Программируемый логический контроллер

PROMPOWER

серии PMP3xx

Руководство по эксплуатации

(Среда разработки Codesys V3)



PROM POWER

Введение

Общая информация

Благодарим вас за покупку контроллера PROMPOWER серии PMP3xx с поддержкой стандарта PLCopen.

В настоящем руководстве представлены инструкции по программированию контроллера серии PMP3xx с использованием среду разработки Codesys V3.

Перед началом использования внимательно прочтите настоящее руководство и запрограммируйте изделие, полностью поняв содержащуюся в руководстве информацию.

Пожалуйста, передайте это руководство конечному пользователю.

Уведомление пользователей

К работе с оборудованием допускаются только квалифицированные сотрудники. Если возникнут какие-либо проблемы, свяжитесь с нашим техническим отделом. Приведенные примеры используются для того, чтобы помочь пользователям лучше понять процесс работы с оборудованием. При подключении ПЛК к другому оборудованию, убедитесь, что технические характеристики и принципы работы ПЛК соответствуют требованиям. При использовании ПЛК соблюдайте меры безопасности. Ошибки оператора могут привести к повреждению оборудования.

Исключение ответственности

Содержание руководства было тщательно проверено, однако потенциальные ошибки при работе не могут быть исключены. В связи с этим мы не гарантирует полного соответствия.

Мы регулярно проверяем руководство и исправляем возникающие проблемы в последующих версиях. Мы всегда рады обратной связи.

Авторские права PROMPOWER

Не копируйте руководство или информацию из него без письменного разрешения PROMPOWER. Нарушители понесут ответственность за убытки. Пожалуйста, сохраняйте все авторские права нашей компании, включая практические модули, разработанные патенты и авторские права.

Содержание

1. Основные команды 20
1-1. Битовые логические команды 20
1-1-1. Основные битовые логические команды 20
1-1-1-1. Битовая операция AND 20
1-1-1-2. Булева операция AND 22
1-1-1-3. Битовая операция OR 23
1-1-1-4. Булева операция OR 24
1-1-1-5. Битовая операция NOT 25
1-1-1-6. Булева операция NOT 26
1-1-1-7. Битовая операция XOR 27
1-1-1-8. Булева операция XOR 28
1-1-2. Битовая логическая команда 29
1-1-2-1. Вывод при достижении нарастающего фронта PLS 29
1-1-2-2. Вывод при достижении спадающего фронта PLF 31
1-1-2-3. Переменный выход ALT 32
1-1-2-4. Вывод битовых данных BOUT 34
1-1-2-5. Задание бита BSET 35
1-1-2-6. Сброс бита BRST 37
1-1-2-7. Логическая операция AND/логические операции OR/XOR/XNOR AryAnd/AryOr/AryXor/AryXorN 38
1-1-3. Триггеры установки приоритета и сброса приоритета 42
1-1-3-1. Триггер установки приоритета SR 43
1-1-3-2. Триггер сброса приоритета RS 45
1-1-4. Команда обнаружения фронта 46
1-1-4-1. Обнаружение нарастающего фронта R_TRIG 47
1-1-4-2. Обнаружение спадающего фронта F_TRIG 48

1-2. Таймер 49
1-2-1. Импульсный таймер TP 49
1-2-2. Таймер задержки при включении питания TON 51
1-2-3. Таймер задержки при выключении питания TOF 53
1-2-4. Часы реального времени 55
1-2-5. Таймер накопления 57
1-2-6. Таймер 100 мс 59
1-3. Счетчики 62
1-3-1. Счетчик прямого действия CTU 62
1-3-2. Счетчик обратного действия CTD 64
1-3-3. Счетчик прямого/обратного действия CTUD 65
1-3-4. Вычитающий счетчик CTD_XXX 68
1-3-5. Счетчик прямого/обратного действия CTUD_XXX 71
1-3-6. Счетчик прямого действия CTU_XXX 76
1-4. Команды обработки данных 80
1-4-1. Команды выбора 80
1-4-1-1. Команда двоичного выбора SEL 80
1-4-1-2. Выбор максимального значения MAX 82
1-4-1-3. Выбор минимального значения MIN 83
1-4-1-4. Предельное значение LIMIT 84
1-4-1-5. Множественный выбор MUX 86
1-4-2. Команды сравнения 87
1-4-2-1. Основные команды сравнения EQ/NE/GT/GE/LT/LE 87
1-4-2-2. Получение максимального/минимального значения массивов AryMax/AryMin 90
1-4-2-3. Получение массива ArySearch 93
1-4-2-4. Пакетное сравнение массивов EQ AryCmpEQ 97

1-4-2-5. Сравнение элементов массива EQV AryCmpEQV 99
1-4-2-6. Пакетное сравнение массивов GE AryCmpGE 102
1-4-2-7. Сравнение элементов массива GEV AryCmpGEV 104
1-4-2-8. Пакетное сравнение массивов GT AryCmpGT 106
1-4-2-9. Сравнение элементов массива GTV AryCmpGTV 109
1-4-2-10. Пакетное сравнение массивов LE AryCmpLE 111
1-4-2-11. Сравнение элементов массива LEV AryCmpLEV 113
1-4-2-12. Пакетное сравнение массивов LT AryCmpLT 115
1-4-2-13. Сравнение элементов массива LTV AryCmpLTV 117
1-4-2-14. Пакетное сравнение массивов NE AryCmpNE 119
1-4-2-15. Сравнение элементов массива NEV AryCmpNEV 122
1-4-2-16. Сравнение таблиц TableCmp 124
1-4-2-17. Сравнение зоны ZoneCmp 128
1-4-3. Команды сдвига 130
1-4-3-1. Сдвиг влево SHL 130
1-4-3-2. Сдвиг вправо SHR 132
1-4-3-3. Поворот влево ROL 134
1-4-3-4. Поворот вправо ROR 135
1-4-3-5. Сдвиговый регистр AryShiftReg 137
1-4-3-6. Регистр сдвига влево-вправо AryShiftRegLR 140
1-4-3-7. Циклический сдвиг влево/вправо с переносом RCL/RCR 143
1-4-3-8. Сдвиг N элементов массива влево/вправо ArySHL/ArySHR 147
1-4-3-9. Сдвиг N элементов массива влево/вправо NSHLC/NSHRC 150
1-4-3-10. Копирование битовых данных влево/вправо SFTL/SFTR 152
1-4-3-11. Копирование данных слова влево/вправо WSFL/WSFR 161
1-4-3-12. Чтение/запись данных в порядке поступления SFRD/SFWR 163
1-4-4. Команды перемещения 168

1-4-4-1. Групповая пересылка данных BMOV 168
1-4-4-2. Определение состояния бита данных BON 170
1-4-4-3. Объединение данных в единице байта BTOW 171
1-4-4-4. Передача одного бита в несколько битов данных FMOV 173
1-4-4-5. Бит включения SUM 175
1-4-4-6. Передача массива AryMove 176
1-4-4-7. Передача цифр MoveDigit 179
1-4-4-8. Передача нескольких битов TransBits 181
1-4-4-9. Задание блока SetBlock 183
1-4-4-10. Дешифратор битов Decoder 185
1-4-4-11. Шифратор битов Encoder 187
1-4-4-12. Число битов BitCnt 189
1-4-4-13. Перестановка старших и младших байтов SWAP 191
1-4-4-14. Обмен данными XCH 192
1-4-4-15. Обмен данными массива AryExchange 193
1-4-4-16. Обмен данными Exchange 196
1-4-5. Команды разделения и объединения данных 198
1-4-5-1. Разделение данных на единицы байта Dispart8Bit 198
1-4-5-2. Разделение на 4 бита DispartDigit 200
1-4-5-3. Разложение действительных чисел на мантиссы и порядки DispartReal 202
1-4-5-4. Группа объединения данных в единицах по 1 байту Unite8Bit_XXX 204
1-4-5-5. Группа объединения 4-битных данных UniteDigit_XXX 206
1-4-5-6. Объединение мантиссы и порядков в действительные числа UniteReal 208
1-4-5-7. Разделение данных на единицы байта WTOB 210
1-4-6. Команды считывания и записи данных 212

1-4-6-1. Запись N битов в группу WriteNbit_XXX 212
1-4-6-2. Группа считывания N битов ReadNbit_XXX 214
1-4-7. Команды преобразования данных 216
1-4-7-1. Преобразование кода BCD и целочисленных данных 217
1-4-7-2. Данные преобразования булева типа BOOL_TO_<TYPE> 220
1-4-7-3. Данные преобразования байтового типа BYTE_TO_<TYPE> 222
1-4-7-4. Команда преобразования целочисленных данных < Integer data > _TO_<TYPE> 224
1-4-7-5. Команда преобразования действительных чисел REAL_TO_<TYPE> 226
1-4-7-6. Команда преобразования временных данных TIME_TO_<TYPE> 228
1-4-7-7. Команда преобразования данных о датах DATE_TO_<TYPE> ... 229
1-4-7-8. Команда преобразования данных о датах и времени DT_TO_<TYPE> 231
1-4-7-9. Команда преобразования временных данных TOD_TO_<TYPE> 232
1-4-7-10. Команда преобразования символьных данных STRING_TO_<TYPE> 233
1-4-7-11. Округление TRUNC 235
1-4-7-12. Преобразование массива в строку AryToString 236
1-4-7-13. Преобразование данных о датах в строку DateToString 238
1-4-7-14. Преобразование данных о датах и времени в строку DtToString 239
1-4-7-15. Преобразование шестнадцатеричных строк в целые числа HexStringToNum_XXX 241
1-4-7-16. Преобразование целых чисел в десятичные/шестнадцатеричные строки фиксированной длины NumToDecString/NumToHexString 242
1-4-7-17. Преобразование строки в массив StringToAry 246
1-4-7-18. Преобразование времени в строку TodToString 248

1-4-7-19. Преобразование из массива байтов AryByteTo 250
1-4-7-20. Пакетное преобразование массивов в BCD AryToBCD 255
1-4-7-21. Пакетное преобразование массивов в битовые строки AryToBin 258
1-4-7-22. Преобразование битовой строки в код Грея BinToGray_XXX 260
1-4-7-23. Преобразование битовой последовательности/битовой строки в массив ColmToLine_XXX 262
1-4-7-24. Режим копирования битов CopyByteToNum/CopyWordToNum/CopyDwordToNum/ CopyLwordToNum 265
1-4-7-25. Режим копирования битов CopyDwordToReal/CopyLwordToLreal 267
1-4-7-26. Режим копирования битов CopyRealToDWord/CopyLrealToLword 268
1-4-7-27. Режим копирования битов CopyNumToXXX 270
1-4-7-28. Режим копирования битов CopyNumToReal/CopyNumToLreal ... 272
1-4-7-29. Режим копирования битов CopyRealToNum/CopyLrealToNum ... 274
1-4-7-30. Преобразование кода Грея в битовую строку GrayToBin_XXX .. 275
1-4-7-31. Преобразование кода Грея в угол GRAY 277
1-4-7-32. Преобразование чисел с фиксированной десятичной точкой в десятичные строки FixNumToString 282
1-4-7-33. Преобразование строки в число с фиксированной десятичной точкой StringToFixNum 284
1-4-7-34. Преобразование строк/столбцов LineToColm 288
1-4-7-35. Преобразование в массив байтов ToAryByte 290
1-4-8. Команды инициализации 294
1-4-8-1. Инициализация Clear 294
1-4-9. Команды регулировки зоны 296
1-4-9-1. Команда регулировки зоны нечувствительности BZAND_TAB 296
1-4-9-2. Вычисление среднего значения данных MEAN_TAB 299
1-4-9-3. Команда наклона RAMP_TAB 300

1-4-9-4. Команда получения табличных координат SCL_TAB 304
1-4-9-5. Команда сортировки табличных данных SORT_TAB 306
1-4-9-6. Команда вычисления суммарного количества данных WSUM_TAB 309
1-4-9-7. Команда контроля зоны ZONE_TAB 311
1-4-9-8. Команда пакетного сброса данных ZRST_TAB 312
1-5. Рабочие команды 314
1-5-1. Команда присвоения 314
1-5-1-1. Команда присвоения MOVE 314
1-5-2. Команды арифметических операций 315
1-5-2-1. Операция сложения ADD 315
1-5-2-2. Операция вычитания SUB 317
1-5-2-3. Операция умножения MUL 319
1-5-2-4. Операция деления DIV 320
1-5-2-5. Операция вычисления остатка MOD 322
1-5-3. Команды математических операций 323
1-5-3-1. Абсолютное значение ABS 323
1-5-3-2. Квадратный корень SQRT 324
1-5-3-3. Экспонента EXP 326
1-5-3-4. Натуральный логарифм LN 327
1-5-3-5. Логарифм с основанием 10 LOG 328
1-5-3-6. Синусоидальная функция SIN 329
1-5-3-7. Косинусоидная функция COS 331
1-5-3-8. Функция арккосинуса ACOS 332
1-5-3-9. Функция арксинуса ASIN 333
1-5-3-10. Функция тангенса TAN 334
1-5-3-11. Функция арктангенса ATAN 335

1-5-3-12. Преобразование углов с плавающей точкой в радианы RAD/DEG 336
1-5-3-13. Прибавление к элементу массива AryAddV 338
1-5-3-14. Вычисление среднего значения элементов массива AryMean 341
1-5-3-15. Стандартное отклонение элементов массива ArySD 343
1-5-3-16. Вычитание значений из элементов массива ArySubV 345
1-5-3-17. Проверка действительных чисел CheckReal 347
1-5-3-18. Увеличение/уменьшение Inc/Dec 349
1-5-3-19. Фактический остаток ModReal 351
1-5-3-20. Длинный фактический остаток ModReal_LR 353
1-5-3-21. Скользящее среднее значение MovingAverage 356
1-5-3-22. Перенос знака действительного числа RoundUp 359
1-5-3-23. Генерирование случайных чисел RAND 361
1-5-4. Команды для работы с адресами 363
1-5-4-1. Размер типа данных SIZEOF 363
1-5-4-2. Оператор взятия адреса ADR 364
1-5-4-3. Оператор битового адреса BITADR 367
1-6. Строковые команды 369
1-6-1. Вычисление суммы StringSum 369
1-6-2. Очистка строки Clearstring 371
1-6-3. Вычисление количества байтов в строке GetByteLen 372
1-6-4. Удаление разделителя после считывания строки действительных чисел SubDelimiter 374
1-6-5. Удаление разделителя после считывания длинной строки действительных чисел SubDelimiter_LR 379
1-6-6. Преобразование букв строки ToUCase/ToLCase 385
1-6-7. Обрезка строки слева/обрезка строки справа TrimL/TrimR 387
2. Специальные команды 390

2-1. Команды высокоскоростного счетчика 390
2-1-1. Обзор функции 390
2-1-2. Функциональный блок 390
2-1-2-1. Включение высокоскоростного счетчика [PP_Counter_Enable] ... 390
2-1-2-2. Постоянный выход сравнения [PP_Counter_Compare] 394
2-1-2-3. Запись заданного значения [PP_Counter_PresetValue] 397
2-1-2-4. Контактный датчик [PP_TouchProbe] 400
2-1-2-5. Считывание измеренного значения ширины импульса счетчика [PP_MeasurePulseWidth] 403
2-1-2-6. Выборка счетчика [PP_Counter_Sample] 405
2-1-2-7. Многосегментное сравнение [PP_Counter_CompareArray] 406
2-1-2-8. Кольцевой счет [PP_Counter_SetRing] 409
2-1-2-9. Сброс порта постоянного выхода сравнения [PP_ResetCmpOutput] 411
2-1-2-10. Сброс ошибок [PP_Counter_Reset] 412
2-1-3. Настройки параметров 414
2-2. Команды высокоскоростного счетчика контроллеров серии PMP3xx 415
2-2-1. Обзор функции 415
2-2-2. Функциональный блок 415
2-2-2-1. Высокоскоростной счетчик [PP_Counter] 415
2-2-2-2. Считывание показаний высокоскоростного счетчика [PP_CounterGetValue] 416
2-2-2-3. Запись показаний высокоскоростного счетчика [PP_CounterSetValue] 418
2-2-3. Конфигурация параметров 421
2-2-4. Пример использования 422
2-3. Команды внешнего прерывания и постоянного прерывания при сравнении 423
2-3-1. Обзор функции 423

2-3-2. Функциональный блок 423
2-3-2-1. Запуск внешних прерываний и постоянных прерываний при сравнении [PP_EnableInterrupt] 423
2-3-2-2. Запись параметров прерывания [PP_WriteInterruptParameter] 427
2-3-3. Конфигурация параметров 429
2-3-4. Пример использования 430
2-4. Команды ПИД-регулирования 431
2-4-1. Формат команд 431
2-4-2. Связанные переменные 432
2-4-3. Описание функции 435
2-4-4. Пример использования 438
2-5. Библиотека системы 445
2-5-1. Обзор функции 445
2-5-2. Введение в функциональные блоки 445
2-5-2-1. Основная частота ЦП [PP_GetCPUFrequency] 445
2-5-2-2. Температура ЦП [PP_GetCPUTemperature] 446
2-5-2-3. Загрузка ЦП [PP_GetCPUUsage] 446
2-5-2-4. Объем памяти [PP_GetMemSize] 447
2-5-2-5. Загрузка памяти [PP_GetMemUsage] 448
2-5-2-6. Время пребывания во включенном состоянии [PP_GetBootTime] 449
2-5-2-7. Название ПЛК [PP_GetPLCName] 450
2-5-2-8. Идентификатор ПЛК [PP_GetPLCID] 451
2-5-2-9. Версия прошивки [PP_GetPLCVersion] 452
2-5-2-10. Получение информации о сетевом порте [PP_GetNetInfo] 453
2-5-2-11. Настройка информации о сетевом порте [PP_SetNetInfo] 456
2-5-2-12. Исполняемая версия [PP_GetRuntimeVersion] 457

2-5-2-13. Версия BOOT [PP_GetBootVersion] 458
2-5-2-14. Получение данных о дате и времени [PP_GetTime] 459
2-5-2-15. Настройка даты и времени [PP_SetTime] 460
2-5-3. Конфигурация параметров 463
2-6. ECAT_FROMTO 464
2-6-1. Обзор функции 464
2-6-2. Введение в функциональные блоки 464
2-6-2-1. Считывание данных на дистанционном входе/выходе [PP_ECATFromTo.PP_EC_FROM] 464
2-6-2-2. Запись данных на дистанционном входе/выходе [PP_ECATFromTo.PP_EC_TO] 466
2-6-3. Конфигурация параметров 468
2-6-4. Применение 469
2-7. Дата и время 470
2-7-1. Получение данных о текущем системном времени GetSystemDate 470
2-7-2. Получение данных о времени выполнения программы в системе GetSystemTime 471
2-7-3. Настройка текущего системного времени SetSystemDate 473
2-7-4. Добавление даты и времени ADD_DT_TIME 475
2-7-5. Сложение значений времени ADD_TOD_TIME 476
2-7-6. Проверка високосного года ChkLeapYear 478
2-7-7. Объединение даты и времени DateStructToDt 479
2-7-8. Преобразование даты в секунды DateToSec 481
2-7-9. Преобразование дней в месяцы DaysToMonth 483
2-7-10. Деление времени DIVTIME 484
2-7-11. Разложение даты и времени DtToDateStruct 487
2-7-12. Преобразование даты и времени в секунды DtToSec 489
2-7-13. Получение информации о дне недели GetDayOfWeek 490

2-7-14. Получение информации о количестве дней в указанном месяце GetDaysOfMonth 492
2-7-15. Получение информации о структуре текущего системного времени GetSystemDate_sDt 494
2-7-16. Получение информации о неделе года GetWeekOfYear 496
2-7-17. Умножение времени MULTIME 498
2-7-18. Преобразование наносекунд во время NanoSecToTime 500
2-7-19. Преобразование секунд в дату SecToDate 501
2-7-20. Преобразование секунд в дату и время SecToDt 503
2-7-21. Преобразование секунд во время SecToTime 504
2-7-22. Преобразование секунд во время SecToTod 506
2-7-23. Вычитание даты SUB_DATE_DATE 507
2-7-24. Вычитание даты и времени SUB_DT_DT 509
2-7-25. Вычитание времени из даты и времени SUB_DT_TIME 511
2-7-26. Вычитание времени из времени SUB_TOD_TIME 513
2-7-27. Вычитание времени SUB_TOD_TOD 514
2-7-28. Преобразование времени в наносекунды TimeToNanoSec 516
2-7-29. Преобразование времени в секунды TimeToSec 517
2-7-30. Преобразование времени в секунды TodToSec 518
2-7-31. Округление даты и времени TruncDt 520
2-7-32. Округление времени TruncTime 522
2-7-33. Округление времени TruncTod 523
2-8. Команды связи 525
2-8-1. Связь по протоколу TCP 525
2-8-1-1. Прием данных при связи по протоколу TCP TCP_Receive 525
2-8-1-2. Передача данных при связи по протоколу TCP TCP_Send 527
2-8-1-3. Создание TCP-соединения TCP_Connect 529

2-8-1-4. Создание серверной коммуникационной службы для TCP соединений TCP_Server 531
2-8-1-5. Создание клиентской коммуникационной службы для TCP соединений TCP_Client 532
2-8-1-6. Пример команды для установления связи по протоколу TCP 534
2-8-2. Связь по протоколу UDP 542
2-8-2-1. Создание соединения для связи по протоколу UDP UDP_Peer 542
2-8-2-2. Прием данных при связи по протоколу UDP UDP_Receive 544
2-8-2-3. Передача данных при связи по протоколу UDP_Send 546
2-8-2-4. Пример связи по протоколу UDP 547
2-8-3. Коды ошибок связи по протоколу TCP/UDP 551
2-9. Обработка файлов 553
2-9-1. Создание каталога DirCreate 553
2-9-2. Удаление каталога DirRemove 555
2-9-3. Открытие файла FileOpen 558
2-9-4. Закрытие файла FileClose 562
2-9-5. Поиск файла FileSeek 565
2-9-6. Запись в файл FileWrite 569
2-9-7. Считывание данных из файла FileRead 573
2-9-8. Запись строки в файл FilePuts 577
2-9-9. Считывание строки из файла FileGets 580
2-9-10. Запись переменной в файл FileWriteVar 584
2-9-11. Считывание переменной из файла FileReadVar 587
2-9-12. Копирование файла FileCopy 590
2-9-13. Удаление файла FileRemove 594
2-9-14. Переименование файла FileRename 596
3. Команды перемещения 600

3-1. Одна ось 600
3-1-1. Обзор команд для одной оси 600
3-1-2. Команды для одной оси 602
3-1-2-1. Включение оси [MC_Power] 602
3-1-2-2. Сброс ошибок оси [MC_Reset] 605
3-1-2-3. Останов перемещения по командам контроллера [MC_Stop] 607
3-1-2-4. Приостановка перемещения [MC_Halt] 609
3-1-2-5. Возврат оси в исходное положение [MC_Home] 611
3-1-2-6. Запуск в толчковом режиме [MC_Jog] 616
3-1-2-7. Абсолютное положение [MC_MoveAbsolute] 619
3-1-2-8. Наложение положения [MC_MoveAdditive] 623
3-1-2-9. Относительное положение [MC_MoveRelative] 628
3-1-2-10. Наложение на команду относительного перемещения [MC_MoveSuperImposed] 631
3-1-2-11. Управление скоростью [MC_MoveVelocity] 636
3-1-2-12. Положение профиля [MC_PositionProfile] 640
3-1-2-13. Профиль скорости [MC_VelocityProfile] 645
3-1-2-14. Профиль ускорения [MC_AccelerationProfile] 650
3-1-2-15. Считывание данных о фактическом положении [MC_ReadActualPosition] 653
3-1-2-16. Считывание данных о текущем крутящем моменте [MC_ReadActualTorque] 655
3-1-2-17. Считывание данных о текущей скорости [MC_ReadActualVelocity] 657
3-1-2-18. Считывание состояния ошибки оси [MC_ReadAxisError] 658
3-1-2-19. Считывание битового параметра оси [MC_ReadBoolParameter] 661
3-1-2-20. Считывание параметра оси [MC_ReadParameter] 662
3-1-2-21. Считывание заданного положения оси [SMC_ReadSetPosition] 664

3-1-2-22. Считывание ошибок функционального блока [SMC_ReadFBError] 666
3-1-2-23. Установка битового параметра оси [MC_WriteBoolParameter] .. 668
3-1-2-24. Установка параметра оси [MC_WriteParameter] 670
3-1-2-25. Сброс ошибок [SMC_ClearFBError] 671
3-1-2-26. Считывание информации об ошибках [SMC_ErrorString] 672
3-1-2-27. Повторный вызов привода/оси [SMC3_ReinitDrive] 673
3-1-2-28. Настройка параметров COE для оси [SMC3_ETC_WriteParameter_CoE] 675
3-1-2-29. Контактный датчик [MC_TouchProbe] 678
3-1-3. Применение функции одной оси 684
3-2. Групповые функции оси 685
3-2-1. Групповые команды оси 685
3-2-2. Групповые команды оси 686
3-2-2-1. Добавление оси в группу осей [MC_AddAxisToGroup] 686
3-2-2-2. Удаление оси из группы осей [MC_RemoveAxisFromGroup] 688
3-2-2-3. Удаление всех осей из группы осей [MC_UngroupAllAxes] 690
3-2-2-4. Включение группы осей [MC_GroupEnable] 692
3-2-2-5. Выключение группы осей [MC_GroupDisable] 694
3-2-2-6. Сброс группы осей [MC_GroupReset] 695
3-2-2-7. Настройка положения группы осей [MC_GroupSetPosition] 698
3-2-2-8. Преобразование системы координат [MC_SetCoordinateTransform] 700
3-2-2-9. Преобразование динамической системы координат [MC_SetDynCoordTransform] 702
3-2-2-10. Продолжение работы группы осей [MC_GroupContinue] 705
3-2-2-11. Остановка работы группы осей [MC_GroupHalt] 706
3-2-2-12. Прерывание работы группы осей [MC_GroupInterrupt] 709

3-2-2-13. Останов группы осей [MC_GroupStop] 711

3-2-2-14. Кинематическое преобразование системы координат [MC_SetKinTransform] 714

3-2-2-15. Настройка перебега группы осей [MC_GroupSetOverride] 715

3-2-2-16. Дуговая интерполяция в режиме абсолютного позиционирования [MC_MoveCircularAbsolute] 721

3-2-2-17. Дуговая интерполяция в режиме относительного позиционирования [MC_MoveCircularRelative] 729

3-2-2-18. Быстрое перемещение в абсолютное положение [MC_MoveDirectAbsolute] 737

3-2-2-19. Быстрое перемещение в относительное положение [MC_MoveDirectRelative] 741

3-2-2-20. Линейная интерполяция в режиме абсолютного позиционирования [MC_MoveLinearAbsolute] 746

3-2-2-21. Линейная интерполяция в режиме относительного позиционирования [MC_MoveLinearRelative] 753

3-2-2-22. Считывание данных обратной связи по положению группы осей [MC_GroupReadActualPosition] 759

3-2-2-23. Считывание данных обратной связи по скорости группы осей [MC_GroupReadActualVelocity] 761

3-2-2-24. Считывание параметров конфигурации группы осей [MC_GroupReadConfiguration] 764

3-2-2-25. Считывание информации об ошибках для группы осей [MC_GroupReadError] 766

3-2-2-26. Считывание текущего рабочего состояния группы осей [MC_GroupReadStatus] 767

3-2-2-27. Запуск группы осей [SMC_StartupAxisGroup] 771

3-2-2-28. Включение группы осей [SMC_GroupPower] 772

3-2-2-29. Настройка перерыва в перемещении [SMC_GroupInterruptAt] .. 774

3-2-2-30. Перезапуск после сброса ошибок [SMC_GroupEnableResumeAfterError] 776

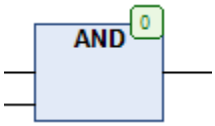
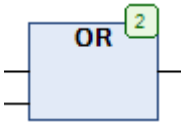
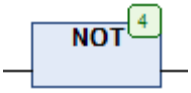
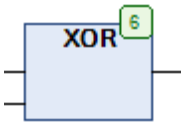
3-2-2-31. Запуск группы осей в толчковом режиме [SMC_GroupJog] 777

3-2-2-32. Задержка в работе группы осей [SMC_GroupWait] 782
3-2-3. Применение функции для группы осей 785
3-3. Функции для кулачков 786
3-3-1. Перечень команд для кулачков 786
3-3-2. Команды для кулачков 787
3-3-2-1. Выбор таблицы кулачков [MC_CamTableSelect] 787
3-3-2-2. Связь через кулачок [MC_CamIn] 790
3-3-2-3. Разблокировка связи через кулачок [MC_CamOut] 794
3-3-2-4. Связь через кулачок [MC_GearIn] 796
3-3-2-5. Работа редуктора при заданных положениях [MC_GearInPos] ... 799
3-3-2-6. Отключение редуктора [MC_GearOut] 805
3-3-2-7. Сдвиг фаз между главной осью и ведомой осью [MC_Phasing] .. 806
3-3-2-8. Диапазон характеристик кулачков [SMC_CAMBounds] 809
3-3-2-9. Диапазон положений при связи с кулачком [SMC_CAMBounds_Pos] 815
3-3-2-10. Визуализация таблицы кулачков [SMC_CamEditor] 817
3-3-2-11. Управление толкателями кулачков [SMC_CamRegister] 819
3-3-2-12. Считывание положения ведомой оси из таблицы кулачков [SMC_GetCamSlaveSetPosition] 823
3-3-2-13. Считывание выходного значения для толкателей [SMC_GetTappetValue] 826
3-3-2-14. Считывание данных из таблицы кулачков [SMC_ReadCAM] 828
3-3-2-15. Внесение изменений в таблицу кулачков [SMC_WriteCAM] 830
3-3-3. Применение функции для кулачков 832
Приложение 834

1. Основные команды

1-1. Битовые логические команды

1-1-1. Основные битовые логические команды

Команда	Пиктограммы команд	Функция
AND		Логический оператор «И»
OR		Логический оператор «ИЛИ»
NOT		Логический оператор «НЕ»
XOR		Логический оператор «исключающее ИЛИ»

Основные битовые логические команды: AND, OR, NOT и XOR. В программном обеспечении Codesys функции подразделяются на битовые логические операции и булевы логические операции.

- Битовая логическая операция: поочередное выполнение булевой логической операции над соответствующими битами двух целочисленных данных и получение совместимых целочисленных результатов.
- Булева логическая операция: выполнение логической операции над двумя данными булевого типа.

1-1-1-1. Битовая операция AND

Функция: Битовая команда AND (И) сравнивает соответствующие биты двух целых чисел. Если соответствующие биты двух чисел оба равны 1, то соответствующий возвращаемый бит результата равен "1". Если соответствующие биты двух целых чисел оба равны "0", или один из битов равен "0", то соответствующий возвращаемый бит результата равен "0".

Битовая операция AND [AND]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

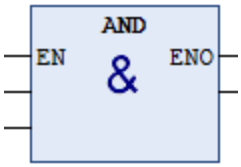
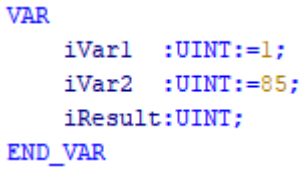
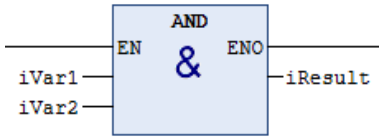
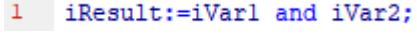
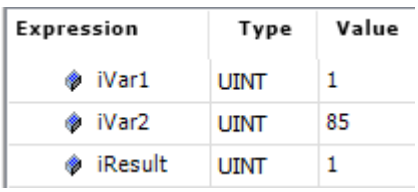
Команда	Название	Графическое представление	ST
AND	Битовая операция AND		AND

Таблица логических отношений AND

Вход 1	Вход 2	Результат
0	0	0
0	1	0
1	0	0
1	1	1

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

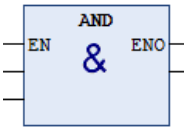
Анализ результатов	Десятичное число 1 соответствует двоичному числу 0000 0001, а десятичное число 85 соответствует двоичному числу 0101 0101. Согласно определению битовой операции AND (И), каждый независимый бит поочередно подвергается операции AND, в результате чего получается конечный результат 0000 0001, который представляет собой десятичное значение 1.

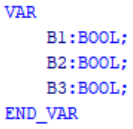
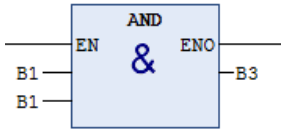
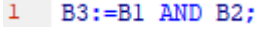
1-1-1-2. Булева операция AND

Функция: Булева операция AND (И) используется для вычисления результатов AND двух булевых выражений. Если результат двух булевых выражений истинен, он возвращается как истина (true). Если одно из них ложно, результат возвращается как ложь (false).

Пример: Создайте POU и используйте булеву операцию AND для определения возвращаемого значения операции. Полученный код:

Булева операция AND [AND]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
AND	Булева операция AND		AND

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Анализ результатов	Если оба значения B1 и B2 истинны, результатом выполнения программы будет истинное значение B3.	

1-1-1-3. Битовая операция OR

Функция: Битовая команда OR (ИЛИ) сравнивает соответствующие биты двух целых чисел. Если один из соответствующих битов двух чисел равен "1", или оба бита равны "1", то соответствующий возвращаемый бит результата равен "1". Если соответствующие биты двух целых чисел оба равны "0", то соответствующий возвращаемый бит результата равен "0".

Битовая операция OR [OR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

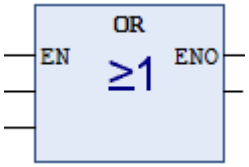
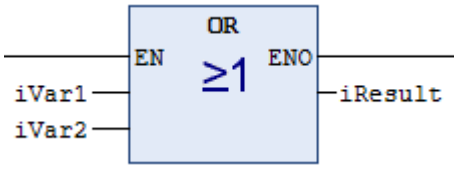
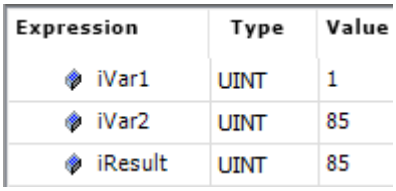
Команда	Название	Графическое представление	ST
OR	Битовая операция OR		OR

Таблица логических отношений OR

Вход 1	Вход 2	Результат
0	0	0
0	1	1
1	0	1
1	1	1

ДЕМО	LD	ST

Задание переменных	VAR
	iVar1 :UINT:=1;
	iVar2 :UINT:=85;
	iResult:UINT;
	END_VAR	

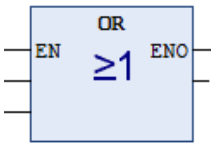
Пример программы		1 iResult:=iVar1 OR iVar2;
Результат операции		
Анализ результатов	Конечным результатом выполнения программы будет значение 85.	

1-1-1-4. Булева операция OR

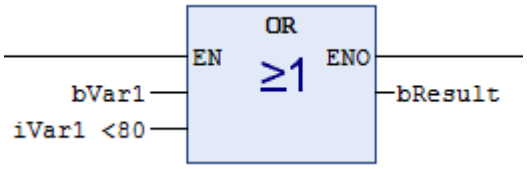
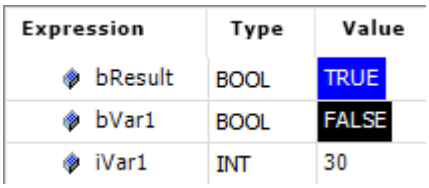
Функция: Булева команда OR (ИЛИ) используется для вычисления результатов OR двух булевых выражений. Если одно из двух булевых выражений истинно, результат возвращается как истина (true). Если результат двух булевых выражений ложен, результат возвращается как ложь (false).

Пример: Создайте POU и используйте булеву операцию OR для определения возвращаемого значения операции. Полученный код:

Булева операция OR [OR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
OR	Булева операция OR		OR

ДЕМО	LD	ST
Задание переменных	VAR bResult:BOOL; bVar1 :BOOL; iVar1 :INT:=30; END_VAR	

Пример программы		1 bResult:=bVar1 OR (iVar1 <80);
Результат операции		
Анализ результатов	Поскольку начальное значение iVar1 равно 30, условие iVar1<80 истинно, а начальное значение bVar1 равно "0", поэтому оно ложно. Логическим результатом одного истинного и одного ложного ИЛИ является истина. Таким образом, условие в правой части уравнения истинно, и результатом выполнения программы является значение bResult, равное true.	

1-1-1-5. Битовая операция NOT

Функция: Отрицание логической строки, изменение текущего значения с "0" на "1" или с "1" на "0". Битовая команда NOT (НЕ) поочередно отрицает переменные или константы.

Битовая операция NOT [NOT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

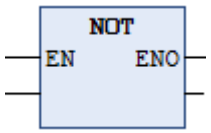
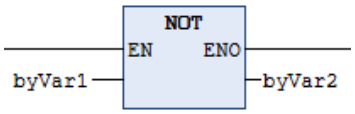
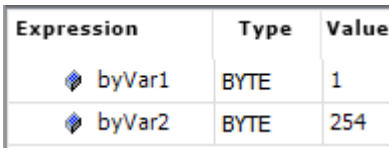
Команда	Название	Графическое представление	ST
NOT	Битовая операция NOT		NOT

Таблица логических отношений NOT

Вход	Результат
0	1
1	0

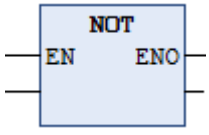
ДЕМО	LD	ST
Задание переменных	VAR byVar1:BYTE:=1; byVar2:BYTE; END_VAR	
Пример программы		1 byVar2:=NOT byVar1;
Результат операции		
Анализ результатов	Поскольку значение byVar1 равно 1, преобразование его в двоичное число дает 0000 0001. После выполнения битового отрицания результат будет 1111 1110. Конечный результат — 254.	

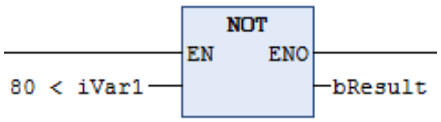
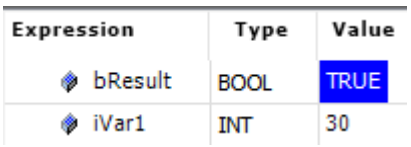
1-1-1-6. Булева операция NOT

Функция: Булева команда NOT (НЕ) используется для вычисления результатов одного булева выражения. Когда входные данные истинны, результат будет ложным. Когда входные данные ложны, результат будет истинным.

Пример: Создайте POU и используйте булеву операцию NOT для определения возвращаемого значения операции. Код будет следующим:

Булева операция NOT [NOT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
NOT	Булева операция NOT		NOT

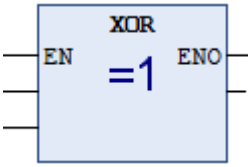
ДЕМО	LD	ST
Задание переменных	VAR bResult:BOOL; iVar1:INT:=30; END_VAR	
Пример программы		1 bResult:=NOT (80 < iVar1);
Результат операции		
Анализ результатов	При 80<30 это утверждение ложно. После использования команды NOT для отрицания булева выражения результат будет истинным, поэтому конечное значение bResult будет true (истинно).	

1-1-1-7. Битовая операция XOR

Функция: Сравнение соответствующих битов двух целых чисел с помощью битовой команды XOR (исключающее ИЛИ). Если один бит двух целых чисел равен "1", а второй бит равен "0", то соответствующий возвращаемый бит результата равен "1". Если соответствующие биты двух целых чисел оба равны "1" или "0", то соответствующий возвращаемый бит результата равен "0".

Пример: Создайте POU, выполните битовую операцию XOR для переменных iVar1 и iVar2 и выведите результаты. Код реализации будет следующим:

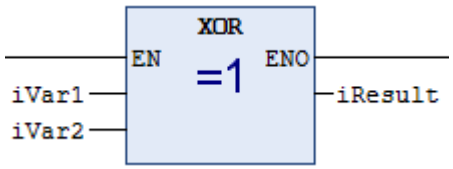
Битовая операция XOR [XOR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
XOR	Битовая операция XOR		XOR

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    iVar1  :INT:=1;
    iVar2  :INT:=85;
    iResult:INT;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  iResult:=iVar1 XOR iVar2;
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
iVar1	INT	1
iVar2	INT	85
iResult	INT	84

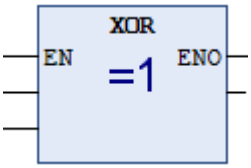
</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Десятичное число 1 соответствует двоичному числу 0000 0001, а десятичное число 85 соответствует двоичному числу 0101 0101. Согласно определению битовой команды XOR (исключающее ИЛИ) результатом будет 84.
Битовая операция XOR. Только когда входное состояние одного контакта равно "1", а входное состояние другого контакта равно "0", выходное состояние будет равно "1". Если оба состояния контактов одновременно равны "1" или "0", выходное состояние будет равно "0".</td></tr>
</table>

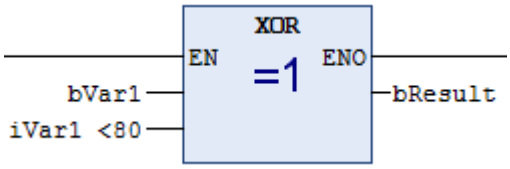
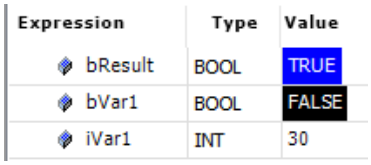
1-1-1-8. Булева операция XOR

Функция: Булева команда XOR (исключающее-ИЛИ) используется для вычисления результатов двух булевых выражений. Только когда одно выражение истинно, а другое ложно, результат, возвращаемый выражением, является истинным. Если оба выражения оцениваются как истинные или ложные, возвращаемый результат будет ложным.

Пример: Создайте POU и используйте булеву команду XOR для определения возвращаемого значения (TRUE или FALSE). Код реализации будет следующим:

Булева операция XOR [XOR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
XOR	Булева операция XOR		XOR

ДЕМО	LD	ST
Задание переменных	VAR bResult:BOOL; bVar1 :BOOL; iVar1 :INT:=30; END_VAR	
Пример программы		1 bResult:=bVar1 XOR (iVar1 <80);
Результат операции		
Анализ результатов	Результатом выполнения программы будет значение TRUE.	

Expression	Type	Value
bResult	BOOL	TRUE
bVar1	BOOL	FALSE
iVar1	INT	30

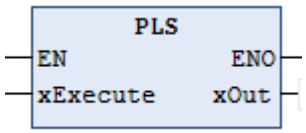
1-1-2. Битовая логическая команда

1-1-2-1. Вывод при достижении нарастающего фронта PLS

(1) Обзор команды

Когда PLS команда xExecute обнаружит нарастающий фронт, xOut примет состояние true, которое перейдет в состояние false через один цикл сканирования.

Вывод при достижении нарастающего фронта [PLS]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
PLS	Вывод при достижении нарастающего фронта		PLS(xExecute:= , xOut=>);

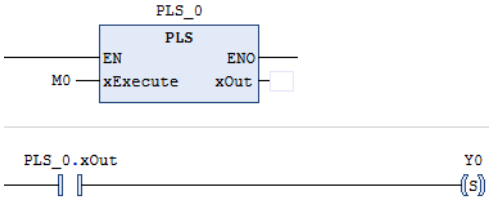
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xExecute	Входной сигнал	BOOL	FALSE,TRUE	FALSE	Определяемое значение BOOL
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Выходной сигнал	BOOL	FALSE,TRUE	FALSE	Выходной результат

(3) Функция

- Вывод команды цикла сканирования при достижении нарастающего фронта.

(4) Пример программы

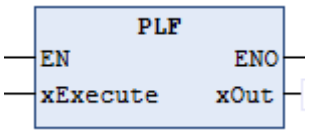
ДЕМО	LD	ST
Задание переменной		VAR PLS_0: PLS; M0: BOOL; Y0: BOOL; END_VAR
Пример программы		PLS_0(xExecute:=M0, xOut=>); IF PLS_0.xOut THEN Y0:=TRUE; END_IF
Результат операции		

1-1-2-2. Вывод при достижении спадающего фронта PLF

(1) Обзор команды

Когда PLF команда xExecute обнаружит спадающий фронт, xOut примет состояние true, которое перейдет в состояние false через один цикл сканирования.

Вывод при достижении спадающего фронта [PLF]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
PLF	Вывод при достижении спадающего фронта		PLF_0(xExecute:=, xOut=>);

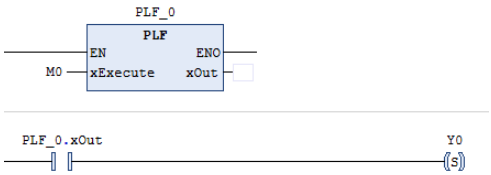
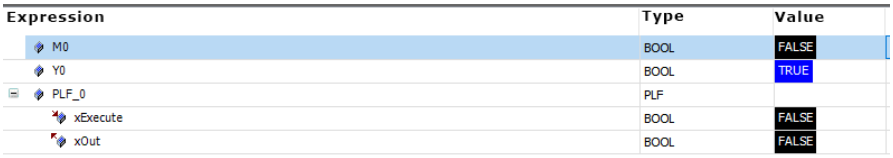
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xExecute	Входной сигнал	BOOL	FALSE,TRUE	FALSE	Определяемое значение BOOL
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
xOut	Выходной сигнал	BOOL	FALSE,TRUE	FALSE	Выходной результат

(3) Функция

- Вывод команды цикла сканирования при достижении спадающего фронта

(4) Пример программы

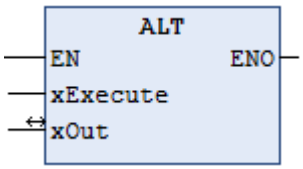
ДЕМО	LD	ST
Задание переменной		VAR PLF_0: PLF; M0: BOOL; Y0: BOOL; END_VAR
Пример программы		PLF_0(xExecute:= M0, xOut=>); IF PLF_0.xOut THEN Y0:=TRUE; END_IF
Результат операции		

1-1-2-3. Переменный выход ALT

(1) Обзор команды

Когда ALT команда xExecute обнаружит нарастающий фронт, xOut изменит свое состояние на противоположное.

Переменный выход [ALT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ALT	Переменный выход		ALT_0(xExecute:= , xOut:=);

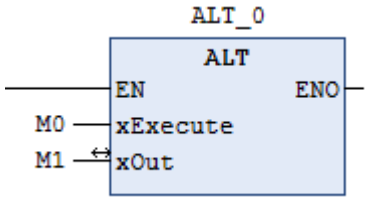
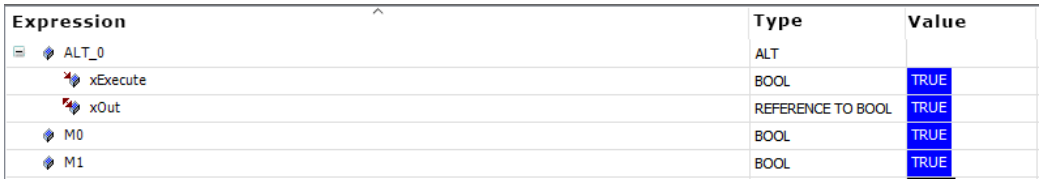
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xExecute	Входной сигнал	BOOL	FALSE,TRUE	FALSE	Определяемое значение BOOL
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
xOut	Выходной сигнал	BOOL	FALSE,TRUE	FALSE	Если обнаружен нарастающий фронт xExecute, и его состояние изменено на противоположное.

(3) Функция

- После обнаружения нарастающего фронта сигнала xExecute состояние выходного сигнала изменится на противоположное; преобразование из исходного состояния FALSE в True, либо из True в FALSE.

(4) Пример программы

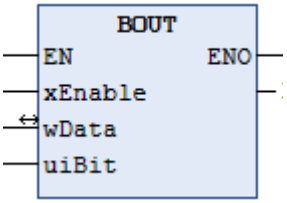
ДЕМО	LD	ST
Задание переменной		VAR M0: BOOL; M1: BOOL; ALT_0: ALT; END_VAR
Пример программы		ALT_0(xExecute:=M0 , xOut:=M1);
Результат операции		

1-1-2-4. Вывод битовых данных BOUT

(1) Обзор команды

Задает состояние бита.

Вывод битовых данных [BOUT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BOUT	Вывод значения бита		BOUT(xEnable:= , wData:= , uiBit:=);

(2) Связанная переменная

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xExecute	Входной сигнал	BOOL	FALSE,TRUE	FALSE	Определяемое значение BOOL
uiBit	Входной сигнал	UINT	В качестве типа данных	0	Задаваемый бит
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
wData	Выходной сигнал	Слово	В качестве типа данных	0	Задаваемое значение

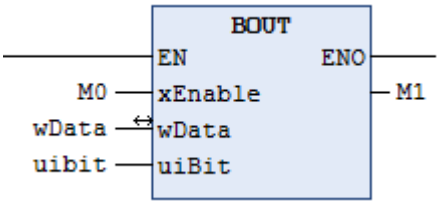
(3) Функция

- Обнаружение нарастающего фронта сигнала xExecute и задание бита, заданного по переменной uiBit, соответствующей данным wData.

(4) Примечание

- WDate — это входные и выходные данные, которые необходимо задать в таблице переменных и преобразовать в двоичный формат. Подтвердите бит преобразования.

(5) Пример программы

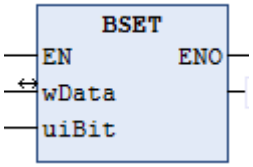
ДЕМО	LD	ST
Задание переменной	VAR M0:BOOL; wData: WORD; uibit: UINT; M1: BOOL; END_VAR	
Пример программы		BOUT(xEnable:= M0, wData:=wData, uiBit:=uibit);
Результат операции		
Анализ результатов	После настройки начальных данных wDate на 22 (двоичный код: 10110) uiBit принимает значение 3, M0 устанавливается на ON, бит 3 данных wDate устанавливается на ON, значение wDate изменяется на 30 (двоичный код: 11110), а ответный сигнал M1 устанавливается на TRUE.	

1-1-2-5. Задание бита BSET

(1) Обзор команды

Задание бита.

Задание бита [BSET]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BSET	Задание бита		BSET(wData:=, uiBit:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
uiBit	Входной сигнал	UINT	В качестве типа данных	0	Задаваемый бит
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
wData	Выходной сигнал	Слово	В качестве типа данных	0	Задаваемое значение

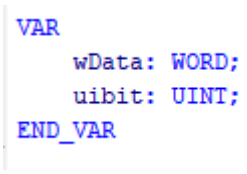
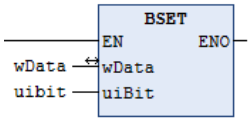
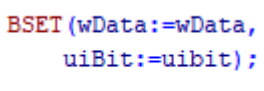
(3) Функция

- Задание бита, заданного по переменной uiBit, соответствующей wData.

(4) Примечание

- WDate — это входные и выходные данные, которые необходимо задать в таблице переменных и преобразовать в двоичный формат. Подтвердите бит преобразования.
- Независимо от того, будет ли позже использована команда BSET или нет, состояние ON всегда будет сохраняться, а команду BRST можно использовать для установки этого бита в состояние OFF.

(5) Пример программы

ДЕМО	LD	ST
Задание переменной		
Пример программы		

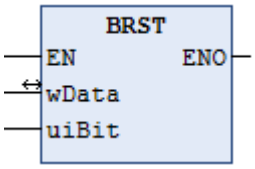
Результат операции	
Анализ результатов	Начальные данные wDate установлены на 22 (двоичный код: 10110), а uiBit принимает значение 3. Бит 3 данных wDate устанавливается на ON, значение wDate изменяется на 30 (двоичный код: 11110).

1-1-2-6. Сброс бита BRST

(1) Обзор команды

Сброс бита.

Сброс бита [BRST]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BRST	Сброс бита		BRST(wData:=, uiBit:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
uiBit	Входной сигнал	UINT	В качестве типа данных	0	Задаваемый бит
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
wData	Выходной сигнал	Слово	В качестве типа данных	0	Задаваемое значение

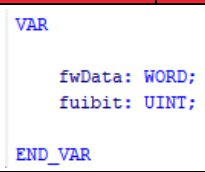
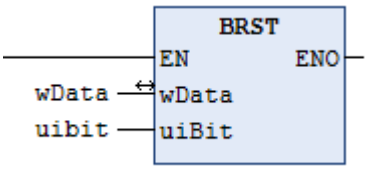
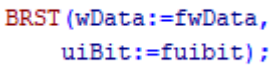
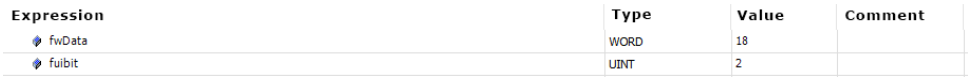
(3) Функция

- Сброс бита, заданного по переменной uiBit, соответствующей wData.

(4) Примечание

- WDate — это входные и выходные данные, которые необходимо задать в таблице переменных и преобразовать в двоичный формат. Подтвердите бит преобразования.

(5) Пример программы

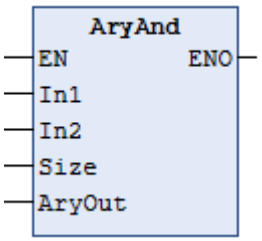
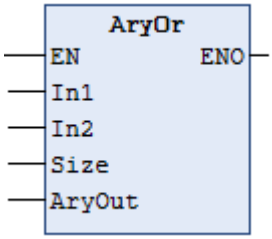
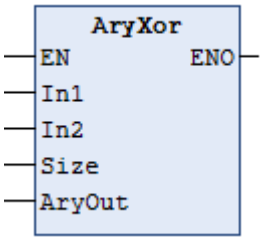
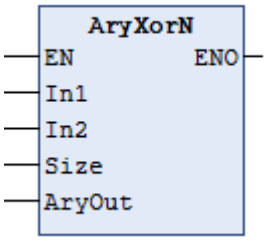
ДЕМО	LD	ST
Задание переменной		
Пример программы		
Результат операции		
Анализ результатов	Начальные данные wDate установлены на 22 (двоичный код: 10110), а uiBit принимает значение 2. Бит 3 данных wDate сбрасывается, значение wDate изменяется на 18 (двоичный код: 10010).	

1-1-2-7. Логическая операция AND/логические операции OR/XOR/XNOR AryAnd/AryOr/AryXor/AryXorN

(1) Обзор команды

Выполнение операций над каждым битом булевой и битовой строки элементов между массивами.

Логическая операция AND/логические операции OR/XOR/XNOR [AryAnd/AryOr/AryXor/AryXorN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryAnd	Логическое И		AryAnd(In1:= , In2:= , Size:= , AryOut:=);
AryOr	Логическое ИЛИ		AryOr(In1:=, In2:= , Size:= , AryOut:=]);
AryXor	Исключающее ИЛИ		AryXor(In1:= , In2:= , Size:= , AryOut:=);
AryXorN	То же самое ИЛИ		AryXorN(In1:=, In2:=, Size:=, AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In1[], In2[]	Массив арифметических объектов	BOOL/BYTE/ WORD/DWORD / LWORD	В качестве типа данных	-	Массив арифметических объектов
Size	Элементы данных	UINT	В качестве типа данных	-	Количество элементов данных объекта операции

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Aryout[]	Массив результатов операции	BOOL/BYTE/ WORD/DWORD /LWORD	В качестве типа данных	-	Массив результатов операции

(3) Функция

- Выполнение логических операций над соответствующими битами каждого из элементов "Size", начиная с In1[] и In2[] в массиве объектов операции. Результат операции сохраняется в соответствующем элементе AryOut[]. Следовательно, типы данных In1[], In2[] и AryOut[] должны быть одинаковыми.
- Входные и выходные связи операции:

AryAnd

Если один из двух элементов имеет значение FALSE, то результатом операции будет FALSE. Только если оба элемента имеют значение TRUE, результатом операции будет TRUE.

Бит элемента In1[]	Бит элемента In1[]	Бит результата элемента AryOut[]
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

AryoR

Если один из двух элементов имеет значение TRUE, то результатом операции будет TRUE. Только если оба элемента имеют значение FALSE, результатом операции будет FALSE.

Бит элемента In1[]	Бит элемента In1[]	Бит результата элемента AryOut[]
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

AryXor

Если два элемента одинаковы, то результатом операции будет FALSE. Если два элемента разные, результатом операции будет TRUE.

Бит элемента In1[]	Бит элемента In1[]	Бит результата элемента AryOut[]
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

AryXorN

Если два элемента одинаковы, то результатом операции будет TRUE. Если два элемента разные, результатом операции будет FALSE.

Бит элемента In1[]	Бит элемента In1[]	Бит результата элемента AryOut[]
FALSE	FALSE	TRUE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

(4) Примечание

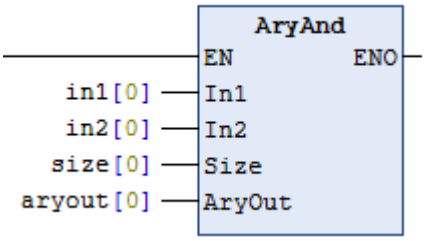
- Типы данных In1[], In2[] и AryOut[] должны быть одинаковыми. Если они разные, функция возвращает значение FALSE, а значение "AryOut" остается неизменным.
- Во избежание исключений ПЛК значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].
- Если значение "Size" равно 0, значение AryOut[] остается неизменным, а функция возвращает значение TRUE.
- Для In1[] и In2[] требуются булевы и битовые строки, в противном случае программа выдаст ошибку во время выполнения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменной</td><td colspan="2">

```
VAR
    in1:ARRAY[0..5] OF BOOL;
    in2:ARRAY[0..5] OF BOOL;
    size:ARRAY[0..3] OF UINT;
    aryout:ARRAY[0..5] OF BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
AryAnd(In1:=in1[0] ,
    In2:=in2[0] ,
    Size:=size[0] ,
    AryOut:=aryout[0] );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value	Comment
in1	ARRAY [0..5] OF BO...		
in1[0]	BOOL	FALSE	
in1[1]	BOOL	FALSE	
in1[2]	BOOL	TRUE	
in1[3]	BOOL	TRUE	
in1[4]	BOOL	FALSE	
in1[5]	BOOL	FALSE	
in2	ARRAY [0..5] OF BO...		
in2[0]	BOOL	FALSE	
in2[1]	BOOL	FALSE	
in2[2]	BOOL	FALSE	
in2[3]	BOOL	TRUE	
in2[4]	BOOL	FALSE	
in2[5]	BOOL	FALSE	
size	ARRAY [0..3] OF UINT		
aryout	ARRAY [0..5] OF BO...		
aryout[0]	BOOL	FALSE	
aryout[1]	BOOL	FALSE	
aryout[2]	BOOL	FALSE	
aryout[3]	BOOL	TRUE	
aryout[4]	BOOL	FALSE	
aryout[5]	BOOL	FALSE	
fwData	WORD	0	

</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Если in1[0] имеет значение FALSE, in2[0] имеет значение FALSE, то результатом aryout[0] будет FALSE.

Если in1[0] имеет значение FALSE, in2[0] имеет значение TRUE, то результатом aryout[0] будет FALSE.

Если in1[0] имеет значение TRUE, in2[0] имеет значение FALSE, то результатом aryout[0] будет FALSE.

Если in1[0] имеет значение TRUE, in2[0] имеет значение TRUE, то результатом aryout[0] будет TRUE.</td></tr>
</table>

1-1-3. Триггеры установки приоритета и сброса приоритета

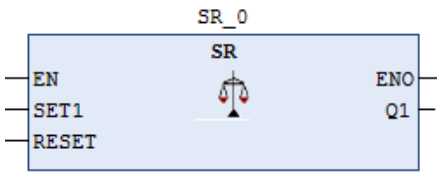
В релейной системе несколько пар контактов реле действуют одновременно. В ПЛК команды выполняются одна за другой, то есть последовательно, без одновременного выполнения. Поэтому команды установки и сброса формата катушки имеют приоритет. Входы установки и сброса SR-триггера и RS-

триггера находятся в одной команде, и установка и сброс выполняются входом, расположенным ниже входа команд. SR-триггер — это триггер «установки приоритета». Если сигнал установки (SET1) и сигнал сброса (RESET) равны 1, триггер окончательно переходит в состояние установки. RS-триггер — это триггер «сброса приоритета». Если сигнал установки (SET) и сигнал сброса (RESET1) равны 1, триггер окончательно переходит в состояние сброса.

1-1-3-1. Триггер установки приоритета SR

Оба входных сигнала установки и сброса действительны. Приоритет входного сигнала установки.

Триггер установки приоритета [SR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	Функция
SR		SR_0(SET1:= , RESET:= , Q1=>);

Связанные переменные

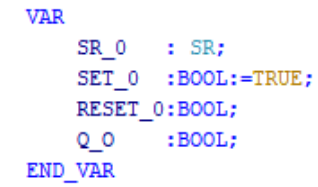
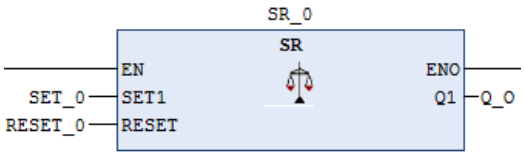
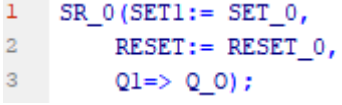
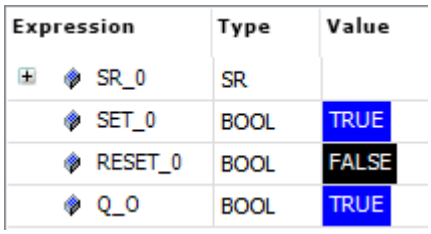
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
SET1	Сигнал установки	BOOL	[FALSE,TRUE]	FALSE	-
RESET	Сигнал сброса	BOOL	[FALSE,TRUE]	FALSE	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q1	Состояние выхода	BOOL	[FALSE,TRUE]	FALSE	Выход установки приоритета

Функция: Задание бистабильного триггера с установленным приоритетом. Логическое отношение: Q1=(NOT RESET AND Q1) OR SET1, где SET1 — сигнал установки, а RESET — сигнал сброса.

Синтаксис: Когда SET1 равен "1", независимо от того, установлен ли RESET на "1" или нет, выход Q1 равен "1". Когда SET1 равен "0", если выход Q1 равен "1", после того, как RESET примет значение "1" выход Q1 немедленно сбросится на "0". Когда выход Q1 равен "0", независимо от того, установлен ли RESET на "1" или "0", выход Q1 остается равным "0".

Таблица состояний SR

SET1	RESET	Выход Q1
0	0	Сохранение исходного состояния
1	0	1
0	1	0
1	1	1

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    SR_0    : SR;
    SET_0   :BOOL:=TRUE;
    RESET_0:BOOL;
    Q_0     :BOOL;
END_VAR
```



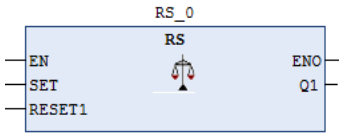
```
1  SR_0(SET1:= SET_0,
2       RESET:= RESET_0,
3       Q1=> Q_0);
```

Expression	Type	Value
SR_0	SR	
SET_0	BOOL	TRUE
RESET_0	BOOL	FALSE
Q_0	BOOL	TRUE

1-1-3-2. Триггер сброса приоритета RS

Оба входных сигнала установки и сброса действительны. Приоритет входного сигнала сброса.

Триггер сброса приоритета [RS]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	Функция
RS		RS_0(SET:= , RESET1:= , Q1=>);

Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
SET1	Сигнал установки	BOOL	[FALSE,TRUE]	FALSE	-
RESET1	Сигнал сброса	BOOL	[FALSE,TRUE]	FALSE	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q1	Состояние выхода	BOOL	[FALSE,TRUE]	FALSE	Выход сброса приоритета

Функция: Сброс бистабильного триггера с приоритетом сброса. Логическое отношение: Q1=NOT RESET1 AND (Q1 OR SET), где SET — сигнал установки, а RESET1 — сигнал сброса.

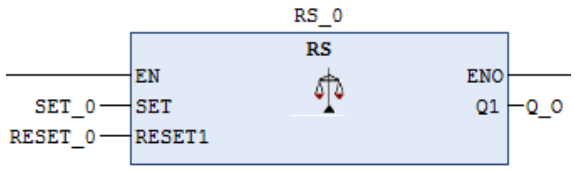
Синтаксис: Когда RESET1 равен "1", независимо от того, установлен ли SET на "1" или нет, выход Q1 равен "0". Когда RESET1 равен "0", если выход Q1 равен "0", после того, как SET примет значение "1" выход Q1 немедленно сбросится на "1". Когда выход Q1 равен "1", независимо от того, установлен ли SET на "1" или "0", выход Q1 остается равным "1".

Таблица состояний RS

SET	RESET1	Выход Q1
0	0	Сохранение исходного состояния
1	0	1
0	1	0
1	1	0

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    RS_0: RS;
    SET_0  :BOOL;
    RESET_0:BOOL:=TRUE;
    Q_0    :BOOL;
END_VAR
```




```
1  RS_0(SET:= SET_0,
2       RESET1:= RESET_0,
3       Q1=> Q_0);
```

Expression	Type	Value
RS_0	RS	
SET_0	BOOL	FALSE
RESET_0	BOOL	TRUE
Q_0	BOOL	FALSE

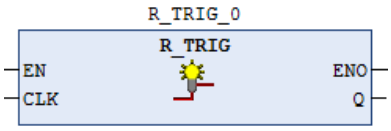
1-1-4. Команда обнаружения фронта

Команда обнаружения фронта используется для обнаружения изменений в нарастающем фронте (сигнал 0-->1) и спадающем фронте (сигнал 1-->0) сигнала BOOL. В каждом цикле сканирования состояние сигнала сравнивается с его состоянием в предыдущем цикле сканирования. Если состояния разные, это указывает на скачкообразное изменение фронта. Поэтому состояние сигнала в предыдущем цикле должно быть сохранено для сравнения с новым состоянием сигнала.

1-1-4-1. Обнаружение нарастающего фронта R_TRIG

Для обнаружения нарастающего фронта.

Обнаружение нарастающего фронта [R_TRIG]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

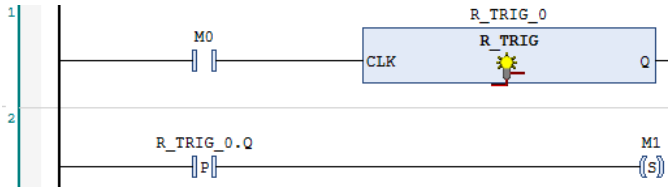
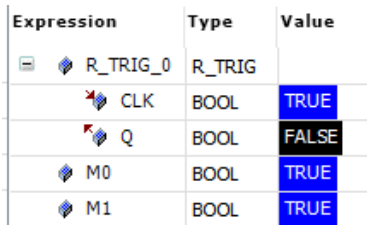
Команда	Пиктограммы команд	Функция
R_TRIG		R_TRIG_0(CLK:= , Q=>);

Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CLK	Входной сигнал	BOOL	[FALSE,TRUE]	FALSE	Действует нарастающий фронт сигнала (от 0 до 1).
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Выходной сигнал	BOOL	[FALSE,TRUE]	FALSE	Обнаружение нарастающего фронта

Функция: Используется для обнаружения нарастающего фронта.

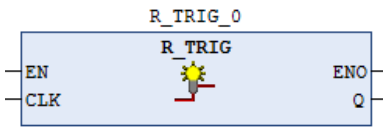
Синтаксис: После изменения CLK с "0" на "1" запускается детектор нарастающего фронта, и выход Q сначала переключается с "1" на "0" в течение одного цикла работы ПЛК. Если CLK постоянно находится в состоянии "1" или "0", выход Q остается в состоянии "0".

ДЕМО	LD	ST
Задание переменных	VAR R_TRIG_0: R_TRIG; M0:BOOL; M1:BOOL; END_VAR	
Пример программы		1 R_TRIG_0(CLK:= M0, Q=>); 2 IF R_TRIG_0.Q THEN 3 M1:=TRUE; 4 END_IF
Результат операции		

1-1-4-2. Обнаружение спадающего фронта F_TRIG

Для обнаружения спадающего фронта.

Обнаружение спадающего фронта [F_TRIG]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	Функция
F_TRIG		F_TRIG_0(CLK:= , Q=>);

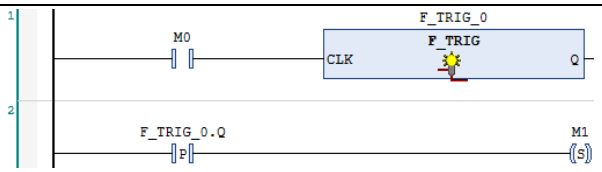
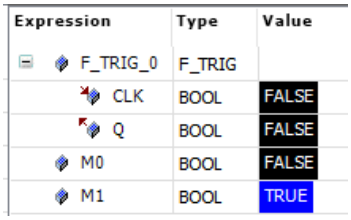
Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CLK	Входной сигнал	BOOL	[FALSE,TRUE]	FALSE	Действует спадающий фронт сигнала (от 1 до 0).

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Q	Выходной сигнал	BOOL	[FALSE,TRUE]	FALSE	Обнаружение спадающего фронта

Функция: Используется для обнаружения спадающего фронта.

Синтаксис: После изменения CLK с "1" на "0" запускается детектор спадающего фронта, и выход Q сначала переключается с "1" на "0" в течение одного цикла работы ПЛК. Если CLK постоянно находится в состоянии "1" или "0", выход Q остается в состоянии "0".

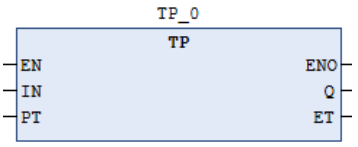
ДЕМО	LD	ST
Задание переменных	VAR F_TRIG_0: F_TRIG; M0:BOOL; M1:BOOL; END_VAR	
Пример программы		1 F_TRIG_0(CLK:= M0, Q=>); 2 IF F_TRIG_0.Q THEN 3 M1:=TRUE; 4 END_IF
Результат операции		

1-2. Таймер

1-2-1. Импульсный таймер TP

Импульсный таймер.

Импульсный таймер [TP]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	ST
TP		TP_0(IN:= , PT:= , Q=> , ET=>);

Связанные переменные

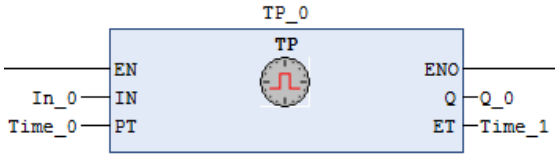
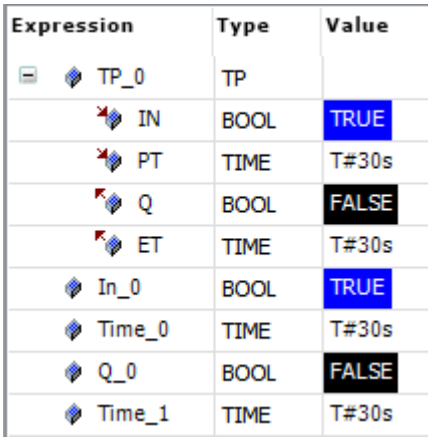
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN	Входной сигнал	BOOL	[FALSE,TRUE]	FALSE	Запуск отсчета времени FALSE → TRUE
PT	Настройка времени	TIME	-	T#0 мс	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Отсчет времени завершен	BOOL	[FALSE,TRUE]	FALSE	Отсчет времени TRUE → FALSE завершен
ET	Текущее время	TIME	-	T#0 мс	-

Функция: Отсчет времени импульсов.

Как только IN примет значение "True", Q станет "True", и время начнет отсчитываться в миллисекундах в ET, пока его значение не станет равным PT, и Q не станет "FALSE".

Когда время PT истечет, ET будет сохраняться заданное время, пока IN не станет "FALSE" и не вернется к T#0 мс. Если вход IN уже изменился на "FALSE" до достижения времени PT, ET немедленно сбросится на T # 0 мс после достижения заданного времени PT.

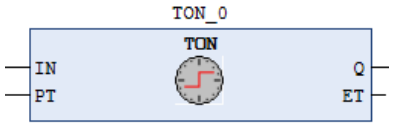
Если вход IN принял значение FALSE, а PT=T#0 мс, таймер можно сбросить.

ДЕМО	LD	ST
Задание переменных		VAR TP_0 : TP; In_0 :BOOL; Time_0:TIME; Q_0 :BOOL; Time_1:TIME; END_VAR
Пример программы		1 TP_0(IN:= In_0, 2 PT:= Time_0, 3 Q=> Q_0, 4 ET=> Time_1);
Результат операции		

1-2-2. Таймер задержки при включении питания TON

После задержки при включении питания на определенный промежуток времени выдается сигнал о завершении отсчета времени.

Таймер задержки при включении питания [TON]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	Функция
TON		TON_0(IN:= , PT:= , Q=> , ET=>);

Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN	Входной сигнал	BOOL	[FALSE,TRUE]	FALSE	Запуск отсчета времени FALSE → TRUE; сохранение значения истинным (true) во время отсчета времени
PT	Настройка времени	TIME	-	T#0 мс	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Отсчет времени завершен	BOOL	[FALSE,TRUE]	FALSE	После завершения отсчета времени FALSE изменяется на True.
ET	Текущее время	TIME	-	T#0 мс	-

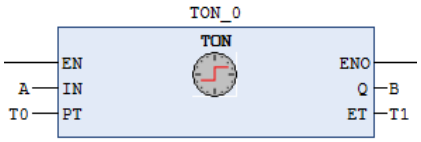
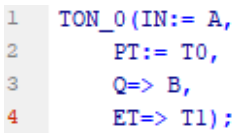
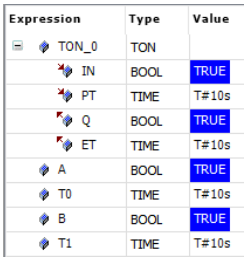
Функция: Отсчет времени задержки при включении питания.

Таймер запускается после изменения входа IN с "FALSE" на "True". Если сигнал IN на входе всегда сохраняет значение "True", а ET достигает заданного времени PT, выход Q будет иметь значение "True".

Выход ET обеспечивает время отсчета, которое начинается с T#0c и заканчивается в заданное время PT. Когда ET достигнет PT, оно будет сохраняться заданное время, пока IN не примет значение "FALSE".

Если вход IN станет "FALSE", то выход ET немедленно примет значение T#0c до того, как ET достигнет времени отсчета PT. Чтобы перезапустить таймер, IN можно установить на "FALSE".

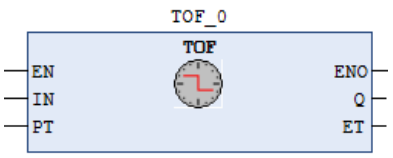
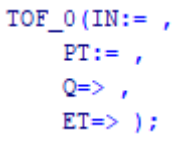
ДЕМО	LD	ST
Задание переменных	VAR TON_0: TON; A:BOOL; T0:TIME; B:BOOL; T1:TIME; END_VAR	

Пример программы		
Результат операции		

1-2-3. Таймер задержки при выключении питания TOF

После задержки при выключении питания на определенный промежуток времени выдается сигнал о завершении отсчета времени.

Таймер задержки при выключении питания [TOF]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	Функция
TOF		

Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN	Входной сигнал	BOOL	[FALSE,TRUE]	FALSE	Запуск отсчета времени TRUE → FALSE ; сохранение значения ложным (FALSE) во время отсчета времени

PT	Настройка времени	TIME	-	T#0 мс	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Отсчет времени завершен	BOOL	[FALSE,TRUE]	FALSE	После завершения отсчета времени TRUE изменяется на FALSE.
ET	Текущее время	TIME	-	T#0 мс	-

Функция: Отсчет времени задержки при выключении питания.

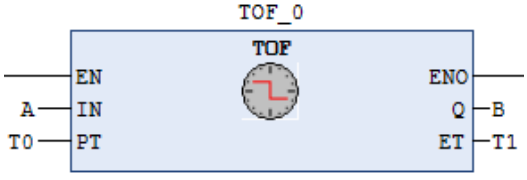
Когда состояние входной клеммы IN изменяется с "FALSE" на "True", выходной сигнал Q таймера принимает значение "True". Таймер запускается, когда состояние входной клеммы IN начинает изменяться на "FALSE". Во время отсчет времени по таймеру выход Q сохраняет значение "True". После достижения времени отсчета PT состояние выходной клеммы Q сбрасывается.

Выход ET обеспечивает время отсчета, которое начинается с T#0c и заканчивается в заданное время PT. Когда ET достигнет PT, оно будет сохраняться заданное время, пока IN не примет значение "True".

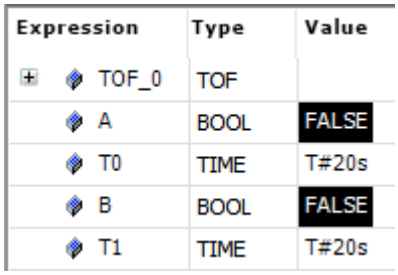
Если вход IN станет "True", то выход ET немедленно примет значение T#0c до того, как ET достигнет времени отсчета PT. Чтобы перезапустить таймер, IN можно установить на "True".

ДЕМО	**LD**	**ST**
Задание переменных		
Пример программы		

```
VAR
    TOF_0: TOF;
    A    :BOOL;
    T0   :TIME;
    B    :BOOL;
    T1   :TIME;
END_VAR
```




```
1  TOF_0(IN:= A,
2        PT:= T0,
3        Q=> B,
4        ET=> T1);
```

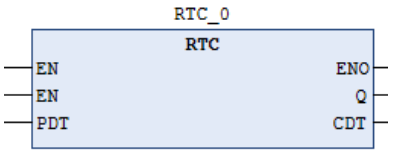
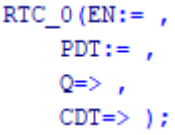
Результат операции	

Expression	Type	Value
TOF_0	TOF	
A	BOOL	FALSE
T0	TIME	T#20s
B	BOOL	FALSE
T1	TIME	T#20s

1-2-4. Часы реального времени

Начало отсчета времени с заданного времени.

Часы реального времени [RTC]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Команда	Пиктограммы команд	Функция
RTC		


```
RTC_0(EN:= ,
      PDT:= ,
      Q=> ,
      CDT=> );
```

Связанная переменная

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
EN	Включение	BOOL	[FALSE,TRUE]	FALSE	Нарастающий фронт: CDT установлено на PDT, запуск по нарастающему фронту при CDT Спадающий фронт: CDT установлено на DT#1970-01-01 00:00:00

PDT	Настройка времени и даты запуска	DATE_AND_TIME	(DT#1970-1-1-0:0:0)-(2106-2-7-6:28:15)	DT#1970-1-1-0:0:0	Заданные дата и время
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Выходной сигнал состояния	BOOL	[FALSE,TRUE]	FALSE	CDT принимает значение TRUE при отсчете времени
CDT	Текущее состояние отсчета времени и даты	DATE_AND_TIME	(DT#1970-1-1-0:0:0)-(2106-2-7-6:28:15)	DT#1970-1-1-0:0:0	Текущее состояние отсчета времени и даты

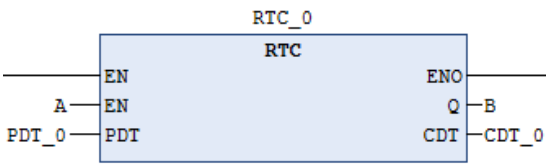
Если EN принимает значение "True", то CDT будет установлено на время, заданное PDT, и отсчет будет вестись в секундах. После сброса EN на "FALSE", и когда выходная переменная Q примет значение "FALSE", CDT будет сброшено на исходное значение DT #1970-01-00:00:00.

Время PDT действует только по нарастающему фронту.

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    RTC_0: RTC;
    A    :BOOL;
    PDT_0:DATE_AND_TIME;
    B    :BOOL;
    CDT_0:DATE_AND_TIME;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  RTC_0(EN:= A,
2        PDT:= PDT_0,
3        Q=> B,
4        CDT=> CDT_0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
RTC_0	RTC	
A	BOOL	TRUE
PDT_0	DATE_...	DT#1990-1-2-2:0:0
B	BOOL	TRUE
CDT_0	DATE_...	DT#1990-1-2-2:0:2

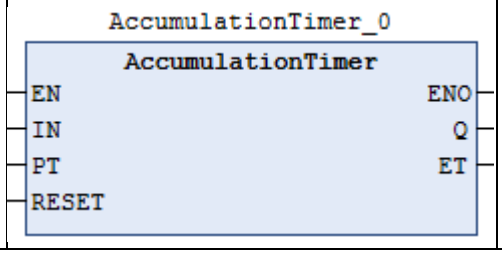
</td></tr>
</table>

1-2-5. Таймер накопления

(1) Обзор команды

Таймер накопления.

Таймер накопления [AccumulationTimer]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
AccumulationTimer	Таймер накопления		AccumulationTimer_0(IN:= , PT:= , RESET:= , Q=> , ET=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Вход таймера	BOOL	FALSE/TRUE	FALSE	FALSE: остановка таймера TRUE: таймер работает
PT	Уставка времени	TIME	В качестве типа данных	0	Максимальное значение отсчета времени
Reset	Сброс	BOOL	FALSE/TRUE	FALSE	FALSE:таймер не сброшен TRUE: таймер сброшен

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Q	Выход таймера	BOOL	FASLE/TRUE	-	TRUE: “ET” достигло “PT” FALSE: “ET” не достигло “PT”
ET	Накопленное время	TIME	В качестве типа данных	-	Накопленное время

(3) Функция

- Таймер, который накапливает время, когда вход "In" принимает значение true. В качестве единицы измерения времени задается «мс», а точность синхронизации — 100 нс.
- При сбросе "Reset" на FALSE и изменении "In" с FALSE на True запускается таймер, и накопленное время "ET" увеличивается одновременно со временем.
- При изменении "In" на FALSE таймер останавливается. Но при этом сохраняется значение "ET".
- При повторном изменении "In" на TRUE таймер снова запускается. "ET" увеличивается от времени удержания.
- Когда "ET" достигнет уставки времени "PT", выход таймера "Q" изменится на True. В этот момент "ET" прекращает увеличиваться.
- Таймер сбрасывается, когда "Reset" примет значение true. Значение "ET" становится равным 0, а "Q" становится FALSE.

(4) Примечание

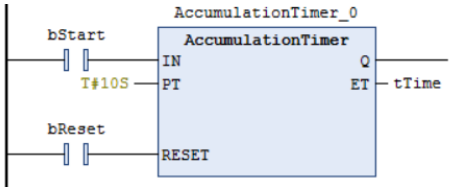
- "ET" и "Q" обновляются во время выполнения этой команды. Поэтому, строго говоря, условием истинности "Q" не является момент, когда совокупное время работы таймера равно "PT". Скорее, это момент, когда суммарное время работы таймера достигнет "PT", и эта команда будет выполнена впервые. Таким образом, возникнет задержка до одного цикла выполнения задачи.
- PT и ET выражены в мс, а точность синхронизации — 100 нс.
- Если оба параметра "In" и "Reset" являются истинными, "Reset" имеет приоритет. Значение "ET" становится равным 0, а "Q" становится FALSE.
- Если в начале выполнения программы значение "In" уже является истинным, отсчет времени начнется с этого момента.

- Если для "PT" задано T#0 мс, значения "In" и "Q" станут истинными (true).
- Пока значение "ET" не достигло значения "PT", значение "PT" можно изменять, но суммарное время "ET" по-прежнему рассчитывается в соответствии с исходным значением "PT".

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   AccumulationTimer_0: AccumulationTimer;
   bStart: BOOL;
   bReset: BOOL;
   tTime: TIME;
   XDONE: BOOL;
END_VAR
```

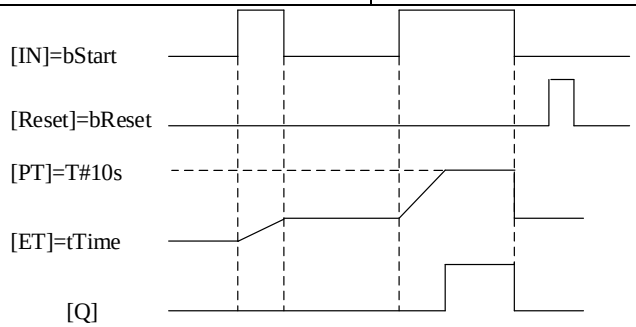
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
AccumulationTimer_0(
    IN:= bStart,
    PT:= T#10S,
    RESET:= bReset,
    Q=> XDONE,
    ET=> tTime);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">



Когда суммарное время, в течение которого переменная bStart будет иметь значение true, достигнет 10 секунд, переменная XDONE примет значение true.

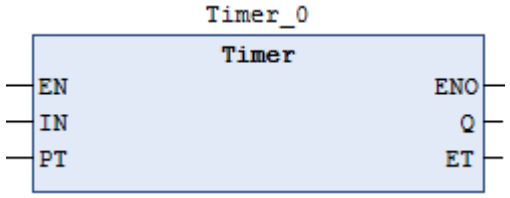
</td></tr>
</table>

1-2-6. Таймер 100 мс

(1) Обзор команды

Таймер 100 мс.

Таймер 100 мс [Timer]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
Timer	Таймер 100 мс		Timer_0(IN:= , PT:= , Q=> , ET=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Вход таймера	BOOL	FALSE/TRUE	FALSE	FALSE: остановка таймера TRUE: таймер работает
PT	Уставка времени	UINT	В качестве типа данных	0	Максимальное значение отсчета времени составляет 100 мс.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Выход таймера	BOOL	FASLE/TRUE	-	TRUE: “ET” достигло “PT” FALSE: “ET” не достигло “PT”
ET	Оставшееся время	UINT	В качестве типа данных	-	Оставшееся время

(3) Функция

- По истечении заданного времени с момента запуска таймер выдает значение true. Время задается в 100 мс.
- Если значение входного сигнала "In" в таймер станет FALSE, таймер сбросится. Оставшееся время "ET" станет равным заданному времени "PT", а значение на выходе таймера "Q" станет FALSE.
- Если "In" изменится с FALSE на True, таймер запустится. Значение "ET" уменьшается с течением времени.
- После того, как "ET" уменьшится до 0, значение на выходе таймера "Q" станет

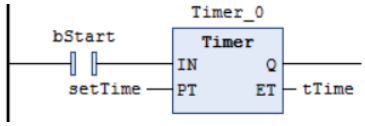
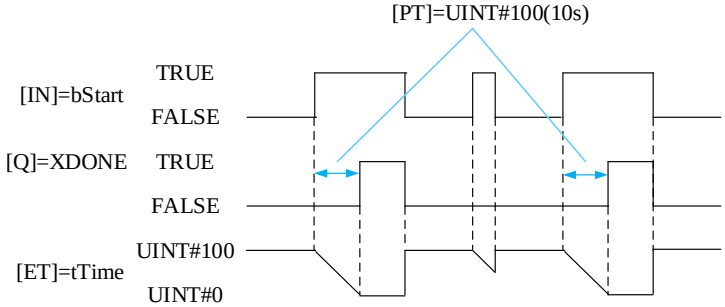
True. В этот момент значение "ET" прекращает уменьшаться.

- После запуска таймера, если "In" станет FALSE, таймер сбросится, даже если "ET" не уменьшится до 0.

(4) Примечание

- Вызов нескольких команд TIMER в одном POU может привести к исключениям в программе.
- Ошибка синхронизации, требуемая для того, чтобы значения "PT" и "Q" стали истинными, составляет 1 цикл выполнения задачи. Таймер определяет, достигло ли время "ET" для каждого цикла выполнения задачи значения "PT". Если время "ET" достигает "PT" после вынесения решения, оно задерживается на 1 цикл выполнения задачи.
- В начале выполнения программы, если значение "In" уже true, немедленно запустите таймер.
- Если значение "PT" изменилось, его нужно применить во время следующего сброса таймера; оно не действует во время синхронизации по времени.
- Ручное изменение значений внутренних переменных параметров (ET, Q) не приведет к желаемому результату.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2"><pre>VAR
   Timer_0: Timer;
   bStart: BOOL;
   tTime: UINT;
   setTime: UINT:= 100;
   XDONE: BOOL;
END_VAR</pre></td></tr>
<tr><td>Пример программы</td><td>
</td><td><pre>Timer_0(
    IN:= bStart,
    PT:= setTime,
    Q=> XDONE,
    ET=> tTime);</pre></td></tr>
<tr><td>Результат операции</td><td colspan="2">

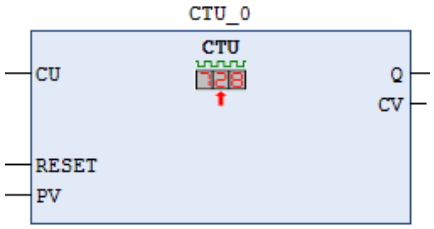
После того как переменная bStart станет True на 100*100 мс (10 с), значение переменной XDONE станет True.</td></tr>
</table>

1-3. Счетчики

1-3-1. Счетчик прямого действия CTU

При однократном запуске по нарастающему фронту сигнала значение счетчика увеличивается на 1.

Счетчик прямого действия [CTU]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Название команды	Пиктограммы команд	Функция
CTU		CTU_0(CU:= , RESET:= , PV:= , Q=> , CV=>);

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CU	Вход счетного сигнала	BOOL	[FALSE,TRUE]	FALSE	Нарастающий фронт: Значение счетчика CV плюс 1
RESET	Сброс входа	BOOL	[FALSE,TRUE]	FALSE	TRUE: Сброс счетчика CV на 0
PV	Установка верхнего предела подсчета	WORD	0-65535	0	Установка верхнего предела подсчета

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Q	Выходной сигнал состояния	BOOL	[FALSE,TRUE]	FALSE	Если CV>=PV, то значением будет TRUE
CV	Текущее значение счета	WORD	0-65535	0	Текущее значение счета

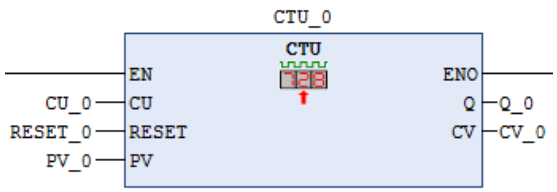
При изменении сигнала CU с "FALSE" на "True" текущее вычисленное значение увеличивается на 1 и отображается через выходную клемму CV. Если значение счетчика достигает верхнего предела 65535, оно больше не увеличивается.

Когда состояние сигнала RESET равно "True", CV счетчика инициализируется до 0, а Q становится "FALSE", и нарастающий фронт больше не оказывает никакого влияния на CU.

Если значение CV больше или равно PV, выход Q примет значение "True". В этот момент CV может продолжать накапливаться, а выход Q продолжает быть равным "True".

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    CU_0   :BOOL;
    RESET_0:BOOL;
    PV_0   :WORD := 3;
    Q_0    :BOOL;
    CV_0   :WORD;
    CTU_0  : CTU;
END_VAR
```



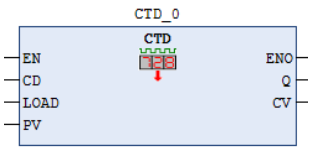

```
1  CTU_0(CU:= CU_0,
2       RESET:= RESET_0,
3       PV:= PV_0,
4       Q=> Q_0,
5       CV=> CV_0);
```

Expression	Type	Value
CU_0	BOOL	TRUE
RESET_0	BOOL	FALSE
PV_0	WORD	3
Q_0	BOOL	TRUE
CV_0	WORD	3
CTU_0	CTU	

1-3-2. Счетчик обратного действия CTD

При однократной подаче сигнала с нарастающим фронтом значение счетчика уменьшается на 1.

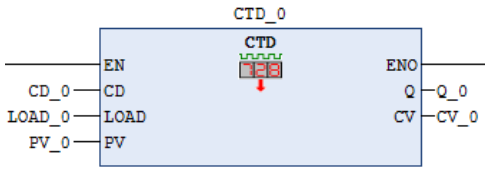
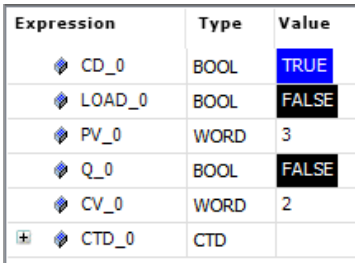
Счетчик обратного действия [CTD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Название команды	Пиктограммы команд	Функция
CTD		CTD_0(CD:= , LOAD:= , PV:= , Q=> , CV=>);

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CU	Вход счетного сигнала	BOOL	[FALSE,TRUE]	FALSE	Нарастающий фронт: Значение счетчика CV минус 1
LOAD	Сигнал нагрузки	BOOL	[FALSE,TRUE]	FALSE	True: CV задается в качестве начального значения PV.
PV	Отклонение начального значения	WORD	0-65535	0	Отклонение начального значения
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Выходной сигнал состояния	BOOL	[FALSE,TRUE]	FALSE	CV=0, это TRUE
CV	Текущее значение счета	WORD	0-65535	0	Текущее значение счета

Сигнал нагрузки должен измениться с "FALSE" на "True", параметр CV будет установлен на начальное значение, заданное по PV, выход Q изменится с начального значения "True" на "FALSE", а нагрузка перейдет в состояние "FALSE" до того, как данный функциональный блок вступит в силу.

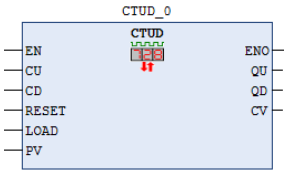
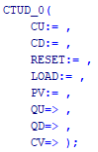
При изменении сигнала CD на входе счетчика с состояния "FALSE" на "True" текущее значение счетчика уменьшится на 1 и отобразится на выходе CV. Если значение CV уменьшится до 0, Q примет значение True, и значение счетчика больше не будет уменьшаться.

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2"><pre>
VAR
    CD_0   :BOOL;
    LOAD_0 :BOOL;
    PV_0   :WORD := 3;
    Q_0   :BOOL;
    CV_0   :WORD;
    CTD_0 : CTD;
END_VAR
</pre></td></tr>
<tr><td>Пример программы</td><td>
</td><td><pre>
1 CTD_0(CD:= CD_0,
2   LOAD:= LOAD_0,
3   PV:= PV_0,
4   Q=> Q_0,
5   CV=> CV_0);
</pre></td></tr>
<tr><td>Результат операции</td><td colspan="2">
</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">После установки LOAD_0 на True Q_0 изменится с True на FALSE, а CV_0 станет равным 3. Повторный сброс LOAD_0 на FALSE. В этот момент сигналы поступают через CD_0, а значение CV_0 уменьшается на 1.</td></tr>
</table>

1-3-3. Счетчик прямого/обратного действия CTUD

Можно увеличить значение счетчика на 1 или уменьшить на 1.

Счетчик прямого/обратного действия [CTUD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Стандартн.		

Название команды	Пиктограммы команд	Функция
CTUD		

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CU	Вход счетчика прямого действия	BOOL	[FALSE,TRUE]	FALSE	Нарастающий фронт: Значение счетчика CV плюс 1
CD	Вход счетчика обратного действия	BOOL	[FALSE,TRUE]	FALSE	Нарастающий фронт: Значение счетчика CV минус 1
RESET	Сброс входа	BOOL	[FALSE,TRUE]	FALSE	TRUE: Сброс счетчика CV на 0
LOAD	Сигнал нагрузки	BOOL	[FALSE,TRUE]	FALSE	True: CV задается в качестве начального значения PV.
PV	Установка верхнего предела подсчета	WORD	0-65535	0	Установка верхнего предела подсчета
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
QU	Выход прямого счета	BOOL	[FALSE,TRUE]	FALSE	Если CV>=PV, то значением будет TRUE
QD	Выход обратного счета	BOOL	[FALSE,TRUE]	FALSE	CV=0, это TRUE
CV	Текущее значение счета	WORD	0-65535	0	Текущее значение счета

При изменении сигнала CU на входе счетчика с состояния "FALSE" на "True" текущее значение счетчика увеличится на 1 и отобразится на выходе CV.

При изменении сигнала CD на входе счетчика с состояния "FALSE" на "True" (сигнал нагрузки должен измениться с "FALSE" на "True", а затем на состояние "FALSE" до того, как данный функциональный блок вступит в силу) текущее значение счетчика уменьшится на 1 и отобразится на выходе CV. Если оба входа имеют нарастающие фронты, текущее значение счетчика останется неизменным.

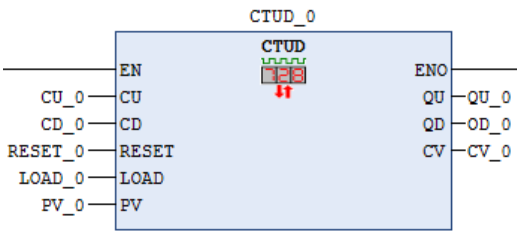
Если значение счетчика достигнет верхнего предела 65535, нарастающий фронт входа счетчика CU больше не будет действовать. Таким образом, даже если на входе счетчика CU есть нарастающий фронт, его значение не будет увеличиваться. Аналогичным образом, если значение счетчика достигнет нижнего предельного значения 0, вход счетчика CD не будет действовать. Следовательно, даже при наличии нарастающего фронта на входе счетчика CD значение счетчика не будет уменьшаться.

Если значение CV больше или равно PV, выход QU примет значение "True". Если значение CV равно 0, выход QD примет значение "True".

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    CU_0    :BOOL;
    CD_0    :BOOL;
    RESET_0:BOOL;
    LOAD_0 :BOOL;
    PV_0    :WORD := 3;
    QU_0    :BOOL;
    OD_0    :BOOL;
    CV_0    :WORD;
    CTUD_0: CTUD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  CTUD_0(
2      CU:= CU_0,
3      CD:= CD_0,
4      RESET:= RESET_0,
5      LOAD:= LOAD_0,
6      PV:= PV_0,
7      QU=> QU_0,
8      QD=> OD_0,
9      CV=> CV_0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
CU_0	BOOL	TRUE
CD_0	BOOL	FALSE
RESET_0	BOOL	FALSE
LOAD_0	BOOL	FALSE
PV_0	WORD	3
QU_0	BOOL	FALSE
OD_0	BOOL	FALSE
CV_0	WORD	1
CTUD_0	CTUD	

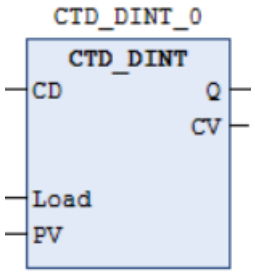
</td></tr>
</table>

1-3-4. Вычитающий счетчик CTD_XXX

(1) Обзор команды

Счетчик, выполняющий операции вычитания при подаче сигнала на вход счетчика. Типы данных для заданных значений и значений счетчика: DINT, LINT, UDIND или ULINT.

Вычитающий счетчик [CTD_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
CTD_XXX	Вычитающий счетчик		CTD_DINT_0(CD:= , Load:= , PV:= , Q=> , CV=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CD	Вход счетчика	BOOL	FALSE/TRUE	FALSE	Вход счетчика
Load	Сигнал нагрузки	BOOL	FALSE/TRUE	FALSE	Если задано TRUE, PV загружается в CV
PV	Заданное значение	DINT/LINT/ UDINT/ULINT	В качестве типа данных	0	Заданное значение счетчика

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Q	Выход счетчика	BOOL	FALSE/TRUE	FALSE	TRUE: выходы счетчика ВКЛЮЧЕНЫ FALSE: выходы счетчика ВЫКЛЮЧЕНЫ
CV	Значение счета	DINT/LINT/UDINT/ULINT	В качестве типа данных	-	Текущее значение счетчика

(3) Функция

- Вычитающий счетчик. Типы данных для заданных значений и значений счетчика: DINT, LINT, UDINT или ULINT.
- Имя команды зависит от типа данных "PV" и "CV". Например, если типом данных обоих параметров является DINT, команда будет иметь имя CTD_DINT.
- Если сигнал нагрузки "Load" принимает значение True, то заданное значение "PV" загружается в значение счетчика "CV", а значение на выходе счетчика "Q" станет FALSE. Если сигнал нагрузки "Load" принимает значение True, а значение "PV" равно 0, то значение на выходе счетчика "Q" станет True.
- Если входной сигнал счетчика "CD" находится на нарастающем фронте, "CV" выполнит операцию вычитания. Если значение "CV" равно 0, значение "Q" становится True. Если значение "CV" вручную изменить так, чтобы оно было меньше 0, "Q" становится True. Кроме того, значение "Q" нельзя изменять во время работы.
- Если значение "CV" меньше 0, то даже если "CD" находится на нарастающем фронте, "CV" не выполняет операции вычитания.
- В период, когда значение "Load" истинно, "CV" нельзя изменять, и операции вычитания не выполняются.

(4) Примечание

- При повторном запуске счетчика после завершения обратного отсчета установите значение "Load" на True, прежде чем устанавливать его на FALSE.
- Задайте одинаковые типы данных для "PV" и "CV".
- Если для параметра "PV" задано отрицательное число, значение "PV"

загружается в "CV", когда значение параметра "Load" станет истинным. Значение "CV" меньше 0, поэтому значение "Q" немедленно станет истинным. После этого, даже если "CD" изменится, "CV" не будет выполнять операции вычитания.

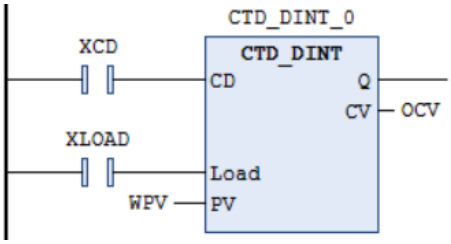
- Если значение "PV" выходит за пределы диапазона значений, "PV" будет неявно преобразовано в значение ошибки, и функция не будет иметь никаких исключений.
- После отключения источника питания, или когда программа управляет режимом действия в состоянии, при котором значение "CD" равно FALSE, и команда перезапускается, "CV" выполнит одну операцию вычитания, если значение "CD" станет истинным.
- При выполнении этой команды через релейно-контактную программу значение "Q" станет FALSE, если передняя часть схемы этой команды нарушена.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   CTD_DINT_0: CTD_DINT;
   XCD: BOOL;
   XLOAD: BOOL;
   WPV: DINT:= 5;
   OCV: DINT;
   XDONE: BOOL;
END_VAR
```

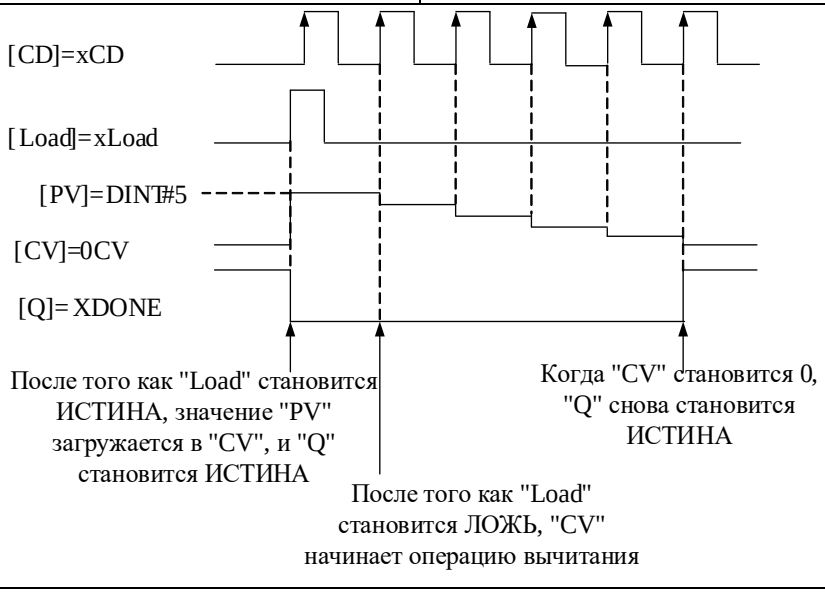
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
CTD_DINT_0(
    CD:= XCD,
    Load:= XLOAD,
    PV:= WPV,
    Q=> XDONE,
    CV=>OCV );
```

</td></tr>
<tr><td>Анализ принципа</td><td colspan="2">



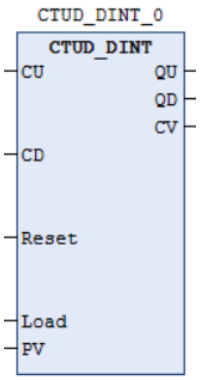
</td></tr>
</table>

1-3-5. Счетчик прямого/обратного действия CTUD_XXX

(1) Обзор команды

Счетчик, выполняющий операции сложения и вычитания на основе входных данных счетчика сложения и счетчика вычитания. Типы данных для заданных значений и значений счетчика: DINT, LINT, UDIND или ULINT.

Счетчик прямого/обратного действия [CTUD_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
CTUD_XXX	Счетчик прямого/обратного действия		CTUD_DINT_0(CU:= , CD:= , Reset:= , Load:= , PV:= , QU=> , QD=> , CV=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CU	Вход счетчика сложения	BOOL	FALSE/TRUE	FALSE	Вход счетчика сложения
CD	Вход счетчика вычитания	BOOL	FALSE/TRUE	FALSE	Вход счетчика вычитания
Reset	Сигнал сброса	BOOL	FALSE/TRUE	FALSE	Если задано TRUE, 'CV' сбрасывается на 0

Load	Сигнал нагрузки	BOOL	FALSE/TRUE	FALSE	Если задано TRUE, 'PV' загружается в 'CV'
PV	Заданное значение	DINT/LINT/ UDINT/ULINT	В качестве типа данных	0	Заданное значение счетчика

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
QU	Выход счетчика сложения	BOOL	FALSE/TRUE	FALSE	TRUE: выход счетчика прямого действия ВКЛЮЧЕН FALSE: выход счетчика прямого действия ВЫКЛЮЧЕН
QD	Выход счетчика вычитания	BOOL	FALSE/TRUE	FALSE	TRUE: выход счетчика обратного действия ВКЛЮЧЕН FALSE: выход счетчика обратного действия ВЫКЛЮЧЕН
CV	Значение счета	DINT/LINT/ UDINT/ULINT	В качестве типа данных	-	Текущее значение счетчика

(3) Функция

- Счетчик, выполняющий операции сложения и вычитания на основе входных сигналов счетчика сложения и счетчика вычитания.
- Счетчик одновременно поддерживает функции счетчика сложения и счетчика вычитания.
- Типы данных для заданных значений и значений счетчика: DINT, LINT, UDINT или ULINT.
- Имя команды зависит от типа данных "PV" и "CV". Например, если типом данных обоих параметров является LINT, команда будет иметь имя CTUD_Lint.
- Функции счетчика сложения

- Если сигнал сброса "Reset" принимает значение True, то значение счетчика "CV" станет равным 0, а значение на выходе счетчика сложения "QU" станет FALSE.
- Если входной сигнал "CU" счетчика сложения находится на нарастающем фронте, "CV" выполнит операцию сложения. Если значение "CV" равно заданному значению "PV", значение "QU" становится True. Если значение "CV" вручную изменено на значение больше значения "PV", значение "QU" становится True.
- Если значение "CV" превышает значение "PV", то "CV" не изменится, даже если повторно запущен входной сигнал "CU".
- Если "CV" меньше или равно 0, "QD" станет True.

- Функции счетчика вычитания
  - Если сигнал нагрузки "Load" принимает значение True, то заданное значение "PV" загружается в значение счетчика "CV", а значение на выходе счетчика вычитания "QD" станет FALSE.
  - Если входной сигнал "CD" счетчика вычитания находится на нарастающем фронте, "CV" выполнит операцию вычитания. Если значение "CV" равно 0, значение "QD" становится True. Вручную измените значение "CV" на значение больше значения "PV" или равное ему, после чего значение "QU" становится True. Вручную измените значение "CV" на значение меньше 0, после чего значение "QD" становится True.
  - Если значение "CV" меньше 0, то "CV" не изменится, даже если повторно запущен входной сигнал "CD".
- Общие функции счетчиков сложения и вычитания
  - В период, когда значение "Load" или "Reset" истинное, значения "CU" и "CD" игнорируются. "CV" не выполняет операции сложения и вычитания.
  - Если оба "CU" и "CD" находятся на нарастающем фронте, "CV" не изменится.
  - Если оба "Reset" и "Load" принимают значение True, то "Reset" имеет приоритет, а значение "CV" становится равным 0.
  - Если "Reset" принимает значение True, то значение "CV" становится равным 0, поэтому значение "QD" становится истинным.
  - Если "Load" принимает значение True, то значение "CV" равно "PV", поэтому значение "QU" становится истинным (True).

“Reset”	“Load”	“CV”	“QU”	“QD”	Операция
FALSE	FALSE	Меньше 0	FALSE	TRUE	Выполнение только операций счетчика сложения Если "CU" находится на нарастающем фронте, то "CV" выполняет операции сложения. Если CD находится на нарастающем фронте, то операции вычитания не выполняются.
		От 0 до “PV”	FALSE	FALSE	Выполнение операций счетчика сложения/вычитания Если "CU" находится на нарастающем фронте, то "CV" выполняет операции сложения. Если CD находится на нарастающем фронте, то CV выполняет операции вычитания.
		Больше “PV”	TRUE	FALSE	Выполнение только операций счетчика вычитания Если "CU" находится на нарастающем фронте, то операции сложения не выполняются. Если CD находится на нарастающем фронте, то CV выполняет операции вычитания.
TRUE	FALSE	0	FALSE	TRUE	
FALSE	TRUE	“PV”	TRUE	FALSE	Состояние сброса счетчика сложения, когда значение "CV" установлено на 0 Состояние сброса счетчика вычитания, значение "CV" установлено на "PV"
TRUE	TRUE	0	FALSE	TRUE	Состояние сброса счетчика сложения "Reset" имеет приоритет над "Load", а значение "CV" установлено на 0.

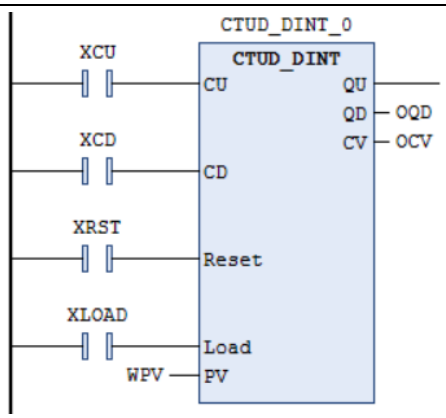
(4) Примечание

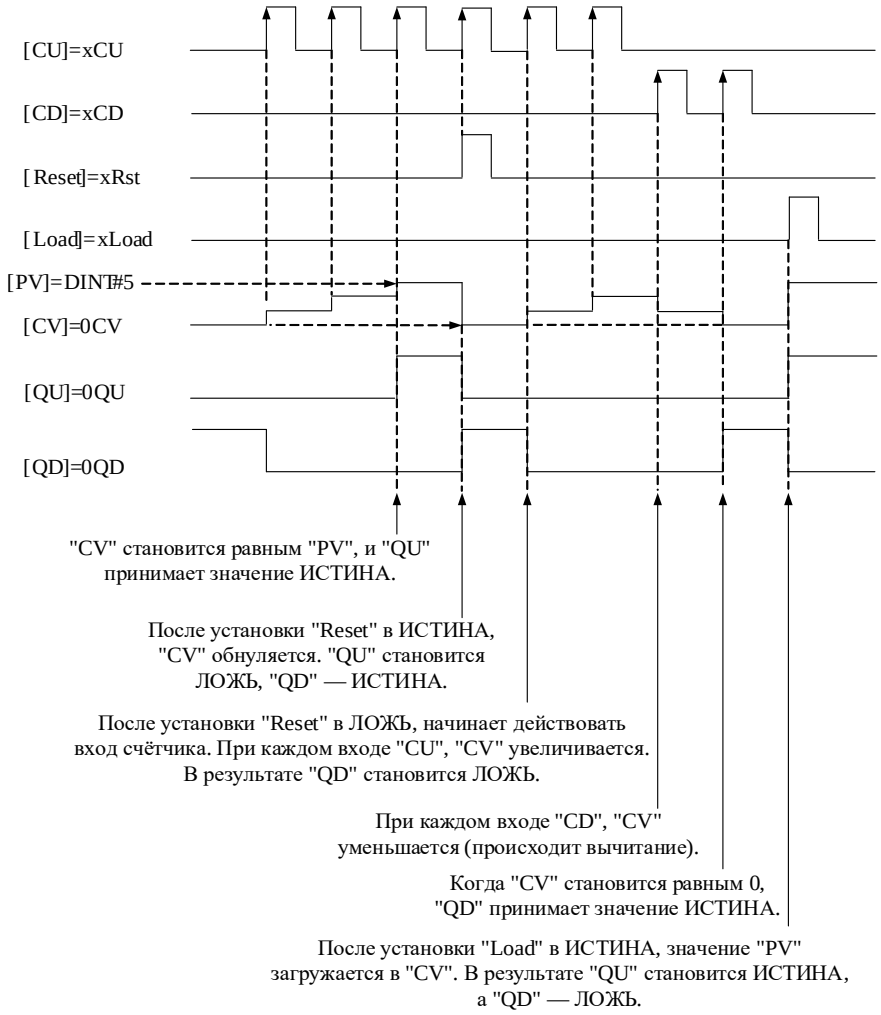
- Обратите внимание, что для сброса счетчика сложения, когда значение "Reset" становится True, значение "QU" становится FALSE, а значение "QD" становится True.
- Обратите внимание, что для сброса счетчика вычитания, когда значение "Load" становится True, значение "QD" становится FALSE, а значение "QU" становится True.
- Если для параметра "PV" задано отрицательное число, значение "PV"

загружается в "CV", когда значение параметра "Load" станет истинным. Значение "CV" меньше 0, поэтому значение "QD" немедленно станет истинным. После этого, даже если "CD" изменится, "CV" не будет выполнять операции вычитания. Значение "Reset" становится True, а значение "CV" становится равным 0. Значение "CV" больше значения "PV", поэтому значение "QU" немедленно станет истинным. После этого, даже если "CU" изменится, "CV" не будет выполнять операции сложения.

- Если значение "PV" выходит за пределы диапазона значений, "PV" будет неявно преобразовано в значение ошибки, и функция не будет иметь никаких исключений.
- Задайте одинаковые типы данных для "PV" и "CV".
- Если значение "CU" или "CD" равно FALSE, после отключения источника питания, или когда программа управляет режимом действия, и команда перезапускается, "CV" только один раз выполнит операцию сложения или вычитания, если значение "CU" или "CD" станет истинным.
- Значение "Q" нельзя изменять во время работы.
- Уставку "PV" можно настроить на отрицательное значение, и после запуска "CU"/"CD" это значение можно нормально прибавлять/вычитать, пока оно не достигнет значения "PV".

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR CTUD_DINT_0: CTUD_DINT; XCU: BOOL; XCD: BOOL; XRST: BOOL; XLOAD: BOOL; WPV: DINT:= 5; OQD: BOOL; OCV: DINT; OQU: BOOL; END_VAR
Пример программы		CTUD_DINT_0(CU:= XCU, CD:= XCD, Reset:= XRST, Load:= XLOAD, PV:= WPV, QU=> OQU, QD=> OQD, CV=> OCV);

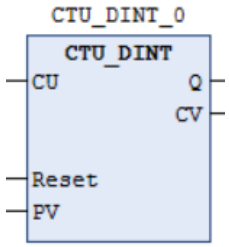
Анализ принципа	

1-3-6. Счетчик прямого действия CTU_XXX

(1) Обзор команды

Счетчик, выполняющий операции сложения при подаче сигнала на вход счетчика. Типы данных для заданных значений и значений счетчика: DINT, LINT, UDIND или ULINT.

Счетчик прямого действия [CTU_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
CTU_XXX	Счетчик прямого действия		CTU_DINT_0(CU:= , Reset:= , PV:= , Q=> , CV=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CU	Вход счетчика	BOOL	FALSE/TRUE	FALSE	Вход счетчика
Reset	Сигнал нагрузки	BOOL	FALSE/TRUE	FALSE	Если задано TRUE, происходит сброс на 0 по направлению к ' CV '
PV	Заданное значение	DINT/LINT/ UDINT/ULINT	В качестве типа данных	0	Заданное значение счетчика
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Q	Выход счетчика	BOOL	FALSE/TRUE	FALSE	TRUE: выход счетчика ВКЛЮЧЕН FALSE: выход счетчика ВЫКЛЮЧЕН
CV	Значение счета	DINT/LINT/ UDINT/ULINT	В качестве типа данных	-	Текущее значение счетчика

(3) Функция

- Счетчик сложения. Типы данных для заданных значений и значений счетчика: DINT, LINT, UDINT или ULINT.
- Имя команды зависит от типа данных "PV" и "CV". Например, если типом данных обоих параметров является DINT, команда будет иметь имя

CTU_DINT.

- Если сигнал сброса "Reset" принимает значение True, то значение счетчика "CV" станет равным 0, а значение на выходе счетчика "Q" станет FALSE.
- Если входной сигнал счетчика "CU" находится на нарастающем фронте, "CV" выполнит операцию сложения. Если значение "CV" равно заданному значению "PV", значение "Q" становится True. Если значение "CV" вручную изменено на значение больше значения "PV", значение "Q" становится True. Кроме того, значение "Q" нельзя изменять во время выполнения программы.
- Если значение "CV" превышает значение "PV", то "CV" не изменится, даже если введено большее значение "CU".
- В период, когда значение "Reset" истинное, значение "CU" игнорируется. CV не выполняет операции сложения.
- В период, когда значение "Reset" истинно, "CV" нельзя изменять, и "CV" не выполняет операции сложения.

(4) Примечание

- При повторном запуске счетчика после завершения отсчета установите значение "Reset" на True, а затем на FALSE.
- Если для параметра "PV" задано отрицательное число, значение "CV" станет 0, когда значение параметра "Reset" станет истинным. Значение "CV" больше значения "PV", поэтому значение "Q" немедленно станет истинным. После этого, даже если "CU" изменится, "CV" не будет выполнять операции сложения.
- Если значение "PV" выходит за пределы диапазона значений, "PV" будет неявно преобразовано в значение ошибки, и функция не будет иметь никаких исключений.
- Задайте одинаковые типы данных для "PV" и "CV".
- Если значение "Reset" становится FALSE, то при изменении значения "PV" будут выполняться операции, указанные в таблице ниже:

Значение “PV”	Описание
"CV" больше, чем в тот момент	Продолжение счета
"CV" меньше, чем в тот момент	Счет окончен. Значение "Q" станет истинным, при этом значение "CV" не изменится.

- Если значение "CU" равно FALSE, после отключения источника питания, или

когда программа управляет режимом действия, и команда перезапускается, "CV" только один раз выполнит операцию сложения, если значение "CU" станет истинным.

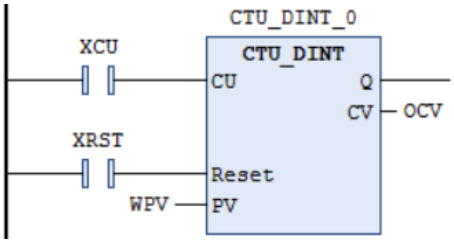
- При выполнении этой команды через релейно-контактную программу значение "Q" станет FALSE, если передняя часть схемы этой команды нарушена.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   CTU_DINT_0: CTU_DINT;
   XCU: BOOL;
   XRST: BOOL;
   WPV: DINT:= 5;
   OCV: DINT;
   XDONE: BOOL;
END_VAR
```

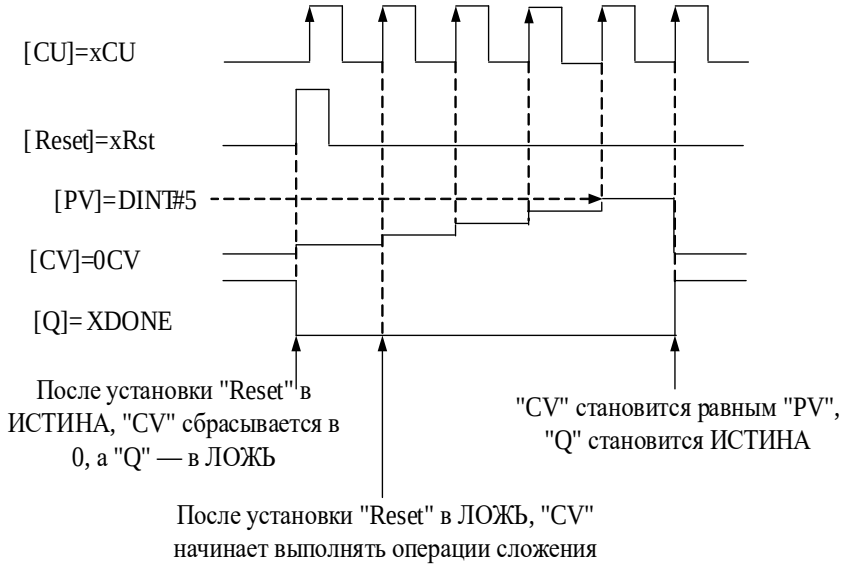
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
CTU_DINT_0(
    CU:= XCU,
    Reset:= XRST,
    PV:= WPV,
    Q=> XDONE,
    CV=> OCV);
```

</td></tr>
<tr><td>Анализ принципа</td><td colspan="2">



</td></tr>
</table>

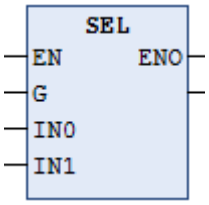
1-4. Команды обработки данных

1-4-1. Команды выбора

1-4-1-1. Команда двоичного выбора SEL

Когда селекторный переключатель находится в состоянии FALSE, выходными данными являются первые входные данные, а когда селекторный переключатель находится в состоянии True, выходными данными являются вторые данные.

Двоичный выбор [SEL]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Пиктограммы команд	Функция
SEL		SEL

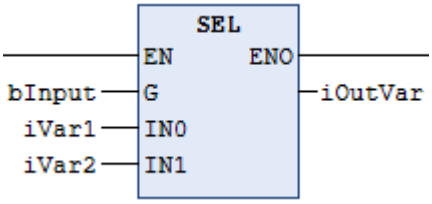
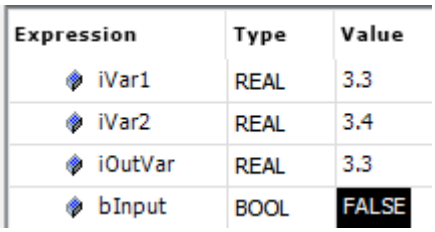
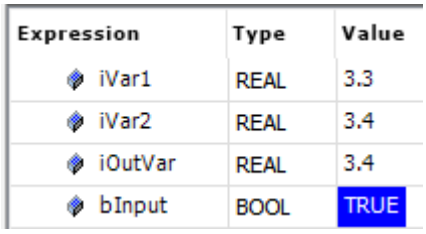
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
G	Входной сигнал	BOOL	[FALSE,TRUE]	FALSE	FALSE: вывод данных IN0 TRUE: вывод данных IN1
IN0	Входная переменная	Любой тип	В качестве типа данных	0	G принимает значение FALSE: вывод данных IN0
IN1	Входная переменная	Любой тип	В качестве типа данных	0	G принимает значение TRUE: вывод данных IN1

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
OUT	Выходная переменная	Любой тип	В качестве типа данных	0	Выходная переменная

С помощью переключателя выберите один из двух наборов входных данных в качестве выходных данных. Если G принимает значение FALSE, результатом возвращаемого значения будет IN0; если G принимает значение True, результатом возвращаемого значения будет IN1.

ДЕМО	LD	ST

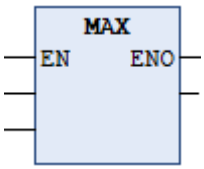
Задание переменной	VAR
	iVar1 :REAL:=3.3;
	iVar2 :REAL:=3.4;
	iOutVar:REAL;
	bInput :BOOL;
	END_VAR	

Пример программы		1 iOutVar:=SEL(bInput,iVar1,iVar2);
Результат операции	 	
Анализ результатов	Если входное значение bInput равно FALSE, выходным значением будет 3.3. И наоборот, если входное значение bInput равно true, выходным значением будет 3.4.	

1-4-1-2. Выбор максимального значения MAX

Выбор максимального из двух значений для вывода.

Выбор максимального значения [MAX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

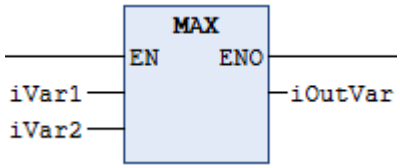
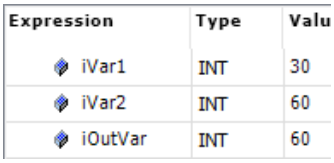
Команда	Пиктограммы команд	Функция
MAX		MAX

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN0	Входная переменная	Любой тип	В качестве типа данных	0	Входные данные 0
INn	Входная переменная	Любой тип	В качестве типа данных	0	Входные данные 1
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Выходная переменная	Любой тип	В качестве типа данных	0	Выходная переменная

Из двух значений нужно выбрать максимальное.

ДЕМО	LD	ST

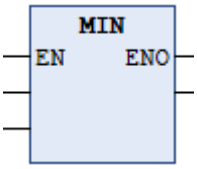
Задание переменной	VAR
	iVar1 :INT:=30;
	iVar2 :INT:=60;
	iOutVar:INT;
	END_VAR	

Пример программы		1 iOutVar:=MAX(iVar1,iVar2);
Результат операции		

1-4-1-3. Выбор минимального значения MIN

Выбор минимального из двух значений для вывода.

Выбор минимального значения [MIN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

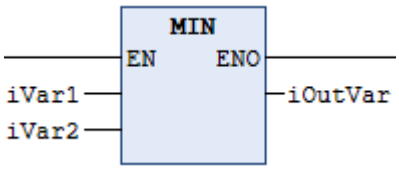
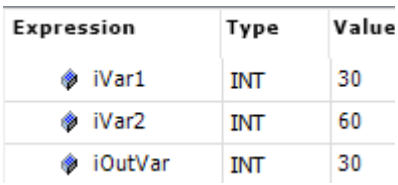
Команда	Пиктограммы команд	Функция
MIN		MIN

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN0	Входная переменная	Любой тип	В качестве типа данных	0	Входные данные 0
INn	Входная переменная	Любой тип	В качестве типа данных	0	Входные данные 1
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Выходная переменная	Любой тип	В качестве типа данных	0	Выходная переменная

Из двух значений нужно выбрать минимальное.

ДЕМО	LD	ST

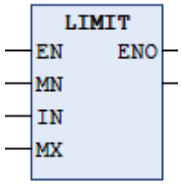
Задание переменной	VAR
	iVar1 :INT:=30;
	iVar2 :INT:=60;
	iOutVar:INT;
	END_VAR	

Пример программы		1 iOutVar:=MIN(iVar1,iVar2);
Результат операции		

1-4-1-4. Предельное значение LIMIT

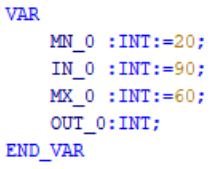
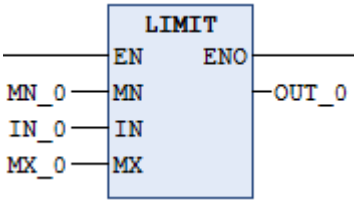
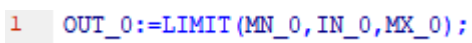
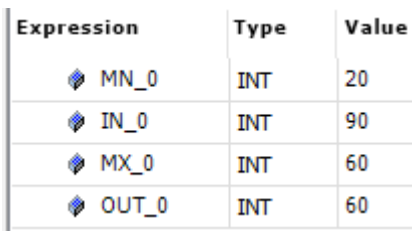
Если входные данные находятся между верхним и нижним предельными значениями, выведите это значение. Если входные данные ниже нижнего предельного значения, выведите нижнее предельное значение. Если входные данные выше верхнего предельного значения, выведите верхнее предельное значение.

Предельное значение [LIMIT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Пиктограммы команд	Функция
LIMIT		LIMIT

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
MN	Нижнее предельное значение	Любой тип	В качестве типа данных	0	Минимальное значение
IN	Входная переменная	Любой тип	В качестве типа данных	0	Входные данные
MX	Верхнее предельное значение	Любой тип	В качестве типа данных	0	Максимальное значение
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Выходная переменная	Любой тип	В качестве типа данных	0	Выходная переменная

Вывод предельного значения. Определите, находятся ли входные данные между минимальным и максимальным значениями. Если входные данные находятся между этими двумя значениями, выведите их напрямую в качестве выходных данных. Если входные данные превышают максимальное значение, то максимальное значение принимают за выходное значение. Если входные данные меньше минимального значения, то минимальное значение принимают за выходное значение.

ДЕМО	LD	ST
Задание переменной		
Пример программы		
Результат операции		

```
VAR
    MN_0 :INT:=20;
    IN_0 :INT:=90;
    MX_0 :INT:=60;
    OUT_0:INT;
END_VAR
```



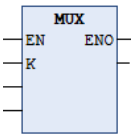
```
1  OUT_0:=LIMIT(MN_0,IN_0,MX_0);
```

Expression	Type	Value
MN_0	INT	20
IN_0	INT	90
MX_0	INT	60
OUT_0	INT	60

1-4-1-5. Множественный выбор MUX

Команда MUX позволяет выбирать значение Kth из набора значений. Первое значение — K=0. Если K больше количества других входов (n), Codesys передает последнее значение.

Множественный выбор [MUX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

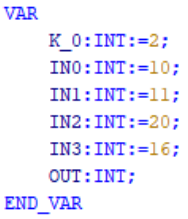
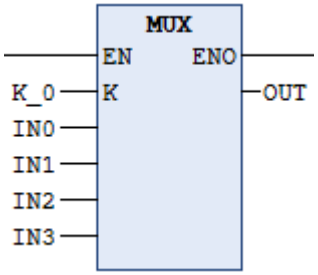
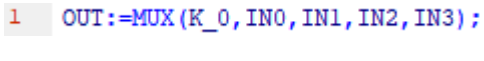
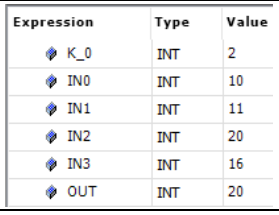
Команда	Пиктограммы команд	Функция
MUX		MUX

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
K	Нижнее предельное значение	BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, LINT, ULINT, UDINT	В качестве типа данных	0	Контрольное число
IN0	Входная переменная	Любой тип	В качестве типа данных	0	Входные данные 0
INn	Верхнее предельное значение	Любой тип	В качестве типа данных	0	Входные данные n

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
OUT	Выходная переменная	Любой тип	В качестве типа данных	0	Выходные данные

Команда MUX позволяет выбирать K-й выход данных из группы переменных.

Если значение K меньше 0 или больше количества входных переменных, выведите значение последней переменной.

ДЕМО	LD	ST
Задание переменной		
Пример программы		
Результат операции		

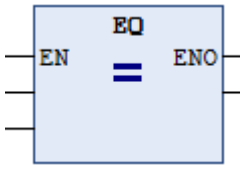
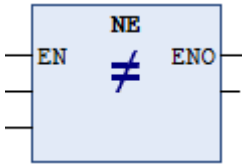
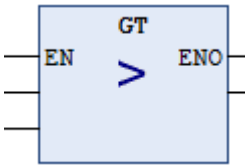
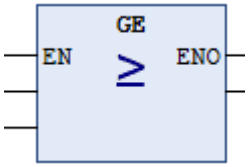
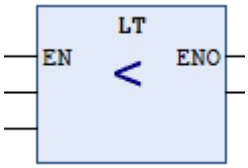
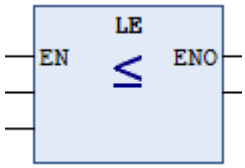
Expression	Type	Value
K_0	INT	2
IN0	INT	10
IN1	INT	11
IN2	INT	20
IN3	INT	16
OUT	INT	20

1-4-2. Команды сравнения

1-4-2-1. Основные команды сравнения EQ/NE/GT/GE/LT/LE

Команда сравнения размера данных.

Основные команды сравнения [EQ/NE/GT/GE/LT/LE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

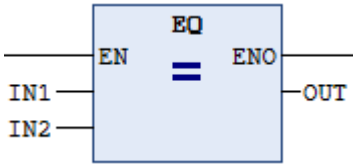
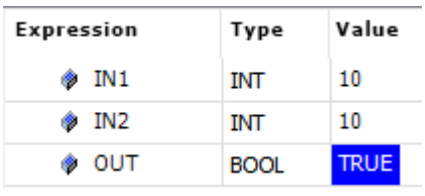
Название команды	Пиктограммы команд	Функция
EQ		=
NE		<>
GT		>
GE		>=
LT		<
LE		<=

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN0	Входная переменная	Любой тип	В качестве типа данных	0	Значение для сравнения
IN1	Входная переменная	Любой тип	В качестве типа данных	0	Значение для сравнения

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
OUT	Результат сравнения	BOOL	[FALSE,TRUE]	0	Выходные данные

Сравнение с помощью основных команд сравнения

Команда	Сравнение	Результат
EQ	IN0=IN1	TRUE
NE	IN0<>IN1	TRUE
GT	IN0>IN1	TRUE
GE	IN0>=IN1	TRUE
LT	IN0<IN1	TRUE
LE	IN0<=IN1	TRUE

ДЕМО	LD	ST
Задание переменной	VAR IN1:INT:=10; IN2:INT:=10; OUT:BOOL; END_VAR	
Пример программы		1 OUT:=IN1=IN2;
Результат операции		

1-4-2-2. Получение максимального/минимального значения массивов AryMax/AryMin

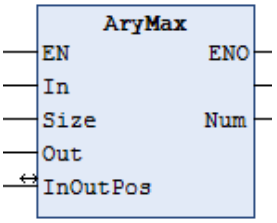
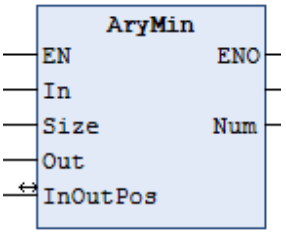
(1) Обзор команды

Получение максимальных/минимальных значений одномерного массива.

AryMax: Получение максимального значения элемента одномерного массива.

AryMin: Получение минимального значения элемента одномерного массива.

Получение максимального/минимального значения массивов [AryMax/AryMin]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryMax	Получение максимального значения массивов		Out:=AryMax(In:= , Size:= , Out:= , InOutPos:= , Num=>);
AryMin	Получение минимального значения массивов		Out:=AryMin(In:= , Size:= , Out:= , InOutPos:= , Num=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In[]	Получение массива объектов	USINT/UINT/UDINT /ULINT/SINT/INT/DINT/ LINT/REAL/LREAL/TIME/ DATE/TOD/DT/STRING	В качестве типа данных	(*)	Получение массива объектов

Size	Получение информации о количестве элементов в массиве объектов	UINT	В качестве типа данных	1	В [] количество элементов в полученном массиве объектов
Out	Результаты поиска	Соответствует типу данных элемента In[]	В качестве типа данных	(*)	Результаты поиска
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Num	Количество полученных значений	UINT	В качестве типа данных	-	Количество полученных значений
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
InOutPos	Полученные элементы	UINT	В качестве типа данных	-	Номера элементов массива для полученных значений

(3) Функция

- Получение элементов массива "Size", начиная с элемента In[] в полученном массиве объектов In.
- Присвойте результату поиска значение "Out", смещению элемента поиска относительно положения входного элемента - "InOutPos", а количеству значений поиска - "Num".. Если Num больше 1, значение InOutPos становится наименьшим номером элемента в полученном значении.
- Оценка соотношения размеров между значениями типов данных, отличных от целых и действительных чисел, представлена в таблице ниже:

Тип данных	Соотношение размеров
TIME	Значение с большей величиной оценивается как большое
DATE, TOD, DT	Последняя дата или время считаются большими.

Тип данных	Соотношение размеров
STRING	Сначала сравните кодировки первых символов в каждой строке. Если кодировка символов отличается, то размер кодировки символов равен размеру строки. Если кодировка первых символов совпадает, продолжите сравнивать второй и третий символы, пока не будет найдена отличающаяся кодировка. Если длина каждой строки различается, в конец более короткой строки будет добавлен символ NULL (16#00), исходя из длины более длинной строки, а затем произведено сравнение.

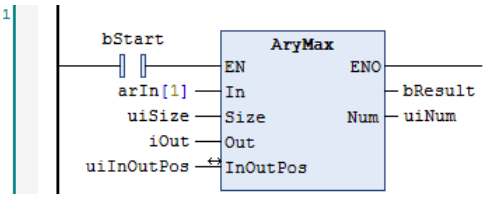
(4) Примечание

- Значение "Size" не должно превышать область массива в In[].
- In[] должен быть задан в качестве одномерного массива.
- Убедитесь, что "Out" и элементы полученного массива объектов In[] имеют одинаковый тип данных.
- Получение номера элемента "InOutPos" в качестве смещения положения входного элемента.
- Если In[] — действительное число, ожидаемый результат может быть не получен из-за влияния ошибок.
- Если значение "Size" равно 0, значение "Num" становится равным 0, то значения "InOutPos" и "Out" остаются неизменными, а функция возвращает значение TRUE.
- Если In[] — тип данных, который не поддерживается, его невозможно выполнить правильно.
- Если In[] и "Out" имеют тип данных STRING, размер области In[] не должен превышать размер "Out". В противном случае, функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Если In[] и "Out" имеют тип данных STRING, то допустимый диапазон поддерживает максимум 1985 символов. В противном случае, возвращаемое значение будет FALSE.
- Для In[] требуются целые числа, действительные числа и типы данных. В противном случае, возвращаемое значение будет FALSE, а "InOutPos" и "Out" останутся неизменными.

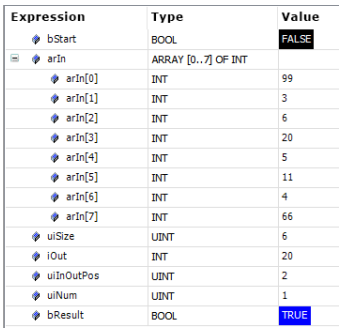
(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    bStart:BOOL;
    arIn:ARRAY[0..7]OF INT:=[99,3,6,20,5,11,4,66];
    uiSize    :UINT:=6;
    iOut      :INT;
    uiInOutPos:UINT;
    uiNum     :UINT;
    bResult   :BOOL;
END_VAR
```




```
IF bStart THEN
    bResult:=AryMax(In:= arIn[1],
        Size:= uiSize,
        Out:= iOut,
        InOutPos:= uiInOutPos,
        Num=> uiNum);
    bStart:=FALSE;
END_IF
```



Expression	Type	Value
bStart	BOOL	FALSE
arIn	ARRAY [0..7] OF INT	
arIn[0]	INT	99
arIn[1]	INT	3
arIn[2]	INT	6
arIn[3]	INT	20
arIn[4]	INT	5
arIn[5]	INT	11
arIn[6]	INT	4
arIn[7]	INT	66
uiSize	UINT	6
iOut	INT	20
uiInOutPos	UINT	2
uiNum	UINT	1
bResult	BOOL	TRUE

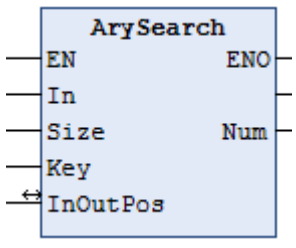
1-4-2-3. Получение массива ArySearch

(1) Обзор команды

Получение указанного значения в одномерном массиве.

Получение массива [ArySearch]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ArySearch	Получение массива		




```
Out:=ArySearch(In:= ,
        Size:= ,
        Key:= ,
        InOutPos:= ,
        Num=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In[]	Получение массива объектов	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/LINT/ REAL/LREAL/TIME/DATE/ TOD/DT/STRING	В качестве типа данных	(*)	Получение массива объектов
Size	Получение информации о количестве элементов в массиве объектов	UINT	В качестве типа данных	1	Получение информации о количестве элементов в массиве объектов
Key	Поиск ключевых слов	Соответствует типу данных элемента In[]	В качестве типа данных	(*)	Получение значения
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Num	Количество полученных значений	UINT	В качестве типа данных	0	Количество полученных значений
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
InOutPos	Получение номеров элементов	UINT	В качестве типа данных	-	Полученное значение является значением смещения положения входного элемента.

(3) Функция

- Получение элементов с тем же значением, что полученное ключевое слово "Key", из элементов массива "Size", начиная с элемента In[] в полученном одномерном массиве объектов.
- В таблице ниже представлены значения для получения номера элемента "InOutPos" и количество полученных значений "Num":

Существование элементов с тем же значением, что и у "Key"	“InOutPos”	“Num”
Существуют	Смещение положения входа самого нижнего элемента в элементе с тем же значением, что и у "Key"	Количество элементов с тем же значением, что и у "Key"
Не существуют	Не меняется	0

- Оценка соотношения размеров между значениями типов данных, отличных от целых и действительных чисел, представлена в таблице ниже:

Тип данных	Соотношение размеров
TIME	Значение с большей величиной оценивается как большое
DATE	Последняя считается большой.
STRING	Сначала сравните кодировки первых символов в каждой строке. Если кодировка символов отличается, то размер кодировки символов равен размеру строки. Если кодировка первых символов совпадает, продолжите сравнивать второй и третий символы, пока не будет найдена отличающаяся кодировка. Если длина каждой строки различается, в конец более короткой строки будет добавлен символ NULL (16#00), исходя из длины более длинной строки, а затем произведено сравнение.

(4) Примечание

- Значение "Size" не должно превышать область массива в In[].
- In[] должен быть задан в качестве одномерного массива.
- Убедитесь, что "Key" и элементы полученного массива объектов In[] имеют одинаковый тип данных, в противном случае, функция возвращает значение FALSE.

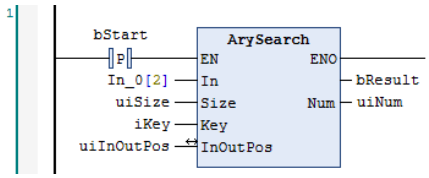
- Убедитесь, что входным параметром для "Key" является переменная.
- Получение номера элемента "InOutPos" в качестве смещения положения входного элемента.
- Если значение "Size" равно 0, значение "Num" инициализируется равным 0, то значение "InOutPos" остается неизменным, а функция возвращает значение TRUE.
- Если In[] и "Key" имеют тип данных STRING, то допустимый диапазон поддерживает максимум 1985 символов. Если допустимый диапазон превышен, функция возвращает значение FALSE, а значение "Num" инициализируется равным 0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    bStart     :BOOL;
    In_0       :ARRAY[0..9] OF INT:=[0,1,2,3,4,5,6,7,8,9];
    uiSize     :UINT:=7;
    iKey       :INT:=5;
    uiInOutPos:UINT;
    uiNum      :UINT;
    bResult    :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



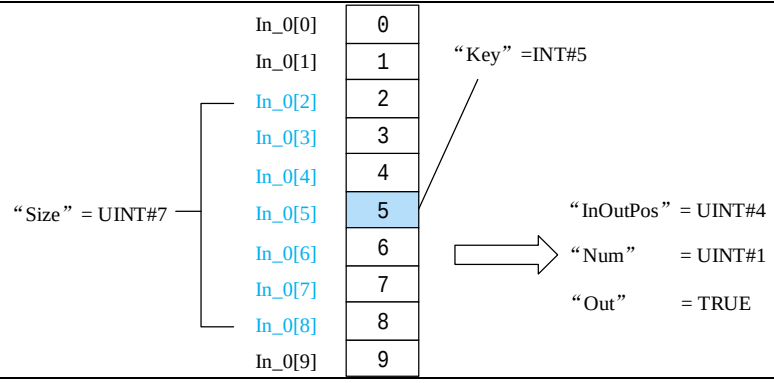
</td><td>

```
IF bStart THEN
    bResult:=ArySearch(In:= In_0[2],
        Size:= uiSize,
        Key:= iKey,
        InOutPos:= uiInOutPos,
        Num=> uiNum);
    bStart:=FALSE;
END_IF
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
bStart	BOOL	TRUE
In_0	ARRAY [0..9] OF INT	
In_0[0]	INT	0
In_0[1]	INT	1
In_0[2]	INT	2
In_0[3]	INT	3
In_0[4]	INT	4
In_0[5]	INT	5
In_0[6]	INT	6
In_0[7]	INT	7
In_0[8]	INT	8
In_0[9]	INT	9
uiSize	UINT	7
iKey	INT	5
uiInOutPos	UINT	3
uiNum	UINT	1
bResult	BOOL	TRUE

</td></tr>
<tr><td>Анализ принципа</td><td colspan="2">



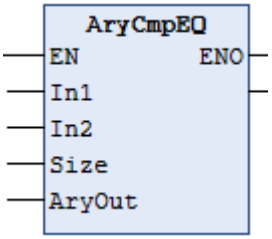
</td></tr>
</table>

1-4-2-4. Пакетное сравнение массивов EQ AryCmpEQ

(1) Обзор команды

Сравните, равны ли элементы двух массивов.

Пакетное сравнение массивов EQ [AryCmpEQ]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpEQ	Пакетное сравнение массивов EQ		bResult:=AryCmpEQ(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива1
Массив In2[]	Сравнение массива2	Массив с тем же типом данных, как и у In1[]	В качестве типа данных	-	Сравнение массива2
Size	Номера элементов сравнения	UINT	В качестве типа данных	1	Номера элементов сравнения

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните элементы Size, начиная с In1[], и элементы Size, начиная с In2[], в двух массивах, чтобы проверить, равны ли они. Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с AryOut[].
- Если In1[i]=In2[i], AryOut[i] имеет значение True, в противном случае — FALSE.

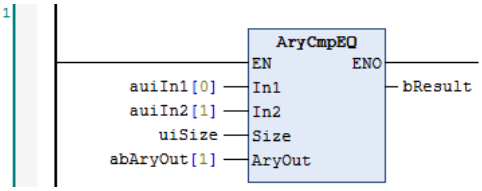
(4) Примечание

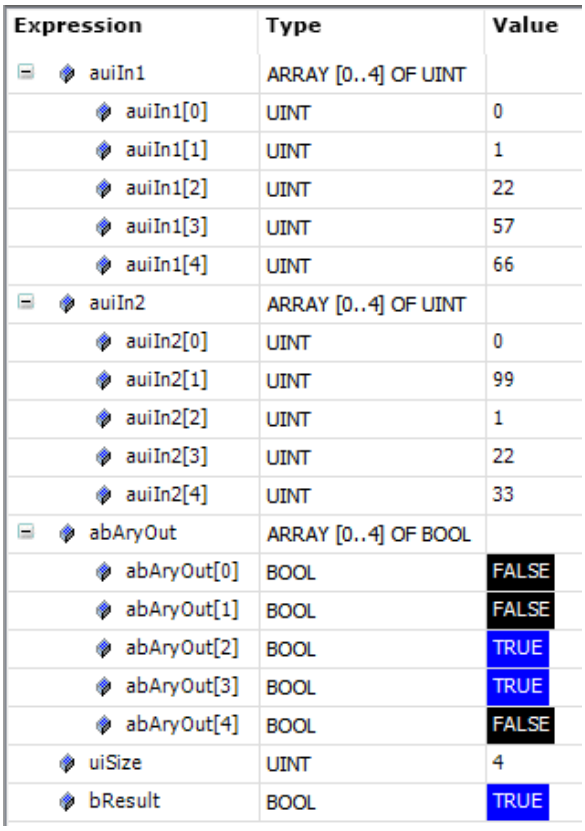
- Типы данных In1[] и In2[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2[] являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].

(5) Пример программы

ДЕМО	LD	ST

Задание переменных	VAR
	auiIn1 :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
	auiIn2 :ARRAY[0..4] OF UINT:=[0,99,1,22,33];
	abAryOut:ARRAY[0..4] OF BOOL;
	uiSize :UINT:=4;
	bResult :BOOL;
	END_VAR	

Пример программы		1 bResult:=AryCmpEQ(In1:= auiIn1[0], 2 In2:= auiIn2[1], 3 Size:= uiSize, 4 AryOut:= abAryOut[1]);

Результат операции	

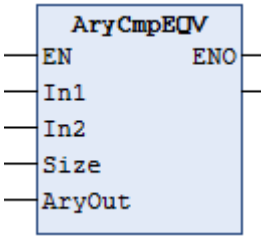
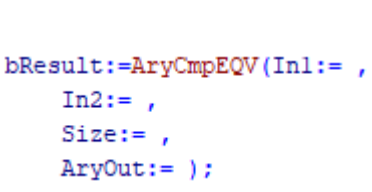
Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
auiIn2	ARRAY [0..4] OF UINT	
auiIn2[0]	UINT	0
auiIn2[1]	UINT	99
auiIn2[2]	UINT	1
auiIn2[3]	UINT	22
auiIn2[4]	UINT	33
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	TRUE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	4
bResult	BOOL	TRUE

1-4-2-5. Сравнение элементов массива EQV AryCmpEQV

(1) Обзор команды

Сравните, равны ли элементы двух массивов.

Сравнение элементов массива EQV [AryCmpEQV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpEQV	Сравнение элементов массива EQV		


```
bResult:=AryCmpEQV(In1:= ,
    In2:= ,
    Size:= ,
    AryOut:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива
Массив In2[]	Сравнение переменной	Тот же тип данных, как у элемента In1[]	В качестве типа данных	-	Сравнение переменной
Size	Сравнение номеров элементов	UINT	В качестве типа данных	1	Сравнение номеров элементов
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

Примечание: In2 также может быть задан в качестве переменной.

(3) Функция

- Сравните последовательные элементы Size, начиная с массива In1[] и переменной "In2", чтобы проверить, равны ли они (или сравните указанные элементы массива In1 и массива In2). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с AryOut[].
- Если In1[i]=In2, AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

- Типы данных In1[] и In2 должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2 являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться

от ожидаемых из-за ошибок округления.

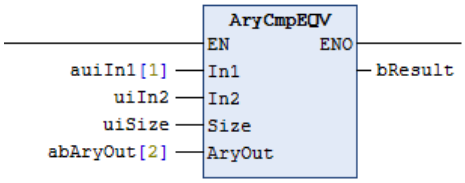
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[] или AryOut[].
- Для In1[] и In2[] необходимо использовать булевы, битовые, целые или действительные числа, в противном случае функция выполнения возвращает FALSE, а AryOut[] остается неизменным.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    uiIn2   :UINT:=57;
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=3;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=AryCmpEQV(In1:= auiIn1[1],
2       In2:= uiIn2,
3       Size:= uiSize,
4       AryOut:= abAryOut[2]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
uiIn2	UINT	57
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	FALSE
abAryOut[4]	BOOL	TRUE
uiSize	UINT	3
bResult	BOOL	TRUE

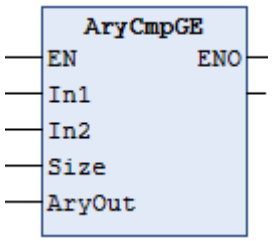
</td></tr>
</table>

1-4-2-6. Пакетное сравнение массивов GE AryCmpGE

(1) Обзор команды

Сравните размер каждого элемента в двух массивах (больше или равно).

Пакетное сравнение массивов GE [AryCmpGE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpGE	Пакетное сравнение массивов GE		bResult:=AryCmpGE(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	USINT/UINT/UDINT/ULINT / SINT/INT/DINT/LINT/REAL / LREAL	В качестве типа данных	-	Сравнение массива1
Массив In2[]	Сравнение массива2	Массив с тем же типом данных, как и у In1[]	В качестве типа данных	-	Сравнение массива2
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните последовательные элементы Size, начиная с In1[], и последовательные элементы Size, начиная с In2[], в двух массивах. Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1[i]>=In2[i], AryOut[i] имеет значение True, в противном случае — FALSE.

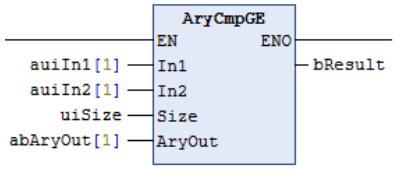
(4) Примечание

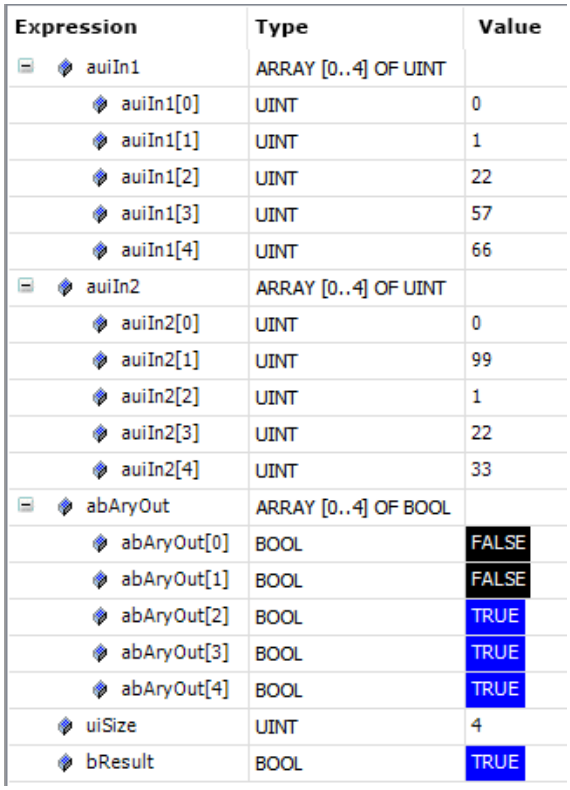
- Типы данных In1[] и In2[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2[] являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].

(5) Пример программы

ДЕМО	LD	ST

Задание переменных	VAR
	auiIn1 :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
	auiIn2 :ARRAY[0..4] OF UINT:=[0,99,1,22,33];
	abAryOut:ARRAY[0..4] OF BOOL;
	uiSize :UINT:=4;
	bResult :BOOL;
	END_VAR	

Пример программы		1 bResult:=AryCmpGE(In1:= auiIn1[1], 2 In2:= auiIn2[1], 3 Size:= uiSize, 4 AryOut:= abAryOut[1]);

Результат операции	

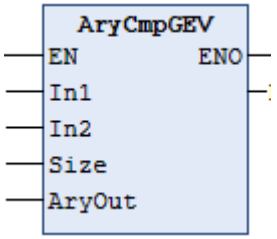
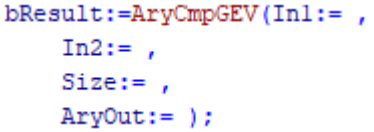
Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
auiIn2	ARRAY [0..4] OF UINT	
auiIn2[0]	UINT	0
auiIn2[1]	UINT	99
auiIn2[2]	UINT	1
auiIn2[3]	UINT	22
auiIn2[4]	UINT	33
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	TRUE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	TRUE
uiSize	UINT	4
bResult	BOOL	TRUE

1-4-2-7. Сравнение элементов массива GEV AryCmpGEV

(1) Обзор команды

Сравните размер каждого элемента и переменной в массиве (больше или равно).

Сравнение элементов массива GEV [AryCmpGEV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpGEV	Сравнение элементов массива GEV		


```
bResult:=AryCmpGEV(In1:= ,
    In2:= ,
    Size:= ,
    AryOut:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива
Массив In2[]	Сравнение переменной	Тот же тип данных, как у элемента In1[]	В качестве типа данных	-	Сравнение переменной
Size	Сравнение номеров элементов	UINT	В качестве типа данных	1	Сравнение номеров элементов
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

Примечание: In2 также может быть задан в качестве переменной.

(3) Функция

- Сравните последовательные элементы Size, начиная с двух массивов In1[] и переменной "In2" (больше или равно). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1[i]>=In2, AryOut[i] имеет значение True, в противном случае — FALSE.

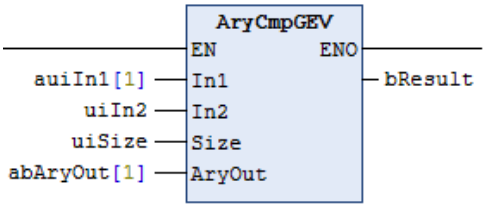
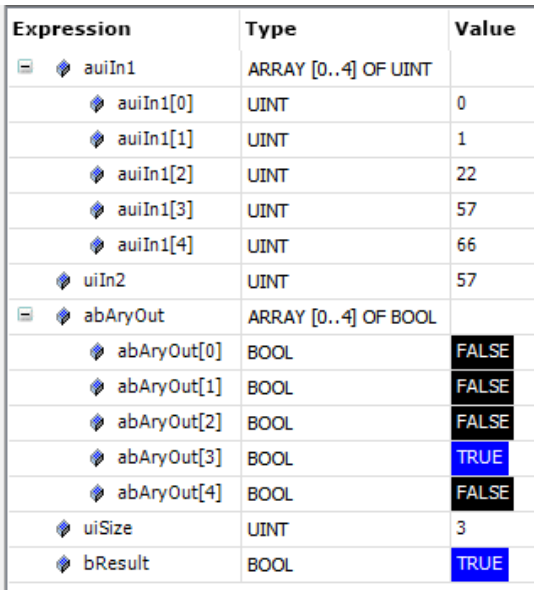
(4) Примечание

- Типы данных In1[] и In2 должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2 являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[] или AryOut[].

(5) Пример программы

ДЕМО	LD	ST

Задание переменных	VAR
	auiIn1 :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
	uiIn2 :UINT:=57;
	abAryOut:ARRAY[0..4] OF BOOL;
	uiSize :UINT:=3;
	bResult :BOOL;
	END_VAR	

Пример программы		1 bResult:=AryCmpGEV(In1:= auiIn1[1], 2 In2:= uiIn2, 3 Size:= uiSize, 4 AryOut:= abAryOut[1]);
Результат операции		

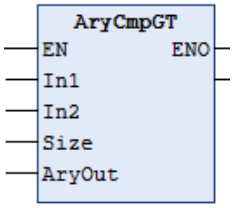
Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
uiIn2	UINT	57
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	3
bResult	BOOL	TRUE

1-4-2-8. Пакетное сравнение массивов GT AryCmpGT

(1) Обзор команды

Сравните размер каждого элемента в двух массивах (больше).

Пакетное сравнение массивов GT [AryCmpGT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpGT	Пакетное сравнение массивов GT		bResult:=AryCmpGT(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива1
Массив In2[]	Сравнение массива2	Массив с тем же типом данных, как и у In1[]	В качестве типа данных	-	Сравнение массива2
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните последовательные элементы Size в In1[] и In2[] (больше). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1 [i]>In2 [i], AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

- Типы данных In1[] и In2[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.

- Если In1[] и In2[] являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].

(5) Пример программы

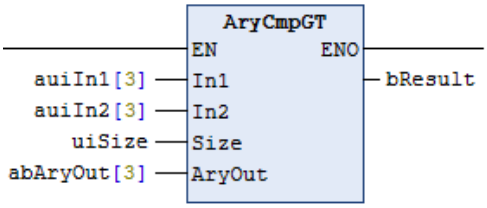
ДЕМО

Задание переменных

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    auiIn2  :ARRAY[0..4] OF UINT:=[0,99,1,22,33];
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=2;
    bResult :BOOL;
END_VAR
```

LD

Пример программы



ST

```
1  bResult:=AryCmpGT(In1:= auiIn1[3],
2      In2:= auiIn2[3],
3      Size:= uiSize,
4      AryOut:= abAryOut[3]);
```

Результат операции

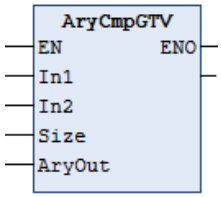
Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
auiIn2	ARRAY [0..4] OF UINT	
auiIn2[0]	UINT	0
auiIn2[1]	UINT	99
auiIn2[2]	UINT	1
auiIn2[3]	UINT	22
auiIn2[4]	UINT	33
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	TRUE
uiSize	UINT	2
bResult	BOOL	TRUE

1-4-2-9. Сравнение элементов массива GTV AryCmpGTV

(1) Обзор команды

Сравните размер каждого элемента и переменной в массиве (больше).

Сравнение элементов массива GTV [AryCmpGTV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpGTV	Сравнение элементов массива GTV		bResult:=AryCmpGTV(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива
Массив In2[]	Сравнение массива2	Тот же тип данных, как у элемента In1[]	В качестве типа данных	-	Сравнение переменной
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните последовательные элементы Size, начиная с In1[] и переменной "In2" (больше). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1 [i]>In2, AryOut [i] имеет значение True, в противном случае — FALSE.

(4) Примечание

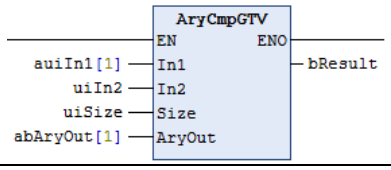
- Типы данных In1[] и In2 должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2 являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[] или AryOut[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    uiIn2   :UINT:=20;
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=3;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=AryCmpGTV(In1:= auiIn1[1],
2      In2:= uiIn2,
3      Size:= uiSize,
4      AryOut:= abAryOut[1]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
uiIn2	UINT	20
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	TRUE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	3
bResult	BOOL	TRUE

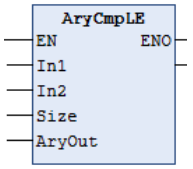
</td></tr>
</table>

1-4-2-10. Пакетное сравнение массивов LE AryCmpLE

(1) Обзор команды

Сравните размер каждого элемента в двух массивах (меньше или равно).

Пакетное сравнение массивов LE [AryCmpLE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpLE	Пакетное сравнение массивов LE		bResult:=AryCmpLE(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива1
Массив In2[]	Сравнение массива2	Массив с тем же типом данных, как и у In1[]	В качестве типа данных	-	Сравнение массива2
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните последовательные элементы Size, начиная с In1[], и последовательные элементы Size, начиная с In2[] (меньше или равно). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut [].
- Если In1[i]<=In2[i], AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

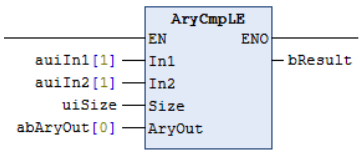
- Типы данных In1[] и In2[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2[] являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    auiIn2  :ARRAY[0..4] OF UINT:=[0,99,1,22,33];
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=4;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=AryCmpLE(In1:= auiIn1[1],
2      In2:= auiIn2[1],
3      Size:= uiSize,
4      AryOut:= abAryOut[0]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
auiIn2	ARRAY [0..4] OF UINT	
auiIn2[0]	UINT	0
auiIn2[1]	UINT	99
auiIn2[2]	UINT	1
auiIn2[3]	UINT	22
auiIn2[4]	UINT	33
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	TRUE
abAryOut[1]	BOOL	FALSE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	FALSE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	4
bResult	BOOL	TRUE

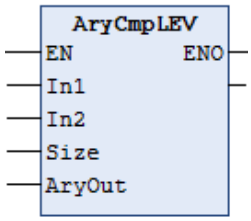
</td></tr>
</table>

1-4-2-11. Сравнение элементов массива LEV AryCmpLEV

(1) Обзор команды

Сравните размер каждого элемента и переменной в массиве (меньше или равно).

Сравнение элементов массива LEV [AryCmpLEV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpLEV	Сравнение элементов массива LEV		bResult:=AryCmpLEV(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива
Массив In2[]	Сравнение переменной	Тот же тип данных, как у элемента In1[]	В качестве типа данных	-	Сравнение переменной
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

Примечание: In2 также может быть задан в качестве переменной.

(3) Функция

- Сравните последовательные элементы Size, начиная с массива сравнения In1[] и переменной "In2" (меньше или равно). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1[i]<=In2, AryOut[i] имеет значение True, в противном случае — FALSE.

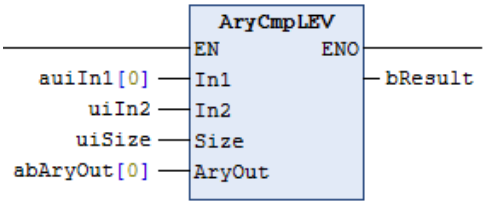
(4) Примечание

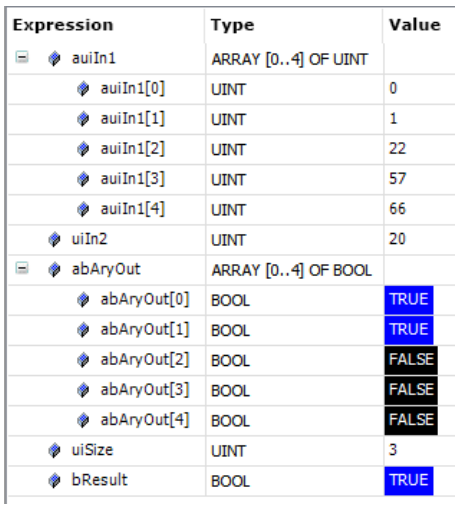
- Типы данных In1[] и In2 должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2 являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[] или AryOut[].

(5) Пример программы

ДЕМО	LD	ST

Задание переменных	VAR
	auiIn1 :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
	uiIn2 :UINT:=20;
	abAryOut:ARRAY[0..4] OF BOOL;
	uiSize :UINT:=3;
	bResult :BOOL;
	END_VAR	

Пример программы		1 bResult:=AryCmpLEV(In1:= auiIn1[0], 2 In2:= uiIn2, 3 Size:= uiSize, 4 AryOut:= abAryOut[0]);

Результат операции	

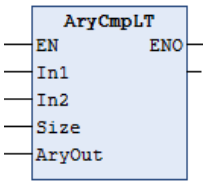
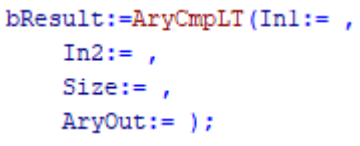
Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
uiIn2	UINT	20
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	TRUE
abAryOut[1]	BOOL	TRUE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	FALSE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	3
bResult	BOOL	TRUE

1-4-2-12. Пакетное сравнение массивов LT AryCmpLT

(1) Обзор команды

Сравните размер каждого элемента в двух массивах (меньше).

Пакетное сравнение массивов LT [AryCmpLT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpLT	Пакетное сравнение массивов LT		


```
bResult:=AryCmpLT(In1:= ,
    In2:= ,
    Size:= ,
    AryOut:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива1

Массив In2[]	Сравнение массива2	Массив с тем же типом данных, как и у In1[]	В качестве типа данных	-	Сравнение массива2
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните последовательные элементы Size, начиная с In1[], и последовательные элементы Size, начиная с In2[] (меньше). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1[i]<In2[i], AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

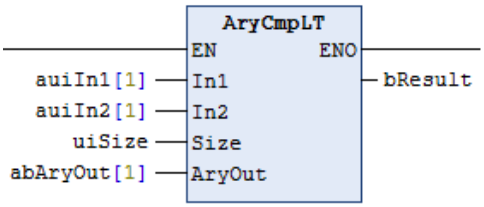
- Типы данных In1[] и In2[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2[] являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    auiIn2  :ARRAY[0..4] OF UINT:=[0,99,1,22,33];
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=4;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=AryCmpLT(In1:= auiIn1[1],
2      In2:= auiIn2[1],
3      Size:= uiSize,
4      AryOut:= abAryOut[1]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
auiIn2	ARRAY [0..4] OF UINT	
auiIn2[0]	UINT	0
auiIn2[1]	UINT	99
auiIn2[2]	UINT	1
auiIn2[3]	UINT	22
auiIn2[4]	UINT	33
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	TRUE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	FALSE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	4
bResult	BOOL	TRUE

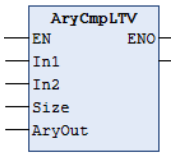
</td></tr>
</table>

1-4-2-13. Сравнение элементов массива LTV AryCmpLTV

(1) Обзор команды

Сравните размер каждого элемента и переменной в массиве (меньше).

Сравнение элементов массива LTV [AryCmpLTV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpLTV	Сравнение элементов массива LT		bResult:=AryCmpLTV(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива
Массив In2[]	Сравнение переменной	Тот же тип данных, как у элемента In1[]	В качестве типа данных	-	Сравнение переменной
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

Примечание: In2 также может быть задан в качестве переменной.

(3) Функция

- Сравните последовательные элементы Size, начиная с массива In1[0] и переменной сравнения "In2" (меньше). Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1[i]<In2, AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

- Типы данных In1[] и In2 должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2 являются действительными числами и включают в себя

неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.

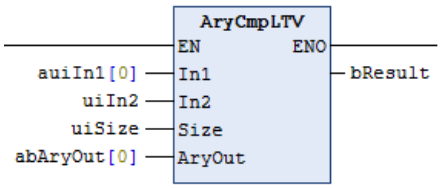
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[] или AryOut[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    uiIn2   :UINT:=20;
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=3;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=AryCmpLTV(In1:= auiIn1[0],
2      In2:= uiIn2,
3      Size:= uiSize,
4      AryOut:= abAryOut[0]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
uiIn2	UINT	20
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	TRUE
abAryOut[1]	BOOL	TRUE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	FALSE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	3
bResult	BOOL	TRUE

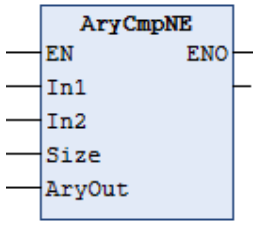
</td></tr>
</table>

1-4-2-14. Пакетное сравнение массивов NE AryCmpNE

(1) Обзор команды

Сравните, отличаются ли элементы двух массивов.

Пакетное сравнение массивов NE [AryCmpNE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpNE	Пакетное сравнение массивов NE		bResult:=AryCmpNE(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива1	BOOL/BYTE/WORD/ DWORD/LWORD/USINT/ UINT/UDINT/ULINT/SINT/ INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Сравнение массива1
Массив In2[]	Сравнение массива2	Массив с тем же типом данных, как и у In1[]	В качестве типа данных	-	Сравнение массива2
Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

(3) Функция

- Сравните последовательные элементы Size, начиная с массива In1[], и последовательные элементы Size, начиная с массива In2[]. Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut[].
- Если In1[i]<>In2[i], AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

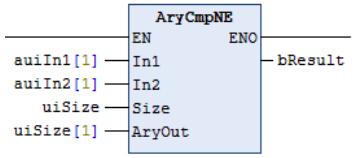
- Типы данных In1[] и In2[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Для In1[] и In2[] необходимо использовать булевы, битовые, целые или действительные числа, в противном случае возвращаемое значение будет FALSE, а AryOut[] остается неизменным.
- Если In1[] и In2[] являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[], In2[] или AryOut[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,22,57,66];
    auiIn2  :ARRAY[0..4] OF UINT:=[0,99,22,45,33];
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=4;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=AryCmpNE(In1:= auiIn1[1],
2      In2:= auiIn2[1],
3      Size:= uiSize,
4      AryOut:= abAryOut[1]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	22
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
auiIn2	ARRAY [0..4] OF UINT	
auiIn2[0]	UINT	0
auiIn2[1]	UINT	99
auiIn2[2]	UINT	22
auiIn2[3]	UINT	45
auiIn2[4]	UINT	33
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	TRUE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	TRUE
uiSize	UINT	4
bResult	BOOL	TRUE

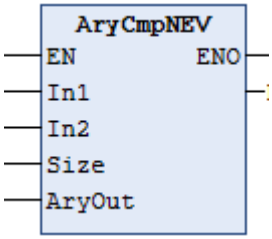
</td></tr>
</table>

1-4-2-15. Сравнение элементов массива NEV AryCmpNEV

(1) Обзор команды

Сравните, отличаются ли элементы двух массивов.

Сравнение элементов массива NEV [AryCmpNEV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryCmpNEV	Сравнение элементов массива NEV		AryCmpNEV (In1 :=, In2 :=, Size := , AryOut :=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In1[]	Сравнение массива	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/LINT/ REAL/LREAL	В качестве типа данных	-	Сравнение массива
Массив In2[]	Сравнение переменной	Тот же тип данных, как у элемента In1[]	В качестве типа данных	-	Сравнение переменной

Size	Сравнение номера элемента	UINT	В качестве типа данных	1	Сравнение номера элемента
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Массив с результатами сравнения	BOOL	В качестве типа данных	-	Массив с результатами сравнения

Примечание: In2 также может быть задан в качестве переменной.

(3) Функция

- Сравните последовательные элементы Size массива сравнения In1[] и переменной сравнения "In2", чтобы проверить, равны ли они. Результаты сравнения сохраняются в соответствующих номерах элементов последовательных элементов Size, начиная с массива с результатами сравнения AryOut [].
- Если In1[i]<>In2, AryOut[i] имеет значение True, в противном случае — FALSE.

(4) Примечание

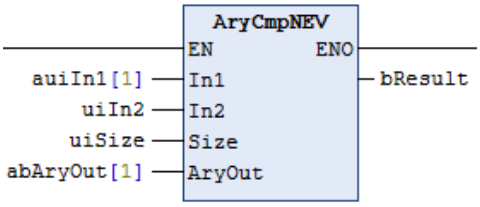
- Типы данных In1[] и In2 должны совпадать, в противном случае возвращаемое значение будет FALSE, а AryOut[] останется неизменным.
- Если In1[] и In2 являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Значение "Size" не должно превышать любую область массива в In1[] или AryOut[].
- Для In1[] и In2[] необходимо использовать булевы, битовые, целые или действительные числа, в противном случае возвращаемое значение будет FALSE, а AryOut[] остается неизменным.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    auiIn1  :ARRAY[0..4] OF UINT:=[0,1,20,57,66];
    uiIn2   :UINT:=20;
    abAryOut:ARRAY[0..4] OF BOOL;
    uiSize  :UINT:=3;
    bResult :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
bResult:=AryCmpNEV(In1:= auiIn1[1],
    In2:= uiIn2,
    Size:= uiSize,
    AryOut:= abAryOut[1]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
auiIn1	ARRAY [0..4] OF UINT	
auiIn1[0]	UINT	0
auiIn1[1]	UINT	1
auiIn1[2]	UINT	20
auiIn1[3]	UINT	57
auiIn1[4]	UINT	66
uiIn2	UINT	20
abAryOut	ARRAY [0..4] OF BOOL	
abAryOut[0]	BOOL	FALSE
abAryOut[1]	BOOL	TRUE
abAryOut[2]	BOOL	FALSE
abAryOut[3]	BOOL	TRUE
abAryOut[4]	BOOL	FALSE
uiSize	UINT	3
bResult	BOOL	TRUE

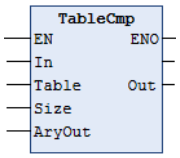
</td></tr>
</table>

1-4-2-16. Сравнение таблиц TableCmp

(1) Обзор команды

Сравните несколько заданных интервалов, указанных в данных сравнения и таблице сравнения.

Сравнение таблиц [TableCmp]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
TableCmp	Сравнение таблиц		bResult:=TableCmp(In:= , Table:= , Size:= , AryOut:= , Out=>);

(2) Связанные переменные

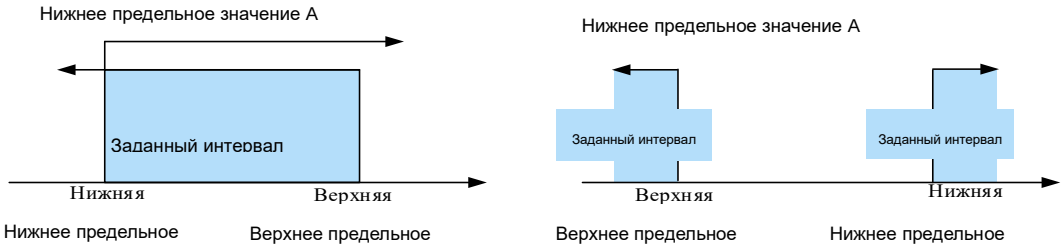
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Сравнение данных	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL	В качестве типа данных	-	Сравниваемое значение
Массив Table[] 2D	Сравнение таблицы	Двумерный массив с элементами, имеющими тот же тип данных, как и у In	В качестве типа данных	-	Двумерный массив с заданными интервалами в виде элементов
Size	Сравнение размера	UINT	В качестве типа данных	1	Количество элементов в Table[] по сравнению с "In"
Массив AryOut[]	Массив с индивидуальными результатами сравнения	BOOL	[FALSE,TRUE]	FALSE	Результаты сравнения для каждого элемента в Table[] True: Непротиворечивые FALSE: Противоречивые
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Возвращаемое значение	BOOL	[FALSE,TRUE]	FALSE	Результаты сравнения

(3) Функция

- Сравните заданный интервал группы “Size” в данных сравнения "In" и таблице сравнения [].
- Table[] — двумерный массив, где одномерный элемент — это номер заданного интервала, нулевой элемент в двумерном массиве представляет нижнее предельное значение A заданного интервала, а первый элемент представляет верхнее предельное значение B заданного интервала.

Заданные интервалы	Нижнее предельное значение A интервала	Верхнее предельное значение B интервала
Интервал 0	Таблица [0,0]	Таблица [0,1]
Интервал 1	Таблица [1,0]	Таблица [1,1]
...	...	...
Интервал “Size”-1	Таблица [“Size”-1,0]	Таблица [“Size”-1,1]

- Метод опредления заданного интервала с помощью нижнего предельного значения A и верхнего предельного значения B интервала заключается в следующем. Нижнее предельное значение A и верхнее предельное значение B интервала включают в заданный интервал.



- Результаты сравнения "In" и Table[] сохраняются в массиве с индивидуальными результатами сравнения AryOut[]. Если "In" находится в пределах i-го заданного интервала, то значение AryOut[i] равно True, в противном случае - FALSE. Если все элементы "Size" в AryOut[] истинны, то значение результата сравнения "Out" истинно, в противном случае, оно ложно.

(4) Примечание

- Типы данных элементов "In" и Table[] должны совпадать, в противном случае возвращаемое значение функции будет FALSE.
- Убедитесь, что для Table[] используется двумерный массив, и размер двумерного массива равен 2, иначе выходной результат будет неожиданным.

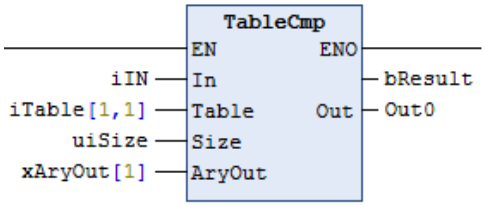
- Если размер массива AryOut[] больше "Size", результаты сравнения будут сохранены в соответствующие номера элементов последовательного элемента Size в начале массива AryOut[]. Другие элементы массива останутся неизменными.
- При сравнении действительных чисел и включении результатов деления, которые нельзя разделить полностью, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Если значение "Size" равно 0, значение "Out" равно FALSE, AryOut[] остается неизменным, а функция возвращает значение TRUE.
- Если значение "Size" превышает размер массива AryOut[] или одномерного массива Table[], это может привести к исключениям при выполнении программы и даже к сбоям в работе ПЛК.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    iIN     :INT:=5;
    iTable  :ARRAY[1..3,1..2] OF INT:=[0,1,4,10,10,20];
    uiSize  :UINT:=3;
    xAryOut:ARRAY[1..4] OF BOOL;
    Out0    :BOOL;
    bResult:BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=TableCmp(In:= iIN,
2      Table:= iTable[1,1],
3      Size:= uiSize,
4      AryOut:= xAryOut[1],
5      Out=> Out0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
iIN	INT	5
⊞ iTable	ARRAY [1..3, 1..2] O...	
uiSize	UINT	3
⊟ xAryOut	ARRAY [1..4] OF BOOL	
xAryOut[1]	BOOL	FALSE
xAryOut[2]	BOOL	TRUE
xAryOut[3]	BOOL	FALSE
xAryOut[4]	BOOL	FALSE
Out0	BOOL	FALSE
bResult	BOOL	TRUE

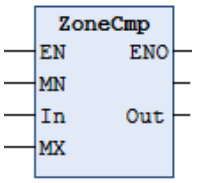
</td></tr>
</table>

1-4-2-17. Сравнение зоны ZoneCmp

(1) Обзор команды

Определите, находятся ли данные сравнения между указанными верхними и нижними пределами.

Сравнение зоны [ZoneCmp]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ZoneCmp	Сравнение зоны		bResult:=ZoneCmp(MN:= , In:= , MX:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
MN	Нижнее предельное значение	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/REAL/ LREAL/TIME/DATE/TOD/DT	В качестве типа данных	-	Нижнее предельное значение
In	Сравнение данных	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/REAL/ LREAL/TIME/DATE/TOD/DT	В качестве типа данных	-	Сравниваемое значение
MX	Верхнее предельное значение	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/REAL/ LREAL/TIME/DATE/TOD/DT	В качестве типа данных	1	Верхнее предельное значение

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Out	Результаты сравнения	BOOL	[FALSE,TRUE]	FALSE	Результаты сравнения

(3) Функция

- Определите, находятся ли данные сравнения "In" между верхним предельным значением "MX" и нижним предельным значением "MN".
- Если "MX"≥"In"≥"MN", значение "Out" является истинным, а остальные значения — ложными.
- Оценка соотношения размеров между значениями типов данных, отличных от целых и действительных чисел, представлена в таблице ниже:

Тип данных	Соотношение размеров
TIME	Значение с большей величиной оценивается как большое
DATA, TOD, DT	Последняя дата или время считаются большими.

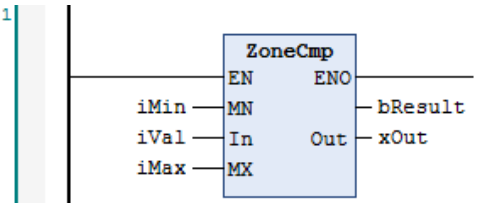
(4) Примечание

- Если типы данных "In", "MX" и "MN" разные, возвращаемое значение будет FALSE, а Out остается неизменным.
- Если "In", "MX" и "MN" являются действительными числами и включают в себя неисчерпаемые результаты деления, результаты обработки могут отличаться от ожидаемых из-за ошибок округления.
- Определите, чему равны значения в диапазоне +∞ или -∞.
- Если значение "MN" больше значения "MX", программа сталкивается с исключением, и функция возвращает значение FALSE.
- "MX" и "MN" не должны иметь нечисловые значения, в противном случае программа столкнется с исключением, и функция вернет значение FALSE.
- Для "In", "MX" и "MN" нужно использовать целые числа, действительные числа и даты. В противном случае возвращаемое значение будет FALSE, а Out останется неизменным.

(5) Пример программы

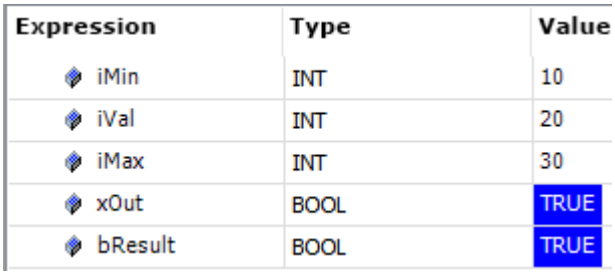
<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    iMin:INT:=10;
    iVal:INT:=20;
    iMax:INT:=30;
    xOut:BOOL;
    bResult:BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
bResult:=ZoneCmp(MN:= iMin,
    In:= iVal,
    MX:= iMax,
    Out=> xOut);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">


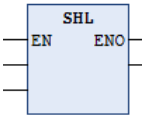
</td></tr>
</table>

1-4-3. Команды сдвига

1-4-3-1. Сдвиг влево SHL

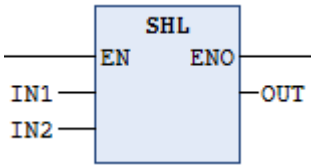
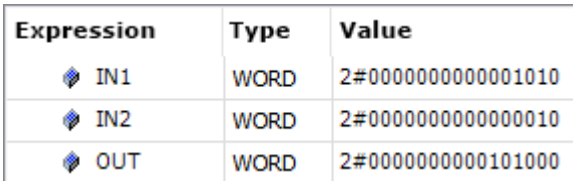
Побитовый сдвиг операнда влево. При этом левый крайний бит остается необработанным, а правый пустой бит автоматически заполняется нулем.

Сдвиг влево [SHL]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Название команды	Пиктограммы команд	Функция
SHL		SHL

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN1	Входные данные	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT	В качестве типа данных	0	Входные данные
IN2	Биты сдвига	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT	В качестве типа данных	0	Биты сдвига
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Данные после сдвига	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT	В качестве типа данных	0	Данные после сдвига

Эта команда может сдвигать данные на входе IN1 влево на биты IN2 и присваивать результат вывода переменной OUT. Сдвиг двоичного числа влево на один бит эквивалентен умножению исходного числа на 2. Если значение IN2 больше, чем ширина типа данных в IN1, значение IN1 будет заполнено нулем.

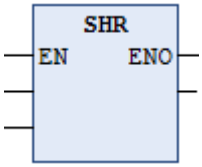
ДЕМО	LD	ST
Задание переменных		VAR IN1:WORD:=10; IN2:WORD:=2; OUT:WORD; END_VAR
Пример программы		1 OUT:=SHL(IN1,IN2);
Результат операции		

Expression	Type	Value
IN1	WORD	2#0000000000001010
IN2	WORD	2#0000000000000010
OUT	WORD	2#0000000000101000

1-4-3-2. Сдвиг вправо SHR

Побитовый сдвиг операнда вправо. При этом правый крайний бит остается необработанным,

Сдвиг вправо [SHR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Название команды	Пиктограммы команд	Функция
SHR		SHR

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN1	Входные данные	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT	В качестве типа данных	0	Входные данные
IN2	Биты сдвига	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT	В качестве типа данных	0	Биты сдвига
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Данные после сдвига	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT	В качестве типа данных	0	Данные после сдвига

Эта команда может сдвигать данные на входе IN1 вправо на биты IN2, присваивать результат вывода переменной OUT. Сдвиг двоичного числа вправо

на один бит эквивалентен делению исходного числа на 2. Если значение IN2 больше, чем ширина типа данных в IN1, значение IN1 будет заполнено нулем.

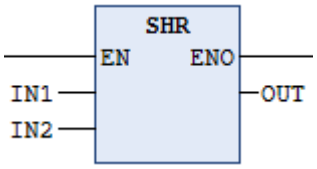
После сдвига беззнакового типа данных вправо, старший бит может быть заполнен нулем, поскольку знаковый бит отсутствует.

При обнаружении знаковых целочисленных данных старший бит нужно сдвинуть вправо, заполнив знаковый бит.

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN1:INT:=16#F100;
    IN2:WORD:=2;
    OUT:INT;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>

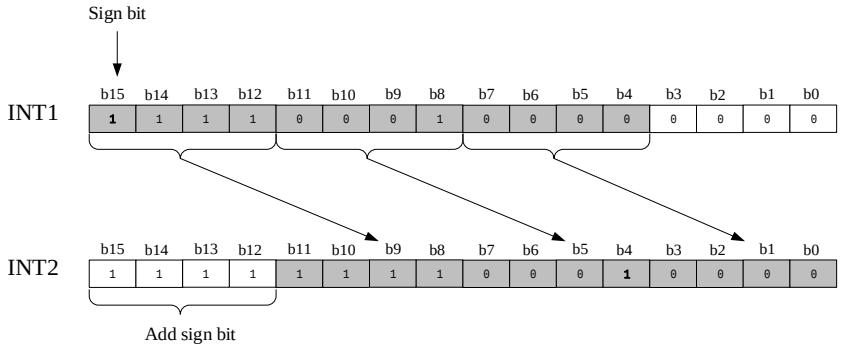


</td><td>

```
1  OUT:=SHR(IN1,IN2);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

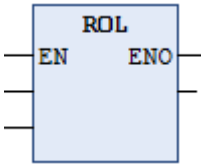
Expression	Type	Value
IN1	INT	2#1111000100000000
IN2	WORD	2#0000000000000010
OUT	INT	2#1111110001000000

</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Поскольку INT — это знаковые битовые данные с допустимыми значениями в диапазоне от -32768 до 32767, IN1 — это шестнадцатеричные знаковые данные F100, а старший бит b15 — это знаковый бит, то после сдвига на 4 бита вправо необходимо добавить данные. Поскольку знаковый бит исходных данных равен 1, старшие 4 бита складываются с четырьмя 1 (т.е. знаковыми битами). Таким образом, конечный результат работы программы — 16#FF10. Процесс сдвига показан на рисунке ниже:

</td></tr>
</table>

1-4-3-3. Поворот влево ROL

Поворачивайте операнд в побитовом цикле влево, и биты, удаленные слева, будут напрямую добавлены к младшему биту справа.

Поворот влево [ROL]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Название команды	Пиктограммы команд	Функция
ROL		ROL

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN1	Входные данные	BYTE/WORD/DWORD/LWORD / USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	0	Входные данные
IN2	Биты сдвига	BYTE/WORD/DWORD/LWORD / USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	0	Биты сдвига
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Данные после сдвига	BYTE/WORD/DWORD/LWORD / USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	0	Данные после сдвига

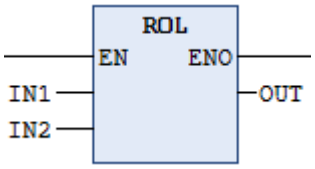
Эту команду можно использовать для циклического побитного переноса всего содержимого входа IN1 влево, при этом пустые биты заполняются сигнальным состоянием сдвинутых битов. Входной параметр IN2 представляет собой числовое значение, обозначающее количество битов для поворота, а OUT — результат операции поворота.

После сдвига беззнакового типа данных вправо, старший бит может быть заполнен нулем, поскольку знаковый бит отсутствует.

При обнаружении знаковых целочисленных данных старший бит нужно сдвинуть вправо, заполнив знаковый бит.

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN1:WORD:=16#AA00;
    IN2:WORD:=4;
    OUT:WORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>
</td><td>

```
1   OUT:=ROL(IN1,IN2);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

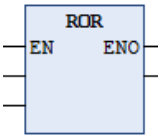
Expression	Type	Value
IN1	WORD	16#AA00
IN2	WORD	16#0004
OUT	WORD	16#A00A

</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Использование выхода OUT после цикла сдвига вправо не заполняет пустой бит нулем, а вводит 1010 из b12 - b15 во входные данные IN1 до битов от b0 до b3.</td></tr>
</table>

1-4-3-4. Поворот вправо ROR

Поворачивайте операнды в побитовом цикле вправо, и биты, удаленные справа, будут напрямую добавлены к старшему биту слева.

<table>
<tr><th colspan="4">Поворот вправо [ROR]</th></tr>
<tr><td>Условие выполнения</td><td>Запуск фронтом сигнала</td><td>Совместимые модели</td><td>PMP3xx</td></tr>
<tr><td>Прошивка</td><td>-</td><td>Программное обеспечение</td><td>-</td></tr>
<tr><td>Библиотека</td><td>-</td><td colspan="2"></td></tr>
</table>

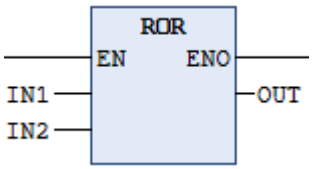
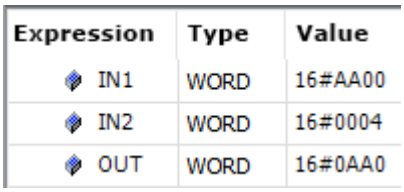
Название команды	Пиктограммы команд	Функция
ROR		ROR

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
IN1	Входные данные	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	0	Входные данные
IN2	Биты сдвига	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	0	Биты сдвига
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
OUT	Данные после сдвига	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT /INT/DINT/LINT	В качестве типа данных	0	Данные после сдвига

Эту команду можно использовать для циклического побитного переноса всего содержимого входа IN1 влево, при этом пустые биты заполняются сигнальным состоянием сдвинутых битов. Входной параметр IN2 представляет собой числовое значение, обозначающее количество битов для поворота, а OUT — результат операции поворота.

После сдвига беззнакового типа данных вправо, старший бит может быть заполнен нулем, поскольку знаковый бит отсутствует.

При обнаружении знаковых целочисленных данных старший бит нужно сдвинуть вправо, заполнив знаковый бит.

ДЕМО	LD	ST
Задание переменных	VAR IN1:WORD:=16#AA00; IN2:WORD:=4; OUT:WORD; END_VAR	
Пример программы		1 OUT:=ROR(IN1,IN2);
Результат операции		

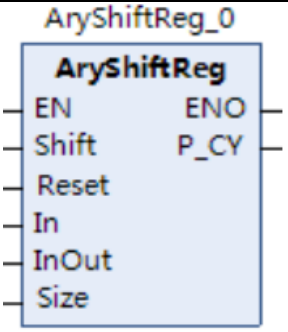
Expression	Type	Value
IN1	WORD	16#AA00
IN2	WORD	16#0004
OUT	WORD	16#0AA0

1-4-3-5. Сдвиговый регистр AryShiftReg

(1) Обзор команды

Сдвиг всех элементов массива на 1 бит влево и вставка входного значения в самое нижнее положение.

Сдвиговый регистр [AryShiftReg]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryShiftReg	Сдвиговый регистр		AryShiftReg_0(Shift:= , Reset:= , In:= , InOut:= , Size:= , P_CY=> , ENO=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Shift	Сдвиг	BOOL	[FALSE,TRUE]	FALSE	Выполнение сдвига при изменении с FALSE на True
Reset	Сброс	BOOL	[FALSE,TRUE]	FALSE	Выполнение сброса при изменении с FALSE на True
In	Входное значение	BOOL	[FALSE,TRUE]	FALSE	Значение, вставленное в самом нижнем положении InOut[]
Size	Количество элементов в массиве положения	UINT	В качестве типа данных	0	Количество элементов, используемых в сдвиговом регистре InOut[]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
P_CY	Флаг переноса	BOOL	В качестве типа данных	-	Значение, сохраненное в флаге переноса
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
InOut[]	Массив положения	BOOL/BYTE/WORD /DWORD/LWORD	В качестве типа данных	-	Массив положения

(3) Функция

- При изменении значения параметра "Shift" с FALSE на True элементы массива "Size", начиная с InOut[], установленного на выводе InOut, сдвигаются влево (в направлении старшего бита) на 1 бит.

- Вставьте входное значение "In" в самом нижнем положении. Старший бит, переполнивший массив, будет выведен во флаг переноса (CY) "P_CY".
- Если "Reset" равен true, все биты элементов "Size", начиная с InOut[], установленного на выводе InOut, и флаги переноса (CY) устанавливаются в FALSE.

(4) Примечание

- Если "Reset" равен true, массиву InOut[] нельзя присвоить значение, и даже если "Shift" изменится с FALSE на true, сдвига не произойдет.
- Если значение "Size" равно 0, InOut[] остается неизменным, а функция возвращает значение True.
- Если значение "Size" превысит область массива InOut[], ПЛК перестает выдавать исключение, как показано на рисунке ниже:



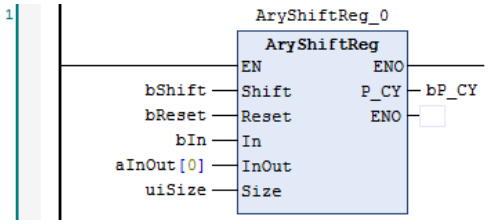
останов программа не изменилась исключение при загрузке программы

- Для InOut[] требуются булева и битовая строка, иначе при выполнении программы ENO будет ложным.

(5) Пример программы

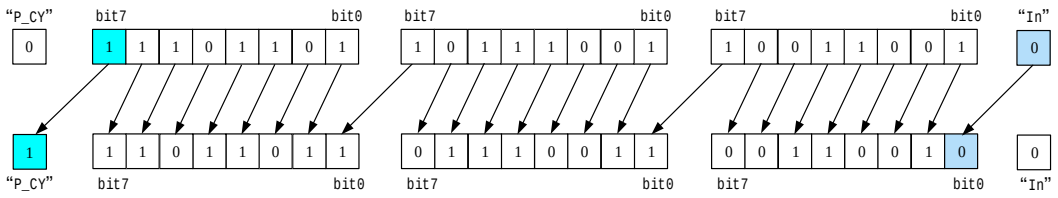
ДЕМО	LD	ST
Задание переменной		
Пример программы		
Результат операции		

```
PROGRAM AryShiftReg_ST
VAR
    AryShiftReg_0:AryShiftReg;
    bShift:BOOL;
    bReset:BOOL;
    bIn:BOOL;
    aInOut:ARRAY[0..3]OF BYTE:=[2#11100100,2#00110000,2#11111100,2#01100100];
    uiSize:UINT;
    bP_CY:BOOL;
END_VAR
```




```
AryShiftReg_0(
    Shift:= bShift,
    Reset:= bReset,
    In:= bIn,
    InOut:= aInOut[0],
    Size:= uiSize,
    P_CY=> bP_CY,
    ENO=> );
```

Expression	Type	Value
AryShiftReg_0	AryShiftReg	
bShift	BOOL	TRUE
bReset	BOOL	FALSE
bIn	BOOL	FALSE
aInOut	ARRAY [0..3] OF BYTE	
aInOut[0]	BYTE	2#11001000
aInOut[1]	BYTE	2#01100001
aInOut[2]	BYTE	2#11111000
aInOut[3]	BYTE	2#01100100
uiSize	UINT	2#0000000000000011
bP_CY	BOOL	TRUE

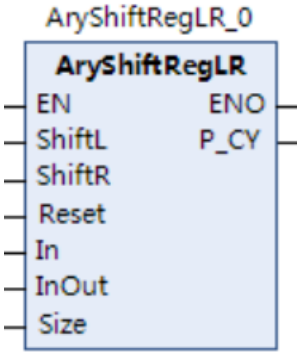
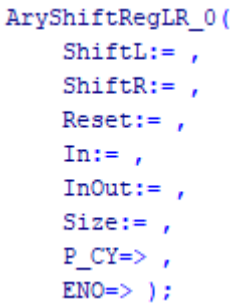
Анализ принципа	

1-4-3-6. Регистр сдвига влево-вправо AryShiftRegLR

(1) Обзор команды

Сдвиг всех элементов массива на 1 бит влево (или вправо) и вставка входного значения в самое нижнее (верхнее) положение.

Регистр сдвига влево-вправо [AryShiftRegLR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryShiftRegLR	Регистр сдвига влево-вправо		


```
AryShiftRegLR_0(
    ShiftL:= ,
    ShiftR:= ,
    Reset:= ,
    In:= ,
    InOut:= ,
    Size:= ,
    P_CY=> ,
    ENO=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
ShiftL	Сдвиг влево	BOOL	[FALSE,TRUE]	FALSE	Выполнение сдвига влево при изменении с FALSE на True

ShiftR	Сдвиг вправо	BOOL	[FALSE,TRUE]	FALSE	Выполнение сдвига вправо при изменении с FALSE на True
Reset	Сброс	BOOL	[FALSE,TRUE]	FALSE	Выполнение сброса при изменении с FALSE на True
In	Входное значение	BOOL	[FALSE,TRUE]	FALSE	Значение, вставленное в самом нижнем или верхнем положении InOut[]
Size	Количество элементов в массиве положения	UINT	В качестве типа данных	0	Количество элементов, используемых в сдвиговом регистре InOut[]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
P_CY	Флаг переноса	BOOL	В качестве типа данных	-	Значение, сохраненное в флаге переноса
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив InOut[]	Массив положения	BOOL/BYTE/WORD /DWORD/LWORD	В качестве типа данных	-	Массив положения

(3) Функция

- При изменении значения параметра "ShiftL" с FALSE на True элементы массива "Size", начиная с InOut[], установленного на выводе InOut, сдвигаются влево (в направлении старшего бита) на 1 бит. Вставьте входное значение "In" в самом нижнем положении. Старший бит, переполнивший массив, будет выведен во флаг переноса (CY) "P_CY".

- При изменении значения параметра "ShiftR" с FALSE на True элементы массива "Size", начиная с InOut[], установленного на выводе InOut, сдвигаются вправо (в направлении младшего бита) на 1 бит. Вставьте входное значение "In" в самом верхнем положении. Младший бит, переполнивший массив, будет выведен во флаг переноса (CY) "P_CY".

- Если "Reset" равен true, все биты элементов "Size", начиная с InOut[], установленного на выводе InOut, и флаги переноса (CY) устанавливаются в FALSE.

(4) Примечание

- Если "Reset" равен true, сдвига не произойдет, даже если "ShiftL" или "ShiftR" изменится с FALSE на true.

- Сдвига не произойдет, если оба "ShiftL" и "ShiftR" изменятся с FALSE на true.

- Если значение "Size" равно 0, InOut[] остается неизменным, а функция возвращает значение True.

- Если значение "Size" превысит область массива InOut[], ПЛК перестает выдавать исключение, как показано на рисунке ниже:



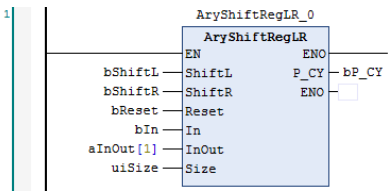
останов программа не изменилась исключение при загрузке программы

- Для InOut[] требуются булева и битовая строка, иначе при выполнении программы ENO будет ложным.

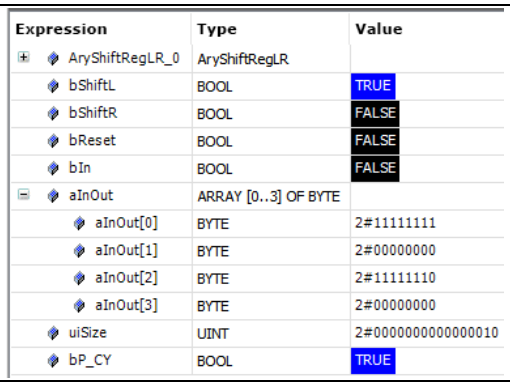
(5) Пример программы

ДЕМО	LD	ST
Задание переменной		
Пример программы		

```
VAR
    AryShiftRegLR_0:AryShiftRegLR;
    bShiftL:BOOL;
    bShiftR:BOOL;
    bReset:BOOL;
    bIn:BOOL;
    aInOut:ARRAY[0..3]OF BYTE:=[2#11111111,2#00000000,2#11111111,2#00000000];
    uiSize:UINT:=2;
    bP_CY:BOOL;
END_VAR
```




```
AryShiftRegLR_0(
    ShiftL:= bShiftL,
    ShiftR:= bShiftR,
    Reset:= bReset,
    In:= bIn,
    InOut:= aInOut[1],
    Size:= uiSize,
    P_CY=> bP_CY,
    ENO=> );
```

Результат операции	

Expression	Type	Value
AryShiftRegLR_0	AryShiftRegLR	
bShiftL	BOOL	TRUE
bShiftR	BOOL	FALSE
bReset	BOOL	FALSE
bIn	BOOL	FALSE
aInOut	ARRAY [0..3] OF BYTE	
aInOut[0]	BYTE	2#11111111
aInOut[1]	BYTE	2#00000000
aInOut[2]	BYTE	2#11111110
aInOut[3]	BYTE	2#00000000
uiSize	UINT	2#0000000000000010
bP_CY	BOOL	TRUE

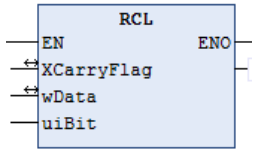
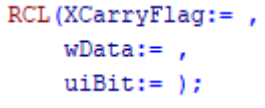
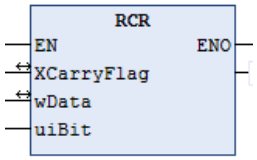
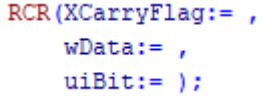
1-4-3-7. Циклический сдвиг влево/вправо с переносом RCL/RCR

(1) Обзор команды

RCL: После выполнения операнд циклически сдвигается влево с флагом переноса, а бит, сдвинутый влево, напрямую заполняет флаг переноса. Исходный флаг переноса сдвигается в самое нижнее положение справа.

RCR: После выполнения операнд циклически сдвигается вправо с флагом переноса, а бит, сдвинутый вправо, напрямую заполняет флаг переноса. Исходный флаг переноса сдвигается в самое верхнее положение слева.

Циклический сдвиг влево/вправо с переносом [RCL/RCR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
RCL	Циклический сдвиг влево с переносом		
RCR	Циклический сдвиг вправо с переносом		


```
RCL(XCarryFlag:= ,
    wData:= ,
    uiBit:= );
```



```
RCR(XCarryFlag:= ,
    wData:= ,
    uiBit:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
uiBit	Номер циклического сдвига	UINT	1-1024	-	Номер циклического сдвига
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
xCarryFlag	Флаг переноса данных	BOOL	-	-	Флаг переноса данных
wData	источники данных	WORD	1-1024	-	источники данных

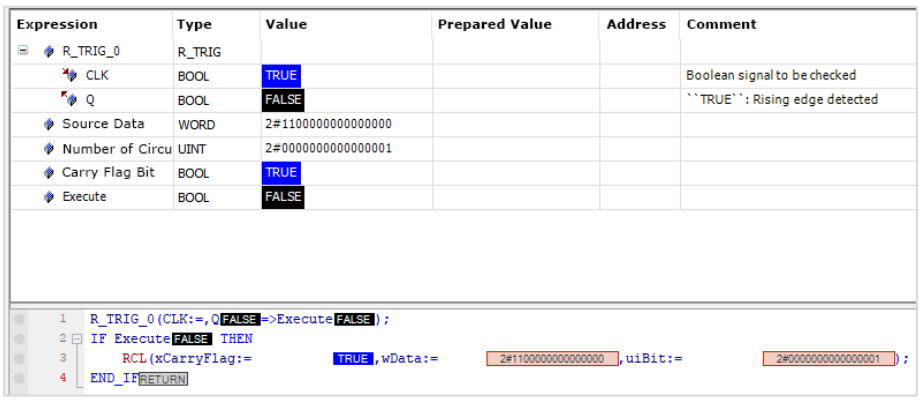
(3) Функция

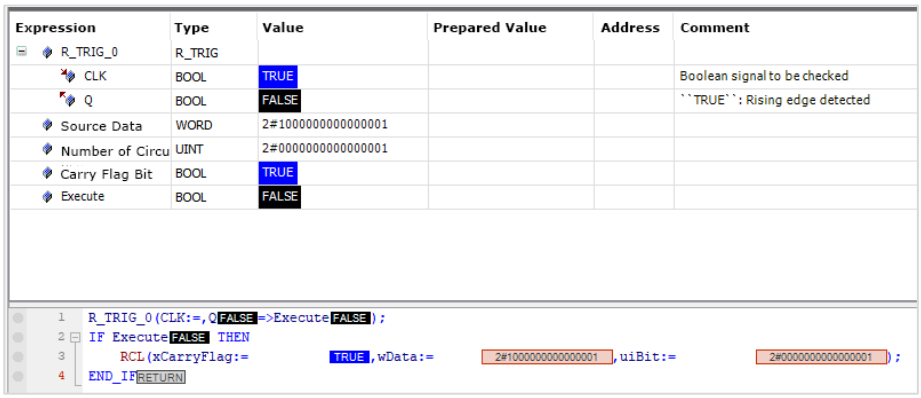
- Сдвиг данных типа WORD во "wData" влево и вправо в зависимости от двоичного отображения.
- Задание количества преобразуемых элементов "In[]" в "Size".

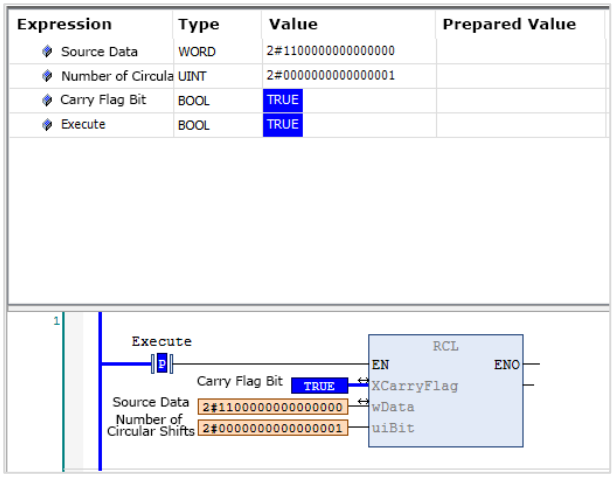
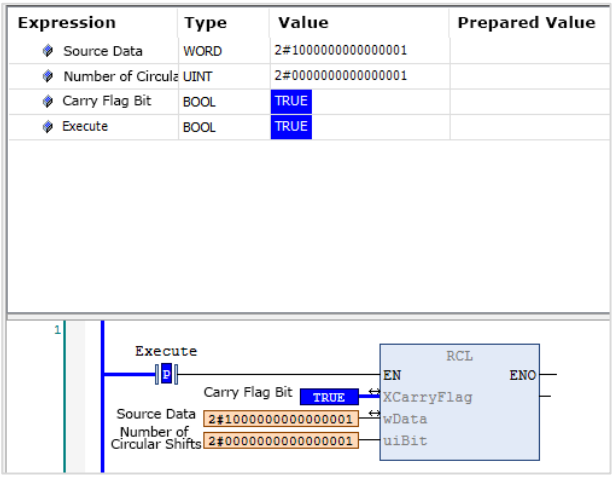
(4) Пример программы

1. RCL

Когда входной сигнал "исходные данные" равен 2 # 111000000, а "номер сдвига" равен 1:

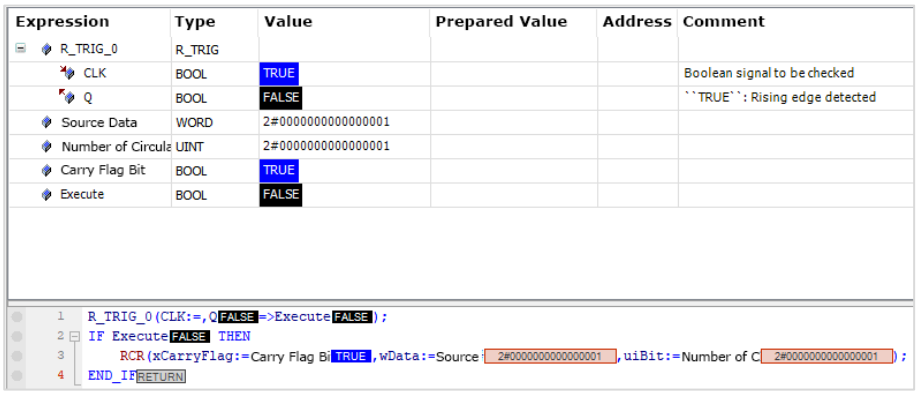
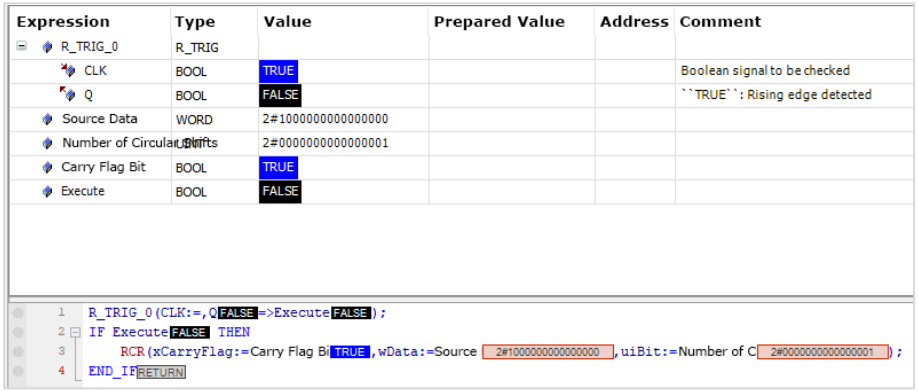
ST:	
	При первом срабатывании выходной сигнал "данные о результатах" равен 2 # 11000000000. Действительные данные сдвигаются влево на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое нижнее положение справа.

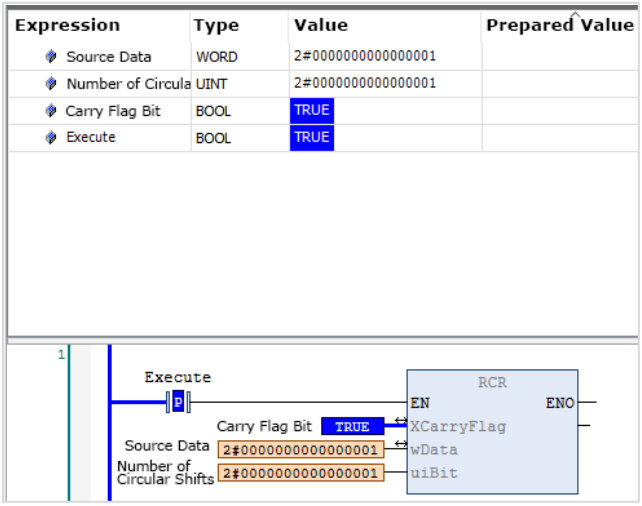
	При втором срабатывании выходной сигнал "данные о результатах" равен 2 # 1000000000001. Действительные данные сдвигаются влево на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое нижнее положение справа.

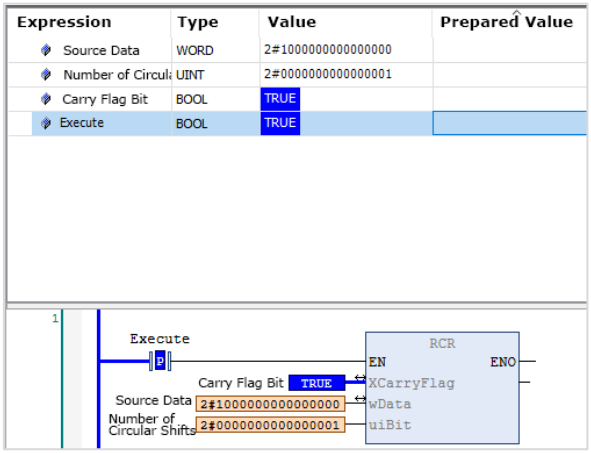
LD:	
	При первом срабатывании выходной сигнал "данные о результатах" равен 2 # 11000000000. Действительные данные сдвигаются влево на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое нижнее положение справа.
	При втором срабатывании выходной сигнал "данные о результатах" равен 2 # 1000000000001. Действительные данные сдвигаются влево на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое нижнее положение справа.

2. RCR

Когда входной сигнал "исходные данные" равен 2 # 000000 0000011, а "номер сдвига" равен 1:

ST:	
	При первом срабатывании выходной сигнал "данные о результатах" равен 2 # 1000000000001. Действительные данные сдвигаются вправо на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое верхнее положение слева.
	При втором срабатывании выходной сигнал "данные о результатах" равен 2 # 100000000000. Действительные данные сдвигаются вправо на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое верхнее положение слева.

LD:	
	При первом срабатывании выходной сигнал "данные о результатах" равен 2 # 1000000000001. Действительные данные сдвигаются вправо на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое верхнее положение слева.

	При втором срабатывании выходной сигнал "данные о результатах" равен 2 # 100000000000. Действительные данные сдвигаются вправо на 1 бит до флага переноса, а исходный флаг переноса сдвигается в самое верхнее положение слева.

1-4-3-8. Сдвиг N элементов массива влево/вправо ArySHL/ArySHR

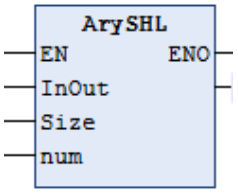
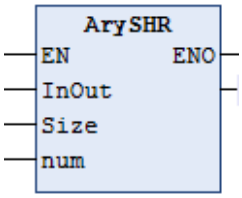
(1) Обзор команды

Сдвиг нескольких элементов массива.

ArySHL: Сдвиг нескольких элементов в массиве влево (направление высокого порядка).

ArySHR: Сдвиг нескольких элементов в массиве вправо (направление низкого порядка).

Сдвиг N элементов массива влево/вправо [ArySHL/ArySHR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ArySHL	Сдвиг N элементов в массиве влево		```ArySHL(InOut:= , Size:= , num:=);```
ArySHR	Сдвиг N элементов в массиве вправо		```ArySHR(InOut:= , Size:= , num:=);```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Size	Количество элементов в сдвинутом массиве объектов	UINT	В качестве типа данных	1	Количество элементов в сдвинутом массиве объектов
Num	Количество сдвинутых цифр	UINT	В качестве типа данных	1	Количество сдвинутых цифр
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив InOut[]	Массив объектов сдвига	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT /ULINT/SINT/INT/DINT/ LINT/REAL/LREAL/TIME/ DATE/TOD/DT/STRING	В качестве типа данных	-	Массив объектов сдвига

(3) Функция

- Сдвиг битов "Num" влево/вправо для младшего/старшего элементов "Size" массива объектов сдвига InOut[].
- Элементы сдвигаются только в пределах "Size", а после сдвига элементы, переполняющие сдвинутый массив объектов, удаляются из него.
- После сдвига инициализируйте пустые элементы массива объектов InOut[], то есть присвойте пустым элементам начальное значение по умолчанию соответствующего типа данных.
- Начальные значения по умолчанию для каждого типа данных приведены в таблице ниже:

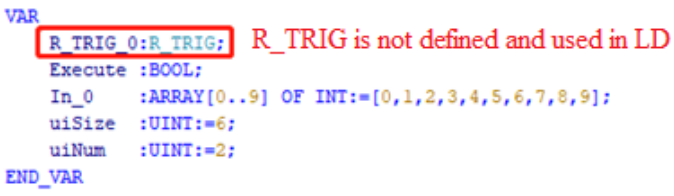
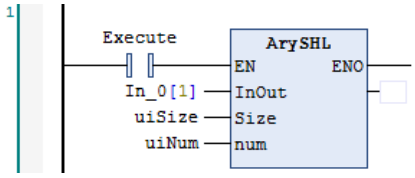
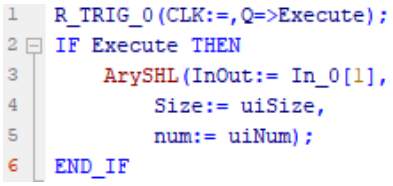
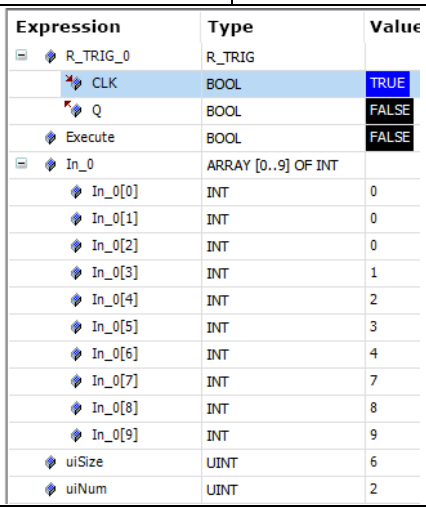
Тип данных	Начальное значение
BOOL	FALSE
BYTE, WORD, DWORD, LWORD	16#0

USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL	0
TIME	T#0 мс
DATE	D#1970-1-1
TOD	TOD#0:0:0
DT	DT#1970-1-1-0:0:0
STRING	‘’

(4) Примечание

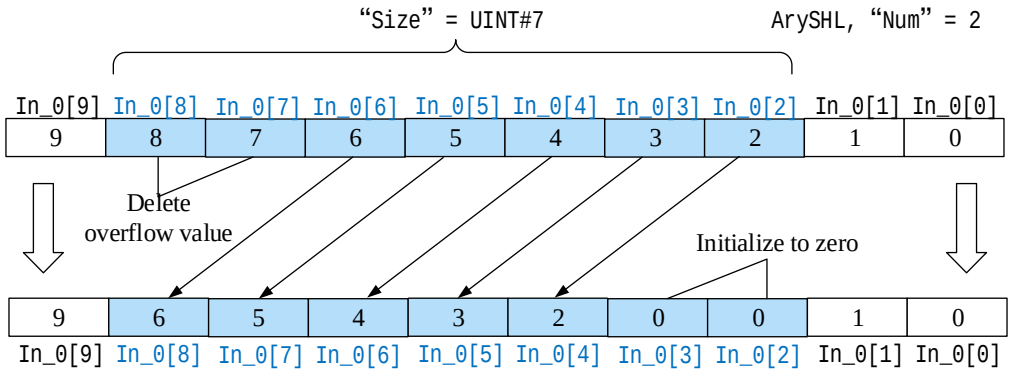
- Значение "Size" не должно превышать область массива в InOut[].
- Если значение "Num" или "Size" равно 0, сдвига не происходит, функция возвращает значение True, а InOut[] остается неизменным.
- Если значение "Num" больше, чем значение "Size", то последовательные значения Size, начиная с InOut[], инициализируются нулем.

(5) Пример программы

ДЕМО	LD	ST
Задание переменной		
Пример программы		
Результат операции		


```
R_TRIG_0(CLK:=,Q=>Execute);
IF Execute THEN
    ArySHL(InOut:= In_0[1],
        Size:= uiSize,
        num:= uiNum);
END_IF
```

Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	TRUE
Q	BOOL	FALSE
Execute	BOOL	FALSE
In_0	ARRAY [0..9] OF INT	
In_0[0]	INT	0
In_0[1]	INT	0
In_0[2]	INT	0
In_0[3]	INT	1
In_0[4]	INT	2
In_0[5]	INT	3
In_0[6]	INT	4
In_0[7]	INT	7
In_0[8]	INT	8
In_0[9]	INT	9
uiSize	UINT	6
uiNum	UINT	2

Анализ принципа	

1-4-3-9. Сдвиг N элементов массива влево/вправо NSHLC/NSHRC

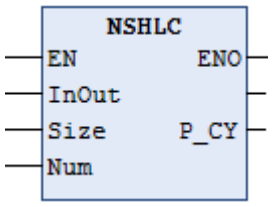
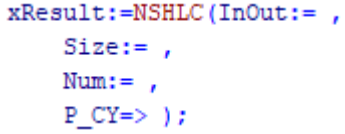
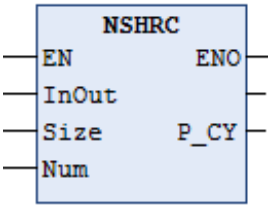
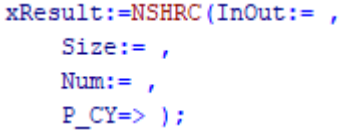
(1) Обзор команды

Перемещение нескольких битов вместе с флагом переноса (CY).

NSHLC: Сдвиг влево (в направлении старшего бита).

NSHRC: Сдвиг вправо (в направлении младшего бита).

Сдвиг N элементов массива влево/вправо [NSHLC/NSHRC]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
NSHLC	Сдвиг N элементов в массиве влево		
NSHRC	Сдвиг N элементов в массиве вправо		


```
xResult:=NSHLC(InOut:= ,
    Size:= ,
    Num:= ,
    P_CY=> );
```



```
xResult:=NSHRC(InOut:= ,
    Size:= ,
    Num:= ,
    P_CY=> );
```

(2) Связанная переменная

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Size	Количество битов в сдвинутом объекте	USINT	В качестве типа данных	1	Количество битов в сдвинутом объекте
Num	Биты сдвига	USINT	В качестве типа данных	1	Биты сдвига
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
P_CY	Флаг переноса	BOOL	В качестве типа данных	-	Флаг переноса
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив InOut[]	Массив объектов сдвига	BOOL/BYTE/WORD /DWORD/LWORD	В качестве типа данных	-	Массив объектов сдвига

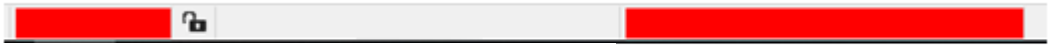
(3) Функция

- Перемещение элементов массива "Size", начиная с InOut [0] сдвинутого массива объектов InOut [], на "Num" битов. Конечное значение бита переполнения будет выдано в флаг переноса (CY), а бит, вставленный с противоположной стороны, будет заполнен нулем.
- NSHLC: Сдвиг влево (от младшего к старшему биту).
- NSHRC: Сдвиг вправо (от старшего к младшему биту).

(4) Примечание

- Если значение "Num" равно 0, сдвига не происходит, функция возвращает значение True, InOut[] остается неизменным, а значение "P_CY" равно FALSE.
- Если значение "Num" больше, чем значение "Size", то значением в диапазоне от bit0 в InOut[] до битов "Size" будет FALSE. Кроме того, значением флага переноса (CY) также будет FALSE.
- Если значение "Size" превысит область массива InOut[], ПЛК будет выдавать исключение, как показано на рисунке ниже. Такое неправильное

использование следует исключить.



останов программа не изменилась исключение при загрузке программы

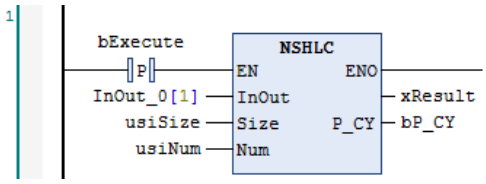
- Для "InOut[]" требуются булевы и битовые числа, в противном случае функция выполнения возвращает FALSE.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменной</td><td colspan="2">

```
VAR
    bExecute:BOOL;
    InOut_0:ARRAY[0..4] OF BYTE:=[2(2#11110000),3(2#00001111)];
    usiSize:USINT:=15;
    usiNum :USINT:=2;
    bP_CY  :BOOL;
    xResult:BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  R_TRIG_0(CLK:=,Q=>bExecute);
2  IF bExecute THEN
3      xResult:=NSHLC(InOut:= InOut_0[1],
4          Size:= usiSize,
5          Num:= usiNum,
6          P_CY=> bP_CY);
7  END_IF
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
bExecute	BOOL	TRUE
InOut_0	ARRAY [0..4] OF BYTE	
InOut_0[0]	BYTE	2#11110000
InOut_0[1]	BYTE	2#11000000
InOut_0[2]	BYTE	2#00111111
InOut_0[3]	BYTE	2#00001111
InOut_0[4]	BYTE	2#00001111
usiSize	USINT	2#00001111
usiNum	USINT	2#00000010
bP_CY	BOOL	FALSE
xResult	BOOL	TRUE

</td></tr>
</table>

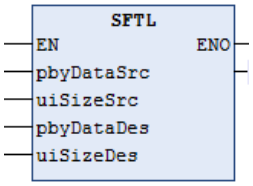
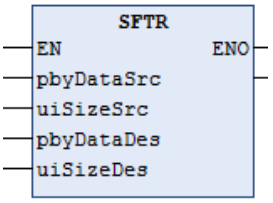
1-4-3-10. Копирование битовых данных влево/вправо SFTL/SFTR

(1) Обзор команды

SFTL: Непрерывное копирование битовых данных заданной длины из исходного массива влево в целевой массив.

SFTR: Непрерывное копирование битовых данных заданной длины из исходного массива вправо в целевой массив.

Копирование битовых данных влево/вправо [SFTL/SFTR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
SFTL	Копирование битовых данных влево		SFTL(pbyDataSrc:= , uiSizeSrc:= , pbyDataDes:= , uiSizeDes:=);
SFTR	Копирование битовых данных вправо		SFTR(pbyDataSrc:= , uiSizeSrc:= , pbyDataDes:= , uiSizeDes:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
pbyDataSrc	Исходные данные	POINTER_TO_BYTE	-	-	Исходные данные
pbyDataDes	Целевые данные	POINTER_TO_BYTE	-	-	Целевые данные
uiSizeSrc	Номера источников данных	BYTE	-	-	Номера источников данных
uiSizeDes	Номера целевых данных	BYTE	-	-	Номера целевых данных

(3) Функция

- Значение uiSizeSrc должно быть больше uiSizeDes, в противном случае ПЛК аварийно завершит работу.

(4) Пример программы

1. SFTL

Скопируйте последовательные биты данных uiSizeSrc (количество элементов в одной копии), начиная с первого адреса указателя pbyDataSrc (исходный массив) в младший диапазон первого адреса pbyDataDes (целевой массив). Перед копированием все биты данных uiSizeDes (целевое смещение) в pbyDataDes перемещаются в старший диапазон битами uiSizeSrc, и только в пределах диапазона битов данных uiSizeDes (целевое смещение) в pbyDataDes (целевой массив).

<table>
<tr><th colspan="2">ST:</th></tr>
<tr><td>

```
PROGRAM SFTL_ST
VAR
    R_TRIG_0:R_TRIG;
    uiSizeSrc:BYTE;
    uiSizeDes:BYTE;
    Source Data:ARRAY[0..3]OF BOOL;
    Target Data:ARRAY[0..5]OF BOOL;
    Execute:BOOL;
END_VAR
```

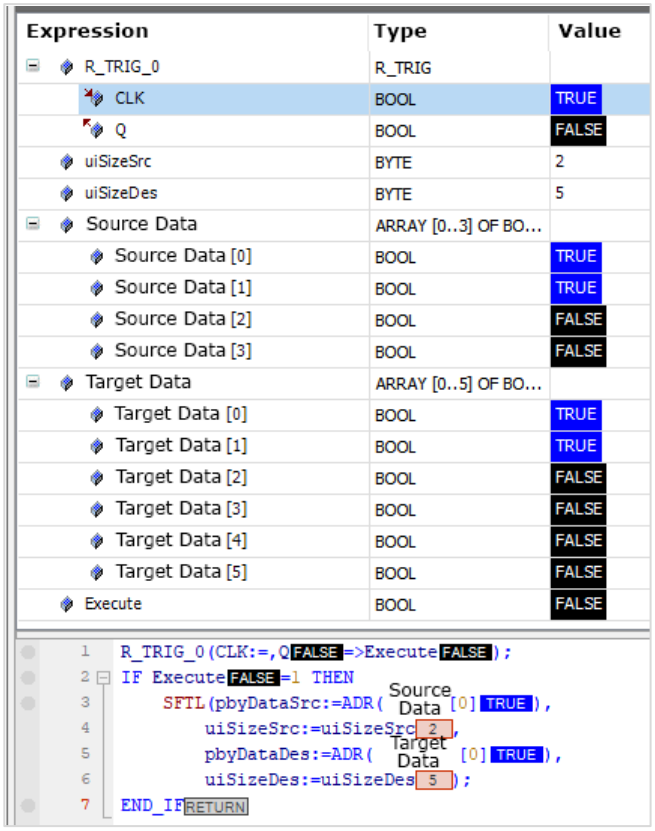
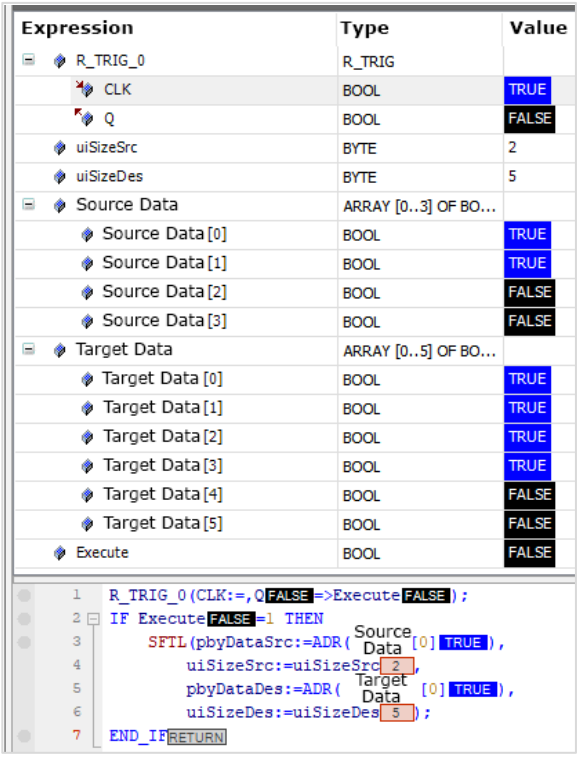
```
R_TRIG_0(CLK:=,Q=>Execute);
IF Execute=1 THEN
    SFTL(pbyDataSrc:=ADR(Source Data[0]),
        uiSizeSrc:=uiSizeSrc,
        pbyDataDes:=ADR(Target Data[0]),
        uiSizeDes:=uiSizeDes);
END_IF
```

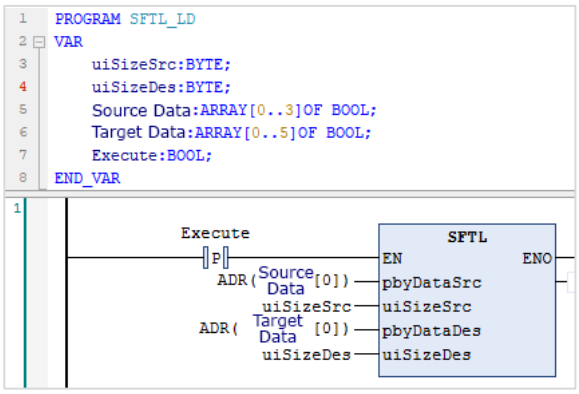
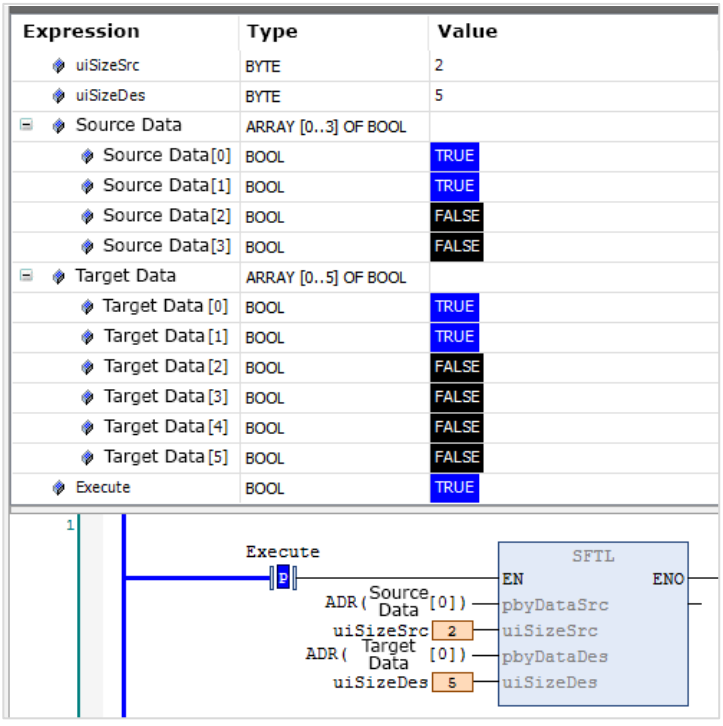
</td><td>Задание переменных и программирование</td></tr>
<tr><td>

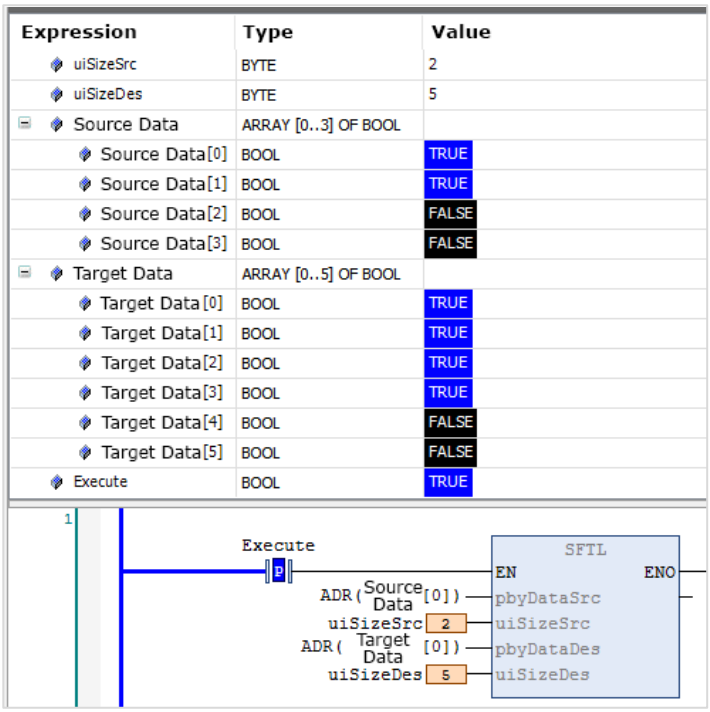
Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	FALSE
Q	BOOL	FALSE
uiSizeSrc	BYTE	2
uiSizeDes	BYTE	5
Source Data	ARRAY [0..3] OF BO...	
Source Data [0]	BOOL	TRUE
Source Data [1]	BOOL	TRUE
Source Data [2]	BOOL	FALSE
Source Data [3]	BOOL	FALSE
Target Data	ARRAY [0..5] OF BO...	
Target Data [0]	BOOL	FALSE
Target Data [1]	BOOL	FALSE
Target Data [2]	BOOL	FALSE
Target Data [3]	BOOL	FALSE
Target Data [4]	BOOL	FALSE
Target Data [5]	BOOL	FALSE
Execute	BOOL	FALSE

```
R_TRIG_0(CLK:=,Q FALSE =>Execute FALSE );
IF Execute FALSE =1 THEN
    SFTL(pbyDataSrc:=ADR(Source Data[0] TRUE ),
        uiSizeSrc:=uiSizeSrc 2 ,
        pbyDataDes:=ADR(Target Data[0] FALSE ),
        uiSizeDes:=uiSizeDes 5 );
END_IF RETURN
```

</td><td>Присваивание начального значения</td></tr>
</table>

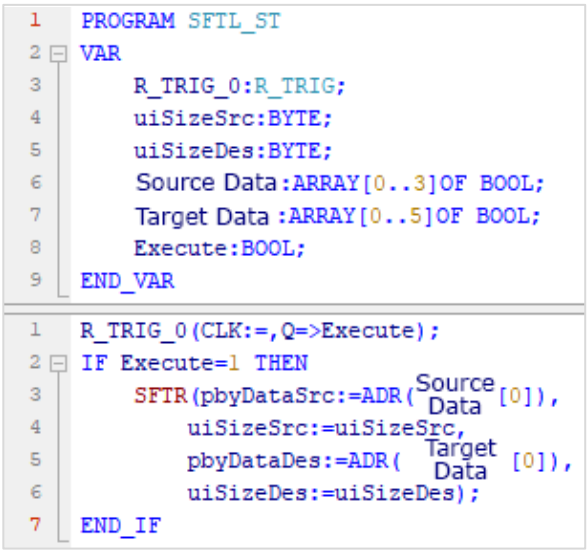
	При срабатывании в первый раз при сдвиге на 2 бита в сторону старшего бита состояние все еще FALSE, так как все целевые данные FALSE. Исходные данные [0], исходные данные [1] передаются в целевые данные [0], и целевые данные [1] становятся истинными.
	При срабатывании во второй раз значение исходных целевых данных [0] - целевых данных [4] перемещается на 2 бита в положение старшего бита, в целевые данные [2] - целевые данные [4], а 2 бита переполнения перемещаются в целевые данные [0] - целевые данные [1]. Далее значения исходных данных [0] и исходных данных [1] передаются в целевые данные [0] и целевые данные [1].

LD:	
	Задание переменных и программирование.
	Присваивание начального значения.
	При срабатывании в первый раз при сдвиге на 2 бита в сторону старшего бита состояние все еще FALSE, так как все целевые данные FALSE. Исходные данные [0], исходные данные [1] передаются в целевые данные [0], и целевые данные [1] становятся истинными.

	При срабатывании во второй раз значение исходных целевых данных [0] - целевых данных [4] перемещается на 2 бита в положение старшего бита, в целевые данные [2] - целевые данные [4], а 2 бита переполнения перемещаются в целевые данные [0] - целевые данные [1]. Далее значения исходных данных [0] и исходных данных [1] передаются в целевые данные [0] и целевые данные [1].

2. SFTR

Скопируйте последовательные биты данных uiSizeSrc (количество элементов массива в одной копии), начиная с первого адреса указателя pbyDataSrc (исходный массив) в область старшего бита pbyDataDes (целевой массив). Перед копированием все биты данных uiSizeSrc (целевое смещение) в pbyDataDes перемещаются в младший диапазон битами uiSizeSrc, и только в пределах диапазона битов данных uiSizeDes (целевое смещение) в pbyDataDes (целевой массив).

ST:	
	Задание переменных и программирование.

<table>
<tr>
<td>

Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	FALSE
Q	BOOL	FALSE
uiSizeSrc	BYTE	2
uiSizeDes	BYTE	5
Source Data	ARRAY [0..3] OF BOOL	
Source Data [0]	BOOL	TRUE
Source Data [1]	BOOL	TRUE
Source Data [2]	BOOL	FALSE
Source Data [3]	BOOL	FALSE
Target Data	ARRAY [0..5] OF BOOL	
Target Data [0]	BOOL	FALSE
Target Data [1]	BOOL	FALSE
Target Data [2]	BOOL	FALSE
Target Data [3]	BOOL	FALSE
Target Data	BOOL	FALSE
Target Data	BOOL	FALSE
Execute	BOOL	FALSE

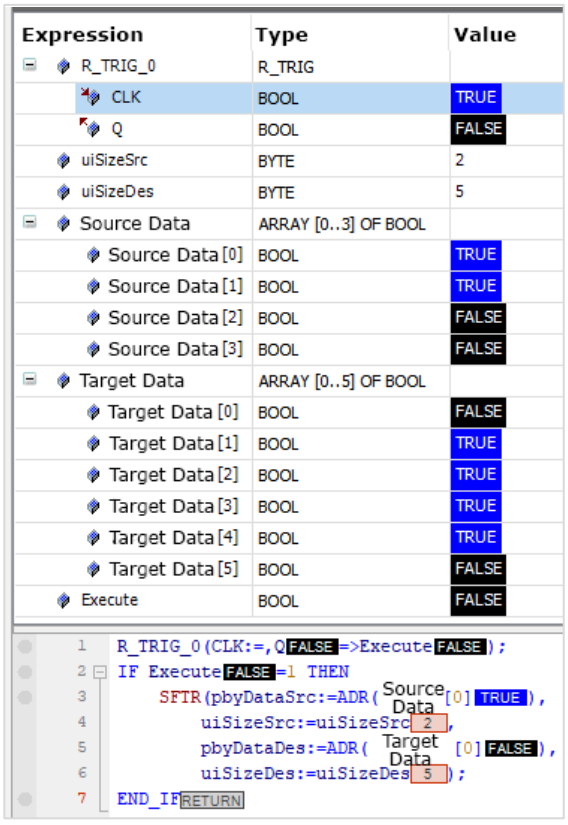
```
1  R_TRIG_0(CLK:=,Q FALSE =>Execute FALSE );
2  IF Execute FALSE =1 THEN
3      SFTR(pbyDataSrc:=ADR( Source Data [0] TRUE ),
4          uiSizeSrc:=uiSizeSrc 2 ,
5          pbyDataDes:=ADR( Target Data [0] FALSE ),
6          uiSizeDes:=uiSizeDes 5 );
7  END_IF RETURN
```

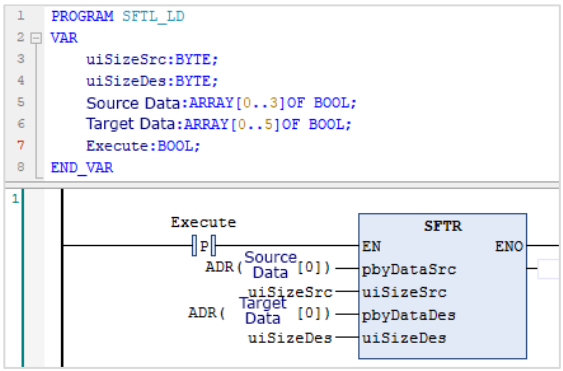
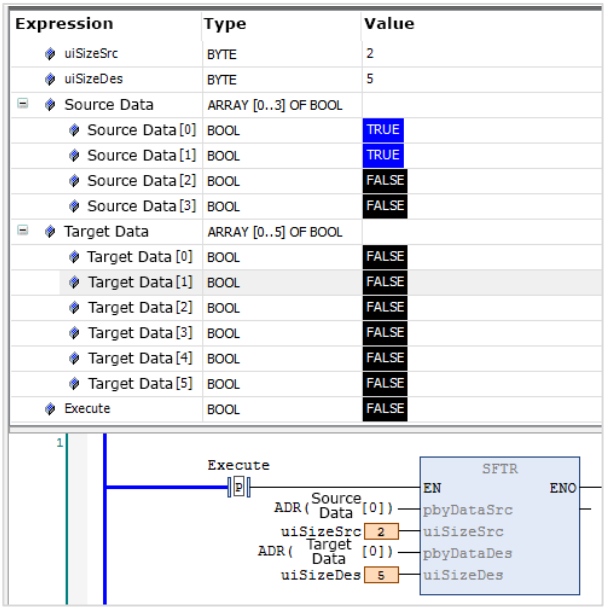
</td>
<td>Присваивание начального значения.</td>
</tr>
<tr>
<td>

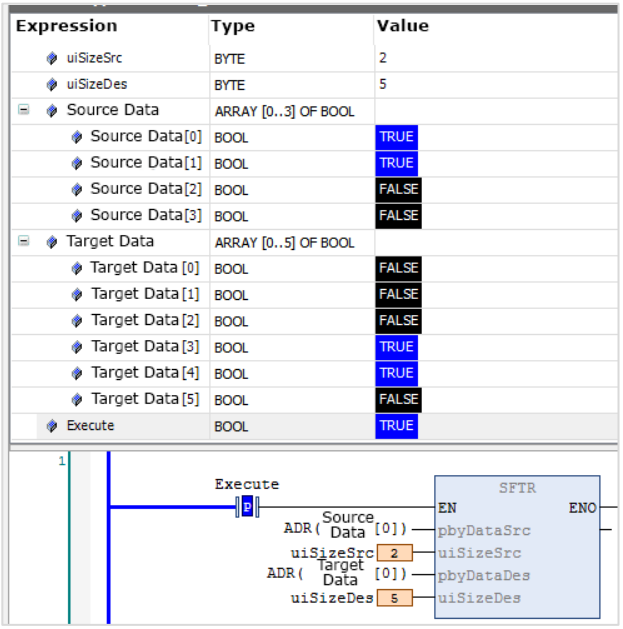
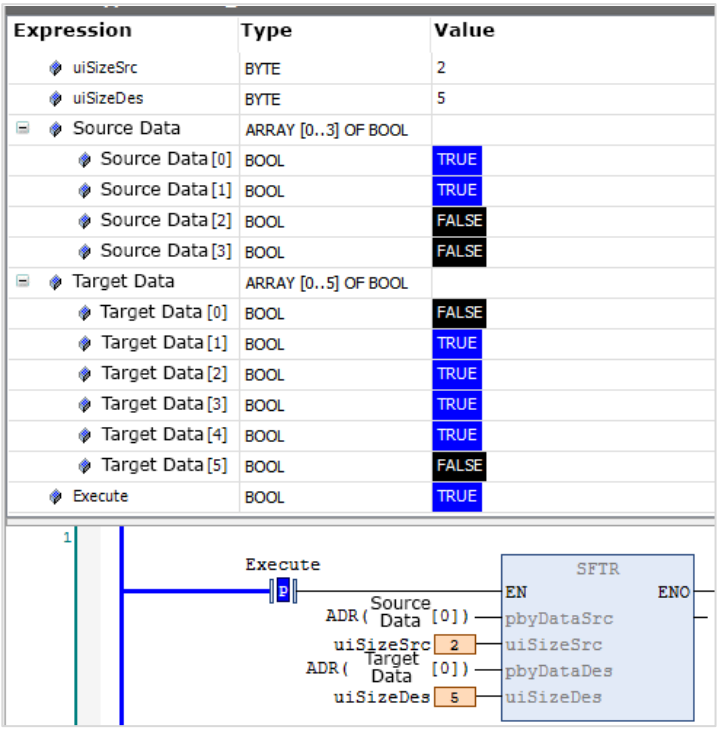
Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	TRUE
Q	BOOL	FALSE
uiSizeSrc	BYTE	2
uiSizeDes	BYTE	5
Source Data	ARRAY [0..3] OF BOOL	
Source Data [0]	BOOL	TRUE
Source Data [1]	BOOL	TRUE
Source Data [2]	BOOL	FALSE
Source Data [3]	BOOL	FALSE
Target Data	ARRAY [0..5] OF BOOL	
Target Data [0]	BOOL	FALSE
Target Data [1]	BOOL	FALSE
Target Data [2]	BOOL	FALSE
Target Data [3]	BOOL	TRUE
Target Data [4]	BOOL	TRUE
Target Data [5]	BOOL	FALSE
Execute	BOOL	FALSE

```
1  R_TRIG_0(CLK:=,Q FALSE =>Execute FALSE );
2  IF Execute FALSE =1 THEN
3      SFTR(pbyDataSrc:=ADR( Source Data [0] TRUE ),
4          uiSizeSrc:=uiSizeSrc 2 ,
5          pbyDataDes:=ADR( Target Data [0] FALSE ),
6          uiSizeDes:=uiSizeDes 5 );
7  END_IF RETURN
```

</td>
<td>При срабатывании в первый раз при сдвиге на 2 бита в сторону младшего бита состояние все еще FALSE, так как все целевые данные FALSE. Исходные данные [0], исходные данные [1] передаются в целевые данные [3], и целевые данные [4] становятся истинными.</td>
</tr>
</table>

	При срабатывании во второй раз значение исходных целевых данных [0] - целевых данных [4] перемещается на 2 бита в положение младшего бита, в целевые данные [0] - целевые данные [2], а 2 бита переполнения перемещаются в целевые данные [3] - целевые данные [4]. Далее значения исходных данных [0] и исходных данных [1] передаются в целевые данные [3] и целевые данные [4].

LD:	
	Задание переменных и программирование.
	Присваивание начального значения.

	При срабатывании в первый раз при сдвиге на 2 бита в сторону младшего бита состояние все еще FALSE, так как все целевые данные FALSE. Исходные данные [0], исходные данные [1] передаются в целевые данные [3], и целевые данные [4] становятся истинными.
	При срабатывании во второй раз значение исходных целевых данных [0] - целевых данных [4] перемещается на 2 бита в положение младшего бита, в целевые данные [0] - целевые данные [2], а 2 бита переполнения перемещаются в целевые данные [3] - целевые данные [4]. Далее значения исходных данных [0] и исходных данных [1] передаются в целевые данные [3] и целевые данные [4].

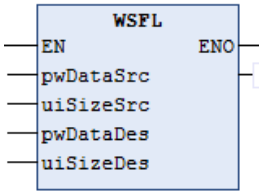
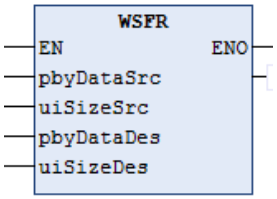
1-4-3-11. Копирование данных слова влево/вправо WSFL/WSFR

(1) Обзор команды

WSFL: Непрерывное копирование данных слова заданной длины из исходного массива влево в целевой массив.

WSFR: Непрерывное копирование данных слова заданной длины из исходного массива вправо в целевой массив.

Копирование данных словах влево/вправо [WSFL/WSFR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
WSFL	Копирование данных слова влево		WSFL(pwDataSrc:= ,uiSizeSrc:= , pwDataDes:= , uiSizeDes:=);
WSFR	Копирование данных слова вправо		WSFR(pwDataSrc:= , uiSizeSrc:= , pwDataDes:= , uiSizeDes:=);

(2) Связанная переменная

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение
pbyDataSrc	Исходные данные	POINTER_TO_WORD	-	-
pbyDataDes	Целевые данные	POINTER_TO_WORD	-	-
uiSizeSrc	Номера исходных данных	UINT	1-1024	-

uiSizeDes	Номера целевых данных	UINT	1-1024	-

(3) Примечание

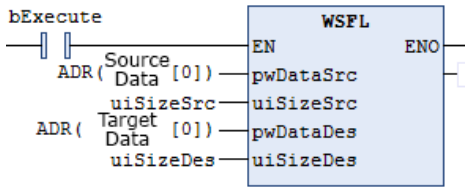
- Значение UiSizeSrc не должно быть больше uiSizeDes, в противном случае ПЛК аварийно завершит работу.

(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    R_TRIG_0:R_TRIG;
    bExecute: BOOL;
Source Data:ARRAY[0..3] OF INT:=[10,20,30,40];
  Target Data:ARRAY[0..9]OF INT:=[1,2,3,4,5,6,7,8,9,10];
    uiSizeSrc:UINT;
    uiSizeDes:UINT;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



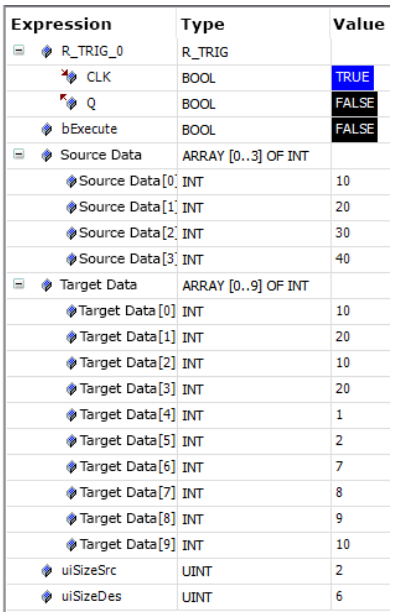
</td><td>

```
R_TRIG_0(CLK:=,Q=>bExecute);
IF bExecute=1 THEN
    WSFL(pwDataSrc:= ADR( Source Data [0]),
        uiSizeSrc:= uiSizeSrc,
        pwDataDes:= ADR( Target Data [0]),
        uiSizeDes:= uiSizeDes);
END_IF
```

</td></tr>
<tr><td>Присваивание переменной</td><td colspan="2">

Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	FALSE
Q	BOOL	FALSE
bExecute	BOOL	FALSE
Source Data	ARRAY [0..3] OF INT	
Source Data[0]	INT	10
Source Data[1]	INT	20
Source Data[2]	INT	30
Source Data[3]	INT	40
Target Data	ARRAY [0..9] OF INT	
Target Data[0]	INT	1
Target Data[1]	INT	2
Target Data[2]	INT	3
Target Data[3]	INT	4
Target Data[4]	INT	5
Target Data[5]	INT	6
Target Data[6]	INT	7
Target Data[7]	INT	8
Target Data[8]	INT	9
Target Data[9]	INT	10
uiSizeSrc	UINT	2
uiSizeDes	UINT	6

</td></tr>
</table>

Двойное выполнение операции	

Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	TRUE
Q	BOOL	FALSE
bExecute	BOOL	FALSE
Source Data	ARRAY [0..3] OF INT	
Source Data[0]	INT	10
Source Data[1]	INT	20
Source Data[2]	INT	30
Source Data[3]	INT	40
Target Data	ARRAY [0..9] OF INT	
Target Data[0]	INT	10
Target Data[1]	INT	20
Target Data[2]	INT	10
Target Data[3]	INT	20
Target Data[4]	INT	1
Target Data[5]	INT	2
Target Data[6]	INT	7
Target Data[7]	INT	8
Target Data[8]	INT	9
Target Data[9]	INT	10
uiSizeSrc	UINT	2
uiSizeDes	UINT	6

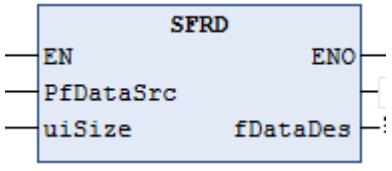
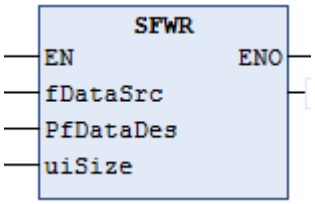
1-4-3-12. Чтение/запись данных в порядке поступления SFRD/SFWR

(1) Обзор команды

SFRD: Считывание содержимого данных заданной длины после выполнения.

SFWR: Запись содержимого данных заданной длины после выполнения.

Чтение/запись данных в порядке поступления [SFRD/SFWR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
SFRD	Чтение данных в порядке поступления		SFRD(pfDataSrc:=, uiSize:=, fDataDes=>);
SFWR	Запись данных в порядке поступления		SFWR(fDataSrc:=, pfDataDes:=, uiSize:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение
PfDataSrc	Исходные данные	POINTER_TO_REAL	-	-
uiSize	Количество элементов данных с плавающей точкой или очередь исходных данных	UINT	-	-
fDataDes	Считывание данных	REAL	-	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**
fDataSrc	Исходные данные	REAL	-	-
PfDataDes	Целевые данные	POINTER_TO_REAL	-	-
uiSize	Количество элементов данных с плавающей точкой или очередь исходных данных	UINT	-	-

(3) Примечание

- Первый элемент исходных данных нужно назначать заранее, в противном случае выполнение программы будет проходить в нештатном режиме.

(4) Пример программы

1. SFRD

Считайте первый элемент очереди «в порядке поступления» pfDataSrc (указатель, указывающий на исходные данные) в fDataDes (целевые данные), а затем пословно перемещайте очередь pfDatasrc вправо на одно слово, уменьшая указатель очереди. Если первое устройство с номером используется в качестве указателя, то при выполнении команды содержимое указателя сначала уменьшается на 1, а затем содержимое устройства, заданного по pfDataSrc, записывается в положении, обозначенным указателем в строке данных fDataDes (в порядке поступления). Если указатель уже равен 0, то команда не будет обрабатывать вышеупомянутую операцию.

ST:

Expression	Type	Value
Source Data	ARRAY [0..9] OF REAL	
Source Data[0]	REAL	1
Source Data[1]	REAL	1
Source Data[2]	REAL	2
Source Data[3]	REAL	3
Source Data[4]	REAL	4
Source Data[5]	REAL	5
Source Data[6]	REAL	6
Source Data[7]	REAL	7
Source Data[8]	REAL	8
Source Data[9]	REAL	9
Data Length	UINT	10
Result Data	REAL	0
bExecute	BOOL	FALSE
R_TRIG_0	R_TRIG	
CLK	BOOL	FALSE
Q	BOOL	FALSE

```
1  Source Data[0] 1 :=1;
2  R_TRIG_0(CLK:= , Q FALSE => bExecute FALSE);
3  IF bExecute FALSE =1 THEN
4      SFRD(PfDataSrc:= ADR( Source Data ),
5           uiSize:=  Data Length 10 ,
6           fDataDes=>  Result Data 0 );
7  END_IF RETURN
```

Начальное состояние.

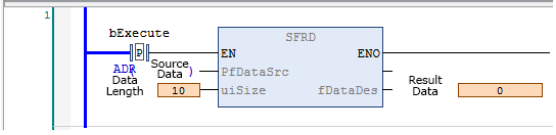
Expression	Type	Value
Source Data	ARRAY [0..9] OF REAL	
Source Data[0]	REAL	1
Source Data[1]	REAL	4
Source Data[2]	REAL	5
Source Data[3]	REAL	6
Source Data[4]	REAL	7
Source Data[5]	REAL	8
Source Data[6]	REAL	9
Source Data[7]	REAL	9
Source Data[8]	REAL	9
Source Data[9]	REAL	9
Data Length	UINT	10
Result Data	REAL	3
bExecute	BOOL	FALSE
R_TRIG_0	R_TRIG	
CLK	BOOL	TRUE
Q	BOOL	FALSE

```
1  Source Data[0] 1 :=1;
2  R_TRIG_0(CLK:= , Q FALSE => bExecute FALSE);
3  IF bExecute FALSE =1 THEN
4      SFRD(PfDataSrc:= ADR( Source Data ),
5           uiSize:=  Data Length 10 ,
6           fDataDes=>  Result Data 3 );
7  END_IF RETURN
```

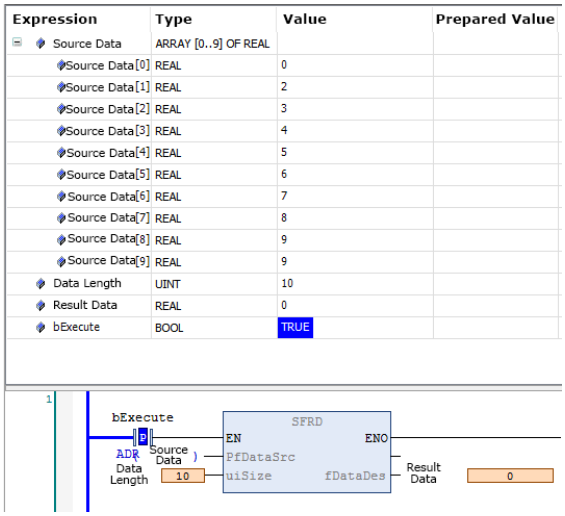
После трехкратного срабатывания.

LD:

Expression	Type	Value	Prepared Value
Source Data	ARRAY [0..9] OF REAL		
Source Data[0]	REAL	1	
Source Data[1]	REAL	1	
Source Data[2]	REAL	2	
Source Data[3]	REAL	3	
Source Data[4]	REAL	4	
Source Data[5]	REAL	5	
Source Data[6]	REAL	6	
Source Data[7]	REAL	7	
Source Data[8]	REAL	8	
Source Data[9]	REAL	9	
Data Length	UINT	10	
Result Data	REAL	0	
bExecute	BOOL	FALSE	

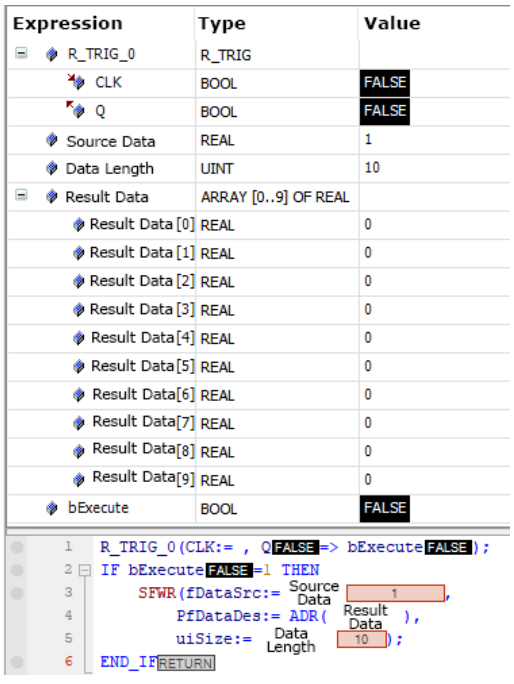


Начальное состояние.

	После трехкратного срабатывания.

2. SFWR

Запишите значение fDataSrc (исходные данные) в очередь «в порядке поступления», начиная с адреса pfDataDes (целевые данные) с номером uiSize (длина данных), используя первое пронумерованное устройство в качестве указателя. При выполнении команды содержимое указателя сначала увеличивается на 1, а затем содержимое устройства, заданного по fDataSrc, записывается в положении, обозначенным указателем в строке данных pfDataDes (в порядке поступления).

ST:	
	Начальное состояние.

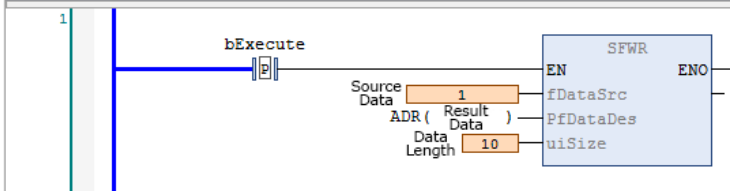
Expression	Type	Value
R_TRIG_0	R_TRIG	
CLK	BOOL	TRUE
Q	BOOL	FALSE
Source Data	REAL	1
Data Length	UINT	10
Result Data	ARRAY [0..9] OF REAL	
Result Data [0]	REAL	3
Result Data [1]	REAL	1
Result Data [2]	REAL	1
Result Data [3]	REAL	1
Result Data [4]	REAL	0
Result Data [5]	REAL	0
Result Data [6]	REAL	0
Result Data [7]	REAL	0
Result Data [8]	REAL	0
Result Data [9]	REAL	0
bExecute	BOOL	FALSE

```
1  R_TRIG_0(CLK:= , Q=> bExecute);
2  IF bExecute=1 THEN
3      SFWR(fDataSrc:= Source Data 1,
4          PfDataDes:= ADR( Result Data ),
5          uiSize:= Data Length 10);
6  END_IF
```

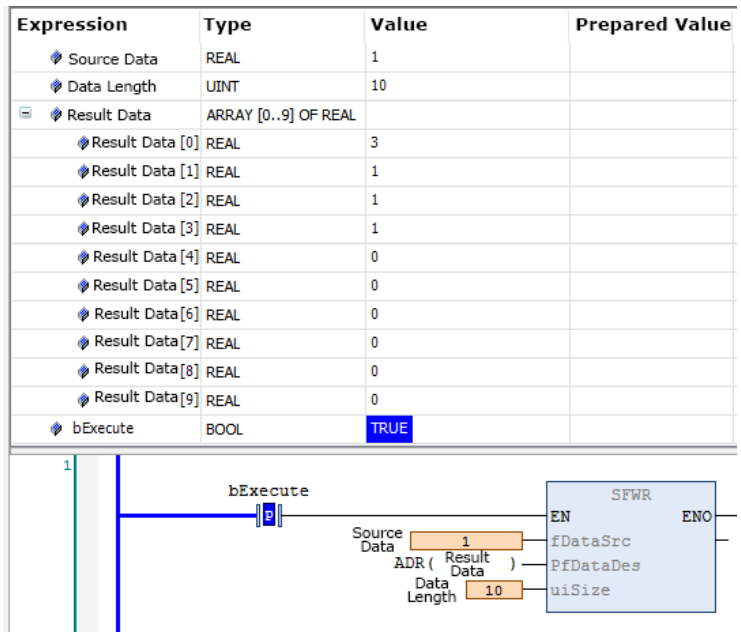
После трехкратного срабатывания.

LD:

Expression	Type	Value	Prepared Value
Source Data	REAL	1	
Data Length	UINT	10	
Result Data	ARRAY [0..9] OF REAL		
Result Dat[0]	REAL	0	
Result Dat[1]	REAL	0	
Result Dat[2]	REAL	0	
Result Dat[3]	REAL	0	
Result Dat[4]	REAL	0	
Result Dat[5]	REAL	0	
Result Dat[6]	REAL	0	
Result Dat[7]	REAL	0	
Result Dat[8]	REAL	0	
Result Dat[9]	REAL	0	
bExecute	BOOL	FALSE	



Начальное состояние.

	После трехкратного срабатывания.

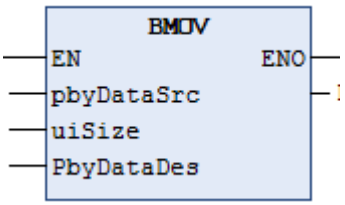
1-4-4. Команды перемещения

1-4-4-1. Групповая пересылка данных BMOV

(1) Обзор команды

Копирование исходных данных в целевые данные с указанной длиной.

Групповая пересылка данных [BMOV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BMOV	Пакетная передача данных		BMOV(pbyDataSrc:= , uiSize:= , PbyDataDes:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение
pbyDataSrc	Начальный адрес для репликации данных	POINTER TO BYTE	-	-
uiSize	Число переменных	UINT	В качестве типа данных	0
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**
PbyDataDes	Начальный адрес для хранения данных	POINTER TO BYTE	-	-

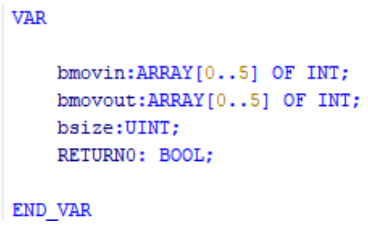
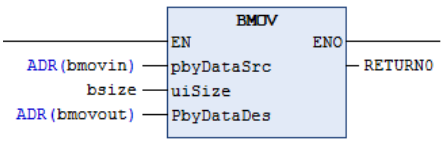
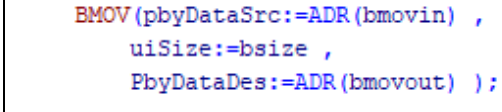
(3) Функция

- Эта команда управляется контактами путем копирования значений переменной uiSize указанного начального адреса pbyDataSrc в единицы uiSize указанного начального адреса pbyDataDes. Диапазон значений uiSize – 1-1024. Возвращаемое значение, True: Успешное выполнение, FALSE: Ошибка выполнения.

(4) Примечание

- Uisize — это количество байтов.

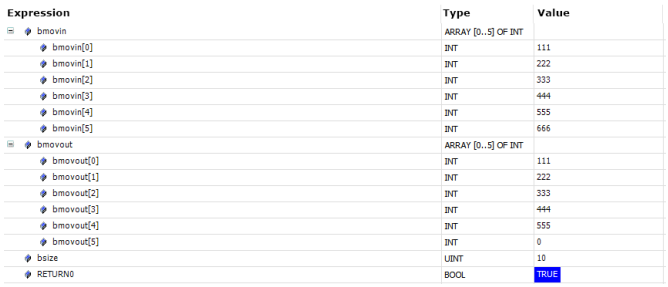
(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		

```
VAR

    bmovin:ARRAY[0..5] OF INT;
    bmovout:ARRAY[0..5] OF INT;
    bsize:UINT;
    RETURN0: BOOL;

END_VAR
```



```
BMOV(pbyDataSrc:=ADR(bmovin) ,
    uiSize:=bsize ,
    PbyDataDes:=ADR(bmovout) );
```

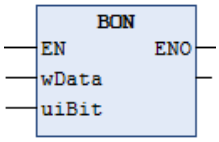
Результат операции	
Анализ результатов	Uisize представляет собой количество байт, при этом данные установлены на 10. Элемент массива определяется как INT, что составляет 2 байта. Поэтому он передает данные из bmovin[0] - bmovin[4] в bmovout[0] - bmovout[4].

1-4-4-2. Определение состояния бита данных BON

(1) Обзор команды

Определите состояние бита uiBit данных wData, сохраните результат в BON и выведите его с правой стороны.

Определение состояния бита данных [BON]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BON	Определение состояния бита данных		BON(wData:= , uiBit:=);

(2) Связанные переменные

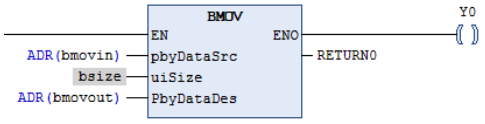
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
wData	Данные, которые необходимо оценивать	WORD	遵照数据类型	0	Данные, которые необходимо оценивать (десятичные значения)

uiBit	Количество битов данных	UINT	遵照数据类型	0	Количество битов данных (двоичные битовые числа)

(3) Функция

- Определите состояние бита uiBit данных wData, сохраните результат в BON и выведите его с правой стороны.

(4) Пример программы

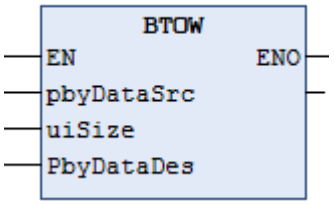
ДЕМО	LD	ST
Задание переменных	VAR wData0: WORD; uibit0: UINT; Y0: BOOL; RETURN0: BOOL; END_VAR	
Пример программы		Y0:=BON(wData:=wData0, uiBit:=uisize0);
Результат операции		
Анализ результатов	Введите значение 22 для wData0 (двоичное отображение 10110), значение 2 для uibit0, то есть bit2, со значением 1, и выведите результат как True.	

1-4-4-3. Объединение данных в единице байта BTOW

(1) Обзор команды

Команда, объединяющая младшие 8 бит (младшие байты) непрерывных 16-битных данных.

Объединение данных в единице байта [BTOW]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BTOW	Объединение данных в единице байта		BTOW(pbyDataSrc:= , uiSize:= , PbyDataDes:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
pbyDataSrc	Начальный адрес объединенных данных	POINTER TO BYTE	-	-	Начальный адрес объединенных данных
uiSize	Количество объединенных данных	UINT	В качестве типа данных	0	Количество объединенных данных
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
PbyDataDes	Целевой адрес для хранения объединенных данных	POINTER TO BYTE	-	-	Целевой адрес для хранения объединенных данных

(3) Функция

- Введите исходные данные в byDataSrc и количество объединенных данных в uiSize. Эта команда объединяет младшие 8 бит нескольких 16-битных данных во входных исходных данных, чтобы сформировать новые 16-битные данные. Новые сгенерированные данные сохраняются в начальный адрес pwDataDes, а старшие 8 битов pbyDataSrc игнорируются.

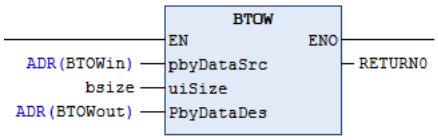
(4) Примечание

- Если значение SIZE равно 0, функция возвращает значение True.

(5) Пример программы

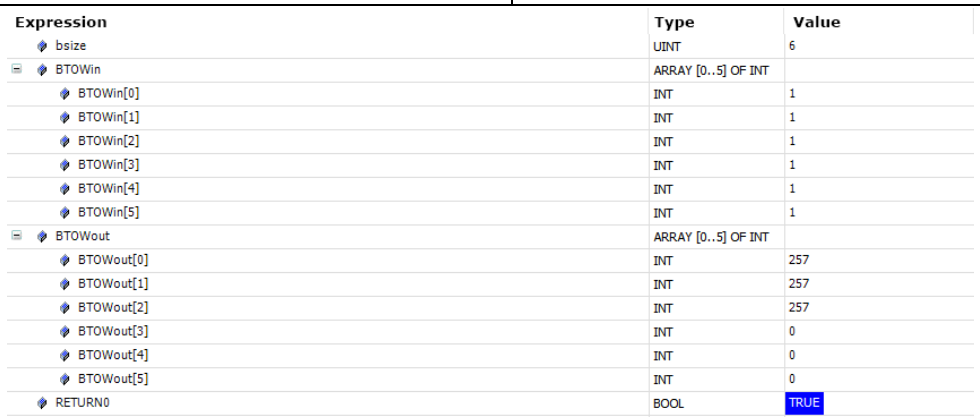
<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    bsize:UINT;
    BTOWin:ARRAY[0..5] OF INT;
    BTOWout:ARRAY[0..5] OF INT;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
BTOW(pbyDataSrc:=ADR(BTOWin) ,
     uiSize:=bsize ,
     PbyDataDes:=ADR(BTOWout) );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">


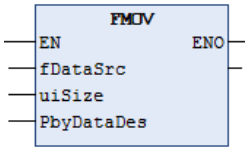
</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Bsize представляет собой количество байт, а данные устанавливаются равными 6. Преобразуйте два младших октета BTOWin [0] - BTOWin [5] по порядку, т.е. данные 01, в новые 16-битные данные (0101) и последовательно сохраните их в BTOWout [0] - BTOWout [2].</td></tr>
</table>

1-4-4-4. Передача одного бита в несколько битов данных FMOV

(1) Обзор команды

Скопируйте указанный бит исходного массива данных в несколько битов целевого массива, например, передайте значение A1 [1] в A2 [0,1,2].

Передача одного бита в несколько битов данных [FMOV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
FMOV	Передача одного бита в несколько битов данных		FMOV(fDataSrc:= , uiSize:= , byDataDes:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
fDataSrc	Начальный адрес для репликации данных	REAL	В качестве типа данных	0	Начальный адрес для репликации данных
uiSize	Число переменных	UINT	В качестве типа данных	0	Число переменных
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
PbyDataDes	Начальный адрес для хранения данных	POINTER TO BYTE	-	-	Начальный адрес для хранения данных

(3) Функция

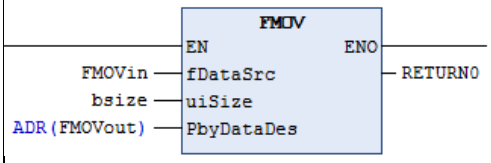
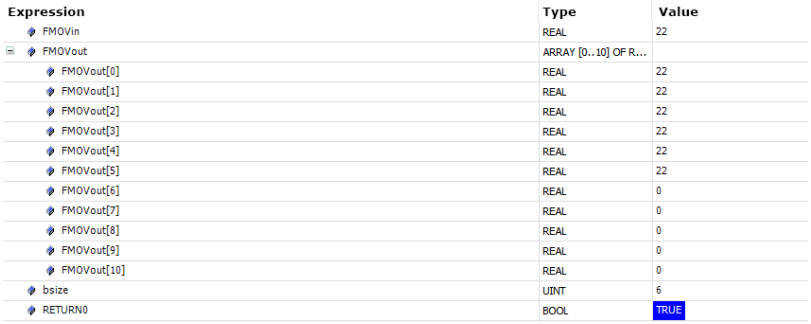
- Передача данных из fDataSrc в единицы uiSize по указанному начальному адресу PbyDataDes, при этом диапазон данных uiSize составляет от 0 до 1024.

(4) Примечание

- UiSize обозначает количество передач.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR FMOVin: REAL; FMOVout:ARRAY[0..10] OF REAL; bsize:UINT; RETURN0: BOOL; END_VAR

Пример программы		FMOV(fDataSrc:=FMOVin , uiSize:=bsize , PbyDataDes:=ADR(FMOVout));
Результат операции		

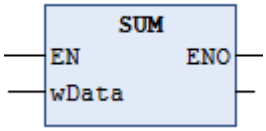
Expression	Type	Value
FMOVin	REAL	22
FMOVout	ARRAY [0..10] OF R...	
FMOVout[0]	REAL	22
FMOVout[1]	REAL	22
FMOVout[2]	REAL	22
FMOVout[3]	REAL	22
FMOVout[4]	REAL	22
FMOVout[5]	REAL	22
FMOVout[6]	REAL	0
FMOVout[7]	REAL	0
FMOVout[8]	REAL	0
FMOVout[9]	REAL	0
FMOVout[10]	REAL	0
bsize	UINT	6
RETURN0	BOOL	TRUE

1-4-4-5. Бит включения SUM

(1) Обзор команды

Подсчитайте количество "1" в двоичных данных, сохраните статистические результаты в SUM и выведите их с правой стороны.

Сумма битов включения [SUM]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
SUM	Сумма битов включения		SUM(wData:=);

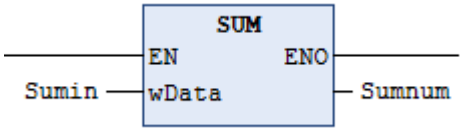
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
wData	Данные, которые необходимо оценивать	WORD	В качестве типа данных	0	Данные, которые необходимо оценивать (десятичные значения)

(3) Функция

- Подсчитайте количество "1" в двоичных данных, сохраните статистические результаты в SUM.

(4) Пример программы

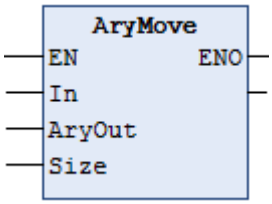
ДЕМО	LD	ST
Задание переменных	VAR Sumin: WORD; sumnum: INT; END_VAR	
Пример программы		SUMnum:=SUM(wData:=SUMin);
Результат операции		
Анализ результатов	Вход данных Sumin равен 22 (двоичное отображение 10110), поэтому для Sumnum отображается 3.	

1-4-4-6. Передача массива AryMove

(1) Обзор команды

Передача нескольких элементов массива.

Передача массива [AryMove]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryMove	Передача массива		AryMove(In:= , AryOut:= , Size:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Передаваемый массив	BOOL/BYTE/WORD /DWORD/LWORD/ USINT/UINT/UDINT /ULINT/SINT/INT/DINT /LINT/REAL/LREAL/TIME /DATE/TOD/DT/STRING	В качестве типа данных	-	Передаваемый массив
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
AryOut	Массив с результатами передачи	BOOL/BYTE/WORD /DWORD/LWORD/ USINT/UINT/UDINT /ULINT/SINT/INT/DINT /LINT/REAL/LREAL/TIME /DATE/TOD/DT/STRING	В качестве типа данных	-	Массив с результатами передачи
Size	Количество передаваемых элементов	UINT	В качестве типа данных	0	Количество передаваемых элементов

(3) Функция

- Передача элементов "Size", начиная с In[0] массива объектов передачи In[], в элементы, начиная с AryOut[0] массива с результатами передачи AryOut[].

(4) Примечание

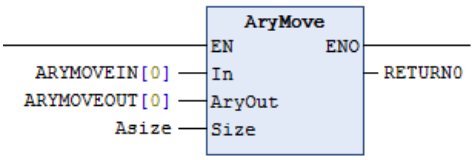
- Типы данных элементов In[] и AryOut[] должны совпадать, в противном случае возвращаемое значение функции выполнения будет FALSE, а AryOut[] останется неизменным.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Во избежание исключений ПЛК длина In[] не должна превышать размер памяти AryOut[].
- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в In[] и AryOut[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    ARYMOVEIN:ARRAY[0..5] OF INT;
    ARYMOVEOUT:ARRAY[0..5] OF INT;
    Asize: UINT;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
AryMove(In:=ARYMOVEIN[0],
    AryOut:=ARYMOVEOUT[0] ,
    Size:=bsize );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
ARYMOVEIN	ARRAY [0..5] OF INT	
ARYMOVEIN[0]	INT	11
ARYMOVEIN[1]	INT	22
ARYMOVEIN[2]	INT	33
ARYMOVEIN[3]	INT	44
ARYMOVEIN[4]	INT	55
ARYMOVEIN[5]	INT	66
ARYMOVEOUT	ARRAY [0..5] OF INT	
ARYMOVEOUT[0]	INT	11
ARYMOVEOUT[1]	INT	22
ARYMOVEOUT[2]	INT	33
ARYMOVEOUT[3]	INT	44
ARYMOVEOUT[4]	INT	55
ARYMOVEOUT[5]	INT	66
Asize	UINT	6
RETURN0	BOOL	TRUE

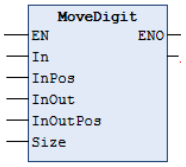
</td></tr>
</table>

1-4-4-7. Передача цифр MoveDigit

(1) Обзор команды

Передача нескольких цифр в последовательности (1 цифра — это 4 бита).

Передача цифр [MoveDigit]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
MoveDigit	Передача цифр		MoveDigit(In:= , InPos:= , InOut:= , InOutPos:= , Size:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Отправитель	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Отправитель
Inpos	Числовое положение отправителя	USINT	В качестве типа данных	0	Положение числа, переданного в "In"
InOut	Получатель	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Получатель
InOutPos	Числовое положение получателя	USINT	В качестве типа данных	0	Числовое положение получателя в "Out"
Size	Количество передаваемых цифр	USINT	В качестве типа данных	1	Количество передаваемых цифр

(3) Функция

- Передача цифр "Size", начиная с числового положения "InPos" отправителя "In" до числового положения "InOutPos" получателя "InOut". Одна цифра равна 4 битам.

(4) Примечание

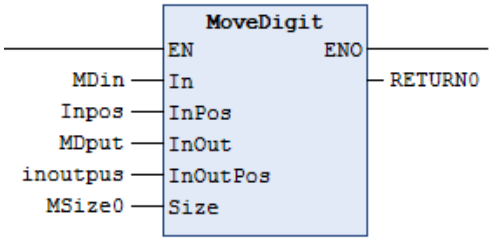
- Если указанное числовое положение получателя превышает максимальное положение "InOut", избыточные данные будут сохранены в минимальном положении "InOut".
- Если указанное числовое положение отправителя превышает максимальное положение "In", избыточные данные будут переданы в минимальное положение "In".
- Если значение "Size" равно 0, то функция возвращает значение True, а "InOut" остается неизменным.
- Исключения возникают в описанных ниже случаях. Возвращаемое значение функции становится FALSE, а "InOut" остается неизменным.
  - Когда значение "InPos" превышает допустимый диапазон.
  - Когда значение "InOutPos" превышает допустимый диапазон.
  - Когда значение "Size" превышает допустимый диапазон.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    MDin:WORD;
    MDput:WORD;
    Inpos: USINT;
    inoutpus: USINT;
    MSize0: USINT;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
MoveDigit(In:= MDin,
    InPos:= Inpos,
    InOut:=MDput,
    InOutPos:= inoutpus,
    Size:=MSize0 );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
MDin	WORD	16#1234
MDput	WORD	16#9934
Inpos	USINT	16#00
inoutpus	USINT	16#00
MSize0	USINT	16#02
RETURN0	BOOL	TRUE

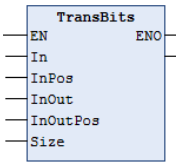
</td></tr>
</table>

1-4-4-8. Передача нескольких битов TransBits

(1) Обзор команды

Передача нескольких битов последовательности.

Передача нескольких битов [TransBits]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
TransBits	Передача нескольких битов		TransBits(In:= , InPos:= , InOut:= , InOutPos:= , Size:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Отправитель	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Отправитель
Inpos	Битовое положение отправителя	USINT	0~’in’ биты -1	-	Положение переданного бита в "In"
InOut	Получатель	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Получатель
InOutPos	Битовое положение получателя	USINT	0~’inout’ биты -1	-	Положение бита получателя в "Out"
Size	Количество передаваемых цифр	USINT	0~’in’ и ’inout’ биты -1	-	Количество передаваемых цифр

(3) Функция

- Передача битов "Size", начиная с битового положения "InPos" отправителя "In" до битового положения "InOutPos" получателя "InOut".

(4) Примечание

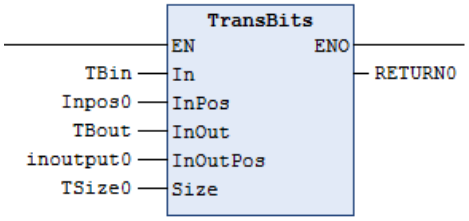
- Если значение "Size" равно 0, оно не передается, а функция возвращает значение True.
- Биты в "InOut", которые не относятся к передаче, остаются неизменными.
- В указанных ниже случаях возникает исключение, при этом функция возвращает значение FALSE, а InOut остается неизменным.
  - Когда значение "InPos" превышает допустимый диапазон.
  - Когда значение "InOutPos" превышает допустимый диапазон.
  - Когда значение "Size" превышает допустимый диапазон.
  - Когда заданные значения "InPos" и "Size" превышают количество цифр в "In".
  - Когда заданные значения "InOutPos" и "Size" превышают количество цифр в "InOut".

5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    RETURN0: BOOL;
    TBin:WORD;
    TBout:WORD;
    Inpos0: USINT;
    inoutput0: USINT;
    TSize0: USINT;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
TransBits(In:= TBin,
    InPos:= Inpos0,
    InOut:=TBout ,
    InOutPos:=inoutput0 ,
    Size:=TSize0 );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
RETURN0	BOOL	TRUE
TBin	WORD	2#0000000000000000
TBout	WORD	2#0000000000011100
Inpos0	USINT	2#00000000
inoutput0	USINT	2#00000000
TSize0	USINT	2#00000010

</td></tr>
</table>

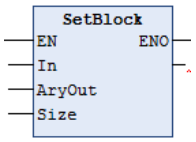
Анализ результатов	Задайте TBin (данные установлены на 0), начиная с input0 (данные установлены на 0). Два фрагмента данных передаются в TBout (начальный бит данных 2#11111), начиная с inoutpos0 (значение 0), два фрагмента данных передаются TBout (данные TBout передаются в 2#11100), и функция возвращает значение TRUE.

1-4-4-9. Задание блока SetBlock

(1) Обзор команды

Передача значений переменных и констант в несколько элементов массива.

Задание блока [SetBlock]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
SetBlock	Задание блока		SetBlock(In:= , AryOut:= , Size:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Источник передачи	BOOL/BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Передаваемый массив
Size	Количество передаваемых элементов	UINT	В качестве типа данных	0	Количество передаваемых элементов

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AryOut	Целевой массив передачи	BOOL/BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Массив с результатами передачи

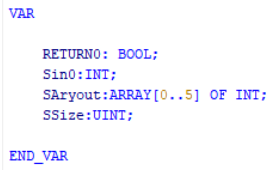
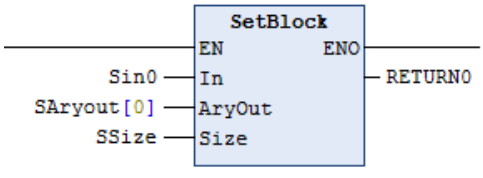
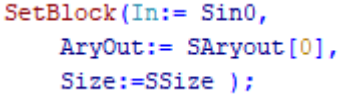
(3) Функция

- Передача значения источника передачи "In" в положения "Size", начиная с AryOut[0] целевого массива передачи AryOut[].

(4) Примечание

- Если типы данных "In" и "AryOut" разные, функция возвращает значение FALSE, а "AryOut" остается неизменным.
- Размеры областей "In" и "AryOut" должны быть одинаковыми, чтобы исключить сбои в работе ПЛК.
- Если значение "Size" равно 0, то функция возвращает значение True, а AryOut[] остается неизменным.
- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в AryOut[].
- Если "In" и AryOut[] являются неподдерживаемыми типами данных, при компиляции возникнет исключение.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		

Результат операции	

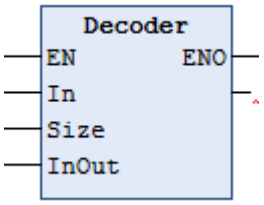
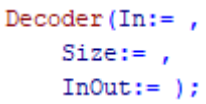
Expression	Type	Value
RETURN0	BOOL	TRUE
Sin0	INT	11
SAryout	ARRAY [0..5] OF INT	
SAryout[0]	INT	11
SAryout[1]	INT	11
SAryout[2]	INT	11
SAryout[3]	INT	11
SAryout[4]	INT	0
SAryout[5]	INT	0
SSize	UINT	4

1-4-4-10. Дешифратор битов Decoder

(1) Обзор команды

Настройте один из упорядоченных элементов, состоящий из 256 бит, как True, а остальные биты - как FALSE.

Дешифратор битов [Decoder]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Декодер	Дешифратор битов		


```
Decoder(In:= ,
    Size:= ,
    InOut:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Положение бита преобразования	BYTE	В качестве типа данных	0	Положение бита преобразования
Size	Биты преобразования	USINT	0~8	1	Биты преобразования

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
InOut	Массив объектов преобразования	BOOL/BYTE/WORD /DWORD/LWORD	В качестве типа данных	-	Массив объектов преобразования

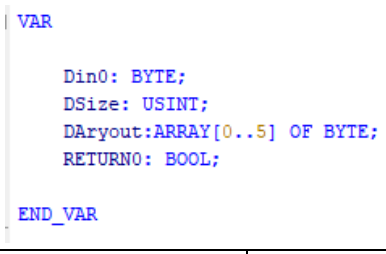
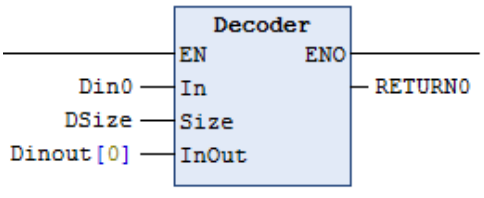
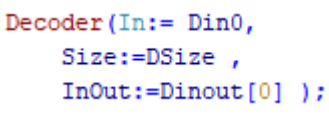
(3) Функция

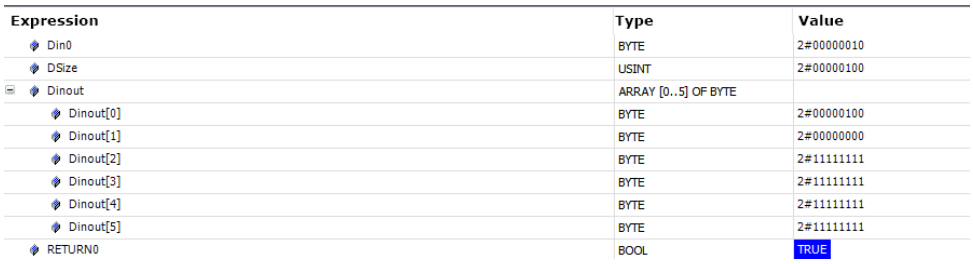
- Установите указанный бит из 2 битов "Size", начиная с элемента InOut[0] в массиве объектов преобразования InOut[], в True, а остальные биты - в FALSE.
- Среди них положение бита, устанавливаемое как True, задается по положению бита преобразования "In".
- Входные параметры, передаваемые в InOut[], должны быть указаны вместе с номером элемента, например array[3].

(4) Примечание

- Во избежание исключений ПЛК значение "Size" не должно превышать количество цифр в элементе массива "InOut".
- Во избежание сбоев в работе ПЛК значение "Size" не должно превышать допустимый диапазон.
- Если значение "Size" равно 0, то функция возвращает значение True, а InOut[] остается неизменным.
- Для InOut[] требуются булевы и битовые строки, в противном случае программа выдаст ошибку во время выполнения.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		

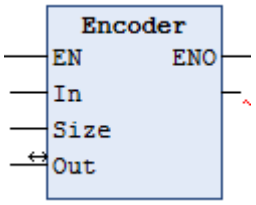
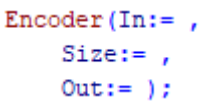
Результат операции	
Анализ результатов	Установите данные массива Dinout на 2#11111111. Выполните команду, чтобы установить 2^4(DSize) бита, начиная с данных Dinout[0], на 0, кроме данных bit2(Din0), которые устанавливаются на 1.

1-4-4-11. Шифратор битов Encoder

(1) Обзор команды

Вычислите положение бита со значением True в элементе перестановки, состоящем максимум из 256 битов.

Шифратор битов [Encoder]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Encoder	Шифратор битов		

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Расположение объектов преобразования	BOOL/BYTE/WORD /DWORD/LWORD	В качестве типа данных	-	Расположение объектов преобразования

Size	Биты преобразования	USINT	0~8	1	Биты преобразования
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
AryOut	Результат преобразования	BYTE	В качестве типа данных	0	Результат преобразования

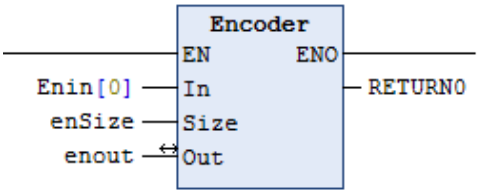
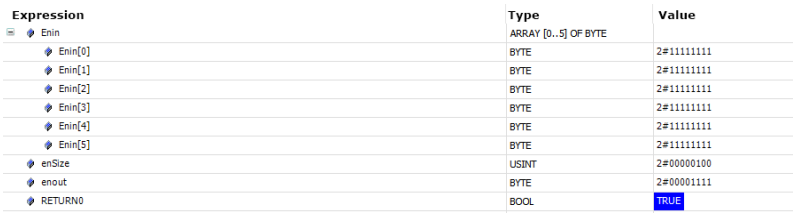
(3) Функция

- Вычисление положения бита со значением True в любом диапазоне расположения объектов преобразования In[].
- Диапазоном для вычисления положений битов будет 2 бита "Size", начиная с In[0]. Положения битов со значением True в этом диапазоне представлены в двоичном виде и сохраняются в результате преобразования "Out". После этого в остальных битах "Out" сохраняется FALSE.
- Если в заданном диапазоне есть несколько битов со значением True, вычислите положение самого старшего бита.
- Входные параметры, передаваемые в In[], должны быть указаны вместе с номером элемента, например Array[3].
- Если "Size"=USINT#4 и In[] — это BYTE, то, поскольку "Size"=USINT#4, диапазон объектов для вычисления положения бита TRUE равен 2^4=16, начиная с In[0].

(4) Примечание

- Если значение "Size" равно 0, то функция возвращает значение True, а "Out" остается неизменным.
- Когда значение "Size" превышает допустимый диапазон, исключите сбои в работе ПЛК.
- Если значение "Size" превышает количество цифр в упорядоченных элементах In[], может быть получено неверное значение.
- Если все значения, указанные в "Size" в бите In[], равны FALSE, программа сталкивается с исключением, и функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Для In[] требуются булевы и битовые строки, в противном случае программа выдаст ошибку при компиляции.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR Enin:ARRAY[0..5] OF BYTE; enSize: USINT; enout: BYTE; RETURN0: BOOL; END VAR	
Пример программы		Encoder(In:=enin[0] , Size:=enSize , Out:=enout);
Результат операции		

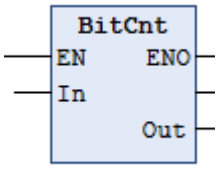
Expression	Type	Value
Enin	ARRAY [0..5] OF BYTE	
Enin[0]	BYTE	2#11111111
Enin[1]	BYTE	2#11111111
Enin[2]	BYTE	2#11111111
Enin[3]	BYTE	2#11111111
Enin[4]	BYTE	2#11111111
Enin[5]	BYTE	2#11111111
enSize	USINT	2#00000100
enout	BYTE	2#00001111
RETURN0	BOOL	TRUE

1-4-4-12. Число битов BitCnt

(1) Обзор команды

Вычислите общее количество битов в строке со значением True.

Число битов [BitCnt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BitCnt	Число битов		BitCnt(In:= , Out=>);

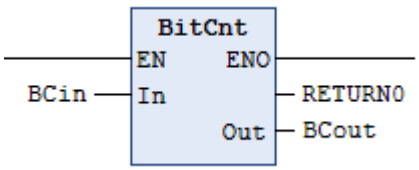
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Объект подсчета	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Объект подсчета
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Вычисление количества битов	USINT	В качестве типа данных	0	Вычисление количества битов со значением True в объекте

(3) Функция

- Вычисление общего количества битов в объекте подсчета "In", когда значение True

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR BCin: WORD; BCout: USINT; RETURN0: BOOL; END_VAR	
Пример программы		BitCnt(In:=BCin , Out=> BCout);
Результат операции		

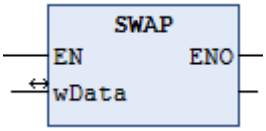
Expression	Type	Value
BCin	WORD	17
BCout	USINT	2
RETURN0	BOOL	TRUE

1-4-4-13. Перестановка старших и младших байтов SWAP

(1) Обзор команды

Перестановка старших и младших 8 битов в 16-битных данных.

Перестановка старших и младших байтов [SWAP]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
SWAP	Перестановка старших и младших байтов		SWAP(wData:=);

(2) Связанные переменные

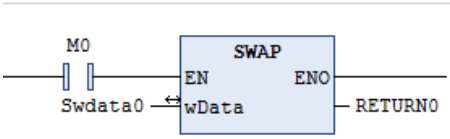
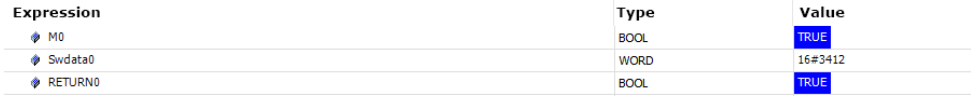
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
wData	Данные, которыми необходимо управлять	Слово	В качестве типа данных	-	Данные, которыми необходимо управлять

(3) Функция

- Перестановка старших и младших 8 битов в 16-битных данных.

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR M0:BOOL; Swdata0: WORD; RETURN0: BOOL; END_VAR	

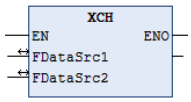
Пример программы		IF M0 THEN SWAP(wData:=Swdata0); END_IF
Результат операции		
Анализ результатов	M0 устанавливается в значение True, и старшие и младшие биты Swdata0 преобразуются из 16#1234 в 16#3412; M0 постоянно срабатывает, а старшие и младшие биты будут продолжать изменяться.	

1-4-4-14. Обмен данными XCH

(1) Обзор команды

Перестановка двух чисел с плавающей точкой.

Обмен данными [XCH]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
XCH	Обмен данными		XCH(FDataSrc1:= , FDataSrc2:=);

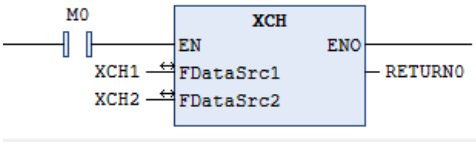
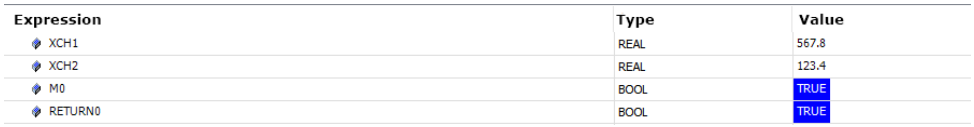
(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
FDataSrc1	Данные1, подлежащие обмену	REAL	В качестве типа данных	0	Данные, подлежащие обмену
FDataSrc1	Данные2, подлежащие обмену	REAL	В качестве типа данных	0	Данные, подлежащие обмену

(3) Функция

- Перестановка двух чисел с плавающей точкой.

(4) Пример программы

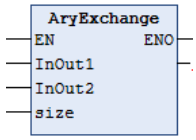
ДЕМО	LD	ST
Задание переменных		VAR XCH1: REAL; XCH2: REAL; M0:BOOL; RETURN0: BOOL; END_VAR
Пример программы		IF M0 THEN XCH(FDataSrc1:= XCH1, FDataSrc2:=XCH2); END_IF
Результат операции	Перед операцией  После операции 	
Анализ результатов	Установите M0 в True. Будет происходить обмен данными XCH1 и XCH2, при этом данные 123.4 в XCH1 будут заменяться на данные 567.8 в XCH2; M0 постоянно срабатывает, и будет продолжаться обмен данными XCH1 и XCH2.	

1-4-4-15. Обмен данными массива AryExchange

(1) Обзор команды

Перестановка элементов в двух массивах.

Обмен данными массива [AryExchange]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryExchange	Обмен данными массива		AryExchange(InOut1:= , InOut2:= , size:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Передаваемый массив	BOOL/BYTE/WORD/ DWORD/LWORD/ USINT/UINT/UDINT/ ULINT/SINT/INT/DINT /LINT/REAL/LREAL TIME/DATE/TOD/DT/ STRING	В качестве типа данных	-	Передаваемый массив
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
AryOut	Массив с результатами передачи	BOOL/BYTE/WORD/ DWORD/LWORD/ USINT/UINT/UDINT/ ULINT/SINT/INT/DINT /LINT/REAL/LREAL TIME/DATE/TOD/DT/ STRING	В качестве типа данных	-	Массив с результатами передачи
Size	Количество передаваемых элементов	UINT	В качестве типа данных	0	Количество передаваемых элементов

(3) Функция

- Обмен значениями "InOut1" и "InOut2".
- Поменяйте местами элементы "Size", начиная с InOut1[0] в массиве заменяемых объектов InOut1[], с элементами "Size", начиная с InOut2[0] в InOut2[].

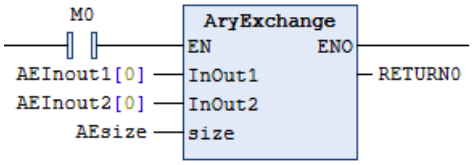
(4) Примечание

- Типы данных элементов InOut1[] и InOut2[] должны совпадать, в противном случае возвращаемое значение функции будет FALSE, а InOut1[] и InOut2[] останутся неизменными.
- Если значение "Size" равно 0, то функция возвращает значение True, а InOut1[] и InOut2[] остаются неизменными.
- Элементы InOut1[] и InOut2[] имеют тип STRING, и длина строки одного из них не должна быть больше объема памяти другого элемента во избежание исключений ПЛК.
- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в InOut1[] или InOut2[].

(5) Пример программы

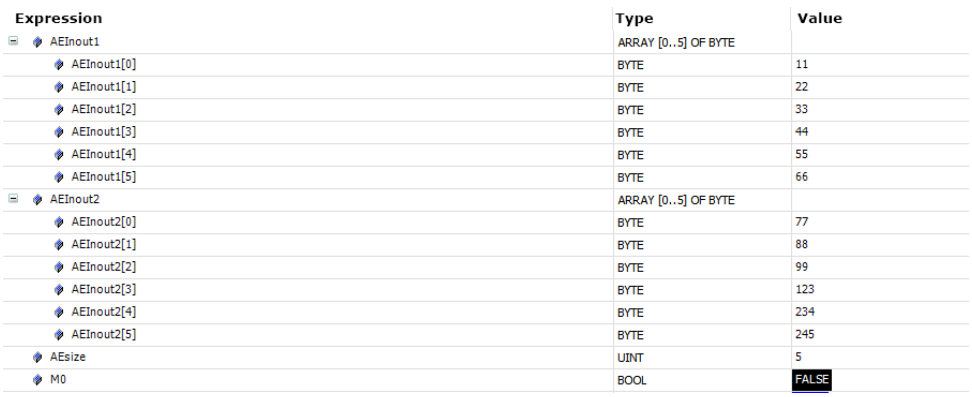
<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    AEInout1:ARRAY[0..5] OF BYTE;
    AEInout2:ARRAY[0..5] OF BYTE;
    AEsize: UINT;
    M0:BOOL;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


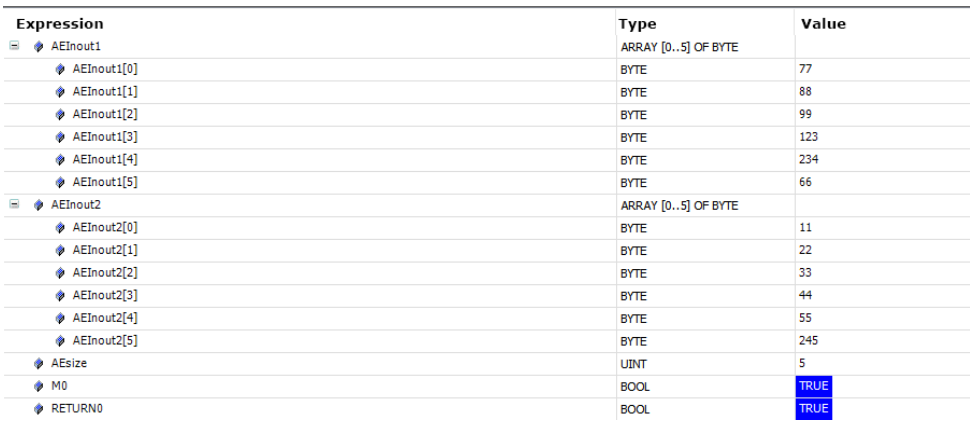
</td><td>

```
IF M0 THEN
    AryExchange(InOut1:= AEInout1[0],
        InOut2:= AEInout2[0],
        size:=AEsize );
END_IF
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">
Перед операцией


Expression	Type	Value
AEInout1	ARRAY [0..5] OF BYTE	
AEInout1[0]	BYTE	11
AEInout1[1]	BYTE	22
AEInout1[2]	BYTE	33
AEInout1[3]	BYTE	44
AEInout1[4]	BYTE	55
AEInout1[5]	BYTE	66
AEInout2	ARRAY [0..5] OF BYTE	
AEInout2[0]	BYTE	77
AEInout2[1]	BYTE	88
AEInout2[2]	BYTE	99
AEInout2[3]	BYTE	123
AEInout2[4]	BYTE	234
AEInout2[5]	BYTE	245
AEsize	UINT	5
M0	BOOL	FALSE

После операции
</td></tr>
</table>

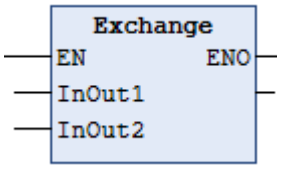
	 Expression / Type / Value AEInout1 — ARRAY [0..5] OF BYTE AEInout1[0] — BYTE — 77 AEInout1[1] — BYTE — 88 AEInout1[2] — BYTE — 99 AEInout1[3] — BYTE — 123 AEInout1[4] — BYTE — 234 AEInout1[5] — BYTE — 66 AEInout2 — ARRAY [0..5] OF BYTE AEInout2[0] — BYTE — 11 AEInout2[1] — BYTE — 22 AEInout2[2] — BYTE — 33 AEInout2[3] — BYTE — 44 AEInout2[4] — BYTE — 55 AEInout2[5] — BYTE — 245 AEsize — UINT — 5 M0 — BOOL — TRUE RETURN0 — BOOL — TRUE
Анализ результатов	Установите M0 в True, и будет происходить обмен данными AEsize между AEInOut1 и AEinout2; M0 постоянно срабатывает, и будет продолжаться обмен данными XCH1 и XCH2.

1-4-4-16. Обмен данными Exchange

(1) Обзор команды

Перестановка значений в двух переменных.

Обмен данными [Exchange]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Exchange	Обмен данными	 Exchange EN ENO InOut1 InOut2	`Exchange(InOut1:= , InOut2:= );`

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
InOut1	Объект	BOOL/BYTE/WORD/	В качестве	0	Объект

	обмена 1	DWORD/LWORD/USINT/UINT/ UDINT/ULINT/SINT/INT/DINT /LINT/REAL/LREAL/TIME/DATE/ TOD/DT/STRING	типа данных		обмена 1
InOut2	Объект обмена 2	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/LINT/REAL / LREAL/TIME/DATE/TOD/DT/STRING	В качестве типа данных	0	Объект обмена 2

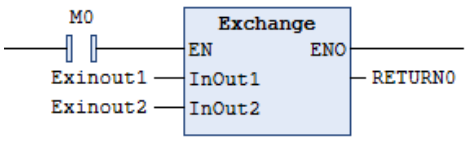
(3) Функция

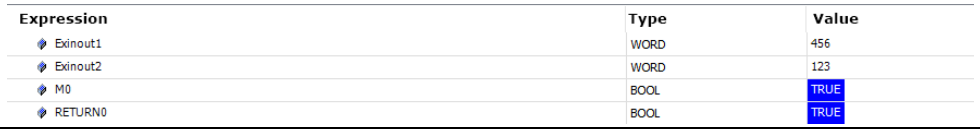
- Обмен значениями "InOut1" и "InOut2".

(4) Примечание

- Типы данных элементов "InOut1" и "InOut2" должны совпадать, иначе функция выполнения возвращает значение FALSE.
- Если эта команда используется в программе ST, возвращаемое значение "Out" не будет использовано.
- Переменные "InOut1" и "InOut2" имеют тип данных STRING. Если длина какой-либо строки не совпадает с размером другого элемента, перестановка не произойдет.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR Exinout1: WORD; Exinout2: WORD; M0:BOOL; RETURN0: BOOL; END_VAR	
Пример программы		IF M0 THEN Exchange(InOut1:=Exinout1, InOut2:=Exinout2); END_IF

Результат операции	Перед операцией  После операции 
Анализ результатов	Установите M0 в True, и будет происходить обмен данными ExInOut1 и Exinout2; M0 постоянно срабатывает, а ExInOut1 и Exinout2 всегда будут переключаться.

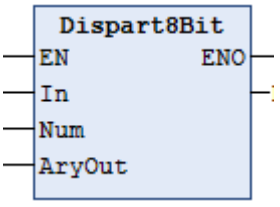
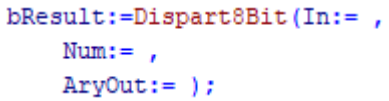
1-4-5. Команды разделения и объединения данных

1-4-5-1. Разделение данных на единицы байта Dispart8Bit

(1) Обзор команды

Разделение битовых строк на единицы 1 байта.

Разделение данных на единицы байта [Dispart8Bit]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Dispart8Bit	Разделение данных на единицы байта		


```
bResult:=Dispart8Bit(In:= ,
    Num:= ,
    AryOut:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Отдельные объекты	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Битовая строка отдельного объекта
Num	Отдельные байты	USINT	В качестве типа данных	1	Количество разделенных байтов
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Отдельный массив с результатами	BYTE	В качестве типа данных	-	Отдельный массив с результатами

Примечание: Для входной и выходной переменной AryOut нужно указать элементы массива.

(3) Функция

- Разделение "In" отдельного объекта на единицы 1 байта и сохранение в отдельном массиве с результатами AryOut[ ].
- Сначала разделите "In" на единицы 1 байта. Затем сохраните младший байт в начальном адресе AryOut[], установленном на выводе AryOut. Сохраните следующий байт в адресе AryOut[]+1. Выполните ту же обработку для определенного количества разделенных байтов "Num".

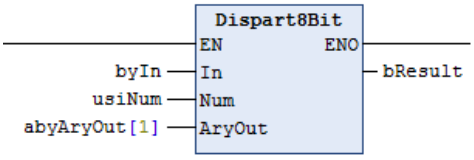
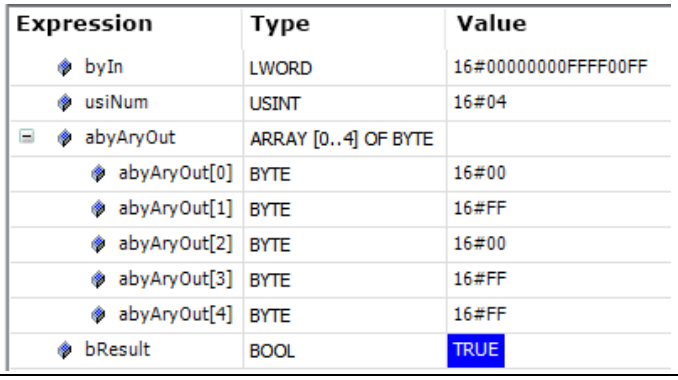
(4) Примечание

- Когда значение "Num" превышает допустимый диапазон, функция возвращает значение FALSE, а значение AryOut[] остается неизменным.
- Значение "Num" не должно превышать область массива в AryOut[].

(5) Пример программы

ДЕМО	LD	ST

Задание переменных	VAR
	byIn :LWORD:=16#FFFF00FF;
	usiNum :USINT:=4;
	abyAryOut:ARRAY[0..4] OF BYTE:=[0,0,0,0,0];
	bResult :BOOL;
	END_VAR	

Пример программы		1 bResult:=Dispart8Bit(In:= byIn, 2 Num:= usiNum, 3 AryOut:= abyAryOut[1]);
Результат операции		

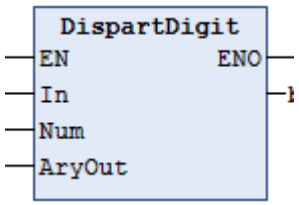
Expression	Type	Value
byIn	LWORD	16#00000000FFFF00FF
usiNum	USINT	16#04
abyAryOut	ARRAY [0..4] OF BYTE	
abyAryOut[0]	BYTE	16#00
abyAryOut[1]	BYTE	16#FF
abyAryOut[2]	BYTE	16#00
abyAryOut[3]	BYTE	16#FF
abyAryOut[4]	BYTE	16#FF
bResult	BOOL	TRUE

1-4-5-2. Разделение на 4 бита DispartDigit

(1) Обзор команды

Разделение битовых строк на единицы 4 бита.

Разделение на 4 бита [DispartDigit]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
DispartDigit	Разделение на 4 бита		bResult:=DispartDigit(In:= , Num:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Отдельные объекты	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Битовые строки отдельного объекта
Num	Отдельные биты	USINT	В качестве типа данных	1	Количество разделенных разрядов
VAR_IN_OUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Массив AryOut[]	Отдельный массив с результатами	BYTE	В качестве типа данных	-	Отдельный массив с результатами

Примечание: Для входной и выходной переменной AryOut нужно указать элементы массива.

(3) Функция

- Разделение "In" отдельного объекта на единицы 4 разрядов (цифровые единицы) и сохранение в отдельном массиве с результатами AryOut[].
- Сначала разделите "In" на единицы 4 разрядов. Затем сохраните 4 младших бита в начальном адресе AryOut[], установленном на выводе AryOut. На этом этапе добавьте 16#0 в биты с 4 по 7 AryOut[] типа BYTE.
- Выполните ту же обработку для определенного количества разделенных разрядов "Num".

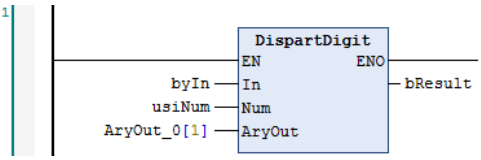
(4) Примечание

- Если значение "Num" равно 0, то функция возвращает значение True, а значение AryOut[] остается неизменным.
- Когда значение "Num" превышает допустимый диапазон, функция возвращает значение FALSE, а значение AryOut[] остается неизменным.
- Значение "Num" не должно превышать область массива в AryOut[].

(5) Пример программы

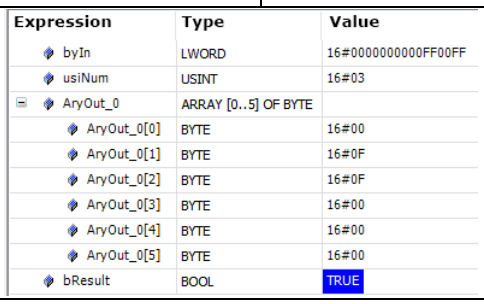
<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    byIn:LWORD:=16#FF00FF;
    usiNum:USINT:=3;
    AryOut_0:ARRAY[0..5] OF BYTE:=[0,0,0,0,0,0];
    bResult:BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
1  bResult:=DispartDigit(In:= byIn,
2      Num:= usiNum,
3      AryOut:= AryOut_0[1]);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">


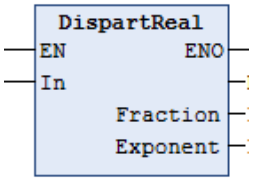
</td></tr>
</table>

1-4-5-3. Разложение действительных чисел на мантиссы и порядки DispartReal

(1) Обзор команды

Разложите действительные числа на мантиссы и порядки со знаком.

Разложение действительных чисел на мантиссы и порядки [DispartReal]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
DispartReal	Разложение действительных чисел на мантиссы и порядки		`bResult:=DispartReal(In:= , Fraction=> , Exponent=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Действительное число	REAL/LREAL	В качестве типа данных	-	Разлагаемое действительное число
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Fraction	Мантисса со знаком	LINT	В качестве типа данных	0	Мантисса со знаком
Exponent	Порядок	INT	В качестве типа данных	0	Порядок

(3) Функция

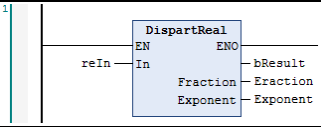
- Разложение "In" действительного числа на мантиссу "Fraction" и порядок "Exponent" со знаком.
- Если "In" имеет тип REAL, то "Fraction" — это 7-битное целое число. Если "In" имеет тип LREAL, то "Fraction" — это 15-битное целое число. Эффективный диапазон "Fraction" зависит от типа данных "In".

Тип данных “In”	Эффективный диапазон "Fraction"
REAL	от -9999999 до 9999999
LREAL	от -999999999999999 до 999999999999999

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    reIn:REAL:=12.34567E+8;
    Eraction:LINT;
    Exponent:INT;
    bResult:BOOL;
END_VAR
```




```
1  bResult:=DispartReal(In:= reIn,
2      Fraction=> Eraction,
3      Exponent=> Exponent);
```

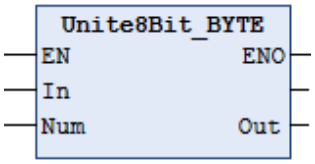
Expression	Type	Value
reIn	REAL	1.234567E+09
Eraction	LINT	1234567
Exponent	INT	3
bResult	BOOL	TRUE

1-4-5-4. Группа объединения данных в единицах по 1 байту Unite8Bit_XXX

(1) Обзор команды

Объединение данных в единицах по 1 байту в битовую строку.

Группа объединения данных в единицах по 1 байту [Unite8Bit_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Unite8Bit_XXX	Группа объединения данных в единицах по 1 байту		bResult:=Unite8Bit_BYTE(In:= , Num:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In[]	Объединенный массив объектов	BYTE	В качестве типа данных	-	Объединяемый массив
Num	Объединенные байты	USINT	В качестве типа данных	1	Объединяемые байты
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Объединенный результат	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Битовая строка объединенных результатов

Примечание: Входная переменная In должна передавать указанный элемент массива.

(3) Функция

- Объединение элементов объединенного массива объектов In[] для генерирования битовой строки "Out" с результатами объединения.
- Количество объединяемых элементов массива зависит от количества объединенных байтов "Num". Сначала объедините In[] с последовательными байтами "Num", чтобы создать битовую строку байтов "Num". В старшем бите примите за "Out" битовую строку 16#00 комбинации {(байты "Out") - "Num"}.
- Имя команды зависит от типа данных "Out". Например, если типом данных для "Out" является DWORD, команда будет иметь имя Unite8Bit_DWORD.

(4) Примечание

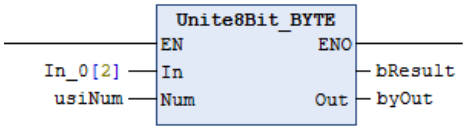
- Если значение "Num" равно 0, значение "Out" инициализируется равным 0, а функция возвращает значение True.
- Если значение "Num" превышает допустимый диапазон, функция возвращает значение FALSE, и значение "Out" инициализируется равным 0.
- Значение "Num" не может превышать диапазон массива In[] и не может превышать диапазон "Out".
- Входная переменная должна передавать указанные элементы массива, а длину массива нельзя определить. В противном случае, возвращаемое значение функции будет FALSE.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    In_0     :ARRAY[1..10] OF BYTE:=[10(1)];
    usiNum   :USINT:=1;
    byOut    :BYTE;
    bResult  :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=Unite8Bit_BYTE(In:= In_0[2],
2      Num:= usiNum,
3      Out=> byOut);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
In_0	ARRAY [1..10] OF BYTE	
In_0[1]	BYTE	16#01
In_0[2]	BYTE	16#01
In_0[3]	BYTE	16#01
In_0[4]	BYTE	16#01
In_0[5]	BYTE	16#01
In_0[6]	BYTE	16#01
In_0[7]	BYTE	16#01
In_0[8]	BYTE	16#01
In_0[9]	BYTE	16#01
In_0[10]	BYTE	16#01
usiNum	USINT	16#01
byOut	BYTE	16#01
bResult	BOOL	TRUE

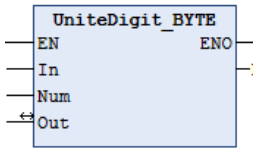
</td></tr>
</table>

1-4-5-5. Группа объединения 4-битных данных UniteDigit_XXX

(1) Обзор команды

Объединение 4-битных данных в битовую строку.

Группа объединения 4-битных данных [UniteDigit_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
UniteDigit_XXX	Группа объединения 4-битных данных		bResult:=UniteDigit_BYTE(In:= , Num:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Массив In[]	Объединенный массив объектов	BYTE	В качестве типа данных	-	Объединяемый массив
Num	Объединенные биты	USINT	В качестве типа данных	1	Объединяемые биты
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Объединенный результат	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Битовая строка объединенных результатов

(3) Функция

- Объединение элементов объединенного массива объектов In[] по младшим 4 битам (1 разряд=4 бита) для генерирования битовой строки "Out" с результатами объединения.

- Количество объединяемых элементов массива зависит от количества объединенных разрядов "Num". Сначала объедините каждые младшие 4 бита последовательных "Num", начиная с In[], чтобы создать битовую строку из битов "Num". В старшем бите примите за "Out" битовую строку 16#0 комбинации {(количество битов "Out") - "Num"}.

- Имя команды зависит от типа данных "Out". Например, если типом данных для "Out" является WORD, команда будет иметь имя UniteDigit_WORD.

(4) Примечание

- Если значение "Num" равно 0, значение "Out" будет равно 0, а функция возвращает значение True.

- Если значение "Num" превышает допустимый диапазон, функция возвращает значение FALSE, а значение "Out" остается неизменным.

- Значение "Num" не может превышать диапазон массива In[] и не может превышать диапазон "Out".

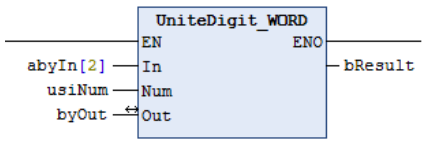
- Входная переменная In должна передавать указанные элементы массива, а длину массива нельзя определить. В противном случае, возвращаемое значение функции будет FALSE.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    abyIn:ARRAY[1..10] OF BYTE:=[10(1)];
    usiNum:USINT:=2;
    byOut:WORD;
    bResult:BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=UniteDigit_WORD(In:= abyIn[2],
2      Num:= usiNum,
3      Out:= byOut);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
abyIn	ARRAY [1..10] OF BYTE	
abyIn[1]	BYTE	16#01
abyIn[2]	BYTE	16#01
abyIn[3]	BYTE	16#01
abyIn[4]	BYTE	16#01
abyIn[5]	BYTE	16#01
abyIn[6]	BYTE	16#01
abyIn[7]	BYTE	16#01
abyIn[8]	BYTE	16#01
abyIn[9]	BYTE	16#01
abyIn[10]	BYTE	16#01
usiNum	USINT	16#02
byOut	WORD	16#0011
bResult	BOOL	TRUE

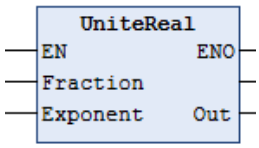
</td></tr>
</table>

1-4-5-6. Объединение мантиссы и порядков в действительные числа UniteReal

(1) Обзор команды

Объединение мантиссы и порядка со знаком для получения действительных чисел.

Объединение мантиссы и порядков в действительные числа [UniteReal]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
UniteReal	Объединение мантиссы и порядков в действительные числа		bResult:=UniteReal(Fraction:= , Exponent:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Fraction	Мантисса со знаком	DINT/LINT	В качестве типа данных	-	Мантисса со знаком
Exponent	Порядок	INT	В качестве типа данных	0	Порядок
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Действительное число	LREAL	В качестве типа данных	0	Действительные числа, полученные объединением

Примечание: Если тип данных "Fraction" — DINT, то тип данных "Out" — REAL; если тип данных "Fraction" — LINT, то тип данных "Out" — LREAL.

(3) Функция

- Объединение мантиссы "Fraction" и порядка "Exponent" со знаком, чтобы получить действительное число "Out".

(4) Примечание

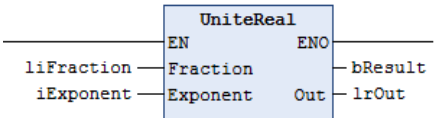
- Из-за разных значений "Fraction" и "Exponent" возможны ошибки при преобразовании целых чисел в действительные.
- Если результат объединения выходит за пределы допустимого диапазона "Out", если "Exponent" — положительное число, то значение "Out" будет бесконечным с тем же знаком, что и "Fraction". Если "Exponent" — отрицательное число, то значение "Out" будет равно 0.
- Для "Fraction" требуются типы данных Dint и Lint. Если тип данных не относится к целому числу, выдается сообщение об ошибке при компиляции. В противном случае, функция выполнения возвращает значение FALSE, а значение "Out" равно 0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    liFraction:LINT:=123456;
    iExponent :INT:=10;
    lrOut     :LREAL;
    bResult   :BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=UniteReal(Fraction:= liFraction,
2      Exponent:= iExponent,
3      Out=> lrOut);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
liFraction	LINT	123456
iExponent	INT	10
lrOut	LREAL	1.23456E+15
bResult	BOOL	TRUE

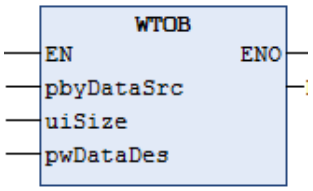
</td></tr>
</table>

1-4-5-7. Разделение данных на единицы байта WTOB

(1) Обзор команды

Разделение непрерывных 16-битных данных на единицы байта.

Разделение данных на единицы байта [WTOB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
WTOB	Разделение данных на единицы байта		bResult:=WTOB(pbyDataSrc:= , uiSize:= , pwDataDes:=);

(2) Связанная переменная

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
pbyDataSrc	Начальный адрес для разделения данных	POINTER_TO_BYTE	В качестве типа данных	-	Начальный адрес для разделения данных
uiSize	Количество разделенных данных	UINT	В качестве типа данных	0	Количество разделенных данных
pwDataDes	Конечный адрес назначения хранения данных	POINTER_TO_WORD	В качестве типа данных	-	Конечный адрес назначения хранения данных при разделении данных

(3) Описание функции

- Сохранение 16-битных данных, начиная с адреса pbyDataSrc, в байтах, в младшие 8 битов uiSize*2, начиная с адреса pwDataDes, а старшие 8 битов в 16#00.

(4) Примечание

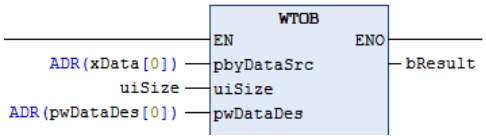
- Если значение SIZE равно 0, функция возвращает значение True.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    bResult:BOOL;
    xData:ARRAY[0..8] OF BYTE:=[9(16#11)];
    uiSize:UINT:=3;
    pwDataDes:ARRAY[0..10] OF WORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bResult:=WTOB(pbyDataSrc:= ADR(xData[0]),
2      uiSize:= uiSize,
3      pwDataDes:= ADR(pwDataDes[0]));
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
bResult	BOOL	TRUE
xData	ARRAY [0..8] OF BYTE	
xData[0]	BYTE	16#11
xData[1]	BYTE	16#11
xData[2]	BYTE	16#11
xData[3]	BYTE	16#11
xData[4]	BYTE	16#11
xData[5]	BYTE	16#11
xData[6]	BYTE	16#11
xData[7]	BYTE	16#11
xData[8]	BYTE	16#11
uiSize	UINT	16#0003
pwDataDes	ARRAY [0..10] OF WORD	
pwDataDes[0]	WORD	16#0011
pwDataDes[1]	WORD	16#0011
pwDataDes[2]	WORD	16#0011
pwDataDes[3]	WORD	16#0011
pwDataDes[4]	WORD	16#0011
pwDataDes[5]	WORD	16#0011
pwDataDes[6]	WORD	16#0000
pwDataDes[7]	WORD	16#0000
pwDataDes[8]	WORD	16#0000
pwDataDes[9]	WORD	16#0000
pwDataDes[10]	WORD	16#0000

</td></tr>
</table>

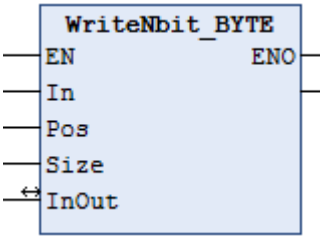
1-4-6. Команды считывания и записи данных

1-4-6-1. Запись N битов в группу WriteNbit_XXX

(1) Обзор команды

Запись нескольких бит в битовую последовательность.

Запись N битов в группу [WriteNbit_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
WriteNbit_XXX	Запись N битов в группу		WriteNbit_BYTE(In:= , Pos:= , Size:= , InOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Считывание источника	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Эта битовая последовательность позволяет считывать значение, записываемое в "InOut".
Pos	Положение записи	USINT	В качестве типа данных	0	Положение записываемого бита
Size	Размер записи	USINT	В качестве типа данных	0	Количество записываемых битов

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
InOut	Объект записи	То же, что для типа "In"	В качестве типа данных	-	Запись результата

(3) Функция

- Сначала считайте младший бит "Size" источника "In". Затем запишите считанное значение в положение записи "Pos" объекта записи "InOut".
- Имя команды зависит от типа данных "In" и "InOut". Например, если типом данных параметров "In" и "InOut" является BYTE, команда будет иметь имя WriteNbit_BYTE.

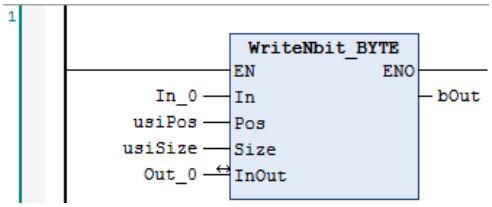
(4) Примечание

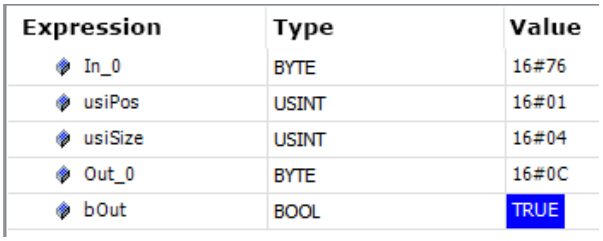
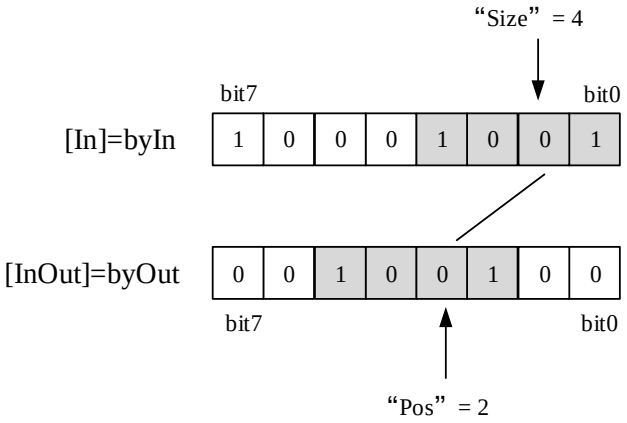
- Задайте одинаковые типы данных для "In" и "InOut".
- Если значение "Size" равно 0, значение "InOut" остается неизменным, а функция возвращает значение True.
- Если эта команда используется в программе ST, возвращаемое значение "Out" не будет использовано.
- Если значения "Size" и "Pos" превышают допустимый диапазон, программа сталкивается с исключением, и функция возвращает значение FALSE, а значение "InOut" остается неизменным.
- Если количество битов в "InOut" недостаточно от положения, указанного в "Pos", до положения, указанного в "Size", программа сталкивается с исключением, и функция возвращает значение FALSE, а значение "InOut" остается неизменным.

(5) Пример программы

ДЕМО	LD	ST

Задание переменных	VAR
	In_0 :BYTE:=16#76;
	usiPos :USINT:=1;
	usiSize:USINT:=4;
	Out_0 :BYTE;
	bOut :BOOL;
	END_VAR	

Пример программы		1 WriteNbit_BYTE(In:= In_0, 2 Pos:= usiPos, 3 Size:= usiSize, 4 InOut:= Out_0);

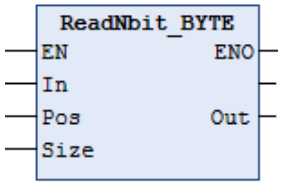
Результат операции	
Анализ принципа	

1-4-6-2. Группа считывания N битов ReadNbit_XXX

(1) Обзор команды

Считывание нескольких битов из битовой последовательности.

Группа считывания N битов [ReadNbit_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ReadNbit_XXX	Группа считывания N битов		ReadNbit_BYTE(In:= , Pos:= , Size:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Считывание источника	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Эта битовая последовательность позволяет считывать значение, записываемое в "InOut".
Pos	Положение считывания	USINT	В качестве типа данных	0	Считывание положения бита
Size	Размер считывания	USINT	В качестве типа данных	0	Количество считываемых битов
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат считывания	То же, что для типа "In"	В качестве типа данных	-	Результат считывания

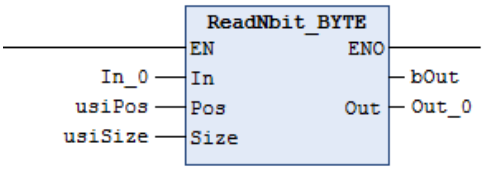
(3) Функция

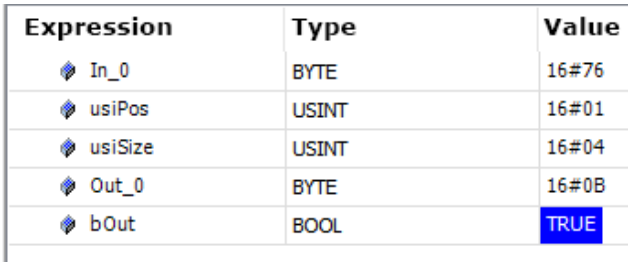
- Подстановка битов "Size", начиная с положения считывания "Pos" столбца исходных битов "In", в результат считывания "Out".
- Имя команды зависит от типа данных "In" и "Out". Например, если типом данных параметров "In" и "Out" является BYTE, команда будет иметь имя ReadNbit_BYTE.

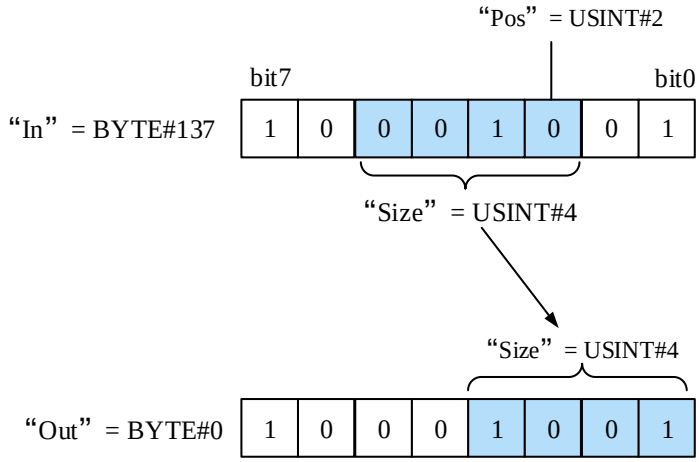
(4) Примечание

- Рекомендуется унифицировать типы данных для "In" и "Out".
- Если значение "Size" равно 0, то значением "Out" будет 16#0.
- В указанных ниже случаях возникает исключение, при этом функция возвращает значение FALSE, а значение "Out" равно 16#0.
  - **Когда значение "Pos" превышает допустимый диапазон.**
  - **Когда значение "Size" превышает допустимый диапазон.**
  - **Когда заданные значения "Pos" и "Size" превышают количество цифр в "In".**

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">
<pre>
VAR
    In_0   :BYTE:=16#76;
    usiPos :USINT:=1;
    usiSize:USINT:=4;
    Out_0 :BYTE;
    bOut   :BOOL;
END_VAR
</pre>
</td></tr>
<tr><td>Пример программы</td><td>


</td><td>
<pre>
1 ReadNbit_BYTE(In:= In_0,
2   Pos:= usiPos,
3   Size:= usiSize,
4   Out=> Out_0);
</pre>
</td></tr>
<tr><td>Результат операции</td><td colspan="2">


</td></tr>
<tr><td>Анализ принципа</td><td colspan="2">


</td></tr>
</table>

1-4-7. Команды преобразования данных

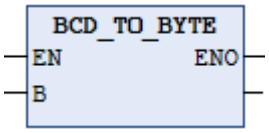
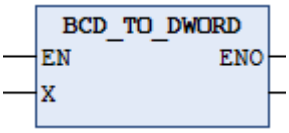
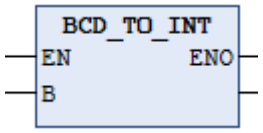
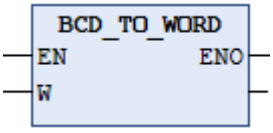
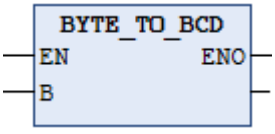
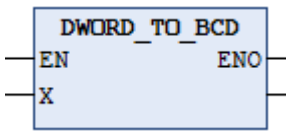
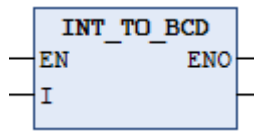
※	Категорически запрещено неявно преобразовывать «большие» данные в «меньшие» для последующего использования, так как существует риск потери информации при переходе от больших данных к меньшим.
※	Если преобразованное значение превышает диапазон хранения целевых данных, старший байт этого числа будет проигнорирован. Пример: Преобразование типа INT в тип BYTE или типа DINT в тип WORD.

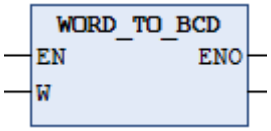
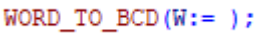
1-4-7-1. Преобразование кода BCD и целочисленных данных

(1) Обзор команды

Код BCD преобразуется в данные BYTE, DWORD, INT и WORD.

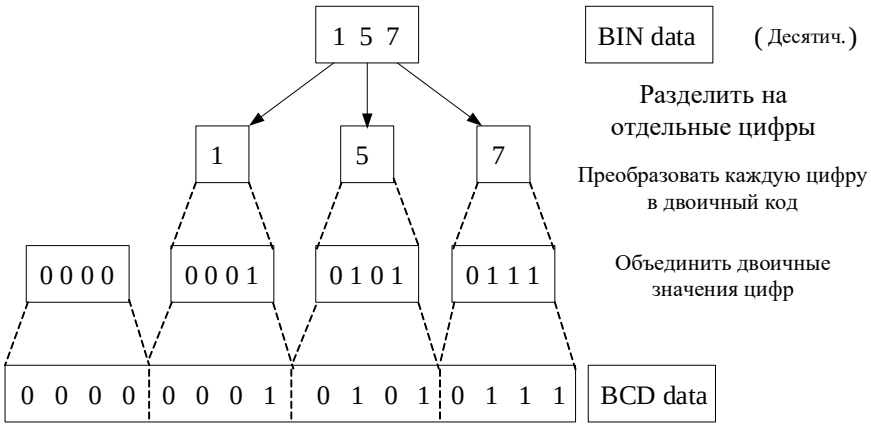
Преобразование кода BCD и целочисленных данных [BCD_TO_XXX/XXX_TO_BCD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	Util		

Команда	Название	Графическое представление	ST
BCD_TO_BYTE	Преобразование BCD в BYTE		`BCD_TO_BYTE(B:= );`
BCD_TO_DWORD	Преобразование BCD в DWORD		`BCD_TO_DWORD(X:= );`
BCD_TO_INT	Преобразование BCD в INT		`BCD_TO_INT(B:= );`
BCD_TO_WORD	Преобразование BCD в WORD		`BCD_TO_WORD(W:= );`
BYTE_TO_BCD	Преобразование BYTE в BCD		`BYTE_TO_BCD(B:= );`
DWORD_TO_BCD	Преобразование DWORD в BCD		`DWORD_TO_BCD(X:= );`
INT_TO_BCD	Преобразование INT в BCD		`INT_TO_BCD(I:= );`

Команда	Название	Графическое представление	ST
WORD_TO_BCD	Преобразование WORD в BCD		

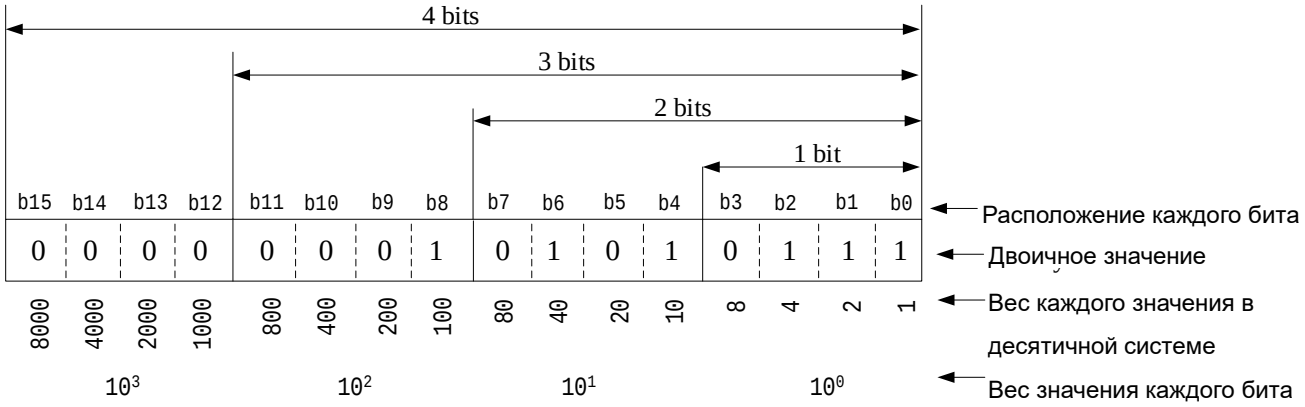
(2) Функция

BCD (двоично-кодированное десятичное число) означает использование 4-битных двоичных чисел для параллельного представления значений каждой цифры десятичного числа. Например, в данных BIN данные BCD 0000 0001 0101 0111 (343) используются для представления десятичного числа "157" указанным ниже образом.



Пример BCD

Когда данные BCD хранятся в 16-битной памяти, они могут обрабатывать значения в диапазоне от 0 до 9999 (максимальное значение - 4 бита). Вес каждого бита показан на рисунке ниже:



Вес каждого значения в десятичной системе, представленный с помощью BCD

(3) Примечание

Типы данных в команде преобразования кода BCD и целочисленных данных

Команда	Тип входных данных	Тип возвращаемого значения
BCD_TO_BYTE	BYTE	BYTE
BCD_TO_DWORD	DWORD	DWORD
BCD_TO_INT	BYTE	INT
BCD_TO_WORD	WORD	WORD
BYTE_TO_BCD	BYTE	BYTE
DWORD_TO_BCD	DWORD	DWORD
INT_TO_BCD	INT	BYTE
WORD_TO_BCD	WORD	WORD

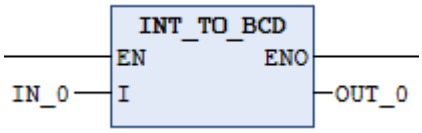
(4) Пример программы

1. Преобразование целочисленных данных 73 в код BCD

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN_0 :INT:=73;
    OUT_0:BYTE;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



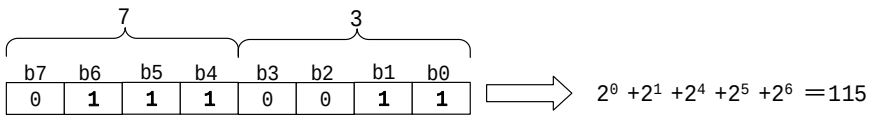
</td><td>

```
1  OUT_0:=INT_TO_BCD(I:= IN_0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

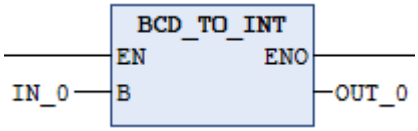
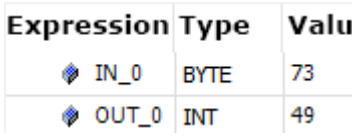
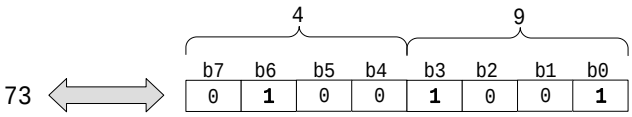
Expression	Type	Valu
IN_0	INT	73
OUT_0	BYTE	115

</td></tr>
<tr><td>Анализ принципа</td><td colspan="2">



</td></tr>
</table>

2. Преобразование кода BCD 73 в целочисленные данные

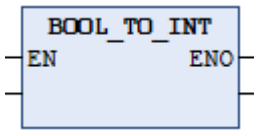
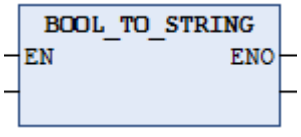
ДЕМО	LD	ST
Задание переменных	VAR IN_0 :BYTE:=73; OUT_0:INT; END_VAR	
Пример программы		1 OUT_0:=BCD_TO_INT(B:= IN_0);
Результат операции		
Анализ принципа		

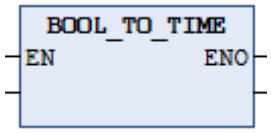
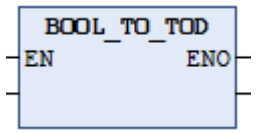
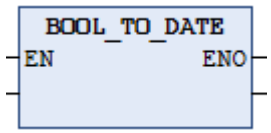
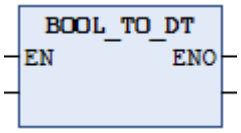
1-4-7-2. Данные преобразования булева типа BOOL_TO_<TYPE>

(1) Обзор команды

Преобразование булевых данных в данные других типов.

Данные преобразования булева типа [BOOL_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

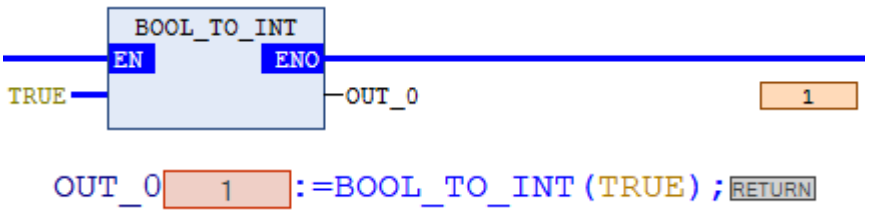
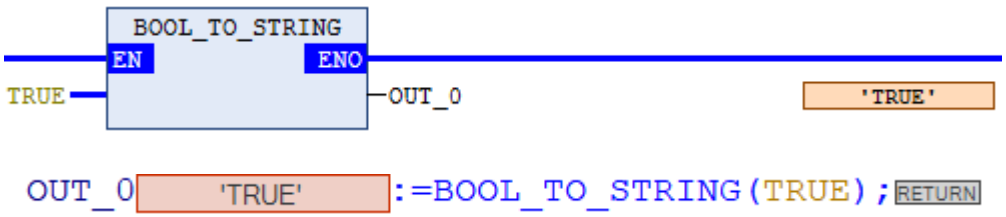
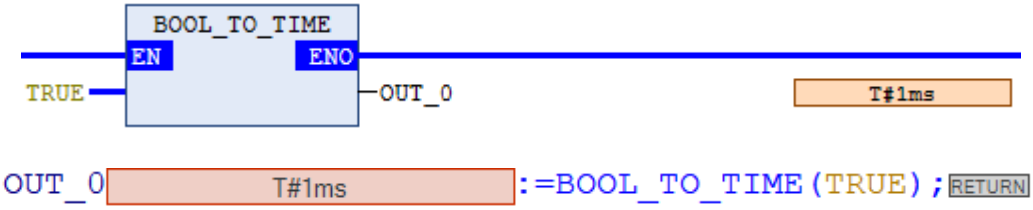
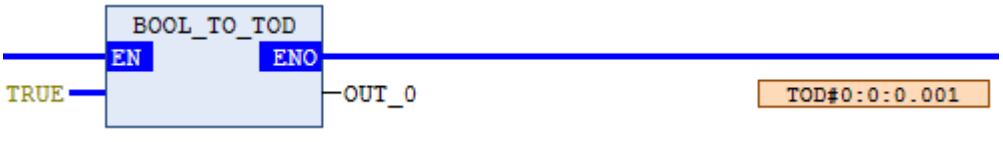
Команда	Название	Графическое представление	ST
BOOL_TO_INT	Преобразование BOOL в INT		BOOL_TO_INT
BOOL_TO_STRING	Преобразование BOOL в STRING		BOOL_TO_STRING

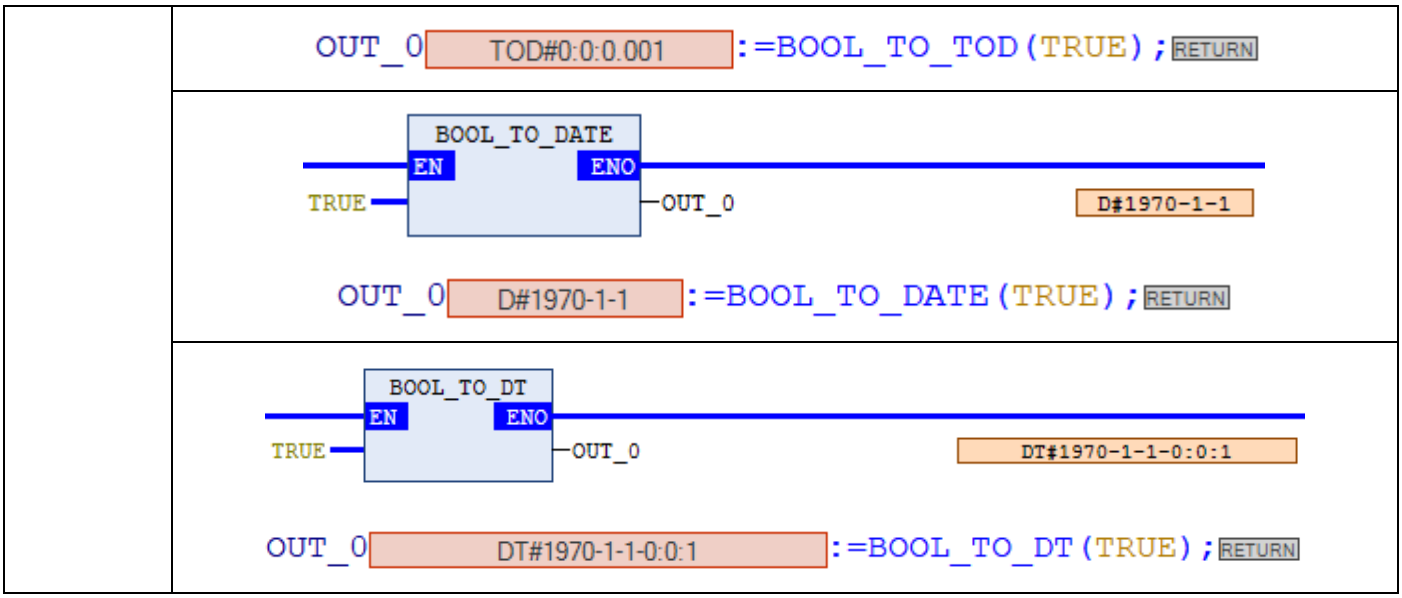
BOOL_TO_TIME	Преобразование BOOL в TIME		BOOL_TO_TIME
BOOL_TO_TOD	Преобразование BOOL в TOD		BOOL_TO_TOD
BOOL_TO_DATE	Преобразование BOOL в DATE		BOOL_TO_DATE
BOOL_TO_DT	Преобразование BOOL в DT		BOOL_TO_DT

(2) Функция

Поддерживаемые типы данных: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.

(3) Пример программы

ДЕМО	LD/ST
Пример программы	   

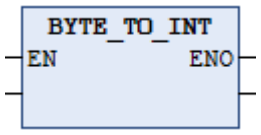
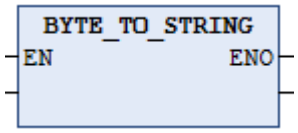
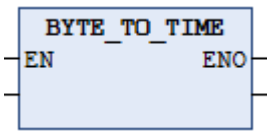


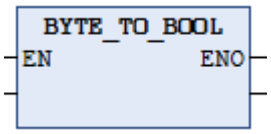
1-4-7-3. Данные преобразования байтового типа BYTE_TO_<TYPE>

(1) Обзор команды

Преобразование байтовых данных в данные других типов.

Данные преобразования байтового типа [BYTE_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
BYTE_TO_INT	Преобразование BYTE в INT		BYTE_TO_INT
BYTE_TO_STRING	Преобразование BYTE в STRING		BYTE_TO_STRING
BYTE_TO_TIME	Преобразование BYTE в TIME		BYTE_TO_TIME

BYTE_TO_BOOL	Преобразование BYTE в BOOL		BYTE_TO_BOOL
.....			

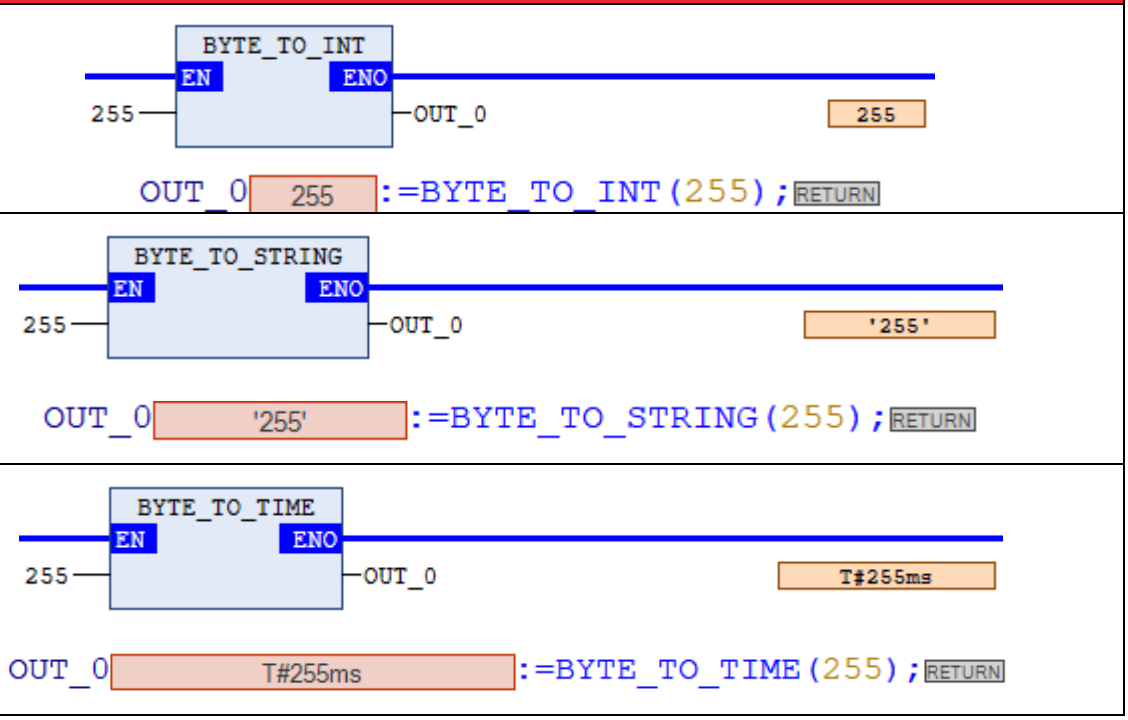
(2) Функция

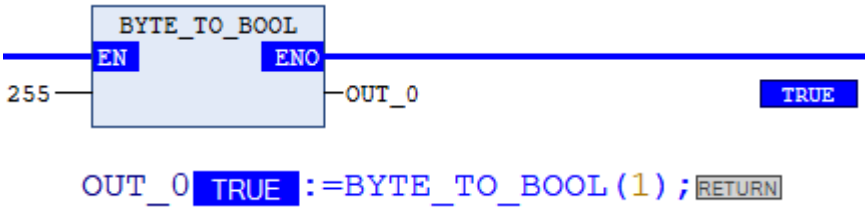
- Поддерживаемые типы данных: BOOL, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.
- Если выходные данные относятся к типу BOOL: Если значение на входе не равно 0, выход примет значение True. Если входные данные равны 0, выход примет значение FALSE.
- Если выходные данные относятся к типу TIME или TOD, входные данные будут преобразованы в миллисекунды.
- Если выходные данные относятся к типу DATE или DT: Входные данные будут преобразованы в секунды.

(3) Примечание

- Другие команды преобразования байтовых данных описаны в интерактивной справке по ПО.

(4) Пример программы

ДЕМО	LD/ST
Пример программы	

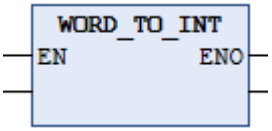
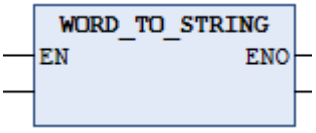
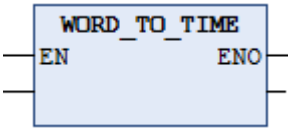
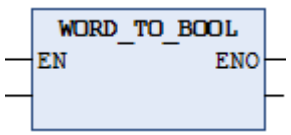
ДЕМО	LD/ST
	

1-4-7-4. Команда преобразования целочисленных данных < Integer data > _TO_<TYPE>

(1) Обзор команды

Преобразование целочисленных данных в данные других типов.

Команда преобразования целочисленных данных [< Integer data >_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
WORD_TO_INT	Преобразование WORD в INT		WORD_TO_INT
WORD_TO_STRING	Преобразование WORD в STRING		WORD_TO_STRING
WORD_TO_TIME	Преобразование WORD в TIME		WORD_TO_TIME
WORD_TO_BOOL	Преобразование WORD в BOOL		WORD_TO_BOOL
.....			

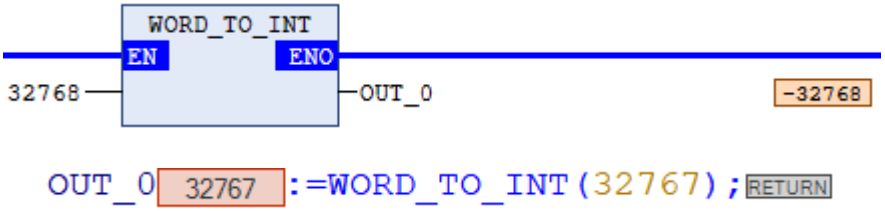
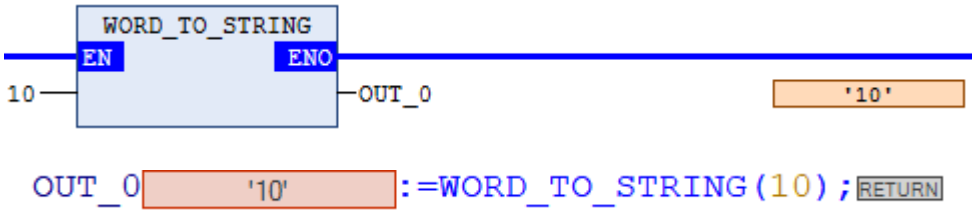
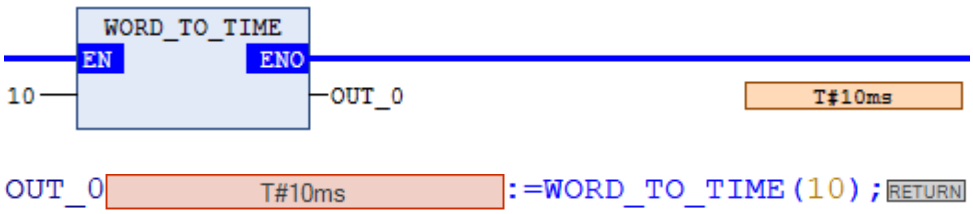
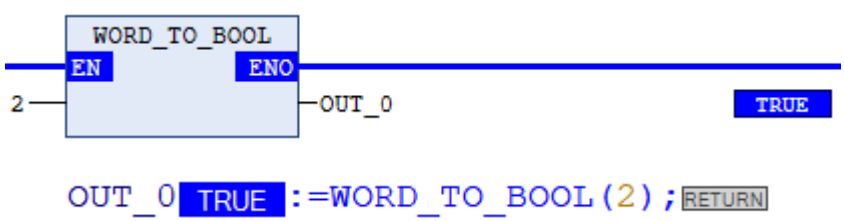
(2) Функция

- Поддерживаемые типы данных: BOOL, BYTE, SINT, WORD, DWORD, USINT, INT, UINT, force, UDINT, REAL, TIME, DATE, TOD, DT, а также строки.
- Если выходные данные относятся к типу BOOL: Если значение на входе не равно 0, выход примет значение TRUE; если значение на входе равно 0, выход примет значение FALSE.
- Если выходные данные относятся к типу TIME или TOD, входные данные будут преобразованы в миллисекунды.
- Если выходные данные относятся к типу DATE или DT, входные данные будут преобразованы в секунды.

(3) Примечание

Другие команды преобразования целочисленных данных описаны в интерактивной справке по ПО.

(4) Пример программы

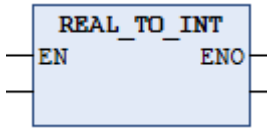
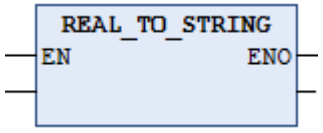
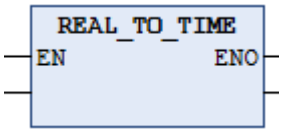
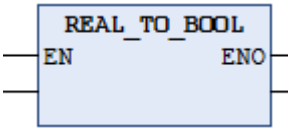
ДЕМО	LD/ST
Пример программы	
	
	
	

1-4-7-5. Команда преобразования действительных чисел REAL_TO_<TYPE>

(1) Обзор команды

Преобразование чисел с плавающей точкой в другие типы данных. При преобразовании числа с плавающей точкой в другой тип данных округлите значение до целого значения (за исключением преобразования в STRING), а затем преобразуйте его в новый тип данных.

Команда преобразования действительных чисел [REAL_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
REAL_TO_INT	Преобразование REAL в INT		REAL_TO_INT
REAL_TO_STRING	Преобразование REAL в STRING		REAL_TO_STRING
REAL_TO_TIME	Преобразование REAL в TIME		REAL_TO_TIME
REAL_TO_BOOL	Преобразование REAL в BOOL		REAL_TO_BOOL
.....			

(2) Функция

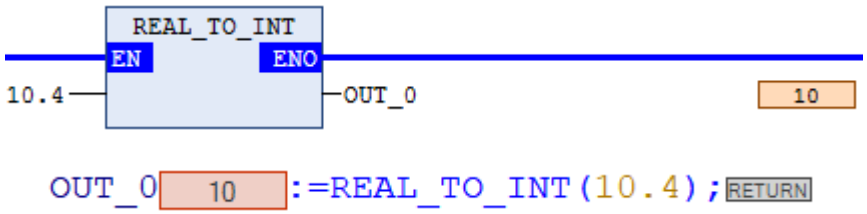
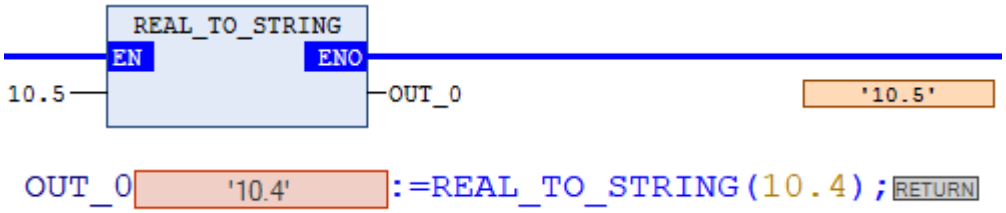
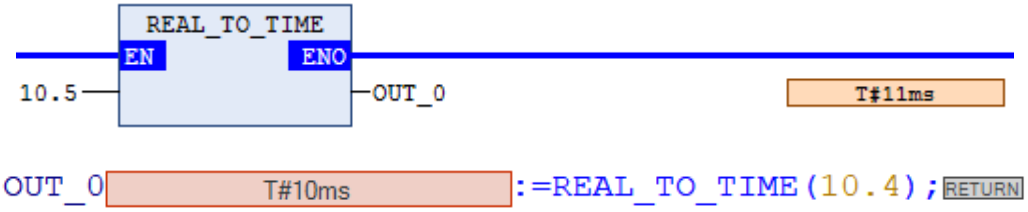
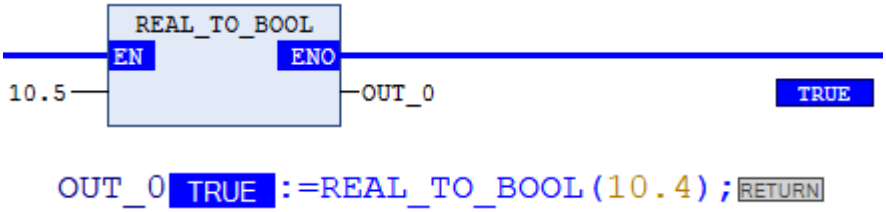
- Поддерживаемые типы данных: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT, а также STRING.

- Если выходные данные относятся к типу BOOL: Если значение на входе не равно 0, выход примет значение TRUE. Если входные данные равны 0, выход примет значение FALSE.
- Если выходные данные относятся к типу TIME или TOD, входные данные будут преобразованы в миллисекунды.
- Если выходные данные относятся к типу DATE или DT, входные данные будут преобразованы в секунды.

(3) Примечание

Другие команды преобразования чисел с плавающей точкой описаны в интерактивной справке по ПО.

(4) Пример программы

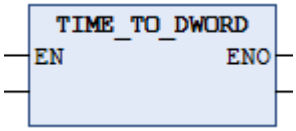
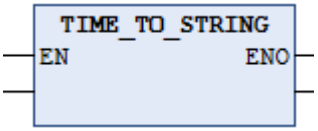
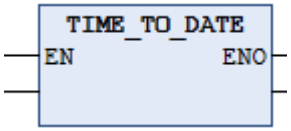
ДЕМО	LD/ST
Пример программы	
	
	
	

1-4-7-6. Команда преобразования временных данных TIME_TO_<TYPE>

(1) Обзор команды

Преобразование временных данных в другие типы данных, при этом время хранится внутри как тип данных DWORD в миллисекундах (начиная с 00:00 для переменной TIME_OF_DAY).

Команда преобразования временных данных [TIME_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
TIME_TO_DWORD	Преобразование TIME в DWORD		TIME_TO_DWORD
TIME_TO_STRING	Преобразование TIME в STRING		TIME_TO_STRING
TIME_TO_DATE	Преобразование TIME в DATE		TIME_TO_DATE
.....			

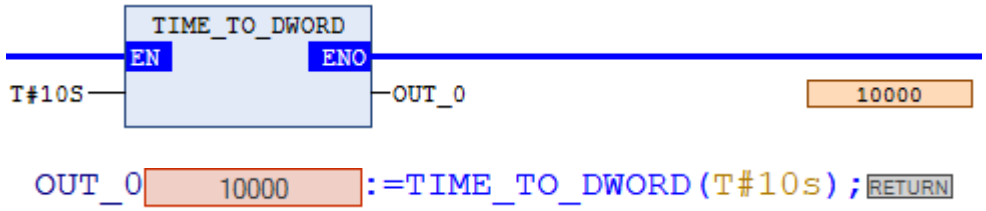
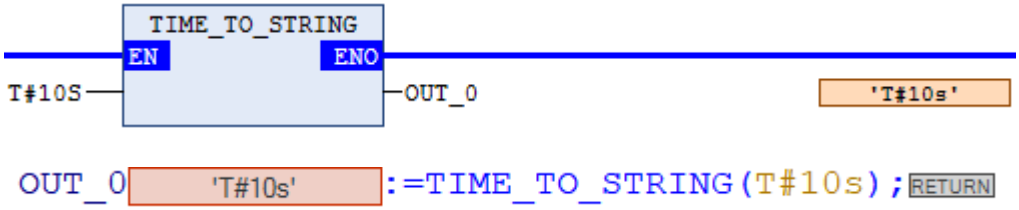
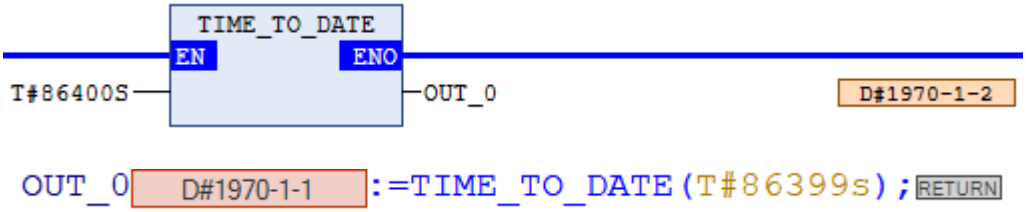
(2) Функция

- Поддерживаемые типы данных: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.

(3) Примечание

- Другие команды преобразования временных данных описаны в интерактивной справке по ПО.

(4) Пример программы

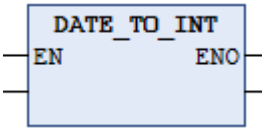
ДЕМО	LD/ST
Пример программы	
	
	

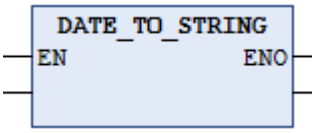
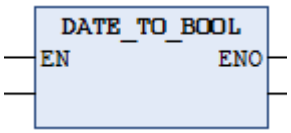
1-4-7-7. Команда преобразования данных о датах DATE_TO_<TYPE>

(1) Обзор команды

Преобразование данных о датах в другие типы данных, при этом даты хранятся внутри в секундах, начиная с 1 января 1970 г.

Команда преобразования данных о датах [DATE_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
DATE_TO_INT	Преобразование DATE в INT		DATE_TO_INT

DATE_TO_STRING	Преобразование DATE в STRING		DATE_TO_STRING
DATE_TO_BOOL	Преобразование DATE в BOOL		DATE_TO_BOOL
.....			

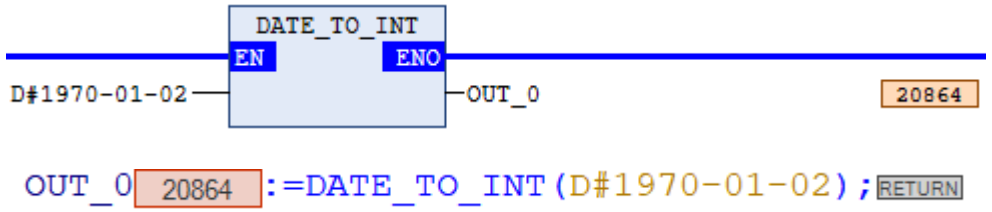
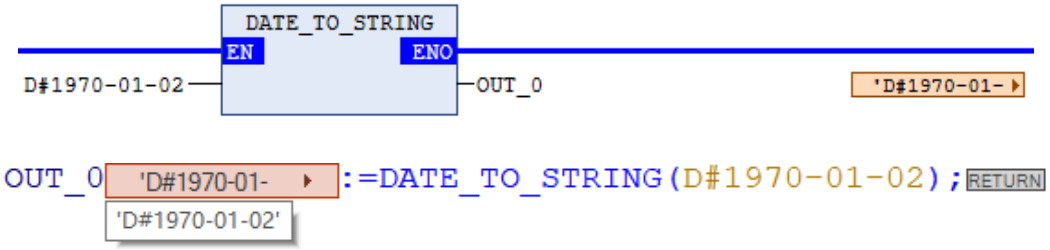
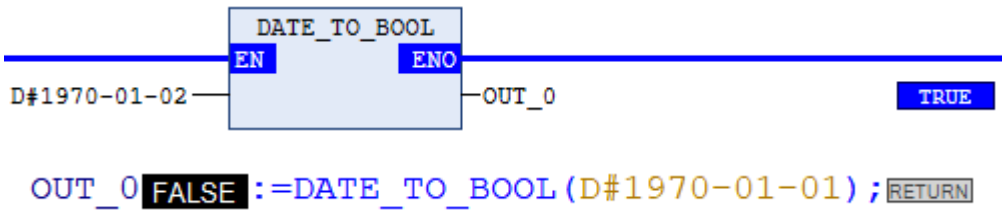
(2) Функция

- Поддерживаемые типы данных: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.

(3) Примечание

- Другие команды преобразования данных о датах описаны в интерактивной справке по ПО.

(4) Пример программы

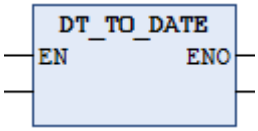
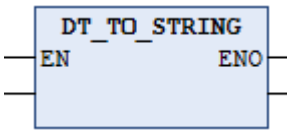
ДЕМО	LD/ST
Пример программы	  

1-4-7-8. Команда преобразования данных о датах и времени DT_TO_<TYPE>

(1) Обзор команды

Преобразование данных о датах и времени в другие типы данных, при этом даты хранятся внутри в секундах, начиная с 1 января 1970 г.

Команда преобразования данных о датах и времени [DT_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
DT_TO_DATE	Преобразование DT в DATE		DT_TO_DATE
DT_TO_STRING	Преобразование DT в STRING		DT_TO_STRING
.....			

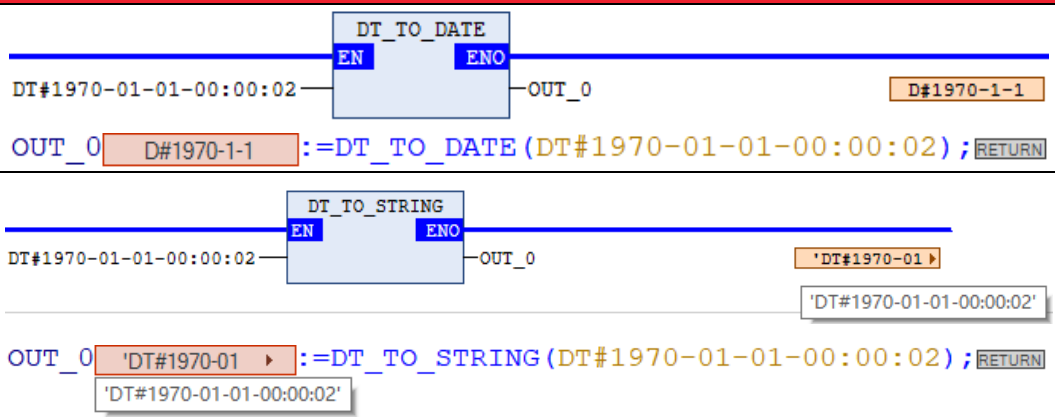
(2) Функция

- Поддерживаемые типы данных: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.

(3) Примечание

- Другие команды преобразования данных о датах и времени описаны в интерактивной справке по ПО.

(4) Пример программы

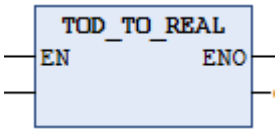
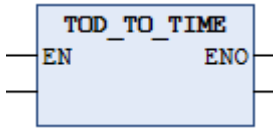
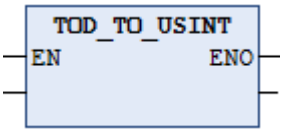
ДЕМО	LD/ST
Пример программы	

1-4-7-9. Команда преобразования временных данных TOD_TO_<TYPE>

(1) Обзор команды

Преобразование временных данных в другие типы данных, при этом даты преобразуются внутри в миллисекунды.

Команда преобразования временных данных [TOD_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
TOD_TO_REAL	Преобразование TOD в REAL		TOD_TO_REAL
TOD_TO_TIME	Преобразование TOD в TIME		TOD_TO_TIME
TOD_TO_USINT	Преобразование TOD в USINT		TOD_TO_USINT

.....

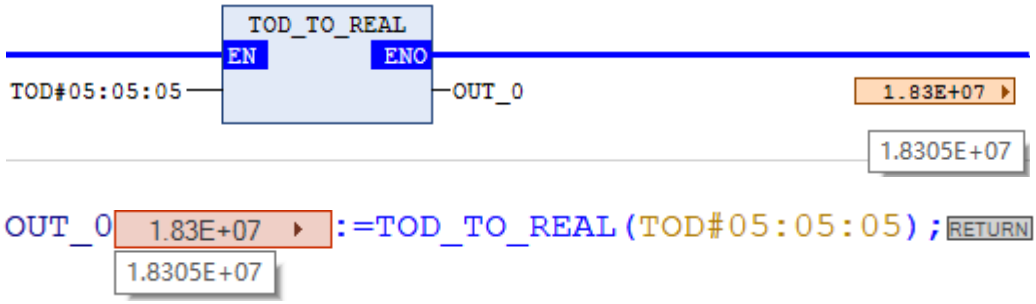
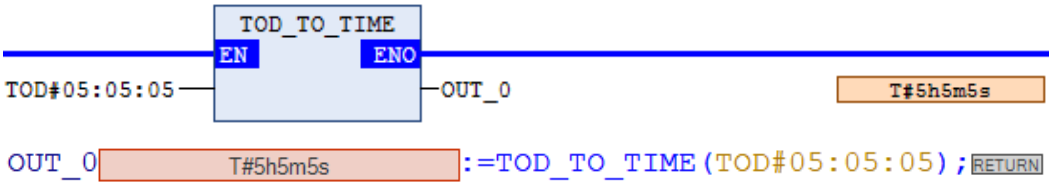
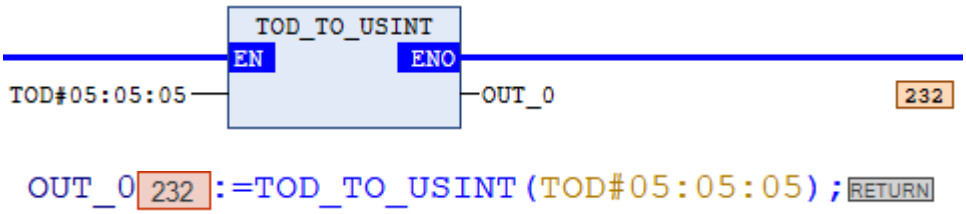
(2) Функция

- Поддерживаемые типы данных: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.

(3) Примечание

Другие команды преобразования временных данных описаны в интерактивной справке по ПО.

(4) Пример программы

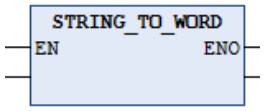
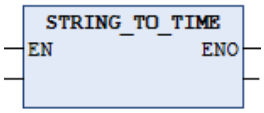
ДЕМО	LD/ST
Пример программы	
	
	

1-4-7-10. Команда преобразования символьных данных STRING_TO_<TYPE>

(1) Обзор команды

Чтобы преобразовать строку в другой тип данных, строковая переменная должна содержать допустимое значение целевой переменной, в противном случае результатом преобразования будет 0.

Команда преобразования символьных данных [STRING_TO_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
STRING_TO_WORD	Преобразование STRING в WORD		STRING_TO_WORD
STRING_TO_TIME	Преобразование STRING в TIME		STRING_TO_TIME
.....			

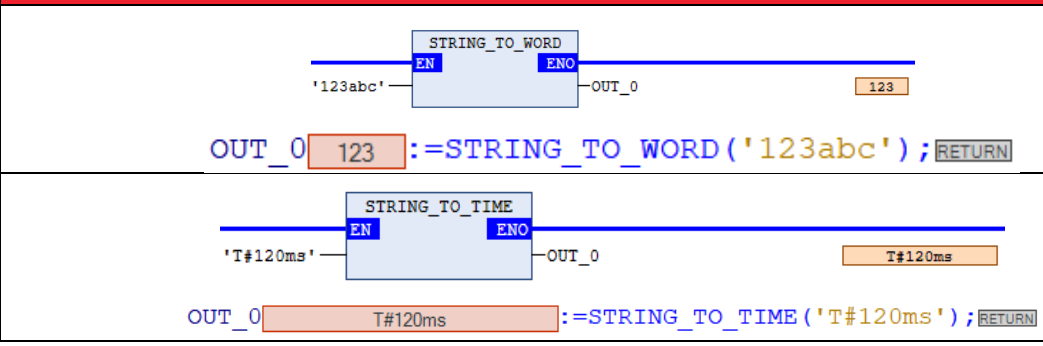
(2) Функция

- Поддерживаемые типы данных: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, TIME, DATE, TOD, DT и STRING.

(3) Примечание

- Другие команды преобразования строк описаны в интерактивной справке по ПО.

(4) Пример программы

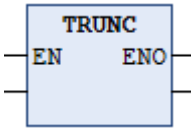
ДЕМО	LD/ST
Пример программы	

1-4-7-11. Округление TRUNC

(1) Обзор команды

Отбрасывание десятичной части данных и сохранение только целой части.

Округление [TRUNC]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
TRUNC	Округление		TRUNC

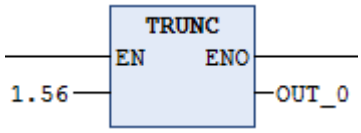
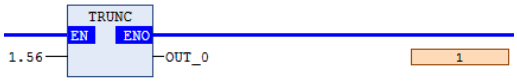
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входные данные	REAL	В качестве типа данных	0	Входные данные
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Данные после округления	DINT	В качестве типа данных	0	Данные после округления

(3) Примечание

- Другие команды преобразования строк описаны в интерактивной справке по ПО.
- При переходе от большего типа данных к меньшему типу данных может произойти потеря информации.
- Эта команда отсекает только целое число. При необходимости его округления используйте команду REAL_TO_INT.

(4) Пример программы

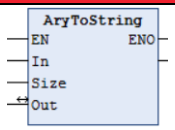
ДЕМО	LD	ST
Задание переменных	VAR OUT_0:DINT; END_VAR	
Пример программы		1 OUT_0:=TRUNC(-3.56);
Результат операции		1 OUT_0 -3 :=TRUNC(-3.56);

1-4-7-12. Преобразование массива в строку AryToString

(1) Обзор команды

Преобразование массива BYTE в строку.

Преобразование массива в строку [AryToString]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryToString	Преобразование массива в строку		XDONE:=AryToString(In:= , Size:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразованного массива	BYTE	В качестве типа данных	-	Массив BYTE, максимальное количество элементов - 1985.

Size	Размер преобразования	UINT	В качестве типа данных	1	Количество элементов массива In[], подлежащего преобразованию

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Out	Результат преобразования	STRING	1985 байта	-	Строка

(3) Функция

- Значения элементов после массива BYTE "In[X]" (включительно) обрабатываются как кодировка символов и сохраняются в строке "Out". Добавьте символ NULL в конце строки "Out".
- Задайте количество элементов "In[]", преобразуемых в "Size".
- Если между In[X] и In[X+“Size-1”] есть символы NULL, то в строке "Out" сохраняются только данные перед символом NULL.

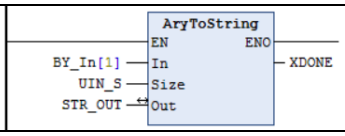
(4) Примечание

- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в In[].
- In[] должен быть массивом BYTE; в противном случае, функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Если значение "Size" равно 0, то "Out" — это строка, содержащая только нули, а функция возвращает значение True.
- Если значение Size превышает 1985 байтов, происходит усечение, а функция возвращает значение True.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		

```
VAR
   BY_In: ARRAY[0..5] OF BYTE:= [51,56,57,68,69,70];
   UIN_S: UINT:= 4;
   STR_OUT: STRING(1985);
   XDONE: BOOL;
END_VAR
```

Пример программы			XDONE:=AryToString(In:= BY_In[1], Size:= UIN_S, Out:= STR_OUT);
Результат операции			

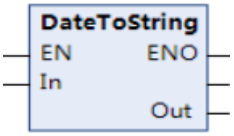
Expression	Type	Value
BY_In	ARRAY [0..5] OF BYTE	
BY_In[0]	BYTE	16#33
BY_In[1]	BYTE	16#38
BY_In[2]	BYTE	16#39
BY_In[3]	BYTE	16#44
BY_In[4]	BYTE	16#45
BY_In[5]	BYTE	16#46
UIN_S	UINT	16#0004
STR_OUT	STRING(1985)	'89DE'
XDONE	BOOL	TRUE

1-4-7-13. Преобразование данных о датах в строку DateToString

(1) Обзор команды

Преобразование данных о датах в строку.

Преобразование данных о датах в строку [DateToString]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
DateToString	Преобразование данных о датах в строку		xDone:=DateToString(In:= , Out=>);

(2) Связанная переменная

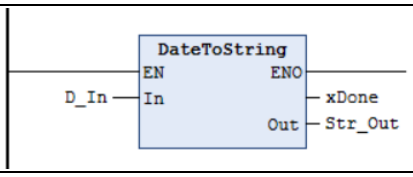
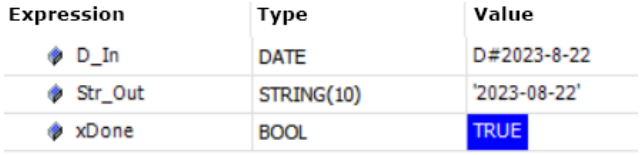
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразования	DATE	В качестве типа данных	D#1970-1-1	Дата

VAR_OUTPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Out	Результат преобразования	STRING[10]	10 байтов	-	Строка с результатами преобразования

(3) Функция

- Преобразование "In" с датой в строку.
- Добавьте символ NULL в конце строки "Out".

(4) Пример программы

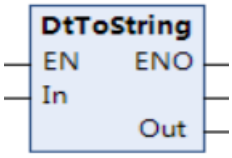
ДЕМО	LD	ST
Задание переменных	VAR D_In: DATE := D#2023-8-22; Str_Out: STRING(10); xDone: BOOL; END_VAR	
Пример программы		xDone := DateToString(In:= D_In, Out=> Str_Out);
Результат операции		

1-4-7-14. Преобразование данных о датах и времени в строку DtToString

(1) Обзор команды

Преобразование данных о датах и времени в строку.

Преобразование данных о датах и времени в строку [DtToString]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
DtToString	Преобразование данных о датах и времени в строку		xDone:=DtToString(In:= , Out=>);

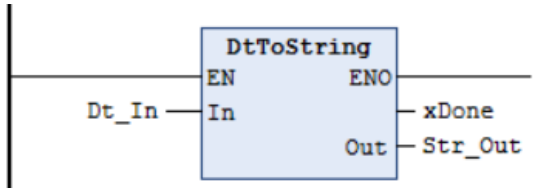
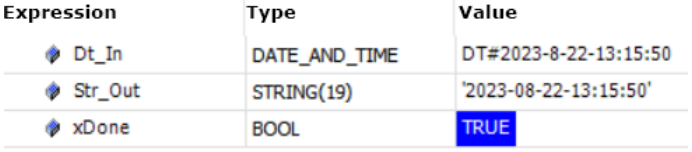
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразования	DT	В качестве типа данных	D#1970-1-1-0:0:0	Дата и время
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	STRING[19]	19 байтов	-	строка

(3) Функция

- Преобразование "In" с датой и временем в строку.
- Добавьте символ NULL в конце строки "Out".

(4) Пример программы

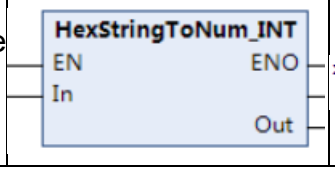
ДЕМО	LD	ST
Задание переменных	VAR Dt_In: DATE_AND_TIME:= DT#2023-8-22-13:15:50; Str_Out: STRING(19); xDone: BOOL; END_VAR	
Пример программы		xDone := DtToString(In:= Dt_In, Out=> Str_Out);
Результат операции		

1-4-7-15. Преобразование шестнадцатеричных строк в целые числа HexStringToNum_XXX

(1) Обзор команды

Преобразование шестнадцатеричных строк в целые числа.

Преобразование шестнадцатеричных строк в целые числа [HexStringToNum_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
HexStringToNum_XXX	шестнадцатеричные строки →целое число		xDone:=HexStringToNum_INT(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразования	STRING	В качестве типа данных	-	шестнадцатеричные строки
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	USINT/UINT/UDINT/ULINT SINT/INT/DINT/LINT	В качестве типа данных	-	Целочисленный тип данных

(3) Функция

- Преобразование "In" в виде шестнадцатеричной строки в целое число. Не зависит от старшего полупробела (16#20) и '0' (16#30). Одновременное игнорирование подчеркивания в строке (16#5F).
- Имя команды зависит от типа данных выходной переменной "Out". Например,

если типом данных для "Out" является INT, команда будет иметь имя HexStringToNum_INT.

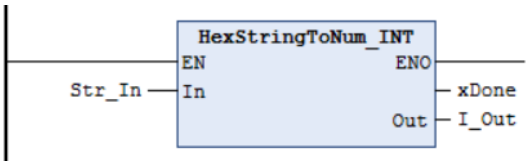
(4) Примечание

- Даже если результат преобразования выходит за пределы допустимого диапазона "Out", исключения не произойдет, а недопустимые значения будут сохранены в "Out".
- Если содержимое "In" не заканчивается символом NULL, при преобразовании могут произойти исключения.
- Если содержимое "In" представляет собой символ, который нельзя преобразовать в числовое значение, программа сталкивается с исключением, функция возвращает значение FALSE, а "Out" остается неизменным.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th colspan="2">LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="3">

```
VAR
    Str_In: STRING(1985);
    I_Out: INT;
    xDone: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td colspan="2">
</td><td>

```
xDone := HexStringToNum_INT(
    In:= Str_In,
    Out=> I_Out);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="3">

Expression	Type	Value
Str_In	STRING(1985)	'200'
I_Out	INT	512
xDone	BOOL	TRUE

Expression	Type	Value
Str_In	STRING(1985)	'-F'
I_Out	INT	-15
xDone	BOOL	TRUE

</td></tr>
</table>

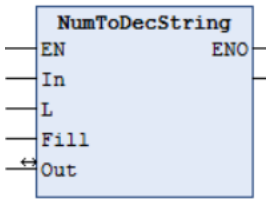
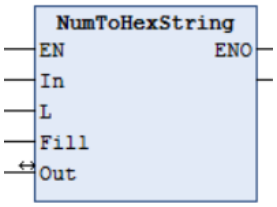
1-4-7-16. Преобразование целых чисел в десятичные/шестнадцатеричные строки фиксированной длины NumToDecString/NumToHexString

(1) Обзор команды

NumToDecString: Преобразование целого числа в десятичную строку фиксированной длины.

NumToHexString: Преобразование целого числа в шестнадцатеричную строку фиксированной длины.

Преобразование целых чисел в десятичные/шестнадцатеричные строки фиксированной длины [NumToDecString/NumToHexString]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
NumToDecString	Преобразование в десятичные строки фиксированной длины		xDone:=NumToDecString(In:= , L:= , Fill:= , Out:=);
NumToHexString	Преобразование в шестнадцатеричные строки фиксированной длины		xDone:=NumToHexString(In:= , L:= , Fill:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразования	USINT/UINT/UDINT/ ULINT SINT/INT/DINT/LINT	В качестве типа данных	-	Целое число
L	Символы	UINT	0-1985	1	Вывод символов
Fill	Добавление символов	_eFiLL_CHR	1 (_BLANK) 2 (_ZERO)	1 (_BLANK)	Добавление символов

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Out	Результат преобразования	STRING	В качестве типа данных	-	Строка

(3) Функция

- NumToDecString: Преобразование целого числа "In" в десятичную строку формата UTF-8, состоящую из английских цифр половинной ширины. Если "In" — отрицательное число, добавьте "-" (знак минус) в начале.
- NumToHexString: Преобразование целого числа "In" в шестнадцатеричную строку формата UTF-8, состоящую из английских цифр половинной ширины. Если "In" — отрицательное число, оно выражается в виде дополнения к 2 (берется обратное число и прибавляется 1).
- Приведенные выше команды устанавливают количество символов в строке "Out" равным "L". Если количество символов недостаточное, добавьте символ, обозначенный переменной "Fill", в верхнее положение.
- Если количество символов в результате преобразования больше, чем "L", в "Out" из самого нижнего положения результата преобразования берется только символ "L".
- Типом данных "Fill" является элемент перечисления _eFilL_CHR. Значение тела элемента перечисления представлено на рисунке ниже:

Элементы перечисления	Значение
1 (_BLANK)	‘’ (Пустой символ половинной ширины)
2 (_ZERO)	‘0’

(4) Примечание

- Если значение "L" равно 0, значение "Out" остается неизменным.
- Если количество символов в результате преобразования превышает "L", сохраните биты "L" в "Out" из положения младшего бита результата преобразования, как показано в примере ниже:

Команда	Значение “In”	Значение “L”	Значение “Out”
NumToDecString	128	2	28

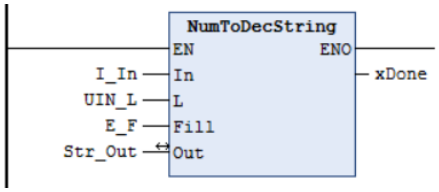
NumToHexString			80

- Если значение "L" превышает допустимый диапазон, программа сталкивается с исключением, и функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Если значение "Fill" превышает допустимый диапазон, при компиляции возникнет исключение.
- Если результат преобразования выходит за пределы диапазона "Out", выдается значение ошибки.

(5) Пример программы

NumToDecString	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    I_In: INT;
    UIN_L: UINT;
    E_F: _eFILL_CHR;
    Str_Out: STRING(1985);
    xDone: BOOL;
END_VAR
```




```
xDone:=NumToDecString(
    In:= I_In,
    L:= UIN_L,
    Fill:= E_F,
    Out:= Str_Out);
```

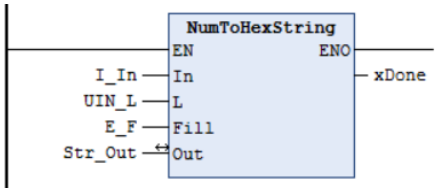
Expression	Type	Value
I_In	INT	512
UIN_L	UINT	8
E_F	_EFILL_CHR	_BLANK
Str_Out	STRING(1985)	' 512'
xDone	BOOL	TRUE

Expression	Type	Value
I_In	INT	-512
UIN_L	UINT	8
E_F	_EFILL_CHR	_ZERO
Str_Out	STRING(1985)	'-0000512'
xDone	BOOL	TRUE

<table>
<tr><th>NumToHexString</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    I_In: INT;
    UIN_L: UINT;
    E_F: _eFILL_CHR;
    Str_Out: STRING(1985);
    xDone: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
xDone:=NumToHexString(
    In:= I_In,
    L:= UIN_L,
    Fill:= E_F,
    Out:= Str_Out);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
I_In	INT	128
UIN_L	UINT	8
E_F	_EFILL_CHR	_BLANK
Str_Out	STRING(1985)	' 80'
xDone	BOOL	TRUE

Expression	Type	Value
I_In	INT	-128
UIN_L	UINT	8
E_F	_EFILL_CHR	_BLANK
Str_Out	STRING(1985)	'FFFFFF80'
xDone	BOOL	TRUE

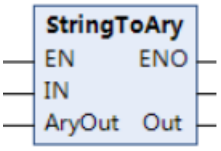
</td></tr>
</table>

1-4-7-17. Преобразование строки в массив StringToAry

(1) Обзор команды

Преобразование строки в массив BYTE.

Преобразование строки в массив [StringToAry]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
StringToAry	Преобразование строки в массив		XDONE:=StringToAry(In:= , AryOut:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразования	STRING	Не более 1985 символов	-	Строка
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
AryOut	Массив после преобразования	BYTE	В качестве типа данных	-	Массив типа BYTE
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Байты после преобразования	UINT	0-1985	-	Байты после преобразования

(3) Функция

- Рассмотрение кодировки символов строки "n" в качестве числового значения и сохранение каждого символа по одному в BYTE-массиве "AryOut[]". Сохранение указанного количества преобразованных байтов в "Out".

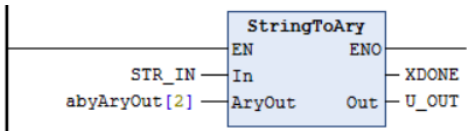
(4) Примечание

- Символ NULL в конце "In" не будет сохраняться в "AryOut[]".
- Если "In" представляет собой строку только с символами NULL, значение "Out" будет равно 0, а AryOut[] остается неизменным.
- Если количество байтов в "In" превышает количество элементов в "AryOut[]", обеспечьте отсутствие исключений ПЛК.
- Если количество байтов в "In" превышает допустимый диапазон (1985), символы за пределами допустимого диапазона будут усечены, а функция преобразует 1985 символов в нормальном виде.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    STR_IN: STRING(1985):= 'XYZ';
    abyAryOut: ARRAY[0..5] OF BYTE;
    U_OUT: UINT;
    XDONE: BOOL;
END_VAR
```

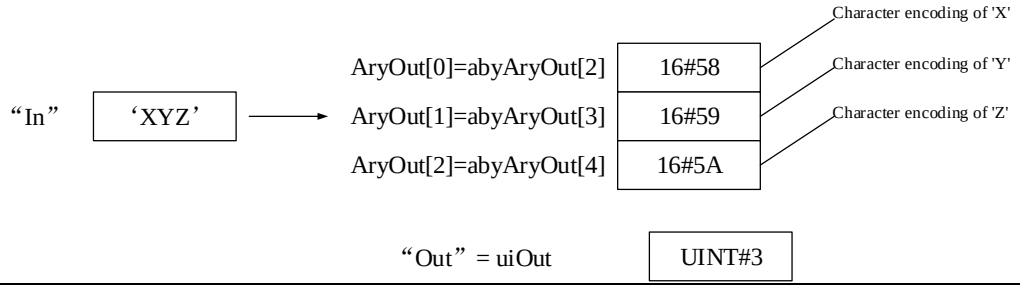
</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
XDONE:=StringToAry(In:= STR_IN,
    AryOut:= abyAryOut[2],
    Out=> U_OUT);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
STR_IN	STRING(1985)	'XYZ'
abyAryOut	ARRAY [0..5] OF BYTE	
abyAryOut[0]	BYTE	16#00
abyAryOut[1]	BYTE	16#00
abyAryOut[2]	BYTE	16#58
abyAryOut[3]	BYTE	16#59
abyAryOut[4]	BYTE	16#5A
abyAryOut[5]	BYTE	16#00
U_OUT	UINT	16#0003
XDONE	BOOL	TRUE

</td></tr>
<tr><td>Анализ принципа</td><td colspan="2">


</td></tr>
</table>

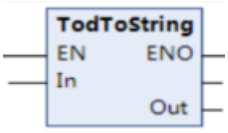
1-4-7-18. Преобразование времени в строку TodToString

(1) Обзор команды

Преобразование времени в строку.

Преобразование времени в строку [TodToString]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

Библиотека	PP_DataUtils	

Команда	Название	Графическое представление	ST
TodToString	Преобразование времени в строку		xDone:=TodToString(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник преобразования	TOD	В качестве типа данных	TOD#0:0:0	Часы, минуты, секунды
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	STRING[13]	13 байтов	-	Строка с результатами преобразования

(3) Функция

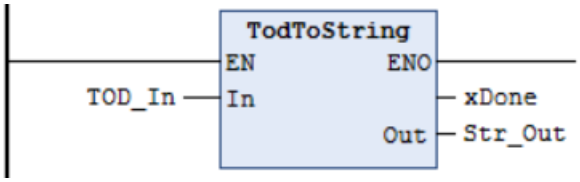
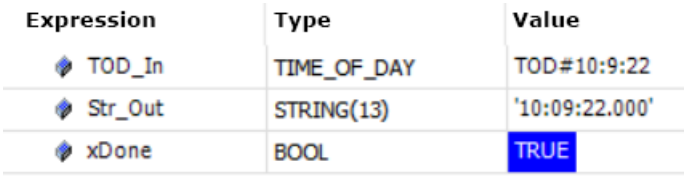
- Преобразование времени "In" в строку. Добавьте символ NULL в конце строки "Out".

(4) Примечание

- Параметр "Out" выражен в миллисекундах.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR TOD_In: TIME_OF_DAY; Str_Out: STRING(13); xDone: BOOL; END_VAR	

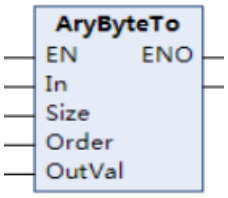
Пример программы		xDone:=TodToString(In:= TOD_In, Out=> Str_Out);
Результат операции		

1-4-7-19. Преобразование из массива байтов AryByteTo

(1) Обзор команды

Объединение элементов массива BYTE и их сохранение в переменные.

Преобразование из массива байтов [AryByteTo]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryByteTo	Преобразование из массива байтов		XDONE:=AryByteTo(In:= , Size:= , Order:= , OutVal:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Массив объектов преобразования	BYTE	В качестве типа данных	-	Массив объектов преобразования

Size	Размер элементов объектов преобразования	UINT	В качестве типа данных	-	Количество преобразуемых элементов In[]
Order	Порядок преобразования	_eBYTE_ORDER	В качестве типа данных	_LOW_HIGH	Преобразуемое положение
OutVal	Результат преобразования	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/REAL/LREAL/ TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Результат преобразования

(3) Функция

- Объедините начальные элементы "Size" массива объектов преобразования "In[X]" в соответствии с размером типа данных результата преобразования "OutVal", а затем сохраните их в "OutVal".
- При объединении элемента "In[]" задайте порядок через переменную порядка преобразования "Order". Тип данных "Order" — элемент перечисления _eBYTE_ORDER. Значения элементов перечисления:

Элемент перечисления	Значение
_LOW_HIGH	Сначала младший байт, затем старший байт
_HIGH_LOW	Сначала старший байт, затем младший байт

- Функции AryByteTo описаны в двух примерах ниже:

1. Тип данных "OutVal" размером 1 байт.

Если тип данных "OutVal" имеет размер 1 байт, сохраните "In[]" в "OutVal" в единицах по 1 байту.

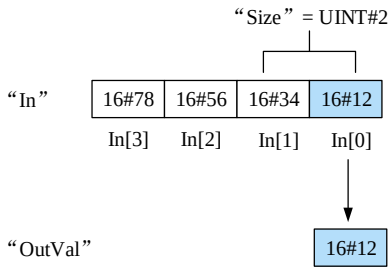
Существуют несколько типов данных размером 1 байт:

Тип	Тип данных
Битовая строка	BYTE
Целое число	USINT, SINT
Действительное число	Отсутствует
Время, продолжительность, дата, строка	Отсутствует

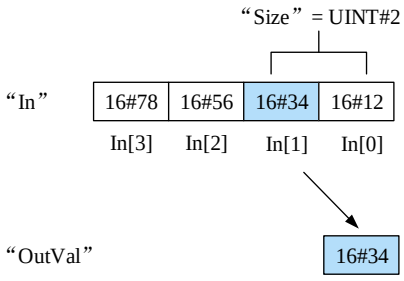
Метод сохранения:

Значение “Order”	Тип данных “OutVal”	Метод сохранения в "OutVal"
LOW_HIGH	BOOL	Сохранение значения нулевого бита In[0] в "OutVal"
	Не BOOL	Сохранение значения In[0] в "OutVal"
HIGH_LOW	BOOL	Сохранение значения нулевого бита In[1] в "OutVal"
	Не BOOL	Сохранение значения In[1] в "OutVal"

Пример, когда "OutVal" является переменной типа SINT с "Size"=UINT#2 и "Order"=_LOW_HIGH:

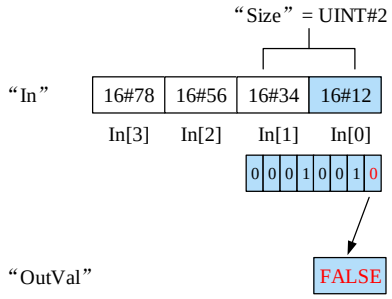


Пример, когда "OutVal" является переменной типа SINT с "Size"=UINT#2 и "Order"=_HIGH_LOW:

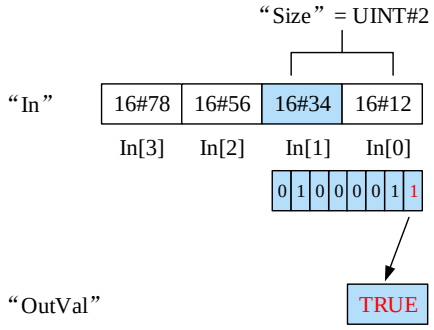


Пример, когда "OutVal" является переменной типа BOOL с "Size"=UINT#2 и

"Order"=_LOW_HIGH:



Пример, когда "OutVal" является переменной типа BOOL с "Size"=UINT#2 и "Order"=_HIGH_LOW:



2. Тип данных "OutVal" размером 2 байта или больше.

Если тип данных "OutVal" имеет размер 2 байта или больше, объедините "In[]" в единицы размера типа данных "OutVal" и сохраните его в "OutVal".

Существуют несколько типов данных размером 2 байта или больше:

Тип	Тип данных
Битовая строка	WORD, DWORD, LWORD
Целое число	UINT, UDINT, ULINT, INT, DINT, LINT
Действительное число	REAL, LREAL
Время, продолжительность, дата, строка	TIME, DATE, TOD, TD, STRING размером 2 байта или больше

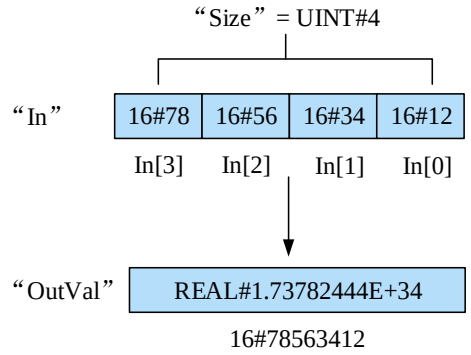
Метод сохранения:

- Создайте 1 слово (2 байта) данных на основе значения "Order", объединенного с "In[0]" и "In[1]". Если значение "Order" равно _LOW_HIGH, то старший байт будет равен "In[1]", а младший — "In[0]". Если значение "Order" равно _HIGH_LOW, то старший байт будет равен "In[0]", а младший байт —

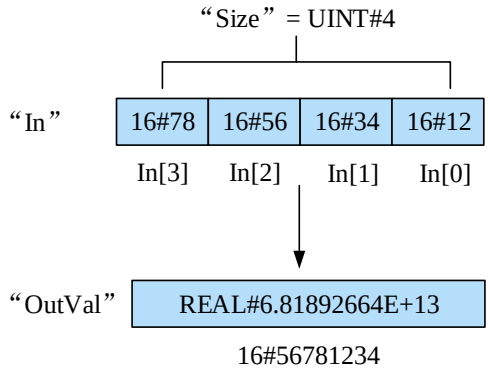
"In[1]". Используя тот же метод, объедините значения после "In[2]" и "In[3]", чтобы создать несколько данных, состоящих из одного слова.

- На основе размера типа данных "OutVal" объедините несколько данных, состоящих из одного слова. Например, если типом данных "OutVal" является DWORD, объедините четыре набора данных, состоящих из одного слова. Если "OutVal" — это весь массив, сохраните "In[i]" (где i — четное число) в "OutVal[i+1]". Сохраните "In[i]" (где i — нечетное число) в "OutVal[i-1]". Если значение "Size" нечетное, оно будет сохранено в "OutVal ["Size"]", а 16#00 будет сохранено в "OutVal[" Size "-1]". Сохраните полученные данные в "OutVal".

Пример, когда "OutVal" является переменной типа REAL с "Size"=UINT#4 и "Order"=_LOW_HIGH:



Пример, когда "OutVal" является переменной типа REAL с "Size"=UINT#4 и "Order"=_HIGH_LOW:



(4) Примечание

- Если значение "Size" превышает область массива в In[], нужно избежать исключений ПЛК.
- "OutVal" не может быть целым массивом типа BOOL, иначе результат выполнения программы будет неверным.
- Если значение "Size" равно 0, то значением "Out" будет True, а "OutVal"

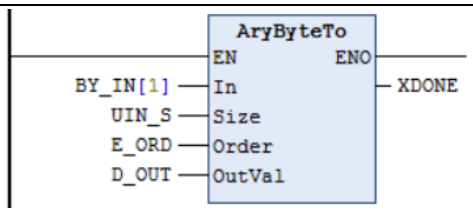
остается неизменным.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   BY_IN: ARRAY[0..5] OF BYTE:= [16#21,16#52,16#62,16#81,16#66,16#55];
   UIN_S: UINT:= 3;
   E_ORD: _eBYTE_ORDER:= _eBYTE_ORDER._LOW_HIGH;
   D_OUT: DWORD;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:=AryByteTo(
    In:= BY_IN[1],
    Size:= UIN_S,
    Order:= E_ORD,
    OutVal:= D_OUT);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
BY_IN	ARRAY [0..5] OF BYTE	
BY_IN[0]	BYTE	16#21
BY_IN[1]	BYTE	16#52
BY_IN[2]	BYTE	16#62
BY_IN[3]	BYTE	16#81
BY_IN[4]	BYTE	16#66
BY_IN[5]	BYTE	16#55
UIN_S	UINT	16#0003
E_ORD	_EBYTE_ORDER	_LOW_HIGH
D_OUT	DWORD	16#00816252
XDONE	BOOL	TRUE

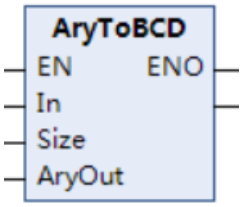
</td></tr>
</table>

1-4-7-20. Пакетное преобразование массивов в BCD AryToBCD

(1) Обзор команды

Преобразование элементов массива целых чисел без знака в BCD.

Пакетное преобразование массивов в BCD [AryToBCD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryToBCD	Пакетное преобразование массивов в BCD		AryToBCD(In:=, Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник массива преобразования	USINT/UINT/ UDINT/ULINT	В качестве типа данных	-	Массив целых чисел без знака
Size	Размер преобразования	UINT	В качестве типа данных	1	Количество преобразуемых элементов In[]
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
AryOut	Массив с результатами	Соответствует элементу In	В качестве типа данных	-	Массив BCD

(3) Функция

- Преобразование элементов "Size", начиная с In[0] массива целых чисел без знака In[], в BCD и вывод результатов преобразования в BCD-массив AryOut[].
- Эффективный диапазон In[] и AryOut[] зависит от комбинации типов данных элементов.

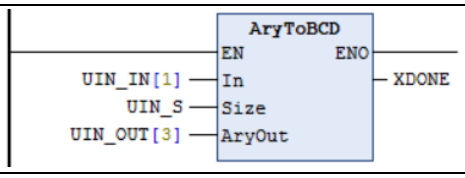
Тип данных элемента In[]	Эффективный диапазон In[]	Эффективный диапазон AryOut[]
USINT	0-99	16#00-16#99(BCD)
UINT	0-9999	16#0000-16#9999(BCD)
UDINT	0-99999999	16#0000_0000-16#9999_9999(BCD)
ULINT	0-9999999999999999	16#0000_0000_0000_0000-

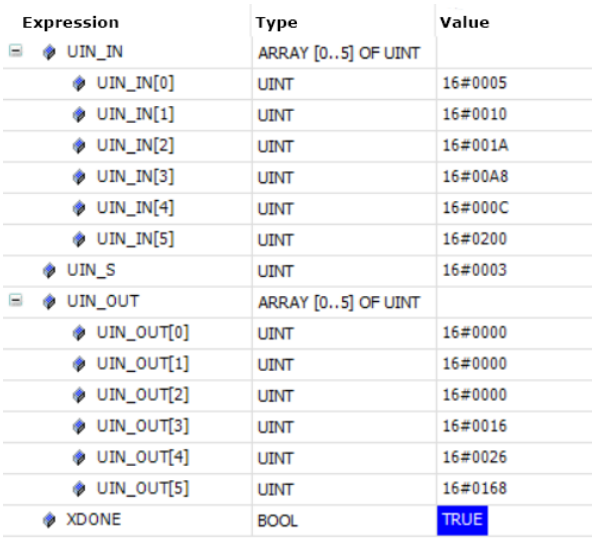
		16#9999_9999_9999_9999(BCD)

(4) Примечание

- Невозможно преобразовать BCD со знаком. Поэтому установите тип данных In[] на любое из целых чисел без знака (USINT, UINT, UDINT, ULINT).
- Если значение "Size" равно 0, то функция возвращает значение True, а значение AryOut[] остается неизменным.
- Если значение In[] превышает допустимый диапазон, функция возвращает значение FALSE, а значение AryOut[] остается неизменным.
- Если значение "Size" превышает область массива в In[] или AryOut[], нужно избежать исключений ПЛК.
- В "In[] array" необходимо использовать целочисленный тип данных. Если тип данных не относится к целому числу, выдается сообщение об ошибке при компиляции.
- Тип "In[] array" не является типом USINT/UINT/UDINT/ULINT, функция выполнения возвращает значение FALSE, а значение AryOut[] остается неизменным.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR UIN_IN: ARRAY[0..5] OF UINT:= [5,16,26,168,12,512]; UIN_S: UINT:= 3; UIN_OUT: ARRAY[0..5] OF UINT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:=AryToBCD(In:= UIN_IN[1], Size:= UIN_S, AryOut:= UIN_OUT[3]);

Результат операции	

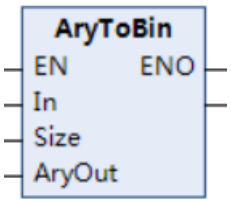
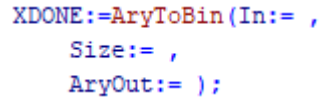
Expression	Type	Value
UIN_IN	ARRAY [0..5] OF UINT	
UIN_IN[0]	UINT	16#0005
UIN_IN[1]	UINT	16#0010
UIN_IN[2]	UINT	16#001A
UIN_IN[3]	UINT	16#00A8
UIN_IN[4]	UINT	16#000C
UIN_IN[5]	UINT	16#0200
UIN_S	UINT	16#0003
UIN_OUT	ARRAY [0..5] OF UINT	
UIN_OUT[0]	UINT	16#0000
UIN_OUT[1]	UINT	16#0000
UIN_OUT[2]	UINT	16#0000
UIN_OUT[3]	UINT	16#0016
UIN_OUT[4]	UINT	16#0026
UIN_OUT[5]	UINT	16#0168
XDONE	BOOL	TRUE

1-4-7-21. Пакетное преобразование массивов в битовые строки AryToBin

(1) Обзор команды

Преобразование элементов массива BCD в битовые строки.

Пакетное преобразование массивов в битовые строки [AryToBin]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryToBin	Пакетное преобразование массивов в битовые строки		


```
XDONE:=AryToBin(In:= ,
    Size:= ,
    AryOut:= );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник массива	BYTE/WORD/ DWORD/LWOR	В качестве типа данных	-	Массив BCD

	преобразования	D			
Size	Количество элементов преобразования	UINT	В качестве типа данных	1	Количество преобразуемых элементов In[]
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
AryOut	Массив с результатами	Соответствует элементу In	В качестве типа данных	-	Преобразованный массив битовых строк

(3) Функция

- Преобразование элементов "Size", начиная с In[0] массива BCD In[], в битовые строки и вывод массива битовых строк AryOut[].
- Эффективный диапазон In[] и AryOut[] зависит от комбинации типов данных элементов.

Тип данных элемента In[]	Эффективный диапазон In[]	Эффективный диапазон AryOut[]
BYTE	16#00-16#99(BCD)	0-99
WORD	16#0000-16#9999(BCD)	0-9999
DWORD	16#0000_0000-16#9999_9999(BCD)	0-99999999
LWORD	16#0000_0000_0000_0000-16#9999_9999_9999_9999(BCD)	0-9999999999999999

(4) Примечание

- Невозможно преобразовать BCD со знаком. Поэтому установите тип данных AryOut[] на любую из битовых строк (BYTE, WORD, DWORD, LWORD).
- Если значение "Size" равно 0, то функция возвращает значение True, а значение AryOut[] остается неизменным.
- Если значение In[] превышает допустимый диапазон, функция возвращает значение FALSE, а значение AryOut[] остается неизменным.
- Если значение "Size" превышает область массива в In[] или AryOut[], нужно избежать исключений ПЛК.

- Типы данных In[] и AryOut[] должны совпадать, в противном случае возвращаемое значение будет FALSE, а значение AryOut[] останется неизменным.

(5) Пример программы

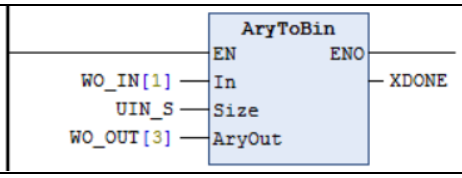
ДЕМО

Задание переменных

```
VAR
   WO_IN: ARRAY[0..5] OF WORD:= [16#9,16#16,16#26,16#168,16#12,16#512];
   UIN_S: UINT:= 3;
   WO_OUT: ARRAY[0..5] OF WORD;
   XDONE: BOOL;
END_VAR
```

Пример программы

LD



ST

```
XDONE:=AryToBin(
    In:= WO_IN[1],
    Size:= UIN_S,
    AryOut:= WO_OUT[3]);
```

Результат операции

Expression	Type	Value
WO_IN	ARRAY [0..5] OF WO...	
WO_IN[0]	WORD	16#0009
WO_IN[1]	WORD	16#0016
WO_IN[2]	WORD	16#0026
WO_IN[3]	WORD	16#0168
WO_IN[4]	WORD	16#0012
WO_IN[5]	WORD	16#0512
UIN_S	UINT	16#0003
WO_OUT	ARRAY [0..5] OF WO...	
WO_OUT[0]	WORD	16#0000
WO_OUT[1]	WORD	16#0000
WO_OUT[2]	WORD	16#0000
WO_OUT[3]	WORD	16#0010
WO_OUT[4]	WORD	16#001A
WO_OUT[5]	WORD	16#00A8
XDONE	BOOL	TRUE

1-4-7-22. Преобразование битовой строки в код Грея BinToGray_XXX

(1) Обзор команды

Преобразование битовой строки в код Грея.

Преобразование битовой строки в код Грея [BinToGray_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BinToGray_XXX	Преобразование битовой строки в код Грея		xDone:=BinToGray_BYTE(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	BYTE/WORD/DWORD/LWORD	В качестве типа данных	0	Битовая строка
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	Соответствует элементу In	В качестве типа данных	-	Код Грея

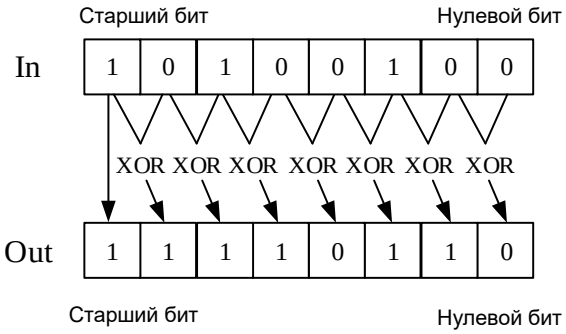
(3) Функция

- Преобразование объекта битовой строки "In" в код Грея.
- Имя команды зависит от типа данных "In" и "Out". Например, если типом данных параметров "In" и "Out" является WORD, команда будет иметь имя BinToGray_WORD.
- На примере типа данных BYTE для параметров "In" и "Out" рассмотрим процедуру преобразования:

(1) Старший бит (7-й бит) в "In" напрямую используется как старший бит (7-й бит) в "Out".

(2) XOR 7-го значения "In" и 6-го значения "In" — это 6-е значение "Out".

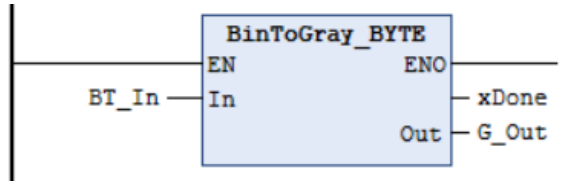
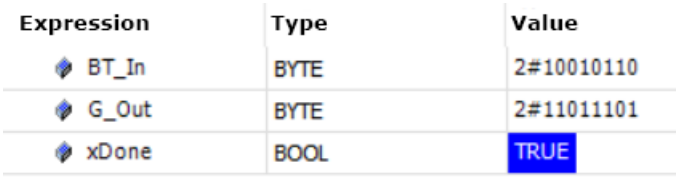
(3) Тем же способом определите значение младшего бита (нулевого) в "Out".



(4) Примечание

- Рекомендуется унифицировать типы данных для "In" и "Out".

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR BT_In: BYTE; G_Out: BYTE; xDone: BOOL; END_VAR	
Пример программы		xDone:=BinToGray_BYTE(In:= BT_In, Out=> G_Out);
Результат операции		

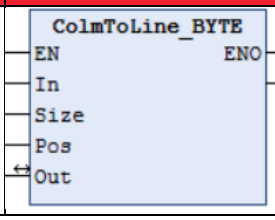
1-4-7-23. Преобразование битовой последовательности/битовой строки в массив ColmToLine_XXX

(1) Обзор команды

Извлечение значения битов в указанном положении каждого элемента массива и вывод его в виде битовой строки.

Преобразование битовой последовательности/битовой строки в массив [ColmToLine_XXX]			
Условие	Запуск фронтом сигнала	Совместимые	PMP3xx

выполнения		модели	
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ColmToLine_XXX	Преобразование битовой последовательности/битовой строки в массив		xDone:=ColmToLine_BYTE(In:= , Size:= , Pos:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Массив объектов преобразования	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Массив объектов преобразования
Size	Количество элементов для объектов преобразования	USINT	В качестве типа данных	1	Количество элементов в In[], являющихся объектами преобразования
Pos	Преобразуемое положение	USINT	В качестве типа данных	0	Преобразуемое положение
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	BYTE/WORD/ DWORD/LWORD	В качестве типа данных	-	Результат преобразования

(3) Функция

- Извлечение значений битов в указанных положениях каждого элемента в массиве, их упорядочивание и вывод в виде битовых строк.
- Сначала извлеките элементы "Size" массива объектов преобразования In[],

начиная с входного элемента массива In[]. После этого извлеките значение положения "Pos" упомянутых выше элементов. Добавьте значение в битовую строку из битов "Size" и сохраните его в положении младшего бита в результате преобразования "Out". Сохраните FALSE в остальных битах "Out".

- Имя команды зависит от типа данных "Out". Например, если типом данных для "Out" является BYTE, команда будет иметь имя ColmToLine_BYTE.

(4) Примечание

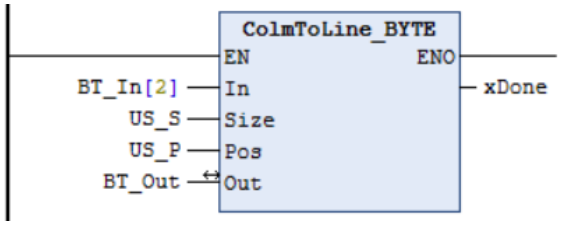
- Если значение "Size" равно 0, то функция возвращает значение True, а все биты в "Out" будут иметь значение FALSE.
- Если значение "Size" превышает допустимый диапазон, функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Если значение "Pos" превышает допустимый диапазон, функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Если значение "Size" превышает область массива в In[], нужно избежать исключений ПЛК.
- Если тип данных, передаваемый через "In[]", не относится к диапазону битовых строк, то при компиляции возникает ошибка.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   BT_In: ARRAY[0..5] OF BYTE:= [82,176,13,47,125,34];
   US_S: USINT:= 4;
   US_P: USINT:= 1;
   BT_Out: BYTE;
   xDone: BOOL;
END_VAR
```

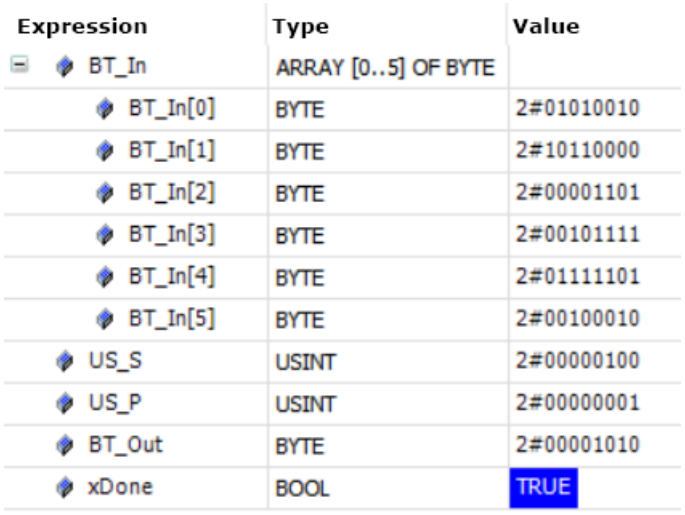
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
xDone:=ColmToLine_BYTE(
    In:= BT_In[2],
    Size:= US_S,
    Pos:= US_P,
    Out:= BT_Out);
```

</td></tr>
</table>

Результат операции	

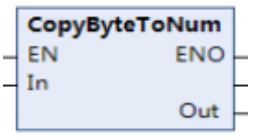
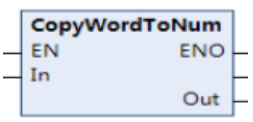
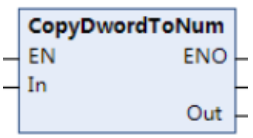
Expression	Type	Value
BT_In	ARRAY [0..5] OF BYTE	
BT_In[0]	BYTE	2#01010010
BT_In[1]	BYTE	2#10110000
BT_In[2]	BYTE	2#00001101
BT_In[3]	BYTE	2#00101111
BT_In[4]	BYTE	2#01111101
BT_In[5]	BYTE	2#00100010
US_S	USINT	2#00000100
US_P	USINT	2#00000001
BT_Out	BYTE	2#00001010
xDone	BOOL	TRUE

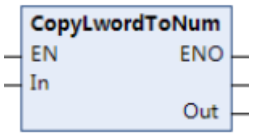
1-4-7-24. Режим копирования битов CopyByteToNum/CopyWordToNum/CopyDwordToNum/ CopyLwordToNum

(1) Обзор команды

Копирование содержимого битовой строки напрямую в целое число со знаком.

Режим копирования битов [CopyByteToNum/CopyWordToNum/CopyDwordToNum/ CopyLwordToNum]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
CopyByteToNum	Режим копирования битов		XDONE:=CopyByteToNum(In:= , Out=>);
CopyWordToNum	Режим копирования битов		XDONE:=CopyWordToNum(In:= , Out=>);
CopyDwordToNum	Режим копирования битов		XDONE:=CopyDwordToNum(In:= , Out=>);

CopyLwordToNum	Режим копирования битов		XDONE:=CopyLwordToNum(In:= , Out=>);

(2) Связанные переменные

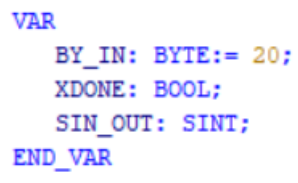
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник копирования	BYTE/WORD/DWORD/LWORD	В качестве типа данных	0	Источник копирования
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Область вставки	SINT/INT/DINT/LINT	В качестве типа данных	(*)	Область вставки

(3) Функция

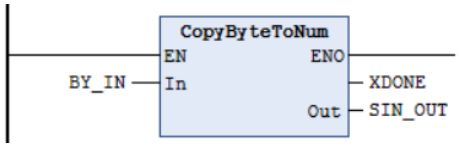
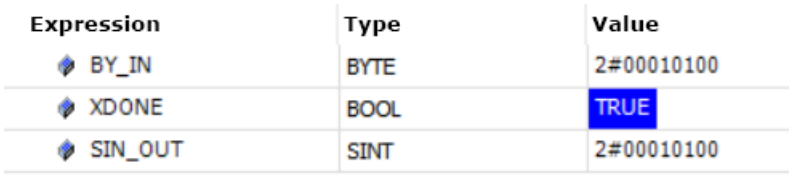
- Копирование содержимого источника без какой-либо обработки и копирование его напрямую в область вставки "Out".
- Из-за разных сочетаний типов данных для "In" и "Out" существуют четыре типа названий команд.

“In”	“Out”	Название команды
BYTE	SINT	CopyByteToNum
WORD	INT	CopyWordToNum
DWORD	DINT	CopyDwordToNum
LWORD	LINT	CopyLwordToNum

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		

```
VAR
   BY_IN: BYTE:= 20;
   XDONE: BOOL;
   SIN_OUT: SINT;
END_VAR
```

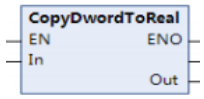
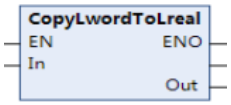
Пример программы		XDONE:=CopyByteToNum(In:= BY_IN, Out=> SIN_OUT);
Результат операции		

1-4-7-25. Режим копирования битов CopyDwordToReal/CopyLwordToLreal

(1) Обзор команды

Копирование содержимого битовой строки напрямую в действительное число.

Режим копирования битов [CopyDwordToReal/CopyLwordToLreal]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
CopyDwordToReal	Режим копирования битов		XDONE:=CopyDwordToReal(In:= , Out=>);
CopyLwordToLreal	Режим копирования битов		XDONE:=CopyLwordToLreal(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник копирования	DWORD/LWORD	В качестве типа данных	0	Источник копирования

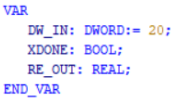
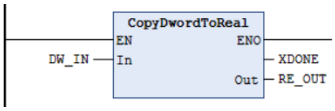
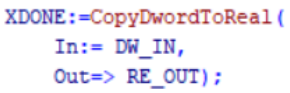
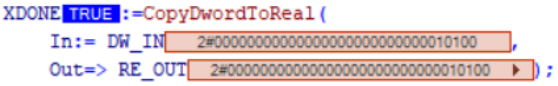
VAR_OUTPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Out	Область вставки	REAL/LREAL	В качестве типа данных	(*)	Область вставки

(3) Функция

- Копирование содержимого источника без какой-либо обработки и копирование его напрямую в область вставки "Out".
- Из-за разных сочетаний типов данных для "In" и "Out" существуют два типа названий команд.

“In”	“Out”	Название команды
DWORD	REAL	CopyDwordToReal
LWORD	LREAL	CopyLwordToLreal

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

1-4-7-26. Режим копирования битов CopyRealToDWord/CopyLrealToLword

(1) Обзор команды

Копирование содержимого действительных чисел напрямую в битовую строку.

Режим копирования битов [CopyRealToDWord/CopyLrealToLword]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

Библиотека	PP_DataUtils	

Команда	Название	Графическое представление	ST
CopyRealToDWord	Режим копирования битов		XDONE:=CopyRealToDWord(In:= , Out=>);
CopyLrealToLword	Режим копирования битов		XDONE:=CopyLrealToLword(In:= , Out=>);

(2) Связанные переменные

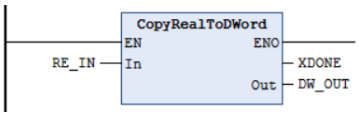
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник копирования	REAL/LREAL	В качестве типа данных	0	Источник копирования
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Область вставки	DWORD/LWORD	В качестве типа данных	(*)	Область вставки

(3) Функция

- Копирование содержимого источника без какой-либо обработки и копирование его напрямую в область вставки "Out".
- Из-за разных сочетаний типов данных для "In" и "Out" существуют два типа названий команд.

“In”	“Out”	Название команды
REAL	DWORD	CopyRealToDWord
LREAL	LWORD	CopyLrealToLword

(4) Пример программы

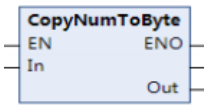
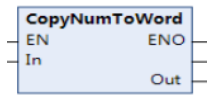
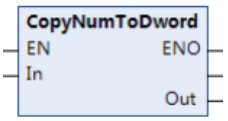
ДЕМО	LD	ST
Задание переменных		VAR RE_IN: REAL:= 8.9; DW_OUT: DWORD; XDONE: BOOL; END_VAR
Пример программы		XDONE:=CopyRealToDWord(In:= RE_IN, Out=> DW_OUT);
Результат операции	XDONE TRUE :=CopyRealToDWord(In:= RE_IN 2#01000001000011100110011001100110 , Out=> DW_OUT 2#01000001000011100110011001100110);	

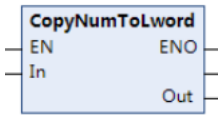
1-4-7-27. Режим копирования битов CopyNumToXXX

(1) Обзор команды

Копирование содержимого целых чисел со знаком напрямую в действительные числа.

Режим копирования битов [CopyNumToXXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
CopyNumToByte	Режим копирования битов		XDONE:=CopyNumToByte(In:= , Out=>);
CopyNumToWord	Режим копирования битов		XDONE:=CopyNumToWord(In:= , Out=>);
CopyNumToDword	Режим копирования битов		XDONE:=CopyNumToDword(In:= , Out=>);

CopyNumToLword	Режим копирования битов		XDONE:=CopyNumToLword(In:= , Out=>);

(2) Связанные переменные

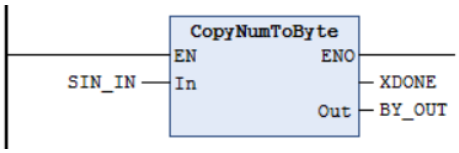
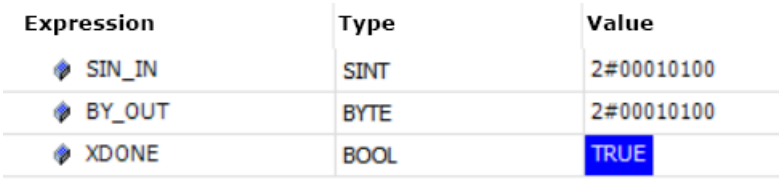
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник копирования	SINT/INT/DINT/LINT	В качестве типа данных	0	Источник копирования
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Область вставки	BYTE/WORD/DWORD/LWORD	В качестве типа данных	(*)	Область вставки

(3) Функция

- Копирование содержимого источника без какой-либо обработки и копирование его напрямую в область вставки "Out".
- Из-за разных сочетаний типов данных для "In" и "Out" существуют четыре типа названий команд.

“In”	“Out”	Название команды
SINT	BYTE	CopyNumToByte
INT	WORD	CopyNumToWord
DINT	DWORD	CopyNumToDword
LINT	LWORD	CopyNumToLword

(4) Пример программы

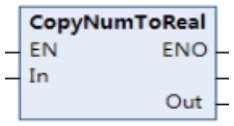
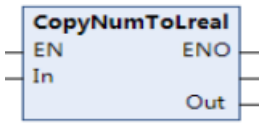
ДЕМО	LD	ST
Задание переменных	VAR SIN_IN: SINT:= 20; BY_OUT: BYTE; XDONE: BOOL; END_VAR	
Пример программы		XDONE:=CopyNumToByte(In:= SIN_IN, Out=> BY_OUT);
Результат операции		

1-4-7-28. Режим копирования битов CopyNumToReal/CopyNumToLreal

(1) Обзор команды

Копирование содержимого целых чисел со знаком напрямую в действительные числа.

Режим копирования битов [CopyNumToReal/CopyNumToLreal]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
CopyNumToReal	Режим копирования битов		XDONE:=CopyNumToReal(In:= , Out=>);
CopyNumToLreal	Режим копирования битов		XDONE:=CopyNumToLreal(In:= , Out=>);

(2) Связанные переменные

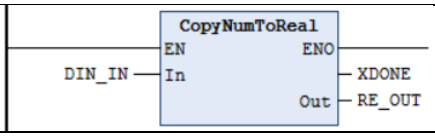
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник копирования	DINT/LINT	В качестве типа данных	0	Источник копирования
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Область вставки	REAL/LREAL	В качестве типа данных	(*)	Область вставки

(3) Функция

- Копирование содержимого источника без какой-либо обработки и копирование его напрямую в область вставки "Out".
- Из-за разных сочетаний типов данных для "In" и "Out" существуют два типа названий команд.

“In”	“Out”	Название команды
DINT	REAL	CopyNumToReal
LINT	LREAL	CopyNumToLreal

(4) Пример программы

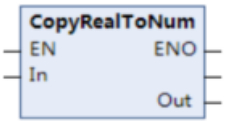
ДЕМО	LD	ST
Задание переменных		VAR DIN_IN: DINT:= 20; RE_OUT: REAL; XDONE: BOOL; END_VAR
Пример программы		XDONE:=CopyNumToReal(In:= DIN_IN, Out=> RE_OUT);
Результат операции	XDONE TRUE :=CopyNumToReal(In:= DIN_IN 2#00000000000000000000000000010100, Out=> RE_OUT 2#00000000000000000000000000010100);	

1-4-7-29. Режим копирования битов CopyRealToNum/CopyLrealToNum

(1) Обзор команды

Копирование содержимого действительных чисел напрямую в целые числа.

Режим копирования битов [CopyRealToNum/CopyLrealToNum]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
CopyRealToNum	Режим копирования битов		XDONE:=CopyRealToNum(In:= , Out=>);
CopyLrealToNum	Режим копирования битов		XDONE:=CopyLrealToNum(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Источник копирования	REAL/LREAL	В качестве типа данных	0	Источник копирования
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Область вставки	DINT/LINT	В качестве типа данных	(*)	Область вставки

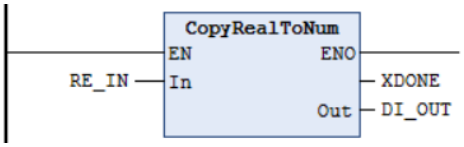
(3) Функция

- Копирование содержимого источника без какой-либо обработки и копирование его напрямую в область вставки "Out".
- Из-за разных сочетаний типов данных для "In" и "Out" существуют два типа

названий команд.

“In”	“Out”	Название команды
REAL	DINT	CopyRealToNum
LREAL	LINT	CopyLrealToNum

(4) Пример программы

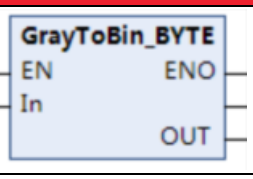
ДЕМО	LD	ST
Задание переменных		VAR RE_IN: REAL:= 8.8; DI_OUT: DINT; XDONE: BOOL; END_VAR
Пример программы		XDONE:=CopyRealToNum(In:= RE_IN, Out=> DI_OUT);
Результат операции	XDONE TRUE :=CopyRealToNum(In:= RE_IN 2#01000001000011001100110011001101 , Out=> DI_OUT 2#01000001000011001100110011001101);	

1-4-7-30. Преобразование кода Грея в битовую строку GrayToBin_XXX

(1) Обзор команды

Преобразование кода Грея в битовую строку

Преобразование кода Грея в битовую строку [GrayToBin_XXX]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
GrayToBin_XXX	Преобразование кода Грея в битовую строку		XDONE:=GrayToBin_BYTE(In:= , Out=>);

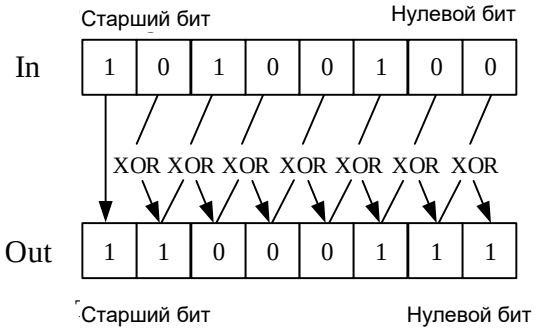
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	BYTE/WORD/DWORD/LWORD	В качестве типа данных	0	Код Грея
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	Соответствует элементу In	В качестве типа данных	-	Битовая строка

(3) Функция

- Преобразование объекта преобразования кода Грея "In" в битовую строку.
- Имя команды зависит от типа данных "In" и "Out". Например, если типом данных параметров "In" и "Out" является WORD, команда будет иметь имя GrayToBin_WORD.
- На примере типа данных BYTE для параметров "In" и "Out" рассмотрим процедуру преобразования:

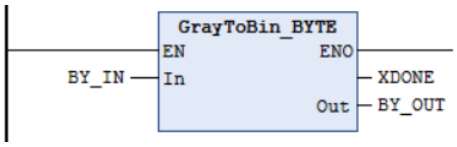
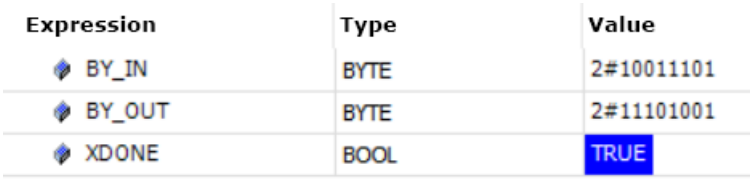
  1. Старший бит (7-й бит) в "In" напрямую используется как старший бит (7-й бит) в "Out".
  2. XOR 6-го значения "In" и 7-го значения "Out" — это 6-е значение "Out".
  3. Тем же способом определите значение младшего бита (нулевого) в "Out".



(4) Примечание

- Рекомендуется унифицировать типы данных для "In" и "Out".

(5) Пример программы

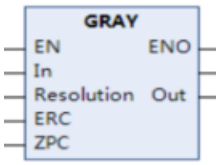
ДЕМО	LD	ST
Задание переменных	VAR BY_IN: BYTE:= 2#10011101; BY_OUT: BYTE; XDONE: BOOL; END_VAR	
Пример программы		XDONE:=GrayToBin_BYTE(In:= BY_IN, Out=> BY_OUT);
Результат операции		

1-4-7-31. Преобразование кода Грея в угол GRAY

(1) Обзор команды

Преобразование кода Грея в угол.

Преобразование кода Грея в угол [Gray]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
GRAY	Преобразование кода Грея в угол		XDONE:=GRAY(In:= , Resolution:= , ERC:= , ZPC:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	WORD	В качестве типа данных	0	Код Грея
Resolution	Разрешение	_eGRY_RESOLUTION	В качестве типа данных	_R256	Разрешение
ERC	Значение остаточной коррекции энкодера	UINT	В качестве типа данных	0	Значение остаточной коррекции энкодера
ZPC	Значение коррекции начала координат	UINT	В качестве типа данных	0	Значение коррекции начала координат
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	LREAL	В качестве типа данных	-	Результат преобразования

(3) Функция

- Преобразование выходного значения "In" поворотного энкодера, представленного в виде кода Грея, в значение угла. Единицей результата преобразования "Out" являются «градусы».
- Тип данных "Resolution" — это элемент перечисления _eGRY_RESOLUTION. Значения элементов перечисления:

Элемент перечисления	Значение	Элемент перечисления	Значение
_R256	256	_R10B	10 битов (1024)
_R1B	1 бит (2)	_R11B	11 битов (2048)
_R2B	2 бита (4)	_R12B	12 битов (4096)

_R3B	3 бита (8)	_R13B	13 битов (8192)
_R4B	4 бита (16)	_R14B	14 битов (16384)
_R5B	5 битов (32)	_R15B	15 битов (32768)
_R6B	6 битов (64)	_R360	360
_R7B	7 битов (128)	_R720	720
_R8B	8 битов (256)	_R1024	1024
_R9B	9 битов (512)		

- Код Грея — это тип переменного двоичного кода. Его характерной особенностью является то, что любые два соседних кода различаются только одной двоичной цифрой.
- Код Грея обычно используют для вывода данных с абсолютных энкодеров и других устройств.
- Ниже представлены 4-битные двоичные коды и коды Грея.

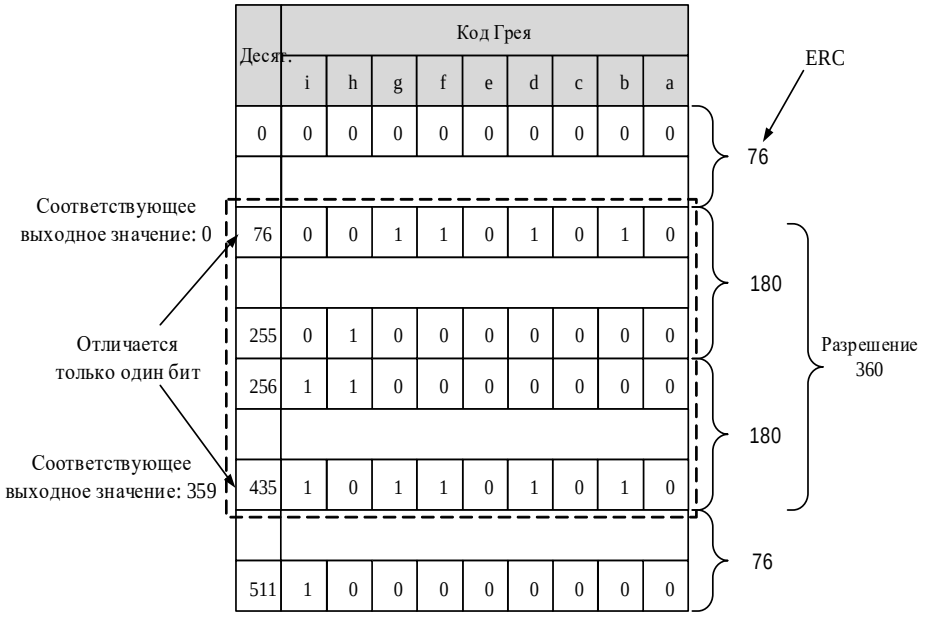
Десятичное число	Даоичный код				Код Грея			
	2^3	2^2	2^1	2^0	d	c	b	a
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0

Десятичное число	Даоичный код				Код Грея			
	2^3	2^2	2^1	2^0	d	c	b	a
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

- Если используется код Грея, то при увеличении или уменьшении выходного значения энкодера на 1 меняется только один бит значения, что исключает вывод неверных значений. Значение остаточной коррекции энкодера "ERC" — это значение коррекции, заданное в диапазоне кодов Грея для того, чтобы максимальное и минимальное выходные значения энкодера отличались только на один бит в коде Грея, когда разрешение энкодера не является степенью числа 2.

- Пример: при использовании абсолютного энкодера с разрешением 360. Количество цифр в коде Грея — 9. Диапазон значений, которые могут быть представлены 9 битами, составляет 0-511. При этом в диапазоне 180 градусов от центра 0-511, то есть 76-435, следует использовать код Грея. Таким образом, код Грея, представляющий выходное значение 0, равен 001101010 (десятичное число 76), а код Грея, представляющий выходное значение 359, равен 101101010 (десятичное число 435), причем между ними есть только одно различие.

- Значение остаточной коррекции энкодера "ERC" составляет 76.



- Значение коррекции начала координат "ZPC" устанавливается, когда

необходимо сместить угол начала координат поворотного энкодера. Например, когда начало координат поворотного энкодера с разрешением 256 смещено на 90°, значение "ZPC" равно 256 x (90/360)=64.

- Если “In”=WORD#16#256, “Resolution”=_R10B, “ERC”=UINT#0, “ZPC”=UINT#2, можно привести следующие примеры:

Во-первых, из-за разрешения в 10 битов одна единица кода Грея составляет 360/1024=0,3515625°.

Значение десятичного числа, соответствующего коду Грея 16#256, равно 923. Таким образом, значение угла до коррекции составляет 0,3515625°×923=324,492188°.

Значение "ERC" равно 0, а значение "ZPC" равно 2. Таким образом, значение скорректированного угла составляет 324,492188°- (0+2) ×0,3515625°=323,7890625°.

Значение "Out" — LREAL #323,7890625

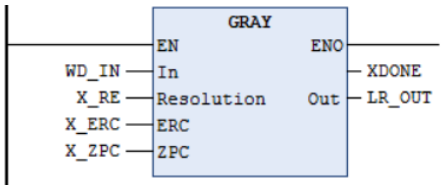
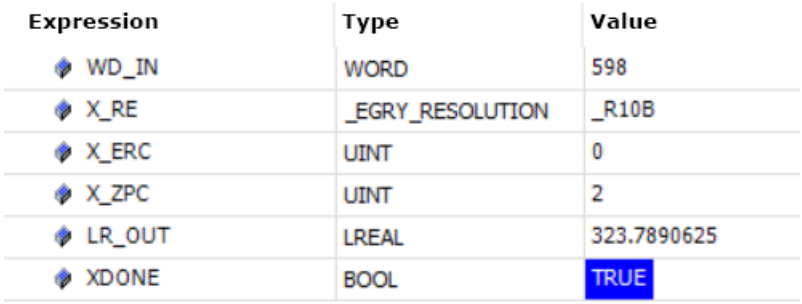
(4) Примечание

Исключения возникают в описанных ниже случаях. Функция возвращает значение FALSE, а "Out" инициализируется равным 0.

- Когда значение "ERC" превышает разрешение, указанное в "Resolution".
- Когда значение "ZPC" превышает разрешение, указанное в "Resolution".
- Если значение после преобразования "In" в ранг меньше, чем значение "ERC".
- Если значение, скорректированное по "ERC", превышает разрешение, указанное в "Resolution".

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR WD_IN: WORD:= 16#256; X_RE: _eGRY_RESOLUTION:= _eGRY_RESOLUTION._R10B; X_ERC: UINT:= 0; X_ZPC: UINT:= 2; LR_OUT: LREAL; XDONE: BOOL; END_VAR	

Пример программы		XDONE:=GRAY(In:= WD_IN, Resolution:= X_RE, ERC:= X_ERC, ZPC:= X_ZPC, Out=> LR_OUT);
Результат операции		

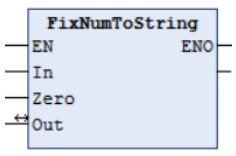
Expression	Type	Value
WD_IN	WORD	598
X_RE	_EGRY_RESOLUTION	_R10B
X_ERC	UINT	0
X_ZPC	UINT	2
LR_OUT	LREAL	323.7890625
XDONE	BOOL	TRUE

1-4-7-32. Преобразование чисел с фиксированной десятичной точкой в десятичные строки FixNumToString

(1) Обзор команды

Преобразование чисел с фиксированной десятичной точкой со знаком в десятичные строки.

Преобразование чисел с фиксированной десятичной точкой в десятичные строки [FixNumToString]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	PMP3xx: версия 2.2.0 и более поздние версии	Программное обеспечение	Codesys версии 2.2.0 и более поздние версии (модели серии PMP3xx)
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
FixNumToString	Преобразование чисел с фиксированной десятичной точкой в десятичные строки		XDONE:=FixNumToString(In:= , Zero:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	DWORD	В качестве типа данных	0	Число с фиксированной десятичной точкой со знаком
Zero	Отображение нулей	BOOL	TRUE/FALSE	FALSE	Индикация, появляющаяся, когда количество десятичных знаков меньше 3. True: Заполнение нулями FALSE: Нули не добавляются
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	STRING	В качестве типа данных	-	Строка с 10 символами

(3) Функция

- Преобразование числа с фиксированной десятичной точкой со знаком "In" в десятичную строку. Последовательность преобразования:

1. Преобразование "In" из шестнадцатеричного формата в десятичный формат.

2. Деление значения на 1000.

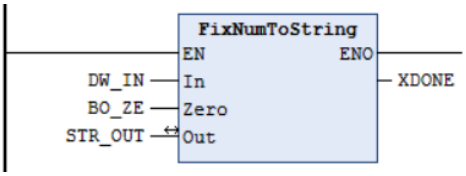
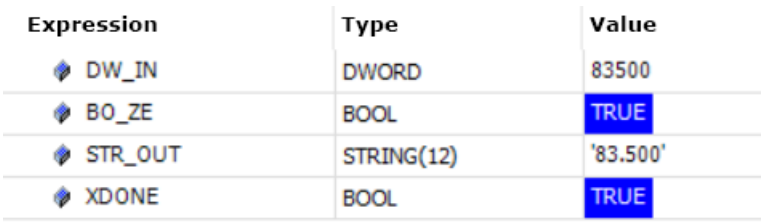
- Когда количество десятичных знаков в "In" меньше 3, отображается ноль. "Zero" определяет, нужно ли вводить "0" перед третьим десятичным знаком в "Out". Если значение "Zero" — true, вводится "0". Если значение FALSE, "0" не вводится.

- На конце "Out" стоит символ NULL.

(4) Примечание

- Если результат преобразования превышает допустимый диапазон параметра "Out", функция возвращает значение FALSE, а значение "Out" остается неизменным.

(5) Пример программы

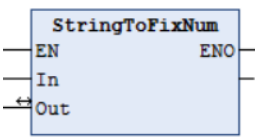
ДЕМО	LD	ST
Задание переменных		VAR DW_IN: DWORD:= 16#1462C; BO_ZE: BOOL:= TRUE; STR_OUT: STRING(12); XDONE: BOOL; END_VAR
Пример программы		XDONE:=FixNumToString(In:= DW_IN, Zero:= BO_ZE, Out:= STR_OUT);
Результат операции		

1-4-7-33. Преобразование строки в число с фиксированной десятичной точкой StringToFixNum

(1) Обзор команды

Преобразование десятичной строки в число с фиксированной десятичной точкой со знаком.

Преобразование строки в число с фиксированной десятичной точкой [StringToFixNum]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
StringToFixNum	Преобразование строки в число с фиксированной десятичной точкой		XDONE:=StringToFixNum(In:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	STRING	В качестве типа данных	0	Строка с 10 символами
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	DWORD	В качестве типа данных	-	Фиксированное количество десятичных знаков

(3) Функция

- Преобразование десятичной строки "In" в фиксированное количество десятичных знаков. Последовательность преобразования:

  1. Умножение значения, представленного параметром "In", на 1000.
  2. Округление десятичной точки этого значения.
  3. Преобразование значения в 32-битный шестнадцатеричный формат (тип DWORD).

Название	Формат
Символьная часть	• Одиночные символы '+' или '-', которые появляются после последовательных символов ' ' (пробелы 16#20), появляющихся в начале строки, не учитываются. • Требования можно не соблюдать. • В то же время не учитываются последовательные пробелы после символов.

Целая часть	• В качестве целой части возьмите последовательные числовые значения ('0' - '9') от символа до десятичной точки. Иногда символы также опускаются. Между символьной и целой частями иногда могут быть последовательные пробелы. • Если десятичная точка или десятичная часть отсутствует, то экспоненциальной части предшествует целая часть. • Если десятичная точка, десятичная и экспоненциальная части отсутствуют, то целая часть находится до конца строки. • Целую часть нельзя опустить. • Максимальное количество цифр в целой части — это максимальная длина строки в 1986 знаков, за вычетом символа, десятичной точки, десятичной части, экспоненциальной части и всех байтов пространства до и после символа.
Десятичная точка	• Одиночный знак '.' после целой части (точка) — это десятичная точка. • Если десятичная часть отсутствует, десятичную точку опускают.
Десятичная часть	• В качестве десятичной части принимают последовательные числовые значения (от '0' до '9') от десятичной точки до экспоненциальной части. • Если экспоненциальная часть отсутствует, то до конца строки используют десятичную часть. • Десятичные части можно опускать. Если нет десятичной точки, то нет и десятичной части. • Максимальное количество десятичных знаков — 15.
Экспоненциальная часть	• Одиночные символы 'e' или 'E' после десятичной части и последовательные числовые значения (от '0' до '9') от одиночного символа '+' или '-' до конца строки — это экспоненциальная часть. • Если десятичная часть отсутствует, то строка после десятичной точки является экспоненциальной частью. • Если десятичная точка или десятичная часть отсутствует, то приведенная выше строка после целого числа является экспоненциальной частью. • Экспоненциальную часть можно опустить. • Максимальное количество цифр в экспоненциальной части — 3.

(4) Примечание

- Округление после 4 десятичных знаков для "In".

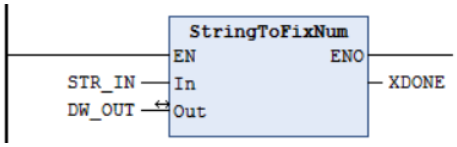
- Игнорирование подчеркивания (16#5F) в строке "In".
- Если "In" не заканчивается символом NULL, функция возвращает значение FALSE, а "Out" остается неизменным.
- Если содержимое "In" представляет собой символ, который нельзя преобразовать в числовое значение, функция возвращает значение FALSE, а "Out" остается неизменным.
- Если в содержимом "In" есть десятичная точка, но нет десятичной части, функция возвращает значение FALSE, а "Out" остается неизменным.
- Если результат преобразования превышает допустимый диапазон параметра "Out", функция возвращает значение FALSE, а значение "Out" остается неизменным.
- В символьной части знак "+" можно опустить, а отрицательные числа используются для обработки переполнения. При появлении двух или более символов функция выполнения возвращает значение FALSE.
- Если в середине и конце входных данных есть пробелы, функция возвращает значение FALSE, а "Out" остается неизменным.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   STR_IN: STRING(1985);
   DW_OUT: DWORD;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:=StringToFixNum(
    In:= STR_IN,
    Out:= DW_OUT);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
STR_IN	STRING(1985)	'2.2563'
DW_OUT	DWORD	2256
XDONE	BOOL	TRUE

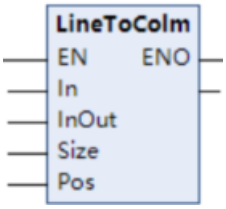
</td></tr>
</table>

1-4-7-34. Преобразование строк/столбцов LineToColm

(1) Обзор команды

Разложение битовой строки и ее вывод в указанное положение бита элемента массива.

Преобразование строк/столбцов [LineToColm]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
LineToColm	Преобразование строк/столбцов		XDONE:=LineToColm(In:= , InOut:= , Size:= , Pos:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	BYTE/WORD DWORD/LWORD	В качестве типа данных	-	Объект преобразования
Size	Количество элементов в результате преобразования	UINT	В качестве типа данных	1	Количество элементов в результате преобразования
Pos	Положение бита преобразования	UINT	В качестве типа данных	0	Преобразуемое положение

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
InOut	Массив с результатами преобразования	BYTE/WORD DWORD/LWORD	В качестве типа данных	-	Массив с результатами преобразования

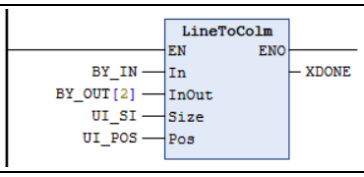
(3) Функция

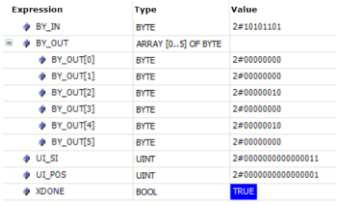
- Разложение битовой строки и ее вывод в указанное положение бита элемента массива.
- Сначала извлеките бит "Size" из младшего бита объекта преобразования "In" и разложите его на каждый бит.
- После этого сохраните его в положении "Pos" каждого элемента, начиная с InOut[0], в массиве с результатами преобразования InOut[]. Существуют сохраненные элементы массива "Size".
- Для сохранения этого значения используются несохраненные биты значения.

(4) Примечание

- Если значение "Size" равно 0, значение InOut[] остается неизменным, а функция возвращает значение True.
- Если значение "Size" превышает допустимый диапазон, значение InOut[] остается неизменным, а функция возвращает значение FALSE.
- Если значение "Pos" превышает допустимый диапазон, значение InOut[] остается неизменным, а функция возвращает значение FALSE.
- Если значение "Size" превышает область массива в InOut[], нужно избежать исключений ПЛК.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR BY_IN: BYTE:= 173; BY_OUT: ARRAY[0..5] OF BYTE; UI_SI: UINT:= 3; UI_POS: UINT:= 1; XDONE: BOOL; END_VAR
Пример программы		XDONE:=LineToColm(In:= BY_IN, InOut:= BY_OUT[2], Size:= UI_SI, Pos:= UI_POS);

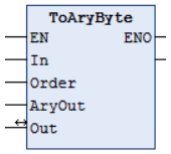
Результат операции	

1-4-7-35. Преобразование в массив байтов ToAryByte

(1) Обзор команды

Деление переменных на 1 байт и их сохранение в массиве типа BYTE.

Преобразование в массив байтов [ToAryByte]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ToAryByte	Преобразование в массив байтов		XDONE:=ToAryByte(In:= , Order:= , AryOut:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	BOOL/ BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/ REAL/LREAL/ TIME/DATE/TOD/DT	В качестве типа данных	-	Объект преобразования

Order	Порядок преобразования	_eBYTE_ORDER	В качестве типа данных	_LOW_HIGH	Преобразуемое положение

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
AryOut	Массив с результатами преобразования	BYTE	В качестве типа данных	-	Массив с результатами преобразования
Out	Количество элементов в результате преобразования	UINT	В качестве типа данных	-	Количество элементов в результате преобразования

(3) Функция

- Разделите объект преобразования "In" на единицы по 1 байту и последовательно сохраните их из массива с результатами преобразования AryOut[], начиная с AryOut [0]. Сохраните количество элементов, уже сохраненных в AryOut[], в количество элементов в результате преобразования "Out".

- Порядок, в котором значение "In" преобразуется в единицы по 1 байту, задается порядком преобразования "Order". Тип данных "Order" — это элемент перечисления _eBYTEE_ORDER. Значения элементов перечисления:

Элемент перечисления	Значение
_LOW_HIGH	Сначала младший байт, затем старший байт
_HIGH_LOW	Сначала старший байт, затем младший байт

- Если тип данных "In" имеет размер свыше 2 байтов, разделите "In" на 1 байт и сохраните его в AryOut[]. Типы данных размером 2 байта или больше:

Тип	Тип данных
Битовая строка	WORD, DWORD, LWORD
Целое число	UINT, UDINT, ULINT, INT, DINT, LINT

Действительное число	REAL, LREAL
Время, продолжительность, дата, строка	TIME, DATE, TOD, TD

- Последовательность обработки:

1. Сначала разделите значение "In" на единицы по 1 слову (2 байта).

2. Разделите слово с младшим битом на 1 байт (2 байта).

3. Если значение "Order" равно _LOW_HIGH, сохраните младший байт в AryOut[0], а старший байт — в AryOut[1]. Если значение "Order" равно _HIGH_LOW, сохраните старший байт в AryOut[0], а младший байт — в AryOut[1].

4. Точно так же разделите следующее слово на 1 байт и сохраните его в AryOut[2] и AryOut[3]. Выполняйте обработку аналогичным образом до конца значения In.

- Если тип данных "In" имеет размер 1 байт, сохранение так же происходит в AryOut[] в единицах по 1 байту. Типы данных размером 1 байт:

Тип	Тип данных
Битовая строка	BYTE
Целое число	USINT, SINT
Действительное число	Отсутствует
Время, продолжительность, дата, строка	Отсутствует

- Метод сохранения:

Значение “Order”	Метод сохранения в "AryOut[]"
LOW_HIGH	Сохраните значение "In" в AryOut[0]
HIGH_LOW	Сохраните значение "In" в AryOut[1] Сохраните 16#00 в AryOut[0].

- Если "In" имеет тип данных BOOL (1 бит), то метод сохранения в AryOut[] будет следующим:

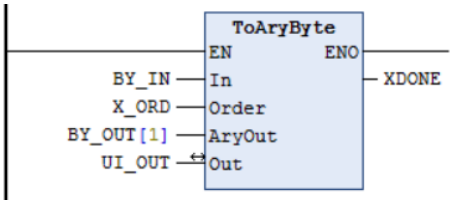
Значение “Order”	Метод сохранения в "AryOut[]"
LOW_HIGH	Сохраните значение "In" с логической суммой 16#00 в AryOut[0].

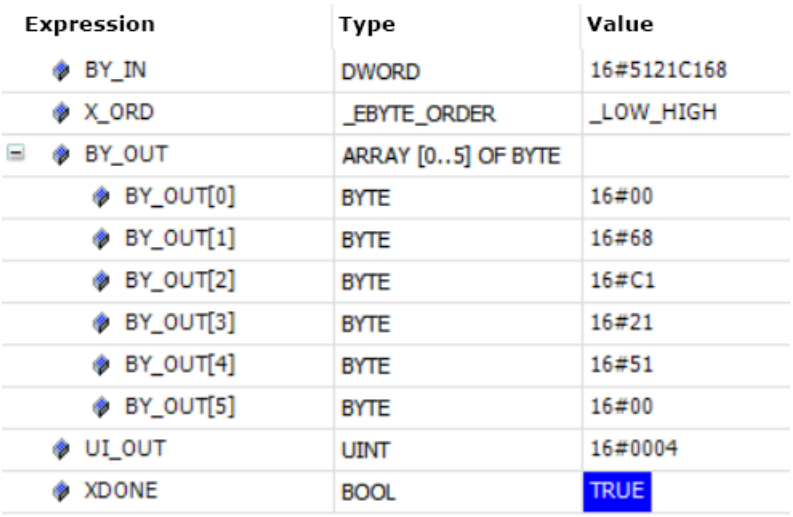
HIGH_LOW	Добавьте значение In[0] и сохраните его в AryOut[0]. Добавьте значение In[1], сохраните его в AryOut[1], после чего используйте тот же метод сохранения. Обязательно установите четное значение параметра "Out".

(4) Примечание

- Убедитесь, что входные параметры, передаваемые в "In", являются переменными. Если передана константа, во время компиляции возникнет исключение.
- Если значение "Order" равно _HIGH_LOW, а общий размер "In" равен нечетному количеству байтов, добавьте 16#00 в конец "In", установите его равным четным байтам, а затем начните преобразование.
- Если значение "Order" превышает допустимый диапазон, функция возвращает значение FALSE, а значения "Out" и "AryOut[] остаются неизменными.
- Если результат преобразования превышает область массива в AryOut[], нужно избежать исключений ПЛК.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR BY_IN: DWORD:= 16#5121C168; X_ORD: _eBYTE_ORDER:=_eBYTE_ORDER._LOW_HIGH; BY_OUT: ARRAY[0..5] OF BYTE; UI_OUT: UINT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:=ToAryByte(In:= BY_IN, Order:= X_ORD, AryOut:= BY_OUT[1], Out:= UI_OUT);

Результат операции	

Expression	Type	Value
BY_IN	DWORD	16#5121C168
X_ORD	_EBYTE_ORDER	_LOW_HIGH
BY_OUT	ARRAY [0..5] OF BYTE	
BY_OUT[0]	BYTE	16#00
BY_OUT[1]	BYTE	16#68
BY_OUT[2]	BYTE	16#C1
BY_OUT[3]	BYTE	16#21
BY_OUT[4]	BYTE	16#51
BY_OUT[5]	BYTE	16#00
UI_OUT	UINT	16#0004
XDONE	BOOL	TRUE

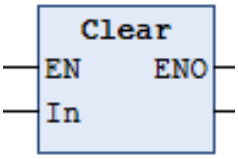
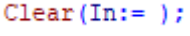
1-4-8. Команды инициализации

1-4-8-1. Инициализация Clear

(1) Обзор команды

Переменные инициализации.

Инициализация [Clear]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Clear	Инициализация		


```
Clear(In:= );
```

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект инициализации	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/LINT/ REAL/LREAL/TIME/DATE/ TOD/DT/STRING	В качестве типа данных	-	Объект инициализации

(3) Функция

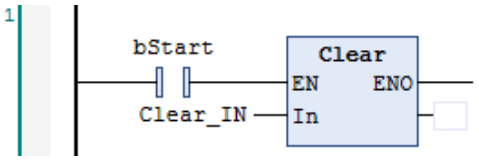
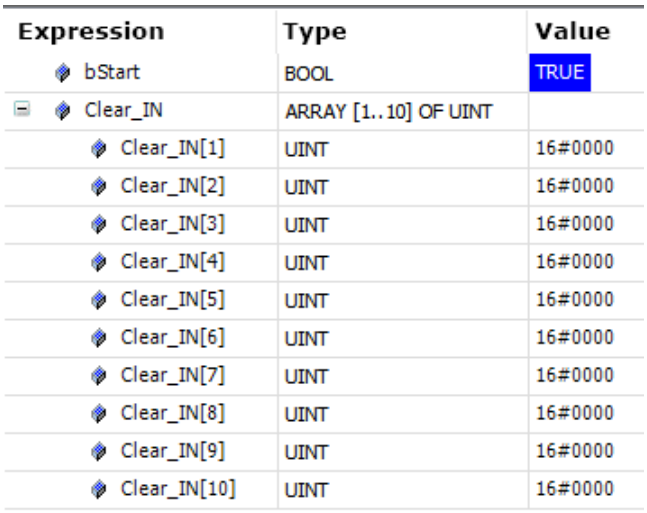
- Инициализация значения объекта инициализации "In" начальным значением по умолчанию для каждого типа данных.
- Начальные значения по умолчанию для каждого типа данных приведены в таблице ниже:

Тип данных	Начальное значение
BOOL	FALSE
BYTE, WORD, DWORD, LWORD	16#0
USINT, UINT, UDINT, ULINT, SINT, INT, DINT, LINT, REAL, LREAL	0
TIME	T#0 мс
DATE	D#1970-1-1
TOD	TOD#0:0:0
DT	DT#1970-1-1-0:0:0
STRING	‘’

- Если "In" представляет весь массив с одним элементом и одной переменной, в таблице ниже представлена соответствующая процедура обработки:

“In”	Процедура обработки
Весь массив	Инициализация всех элементов массива
Элемент массива	Инициализация только этого элемента
Вся структура	Инициализация всех элементов структуры
Один элемент структуры	Инициализация только этого элемента структуры

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR bStart:BOOL; Clear_IN:ARRAY[1..10] OF UINT:=[66,77,52,3,5,789,56,366,32,75]; END_VAR	
Пример программы		1 Clear(In:= Clear_IN);
Результат операции		

Expression	Type	Value
bStart	BOOL	TRUE
Clear_IN	ARRAY [1..10] OF UINT	
Clear_IN[1]	UINT	16#0000
Clear_IN[2]	UINT	16#0000
Clear_IN[3]	UINT	16#0000
Clear_IN[4]	UINT	16#0000
Clear_IN[5]	UINT	16#0000
Clear_IN[6]	UINT	16#0000
Clear_IN[7]	UINT	16#0000
Clear_IN[8]	UINT	16#0000
Clear_IN[9]	UINT	16#0000
Clear_IN[10]	UINT	16#0000

1-4-9. Команды регулировки зоны

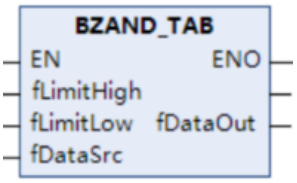
1-4-9-1. Команда регулировки зоны нечувствительности BZAND_TAB

(1) Обзор команды

Определяет, находится ли входное значение в указанных верхних и нижних пределах зоны нечувствительности, чтобы можно было отрегулировать

выходное значение.

Команда регулировки зоны нечувствительности [BZAND_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
BZAND_TAB	Команда регулировки зоны нечувствительности		XDONE:=BZAND_TAB(fLimitHigh:= , fLimitLow:= , fDataSrc:= , fDataOut=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
fLimitHigh	Верхнее предельное значение зоны нечувствительности	REAL	В качестве типа данных	0	Верхнее предельное значение зоны нечувствительности
fLimitLow	Нижнее предельное значение зоны нечувствительности	REAL	В качестве типа данных	0	Нижнее предельное значение зоны нечувствительности
fDataSrc	Исходные данные	REAL	В качестве типа данных	0	Входное значение через зону нечувствительности
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
fDataOut	Выходные данные	REAL	В качестве типа данных	-	Выходное значение через зону нечувствительности

(3) Функция

- Регулирование выходного значения путем определения, находится ли входное значение в пределах заданных верхнего и нижнего предела зоны нечувствительности. Настройте диапазон зоны нечувствительности в fLimitLow, fLimitHigh для вывода входного значения fDatasrc за пределы диапазона зоны нечувствительности в fDataOut.
- Когда fLimitLow>fDatasrc, fDataOut = fDataSrc - fLimitLow;
- Когда fLimitHigh<fDatasrc, fDataOut = fDataSrc - fLimitHigh;
- Когда fLimitLow ≤ fDataSrc ≤ fLimitHigh, fDataOut = 0.

(4) Примечание

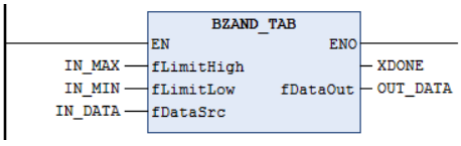
- Если вход fDataSrc — не числовое значение, выход FDataOut равен 0.
- Если fLimitLow не является числовым значением и fDataSrc>fLimitHigh, когда fDataOut=fDataSrc-fLimitHigh; fDataSrc<fLimitHigh, FDataOut равен 0.
- Если fLimitHigh не является числовым значением, и fDataSrc<fLimitLow, когда fDataOut=fDataSrc fLimitLow; fDataSrc>fLimitLow, FDataOut равен 0.
- Если fLimitHigh, fLimitLow не являются числовыми значениями, FDataOut равен 0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN_MAX: REAL:= 10;
    IN_MIN: REAL:= 1;
    IN_DATA: REAL:= -2;
    OUT_DATA: REAL;
    XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= BZAND_TAB(
    fLimitHigh:= IN_MAX,
    fLimitLow:= IN_MIN,
    fDataSrc:= IN_DATA,
    fDataOut=> OUT_DATA);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_MAX	REAL	10
IN_MIN	REAL	1
IN_DATA	REAL	-2
OUT_DATA	REAL	-3
XDONE	BOOL	TRUE

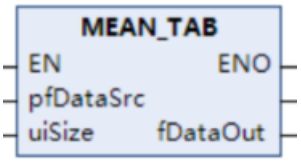
</td></tr>
</table>

1-4-9-2. Вычисление среднего значения данных MEAN_TAB

(1) Обзор команды

Вычисление среднего значения указанных данных.

Вычисление среднего значения данных [MEAN_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
MEAN_TAB	Вычисление среднего значения данных		XDONE:=MEAN_TAB(pfDataSrc:= , uiSize:= , fDataOut=>);

(2) Связанные переменные

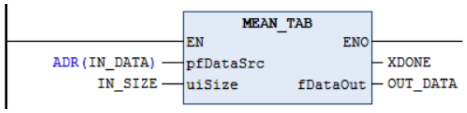
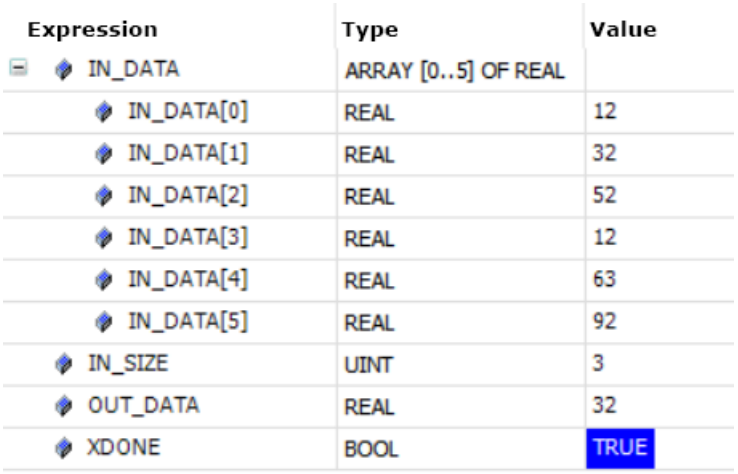
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
pfDataSrc	Исходные данные	POINTER TO REAL	-	-	Адрес входных данных
uiSize	Длина данных	UINT	1-1024	0	Длина данных
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
fDataOut	Выходные данные	REAL	В качестве типа данных	-	Выходное среднее значение

(3) Функция

- Усреднение числа uiSize, начиная с pfDataSrc, и сохранения результата в

fDataOut.

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_DATA: ARRAY[0..5] OF REAL:= [12,32,52,12,63,92]; IN_SIZE: UINT:= 3; OUT_DATA: REAL; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= MEAN_TAB(pfDataSrc:= ADR(IN_DATA), uiSize:= IN_SIZE, fDataOut=> OUT_DATA);
Результат операции		

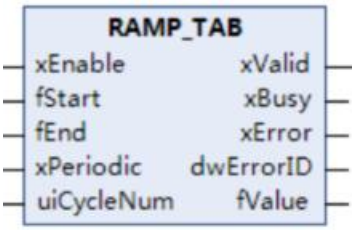
Expression	Type	Value
IN_DATA	ARRAY [0..5] OF REAL	
IN_DATA[0]	REAL	12
IN_DATA[1]	REAL	32
IN_DATA[2]	REAL	52
IN_DATA[3]	REAL	12
IN_DATA[4]	REAL	63
IN_DATA[5]	REAL	92
IN_SIZE	UINT	3
OUT_DATA	REAL	32
XDONE	BOOL	TRUE

1-4-9-3. Команда наклона RAMP_TAB

(1) Обзор команды

Команда наклона с указанным интервалом изменяет выходное значение от самого низкого значения до самого высокого значения или от самого высокого значения до самого низкого значения, обеспечивая линейную интерполяцию между высокими и низкими значениями.

Команда наклона [RAMP_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
RAMP_TAB	Команда наклона		RAMP_TAB_0(xEnable:= , fStart:= , fEnd:= , xPeriodic:= , uiCycleNum:= xValid=> , xBusy=> , xError=> , dwErrorID=> , fValue=>);

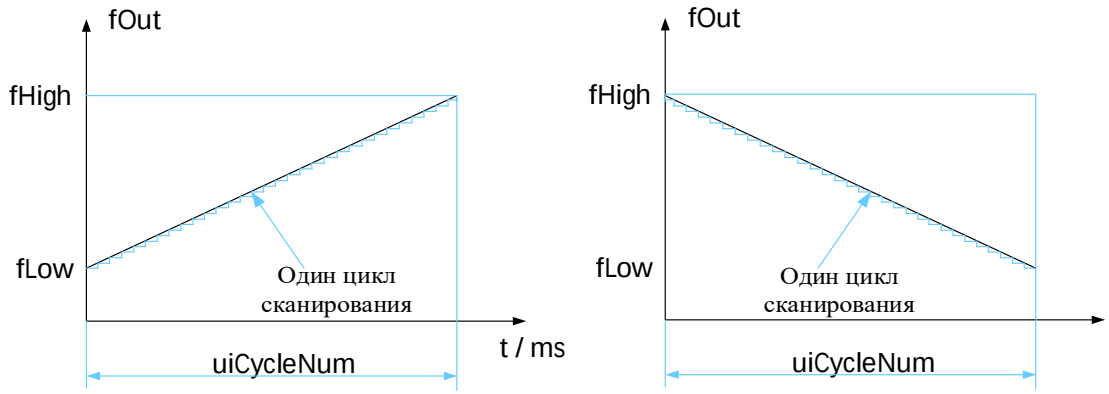
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xEnable	Выполнить	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
fStart	Начальное значение	REAL	В качестве типа данных	0	Начальное значение
fEnd	Конечное значение	REAL	В качестве типа данных	0	Конечное значение
xPeriodic	Циклический вывод	BOOL	FALSE/TRUE	FALSE	FALSE: Однократное выполнение True: Циклическое выполнение
uiCycleNum	Количество выполненных циклов	UINT	В качестве типа данных	0	Количество выполненных циклов
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
xValid	Выполнено	BOOL	FALSE/TRUE	FALSE	True: Выполнение команды завершено FALSE: Выполнение команды не

					завершено
xBusy	В процессе выполнения	BOOL	FALSE/TRUE	FALSE	True: Выполняется команда FALSE: Команда не выполняется
xError	Ошибка	BOOL	FALSE/TRUE	FALSE	True: Ошибка при выполнении команды FALSE: Во время выполнения команды ошибок не возникало
dwErrorID	Код ошибки	DWORD	-	0	Код ошибки
fValue	Текущее значение	REAL	-	0	Текущее выходное значение

(3) Функция

- Функция команды заключается в выполнении линейной интерполяции между двумя заданными точками данных в пределах заданного временного интервала и последовательном выводе значений технологических параметров в соответствии со временем выполнения сканирования до достижения конечной точки в конце интервала.



(4) Примечание

- xEnable изменяется с FALSE на True. Значение fValue снова начинает изменяться, xBusy становится True. После завершения выполнения xValid, xBusy становятся True.

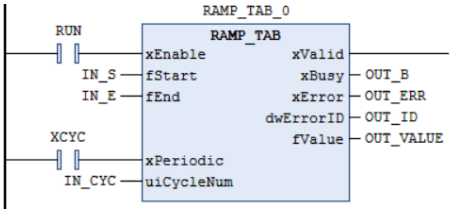
- После изменения xEnable с True на False значение fValue сохраняется.
- Только параметр xEnable равен False, а fStart, fEnd, xPeriodic, uiCycleNum можно записывать, и они могут вступать в силу, когда xEnable равен true.
- Если параметр xEnable равен False, значение fValue можно изменять, но оно не вступит в силу.
- Если параметр xEnable равен True, значения всех входных и выходных переменных могут быть изменены, но не могут вступить в силу.
- Если uiCycleNum равен 0, когда xEnable — True, то XError — True, dwErrorID — 1, xValid, xBusy — False.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   RAMP_TAB_0: RAMP_TAB;
   RUN: BOOL;
   IN_S: REAL:= 5;
   IN_E: REAL:=100;
   XCYC: BOOL;
   IN_CYC: UINT:=95;
   OUT_B: BOOL;
   OUT_ERR: BOOL;
   OUT_ID: DWORD;
   OUT_VALUE: REAL;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
RAMP_TAB_0(
   xEnable:= RUN,
   fStart:= IN_S,
   fEnd:= IN_E,
   xPeriodic:= XCYC,
   uiCycleNum:= IN_CYC,
   xValid=> XDONE,
   xBusy=> OUT_B,
   xError=> OUT_ERR,
   dwErrorID=> OUT_ID,
   fValue=> OUT_VALUE);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
RAMP_TAB_0	RAMP_TAB	
RUN	BOOL	TRUE
IN_S	REAL	5
IN_E	REAL	100
XCYC	BOOL	FALSE
IN_CYC	UINT	95
OUT_B	BOOL	TRUE
OUT_ERR	BOOL	FALSE
OUT_ID	DWORD	0
OUT_VALUE	REAL	100
XDONE	BOOL	TRUE

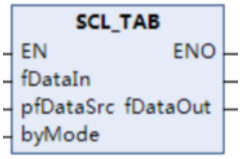
</td></tr>
</table>

1-4-9-4. Команда получения табличных координат SCL_TAB

(1) Обзор команды

Предварительно сохраните табличные данные, входные значения, а затем получите соответствующие выходные значения на основе табличных данных.

Команда получения табличных координат [SCL_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
SCL_TAB	Команда получения табличных координат		XDONE:=SCL_TAB(fDataIn:= , pfDataSrc:= , byMode:= , fDataOut=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
fDataIn	Входное значение	REAL	В качестве типа данных	0	Входные данные
pfDataSrc	Исходные данные	POINTER TO REAL	-	-	Табличные данные
byMode	Выбор режима	_modeType	1/2	_xymode	1：x1,y1;x2,y2;....xn,yn 2：x1,x2,...xn;y1,y2...yn
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
fDataOut	Выходное значение	REAL	-	-	Выходное значение, полученное по данным таблицы

(3) Функция

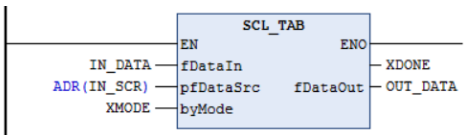
- На основе графика, определенного по таблице в pfDatasrc, найдите выходное значение fDataOut, соответствующее входному значению fDataIn.

(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   IN_DATA: REAL;
   IN_SCR: ARRAY[0..6] OF REAL:= [10,12,12,24,24,31,31];
   XMODE: _modeType:= 1;
   OUT_DATA: REAL;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= SCL_TAB(
     fDataIn:= IN_DATA,
     pfDataSrc:=ADR(IN_SCR),
     byMode:= XMODE,
     fDataOut=> OUT_DATA);
```

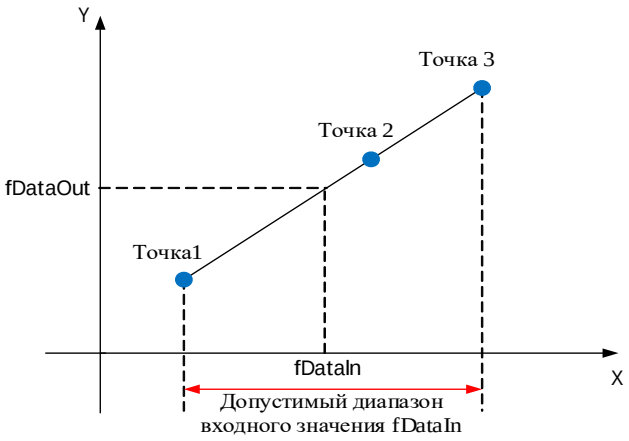
</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_DATA	REAL	18
IN_SCR	ARRAY [0..6] OF REAL	
IN_SCR[0]	REAL	10
IN_SCR[1]	REAL	12
IN_SCR[2]	REAL	12
IN_SCR[3]	REAL	24
IN_SCR[4]	REAL	24
IN_SCR[5]	REAL	31
IN_SCR[6]	REAL	31
XMODE	_MODETYPE	xymode
OUT_DATA	REAL	18
XDONE	BOOL	TRUE

</td></tr>
</table>

Анализ принципа выполнения программы представлен ниже:

<table>
<tr><th colspan="2">Анализ принципа</th></tr>
<tr><td>График табличных данных</td><td>



</td></tr>
</table>

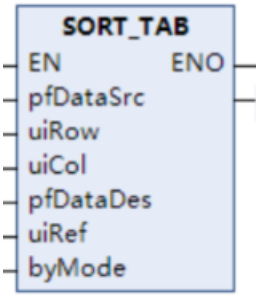
Исходные данные (pfDataSrc)		Адрес табличных данных
Данные о точках координат		(pfDataSrc)^
Точка 1	Координата X1	(pfDataSrc+1)^
	Координата Y1	(pfDataSrc+2)^
Точка 2	Координата X2	(pfDataSrc+3)^
	Координата Y2	(pfDataSrc+4)^
Точка 3	Координата X3	(pfDataSrc+5)^
	Координата Y3	(pfDataSrc+6)^

1-4-9-5. Команда сортировки табличных данных SORT_TAB

(1) Обзор команды

Сортировка массива строк uiRow × столбцов uiCol по параметру столбца или строки uiRef и сохранение результатов сортировки в области переменных, начиная с блока pfDataDes.

Команда сортировки табличных данных [SORT_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
SORT_TAB	Команда сортировки табличных данных		XDONE:=SORT_TAB(pfDataSrc:= , uiRow:= , uiCol:= , pfDataDes:= , uiRef:= , byMode:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
pfDataSrc	Исходные данные	POINTER TO REAL	-	-	Упорядочивание объектов данных
uiRow	Строка	UINT	1-1024	0	Строка данных
uiCol	Столбец	UINT	1-1024	0	Столбец данных
pfDataDes	Выходные данные	POINTER TO REAL	-	-	Вывод данных
uiRef	Строка/столбец	UINT	1-1024	0	На основе строк или столбцов
byMode	Метод размещения	BYTE	1-4	0	Метод размещения 1: Ссылка на столбец, обработка в порядке возрастания 2: Ссылка на столбец, обработка в порядке убывания 3: Ссылка на строку, обработка в порядке возрастания 4: Ссылка на строку, обработка в порядке убывания

(3) Функция

- pfDatasrc — начальная единица первой переменной в первой строке (или первой записи).
- Количество строк в массиве uiRow, также известное как количество записей.
- Количество столбцов в массиве uiCol, также известное как количество

столбцов на запись.

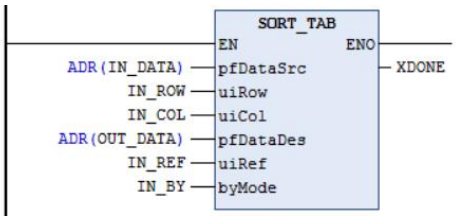
- Начальная единица, хранящаяся после сортировки в pfDataDes, занимает столько же единиц последующих переменных, сколько было переменных массива до сортировки.
- uiRef — номер строки/столбца, основанный на сортировке.
- Режим сортировки byMode 1: ссылка на столбец, обработка в порядке возрастания, 2: ссылка на столбец, обработка в порядке убывания, 3: ссылка на строку, обработка в порядке возрастания, 4: ссылка на строку, обработка в порядке убывания.

(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   IN_DATA: ARRAY[0..8] OF REAL:= [1,2,3,4,5,6,7,8,9];
   IN_ROW: UINT:= 3;
   IN_COL: UINT:= 3;
   OUT_DATA: ARRAY[0..8] OF REAL;
   IN_REF: UINT:=2;
   IN_BY: BYTE:= 2;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= SORT_TAB(
    pfDataSrc:= ADR(IN_DATA),
    uiRow:= IN_ROW,
    uiCol:= IN_COL,
    pfDataDes:= ADR(OUT_DATA),
    uiRef:= IN_REF,
    byMode:= IN_BY);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_DATA	ARRAY [0..8] OF REAL	
IN_DATA[0]	REAL	1
IN_DATA[1]	REAL	2
IN_DATA[2]	REAL	3
IN_DATA[3]	REAL	4
IN_DATA[4]	REAL	5
IN_DATA[5]	REAL	6
IN_DATA[6]	REAL	7
IN_DATA[7]	REAL	8
IN_DATA[8]	REAL	9
IN_ROW	UINT	3
IN_COL	UINT	3
OUT_DATA	ARRAY [0..8] OF REAL	
OUT_DATA[0]	REAL	3
OUT_DATA[1]	REAL	2
OUT_DATA[2]	REAL	1
OUT_DATA[3]	REAL	6
OUT_DATA[4]	REAL	5
OUT_DATA[5]	REAL	4
OUT_DATA[6]	REAL	9
OUT_DATA[7]	REAL	8
OUT_DATA[8]	REAL	7
IN_REF	UINT	2
IN_BY	BYTE	2
XDONE	BOOL	TRUE

</td></tr>
</table>

- Анализ принципа выполнения программы приводит к следующему выводу: выполнить обработку во втором столбце в порядке убывания.

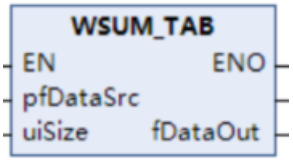
Данные до выполнения			
-	Столбец 1	Столбец 2	Столбец 3
Строка 1	pfDataSrc[0]=1	pfDataSrc[3]=4	pfDataSrc[6]=7
Строка 2	pfDataSrc[1]=2	pfDataSrc[4]=5	pfDataSrc[7]=8
Строка 3	pfDataSrc[2]=3	pfDataSrc[5]=6	pfDataSrc[8]=9
Данные после выполнения			
-	Столбец 1	Столбец 2	Столбец 3
Строка 1	pfDataSrc[0]=3	pfDataSrc[3]=6	pfDataSrc[6]=9
Строка 2	pfDataSrc[1]=2	pfDataSrc[4]=5	pfDataSrc[7]=8
Строка 3	pfDataSrc[2]=1	pfDataSrc[5]=4	pfDataSrc[8]=7

1-4-9-6. Команда вычисления суммарного количества данных WSUM_TAB

(1) Обзор команды

Вычисление суммарного количества табличных данных.

Команда вычисления суммарного количества данных [WSUM_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
WSUM_TAB	Команда вычисления суммарного количества данных		XDONE:=WSUM_TAB(pfDataSrc:= , uiSize:= , fDataOut=>);

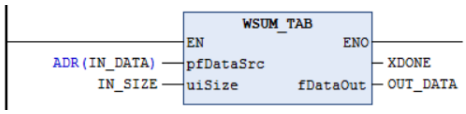
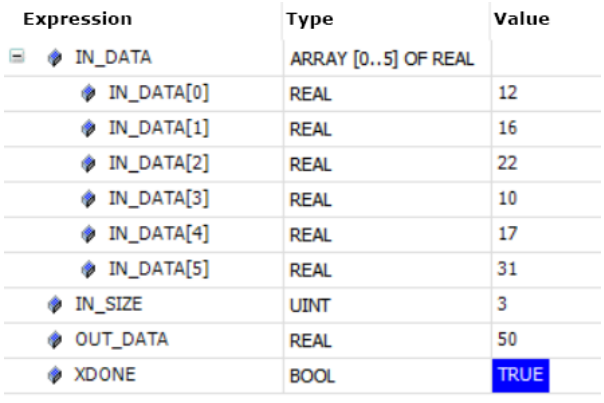
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
pfDataSrc	Исходные данные	POINTER TO REAL	-	-	Адрес входных данных
uiSize	Длина данных	UINT	1-1024	0	Длина данных
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
fDataOut	Выходные данные	REAL	В качестве типа данных	-	Суммарное значение на выходе

(3) Функция

- Вычисление общей длины данных uiSize во входных данных pfDataSrc и вывод результатов в fDataOut.

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR IN_DATA: ARRAY[0..5] OF REAL:= [12,16,22,10,17,31]; IN_SIZE: UINT:= 3; OUT_DATA: REAL; XDONE: BOOL; END_VAR
Пример программы		XDONE:= WSUM_TAB(pfDataSrc:= ADR(IN_DATA), uiSize:= IN_SIZE, fDataOut=> OUT_DATA);
Результат операции		

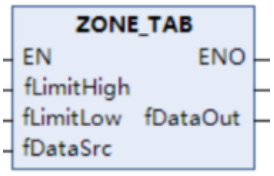
Expression	Type	Value
IN_DATA	ARRAY [0..5] OF REAL	
IN_DATA[0]	REAL	12
IN_DATA[1]	REAL	16
IN_DATA[2]	REAL	22
IN_DATA[3]	REAL	10
IN_DATA[4]	REAL	17
IN_DATA[5]	REAL	31
IN_SIZE	UINT	3
OUT_DATA	REAL	50
XDONE	BOOL	TRUE

1-4-9-7. Команда контроля зоны ZONE_TAB

(1) Обзор команды

Контроль выходного значения с заданным значением отклонения в зависимости от того, является ли входное значение положительным или отрицательным (прибавьте значение отклонения к входному значению).

Команда контроля зоны [ZONE_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
ZONE_TAB	Команда контроля зоны		XDONE:=ZONE_TAB(fLimitHigh:= , fLimitLow:= , fDataSrc:= , fDataOut=>);

(2) Связанные переменные

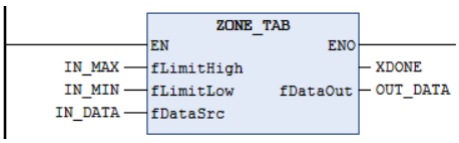
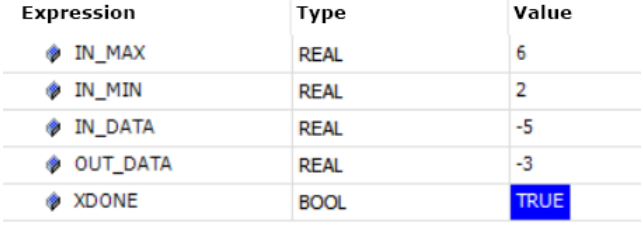
VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
fLimitHigh	Положительное отклонение	REAL	В качестве типа данных	0	Положительное отклонение, значение отклонения добавляется к входному значению
fLimitLow	Отрицательное отклонение	REAL	В качестве типа данных	0	Отрицательное отклонение, значение отклонения добавляется к входному значению

fDataSrc	Входное значение	REAL	В качестве типа данных	0	Исходные данные
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
fDataOut	Выходное значение	REAL	В качестве типа данных	-	Выходное значение

(3) Функция

- Определение знака входного значения fDataSrc, прибавление fLimitHigh или fLimitLow и сохранение результата выполнения в fDataOut.
- Когда fDataSrc<0, fDataOut = fDataSrc + fLimitLow;
- Когда fDataSrc>0, fDataOut = fDataSrc + fLimitHigh;
- Когда fDataSrc=0, fDataOut = 0.

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR IN_MAX: REAL:= 6; IN_MIN: REAL:= 2; IN_DATA: REAL; OUT_DATA: REAL; XDONE: BOOL; END_VAR
Пример программы		XDONE:= ZONE_TAB(fLimitHigh:= IN_MAX, fLimitLow:= IN_MIN, fDataSrc:= IN_DATA, fDataOut=> OUT_DATA);
Результат операции		

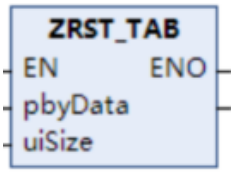
Expression	Type	Value
IN_MAX	REAL	6
IN_MIN	REAL	2
IN_DATA	REAL	-5
OUT_DATA	REAL	-3
XDONE	BOOL	TRUE

1-4-9-8. Команда пакетного сброса данных ZRST_TAB

(1) Обзор команды

Пакетный сброс данных в таблице.

Команда пакетного сброса данных [ZRST_TAB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
ZRST_TAB	Команда пакетного сброса данных		`XDONE:=ZRST_TAB(pbyData:= , uiSize:= );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
pbyData	Исходные данные	POINTER TO BYTE	-	-	Адрес заголовка данных
uiSize	Длина данных	UINT	1-1024	0	Длина данных

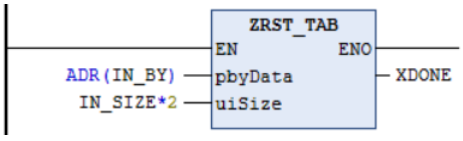
(3) Функция

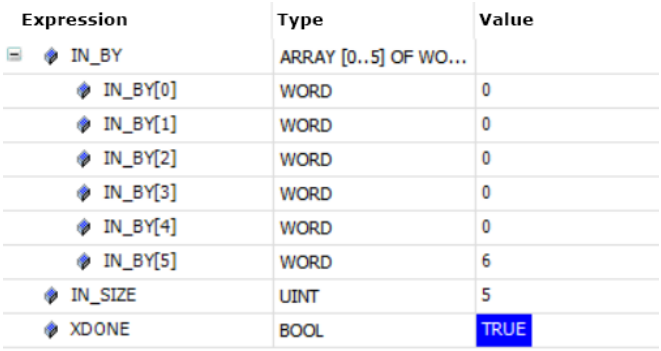
- Пакетный сброс входных исходных данных заданной длины.

(4) Пример программы

ДЕМО	LD	ST

Задание переменных		`VAR`
		`  IN_BY: ARRAY[0..5] OF WORD:= [9,5,3,2,4,6];`
		`  IN_SIZE: UINT:= 5;`
		`  XDONE: BOOL;`
		`END_VAR`

Пример программы		`XDONE:= ZRST_TAB(`
		`  pbyData:= ADR(IN_BY),`
		`  uiSize:= IN_SIZE*2);`

Результат операции	

Expression	Type	Value
IN_BY	ARRAY [0..5] OF WO...	
IN_BY[0]	WORD	0
IN_BY[1]	WORD	0
IN_BY[2]	WORD	0
IN_BY[3]	WORD	0
IN_BY[4]	WORD	0
IN_BY[5]	WORD	6
IN_SIZE	UINT	5
XDONE	BOOL	TRUE

1-5. Рабочие команды

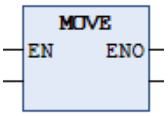
1-5-1. Команда присвоения

1-5-1-1. Команда присвоения MOVE

(1) Обзор команды

Функция: Присвоение значения константы или переменной другой переменной.

Команда присвоения [MOVE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

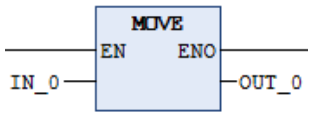
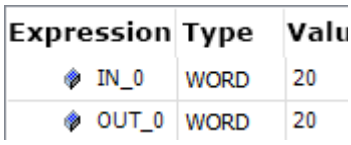
Команда	Название	Графическое представление	ST
MOVE	Команда присвоения		MOVE

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Передача данных	BOOL/BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Переменные для передачи данных

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
OUT	Прием данных	BOOL/BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Переменные для приема данных

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_0 :WORD:=20; OUT_0:WORD; END_VAR	
Пример программы		1 OUT_0:=MOVE(IN_0);
Результат операции		

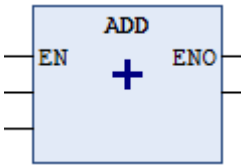
1-5-2. Команды арифметических операций

1-5-2-1. Операция сложения ADD

(1) Обзор команды

Функция: Команда операции сложения, сложение двух (или более) переменных или констант. Две переменные времени также можно сложить, получив еще одну переменную времени.

Операция сложения [ADD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

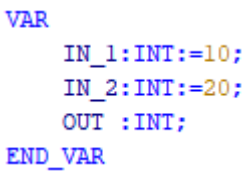
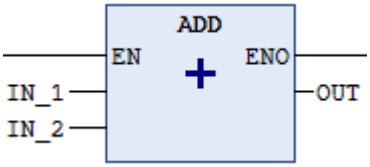
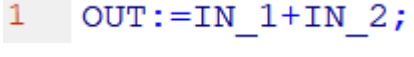
Команда	Название	Графическое представление	ST
ADD	Операция сложения		+

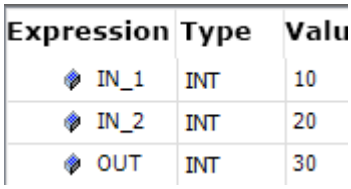
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN1	Слагаемое 1	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME	В качестве типа данных	-	Слагаемое 1
IN2	Слагаемое 2	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME	В качестве типа данных	-	Слагаемое 2
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Сумма	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME	В качестве типа данных	-	Сумма

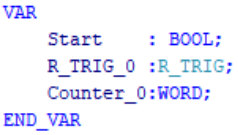
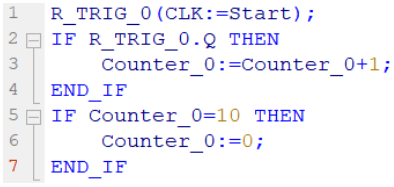
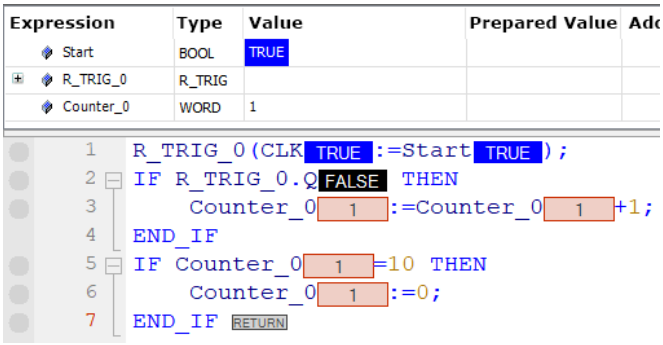
(3) Пример программы

1. Сложение значений двух переменных INT:

ДЕМО	LD	ST
Задание переменных		
Пример программы		

Результат операции	

2. Используя язык программирования ST, функциональный блок триггера нарастающего фронта накапливает слагаемое Counter_0. Когда число достигает 10, кумулятивная переменная сбрасывается до нуля.

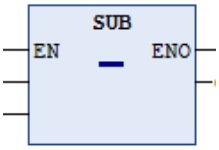
ДЕМО	ST
Задание переменных	
Пример программы	
Результат операции	

1-5-2-2. Операция вычитания SUB

(1) Обзор команды

Функция: Команда операции вычитания, вычитание двух переменных или констант.

Операция вычитания [SUB]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
SUB	Операция вычитания		-

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN1	Уменьшаемое	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/TOD	В качестве типа данных	-	Уменьшаемое
IN2	Вычитаемое	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/TOD	В качестве типа данных	-	Вычитаемое
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Разность	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME	В качестве типа данных	-	Разность

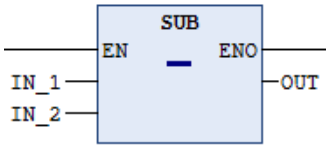
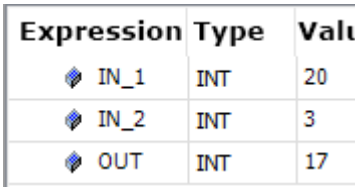
(3) Примечание

Переменные типа TOD также могут использовать функцию вычитания, когда два типа TOD вычитаются для получения новых данных типа TIME.

Пример: TOD#20:10:0- TOD#20:9:59=T#1c, но результат времени не может быть отрицательным.

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR IN_1:INT:=20; IN_2:INT:=3; OUT :INT; END_VAR

Пример программы		1 OUT:=IN_1-IN_2;
Результат операции		

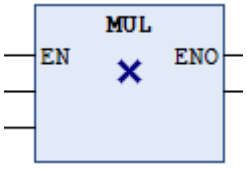
Expression	Type	Valu
IN_1	INT	20
IN_2	INT	3
OUT	INT	17

1-5-2-3. Операция умножения MUL

(1) Обзор команды

Функция: Команда операции умножения, перемножение двух (или более) переменных или констант.

Операция умножения [MUL]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

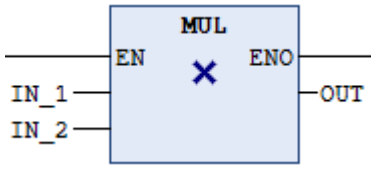
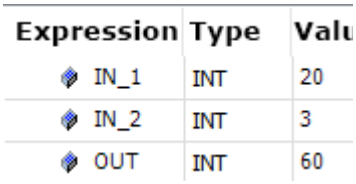
Команда	Название	Графическое представление	ST
MUL	Операция умножения		*

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN1	Множитель 1	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/TOD	В качестве типа данных	-	Множитель 1
IN2	Множитель 2	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/	В качестве типа данных	-	Множитель 2

		SINT/INT/DINT/LINT/REAL/LREAL/TIME/TOD			
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Произведение	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/TOD	В качестве типа данных	-	Произведение

(3) Пример программы

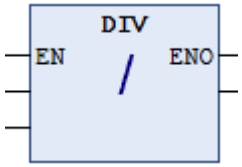
ДЕМО	LD	ST
Задание переменных	VAR IN_1:INT:=20; IN_2:INT:=3; OUT :INT; END_VAR	
Пример программы		1 OUT:=IN_1*IN_2;
Результат операции		

1-5-2-4. Операция деления DIV

(1) Обзор команды

Функция: Команда операции деления, деление двух переменных или констант.

Операция деления [DIV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
DIV	Операция деления		/

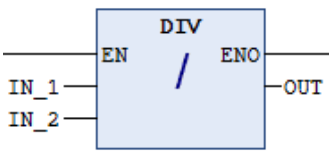
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN1	Делимое	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Делимое
IN2	Делитель	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Делитель
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Частное	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Частное

(3) Примечание

Если команда DIV применяется в проектировании, то можно использовать CheckDivByte, CheckDivWord, CheckDivDWord, and CheckDivReal для проверки того, равен ли делитель нулю. Это исключит ситуацию, когда делители равны нулю.

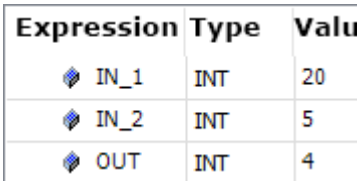
(4) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		

```
VAR
    IN_1:INT:=20;
    IN_2:INT:=5;
    OUT :INT;
END_VAR
```



```
1   OUT:=IN_1/IN_2;
```

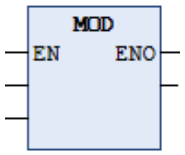
Результат операции	

1-5-2-5. Операция вычисления остатка MOD

(1) Обзор команды

Функция: Деление переменных или констант, чтобы получить остаток. Результат — это остаток после деления двух чисел, который представляет собой целочисленные данные.

Операция вычисления остатка [MOD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

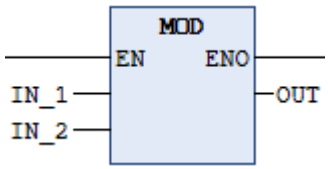
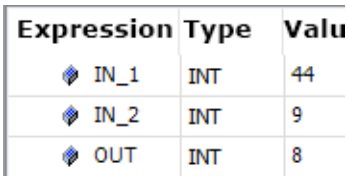
Команда	Название	Графическое представление	ST
MOD	Операция вычисления остатка		MOD

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN1	Делимое	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Делимое
IN2	Делитель	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Делитель

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
OUT	Остаток	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Остаток

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_1:INT:=44; IN_2:INT:=9; OUT :INT; END_VAR	
Пример программы		1 OUT:=IN_1 MOD IN_2;
Результат операции		

Expression	Type	Valu
IN_1	INT	44
IN_2	INT	9
OUT	INT	8

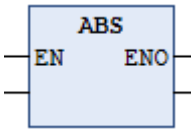
1-5-3. Команды математических операций

1-5-3-1. Абсолютное значение ABS

(1) Обзор команды

Функция: Эта функциональная команда позволяет вычислить абсолютное значение числа. Это никак не связано со знаком положительных и отрицательных чисел.

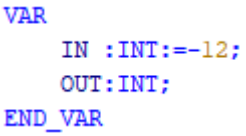
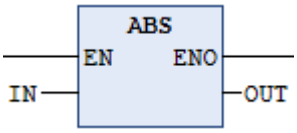
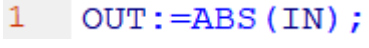
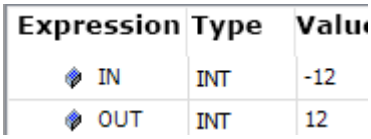
Абсолютное значение [ABS]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
ABS	Абсолютное значение		ABS

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входные переменные	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Параметры, принимаемые за абсолютные значения
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Выходное значение

(3) Пример программы

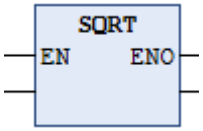
ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

1-5-3-2. Квадратный корень SQRT

(1) Обзор команды

Функция: Квадратный корень из неотрицательных действительных чисел.

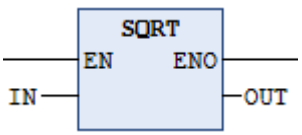
Квадратный корень [SQRT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

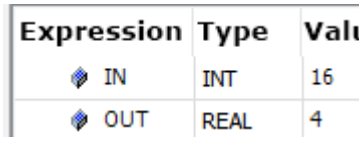
Команда	Название	Графическое представление	ST
SQRT	Квадратный корень		SQRT

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Результаты вычислений	REAL/LREAL	В качестве типа данных	-	Результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR IN :INT:=16; OUT:REAL; END_VAR
Пример программы		1 OUT:=SQRT(IN);

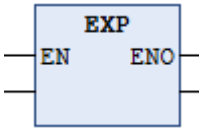
Результат операции	

1-5-3-3. Экспонента EXP

(1) Обзор команды

Функция: возвращает степень числа e (основание натурального логарифма). e — константа, равная 2,71828.

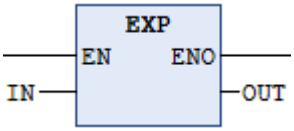
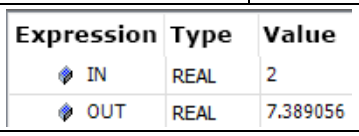
Экспонента [EXP]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
EXP	Экспонента		EXP

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT /INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=2; OUT:REAL; END_VAR	
Пример программы		1 OUT:=EXP(IN);
Результат операции		

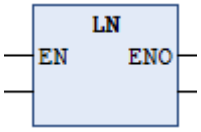
Expression	Type	Value
IN	REAL	2
OUT	REAL	7.389056

1-5-3-4. Натуральный логарифм LN

(1) Обзор команды

Функция: возвращает натуральный логарифм числа. Основанием натурального логарифма является постоянный член e (2,71828182845904).

Натуральный логарифм [LN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

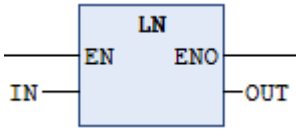
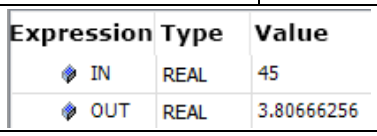
Команда	Название	Графическое представление	ST
LN	Натуральный логарифм		LN

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT	В качестве типа данных	-	Вычисляемая

		/INT/DINT/LINT/REAL/LREAL			переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=45; OUT:REAL; END_VAR	
Пример программы		1 OUT:=LN(IN);
Результат операции		

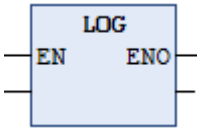
Expression	Type	Value
IN	REAL	45
OUT	REAL	3.80666256

1-5-3-5. Логарифм с основанием 10 LOG

(1) Обзор команды

Функция: Возвращает логарифм с основанием 10.

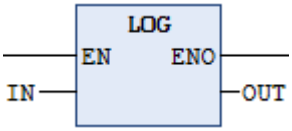
Логарифм с основанием 10 [LOG]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
LOG	Логарифм с основанием 10		LOG

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=314.5; OUT:REAL; END_VAR	
Пример программы		1 OUT:=LOG(IN);
Результат операции		

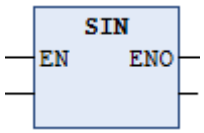
1-5-3-6. Синусоидальная функция SIN

(1) Обзор команды

Функция: Синусоидальная функция.

Синусоидальная функция [SIN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

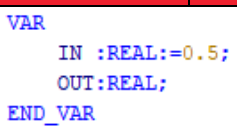
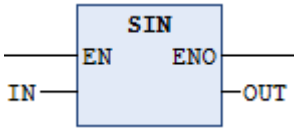
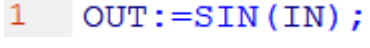
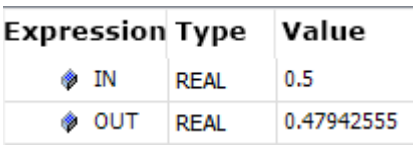
Библиотека	-	

Команда	Название	Графическое представление	ST
SIN	Синусоидальная функция		SIN

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT/ INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

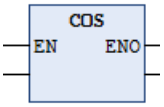
ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

1-5-3-7. Косинусоидная функция COS

(1) Обзор команды

Функция: Косинусоидная функция.

Косинусоидная функция [COS]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

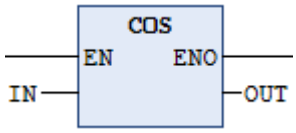
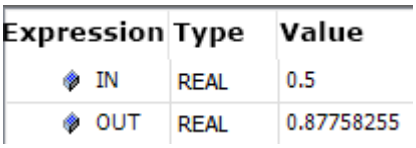
Команда	Название	Графическое представление	ST
COS	Косинусоидная функция		COS

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=0.5; OUT:REAL; END_VAR	

Пример программы		1 OUT:=COS(IN);
Результат операции		

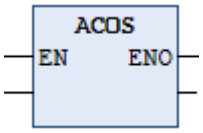
Expression	Type	Value
IN	REAL	0.5
OUT	REAL	0.87758255

1-5-3-8. Функция арккосинуса ACOS

(1) Обзор команды

Радиан косинуса (функция арккосинуса).

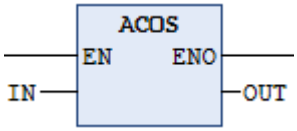
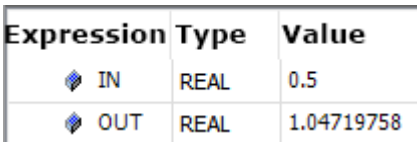
Функция арккосинуса [ACOS]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
ACOS	Функция арккосинуса		ACOS

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT /INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=0.5; OUT:REAL; END_VAR	
Пример программы		1 OUT:=ACOS(IN);
Результат операции		

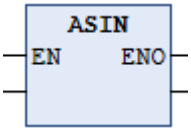
Expression	Type	Value
IN	REAL	0.5
OUT	REAL	1.04719758

1-5-3-9. Функция арксинуса ASIN

(1) Обзор команды

Функция: Радиан синуса (функция арксинуса).

Функция арксинуса [ASIN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

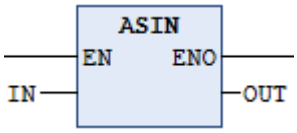
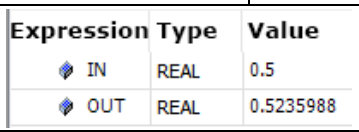
Команда	Название	Графическое представление	ST
ASIN	Функция арксинуса		ASIN

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT	В качестве типа данных	-	Вычисляемая переменна

		/INT/DINT/LINT/REAL/LREAL			я
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=0.5; OUT:REAL; END_VAR	
Пример программы		1 OUT:=ASIN(IN);
Результат операции		

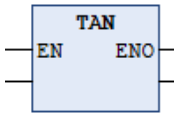
Expression	Type	Value
IN	REAL	0.5
OUT	REAL	0.5235988

1-5-3-10. Функция тангенса TAN

(1) Обзор команды

Функция: Функция тангенса.

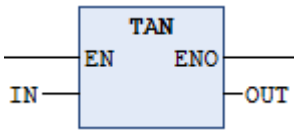
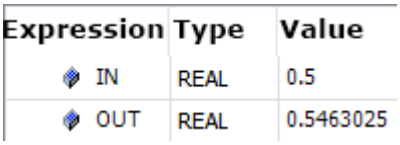
Функция тангенса [TAN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
TAN	Функция тангенса		TAN

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/SINT /INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

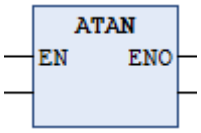
ДЕМО	LD	ST
Задание переменных	VAR IN :REAL:=0.5; OUT:REAL; END_VAR	
Пример программы		1 OUT:=TAN(IN);
Результат операции		

1-5-3-11. Функция арктангенса ATAN

(1) Обзор команды

Функция: Радиан тангенса (функция арктангенса).

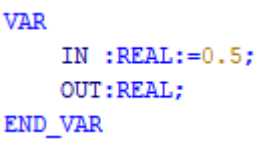
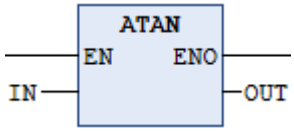
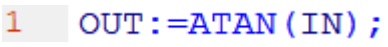
Функция арктангенса [ATAN]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
ATAN	Функция арктангенса		ATAN

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Входной параметр	BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL	В качестве типа данных	-	Вычисляемая переменная
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Выходное значение	REAL/LREAL	В качестве типа данных	-	Выходной результат

(3) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

1-5-3-12. Преобразование углов с плавающей точкой в радианы RAD/DEG

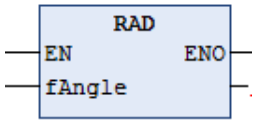
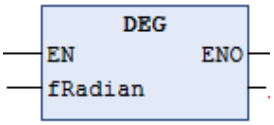
(1) Обзор команды

RAD: Операция преобразования углов с плавающей точкой в радианы. Формула расчета: [единица радиан=единица угла ×л /180].

DEG: Операция преобразования чисел с плавающей точкой из радиан в углы.

Формула расчета: [единица угла =единица радиан ×л /180].

Преобразование углов с плавающей точкой в радианы [RAD/DEG]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
RAD	Преобразование углов в радианы		RAD(fAngle:=);
DEG	Преобразование радиан в углы		DEG(fRadian:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
fAngle	Входной угол	LREAL	В качестве типа данных	0	Входной угол
	Выходной радиан	LREAL	В качестве типа данных	0	Выходной радиан
fRadian	Входной радиан	LREAL	В качестве типа данных	0	Входной радиан
	Выходной угол	LREAL	В качестве типа данных	0	Выходной угол

(3) Функция

- RAD: Операция преобразования углов с плавающей точкой в радианы. Формула расчета: [единица радиан=единица угла ×л /180].
- DEG: Операция преобразования чисел с плавающей точкой из радиан в углы.

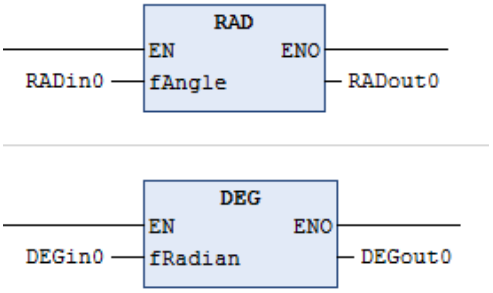
Формула расчета: [единица угла =единица радиан ×л /180].

(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    RADout0: LREAL;
    RADin0: LREAL;
    DEGin0: LREAL;
    DEGout0: LREAL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
RADout0:=RAD(fAngle:= RADin0);
DEGout0:=DEG(fRadian:= DEGin0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
RADout0	LREAL	3.1415927410125732
RADin0	LREAL	180
DEGin0	LREAL	3.14
DEGout0	LREAL	179.90874266466164

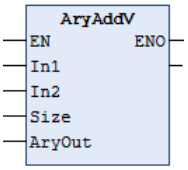
</td></tr>
</table>

1-5-3-13. Прибавление к элементу массива AryAddV

(1) Обзор команды

Прибавление того же значения к каждому элементу массива.

Прибавление к элементу массива [AryAddV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryAddV	Прибавление к элементу массива		AryAddV(In1:= , In2:= , Size:= , AryOut:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In1[]	Массив с аддитивной компоновкой	USINT/UINT/UDINT/ ULINT/SINT/INT/ DINT/LINT	В качестве типа данных	-	Массив с аддитивной компоновкой
In2	Аддитивное значение	USINT/UINT/UDINT/ ULINT/SINT/INT/ DINT/LINT	В качестве типа данных	-	Аддитивное значение
Size	Количество прибавляемых элементов In1[]	UINT	В качестве типа данных	1	Количество прибавляемых элементов In1[]
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
AryOut	Добавление массива с результатами	USINT/UINT/UDINT/ ULINT/SINT/INT/ DINT/LINT	В качестве типа данных	-	Добавление массива с результатами

(3) Функция

- Прибавление аддитивного значения "In2" к элементам "Size", начиная с In1[0] в аддитивном массиве In1[], и его вывод в аддитивный массив с результатами AryOut[].

(4) Примечание

- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в In1[] или AryOut[].

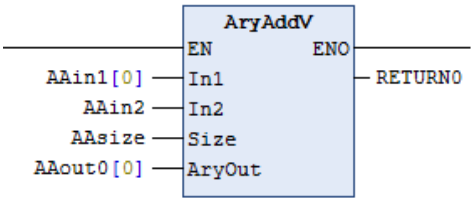
- Типы данных для "In[]", "In2" и "AryOut" должны совпадать. В противном случае, функция выполнения возвращает значение FALSE, а значение AryOut[] остается неизменным.
- Если сумма выходит за пределы допустимого диапазона AryOut[], то элемент AryOut[] имеет неверное значение. При этом никаких отклонений не происходит. Кроме того, это не повредит прилегающую зону хранения элемента.
- Если значение "Size" равно 0, значение AryOut[] остается неизменным, а функция возвращает значение TRUE.
- In1, In2, AryOut[] поддерживают целые или действительные числа, в противном случае функция выполнения возвращает FALSE.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    AAin1:ARRAY[0..5] OF INT;
    AAsize: UINT;
    AAin2: INT;
    AAout0:ARRAY[0..5] OF INT;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
AryAddV(In1:=AAin1[0] ,
    In2:= AAin2,
    Size:= AAsize ,
    AryOut:=AAout0[0] );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
AAin1	ARRAY [0..5] OF INT	
AAin1[0]	INT	11
AAin1[1]	INT	22
AAin1[2]	INT	33
AAin1[3]	INT	44
AAin1[4]	INT	55
AAin1[5]	INT	66
AAsize	UINT	4
AAin2	INT	12
AAout0	ARRAY [0..5] OF INT	
AAout0[0]	INT	23
AAout0[1]	INT	34
AAout0[2]	INT	45
AAout0[3]	INT	56
AAout0[4]	INT	0
AAout0[5]	INT	0
RETURN0	BOOL	TRUE

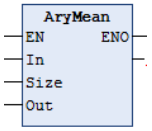
</td></tr>
</table>

1-5-3-14. Вычисление среднего значения элементов массива AryMean

(1) Обзор команды

Вычисление среднего значения элементов массива.

Вычисление среднего значения элементов массива [AryMean]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
AryMean	Вычисление среднего значения элементов массива		AryMean(In:= , Size:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In[]	Массив арифметических объектов	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL/TIME	В качестве типа данных	-	Массив арифметических объектов
Size	Количество элементов In[]	UINT	В качестве типа данных	-	Количество элементов In[]
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат операции	USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/ LINT/REAL/LREAL/TIME	В качестве типа данных	-	Результат операции

(3) Функция

- Вычисление среднего значения элементов массива.

(4) Примечание

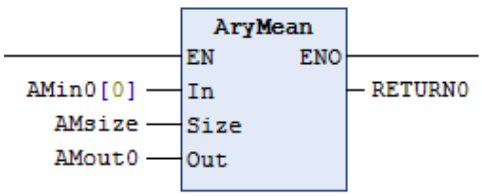
- Четыре арифметические операции не поддерживаются, если значение In[] равно +∞/-∞ и не является числом.
- Если типы данных In[] и "Out" разные, функция возвращает значение FALSE, а "Out" остается неизменным.
- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в In[].
- Если "In" и "Out" — целые числа, то число после десятичной точки среднего округляется.
- Если значение "Size" равно 0, значение "Out" будет равно 0, а функция возвращает значение True.
- Для In[], "Out" необходимо использовать целые числа, действительные числа и типы TIME, в противном случае функция выполнения возвращает значение FALSE, а "Out" остается неизменным.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    AMin0:ARRAY[0..5] OF INT;
    AMsize: UINT;
    AMout0: INT;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
AryMean(In:= AMin0[0],
    Size:= AMsize,
    Out:= AMout0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
AMin0	ARRAY [0..5] OF INT	
AMin0[0]	INT	12
AMin0[1]	INT	22
AMin0[2]	INT	32
AMin0[3]	INT	42
AMin0[4]	INT	0
AMin0[5]	INT	0
AMsize	UINT	4
AMout0	INT	27
RETURN0	BOOL	TRUE

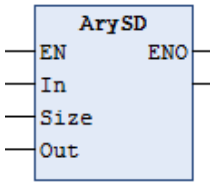
</td></tr>
</table>

1-5-3-15. Стандартное отклонение элементов массива ArySD

(1) Обзор команды

Вычисление стандартного отклонения упорядоченных элементов.

Стандартное отклонение элементов массива [ArySD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ArySD	Стандартное отклонение элементов массива		```ArySD(In:= , Size:= , Out:=);```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In[]	Массив арифметических объектов	REAL/LREAL	В качестве типа данных	-	Массив арифметических объектов
Size	Количество элементов In[]	UINT	В качестве типа данных		Количество элементов In[]
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	стандартное отклонение	REAL/LREAL	В качестве типа данных	-	стандартное отклонение

(3) Функция

- Вычисление стандартного отклонения элементов "Size", начиная с In[0] в массиве объектов операции In[].

i: Суффикс In[] (0~"Size" -1);

LnM: Среднее значение In[0]~In["Size" -1].

(4) Примечание

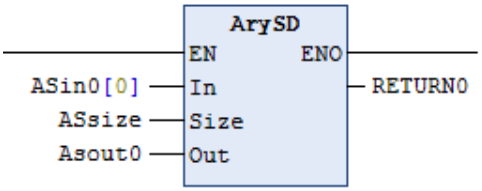
- Если значение "Size" равно 0 или 1, значение "Out" будет равно 0, а функция возвращает значение FALSE.
- Если во время выполнения операции промежуточное значение выходит за пределы допустимого диапазона In[], значение "Out" является недопустимым. При этом никаких отклонений не происходит.
- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в In[].
- Типы данных элементов In[] и Out должны совпадать, в противном случае функция выполнения возвращает значение FALSE, а Out останется неизменным.
- In[] поддерживает типы действительных чисел, в противном случае программа выдаст ошибки при компиляции.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    ASin0:ARRAY[0..5] OF REAL;
    ASsize: UINT;
    Asout0: REAL;
    RETURN0: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
ArySD(In:= ASin0[0],
    Size:= ASsize,
    Out:= ASout0[0] );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
ASin0	ARRAY [0..5] OF REAL	
ASin0[0]	REAL	5
ASin0[1]	REAL	6
ASin0[2]	REAL	7
ASin0[3]	REAL	8
ASin0[4]	REAL	9
ASin0[5]	REAL	10
ASsize	UINT	6
Asout0	REAL	1.87082875
RETURN0	BOOL	TRUE

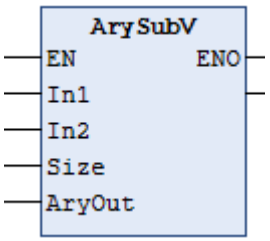
</td></tr>
</table>

1-5-3-16. Вычитание значений из элементов массива ArySubV

(1) Обзор команды

Вычитание одного и того же значения из каждого элемента массива.

Вычитание значений из элементов массива [ArySubV]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ArySubV	Вычитание значений из элементов массива		```ArySubV(In1:= , In2:= , Size:= , AryOut:=);```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In1[]	Массив после вычитания	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	-	Массив после вычитания
In2	Вычитаемое значение	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	-	Вычитаемое значение
Size	Количество элементов, вычитаемых из In1[]	UINT	В качестве типа данных	1	Количество элементов, вычитаемых из In1[]

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
AryOut	Массив с результатами вычитания	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	-	Массив с результатами вычитания

(3) Функция

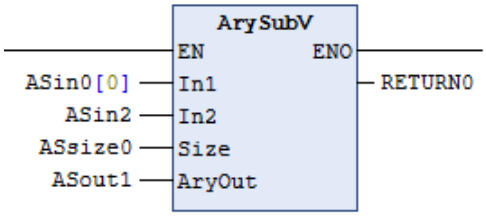
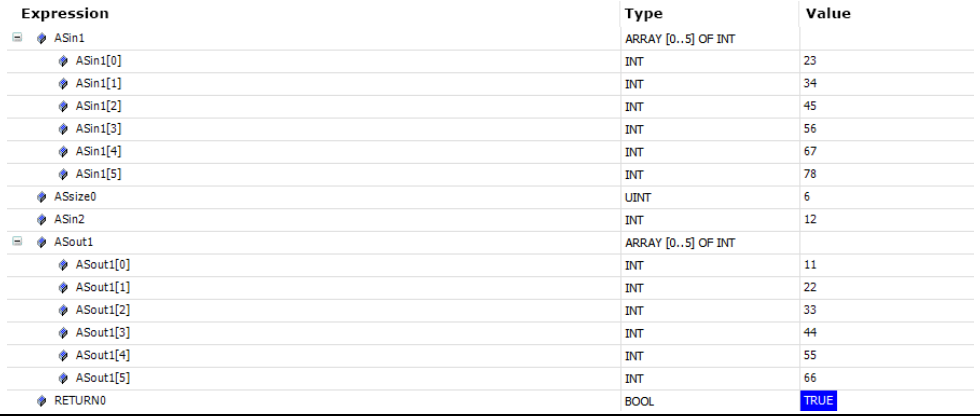
- Вычитание "In2" из элементов "Size", начиная с In1[0] в вычитаемой группе In1[], и его вывод в массив с результатами вычитания AryOut[].

(4) Примечание

- Во избежание исключений ПЛК значение "Size" не должно превышать область массива в In1[] или AryOut[].
- Типы данных для "In[]", "In2" и "AryOut" должны совпадать. В противном случае, функция выполнения возвращает значение FALSE, а значение AryOut[] остается неизменным.
- Если результат вычитания выходит за пределы допустимого диапазона AryOut[], то элементом AryOut[] становится неожиданное значение. При этом никаких отклонений не происходит, и повреждения в прилегающей зоне хранения элемента отсутствуют.
- Если значение "Size" равно 0, значение AryOut[] остается неизменным, а функция возвращает значение TRUE.
- In1, In2, AryOut[] поддерживают целочисленные входные данные, в противном случае программа выдаст ошибки при компиляции.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR ASin1:ARRAY[0..5] OF INT; ASsize0: UINT; ASin2: INT; ASout1:ARRAY[0..5] OF INT; RETURN0: BOOL; END_VAR	

Пример программы		``` ArySubV(In1:= ASin1[0], In2:= ASin2[0], Size:=ASsize0 , AryOut:= ASout1[0]); ```
Результат операции		

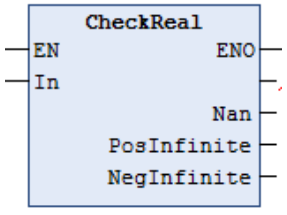
Expression	Type	Value
ASin1	ARRAY [0..5] OF INT	
ASin1[0]	INT	23
ASin1[1]	INT	34
ASin1[2]	INT	45
ASin1[3]	INT	56
ASin1[4]	INT	67
ASin1[5]	INT	78
ASsize0	UINT	6
ASin2	INT	12
ASout1	ARRAY [0..5] OF INT	
ASout1[0]	INT	11
ASout1[1]	INT	22
ASout1[2]	INT	33
ASout1[3]	INT	44
ASout1[4]	INT	55
ASout1[5]	INT	66
RETURN0	BOOL	TRUE

1-5-3-17. Проверка действительных чисел CheckReal

(1) Обзор команды

Определите, является ли входное значение нечисловым, с положительной бесконечностью, отрицательной бесконечностью.

Проверка действительных чисел [CheckReal]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
CheckReal	Проверка действительных чисел		``` CheckReal(In:= , Nan=> , PosInfinite=> , NegInfinite=>); ```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Действительное число	REAL/LREAL	В качестве типа данных	-	Действительное число
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Nan	Суждение о нечисловом значении	BOOL	[FALSE,TRUE]	FALSE	True: Нечисловое значение FALSE: Не является нечисловым значением
PosInfinite	Суждение о положительной бесконечности	BOOL	[FALSE,TRUE]	FALSE	True: Положительная бесконечность FALSE: Не положительная бесконечность
NegInfinite	Суждение об отрицательной бесконечности	BOOL	[FALSE,TRUE]	FALSE	True: Отрицательная бесконечность FALSE: Не отрицательная бесконечность

(3) Функция

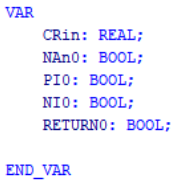
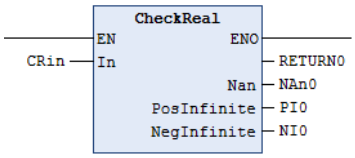
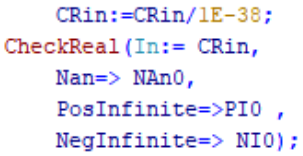
- Проверка, является ли действительное число "In" нечисловым, с положительной бесконечностью и отрицательной бесконечностью, и вывод результатов в "Nan", "PosInfinite" и "NegInfinite", соответственно.

(4) Примечание

- Если параметр, переданный в "In", имеет тип данных, отличный от действительных чисел, при компиляции возникает ошибка.
- Если результат суждения не является числовым, положительной бесконечностью или отрицательной бесконечностью, "Nan", "PosInfinite" и "NegInfinite" возвращают True, и функция возвращает True.

- Если результат суждения является нормальным значением, функция возвращает значение FALSE.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

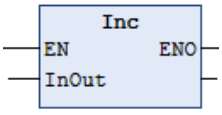
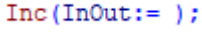
1-5-3-18. Увеличение/уменьшение Inc/Dec

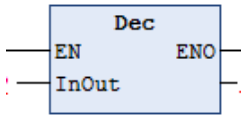
(1) Обзор команды

Inc: Увеличение целых значений.

Dec: Уменьшение целого значения.

Увеличение/уменьшение [Inc/Dec]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Inc	Увеличение		

Dec	Уменьшение		Dec(InOut:=);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Inout	Данные объекта	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT	В качестве типа данных	-	Данные объекта

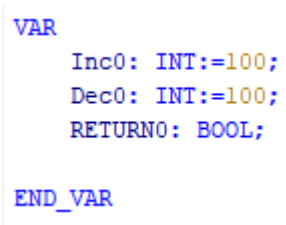
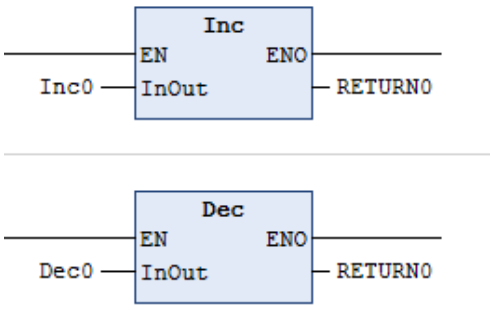
(3) Функция

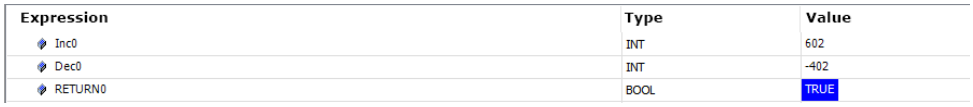
- Inc: Увеличение данных объекта "InOut". Наконец, если максимальное значение "InOut" превышено, оно возвращается к минимальному значению.
- Dec: Уменьшение данных объекта "InOut". Наконец, если минимальное значение "InOut" превышено, оно возвращается к максимальному значению.

(4) Примечание

- Если эта команда используется в программе ST, возвращаемое значение "Out" не будет использовано.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		Inc(InOut:= Inc0); Dec(InOut:= Dec0);

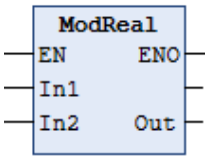
Результат операции	
Анализ результатов	В процессе выполнения команда Inc будет продолжать увеличение, а команда Dec будет продолжать уменьшение.

1-5-3-19. Фактический остаток ModReal

(1) Обзор команды

Вычислите остаток, полученный при делении делимого "In1" на делитель "In2".

Фактический остаток [ModReal]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ModReal	Фактический остаток		ModReal(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In1	Делимое	REAL	В качестве типа данных	(*)	Делимое
In2	Делитель	REAL	В качестве типа данных	(*)	Делитель
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Остаток	REAL	В качестве типа данных	(*)	Остаток

(3) Функция

- Вычислите остаток, полученный при делении делимого "In1" на делитель "In2".
- Эта команда выполняется согласно следующему выражению:
- "Out"="In1" - ("In1"/"In2") * "In2", где операция деления в скобках — это целое значение, округленное в меньшую сторону.
- Примеры значений для "In1", "In2" и "Out":

Значение “In1”	Значение “In2”	Значение “Out”
8,8	3,2	2,4
8,8	-3,2	-0,8
-8,8	3,2	0,8
-8,8	-3,2	-2,4

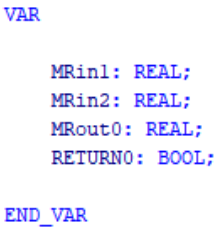
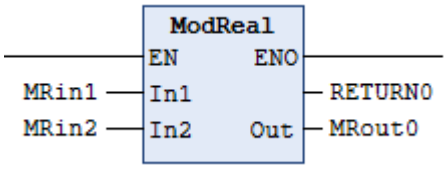
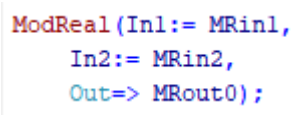
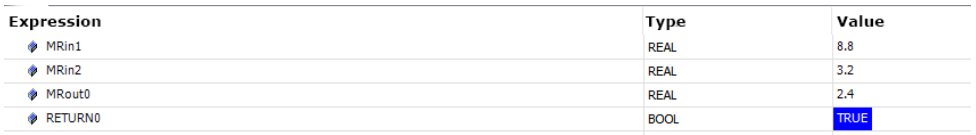
(4) Примечание

- С учетом различных комбинаций значений для "In1" и "In2", значения для "Out" приведены в таблице ниже:

0		“In1”				
		Значение	+∞	-∞	Нечисловое значение	
“In2”	0	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение
	Значение	0	Остаток от деления "In1" на "In2"	Нечисловое значение	Нечисловое значение	Нечисловое значение
	+∞	0	Значение “In1”	Нечисловое значение	Нечисловое значение	Нечисловое значение
	-∞	0	Значение “In1”	Нечисловое значение	Нечисловое значение	Нечисловое значение
	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение

- Если параметры, переданные в "In1" и "In2", являются типами данных, отличными от действительных чисел, при компиляции возникнет исключение.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR

    MRin1: REAL;
    MRin2: REAL;
    MRout0: REAL;
    RETURN0: BOOL;

END_VAR
```



```
ModReal(In1:= MRin1,
    In2:= MRin2,
    Out=> MRout0);
```

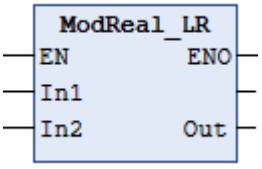
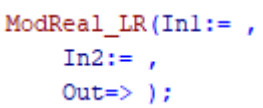
Expression	Type	Value
MRin1	REAL	8.8
MRin2	REAL	3.2
MRout0	REAL	2.4
RETURN0	BOOL	TRUE

1-5-3-20. Длинный фактический остаток ModReal_LR

(1) Обзор команды

Вычислите остаток, полученный при делении делимого "In1" на делитель "In2".

Длинный фактический остаток [ModReal_LR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ModReal_LR	Длинный фактический остаток		


```
ModReal_LR(In1:= ,
    In2:= ,
    Out=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In1	Делимое	LREAL	В качестве типа данных	(*)	Делимое
In2	Делитель	LREAL	В качестве типа данных	(*)	Делитель
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Остаток	LREAL	В качестве типа данных	(*)	Остаток

(3) Функция

- Вычислите остаток, полученный при делении делимого "In1" на делитель "In2".
- Эта команда выполняется согласно следующему выражению:
- "Out"="In1" - ("In1"/"In2") * "In2", где операция деления в скобках — это целое значение, округленное в меньшую сторону.
- Примеры значений для "In1", "In2" и "Out":

Значение “In1”	Значение “In2”	Значение “Out”
8,8	3,2	2,4
8,8	-3,2	-0,8
-8,8	3,2	0,8
-8,8	-3,2	-2,4

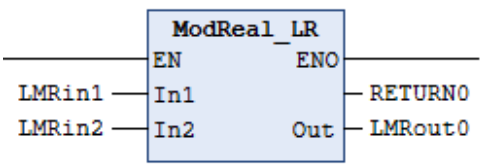
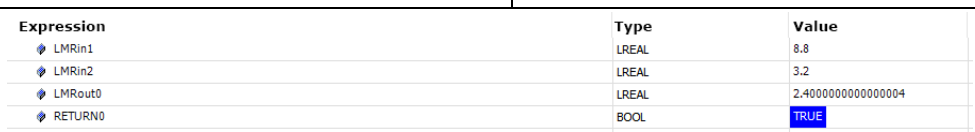
(4) Примечание

- С учетом различных комбинаций значений для "In1" и "In2", значения для "Out" приведены в таблице ниже:

0		“In1”				
		Значение	+∞	-∞	Нечисловое значение	
“In2”	0	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение
	Значение	0	Остаток от деления "In1" на "In2"	Нечисловое значение	Нечисловое значение	Нечисловое значение
	+∞	0	Значение “In1”	Нечисловое значение	Нечисловое значение	Нечисловое значение
	-∞	0	Значение “In1”	Нечисловое значение	Нечисловое значение	Нечисловое значение
	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение	Нечисловое значение

- Если параметры, переданные в "In1" и "In2", являются типами данных, отличными от действительных чисел, при компиляции возникнет исключение.

(5) Пример программы

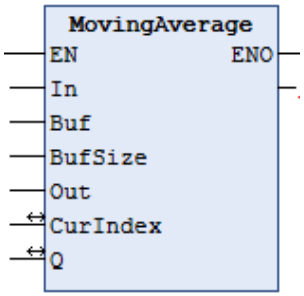
ДЕМО	LD	ST
Задание переменных		VAR LMRin1: LREAL; LMRin2: LREAL; LMRout0: LREAL; RETURN0: BOOL; END_VAR
Пример программы		ModReal_LR(In1:= LMRin1, In2:= LMRin2, Out=> LMRout0);
Результат операции		

1-5-3-21. Скользящее среднее значение MovingAverage

(1) Обзор команды

Вычисление скользящего среднего значения.

Скользящее среднее значение [MovingAverage]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
MovingAverage	Скользящее среднее значение		MovingAverage(In:= , Buf:= , BufSize:= , Out:= , CurIndex:= , Q:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Используется для вычисления среднего значения	USINT/UINT/UDINT/ ULINT/SINT/INT/ DINT/LINT/REAL/ LREAL	В качестве типа данных	-	Используется для вычисления среднего значения
Buf	Используется для сохранения массива "In"	USINT/UINT/UDINT/ ULINT/SINT/INT/ DINT/LINT/REAL/ LREAL	В качестве типа данных	1	Используется для сохранения массива "In"

BufSize	Используется для вычисления максимального количества элементов со средним значением	UINT	В качестве типа данных	-	Используется для вычисления максимального количества элементов со средним значением
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
CurIndex	Расположение Buf[] для сохранения "In"	UINT	В качестве типа данных	0	Расположение Buf[] для сохранения "In"
Q	True: Количество значений, сохраненных в Buf[], превышает "BufSize". FALSE: Количество значений, сохраненных в Buf[], меньше, чем "BufSize".	BOOL	[FALSE,TRUE]	FALSE	True: Количество значений, сохраненных в Buf[], превышает "BufSize". FALSE: Количество значений, сохраненных в Buf[], меньше, чем "BufSize".
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат операции	USINT/UINT/UDINT/ ULINT/SINT/INT/ DINT/LINT/REAL/ LREAL	В качестве типа данных	0	Результат операции

(3) Функция

- При каждом выполнении этой команды входное значение "In" будет

сохраняться в массиве сохранения входных значений Buf[]. Среднее значение сохраненных значений сохраняется в результате операции "Out". Максимальное количество элементов, используемых для вычисления среднего значения, задается в параметре "BufSize".

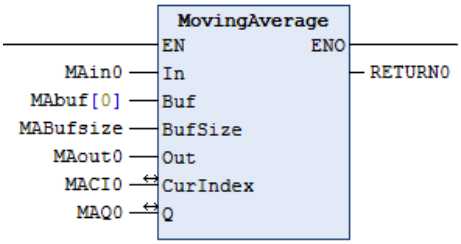
(4) Примечание

- Типы данных элементов "In", "Out" и Buf[] должны совпадать, в противном случае функция возвращает значение FALSE, а Out[] останется неизменным.
- Даже если результат операции выходит за пределы допустимого диапазона "Out", исключения не произойдет, а недопустимые значения будут сохранены в "Out".
- Если значение "BufSize" равно 0, значения "Out" и "CurIndex" будут равны 0, а значение "Q" станет True.
- После изменения значения "BufSize" параметр "CurIndex" сохранит свое текущее значение.
- Во избежание сбоев в работе ПЛК значение "BufSize" не должно превышать размер Buf[].

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

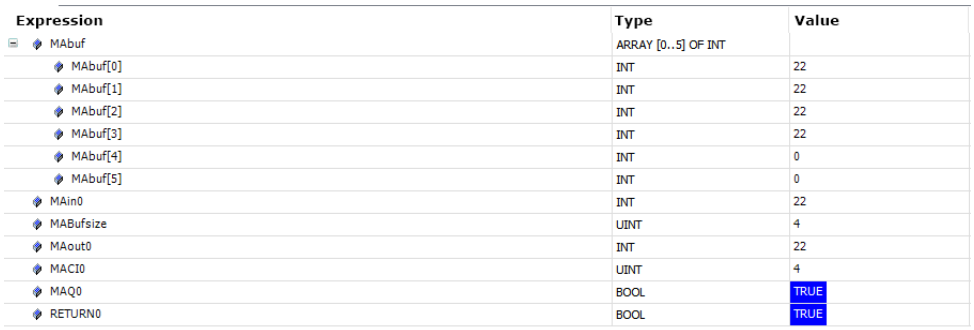
```
VAR

    MAbuf:ARRAY[0..5] OF INT;
    MAin0: INT;
    MABufsize: UINT;
    MAout0: INT;
    MACI0: UINT;
    MAQ0: BOOL;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
MovingAverage(In:= MAin0,
    Buf:= MAbuf[0],
    BufSize:= MABufsize,
    Out:= MAout0,
    CurIndex:= MACI0,
    Q:= MAQ0);
```

</td></tr>
</table>

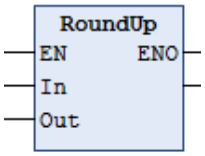
Результат операции	
Анализ результатов	При каждом выполнении команды данные MAin0 будут сохраняться в начальный массив MAbuf[0]. Выполните MABufsize в общей сложности несколько раз и сохраните данные MABufsize в MAbuf.

1-5-3-22. Перенос знака действительного числа RoundUp

(1) Обзор команды

Перенесите первый знак десятичной части действительного числа и возьмите целое число.

Перенос знака действительного числа [RoundUp]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
RoundUp	Перенос знака действительного числа		`RoundUp(In:= , Out:= );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	REAL/LREAL	В качестве типа данных	-	Объект преобразования

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Out	Результат преобразования	DINT/LINT	В качестве типа данных	-	Результат преобразования

(3) Функция

- Обработка первого десятичного знака действительного числа "In", чтобы превратить его в целое число.

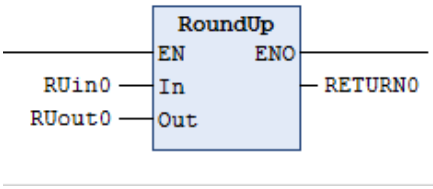
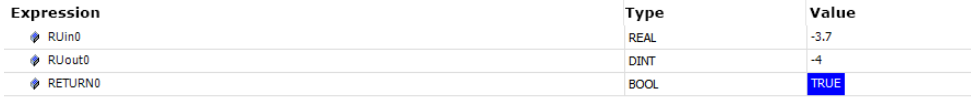
Входное значение	Выходное значение	Входное значение	Выходное значение
REAL#1.6	DINT#2	REAL#-1.4	DINT#-2
REAL#1.5	DINT#2	REAL#-2.5	DINT#-3
REAL#1.4	DINT#2	REAL#3.2	DINT#4
REAL#2.5	DINT#3	REAL#3.7	DINT#4
REAL#-1.6	DINT#-2	REAL#-3.2	DINT#-4
REAL#-1.5	DINT#-2	REAL#-3.7	DINT#-4

(4) Примечание

- Если результат преобразования выходит за пределы допустимого диапазона "Out", значение "Out" является неправильным.
- Типы данных элементов "In" and "Out" должны совпадать, например, REAL соответствует данным DINT. В противном случае, функция возвращает значение FALSE, а значение "Out" остается неизменным.
- Для "Out" необходимо использовать типы данных Dint и Lint, в противном случае функция выполнения возвращает значение FALSE, а "Out" остается неизменным.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR RUin0: REAL; RUout0: DINT; END_VAR

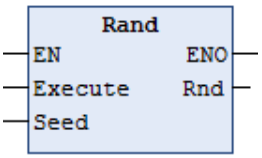
Пример программы		RoundUp(In:= RUin0, Out:=RUout0);
Результат операции		

1-5-3-23. Генерирование случайных чисел RAND

(1) Обзор команды

Генерирование псевдослучайных чисел.

Генерирование случайных чисел [RAND]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
RAND	Генерирование случайных чисел		RAND_0(Execute:= , Seed:= , Rnd=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Execute	Триггер	BOOL	[FALSE,TRUE]	FALSE	Генерирование случайных чисел по триггеру
Seed	Режим случайных чисел	UINT	В качестве типа данных	0	Задание последовательности случайных чисел 0: Не задано

VAR_OUTPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
Rnd	Случайное число	LREAL	0.00000000000000e+0～1.00000000000000e+0	-	Случайное число

(3) Функция

- Генерирование случайного числа "Rnd". Значение "Rnd" изменяется после каждого выполнения команды.
- Режим случайных чисел "Seed" используется для задания последовательности случайных чисел. Если значение параметра "Seed" одинаково, то при каждом включении питания будет генерироваться одна и та же последовательность случайных чисел. Таким образом, можно генерировать воспроизводимые случайные числа.
- Если значение параметра "Seed" равно 0, генерируется неповторяющееся случайное число. Если не нужно генерировать одну и ту же последовательность случайных чисел при каждом включении питания, установите значение параметра "Seed" на 0.

(4) Примечание

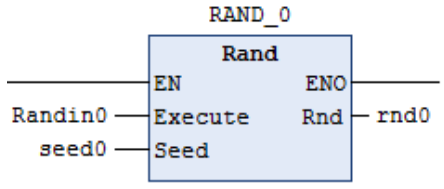
- Значение "Rnd" является действительным числом в диапазоне от 0 до 1.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    RAND_0: Rand;
    Randin0: BOOL;
    seed0: UINT;
    rnd0: LREAL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
RAND_0(Execute:= Randin0,
    Seed:= seed0,
    Rnd=> rnd0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
RAND_0	RAND	
Execute	BOOL	TRUE
Seed	UINT	0
Rnd	LREAL	0.89031982421875
Randin0	BOOL	TRUE
seed0	UINT	0
rnd0	LREAL	0.89031982421875

</td></tr>
</table>

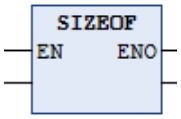
1-5-4. Команды для работы с адресами

1-5-4-1. Размер типа данных SIZEOF

(1) Обзор команды

Функция: Выполните эту функцию, чтобы определить количество байт, необходимых для заданного типа данных. Иными словами, функция должна возвратить количество байт памяти, занимаемых объектом или типом.

Размер типа данных [SIZEOF]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
SIZEOF	Размер типа данных		SIZEOF

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Целевые данные	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/LINT/ REAL/LREAL/TIME/DATE/TOD/ DT/STRING	В качестве типа данных	-	Целевые данные
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Длина данных	USINT/UINT/UDINT/ULINT	В качестве типа данных	-	Длина данных

(3) Функция

Возвращаемое значение SIZEOF — это значение без знака, а возвращаемое значение типа будет использовано для определения размера переменной IN, выходное значение OUT выражено в байтах, а IN может иметь любой тип данных.

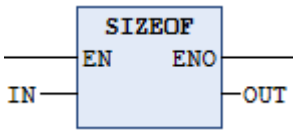
Возвращаемые данные типа SIZEOF

Возвращаемое значение SIZEOF	Неявный тип данных
0 <= размер x < 256	USINT
256 <= размер x < 65536	UINT
65536 <= размер x < 4294967296	UDINT
4294967296 <= размер x	ULINT

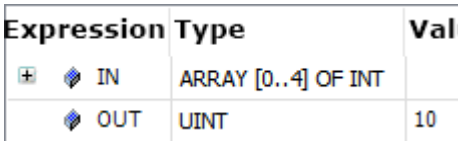
(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN :ARRAY[0..4] OF INT;
    OUT:UINT;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>
</td><td>

```
1   OUT:=SIZEOF(IN);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">
</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Программа присваивает результат переменной OUT, и конечное значение переменной OUT равно 10. Поскольку массив IN состоит из 5 целых элементов INT, единицей результата SIZEOF является BYTE, поэтому конечный результат выполнения программы составляет 10 BYTE, что указывает на то, что IN занимает 10 байт памяти.</td></tr>
</table>

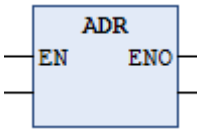
1-5-4-2. Оператор взятия адреса ADR

(1) Обзор команды

ADR: Получение адреса памяти входной переменной, присвоение результата выходной переменной, оператор, который является расширением стандарта IEC 61131-3.

^: Получение адресного содержимого входной переменной и присвоение результата выходной переменной.

Оператор взятия адреса [ADR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
ADR	Команда получения адреса		ADR
^	Команда получения адресного содержимого	-	^

(2) Связанные переменные

ADR

VAR_INOUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Целевые данные	BOOL/BYTE/WORD/DWORD/ LWORD/USINT/UINT/UDINT/ ULINT/SINT/INT/DINT/LINT/ REAL/LREAL/TIME/DATE/ TOD/DT/STRING	В качестве типа данных	-	Целевые данные
OUT	Адрес целевых данных	POINTER TO	-	-	Адрес целевых данных

^

VAR_OUTPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
OUT	Указатель на адресное содержимое	BOOL/BYTE/WORD/DWORD/LWORD/USINT/UINT/UDINT/ULINT/SINT/INT/DINT/LINT/REAL/LREAL/TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Указатель на адресное содержимое

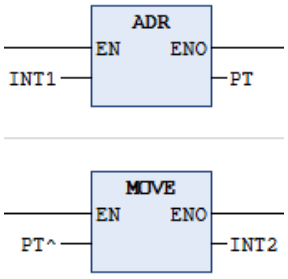
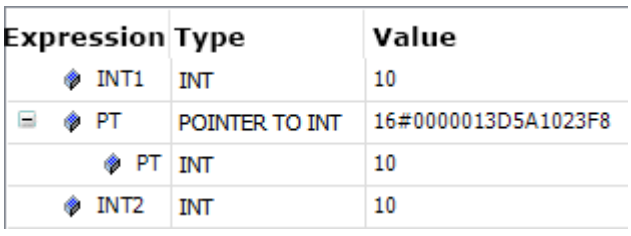
(3) Функция

Возвращаемое значение ADR — это только адрес памяти переменной. Длина данных, которые можно сохранить в этом адресе памяти, составляет 1 БАЙТ. Содержимое соответствующего адреса можно извлечь с помощью оператора содержимого "^". Чтобы получить адрес памяти переменной var_int1, присвойте ее адрес переменной-указателю, а затем извлеките конкретное содержимое в соответствующем адресе с помощью оператора "^" и присвойте его переменной var_int2. Программа реализации имеет следующий вид:

pt := ADR(var_int1);

var_int2:= pt^;

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR INT1:INT:=10; PT :POINTER TO INT; INT2:INT; END_VAR	
Пример программы		1 PT:=ADR(INT1); 2 INT2:=PT^;
Результат операции		

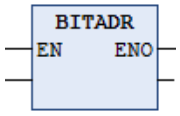
Expression	Type	Value
INT1	INT	10
PT	POINTER TO INT	16#0000013D5A1023F8
PT	INT	10
INT2	INT	10

1-5-4-3. Оператор битового адреса BITADR

(1) Обзор команды

Функция: Возврат смещения информации о битовом адресе переменной распределения.

Оператор битового адреса [BITADR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	-		

Команда	Название	Графическое представление	ST
BITADR	Оператор битового адреса		BITADR

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
IN	Целевые данные	BOOL	FALSE,TRUE	-	Целевые данные
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
OUT	Указатель на целевой адрес	DWORD	В качестве типа данных	-	Указатель на целевой адрес

(3) Функция

BITADR возвращает числовой адрес смещения бита в виде переменной типа DWORD. Обратите внимание, что значение смещения зависит от того, можно ли получить адрес типа опции из целевой системы. Максимальное значение DWORD определяет область памяти, показанную в таблице ниже:

Адреса смещения BITADR для каждого адресного пространства

Адресное пространство	Начальный адрес	Описание
Память	16x40000000	%M

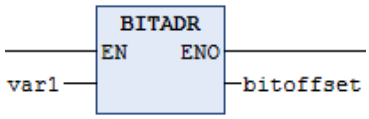
Вход	16x80000000	%I
Выход	16xC0000000	%Q

(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    var1 AT %IX2.3:BOOL;
    bitoffset     : DWORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
1  bitoffset:=BITADR(var1);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
var1	BIT	FALSE
bitoffset	DWORD	16#80000013

</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Результат выполнения — число 80000013 в шестнадцатеричном формате, при этом "2" в %IX2.3 — это 2 байта, а ". 3" — это четвертый бит, поэтому адрес равен 2*8+4=20. Переведите десятичное число 20 в шестнадцатеричное 14. Кроме того, поскольку первый адрес, соответствующий зоне I, хранится, начиная с числа 80000000, нетрудно понять, что фактический адрес, соответствующий шестнадцатеричному 14, — это 16#80000013. Схема представлена в таблице ниже:
Пример BITADR
<table>
<tr><th>Значение смещения</th><th>Содержимое данных</th></tr>
<tr><td>16#80000000</td><td></td></tr>
<tr><td>16#80000001</td><td></td></tr>
<tr><td>16#80000002</td><td></td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>16#80000011</td><td></td></tr>
<tr><td>16#80000012</td><td></td></tr>
<tr><td>16#80000013</td><td>смещение бита</td></tr>
</table>
</td></tr>
</table>

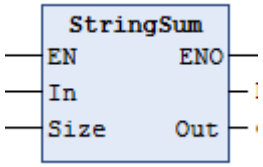
1-6. Строковые команды

1-6-1. Вычисление суммы StringSum

(1) Обзор команды

Вычисление значения SUM (сумма) строки.

Вычисление суммы [StringSum]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
StringSum	Вычисление суммы		StringSum(In:= , Size:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объектная строка	STRING	В качестве типа данных	-	Объектная строка
Size	Размер байта	USINT	1,2	1	Размер байтов в значении суммы
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Значение суммы	STRING	Количество байтов в переменной 'size'	-	Строка после преобразования значения суммы (Sum)

(3) Функция

- Вычисление значения SUM объектной строки "In" (сумма кодировок для каждого символа). Значение суммы (SUM) в "Out" — это количество байт, указанное в размере байта "Size". Переменная "Out" представлена в виде строки шестнадцатеричных чисел с символом NULL, сохраненным в конце строки.

(4) Примечание

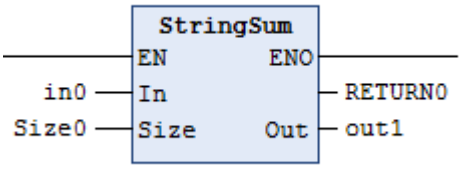
- Если суммарная кодировка символов в "In" превышает количество битов, которые можно представить в "Size", старшие биты округляются, и функция возвращает значение FALSE.
- Если значение "Size" превышает допустимый диапазон, программа сталкивается с исключением, и функция возвращает значение FALSE.
- Если количество байтов в "In" равно 0 (NULL), программа сталкивается с исключением, и функция возвращает значение FALSE.
- Если "In" не заканчивается символом NULL, это может привести к неправильным результатам.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    Size0: USINT;
    in0:STRING;
    out1: STRING(2);
    RETURN0: BOOL;


END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
StringSum(In:=in0 ,
    Size:=size0,
    Out=>out1);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
Size0	USINT	2
in0	STRING	'1234'
out1	STRING(2)	'CA'
RETURN0	BOOL	TRUE

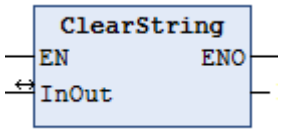
</td></tr>
<tr><td>Анализ результатов</td><td colspan="2">Символьные данные в In преобразуются в шестнадцатеричные данные, младшие биты складываются, старшие биты складываются, чтобы получить вычисленное значение. Если сумма младших битов превышает 16#F, то старший бит увеличивается на 1. Если сумма старших бит превышает 16#F, данные не отображаются; например, '1234'=31 32 33 34 =3+3+3+3 1+2+3+4=CA.</td></tr>
</table>

1-6-2. Очистка строки Clearstring

(1) Обзор команды

Очистка строки.

Очистка строки [Clearstring]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
Clearsteing	Очистка строки		`ClearString(InOut:= );`

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
InOut	Очистка строки	STRING	В качестве типа данных	-	Требуется очистка строки.

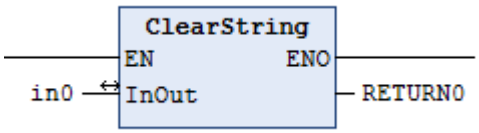
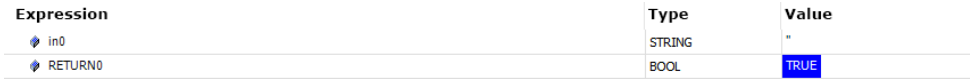
(3) Функция

- Очистка строки "InOut". Сохранение символов NULL во всех областях "InOut".

(4) Примечание

- Возвращаемое значение остается True.

(5) Пример программы

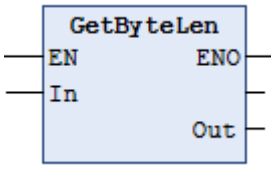
ДЕМО	LD	ST
Задание переменных	VAR in0:STRING; RETURN0: BOOL; END_VAR	
Пример программы		RETURN0:=ClearString(InOut:=string0);
Результат операции		
Анализ результатов	Значение In0 равно '1234', и строковые данные очищаются после записи.	

1-6-3. Вычисление количества байтов в строке GetByteLen

(1) Обзор команды

Вычисление количества байтов в строке.

Вычисление количества байтов в строке [GetByteLen]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
GetByteLen	Вычисление количества байтов в строке		GetByteLen(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объектная строка	STRING	В качестве типа данных	-	Объектная строка для вычисления количества байтов
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Номера байтов в строке	UINT	0~1985	-	Номера байтов

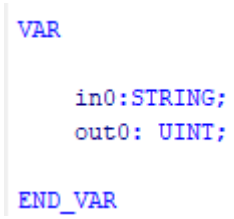
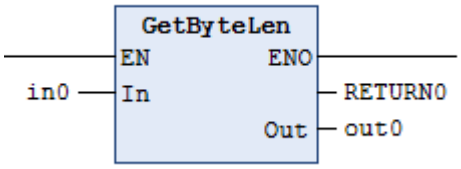
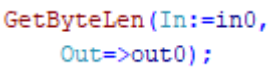
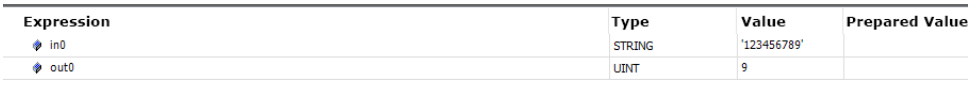
(3) Функция

- Вычисление количества байтов от начала до конца объектной строки "In". Символ NULL в конце строки не входит в указанное количество байтов.

(4) Примечание

- Длина строки ввода ограничена типом данных STRING.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR

    in0:STRING;
    out0: UINT;

END_VAR
```



```
GetByteLen(In:=in0,
Out=>out0);
```

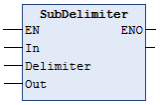
Expression	Type	Value	Prepared Value
in0	STRING	'123456789'	
out0	UINT	9	

1-6-4. Удаление разделителя после считывания строки действительных чисел SubDelimiter

(1) Обзор команды

Считывание данных, разделенных разделителями, из строки и сохранение их в виде массива действительных чисел.

Удаление разделителя после считывания строки действительных чисел [SubDelimiter]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
SubDelimiter	Удаление разделителя после считывания строки действительных чисел		SubDelimiter(In:= , Delimiter:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Строка ввода	STRING	Не более 1985 байтов	-	Считываемая строка, разделенная разделителями
Delimiter	Разделитель	_eDELIMITER	В качестве типа данных	_eDELIMITER_COMMA	Разделитель
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Сохранение места хранения массива действительных чисел	REAL	В качестве типа данных	-	Сохранение больших массивов действительных чисел преобразованных данных

(3) Функция

- Преобразование строки, разделенной разделителем "Delimiter" в строке ввода "In" в массив действительных чисел "Out" и последовательное сохранение его в качестве значений каждого элемента массива.
- Типом данных разделителя "Delimiter" является элемент перечисления _eDELIMITER. Значения элементов перечисления:

Элемент перечисления	Значение
_COMMA	‘,’ (запятая)
_TAB	‘$T’ (символ табуляции)
_SEMICOLON	‘;’ (точка с запятой)
_SPACE	‘ ’ (пробел)

- Если количество данных, разделенных разделителями в "In", превышает количество членов в "Out", лишние данные не будут учитываться.
- Если количество данных, разделенных разделителями в "In", меньше, чем количество членов в "Out", значение членов с недостаточным количеством данных остается неизменным.
- Данные, разделенные разделителями в "In", относятся к строковому типу. Формат строки для представления вещественных (действительных) данных:



Название	Формат
Знаковая часть	Игнорируя последовательные символы ' ' (пробелы) в начале строки, отдельные элементы '+' или '-' следует рассматривать в качестве символов Символ '+' можно опустить. Последовательные пробелы после этого символа будут игнорироваться.
Целая часть	Целая часть — от конца символа до передней части десятичной точки. Однако последовательные пробелы после символа не входят в целую часть. Кроме того, символы можно опускать. Если десятичная точка или десятичная часть отсутствует, то

Название	Формат
	следует рассматривать переднюю часть экспоненциальной части в качестве целой части. Если десятичная точка, десятичная часть или экспоненциальная часть отсутствует, то всю строку следует рассматривать в качестве целой части. Целая часть состоит из чисел в диапазоне от 0 до 9. Целую часть нельзя опустить. Максимальное количество цифр в целой части — это максимальная длина строки в 1985 знаков за вычетом цифр в символе, десятичной точки, десятичной части, экспоненциальной части и пробелов до и после символа.
Десятичная точка	Отдельный символ '.' (точка), за которым следует целая часть, является десятичной точкой. Если десятичная часть отсутствует, десятичную точку опускают.
Десятичная часть	Десятичная часть начинается после десятичной точки и заканчивается перед экспоненциальной частью. Если экспоненциальная часть отсутствует, то вся строка после десятичной точки является десятичной частью. Десятичная часть состоит из чисел в диапазоне от 0 до 9. Десятичную часть можно опустить. Максимальное количество десятичных знаков — 15. Если десятичная точка отсутствует, десятичной части также не существует.
Экспоненциальная часть	Экспоненциальная часть начинается с элемента 'e' , за которым следует десятичная часть и элемент '+' или '-' после нее, и продолжается до конца строки. Если десятичная часть отсутствует, то вышеупомянутая строка после десятичной точки является экспоненциальной частью. Если десятичная точка или десятичная часть отсутствует, то приведенная выше строка после целой части является экспоненциальной частью. Числовая часть экспоненты состоит из цифр от 0 до 9. Экспоненциальную часть можно опустить. Максимальное количество цифр в числовой части экспоненты — 3.

Пример правил преобразования:

Формат	STRING	REAL
Со знаком, с десятичной точкой, с десятичной частью, без экспоненциальной части	‘-123,4567’	-123,4567
Со знаком, с десятичной точкой, с десятичной частью, с экспоненциальной частью	‘+123,4567e+02’	12345.67
Без знака, с десятичной точкой, с десятичной частью, с экспоненциальной частью	‘123,4567e-02’	1.234567
Без знака, без десятичной точкой, без десятичной части, без экспоненциальной части	‘1’	1

Меры предосторожности при преобразовании:

- Если числовое значение, представленное строкой ввода, выходит за пределы допустимого диапазона данных REAL (1.0E-44~3.402823E+38), если это положительное число, то результатом преобразования будет бесконечность. Если это отрицательное число, результатом преобразования будет отрицательная бесконечность.
- Строка ввода не может содержать других символов, не являющихся строками действительных чисел, иначе при преобразовании возникнет исключение, и функция вернет значение FALSE.
- Если в любом положении в середине строки ввода есть два или более последовательных символа "_" (подчеркивание), преобразование происходит нормально.
- Если между '-' (отрицательный знак) или '+' (положительный знак) в начале строки ввода и числа стоит символ '_' (подчеркивание), преобразование происходит нормально.
- Если в начале или конце строки ввода есть символ '_' (подчеркивание), возникает исключение преобразования (функция возвращает FALSE).
- Если в любом положении в середине строки ввода есть отдельный символ '_' (подчеркивание), он будет проигнорирован.
- Если содержимое строки ввода превышает точность, с которой оно может быть представлено типом данных REAL, оно будет округлено.
- Если содержимое строки ввода ближе к 0, чем значение, которое может быть представлено типом данных REAL, то результатом преобразования будет 0.
- Если десятичная часть строки ввода превышает максимум 15 цифр,

программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

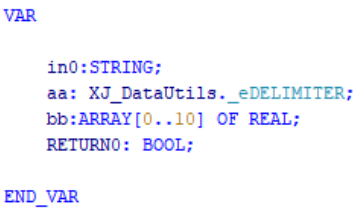
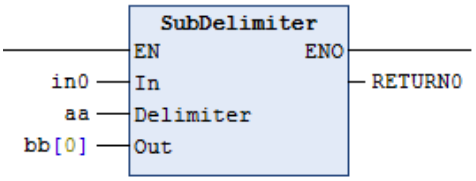
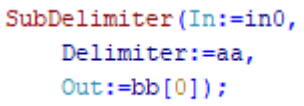
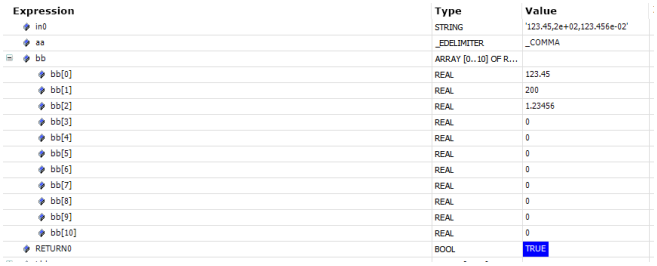
- Если экспоненциальная часть строки ввода превышает максимальное количество в 3 цифры, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Если строка ввода содержит целые числа, десятичные числа и экспоненты с номерами, отличающимися от 0~... 9, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Строка ввода не содержит десятичной части, но десятичная точка не опущена. Программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Экспоненциальная часть строки ввода не имеет числовой части, но в ней есть символы "e+"/"e -". Программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Если другие строки ввода содержат любые иррегулярные числа, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Если до и после экспоненциальной части строки ввода стоит символ пробела, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Непрерывные символы пробела в начале и конце строки ввода (в том числе до и после разделителя) будут игнорироваться, и команда будет нормально выполнена.
- Если экспоненциальная часть является десятичной дробью, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

(4) Примечание

- При соединении с разделителями в "In" данные, разделенные разделителями, отсутствуют. Если данных не существует, то значение соответствующего члена в Out[] равно REAL#0.
- Если строка в "In" содержит данные, не соответствующие формату вещественных данных, недействительные данные и последующие данные не преобразуются. Функция возвращает значение FALSE, и успешно преобразованные значения остаются неизменными.
- Разделители следует использовать только для разделения в "In". Даже если они используются в других целях, они все равно будут распознаваться как разделитель в этой команде.

- Если результат преобразования превышает область массива Out[], нужно избежать исключений ПЛК, и данные, не превышающие область массива Out[], будут сохранены.
- Если строка ввода пустая, то функция возвращает значение True, а "Out" остается неизменным.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

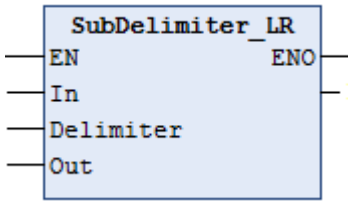
1-6-5. Удаление разделителя после считывания длинной строки действительных чисел SubDelimiter_LR

(1) Обзор команды

Считывание данных, разделенных разделителями, из строки и сохранение их в виде длинного массива действительных чисел.

Удаление разделителя после считывания длинной строки действительных чисел [SubDelimiter_LR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

Библиотека	PP_DataUtils	

Команда	Название	Графическое представление	ST
SubDelimiter_LR	Удаление разделителя после считывания длинной строки действительных чисел		SubDelimiter_LR(In:= , Delimiter:= , Out:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Строка ввода	STRING	Не более 1985 байтов	-	Считываемая строка, разделенная разделителями
Delimiter	Разделитель	_eDELIMITER	В качестве типа данных	_eDELIMITER._COMMA	Разделитель
VAR_IN_OUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Сохранение места хранения массива действительных чисел	LREAL	В качестве типа данных	-	Сохранение больших массивов действительных чисел преобразованных данных

(3) Функция

- Преобразование строки, разделенной разделителем "Delimiter" в строке ввода "In" в массив действительных чисел "Out" и последовательное сохранение его в качестве значений каждого элемента массива.
- Типом данных разделителя "Delimiter" является элемент перечисления _eDELIMITER. Значения элементов перечисления:

Элемент перечисления	Значение
_COMMA	‘,’ (запятая)
_TAB	‘$T’ (символ табуляции)
_SEMICOLON	‘;’ (точка с запятой)
_SPACE	‘ ’ (пробел)

- Если количество данных, разделенных разделителями в "In", превышает количество членов в "Out", лишние данные не будут учитываться.
- Если количество данных, разделенных разделителями в "In", меньше, чем количество членов в "Out", значение членов с недостаточным количеством данных остается неизменным.
- Данные, разделенные разделителями в "In", относятся к строковому типу. Формат строки для представления вещественных (действительных) данных:



Название	Формат
Знаковая часть	Игнорируя последовательные символы ' ' (пробелы) в начале строки, отдельные элементы '+' или '-' следует рассматривать в качестве символов Символ '+' можно опустить. Последовательные пробелы после этого символа будут игнорироваться.
Целая часть	Целая часть — от конца символа до передней части десятичной точки. Однако последовательные пробелы после символа не входят в целую часть. Кроме того, символы можно опускать. Если десятичная точка или десятичная часть отсутствует, то следует рассматривать переднюю часть экспоненциальной части в качестве целой части. Если десятичная точка, десятичная часть или экспоненциальная часть отсутствует, то всю строку следует рассматривать в качестве целой части. Целая часть состоит из чисел в диапазоне от 0 до 9. Целую часть нельзя опустить.

Название	Формат
	Максимальное количество цифр в целой части — это максимальная длина строки в 1985 знаков за вычетом цифр в символе, десятичной точки, десятичной части, экспоненциальной части и пробелов до и после символа.
Десятичная точка	Отдельный символ '.' (точка), за которым следует целая часть, является десятичной точкой. Если десятичная часть отсутствует, десятичную точку опускают.
Десятичная часть	Десятичная часть начинается после десятичной точки и заканчивается перед экспоненциальной частью. Если экспоненциальная часть отсутствует, то вся строка после десятичной точки является десятичной частью. Десятичная часть состоит из чисел в диапазоне от 0 до 9. Десятичную часть можно опустить. Максимальное количество десятичных знаков — 15. Если десятичная точка отсутствует, десятичной части также не существует.
Экспоненциальная часть	Экспоненциальная часть начинается с элемента 'e' , за которым следует десятичная часть и элемент '+' или '-' после нее, и продолжается до конца строки. Если десятичная часть отсутствует, то вышеупомянутая строка после десятичной точки является экспоненциальной частью. Если десятичная точка или десятичная часть отсутствует, то приведенная выше строка после целой части является экспоненциальной частью. Числовая часть экспоненты состоит из цифр от 0 до 9. Экспоненциальную часть можно опустить. Максимальное количество цифр в числовой части экспоненты — 3.

Пример правил преобразования:

Формат	STRING	REAL
Со знаком, с десятичной точкой, с десятичной частью, без экспоненциальной части	‘-123,4567’	-123,4567
Со знаком, с десятичной точкой, с десятичной частью, с экспоненциальной частью	‘+123,4567e+02’	12345.67

Без знака, с десятичной точкой, с десятичной частью, с экспоненциальной частью	‘123,4567e-02’	1.234567
Без знака, без десятичной точкой, без десятичной части, без экспоненциальной части	‘1’	1

Меры предосторожности при преобразовании:

- Если числовое значение, представленное строкой ввода, выходит за пределы допустимого диапазона данных LREAL (4.940656458412 47E-324~1.7976931348623157E+308), если это положительное число, то результатом преобразования будет бесконечность. Если это отрицательное число, результатом преобразования будет отрицательная бесконечность.
- Если строка ввода не является длинным действительным числом,то результатом преобразования будет 0, а функция возвращает значение FALSE.
- Если в любом положении в середине строки ввода есть два или более последовательных символа "_" (подчеркивание), преобразование происходит нормально.
- Если между '-' (отрицательный знак) или '+' (положительный знак) в начале строки ввода и числа стоит символ '_' (подчеркивание), преобразование происходит нормально.
- Если в начале или конце строки ввода есть символ '_' (подчеркивание), возникает исключение преобразования (функция возвращает FALSE).
- Если в любом положении в середине строки ввода есть отдельный символ '_' (подчеркивание), он будет проигнорирован.
- Если содержимое строки ввода превышает точность, с которой оно может быть представлено типом данных LREAL, оно будет округлено.
- Если содержимое строки ввода ближе к 0, чем значение, которое может быть представлено типом данных LREAL, то результатом преобразования будет 0.
- Если десятичная часть строки ввода превышает максимум 15 цифр, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Если экспоненциальная часть строки ввода превышает максимальное количество в 3 цифры, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.
- Если строка ввода содержит целые числа, десятичные числа и экспоненты с номерами, не находящимися в диапазоне 0-9, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

- Строка ввода не содержит десятичной части, но десятичная точка не опущена. Программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

- Экспоненциальная часть строки ввода не имеет числовой части, но в ней есть символы 'e+'/'e-'. Программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

- Если другие строки ввода содержат любые иррегулярные числа, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

- Если до и после ‘+/-’ экспоненциальной части строки ввода стоит символ пробела, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

- Непрерывные символы пробела в начале и конце строки ввода (в том числе до и после разделителя) будут игнорироваться, и команда будет нормально выполнена.

- Если экспоненциальная часть является десятичной дробью, программа не будет выполняться и не может быть преобразована. Функция возвращает значение FALSE.

(4) Примечание

- "In" содержит данные, разделенные разделителями. Если данных не существует, то значение соответствующего члена в Out[] равно LREAL#0.

- Если строка в "In" содержит данные, не соответствующие формату вещественных данных, недействительные данные и последующие данные не преобразуются. Функция возвращает значение FALSE, и успешно преобразованные значения остаются неизменными.

- Разделители следует использовать только для разделения в "In". Даже если они используются в других целях, они все равно будут распознаваться как разделитель в этой команде.

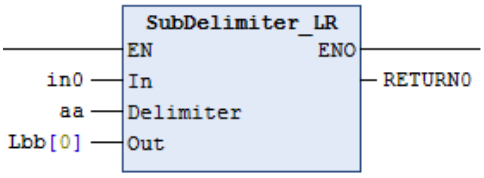
- Если результат преобразования превышает область массива Out[], нужно избежать исключений ПЛК, и данные, не превышающие область массива Out[], будут сохранены.

- Если строка ввода пустая, то функция возвращает значение True, а "Out" остается неизменным.

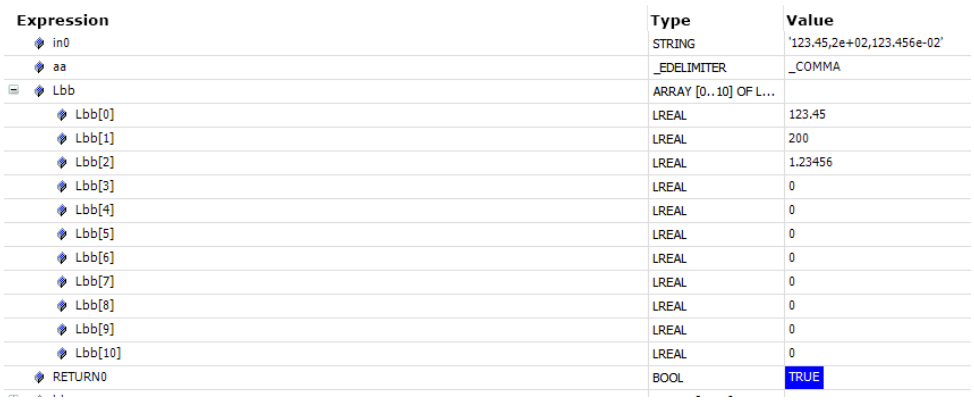
(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    in0:STRING;
    aa: XJ_DataUtils._eDELIMITER;
    Lbb:ARRAY[0..10] OF LREAL;
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>
</td><td>

```
SubDelimiter_LR(In:=in0,
    Delimiter:=aa ,
    Out:=lbb[0] );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">
</td></tr>
</table>

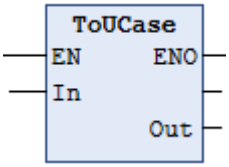
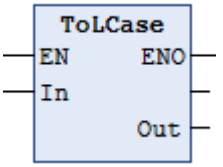
1-6-6. Преобразование букв строки ToUCase/ToLCase

(1) Обзор команды

ToUCase: Преобразует все буквы половинной ширины в строке в заглавные буквы.

ToLCase: Преобразует все буквы половинной ширины в строке в строчные буквы.

Преобразование букв строки [ToUCase/ToLCase]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
ToUCase	Преобразование букв в строке в заглавные буквы		ToUCase(In:= , Out=>);
ToLCase	Преобразование букв в строке в строчные буквы		ToLCase(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Объект преобразования	STRING	Не более 1985 байтов	-	Строка с преобразованным объектом
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	STRING	Не более 1985 байтов	-	Преобразованная строка

(3) Функция

- ToUCase: Преобразует буквы половинной ширины в строке с преобразованным объектом "In" в заглавные буквы.
- ToLCase: Преобразует буквы половинной ширины в строке с преобразованным объектом "In" в строчные буквы.
- Все выводимые команды содержат символ NULL в конце строки. Символы с буквами не половинной ширины не меняются.

(4) Примечание

- Буквы полной ширины не относятся к объекту преобразования.
- Если текстовый код в "In" неправильный, может возникнуть исключение, и значение "Out" будет неточным.
- Если "In" не заканчивается символом NULL, это может привести к

неправильным результатам преобразования.

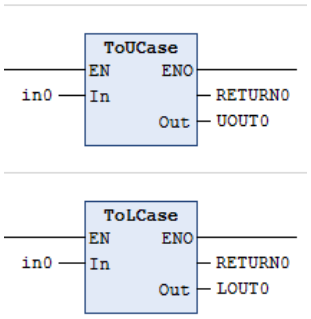
- Если результат преобразования превышает размер "Out", будет отображаться только значение преобразования длины Out.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    in0:STRING;
    UOUT0: STRING(1985);
    LOUT0: STRING(1985);
    RETURN0: BOOL;

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
ToLCase(In:=in0 ,
    Out=>Lout0);

ToUCase(In:=in0 ,
    Out=>Uout0);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
in0	STRING	'ABC'
UOUT0	STRING(1985)	'ABC'
LOUT0	STRING(1985)	'abc'
RETURN0	BOOL	TRUE

Expression	Type	Value
in0	STRING	'abc'
UOUT0	STRING(1985)	'ABC'
LOUT0	STRING(1985)	'abc'
RETURN0	BOOL	TRUE

</td></tr>
</table>

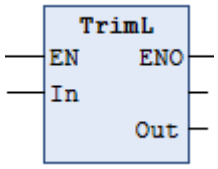
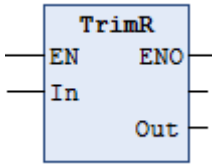
1-6-7. Обрезка строки слева/обрезка строки справа TrimL/TrimR

(1) Обзор команды

TrimL: Удаляет пробелы в начале текстовой строки.

TrimR: Удаляет пробелы в конце текстовой строки.

Обрезка строки слева/обрезка строки справа [TrimL/TrimR]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_DataUtils		

Команда	Название	Графическое представление	ST
TrimL	Обрезка строки слева		TrimL(In:= , Out=>);
TrimR	Обрезка строки справа		TrimR(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
In	Удаление объекта	STRING	Не более 1985 байтов	-	Объектная строка удалена
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
Out	Удаление результата	STRING	Не более 1985 байтов	-	Строка после удаления

(3) Функция

- TrimL: Удаление пробелов от начала объектной строки "In" до первого появившегося символа. Если в начале текстовой строки нет пробелов, никаких действий не производится.
- TrimR: Удаление пробелов от последнего символа объектной строки "In" до конца. Если в конце текстовой строки нет пробелов, никаких действий не производится.
- Обе команды выводят результаты, включая символ NULL в конце строки.

(4) Примечание

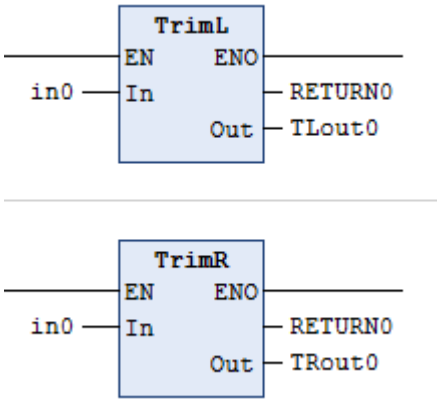
- Обе команды выводят результаты, включая символ NULL в конце строки.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR

    in0:STRING;
    TLout0: STRING(1985);
    TRout0: STRING(1985);

END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
TrimL(In:=string0 ,
    Out=>outl[2] );
TrimR(In:=string0 ,
    Out=>outl[3] );
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
in0	STRING	' ABC '
TLout0	STRING(1985)	'ABC '
TRout0	STRING(1985)	' ABC'

</td></tr>
</table>

2. Специальные команды

2-1. Команды высокоскоростного счетчика

2-1-1. Обзор функции

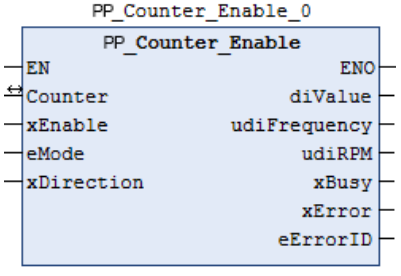
При выборе счетчиков можно измерять высокоскоростные входные сигналы, например, сигналы измерительных датчиков и поворотных энкодеров, с максимальной частотой измерения 1 МГц. Библиотеки команд высокоскоростных счетчиков для, PMP30-60A32-E и PMP301-24A8 представлены в главах 2-2.

2-1-2. Функциональный блок

※	Графическое представление некоторых команд и их представление на языке программирования ST выделены красными волнистыми линиями, так как для компиляции и загрузки программы параметру VAR_IN_OUT'Counter при вызове функции должно присваиваться значение.

2-1-2-1. Включение высокоскоростного счетчика [PP_Counter_Enable]

Включение высокоскоростного счетчика [PP_Counter_Enable]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_Enable	Включение высокоскоростного счетчика		PP_Counter_Enable_0(Counter:= , xEnable:= , eMode:= , xDirection:= , diValue=> , udiFrequency=> , udiRPM=> , xBusy=> , xError=> , eErrorID=>);

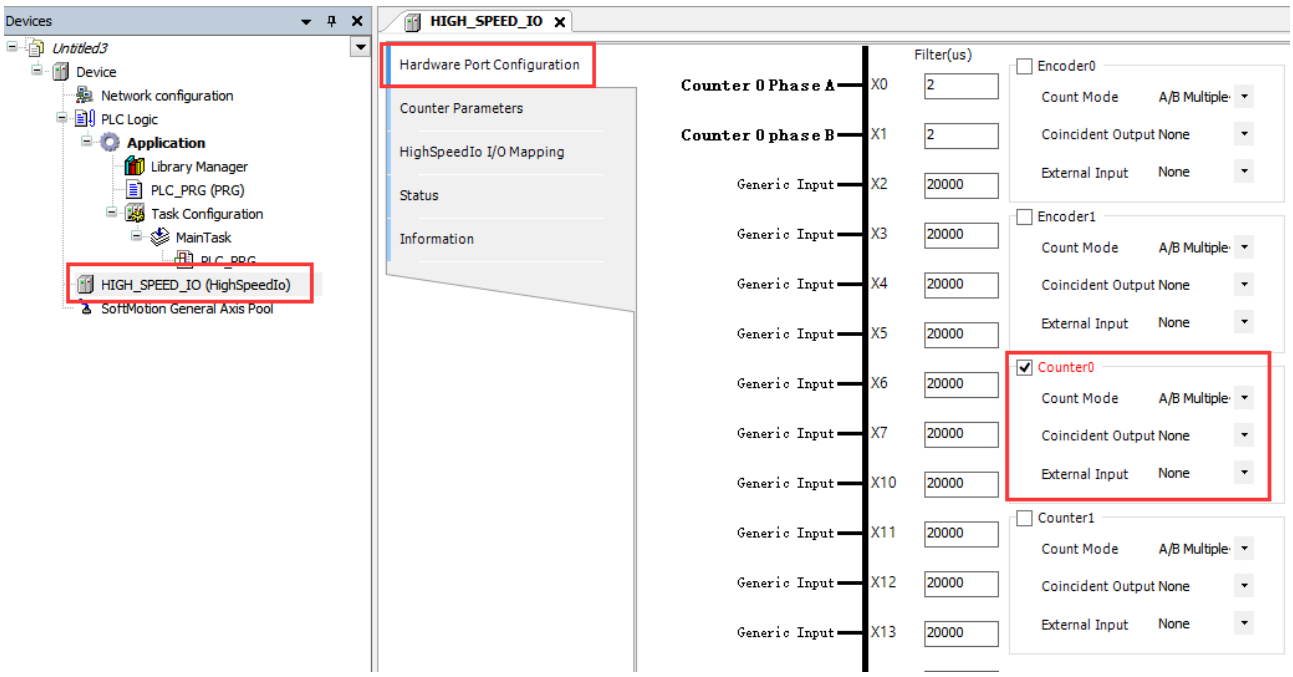
(1) Входные переменные

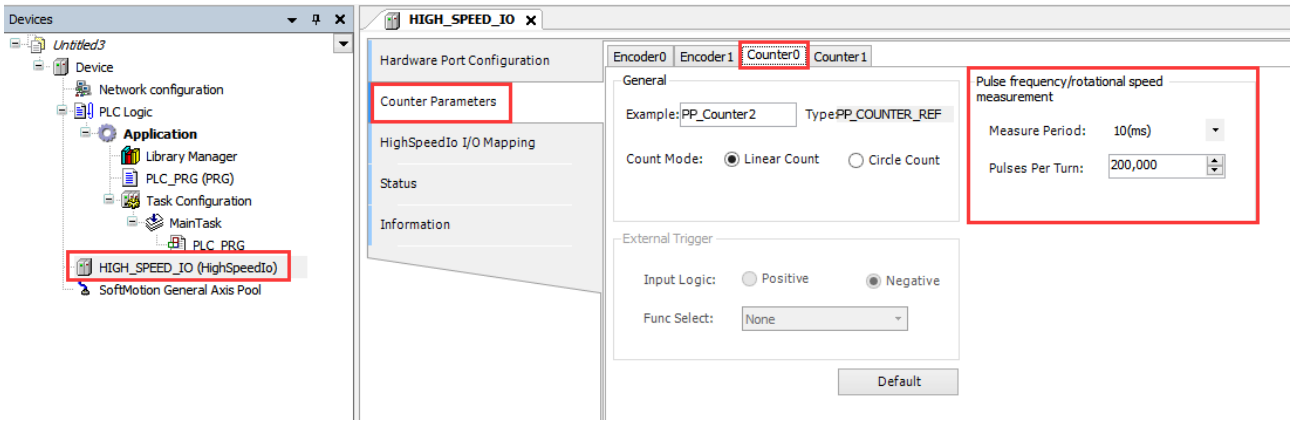
Входная переменная	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	После настройки входа высокоскоростного счетчика в интерфейсе высокоскоростного ввода-вывода он будет автоматически реализован. Выберите имя экземпляра высокоскоростного счетчика.
xEnable	Включение	BOOL	TRUE, FALSE	FALSE	Чтобы включить режим счета, должно быть состояние ON (ВКЛ).
eMode	Режим счета	HSC_EDGE_MODE	PosEdge, NegEdge, BothEdge	PosEdge	PosEdge: подсчет по нарастающему фронту NegEdge: подсчет по спадающему фронту BothEdge: подсчет по двум фронтам
xDirection	Направление	BOOL	TRUE, FALSE	FALSE	FALSE: прямой счет TRUE: обратный счет

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
diValue	Показания высокоскоростного счетчика	DINT	Диапазон данных	0	Показания высокоскоростного счетчика
udiFrequency	Измеренное значение частоты импульсов	UDINT		0	Единица измерения: Гц; если это низкая частота, то ее можно использовать вместе с периодом измерения через интерфейс
udiRPM	Число оборотов в минуту	UDINT		0	Единица измерения: об/мин, используется вместе с количеством импульсов на оборот в конфигурации интерфейса
xBUSY	Занято	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

Для работы в режиме высокоскоростного счета нужно проверить и настроить высокоскоростной счетчик через параметры аппаратного обеспечения, задать способ счета и параметры счетчика, связанные с измерением частоты импульсов/скорости вращения.





Примечание:

(1) Присвойте значение параметру VAR_IN_OUT'Counter' при вызове PP_Counter_Enable, после чего программу можно будет компилировать и загружать.

(2) Примечания по настройке переменной xDirection:

- При наличии одной фазы или фаз A и B подсчет осуществляется в прямом/обратном направлении с учетом состояния клеммы xDirection.
- В режиме P+D клемма направления со значением FALSE и клемма xDirection со значением FALSE обеспечивают подсчет в прямом направлении. Клемма направления со значением FALSE, клемма xDirection

со значением TRUE обеспечивают вычитание. Клемма направления имеет состояние TRUE, а клемма xDirection — FALSE, что указывает на вычитание. Клемма направления и клемма xDirection имеют состояние TRUE, что указывает на сложение.

(3) За цикл измерения осуществляется сбор данных о количестве импульсов за определенный период для расчета:

- Цикл измерения составляет 1000 мс, а минимальная измеряемая частота — 1 Гц. Цикл измерения составляет 1 мс, а минимальная измеряемая частота — 1 кГц.
- Цикл измерения составляет 10 мс, а минимальная измеряемая частота — 100 Гц. Цикл измерения составляет 100 мс, а минимальная измеряемая частота — 10 Гц. Максимальная частота измерения — это частота, которую можно определить при высокоскоростном подсчете.

(4) Единицей измерения скорости вращения в минуту является об/мин, и результаты измерений могут колебаться в ту или иную сторону из-за воздействия аппаратного обеспечения.

(5) Разница между линейным счетом и кольцевым счетом:

- Линейный счет: (DownLimitValue, UpLimitValue)
- Кольцевой счет: [iRingDownValue, iRingUpValue]
- Счетчик 0-3: Диапазон линейного счета — (-2147483648, 2147483647), кроме -2147483648 и 2147483647.
- Диапазон кольцевого счета — [-2147483648, 2147483647], включая -2147483648 и 2147483647.

(6) Если для счетчика 0/счетчика 1 (Counter0/Counter1) выбран режим счетчика фаз A и B (AB phase counting), отображается только частота фазы A. Настраивается через интерфейс одной фазы, фаз A и B.

(7) Если значение счетчика приближается к граничному значению, то при переполнении счетчика ошибка может не выдаваться из-за влияния цикла сканирования.

(8) При загрузке программы счетчик не обнуляется.

(9) Включите функциональный блок PP_Counter_Enable перед использованием функционального блока PP_Counter_Sample.

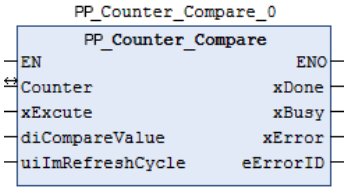
2-1-2-2. Постоянный выход сравнения [PP_Counter_Compare]

Настройте постоянный вывод данных. Если требуется функция прерывания

высокоскоростного подсчета, необходимо вызвать PP_EnableInterrupt для включения прерывания.

※	Многосегментное сравнение и односегментное сравнение выполняются в порядке срабатывания, и более позднее срабатывание не прерывает первое срабатывание. Выполнение функции, запущенной позже, начнется после выполнения функции, запущенной раньше.

Постоянный выход сравнения [PP_Counter_Compare]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_Compare	Постоянный выход сравнения		PP_Counter_Compare_0(Counter:= , xExcute:= , diCompareValue:= , uiImRefreshCycle:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

(1) Входная переменная

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExecute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер

diCompareValue	Назначение значения для сравнения	DINT		0	
uiImRefreshCycle	Время прямого вывода данных аппаратным обеспечением	UINT		0	Единица измерения: 100 мкс, максимальное время вывода данных — 3000 мс.

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После выполнения команды признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErr1orID	Тип ошибки	HSIO_ERROR	-	0	

Примечание:

(1) Управление выходами Y0-Y7 можно настраивать в фоновом режиме, при этом каждый счетчик может произвольно выбирать выходы Y0-Y7 с определенной задержкой при выводе данных. Время вывода данных: 100 мкс - 3000 мс. Если ImRefreshCycle равен 0, вызовите PP_Reset CmpOutput, чтобы уменьшить объем выводимых данных.

(2) ImRefresh=1: Аппаратное обеспечение немедленно осуществляет вывод данных, выходы Y0-Y7 нельзя произвольно выбирать, задержка отсутствует, а время вывода данных задается через ImRefreshCycle. Время вывода данных: 0 - 3000 мс.

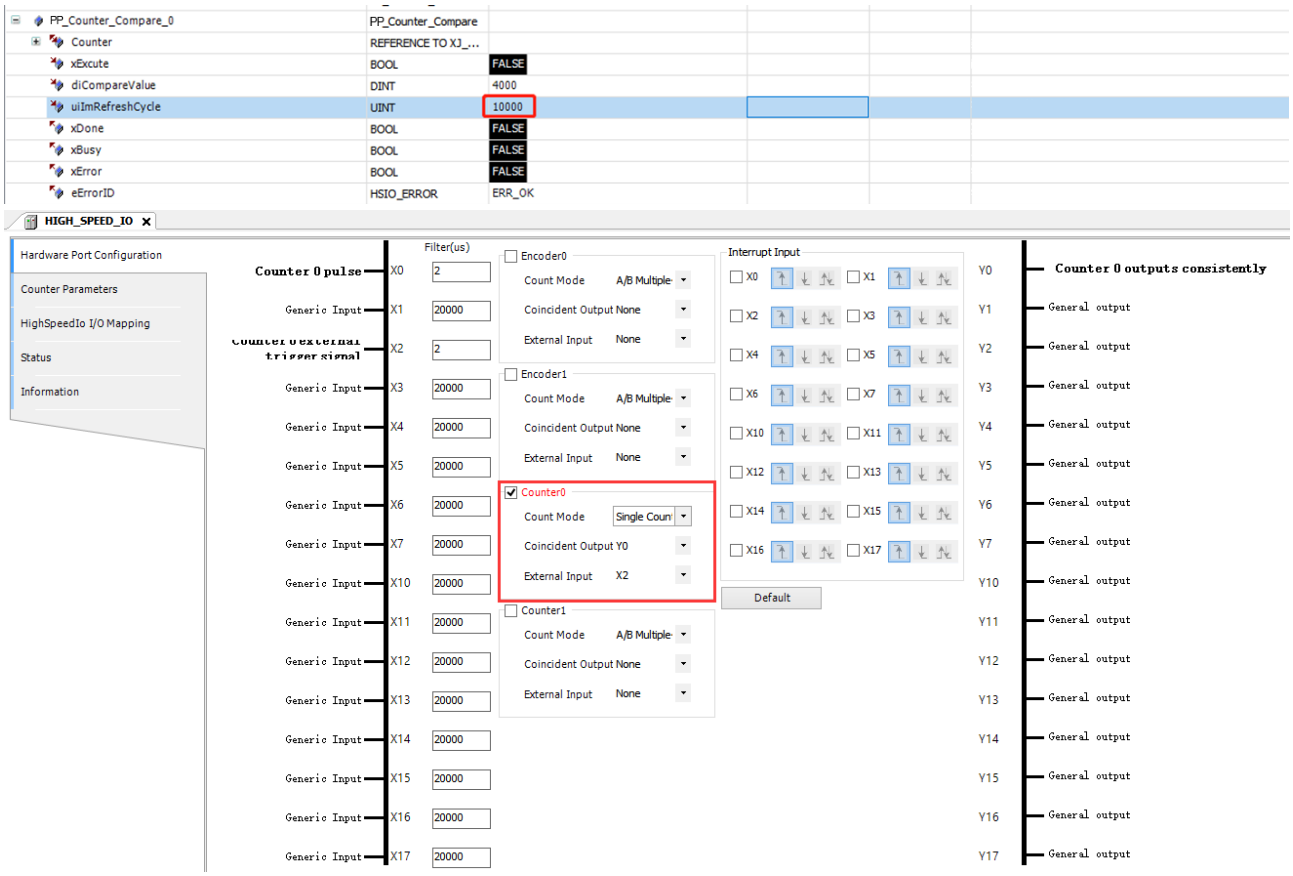
(3) PP_EnableInterrupt необходимо вызвать заранее, чтобы запустить постоянное прерывание при сравнении.

(4) После горячего и холодного сброса сохраняется предыдущее значение для сравнения.

Настройте постоянный выход сравнения и вызовите PP_EnableInterrupt, чтобы запустить прерывание.

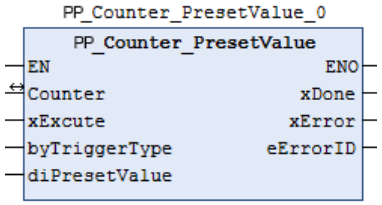
Если аппаратный выход не настроен, необходимо запустить постоянное прерывание при сравнении. Если аппаратный выход настроен, постоянное прерывание при сравнении запускать необязательно.

Например, если время вывода данных составляет 1 секунду, то после сравнения значений аппаратный выход Y будет равен 1 секунде, а программное состояние Y останется неизменным, и выходные данные отображаться не будут. Выполните программу прерывания после ввода внешнего триггера.



2-1-2-3. Запись заданного значения [PP_Counter_PresetValue]

Запись заданного значения [PP_Counter_PresetValue]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_PresetValue	Запись заданного значения		PP_Counter_PresetValue_0(Counter:= , xExcute:= , byTriggerType:= , diPresetValue:= , xDone=> , xError=> , eErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExecute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер
byTriggerType	Тип триггера	HSC_PSV_TIGGLE		0	0: Запись, запускаемая по нарастающему фронту 1: Запуск по внешнему входному сигналу 2: При сравнении во время постоянного вывода данных настройка и запуск происходят по нарастающему фронту.
diPresetValu	Заданное	DINT	Диапазон	0	Запись заданного

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
e	значение		данных		значения высокоскоростного счетчика

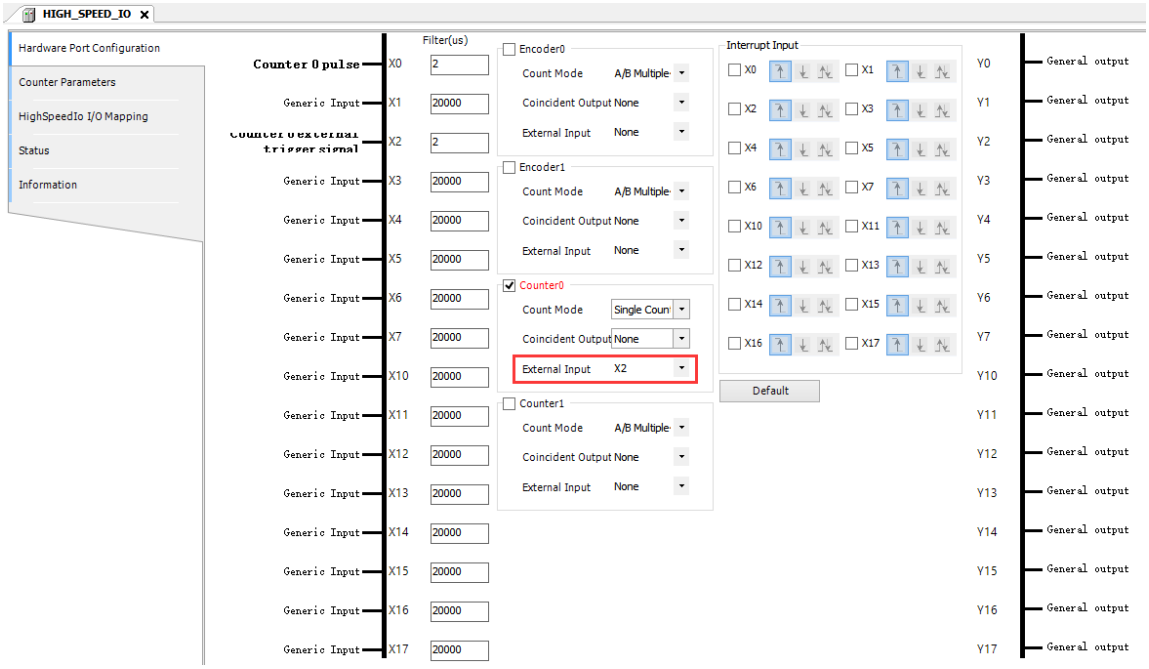
(2) Выходные переменные

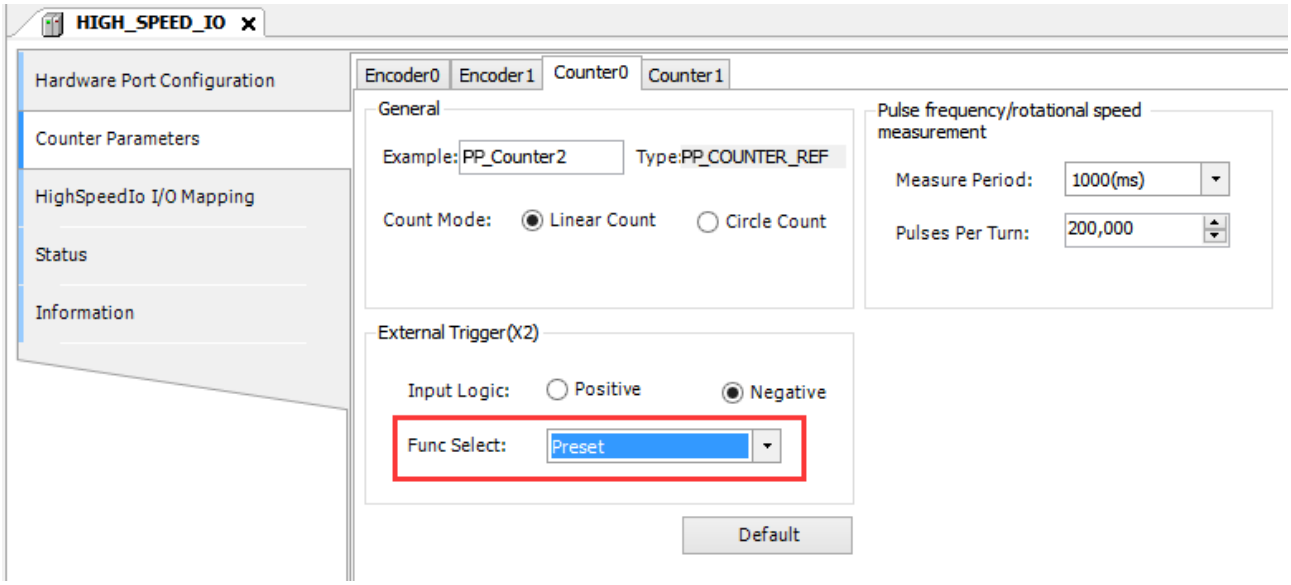
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения записи признак будет иметь состояние TRUE.
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

Примечание: Если значение, отображаемое для ErrorID, равно 2, то CounterID не находится в диапазоне от 0 до 3.

0: Запись, запускаемая по нарастающему фронту. После запуска команды заданное значение можно будет записать в высокоскоростной счетчик.

1: Запуск по внешнему входному сигналу. После запуска команды дождитесь сигнала с внешней входной клеммы, и если сигнал получен, то можно записывать заданное значение.





2: При сравнении во время постоянного вывода данных настройка и запуск происходят по нарастающему фронту.

2-1-2-4. Контактный датчик [PP_TouchProbe]

Контактный датчик [PP_TouchProbe]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_TouchProbe	Контактный датчик	PP_TouchProbe_0 PP_TouchProbe EN Counter xExcute ENO xDone xBusy diTouchValue xError eErrorID	PP_TouchProbe_0(Counter:= , xExcute:= , xDone=> , xBusy=> , diTouchValue=> , xError=> , eErrorID=>);

(1) Входные переменные

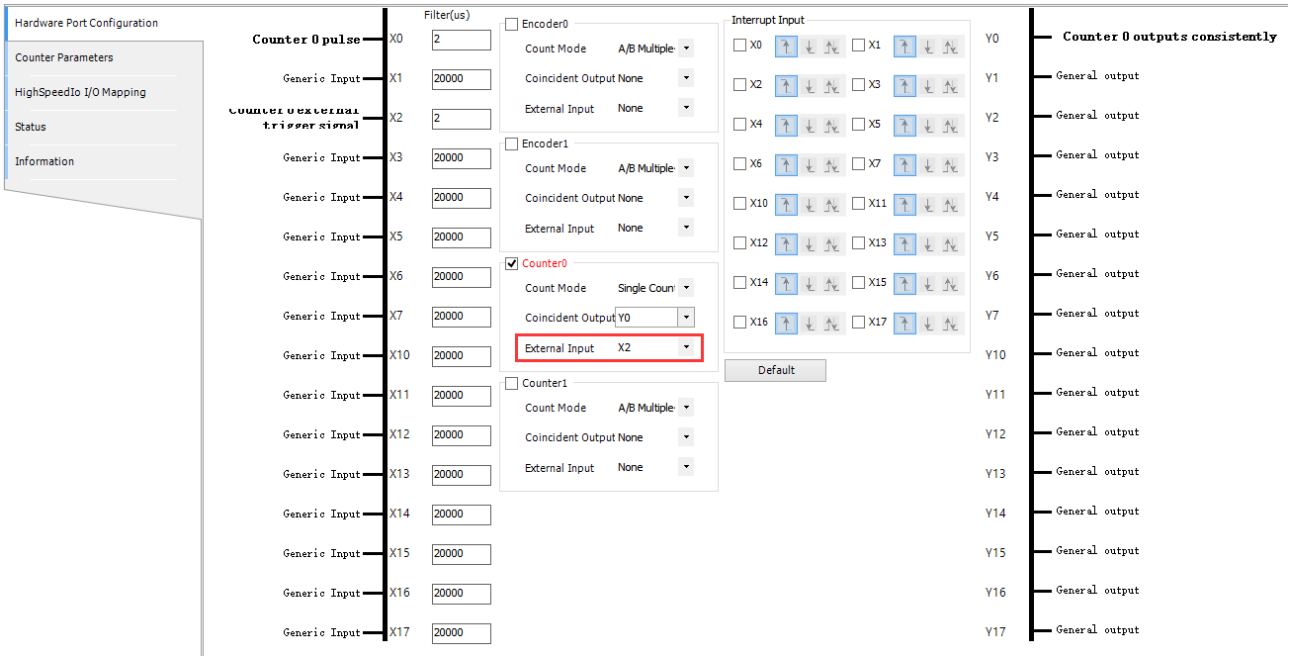
Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExcute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер

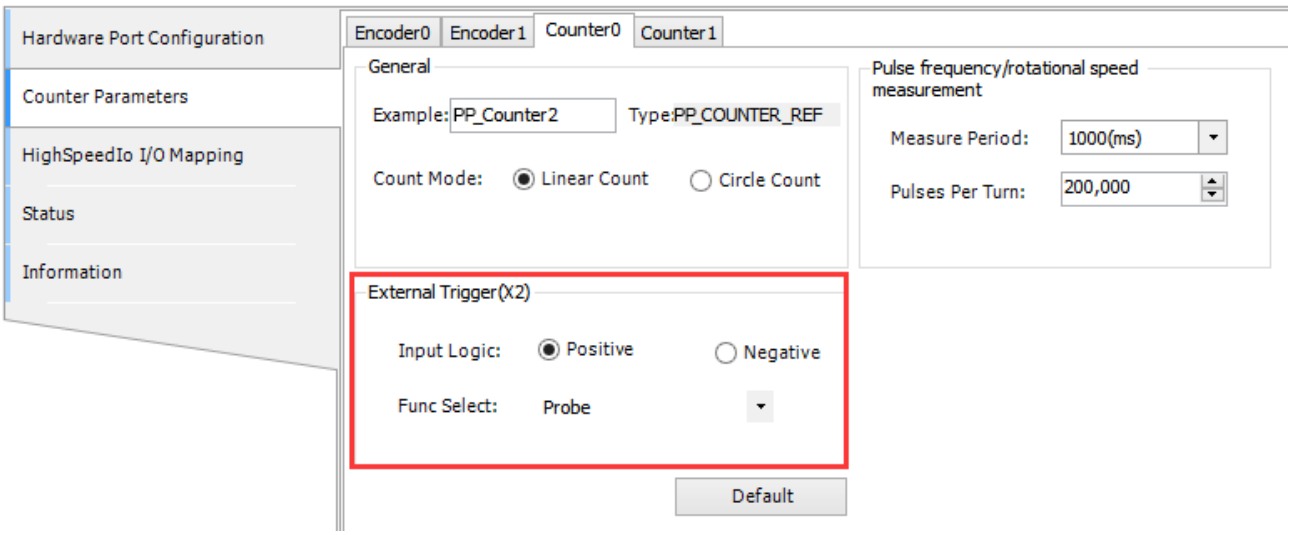
(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения записи признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
diTouchVaule	Значение триггера-защелки	DINT		0	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

Настройте клемму на функцию контактного датчика. После срабатывания этой клеммы текущее показание высокоскоростного счетчика можно заблокировать, как показано на рисунке ниже. Положительная логика будет задействована после запуска по нарастающему фронту.

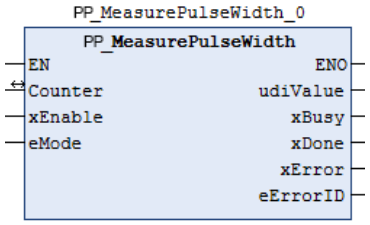
PP_TouchProbe_0	PP_TouchProbe				
Counter	REFERENCE TO PP_COUNTER_REF				PP_COUNTER_REF
xExcute	BOOL	FALSE			
xDone	BOOL	FALSE			
xBusy	BOOL	FALSE			
diTouchValue	DINT	0			
xError	BOOL	FALSE			
eErrorID	HSIO_ERROR	ERR_OK			





2-1-2-5. Считывание измеренного значения ширины импульса счетчика [PP_MeasurePulseWidth]

Считывание измеренного значения ширины импульса счетчика [PP_MeasurePulseWidth]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_MeasurePulseWidth	Считывание измеренного значения ширины импульса счетчика		PP_MeasurePulseWidth_0(Counter:= , xEnable:= , eMode:= , udiValue=> , xBusy=> , xDone=> , xError=> , eErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExecute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер
eMode	Измерение ширины	HSC_PULSEWI	0,1	0	0: Высокий уровень

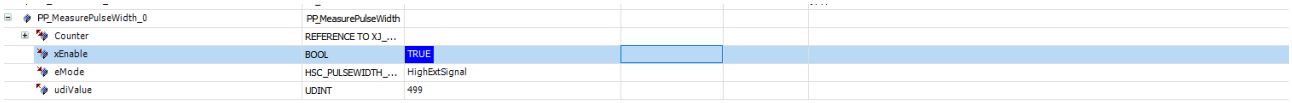
	импульса высокого/низкого уровня	DTH_TYPE			внешнего сигнала (измерение ширины импульса высокого уровня) 1: Низкий уровень внешнего сигнала (измерение ширины импульса низкого уровня)

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
udiValue	Измеренное значение	UDINT		0	
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения записи признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

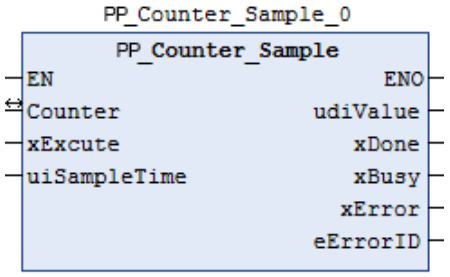
Примечание: Измерение длительности высокого и низкого уровней в микросекундах (мкс).

Пример: Если частота составляет 1 кГц, это эквивалентно отправке импульса каждую 1 мс, а если высокий или низкий уровень присутствует каждые 0,5 мс, то измеренный высокий или низкий уровень составляет около 500 мкс.



2-1-2-6. Выборка счетчика [PP_Counter_Sample]

Выборка счетчика [PP_Counter_Sample]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_Sample	Выборка счетчика		PP_Counter_Sample_0(Counter:= , xExcute:= , uiSampleTime:= , udiValue=> , xDone=> , xBusy=> , xError=> , eErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
eExecute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер
uiSampleTime	Период выборки	UINT		0	Период выборки (10-65535 мс)

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
udiValue	Значение выборки	UDINT		0	
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения записи признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

Примечание:

(1) Команда PP_Counter_Sample может выполняться после включения PP_Counter_Enable.

(2) После запуска команды количество сигналов выборки не изменится в реальном времени, но отобразится после выполнения команды. При этом будут выведены показания высокоскоростного счетчика, полученные за заданное время.

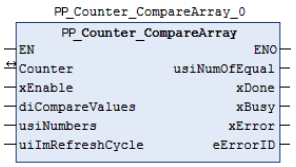
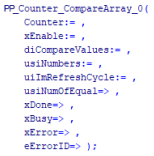
2-1-2-7. Многосегментное сравнение [PP_Counter_CompareArray]

Постоянный выход сравнения: Можно задавать до 100 значений для сравнения, и после сравнения появится сигнал Done (Готово).

※	Многосегментное сравнение и односегментное сравнение выполняются в порядке срабатывания, и более позднее срабатывание не прерывает первое срабатывание. Выполнение функции, запущенной позже, начнется после выполнения функции, запущенной раньше.

Многосегментное сравнение [PP_Counter_CompareArray]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP301-24
Прошивка	-	Программное	-

		обеспечение	
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_CompareArray	Многосегментное сравнение		

(1) Входные переменные

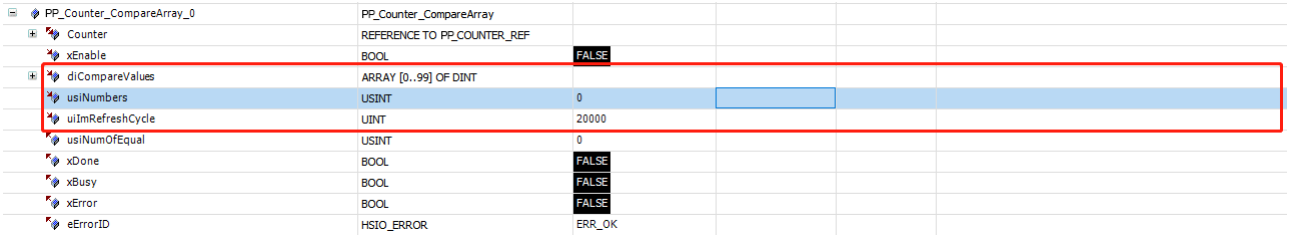
Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xEnable	Включение	BOOL	TRUE,FALSE	FALSE	Включение
diCompareValues	Одномерный массив значений для сравнения	ARRAY		0	Одномерный массив значений сравнения для набора счетчиков (поддержка максимум 100 значений)
usiNumbers	Заданное фактическое значение	USINT		0	Заданное фактическое значение, допустимое значение 1, максимум 100 значений

uiImRefreshcycle	Время прямого вывода данных аппаратным обеспечением	UINT		0	Единица измерения — 100 мкс, а максимальное время вывода данных — 3000 мс. Например, уставка 10000 означает 1000 мс.

(2) Выходные переменные

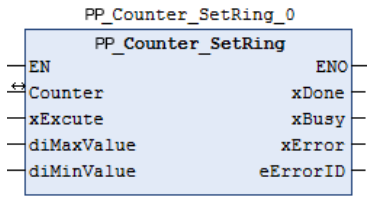
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
usiNumOfEqual	Равные числа	USINT		0	
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения записи признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

- Можно задать несколько значений для сравнения, и при достижении значения для сравнения будет выполнен вход в команду выполнения задачи внешних событий с аппаратным выводом.
- Отличие от односегментного сравнения: Нужно запустить постоянное прерывание при сравнении, иначе команда выдаст ошибку.
- Настройте соответствующие параметры, как показано на рисунке ниже:



2-1-2-8. Кольцевой счет [PP_Counter_SetRing]

Кольцевой счет [PP_Counter_SetRing]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_SetRing	Настройка кольцевого счетчика		PP_Counter_SetRing_0(Counter:= , xExcute:= , diMaxValue:= , diMinValue:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

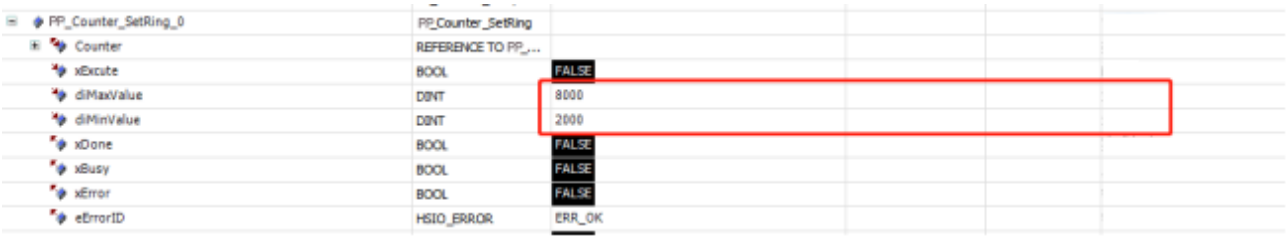
(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExcute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер
diMaxValue	Максимальное значение	DINT		0	Максимальное значение
diMinValue	Минимальное значение	DINT		0	Минимальное значение

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения записи признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

- Текущее значение счетчика должно находиться в пределах заданного диапазона значений, в противном случае команда выдаст сообщение об ошибке. После выполнения данной команды никакая другая команда не может переключиться обратно на линейный счет. Если нужно переключиться обратно на линейный счет, нужно изменить диапазон счета на линейный.

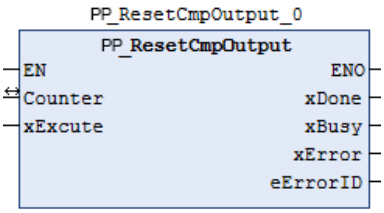


- Минимальное значение для линейного счета составляет -2147483647, а максимальное — 2147483646. Например, при добавлении одного сигнального входа после увеличения счета до 2147483646 значение изменится на -2147483647, а затем продолжится отсчет в прямом направлении (в сторону увеличения). Он не достигнет минимального значения для двухсловных знаковых чисел -2147483648 и максимального значения 2147483647.

- Существует два способа настройки кольцевого счета: во-первых, если вы еще не вошли в систему, в пункте меню "High speed IO" (Высокоскоростной ввод-вывод) - "Counter parameter settings" (Настройки параметров счетчика), можно нажать "Ring counting" (Кольцевой счет) и задать диапазон. Второй способ — можно использовать настройку команды PP_Counter_SetRing, которая позволяет напрямую переключать линейный счет на кольцевой счет во время входа в систему.

2-1-2-9. Сброс порта постоянного выхода сравнения [PP_ResetCmpOutput]

Сброс порта постоянного выхода сравнения [PP_ResetComOutput]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_ResetCmpOutput	Сброс порта постоянного выхода сравнения		PP_ResetCmpOutput_0(Counter:= , xExcute:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

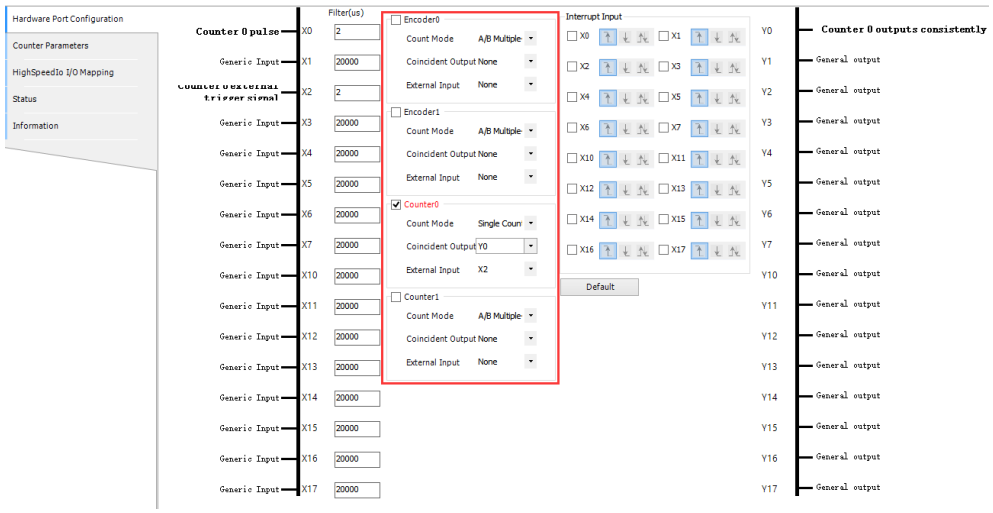
(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExcute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения настройки признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

Выполните сброс всех портов постоянного выхода сравнения. При этом выход Y, который изначально был настроен как постоянный выход сравнения, можно использовать как обычный выход Y. Что касается использования точки Y в качестве постоянного выхода сравнения или обычного выхода, то это зависит от того, какая команда порта срабатывает первой для команды постоянного выхода сравнения и команды сброса постоянного выхода сравнения.

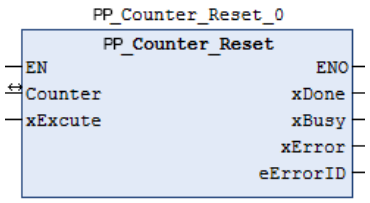


2-1-2-10. Сброс ошибок [PP_Counter_Reset]

Сброс ошибок при выполнении команд высокоскоростного подсчета.

Сброс ошибок [PP_Counter_Reset]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное	-

		обеспечение	
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter_Reset	Сброс ошибок		PP_Counter_Reset_0(Counter:= , xExcute:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Входной порт высокоскоростного счетчика	PP_COUNTER_REF	-	-	Высокоскоростной счетчик; необходимо определить входную клемму для высокоскоростного подсчета (см. функциональное описание).
xExcute	Триггер	BOOL	TRUE,FALSE	FALSE	Триггер

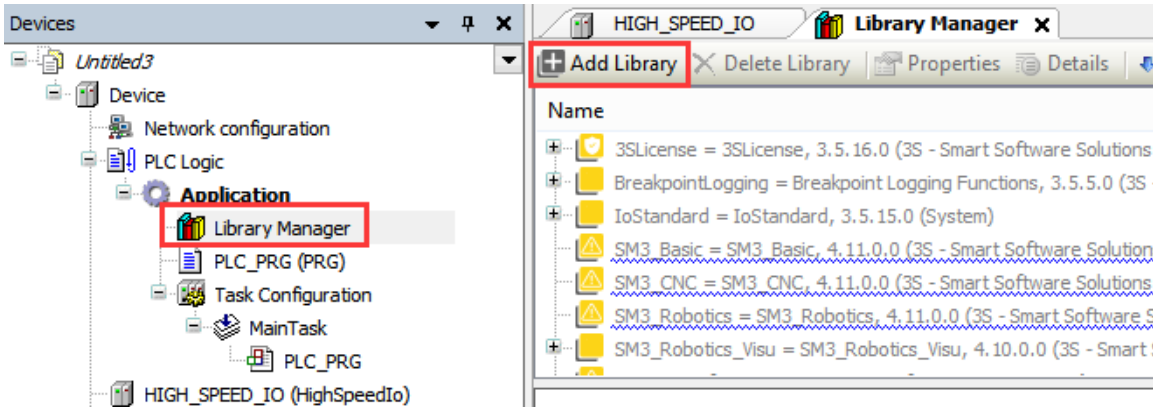
(2) Выходные переменные

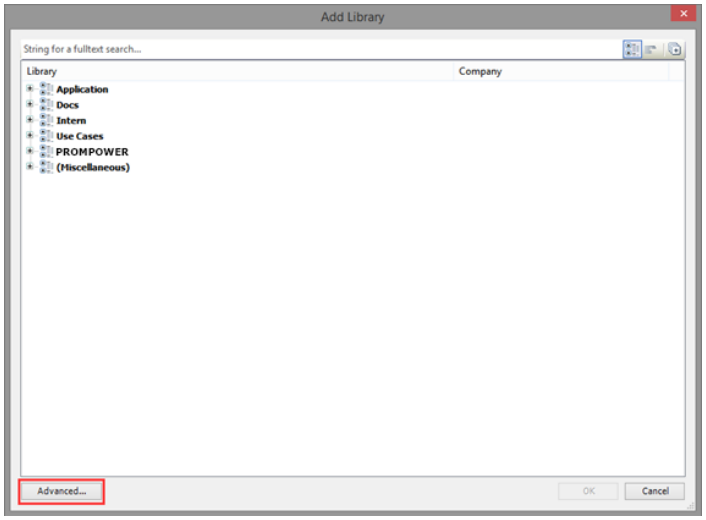
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения настройки признак будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	

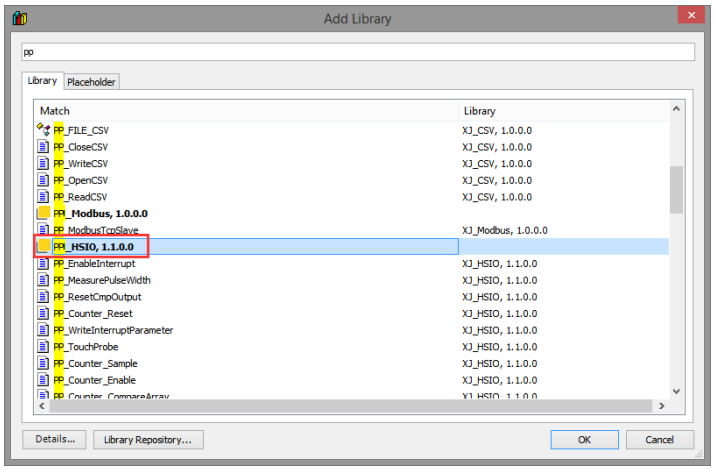
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

2-1-3. Настройки параметров

Добавление библиотечного файла: Library Manager - Add Library - Advanced - Add PP_HSIO (Диспетчер библиотек - Добавить библиотеку - Дополнительно - Добавить PP_HSIO).







2-2. Команды высокоскоростного счетчика контроллеров серии PMP3xx

2-2-1. Обзор функции

ПЛК серии PMP3xx поддерживают функцию высокоскоростного подсчета (счета), которая позволяет измерять высокоскоростные входные сигналы, например, сигналы измерительных датчиков и поворотных энкодеров, путем выбора различных счетчиков. Максимальная частота измерения может достигать 200 кГц.

※	Библиотека команд, описанная в настоящей главе, подходит только для устройств PMP30-60A32-E и PMP301-24A8.

2-2-2. Функциональный блок

2-2-2-1. Высокоскоростной счетчик [PP_Counter]

Высокоскоростной счетчик [PP_Counter]			
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP30, PMP301-24A8
Прошивка	-	Программное обеспечение	-
Библиотека	PROMPOWERCnt		

Команда	Название	Графическое представление	Представление на языке ST
PP_Counter	Высокоскоростной счетчик	PP_Counter_0 PP_Counter EN — ENO Counter — CounterValue Enable — Error Mode — ErrorID	PP_Counter(Counter:= , Enable:= , Mode:= , CounterValue=> , Error=> , ErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Counter	COUNTER_REF	-	-	Высокоскоростной счетчик, задающий

					входную клемму и начальное значение высокоскоростного подсчета
Enable	Включение	BOOL	TRUE,FALSE	FALSE	Включение в нормально разомкнутом состоянии для запуска счета
Mode	Режим счета	Режим	Режим фаз A и B, режим одной фазы	FALSE	Режим высокоскоростного счета: MODE=PP. Режим фаз A и B — это режим работы высокоскоростного счетчика для фаз A и B. MODE=PP. Режим одной фазы — это режим работы высокоскоростного счетчика для одной фазы.

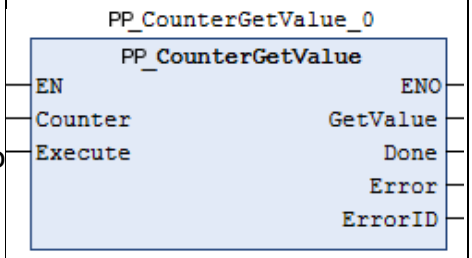
(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CounterValue	Значение счета	DINT	Диапазон данных	0	Показания высокоскоростного счетчика
Error	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
ErrorID	Тип ошибки	UINT	-	0	

2-2-2-2. Считывание показаний высокоскоростного счетчика [PP_CounterGetValue]

Считывание показаний высокоскоростного счетчика [PP_CounterGetValue]			
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP30, PMP301-24A8
Прошивка	-	Программное обеспечение	-

Библиотека	PROMPOWERCnt	

Команда	Название	Графическое представление	Представление на языке ST
PP_CounterGetValue	Считывание показаний высокоскоростного счетчика		PP_CounterGetValue(Counter:= , Execute:= , GetValue=> , Done=> , Error=> , ErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Counter	COUNTER_REF	-	-	Высокоскоростной счетчик, задающий входную клемму и начальное значение высокоскоростного подсчета
Execute	Включение	BOOL	TRUE,FALSE	FALSE	Запуск по нарастающему фронту, считывание текущих показаний высокоскоростного счетчика

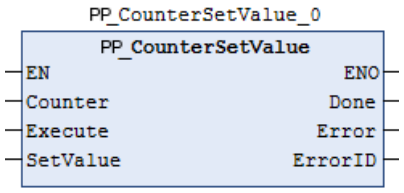
(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetValue	Считывание значения	DINT	Диапазон данных	0	Текущее значение счета
Done	Завершение	BOOL	TRUE,FALSE	FALSE	После завершения считывания признак будет иметь состояние TRUE.
Error	Признак ошибки	BOOL	TRUE,FALSE	FALSE	

ErrorID	Тип ошибки	UINT	-	0	

2-2-2-3. Запись показаний высокоскоростного счетчика [PP_CounterSetValue]

Запись показаний высокоскоростного счетчика [PP_CounterSetValue]			
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP30, PMP301-24A8
Прошивка	-	Программное обеспечение	-
Библиотека	PROMPOWERCnt		

Команда	Название	Графическое представление	Представление на языке ST
PP_CounterSetValue	Запись показаний высокоскоростного счетчика		PP_CounterSetValue(Counter:= , Execute:= , SetValue:= , Done=> , Error=> , ErrorID=>);

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Counter	Counter	COUNTER_REF	-	-	Высокоскоростной счетчик, задающий входную клемму и начальное значение высокоскоростного подсчета
Execute	Включение	BOOL	TRUE,FALSE	FALSE	Запуск по нарастающему фронту, запись показаний высокоскоростного счетчика, запись SetValue в CounterValue
SetValue	Запись значения	DINT	Диапазон данных	0	Запись заданного значения

					высокоскоростного счетчика

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Завершение	BOOL	TRUE,FALSE	FALSE	После завершения записи бит признака будет иметь состояние TRUE.
Error	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
ErrorID	Тип ошибки	UINT	-	0	

Примечание: Если значение, отображаемое для ErrorID, равно 2, то CounterID не находится в диапазоне от 0 до 3.

(3) Описание функции

- Функция высокоскоростного подсчета состоит из трех функциональных блоков: функционального блока высокоскоростного подсчета, функционального блока считывания показаний высокоскоростного счетчика и функционального блока записи показаний высокоскоростного счетчика. Высокоскоростной вход ПЛК серии PMP3xx может принимать только дифференциальный сигнал (DIFF), но не может принимать сигнал разомкнутого коллектора (OC). Необходимо правильно выбрать устройство кодирования дифференциальных сигналов. Высокоскоростной вход ПЛК серии PMP30 предназначен для приема сигнала разомкнутого коллектора (OC).

- Счетчик поддерживает тип данных COUNTER_REF:

Описание типа данных COUNTER_REF:

Элемент	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CounterID	Клемма счетчика	INT	0,1,2,3	0	Выбор входного порта высокоскоростного счетчика
CounterValue	Начальное значение счетчика	DINT	Диапазон данных	0	Настройка начального значения счетчика

- Высокоскоростные счетчики ПЛК серии PMP30 поддерживают два режима работы: однофазный инкрементный режим и режим фаз A и B.

(1) Mode= Single_Mode (однофазный режим)

В этом режиме подсчитываются входные импульсные сигналы, и значение подсчета увеличивается с нарастанием фронта каждого импульсного сигнала.

(2) Mode=AB_Mode (режим фаз A и B)

В этом режиме показания высокоскоростного счетчика увеличиваются или уменьшаются в зависимости от импульсного сигнала (фаза A и фаза B) со сдвигом фаз 90°. Режим счета по умолчанию — 4-кратная частота.

- Выбор входного порта высокоскоростного счетчика для ПЛК серии PMP3xx

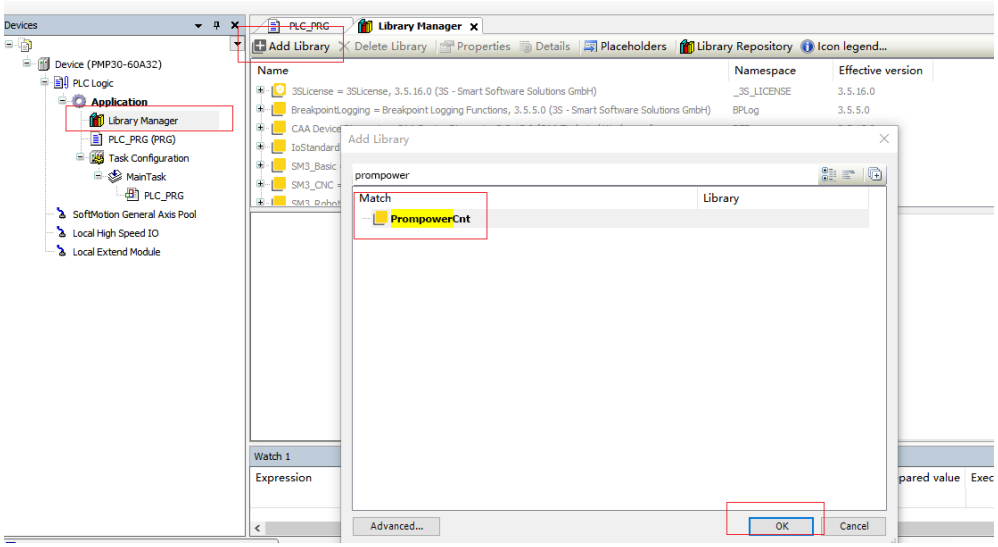
PMP30-60A32-E								
	Однофазный режим				Режим фаз A и B			
CounterID	0	1	2	3	0	1	2	3
Макс. частота	200k	200k	200k	200k	100k	100k	100k	100k
X0	U				A			
X1					B			
X2								
X3		U				A		
X4						B		
X5								
X6			U				A	
X7							B	
X10								
X11				U				A
X12								B
X13								

PMP301-24A8								
	Однофазный режим				Режим фаз A и B			
HSC	0	1	2	3	0	1	2	3
Макс частота	80k	80k	80k	80k	50k	50k	50k	50k
Удвоение частоты					2/4	2/4	2/4	2/4
Прерывание счётчика	√	√	√	√	√	√	√	√
X0	U				A			
X1					B			
X2								
X3		U				A		
X4						B		
X5								
X6			U				A	
X7							B	
X10								
X11				U				A
X12								B

2-2-3. Конфигурация параметров

Добавление библиотечного файла:

Добавьте "PROMPOWERCnt" в пункте меню "Library Manager" (Диспетчер библиотек), и после добавления можно будет задействовать функцию высокоскоростного подсчета.



2-2-4. Пример использования

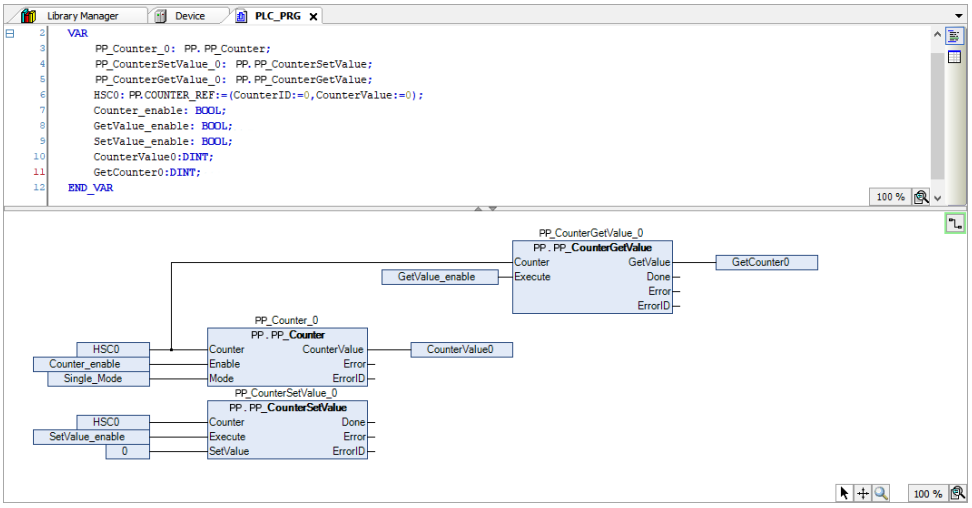
Пример 1: Используйте первый канал высокоскоростного счетчика и считайте текущее значение счетчика, чтобы внести в него изменения.

Работа с программой:

(1) Установите требуемые библиотеки согласно процедуре, описанной в разделах 2-1-3.

(2) Запишите программу высокоскоростного подсчета.

Программирование: С помощью функциональных блоков "PP.PP_Counter", "PP.PP_CounterGetValue" и "PP.PP_CounterSetValue" настройте порт порта высокоскоростного счетчика, режим высокоскоростного счета и значение высокоскоростного счетчика, используемое в программе.



2-3. Команды внешнего прерывания и постоянного прерывания при сравнении

2-3-1. Обзор функции

ПЛК серии прерывание по клемме X, и эта же клемма поддерживает прерывания по нарастающему фронту и спадающему фронту. В ПО Codesys прерывания используются через внешние события в типе задачи. Например, X2R_TRIG обеспечивает прерывание по нарастающему фронту X2, X2F_TRIG — прерывание по спадающему фронту, а количество и тип прерываний, поддерживаемых каждой моделью, можно настроить в пункте меню "External event" (Внешнее событие).

※	PMP30-60A32-E и PMP301-30A32 в настоящее время не поддерживают эту библиотеку команд. Информация о внешних прерываниях приведена в руководстве «Руководство пользователя для контроллера серии PMP3xx с поддержкой стандарта PLCopen [программное обеспечение]».

2-3-2. Функциональный блок

2-3-2-1. Запуск внешних прерываний и постоянных прерываний при сравнении [PP_EnableInterrupt]

Запуск внешних прерываний и постоянных прерываний при сравнении [PP_EnableInterrupt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_EnableInterrupt	Запуск внешних прерываний и постоянных	PP_EnableInterrupt_0 PP_EnableInterrupt EN — ENO xEnable — xValid udiExternal — xBusy uiCompare — xError eErrorID	PP_EnableInterrupt(xEnable:= , udiExternal:= , uiCompare:= , xValid=> , xBusy=> , xError=> , eErrorID=>);

	прерываний при сравнении		

(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xEnable	Включение	BOOL	TRUE,FALSE	FALSE	Включение
udiExternal	Запуск внешнего прерывания на входе	UINT		0	
uiCompare	Запуск постоянного прерывания при сравнении	UINT		0	Например, 4 означает CounterID:=2

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xValid	Прерывание вступает в силу	BOOL	TRUE,FALSE	FALSE	После вступления прерывания в силу бит признака будет иметь состояние TRUE.
xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

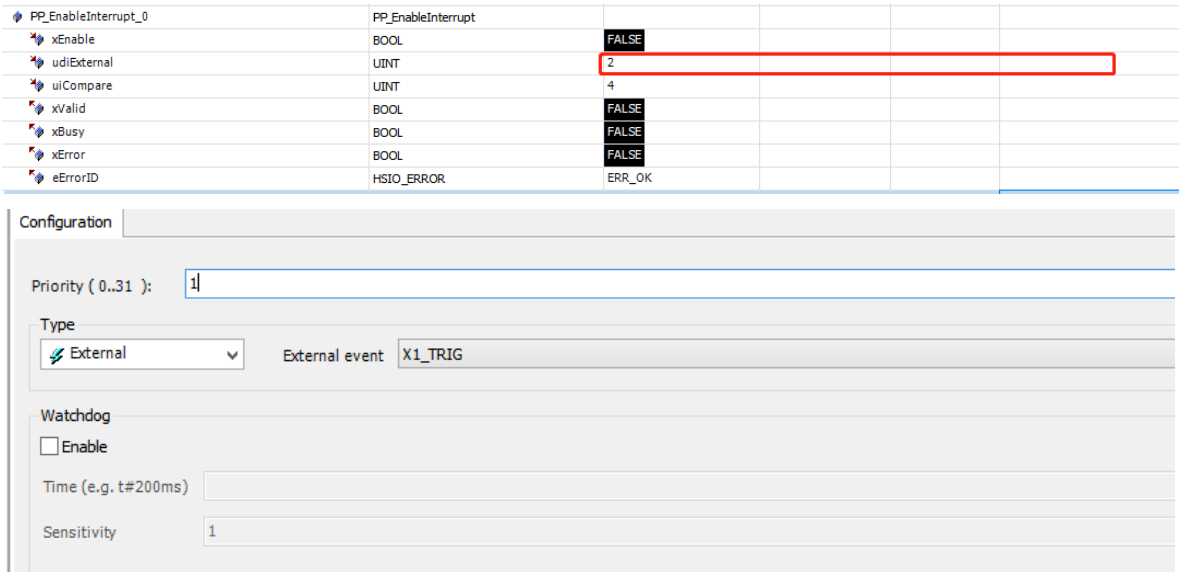
- **Запуск внешнего прерывания на входе:**

1. Выберите клемму внешних прерываний, например, X1



2. Откройте PP_EnableInterrupt, чтобы добавить задачу прерывания, где 1 обозначает X0, 2 обозначает X1, 3 обозначает X0, X1, и т.д. Как показано на рисунке ниже, задачи внешнего прерывания выполняются, когда действует нарастающий фронт клеммы X1.

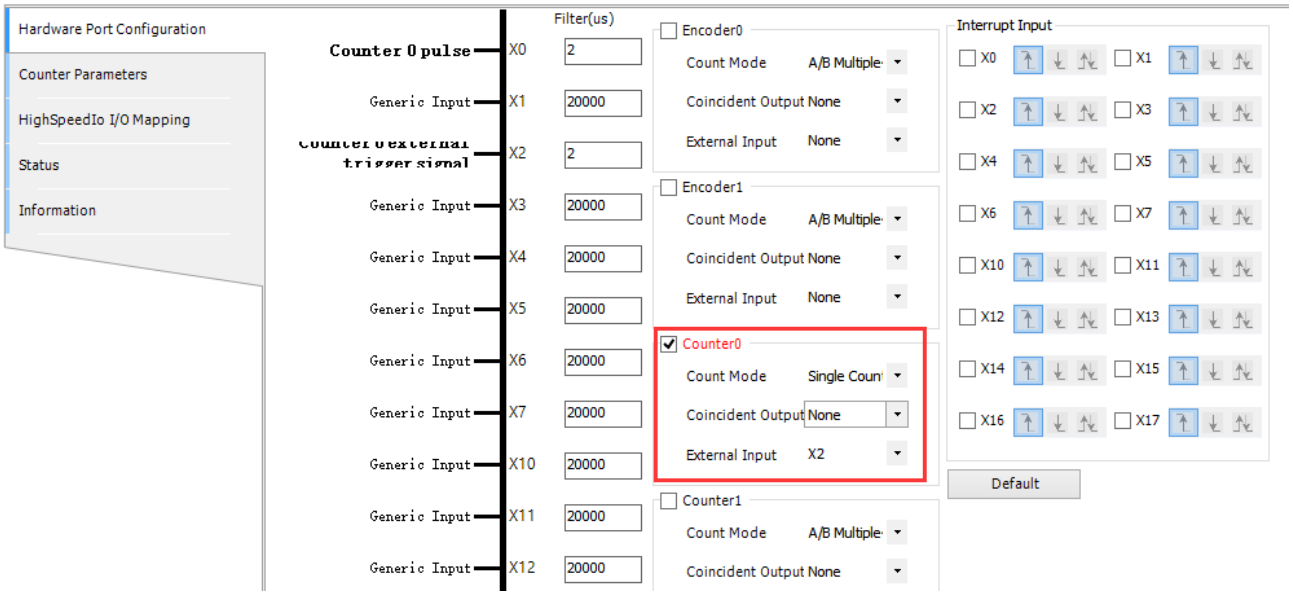
То есть udiExternal запускает внешнее прерывание на выходе и отображает его в виде двоичного адреса, а затем выдает значение. Например, если вы вводите двоичное число 2#11 или десятичное 10#3, откройте X0 и X1. В пункте меню "Hardware Parameter Configuration" (Конфигурация параметров аппаратного обеспечения) нужно также проверить требуемые входы X0 и X1, которые можно выбрать как «фронт» (edge) или «два фронта» (double edge). При поступлении сигнала от X0 и X1 задача, сконфигурированная как "X0" или "X1", выполнит программу.

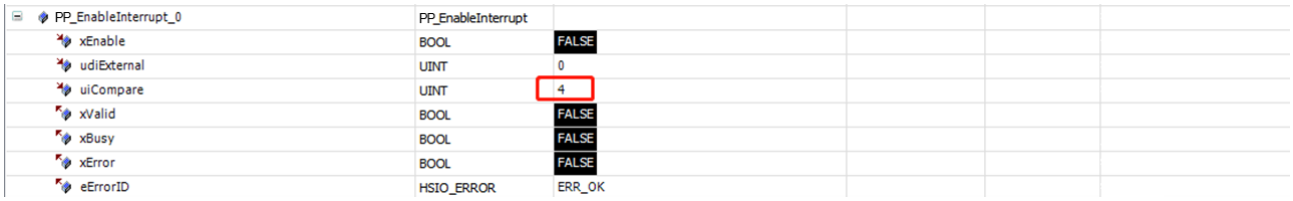


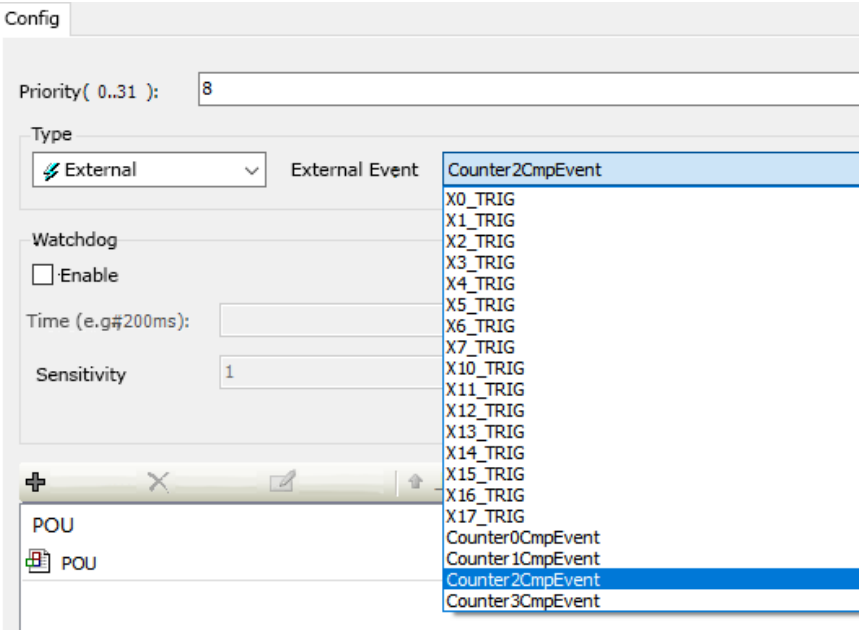
Чтобы порт стал активным, его следует настраивать через интерфейс и команды.

- **Запуск постоянного прерывания при сравнении:**

Настройте идентификатор счетчика, который нужно открыть, например, 3, вводы CounterID:=0 и CounterID:=1. В данном случае используются M210 и CounterID:=2, поэтому следует запустить постоянное прерывание при сравнении для записи 4. После достижения значения для сравнения счетчик начнет работать с командой выполнения задачи внешних событий.

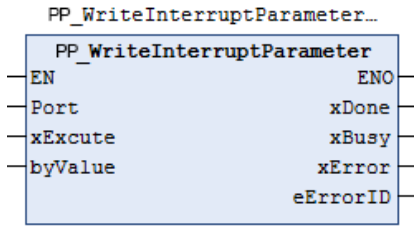






2-3-2-2. Запись параметров прерывания [PP_WriteInterruptParameter]

Запись параметров прерывания [PP_WriteInterruptParameter]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP301-24
Прошивка	-	Программное обеспечение	-
Библиотека	PP_HSIO		

Команда	Название	Графическое представление	Представление на языке ST
PP_WriteInterruptParameter	Запись параметров прерывания		PP_WriteInterruptParameter(Port:= , xExcute:= , byValue:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

(1) Входные переменные

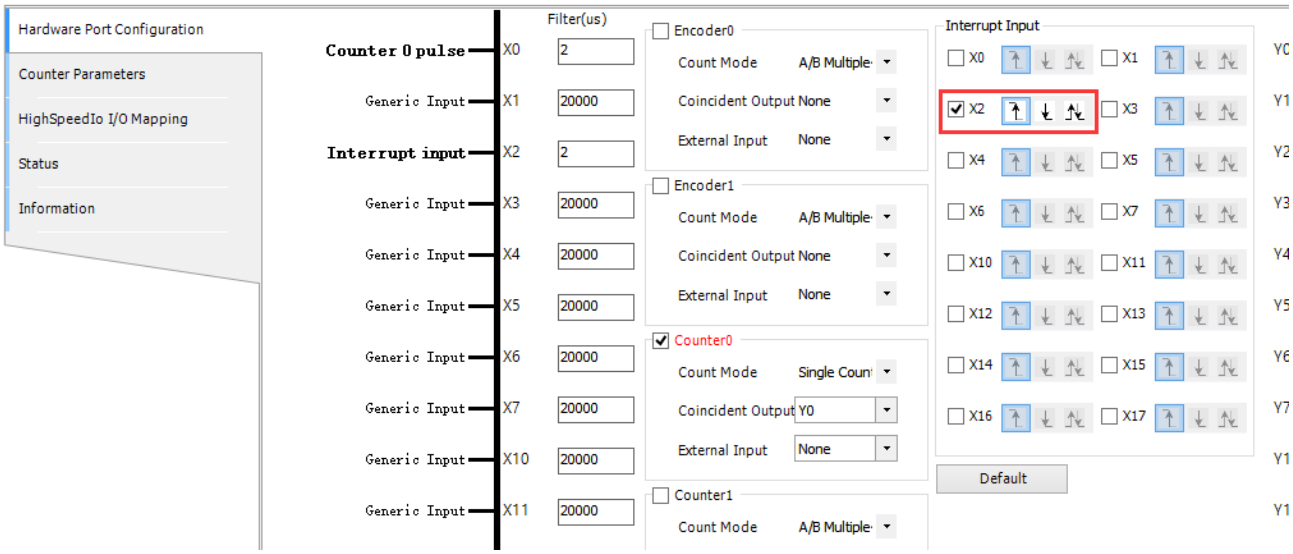
Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Port	Номер порта	UINT		0	Двоичное число, например, 2#11, обозначающее входы X0 и X1
xExcute	Триггер	BOOL	TRUE,FALSE	FALSE	
byValue	Значение	BYTE		0	

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Признак завершения	BOOL	TRUE,FALSE	FALSE	После завершения настройки бит признака будет иметь состояние TRUE.

xBusy	Выполняется	BOOL	TRUE,FALSE	FALSE	
xError	Признак ошибки	BOOL	TRUE,FALSE	FALSE	
eErrorID	Тип ошибки	HSIO_ERROR	-	0	

Параметры внешнего прерывания можно настроить через интерфейс или изменить с помощью команды.



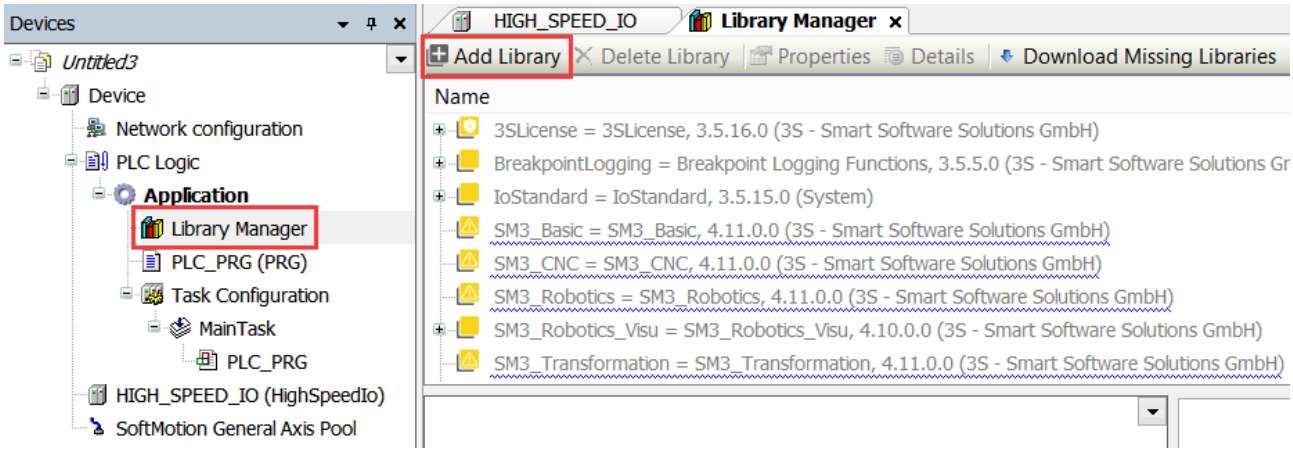
PP_WriteInterruptParameter_0	PP_HSIO.PP_WriteInterruptParameter	
Port	UINT	4
xExcute	BOOL	FALSE
byValue	BYTE	0
xDone	BOOL	FALSE
xBusy	BOOL	FALSE
xError	BOOL	FALSE
eErrorID	HSIO_ERROR	ERR_OK

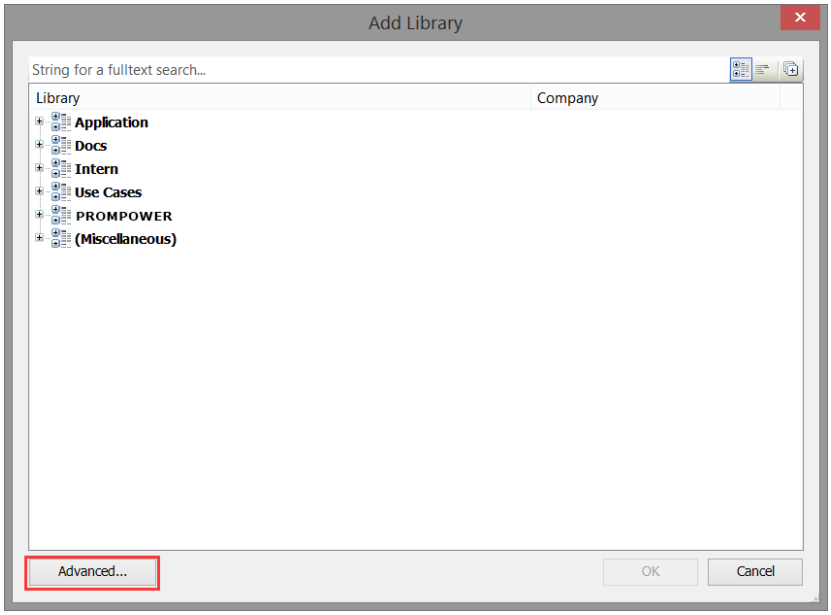
- Если PP_WriteInterruptParameter выполняется до PP_EnableInterrupt, то параметры PP_WriteInterruptParameter являются допустимыми.
- Если PP_EnableInterrupt выполняется до PP_WriteInterruptParameter, то параметры прерывания в фоновом режиме являются допустимыми. Если PP_WriteInterruptParameter выполняется повторно, то параметры PP_WriteInterruptParameter являются допустимыми.
- Во время выполнения этой команды необходимо также запустить внешние прерывания на входе.

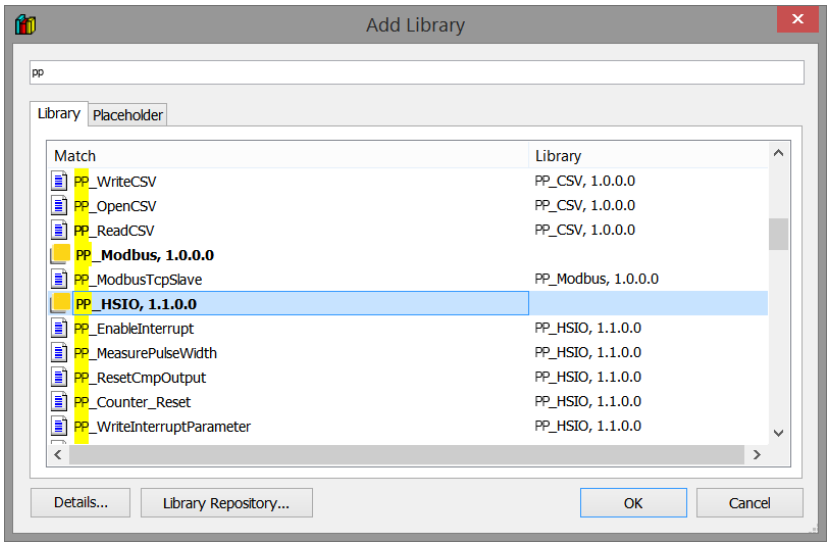
PP_EnableInterrupt_0	PP_EnableInterrupt	
xEnable	BOOL	FALSE
udiExternal	UINT	4
uiCompare	UINT	4
xValid	BOOL	FALSE
xBusy	BOOL	FALSE
xError	BOOL	FALSE
eErrorID	HSIO_ERROR	ERR_OK

2-3-3. Конфигурация параметров

Добавление библиотечного файла: Library Manager - Add Library - Advanced - Add PP_HSIO (Диспетчер библиотек - Добавить библиотеку - Дополнительно - Добавить PP_HSIO).

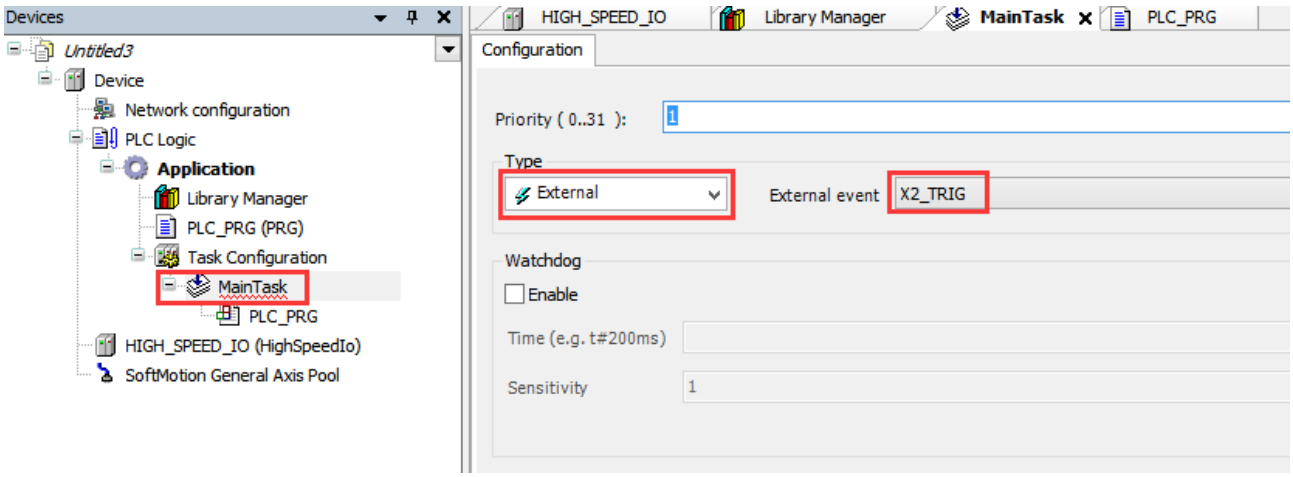




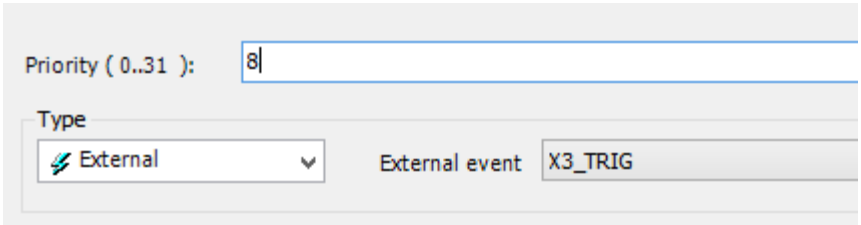


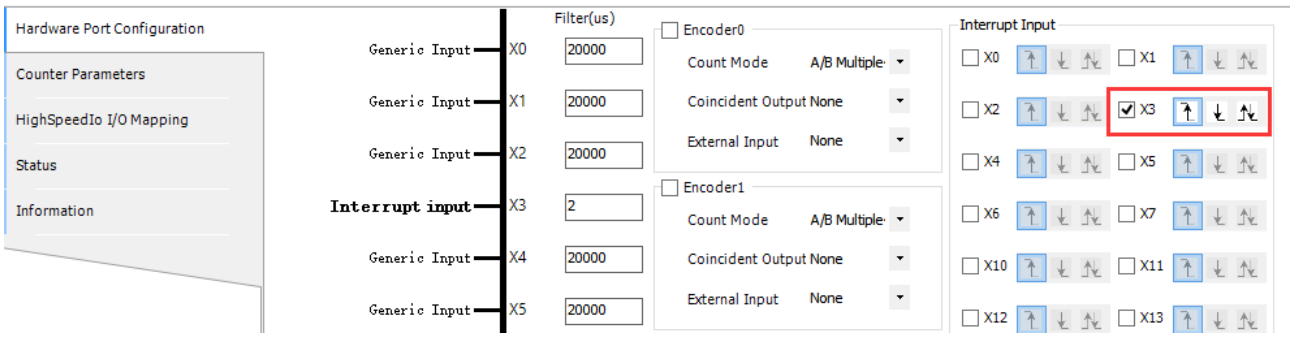
2-3-4. Пример использования

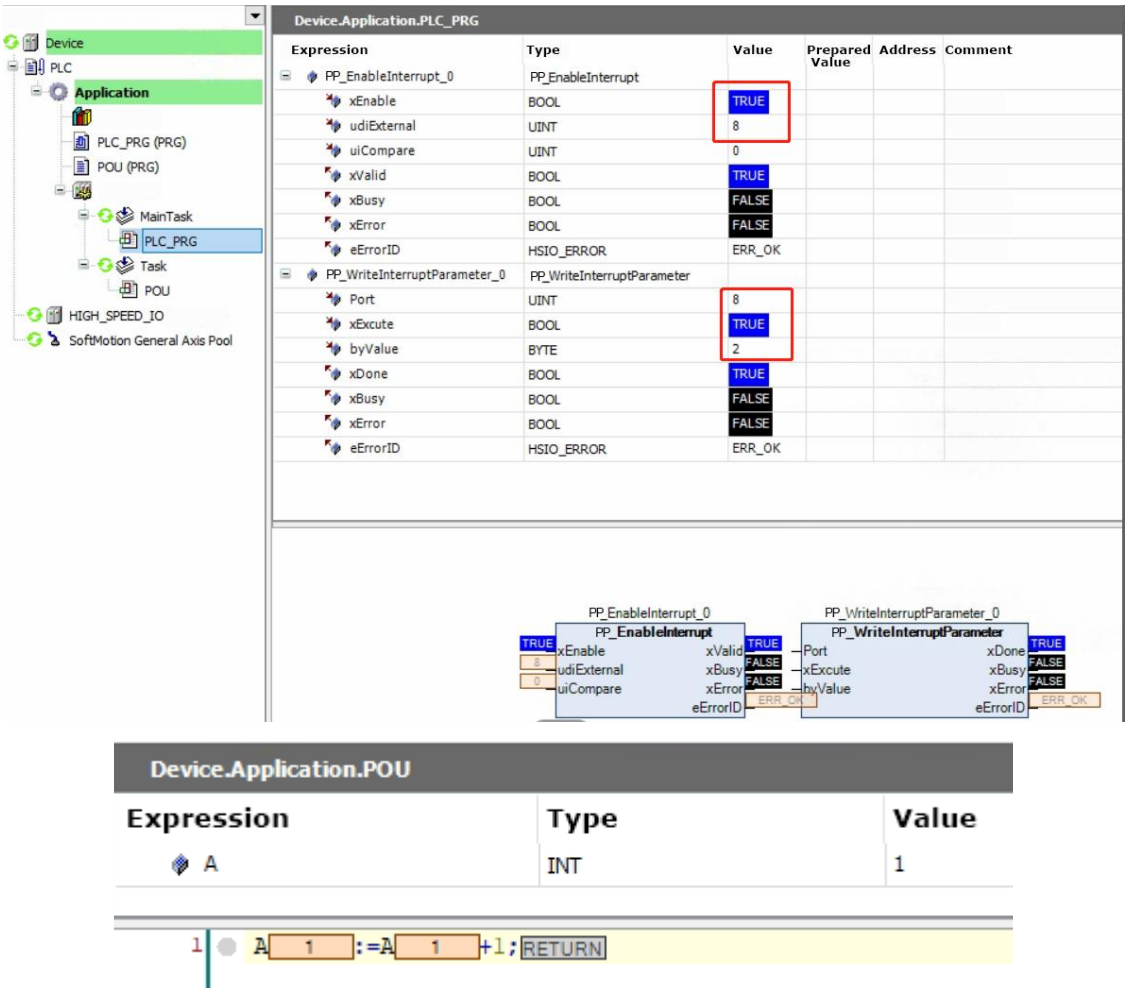
Дважды щелкните по “Task” (Задача), во всплывающем окне выберите тип "External" (Внешние). Клемма X будет использоваться для внешних прерываний. Также можно установить приоритет событий внешнего прерывания.



Пример 1: Работа с командами [PP_EnnableInterrupt][PP_WriteInterruptParameter] Установите X3 в качестве входа внешнего прерывания, возьмите его сигнал двустороннего фронта, который можно настроить через интерфейс настройки конфигурации параметров аппаратного обеспечения или по команде PP_WriteInterruptParameter. Выполните команду self adding 1 (автоматическое добавление 1) в программе POU под другой задачей (сконфигурированной как внешняя, X3_TRIG) после подачи сигнала фронта импульса входа X3. На рисунке ниже показаны конфигурация параметров и команды.





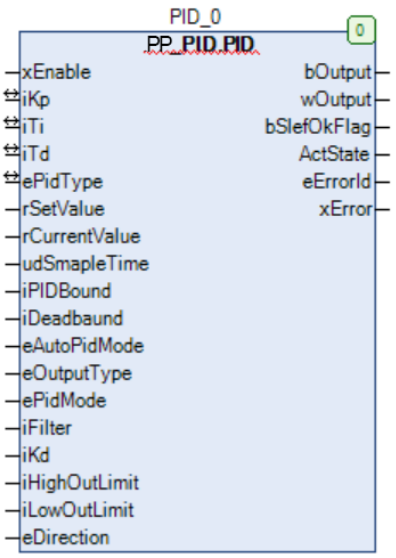


2-4. Команды ПИД-регулирования

※	Функциональный блок PID защищен авторским правом, и его разрешено применять только на ПЛК PROMPOWER! Его нельзя подделать!

2-4-1. Формат команд

Функциональный блок ПИД-регулирования [PID]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_PID		

Команда	Название	Графическое представление	Представление на языке ST
PP_PID.PID	Команда ПИД-регулирования		PP_PID.PID(xEnable:= , iKp:= , iTi:= , iTd:= , ePidType:= , rSetValue:= , rCurrentValue:= , udSmapleTime:= , iPIDBound:= , iDeadbaund:= , eAutoPidMode:= , eOutputType:= , ePidMode:= , iFilter:= , iKd:= , iHighOutLimit:= , iLowOutLimit:= , eDirection:= , bOutput=> , wOutput=> , bSlefOkFlag=> , ActState=> , eErrorId=> , xError=>);

2-4-2. Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xEnable	Включение	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
iKp	Пропорциональное усиление	INT	0~32767	0	Пропорциональное усиление
iTi	Интегральное время	INT	0~32767	0	Интегральное время (*100 мс)
iTd	Дифференциа	INT	0~32767	0	Дифференциальное время (*10

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
	льное время				мс)
ePidType	Режим настройки	PIDTYPE	-	-	0: Ручной режим 1: Режим автоматической регулировки колебаний
rSetValue	Целевое значение	REAL	Диапазон данных	0	*Значение уставки
rCurrentValue	Значение обратной связи	REAL	Диапазон данных	0	*Текущее значение обратной связи
udSmapleTime	Период выборки	UDINT	Диапазон данных	1000	* Период выборки (мс) Рекомендуемое значение: от 1000 до 2000
iPIDBound	Рабочий диапазон	INT	0~32767	0	Рабочий диапазон ПИД-регулятора
iDeadbaund	Зона нечувствительности регулирования	INT	0~32767	0	Зона нечувствительности ПИД-регулирования
eAutoPidMode	Режим автоматического ПИД-регулирования	PID CONTRLMODE	-	-	Режим автоматического ПИД-регулирования 0: ПИД-регулирование 1: ПИ-регулирование 2: П-регулирование
eOutputType	Режим вывода	OUTPUTTYPE	-	-	0: Выход канала ввода-вывода

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					1: Цифровой выход
ePidMode	Режим управления	PIDMODE	-	-	Режим управления: общий режим, расширенный режим
iFilter	Константа входной фильтрации	INT	0~100	0	Константа входной фильтрации, доступна в расширенном режиме в диапазоне от 0 до 100
iKd	Дифференциальное усиление	INT	-	50	Дифференциальное усиление; значение по умолчанию 50, доступно в расширенном режиме
iHighOutLimit	Уставка верхнего предела выходного сигнала	INT	0~32767	0	Уставка верхнего предела выходного сигнала (по умолчанию: 4095)
iLowOutLimit	Уставка нижнего предела выходного сигнала	INT	0~32767	0	Уставка нижнего предела выходного сигнала (по умолчанию: 0)
eDirection	Обратное действие/прямое действие	DIRECTIONPARAMETER	-	-	ATHWART: обратное действие (нагрев); POSITIVE: Прямое действие

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					(охлаждение)
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bOutpu	Булев выход	BOOL	TRUE/FALSE	FALSE	Выход ввода/вывода имеет состояние TRUE
wOutput	Цифровой выход	INT	TRUE/FALSE	FALSE	Цифровой выход имеет состояние TRUE
bSlefOkFlag	Признак завершения автоматической регулировки	BOOL	TRUE/FALSE	FALSE	Завершение автоматической регулировки имеет состояние TRUE
ActState	Текущий режим	AUTOPIDSTATE	-	-	Текущий режим
eErrorId	Код ошибки	ERRID_PID	-	-	В нормальном режиме значение равно 0. Если возникает исключение, выдается код ошибки.
xError	Признак ошибки	BOOL	TRUE/FALSE	FALSE	Если возникает исключение, состояние будет TRUE.

2-4-3. Описание функции

- Направление действия

Прямое действие: Действие, при котором выходное значение MV операции увеличивается с увеличением измеренного значения PV, и которое обычно используется для управления охлаждением.

Обратное действие: Действие, при котором выходное значение MV

уменьшается с увеличением измеренного значения PV, и которое обычно используется для управления нагревом.

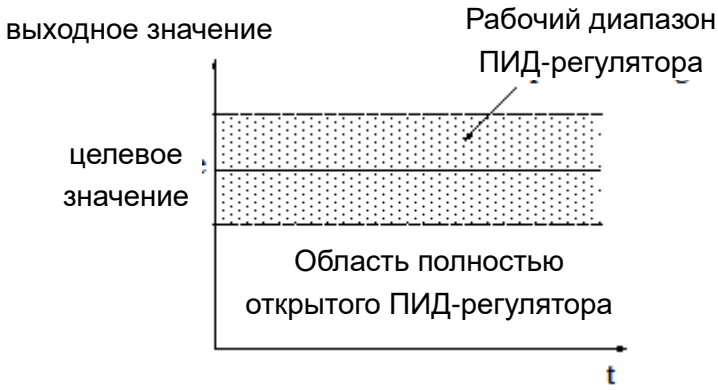
- Настройка расширенного режима: Включите расширенный режим, после чего пользователи смогут настроить такие параметры, как время фильтрации, дифференциальное усиление, верхний и нижний пределы выходного сигнала.

- Период выборки

Система считывает текущее значение через определенный промежуток времени и сравнивает его с выходным значением. Этот промежуток времени является периодом выборки T. Если выводится DA, T не имеет ограничений. Если с порта поступает сигнал, T должен быть больше 1 цикла сканирования программы ПЛК. Значение T должно быть в диапазоне 100-1000 циклов сканирования ПЛК.

- Рабочий диапазон ПИД-регулятора

Когда система работает, ее ПИД-регулятор полностью открыт, благодаря чему к целевому значению можно приблизиться на максимальной скорости (по умолчанию 4095). После достижения рабочего диапазона ПИД-регулятора, параметры Kp, TI и TD начинают брать управление на себя. Это показано на рисунке ниже:



Если пользователь не знает уставки параметров ПИД-регулятора, можно включить режим автоматической регулировки, в котором система автоматически определяет оптимальные параметры управления (пропорциональное усиление Kp, время интегрирования Ti, время дифференцирования TD).

- Объекты управления в режиме автоматической регулировки: температура и давление. Неподходящие объекты управления: уровень жидкости, расход, и т.д.

- Автоматическая регулировка — это процесс извлечения параметров ПИД-регулятора. Иногда в режиме автоматической регулировки не удается найти оптимальные параметры за один раз, поэтому автоматическую регулировку

придется выполнить несколько раз. Колебания во время регулировки допускаются. После завершения автоматической регулировки и определения оптимальных параметров нужно переключиться в режим ручного ПИД-регулирования. Если объект управления ведет себя неустойчиво во время ручного ПИД-регулирования, и им нельзя управлять при постоянном целевом значении, причина может быть в неправильной регулировке параметров, поэтому необходимо повторно отрегулировать параметры ПИД-регулирования для обеспечения устойчивости управления.

- В начале автоматического регулирования по методу критических колебаний пользователь должен заранее задать цикл ПИД-регулирования (период выборки). Контрольное значение: Как правило, системы с медленным откликом можно настроить на 1000 мс, а системы с быстрым откликом — на 10-100 мс.

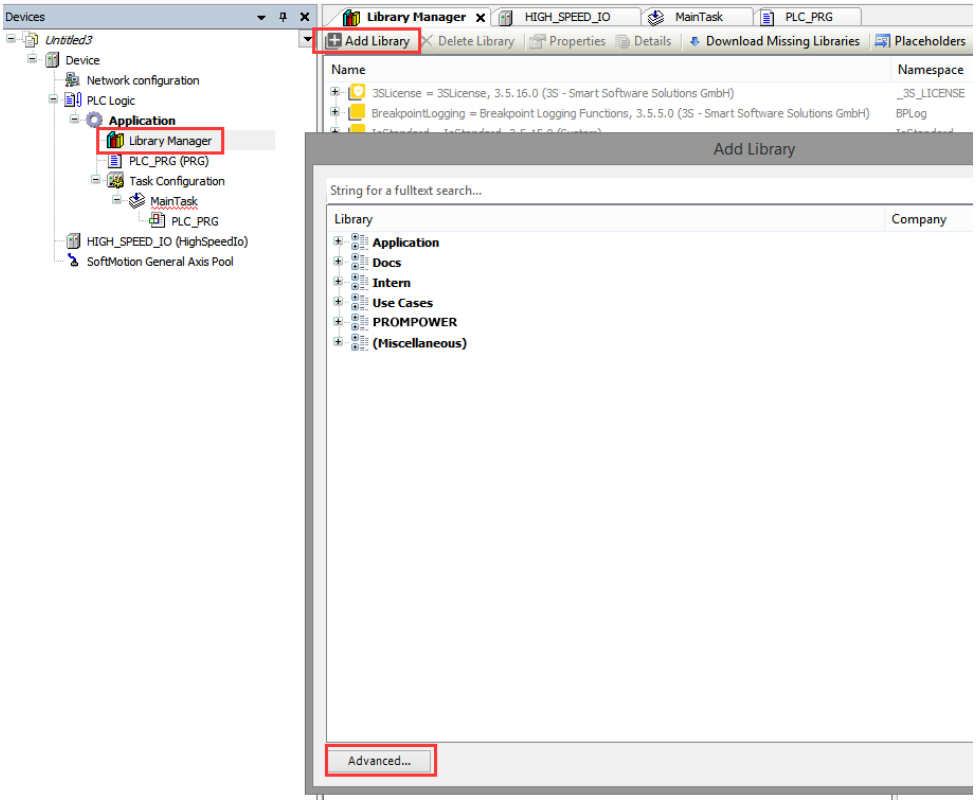
- Благодаря использованию метода критических колебаний для автоматической регулировки систему можно запускать из любого состояния. Для объектов терморегулирования текущая измеренная температура не обязательно должна соответствовать температуре окружающей среды. Она может быть ниже или выше заданной температуры.

- В режиме автоматической регулировки необходимо установить "ePidType" в состояние TRUE и выбрать режим pid (p, pi, pid) (ПИД (П-, ПИ-, ПИД-регулирование). После завершения настройки, переключите состояние xEnable на TRUE, ПЛК перейдет в состояние автоматической регулировки, а состоянием ActState будет AutoPidBusy. Когда "ActState" принимает состояние AutoPidDone, это означает, что автоматическая регулировка выполнена успешно, а "bSlefOkFlag" находится в состоянии true. После выполнения цикла состояние ПИД-регулятора автоматически изменится на CommonPID, при этом ПИД-регулятор управляет системой управления в ручном режиме с параметрами автоматической регулировки.

- Переключение выходных значений. Если выходное значение, необходимое пользователю, является цифровой величиной, установите значение eOutputType на 1. Если выходное значение, необходимое пользователю, является изменяемой величиной, установите значение eOutputType на 0.

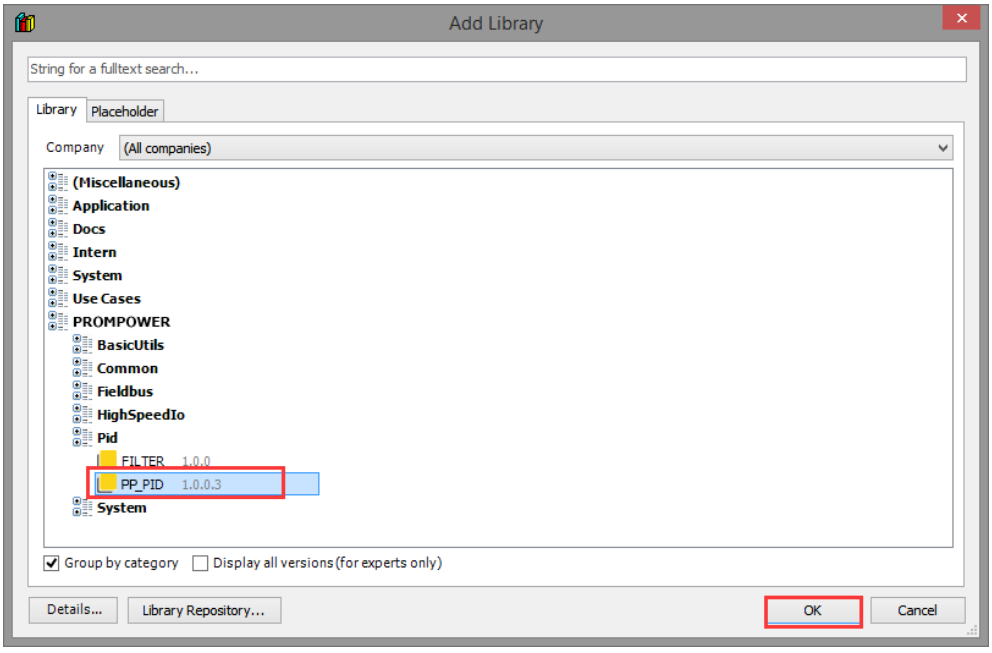
2-4-4. Пример использования

Дважды щелкните Library Manager - Add Library - Advanced (Диспетчер библиотек - Добавить библиотеку - Дополнительно).



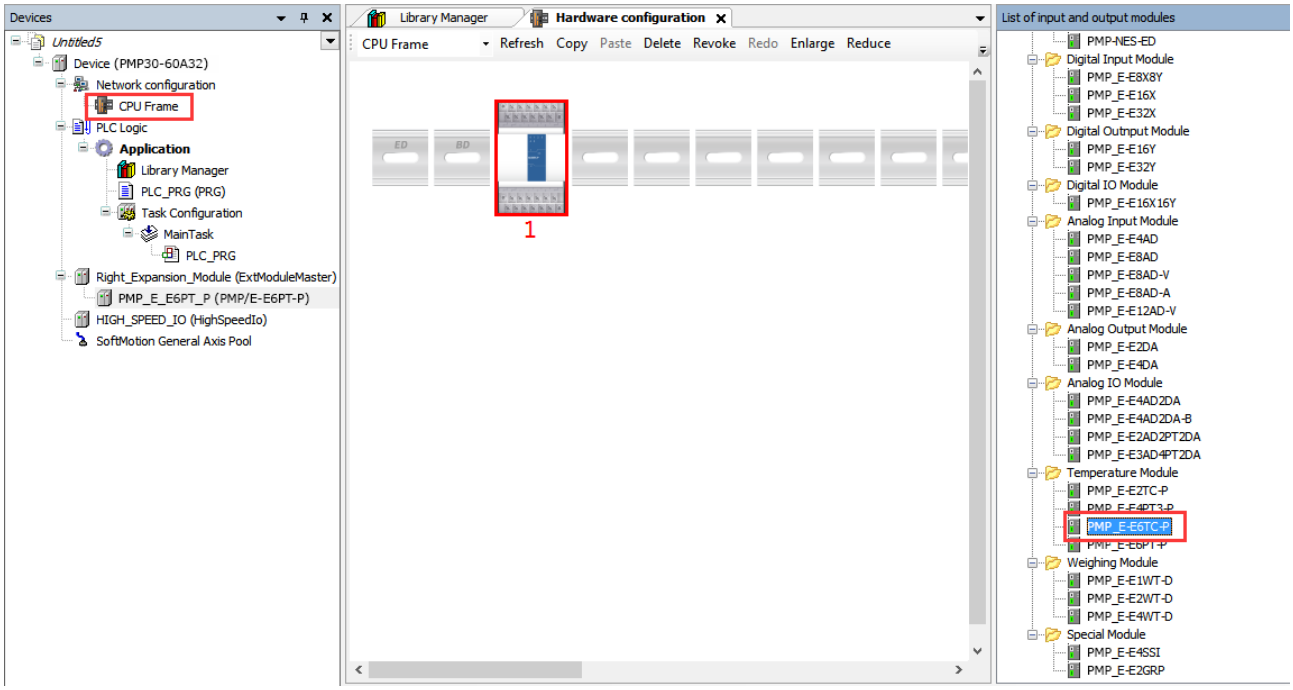
※	Функциональный блок PID защищен авторским правом, и его разрешено применять только на ПЛК PROMPOWER! Его нельзя подделать!

Добавьте PP_PID.

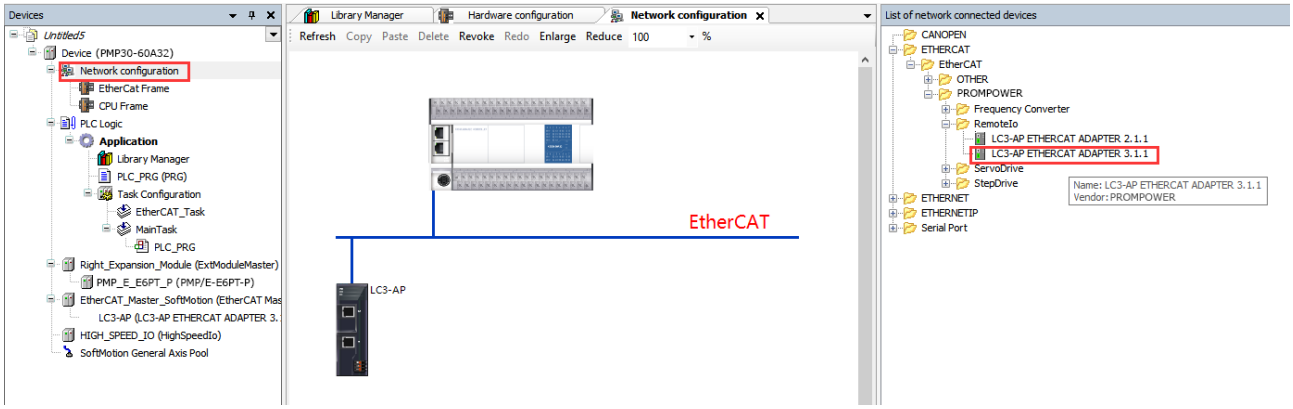


Добавьте модуль температурного расширения.

- Дважды щелкните по CPU Frame (Кадр ЦП), выберите положение 1, после чего щелкните два раза по PMP-E6TC-P.

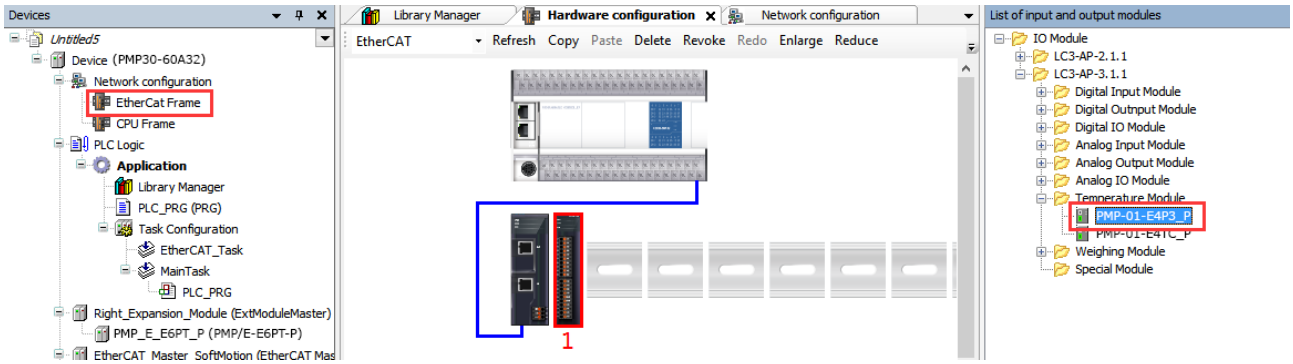


- Модуль дистанционного ввода/вывода: Дважды щелкните по Network configuration (Конфигурация сети), щелкните два раза по пункту PMP-SEC Ethercat adapter 3.1.1.

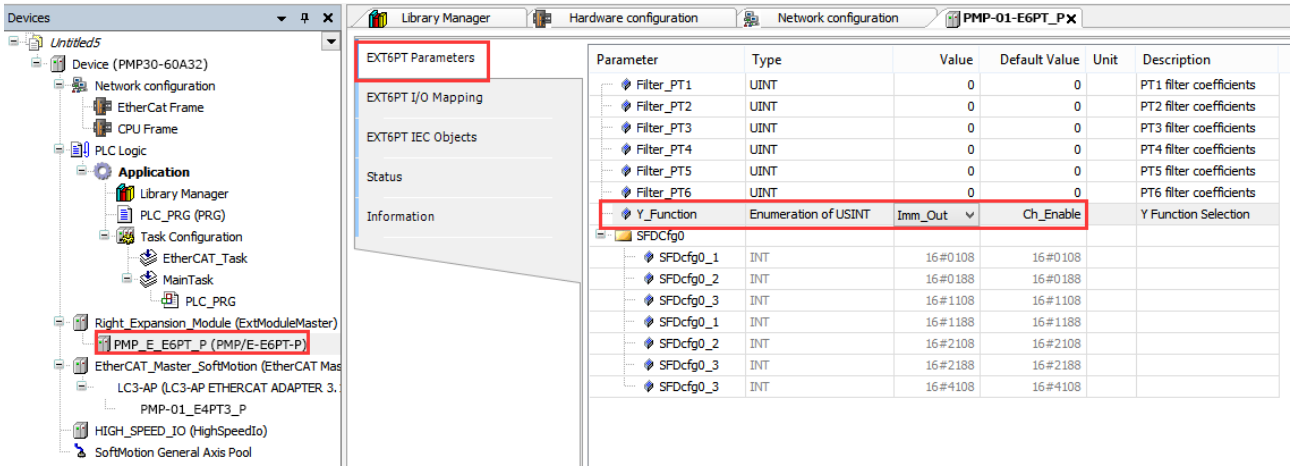


Примечание: После этого будут выбраны только PMP-SEC V3.1 и команда EC_From TO.

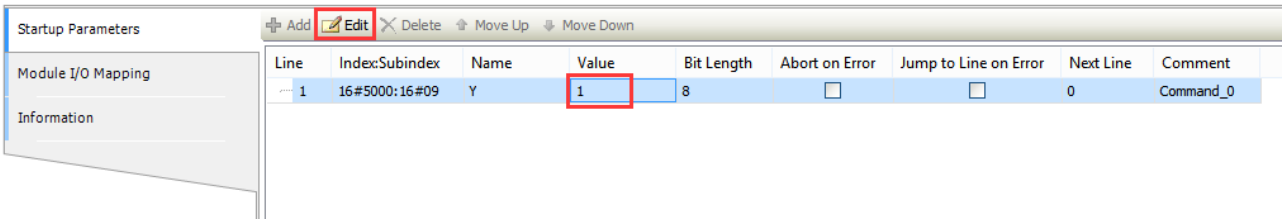
Дважды щелкните по EtherCat Frame (Кадр EtherCat), выберите положение 1, после чего щелкните два раза по PMP-01-E4PT3-P.



Если требуется реле управления количеством переключений, модуль на основном корпусе должен настроить параметр Y_Function на значение 'Imm Out'.

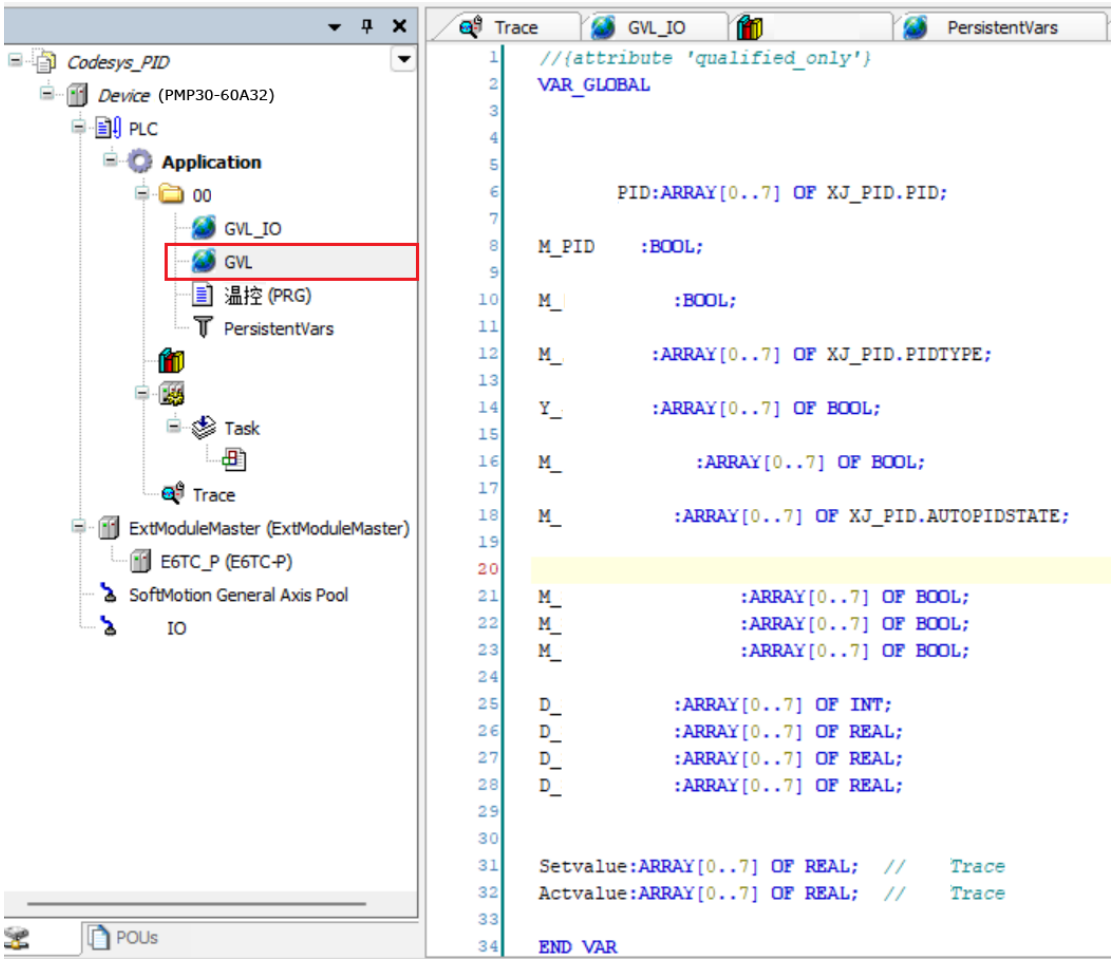


Модуль дистанционного ввода/вывода должен добавить соответствующий параметр запуска к значению по умолчанию 1.

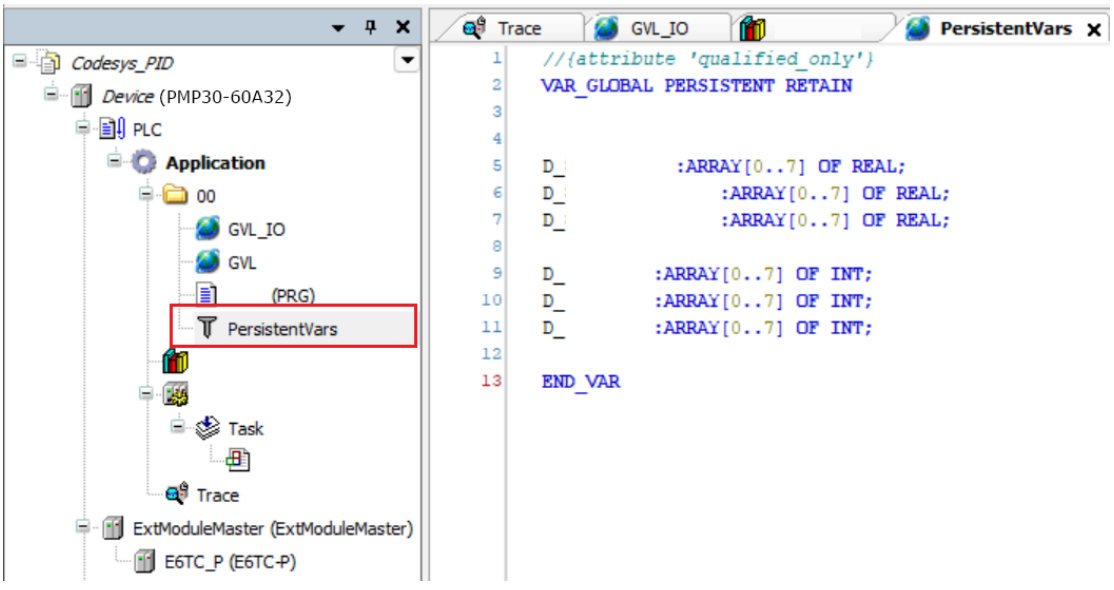


После изменения параметров нужно повторно включить питание модуля, чтобы параметры вступили в силу.

Задание требуемой глобальной переменной (Global).



1) Задайте требуемые переменные сохранения параметров при отключении питания, чтобы данные не были потеряны после перезапуска ПЛК.



2) Создание программы ПИД-регулирования.

1. Преобразуйте фактическое значение температуры, полученное от температурного модуля, в правильное значение температуры на дисплее. Например, если переданные данные равны 289, их нужно разделить на 10, чтобы преобразовать в число с плавающей точкой 28.9, соответствующее 28,9°. Это в дальнейшем облегчит персоналу проведение обслуживания. На объекте могут возникать различные условия, требующие дополнительной компенсации, поэтому значение компенсации добавляется, чтобы получить окончательное значение. Используйте это окончательное значение для регулировки ПИД-регулятора.

```
FOR a:=0 TO 2 BY 1 DO

    D_          [a] :=TO_REAL(D_          [a]) / 10;
    D_          [a] := D_          [a] + D_          [a];

    Setvalue[a]:=D_          [a];  //    Trace
    Actvalue[a]:=D_          [a];  //    Trace

END_FOR
```

2. Сигнализация температуры: При слишком высокой или слишком низкой фактической температуре, а также при обрыве цепи, должен поступать сигнал тревоги.

```
FOR b:=0 TO 2 BY 1 DO

    IF D_          [b] < D_          [b] THEN
        M_              [b]:= TRUE;
    ELSE
        M_              [b]:= FALSE;
    END_IF


    IF D_          [b] > D_          [b] THEN
        M_              [b]:= TRUE;
    ELSE
        M_              [b]:= FALSE;
    END_IF


    IF D_          [b] = -0.1 THEN
        M_              [b]:= TRUE;
    ELSE
        M_              [b]:= FALSE;
    END_IF

END_FOR
```

3. Используется одна клавиша Self-tuning (Автоматическая регулировка), когда активна опция "M_ Self-tuning switch" (M_Переключатель

автоматической регулировки) (она может быть всегда активна, либо установлен запрет на ее постоянное включение), все функциональные блоки ПИД-регулирования находятся в режиме автоматической регулировки, что удобно для пакетной отладки.

```
IF M_          THEN
    M_        [0]:= 1;
    M_        [1]:= 1;
    M_        [2]:= 1;
END_IF
```

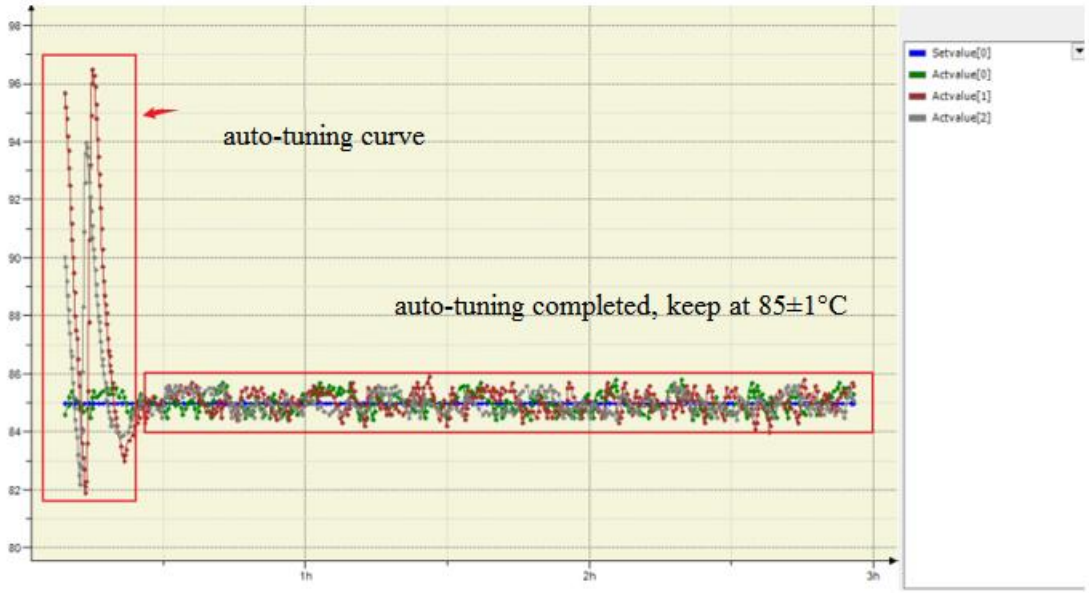
4. Чтобы включить функциональный блок PID (ПИД-регулирование), просто выберите ON M_PID. (Примечание: Перед изменением параметров ПИД-регулирования выключите ПИД-регулятор!)

```
IF NOT M_              [0] THEN
        _PID[0](
    xEnable:= M_PID    ,
    iKp:= D_        [0],
    iTi:= D_        [0],
    iTd:= D_        [0],
    rSetValue:=  D_          [0],
    rCurrentValue:= D_          [0],
    ePidType:= M_        [0],
    udSmapleTime:= 1000,
    iPIDBound:= ,
    iDeadbaund:= ,
    eAutoPidMode:= 0,
    eOutputType:= XJ_PID.OUTPUTTYPE.BoolType,
    ePidMode:= 0,
    iFilter:= ,
    iKd:= ,
    iHighOutLimit:= ,
    iLowOutLimit:= ,
    eDirection:= 0,
    bOutput=> Y_        [0],
    wOutput=> ,
    bSlefOkFlag=> M_            [0] ,
    ActState=> M_          [0],
    eErrorId=> ,
    xError=>  );
END_IF
```

5. Время автоматической регулировки ПИД-регулятора составляет 15-25 минут. После завершения автоматической регулировки всем параметрам iKp, iTi и iTd будут присвоены значения. ePidype автоматически переключится на "ManualType", признак bSelfOKFlag будет включен, а ActState отобразит "AutoPIDrea", чтобы облегчить последующую автоматическую регулировку.

```
(*****PID*****)
PID[0](
    xEnable TRUE := M_PID     TRUE,
    iKp 31929 := D_           31929,
    iTi 1636 := D_            1636,
    iTd 4091 := D             4091,
    rSetValue 85 :=  D_       [0] 85,
    rCurrentValue 91.2 := D_  [0] 91.2,
    ePidType 0 := M_          [0] ManulType,
    udSmapleTime 1000 := 1000,
    iPIDBound:= ,
    iDeadbaund:= ,
    eAutoPidMode PIDMode := 0,
    eOutputType BoolType := PP_PID.OUTPUTTYPE.BoolType,
    ePidMode Common := 0,
    iFilter:= ,
    iKd:= ,
    iHighOutLimit:= ,
    iLowOutLimit:= ,
    eDirection Athwart := 0,
    bOutput FALSE => Y_       [0] FALSE,
    wOutput=> ,
    bSlefOkFlag TRUE => M_    [0] TRUE ,
    ActState AutoPIDRea => M_ [0] AutoPIDRea,
    eErrorId=> ,
    xError=>  ,
    PreI_Item 0 => PreI_Item[0] 0,
    PreD_Item 293 => PreD_Item[0] 293,
    PrePID_Item 0 => PrePID_Item[0] 0,
    Overshoot 7 => Overshoot[0] 7)
    ;
```

10. Точность ПИД-регулирования составит примерно 2% от уставки температуры. После фактического измерения уставка температуры составит 85 °C. При ПИД-регулировке температура поддерживается на уровне 85±1 °C.



2-5. Библиотека системы

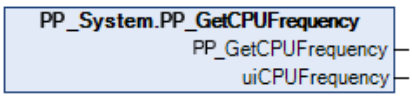
2-5-1. Обзор функции

Пользователи могут считывать или записывать системные параметры контроллера с помощью команд.

2-5-2. Введение в функциональные блоки

2-5-2-1. Основная частота ЦП [PP_GetCPUFrequency]

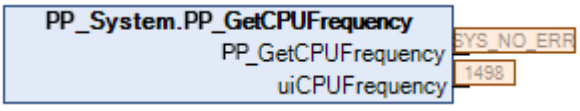
Основная частота ЦП [PP_GetCPUFrequency]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetCPUFrequency	Основная частота ЦП		PP_GetCPUFrequency(uiCPUFrequency=>);

(1) Выходные переменные

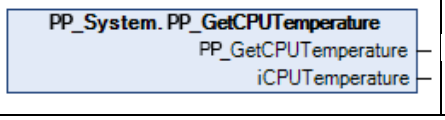
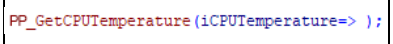
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetCPUFrequency	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
CPUFrequency	Основная частота ЦП	UINT	-	0	M — единица измерения

Результаты выполнения:



2-5-2-2. Температура ЦП [PP_GetCPUTemperature]

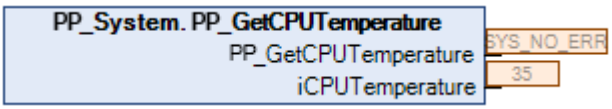
Температура ЦП [PP_GetCPUTemperature]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetCPUTemperature	Температура ЦП		

(1) Выходные переменные

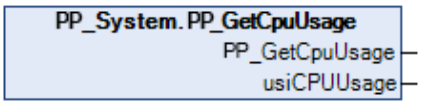
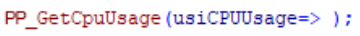
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetCPUTemperature	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
CPUTemperature	Температура ЦП	INT	-	0	Температура ЦП

Результаты выполнения:



2-5-2-3. Загрузка ЦП [PP_GetCPUUsage]

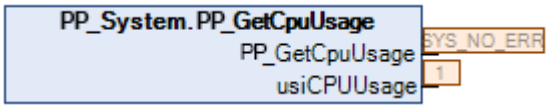
Загрузка ЦП [PP_GetCPUUsage]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetCPUUsage	Загрузка ЦП		

(1) Выходные переменные

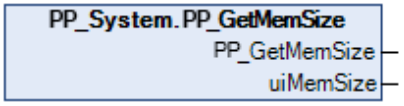
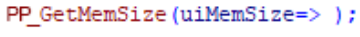
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetCPUUsage	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
CPUUsage	Загрузка ЦП	USINT	-	0	Процент загрузки (0-100%)

Результаты выполнения:



2-5-2-4. Объем памяти [PP_GetMemSize]

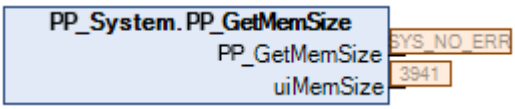
Объем памяти [PP_GetMemSize]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetMemSize	Объем памяти		

(1) Выходные переменные

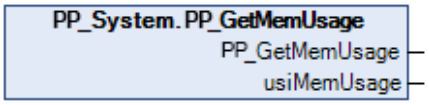
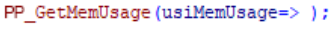
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetMemSize	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
MemSize	Объем памяти	UINT	-	0	Мб — единица измерения.

Результаты выполнения:



2-5-2-5. Загрузка памяти [PP_GetMemUsage]

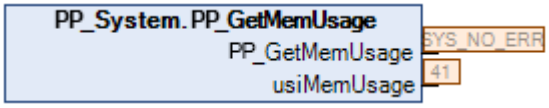
Загрузка памяти [PP_GetMemUsage]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetMemUsage	Загрузка памяти		

(1) Выходные переменные

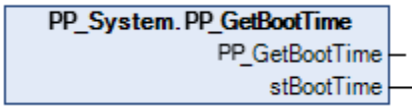
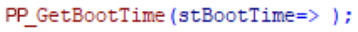
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetMemUsage	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
MemUsage	Загрузка памяти	USINT	-	0	Процент загрузки (0-100%)

Результаты выполнения:



2-5-2-6. Время пребывания во включенном состоянии [PP_GetBootTime]

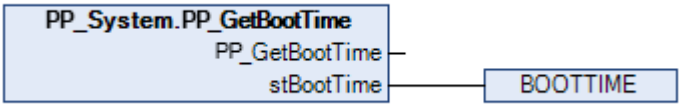
Время пребывания во включенном состоянии [PP_GetBootTime]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetBootTime	Время пребывания во включенном состоянии		

(1) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetBootTime	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
BootTime	Время пребывания во включенном состоянии	SYS_PP_TIME	-	0	Структурная морфология

Результаты выполнения:




```
PROGRAM POU
VAR

    PP_GetNetInfo_0: PP_System.PP_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: PP_SYSTEM.SYS_PP_TIME;
    AA: BOOL;
    NETID: BYTE;
    PP_SetNetInfo_0: PP_System.PP_SetNetInfo;
    runtimeversion: DWORD;
    STTIME: PP_SYSTEM.SYS_PP_TIME;
    timezone: INT;
    SetTIME: PP_System.SYS_PP_TIME;
    itimezone: INT;
END_VAR
```

BOOTTIME	PP_SYSTEM.SYS_PP_TIME					
Year	UINT	2022				
Month	UINT	11				
Day	UINT	19				
Hour	UINT	1				
Minute	UINT	55				
Second	UINT	43				
Milliseconds	UINT	0				
DayOfWeek	UINT	6				

2-5-2-7. Название ПЛК [PP_GetPLCName]

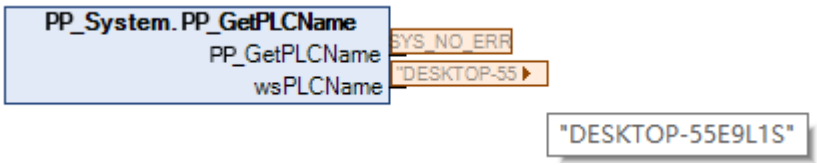
Название ПЛК [PP_GetPLCName]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetPLCName	Название ПЛК	PP_System.PP_GetPLCName PP_GetPLCName wsPLCName	`PP_GetPLCName(wsPLCName=> );`

(1) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetPLCName	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
PLCName	Название ПЛК	WSTRING	-	0	Название ПЛК

Результаты выполнения:



2-5-2-8. Идентификатор ПЛК [PP_GetPLCID]

Идентификатор ПЛК [PP_GetPLCID]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

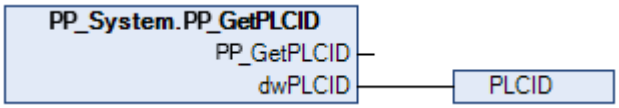
Команда	Название	Графическое представление	Представление на языке ST
PP_GetPLCID	Идентификатор ПЛК	PP_System.PP_GetPLCID PP_GetPLCID dwPLCID	PP_GetPLCID(dwPLCID=>);

(1) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetPLCID	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
PLCID	Идентификатор ПЛК	DWORD	-	0	Информация об идентификаторе ПЛК,

					шестнадцатеричный формат

Ниже приведен результат выполнения. Информация об идентификаторе ПЛК отображается в шестнадцатеричном формате:




```
PROGRAM POU
VAR

    XJ_GetNetInfo_0: XJ_System.XJ_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: XJ_SYSTEM.SYS_XJ_TIME;
    AA: BOOL;
    NETID: BYTE;
    XJ_SetNetInfo_0: XJ_System.XJ_SetNetInfo;
    runtimeversion: DWORD;
    STTIME: XJ_SYSTEM.SYS_XJ_TIME;
    timezone: INT;
    SetTIME: XJ_System.SYS_XJ_TIME;
    itimezone: INT;
END_VAR
```

Expression	Type	Value	Pr
PP_GetNetInfo_0	PP_System.PP_GetNetInfo		
PLCID	DWORD	16#17070200	
BOOTTIME	PP_SYSTEM.SYS_PP_TIME		
AA	BOOL	FALSE	
NETID	BYTE	16#00	
PP_SetNetInfo_0	PP_System.PP_SetNetInfo		
runtimeversion	DWORD	16#03050F28	
STTIME	PP_SYSTEM.SYS_PP_TIME		
timezone	INT	16#0000	
SetTIME	PP_System.SYS_PP_TIME		
itimezone	INT	16#0000	

2-5-2-9. Версия прошивки [PP_GetPLCVersion]

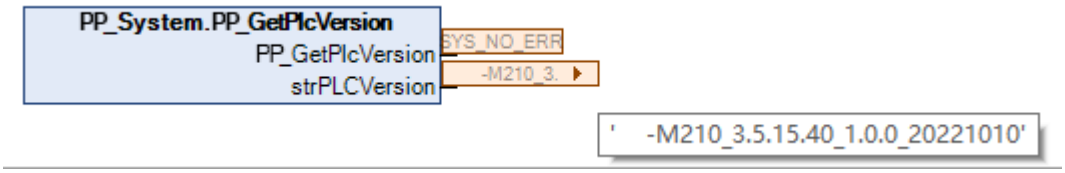
Версия прошивки [PP_GetPLCVersion]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetPLCVersion	Версия прошивки ПЛК	PP_System.PP_GetPlcVersion PP_GetPlcVersion strPLCVersion	PP_GetPlcVersion(strPLCVersion=>);

(1) Выходные переменные

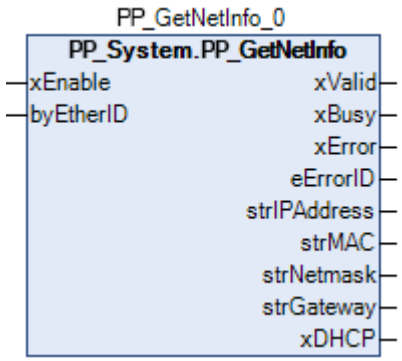
Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetPLCVersion	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
PLCVersion	Версия прошивки ПЛК	STRING	-	0	Версия прошивки ПЛК

Результат выполнения:



2-5-2-10. Получение информации о сетевом порте [PP_GetNetInfo]

Получение информации о сетевом порте [PP_GetNetInfo]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetNetInfo	Получение информации о сетевом порте		PP_GetNetInfo_0(xEnable:= , byEtherID:= , xValid=> , xBusy=> , xError=> , eErrorID=> , strIPAddress=> , strMAC=> , strNetmask=> , strGateway=> , xDHCP=>);

(1) Входные переменные

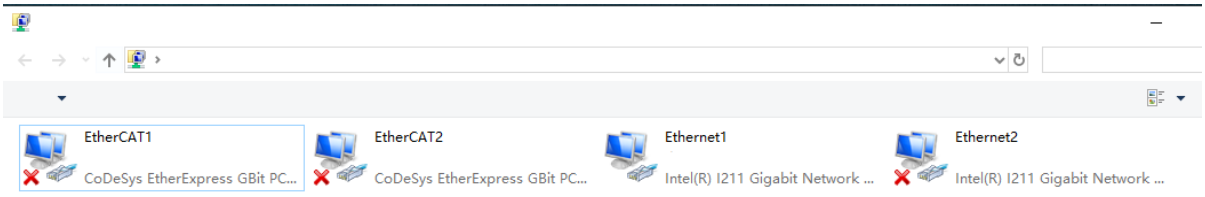
Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Enable	Включение	BOOL	TRUE,FALSE	FALSE	Включение
EtherID	Идентификатор сетевого порта	BYTE	-	0	Номер соответствующего сетевого порта на экране дисплея

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Valid	Допустимые выходные данные	BOOL	TRUE,FALSE	FALSE	Допустимые выходные данные
Busy	В процессе выполнения	BOOL	TRUE,FALSE	FALSE	В процессе выполнения
Error	Признак ошибки	BOOL	TRUE,FALSE	FALSE	Признак ошибки
ErrorID	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
IPAddress	IP-адрес	STRING	-	0	IP-адрес
MAC	MAC-адрес	STRING	-	0	MAC-адрес

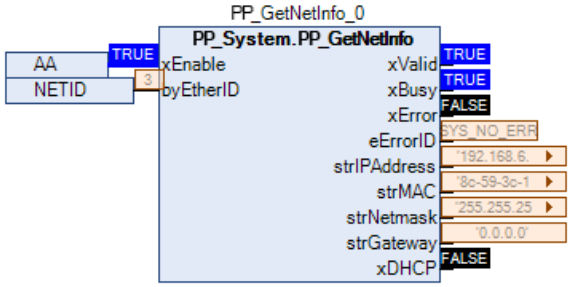
Netmask	Адрес подсети	STRING	-	0	Адрес подсети
Gateway	Информация о шлюзе	STRING	-	0	Информация о шлюзе
DHCP	Свойства DHCP	BOOL	TRUE，FALSE	FALSE	Автоматическое получение данных о свойствах

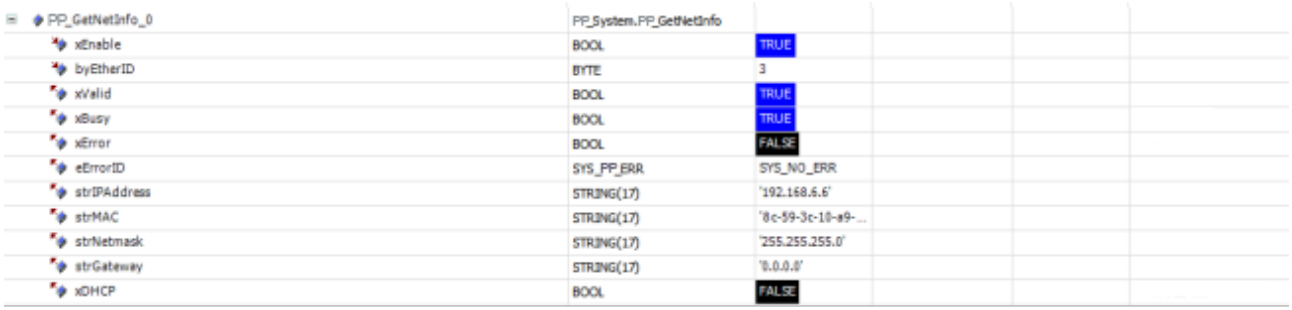
Идентификаторы сетевых портов промышленного компьютера расположены в сетевом адаптере в порядке 0-3 слева направо, как показано на рисунке Ethernet1, и записаны в пункте 2.




```
PROGRAM POU
VAR

    PP_GetNetInfo_0: PP_System.PP_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: PP_SYSTEM.SYS_PP_TIME;
    AA: BOOL;
    NETID: BYTE;
END_VAR
```

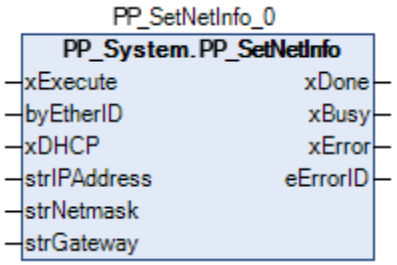
Результат выполнения:





2-5-2-11. Настройка информации о сетевом порте [PP_SetNetInfo]

Настройка информации о сетевом порте [PP_SetNetInfo]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_SetNetInfo	Настройка информации о сетевом порте		PP_SetNetInfo_0(xExecute:= , byEtherID:= , xDHCP:= , strIPAddress:= , strNetmask:= , strGateway:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

(1) Входные переменные

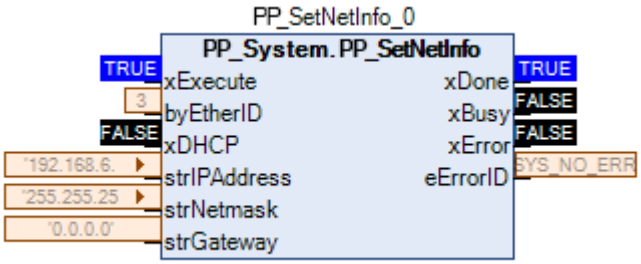
Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Excute	Выполнить	BOOL	TRUE,FALSE	FALSE	Выполнение функционального блока при нарастающем фронте
EtherID	Идентификатор сетевого порта	BYTE	-	0	Соответствующий сетевой порт на экране дисплея
DPCH	Свойства DPCH	BOOL	TRUE,FALSE	FALSE	Автоматическое получение данных о свойствах
IPAddress	IP-адрес	STRING	-	0	IP-адрес

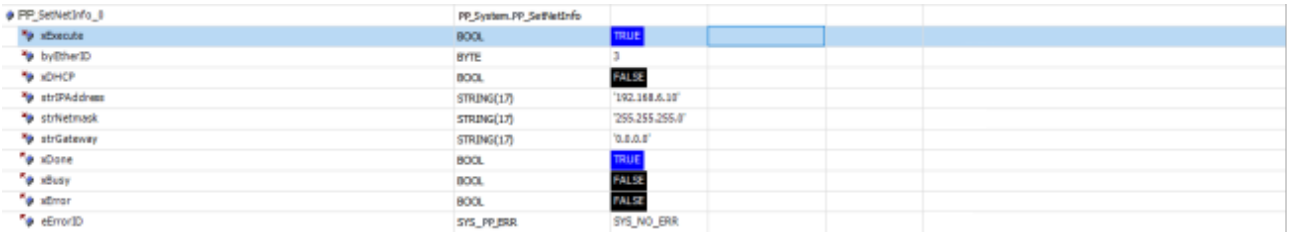
Netmask	Адрес подсети	STRING	-	0	Адрес подсети
Gateway	Информация о шлюзе	STRING	-	0	Информация о шлюзе

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнение завершено	BOOL	TRUE,FALSE	FALSE	Выполнение завершено
Busy	В процессе выполнения	BOOL	TRUE,FALSE	FALSE	В процессе выполнения
Error	Признак ошибки	BOOL	TRUE,FALSE	FALSE	Признак ошибки
ErrorID	Код ошибки	SYS_PP_ERR	-	0	Код ошибки

Изменения вступят в силу только после выключения питания и перезапуска ПЛК или RTE.

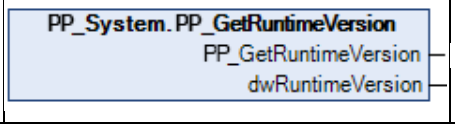




2-5-2-12. Исполняемая версия [PP_GetRuntimeVersion]

Исполняемая версия [PP_GetRuntimeVersion]			
Условие выполнения	-	Совместимые модели	PMP3xx

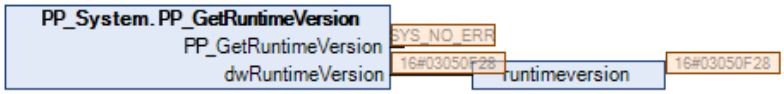
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetRuntimeVersion	Исполняемая версия		

(1) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetRuntimeVersion	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
RuntimeVersion	Исполняемая версия	DWORD	-	0	Исполняемая версия

Исполняемая версия представлена в шестнадцатеричном формате.





2-5-2-13. Версия BOOT [PP_GetBootVersion]

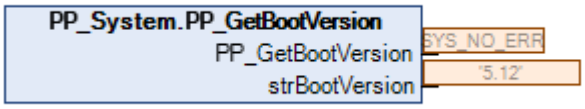
Версия BOOT [PP_GetBootVersion]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetBootVersion	Версия BOOT	PP_System.PP_GetBootVersion PP_GetBootVersion strBootVersion	PP_GetBootVersion(strBootVersion=>);

(1) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetBootVersion	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
BootVersion	Версия BOOT	STRNG	-	0	Версия BOOT

Результаты выполнения:



2-5-2-14. Получение данных о дате и времени [PP_GetTime]

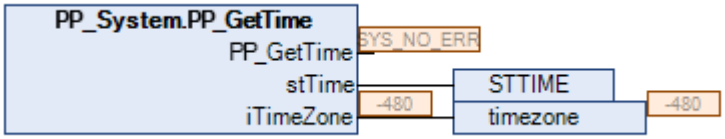
Получение данных о дате и времени [PP_GetTime]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_GetTime	Получение данных о дате и времени	PP_System.PP_GetTime PP_GetTime stTime iTimeZone	PP_GetTime(stTime=> , iTimeZone=>);

(1) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
GetTime	Код ошибки	SYS_PP_ERR	-	0	Код ошибки
Time	Время	SYS_PP_TIME	-	0	Структурная морфология
TimeZone	Получение данных о часовом поясе	INT	-	-	Единица измерения — минута.

Результаты выполнения:



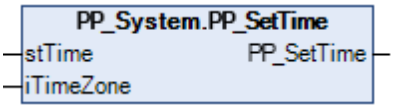

```
PROGRAM POU
VAR

    PP_GetNetInfo_0: PP_System.PP_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: PP_SYSTEM.SYS_PP_TIME;
    AA: BOOL;
    NETID: BYTE;
    PP_SetNetInfo_0: PP_System.PP_SetNetInfo;
    runtimeversion: DWORD;
    STTIME: PP_SYSTEM.SYS_PP_TIME;
    timezone: INT;
    SetTIME: PP_System.SYS_PP_TIME;
    itimezone: INT;
END_VAR
```

STTIME	PP_SYSTEM.SYS_PP_TIME					
Year	UINT	2022				
Month	UINT	11				
Day	UINT	18				
Hour	UINT	13				
Minute	UINT	9				
Second	UINT	8				
Milliseconds	UINT	909				
DayOfWeek	UINT	5				

2-5-2-15. Настройка даты и времени [PP_SetTime]

Настройка даты и времени [PP_SetTime]			
Условие	Запуск фронтом сигнала	Совместимые	PMP3xx

выполнения		модели	
Прошивка	версия 1.1.0 и более поздние версии	Программное обеспечение	-
Библиотека	PP_System		

Команда	Название	Графическое представление	Представление на языке ST
PP_SetTime	Настройка даты и времени		`PP_SetTime(stTime:= , iTimeZone:= );`

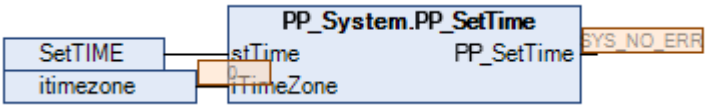
(1) Входные переменные

Входные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Time	Настройка времени	SYS_PP_TIME	-	0	Настройка времени
TimeZone	Настройка часового пояса	INT	-	-480	Единица измерения — минута.

(2) Выходные переменные

Выходные переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
SetTime	Код ошибки	SYS_PP_ERR	-	0	Код ошибки

Изменения времени вступают в силу сразу после записи значений. Из-за отсутствия условий проводимости время после записи остается неизменным. Результат выполнения:



```
PROGRAM POU
VAR

    PP_GetNetInfo_0: PP_System.PP_GetNetInfo;
    PLCID: DWORD;
    BOOTTIME: PP_SYSTEM.SYS_PP_TIME;
    AA: BOOL;
    NETID: BYTE;
    PP_SetNetInfo_0: PP_System.PP_SetNetInfo;
    STTIME: PP_SYSTEM.SYS_PP_TIME;
    runtimeversion: DWORD;
    timezone: INT;
    SetTIME: PP_System.SYS_PP_TIME;
    itimezone: INT;
END_VAR
```

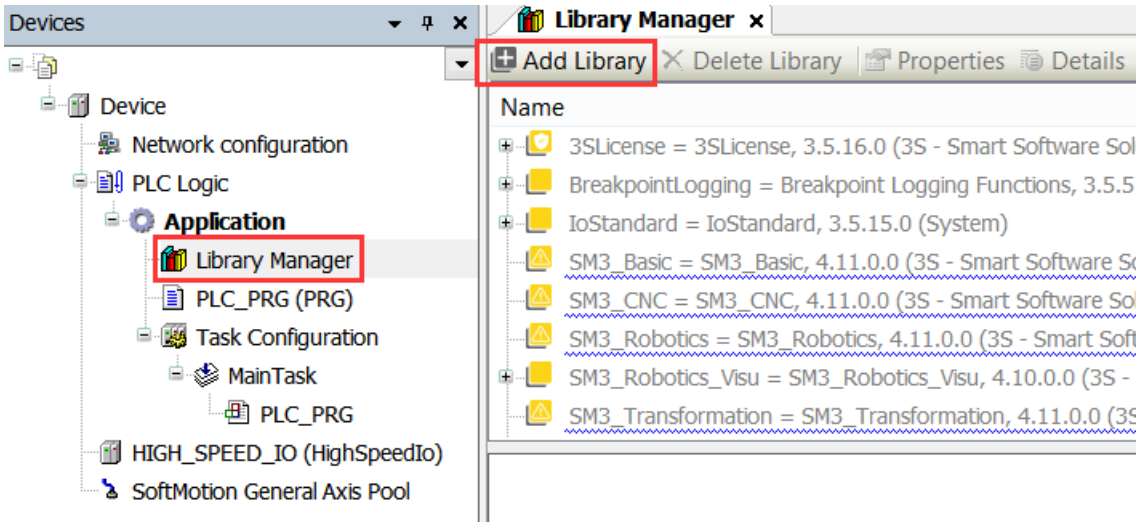
SetTIME	PP_System.SYS_PP_TIME	
Year	UINT	2022
Month	UINT	11
Day	UINT	19
Hour	UINT	13
Minute	UINT	30
Second	UINT	10
Milliseconds	UINT	855
DayOfWeek	UINT	5

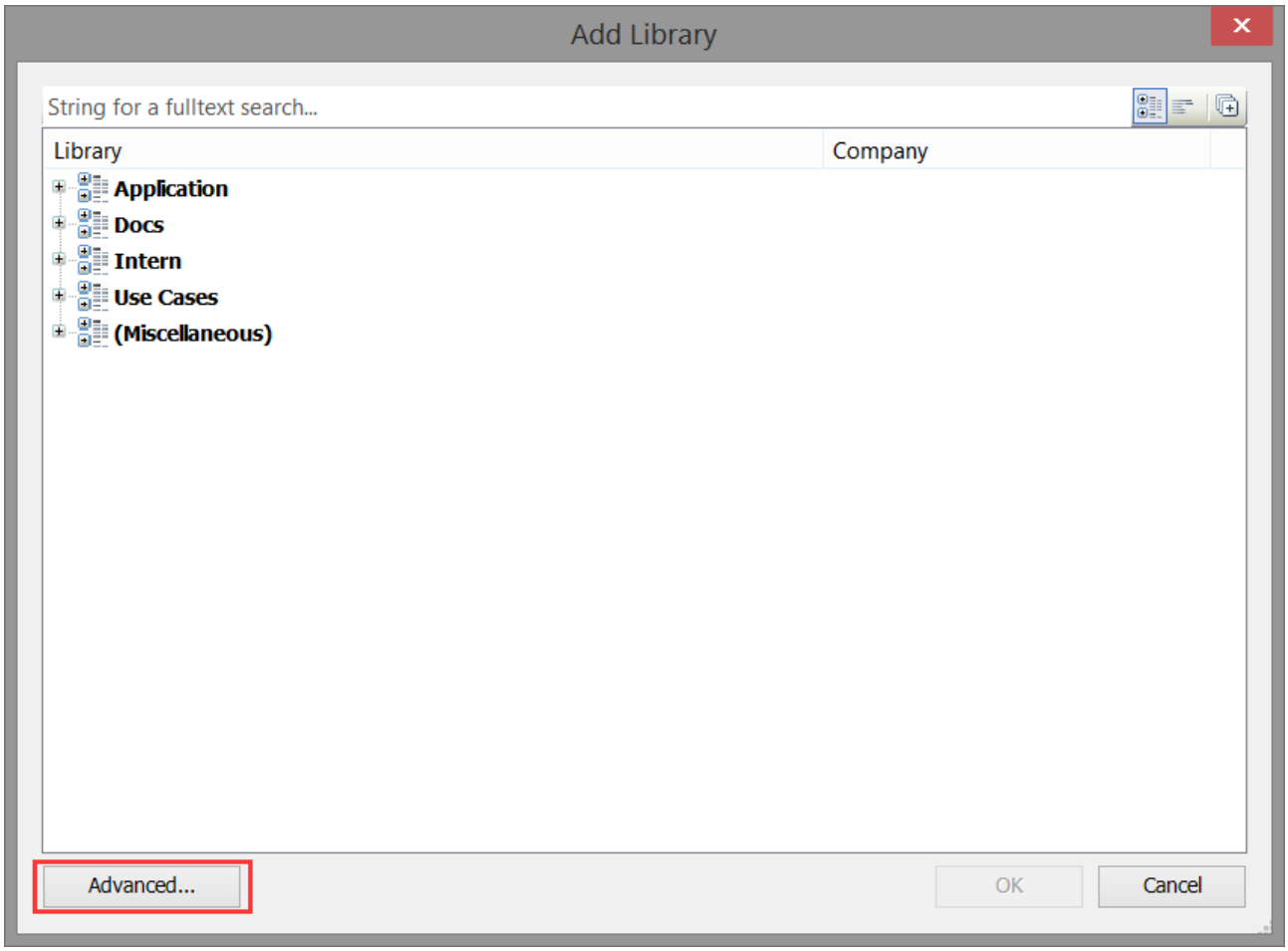
Полученное время соответствует заданному времени.

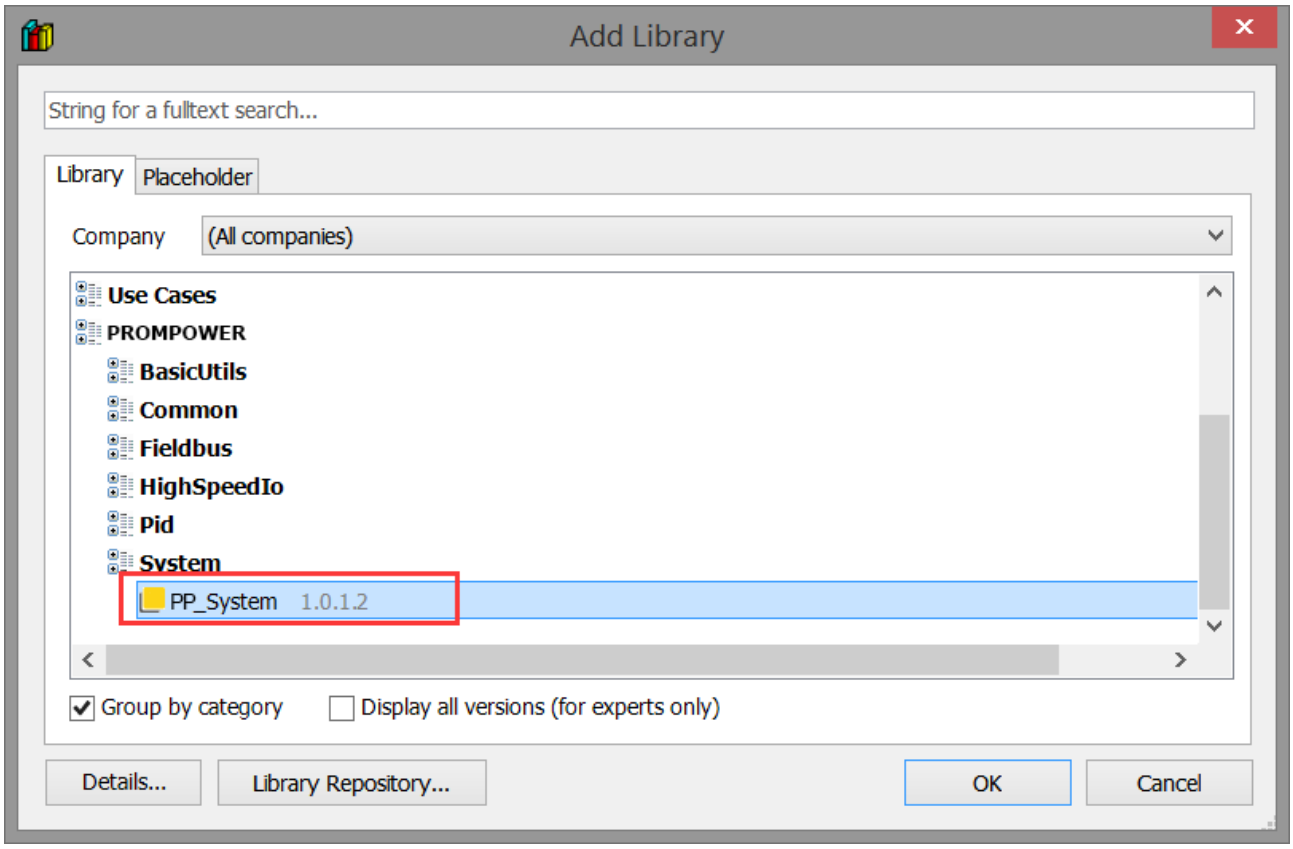
STTIME	PP_SYSTEM.SYS_PP_TIME	
Year	UINT	2022
Month	UINT	11
Day	UINT	19
Hour	UINT	13
Minute	UINT	30
Second	UINT	11
Milliseconds	UINT	938
DayOfWeek	UINT	6

2-5-3. Конфигурация параметров

Добавление библиотечного файла: Library Manager - Add Library - Advanced - Add PP_System (Диспетчер библиотек - Добавить библиотеку - Дополнительно - Добавить PP_System).







2-6. ECAT_FROMTO

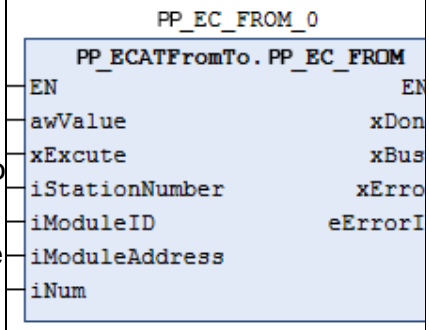
2-6-1. Обзор функции

Используется для считывания и записи параметров модулей на дистанционном входе/выходе, для температурных модулей и модулей взвешивания.

2-6-2. Введение в функциональные блоки

2-6-2-1. Считывание данных на дистанционном входе/выходе [PP_ECATFromTo.PP_EC_FROM]

Считывание данных на дистанционном входе/выходе [PP_ECATFromTo.PP_EC_FROM]			
Условие выполнения	Запуск по нарастающему фронту	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_ECATFromTo		

Команда	Название	Графическое представление	Представление на языке ST
PP_ECATFromTo.PP_EC_FROM	Считывание данных на дистанционном входе/выходе		PP_ECATFromTo.PP_EC_FROM_ awValue:= , xExcute:= , iStationNumber:= , iModuleID:= , iModuleAddress:= , iNum:= , xDone=> , xBusy=> , xError=> , eErrorID=>);

(1) Входные переменные

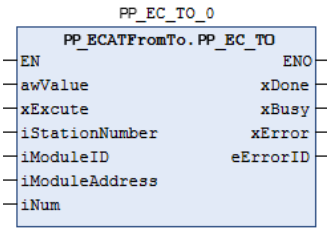
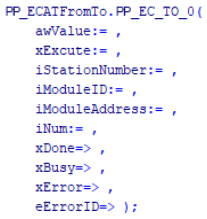
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
awValue	Считывание значения	POINTER TO WORD	-	-	Длина самого длинного массива [0..100]
xExcute	Запуск по нарастающему фронту	BOOL	-	-	
iStationNumber	Номер станции	UINT		-	Номер станции ECAT, например, 1001
iModuleID	Номер модуля	INT	0-15	-	Один дистанционный вход/выход максимум с 16 модулями
iModuleAddress	Адрес модуля	DWORD	-	-	См. адрес модуля в руководстве по эксплуатации модуля расширения.
iNum	Номера в регистре	INT	-	-	Количество последовательно считанных слов

(2) Выходные переменные

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Булев выход	BOOL	TRUE,FALSE	FALSE	Завершение считывания имеет состояние TRUE.
xBusy	Булев выход	BOOL	TRUE,FALSE	FALSE	Считывание имеет состояние TRUE.
xError	Булев выход	BOOL	TRUE,FALSE	FALSE	Если возникает исключение, состояние будет TRUE.
eErrorID	Текущий режим	EC_FT_ERRID	-	-	В нормальном режиме значение равно 0. Если возникает исключение, выдается код ошибки.

2-6-2-2. Запись данных на дистанционном входе/выходе [PP_ECATFromTo.PP_EC_TO]

Запись данных на дистанционном входе/выходе [PP_ECATFromTo.PP_EC_FROM]			
Условие выполнения	Запуск по нарастающему фронту	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_ECATFromTo		

Команда	Название	Графическое представление	Представление на языке ST
PP_ECATFromTo.PP_EC_TO	Запись данных на дистанционном входе/выходе		

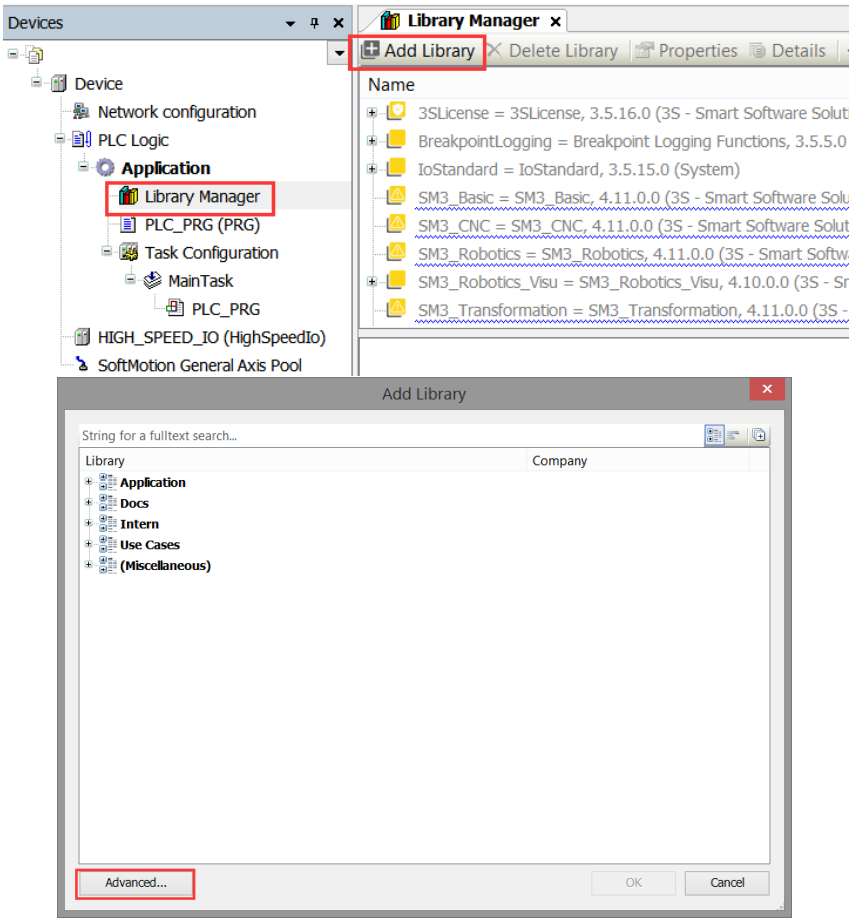
(1) Входные переменные

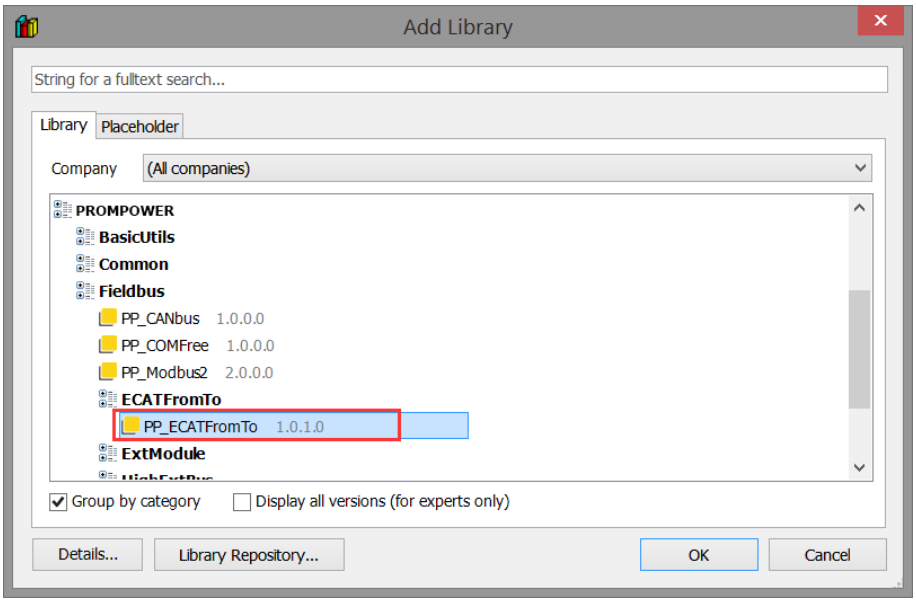
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
awValue	Запись значения	POINTER TO WORD	-	-	Длина самого длинного массива [0..100]
xExcute	Запуск по нарастающему фронту	BOOL	-	-	
iStationNumber	Номер станции	UINT		-	Номер станции ECAT, например, 1001
iModuleID	Номер модуля	INT	0-15	-	Один дистанционный вход/выход максимум с 16 модулями
iModuleAddress	Адрес модуля	DWORD	-	-	См. адрес модуля в руководстве по эксплуатации модуля расширения.
iNum	Номера в регистре	INT	-	-	Количество последовательно записанных слов

(2) Выходные переменные

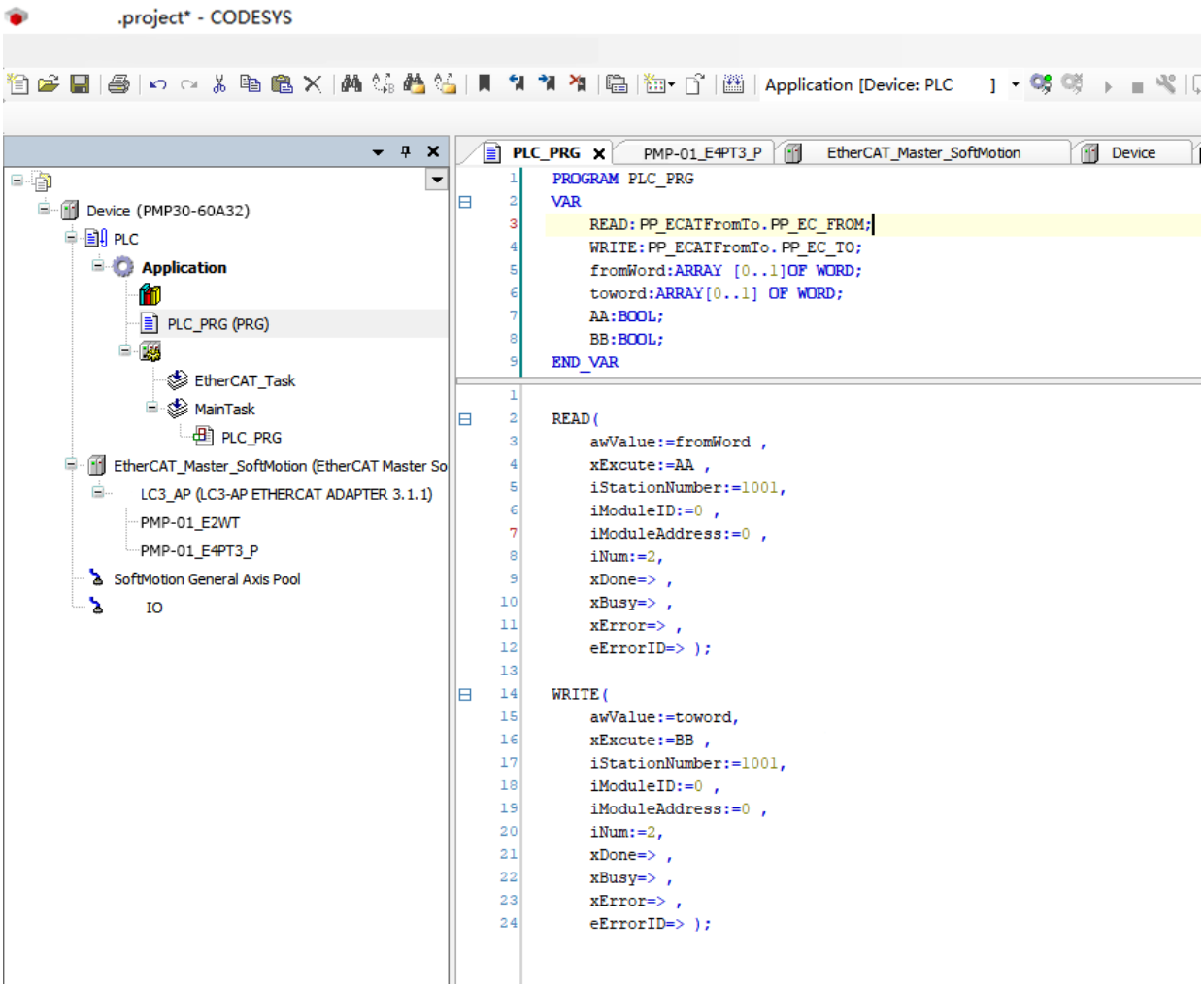
VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Булев выход	BOOL	TRUE,FALSE	FALSE	Завершение записи имеет состояние TRUE.
xBusy	Булев выход	BOOL	TRUE,FALSE	FALSE	Запись имеет состояние TRUE.
xError	Булев выход	BOOL	TRUE,FALSE	FALSE	Если возникает исключение, состояние будет TRUE.
eErrorID	Текущий режим	EC_FT_ERRID	-	-	В нормальном режиме значение равно 0. Если возникает исключение, выдается код ошибки.

2-6-3. Конфигурация параметров





2-6-4. Применение



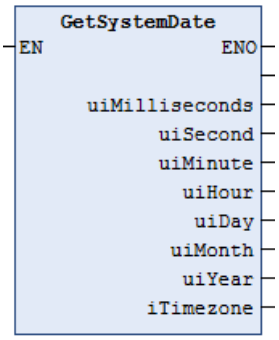
2-7. Дата и время

2-7-1. Получение данных о текущем системном времени GetSystemDate

(1) Обзор команды

Получение данных о текущем системном часовом поясе, дате и времени.

Получение данных о текущем системном времени [GetSystemDate]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	PMP3xx: версия 2.2.0 и более поздние версии	Программное обеспечение	Codesys версии 2.2.0 и более поздние версии (модели серии PMP3xx)
Библиотека	PP_FunctionUtils		

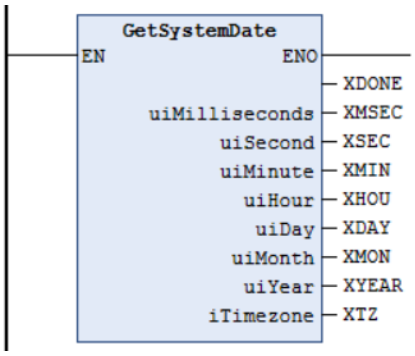
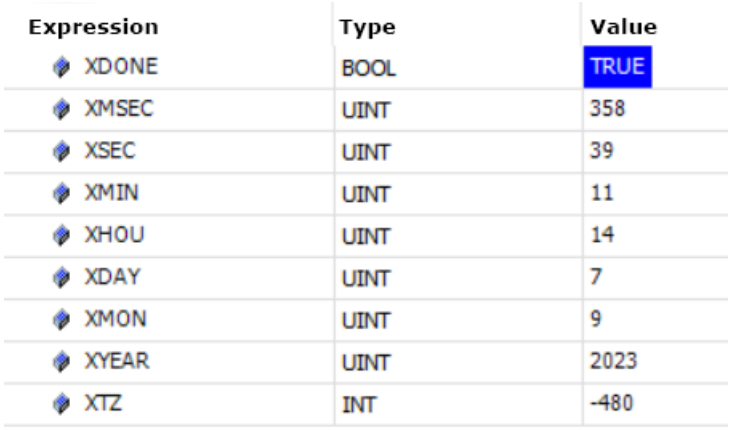
Команда	Название	Графическое представление	Представление на языке ST
GetSystemDate	Получение данных о текущем системном времени		GetSystemDate(uiMilliseconds=> , uiSecond=> , uiMinute=> , uiHour=> , uiDay=> , uiMonth=> , uiYear=> , iTimezone=>);

(2) Связанные переменные

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
uiMilliseconds	Миллисекунда	UINT	-	-	Миллисекунда
uiSecond	Секунда	UINT	-	-	Секунда
uiMinute	Минута	UINT	-	-	Минута
uiHour	Час	UINT	-	-	Час
uiDay	День	UINT	-	-	День
uiMonth	Месяц	UINT	-	-	Месяц

uiYear	Год	UINT	-	-	Год
iTimezone	Часовой пояс	INT	-	-	Часовой пояс

(3) Пример программы

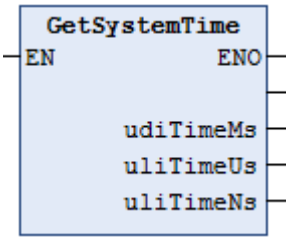
ДЕМО	LD	ST
Задание переменных		VAR XDONE: BOOL; XMSEC: UINT; XSEC: UINT; XMIN: UINT; XHOU: UINT; XDAY: UINT; XMON: UINT; XYEAR: UINT; XTZ: INT; END_VAR
Пример программы		XDONE:= GetSystemDate(uiMilliseconds=> XMSEC, uiSecond=> XSEC, uiMinute=> XMIN, uiHour=> XHOU, uiDay=> XDAY, uiMonth=> XMON, uiYear=> XYEAR, iTimezone=> XTZ);
Результат операции		

2-7-2. Получение данных о времени выполнения программы в системе GetSystemTime

(1) Обзор команды

Получение данных о времени работы после запуска.

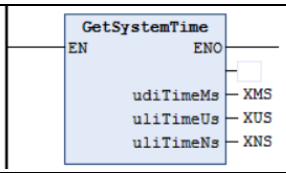
Получение данных о времени выполнения программы в системе [GetSystemTime]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

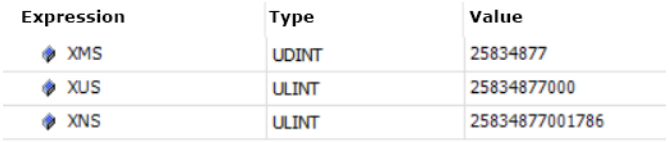
Команда	Название	Графическое представление	Представление на языке ST
GetSystemTime	Получение данных о времени выполнения программы в системе		GetSystemTime(udiTimeMs=> , uliTimeUs=> , uliTimeNs=>);

(2) Связанные переменные

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
udiTimeMs	Время выполнения, мс	UDINT	-	-	Время выполнения, мс
uliTimeUs	Время выполнения, мкс	ULINT	-	-	Время выполнения, мкс
uliTimeNs	Время выполнения, нс	ULINT	-	-	Время выполнения, нс

(3) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR XMS: UDINT; XUS: ULINT; XNS: ULINT; END_VAR
Пример программы		GetSystemTime(udiTimeMs=> XMS, uliTimeUs=> XUS, uliTimeNs=> XNS);

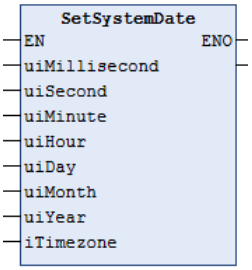
Результат операции	

2-7-3. Настройка текущего системного времени SetSystemDate

(1) Обзор команды

Настройка текущего системного часового пояса, даты и времени.

Настройка текущего системного времени [SetSystemDate]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	PMP3xx: версия 2.2.0 и более поздние версии	Программное обеспечение	Codesys версии 2.2.0 и более поздние версии (модели серии PMP3xx)
Библиотека	PP_FunctionUtils		

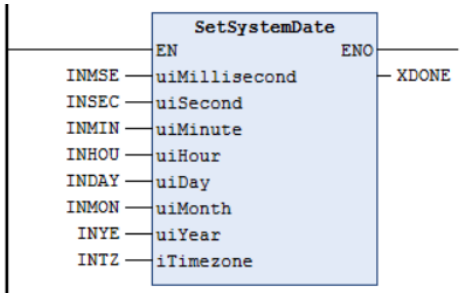
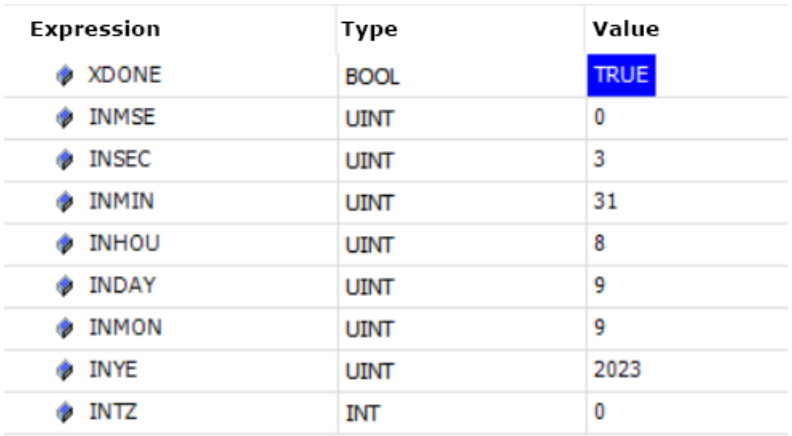
Команда	Название	Графическое представление	Представление на языке ST
SetSystemDate	Настройка текущего системного времени		SetSystemDate(uiMillisecond:= , uiSecond:= , uiMinute:= , uiHour:= , uiDay:= , uiMonth:= , uiYear:= , iTimezone:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
uiMilliseconds	Миллисекунда	UINT	-	-	Миллисекунда
uiSecond	Секунда	UINT	-	-	Секунда
uiMinute	Минута	UINT	-	-	Минута
uiHour	Час	UINT	-	-	Час
uiDay	День	UINT	-	-	День

uiMonth	Месяц	UINT	-	-	Месяц
uiYear	Год	UINT	-	-	Год
iTimezone	Часовой пояс	INT	-	-	Часовой пояс

(3) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR XDONE: BOOL; INMSE: UINT; INSEC: UINT:= 3; INMIN: UINT:= 31; INHOU: UINT:= 8; INDAY: UINT:= 9; INMON: UINT:= 9; INYE: UINT:= 2023; INTZ: INT; END_VAR
Пример программы		XDONE:= SetSystemDate(uiMillisecond:= INMSE, uiSecond:= INSEC, uiMinute:= INMIN, uiHour:= INHOU, uiDay:= INDAY, uiMonth:= INMON, uiYear:= INYE, iTimezone:= INTZ);
Результат операции		

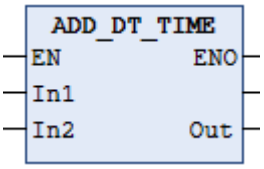
Expression	Type	Value
XDONE	BOOL	TRUE
INMSE	UINT	0
INSEC	UINT	3
INMIN	UINT	31
INHOU	UINT	8
INDAY	UINT	9
INMON	UINT	9
INYE	UINT	2023
INTZ	INT	0

2-7-4. Добавление даты и времени ADD_DT_TIME

(1) Обзор команды

Добавление даты и времени.

Добавление даты и времени [ADD_DT_TIME]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
ADD_DT_TIME	Добавление даты и времени		ADD_DT_TIME(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Добавляемая дата	DT	В качестве типа данных	-	Добавляемая дата
In2	Добавляемое время	TIME	В качестве типа данных	-	Добавляемое время
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат	DT	В качестве типа данных	-	Результат

(3) Функция

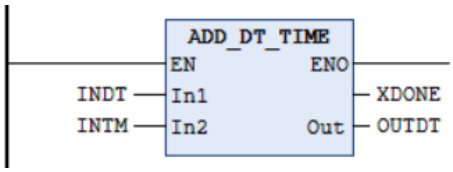
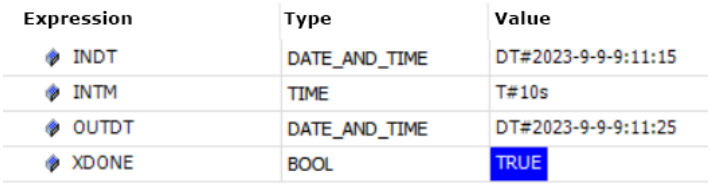
- Дата "In1" и время "In2" складываются, и сумма "Out" является датой и временем.

(4) Примечание

- Если сумма превышает допустимый диапазон "Out", программа не будет иметь никаких исключений. Выполните следующие действия:

DT#2106-2-7-6:28:15 + T#1 с → DT#1970-1-1-0:0:0

(5) Пример программы

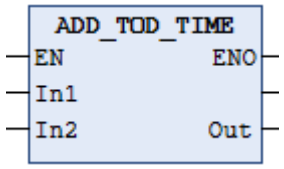
ДЕМО	LD	ST
Задание переменных	VAR INDT: DT:= DT#2023-9-9-9:11:15; INTM: TIME:= T#10S; OUTDT: DT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= ADD_DT_TIME(In1:= INDT, In2:= INTM, Out=> OUTDT);
Результат операции		

2-7-5. Сложение значений времени ADD_TOD_TIME

(1) Обзор команды

Сложение значений времени.

Сложение значений времени [ADD_TOD_TIME]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
ADD_TOD_TIME	Сложение значений времени		ADD_TOD_TIME(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Прибавляемое время	TOD	В качестве типа данных	-	Прибавляемое время
In2	Прибавляемое время	TIME	В качестве типа данных	-	Прибавляемое время
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Сумма значений времени	TOD	В качестве типа данных	-	Сумма значений времени

(3) Функция

- Сложите время "In1" со временем "In2", и полученная сумма "Out" и будет временем.

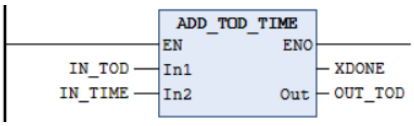
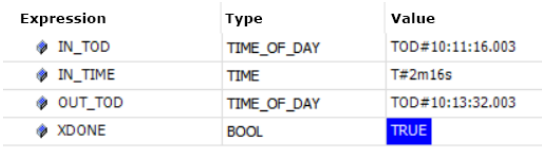
(4) Примечание

- Если сумма превышает допустимый диапазон "Out", программа не будет иметь никаких исключений. Выполните следующие действия:

TOD#23:59:59.999 + T#1 с 1 мс → TOD#0:0:1

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR IN_TOD: TOD:= TOD#10:11:16.003; IN_TIME: TIME:= T#2M16S; OUT_TOD: TOD; XDONE: BOOL; END_VAR

Пример программы		XDONE:= ADD_TOD_TIME(In1:= IN_TOD, In2:= IN_TIME, Out=> OUT_TOD);
Результат операции		

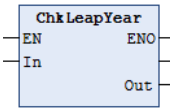
Expression	Type	Value
IN_TOD	TIME_OF_DAY	TOD#10:11:16.003
IN_TIME	TIME	T#2m16s
OUT_TOD	TIME_OF_DAY	TOD#10:13:32.003
XDONE	BOOL	TRUE

2-7-6. Проверка високосного года ChkLeapYear

(1) Обзор команды

Определение, является ли конкретный год високосным.

Проверка високосного года [ChkLeapYear]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
ChkLeapYear	Проверка високосного года		ChkLeapYear(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Год	UINT	1970-2106	-	Год
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат проверки	BOOL	FALSE/TRUE	-	True: високосный год FALSE: Невисокосный год

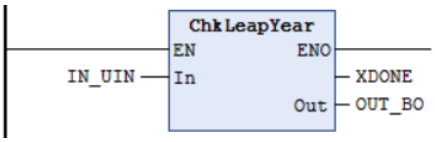
(3) Функция

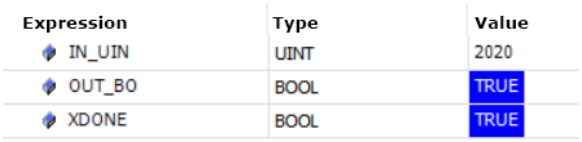
- Определение, является ли год "In" високосным. Если год високосный, то значением результата проверки "Out" будет true. В противном случае, значение результата — FALSE.

(4) Примечание

- Когда значение "In" превышает допустимый диапазон, программа столкнется с исключением. Выходное значение "Out" — FALSE, и функция также возвращает значение FALSE.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">
<pre>VAR
   IN_UIN: UINT:= 2020;
   OUT_BO: BOOL;
   XDONE: BOOL;
END_VAR</pre>
</td></tr>
<tr><td>Пример программы</td><td>


</td><td>
<pre>XDONE:= ChkLeapYear(
    In:= IN_UIN,
    Out=> OUT_BO);</pre>
</td></tr>
<tr><td>Результат операции</td><td colspan="2">


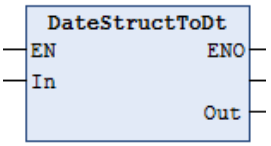
</td></tr>
</table>

2-7-7. Объединение даты и времени DateStructToDt

(1) Обзор команды

Объединение дат и моментов, разложенных на годы, месяцы, дни, часы, минуты и секунды.

Объединение даты и времени [DateStructToDt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
DateStructToDt	Объединение даты и времени		`DateStructToDt(In:= , Out=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Дата и время	_sDT	В качестве типа данных	-	Дата и время до объединения
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Дата и время	DT	DT#1970-1-1-0:0:0-DT#2106-2-7-6:28:15	DT#1970-1-1-0:0:0	Дата и время после объединения

(3) Функция

- Объединение даты и времени "In", разложенных на «год», «месяц», «день», «час», «минуту» и «секунду».
- Элементы структуры _sDT:

Переменная	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
uiYear	Год	UINT	1970-2106	1970	Год
uiMonth	Месяц	UINT	1-12	1	Месяц
uiDay	День	UINT	1-31	1	День
uiHour	Час	UINT	0-23	0	Час
uiMinute	Минута	UINT	0-59	0	Минута
uiSecond	Секунда	UINT	0-59	0	Секунда
uiMillisecond	Миллисекунда	UINT	0-999	0	Миллисекунда
uiDayOfWeek	Неделя	UINT	1-7	1	Неделя

(4) Примечание

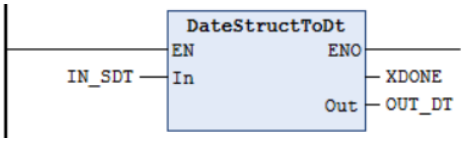
- Допустимый диапазон для вывода даты и времени "Out": DT#1970-1-1-0:0:0 - DT#2106-2-7-6:28:15.
- Если выходное значение "Out" превышает допустимый диапазон, функция возвращает значение FALSE, а вычисленный результат — DT#1970-1-1-0:0:0.
- Если входное значение "In" превышает допустимый диапазон каждого элемента в структуре "_sDT", функция возвращает значение FALSE, а вычисленный результат — DT#1970-1-1-0:0:0.
- Начальные значения каждого элемента в структуре "_sDT" являются значениями разложения DT#1970-1-1-0:0:0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN_SDT: _sDT;
    OUT_DT: DT;
    XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= DateStructToDt(
    In:= IN_SDT,
    Out=> OUT_DT);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_SDT	_sDT	
uiYear	UINT	2023
uiMonth	UINT	9
uiDay	UINT	9
uiHour	UINT	10
uiMinute	UINT	31
uiSecond	UINT	15
uiMillisecond	UINT	256
uiDayOfWeek	UINT	6
OUT_DT	DATE_AND_TIME	DT#2023-9-9-10:31:15
XDONE	BOOL	TRUE

</td></tr>
</table>

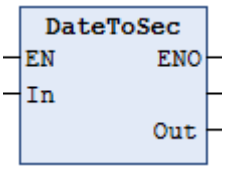
2-7-8. Преобразование даты в секунды DateToSec

(1) Обзор команды

Преобразование введенной даты в секунды, начиная с 1 января 1970 года, 00:00:00.

Преобразование даты в секунды [DateToSec]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
DateToSec	Преобразование даты в секунды		`DateToSec(In:= , Out=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Дата	DATE	В качестве типа данных	D#1970-1-1	Дата
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	LINT	0-4294967295	-	Результат преобразования

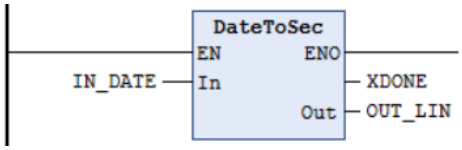
(3) Функция

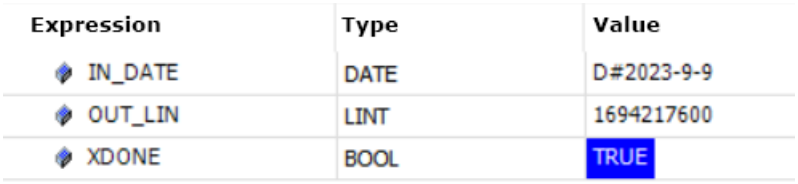
- Преобразование 0:00:00 с даты "In" в секунды, начиная с 0:00:00 с 1 января 1970 года. Преобразованное значение измеряется в секундах.

(4) Пример программы

ДЕМО	LD	ST

Задание переменных		`VAR`
		`  IN_DATE: DATE:= D#2023-9-9;`
		`  OUT_LIN: LINT;`
		`  XDONE: BOOL;`
		`END_VAR`

Пример программы		`XDONE:= DateToSec(`
		`  In:= IN_DATE,`
		`  Out=> OUT_LIN);`

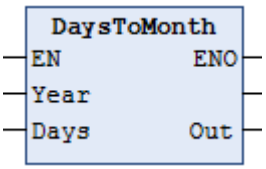
Результат операции	

2-7-9. Преобразование дней в месяцы DaysToMonth

(1) Обзор команды

Вычисление месяца даты в зависимости от количества дней, начиная с 1-го января.

Преобразование дней в месяцы [DaysToMonth]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
DaysToMonth	Преобразование дней в месяцы		DaysToMonth(Year:= , Days:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Year	Год	UINT	1970-2106	1970	Год
Days	Дни	UINT	1-365 (Для високосного года получим 1-366)	1	Кол-во дней с 1-го января
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Месяц	USINT	1-12	-	Месяц

(3) Функция

- Вычисление месяца дня по количеству дней, начиная с 1 января года.

(4) Примечание

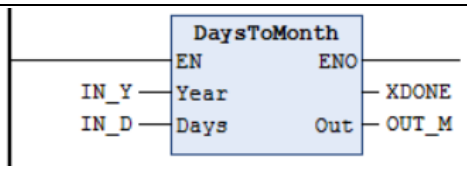
- Если значение параметра "Year" выходит за пределы допустимого диапазона, исключения не произойдет, а значение параметра "Out" будет недопустимым.
- Если значение параметра "Days" превышает допустимый диапазон, функция возвращает значение FALSE, а параметр "Out" будет равен 0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   IN_Y: UINT:= 2023;
   IN_D: UINT:= 156;
   OUT_M: USINT;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= DaysToMonth(
    Year:= IN_Y,
    Days:= IN_D,
    Out=> OUT_M);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_Y	UINT	2023
IN_D	UINT	156
OUT_M	USINT	6
XDONE	BOOL	TRUE

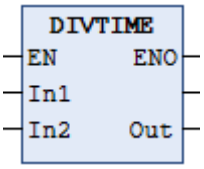
</td></tr>
</table>

2-7-10. Деление времени DIVTIME

(1) Обзор команды

Деление времени на заданный делитель.

Деление времени [DIVTIME]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
DIVTIME	Деление времени		DIVTIME(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Делимое время	TIME	В качестве типа данных	-	Делимое время
In2	Делитель	USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT REAL/LREAL	В качестве типа данных	-	Делитель
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат деления	TIME	В качестве типа данных	-	Результат деления времени

(3) Функция

- Время "In1" делится на делитель "In2", и результат деления "Out" является временем.

(4) Примечание

- Если значение "In2" отрицательное, программа столкнется с исключением. Если выходное значение "Out" равно T#0 мс, функция возвращает значение FALSE.
- Если результат деления превышает допустимый диапазон параметра "Out", выходное значение параметра "Out" равно T#0 мс, а возвращаемое значение функции — FALSE.
- Если входное значение параметра "In2" приводит к тому, что выходное значение параметра "Out" превышает верхний предел типа TIME, выходное значение "Out" будет равно T#0 мс, а функция возвращает значение FALSE.

- Если "In2" имеет тип integer и значение 0, функция возвращает значение FALSE, а "Out" равен 0.
- Если "In2" — действительное число, то значения, меньшие 1 мс в результате деления, будут округлены.
- Если "In2" — действительное число, то результат деления с большим количеством значащих цифр может привести к ошибке в несколько миллисекунд.
- Если "In2" — действительное число, а значение "In2" равно 0, +∞, -∞ и не является числовым значением, функция возвращает значение True, а значение параметра "Out" выглядит следующим образом:

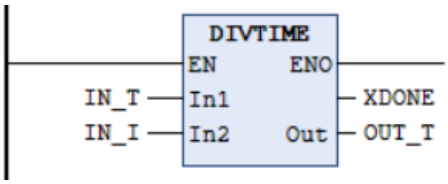
Входное значение “In2”	Выходное значение “Out”
0	T#49 д 17 ч 2 м 47 с 295 мс
+∞	T#0 мс
-∞	T#0 мс
числовое значение	T#0 мс

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   IN_T: TIME:= T#2D16H8M6S200MS;
   IN_I: INT:= 2;
   OUT_T: TIME;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= DIVTIME(
    In1:= IN_T,
    In2:= IN_I,
    Out=> OUT_T);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_T	TIME	T#2d16h8m6s200ms
IN_I	INT	2
OUT_T	TIME	T#1d8h4m3s100ms
XDONE	BOOL	TRUE

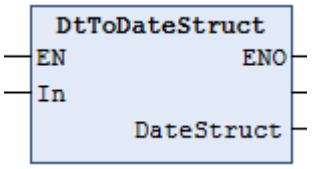
</td></tr>
</table>

2-7-11. Разложение даты и времени DtToDateStruct

(1) Обзор команды

Разложите дату и время на годы, месяцы, дни, часы, минуты и секунды.

Разложение даты и времени [DtToDateStruct]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
DtToDateStruct	Разложение даты и времени		DtToDateStruct(In:= , DateStruct=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Дата и время	DT	DT#1970-1-1-0:0:0-DT#2106-2-7-6:28:15	DT#1970-1-1-0:0:0	Дата и время до разложения
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Дата и время	_sDT	В качестве типа данных	-	Дата и время после разложения

(3) Функция

- Разложите дату и время "In" на «год», «месяц», «день», «час», «минуту» и «секунду» и выведите их в формате структуры _sDT.

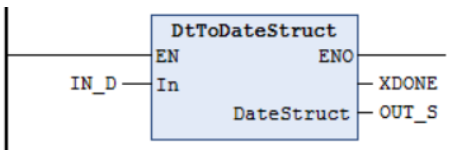
- Элементы структуры _sDT:

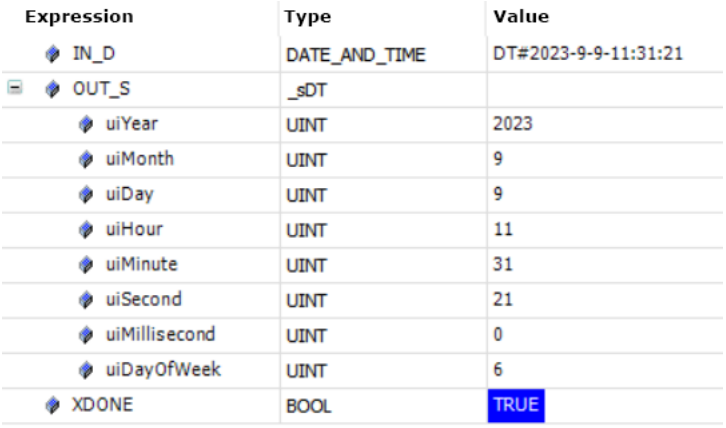
Переменная	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
uiYear	Год	UINT	1970-2106	1970	Год
uiMonth	Месяц	UINT	1-12	1	Месяц
uiDay	День	UINT	1-31	1	День
uiHour	Час	UINT	0-23	0	Час
uiMinute	Минута	UINT	0-59	0	Минута
uiSecond	Секунда	UINT	0-59	0	Секунда
uiMillisecond	Миллисекунда	UINT	0-999	0	Миллисекунда
uiDayOfWeek	Неделя	UINT	1-7	1	Неделя

(4) Примечание

- Допустимый диапазон для ввода даты и времени In: DT#1970-1-1-0:0:0 - DT#2106-2-7-6:28:15.
- Если входное значение In превышает допустимый диапазон, функция возвращает значение FALSE и вычисляет результат разложения DT#1970-1-1-0:0:0.
- Начальные значения каждого элемента в структуре _sDT — это значения разложения DT#1970-1-1-0:0:0.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_D: DT:= DT#2023-9-9-11:31:21; OUT_S: _sDT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= DtToDateStruct(In:= IN_D, DateStruct=> OUT_S);

Результат операции	

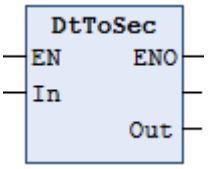
Expression	Type	Value
IN_D	DATE_AND_TIME	DT#2023-9-9-11:31:21
OUT_S	_sDT	
uiYear	UINT	2023
uiMonth	UINT	9
uiDay	UINT	9
uiHour	UINT	11
uiMinute	UINT	31
uiSecond	UINT	21
uiMillisecond	UINT	0
uiDayOfWeek	UINT	6
XDONE	BOOL	TRUE

2-7-12. Преобразование даты и времени в секунды DtToSec

(1) Обзор команды

Преобразование даты и времени в секунды, начиная с 1 января 1970 года, 00:00:00.

Преобразование даты и времени в секунды [DtToSec]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
DtToSec	Преобразование даты и времени в секунды		DtToSec(In:= , Out=>);

(2) Связанные переменные

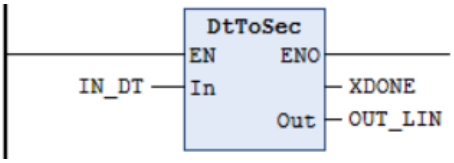
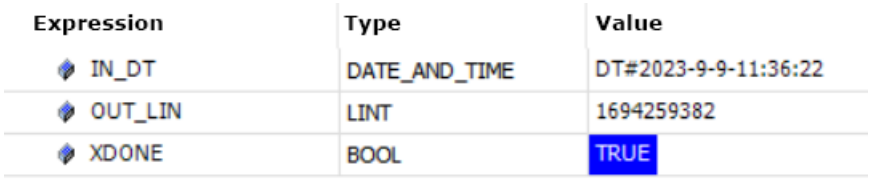
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Дата и время	DT	В качестве типа данных	DT#1970-1-1-0:0:0	Дата и время

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Out	Результат преобразования	LINT	0-4294967295	-	Результат преобразования

(3) Функция

- Преобразование даты и времени "In" в секунды, начиная с 1 января 1970 года, 00:00:00. Преобразованное значение измеряется в секундах. Значения менее 1 секунды округляются.

(4) Пример программы

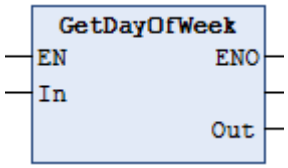
ДЕМО	LD	ST
Задание переменных	VAR IN_DT: DT:= DT#2023-9-9-11:36:22; OUT_LIN: LINT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= DtToSec(In:= IN_DT, Out=> OUT_LIN);
Результат операции		

2-7-13. Получение информации о дне недели GetDayOfWeek

(1) Обзор команды

Получение информации о дне недели для указанного года, месяца и дня.

Получение информации о дне недели [GetDayOfWeek]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
GetDayOfWeek	Получение информации о дне недели		GetDayOfWeek(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Дата	DATE/DT	DATE#1970-1-1-DATE#2106-2-7 DT#1970-1-1-0:0:0-DT#2106-2-7-6:28:15	-	Год, месяц, день
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Неделя	_eDAYOFWEEK	В качестве типа данных	_MON	Информация о дне недели

(3) Функция

- Получение информации о дне недели для года, месяца и дня, указанного в "In".
- Типом данных переменной "Out" является элемент перечисления __eDAYOFWEEK. Значения элементов перечисления:

Элемент перечисления _eDAYOFWEEK	Значение
_MON	Понедельник
_TUE	Вторник
_WED	Среда
_THU	Четверг
_FRI	Пятница
_SAT	Суббота

_SUN	Воскресенье

(4) Примечание

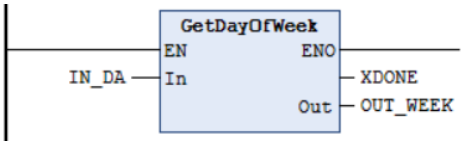
- Когда значение "In" превышает допустимый диапазон, программа столкнется с исключением. Если выходное значение "Out" равно _MON, функция возвращает значение FALSE.
- Входной параметр, переданный в "In", должен являться переменной. Если передана константа, произойдет ошибка при компиляции.
- Для "In" требуются типы данных DATA и DT. Для типов данных TIME и TOD компиляция проходит без ошибок, и после выполнения функция возвращает FALSE. Для других типов данных компиляция происходит с ошибками.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN_DA: DATE:= DATE#2023-9-9;
    OUT_WEEK: _eDAYOFWEEK;
    XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= GetDayOfWeek(
    In:= IN_DA,
    Out=> OUT_WEEK);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_DA	DATE	D#2023-9-9
OUT_WEEK	_EDAYOFWEEK	_SAT
XDONE	BOOL	TRUE

</td></tr>
</table>

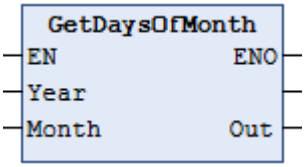
2-7-14. Получение информации о количестве дней в указанном месяце GetDaysOfMonth

(1) Обзор команды

Получение информации о количестве дней в указанном месяце.

Получение информации о количестве дней в указанном месяце [GetDaysOfMonth]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

Библиотека	PP_FunctionUtils	

Команда	Название	Графическое представление	Представление на языке ST
GetDaysOfMonth	Получение информации о количестве дней в указанном месяце		GetDaysOfMonth(Year:= , Month:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Year	Год	UINT	1970-2106	-	Год
Month	Месяц	USINT	1-12	-	Месяц
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Дней	USINT	28-31	-	Дней

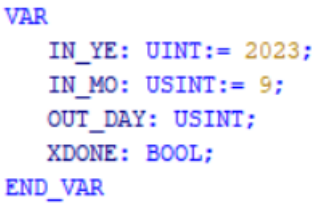
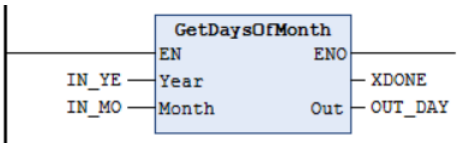
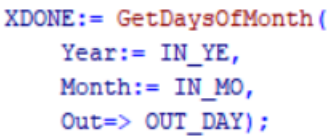
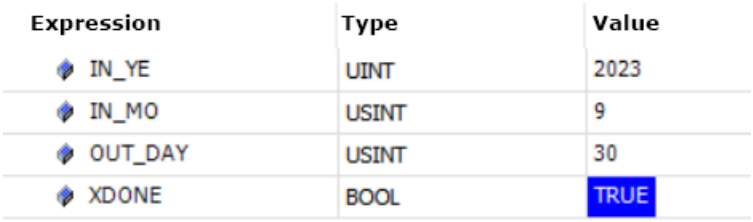
(3) Функция

- Получение информации о количестве дней в указанном году "Year" и указанном месяце "Month".

(4) Примечание

- Если значение "Year" превышает допустимый диапазон, программа столкнется с исключением. Выходное значение "Out" равно 0, и функция возвращает значение FALSE.
- Если значение "Month" превышает допустимый диапазон, программа столкнется с исключением. Выходное значение "Out" равно 0, и функция возвращает значение FALSE.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    IN_YE: UINT:= 2023;
    IN_MO: USINT:= 9;
    OUT_DAY: USINT;
    XDONE: BOOL;
END_VAR
```



```
XDONE:= GetDaysOfMonth(
    Year:= IN_YE,
    Month:= IN_MO,
    Out=> OUT_DAY);
```

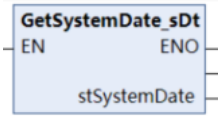
Expression	Type	Value
IN_YE	UINT	2023
IN_MO	USINT	9
OUT_DAY	USINT	30
XDONE	BOOL	TRUE

2-7-15. Получение информации о структуре текущего системного времени GetSystemDate_sDt

(1) Обзор команды

Получение информации о структуре текущего системного времени.

Получение информации о структуре текущего системного времени [GetSystemDate_sDt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
GetSystemDate_sDt	Получение информации о структуре текущего системного времени		GetSystemDate_sDt(stSystemDate =>);

(2) Связанные переменные

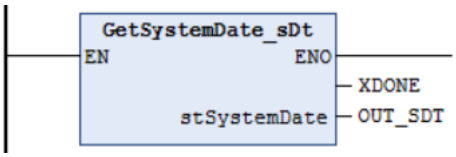
VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
stSystemDate	Системное время	_sDT	-	-	Структура системного времени

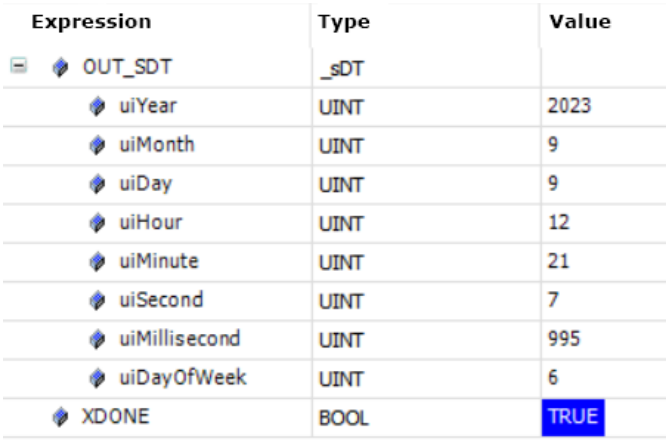
(3) Функция

- Считывание текущего системного времени и его вывод в виде структуры _sDT.
- Элементы структуры _sDT:

Переменные	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
uiYear	Год	UINT	1970-2106	1970	Год
uiMonth	Месяц	UINT	1-12	1	Месяц
uiDay	День	UINT	1-31	1	День
uiHour	Час	UINT	0-23	0	Час
uiMinute	Минута	UINT	0-59	0	Минута
uiSecond	Секунда	UINT	0-59	0	Секунда
uiMillisecond	Миллисекунда	UINT	0-999	0	Миллисекунда
uiDayOfWeek	Неделя	UINT	1-7	1	Неделя

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR OUT_SDT: _sDT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= GetSystemDate_sDt(stSystemDate=> OUT_SDT);

Результат операции	

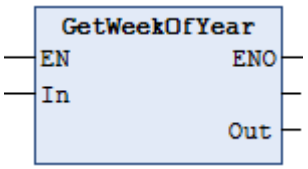
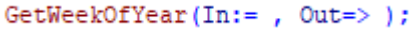
Expression	Type	Value
OUT_SDT	_sDT	
uiYear	UINT	2023
uiMonth	UINT	9
uiDay	UINT	9
uiHour	UINT	12
uiMinute	UINT	21
uiSecond	UINT	7
uiMillisecond	UINT	995
uiDayOfWeek	UINT	6
XDONE	BOOL	TRUE

2-7-16. Получение информации о неделе года GetWeekOfYear

(1) Обзор команды

Вычисление указанного года, месяца и дня в качестве недели текущего года.

Получение информации о неделе года [GetWeekOfYear]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
GetWeekOfYear	Получение информации о неделе года		


```
GetWeekOfYear(In:= , Out=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Год, месяц, день	DATE/DT	DATE#1970-1-1-DATE#2106-2-7 DT#1970-1-1-0:0:0-DT#2106-2-7-6:28:15	-	Год, месяц, день

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Out	Неделя	USINT	1-54	-	Недели указанного года

(3) Функция

- Вычисление недель указанного года, месяца, дня “In”.
- Период с понедельника по воскресенье считается одной неделей, а когда воскресенье сменяется понедельником, то считается за две недели.
- 1 января должно относиться к первой неделе.

(4) Примечание

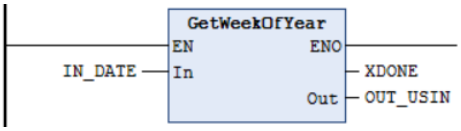
- Когда значение "In" превышает допустимый диапазон, программа столкнется с исключением. Выходное значение "Out" равно 0, и функция возвращает значение FALSE.
- Входной параметр, переданный в "In", должен являться переменной. Если передана константа, произойдет ошибка при компиляции.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   IN_DATE: DATE:= DATE#2023-9-9;
   OUT_USIN: USINT;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= GetWeekOfYear(
    In:= IN_DATE,
    Out=> OUT_USIN);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_DATE	DATE	D#2023-9-9
OUT_USIN	USINT	37
XDONE	BOOL	TRUE

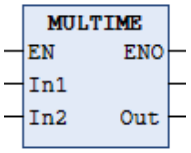
</td></tr>
</table>

2-7-17. Умножение времени MULTIME

(1) Обзор команды

Умножение времени на заданный множитель.

Умножение времени [MULTIME]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
MULTIME	Умножение времени		MULTIME(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Умножаемое время	TIME	В качестве типа данных	-	Умножаемое время
In2	Множитель	USINT/UINT/UDINT/ULINT / SINT/INT/DINT/LINT/ REAL/LREAL	В качестве типа данных	-	Множитель
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат умножения	TIME	В качестве типа данных	-	Результат умножения

(3) Функция

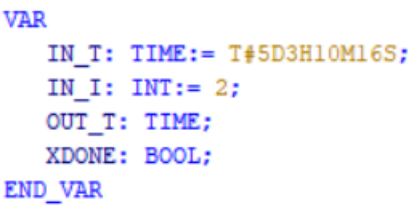
- Время "In1" умножается на множитель "In2", и результат умножения "Out" является временем.

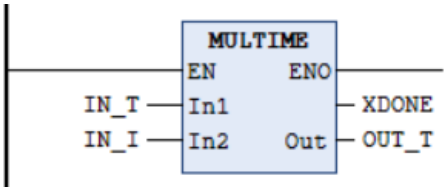
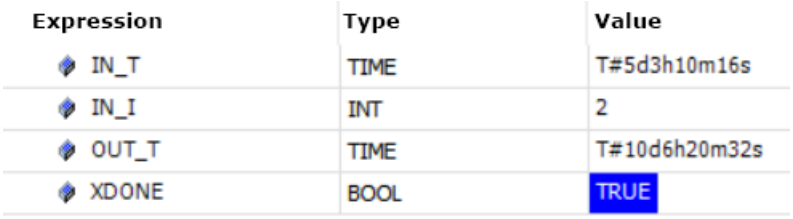
(4) Примечание

- Если значение "In2" отрицательное, программа столкнется с исключением. Выходное значение "Out" равно T#0 мс, и функция возвращает значение FALSE.
- Если результат умножения превышает допустимый диапазон параметра "Out", выходное значение параметра "Out" равно T#0 мс, а возвращаемое значение функции — FALSE.
- Если входное значение параметра "In2" приводит к тому, что выходное значение параметра "Out" превышает верхний предел типа TIME, выходное значение "Out" будет равно T#0 мс, а функция возвращает значение FALSE.
- Если "In2" — действительное число, то результат умножения меньше 1 мс будет округлен.
- Если "In2" — действительное число, то результат умножения с большим количеством значащих цифр может привести к ошибке в несколько миллисекунд.
- Значение "In2" — +∞, -∞, а если это не числовое значение, то значение параметра "Out" будет следующим:

Входное значение “In2”	Выходное значение “Out”
+∞	T#0 мс
-∞	T#0 мс
Нечисловое значение	T#0 мс

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		

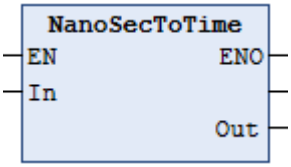
Пример программы		XDONE:= MULTIME(In1:= IN_T, In2:= IN_I, Out=> OUT_T);
Результат операции		

2-7-18. Преобразование наносекунд во время NanoSecToTime

(1) Обзор команды

Преобразование наносекунд во время.

Преобразование наносекунд во время [NanoSecToTime]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
NanoSecToTime	Преобразование наносекунд во время		NanoSecToTime(In:= , Out=>);

(2) Связанные переменные

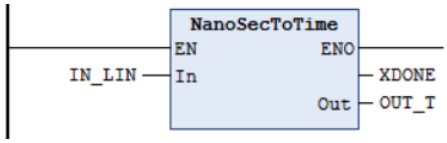
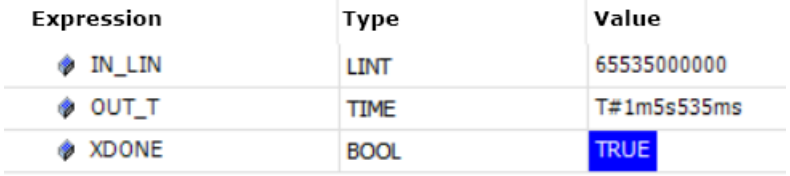
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Наносекунды	LINT	0-4294967295000000	0	Наносекунды

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Out	Результат преобразования	TIME	В качестве типа данных	-	Время после преобразования

(3) Функция

- Преобразование наносекунд "In" во время.

(4) Пример программы

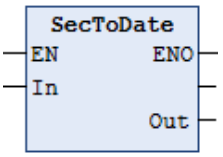
ДЕМО	LD	ST
Задание переменных	VAR IN_LIN: LINT:= 65535000000; OUT_T: TIME; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= NanoSecToTime(In:= IN_LIN, Out=> OUT_T);
Результат операции		

2-7-19. Преобразование секунд в дату SecToDate

(1) Обзор команды

Преобразование секунд, отсчитываемых с 1 января 1970 года, 00:00:00, в дату.

Преобразование секунд в дату [SecToDate]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SecToDate	Преобразование секунд в дату		`SecToDate(In:= , Out=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Секунды	LINT	0-4294967295	0	Количество секунд, отсчитываемых с 00:00:00 1 января 1970 года
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	DATE	В качестве типа данных	-	Дата после преобразования

(3) Функция

- Преобразование количества секунд "In", отсчитываемых с 00:00:00 1 января 1970 года, в дату. Значения менее 1 дня округляются.

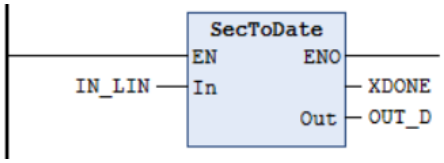
(4) Примечание

- Если значение "In" превышает допустимый диапазон, функция возвращает значение FALSE, а значение "Out" становится равным D#1970-1-1.

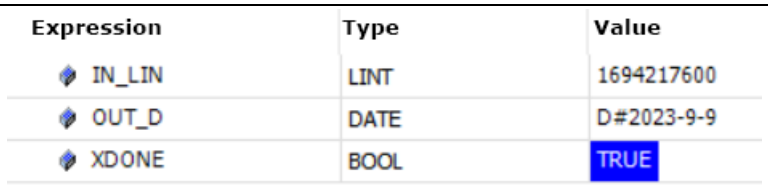
(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		

```
VAR
    IN_LIN: LINT:= 1694217600;
    OUT_D: DATE;
    XDONE: BOOL;
END_VAR
```




```
XDONE:= SecToDate(
    In:= IN_LIN,
    Out=> OUT_D);
```

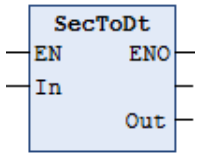
Результат операции	

2-7-20. Преобразование секунд в дату и время SecToDt

(1) Обзор команды

Преобразование количества секунд, отсчитываемых с 00:00:00 1 января 1970 года, в дату и время.

Преобразование секунд в дату и время [SecToDt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SecToDt	Преобразование секунд в дату и время		SecToDt(In:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Секунды	LINT	0-4294967295	0	Количество секунд, отсчитываемых с 00:00:00 1 января 1970 года
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	DT	В качестве типа данных	-	Дата и время после преобразования

(3) Функция

- Преобразование количества секунд "In", отсчитываемых с 00:00:00 1 января 1970 года, в дату и время.

(4) Примечание

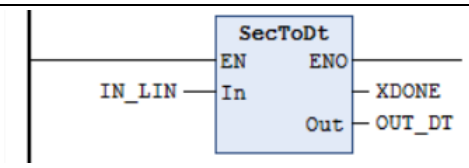
- Если значение "In" превышает допустимый диапазон, функция возвращает значение FALSE, а значение "Out" становится равным DT#1970-1-1-0:0:0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    IN_LIN: LINT:= 1694263261;
    OUT_DT: DT;
    XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= SecToDt(
    In:= IN_LIN,
    Out=> OUT_DT);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_LIN	LINT	1694263261
OUT_DT	DATE_AND_TIME	DT#2023-9-9-12:41:1
XDONE	BOOL	TRUE

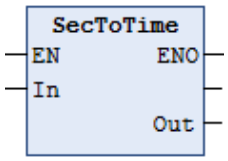
</td></tr>
</table>

2-7-21. Преобразование секунд во время SecToTime

(1) Обзор команды

Преобразование секунд во время.

Преобразование секунд во время [SecToTime]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SecToTime	Преобразование секунд во время		`SecToTime(In:= , Out=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Секунды	LINT	0-4294967295	0	Секунды
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	TIME	В качестве типа данных	-	Время после преобразования

(3) Функция

- Преобразование количества секунд "In" во время.

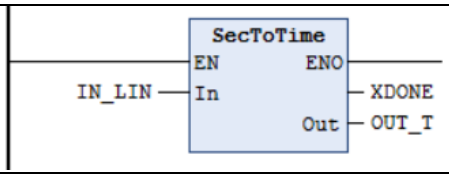
(4) Примечание

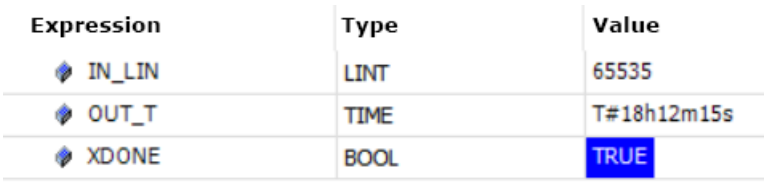
- Единица измерения "In" — секунды, а единица измерения "Out" — миллисекунды.
- Если значение параметра "In" превышает допустимый диапазон, функция возвращает значение FALSE, а параметр "Out" становится равен 0.

(5) Пример программы

ДЕМО	LD	ST

Задание переменных		`VAR`
		`  IN_LIN: LINT:= 65535;`
		`  OUT_T: TIME;`
		`  XDONE: BOOL;`
		`END_VAR`

Пример программы		`XDONE:= SecToTime(`
		`  In:= IN_LIN,`
		`  Out=> OUT_T);`

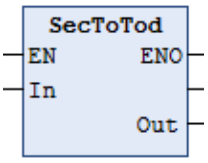
Результат операции	

2-7-22. Преобразование секунд во время SecToTod

(1) Обзор команды

Преобразование количества секунд, отсчитываемых с 0:0:00 с, во время.

Преобразование секунд во время [SecToTod]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SecToTod	Преобразование секунд во время		`SecToTod(In:= , Out=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Секунды	LINT	0-4294967295	0	Количество секунд, отсчитываемых с 0:0:00 с
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	TOD	В качестве типа данных	-	Время после преобразования

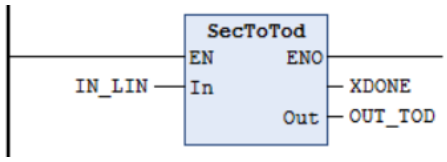
(3) Функция

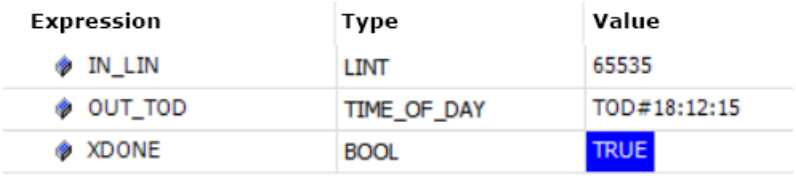
- Преобразование количества секунд "In", отсчитываемых с 0:0:00 с, во время. Если значение "In" больше 24 часов, разделите значение "In" на 24 часа, а затем преобразуйте остаток во время.

(4) Примечание

- Если значение "In" превышает допустимый диапазон, функция возвращает значение FALSE, а значение "Out" становится TOD#0:0:0.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">
<pre>VAR
   IN_LIN: LINT:= 65535;
   OUT_TOD: TIME_OF_DAY;
   XDONE: BOOL;
END_VAR</pre>
</td></tr>
<tr><td>Пример программы</td><td>


</td><td>
<pre>XDONE:= SecToTod(
    In:= IN_LIN,
    Out=> OUT_TOD);</pre>
</td></tr>
<tr><td>Результат операции</td><td colspan="2">


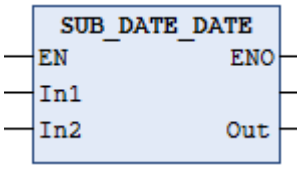
</td></tr>
</table>

2-7-23. Вычитание даты SUB_DATE_DATE

(1) Обзор команды

Вычитание даты из даты.

Вычитание даты [SUB_DATE_DATE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SUB_DATE_DATE	Вычитание даты		SUB_DATE_DATE(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Дата, которую вычитают	DATE	В качестве типа данных	-	Дата, которую вычитают
In2	Дата, из которой вычитают	DATE	В качестве типа данных	-	Дата, из которой вычитают
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Время в качестве результата вычитания	TIME	В качестве типа данных	-	Время в качестве результата вычитания

(3) Функция

- Вычитание даты "In1" из даты "In2", а результатом вычитания "Out" будет время.

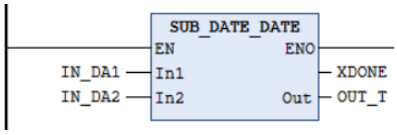
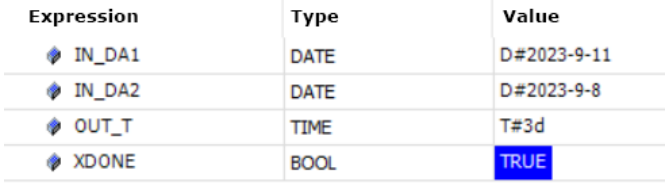
(4) Примечание

- Если результат вычитания превышает допустимый диапазон параметра "Out", программа столкнется с исключением. Выходное значение "Out" равно T#0 мс, и функция возвращает значение FALSE.
- Разница между датой "In1" и датой "In2" должна быть в пределах верхнего предела типа Time (T#49д17ч2м47с295мс), а значение "In1" должно быть больше, чем "In2", иначе программа столкнется с исключением, и функция вернет значение "FALSE".

Примечание: Когда из "In1" (близко к нижнему предельному значению) вычитают "In2" (близко к верхнему предельному значению), а разница между "In1" и "In2" находится в пределах верхнего предельного значения типа Time

(T#49д17ч2м47с295мс), программа не будет иметь никаких исключений. Например, вычитание значения нижнего предела (DATE#1970-1-1) из значения верхнего предела (DATE#2106-2-7) дает результат (T#1d).

(5) Пример программы

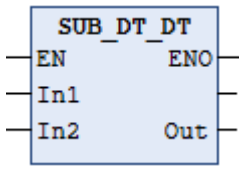
ДЕМО	LD	ST
Задание переменных	VAR IN_DA1: DATE:= DATE#2023-9-11; IN_DA2: DATE:= DATE#2023-9-8; OUT_T: TIME; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= SUB_DATE_DATE(In1:= IN_DA1, In2:= IN_DA2, Out=> OUT_T);
Результат операции		

2-7-24. Вычитание даты и времени SUB_DT_DT

(1) Обзор команды

Вычитание даты и времени из даты и времени.

Вычитание даты и времени [SUB_DT_DT]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SUB_DT_DT	Вычитание даты и времени		SUB_DT_DT(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Дата и время, которые вычитают	DT	В качестве типа данных	-	Дата и время, которые вычитают
In2	Дата и время, из которых вычитают	DT	В качестве типа данных	-	Дата и время, из которых вычитают
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Время в качестве результата вычитания	TIME	В качестве типа данных	-	Время в качестве результата вычитания

(3) Функция

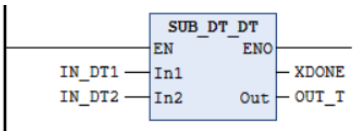
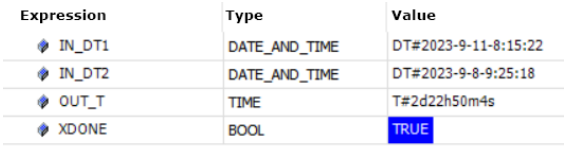
- Вычитание даты и времени "In1" из даты и времени "In2", а результатом вычитания "Out" будет время.

(4) Примечание

- Если результат вычитания превышает допустимый диапазон параметра "Out", программа столкнется с исключением. Выходное значение "Out" равно T#0 мс, и функция возвращает значение FALSE.
- Если значение вычитаемой даты "In1" меньше значения вычитаемой даты "In2", программа не будет иметь никаких исключений. Выполните следующие действия: DT#1970-1-1-0:0:0 - DT#2106-2-6-6:28:15 → T#1 д 1 с.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_DT1: DT:= DT#2023-9-11-8:15:22; IN_DT2: DT:= DT#2023-9-8-9:25:18; OUT_T: TIME; XDONE: BOOL; END_VAR	

Пример программы		XDONE:= SUB_DT_DT(In1:= IN_DT1, In2:= IN_DT2, Out=> OUT_T);
Результат операции		

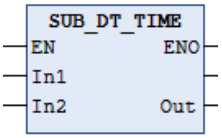
Expression	Type	Value
IN_DT1	DATE_AND_TIME	DT#2023-9-11-8:15:22
IN_DT2	DATE_AND_TIME	DT#2023-9-8-9:25:18
OUT_T	TIME	T#2d22h50m4s
XDONE	BOOL	TRUE

2-7-25. Вычитание времени из даты и времени SUB_DT_TIME

(1) Обзор команды

Вычитание времени из даты и времени.

Вычитание времени из даты и времени [SUB_DT_TIME]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SUB_DT_TIME	Вычитание времени из даты и времени		SUB_DT_TIME(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Дата и время, которые вычитают	DT	В качестве типа данных	-	Дата и время, которые вычитают
In2	Время, из которого вычитают	TIME	В качестве типа данных	-	Время, из которого вычитают

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Out	Дата и время в качестве результата вычитания	DT	В качестве типа данных	-	Дата и время в качестве результата вычитания

(3) Функция

- Вычитание времени "In2" из даты и времени "In1", а результатом вычитания "Out" будет дата и время.

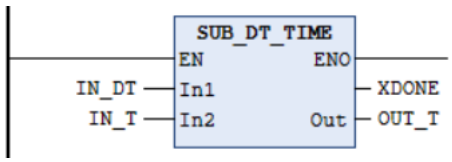
(4) Примечание

- Если результат вычитания превышает допустимый диапазон "Out", программа не будет иметь никаких исключений. Выполните следующие действия: DT#1970-1-1-0:0:0 - T#1 д → DT#2106-2-6-6:28:15.
- Минимальная единица измерения значения "In2" должна быть в с (больше или равно 1000 мс). Если "In2" меньше 1000 мс, значение "Out" соответствует значению "In1".

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
   IN_DT: DT:= DT#2023-9-11-9:21:11;
   IN_T: TIME:= T#1D1H1M1S;
   OUT_T: DT;
   XDONE: BOOL;
END_VAR
```




```
XDONE:= SUB_DT_TIME(
    In1:= IN_DT,
    In2:= IN_T,
    Out=> OUT_T);
```

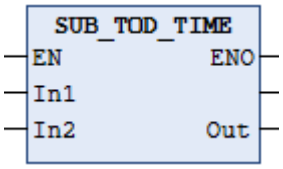
Expression	Type	Value
IN_DT	DATE_AND_TIME	DT#2023-9-11-9:21:11
IN_T	TIME	T#1d1h1m1s
OUT_T	DATE_AND_TIME	DT#2023-9-10-8:20:10
XDONE	BOOL	TRUE

2-7-26. Вычитание времени из времени SUB_TOD_TIME

(1) Обзор команды

Вычитание времени из времени.

Вычитание времени из времени [SUB_TOD_TIME]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SUB_TOD_TIME	Вычитание времени из времени		SUB_TOD_TIME(In1:= , In2:= , Out=>);

Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Время, которое вычитают	TOD	В качестве типа данных	-	Время, которое вычитают
In2	Время, из которого вычитают	TIME	В качестве типа данных	-	Время, из которого вычитают
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Время в качестве результата вычитания	TOD	В качестве типа данных	-	Время в качестве результата вычитания

(2) Функция

- Вычитание времени "In2" из времени "In1", а результатом вычитания "Out" будет время.

(3) Примечание

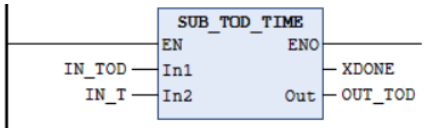
- TOD#0:0:0 - T#1мс → TOD#23:59:59:999 Если результат вычитания превышает допустимый диапазон "Out", программа не будет иметь никаких исключений. Выполните следующие действия: TOD#0:0:0 - T#1 мс → TOD#23:59:59:999.

(4) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   IN_TOD: TOD:= TOD#18:31:29;
   IN_T: TIME:= T#3H11M2S;
   OUT_TOD: TOD;
   XDONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
XDONE:= SUB_TOD_TIME(
    In1:= IN_TOD,
    In2:= IN_T,
    Out=> OUT_TOD);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
IN_TOD	TIME_OF_DAY	TOD#18:31:29
IN_T	TIME	T#3h11m2s
OUT_TOD	TIME_OF_DAY	TOD#15:20:27
XDONE	BOOL	TRUE

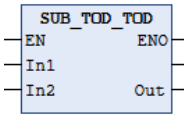
</td></tr>
</table>

2-7-27. Вычитание времени SUB_TOD_TOD

(1) Обзор команды

Вычитание времени из времени.

Вычитание времени [SUB_TOD_TOD]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
SUB_TOD_TOD	Вычитание времени		SUB_TOD_TOD(In1:= , In2:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In1	Время, которое вычитают	TOD	В качестве типа данных	-	Время, которое вычитают
In2	Время, из которого вычитают	TOD	В качестве типа данных	-	Время, из которого вычитают
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Время в качестве результата вычитания	TIME	В качестве типа данных	-	Время в качестве результата вычитания

(3) Функция

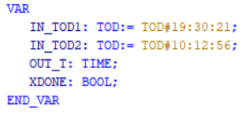
- Вычитание времени "In1" из времени "In2", а результатом вычитания "Out" будет время.

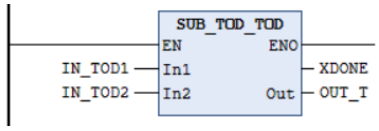
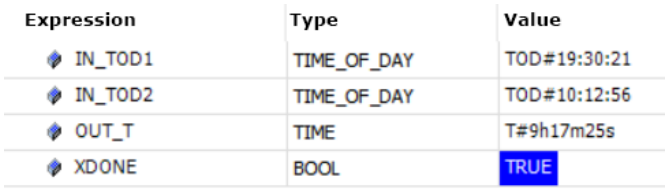
(4) Примечание

- Если значение вычитаемого времени "In1" меньше значения вычитаемого времени "In2", программа не будет иметь никаких исключений. Выполните следующие действия:

TOD#13:34:56.789 - TOD#13:35:56.789 → T#23 ч 59 м

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		

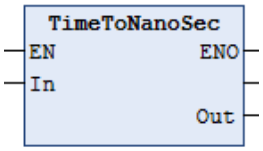
Пример программы		XDONE:= SUB_TOD_TOD(In1:= IN_TOD1, In2:= IN_TOD2, Out=> OUT_T);
Результат операции		

2-7-28. Преобразование времени в наносекунды TimeToNanoSec

(1) Обзор команды

Преобразование времени в наносекунды.

Преобразование времени в наносекунды [TimeToNanoSec]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TimeToNanoSec	Преобразование времени в наносекунды		TimeToNanoSec(In:= , Out=>);

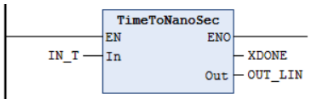
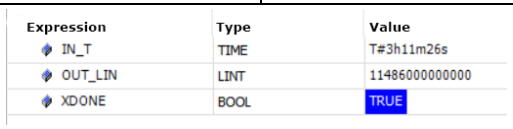
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Время	TIME	В качестве типа данных	T#0 c	Время
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	LINT	В качестве типа данных	-	Наносекунды

(3) Функция

- Преобразование времени "In" в наносекунды.

(4) Пример программы

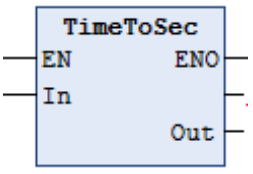
ДЕМО	LD	ST
Задание переменных		VAR IN_T: TIME:= T#3H11M26S; OUT_LIN: LINT; XDONE: BOOL; END_VAR
Пример программы		XDONE:= TimeToNanoSec(In:= IN_T, Out=> OUT_LIN);
Результат операции		

2-7-29. Преобразование времени в секунды TimeToSec

(1) Обзор команды

Преобразование времени в секунды.

Преобразование времени в секунды [TimeToSec]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TimeToSec	Преобразование времени в секунды		TimeToSec(In:= , Out=>);

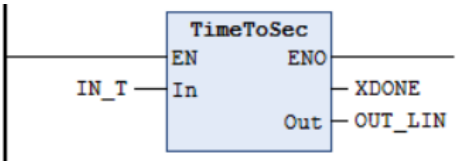
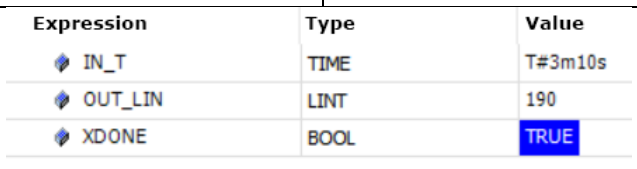
(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Время	TIME	В качестве типа данных	T#0 с	Время
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	LINT	В качестве типа данных	-	Секунды

(3) Функция

- Преобразование времени "In" в секунды. Значения менее 1 секунды округляются.

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_T: TIME:= T#3M10S; OUT_LIN: LINT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= TimeToSec(In:= IN_T, Out=> OUT_LIN);
Результат операции		

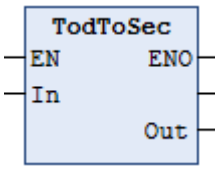
2-7-30. Преобразование времени в секунды TodToSec

(1) Обзор команды

Преобразование времени в количество секунд, отсчитываемых с 0:0:00 с.

Преобразование времени в секунды [TodToSec]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TodToSec	Преобразование времени в секунды		`TodToSec(In:= , Out=> );`

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Время	TOD	В качестве типа данных	TOD#0:0:0	Время
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат преобразования	LINT	0-86399	-	Секунды

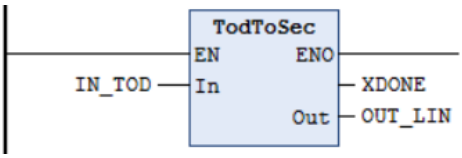
(3) Функция

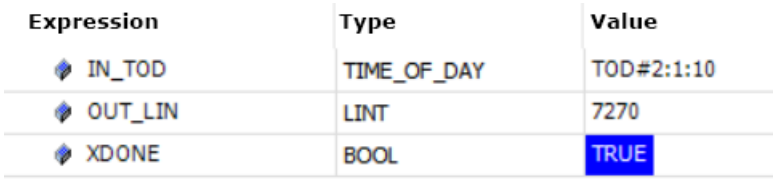
- Преобразование времени "In" в количество секунд, отсчитываемых с 0:0:0:0. Преобразованное значение измеряется в секундах. Значения менее 1 секунды округляются.

(4) Пример программы

ДЕМО	LD	ST

Задание переменных		`VAR`
		`  IN_TOD: TOD:= TOD#2:1:10;`
		`  OUT_LIN: LINT;`
		`  XDONE: BOOL;`
		`END_VAR`

Пример программы		`XDONE:= TodToSec(`
		`  In:= IN_TOD,`
		`  Out=> OUT_LIN);`

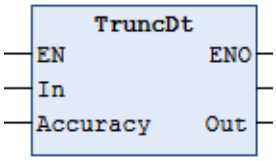
Результат операции	

2-7-31. Округление даты и времени TruncDt

(1) Обзор команды

Округление значений, меньших указанной единицы, в переменных типа DT.

Округление даты и времени [TruncDt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TruncDt	Округление даты и времени		TruncDt(In:= , Accuracy:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Заданные дата и время	DT	В качестве типа данных	-	Отброшенные объекты
Accuracy	Единица округления	_eSUBSEC	В качестве типа данных	_SEC	Минимальная зарезервированная единица времени
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результат обработки	DT	В качестве типа данных	-	Дата и время после округления

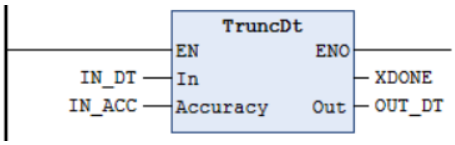
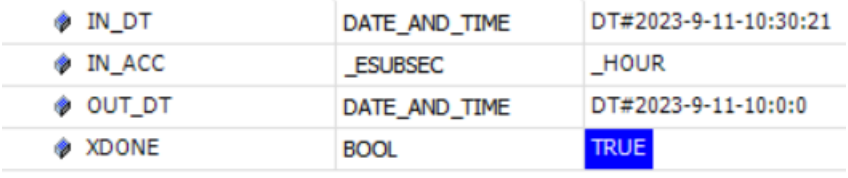
(3) Функция

- В объекте с датой и временем "In" значения, меньшие, чем единица округления "Accuracy", будут округлены, и округленное значение будет сохранено в округленной переменной даты и времени "Out".
- Типом данных переменной "Accuracy" является элемент перечисления _eSUBSEC. Значения элементов перечисления:

Элемент перечисления	Значение
_MILLISEC	Миллисекунда
_SEC	Секунда
_MINUTE	Минута
_HOUR	Час

(4) Примечание

(5) Значение тела элемента перечисления нельзя устанавливать на _MILLISEC, иначе программа столкнется с исключением, и функция вернет значение "FALSE".Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_DT: DT:= DT#2023-9-11-10:30:21; IN_ACC: _eSUBSEC:=_eSUBSEC._HOUR; OUT_DT: DT; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= TruncDt(In:= IN_DT, Accuracy:= IN_ACC, Out=> OUT_DT);
Результат операции		

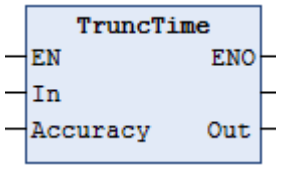
Expression	Type	Value
IN_DT	DATE_AND_TIME	DT#2023-9-11-10:30:21
IN_ACC	_ESUBSEC	_HOUR
OUT_DT	DATE_AND_TIME	DT#2023-9-11-10:0:0
XDONE	BOOL	TRUE

2-7-32. Округление времени TruncTime

(1) Обзор команды

Округление значений, меньших указанной единицы, в переменных типа TIME.

Округление времени [TruncTime]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TruncTime	Округление времени		TruncTime(In:= , Accuracy:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Заданное время	TIME	В качестве типа данных	-	Округление заданной единицы времени
Accuracy	Единица округления	_eSUBSEC	В качестве типа данных	_SEC	Минимальная сохраненная единица времени
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результаты после обработки	TIME	В качестве типа данных	-	Время после округления

(3) Функция

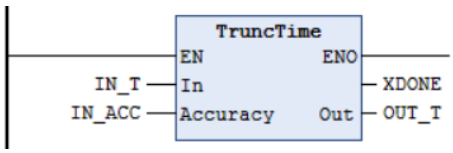
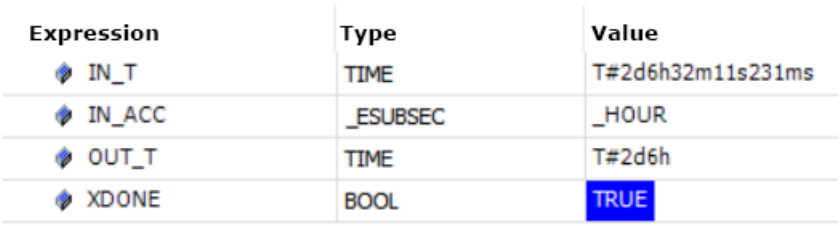
- В объекте со временем "In" значения, меньшие, чем единица округления

"Accuracy", будут округлены, и округленное значение будет сохранено в округленной переменной времени "Out".

- Типом данных переменной "Accuracy" является элемент перечисления _eSUBSEC. Значения элементов перечисления:

Элемент перечисления	Значение
_MILLISEC	Миллисекунда
_SEC	Секунда
_MINUTE	Минута
_HOUR	Час

(4) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR IN_T: TIME:= T#2D6H32M11S231MS; IN_ACC: _eSUBSEC:= _eSUBSEC._HOUR; OUT_T: TIME; XDONE: BOOL; END_VAR	
Пример программы		XDONE:= TruncTime(In:= IN_T, Accuracy:= IN_ACC, Out=> OUT_T);
Результат операции		

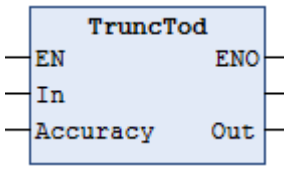
2-7-33. Округление времени TruncTod

(1) Обзор команды

Округление значений, меньших указанной единицы, в переменных типа TOD.

Округление времени [TruncTod]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TruncTod	Округление времени		TruncTod(In:= , Accuracy:= , Out=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
In	Заданное время	TOD	В качестве типа данных	-	Округление заданной единицы времени
Accuracy	Единица округления	_eSUBSEC	В качестве типа данных	_SEC	Минимальная сохраненная единица времени
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Out	Результаты после обработки	TOD	В качестве типа данных	-	Время после округления

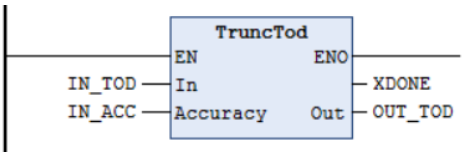
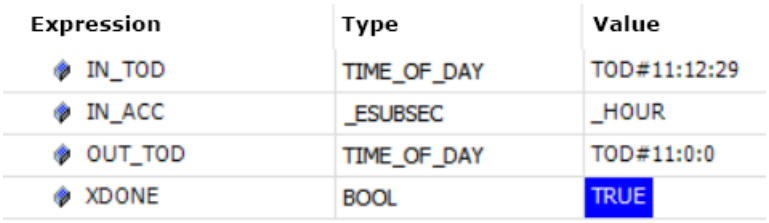
(3) Функция

- В объекте со временем "In" значения, меньшие, чем единица округления "Accuracy", будут округлены, и округленное значение будет сохранено в округленной переменной времени "Out".
- Типом данных переменной "Accuracy" является элемент перечисления _eSUBSEC. Значения элементов перечисления:

Элемент перечисления	Значение
_MILLISEC	Миллисекунда
_SEC	Секунда

_MINUTE	Минута
_HOUR	Час

(4) Пример программы

ДЕМО	LD	ST
Задание переменных		VAR IN_TOD: TOD:= TOD#11:12:29; IN_ACC: _eSUBSEC:= _eSUBSEC._HOUR; OUT_TOD: TOD; XDONE: BOOL; END_VAR
Пример программы		XDONE:= TruncTod(In:= IN_TOD, Accuracy:= IN_ACC, Out=> OUT_TOD);
Результат операции		

2-8. Команды связи

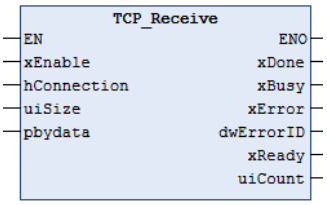
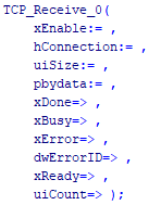
2-8-1. Связь по протоколу TCP

2-8-1-1. Прием данных при связи по протоколу TCP TCP_Receive

(1) Обзор команды

Прием данных при связи по протоколу TCP.

Прием данных при связи по протоколу TCP [TCP_Receive]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TCP_Receive	Прием данных при связи по протоколу TCP		

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xEnable	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
hConnection	Дескриптор соединения	DWORD	В качестве типа данных	-	Дескриптор соединения для связи по протоколу TCP
uiSize	Размер принимаемых данных	UINT	В качестве типа данных	-	Размер области принятых данных (BYTE)
pbyData	Указатель области приема	POINTER TO BYTE	-	-	Указатель на начальный байт области приема
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
xDone	Сигнал окончания	BOOL	FALSE/TRUE	FALSE	Сигнал завершения приема данных
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки
xReady	Признак успешно	BOOL	FALSE/TRUE	FALSE	Считывание данных из буфера, и если буфер

	установленного соединения				данных не пустой, установка признака на один цикл сканирования
uiCount	Дескриптор соединения	_UXINT	-	-	Считывание фактического объема данных из приемного буфера

(3) Функция

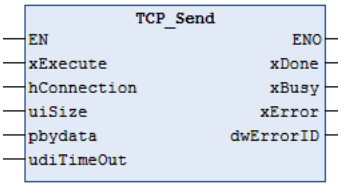
- Если переменная xEnable команды TCP_Receive обнаружит высокий уровень мощности, она попытается считать данные из буфера связи по протоколу TCP и установит переменную xBusy в состояние True.
- Если считывание данных было выполнено успешно, считанные данные будут помещены в массив с pbyData в качестве первого адреса, а uiCount выведет фактический объем считанных данных. Переменная xReady будет установлена в состояние TRUE, и две переменные будут обеспечивать цикл сканирования.
- Если в период считывания данных связь по сети будет нарушена, что приведет к сбою в считывании данных, переменная xDone будет установлена в состояние True, и функция считывания данных не сработает. xEnable=True необходимо запустить повторно. Даже можно повторно запустить xEnable=True функционального блока TCP_Connection или TCP_Client.

2-8-1-2. Передача данных при связи по протоколу TCP TCP_Send

(1) Обзор команды

Передача данных при связи по протоколу TCP.

Передача данных при связи по протоколу TCP [TCP_Send]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TCP_Send	Передача данных при связи по протоколу TCP		TCP_Send_0(xExecute:= , hConnection:= , uiSize:= , pbydata:= , udiTimeOut:= , xDone=> , xBusy=> , xError=> , dwErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xExecute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
hConnection	Дескриптор соединения	_XWORD	-	-	Дескриптор соединения для связи по протоколу TCP
uiSize	Размер передаваемых данных	UINT	В качестве типа данных	-	Размер области переданных данных (BYTE)
pbyData	Указатель на область передачи	POINTER TO BYTE	-	-	Указатель на начальный байт области передачи
UditimeOut	Таймаут	UDINT	В качестве типа данных	500000	Таймаут передачи данных, единица измерения - мкс, значение по умолчанию - 500 мс
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
xDone	Сигнал о завершении	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении передачи данных
xBusy	Сигнал занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости

xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки

(3) Функция

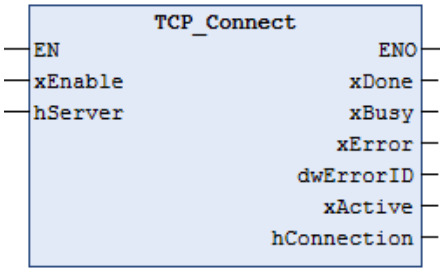
- Если в переменной xExecute команды TCP_Send будет обнаружен нарастающий фронт импульса, она отправит данные длиной uiSize, начиная с установленного пользователем буфера передачи pbyData, на целевое устройство. Если передача данных прошла успешно в течение периода таймаута, xDone устанавливается в True.

2-8-1-3. Создание TCP-соединения TCP_Connect

(1) Обзор команды

Создание TCP-соединения и подключение к серверу.

Создание TCP-соединения [TCP_Connect]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TCP_Connect	Создание TCP-соединения и подключение к серверу		TCP_Connect_0(xEnable:= , hServer:= , xDone=> , xBusy=> , xError=> , dwErrorID=> , xActive=> , hConnection=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xEnable	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
hServer	Дескриптор сервера	_XWORD	-	-	Дескриптор TCP-сервера
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
xDone	Сигнал завершения	BOOL	FALSE/TRUE	FALSE	Сигнала завершения прослушивания
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки
xActive	Признак установления соединения	BOOL	FALSE/TRUE	FALSE	Признак прослушивания сервером запросов на подключения от удаленных клиентов
hConnection	Дескриптор соединения	_XWORD	-	-	Дескриптор соединения для связи между сервером и клиентом

(3) Функция

- Если в переменной xEnable команды TCP_Connect будет обнаружен высокий уровень мощности, локальный сервер будет прослушивать запросы на подключение от удаленных клиентов. После успешного подключения клиента к серверу между сервером и удаленным клиентом будет создан дескриптор соединения для связи hConnection.

(4) Примечание

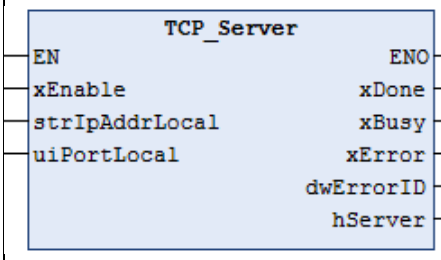
- После установки связи, xActive принимает состояние True, а когда связь прерывается, xActive автоматически сбрасывается.
- Для контроля связи требуется взаимодействие с функциональным блоком TCP_Receive, которое может быть успешным только в том случае, если TCP_Receive.xBusy принимает состояние True.

2-8-1-4. Создание серверной коммуникационной службы для TCP соединений TCP_Server

(1) Обзор команды

Создание серверной коммуникационной службы для TCP соединений.

Создание серверной коммуникационной службы для TCP соединений [TCP_Server]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TCP_Server	Создание серверной коммуникационной службы для TCP соединений		TCP_Server_0(xEnable:= , strIpAddrLocal:= , uiPortLocal:= , xDone=> , xBusy=> , xError=> , dwErrorID=> , hServer=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xEnable	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды

strIpAddrLocal	Локальный IP-адрес	STRING	В качестве типа данных	-	IP-адрес локального сетевого порта связи ПЛК
uiPortLocal	Локальный порт	UINT	В качестве типа данных	-	Номер порта локальной связи
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
xDone	Признак завершения	BOOL	FALSE/TRUE	FALSE	Признак завершения связи со стороны сервера
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки
hServer	Дескриптор сервера	_XWORD	-	-	Дескриптор TCP-сервера

(3) Функция

- Команда создания серверной коммуникационной службы для TCP соединений. Если переменная xEnable команды TCP_Server имеет значение true, будет создан допустимый дескриптор для TCP-связи между локальным сервером и удаленным клиентом (значение hServer не равно 0), а xBusy останется в состоянии true. Если во время связи возникнет ошибка в функции, связанной со службой, xDone примет состояние True, функциональные блоки, связанные с сервером, нельзя будет использовать, и команду TCP_Server нужно будет запустить повторно.

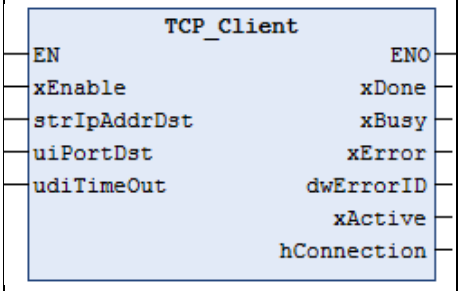
2-8-1-5. Создание клиентской коммуникационной службы для TCP соединений TCP_Client

(1) Обзор команды

Создание клиентской коммуникационной службы для TCP соединений.

Создание клиентской коммуникационной службы для TCP соединений [TCP_Client]			
Условие	Нормальное	Совместимые	PMP3xx

выполнения	включение/выключение триггера	модели	
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	Представление на языке ST
TCP_Client	Создание клиентской коммуникационной службы для TCP соединений		TCP_Client_0(xEnable:= , strIpAddrDst:= , uiPortDst:= , udiTimeOut:= , xDone=> , xBusy=> , xError=> , dwErrorID=> , xActive=> , hConnection=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xEnable	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
strIpAddrDst	IP-адрес удаленного устройства	STRING	В качестве типа данных	-	IP-адрес удаленного устройства
uiPortDst	Номер порта удаленного устройства	UINT	В качестве типа данных	-	Номер порта удаленного устройства
udiTimeOut	Таймаут	UDINT	В качестве типа данных	500000	Таймаут запроса соединения, мкс, значение по умолчанию — 500 мс

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
xDone	Сигнал завершения	BOOL	FALSE/TRUE	FALSE	Признак завершения связи с клиентом
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки
xActive	Признак успешно установленного соединения	BOOL	FALSE/TRUE	FALSE	Бит признака успешного соединения между клиентом и удаленным сервером
hConnection	Дескриптор соединения	_XWORD	-	-	Дескриптор соединения для связи между сервером и клиентом

(3) Функция

- Команда создания клиентской коммуникационной службы для TCP соединений Если в переменной xEnable команды TCP_Client будет обнаружен высокий уровень мощности, локальный клиент будет запрашивать подключение к удаленному серверу. После успешного подключения клиента к серверу между сервером и удаленным клиентом будет создан признак установления соединения для связи hConnection, а xActive примет состояние True.

2-8-1-6. Пример команды для установления связи по протоколу TCP

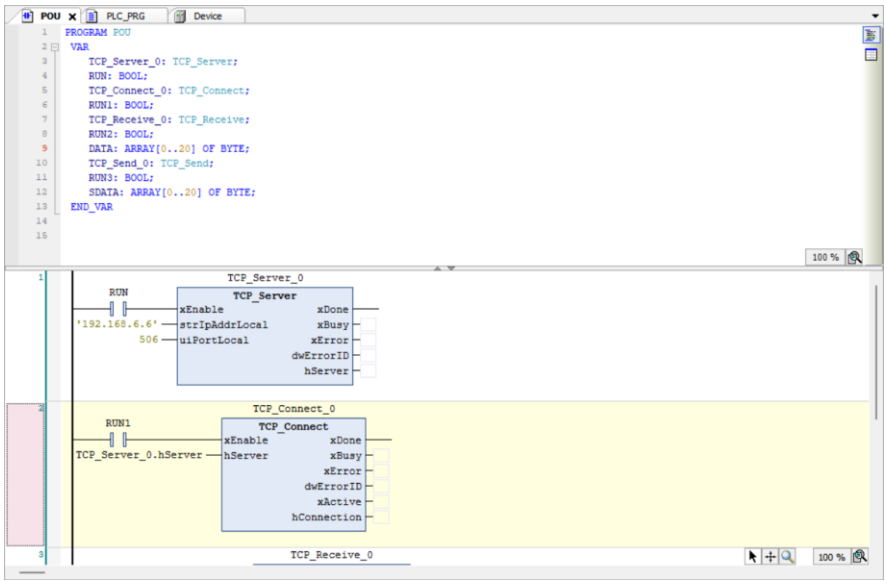
PMP30-60A32 в роли сервера, помощник отладки в роли клиента

IP-адрес PMP30-60A32 — 192.168.6.6

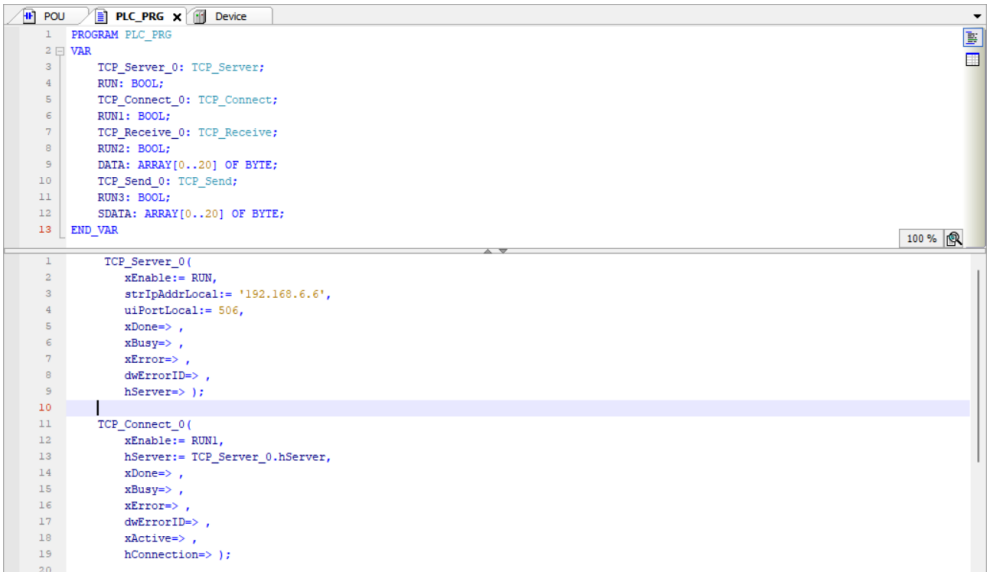
1. Создайте серверные коммуникационные службы для TCP соединений и

установите соединения между клиентами и серверами. Запустите команды TCP Server и TCP Connect, при этом входной контакт hServer команды TCP Connect является выходным контактом hServer команды TCP Server.

2. При установке соединения с помощником отладки проверьте правильность IP-адреса и номер порта. В этом примере номер порта — 506.



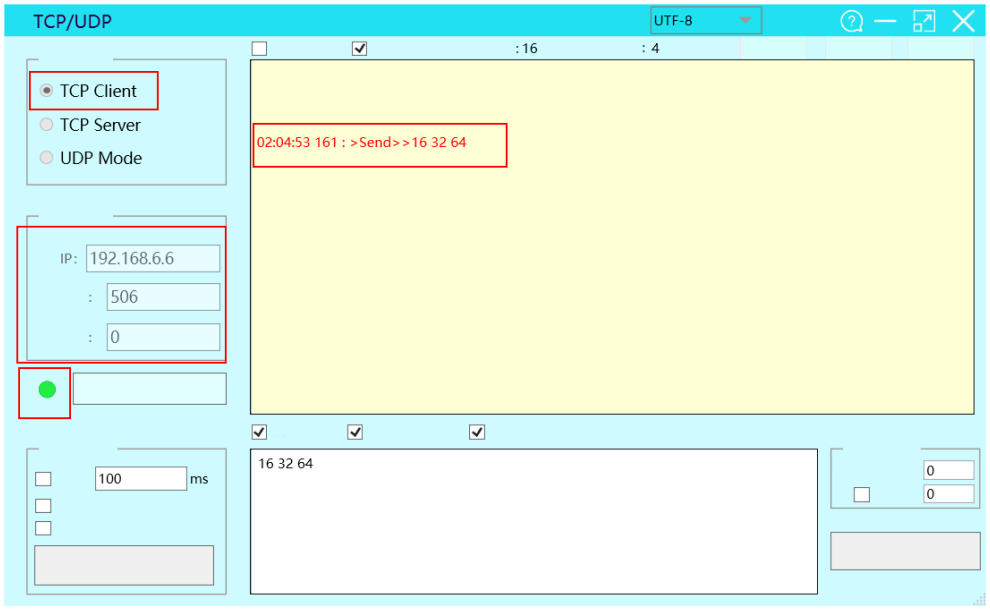
Форма LD



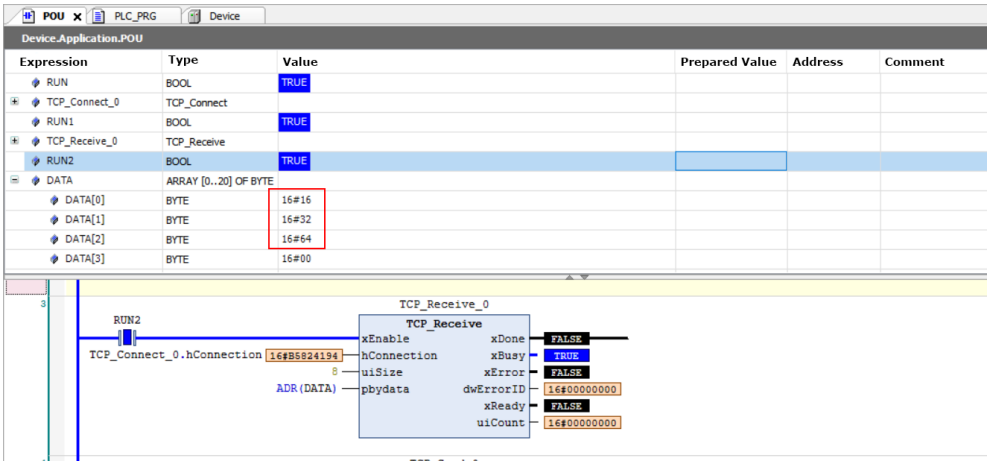
Форма ST

3. Сервер принимает данные, отправленные помощником отладки, выступающим в роли клиента.

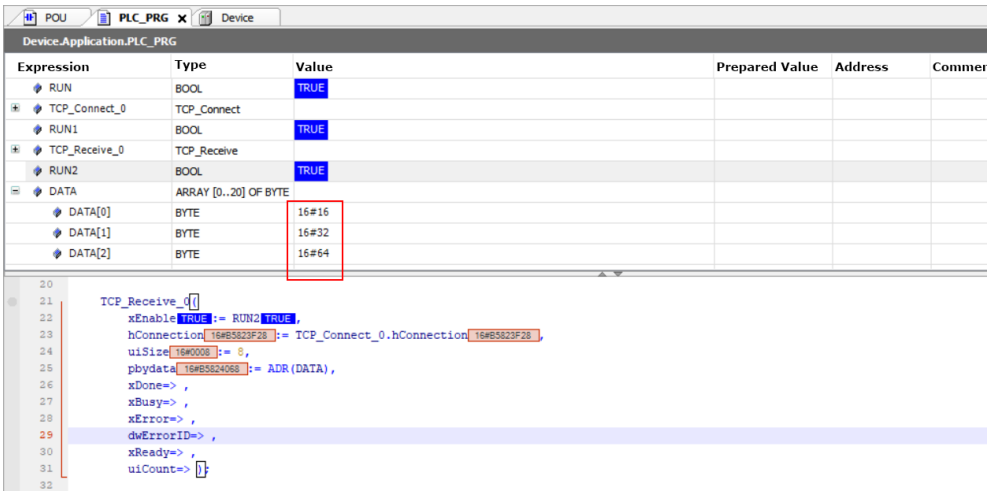
① Передайте шестнадцатеричные данные 16#16 16#32 16#64 через помощник отладки.



② Запустите команду TCP_Receive, после чего принятые данные будут сохранены по адресу, на который указывает указатель pbyData, как показано на рисунке ниже. Принятые данные: 16#16, 16#32, 16#64.



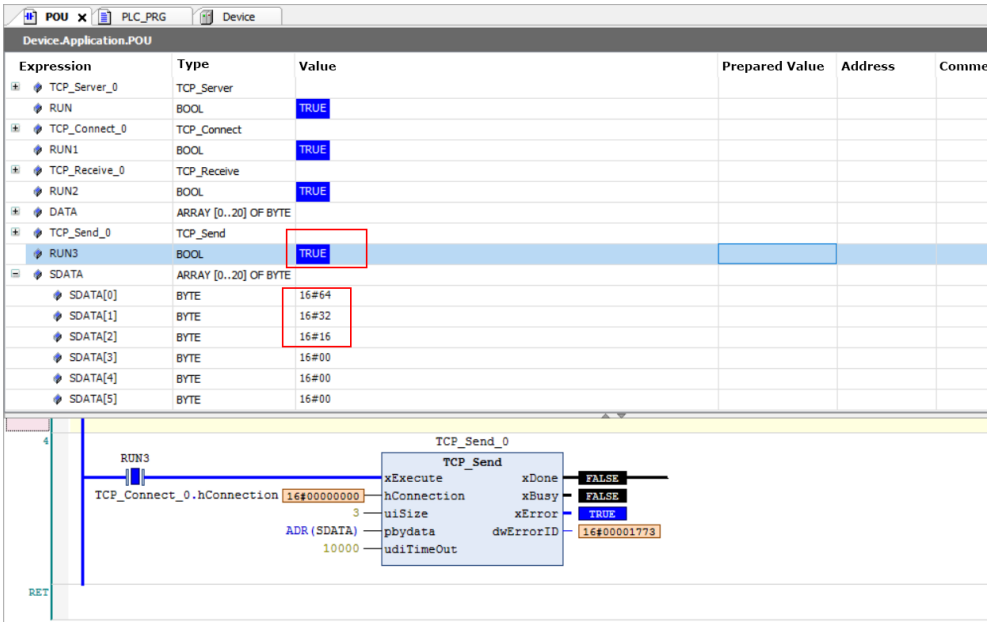
Форма LD



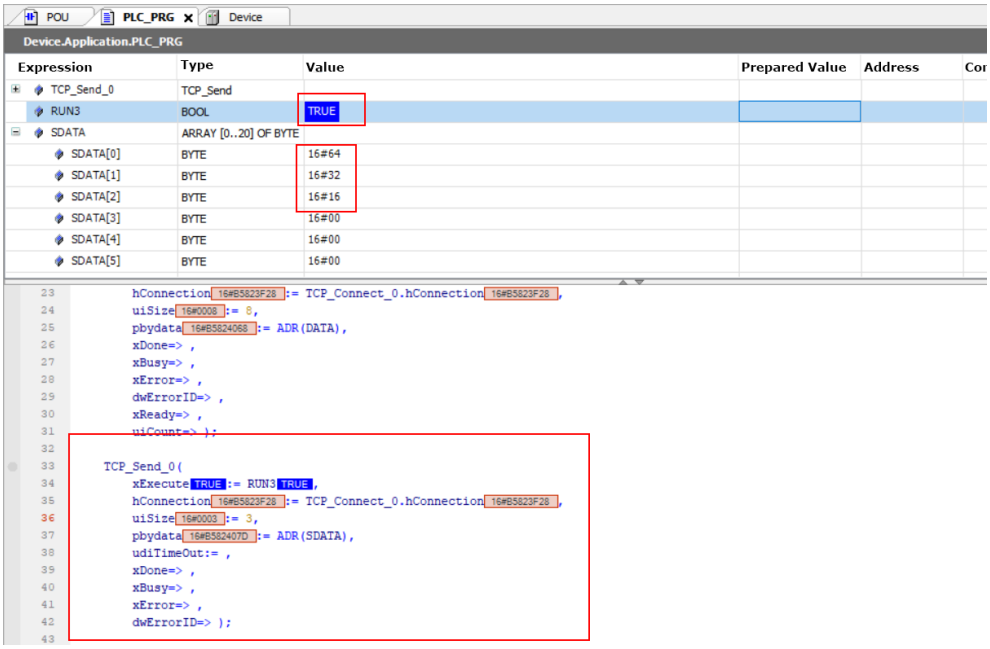
Форма ST

4. Сервер передает данные в помощник отладки, выступающий в роли клиента.

① Запустите команду TCP_Send для передачи данных, сохраненных по адресу, на который указывает указатель pbyData, как показано на рисунке ниже. Передача данных 16#64 16#32 16#16.

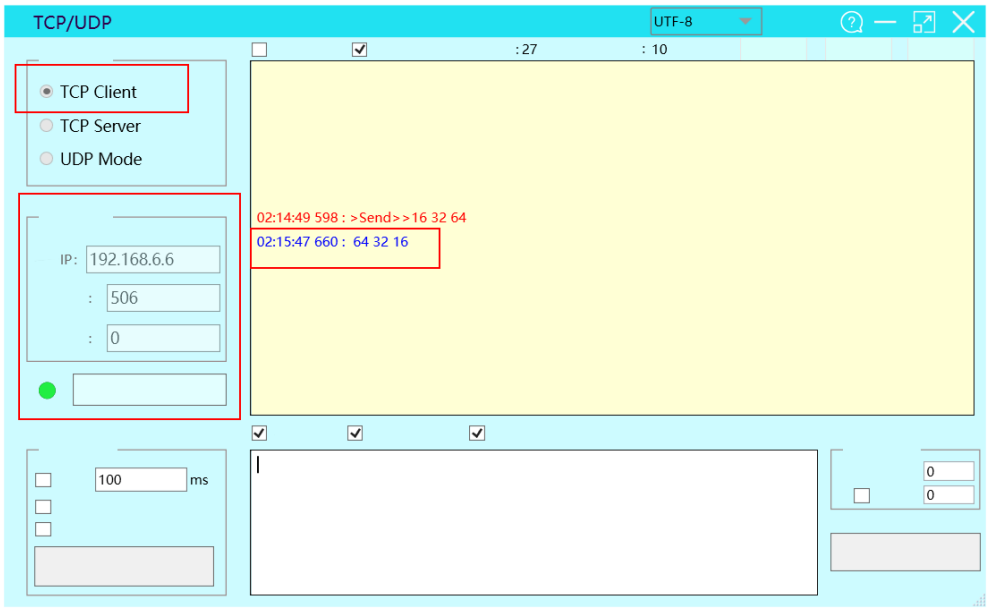


Форма LD



Форма ST

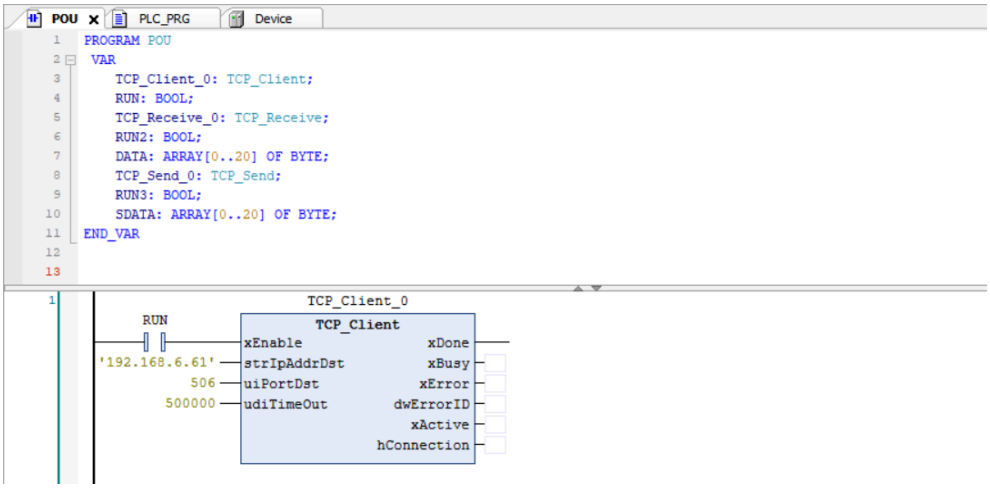
② На рисунке ниже показан помощник отладки клиента, принимающий данные 16#64 16#32 16#16, отправленные сервером ПЛК.



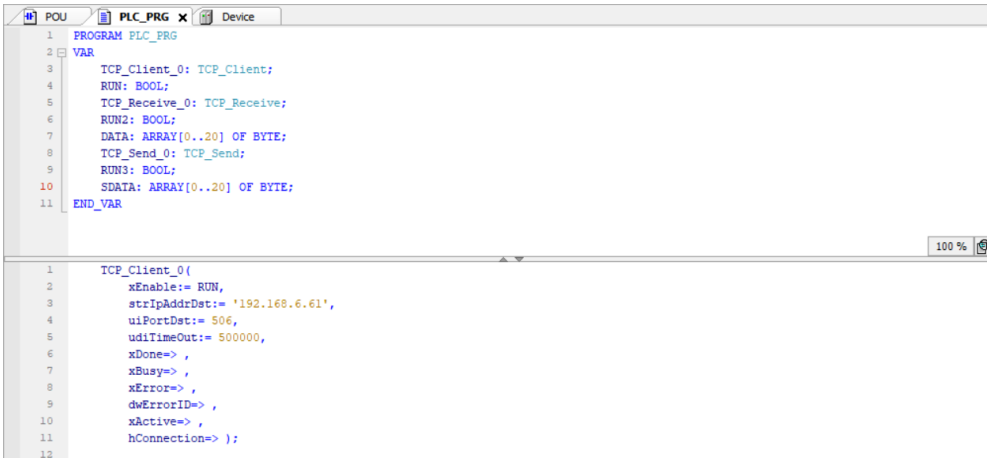
PMP30-60A32 в роли клиента, помощник отладки в роли сервера

1. Создайте клиентскую коммуникационную службу для TCP соединений и установите соединение с помощником отладки.

① Запустите команду TCP_Client и проверьте, что IP-адрес и номер порта являются адресом и номером порта удаленного сервера (IP-адрес 192.168.6.1 помощника отладки — это IP-адрес порта ПК, подключенного к ПЛК). Перед запуском команды откройте помощник отладки, выступающий в роли сервера.

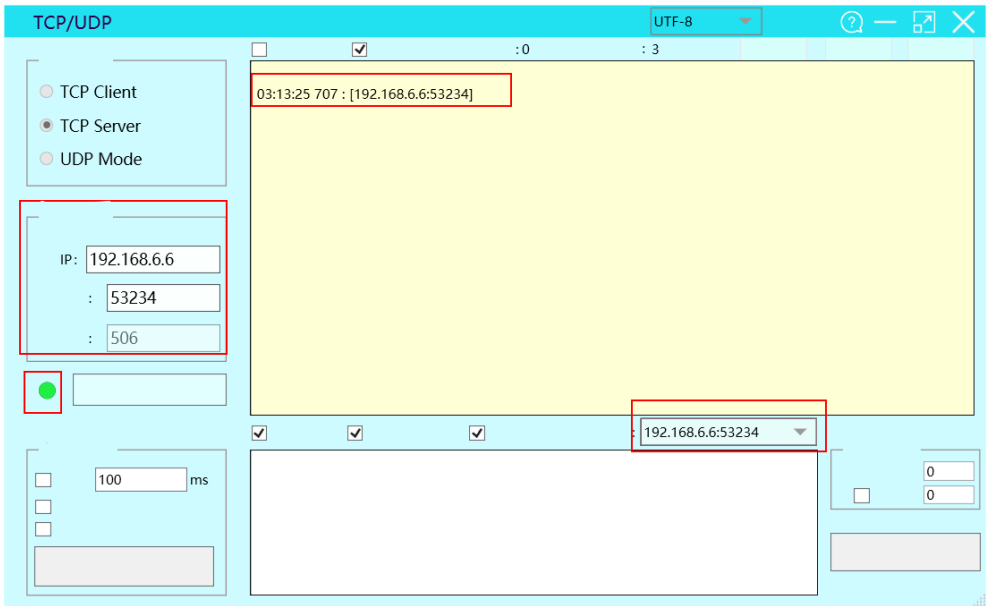


Форма LD



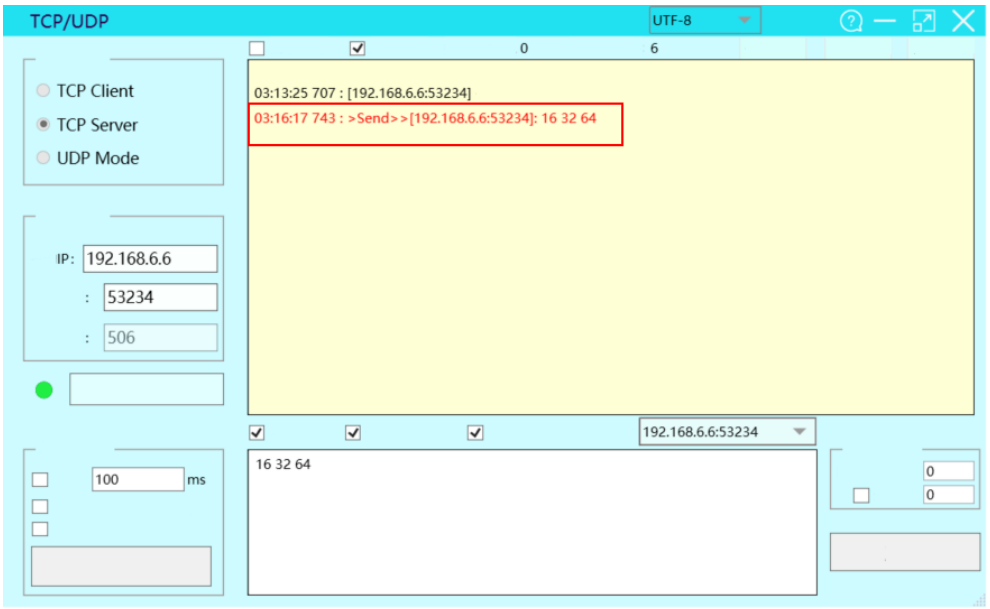
Форма ST

2. Сначала откройте помощник отладки, выступающий в роли сервера, а затем запустите упомянутую выше команду TCP_Client, чтобы создать клиентскую коммуникационную службу. Соединение успешно установлено, как показано на рисунке, приведенном ниже.

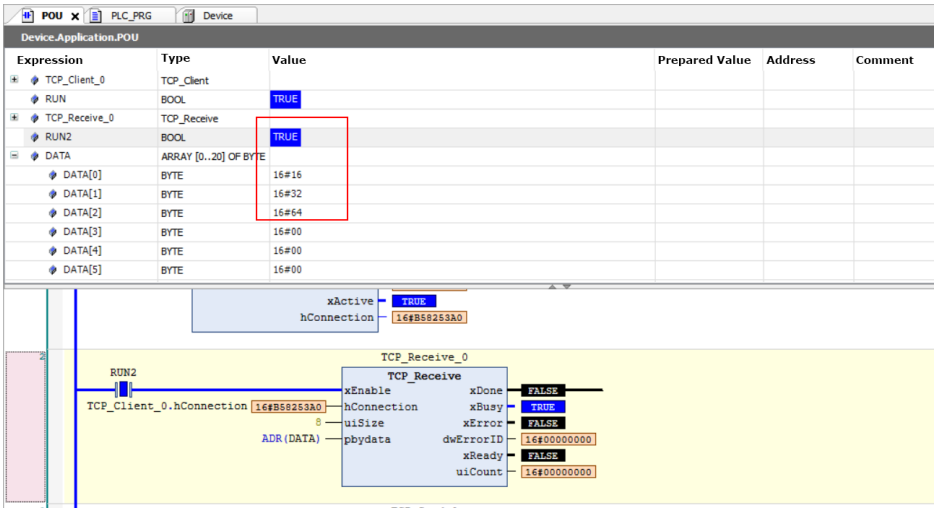


3. Клиент принимает данные, отправленные помощником отладки, выступающим в роли сервера.

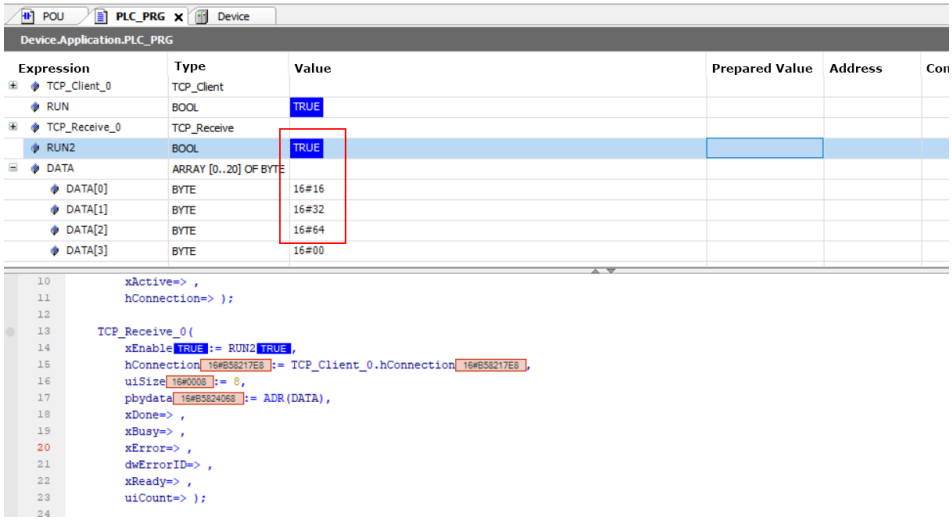
① Передайте шестнадцатеричные данные 16#16 16#32 16#64 через помощник отладки.



② Запустите команду TCP_Receive, после чего принятые данные будут сохранены по адресу, на который указывает указатель pbyData. Клиент принял данные 16#16 16#32 16#64, отправленные сервером, как показано на рисунке ниже. Данные приняты успешно.



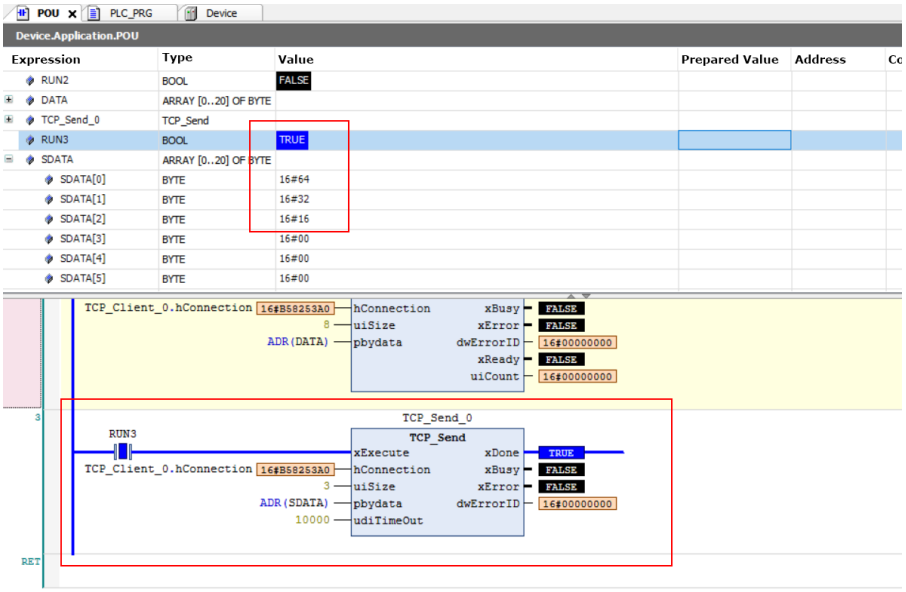
Форма LD



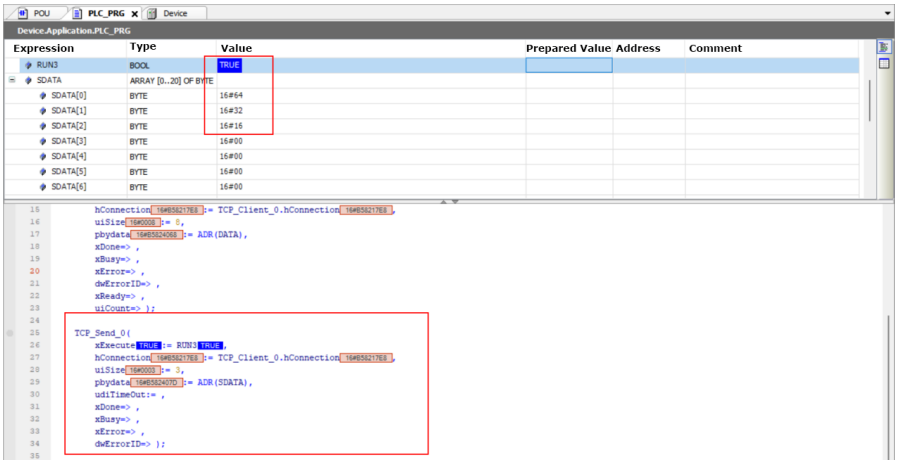
Форма ST

4. Клиент передает данные в помощник отладки, выступающий в роли сервера.

① Запустите команду TCP_Send для передачи данных, сохраненных по адресу, на который указывает указатель pbyData.

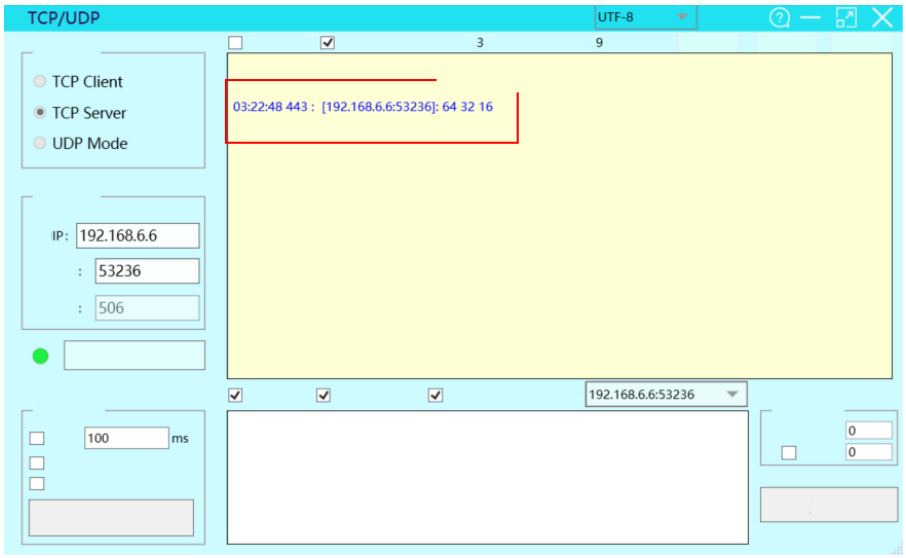


Форма LD



Форма ST

② Помощник отладки на стороне сервера принял данные 16#64 16#32 16#16, переданные клиентом ПЛК, как показано на рисунке ниже. Передача данных выполнена успешно.



2-8-2. Связь по протоколу UDP

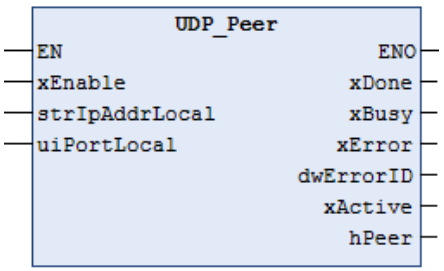
2-8-2-1. Создание соединения для связи по протоколу UDP UDP_Peer

(1) Обзор команды

Создание соединения для связи по протоколу UDP.

Создание соединения для связи по протоколу UDP [UDP_Peer]			
Условие выполнения	Запуск в нормально-разомкнутом/нормально-замкнутом состоянии	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
UDP_Peer	Создание соединения для связи по протоколу UDP		UDP_Peer_0(xEnable:= , strIpAddrLocal:= , uiPortLocal:= , xDone=> , xBusy=> , xError=> , dwErrorID=> , xActive=> , hPeer=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xEnable	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
strIpAddrLocal	Локальный IP-адрес	STRING(80)	В качестве типа данных	-	IP-адрес локального сетевого порта связи ПЛК
uiPortLocal	Номер локального порта	UINT	В качестве типа данных	-	Номер порта локальной связи
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
xDone	Признак завершения	BOOL	FALSE/TRUE	FALSE	Признак завершения связи
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Признак занятости

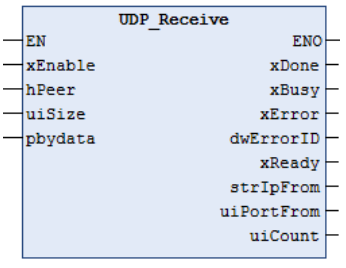
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Признак ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки
xActive	Признак успешной связи	BOOL	FALSE/TRUE	FALSE	Создание признака успешной связи по протоколу UDP
hpeer	Дескриптор связи	_XWORD	-	-	Дескриптор связи по протоколу UDP

2-8-2-2. Прием данных при связи по протоколу UDP UDP_Receive

(1) Обзор команды

Прием данных при связи по протоколу UDP.

Прием данных при связи по протоколу UDP [UDP_Receive]			
Условие выполнения	Запуск в нормально-разомкнутом/нормально-замкнутом состоянии	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
UDP_Receive	Прием данных при связи по протоколу UDP		UDP_Receive_0(xEnable:= , hPeer:= , uiSize:= , pbydata:= , xDone=> , xBusy=> , xError=> , dwErrorID=> , xReady=> , strIpFrom=> , uiPortFrom=> , uiCount=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xEnable	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
hPeer	Дескриптор связи	_XWORD	-	-	Дескриптор связи по протоколу UDP
uiSize	Размер принимаемых данных	UINT	В качестве типа данных	-	Размер области принятых данных (BYTE)
pbyData	Указатель на область приема	POINTER TO BYTE	-	-	Указатель на начальный байт области приема
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
xDone	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал завершения приема данных
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки
xReady	Признак успешного приема данных	BOOL	FALSE/TRUE	FALSE	Считывание данных из буфера, и если буфер данных не пустой, установка признака на один цикл сканирования
strIpFrom	IP-адрес источника текущего пакета	STRING(80)	В качестве типа данных	-	IP-адрес источника текущего пакета
uiPortFrom	Порт источника текущего пакета	UINT	В качестве типа данных	-	Порт источника текущего пакета

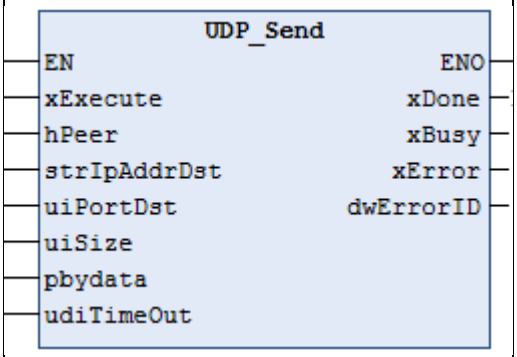
uiCount	Объем данных	_UXINT	-	-	Считывание фактического объема данных из буфера

2-8-2-3. Передача данных при связи по протоколу UDP_Send

(1) Обзор команды

Передача данных при связи по протоколу UDP.

Передача данных при связи по протоколу UDP [UDP_Send]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
UDP_Send	Передача данных при связи по протоколу UDP		UDP_Send_0(xExecute:= , hPeer:= , strIpAddrDst:= , uiPortDst:= , uiSize:= , pbydata:= , udiTimeOut:= , xDone=> , xBusy=> , xError=> , dwErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Действительный диапазон	Начальное значение	Описание
xExecute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
hPeer	Дескриптор связи	_XWORD	-	-	Дескриптор связи по протоколу UDP

strIpAddrDst	Заданный IP-адрес	STRING(80)	В качестве типа данных	-	Заданный IP-адрес для отправки данных
uiPortDst	Заданный порт	UINT	В качестве типа данных	-	Заданный порт для отправки данных
uiSize	Размер передаваемых данных	UINT	В качестве типа данных	-	Размер области переданных данных (BYTE)
pbyData	Указатель на область передачи	POINTER TO BYTE	-	-	Указатель на начальный байт области передачи
UditimeOut	Время таймаута	UDINT	В качестве типа данных	500000	Таймаут передачи данных, единица измерения - мкс, значение по умолчанию - 500 мс
VAR_OUTPUT	**Название**	**Тип данных**	**Действительный диапазон**	**Начальное значение**	**Описание**
xDone	Сигнал о завершении	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении передачи данных
xBusy	Признак занятости	BOOL	FALSE/TRUE	FALSE	Состояние занятости
xError	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Состояние ошибки
dwErrorID	Код ошибки	DWORD	В качестве типа данных	-	Код ошибки

2-8-2-4. Пример связи по протоколу UDP

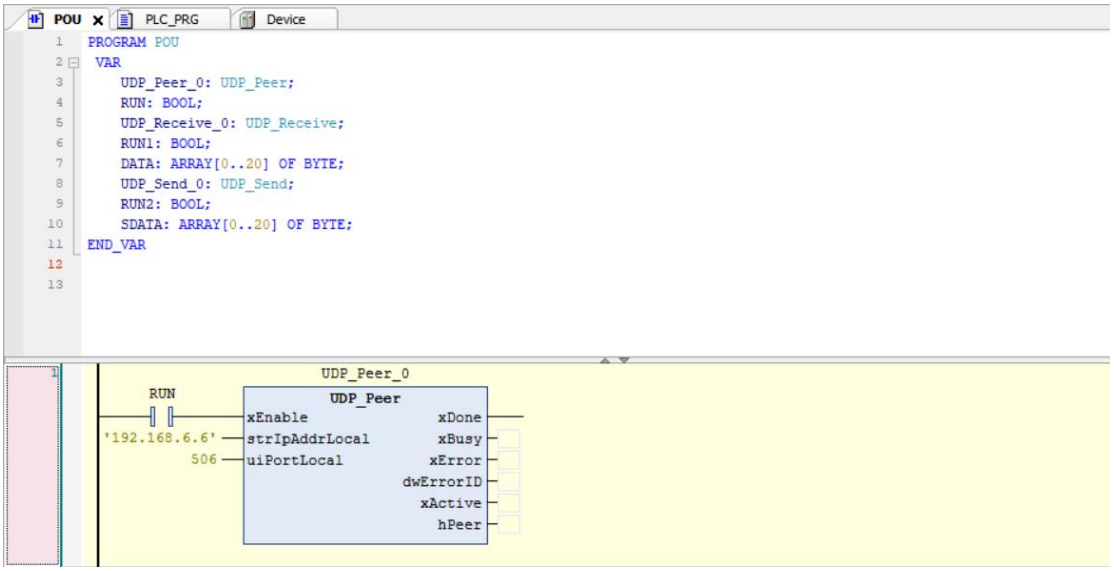
PMP30-60A32 устанавливает связь с помощником отладки компьютера по протоколу UDP

Помощник отладки (порт ПК) имеет IP-адрес 192.168.6.61, номер порта 506

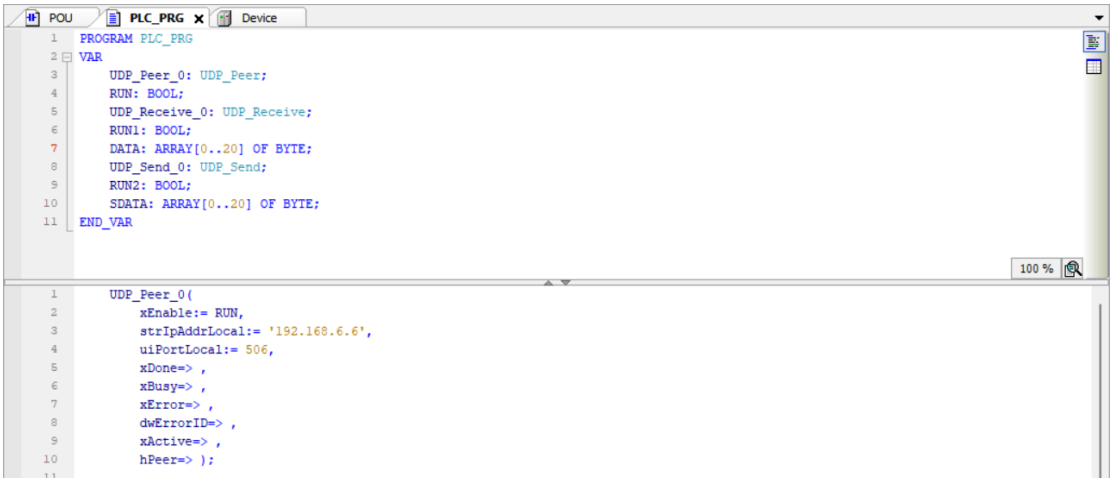
ПЛК имеет IP-адрес 192.168.6.6, номер порта 506

1. Установите соединение между ПЛК и помощником отладки по протоколу UDP. В команде создания соединения для связи по протоколу UDP (UDP_Peer) входной контакт strIpAddrLocal представляет собой адрес порта сети связи с локальным ПЛК, а входной контакт uiPortLocal представляет собой номер

порта локального ПЛК.

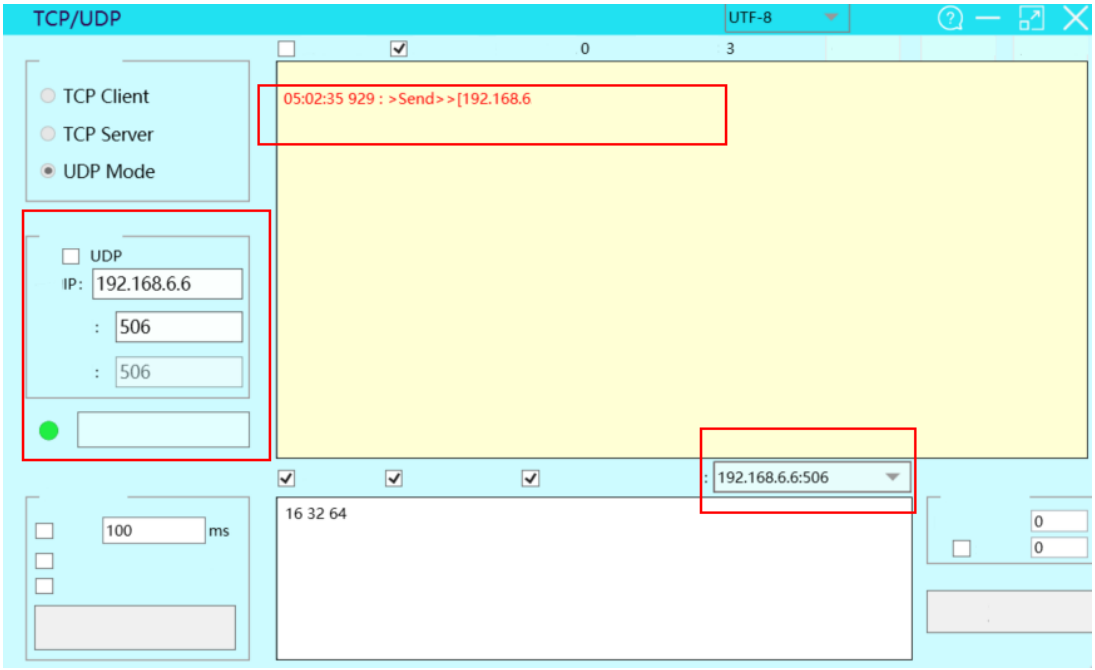


Форма LD

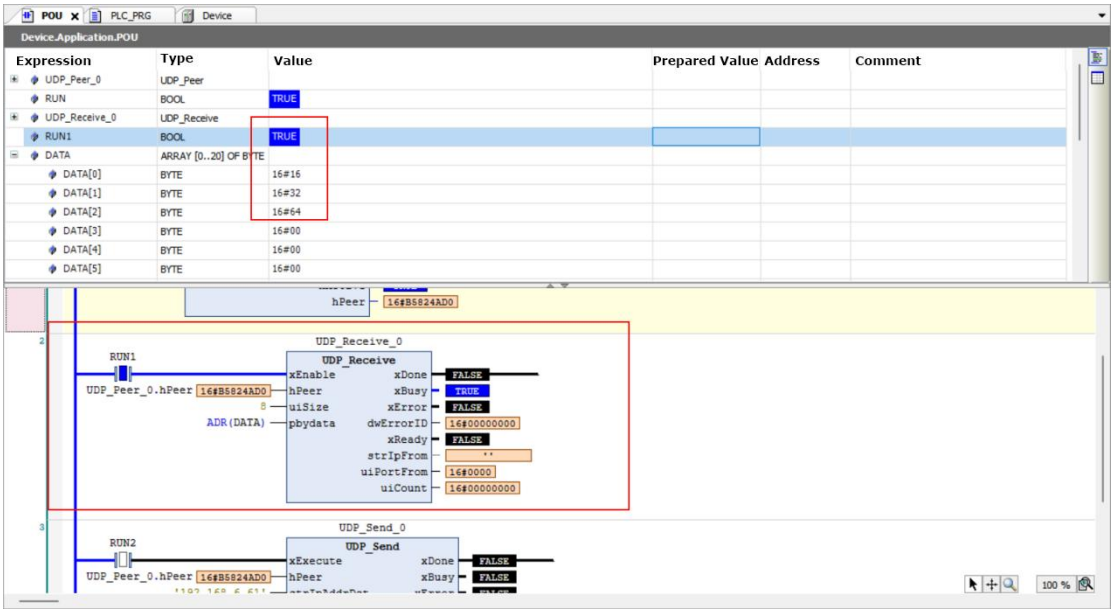


Форма ST

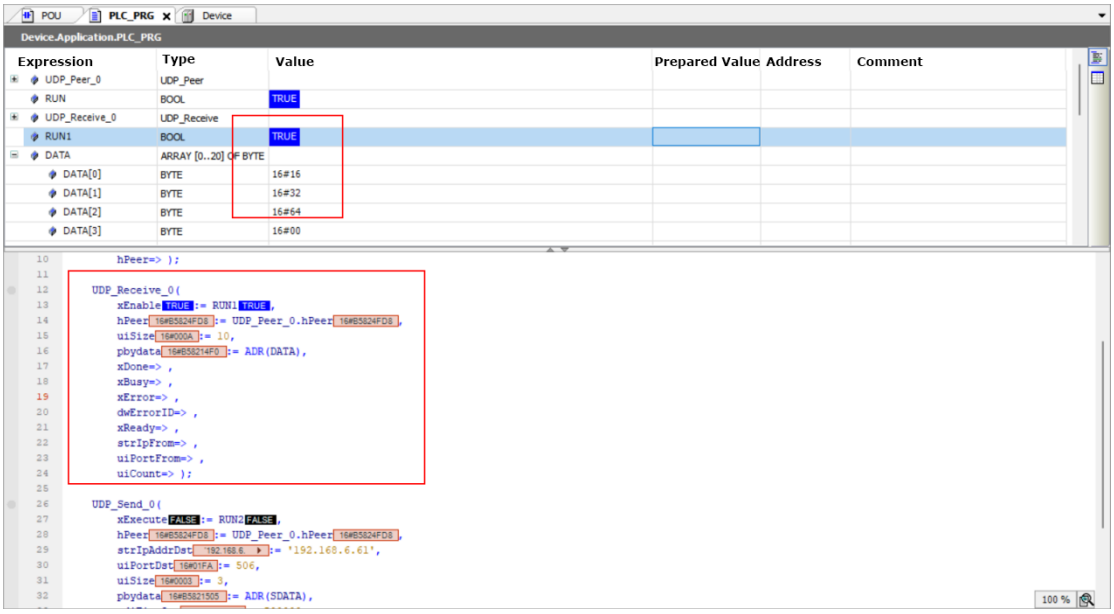
2. Прием данных. ПЛК принимает данные, отправленные помощником отладки ПК.



Запустите команду UDP_Receive, после чего принятые данные будут сохранены по адресу, на который указывает указатель pbyData, как показано на рисунке ниже. Успешно принятые данные: 16#16, 16#32, 16#64.



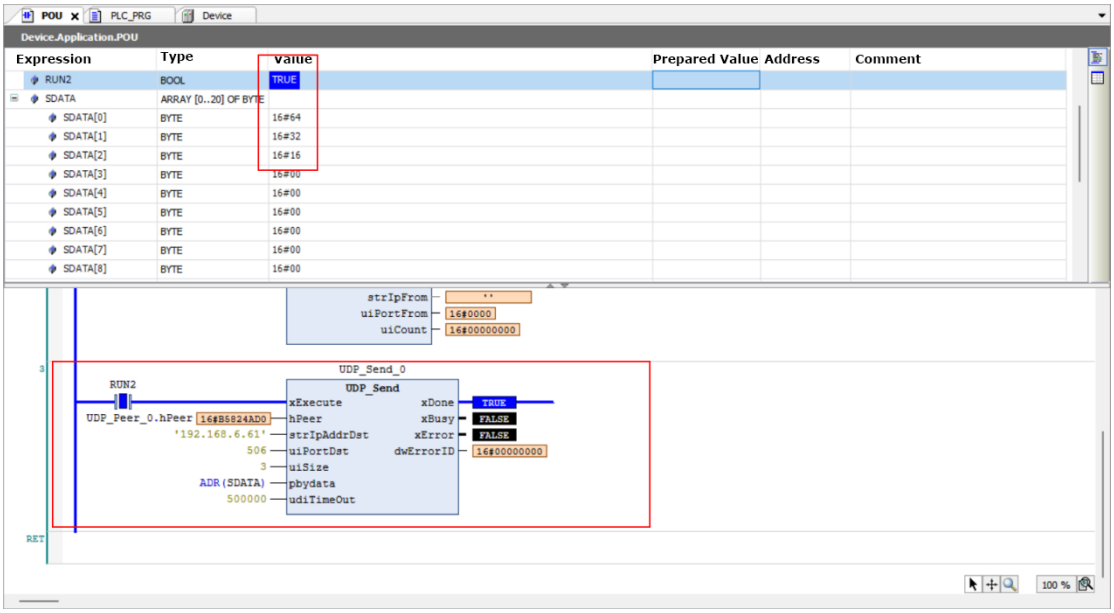
Форма LD



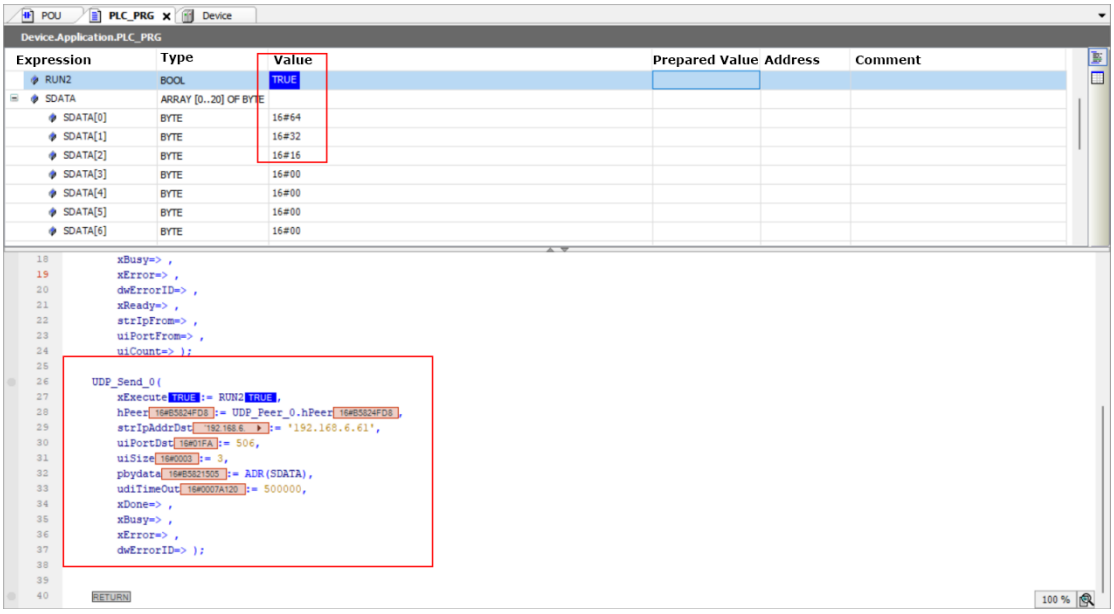
Форма ST

3. Передача данных. ПЛК отправляет данные в помощник отладки ПК.

Запустите команду UDP_Send для передачи данных, сохраненных по адресу, на который указывает указатель pbyData, как показано на рисунке ниже. Передача данных 16#64 16#32 16#16.

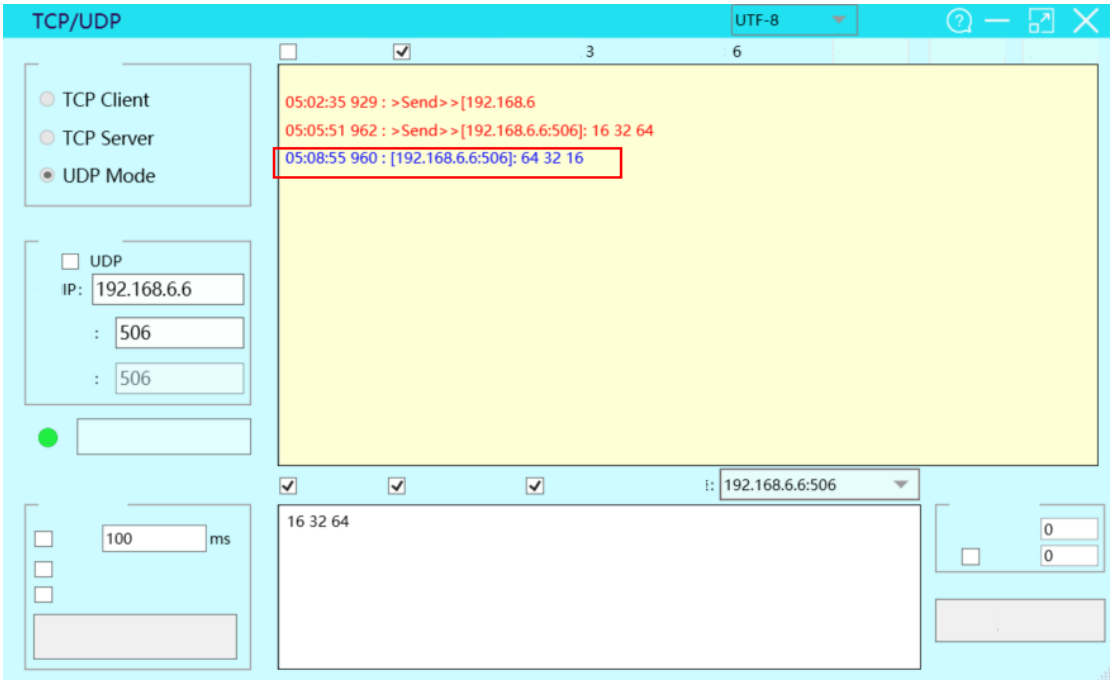


Форма LD



Форма ST

Как показано на рисунке ниже, помощник отладки ПК принял данные 16#64 16#32 16#16, и данные ПЛК были успешно отправлены.



2-8-3. Коды ошибок связи по протоколу TCP/UDP

Идентификатор ошибки	Код ошибки	Описание
NO_ERROR	0	Ошибки отсутствуют
FIRST_ERROR	6000	Зарезервировано
TIME_OUT	6001	Таймаут

INVALID_ADDR	6002	Неверный IP-адрес
INVALID_HANDLE	6003	Неверный дескриптор
INVALID_DATAPOINTER	6004	Неверный указатель данных
INVALID_DATASIZE	6005	Неверная длина данных
UDP_RECEIVE_ERROR	6006	Ошибка при приеме данных по протоколу UDP
UDP_SEND_ERROR	6007	Ошибка при передаче данных по протоколу UDP
UDP_SEND_NOT_COMPLETE	6008	Передача данных по протоколу UDP не завершена
UDP_OPEN_ERROR	6009	Ошибка при открытии протокола UDP
UDP_CLOSE_ERROR	6010	Ошибка при закрытии протокола UDP
TCP_SEND_ERROR	6011	Ошибка при передаче данных по протоколу TCP
TCP_RECEIVE_ERROR	6012	Ошибка при приеме данных по протоколу TCP
TCP_OPEN_ERROR	6013	Ошибка при открытии протокола TCP
TCP_CONNECT_ERROR	6014	Ошибка при установлении связи по протоколу TCP
TCP_CLOSE_ERROR	6015	Ошибка при закрытии протокола TCP
TCP_SERVER_ERROR	6016	Ошибка TCP-сервера
WRONG_PARAMETER	6017	Ошибка параметра
ERROR_UNKNOWN	6018	Неизвестная ошибка
TCP_NO_CONNECTION	6019	Связь по протоколу TCP отсутствует
IOCTL_ERROR	6020	Ошибка IOCTL
FIRST_MF	6050	Зарезервировано
LAST_ERROR	6099	Последняя ошибка отсчета времени

2-9. Обработка файлов

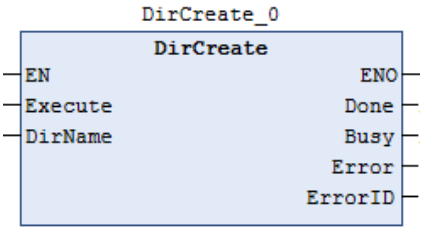
※	Из-за разных методов обработки файлов в операционных системах Windows и Linux одна и та же команда обработки файлов в разных операционных системах может использоваться по-разному.

2-9-1. Создание каталога DirCreate

(1) Обзор команды

Создание каталога с требуемым именем на SD-карте.

Создание каталога [DirCreate]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
DirCreate	Создание каталога		DirCreate_0(Execute:= , DirName:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
DirName	Задание имени каталога	STRING	Не более 1985 символов	-	Имя создаваемого каталога

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

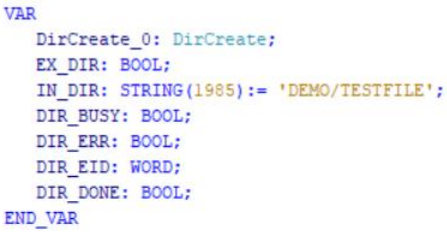
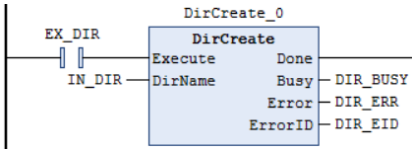
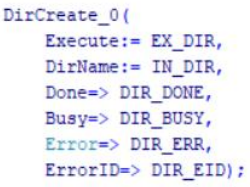
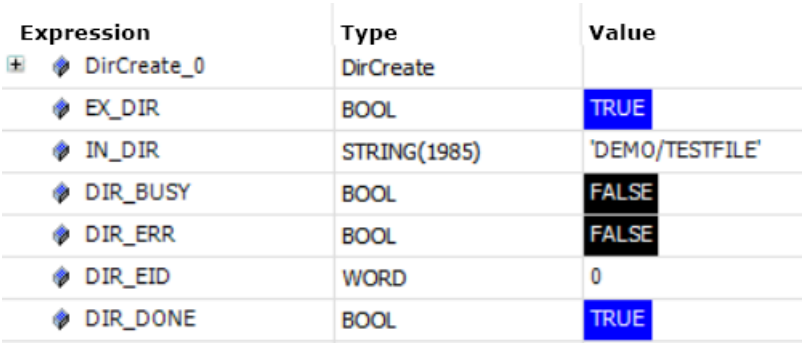
- Создание каталога с требуемым именем в "DirName" на SD-карте.

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Чтобы прочитать и записать файл, откройте его заново.
- Максимальный размер имени каталога вместе с префиксами, указанный в "DirName", составляет 255 символов для систем Linux. В случае превышения размер будет усечен. Системы Windows поддерживают максимум 260 символов вместе с префиксами в имени каталога, и при превышении этого предела будет выдано сообщение об ошибке.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда SD-карта не пригодна для использования.
- SD-карта защищена от записи.
- Если на SD-карте недостаточно свободного места.
- Когда количество создаваемых каталогов превысит допустимое.
- Если каталог, указанный в "DirName", уже существует.

- Если значение "DirName" является недопустимым именем каталога.
- При получении доступа к SD-карте и невозможности получения доступа из-за исключения.
- При создании каталога на SD-карте, путь к переменной IN_DIR должен быть следующим: ../$$USBDev$$/part0, например, IN_DIR: STRING(1985):= '../$$USBDev$$/part0/demo/testfile'.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    DirCreate_0: DirCreate;
    EX_DIR: BOOL;
    IN_DIR: STRING(1985):= 'DEMO/TESTFILE';
    DIR_BUSY: BOOL;
    DIR_ERR: BOOL;
    DIR_EID: WORD;
    DIR_DONE: BOOL;
END_VAR
```



```
DirCreate_0(
    Execute:= EX_DIR,
    DirName:= IN_DIR,
    Done=> DIR_DONE,
    Busy=> DIR_BUSY,
    Error=> DIR_ERR,
    ErrorID=> DIR_EID);
```

Expression	Type	Value
DirCreate_0	DirCreate	
EX_DIR	BOOL	TRUE
IN_DIR	STRING(1985)	'DEMO/TESTFILE'
DIR_BUSY	BOOL	FALSE
DIR_ERR	BOOL	FALSE
DIR_EID	WORD	0
DIR_DONE	BOOL	TRUE

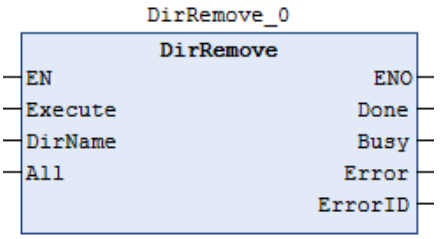
2-9-2. Удаление каталога DirRemove

(1) Обзор команды

Удаление каталога с SD-карты.

Удаление каталога [DirRemove]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

Библиотека	PP_FunctionUtils	

Команда	Название	Графическое представление	ST
DirRemove	Удаление каталога		DirRemove_0(Execute:= , DirName:= , All:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
DirName	Удаление имени каталога	STRING	Не более 1985 символов	-	Удаление имени каталога
All	Выбрать все	BOOL	FALSE/TRUE	FALSE	Операция выполняется, когда каталог содержит файлы/подкаталоги True: Удалить FALSE: Не удалять
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве	-	Идентификатор ошибки при

			типа данных		выполнении команды

(3) Функция

- Удаление каталога, указанного в имени каталога "DirName", с карты памяти SD.
- Если в каталоге, указанном в "DirName", содержатся файлы или подкаталоги, то в зависимости от выбранного значения "All" процесс будет происходить следующим образом:

Значение “All”	Операция
TRUE	Удаление указанного каталога, включая файлы и все подкаталоги
FALSE	Указанный каталог не удаляется, возникло исключение

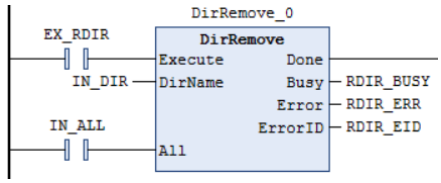
(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Чтобы прочитать и записать файл, откройте его заново.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Когда "All" принимает состояние true, и доступ к каталогу, указанному в "DirName", осуществляется через другие команды.
- Когда значение параметра "All" равно FALSE, и в каталоге, указанном в "DirName", есть файлы или подкаталоги.
- Если каталог, указанный в "DirName", доступен только для чтения.
- Если файл, указанный в "DirName", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   DirRemove_0: DirRemove;
   EX_RDIR: BOOL;
   IN_DIR: STRING(1985):= 'DEMO/TESTFILE';
   IN_ALL: BOOL;
   RDIR_BUSY: BOOL;
   RDIR_ERR: BOOL;
   RDIR_EID: WORD;
   RDIR_DONE: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
DirRemove_0(
    Execute:= EX_RDIR,
    DirName:= IN_DIR,
    All:= IN_ALL,
    Done=> RDIR_DONE,
    Busy=> RDIR_BUSY,
    Error=> RDIR_ERR,
    ErrorID=> RDIR_EID);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
DirRemove_0	DirRemove	
EX_RDIR	BOOL	TRUE
IN_DIR	STRING(1985)	'DEMO/TESTFILE'
IN_ALL	BOOL	TRUE
RDIR_BUSY	BOOL	FALSE
RDIR_ERR	BOOL	FALSE
RDIR_EID	WORD	0
RDIR_DONE	BOOL	TRUE

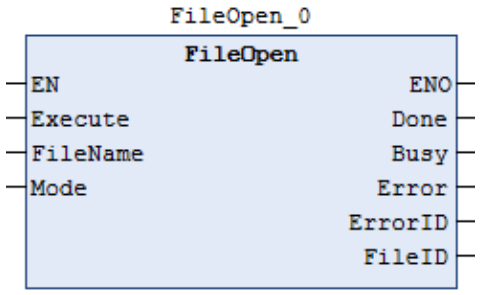
</td></tr>
</table>

2-9-3. Открытие файла FileOpen

(1) Обзор команды

Открытие указанного файла на карте памяти SD.

Открытие файла [FileOpen]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileOpen	Открытие файла		FileOpen_0(Execute:= , FileName:= , Mode:= , Done=> , Busy=> , Error=> , ErrorID=> , FileID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileName	Задание имени файла	STRING	Не более 1985 символов	-	Имя открываемого файла
Mode	Режим открытия	ACCESS_MODE	В качестве типа данных	-	Режим открытия файла
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды

ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды
FileID	Идентификатор файла	DWORD	В качестве типа данных	-	Идентификатор открытого файла

(3) Функция

- Открытие файла, указанного в "FileName", на карте памяти SD в режиме, указанном в "Mode".
- После открытия файла будет выведен идентификатор файла "FileID". "FileID" можно использоваться в командах FileRead и FileWrite для задания файла.
- Тип данных переменной "Mode" — это элемент перечисления ACCESS-MODE. Значения элементов перечисления:

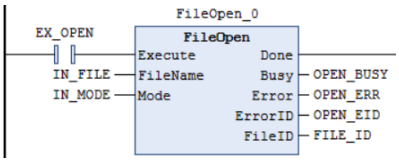
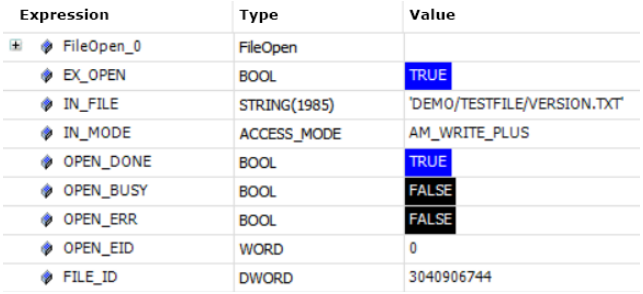
Элемент перечисления	Значение
AM_READ	Если путь к целевому файлу существует, откройте существующий файл только для чтения. Если путь не существует, выдается значение ошибки.
AM_WRITE	Если путь к целевому файлу существует, откройте файл, перезаписав его. Если путь не существует, будет автоматически создан новый файл с разрешением на запись.
AM_APPEND	Если путь к целевому файлу существует, откройте существующий файл только для записи и добавьте содержимое в конец файла. Если путь не существует, выдается значение ошибки.
AM_READ_PLUS	Если путь к целевому файлу существует, откройте существующий файл для чтения и записи. Если путь не существует, выдается значение ошибки.
AM_WRITE_PLUS	Если путь к целевому файлу существует, откройте файл для чтения и записи. Если путь не существует, будет автоматически создан новый файл с разрешением на чтение/запись.
AM_APPEND_PLUS	Если путь к целевому файлу существует, откройте существующий файл для чтения и записи и добавьте содержимое в конец файла. Если путь не существует, будет автоматически создан новый файл с разрешением на чтение/запись.

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Эта команда должна быть выполнена до команд FileSeek, FileWrite, FileRead, FilePuts и FileGets.
- Для закрытия файлов, открытых с помощью этой команды, выполните команду FileClose.
- После выполнения этой команды сохраните значение в "FileID". Когда значение переменной "Done" меняется с FALSE на True.
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Для чтения и записи файла
- При использовании SysFileRead в системе Linux для определения длины символов открытие файла в формате AM_WRITE выдает сообщение об ошибке с идентификатором ошибки 1. При использовании SysFileRead в системе Windows для определения длины символов открытие файла выдает сообщение об ошибке с идентификатором ошибки 2.
- При использовании SysFileRead в системе Linux для определения длины символов открытие файла в формате AM_APPEND выдает сообщение об ошибке с идентификатором ошибки 1. При использовании SysFileRead в системе Windows для определения длины символов ошибка может не появляться.
- Файлы, открытые через SysFileOpen, можно удалять по пути RUNTIME в системе Linux, но нельзя удалять по пути RUNTIME в системе Windows из-за проблем с правами.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Если значение параметра "Mode" равно AM_READ, AM_APPEND или AM_READ_PLUS, а файл, указанный в "FileName", не существует.
- Если значение "FileName" является недопустимым именем файла.
- Когда количество создаваемых файлов и каталогов превысит допустимое.

- При доступе к файлу, указанному в "FileName".
- Если файл, указанный в "FileName", доступен только для чтения.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

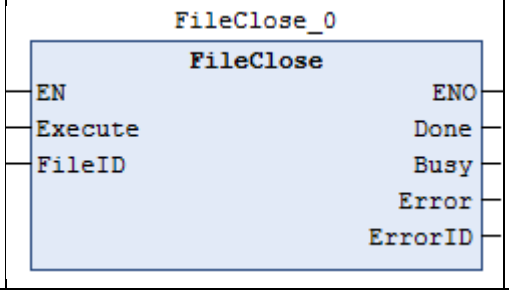
ДЕМО	LD	ST
Задание переменных	VAR FileOpen_0: FileOpen; EX_OPEN: BOOL; IN_FILE: STRING(1985):= 'DEMO/TESTFILE/VERSION.TXT'; IN_MODE: PP_FunctionUtils.ACCESS_MODE:= PP_FunctionUtils.ACCESS_MODE.AM_WRITE_PLUS; OPEN_DONE: BOOL; OPEN_BUSY: BOOL; OPEN_ERR: BOOL; OPEN_EID: WORD; END_VAR	
Пример программы		FileOpen_0(Execute:= EX_OPEN, FileName:= IN_DIR, Mode:= IN_MODE, Done=> OPEN_DONE, Busy=> OPEN_BUSY, Error=> OPEN_ERR, ErrorID=> OPEN_EID, FileID=> FILE_ID);
Результат операции		

2-9-4. Закрытие файла FileClose

(1) Обзор команды

Закрытие указанного файла на карте памяти SD.

Закрытие файла [FileClose]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileClose	Закрытие файла		FileClose_0(Execute:= , FileID:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileID	Идентификатор файла	_XWORD(DWORD)	В качестве типа данных	-	Идентификатор закрываемого файла
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды

ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

Закрытие файлов, указанных в "FileID", на карте памяти SD.

(4) Примечания

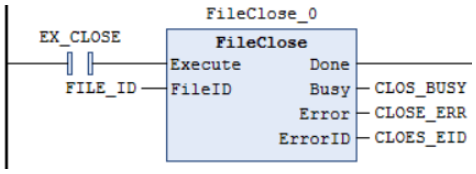
- Запрещается выполнять эту команду несколько раз для одного и того же файла, иначе это может привести к сбоям в работе ПЛК и даже к аварийному завершению работы.
- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.
- Для файлов, открытых с помощью команды FileOpen, обязательно выполните эту команду и закройте файл после использования, иначе могут возникнуть непредвиденные результаты.
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Чтобы прочитать и записать файл, откройте его заново.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- При доступе к файлу, указанному в "FileID".
- Если файл, указанный в "FileID", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    FileClose_0: FileClose;
    EX_CLOSE: BOOL;
    FILE_ID: __XWORD:= 3040906744;
    CLOSE_DONE: BOOL;
    CLOS_BUSY: BOOL;
    CLOSE_ERR: BOOL;
    CLOES_EID: WORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FileClose_0(
    Execute:= EX_CLOSE,
    FileID:= FILE_ID,
    Done=> CLOSE_DONE,
    Busy=> CLOS_BUSY,
    Error=> CLOSE_ERR,
    ErrorID=> CLOES_EID);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
FileClose_0	FileClose	
EX_CLOSE	BOOL	TRUE
FILE_ID	DWORD	3040906744
CLOSE_DONE	BOOL	TRUE
CLOS_BUSY	BOOL	FALSE
CLOSE_ERR	BOOL	FALSE
CLOES_EID	WORD	0

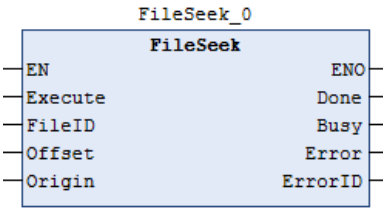
</td></tr>
</table>

2-9-5. Поиск файла FileSeek

(1) Обзор команды

Установка индикатора расположения файла в указанном файле на карте памяти SD.

Поиск файла [FileSeek]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileSeek	Поиск файла		FileSeek_0(Execute:= , FileID:= , Offset:= , Origin:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileID	Идентификатор файла	_XWORD(DWORD)	В качестве типа данных	-	Идентификатор файла, где должен быть задан индикатор расположения файла
Offset	Смещение	DINT	В качестве типа данных	-	Смещение индикатора расположения файла относительно "Origin"
Origin	Исходное положение	_eFSEEK_ORIGIN	В качестве типа данных	_eFSEEK_ORIGIN._SEEK_SET	Исходное положение индикатора расположения файла

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

- Установка индикатора расположения файла в файле на карте памяти SD, указанном в идентификаторе файла "FileID". Индикатор расположения файла указывает на место в файле, где начинается чтение или запись при выполнении команд чтения и записи данных файла. Например, при считывании данных с начала файла следует использовать команду FileSeek, чтобы установить индикатор расположения файла в начале файла, а затем выполнить команду FileRead.
- Установите положение индикатора расположения файла как исходное положение "Origin" плюс смещение "Offset".
- Тип данных "Origin" — это элемент перечисления _eFSEEK_ORIGIN. Значения элементов перечисления:

Элемент перечисления	Значение
_SEEK_SET	В начале файла
_SEEK_CUR	Местоположение текущего индикатора расположения файла
_SEEK_END	В конце файла

(4) Примечания

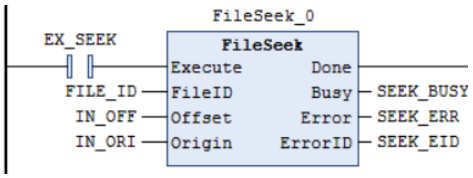
- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- При доступе к файлу, указанному в "FileID".
- Если файл, указанный в "FileID", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    FileSeek_0: FileSeek;
    EX_SEEK: BOOL;
    FILE_ID: __XWORD:= 3040906744;
    IN_OFF: DINT:= 6;
    IN_ORI: _eFSEEK_ORIGIN:= _eFSEEK_ORIGIN._SEEK_SET;
    SEEK_DONE: BOOL;
    SEEK_BUSY: BOOL;
    SEEK_ERR: BOOL;
    SEEK_EID: WORD;
END_VAR
```

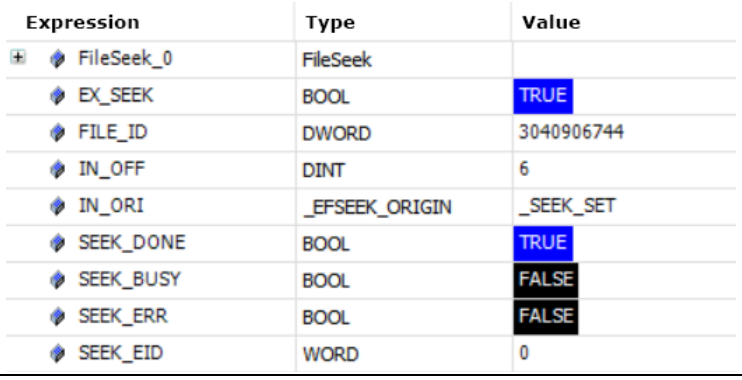
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FileSeek_0(
    Execute:= EX_SEEK,
    FileID:= FILE_ID,
    Offset:= IN_OFF,
    Origin:= IN_ORI,
    Done=> SEEK_DONE,
    Busy=> SEEK_BUSY,
    Error=> SEEK_ERR,
    ErrorID=> SEEK_EID);
```

</td></tr>
</table>

Результат операции	

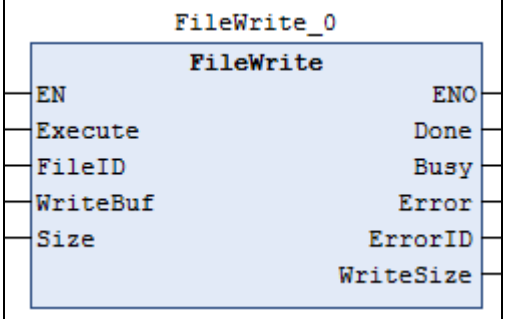
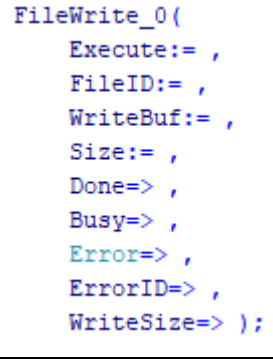
Expression	Type	Value
FileSeek_0	FileSeek	
EX_SEEK	BOOL	TRUE
FILE_ID	DWORD	3040906744
IN_OFF	DINT	6
IN_ORI	_EFSEEK_ORIGIN	_SEEK_SET
SEEK_DONE	BOOL	TRUE
SEEK_BUSY	BOOL	FALSE
SEEK_ERR	BOOL	FALSE
SEEK_EID	WORD	0

2-9-6. Запись в файл FileWrite

(1) Обзор команды

Запись данных в указанный файл на карте памяти SD.

Запись в файл [FileWrite]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileWrite	Запись в файл		


```
FileWrite_0(
    Execute:= ,
    FileID:= ,
    WriteBuf:= ,
    Size:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> ,
    WriteSize=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileID	Идентификатор файла	_XWORD(DWORD)	В качестве типа данных	-	Идентификатор файла, в который записываются данные
Size	Запись количества элементов	UINT	В качестве типа данных	-	Количество записываемых элементов
WriteBuf	Запись массива в кэш	BOOL/ BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/ REAL/LREAL/ TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Объект записи в записанных данных
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды

Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды
WriteSize	Фактическое количество записанных элементов	UINT	В качестве типа данных	-	Фактическое количество записанных элементов

(3) Функция

- Для файла на карте памяти SD, указанного в идентификаторе файла "FileID", запишите данные в кэше WriteBuf[ ], в место, на которое указывает индикатор расположения файла.
- Индикатор расположения файла должен быть заранее установлен в любое место с помощью команды FileSeek.
- Объем записанных данных — (WriteBuf [ ] размер типа данных) × "Size".
- Фактическое количество записанных элементов сохраняется в переменной "WriteSize". Как правило, значение переменной "Size" соответствует значению переменной "ReadSize".

(4) Примечание

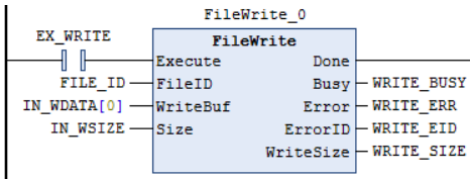
- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если значение "Size" превышает область массива в WriteBuf[], нужно избежать исключений ПЛК.
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.

- При доступе к файлу, указанному в "FileID".
- Если файл, указанный в "FileID", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.
- Когда файл, указанный в "FileID", не открыт в режиме записи.
- ReadBuf[] передает весь массив, а ErrorID отображает код ошибки "16#20".

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   FileWrite_0: FileWrite;
   EX_WRITE: BOOL;
   FILE_ID: __XWORD:= 3040906744;
   IN_WDATA: ARRAY[0..2] OF BYTE:= [16,32,64];
   IN_WSIZE: UINT:= 3;
   WRITE_DONE: BOOL;
   WRITE_BUSY: BOOL;
   WRITE_ERR: BOOL;
   WRITE_EID: WORD;
   WRITE_SIZE: UINT;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
FileWrite_0(
    Execute:= EX_WRITE,
    FileID:= FILE_ID,
    WriteBuf:= IN_WDATA[0],
    Size:= IN_WSIZE,
    Done=> WRITE_DONE,
    Busy=> WRITE_BUSY,
    Error=> WRITE_ERR,
    ErrorID=> WRITE_EID,
    WriteSize=> WRITE_SIZE);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
FileWrite_0	FileWrite	
EX_WRITE	BOOL	TRUE
FILE_ID	DWORD	3040906744
IN_WDATA	ARRAY [0..2] OF BYTE	
IN_WDATA[0]	BYTE	16
IN_WDATA[1]	BYTE	32
IN_WDATA[2]	BYTE	64
IN_WSIZE	UINT	3
WRITE_DONE	BOOL	TRUE
WRITE_BUSY	BOOL	FALSE
WRITE_ERR	BOOL	FALSE
WRITE_EID	WORD	0
WRITE_SIZE	UINT	3

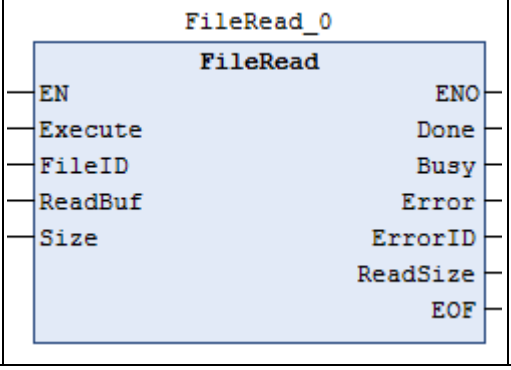
</td></tr>
</table>

2-9-7. Считывание данных из файла FileRead

(1) Обзор команды

Считывание данных из указанного файла на карте памяти SD.

Считывание данных из файла [FileRead]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileRead	Считывание данных из файла		FileRead_0(Execute:= , FileID:= , ReadBuf:= , Size:= , Done=> , Busy=> , Error=> , ErrorID=> , ReadSize=> , EOF=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileID	Идентификатор файла	_XWORD(DWORD)	В качестве типа данных	-	Идентификатор файла, из которого считываются данные

Size	Считываемое количество элементов	UINT	В качестве типа данных	-	Количество считываемых элементов
ReadBuf	Считываемый массива в кэше	BOOL/ BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ ULINT/ SINT/INT/DINT/LINT/ REAL/LREAL/ TIME/DATE/TOD/DT/ STRING	В качестве типа данных	-	Объект записи в считанных данных

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды
ReadSize	Фактическое количество считанных элементов	UINT	В качестве типа данных	-	Фактическое количество считанных элементов

EOF	Признак конца файла	BOOL	FALSE/TRUE	FALSE	Достигнут ли конец файла True: Достигнут FALE: Не достигнут

(3) Функция

- Для файла на карте памяти SD, указанного в идентификаторе файла "FileID", начните считывать данные из места, обозначенного индикатором расположения файла, и сохраните данные в кэш чтения ReadBuf[].
- Индикатор расположения файла должен быть заранее установлен в любое место с помощью команды FileSeek.
- Объем считанных данных — (ReadBuf[ ] размер типа данных) × "Size".
- Фактическое количество считанных элементов сохраняется в переменной "ReadSize". Как правило, значение переменной "Size" соответствует значению переменной "ReadSize". Если объем данных от индикатора расположения файла до конца файла меньше "Size", исключения не произойдет, и данные до конца файла будут сохранены в ReadBuf[]. В этот момент значение переменной "ReadSize" становится меньше значения "Size".
- Кроме того, при чтении до конца файла значение "EOF" в конце файла становится истинным (true). В противном случае, значение "EOF" будет FALSE.

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если при считывании данных до конца файла размер данных нельзя полностью разделить на размер типа данных ReadBuf[], данные, размер которых меньше размера ReadBuf[], также будут обработаны. Индикатор расположения файла дойдет до конца файла, и значение "EOF" станет истинным.
- Если значение "Size" превышает область массива в ReadBuf[], нужно избежать исключений ПЛК. Элемент после "Size" в ReadBuf[] (который не будет перезаписан при считывании данных) сохраняет значение, которое было до выполнения этой команды.

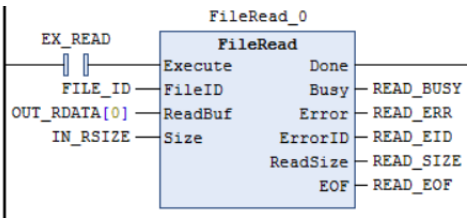
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.
- После выполнения этой команды сохраните значение в "EOF". Когда значение переменной "Done" меняется с FALSE на True.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- При доступе к файлу, указанному в "FileID".
- Если файл, указанный в "FileID", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.
- Когда файл, указанный в "FileID", не открыт в режиме чтения.
- ReadBuf[] передает весь массив, а ErrorID отображает код ошибки "16#20".

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    FileRead_0: FileRead;
    EX_READ: BOOL;
    FILE_ID: __XWORD:= 3040906744;
    OUT_RDATA: ARRAY[0..2] OF BYTE;
    IN_RSIZE: UINT:= 3;
    READ_DONE: BOOL;
    READ_BUSY: BOOL;
    READ_ERR: BOOL;
    READ_EID: WORD;
    READ_SIZE: UINT;
    READ_EOF: BOOL;
END_VAR
```

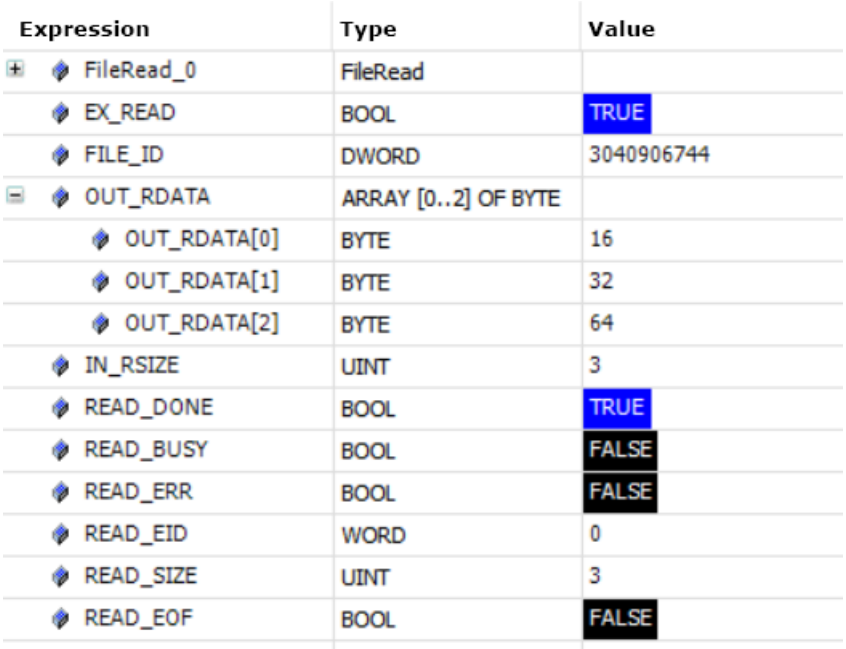
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FileRead_0(
    Execute:= EX_READ,
    FileID:= FILE_ID,
    ReadBuf:= OUT_RDATA,
    Size:= IN_RSIZE,
    Done=> READ_DONE,
    Busy=> READ_BUSY,
    Error=> READ_ERR,
    ErrorID=> READ_EID,
    ReadSize=> READ_SIZE,
    EOF=> READ_EOF);
```

</td></tr>
</table>

Результат операции	

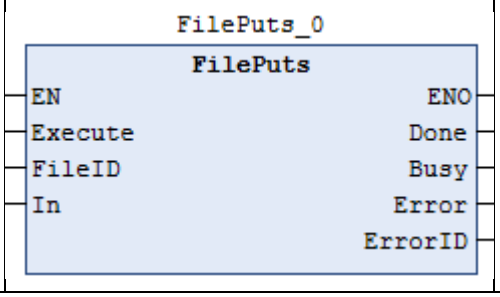
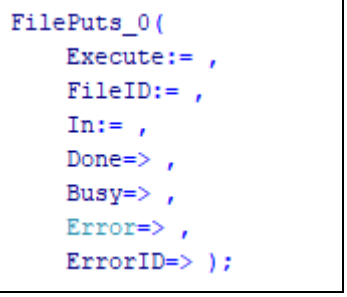
Expression	Type	Value
FileRead_0	FileRead	
EX_READ	BOOL	TRUE
FILE_ID	DWORD	3040906744
OUT_RDATA	ARRAY [0..2] OF BYTE	
OUT_RDATA[0]	BYTE	16
OUT_RDATA[1]	BYTE	32
OUT_RDATA[2]	BYTE	64
IN_RSIZE	UINT	3
READ_DONE	BOOL	TRUE
READ_BUSY	BOOL	FALSE
READ_ERR	BOOL	FALSE
READ_EID	WORD	0
READ_SIZE	UINT	3
READ_EOF	BOOL	FALSE

2-9-8. Запись строки в файл FilePuts

(1) Обзор команды

Запись строки в указанный файл на карте памяти SD.

Запись строки в файл [FilePuts]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FilePuts	Запись строки в файл		


```
FilePuts_0(
    Execute:= ,
    FileID:= ,
    In:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileID	Идентификатор файла	_XWORD(DWORD)	В качестве типа данных	-	Идентификатор файла, в который записываются данные
In	Записываемая строка	STRING	В качестве типа данных	-	Записываемая строка
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

- Запись строки "In" в место, обозначенное индикатором расположения файла на карте памяти SD, указанного в идентификаторе файла "FileID".

- Индикатор расположения файла должен быть заранее установлен в любое место с помощью команды FileSeek.

(4) Примечание

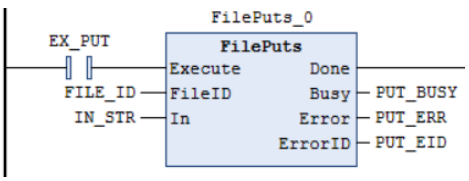
- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Если на карте памяти SD недостаточно свободного места.
- При доступе к файлу, указанному в "FileID".
- Если файл, указанный в "FileID", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.
- Когда файл, указанный в "FileID", не открыт в режиме записи.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    FilePuts_0: FilePuts;
    EX_PUT: BOOL;
    FILE_ID: __XWORD:= 3040906744;
    IN_STR: STRING(1985):= '603416';
    PUT_DONE: BOOL;
    PUT_BUSY: BOOL;
    PUT_ERR: BOOL;
    PUT_EID: WORD;
END_VAR
```

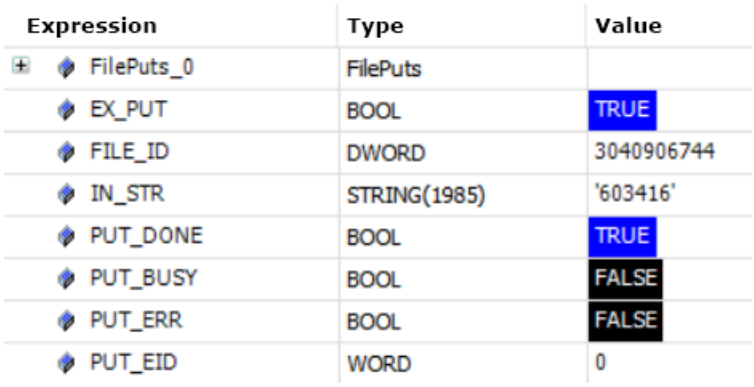
</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FilePuts_0(
    Execute:= EX_PUT,
    FileID:= FILE_ID,
    In:= IN_STR,
    Done=> PUT_DONE,
    Busy=> PUT_BUSY,
    Error=> PUT_ERR,
    ErrorID=> PUT_EID);
```

</td></tr>
</table>

Результат операции	

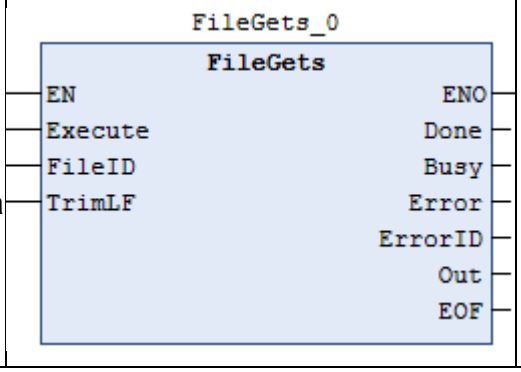
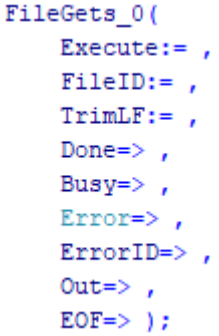
Expression	Type	Value
FilePuts_0	FilePuts	
EX_PUT	BOOL	TRUE
FILE_ID	DWORD	3040906744
IN_STR	STRING(1985)	'603416'
PUT_DONE	BOOL	TRUE
PUT_BUSY	BOOL	FALSE
PUT_ERR	BOOL	FALSE
PUT_EID	WORD	0

2-9-9. Считывание строки из файла FileGets

(1) Обзор команды

Считывание 1 строки из указанного файла на карте памяти SD.

Считывание строки из файла [FileGets]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileGets	Считывание строки из файла		


```
FileGets_0(
    Execute:= ,
    FileID:= ,
    TrimLF:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> ,
    Out=> ,
    EOF=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileID	Идентификатор файла	_XWORD(DWORD)	В качестве типа данных	-	Идентификатор файла, из которого считываются данные
TrimLF	Удаление кода разрыва строки	BOOL	FALSE/TRUE	FALSE	Удаление кода разрыва строки TRUE: удалить FALSE: не удалять
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды

ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды
Out	Считывание строки из файла	STRING	В качестве типа данных	-	Считывание строки из файла
EOF	Признак конца файла	BOOL	FALSE/TRUE	FALSE	Достигнут ли конец файла True: Достигнут FALE: Не достигнут

(3) Функция

- Считывание одной строки из места, обозначенного индикатором расположения файла на карте памяти SD, указанного в идентификаторе файла "FileID".
- Индикатор расположения файла должен быть заранее установлен в любое место с помощью команды FileSeek.
- Интервал между строками определяют по коду новой строки. Вывод считываемой строки в переменную считывания строки из файла "Out".
- Разрывы строк автоматически распознаются из трех типов CR, LF и CR+LF. Если параметр TrimLF, задающий удаление кода разрыва строки, принимает значение true, то после удаления кода разрыва строки в строке значение будет записано в "Out".
- При чтении до конца файла значение "EOF" в конце флага файла становится истинным (true).

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если длина строки превышает 1985 байт, то 1985 байт и предыдущая строка, а также последний символ NULL будут сохранены в "Out".
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.

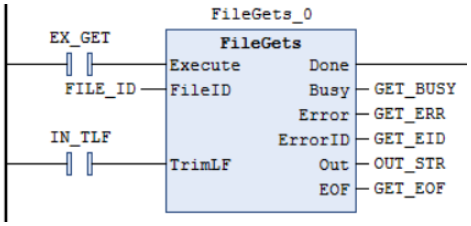
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- При доступе к файлу, указанному в "FileID".
- Если файл, указанный в "FileID", не существует.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.
- Когда файл, указанный в "FileID", не открыт в режиме чтения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   FileGets_0: FileGets;
   EX_GET: BOOL;
   FILE_ID: __XWORD:= 3040906744;
   IN_TLF: BOOL;
   GET_DONE: BOOL;
   GET_BUSY: BOOL;
   GET_ERR: BOOL;
   GET_EID: WORD;
   OUT_STR: STRING(1985);
   GET_EOF: BOOL;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FileGets_0(
    Execute:= EX_GET,
    FileID:= FILE_ID,
    TrimLF:= IN_TLF,
    Done=> GET_DONE,
    Busy=> GET_BUSY,
    Error=> GET_ERR,
    ErrorID=> GET_EID,
    Out=> OUT_STR,
    EOF=> GET_EOF);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
FileGets_0	FileGets	
EX_GET	BOOL	TRUE
FILE_ID	DWORD	3040906744
IN_TLF	BOOL	FALSE
GET_DONE	BOOL	TRUE
GET_BUSY	BOOL	FALSE
GET_ERR	BOOL	FALSE
GET_EID	WORD	0
OUT_STR	STRING(1985)	'416603416603416'
GET_EOF	BOOL	TRUE

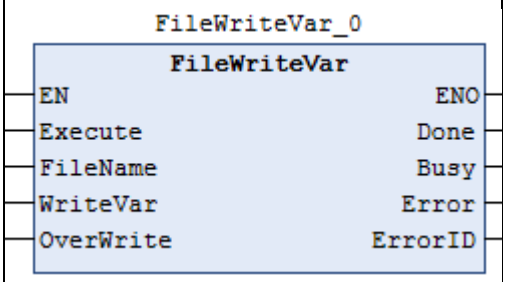
</td></tr>
</table>

2-9-10. Запись переменной в файл FileWriteVar

(1) Обзор команды

Запись значения одной переменной в двоичном виде в указанный файл на карте памяти SD.

Запись переменной в файл [FileWriteVar]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileWriteVar	Запись переменной в файл		FileWriteVar_0(Execute:= , FileName:= , WriteVar:= , OverWrite:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileName	Указанное имя файла	STRING	Не более 1985 символов	-	Имя файла, в который записывают данные переменной

WriteVar	Ввод записываемых переменных	BOOL/ BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/ REAL/LREAL/ TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Записываемая переменная
OverWrite	Выбор функции перезаписи	BOOL	FALSE/TRUE	FALSE	True: Перезапись разрешена FALSE: Перезапись запрещена

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

- Запись переменной "WriteVar" в двоичном виде в файл на карте памяти SD, указанный в имени файла "FileName".

- Если на карте памяти SD нет файла с таким же именем, как в "FileName", будет создан новый файл.
- Если файл с таким же именем, как в "FileName", уже существует на карте памяти SD, то в зависимости от значения, разрешенного для перезаписи "OverWrite", будут выполнены следующие операции:

Значение “OverWrite”	Операция
TRUE	Перезапись указанного файла
FALSE	Файл не перезаписывается, возникло исключение

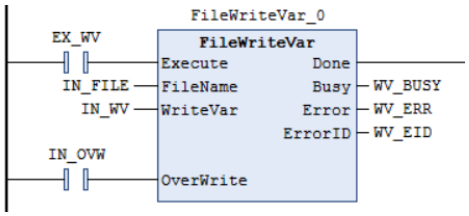
(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- "FileID" необходимо получить, предварительно выполнив команду FileOpen.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Если на карте памяти SD недостаточно свободного места.
- При доступе к файлу, указанному в "FileID".
- Если значение "FileName" является недопустимым именем файла.
- Когда количество создаваемых файлов и каталогов превысит допустимое.
- Если файл с таким же именем, как в "FileName", уже существует, и к нему осуществляется доступ.
- Если файл с таким же именем, как в "FileName", уже существует, а переменная "OverWrite" принимает значение FALSE.
- Если файл с таким же именем, как в "FileName", уже существует и доступен только для чтения.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    FileWriteVar_0: FileWriteVar;
    EX_WV: BOOL;
    IN_FILE: STRING(1985):= 'DEMO/TESTFILE/VERSION.TXT';
    IN_WV: ARRAY[0..2] OF BYTE:= [64,32,16];
    IN_OVW: BOOL;
    WV_DONE: BOOL;
    WV_BUSY: BOOL;
    WV_ERR: BOOL;
    WV_EID: WORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>


</td><td>

```
FileWriteVar_0(
    Execute:= EX_WV,
    FileName:= IN_FILE,
    WriteVar:= IN_WV,
    OverWrite:= IN_OVW,
    Done=> WV_DONE,
    Busy=> WV_BUSY,
    Error=> WV_ERR,
    ErrorID=> WV_EID);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
FileWriteVar_0	FileWriteVar	
EX_WV	BOOL	TRUE
IN_FILE	STRING(1985)	'DEMO/TESTFILE/VERSION.TXT'
IN_WV	ARRAY [0..2] OF BYTE	
IN_WV[0]	BYTE	64
IN_WV[1]	BYTE	32
IN_WV[2]	BYTE	16
IN_OVW	BOOL	TRUE
WV_DONE	BOOL	TRUE
WV_BUSY	BOOL	FALSE
WV_ERR	BOOL	FALSE
WV_EID	WORD	0

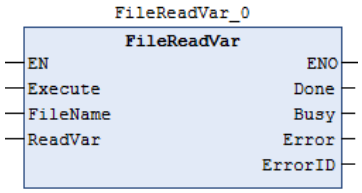
</td></tr>
</table>

2-9-11. Считывание переменной из файла FileReadVar

(1) Обзор команды

Считывание значения из указанного файла на карте памяти SD в двоичном виде и его запись в переменную.

Считывание переменной из файла [FileReadVar]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileReadVar	Считывание переменной из файла		FileReadVar_0(Execute:= , FileName:= , ReadVar:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileName	Указанное имя файла	STRING	Не более 1985 символов	-	Имя файла, из которого считывают данные переменной
ReadVar	Считываемые переменные	BOOL/ BYTE/WORD/DWORD/LWORD/ USINT/UINT/UDINT/ULINT/ SINT/INT/DINT/LINT/ REAL/LREAL/ TIME/DATE/TOD/DT/STRING	В качестве типа данных	-	Сохранение считанных данных переменных
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды

Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

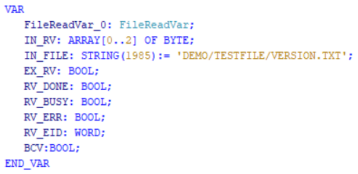
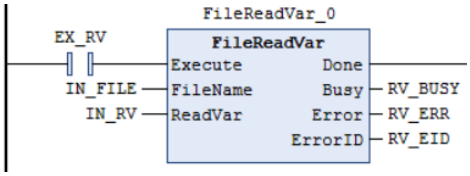
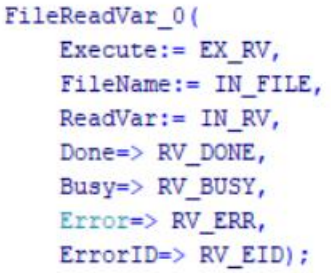
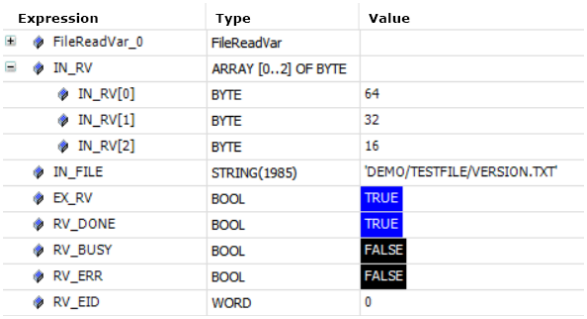
- Считывание данных переменной в двоичном виде из файла на карте памяти SD, указанного в имени файла "FileName", и подстановка считанного значения в считываемую переменную "ReadVar".

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если размер указанного файла больше размера "ReadVar", исключения не произойдет, и будут считаны только данные, соответствующие размеру "ReadVar".
- Если размер указанного файла меньше размера "ReadVar", исключения не произойдет, и будут считаны только данные, соответствующие размеру указанного файла. Оставшаяся область "ReadVar" сохранит свое значение до выполнения этой команды.
- Если "ReadVar" представляет собой всю структуру, области корректировки можно вставлять между элементами с учетом особенностей конкретной структуры.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Если значение "FileName" является недопустимым именем файла.

- Если файл, указанный в "FileName", не существует.
- При доступе к файлу, указанному в "FileName".
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    FileReadVar_0: FileReadVar;
    IN_RV: ARRAY[0..2] OF BYTE;
    IN_FILE: STRING(1985):= 'DEMO/TESTFILE/VERSION.TXT';
    EX_RV: BOOL;
    RV_DONE: BOOL;
    RV_BUSY: BOOL;
    RV_ERR: BOOL;
    RV_EID: WORD;
    BCV:BOOL;
END_VAR
```



```
FileReadVar_0(
    Execute:= EX_RV,
    FileName:= IN_FILE,
    ReadVar:= IN_RV,
    Done=> RV_DONE,
    Busy=> RV_BUSY,
    Error=> RV_ERR,
    ErrorID=> RV_EID);
```

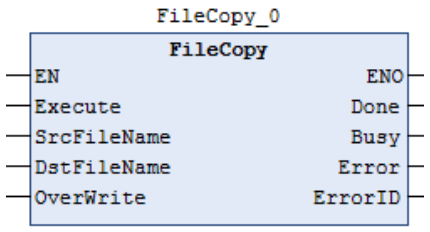
Expression	Type	Value
FileReadVar_0	FileReadVar	
IN_RV	ARRAY [0..2] OF BYTE	
IN_RV[0]	BYTE	64
IN_RV[1]	BYTE	32
IN_RV[2]	BYTE	16
IN_FILE	STRING(1985)	'DEMO/TESTFILE/VERSION.TXT'
EX_RV	BOOL	TRUE
RV_DONE	BOOL	TRUE
RV_BUSY	BOOL	FALSE
RV_ERR	BOOL	FALSE
RV_EID	WORD	0

2-9-12. Копирование файла FileCopy

(1) Обзор команды

Копирование указанного файла с карты памяти SD.

Копирование файла [FileCopy]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileCopy	Копирование файлов		FileCopy_0(Execute:= , SrcFileName:= , DstFileName:= , OverWrite:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
StcFileName	Копирование имени исходного файла	STRING	Не более 1985 символов	-	Копирование имени исходного файла
DstFileName	Имя целевого файла для вставки	STRING	Не более 1985 символов	-	Имя целевого файла для вставки
OverWrite	Выбор функции перезаписи	BOOL	FALSE/TRUE	FALSE	True: Перезапись разрешена FALSE: Перезапись запрещена
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды

ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

- Копирования файла, указанного в имени исходного файла "SrcFileName", с карты памяти SD и его вставка в имя целевого файла "DstFileName".
- Если на карте памяти SD нет файла с таким же именем, как в "DstFileName", будет создан новый файл.
- Если на карте памяти SD уже существует файл с таким же именем, как в "DstFileName", то в зависимости от допустимого значения параметра "OverWrite" будут выполнены следующие операции перезаписи:

Значение “OverWrite”	Операция
TRUE	Перезапись указанного файла
FALSE	Файл не перезаписывается, возникло исключение

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Чтобы прочитать и записать файл, откройте его заново.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Если на карте памяти SD недостаточно свободного места.
- Если файл, указанный в "SrcFileName", не существует.
- Если значение "SrcFileName" является недопустимым именем файла.
- Если значение "DstFileName" является недопустимым именем файла.

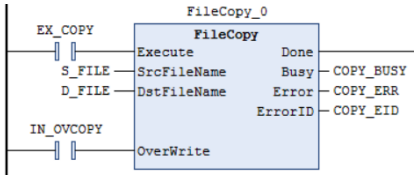
- Когда количество создаваемых файлов и каталогов превысит допустимое.
- Если файл с таким же именем, как в "DstFileName", уже существует, и к нему осуществляется доступ.
- Если файл с таким же именем, как в "DstFileName", уже существует, а переменная "OverWrite" принимает значение FALSE.
- Если файл с таким же именем, как в "DstFileName", уже существует и доступен только для чтения.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
    FileCopy_0: FileCopy;
    EX_COPY: BOOL;
    S_FILE: STRING(1985):= 'DEMO/TESTFILE/VERSION.TXT';
    D_FILE: STRING(1985):= 'DEMO/TESTFILE/VERSION_COPY.TXT';
    IN_OVCOPY: BOOL;
    COPY_DONE: BOOL;
    COPY_BUSY: BOOL;
    COPY_ERR: BOOL;
    COPY_EID: WORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FileCopy_0(
    Execute:= EX_COPY,
    SrcFileName:= S_FILE,
    DstFileName:= D_FILE,
    OverWrite:= IN_OVCOPY,
    Done=> COPY_DONE,
    Busy=> COPY_BUSY,
    Error=> COPY_ERR,
    ErrorID=> COPY_EID);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
EX_COPY	BOOL	TRUE
S_FILE	STRING(1985)	'DEMO/TESTFILE/VERSION.TXT'
D_FILE	STRING(1985)	'DEMO/TESTFILE/VERSION_COPY.TXT'
IN_OVCOPY	BOOL	FALSE
COPY_DONE	BOOL	TRUE
COPY_BUSY	BOOL	FALSE
COPY_ERR	BOOL	FALSE
COPY_EID	WORD	0

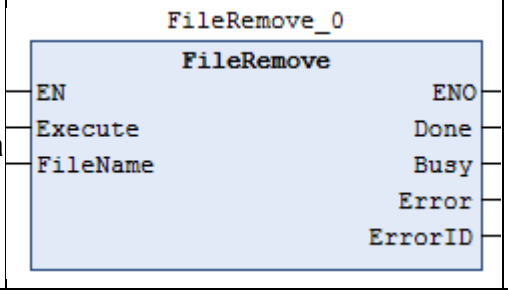
</td></tr>
</table>

2-9-13. Удаление файла FileRemove

(1) Обзор команды

Удаление указанного файла с карты памяти SD.

Удаление файла [FileRemove]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileRemove	Удаление файла		FileRemove_0(Execute:= , FileName:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileName	Указанное имя файла	STRING	Не более 1985 символов	-	Имя удаляемого файла
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды

Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

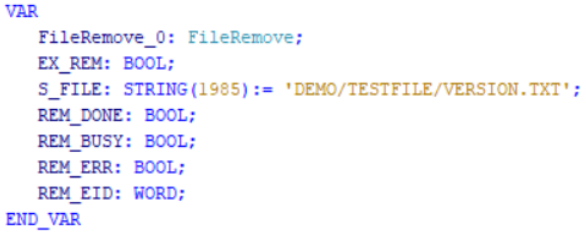
(3) Функция

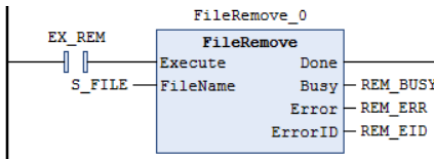
- Удаление файла, указанного в имени файла "FileName", с карты памяти SD.

(4) Примечание

- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Чтобы прочитать и записать файл, откройте его заново.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Если файл, указанный в "FileName", не существует.
- При доступе к файлу, указанному в "FileName".
- Если файл с таким же именем, как в "FileName", уже существует и доступен только для чтения.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		

<table>
<tr><td>Пример программы</td><td>
</td><td>

```
FileRemove_0(
    Execute:= EX_REM,
    FileName:= S_FILE,
    Done=> REM_DONE,
    Busy=> REM_BUSY,
    Error=> REM_ERR,
    ErrorID=> REM_EID);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
FileRemove_0	FileRemove	
EX_REM	BOOL	TRUE
S_FILE	STRING(1985)	'DEMO/TESTFILE/VERSION.TXT'
REM_DONE	BOOL	TRUE
REM_BUSY	BOOL	FALSE
REM_ERR	BOOL	FALSE
REM_EID	WORD	0

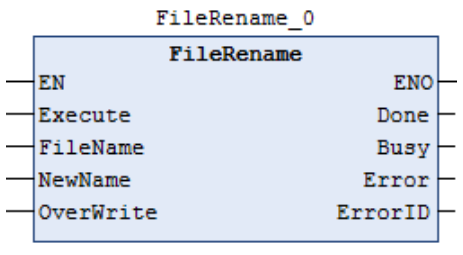
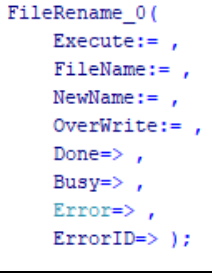
</td></tr>
</table>

2-9-14. Переименование файла FileRename

(1) Обзор команды

Изменение имени указанного файла или каталога на карте памяти SD.

Переименование файла [FileRename]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	PP_FunctionUtils		

Команда	Название	Графическое представление	ST
FileRename	Переименование файла		


```
FileRename_0(
    Execute:= ,
    FileName:= ,
    NewName:= ,
    OverWrite:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Включение команды	BOOL	FALSE/TRUE	FALSE	Ввод нарастающего фронта высокого уровня для запуска команды
FileName	Имя исходного файла	STRING	Не более 1985 символов	-	Имя исходного файла
NewName	Имя измененного файла	STRING	Не более 1985 символов	-	Имя измененного файла
OverWrite	Выбор функции перезаписи	BOOL	FALSE/TRUE	FALSE	True: Перезапись разрешена FALSE: Перезапись запрещена
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Признак завершения	BOOL	FALSE/TRUE	FALSE	Сигнал о завершении выполнения команды
Busy	Признак выполнения	BOOL	FALSE/TRUE	FALSE	Статус выполнения команды
Error	Признак ошибки	BOOL	FALSE/TRUE	FALSE	Статус ошибки при выполнении команды
ErrorID	Код ошибки	WORD	В качестве типа данных	-	Идентификатор ошибки при выполнении команды

(3) Функция

- Изменение имени файла или каталога, указанного в имени исходного файла "FileName", на обновленное имя файла "NewName" для файла на карте памяти SD.
- Если на карте памяти SD нет файла с таким же именем, как в "NewName", будет создан новый файл.
- Если на карте памяти SD уже существует файл с таким же именем, как в

"NewName", то в зависимости от допустимого значения параметра "OverWrite" будут выполнены следующие операции перезаписи:

Значение “OverWrite”	Операция
TRUE	Перезапись файла или каталога
FALSE	Файл или каталог не перезаписывается, возникло исключение

(4) Примечание

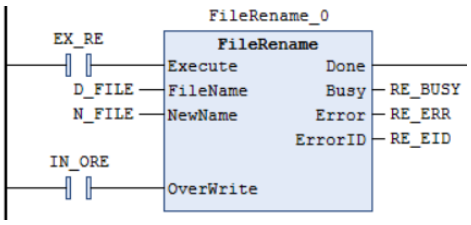
- Даже если значение переменной "Execute" станет FALSE, или время выполнения команды превысит цикл выполнения задачи, эта команда будет выполнена до конца. О нормальном завершении обработки можно судить по изменению значения переменной "Done" на true.
- Если каталоги "FileName" и "NewName" отличаются, файл будет перемещен в каталог, заданный параметром "NewName".
- Если извлечь карту памяти SD, когда файл открыт, он останется открытым. Но если вставить карту памяти SD обратно, чтение и запись файлов будут невозможны. Чтобы прочитать и записать файл, откройте его заново.
- Исключения возникают в описанных ниже случаях. Переменная "Error" принимает состояние true.
- Когда карта памяти SD не пригодна для использования.
- Карта памяти SD защищена от записи.
- Если файл, указанный в "FileName", не существует.
- Если значение "FileName" или "FileName" является недопустимым именем файла.
- При доступе к файлу, указанному в "FileName".
- Когда количество создаваемых файлов и каталогов превысит допустимое.
- Если файл с таким же именем, как в "FileName", уже существует, а переменная "OverWrite" принимает значение FALSE.
- Если файл с таким же именем, как в "DstFileName", уже существует, а переменная "OverWrite" принимает значение TRUE, и файл доступен только для чтения.
- При получении доступа к карте памяти SD и невозможности получения доступа из-за исключения.

(5) Пример программы

<table>
<tr><th>ДЕМО</th><th>LD</th><th>ST</th></tr>
<tr><td>Задание переменных</td><td colspan="2">

```
VAR
   FileRename_0: FileRename;
   EX_RE: BOOL;
   D_FILE: STRING(1985):= 'DEMO/TESTFILE/VERSION_COPY.TXT';
   N_FILE: STRING(1985):= 'DEMO/TESTFILE/NEW_VERSION.TXT';
   IN_ORE: BOOL;
   RE_DONE: BOOL;
   RE_BUSY: BOOL;
   RE_ERR: BOOL;
   RE_EID: WORD;
END_VAR
```

</td></tr>
<tr><td>Пример программы</td><td>



</td><td>

```
FileRename_0(
    Execute:= EX_RE,
    FileName:= D_FILE,
    NewName:= N_FILE,
    OverWrite:= IN_ORE,
    Done=> RE_DONE,
    Busy=> RE_BUSY,
    Error=> RE_ERR,
    ErrorID=> RE_EID);
```

</td></tr>
<tr><td>Результат операции</td><td colspan="2">

Expression	Type	Value
FileRename_0	FileRename	
EX_RE	BOOL	TRUE
D_FILE	STRING(1985)	'DEMO/TESTFILE/VERSION_COPY.TXT'
N_FILE	STRING(1985)	'DEMO/TESTFILE/NEW_VERSION.TXT'
IN_ORE	BOOL	FALSE
RE_DONE	BOOL	TRUE
RE_BUSY	BOOL	FALSE
RE_ERR	BOOL	FALSE
RE_EID	WORD	0

</td></tr>
</table>

3. Команды перемещения

3-1. Одна ось

3-1-1. Обзор команд для одной оси

Команда	Функция
MC_Power	Приведение оси в рабочее состояние
MC_Reset	Сброс внутренних ошибок оси
MC_Stop	Останов перемещения по командам контроллера
MC_Halt	Приостановка выполнения функциональных блоков
MC_Home	Возврат в исходное положение
MC_Jog	Запуск в толчковом режиме
MC_MoveAbsolute	Использование оси управления для перемещения в заданное абсолютное положение
MC_MoveAdditive	Ускорение для перемещения на дополнительное расстояние с заданной скоростью
MC_MoveRelative	Перемещение оси в относительное положение от текущего положения оси
MC_MoveSuperImposed	Исходя из предыдущего перемещения, наложенные скорость и ускорение обеспечивают перемещение на дополнительное расстояние
MC_MoveVelocity	Вал продолжает вращаться с заданной скоростью
MC_PositionProfile	Выполнение перемещений с учетом запланированного времени и положения
MC_VeloctyProfile	Выполнение перемещений с учетом запланированного времени и скорости
MC_AccelerationProfile	Выполнение перемещений с учетом запланированного времени и ускорения
MC_ReadActualPosition	Считывание текущего положения соответствующей оси

MC_ReadActualTorque	Считывание текущего крутящего момента соответствующей оси
MC_ReadActualVlocity	Считывание текущей скорости соответствующей оси
MC_ReadAxisError	Получение кода ошибки
MC_ReadBoolParameter	Получение значений параметров на основе порядкового номера параметра
MC_ReadParameter	Получение значения параметра на основе идентификатора параметра
SMC_ReadSetPosition	Считывание заданного положения текущей оси
SMC_ReadFBError	Считывание информации об ошибках функционального блока из архива
MC_WriteBoolParameter	Изменение значений параметров булевой переменной, заданной пользователем
MC_WriteParameter	Изменение специального параметра, заданного пользователем
SMC_ClearFBError	Очистка информации об ошибках функционального блока из архива
SMC_ErrorString	Считывание описания ошибки, соответствующего коду ошибки
SMC3_ReinitDrive	Повторное включение привода/оси
SMC3_ETC_WriteParameter_CoE	Настройка параметров COE для оси
MC_TouchProbe	Команды для контактного датчика

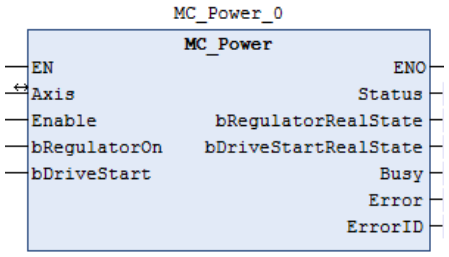
3-1-2. Команды для одной оси

3-1-2-1. Включение оси [MC_Power]

(1) Обзор команды

Эта команда используется для включения выбранной оси, чтобы перевести ее в рабочее состояние/вывести из этого состояния.

Включение оси [MC_Power]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Power	Включение оси		MC_Power_0(Axis:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , bRegulatorRealState=> , bDriveStartRealState=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE,FALSE	FALSE	Для включения функции обработки функциональных блоков должна быть в

					состоянии TRUE
bRegulatorOn	Включение	BOOL	TRUE,FALSE	FALSE	Для включения функционального блока должна быть в состоянии TRUE
bDriveStart	Включение привода	BOOL	TRUE,FALSE	FALSE	Для выключения функции обработки функциональных блоков при аварийном останове должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Status	Готовность к работе	BOOL	TRUE,FALSE	FALSE	TRUE, если ось готова к работе
bRegulatorRealState	Включено	BOOL	TRUE,FALSE	FALSE	Ось включена
bDriveStartRealState	Привод можно эксплуатировать	BOOL	TRUE,FALSE	FALSE	Работа привода не прерывается механизмом быстрого останова — состояние TRUE
Busy	В процессе выполнения	BOOL	TRUE,FALSE	FALSE	Обработка функционального блока не завершена — состояние TRUE
Error	Ошибка	BOOL	TRUE,FALSE	FALSE	TRUE, если возникает исключение
ErrorID	Код ошибки	SMC_ERROR	-	0	В нормальном режиме значение равно

					0. Если возникает исключение, отображается код ошибки.

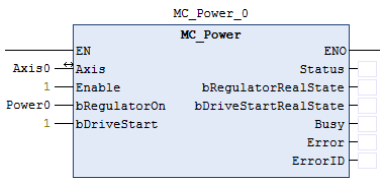
(3) Описание функции

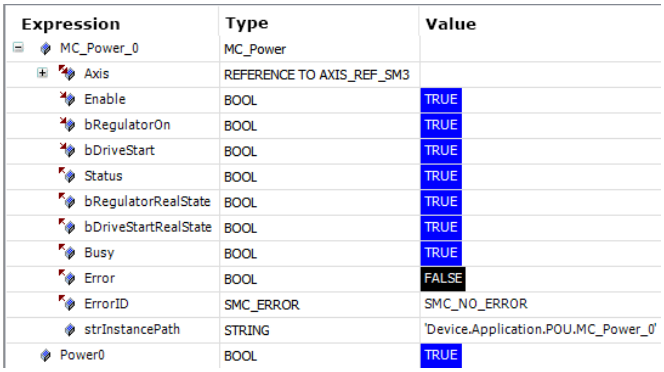
- Обработка входных данных при условии, что переменная Enable имеет состояние TRUE.
- Вызов команды MC_Power с переменной bRegulatoron = FALSE, чтобы установить nAxisState опорной оси в состояние off (выключено). После этого ось не сможет перемещаться.
- Если для отключенной оси ошибки отсутствуют, вызов команды MC_Power с переменной bRegulatorOn = TRUE устанавливает ее AxisState на паузу.
- Если обнаружена ошибка, ось примет состояние errorstop.
- Если входные переменные Enable, bRegulatoron и bDrivestart имеют состояние TRUE, но выходная переменная Status остается в состоянии FALSE в течение длительного времени, возможно, возникли неисправности в источнике питания привода. Если источник питания вышел из строя (в том числе во время эксплуатации), nAxisState опорной оси примет состояние errorstop.

(4) Примечания

Для каждой оси нужно записывать только одну команду MC_Power, иначе может возникнуть ситуация, похожая на наличие двух катушек.

(5) Пример программы

ДЕМО	LD	ST
Задание переменных	VAR MC_Power_0: MC_Power; Power0 :BOOL; END_VAR	
Пример программы		1 MC_Power_0(2 Axis:= Axis0, 3 Enable:= 1, 4 bRegulatorOn:= Power0, 5 bDriveStart:= 1, 6 Status=> , 7 bRegulatorRealState=> , 8 bDriveStartRealState=> , 9 Busy=> , 10 Error=> , 11 ErrorID=>);

Результат операции	

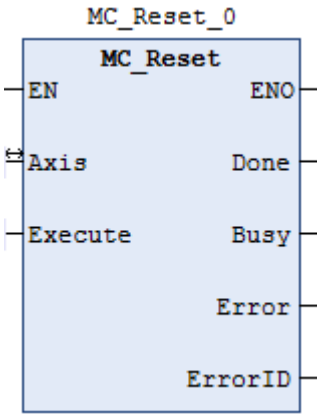
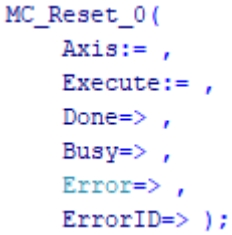
Expression	Type	Value
MC_Power_0	MC_Power	
Axis	REFERENCE TO AXIS_REF_SM3	
Enable	BOOL	TRUE
bRegulatorOn	BOOL	TRUE
bDriveStart	BOOL	TRUE
Status	BOOL	TRUE
bRegulatorRealState	BOOL	TRUE
bDriveStartRealState	BOOL	TRUE
Busy	BOOL	TRUE
Error	BOOL	FALSE
ErrorID	SMC_ERROR	SMC_NO_ERROR
strInstancePath	STRING	'Device.Application.POU.MC_Power_0'
Power0	BOOL	TRUE

3-1-2-2. Сброс ошибок оси [MC_Reset]

(1) Обзор команды

Сброс ошибок оси.

Сброс ошибок оси [MC_Reset]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Reset	Сброс ошибок оси		


```
MC_Reset_0(
    Axis:= ,
    Execute:= ,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE,FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Готово	BOOL	TRUE,FALSE	FALSE	TRUE, если выполнен сброс
Busy	В процессе выполнения	BOOL	TRUE,FALSE	FALSE	TRUE, если выполнение функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE,FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Если с валом установлена надежная связь, состояние оси изменится с errorstop на Standstill, после чего ось перейдет из ненормального состояния в нормальное рабочее состояние.
- Функциональный блок возвращает ошибку SMC_R_NO_ERROR_TO_RESET при его вызове в состоянии, отличающимся от состояния errorstop.

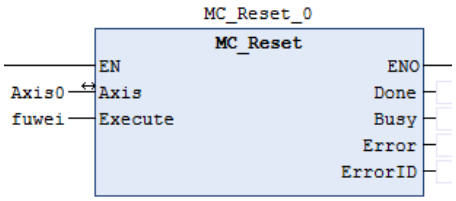
(4) Примечания

Команда MC_RESET не может восстановить связь, и появится сообщение об ошибке " SMC_R_CANNOT_RESET_COMMUNICATION _ERROR ".

(5) Пример программы

ДЕМО	LD	ST
Задание переменных		
Пример программы		
Результат операции		

```
VAR
    power: BOOL;
    fuwei:BOOL;
    MC_Power_0: MC_Power;
    MC_Reset_0: MC_Reset;
    MC_Home_0: MC_Home;
END_VAR
```




```
MC_Reset_0(
    Axis:= Axis0,
    Execute:= fuwei,
    Done=> ,
    Busy=> ,
    Error=> ,
    ErrorID=> );
```

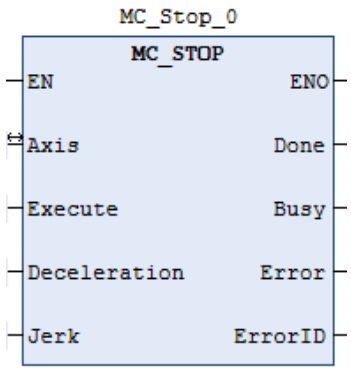
MC_Reset_0	MC_Reset	
Axis	REFERENCE TO AXI...	
Execute	BOOL	TRUE
Done	BOOL	TRUE
Busy	BOOL	FALSE
Error	BOOL	FALSE
ErrorID	SMC_ERROR	SMC_NO_ERROR
OldExecute	BOOL	TRUE
rbp	MC_ReadBoolParam...	
wbp	MC_WriteBoolParam...	
wState	WORD	100
wCount	WORD	0
wCount2	WORD	0
pAxisStored	POINTER TO AXIS_...	16#FFFF870876565...
eErrorIDStored	SMC_ERROR	SMC_NO_ERROR
strInstancePath	STRING	'Device.Application.P...

3-1-2-3. Останов перемещения по командам контроллера [MC_Stop]

(1) Обзор команды

Останов оси с торможением.

Останов перемещения по командам контроллера [MC_Stop]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Stop	Принудительный останов		MC_Stop_0(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск перемещения оси по нарастающему фронту
Deceleration	Скорость торможения	LREAL	Положительное значение	0	Скорость торможения. Единица измерения — командная единица/$с^2$
Jerk	Толчковый режим	LREAL	Положительное значение	0	Перемещение в толчковом режиме. Единица измерения — командная единица/$с^3$
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Готово	BOOL	TRUE/FALSE	FALSE	После выполнения принимает состояние TRUE

Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	После получения команды принимает состояние TRUE
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

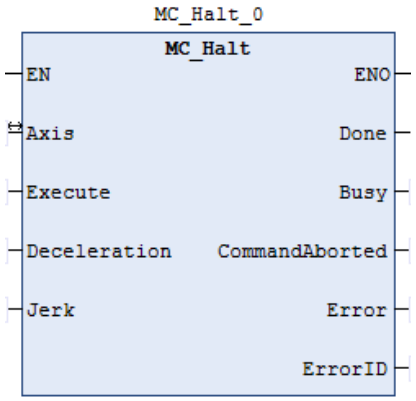
- Команда MC_Stop переводит ось в состояние остановки. Таким образом, перемещение текущего экземпляра функционального блока прерывается. Исключение — выполняемый в данный момент экземпляр команды Mc_Stop. Его выполнение не прерывается. Вместо этого, экземпляр MC_Stop, который был только что запущен, возвращает ошибку).
- Пока ось неподвижна, никакой другой экземпляр не может обеспечить ее перемещение. Если ось достигла нулевой скорости, выход Done принимает состояние TRUE. Пока вход Execute находится в состоянии TRUE, ось остается неподвижной. Если Execute переходит в состояние FALSE, а выход Done принимает состояние TRUE, ось остановится.

3-1-2-4. Приостановка перемещения [MC_Halt]

(1) Обзор команды

Перед остановкой ось замедляется, а после остановки оси можно будет возобновить выполнение функции.

Приостановка перемещения [MC_Halt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Halt	Приостановка перемещения		MC_Halt_0(Axis:= , Execute:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
Deceleration	Скорость торможения	LREAL	Положительное значение	0	Скорость торможения. Единица измерения — командная единица/$с^2$
Jerk	Толчковый режим	LREAL	Положительное значение	0	Перемещение в толчковом режиме. Единица измерения — командная единица/$с^3$
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Готово	BOOL	TRUE/FALSE	FALSE	После выполнения

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					принимает состояние TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	TRUE, если выполнение функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	TRUE, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Этот функциональный блок управляет остановом опорных осей. Если одновременно выполняются другие функциональные блоки, текущая операция прерывается. Ось совершает дискретное перемещение, пока не достигнет скорости 0. Если установлен выход Done команды MC_Halt, ось остановится. Пока команда MC_Halt активна, можно подать новую команду на перемещение, чтобы прервать выполнение команды MC_Halt. В отличие от команды MC_Stop, команду MC_Halt можно прерывать.

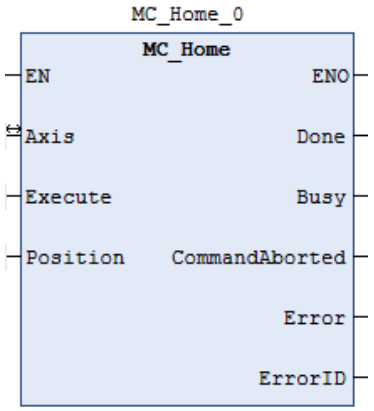
3-1-2-5. Возврат оси в исходное положение [MC_Home]

(1) Обзор команды

Эта команда возвращает шаговый электродвигатель в исходное положение. Процедура возврата в исходное положение зависит от режима возврата в исходное положение, поддерживаемого драйвером шины.

Возврат оси в исходное положение [MC_Home]			
Условие выполнения	Нормальное включение/выключение	Совместимые модели	PMP3xx

	триггера		
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Home	Возврат оси в исходное положение		MC_Home_0(Axis:= , Execute:= , Position:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение при нарастающем фронте
Position	Положение, которого достигла ось	LREAL	Диапазон данных	0	Исходное положение оси
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	После завершения выполнения принимает состояние TRUE

Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Функция

- Этот функциональный блок предназначен для возврата в исходное положение, т.е. нулевое положение оси.
- Рабочее состояние этого функционального блока — Standstill (остановка), а состояние выполнения команды - homing (возврат в исходное положение). Другие состояния запускать нельзя.
- Запустите данную команду при нарастающем фронте Execute.
- Перед возвратом в исходное положение нужно настроить параметры возврата в исходное положение для драйвера шины, такие как режим возврата в исходное положение, скорость, ускорение и т.д. Обратитесь к руководству по эксплуатации драйвера. Например, нужную информацию можно найти в руководстве пользователя сервопривода серии PSD5C1, глава 6.5 «Режим HM».
- Чтобы выполнить возврат к нулю, для драйвера общей шины нужно ввести данные индекса и субиндекса, представленные в таблице ниже:

Индекс	Субиндекс	Описание
0x6098	00h	Режим возврата в исходное положение
0x6099	01h	Скорость выполнения процесса от начала возврата в исходное положение до нахождения нулевой точки. При большей скорости время возврата в исходное положение сокращается.

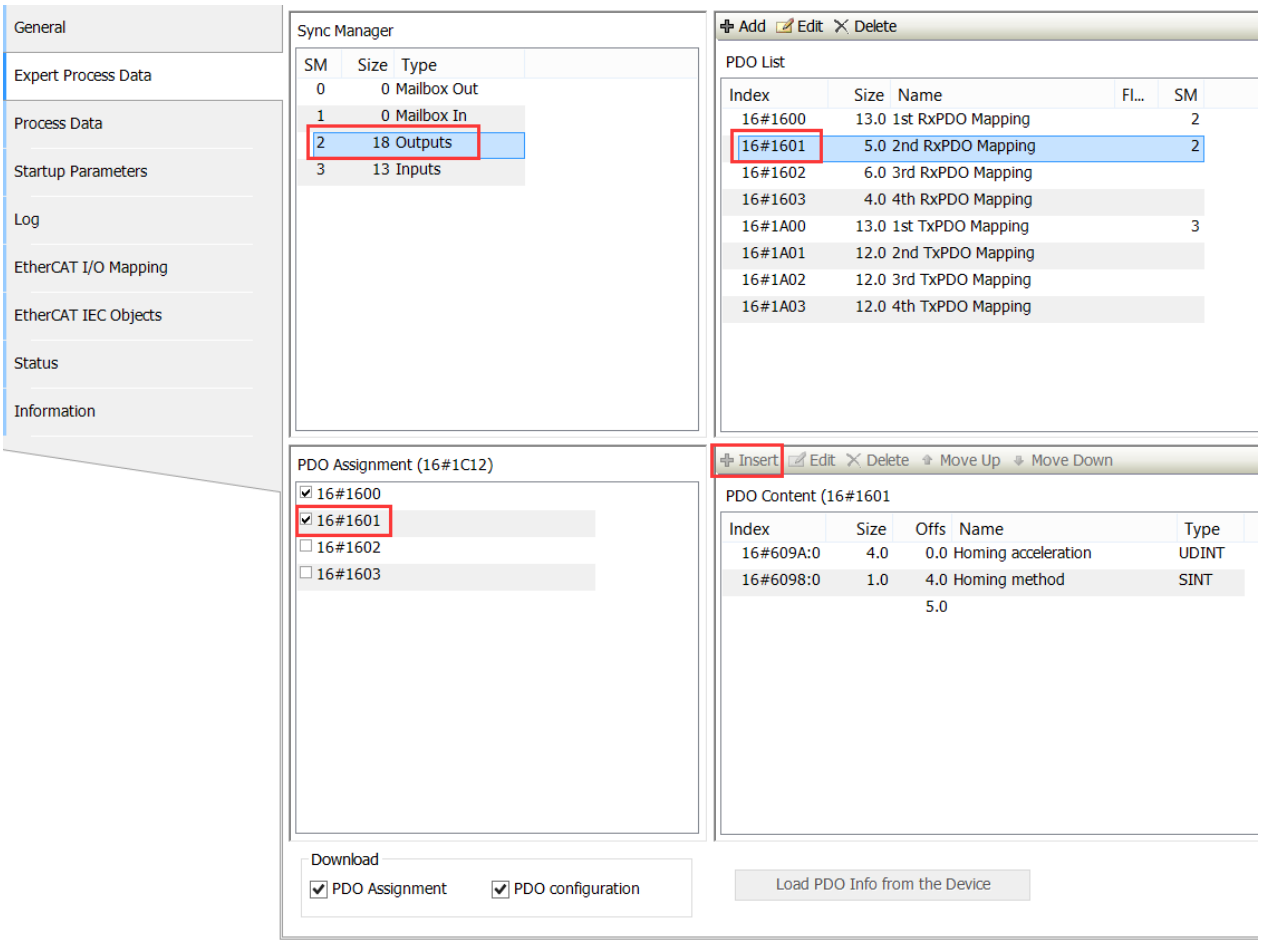
Индекс	Субиндекс	Описание
0x6099	02h	Скорость выполнения процесса от нахождения нулевой точки до завершения возврата в исходное положение. Чем меньше скорость, тем выше точность.
0x609A	00h	Ускорение при возврате в исходное положение

(4) Пример использования

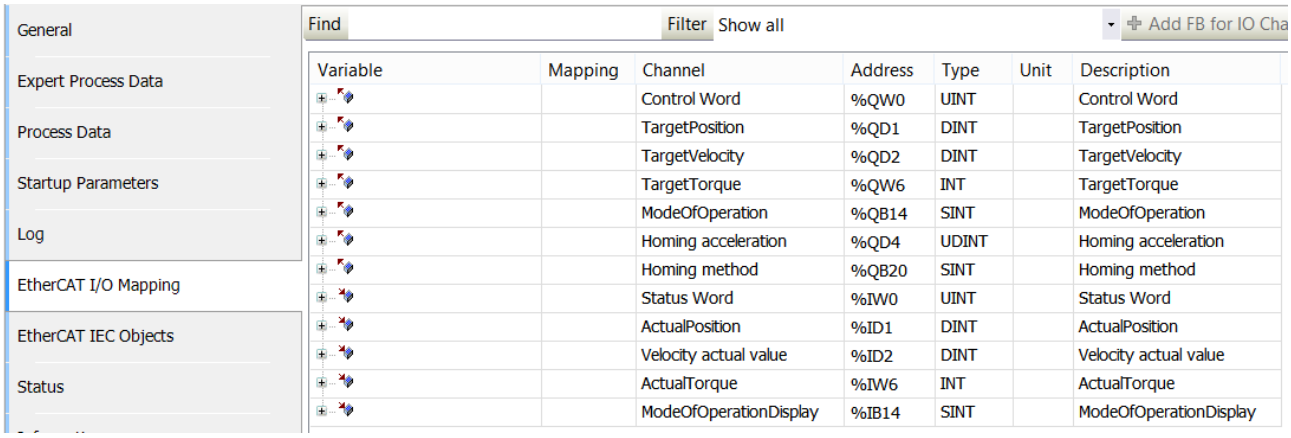
Пример 1: В качестве примера можно привести сервопривод PROMPOWER PSD5C, в котором ось возвращается в исходное положение в режиме возврата в исходное положение 1. P5-22 — это адрес настройки положительного предела, значение по умолчанию — 1, что соответствует клемме сервопривода SI1. P5-23 — это адрес настройки отрицательного предела, значение по умолчанию — 2, что соответствует клемме сервопривода SI2. P5-27 — это адрес исходного положения, значение по умолчанию — 3, что соответствует клемме сервопривода SI3.

Создание программы:

Выберите [expert settings (Настройки для продвинутых пользователей)], выберите 1601 в меню [Expert process data (Экспертные данные процесса)] - [PDO assignment (Распределение PDO)], добавьте 6098h, 609Ah в 1601.

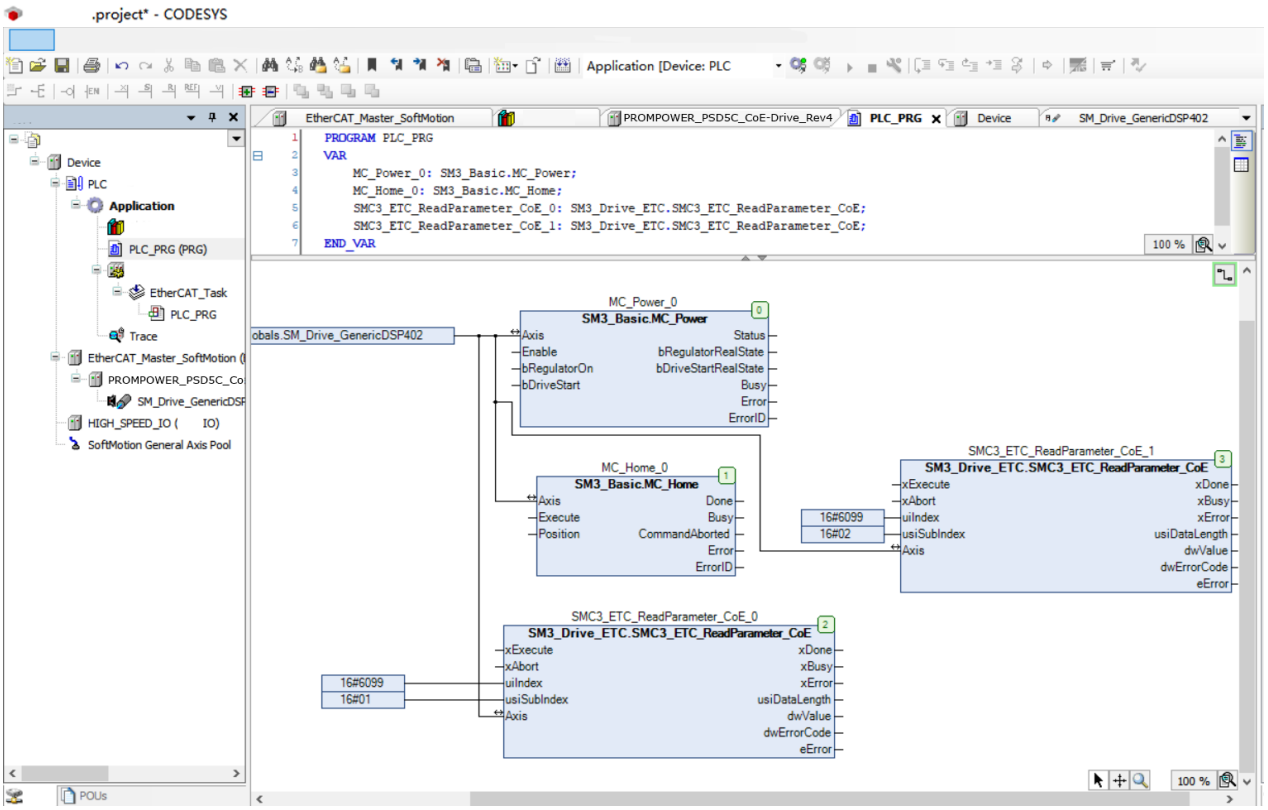


Новые добавленные параметры можно просмотреть в разделе меню [EtherCAT I/O mapping (Отображение вводов/выводов EtherCAT)].

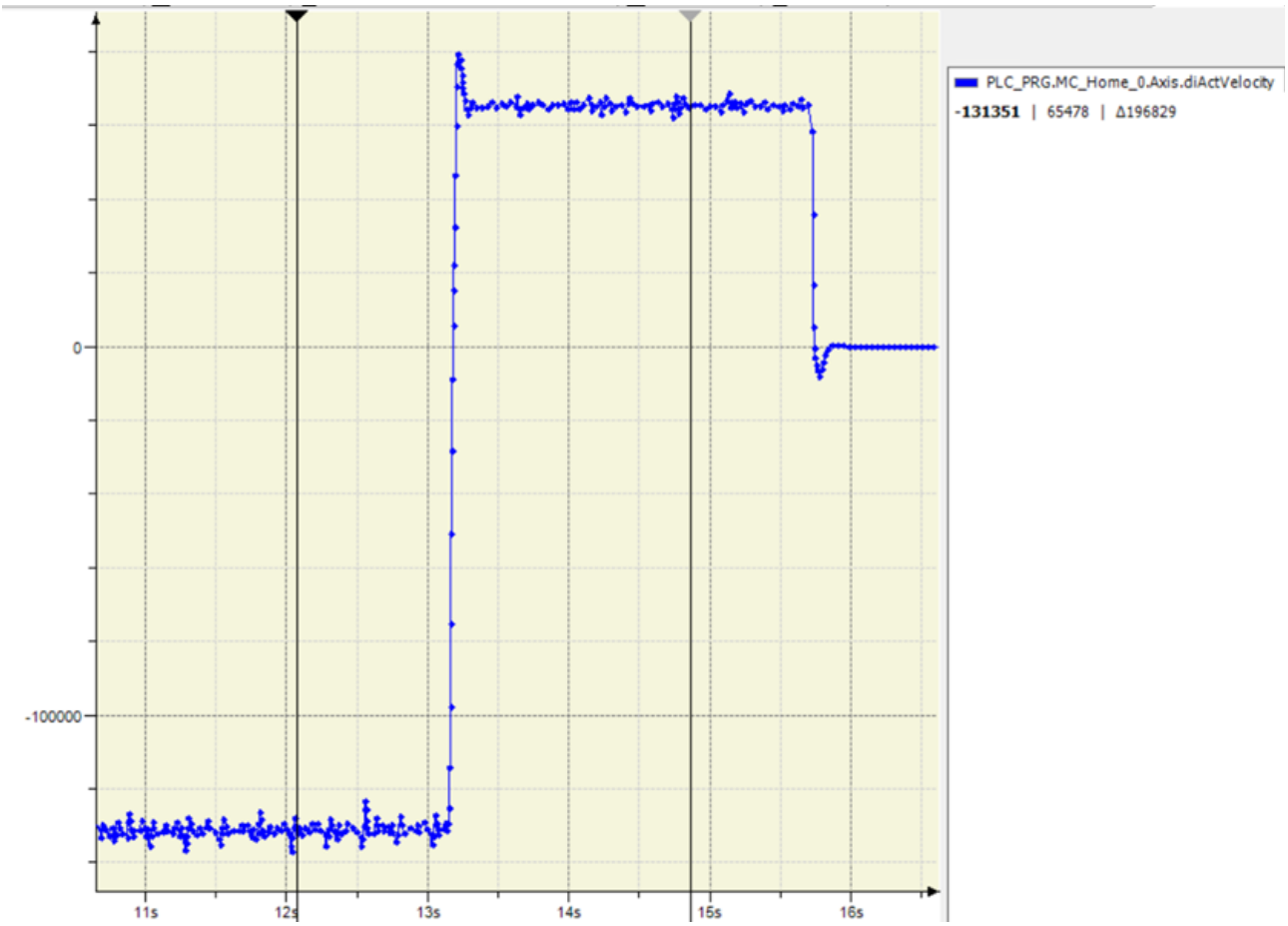


Variable	Mapping	Channel	Address	Type	Unit	Description
		Control Word	%QW0	UINT		Control Word
		TargetPosition	%QD1	DINT		TargetPosition
		TargetVelocity	%QD2	DINT		TargetVelocity
		TargetTorque	%QW6	INT		TargetTorque
		ModeOfOperation	%QB14	SINT		ModeOfOperation
		Homing acceleration	%QD4	UDINT		Homing acceleration
		Homing method	%QB20	SINT		Homing method
		Status Word	%IW0	UINT		Status Word
		ActualPosition	%ID1	DINT		ActualPosition
		Velocity actual value	%ID2	DINT		Velocity actual value
		ActualTorque	%IW6	INT		ActualTorque
		ModeOfOperationDisplay	%IB14	SINT		ModeOfOperationDisplay

Считайте скорость возврата в исходное положение в 6099h через SMC3_ETC_ReadParameter_CoE, установите режим возврата в исходное положение 1 в пункте меню [EtherCAT I/O mapping (Отображение вводов/выводов EtherCAT)], настройте ускорение при возврате в исходное положение на 13107200. После включения через MC_Power, включите функциональный блок MC_Home, затем подайте сигнал с клеммы SI1.



Проверьте скорость возврата в исходное положение через опцию Trace (Отслеживание).

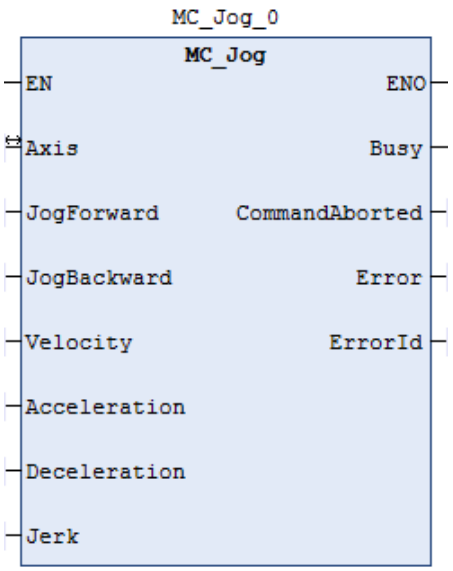


3-1-2-6. Запуск в толчковом режиме [MC_Jog]

(1) Обзор команды

Эта команда используется для ручного управления перемещением оси в заданном направлении.

Запуск в толчковом режиме [MC_Jog]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Jog	Запуск в толчковом режиме		MC_Jog_0(Axis:= , JogForward:= , JogBackward:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Busy=> , CommandAborted=> , Error=> , ErrorId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Ось	Ось	AXIS_REF_SM3	-	-	Выбранная ось
JogForward	Прямой ход в толчковом режиме	BOOL	TRUE/FALSE	FALSE	Если переменная JogForward принимает состояние TRUE, ось будет перемещаться вперед с заданными параметрами (Velocity, Acceleration, Deceleration и Jerk), а если переменная JogBackward также имеет состояние TRUE, ось не будет перемещаться.
JogBackward	Обратный ход в толчковом режиме	BOOL	TRUE/FALSE	FALSE	Если переменная JogBackward принимает состояние TRUE, ось будет

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					перемещаться назад с заданными параметрами (Velocity, Acceleration, Deceleration и Jerk), а если переменная JogForward также имеет состояние TRUE, ось не будет перемещаться.
Velocity	Заданная скорость	LREAL	0, положительное число	0	Максимальная скорость [импульсов/с]
Acceleration	Заданная скорость ускорения	LREAL	положительное число	0	Значение ускорения [импульсов/с^2]
Deceleration	Заданная скорость торможения	LREAL	положительное число	0	Значение торможения [импульсов/с^2]
Jerk	Заданная скорость в толчковом режиме	LREAL	положительное число	0	Значение скорости в толчковом режиме [импульсов/с^3]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Идентификация ошибок

(3) Описание функции

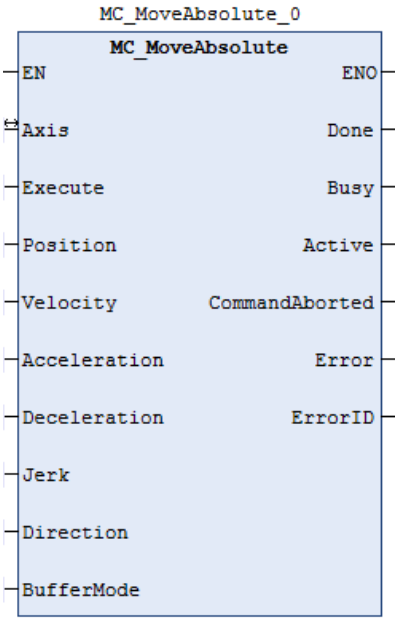
- Выполнение работы в толчковом режиме в соответствии с заданным значением параметра Velocity (заданная скорость).
- Для выполнения прямого хода установите переменную JogForward (прямой ход) в состояние TRUE. Для выполнения обратного хода установите переменную JogBackward (обратный ход) в состояние TRUE.
- Если установить обе переменные JogForward (прямой ход) и JogBackward (обратный ход) в состояние TRUE, то перемещения не будет. Если уставка скорости команды MC_Jog превышает максимальную скорость хода в толчковом режиме, заданную в параметрах оси, работа все равно будет выполняться на максимальной скорости хода в толчковом режиме.

3-1-2-7. Абсолютное положение [MC_MoveAbsolute]

(1) Обзор команды

Эта команда используется для перемещения оси управления в заданное абсолютное положение в соответствии с установленными параметрами.

Абсолютное положение [MC_MoveAbsolute]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveAbsolute	абсолютное положение		MC_MoveAbsolute_0(Axis:= , Execute:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
Position	Положение	LREAL	Диапазон данных	0	Заданное положение перемещения
Velocity	Заданная скорость	LREAL	0, положительное число	0	Максимальная скорость [импульсов/с]
Acceleration	Заданная скорость ускорения	LREAL	положительное число	0	Значение ускорения [импульсов/$с^2$]

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Deceleration	Заданная скорость торможения	LREAL	положительное число	0	Значение торможения [импульсов/с^2]
Jerk	Заданная скорость в толчковом режиме	LREAL	положительное число	0	Значение скорости в толчковом режиме [импульсов/с^3]
Direction	Направление	MC_DIRECTION	3: fastest 2: current 1: Positive 0: shortest -1: Negative	0	Fastest: Автоматический выбор направления перемещения с максимальной скоростью Current: Перемещение в текущем направлении Positive: Прямой ход Shortest: Выбор направления перемещения по кратчайшему пути Negative: Обратный ход
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Если функциональный блок имеет статус Busy (В процессе выполнения), допускается только BufferMode=Aborting.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Идентификация ошибок

(3) Описание функции

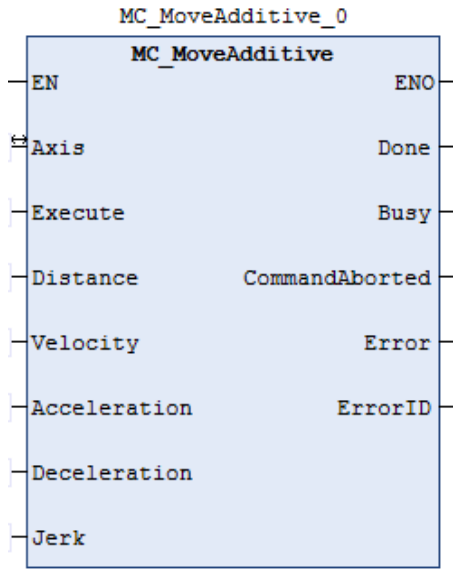
- Этот функциональный блок перемещает вал в абсолютное положение и использует значения скорости, торможения, ускорения и скорости в толчковом режиме. Если соответствующая уставка не задана, выполнение команды завершится при нулевой скорости (т.е. в состоянии остановки)
- Выполнение команды на запуск при нарастающем фронте Execute.

3-1-2-8. Наложение положения [MC_MoveAdditive]

(1) Обзор команды

На исходное заданное положение оси накладываются данные, заданные параметром Distance, которые используются для наложения положения в процессе управления перемещением оси.

Наложение положения [MC_MoveAdditive]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveAdditive	Наложение положения		MC_MoveAdditive_0(Axis:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
Distance	Положение	LREAL	Диапазон данных	0	Эти данные являются данными о наложенном положении
Velocity	Заданная скорость	LREAL	0, положительное число	0	Максимальная скорость [импульсов/с]
Acceleration	Заданная скорость ускорения	LREAL	положительное число	0	Значение ускорения [импульсов/$с^2$]
Deceleration	Заданная скорость торможения	LREAL	положительное число	0	Значение торможения [импульсов/$с^2$]
Jerk	Заданная скорость в толчковом режиме	LREAL	положительное число	0	Значение скорости в толчковом режиме [импульсов/$с^3$]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Идентификация ошибок

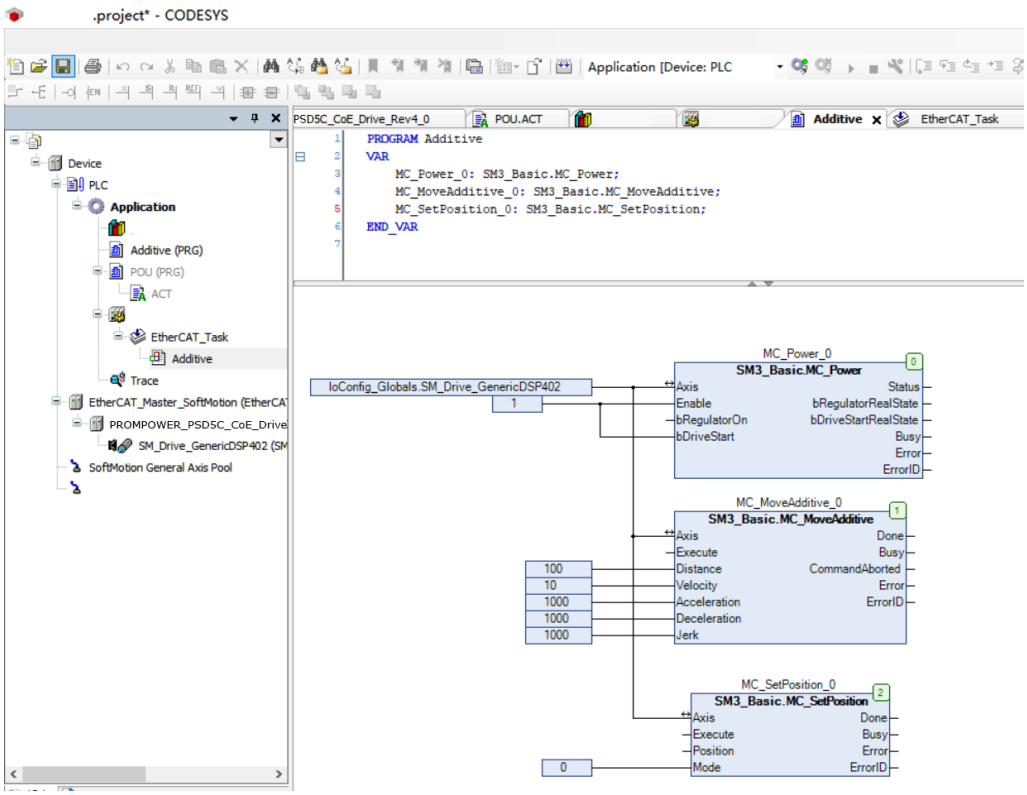
(3) Описание функции

- Этот функциональный блок представляет собой команду наложения положения, а данные переменной Distance - это данные наложения оси.
- Если рабочее состояние этого функционального блока — Discrete Motion, то переменная CommandAbort других команд примет состояние ON во время работы.
- После остановки эту команду можно выполнять независимо для выполнения требований к относительному позиционированию.
- Переменная Acceleration или Deceleration равна нулю, и выполнение команды происходит ненормально, но состояние оси — Discrete Motion (Дискретное перемещение).
- Выполнение команды на запуск при нарастающем фронте Execute.

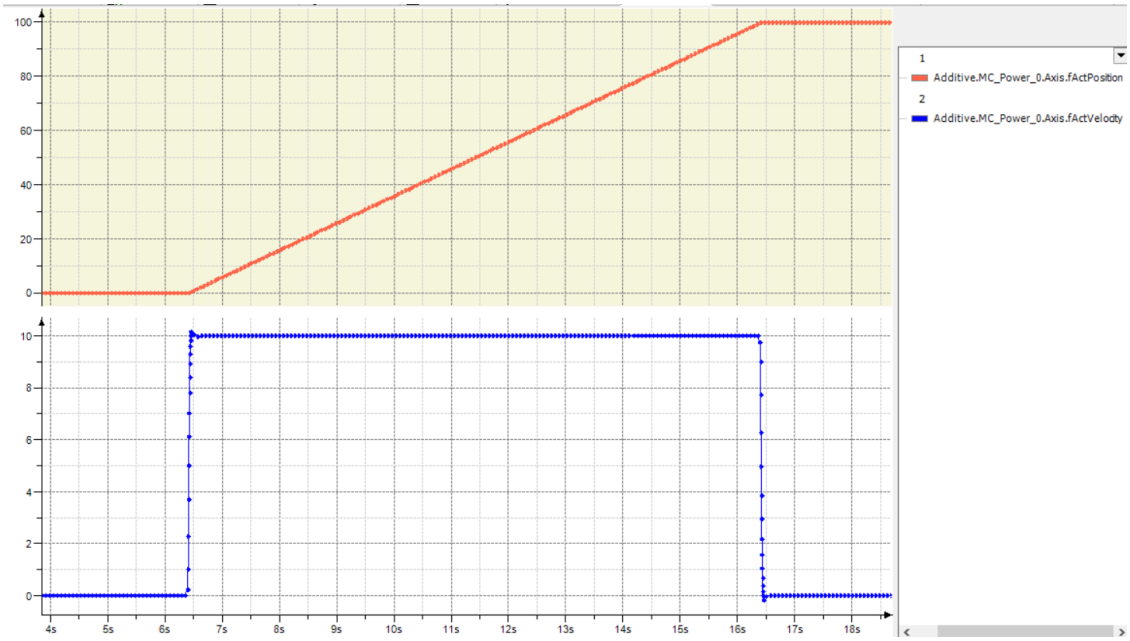
(4) Пример использования

Пример 1: Эту команду можно напрямую использовать и запускать через команду "MC_MoveAdditive" после включения команды "MC_Power".

1. Программирование: Можно сначала установить положение 0 через "MC_SetPosition", включить команду, а затем установить состояние ON для MC_ MoveAdditive. Можно настроить скорость 10 и положение 100.

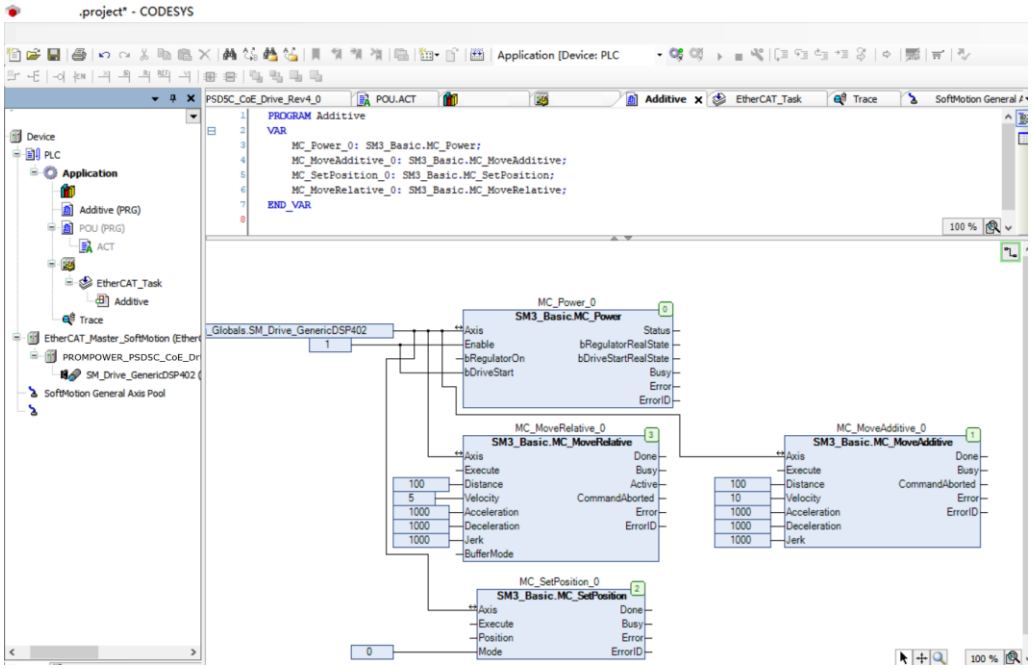


2. Положение и скорость работы можно просматривать через опцию Trace (Отслеживание).

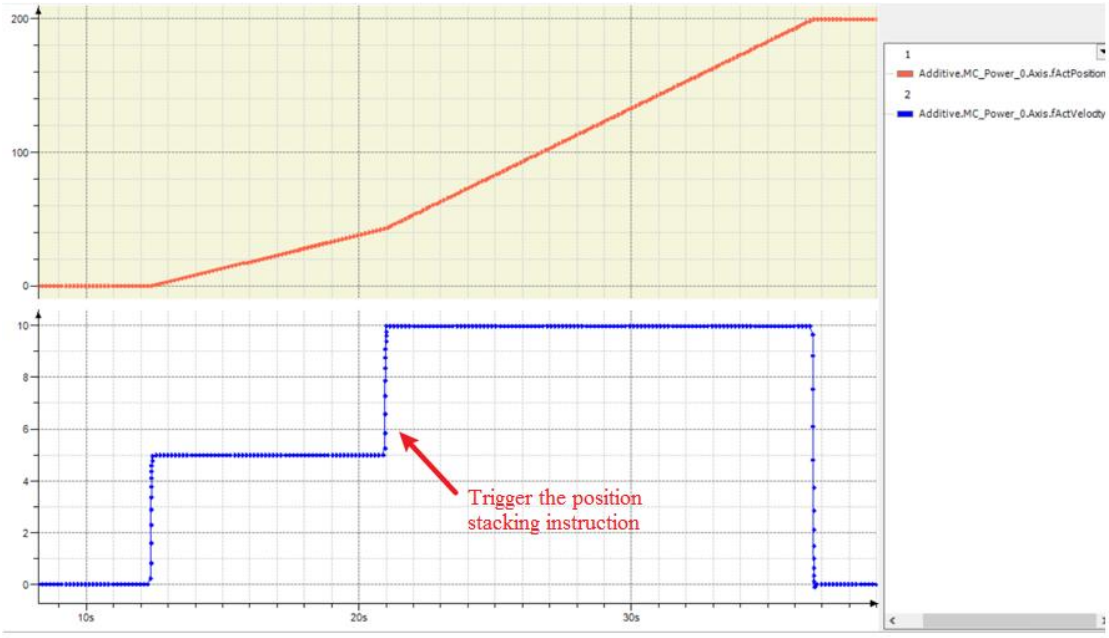


Пример 2: После включения команды "MC_Power" сначала выполните команду "MC_MoveRelative", а затем "MC-MoveAdditive".

1. Запишите программу: Команда "MC_MoveRelative" выполняется при скорости 5 и заданном положении 100. Команду можно включать во время работы или после ожидания завершения выполнения команды "MC_MoveRelative". Скорость, заданная для "MC_MoveAdditive", равна 10, а заданное положение — 100. Скорость выполнения после включения равна 10, а конечное положение — 100. То есть положение накладывается, а ускорение не накладывается, сбрасывается только скорость.



2. Текущие скорость и положение можно просматривать через опцию Trace (Отслеживание).

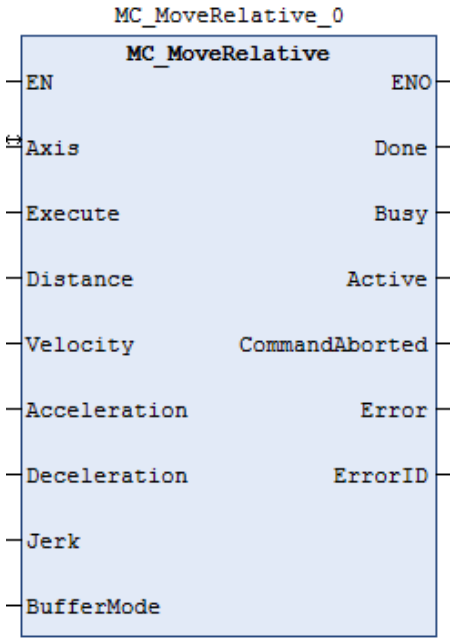


3-1-2-9. Относительное положение [MC_MoveRelative]

(1) Обзор команды

Ось работает в относительном положении, которое задается параметром Distance.

Относительное положение [MC_MoveRelative]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveRelative	Относительное положение		MC_MoveRelative_0(Axis:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , BufferMode:= , Done=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

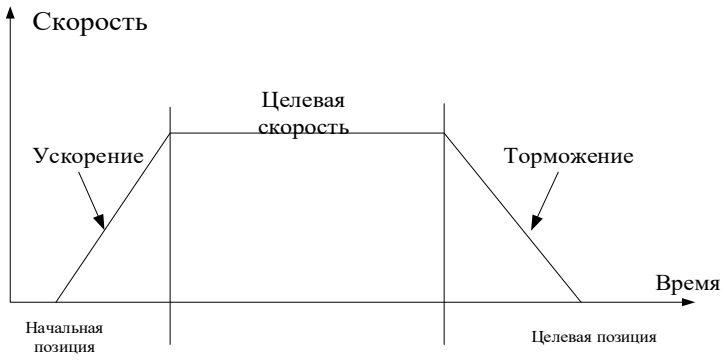
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
Distance	Положение	LREAL	Диапазон данных	0	Относительное расстояние между заданным положением и текущим положением
Velocity	Заданная скорость	LREAL	0, положительное число	0	Максимальная скорость [импульсов/с]
Acceleration	Заданная скорость ускорения	LREAL	положительное число	0	Значение ускорения [импульсов/с^2]
Deceleration	Заданная скорость торможения	LREAL	положительное число	0	Значение торможения [импульсов/с^2]
Jerk	Заданная скорость в толчковом режиме	LREAL	положительное число	0	Значение скорости в толчковом режиме [импульсов/с^3]
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Если функциональный блок имеет статус Busy (В процессе выполнения), допускается только BufferMode=Aborting.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
Command Aborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

- Рабочее состояние этого функционального блока — Standstill (остановка), а состояние выполнения команды - Discrete Motion (Дискретное перемещение). Обращайте внимание на рабочее состояние этой оси во время выполнения команды. Выполнение других команд на этой оси не должно быть прервано, либо текущая команда не должна быть прервана другими командами.
- Выполнение команды на запуск при нарастающем фронте Execute. Эта команда действительна для повторяющихся нарастающих фронтов в режиме Discrete Motion (Дискретное перемещение) и может каждый раз обновлять последнее Position (Положение).
- Переменная Acceleration или Deceleration равна нулю, и выполнение команды происходит ненормально, но состояние оси — Discrete Motion (Дискретное перемещение).

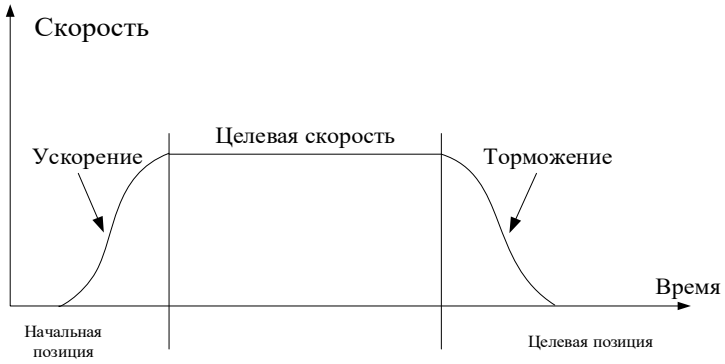
- Трапецеидальное ускорение и торможение

Параметры Velocity, Acceleration и Deceleration имеют значение, но параметр Jerk равен 0.



- S-кривая ускорения и торможения

Параметры Velocity, Acceleration, Deceleration и Jerk имеют значения.

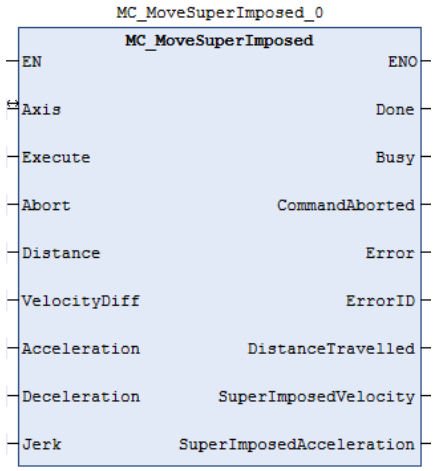


3-1-2-10. Наложение на команду относительного перемещения [MC_MoveSuperImposed]

(1) Обзор команды

Данные о скорости и положении оси накладываются на команду перемещения на основе исходных заданных скорости и положения.

Наложение на команду относительного перемещения [MC_MoveSuperImposed]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveSuperImposed	Наложение на команду относительного перемещения		MC_MoveSuperImposed_0(Axis:= , Execute:= , Abort:= , Distance:= , VelocityDiff:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , DistanceTravelled=> , SuperImposedVelocity=> , SuperImposedAcceleration=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
Abort	Прерывание	BOOL	TRUE/FALSE	FALSE	Прерывание текущего перемещения и сброс всех выходов
Distance	Положение	LREAL	Диапазон данных	0	Эти данные являются данными о наложенном положении
VelocityDiff	Скорость наложения	LREAL	0, положительное число	0	Максимальная скорость наложенного перемещения [импульсов/с]

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Acceleration	Заданная скорость ускорения	LREAL	положительное число	0	Значение ускорения [импульсов/с^2]
Deceleration	Заданная скорость торможения	LREAL	положительное число	0	Значение торможения [импульсов/с^2]
Jerk	Заданная скорость в толчковом режиме	LREAL	положительное число	0	Значение скорости в толчковом режиме [импульсов/с^3]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Этот функциональный блок предназначен для команд наложения положения и скорости, а переменные VelocityDiff и Distance представляют собой

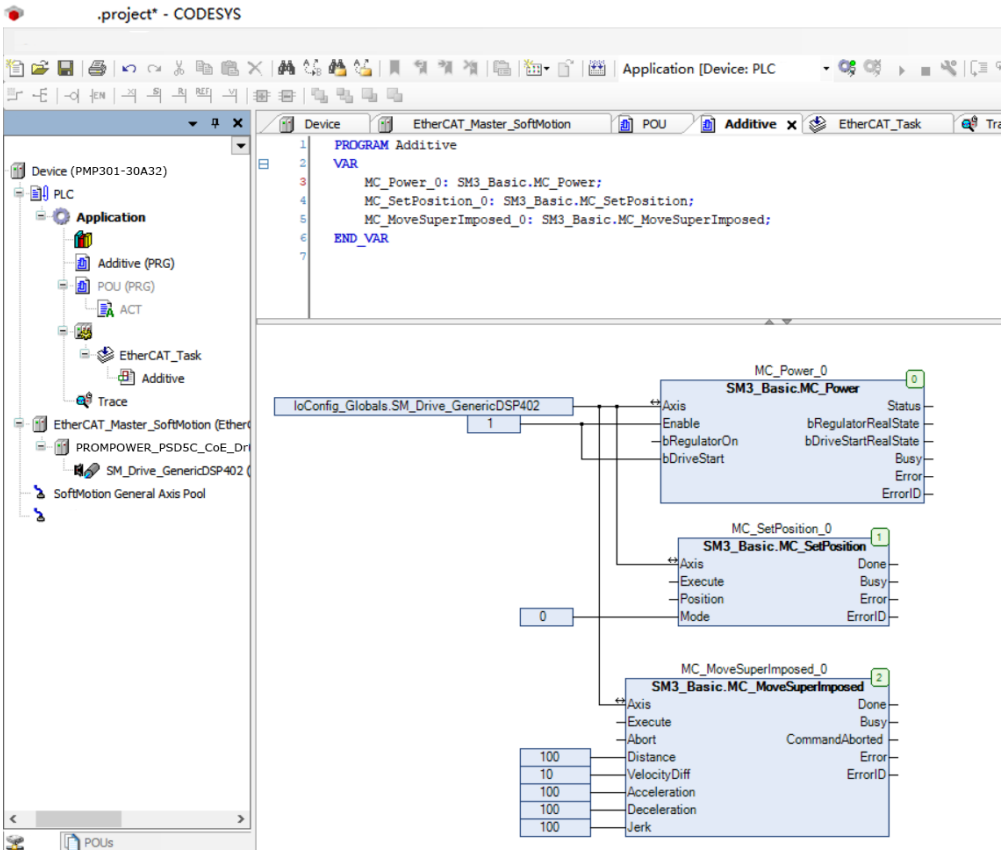
скорость и положение, наложенные на другие команды, соответственно.

- Команда MC_MoveSuperImposed может накладываться на любую другую команду.
- Команду MC_MoveSuperImposed можно прерывать командой MC_ MoveSuperImposed.
- В состоянии StandStill (Остановка) функциональный блок MC_ MoveSuperimposed аналогичен функциональному блоку MC_ MoveRelative.
- Выполнение команды на запуск при нарастающем фронте Execute.

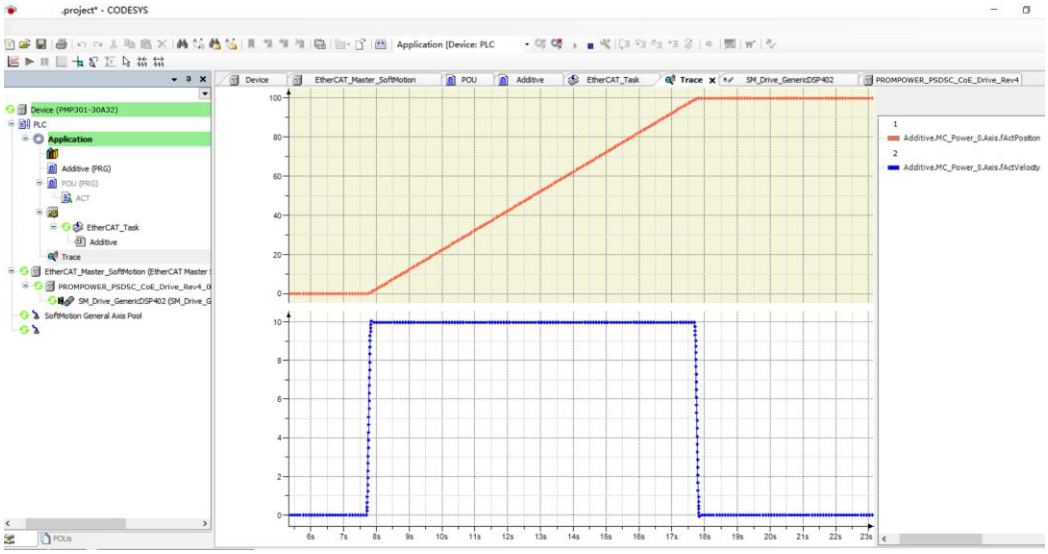
(4) Применение

Пример 1: Эту команду можно напрямую запускать через команду "MC_MoveSuperImposed" после включения команды "MC_Power".

Программирование: Можно сначала установить положение 0 через "MC_SetPosition", включить команду, а затем выполнить MC_ MoveSuperImplied. Можно настроить скорость работы 10 и рабочее положение 100.

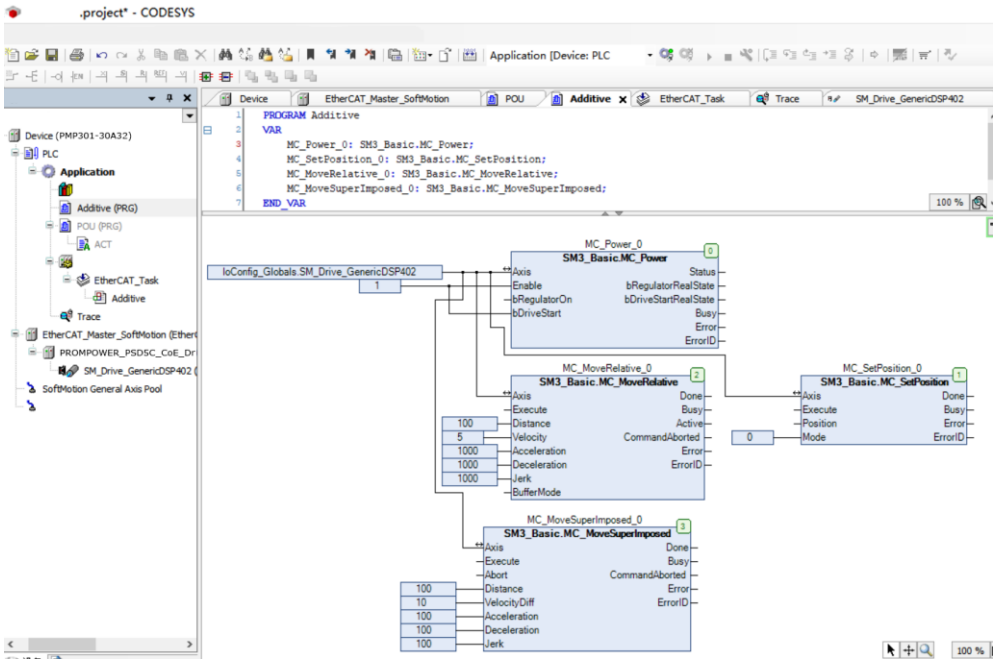


Текущие скорость и положение можно просматривать через опцию Trace (Отслеживание).

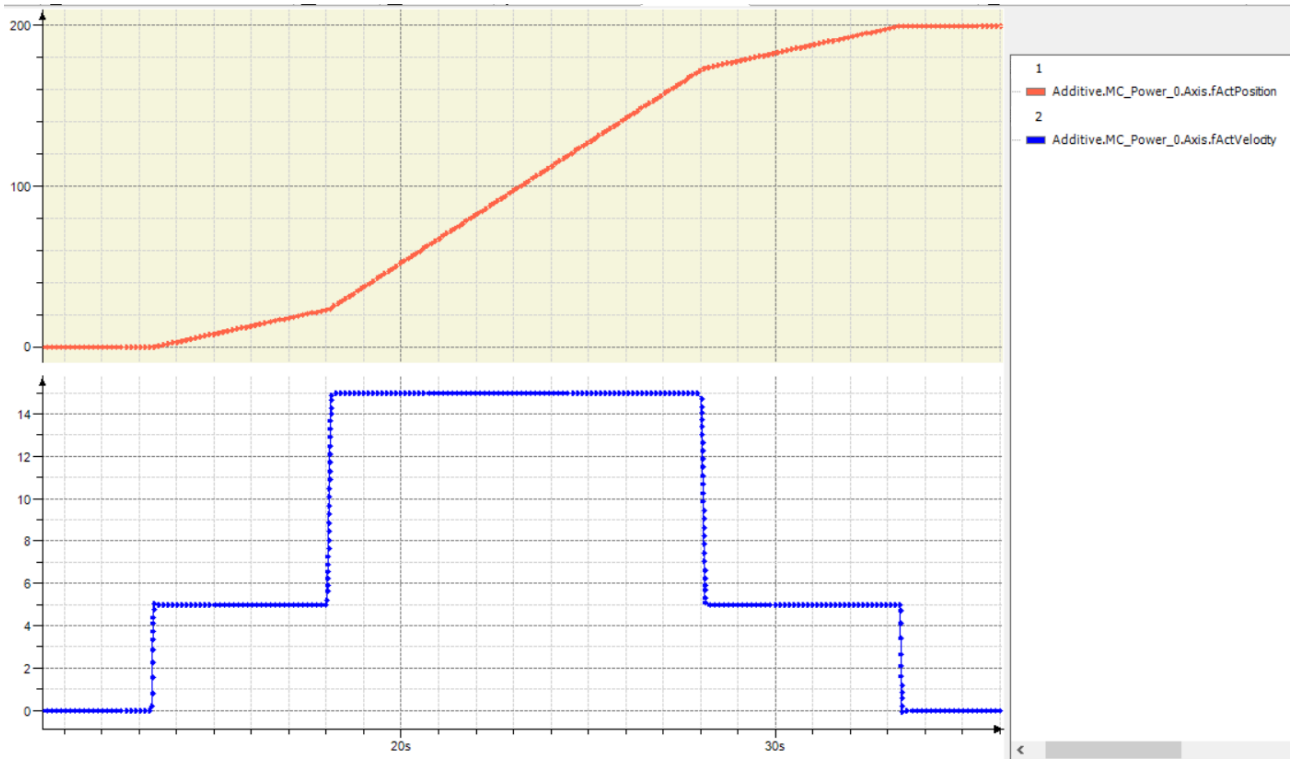


Пример 2: После включения команды "MC_Power" сначала выполните команду "MC_MoveRelative", а затем "MC_MoveSuperImposed".

Программирование: Команда "MC_MoveRelative" выполняется при скорости 5 импульсов/с и заданном положении 100 импульсов. Команду MC_MoveSuperImposed можно включать во время работы или после ожидания завершения выполнения команды "MC_MoveRelative". Скорость, заданная для "MC_MoveSuperImposed", равна 10 импульсов/с, а заданное положение — 100 импульсов. Положение после включения представляет собой сумму заданных положений, настроенных в двух командах. Команда MC_MoveSuperImposed одновременно накладывается на положение и скорость.



Текущие скорость и положение можно просматривать через опцию Trace (Отслеживание).

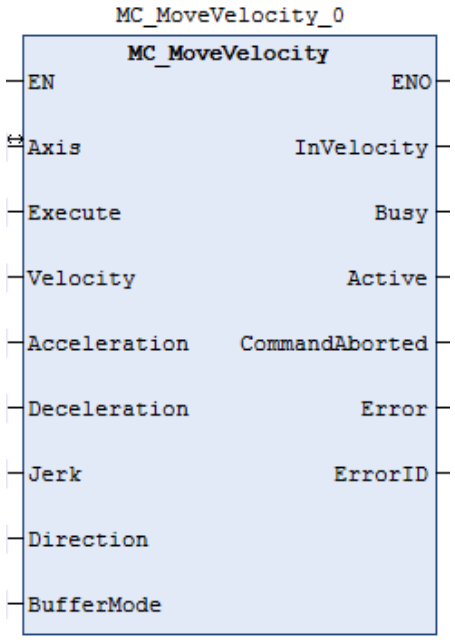


3-1-2-11. Управление скоростью [MC_MoveVelocity]

(1) Обзор команды

Этот функциональный блок движется бесконечно с заданной скоростью.

Управление скоростью [MC_MoveVelocity]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveVelocity	Команда управления скоростью		MC_MoveVelocity_0(Axis:= , Execute:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Direction:= , BufferMode:= , InVelocity=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
Velocity	Скорость	LREAL	0, положительное значение	0	Значение скорости [импульсов/с]
Acceleration	Заданная скорость ускорения	LREAL	0, положительное значение	0	Значение ускорения [импульсов/с^2]
Deceleration	Заданная скорость торможения	LREAL	0, положительное значение	0	Значение торможения [импульсов/с^2]

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Jerk	Заданная скорость в толчковом режиме	LREAL	0, положительное значение	0	Значение скорости в толчковом режиме [импульсов/с^3]
Direction	Направление	MC_DIRECTION	3: fastest 2: current 1: Positive 0: shortest -1: Negative	0	Fastest: Автоматический выбор направления перемещения с максимальной скоростью Current: Перемещение в текущем направлении Positive: прямой ход Shortest: Выбор направления перемещения по кратчайшему пути Negative: обратный ход
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Если функциональный блок имеет статус Busy (В процессе выполнения), допускается только BufferMode=Aborting.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
InVelocity	Признак достижения заданной скорости	BOOL	TRUE/FALSE	FALSE	После достижения заданной скорости эта переменная принимает состояние TRUE

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
Command Aborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

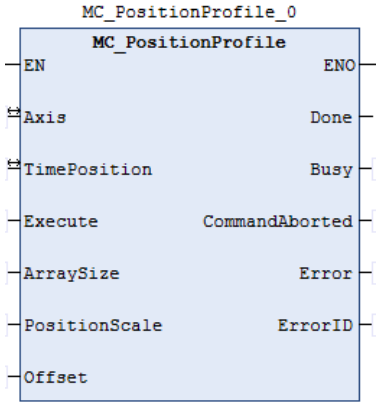
- Изменение параметра Velocity для регулирования скорости привода.
- Этот функциональный блок должен выполняться при нарастающем фронте.
- Переменная InVelocity данного функционального блока указывает на то, что рабочая скорость этой команды достигла заданного значения.
- Переменная Busy данного функционального блока указывает на то, что в данный момент выполняется текущий функциональный блок.

3-1-2-12. Положение профиля [MC_PositionProfile]

(1) Обзор команды

Пользователи могут составлять собственную таблицу данных «время - положение», а контроллер обеспечит перемещение в соответствии с введенными данными.

Положение профиля [MC_PositionProfile]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_PositionProfile	Команда управления положением профиля		MC_PositionProfile_0(Axis:= , TimePosition:= , Execute:= , ArraySize:= , PositionScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
TimePosition	Таблица данных	MC_TP_REF	-	-	Составленная пользователем таблица данных «время - положение»

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
ArraySize	Динамический массив	INT	Положительное значение	0	Количество массивов, используемых в текущем профиле
PositionScale	Комплексный коэффициент	LREAL	Диапазон данных	1	Общий коэффициент масштабирования положения
Offset	Смещение	LREAL	Диапазон данных	0	Смещение положения
VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Этот функциональный блок представляет собой модель перемещения профиля для временных периодов и положений и выполняется в режиме Discrete Motion (Дискретное перемещение) в соответствии с данными, заданными пользователем в переменной TimePosition.
- Рабочее состояние этого функционального блока — Standstill (остановка), а состояние выполнения команды - Discrete Motion (Дискретное перемещение). Другие состояния запускать нельзя.
- Эта команда запускается при нарастающем фронте Execute и повторно выполняется в режиме Discrete Motion (Дискретное перемещение).
- TimePosition — это тип данных MC_ TP_ REF.

Описание типа данных MC_TP_REF:

Элемент	Тип	Начальное значение	Описание
Number_of_pairs	INT	0	Количество сегментов в траектории перемещения профиля
IsAbsolute	BOOL	TRUE	Выбор перемещения в абсолютное положение (TRUE) и относительное положение
MC_TP_ArrayARRAY[1..N] OF SMC_TP			Массив с данными о времени и местоположении

Описание типа данных SMC_TP:

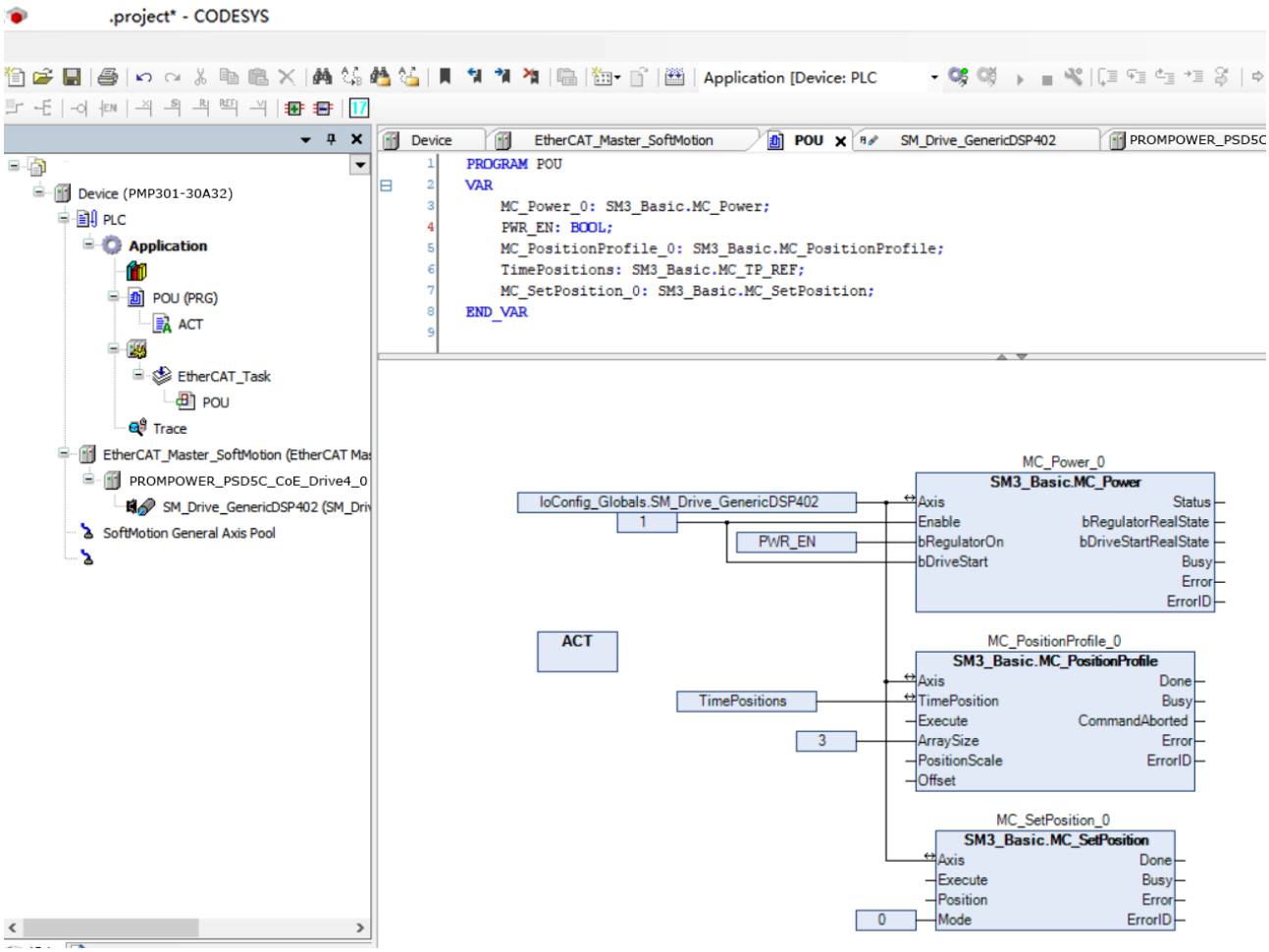
Элемент	Тип	Начальное значение	Описание
delta_time	TIME	TIME#0 мс	Время сегмента в команде на позиционирование
position	LREAL	0	Текущее значение положения

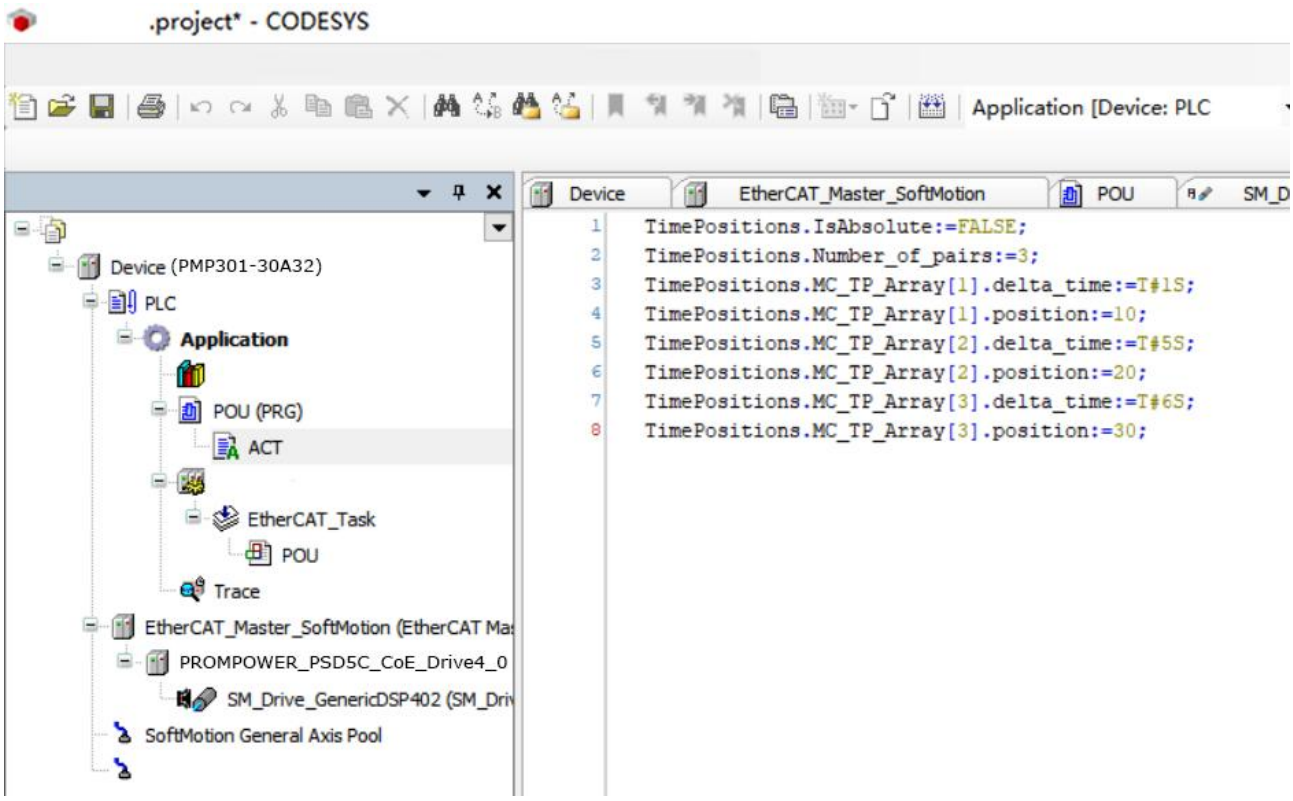
Примечание: При изменении скорости, соответствующей заданным данным о положении, выполняется регулировка согласно S-кривой.

(4) Применение

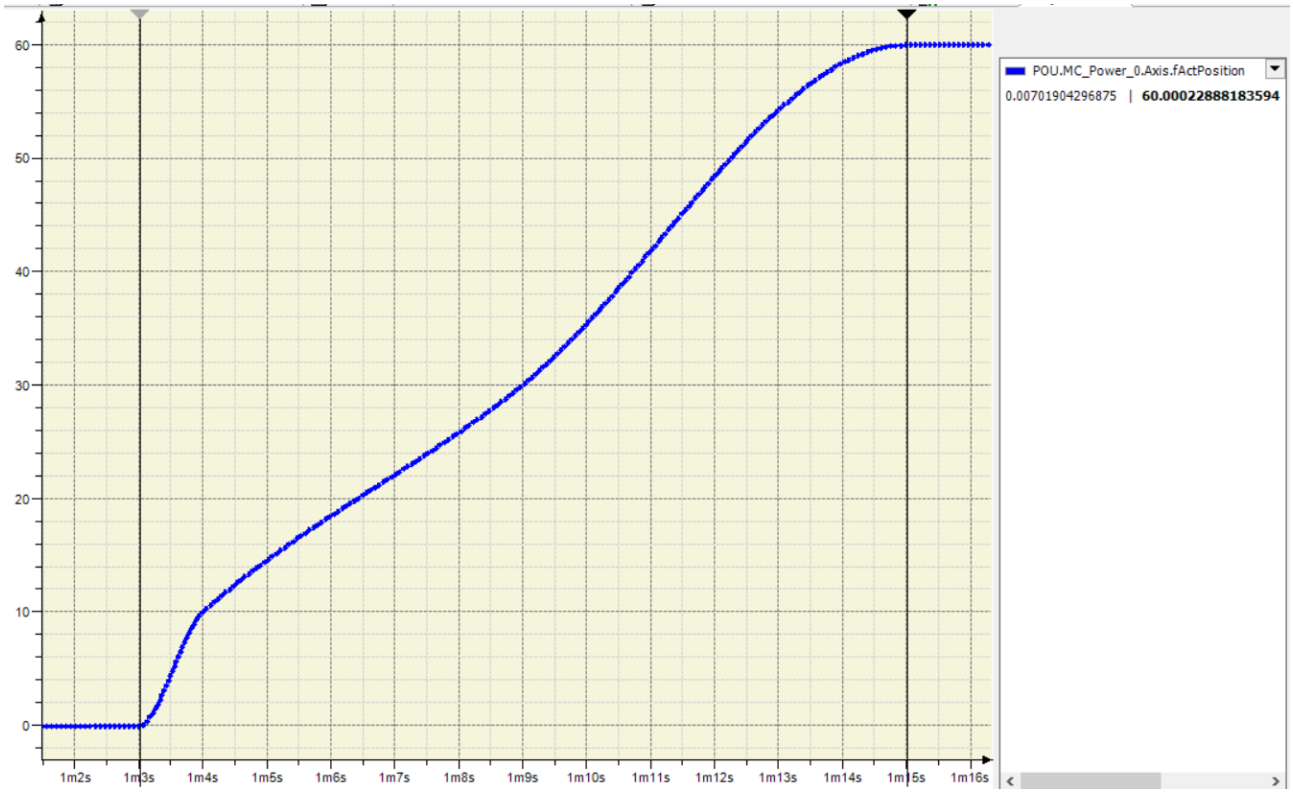
Пример 1: Настройте выполнение трех сегментов и установите для них режим относительного перемещения. Первый сегмент выполняется в положении 10 в течение 1 секунды, второй сегмент выполняется в положении 20 в течение 5 секунд, а третий сегмент выполняется в положении 30 в течение 6 секунд. Общая продолжительность этих трех сегментов составляет 12 секунд, а общее расстояние выполнения - 60.

Напишите программу на основе команд "MC_Power" и "MC_PositionProfile", запустите ее, а затем настройте и запустите три сегмента как команду MC_ PositionProfile. Перед выполнением команды установите положение 0 через команду MC_SetPosition. Настройте время и положение через пункт меню ACT (Action Properties of POU (Свойства операций POU)).





Время выполнения и местоположение можно просматривать через опцию Trace (Отслеживание).

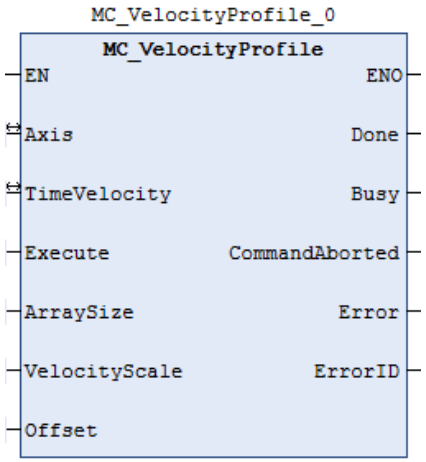


3-1-2-13. Профиль скорости [MC_VelocityProfile]

(1) Обзор команды

Аналогично команде MC_PositionProfile, команда MC_VelocityProfile планирует перемещение путем задания данных "время - скорость".

Профиль скорости [MC_VelocityProfile]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_VelocityProfile	Команда управления профилем скорости		MC_VelocityProfile_0(Axis:= , TimeVelocity:= , Execute:= , ArraySize:= , VelocityScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
TimePosition	Таблица данных	MC_TP_REF	-	-	Составленная пользователем таблица данных «время - скорость»

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
ArraySize	Динамический массив	INT	Положительное значение	0	Количество массивов, используемых в текущем профиле
VelocityScale	Коэффициент скорости	LREAL	Диапазон данных	1	Коэффициент масштабирования скорости
Offset	Смещение	LREAL	Диапазон данных	0	Смещение по скорости
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Этот функциональный блок представляет собой модель перемещения профиля для временных периодов и скоростей и выполняется в режиме Continuous Motion (Непрерывное перемещение) в соответствии с данными, заданными пользователем в переменной TimeVelocity.
- Рабочее состояние этого функционального блока — Standstill (остановка), а состояние выполнения команды - Discrete Motion (Дискретное перемещение). Другие состояния запускать нельзя.
- Эта команда запускается при нарастающем фронте Execute и повторно выполняется в режиме Discrete Motion (Дискретное перемещение).
- TimeVelocity — это тип данных MC_TV_REF:

Подробное описание типа данных MC_TP_REF:

Элемент	Тип	Начальное значение	Описание
Number_of_pairs	INT	0	Количество сегментов в траектории перемещения профиля
IsAbsolute	BOOL	TRUE	Выбор перемещения в абсолютное положение (TRUE) и относительное положение
MC_TP_ArrayARRAY[1..N] OF SMC_TP			Массив с данными о времени и местоположении

Подробное описание SMC_TP:

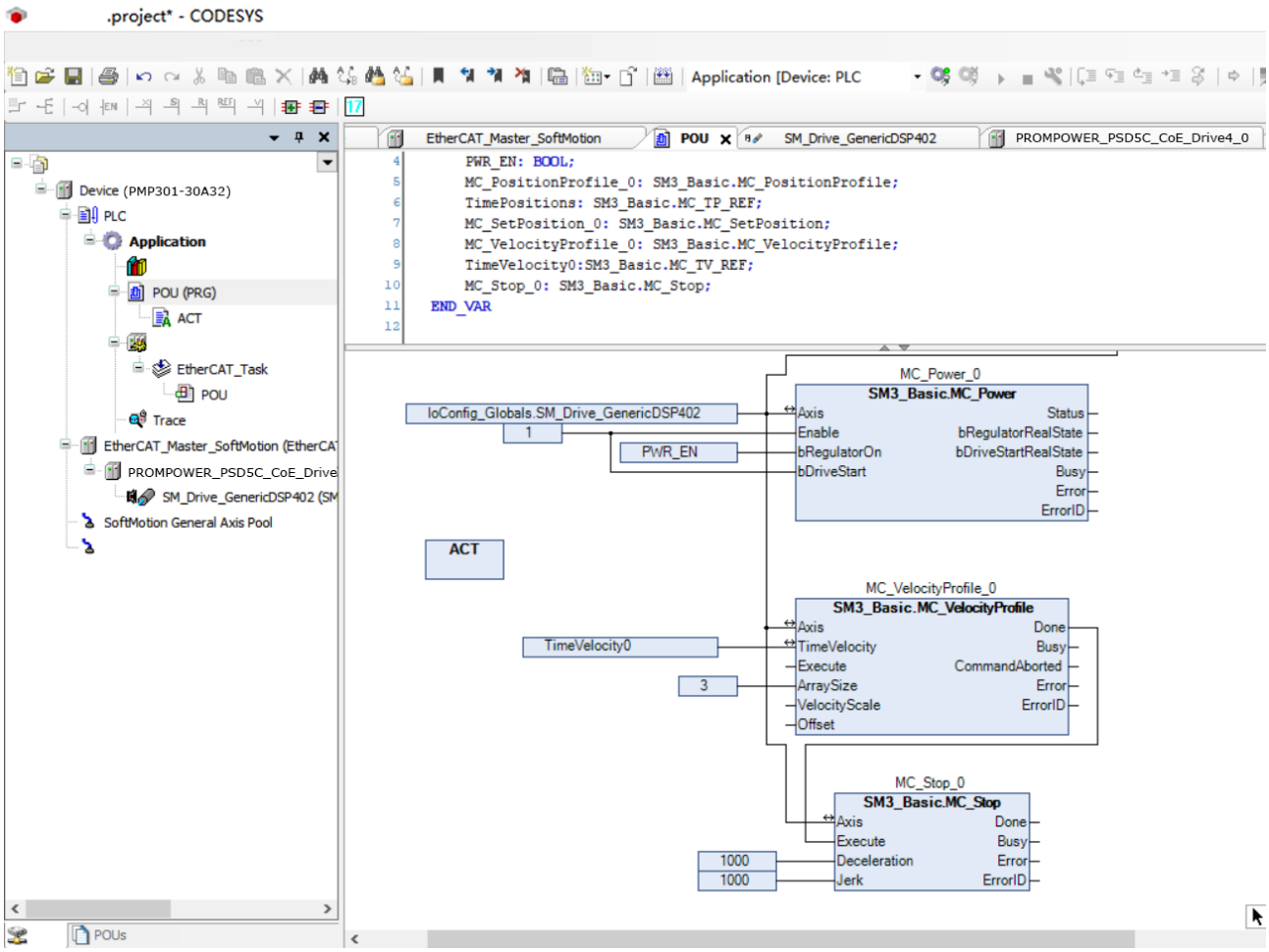
Элемент	Тип	Начальное значение	Описание
delta_time	TIME	TIME#0 мс	Время сегмента в команде на позиционирование
position	LREAL	0	Текущее значение положения

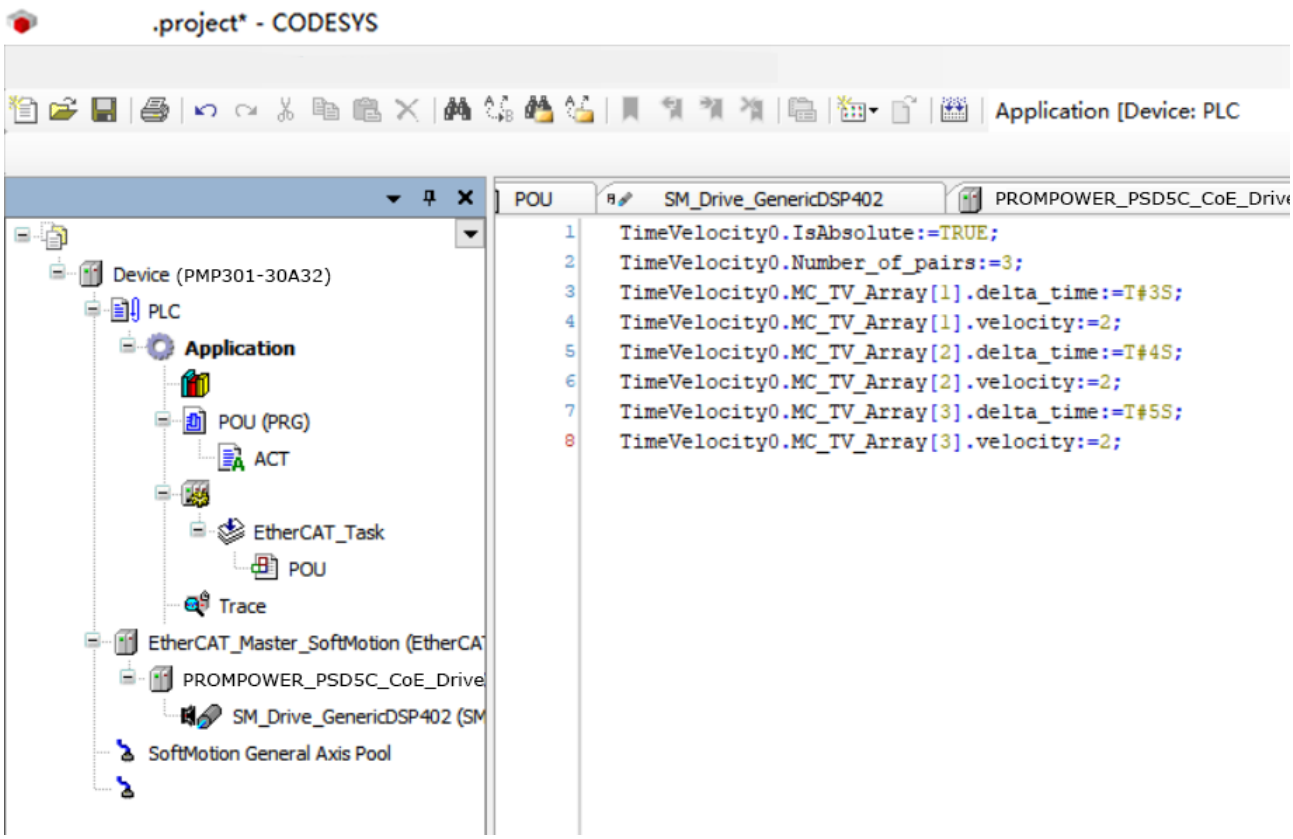
Примечание: Весь процесс управления скоростью рассчитывается по методу S-кривой ускорения и торможения, а скорость каждого сегмента профиля рассчитывается по методу наложения. При многократном выполнении команд значения скорости также суммируются, чтобы избежать превышения предела скорости во время использования команд. Повторное выполнение операции должно вернуть эту ось в состояние Standstill (Остановка).

(4) Применение

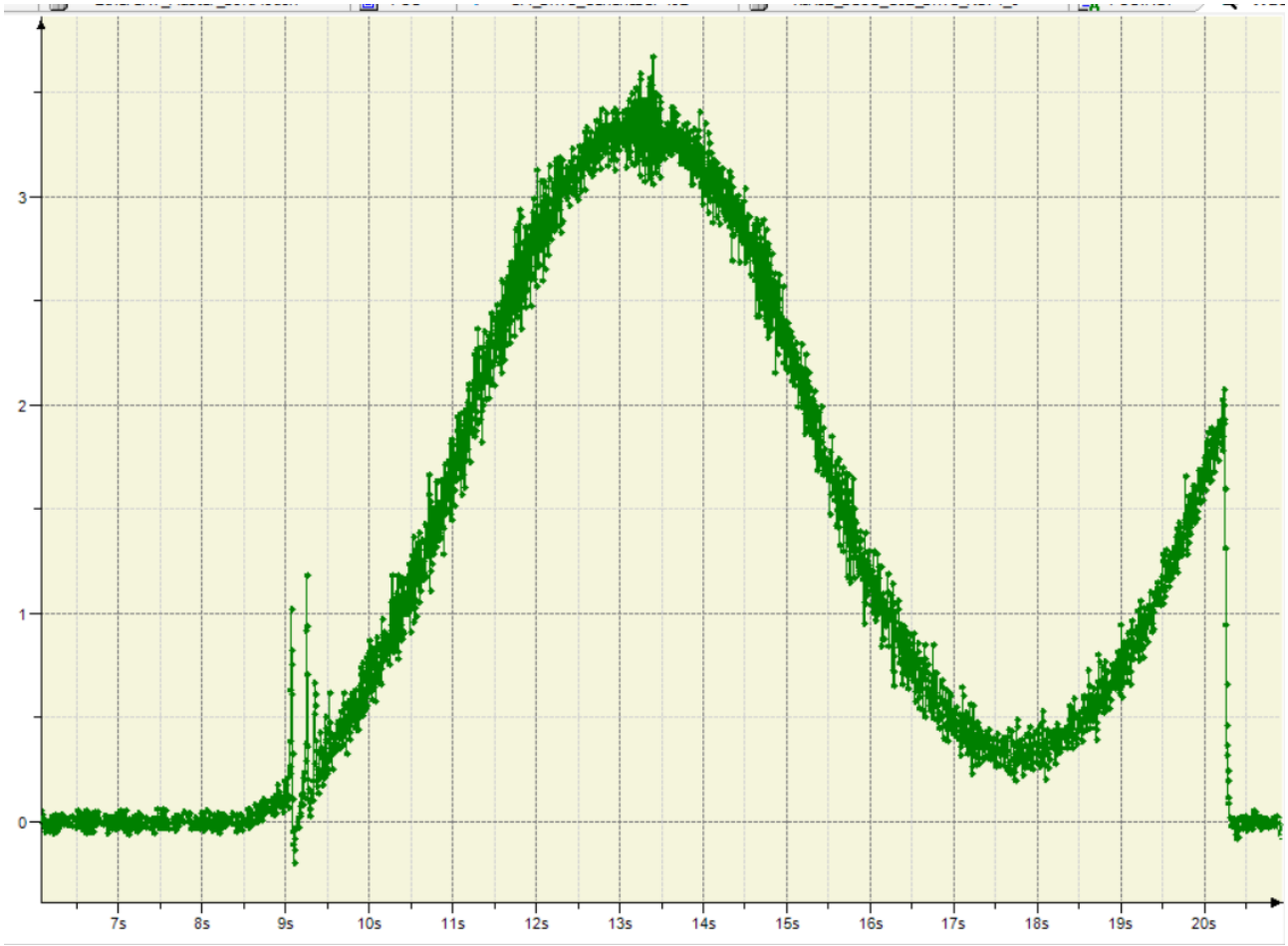
Пример 1: Настройте выполнение трех сегментов и установите для них режим абсолютного перемещения. Первый сегмент достигает скорости 2 в течение 3 секунд, второй сегмент достигает скорости 2 в течение 4 секунд, а третий сегмент достигает скорости 2 в течение 5 секунд. Общая продолжительность этих трех сегментов составляет 12 секунд, а общая скорость выполнения - 2.

Создайте программу на основе команд “MC_Power”, “MC_VelocityProfile”, “MC_Stop”. После включения запустите эти три сегмента в качестве настройки команды MC_VelocityProfile. В пункте меню ACT (POU action) задайте время и скорость.





Время и скорость можно просматривать через опцию Trace (Отслеживание).



Примечание: когда команда MC_VelocityProfile получает статус Done, скорость

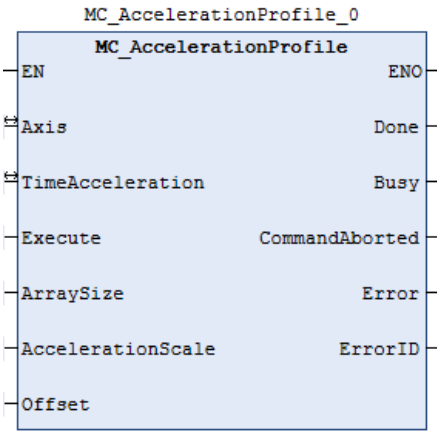
не равна 0. Поэтому нужно будет дополнительно ввести команду STOP или другие рабочие команды и команды управления позднее.

3-1-2-14. Профиль ускорения [MC_AccelerationProfile]

(1) Обзор команды

Аналогично команде MC_PositionProfile, команда MC_AccelerationProfile планирует перемещение путем задания данных "время - ускорение".

Профиль ускорения [MC_AccelerationProfile]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_AccelerationProfile	Команда управления профилем ускорения		MC_AccelerationProfile_0(Axis:= , TimeAcceleration:= , Execute:= , ArraySize:= , AccelerationScale:= , Offset:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
TimeAcceleration	Таблица данных	MC_TA_REF	-	-	Описание начального момента

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					времени/ускорения
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Запуск выполнения по нарастающему фронту
ArraySize	Динамический массив	INT	Положительное значение	0	Количество массивов, используемых в текущем профиле
AccelerationScale	Комплексный коэффициент	LREAL	Диапазон данных	1	Коэффициент масштабирования ускорения или торможения
Offset	Смещение	LREAL	Диапазон данных	0	Смещение по ускорению
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	True, если команда прервана другой командой
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Этот функциональный блок представляет собой модель перемещения профиля для временного периода и ускорения/торможения. Рабочим режимом является Discrete Motion (Дискретное перемещение), который выполняется в соответствии с данными, заданными пользователем в переменной TimeAcceleration.
- Рабочее состояние этого функционального блока — Standstill (остановка), а состояние выполнения команды - Discrete Motion (Дискретное перемещение). Другие состояния запускать нельзя.
- Эта команда запускается при нарастающем фронте Execute, а повторный запуск этой команды в режиме Discrete Motion (Дискретное перемещение) накладывается на предыдущий запуск и может легко вызвать отказ системы.
- TimeVelocity — это тип данных MC_TV_REF.

Описание типа данных MC_TP_REF:

Элемент	Тип	Начальное значение	Описание
Number_of_pairs	INT	0	Количество сегментов в траектории перемещения профиля
IsAbsolute	BOOL	TRUE	Выбор перемещения в абсолютное положение (TRUE) и относительное положение
MC_TP_ArrayARRAY[1..N] OF SMC_TP			Массив с данными о времени и местоположении

Описание типа данных SMC_TP:

Элемент	Тип	Начальное значение	Описание
delta_time	TIME	TIME#0 мс	Время сегмента в команде на позиционирование
position	LREAL	0	Текущее значение положения

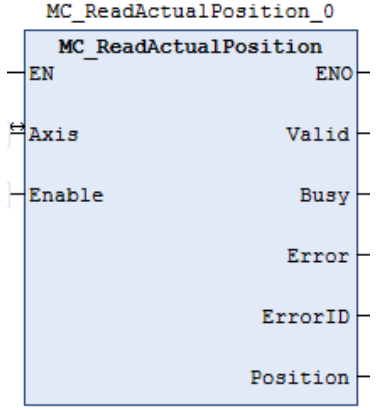
Примечание: Заданное ускорение проявляется в изменении скорости, а изменения ускорения соответствуют S-кривой. Данные об ускорении от изменения конечного результата до [начальное ускорение - A, конечное - B] (A+B)/2 отражаются в конечной скорости.

3-1-2-15. Считывание данных о фактическом положении [MC_ReadActualPosition]

(1) Обзор команды

Используется для считывания данных о текущем фактическом положении оси.

Считывание данных о фактическом положении [MC_ReadActualPosition]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_ReadActualPosition	Считывание данных о фактическом положении		MC_ReadActualPosition_0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Считывание данных о текущем положении сервопривода в состоянии true
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Position	Полученные данные о положении оси	LREAL	Диапазон данных	0	Считывание данных о положении оси из команд

(3) Описание функции

- С помощью этой команды считывается команда определения фактического положения в драйвере, что приводит к разрешению уровня Enable. Команды

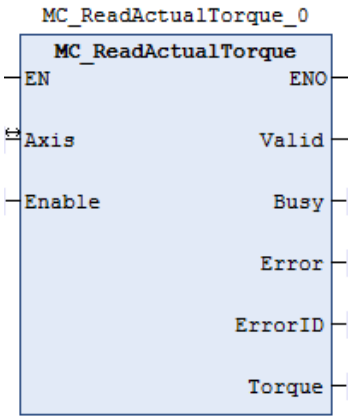
можно использовать несколько раз без воздействия друг на друга.

3-1-2-16. Считывание данных о текущем крутящем моменте [MC_ReadActualTorque]

(1) Обзор команды

Используется для считывания данных о фактическом крутящем моменте вала.

Считывание данных о текущем крутящем моменте [MC_ReadActualTorque]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_ReadActualTorque	Считывание данных о текущем крутящем моменте		MC_ReadActualTorque_0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Torque=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					блока должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Torque	Полученное текущее значение крутящего момента	LREAL	Диапазон данных	0	Данные о текущем крутящем моменте, считанные по этой команде

(3) Описание функции

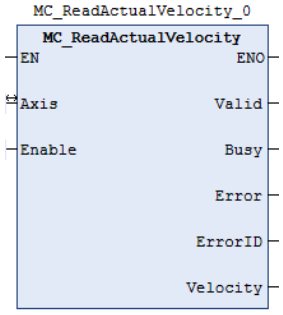
- Считывание команды текущего крутящего момента в драйвере через команду MC_ReadActualTorque, что приводит к разрешению уровня Enable. Команды можно использовать несколько раз без воздействия друг на друга.

3-1-2-17. Считывание данных о текущей скорости [MC_ReadActualVelocity]

(1) Обзор команды

Используется для считывания данных о фактической скорости оси.

Считывание данных о текущей скорости [MC_ReadActualVelocity]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_ReadActualVelocity	Считывание данных о текущей скорости		MC_ReadActualVelocity_0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Velocity	Полученное текущее значение скорости	LREAL	Диапазон данных	0	Данные о текущей скорости, считанные по этой команде

(3) Описание функции

- Считывание команды текущей скорости в драйвере через команду MC_ReadActualVelocity, что приводит к разрешению уровня Enable. Команды можно использовать несколько раз без воздействия друг на друга.

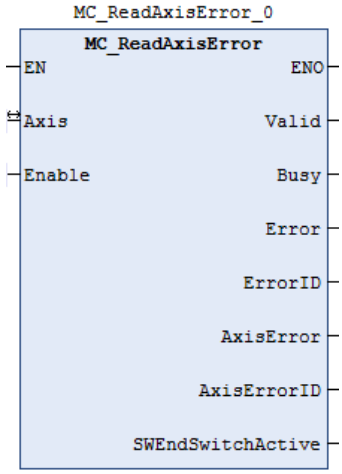
3-1-2-18. Считывание состояния ошибки оси [MC_ReadAxisError]

(1) Обзор команды

Считывание ошибки оси.

Считывание состояния ошибки оси [MC_ReadAxisError]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_AxisReadSettingsualVelocity	Считывание состояния ошибки оси		MC_ReadAxisError_0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , AxisError=> , AxisErrorID=> , SWEndSwitchActive=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE /FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно,

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционально го блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционально го блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
AxisError	Признак ошибки оси	BOOL	TRUE/FALSE	FALSE	Признак ошибки оси
AxisErrorID	Код ошибки оси	DWORD	Положительное значение, 0	0	Считывание кода ошибки оси
SWEndSwitchAc tive	Программно заданный предел	BOOL	TRUE/FALSE	FALSE	Если программно заданный предел превышен, будет состояние TRUE

(3) Описание функции

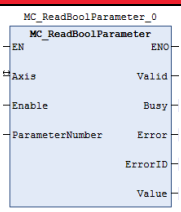
- Считывание кода ошибки в драйвере через команду MC_ReadAxisError; команда получит уровень Enable. Команды можно использовать несколько раз без воздействия друг на друга.

3-1-2-19. Считывание битового параметра оси [MC_ReadBoolParameter]

(1) Обзор команды

Считывание значения заданной переменной типа BOOL.

Считывание битового параметра оси [MC_ReadBoolParameter]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_ReadBoolParameter	Считывание битовых параметров оси		MC_ReadBoolParameter_0(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
ParameterNumbe	Номер параметра	DINT	Положительно	0	Номер

r	оси		е значение, 0		параметра
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Value	Значение параметра	BOOL	TRUE/FALSE	FALSE	Считывание значения параметра

(3) Описание функции

- Считывание состояния битовых данных в драйвере через команду MC_ReadBoolParam; команда получит уровень Enable. Команды можно использовать несколько раз без воздействия друг на друга.

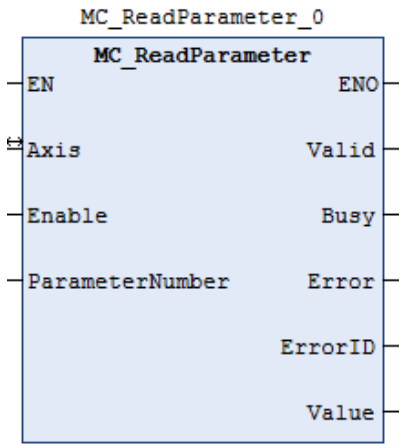
3-1-2-20. Считывание параметра оси [MC_ReadParameter]

(1) Обзор команды

Используется для считывания заданного значения параметра.

Считывание параметра оси [MC_ReadParameter]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_ReadParameter	Считывание параметра оси		MC_ReadParameter_0(Axis:= , Enable:= , ParameterNumber:= , Valid=> , Busy=> , Error=> , ErrorID=> , Value=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
ParameterNumber	Номер параметра оси	DINT	Положительное значение, 0	0	Номер параметра
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Value	Значение параметра	LREAL	Диапазон данных	0	Считывание значения параметра

(3) Описание функции

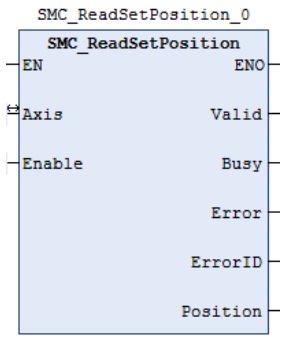
- Считывание состояния битовых данных в драйвере через команду MC_ReadParam; команда получит уровень Enable. Команды можно использовать несколько раз без воздействия друг на друга.

3-1-2-21. Считывание заданного положения оси [SMC_ReadSetPosition]

(1) Обзор команды

Этот функциональный блок можно использовать для считывания текущего заданного положения привода.

Считывание заданного положения оси [MC_ReadSetPosition]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_ReadSetPosition	Считывание заданного положения оси		SMC_ReadSetPosition_0(Axis:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока

ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Position	Положение	LREAL	Диапазон данных	0	Заданное значение положения

(3) Описание функции

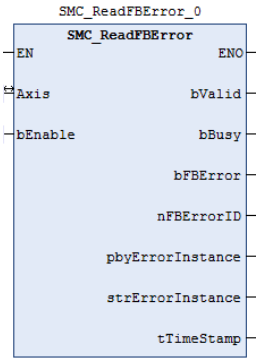
- Переменная Enable имеет состояние true, действительна при отсутствии ошибок, а выход Busy имеет состояние TRUE.
- Выходное значение переменной Position является значением Axis.fSetPosition.
- Если состояние переменной Enable изменилось на FALSE, то выходы Valid и Busy имеют состояние FALSE. Переменная Position сохраняет значение, которое было до состояния FALSE.

3-1-2-22. Считывание ошибок функционального блока [SMC_ReadFBError]

(1) Обзор команды

Считывание сообщений об ошибках функционального блока оси.

Считывание ошибок функционального блока [MC_ReadFBError]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_ReadFBError	Считывание ошибок функционального блока		SMC_ReadFBError_0(Axis:= , bEnable:= , bValid=> , bBusy=> , bFBError=> , nFBErrorID=> , pbyErrorInstance=> , strErrorInstance=> , tTimeStamp=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bEnable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bValid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
bFBError	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
nFBErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
pbyErrorInstance	Указатель на ошибки	POINTER TO BYTE	-	-	Указатель на ошибки функционального блока
strErrorInstance	Указатель на	STRING	-	-	-

	ошибки				
tTimeStamp	Временная метка	TIME	-	0	Временная метка, в которой произошла ошибка

(3) Описание функции

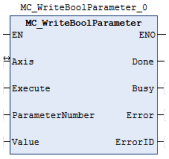
- Переменная Enable имеет состояние true, действительна при отсутствии ошибок, а выход Busy имеет состояние TRUE.
- При наличии ошибок функционального блока выход bFBError имеет состояние true.
- Если состояние переменной Enable изменилось на FALSE, то выходы Valid и Busy имеют состояние FALSE.

3-1-2-23. Установка битового параметра оси [MC_WriteBoolParameter]

(1) Обзор команды

Запись значений параметра типа BOOL.

Установка битового параметра оси [MC_WriteBoolParameter]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_WriteBoolParameter	Установка битового параметра оси		MC_WriteBoolParameter_0(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
Parameter Number	Номер параметра	DINT	Положительное число, 0	0	Идентификатор параметра
Value	Значение параметра	BOOL	TRUE/FALSE	FALSE	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если значение параметра было успешно записано
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

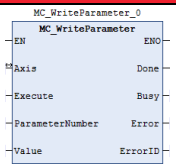
- Установка битового параметра оси через команду MC_WriteBoolParameter; команда запускается при нарастающем фронте Execute. Команды можно использовать несколько раз без воздействия друг на друга.

3-1-2-24. Установка параметра оси [MC_WriteParameter]

(1) Обзор команды

Запись заданного значения параметра.

Установка параметра оси [MC_WriteParameter]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_WriteParameter	Установка параметра оси		MC_WriteParameter_0(Axis:= , Execute:= , ParameterNumber:= , Value:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока

ParameterNumber	Номер параметра	DINT	Положительное значение, 0	0	Идентификатор параметра
Value	Значение параметра	LREAL	Диапазон данных	-	Запись значения, которое нужно установить
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если значение параметра было успешно записано
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Установка параметра оси через команду MC_WriteParameter; команда запускается при нарастающем фронте Execute. Команды можно использовать несколько раз без воздействия друг на друга.

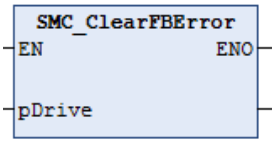
3-1-2-25. Сброс ошибок [SMC_ClearFBError]

(1) Обзор команды

Сброс информации об ошибках функционального блока из архива

Сброс ошибок [SMC_ClearFBError]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_ClearFBError	Сброс ошибок		SMC_ClearFBError(pDrive:=);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
pDrive	Указатель оси	POINTR TO AXIS_REF_SM3	-	-	Указание на ось
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
SMC_ClearFBError	Действительн.	BOOL	TRUE/FALSE	FALSE	Команда сброса ошибок имеет состояние true.

(3) Описание функции

- При возникновении ошибки на оси, после вызова функционального блока сброса для сброса оси необходимо вызвать функциональный блок для сброса информации об ошибках оси из архива. Нарастающий фронт Execute запустит выполнение этой команды.

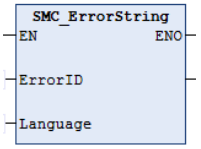
3-1-2-26. Считывание информации об ошибках [SMC_ErrorString]

(1) Обзор команды

Считывание информации с описанием ошибки, соответствующей коду ошибки.

Считывание информации об ошибках [SMC_ClearFBError]			
Условие выполнения	-	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SM3_Error.SMC_ErrorString	Считывание информации об ошибках		SMC_ErrorString(ErrorID:= , Language:=);

(2) Связанные переменные

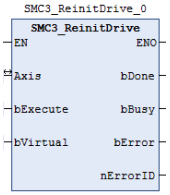
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
ErrorID	Код ошибки	SMC_ERROR	-	-	Обнаружение ошибок
Language	Язык	-	-	-	Требуемый язык
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
SMC_ErrorString	Ошибка	STRING(100)	-	-	Информация с описанием ошибки

3-1-2-27. Повторный вызов привода/оси [SMC3_ReinitDrive]

(1) Обзор команды

Этот функциональный блок служит для повторного вызова привода/оси. Это означает, что этап запуска выполняется повторно, и приложение не сможет управлять приводом до тех пор, пока функциональный блок не будет установлен в состояние bDone:=TRUE.

Повторный вызов привода/оси [SMC3_ReinitDrive]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC3_ReinitDrive	Повторный вызов привода/оси		SMC3_ReinitDrive_0(Axis:= , bExecute:= , bVirtual:= , bDone=> , bBusy=> , bError=> , nErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bExecute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
bVirtual	Язык	BOOL	TRUE/FALSE	FALSE	Вход определяет, существует ли ось на самом деле, или она смоделирована
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bDone	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональ

					ного блока еще не завершено
bError	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональ ного блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

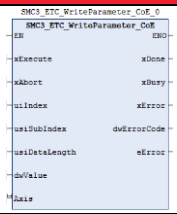
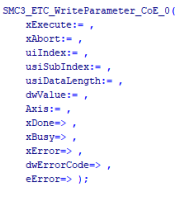
(3) Описание функции

- Используется для сброса привода и повторного подключения к контроллеру.
- Вход bvirtual определяет, существует ли ось на самом деле, или она смоделирована. Если вход bVirtual установлен в состояние TRUE, ось переводится в виртуальный режим. Затем вместо нее появятся смоделированные элементы, похожие на виртуальные приводные устройства. Это не повлияет на работу устройства с шиной Fieldbus. Устройство будет продолжать работать в обычном режиме, но не будет принимать сообщения от фактического устройства или отправлять их на него.

3-1-2-28. Настройка параметров COE для оси [SMC3_ETC_WriteParameter_CoE]

(1) Обзор команды

Настройка параметров COE для оси [SMC3_ETC_WriteParameter_CoE]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Drive_ETC		

Команда	Название	Графическое представление	Представление на языке ST
SMC3_ETC_WriteParameter_CoE	Настройка параметров COE для оси		

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
xExecute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
XAbort	Прерывание	BOOL	TRUE/FALSE	FALSE	Прерывание текущего запроса на запись
uiIndex	Индекс объекта	UINT	-	-	Индекс объекта, например, 16#6060
usiSubInde	Субиндекс объекта	USINT	-	-	Субиндекс объекта, например, 0
usiDataLength	Запись длины данных	USINT	-	-	Длина данных, записанная в байтах (1-4)
dwValue	Запись значения	DWORD	-	-	Запись значения DWORD

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
bDone	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
bError	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Применение

Изменение значения 607D:

```
VAR
  SMC3_ETC_WriteParameter_CoE_0:   SMC3_ETC_WriteParameter_CoE;
  SMC3_ETC_WriteParameter_CoE_1:   SMC3_ETC_WriteParameter_CoE;
  WRITE: BOOL;
  SMC3_ReinitDrive: SMC3_ReinitDrive;
END_VAR
```

```
SMC3_ETC_WriteParameter_CoE_0(
  xExecute: = WRITE, //запуск по нарастающему фронту
  xAbort: = ,
  uiIndex: =16#607D , //индекс объекта, например, 16#6060
  usiSubIndex: =1 , //субиндекс объекта, например, 0
  usiDataLength: =4 , // длина данных, записанная в байтах (1-4)
```

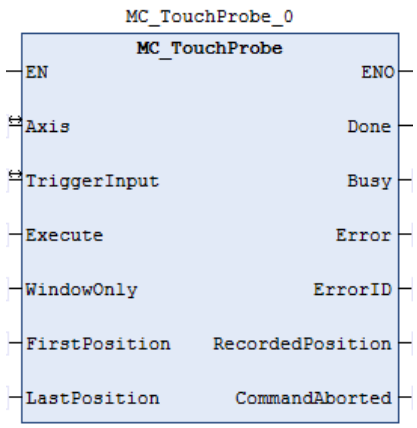
```
        dwValue: = 16#8000, //запись значения DWORD, возможность настройки
переменных DWORD
        Axis: =SM_Drive_GenericDSP402 , // SoftMotion axis
        xDone=> ,
        xBusy=> ,
        xError=> ,
          dwErrorCode=> ,
        eError=> );
      SMC3_ETC_WriteParameter_CoE_1(
        xExecute: = WRITE,
        xAbort: = ,
        uiIndex: =16#607D ,
        usiSubIndex: =2 ,
        usiDataLength: =4 ,
        dwValue: = 16#8000,
        Ось: =SM_Drive_GenericDSP402 ,
        xDone=> ,
        xBusy=> ,
        xError=> ,
        dwErrorCode=> ,
        eError=> );
```

3-1-2-29. Контактный датчик [MC_TouchProbe]

(1) Обзор команды

Этот функциональный блок используется для регистрации положения оси при возникновении запускающего события.

Контактный датчик [MC_TouchProbe]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_TouchProbe	Включение контактного датчика		MC_TouchProbe_0(Axis:= , TriggerInput:= , Execute:= , WindowOnly:= , FirstPosition:= , LastPosition:= , Done=> , Busy=> , Error=> , ErrorID=> , RecordedPosition=> , CommandAborted=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
TruggerInput	Запускающий сигнал	TRIIGGER_ REF	-	-	Опорный источник запускающего сигнала
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение

					функционального блока
WindowOnly	Окно запуска	BOOL	TRUE/FALSE	FALSE	
FirstPosition	Значение параметра	LREAL	-	0	Положение начала приема запускающих событий
LastPosition	Значение параметра	LREAL	-	0	Положение окончания приема запускающих событий
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если значение параметра было успешно записано
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
RecordedPosition	Зарегистрированное положение запуска	LREAL	-	0	Текущее положение запуска
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Текущая команда была прервана и

					приняла состояние TRUE.

(3) Описание функции

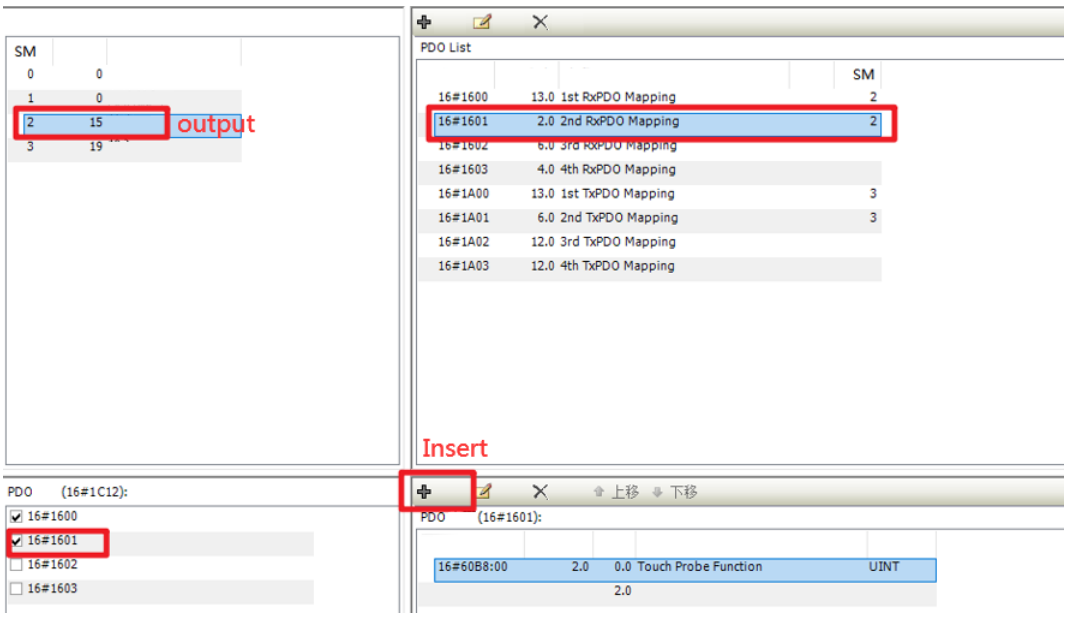
- Запись текущего положения рабочей оси при подаче сигнала TruggerInput через функциональный блок MC_TouchProbe.
- В заблокированном состоянии привод получит сигнал блокировки в положении записи и будет ждать команды контроллера.

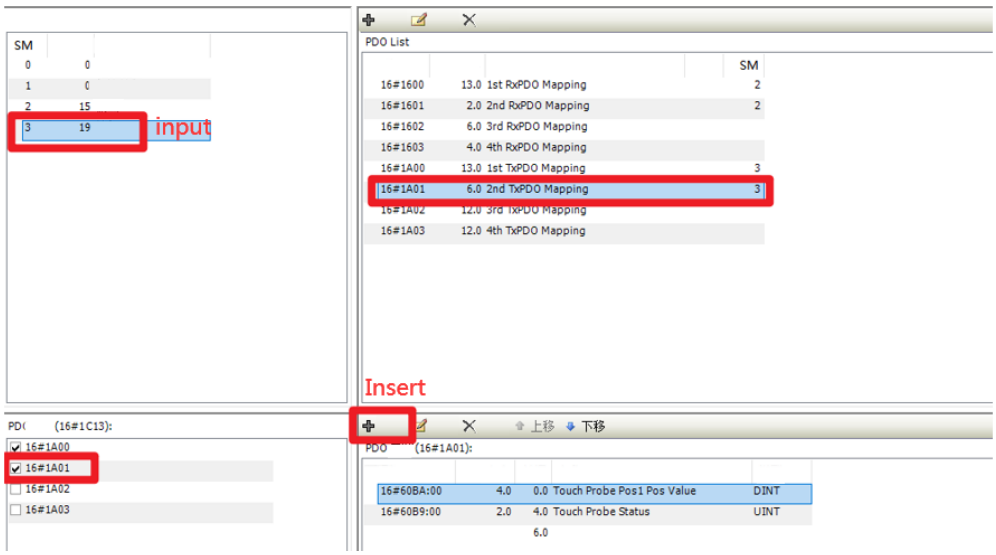
(4) Применение

Пример 1: В качестве примера рассмотрим сервопривод PROMPOWER PSD5C и выполним функцию фиксации положения контактного датчика 1 при нарастающем фронте.

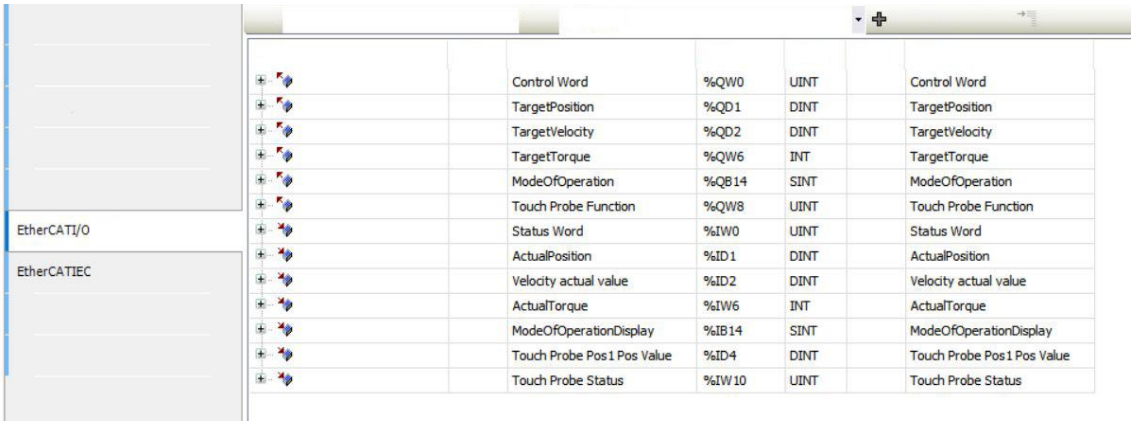
Программирование:

(1) Отметьте галочкой опцию 'Enable Expert Settings' (Включить настройки для продвинутых пользователей), выберите 1601 и 1A01 в пункте меню PDO Allocation (Распределение PDO) в Expert Process Data (Экспертные данные процесса), добавьте 60B8h в 1601, и 60BAh и 60B9h в 1a01.

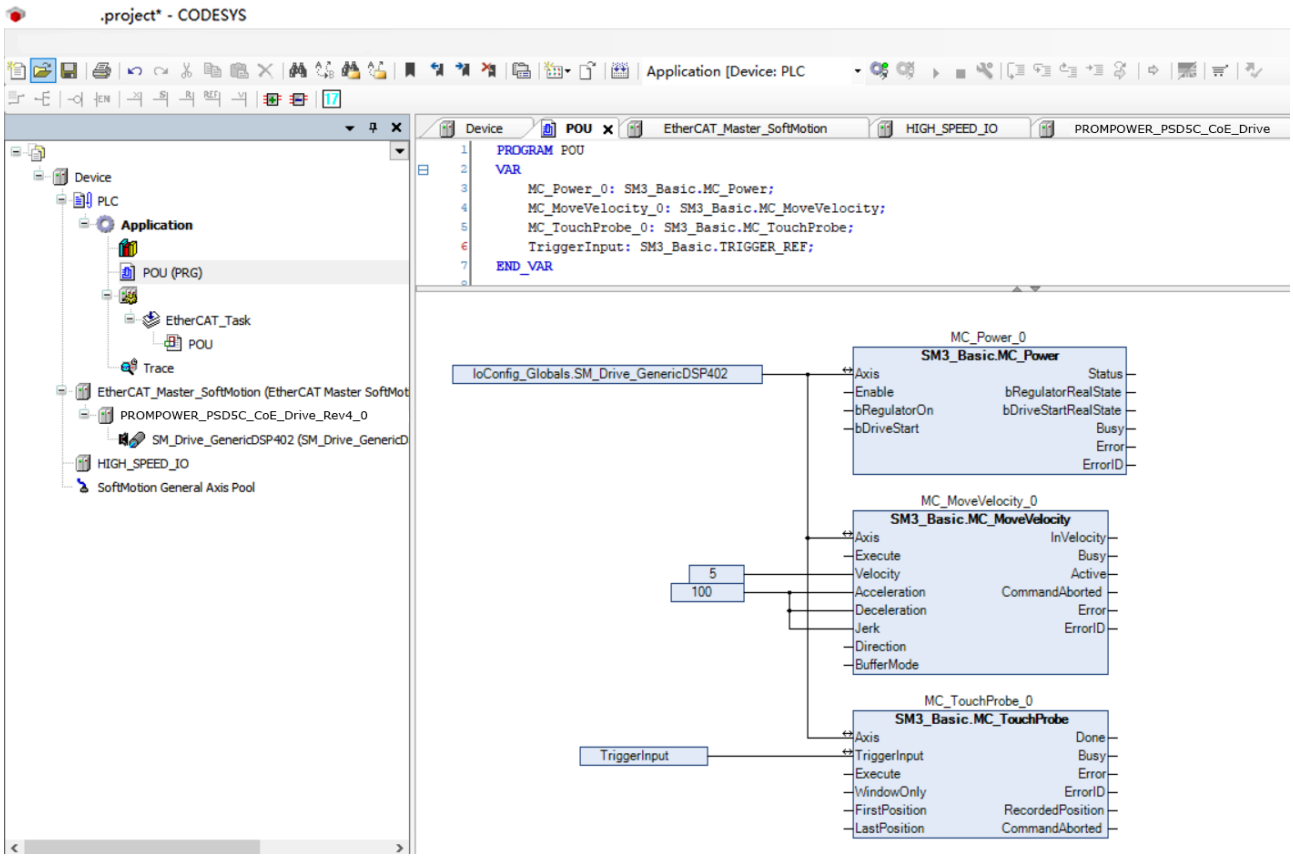




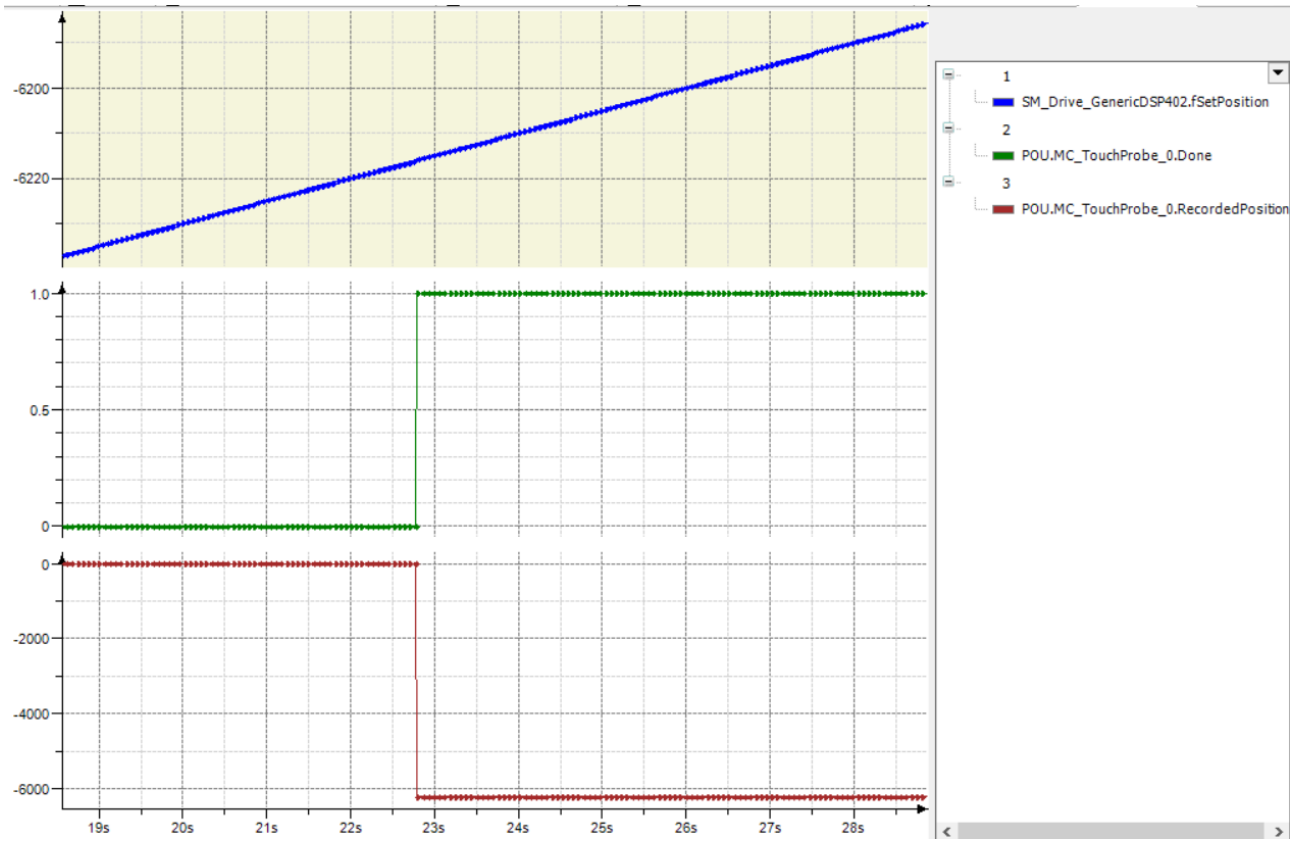
(2) Новые добавленные параметры можно просмотреть в разделе "EthercatI/O Mapping" (Отображение вводов/выводов EtherCAT).



(3) Установите bFastLatching в состояние true, bInput в состояние false, а iTriggerNumber в состояние 0 во входном контакте TriggerInput функционального блока "MC_TouchProbe". После включения функционального блока откроется нарастающий фронт сигнала контактного датчика 1. После выполнения функционального блока можно увидеть, что значение в 60B8h равно 17, что является нарастающим фронтом контактного датчика 1.



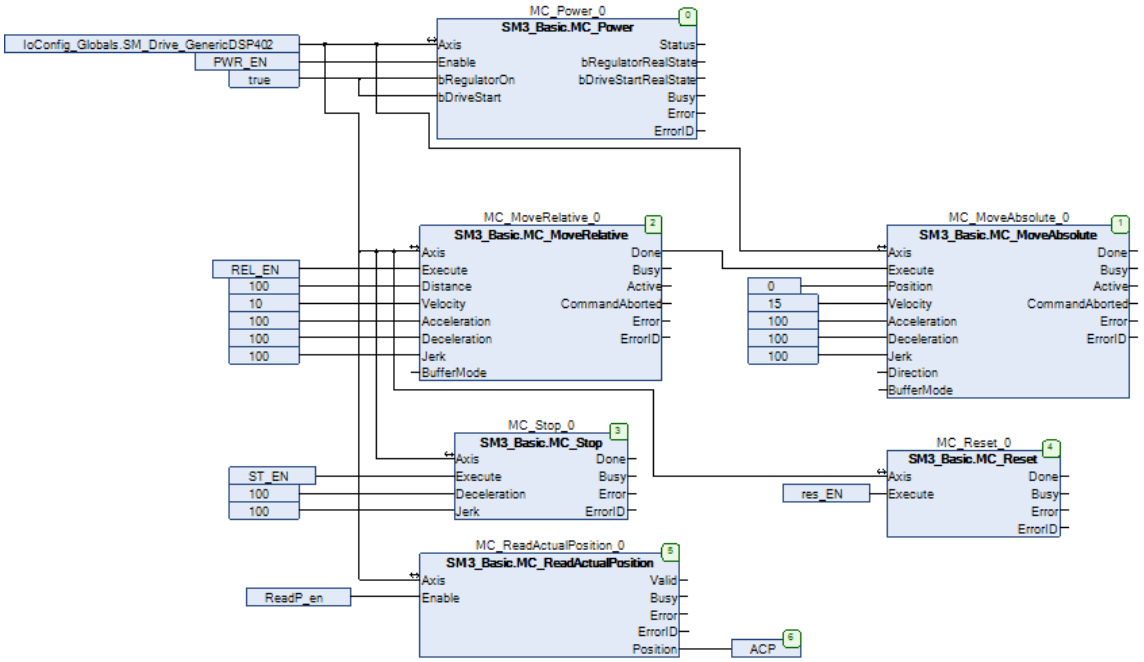
(4) В 'Trace' можно увидеть, что после выполнения команды MC_TouchProbe текущее положение будет немедленно зафиксировано.



3-1-3. Применение функции одной оси

Пример 1: Чтобы обеспечить перемещение оси 0 в двух сегментах для достижения заданного положения 0 командных единиц, первый сегмент работает со скоростью 10 импульсов/с и ускорением 100 импульсов/с^2 для относительного перемещения. Затем выполняется ускорение еще на 100 импульсов относительно начального положения. После достижения заданного относительного положения выполняется абсолютное перемещение со скоростью 15 импульсов/с и ускорением 100 импульсов/с^2. В результате этого перемещения достигается заданное положение 0 импульсов. Во время перемещения можно считывать информацию о положении в режиме реального времени, а также останавливать перемещение. При возникновении ошибки ось можно сбросить.

```
PROGRAM POU
VAR
                    define the function block
    MC_Power_0: SM3_Basic.MC_Power;
    MC_MoveRelative_0: SM3_Basic.MC_MoveRelative;
    MC_MoveAbsolute_0: SM3_Basic.MC_MoveAbsolute;
    MC_Stop_0: SM3_Basic.MC_Stop;
    MC_Reset_0: SM3_Basic.MC_Reset;
    MC_ReadActualPosition_0: SM3_Basic.MC_ReadActualPosition;
    PWR_EN: BOOL;           enable
    ST_EN: BOOL;            stop
    res_EN: BOOL;           reset
    ReadP_en: BOOL;                  read real-time position
    ACP: LREAL;                read position
END_VAR
```



3-2. Групповые функции оси

3-2-1. Групповые команды оси

Команда	Функция
MC_AddAxisToGroup	Добавление оси в группу осей
MC_RemoveAxisFromGroup	Удаление оси из группы осей
MC_UngroupAllAxes	Удаление всех осей из группы осей
MC_GroupEnable	Включение группы осей
MC_GroupDisable	Выключение группы осей
MC_GroupReset	Сброс группы осей
MC_GroupSetPosition	Настройка положения группы осей
MC_SetCoordinateTransform	Преобразование системы координат
MC_SetDynCoordTransform	Объединение двух групп осей
MC_GroupContinue	Продолжение работы группы осей
MC_GroupHalt	Остановка работы группы осей
MC_GroupInterrupt	Прерывание работы группы осей
MC_GroupStop	Останов группы осей
MC_GroupSetOverride	Изменение скорости, ускорения, активные или регулируемые операции
MC_SetKinTransform	Кинематическое преобразование системы координат
MC_MoveCircularAbsolute	Круговое перемещение в абсолютное положение
MC_MoveCircularRelative	Круговое перемещение в относительное положение
MC_MoveDirectAbsolute	Перемещение в абсолютное положение
MC_MoveDirectRelative	Перемещение в относительное положение
MC_MoveLinearAbsolute	Линейное перемещение в абсолютное положение
MC_MoveLinearRelative	Линейное перемещение в относительное положение

Команда	Функция
MC_GroupReadActualPosition	Считывание фактического положения
MC_GroupReadActualVelocity	Считывание фактической скорости
MC_GroupReadConfiguration	Считывание параметров
MC_GroupReadError	Считывание ошибок
MC_GroupReadStatus	Считывание состояния
SMC_StartupAxisGroup	Запуск группы осей
SMC_GroupPower	Включение питания группы осей
SMC_GroupInterruptAt	Настройка перерыва в перемещении
SMC_GroupEnableResumeAfterError	Возобновление перемещения после устранения ошибки оси
SMC_GroupJog	Запуск группы осей в толчковом режиме
SMC_GroupWait	Задержка в работе группы осей

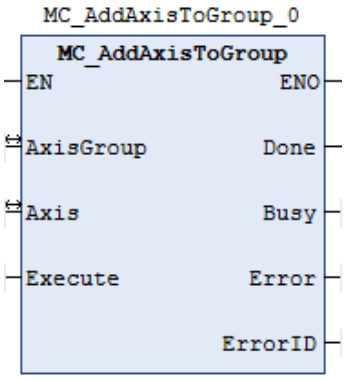
3-2-2. Групповые команды оси

3-2-2-1. Добавление оси в группу осей [MC_AddAxisToGroup]

(1) Обзор команды

Добавление оси в группу осей в командном режиме.

Добавление оси в группу осей [MC_AddAxisToGroup]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_AddAxisToGroup	Добавление оси в группу осей		MC_AddAxisToGroup_0(AxisGroup:= , Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функциональног о блока
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональног о блока еще не завершено

Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

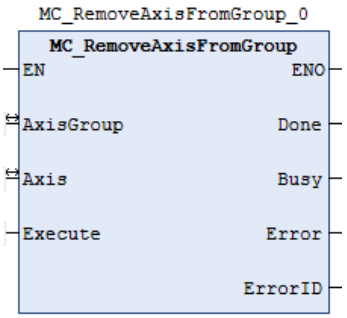
- Эта команда используется для добавления осей в выбранную группу осей и связывания их между собой в составе группы.
- Если переменная Done команды приняла состояние TRUE, это означает, что ось была успешно добавлена в группу осей. Обратите внимание, что установка переменной Execute в состояние FALSE не удаляет ось из группы осей. Если нужно удалить ось из группы осей, следует использовать команду MC_ RemoveAxisFromGroup.
- Эту команду можно выполнить только тогда, когда группа осей находится в состоянии GroupDisabled. Если эта команда выполняется после включения группы осей, будет выдано сообщение об ошибке.

3-2-2-2. Удаление оси из группы осей [MC_RemoveAxisFromGroup]

(1) Обзор команды

Удаление оси из группы осей.

Удаление оси из группы осей [MC_RemoveAxisFromGroup]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_RemoveAxisFromGroup	Удаление оси из группы осей		MC_RemoveAxisFromGroup_0(AxisGroup:= , Axis:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
Axis	Ось	AXIS_REF_SM3	-	-	Выбранная ось
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено

Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональног о блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

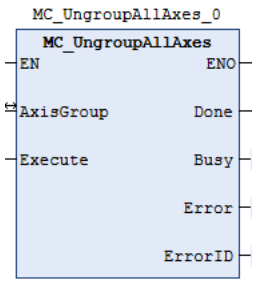
- Эту команду можно выполнить только в том случае, если группа осей выключена. Если эта команда выполняется после включения группы осей, будет выдано сообщение об ошибке.
- Если переменная Done команды приняла состояние TRUE, это означает, что ось была успешно удалена из группы осей.

3-2-2-3. Удаление всех осей из группы осей [MC_UngroupAllAxes]

(1) Обзор команды

Удаление всех осей из определенной группы осей и удаление самой группы осей.

Удаление всех осей из группы осей [MC_UngroupAllAxes]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_UngroupAllAxes	Удаление всех осей из группы осей		MC_UngroupAllAxes_0(AxisGroup:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функциональн ого блока
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональн ого блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональн ого блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

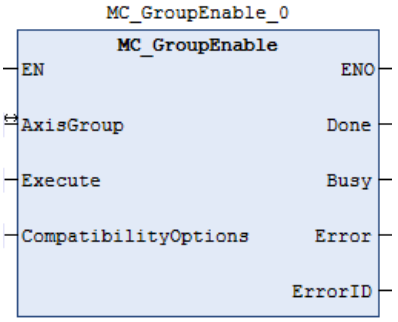
- Удаление всех осей из группы осей. Если состояние группы осей не GroupDisabled, будет сгенерирована ошибка, и ось не будет удалена из группы осей.

3-2-2-4. Включение группы осей [MC_GroupEnable]

(1) Обзор команды

Включение группы осей

Включение группы осей [MC_GroupEnable]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupEnable	Включение группы осей		MC_GroupEnable_0(AxisGroup:= , Execute:= , CompatibilityOptions:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока

CompatibilityOptions	Опция, обеспечивающая совместимость	SMC_AXIS_GROUP_COMPATIBILITY_OPTIONS	-	-	Параметры, обеспечивающие совместимость с предыдущими версиями

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Перед вызовом команды управления перемещением и других команд группы осей необходимо вызвать команду SMC_GroupPower или MC_Power, чтобы включить группу осей.
- Если переменная Done этой команды приняла состояние TRUE, это означает, что группа осей успешно перешла в состояние Standby (Ожидание).
- Типы осей, которые можно задать в группе осей: "servo axis" (Ось сервопривода) и "virtual servo axis" (Виртуальная ось сервопривода). Если заданы другие типы осей, возникнет исключение.

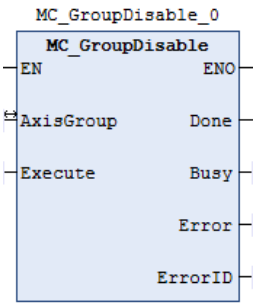
- Во время выполнения этой команды все оси в группе осей должны быть в состоянии остановки.
- Если есть оси, которые уже принадлежат к другим группам осей и уже были включены, команду MC_GroupEnable нельзя выполнить, она вызовет непредвиденную ошибку.
- Условия для отключения группы осей: выполнение команды MC_GroupDisable, переход в режим программирования для прекращения работы и пробный прогон MC.

3-2-2-5. Выключение группы осей [MC_GroupDisable]

(1) Обзор команды

Группа осей переводится в состояние Disabled (Выключено), и управлять перемещением группы осей в этом состоянии будет невозможно.

Выключение группы осей [MC_GroupDisable]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupEnable	Выключение группы осей		MC_GroupDisable_0(AxisGroup:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

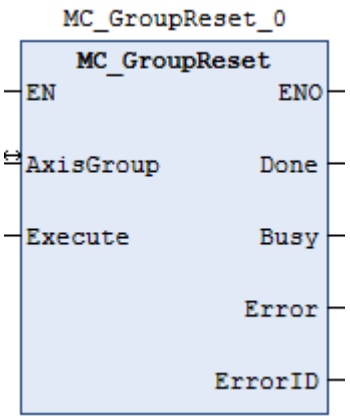
- Эта команда переводит выбранную группу осей в состояние GroupDisable.
- При изменении состояния группы осей на GroupDisable команды для выбранной группы осей (AxesGroup) будут очищены из кэша.

3-2-2-6. Сброс группы осей [MC_GroupReset]

(1) Обзор команды

Сброс ненормального состояния группы осей и оси.

Сброс группы осей [MC_GroupReset]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupReset	Сброс группы осей		MC_GroupReset_0(AxisGroup:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- По нарастающему фронту Execute обрабатываются исключения выбранной группы осей (AxesGroup) в состоянии GroupEnable и исключения оси, связанной с данной группой осей. К исключениям, которые можно устранить, относятся мелкие неисправности в оси и группе осей, неверная контрольная информация, а также сброс ошибок привода.
- Независимо от того, находится ли ось в состоянии «сервопривод ВКЛ» (servo ON) или «сервопривод ВЫКЛ» (servo OFF), может быть выполнена обработка ненормального сброса.
- Для осей с ошибками привода сначала следует выполнить сброс ошибок привода, а затем устранить исключения.
- При обработке сброса ошибок привода возможны два варианта: можно выполнить сброс ошибок привода либо оставить их без изменений в пределах параметров оси [время контроля сброса ошибок привода]. Сброс ошибок привода выполняется одновременно для всех осей, входящих в группу осей.
- Объект исключения, который может быть разрешен, — это исключение, возникающее при активации нарастающего фронта сигнала Execute. Невозможно разрешить исключения, возникшие во время разрешения других исключений.

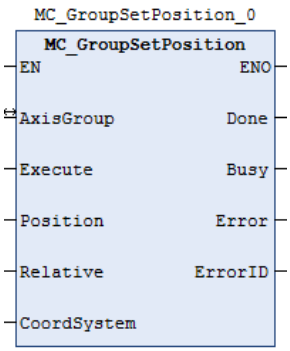
- Если команда выполняется во время неправильного торможения группы осей, ее нельзя выполнить, так как сброс не может быть выполнен до остановки осей. Кроме того, если в самой оси в составе группы осей возникнет аномальная ошибка, ее нельзя будет устранить с помощью этой команды.

3-2-2-7. Настройка положения группы осей [MC_GroupSetPosition]

(1) Обзор команды

Используется для настройки заданных положений каждой оси в группе осей.

Настройка положения группы осей [MC_GroupSetPosition]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupSetPosition	Настройка положения группы осей		MC_GroupSetPosition_0(AxisGroup:= , Execute:= , Position:= , Relative:= , CoordSystem:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока
Position	Положение	SMC_POS_REF	Диапазон данных	0	Заданное положение оси
Relative	Режим позиционирования	BOOL	TRUE/FALSE	FALSE	Режим относительного позиционирования=True, режим абсолютного позиционирования=False (по умолчанию)
CoordSystem	Применяемая система координат	SMC_COORD_SYSTEM	-	-	Применяемая система координат
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока

ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

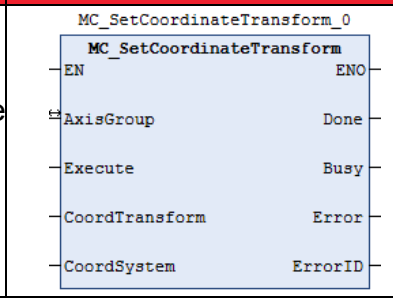
- Настройка координат в заданной для группы осей системе координат.
- Данная команда выполняется в состоянии GroupStandby группы осей и не может выполняться в динамической системе координат или выполняться одновременно с командой MC_GroupContinue.

3-2-2-8. Преобразование системы координат [MC_SetCoordinateTransform]

(1) Обзор команды

Используется для преобразования различных командных координат в эталонных системах координат.

Преобразование системы координат [MC_SetCoordinateTransform]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_SetCoordinateTransform	Преобразование системы координат		MC_SetCoordinateTransform_0(AxisGroup:= , Execute:= , CoordTransform:= , CoordSystem:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функциональн ого блока
CoordTransform	Преобразование системы координат	MC_COORD_REF	-	-	Преобразование системы координат, т.е. системы координат изделия (PCS1 или PCS2) или системы координат станка (MCS), представленной системой мировых координат (WCS)
CoordSystem	Применяемая система координат	SMC_COORD_SYSTEM	-	-	Заданная система координат, допускающая преобразование PCS_1, PCS_2 и MCS
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного

					функциональн ого блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональн ого блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

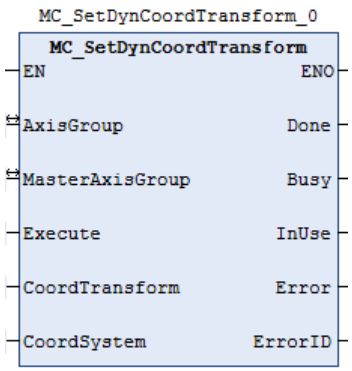
- Настройка системы мировых координат (WCS) и системы координат изделия/станка (PCS */MCS) в группе осей для преобразования системы координат.
- Если PCS является динамической системой координат (PCS перемещается относительно WCS), необходимо использовать команду MC_SetDynCoordTransform.
- Эта команда выполняет только преобразование системы координат и не зависит от команд управления перемещением.

3-2-2-9. Преобразование динамической системы координат [MC_SetDynCoordTransform]

(1) Обзор команды

Если заданная система координат перемещается относительно WCS, необходимо вызвать эту команду, чтобы выполнить преобразование системы координат.

Преобразование динамической системы координат [MC_SetDynCoordTransform]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_SetDynCoordTransform	Преобразование динамической системы координат		MC_SetDynCoordTransform_0(AxisGroup:= , MasterAxisGroup:= , Execute:= , CoordTransform:= , CoordSystem:= , Done=> , Busy=> , InUse=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
MasterAxisGroup	Группа главных осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа главных осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функциональн ого блока
CoordTransform	Преобразуем ая система координат	MC_COORD_REF	-	-	Система координат инструмента в шпиндельной группе относительно координат и направления PCS

CoordSystem	Система координат	SMC_COORD_SYSTEM	-	-	Преобразуемая система координат PCS (PCS1 или PCS2)
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональн ого блока еще не завершено
InUse	Система отсчета	BOOL	-	-	Указывает на динамическую систему координат, на которую группа осей все еще должна ссылаться
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональн ого блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

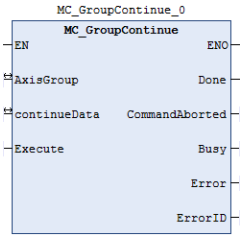
- Обычно применяется при использовании динамических PCS, например, при одновременно вызове команд MC_TrackConveyorBelt или MC_TrackRotaryTable.
- Команда SMC_SetDynCoordTransformEX имеет более универсальный интерфейс.

3-2-2-10. Продолжение работы группы осей [MC_GroupContinue]

(1) Обзор команды

Разблокировка группы осей и продолжение выполнения незавершенных команд.

Продолжение работы группы осей [MC_GroupContinue]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupContinue	Продолжение работы группы осей		MC_GroupContinue_0(AxisGroup:= , continueData:= , Execute:= , Done=> , CommandAborted=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
continueData	Продолжение передачи данных о перемещении	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Положения группы осей в момент прерывания перемещения
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональ ного блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональ ного блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

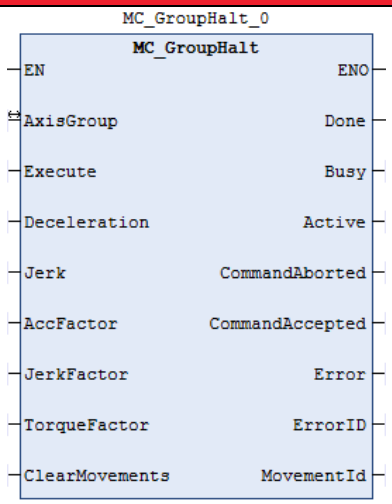
- Разблокировка группы осей и продолжение выполнения незавершенных команд.

3-2-2-11. Остановка работы группы осей [MC_GroupHalt]

(1) Обзор команды

Используется для приостановки перемещения текущей группы осей.

Остановка работы группы осей [MC_GroupHalt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupHalt	Остановка работы группы осей		MC_GroupHalt_0(AxisGroup:= , Execute:= , Deceleration:= , Jerk:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , ClearMovements:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
Deceleration	торможение	LREAL	положительное число	0	Максимальное результирующее торможение [командная единица/$с^2$]
Jerk	Скорость в толчковом режиме	LREAL	положительное число	0	Максимальная результирующая скорость в толчковом режиме [командная единица/$с^3$]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная

					скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии

CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAccepted	Команда на перемещение принята	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется или завершено

3-2-2-12. Прерывание работы группы осей [MC_GroupInterrupt]

(1) Обзор команды

Прерывание перемещения группы осей. Выполняется по команде MC_GroupContinue, при этом незавершенные команды перемещения продолжают выполняться.

Прерывание работы группы осей [MC_GroupInterrupt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupInterrupt	Прерывание работы группы осей		MC_GroupInterrupt_0(AxisGroup:= , continueData:= , Execute:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=> , mvtIdInterruptPosition=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
continueData	Продолжение передачи данных о перемещении	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Информация о перемещении при прерывании перемещения группы осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

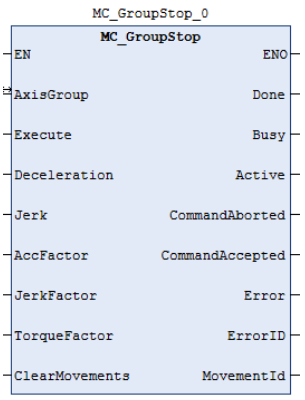
mvtIdInterruptPosition	Идентификатор положения в момент прерывания перемещения	SMC_Movement_Id	-	-	Идентификатор перемещения, соответствующий положению в момент прерывания перемещения

3-2-2-13. Останов группы осей [MC_GroupStop]

(1) Обзор команды

Останов перемещения группы осей.

Останов группы осей [MC_GroupStop]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupStop	Останов группы осей		MC_GroupStop_0(AxisGroup:= , Execute:= , Deceleration:= , Jerk:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , ClearMovements:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
Deceleration	торможение	LREAL	положительное число	0	Максимальное результирующее торможение [командная единица/$с^2$]
Jerk	Скорость в толчковом режиме	LREAL	положительное число	0	Максимальная результирующая скорость в толчковом режиме [командная единица/$с^3$]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом

					режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональн ого блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAbort ed	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAcce pted	Команда на перемещени е принята	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональн ого блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
MovementId	Признак перемещени я	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется

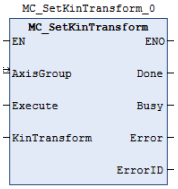
					или завершено

3-2-2-14. Кинематическое преобразование системы координат [MC_SetKinTransform]

(1) Обзор команды

Настройка кинематического преобразования группы осей из системы координат ACS в систему координат MCS.

Кинематическое преобразование системы координат [MC_SetKinTransform]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_SetKinTransform	Кинематическое преобразование системы координат		MC_SetKinTransform_0(AxisGroup:= , Execute:= , KinTransform:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Нарастающий фронт входного значения запускает выполнение функционального блока

KinTransform	Значения кинематического преобразования	TRAFO.MC_KIN_REF_ SM3	-	-	Кинематическое преобразование
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Настройка кинематических преобразований между системой координат осей (ACS) и системой координат станка (MCS).
- После выполнения этого функционального блока будет сброшена коррекция на инструмент.

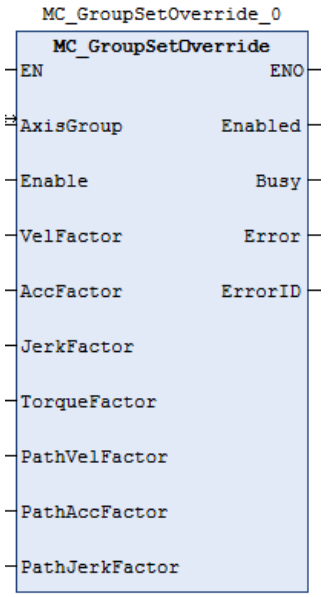
3-2-2-15. Настройка перебега группы осей [MC_GroupSetOverride]

(1) Обзор команды

Когда группа осей находится в состоянии Moving (Перемещение), можно изменять скорость ее перемещения.

Настройка перебега группы осей [MC_GroupSetOverride]			
Условие выполнения	Нормальное включение/выключение	Совместимые модели	PMP3xx

	триггера		
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupSetOverride	Настройка перебега группы осей		MC_GroupSetOverride_0(AxisGroup:= , Enable:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , PathVelFactor:= , PathAccFactor:= , PathJerkFactor:= , Enabled=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот

					коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
PathVelFactor	Коэффициент результирующей скорости	LREAL	0-1	1	Коэффициент результирующей скорости; максимальная результирующая скорость по всей траектории перемещения группы осей, умноженная на

					этот коэффициент скорости; значение находится в диапазоне [0,1]
PathAccFactor	Коэффициент результирующего ускорения	LREAL	0-1	1	Коэффициент результирующего ускорения; максимальное результирующее ускорение по всей траектории перемещения группы осей, умноженное на этот коэффициент результирующего ускорения; значение находится в диапазоне [0,1]
PathJerkFactor	Коэффициент результирующей скорости в толчковом режиме	LREAL	0-1	1	Коэффициент результирующей скорости в толчковом режиме; максимальная результирующая скорость в толчковом режиме по всей траектории перемещения группы осей, умноженная на этот коэффициент результирующей скорости в толчковом режиме;

					значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enabled	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE, если настройка коэффициента перебега завершена
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

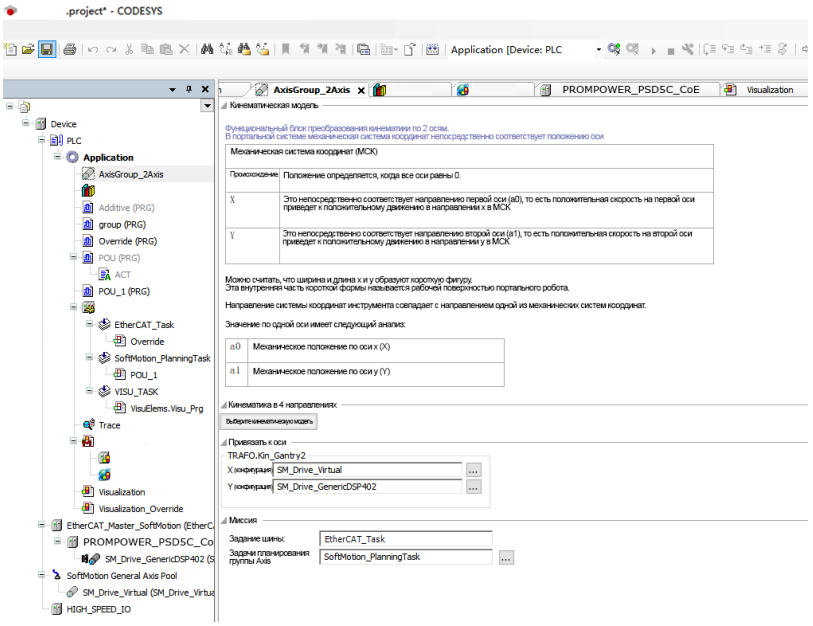
(3) Описание функции

- Группа осей может изменять свою скорость, ускорение и ускорение при перемещении, если выполнить эту команду в состоянии Moving (Перемещение).
- Заданный коэффициент перебега сохранится, пока не будет задано новое значение. Например, для группы осей в режиме перемещения CP (даже после перемещения) установка параметра VelFactor или PathVelFactor на 0 приведет к внезапной остановке перемещения по заданной траектории. Если команда MC_GroupStop в текущий момент находится в состоянии Active, появится сообщение об ошибке.
- На команду MC_GroupStop коэффициент перебега не влияет.
- Повторно включите группу осей, и коэффициент перебега сохранится.
- По умолчанию коэффициент перебега равен 1.

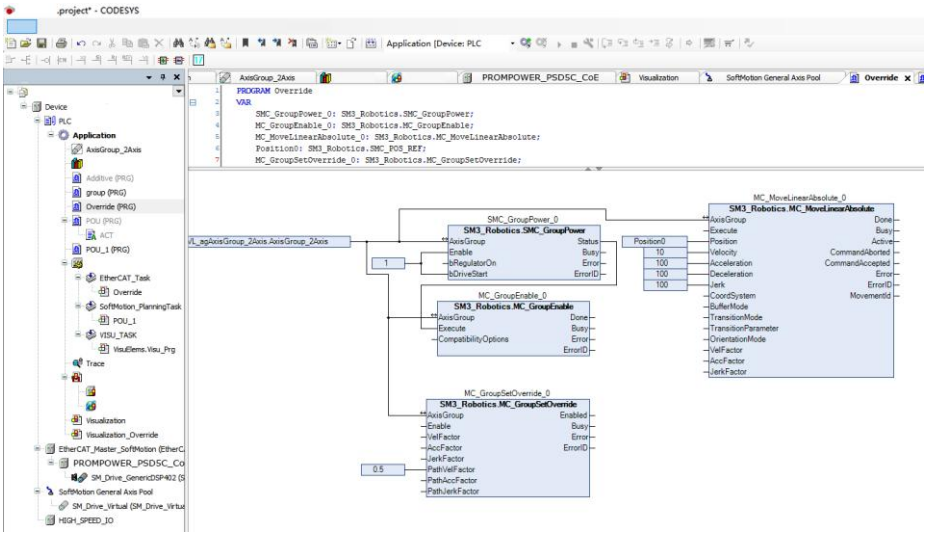
- Уменьшение значения переменной AccFactor или JerkFactor может вызвать перерегулирование по положению (перебег) и привести к повреждению оборудования.

(4) Применение

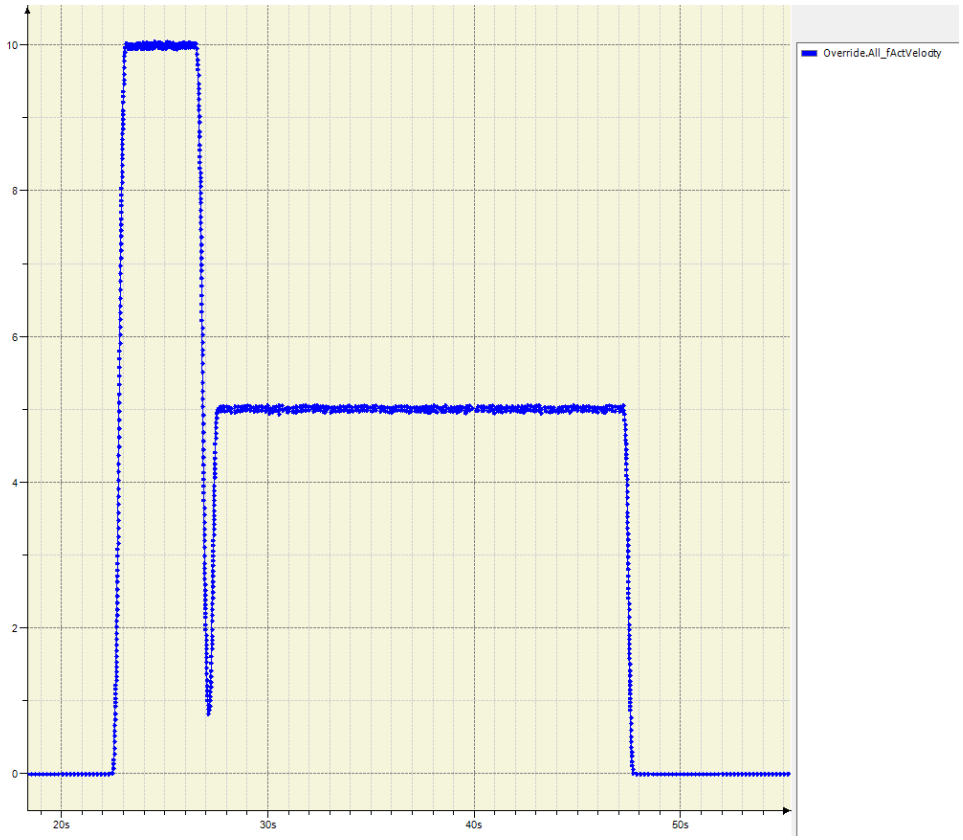
Пример 1: Выберите кинематическую модель для двух осей станка с подвижным порталом, выберите виртуальную ось для оси X, после чего выберите действительную ось Ethercat для оси Y. Во время выполнения абсолютного линейного интерполяционного перемещения по команде "MC-MoveLinearAbsolute" используйте команду настройки перебега группы осей "MC_GroupSetOverride" для изменения скорости группы осей.



Программирование: Установите текущие положения обеих осей на 0, настройте заданное положение (100, 100), выполните абсолютное линейное интерполяционное перемещение из начальной точки (0,0) и установите скорость группы осей на 10 импульсов/с. С помощью команды настройки перебега группы осей измените скорость группы осей на 5 импульсов/с.



Через пункт меню Trace можно контролировать текущую скорость группы осей. После выполнения команды настройки перебега группы осей скорость изменится с 10 импульсов/с на 5 импульсов/с.

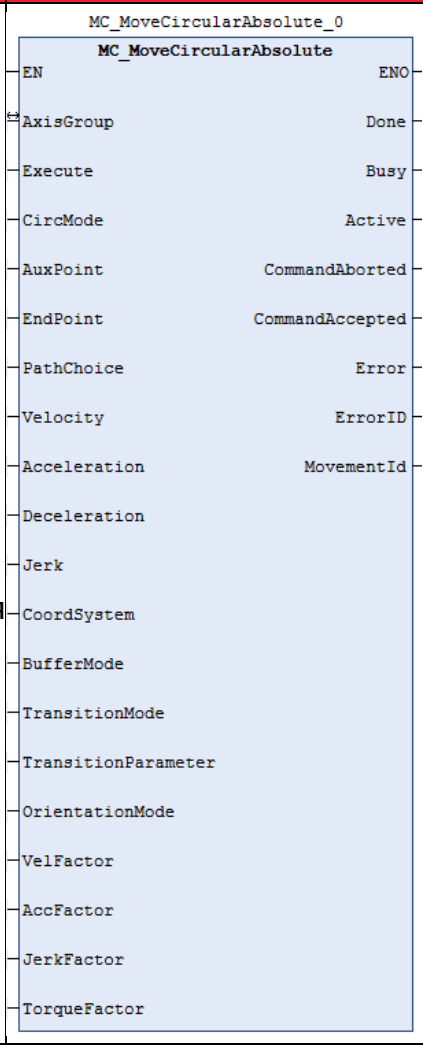


3-2-2-16. Дуговая интерполяция в режиме абсолютного позиционирования [MC_MoveCircularAbsolute]

(1) Обзор команды

Управление группой осей для выполнения перемещения с дуговой интерполяцией в режиме абсолютного позиционирования.

Дуговая интерполяция в режиме абсолютного позиционирования [MC_MoveCircularAbsolute]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveCircularAbsolute	Дуговая интерполяция в режиме абсолютного позиционирования		MC_MoveCircularAbsolute_0(AxisGroup:= , Execute:= , CircMode:= , AuxPoint:= , EndPoint:= , PathChoice:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
CircMode	Режим дуговой интерполяции	SMC_CIRC_MODE	Border/Center/Radius	-	Метод дуговой интерполяции: Border: Трехточечная дуга

					Center: Центральная дуга Radius: Радиусная дуга
AuxPoint	Дополнительная точка	SMC_POS_REF	-	-	Ввод дополнительных точек в системе координат. См. CircMode
EndPoint	Конечная точка	SMC_POS_REF	-	-	Ввод конечного положения в системе координат
PathChoice	направление	MC_CIRC_PATHCHOICE	CLOCKWISE/ COUNTER_CLOCKWISE	-	Направление перемещения: CLOCKWISE: по часовой стрелке COUNTER_CLOCKWISE: против часовой стрелки
Velocity	скорость	LREAL	0, положительное число	0	Максимальная результирующая скорость [командная единица/с]
Deceleration	торможение	LREAL	положительное число	0	Максимальное результирующее торможение [командная единица/$с^2$]
Jerk	Скорость в толчковом режиме	LREAL	положительное число	0	Максимальная результирующая скорость в толчковом режиме [командная единица/$с^3$]

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CoordSystem	Эталонная система координат	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	Эталонная система координат
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Операция при задании нескольких команд на начало перемещения
TransitionMode	Режим углового перехода	MC_TRANSITION_MODE	TMNone/ TMStartVelocity/ TMCornerDistance	-	TMNone: Без смешивания TMStartVelocity: Смешивание на основе скорости TMCornerDistance: Смешивание на основе расстояния
TransitionParameter	Параметры углового перехода	массив [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, положительное число	0	Параметры углового перехода
OrientationMode	Режим позиционирования с интерполяцией	SMC_ORIENTATION_MODE	GreatCircle/ Ось	-	GreatCircle: Перемещение по кратчайшему пути из начального положения в заданное положение. В этом режиме, даже если начальное положение и конечное

					положение находятся в пределах указанной области, фактическая траектория перемещения все равно может выйти за пределы данной области. Axis: Ось позиционирования перемещается в пределах указанной области от начального положения до конечного положения, и в этом режиме поддерживаются не все кинематические преобразования.
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]

AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено

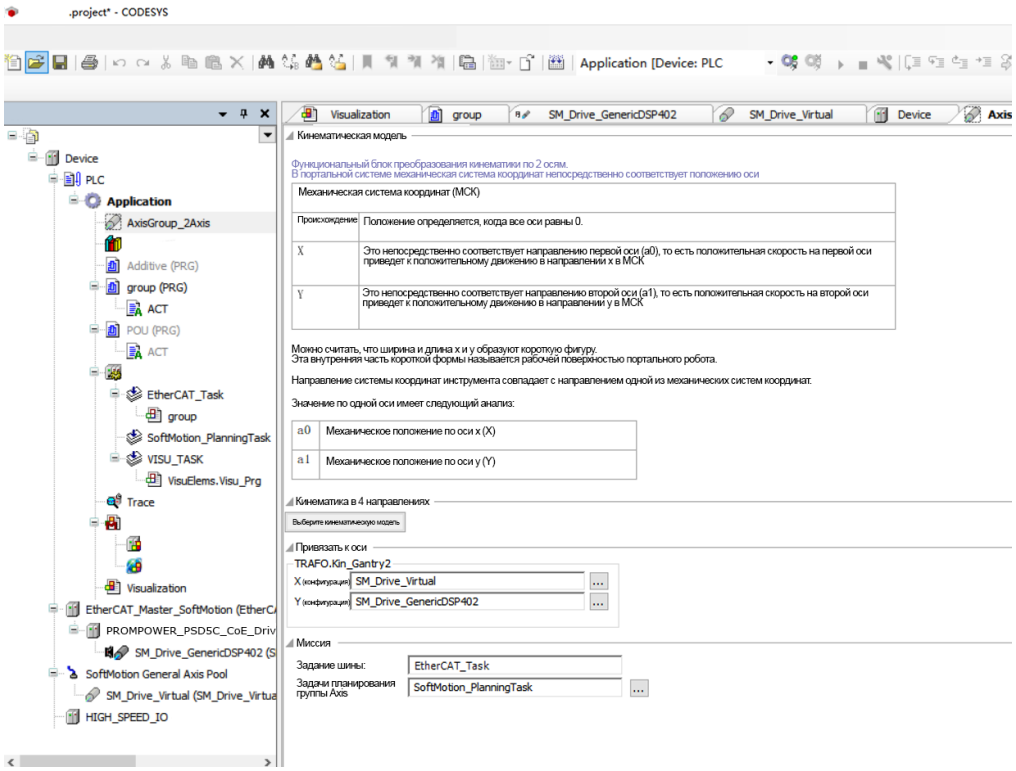
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAccepted	Прием команды на перемещение	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	Действительно, если CommandAccepted или Done имеет состояние TRUE

(3) Описание функции

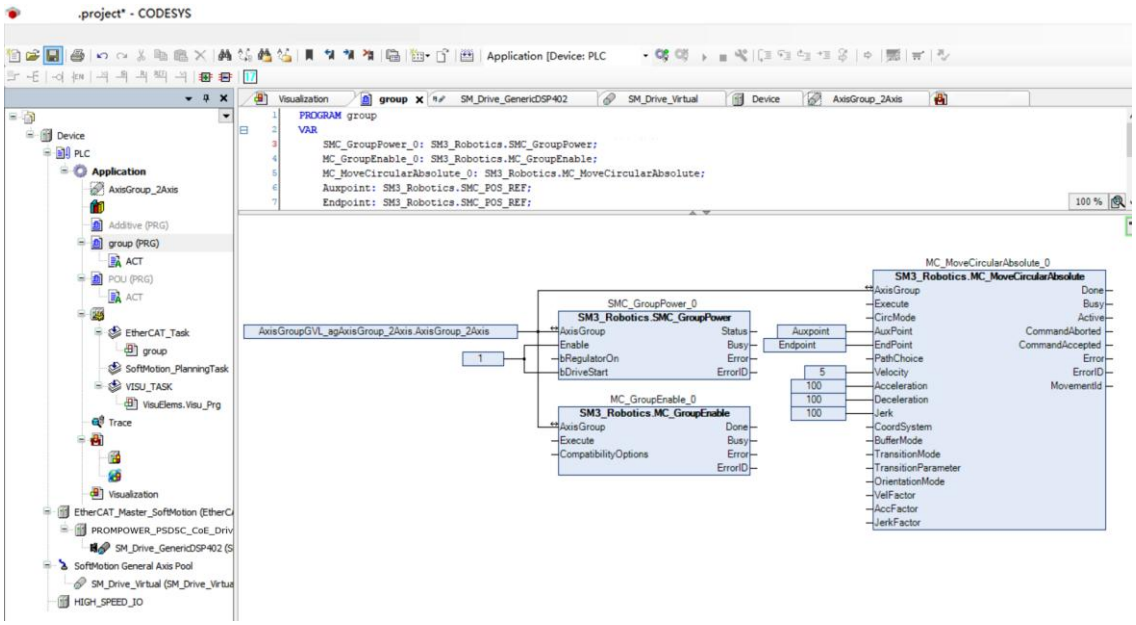
- Нарастающий фронт сигнала Execute запускает заданное перемещение, а спадающий фронт не влияет на заданное перемещение.
- В режиме интерполяции по центральной дуге введите значения параметров AuxPoint [1] и AuxPoint [2] как расстояние между центром окружности и начальной точкой дуги. В режиме интерполяции по радиусной дуге AuxPoint [1] является значением радиуса, а параметр AuxPoint [2] недействителен.
- Числовые значения переменных EndPoint [1] - EndPoint [8] представляют собой координаты конечных точек каждой оси.

(4) Применение

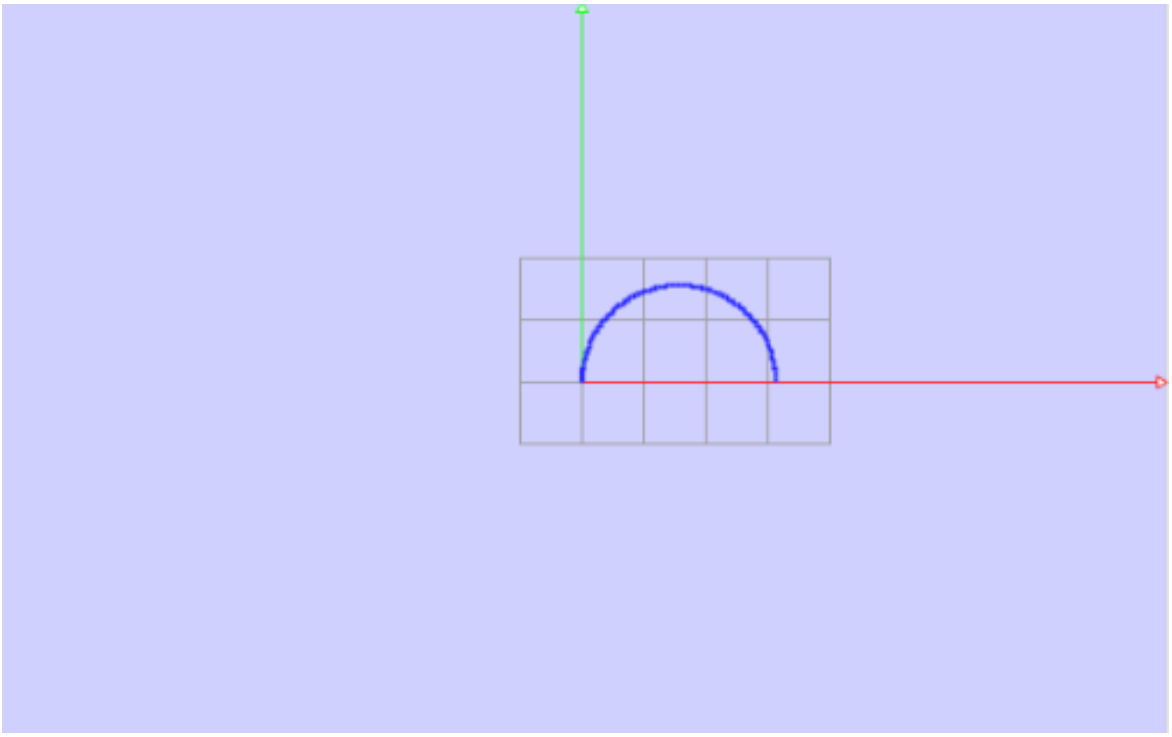
Пример 1: Чтобы выполнить дуговую интерполяцию в режиме абсолютного позиционирования, выберите кинематическую модель для двух осей станка с подвижным порталом, выберите мнимую ось для оси X, после чего выберите действительную ось Ethercat для оси Y.



Программирование: Установите текущие положения обеих осей на 0, выберите режим задания дуги по трем точкам, установите AuxPoint на (50, 50), EndPoint на (100, 0), установите скорость обеих осей на 5, и начертите дугу в начале координат (0, 0).



На рисунке ниже показана траектория дуги.

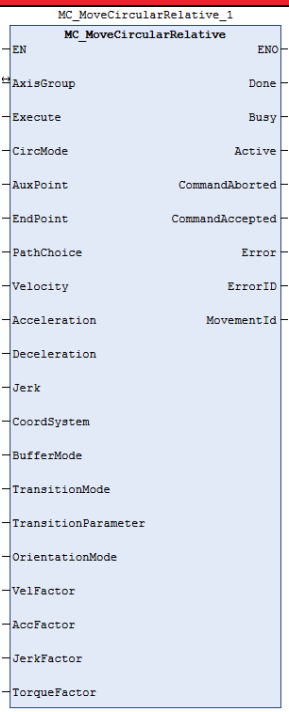


3-2-2-17. Дуговая интерполяция в режиме относительного позиционирования [MC_MoveCircularRelative]

(1) Обзор команды

Управление группой осей для выполнения перемещения с круговой (дуговой) интерполяцией в режиме относительного позиционирования.

Дуговая интерполяция в режиме относительного позиционирования [MC_MoveCircularRelative]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveCircularRelative	Дуговая интерполяция в режиме относительного позиционирования		MC_MoveCircularRelative_0(AxisGroup:= , Execute:= , CircMode:= , AuxPoint:= , EndPoint:= , PathChoice:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
CircMode	Режим дуговой интерполяции	SMC_CIRC_MODE	Border/Center/Radius	-	Метод дуговой интерполяции: Border: Трехточечная дуга Center: Центральная дуга Radius: Радиусная дуга

AuxPoint	Дополнительная точка	SMC_POS_REF	-	-	Ввод дополнительных точек в системе координат. См. CircMode
EndPoint	Конечная точка	SMC_POS_REF	-	-	Ввод конечного положения в системе координат
PathChoice	направление	MC_CIRC_PATHCHOICE	CLOCKWISE/ COUNTER_CLOCKWISE	-	Направление перемещения: CLOCKWISE: по часовой стрелке COUNTER_CLOCKWISE: против часовой стрелки
Velocity	скорость	LREAL	0, положительное число	0	Максимальная результирующая скорость [командная единица/с]
Deceleration	торможение	LREAL	положительное число	0	Максимальное результирующее торможение [командная единица/$с^2$]
Jerk	Скорость в толчковом режиме	LREAL	положительное число	0	Максимальная результирующая скорость в толчковом режиме [командная единица/$с^3$]
CoordSystem	Эталонная система координат	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	Эталонная система координат

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Операция при задании нескольких команд на начало перемещения
TransitionMode	Режим углового перехода	MC_TRANSITION_MODE	TMNone/ TMStartVelocity/ TMCornerDistance	-	TMNone: без смешивания TMStartVelocity: смешивание на основе скорости TMCornerDistance: смешивание на основе расстояния
TransitionParameter	Параметры углового перехода	массив [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, положительное число	0	Параметры углового перехода
OrientationMode	Режим позиционирования с интерполяцией	SMC_ORIENTATION_MODE	GreatCircle/ Ось	-	GreatCircle: Перемещение по кратчайшему пути из начального положения в заданное положение. В этом режиме, даже если начальное положение и конечное положение находятся в пределах указанной области, фактическая траектория перемещения

					все равно может выйти за пределы данной области. Axis: Ось позиционирования перемещается в пределах указанной области от начального положения до конечного положения, и в этом режиме поддерживаются не все кинематические преобразования.
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости в

	в толчковом режиме				толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAccepted	Прием команды на перемещение	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока

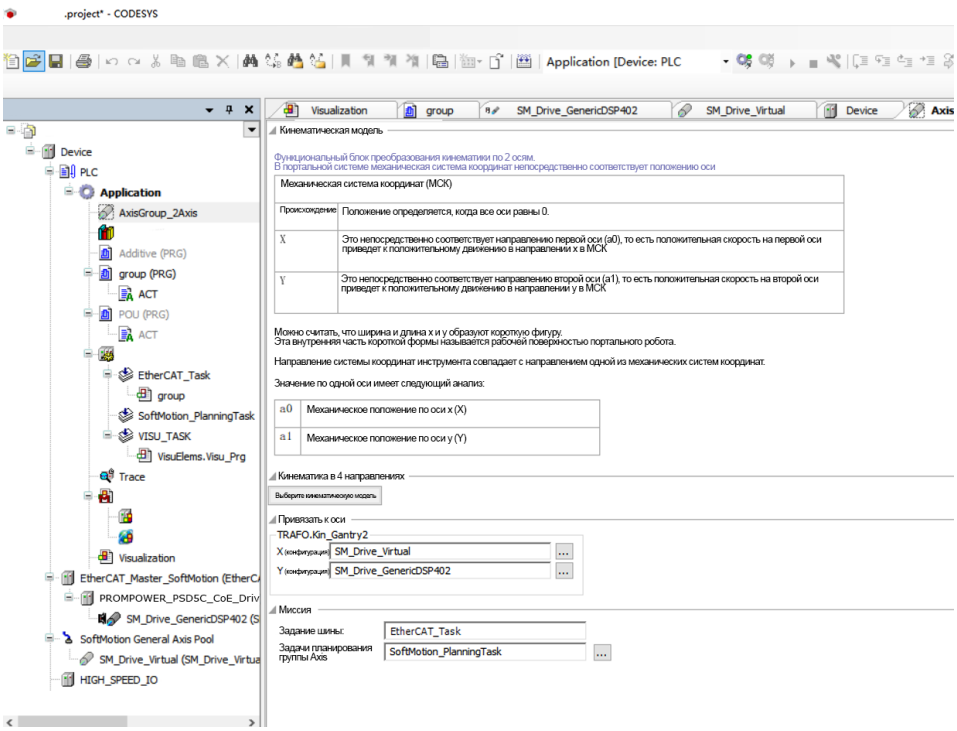
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	Действительно, если CommandAccepted или Done имеет состояние TRUE

(3) Описание функции

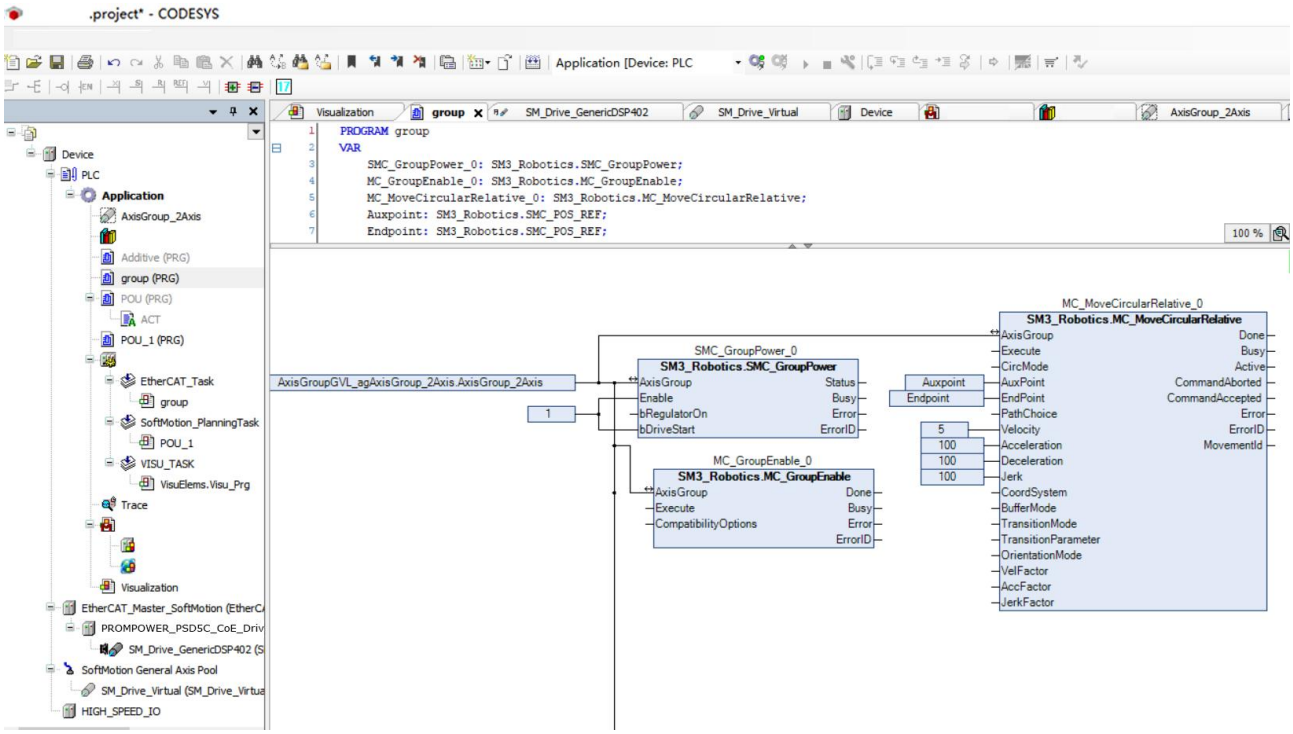
- Нарастающий фронт сигнала Execute запускает заданное перемещение, а спадающий фронт не влияет на заданное перемещение.
- В режиме интерполяции по центральной дуге введите значения параметров AuxPoint [1] и AuxPoint [2] как расстояние между центром окружности и начальной точкой дуги. В режиме интерполяции по радиусной дуге AuxPoint [1] является значением радиуса, а параметр AuxPoint [2] недействителен.
- Числовые значения переменных EndPoint [1] - EndPoint [8] представляют собой координаты конечных точек каждой оси.

(4) Применение

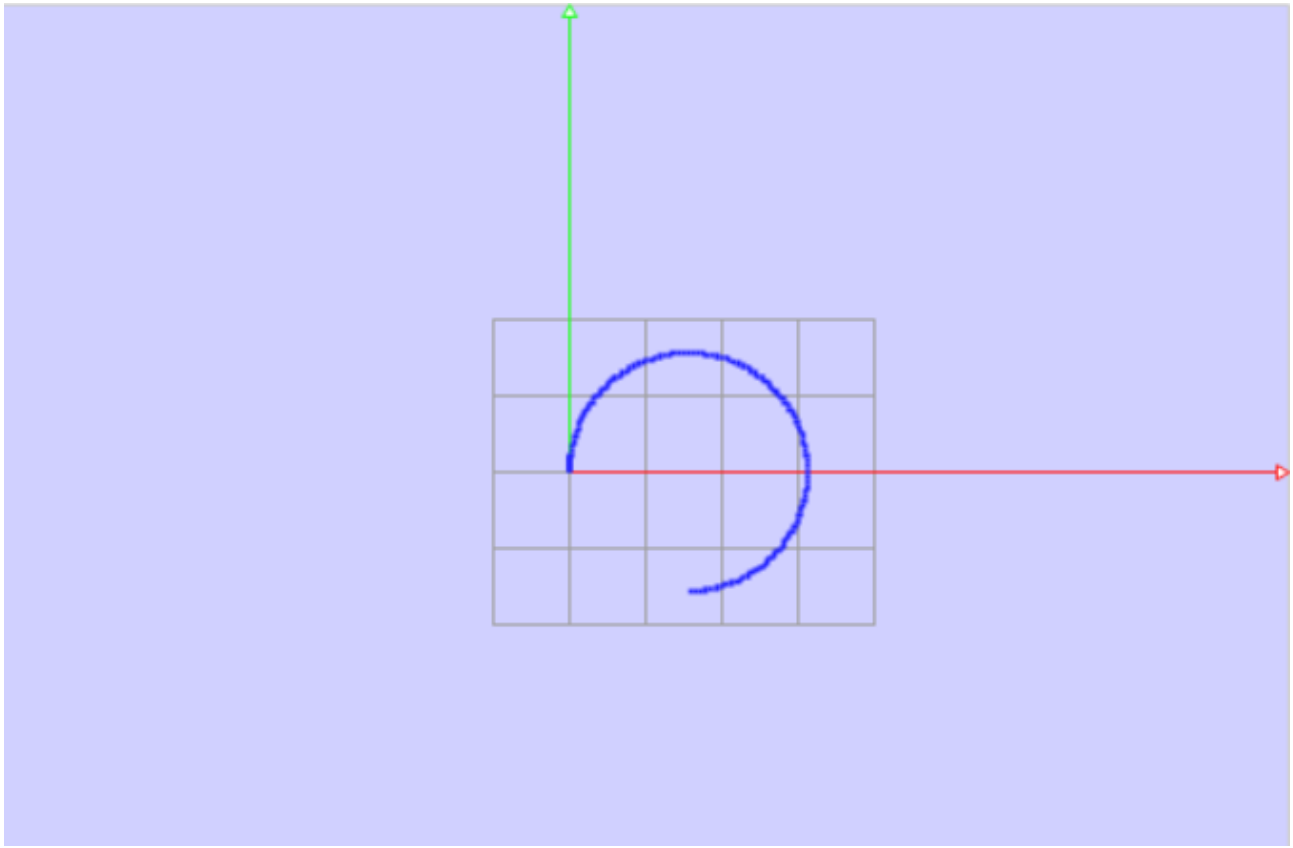
Пример 1: Чтобы выполнить дуговую интерполяцию в режиме относительного позиционирования, выберите кинематическую модель для двух осей станка с подвижным порталом, выберите мнимую ось для оси X, после чего выберите действительную ось Ethercat для оси Y.



Программирование: Установите текущие положения обеих осей на 0, выберите режим задания дуги по трем точкам, установите дополнительную точку AuxPoint на (50, 50), EndPoint на (50, 0), установите скорость обеих осей на 5, и начертите дугу относительно текущей точки (0, 0).



На рисунке ниже показана траектория дуги.

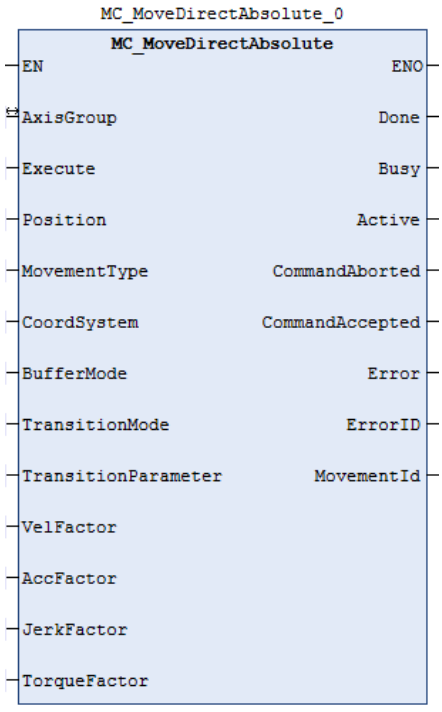


3-2-2-18. Быстрое перемещение в абсолютное положение [MC_MoveDirectAbsolute]

(1) Обзор команды

Управление всеми осями в группе осей для перемещения в конечную точку абсолютного положения с заданной скоростью.

Быстрое перемещение в абсолютное положение [MC_MoveDirectAbsolute]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveDirectAbsolute	Быстрое перемещение в абсолютное положение		MC_MoveDirectAbsolute_0(AxisGroup:= , Execute:= , Position:= , MovementType:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
Position	положение	SMC_POS_REF	-	-	Заданное абсолютное положение в указанной эталонной системе координат
MovementType	Режим перемещения PTP	SMC_PTP_MOVEMENT_TYPE	-	-	Fast (0): Режим перемещения PTP с временным приоритетом Path_Invariant: Перемещение PTP по фиксированной траектории
CoordSystem	Эталонная система координат	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM		Эталонная система координат
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Операция при задании нескольких команд на начало перемещения
TransitionMode	Режим углового перехода	MC_TRANSITION_MODE	TMNone/ TMStartVelocity/ TMCornerDistance	-	TMNone: без смешивания

					TMStartVelocity: смешивание на основе скорости TMCornerDistance: смешивание на основе расстояния
Transition Parameter	Параметры углового перехода	массив [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, положительное число	0	Параметры углового перехода
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент	LREAL	0-1	1	Коэффициент скорости

	скорости в толчковом режиме				в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAccepted	Прием команды на перемещение	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей

Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется или завершено

(3) Описание функции

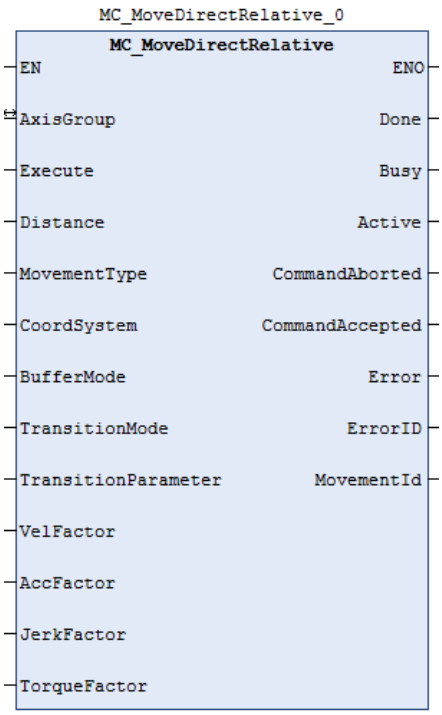
- Каждая ось перемещается в относительно независимое от других положений абсолютное положение, и ее траектория перемещения неизвестна.

3-2-2-19. Быстрое перемещение в относительное положение [MC_MoveDirectRelative]

(1) Обзор команды

Управление всеми осями в группе осей для перемещения в конечную точку относительного положения с заданной скоростью.

Быстрое перемещение в относительное положение [MC_MoveDirectRelative]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveDirectRelative	Быстрое перемещение в относительное положение		MC_MoveDirectRelative_0(AxisGroup:= , Execute:= , Distance:= , MovementType:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
Distance	относительное положение	SMC_POS_REF	-	-	Заданное относительное положение в указанной эталонной системе координат

Movement Type	Режим перемещения PTP	SMC_PTP_MOVEMENT_TYPE	-	-	Fast (0): Режим перемещения PTP с временным приоритетом Path_ Invariant: Перемещение PTP по фиксированной траектории
CoordSystem	Эталонная система координат	SMC_COORD_ SYSTEM	SMC_COORD_SYSTEM		Эталонная система координат
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Операция при задании нескольких команд на начало перемещения
Transition Mode	Режим углового перехода	MC_TRANSITION_MODE	TMNone/TMStartVelocity/TMCornerDistance		TMNone: без смешивания TMStartVelocity: смешивание на основе скорости TMCornerDistance: смешивание на основе расстояния
Transition Parameter	Параметры углового перехода	массив [0..(SMC_RCNST.MA	0, положительное число	0	Параметры углового перехода

		X_TRANS_PARAMS - 1)] OF LREAL			
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная

					на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
Command Aborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Command Accepted	Прием команды на перемещение	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется или завершено

(3) Описание функции

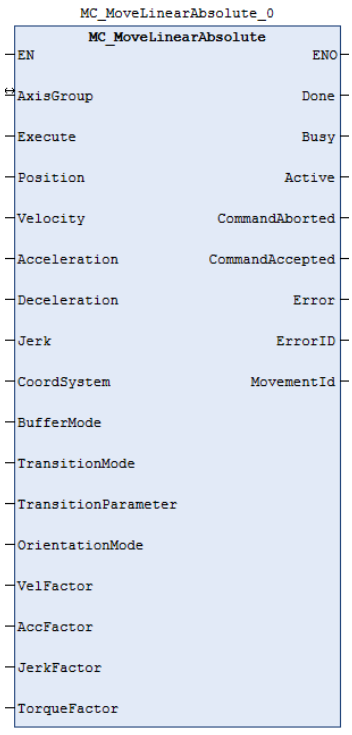
- Каждая ось перемещается в относительно независимое от других положений абсолютное положение, и ее траектория перемещения неизвестна.

3-2-2-20. Линейная интерполяция в режиме абсолютного позиционирования [MC_MoveLinearAbsolute]

(1) Обзор команды

Линейное интерполяционное перемещение группы осей управления в режиме абсолютного позиционирования в заданной системе координат.

Линейная интерполяция в режиме абсолютного позиционирования [MC_MoveLinearAbsolute]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveLinearAbsolute	Линейная интерполяция в режиме абсолютного позиционирования		MC_MoveLinearAbsolute_0(AxisGroup:= , Execute:= , Position:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE /FALSE	FALSE	Выполнение текущей команды
Position	положение	SMC_POS_REF	-	-	Заданное абсолютное положение в указанной эталонной системе координат
Velocity	скорость	LREAL	0, положительное число	0	Максимальная результирующая скорость [командная единица/с]
Acceleration	ускорение	LREAL	положительное число	0	Максимальное результирующее ускорение [командная единица/$с^2$]
Deceleration	торможение	LREAL	положительное число	0	Максимальное результирующее торможение

					[командная единица/с^2]
Jerk	Скорость в толчковом режиме	LREAL	положительное число	0	Максимальная результирующая скорость в толчковом режиме [командная единица/с^3]
CoordSystem	Эталонная система координат	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	Эталонная система координат
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Операция при задании нескольких команд на начало перемещения
TransitionMode	Режим углового перехода	MC_TRANSITION_MODE	TMNone/TMStartVelocity/TMCornerDistance	-	TMNone: без смешивания TMStartVelocity: смешивание на основе скорости TMCornerDistance: смешивание на основе расстояния
TransitionParameter	Параметры углового перехода	массив [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, положительное число	0	Параметры углового перехода

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
OrientationMode	Режим позиционирования с интерполяцией	SMC_ORIENTATION_MODE	GreatCircle/ Ось	-	GreatCircle: Перемещение по кратчайшему пути из начального положения в заданное положение. В этом режиме, даже если начальное положение и конечное положение находятся в пределах указанной области, фактическая траектория перемещения все равно может выйти за пределы данной области. Axis: Ось позиционирования перемещается в пределах указанной области от начального положения до конечного положения, и в этом режиме поддерживаются не все кинематические преобразования.
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная

					скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAccepted	Прием команды на перемещение	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется или завершено

(3) Описание функции

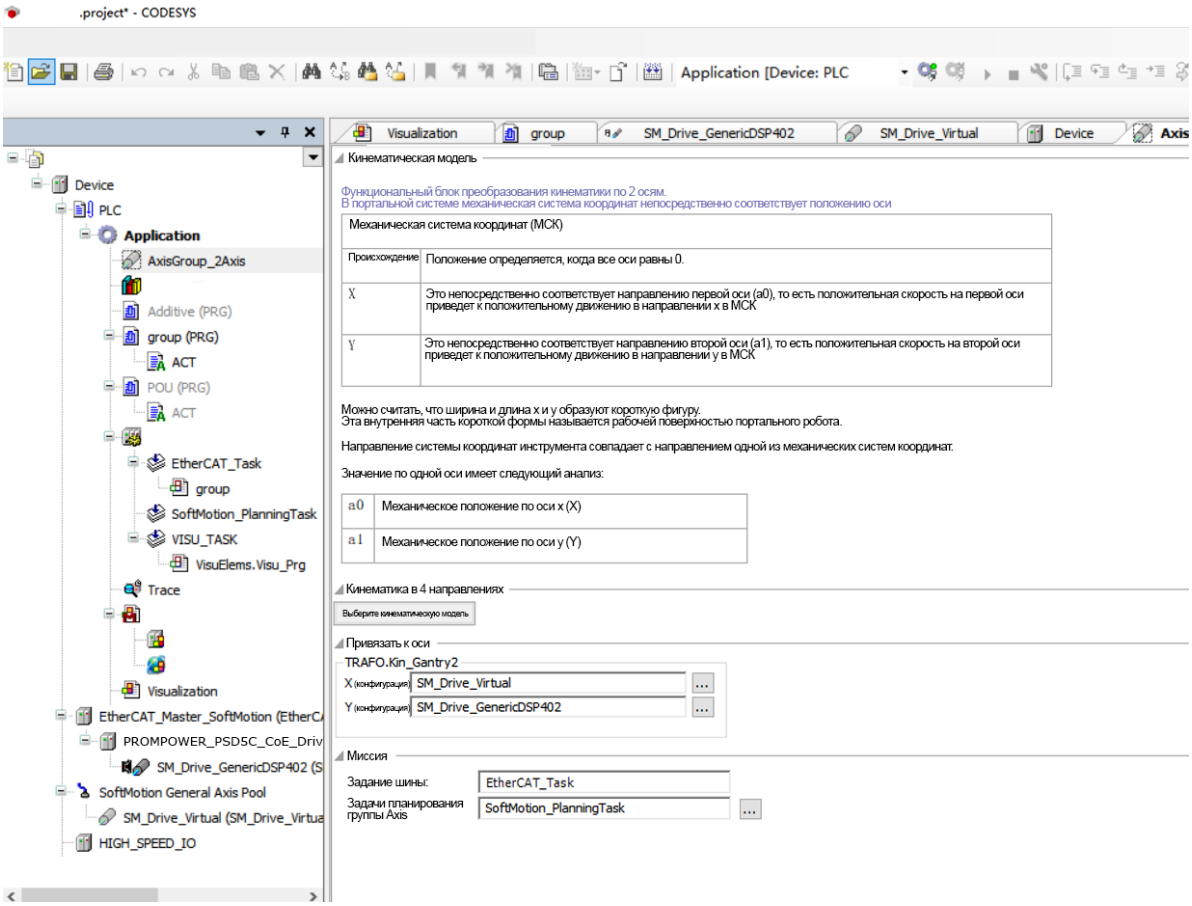
- Управление линейным интерполяционным перемещением группы осей в режиме абсолютного позиционирования в заданной системе координат.
- Отношение между параметрами BufferMode, TransitionMode и TransitionParameter:

(1) Если BufferMode выбирает режим mcBuffered, TransitionMode поддерживает только режим mcTMNone.

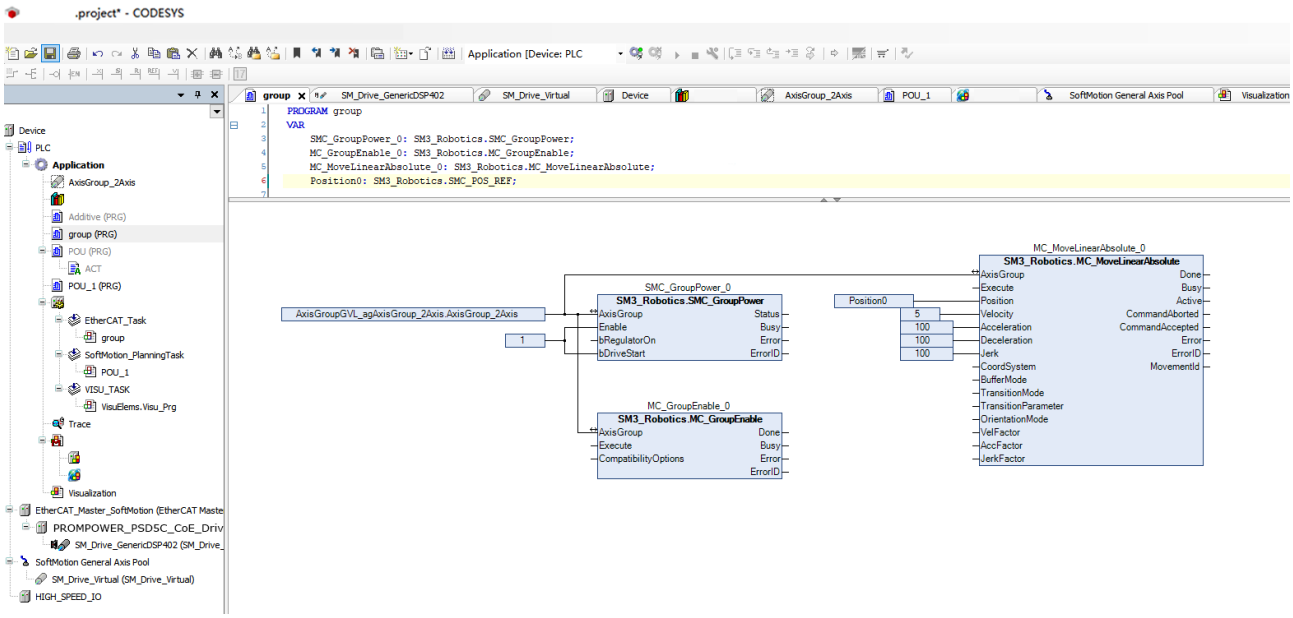
(2) Если BufferMode выбирает режим mcBlendingPrevious, можно будет выбрать TransitionMode.

- Режимы McTMConstantVelocity и mcTMCornerDistance.

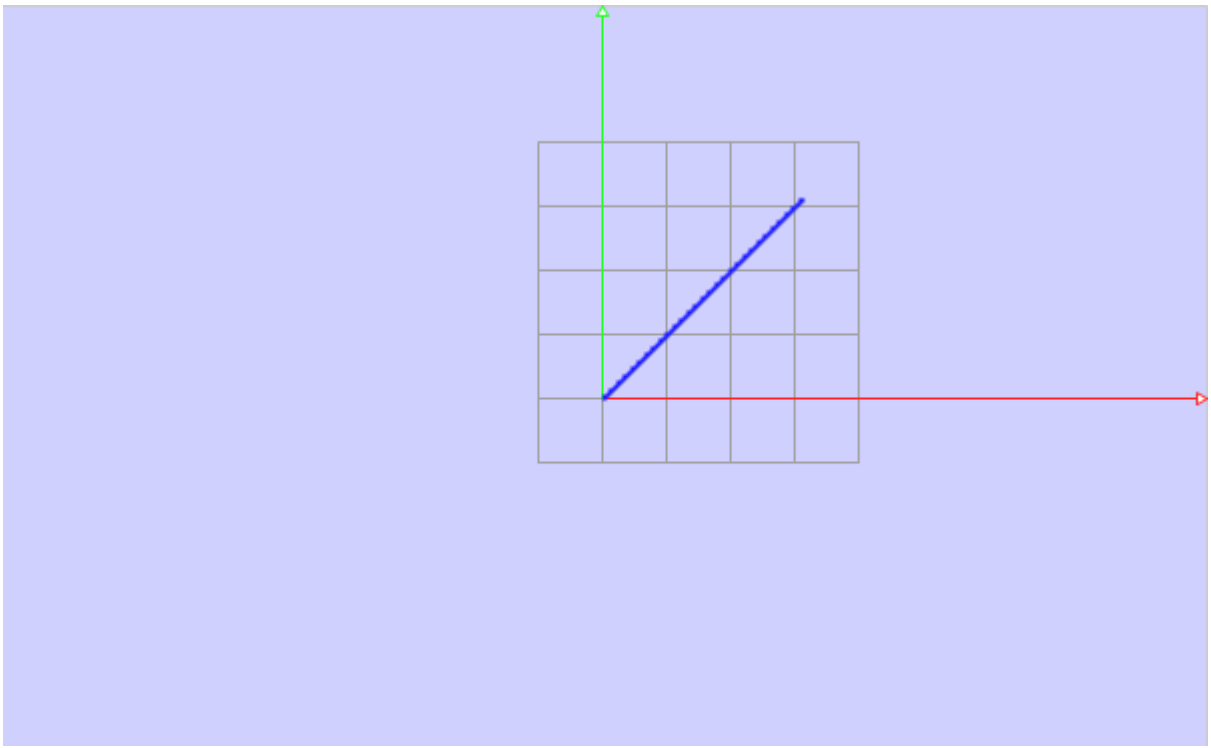
Пример 1: Чтобы выполнить интерполяцию в режиме абсолютного позиционирования, выберите кинематическую модель для двух осей станка с подвижным порталом, выберите мнимую ось для оси X, после чего выберите действительную ось Ethercat для оси Y.



Программирование: Установите текущие положения обеих осей на 0, настройте положение (100, 100) и выполните абсолютное линейное интерполяционное перемещение из начальной точки (0,0).



На рисунке ниже показана траектория прямолинейного перемещения.

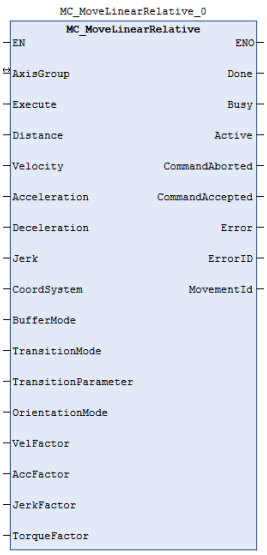


3-2-2-21. Линейная интерполяция в режиме относительного позиционирования [MC_MoveLinearRelative]

(1) Обзор команды

Линейное интерполяционное перемещение группы осей управления в режиме относительного позиционирования в заданной системе координат.

Линейная интерполяция в режиме относительного позиционирования [MC_MoveLinearRelative]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_MoveLinearRelative	Линейная интерполяция в режиме относительного позиционирования		MC_MoveLinearRelative_0(AxisGroup:= , Execute:= , Distance:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , CoordSystem:= , BufferMode:= , TransitionMode:= , TransitionParameter:= , OrientationMode:= , VelFactor:= , AccFactor:= , JerkFactor:= , TorqueFactor:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
Distance	относительное положение	SMC_POS_REF	-	-	Заданное относительное положение в указанной эталонной системе координат
Velocity	скорость	LREAL	0, положительное число	0	Максимальная результирующая скорость [командная единица/с]
Acceleration	ускорение	LREAL	положительное число	0	Максимальное результирующее ускорение [командная единица/$с^2$]
Deceleration	торможение	LREAL	положительное число	0	Максимальное результирующее торможение [командная единица/$с^2$]
Jerk	Скорость в толчковом режиме	LREAL	положительное число	0	Максимальная результирующая скорость в толчковом режиме [командная единица/$с^3$]
CoordSystem	Эталонная система координат	SMC_COORD_SYSTEM	SMC_COORD_SYSTEM	-	Эталонная система координат
BufferMode	Буферный режим	MC_BUFFER_MODE	-	0	Операция при задании нескольких команд на начало перемещения

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Transition Mode	Режим углового перехода	MC_TRANSITION_MODE	TMNone/TMStartVelocity/TMCornerDistance	-	TMNone: без смешивания TMStartVelocity: смешивание на основе скорости TMCornerDistance: смешивание на основе расстояния

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Transition Parameter	Параметры углового перехода	массив [0..(SMC_RCNST.MAX_TRANS_PARAMS - 1)] OF LREAL	0, положительное число	0	Параметры углового перехода
OrientationMode	Режим позиционирования с интерполяцией	SMC_ORIENTATION_MODE	GreatCircle /Axis	-	GreatCircle: Перемещение по кратчайшему пути из начального положения в заданное положение. В этом режиме, даже если начальное положение и конечное положение находятся в пределах указанной области, фактическая траектория перемещения все равно может выйти за пределы данной области. Axis: Ось позиционирования перемещается в пределах указанной

					области от начального положения до конечного положения, и в этом режиме поддерживаются не все кинематические преобразования.
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения

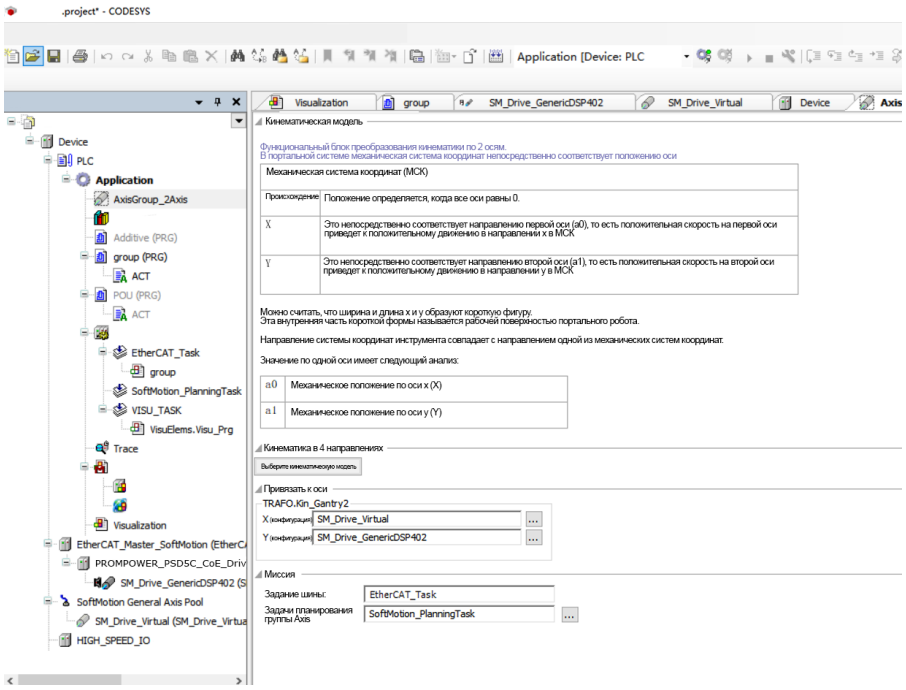
					выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
Command Aborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Command Accepted	Прием команды на перемещение	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Movement Id	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется или завершено

(3) Описание функции

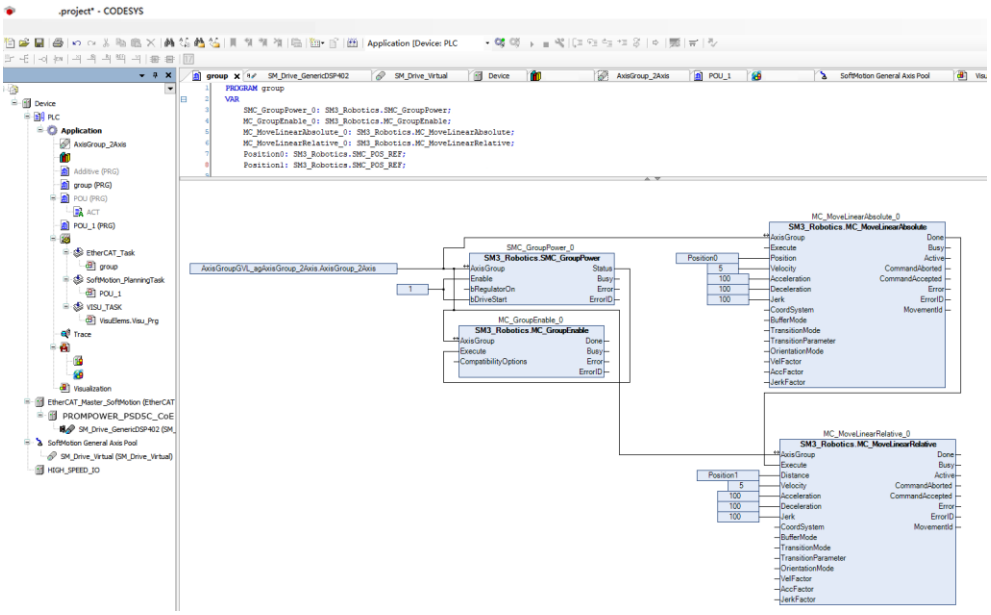
- Параметр TransitionParameter действителен только в том случае, если TransitionMode имеет значение mcTMCornerDistance.

(4) Применение

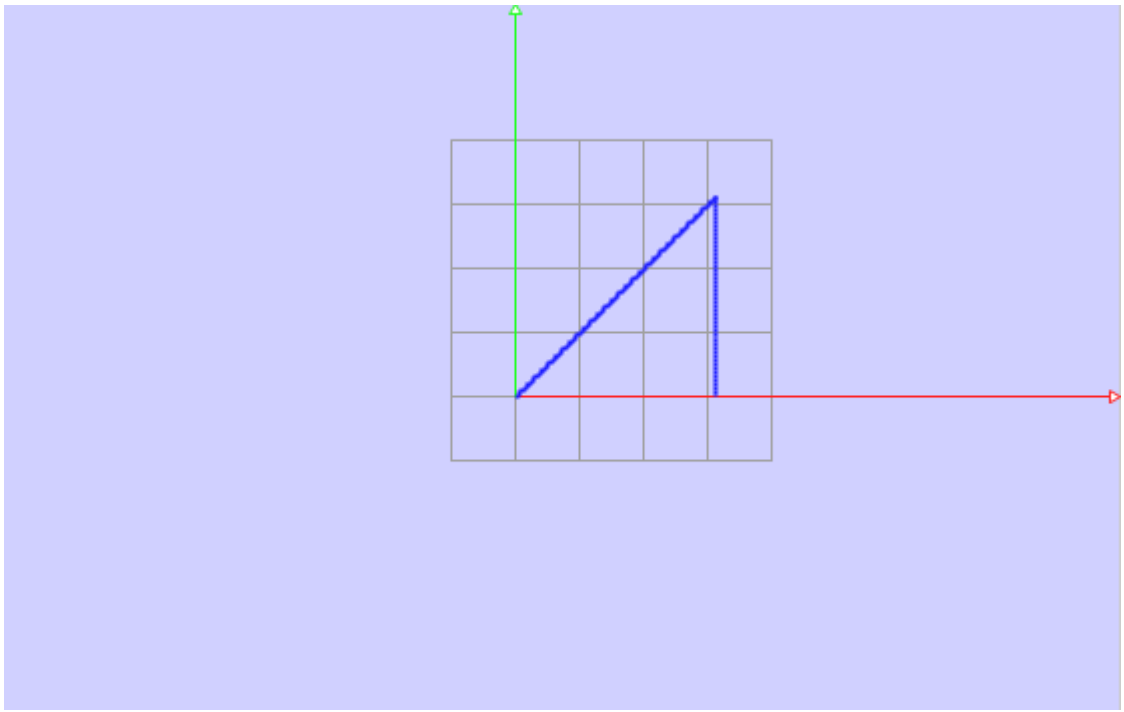
Пример 1: Чтобы выполнить интерполяцию в режиме относительного позиционирования, выберите кинематическую модель для двух осей станка с подвижным порталом, выберите мнимую ось для оси X, после чего выберите действительную ось Ethercat для оси Y.



Программирование: Установите текущие положения обеих осей на 0, то есть начальное положение на (0, 0), положение при линейной интерполяции в режиме абсолютного позиционирования на (100, 100) и положение при линейной интерполяции в режиме относительного позиционирования на (0, -100).



На рисунке ниже показана траектория прямолинейного перемещения.

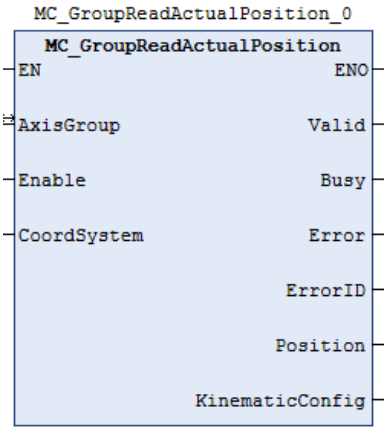


3-2-2-22. Считывание данных обратной связи по положению группы осей [MC_GroupReadActualPosition]

(1) Обзор команды

Считывание данных обратной связи по положению группы осей в заданной системе координат.

Считывание данных обратной связи по положению группы осей [MC_GroupReadActualPosition]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupReadActualPosition	Считывание данных обратной связи по положению группы осей		MC_GroupReadActualPosition_0(AxisGroup:= , Enable:= , CoordSystem:= , Valid=> , Busy=> , Error=> , ErrorID=> , Position=> , KinematicConfig=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
CoordSystem	Применяемая система координат	SMC_COORD_SYSTEM	-	-	Применяемая система координат
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Положение	положение	SMC_POS_REF	-	-	Фактическое положение группы осей
KinematicConfig	Кинематическая конфигурация	TRAFO.CONFIGDATA	-	-	Кинематическая конфигурация текущего положения. Задается только для декартовой системы координат (т.е. не задается в ACS)

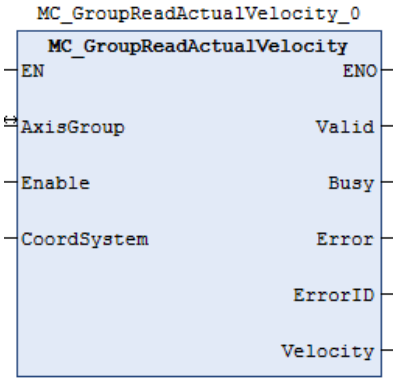
3-2-2-23. Считывание данных обратной связи по скорости группы осей [MC_GroupReadActualVelocity]

(1) Обзор команды

Считывание данных обратной связи по скорости группы осей в заданной системе координат.

Считывание данных обратной связи по скорости группы осей [MC_GroupReadActualVelocity]			
Условие выполнения	Нормальное включение/выключение	Совместимые модели	PMP3xx

	триггера		
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupReadActualVelocity	Считывание данных обратной связи по скорости группы осей		MC_GroupReadActualVelocity_0(AxisGroup:= , Enable:= , CoordSystem:= , Valid=> , Busy=> , Error=> , ErrorID=> , Velocity=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
CoordSystem	Применяемая система координат	SMC_COORD_SYSTEM	-	-	Применяемая система координат
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

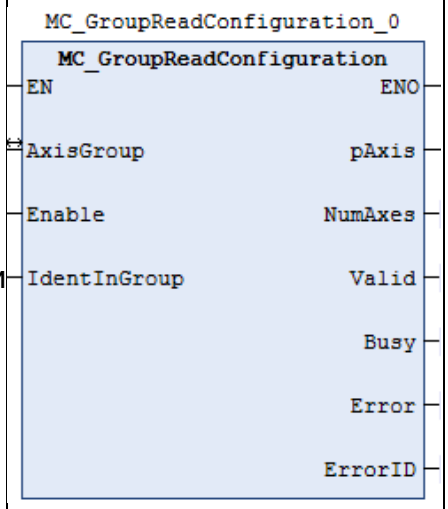
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
Velocity	скорость	SMC_POS_REF	-	-	Текущие данные обратной связи по скорости группы осей. Если выбрана декартова система координат; Velocity. c содержит значения скорости в декартовой системе координат: (X, Y, Z) — это вектор скорости, а (A, B, C) — угловая скорость вокруг осей x, y и z, соответственно.

3-2-2-24. Считывание параметров конфигурации группы осей [MC_GroupReadConfiguration]

(1) Обзор команды

Считывание параметров конфигурации, например, оси и их количество в группе.

Считывание параметров конфигурации группы осей [MC_GroupReadConfiguration]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupReadConfiguration	Считывание параметров конфигурации группы осей		MC_GroupReadConfiguration_0(AxisGroup:= , Enable:= , IdentInGroup:= , pAxis=> , NumAxes=> , Valid=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

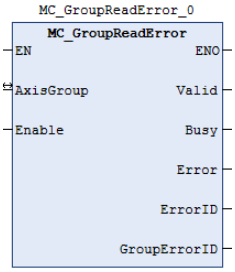
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
IdentInGroup	Номер оси	IDENT_IN_GROUP_REF_SM3	0, положительное число	0	Ввод номера соответствующей оси в группе осей
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
pAxis	Опорная ось	POINTER TOAXIS_REF_SM3	-	-	Выбранная опорная ось
NumAxes	Количество осей	UDINT	0, положительное число	0	Количество осей в группе осей
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

3-2-2-25. Считывание информации об ошибках для группы осей [MC_GroupReadError]

(1) Обзор команды

Получение информации об ошибках для группы осей.

Считывание информации об ошибках для группы осей [MC_GroupReadError]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupReadError	Считывание информации об ошибках для группы осей		MC_GroupReadError_0(AxisGroup:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , GroupErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки для одной оси
GroupErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки для группы осей

(3) Описание функции

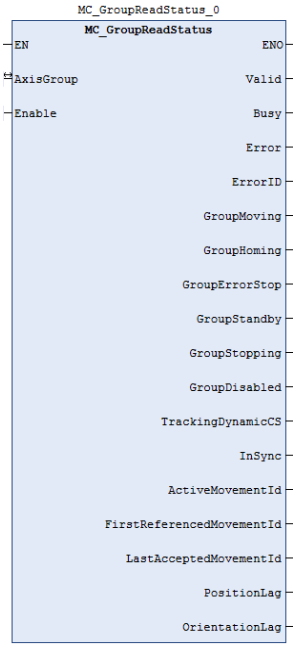
- Получение информации об ошибках для одной оси и группы осей, например, считывание данных о жестком пределе (или мягком пределе) или ошибки одной оси.

3-2-2-26. Считывание текущего рабочего состояния группы осей [MC_GroupReadStatus]

(1) Обзор команды

Используется для получения информации о текущих перемещениях группы осей.

Считывание текущего рабочего состояния группы осей [MC_GroupReadStatus]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
MC_GroupReadStatus	Считывание информации о текущих перемещениях группы осей		MC_GroupReadStatus_0(AxisGroup:= , Enable:= , Valid=> , Busy=> , Error=> , ErrorID=> , GroupMoving=> , GroupHoming=> , GroupErrorStop=> , GroupStandby=> , GroupStopping=> , GroupDisabled=> , TrackingDynamicCS=> , InSync=> , ActiveMovementId=> , FirstReferencedMovementId=> , LastAcceptedMovementId=> , PositionLag=> , OrientationLag=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Для включения функции обработки функционального блока должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE

Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональног о блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональног о блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки для одной оси
GroupErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки для группы осей
GroupMoving	перемещение	BOOL	TRUE/FALSE	FALSE	Группа осей перемещается — TRUE
GroupHoming	возврат в исходное положение	BOOL	TRUE/FALSE	FALSE	Возврат группы осей в исходное положение — TRUE
GroupErrorSt op	Останов при ошибке	BOOL	TRUE/FALSE	FALSE	Останов при ошибке в группе осей — TRUE
GroupStandby	Готовность к перемещению	BOOL	TRUE/FALSE	FALSE	Готовность группы осей к перемещению — TRUE
GroupStoppin g	Остановка	BOOL	TRUE/FALSE	FALSE	True при остановке перемещения группы осей
GroupDisable d	Группа осей не включена	BOOL	TRUE/FALSE	FALSE	Группа осей не работает — TRUE
TrackingDyna micCS	В настоящий момент в динамической	BOOL	TRUE/FALSE	FALSE	True, если в настоящее время используется

	системе координат				динамическая система координат
InSync	На заданной траектории или уже в рабочем положении	BOOL	TRUE/FALSE	FALSE	True при непрерывном интерполяционном перемещении, когда полученное положение принадлежит заданной траектории. True в стандартной точке и при интерполяционном перемещении, когда текущее положение равно заданному положению
ActiveMovementId	Номер сегмента перемещения	SMC_Movement_Id	0, положительное число	-	Идентификатор текущего перемещения. Id=0 указывает на отсутствие перемещения

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
LastAcceptedMovementId	Номер сегмента перемещения	SMC_Movement_Id	0, положительное число	0	Идентификатор последнего принятого перемещения. Id=0 означает, что никаких операций еще не было выполнено.

(3) Описание функции

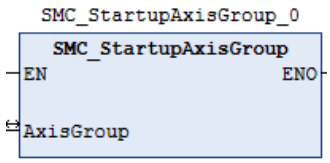
- Эта команда позволяет непрерывно получать такую информацию, как текущее состояние перемещения группы осей, находятся ли оси группы в рабочем положении, а также номер сегмента перемещения.

3-2-2-27. Запуск группы осей [SMC_StartupAxisGroup]

(1) Обзор команды

Используется для испытания и отладки групп осей.

Запуск группы осей [SMC_StartupAxisGroup]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
SMC_StartupAxisGroup	Запуск группы осей		SMC_StartupAxisGroup_0(AxisGroup:=);

(2) Связанные переменные

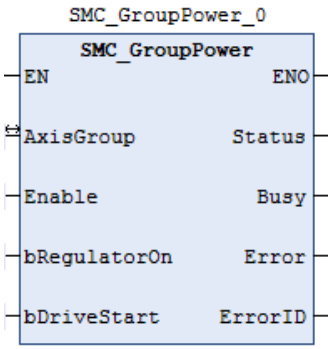
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей

3-2-2-28. Включение группы осей [SMC_GroupPower]

(1) Обзор команды

Включение всех осей в группе осей. Эквивалентно вызову команды MC_Power для всех осей в группе осей.

Включение группы осей [SMC_GroupPower]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
SMC_GroupPower	Включение группы осей		SMC_GroupPower_0(AxisGroup:= , Enable:= , bRegulatorOn:= , bDriveStart:= , Status=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Введите TRUE для запуска модуля

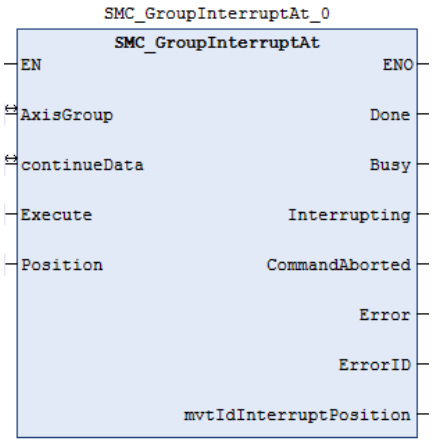
bRegulatorOn	включение	BOOL	TRUE/FALSE	FALSE	Введите TRUE, чтобы включить группу осей
bDriveStart	Включение привода	BOOL	TRUE/FALSE	FALSE	Для выключения функции обработки функциональных блоков при аварийном останове должна быть в состоянии TRUE
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Status	Готовность к работе	BOOL	TRUE/FALSE	FALSE	True, если группа осей готова к работе
Valid	Получение признака	BOOL	TRUE/FALSE	FALSE	Если выходное значение действительно, то состояние будет TRUE
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки

3-2-2-29. Настройка перерыва в перемещении [SMC_GroupInterruptAt]

(1) Обзор команды

Прерывание перемещения группы осей.

Настройка перерыва в перемещении [SMC_GroupInterruptAt]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
SMC_GroupInterruptAt	Настройка перерыва в перемещении		SMC_GroupInterruptAt_0(AxisGroup:= , continueData:= , Execute:= , Position:= , Done=> , Busy=> , Interrupting=> , CommandAborted=> , Error=> , ErrorID=> , mvtIdInterruptPosition=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
continueData	Продолжение передачи данных о перемещении	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Информация о перемещении при прерывании перемещения группы осей

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
Position	положение	SMC_GroupInterrupt Положение	-	-	Место, в котором перемещение будет прерываться
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Interrupting	Прерывание	BOOL	TRUE/FALSE	FALSE	Сигнал, указывающий на прерывание выполнения функционального блока
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки
mvtIdInterrupt Position	Идентификатор положения в момент прерывания	-	-	-	

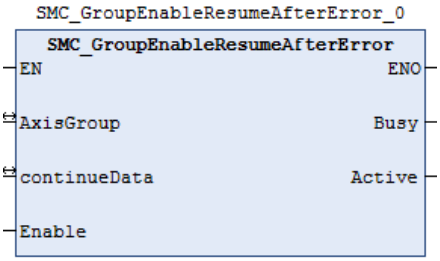
	перемещения				

3-2-2-30. Перезапуск после сброса ошибок [SMC_GroupEnableResumeAfterError]

(1) Обзор команды

Восстановление состояния группы осей, которое было нарушено из-за ошибки.

Перезапуск после сброса ошибок [SMC_GroupEnableResumeAfterError]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
SMC_GroupEnableResumeAfterError	Перезапуск после сброса ошибок		SMC_GroupEnableResumeAfterError_0(AxisGroup:= , continueData:= , Enable:= , Busy=> , Active=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
continueData	Продолжение передачи данных о перемещении	SMC_AXIS_GROUP_CONTINUE_DATA	-	-	Положения группы осей в момент прерывания перемещения

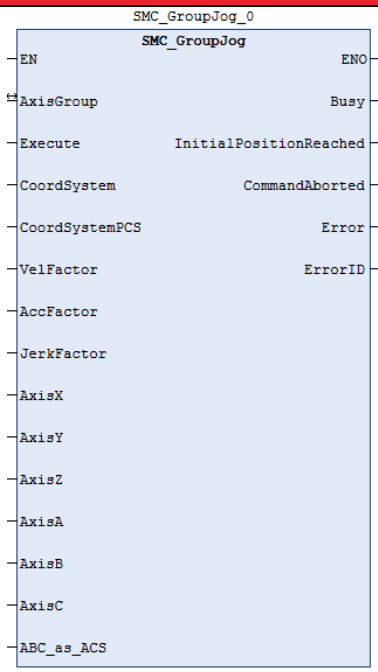
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Enable	Действительн.	BOOL	TRUE/FALSE	FALSE	Включение командной функции
VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В процессе вызова	BOOL	TRUE/FALSE	FALSE	Возобновление записи данных

3-2-2-31. Запуск группы осей в толчковом режиме [SMC_GroupJog]

(1) Обзор команды

Управление группой осей для их перемещения в толчковом режиме в заданной системе координат.

Запуск группы осей в толчковом режиме [SMC_GroupJog]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
SMC_GroupJog	Запуск группы осей в толчковом режиме		SMC_GroupJog_0(AxisGroup:= , Execute:= , CoordSystem:= , CoordSystemPCS:= , VelFactor:= , AccFactor:= , JerkFactor:= , AxisX:= , AxisY:= , AxisZ:= , AxisA:= , AxisB:= , AxisC:= , ABC_as_ACS:= , Busy=> , InitialPositionReached=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
CoordSystem	эталонная система координат	SMC_COORD_SYSTEM	-	-	Эталонная система координат
CoordSystem PCS	Система координат PCS	SMC_COORD_SYSTEM	-	-	PCS — это внутренняя система координат, используемая для работы в толчковом режиме, а команда SMC_SetDynCoordTransformEx применяется для

					внесения изменений.
VelFactor	Коэффициент скорости	LREAL	0-1	1	Коэффициент скорости; максимальная скорость каждой оси, умноженная на этот коэффициент скорости; значение находится в диапазоне [0,1]
AccFactor	коэффициент ускорения	LREAL	0-1	1	Коэффициент ускорения; максимальная скорость каждой оси, умноженная на этот коэффициент ускорения; значение находится в диапазоне [0,1]
JerkFactor	Коэффициент скорости в толчковом режиме	LREAL	0-1	1	Коэффициент скорости в толчковом режиме; максимальная скорость каждой оси, умноженная на этот коэффициент скорости в толчковом режиме; значение находится в диапазоне [0,1]
AxisX	Ось X	IAxisRef	-	0	Ось X в системе координат устанавливается на 0, если она не используется.
AxisY	Ось Y	IAxisRef	-	0	Ось Y в системе координат устанавливается на 0, если она не используется.

AxisZ	Ось Z	IAxisRef	-	0	Ось Z в системе координат устанавливается на 0, если она не используется.
AxisA	Ось A	IAxisRef	-	0	Управление координатой a (вращение вокруг оси z) или первой осью инструмента. Если не применимо, установите значение 0.

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisB	Ось B	IAxisRef	-	0	Управление координатой b (вращение вокруг оси y) или второй осью инструмента. Если не применимо, установите значение 0.
AxisC	Ось C	IAxisRef	-	0	Управление координатой c (вращение вокруг оси z) или третьей осью инструмента. Если не применимо, установите значение 0.
ABC_as_ACS	Запуск преобразования координат	BOOL	TRUE/FALSE	FALSE	В состоянии true положения AxisA, AxisB и AxisC будут интерпретироваться как заданное положение кинематической оси инструмента, в противном случае они будут представлены как направление ZYZ. Если параметр CoordSystem установлен на ACS, он будет игнорироваться. В состоянии true кинематическое преобразование группы осей должно относиться к типу Kin_ Coupled и

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					поддерживать SMC_ORIENTATION_MODE. Axis.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
InitialPositionReached	Принятые данные о положении	BOOL	TRUE/FALSE	FALSE	-
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки

(3) Описание функции

- При перемещении в толчковом режиме группа осей будет двигаться в соответствии с положением и направлением, заданными сконфигурированной осью.
- Команда преобразует входную систему координат в заданную систему координат для вывода, например, перемещение в толчковом режиме в системе координат станка с помощью MCS.
- Настройка параметров ABC_as_ACS осуществляется с помощью гибридного преобразования опорной оси, при котором X/Y/Z — положение декартовой системы координат, а A/B/C — положение системы координат инструмента.
- Если система координат изделия используется для изменения положения в реальном времени, данную операцию можно выполнить, настроив команду CoordSystemPCS входной системы координат, а изменения в группе осей в системе координат в режиме реального времени необходимо будет обозначить.
- Чтобы перезапустить этот функциональный блок без изменения CoordSystemPCS, следует ввести другую команду перемещения, например

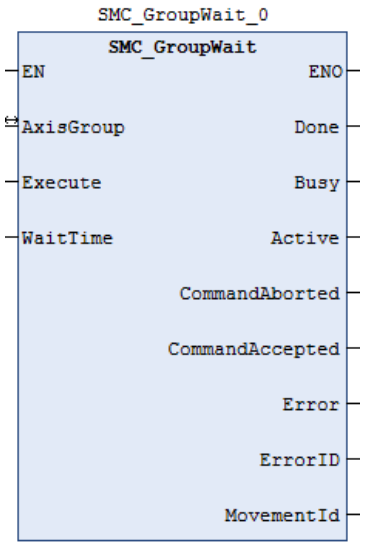
MC_GroupHalt. Если нарушить этот принцип использования, будет возвращена ошибка SMC_ AXIS_ GROUP_ PCS_ STILL_ IN_USE.

3-2-2-32. Задержка в работе группы осей [SMC_GroupWait]

(1) Обзор команды

Настройка задержки в работе группы осей.

Задержка в работе группы осей [SMC_GroupWait]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Robotics		

Команда	Название	Графическое представление	Представление на языке ST
SMC_GroupJog	Задержка в работе группы осей		SMC_GroupWait_0(AxisGroup:= , Execute:= , WaitTime:= , Done=> , Busy=> , Active=> , CommandAborted=> , CommandAccepted=> , Error=> , ErrorID=> , MovementId=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
AxisGroup	Группа осей	AXIS_GROUP_REF_SM3	-	-	Выбранная группа осей
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды
WaitTime	Время ожидания	LREAL	0, положительное число	0	Время ожидания на траектории, в секундах
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
CommandAccepted	Команда на перемещение принята	BOOL	TRUE/FALSE	FALSE	True, если модуль успешно вызвал группу осей
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Код ошибки

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
MovementId	Признак перемещения	SMC_Movement_Id	TRUE/FALSE	FALSE	True, если перемещение выполняется или завершено

(3) Описание функции

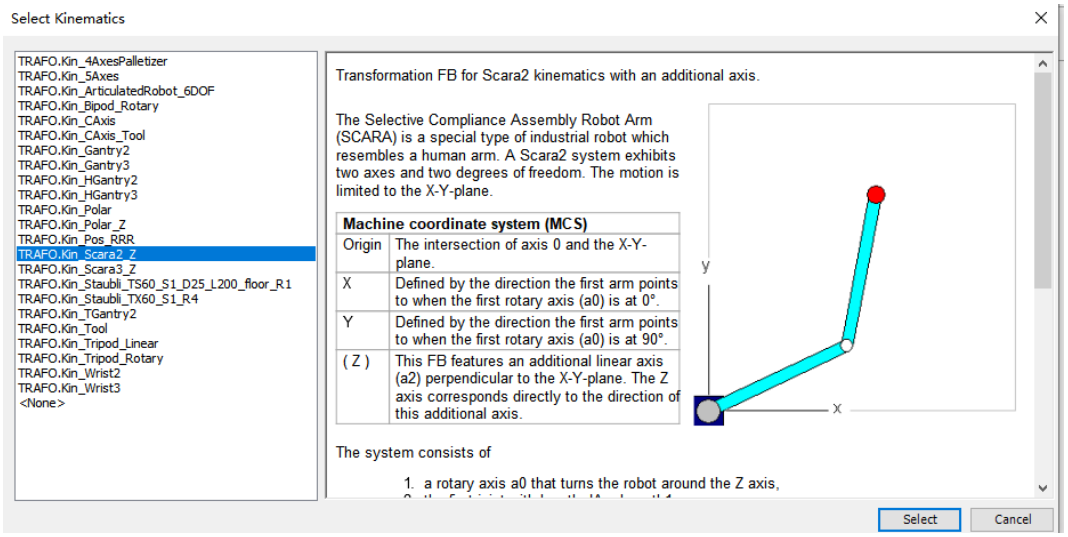
- Если время ожидания между двумя соседними командами и время, выделенное для выполнения задачи, одинаковы, то фактическое время ожидания между двумя перемещениями, заданное через команду SMC_GroupWait станет меньше. Например, выполнение следующей команды обычно начинается со следующего цикла после данного цикла выполнения задачи, но если в середине есть SMC_GroupWait, то после окончания задержки выполнения команды SMC_GroupWait немедленно начнется выполнение следующей команды перемещения до начала следующего цикла.
- Если после отслеживания перемещения подается команда на ожидание, то группа осей будет отслеживать конечную точку предыдущего перемещения в течение заданного времени.
- Если каждая ось не находится в состоянии Standstill (Остановка), не управляется данной группой осей и ожидает вызова, то группа осей выдаст ошибку SMC_AXIS_GROUP_IDLE_WAIT_AXES_MOVING.
- В отличие от обновления каждого цикла в рамках цикла выполнения задачи, если время выполнения команды SMC_ GroupWait отличается от кратного времени цикла шины, то следующая команда может выполняться сразу по истечении времени ожидания, что делает переход к последующим перемещениям более плавным.
- По техническим причинам время ожидания можно увеличивать максимум на один цикл в следующих условиях: - без отслеживания -> ожидание -> отслеживание - отслеживание -> ожидание -> без отслеживания - отслеживание -> ожидание -> отслеживание перемещения PTP.

3-2-3. Применение функции для группы осей

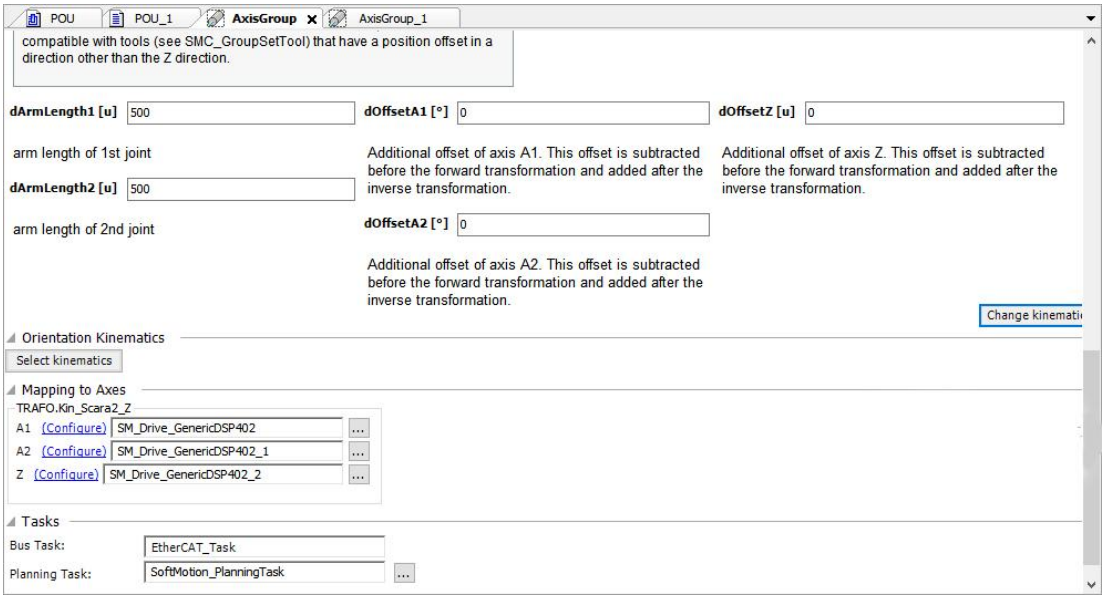
Пример: В этой программе реализовано управление перемещением для системы SCARA с двумя узлами.

Работа с программой:

(1) Щелкните правой кнопкой мыши по "Application" (Применение) на панели инженерного оборудования, выберите "Add Object" - "AxisGroup" ("Добавить объект" - "AxisGroup"), присвойте ему имя AxisGroup и откройте его. Выберите кинематическую модель SCARA через пункт меню "Kinematic Model" (Кинематическая модель).

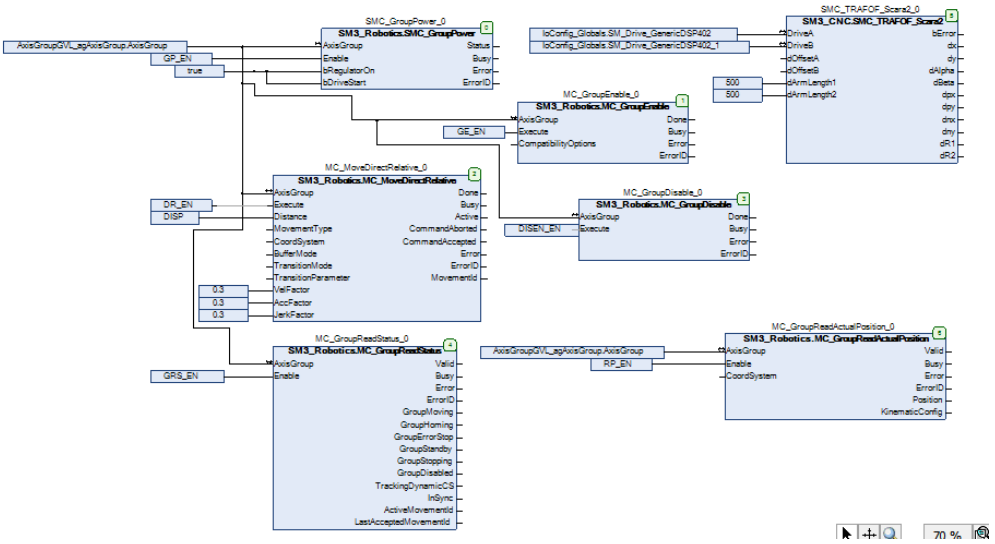


(2) Настройте кинематические параметры: Установите длину большого и малого рычагов на 500, а угловое смещение - на ноль. Настройка параметров осей с отображением положений осей.



(3) Напишите кинематические команды и настройте параметры перемещения на основе траектории перемещения.

```
PROGRAM POU
VAR
    SMC_GroupPower_0: SM3_Robotics.SMC_GroupPower;
    GP_EN: BOOL;              servo enable
    GE_EN: BOOL;              axis group startup
    MC_MoveDirectRelative_0: SM3_Robotics.MC_MoveDirectRelative;
    DR_EN: BOOL;                    move to relative position
    DISP: SM3_Robotics.SMC_POS_REF;                  set relative position
    DIST: SM3_Robotics.SMC_POS_REF;
    MC_GroupEnable_0: SM3_Robotics.MC_GroupEnable;
    MC_GroupReadStatus_0: SM3_Robotics.MC_GroupReadStatus;
    GRS_EN: BOOL;                   read axis group status
    Kin_HGantry2_0: TRAFO.Kin_HGantry2;
    SMC_TRAFOF_Gantry2_0: SM3_CNC.SMC_TRAFOF_Gantry2;
    SMC_TRAFOF_Scara2_0: SM3_CNC.SMC_TRAFOF_Scara2;
    MC_GroupReadActualPosition_0: SM3_Robotics.MC_GroupReadActualPosition;
    RP_EN: BOOL;                    read real-time position
    MC_GroupDisable_0: SM3_Robotics.MC_GroupDisable;
    DISEN_EN: BOOL;                 axis group disable
END VAR
```



3-3. Функции для кулачков

3-3-1. Перечень команд для кулачков

Команда	Описание
MC_CamTableSelect	Связь выбранной таблицы кулачков с фактической таблицей кулачков
MC_CamIn	Связь через кулачок
MC_CamOut	Разблокировка связи через кулачок
MC_GearIn	Настройка передаточного числа главной оси и ведомой оси и включение электронных редукторов

MC_GearInPos	Настройка расстояния синхронизации и передаточного числа главной оси и ведомой оси и включение электронных редукторов
MC_GearOut	Отключение электронных редукторов главной оси и ведомой оси
MC_Phasing	Сдвиг фаз
SMC_CAMBounds	Связь ведомой оси со шпинделем
SMC_CAMBounds_Pos	Связь шпинделя с ведомой осью
SMC_CamEditor	Визуализация таблицы кулачков
SMC_CamRegister	Считывание информации о толкателях
SMC_GetCamSlaveSetPosition	Считывание положения ведомой оси из таблицы кулачков
SMC_GetTappetValue	Оценка выходных толкателей
SMC_ReadCAM	Считывание данных из таблицы кулачков
SMC_WriteCAM	Запись данных в таблицу кулачков

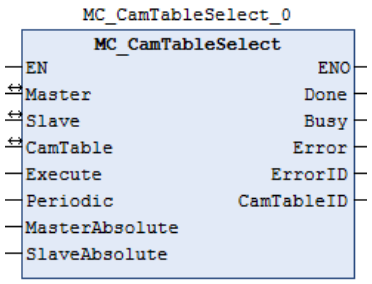
3-3-2. Команды для кулачков

3-3-2-1. Выбор таблицы кулачков [MC_CamTableSelect]

(1) Обзор команды

Используется для выбора выполняемой таблицы кулачков, которую необходимо применять одновременно с командой MC_CamIn.

Выбор таблицы кулачков [MC_CamTableSelect]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_CamTableSelect	Выбор таблицы кулачков		MC_CamTableSelect_0(Master:= , Slave:= , CamTable:= , Execute:= , Periodic:= , MasterAbsolute:= , SlaveAbsolute:= , Done=> , Busy=> , Error=> , ErrorID=> , CamTableID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к ведомой оси
CamTable	Таблица кулачков	MC_CAM_REF	-	-	Привязка к описанию таблицы кулачков
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
Periodic	Циклический режим	BOOL	TRUE/FALSE	FALSE	TRUE: Периодическое и повторное выполнение выбранной таблицы кулачков FALSE: Однократное выполнение таблицы кулачков

MasterAbsolute	Абсолютный режим главной оси	BOOL	TRUE/FALSE	FALSE	TRUE: обозначает абсолютные координаты FALSE: обозначает относительные координаты
SlaveAbsolute	Абсолютный режим ведомой оси	BOOL	TRUE/FALSE	FALSE	TRUE: обозначает абсолютные координаты FALSE: обозначает относительные координаты
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
CamTableID	Действующий идентификатор таблицы кулачков	MC_CAM_ID	-	-	Выбор действующего Cam_ID, используемого вместе с CamTableID в MC_CamIn

(3) Описание функции

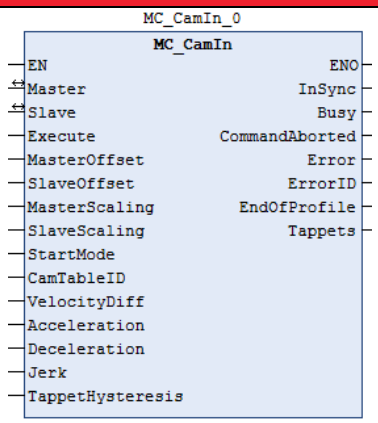
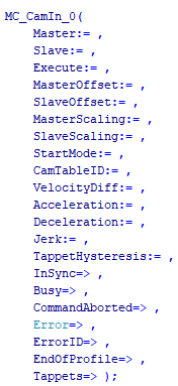
- Данная команда реализована на основе библиотеки "SM3_Basic".
- Эта команда позволяет выбирать таблицу кулачков, необходимую для работы электронных кулачков, поэтому перед началом использования данной команды таблицу нужно отредактировать (в редакторе кулачков или онлайн редакторе).
- Выполните выбранную таблицу кулачков при нарастающем фронте сигнала Execute или обновите выбранную таблицу после внесения в нее изменений.
- Если выходной сигнал Done имеет значение TRUE, генерируется и вступает в силу выходная переменная "CamTableID".
- Главную ось и ведомую ось нельзя задавать как одну ось, иначе появится ошибка.

3-3-2-2. Связь через кулачок [MC_CamIn]

(1) Обзор команды

Привязка главной оси к ведомой оси, также можно задавать режим включения и скорость перемещения ведомой оси.

Связь через кулачок [MC_CamIn]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_CamIn	Связь через кулачок		

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к ведомой оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
MasterOffset	Смещение главной оси	LREAL	-	0	Смещение по таблице главной оси
SlaveOffset	Смещение ведомой оси	LREAL	-	0	Смещение по таблице ведомой оси
MasterScaling	Масштабирование главной оси	LREAL	-	1	Коэффициент масштабирования файла конфигурации главной оси
SlaveScaling	Масштабирование ведомой оси	LREAL	-	1	Коэффициент масштабирования файлов конфигурации ведомой оси
StartMode	Метод включения ведомой оси	MC_StartMode	0-4	абсолютное	0: абсолютное положение 1: относительное положение 2: ramp_in (линейное изменение

					величины) 3: ramp_in_pos (линейное увеличение величины) 4: ramp_in_neg (линейное уменьшение величины)
CamTableID	Таблица кулачков	MC_CAM_ID	-	-	Настройка использования таблицы кулачков, выходная точка MC_CamTableSelect, используется вместе с CamTableID
VelocityDiff	скорость	LREAL	Отрицательное число, 0, положительное число	0	Максимальная скорость наложенного перемещения [импульсов/с]
Acceleration	Заданное ускорение	LREAL	Отрицательное число, 0, положительное число	0	ускорение [импульсов/с^2]
Deceleration	Заданное торможение	LREAL	Отрицательное число, 0, положительное число	0	торможение [импульсов/с^2]
Jerk	Заданная скорость в толчковом режиме	LREAL	Отрицательное число, 0, положительное число	0	скорость в толчковом режиме [импульсов/с^3]
TappetHysteresis	Гистерезис толкателей	LREAL	-	-	Гистерезис толкателей

VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
InSync	Применяется таблица кулачков	BOOL	TRUE/FALSE	FALSE	TRUE указывает на синхронизацию ведомой оси со шпинделем на основе таблицы кулачков
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
EndOfProfile	Кривая построена	BOOL	TRUE/FALSE	FALSE	Импульсный выход: Цикл построения контуров в CAM-системе завершен
Tappets	Таблица толкателей	SMC_TappetData	-	-	Для сигнала толкателей, обработанного с помощью SMC_GetTappetValue

(3) Описание функции

- Данная команда реализована на основе библиотеки "SM3_Basic".
- Эта команда позволяет выполнять синхронные операции с кулачками и является таблицей кулачков для фазы (шпиндель) и смещения (ведомая ось).

- Существуют два способа составления таблицы кулачков, задаваемой данной командой:

(1)Составление в редакторе кулачков

(2)Построение структуры данных таблицы кулачков методом программирования.

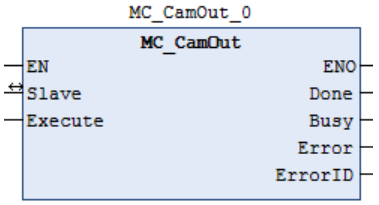
- В CAM-системе для вызова кривой характеристик кулачков сначала следует вызвать команду MC_CamTableSelect, чтобы выбрать соответствующую таблицу кулачков, а затем следует выполнить команду MC_CamIn. Для замены кривой характеристик кулачков повторно вызовите команду MC_CamTableSelect, чтобы снова выбрать таблицу кулачков.
- Используйте команду MC_CamOut, чтобы разблокировать соединение между главной и ведомой осями с помощью кулачка.
- После выполнения этой команды связь между ведомой осью и шпинделем через кулачок будет разблокирована, когда ведомая ось команды выполнит другие команды на перемещение, а переменная Command-Aborted команды MC_CamIn примет состояние TRUE.

3-3-2-3. Разблокировка связи через кулачок [MC_CamOut]

(1) Обзор команды

Разблокировка связи между указанной ведомой осью и соответствующим шпинделем через кулачок.

Разблокировка связи через кулачок [MC_CamOut]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_CamOut	Разблокировка связи через кулачок		MC_CamOut_0(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	TRUE после завершения выполнения команды
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Данная команда реализована на основе библиотеки "SM3_Basic".
- Эта команда используется для разблокировки связи между указанной ведомой осью и соответствующим шпинделем через кулачок.
- Связь ведомой оси через кулачок разблокируется при нарастающем фронте сигнала Execute.
- После разблокировки связи через кулачок ведомая ось может не остановиться. Если перед выполнением этой команды скорость ведомого вала не равна 0, то после поступления сигнала DONE данной команды связь

через кулачок будет разблокирована, но ведомая ось продолжит работать с вышеупомянутой скоростью до отключения.

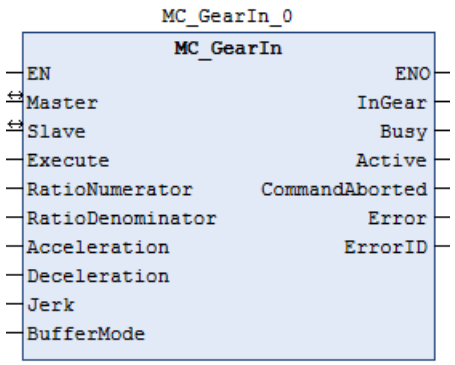
- Если команда выполняется без связи ведомой оси через кулачок, возникает ошибка ERROR.

3-3-2-4. Связь через кулачок [MC_GearIn]

(1) Обзор команды

Настройка передаточного числа главной оси и ведомой оси и включение электронных редукторов.

Связь через кулачок [MC_GearIn]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_GearIn	Связь через кулачок		MC_GearIn_0(Master:= , Slave:= , Execute:= , RatioNumerator:= , RatioDenominator:= , Acceleration:= , Deceleration:= , Jerk:= , BufferMode:= , InGear=> , Busy=> , Active=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к ведомой оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
RatioNumerator	Числитель передаточного числа	DINT	положительное число	1	Смещение по таблице главной оси
RatioDenominator	Знаменатель передаточного числа	UDINT	положительное число	1	Смещение по таблице ведомой оси
Acceleration	Заданное ускорение	LREAL	положительное число	0	ускорение [импульсов/с^2]
Deceleration	Заданное торможение	LREAL	положительное число	0	торможение [импульсов/с^2]
Jerk	Заданная скорость в толчковом режиме	LREAL	положительное число	0	скорость в толчковом режиме [импульсов/с^3]
BufferMode	Буферный режим	MC_BUFFER_MODE	-	-	-
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
InGear	Действующее передаточное число	BOOL	TRUE/FALSE	FALSE	Ведомая станция перемещается с заданным отношением скоростей главной станции, а заданная скорость (Target speed),

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					достигнутая ведомой осью, равна TRUE.
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Связь через кулачок при нарастающем фронте сигнала Execute
- Для разблокировки после завершения работы электронного редуктора нужно выполнить команду GearOut.
- Эта команда представляет собой функцию электронного редуктора скорости, и потеря расстояния синхронизации, возникающая в процессе ускорения, не будет автоматически компенсироваться.
- Если во время выполнения команды сигнал Busy принимает состояние TRUE, повторный запуск по нарастающему фронту сигнала Execute не приведет к перезапуску команды независимо от того, была ли достигнута заданная скорость ведомой оси.

- При достижении заданной скорости переменная InGear примет состояние TRUE, при этом величина перемещения ведомой оси будет соответствовать величине перемещения шпинделя * RatioNumerator/RatioDenominator.
- Если скорость вращения шпинделя изменяется в режиме реального времени, используйте эту команду с осторожностью.

Примечание: Не используйте команду MC_SetPosition во время выполнения других команд, чтобы исключить несчастные случаи, вызванные быстрой работой двигателя.

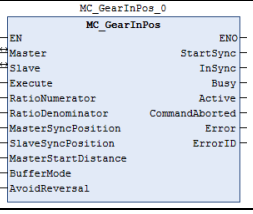
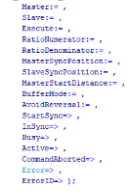
3-3-2-5. Работа редуктора при заданных положениях [MC_GearInPos]

(1) Обзор команды

Настройка передаточного числа электронного редуктора между главной и ведомой осями, чтобы обеспечить работу электронного редуктора.

Задайте положение шпинделя и положение ведомой оси в начале синхронизации, а также расстояние при начальной синхронизации шпинделя и используйте введенные значения для завершения запуска электронного редуктора.

Работа редуктора при заданных положениях [MC_GearInPos]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_GearInPos	Работа редуктора при заданных положениях		

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к ведомой оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
RatioNumerator	Числитель передаточного числа	DINT	положительное число	1	Смещение главной оси
RatioDenominator	Знаменатель передаточного числа	UDINT	положительное число	1	Смещение ведомой оси
MasterSyncPosition	Положение для синхронизации шпинделя	LREAL	Отрицательное число, 0, положительное число	0	Положение для синхронной работы главной оси
SlaveSyncPosition	Положение для синхронизации ведомой оси	LREAL	Отрицательное число, 0, положительное число	0	Положение для синхронной работы ведомой оси
MasterStartDistance	Выполнение синхронизации положения шпинделя	LREAL	Отрицательное число, 0, положительное число	0	Основное расстояние между зубчатыми передачами в программе (запуск ведомой оси для входа в

					режим синхронизации)
BufferMode	Буферный режим	MC_BUFFER_MODE	-	-	-
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
AvoidReversal	Запрет обратного хода	BOOL	TRUE/FALSE	FALSE	FALSE: Указывает на то, что обратный ход ведомой оси физически возможен и допустим. TRUE: Указывает на то, что обратный ход ведомой оси физически невозможен. В противном случае это может привести к опасной ситуации. Касается только модульных приводных осей. Если обратного хода избежать невозможно, ось остановится из-за ошибки.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

StartSync	Начало синхронизации	BOOL	TRUE/FALSE	FALSE	True, если электронный редуктор начинает обработку
InSync	Синхронизация завершена	BOOL	TRUE/FALSE	FALSE	Команда электронного редуктора выполнена
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Active	В управляемом состоянии	BOOL	TRUE/FALSE	FALSE	Изменение на TRUE в управляемом состоянии
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - TRUE
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Данная команда реализована на основе библиотеки "SM3_Basic".
- После запуска оборудования заданная скорость ведомой оси будет равна скорости, полученной умножением скорости шпинделя на передаточное число, и будут выполняться операции ускорения и торможения.
- Процесс от начала синхронизации до конца синхронизации в этом функциональном блоке представляет собой, по сути, электронный кулачок, с помощью которого ведомая ось следует за шпинделем в пределах интервала синхронизации. При этом, учитывая диапазон параметров шпинделя

(MasterSyncPosition-MasterStartDistance, MasterSyncPosition) и диапазон характеристик ведомой оси (текущее положение, SlaveSyncPosition), команда автоматически сгенерирует кривую характеристик кулачка на основе заданного передаточного числа и трех указанных выше параметров. При выполнении синхронизации ведомая ось будет следовать за шпинделем, чтобы завершить перемещение кулачка.

- Если главная ось и ведомая ось работают в линейном режиме, следует убедиться, что указанные выше параметры заданы правильно, в противном случае редуктор будет работать неправильно. По этой причине, рекомендуется использовать эту команду, когда главная ось и ведомая ось находятся в режиме оси модуля. Например, в линейных режимах работы главная ось и ведомая ось перемещаются в прямом направлении. Если положение главной оси MasterSyncPosition-MasterStartDistance или положение ведомой оси>SlaveSyncPosition, то электронный редуктор не включится.

- Одновременно с завершением синхронизации (InSync в состоянии true) достигается заданная скорость, и тогда величина перемещения ведомой оси будет соответствовать величине перемещения шпинделя (RatioNumerator/RatioDenominator).

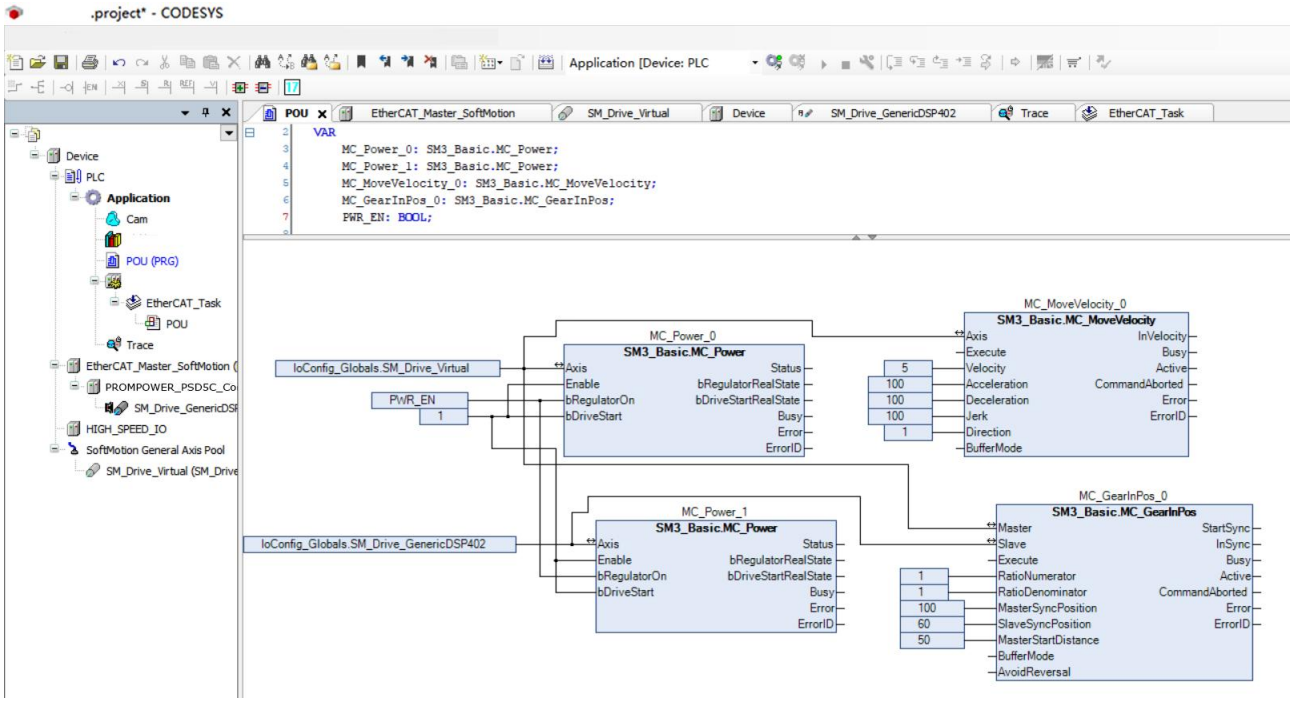
- Для AvoidReversal: Если ведомая ось является модальной осью, а скорость шпинделя (кратное отношение передаточного числа) не относится к отношению скоростей ведомой оси, то с помощью команды MC_GearInPos можно попытаться предотвратить обратный ход ведомой оси. Данная команда пытается «растянуть» перемещение ведомой оси, добавляя 5 циклов ведомой оси. Если это «растяжение» не сработает, возникнет ошибка, и ведомая ось остановится из-за ошибки. Если скорость ведомой оси связана со скоростью шпинделя (которая кратна передаточному числу), возникнет ошибка, и ось остановится из-за ошибки. Если ведомая ось является осью линейного режима, ошибка возникнет при нарастающем фронте сигнала Execute.

(4) Применение

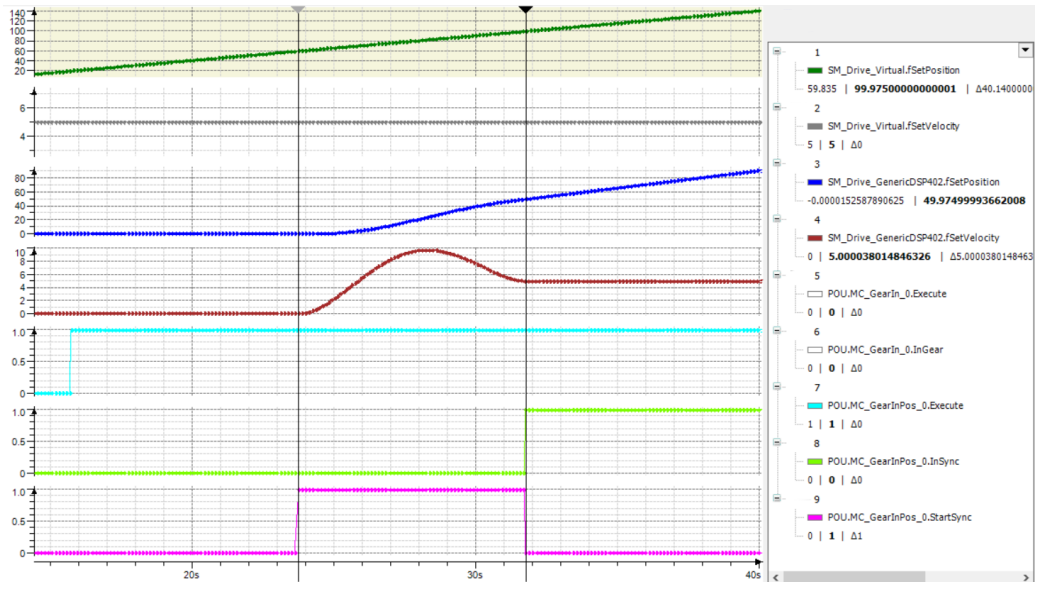
Пример 1: Настройка передаточного числа электронного редуктора между главной и ведомой осями, чтобы обеспечить работу электронного редуктора. Задайте положение шпинделя и положение ведомой оси в начале синхронизации, а также расстояние при начальной синхронизации шпинделя и используйте введенные значения для завершения запуска электронного редуктора. Выберите виртуальную ось для главной оси, после чего выберите действительную ось Ethercat для ведомой оси. Установите скорость главной оси на 5 импульсов/с, а также положительное направление.

(1) Программирование: Начальные положения обеих осей установлены на 0, шпиндель настроен на скорость 5 импульсов/с с помощью функционального

блока "MC_MoveVelocity", и задано положительное направление. С помощью команды "MC_GearInPos" для двух осей задается передаточное число электронного редуктора 1:1, положение для синхронизации шпинделя равно 100 импульсов, положение для синхронизации ведомой оси равно 50 импульсов, а расстояние при начальной синхронизации шпинделя равно 40 импульсов.



(2) Текущее состояние можно просматривать через пункт меню 'Trace'. Как показано на рисунке ниже, синхронизация начинается с 60 импульсов для положения шпинделя. После завершения синхронизации положение шпинделя составляет 100 импульсов, положение ведомой оси — 50 импульсов, а скорость ведомой оси — 5 импульсов/с.

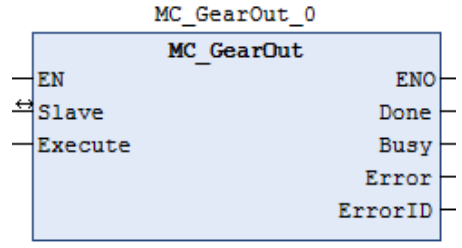


3-3-2-6. Отключение редуктора [MC_GearOut]

(1) Обзор команды

Отключение муфты электронного редуктора между ведомой осью и шпинделем.

Отключение редуктора [MC_GearOut]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_GearOut	Отключение редуктора		MC_GearOut_0(Slave:= , Execute:= , Done=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если ведомая ось отключена от

					электронного редуктора шпинделя
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

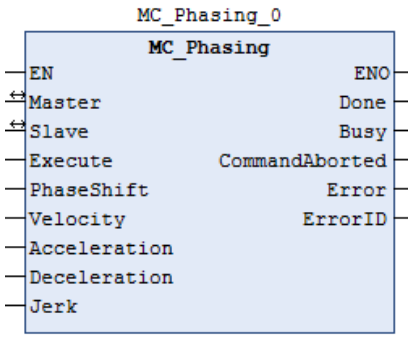
- Данная команда реализована на основе библиотеки "SM3_Basic".
- После отключения электронного редуктора скорость ведомой оси будет равна скорости до отключения, поэтому необходимо ввести команду MC_Stop, чтобы остановить ведомую ось.
- Задайте ось объекта действия через Slave, задайте торможение (Deceleration) и прервите выполнение команды MC_GearIn (включение редуктора), команды MC_ GearInPos (работа редуктора при заданных положениях).
- Эта команда не влияет на команды MC_GearIn и MC_GearInPos, относящиеся к главной оси.

3-3-2-7. Сдвиг фаз между главной осью и ведомой осью [MC_Phasing]

(1) Обзор команды

Задает сдвиг фаз между главной осью и ведомой осью.

Сдвиг фаз между главной осью и ведомой осью [MC_Phasing]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
MC_Phasing	Сдвиг фаз между главной осью и ведомой осью		MC_Phasing_0(Master:= , Slave:= , Execute:= , PhaseShift:= , Velocity:= , Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=> , CommandAborted=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
PhaseShift	Компенсация сдвига фаз	LREAL	Отрицательное число, 0, положительное число	0	Настройка компенсации сдвига фаз для шпинделя
Velocity	скорость	LREAL	0, положительное число	0	Максимальная скорость при сдвиге фаз [командная единица/с]
Acceleration	ускорение	LREAL	0, положительное число	0	Максимальное ускорение при сдвиге фаз

					[командная единица/c^2]
Deceleration	торможение	LREAL	0, положительное число	0	Максимальное торможение при сдвиге фаз [командная единица/c^2]
Jerk	Скорость в толчковом режиме	LREAL	0, положительное число	0	Максимальная скорость в толчковом режиме при сдвиге фаз [командная единица/c^3]
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Done	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если ведомая ось отключена от электронного редуктора шпинделя
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
CommandAborted	Команда прервана	BOOL	TRUE/FALSE	FALSE	Выполнение модуля прервано - true
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Сдвиг фаз при нарастающем фронте сигнала Execute; ведомая ось автоматически вычисляет сглаженную кривую для завершения сдвига фаз от

ведомой оси к главной оси. Разность фаз между главной осью и ведомой осью — это значение PhaseShift входного сигнала, а положительное значение — это отставание ведомой оси от главной оси.

- После завершения сдвига выход сигнала Done примет состояние true.
- Данная функция компенсирует разность фаз между главной осью и ведомой осью на основе заданных значений PhaseShift, Velocity, Acceleration и Deceleration.
- Когда разность фаз между главной осью и ведомой осью достигнет значения PhaseShift, будет подан сигнал Done.
- Во время выполнения этой команды заданное положение шпинделя и обратная связь по положению останутся неизменными, а ведомая ось будет отрегулирована. После завершения выполнения команды разность фаз между ведомой осью и шпинделем равна значению PhaseShift. Конечным результатом выполнения этой команды является сдвиг фаз между заданными значениями оси, поэтому фактическое значение обратной связи по действительной оси может не совпадать с конечным сдвигом.

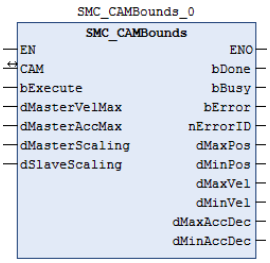
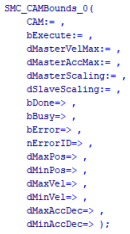
3-3-2-8. Диапазон характеристик кулачков [SMC_CAMBounds]

(1) Обзор команды

Если ведомая ось связана с кулачком шпинделя, то с помощью этого функционального блока можно рассчитать максимальное положение, скорость и ускорение ведомой оси.

Шпиндель перемещается с максимальной скоростью, при этом действуют ограничения на ускорение/торможение. Эта команда позволяет проверить правильность кривой характеристик при составлении таблицы кулачков, если известны максимальное ускорение, торможение, скорость и другие характеристики шпинделя.

Диапазон характеристик кулачков [SMC_CAMBounds]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_CAMBounds	Диапазон характеристик кулачков		

(2) Связанные переменные

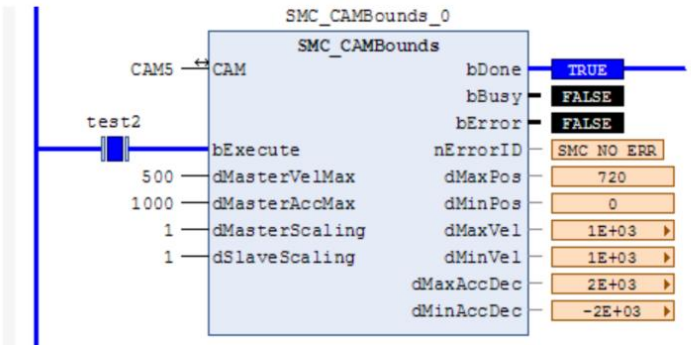
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Cam	КУЛАЧОК	MC_CAM_REF	-	-	Привязка к кулачку
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bExecute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом, выполнение текущей команды
dMasterVelMax	Макс. скорость	LREAL	-	1	Максимальная скорость вращения шпинделя в абсолютном режиме
dMasterAccMax	Макс. ускорение	LREAL	-	0	Максимальное ускорение шпинделя в абсолютном режиме
dMasterScaling	Коэффициент масштабирования	LREAL	-	1	Коэффициент масштабирования для применяемого кулачка шпинделя

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
dSlaveScaling	Коэффициент масштабирования	LREAL	-	1	Коэффициент масштабирования для применяемого кулачка ведомой оси
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bDone	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если ведомая ось отключена от электронного редуктора шпинделя
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
bError	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
dMaxPos	Макс. положение	LREAL	-	0	Вычисление максимального положения ведомой оси на основе таблицы кулачков
dMinPos	Мин. положение	LREAL	-	0	Вычисление минимального положения ведомой оси на основе таблицы кулачков

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
dMaxVel	Макс. скорость	LREAL	-	0	Вычисление максимальной скорости ведомой оси
dMinVel	Мин. скорость	LREAL	-	0	Вычисление минимальной скорости ведомой оси
dMaxAccDec	Макс. ускорение	LREAL	-	0	Вычисление максимального ускорения ведомой оси
dMinAccDec	Мин. ускорение	LREAL	-	0	Вычисление минимального ускорения ведомой оси

(3) Описание функции

- По нарастающему фронту сигнала bExecute значения входных переменных "dMasterVelMax", "dMasterAccMax", "dMasterScaling" и "dSlaveScaling" объединяются с данными таблицы кулачков для расчета эквивалентного значения «максимального положения» и «минимального положения» ведомой оси. Например, если цикл вращения шпинделя равен 360, а таблица кулачков представляет собой прямую линию с наклоном 2, то результат вычислений будет таким, как показано на рисунке ниже:



- Характеристики шпинделя можно рассчитать с помощью этой команды при работе в абсолютном режиме, или когда шпиндель настроен на режим цикла, а модуль настроен на цикл шпинделя.

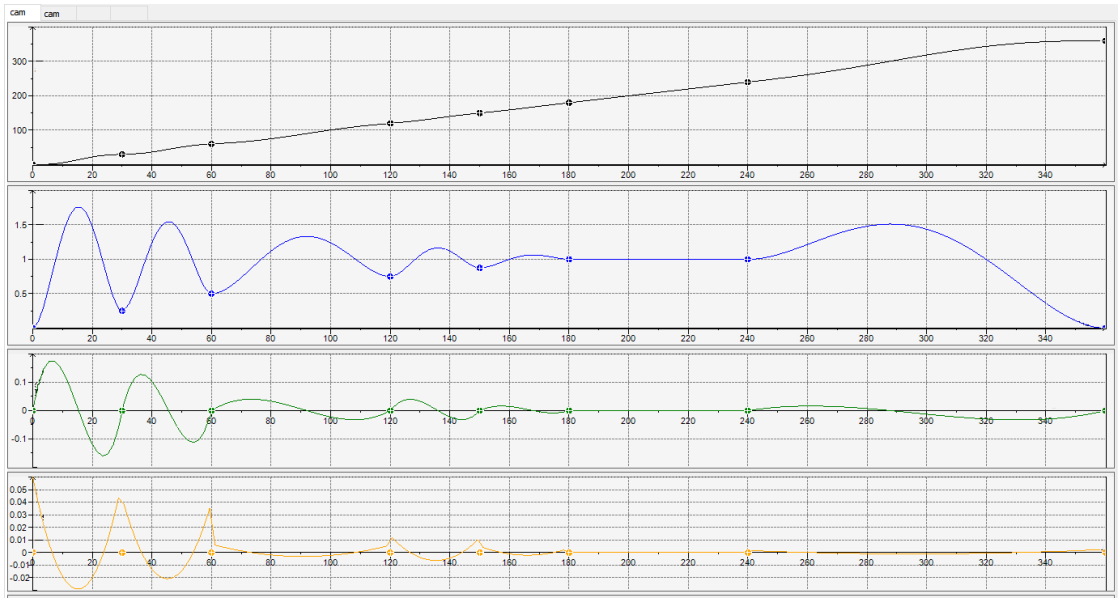
- Таблица кулачков имеет вид XYVA (допустима в полиномиальном режиме), а одномерные массивы, двумерные массивы и т. д. недопустимы.

(4) Применение

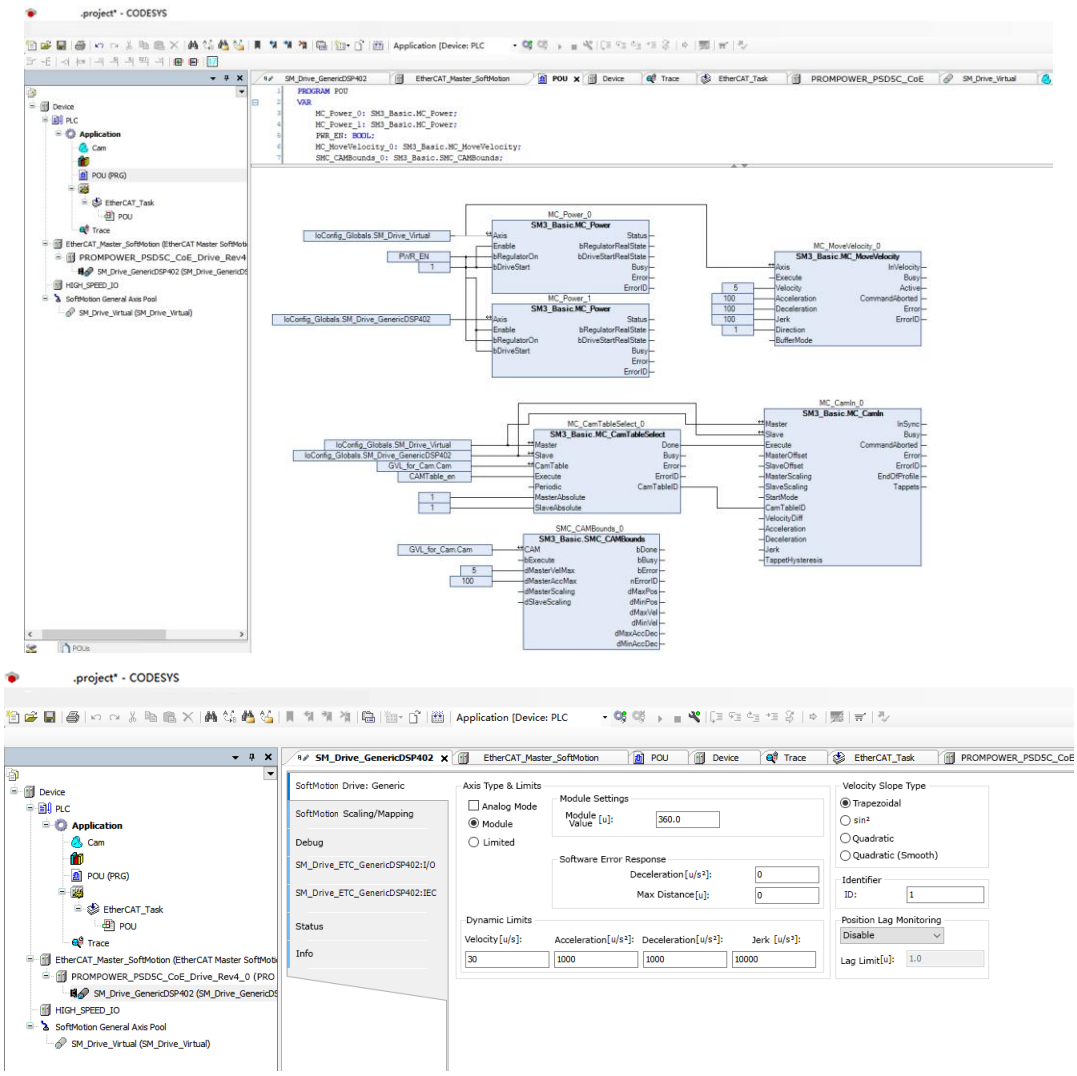
Пример: Создайте таблицу электронных кулачков и рассчитайте максимальное положение, скорость, ускорение/торможение ведомой оси через команду "SMC_CAMBounds" после соединения ведомой оси с кулачком шпинделя. Выберите виртуальную ось для главной оси, после чего выберите действительную ось Ethercat для ведомой оси.

Программа:

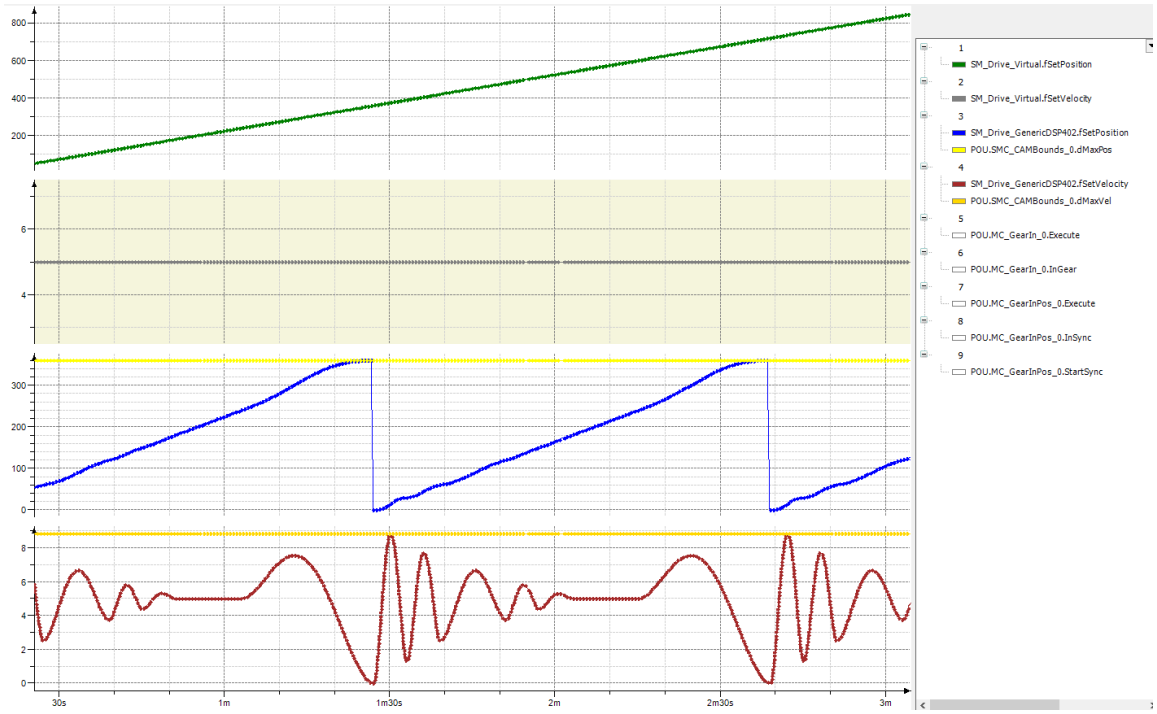
Щелкните правой кнопкой мыши по "Application" (Применение) на панели устройств проекта, выберите "Add Object" - "Cam Table" ("Добавить объект" - "Таблица кулачков"), присвойте ему имя Cam и откройте его, чтобы добавить соответствующие параметры таблицы кулачков, как показано на рисунке ниже.



Выберите таблицу кулачков с помощью команды MC_CamTableSelect, выберите абсолютный режим соединения (Absolute) и настройте электронный модуль кулачков MC_CamIn. Установите скорость вращения шпинделя на 5 и ускорение на 100 с помощью "MC-MoveVelocity" и рассчитайте максимальное положение, скорость и ускорение ведомой станции с помощью "SMC_CAMBounds". Главная станция работает синхронно с главной осью и ведомой осью и задает режим работы ведомой оси по модулю - в диапазоне от 0 до 360.



Контролируя положение и скорость ведомой оси через "Trace", можно убедиться, что максимальное положение и скорость совпадают во время работы.

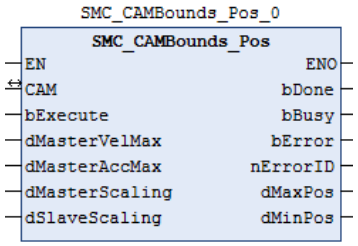


3-3-2-9. Диапазон положений при связи с кулачком [SMC_CAMBounds_Pos]

(1) Обзор команды

Если ведомая ось связана с кулачком шпинделя, то с помощью этого функционального блока можно рассчитать максимальное и минимальное положения ведомой оси. В отличие от SMC_CAMBounds, в этом функциональном блоке отсутствуют такие вычисления, как максимальное ускорение, при этом все остальные рабочие характеристики соответствуют требованиям.

Диапазон положений при связи с кулачком [SMC_CAMBounds_Pos]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_CAMBounds_Pos	Диапазон положений при связи с кулачком		SMC_CAMBounds_Pos_0(CAM:= , bExecute:= , dMasterVelMax:= , dMasterAccMax:= , dMasterScaling:= , dSlaveScaling:= , bDone=> , bBusy=> , bError=> , nErrorID=> , dMaxPos=> , dMinPos=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Cam	КУЛАЧОК	MC_CAM_REF	-	-	Привязка к кулачку
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bExecute	Действительн.	BOOL	TRUE/FALSE	FALSE	Сигнал с нарастающим фронтом,

					выполнение текущей команды
dMasterVelMax	Макс. скорость	LREAL	-	1	Максимальная скорость вращения шпинделя в абсолютном режиме
dMasterAccMax	Макс. ускорение	LREAL	-	0	Максимальное ускорение шпинделя в абсолютном режиме
dMasterScaling	Коэффициент масштабирования	LREAL	-	1	Коэффициент масштабирования для применяемого кулачка шпинделя
dSlaveScaling	Коэффициент масштабирования	LREAL	-	1	Коэффициент масштабирования для применяемого кулачка ведомой оси
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bDone	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если ведомая ось отключена от электронного редуктора шпинделя
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
bError	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении

					функционального блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
dMaxPos	Макс. положение	LREAL	-	0	Вычисление максимального положения ведомой оси на основе таблицы кулачков
dMinPos	Мин. положение	LREAL	-	0	Вычисление минимального положения ведомой оси на основе таблицы кулачков

(3) Описание функции

- По нарастающему фронту сигнала bExecute значения входных переменных "dMasterVelMax", "dMasterAccMax", "dMasterScaling" и "dSlaveScaling" объединяются с данными таблицы кулачков для расчета значения «максимального положения» и «минимального положения» ведомой оси.
- Характеристики шпинделя можно рассчитать с помощью этой команды при работе в абсолютном режиме, или когда шпиндель настроен на режим цикла, а модуль настроен на цикл шпинделя.
- Таблица кулачков имеет вид XYVA (допустима в полиномиальном режиме), а одномерные массивы, двумерные массивы и т. д. недопустимы.

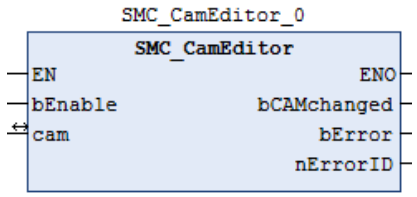
3-3-2-10. Визуализация таблицы кулачков [SMC_CamEditor]

(1) Обзор команды

Этот функциональный блок используется для визуализации таблицы кулачков. Этот функциональный блок не работает без TargetVideo или WebVideo.

Визуализация таблицы кулачков [SMC_CamEditor]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx

Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_CamEditor	Визуализация таблицы кулачков		SMC_CamEditor_0(bEnable:= , bCAMchanged=> , bError=> , nErrorID=> , cam:=);

(2) Связанные переменные

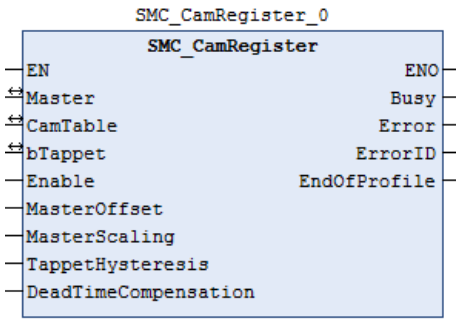
VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Cam	КУЛАЧОК	MC_CAM_REF	-	-	Привязка к кулачку
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bEnable	Действительн.	BOOL	TRUE/FALSE	FALSE	Если TRUE, то в таблицу кулачков вносятся изменения
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bCAMchanged	Выполнено	BOOL	TRUE/FALSE	FALSE	Внешний вид кулачка на экране изменился. Значит, состояние True
bError	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

3-3-2-11. Управление толкателями кулачков [SMC_CamRegister]

(1) Обзор команды

Реализация функции управления толкателями кулачков (кулачковый выключатель). При редактировании таблицы кулачков можно не редактировать кривые характеристик главной и ведомой осей, а можно просто настроить таблицу толкателей для управления толкателями через этот функциональный блок.

Управление толкателями кулачков [SMC_CamRegister]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_CamRegister	Управление толкателями кулачков		SMC_CamRegister_0(Master:= , CamTable:= , bTappet:= , Enable:= , MasterOffset:= , MasterScaling:= , TappetHysteresis:= , DeadTimeCompensation:= , Busy=> , Error=> , ErrorID=> , EndOfProfile=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси
CamTable	Таблица кулачков	MC_CAM_REF	-	-	Привязка к электронному кулачку, т.е. к экземпляру

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					электронного кулачка
bTappet	Выход толкателя	ARRAY [1..MAX_NUM_TAPPETS]OF BOOL	-	-	Выход точки толкателя
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Enable	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды - TRUE
MasterOffset	Положение главной оси	LREAL	-	0	Смещение по таблице главной оси
MasterScaling	Масштабирование главной оси	LREAL	-	1	Коэффициент масштабирования файла конфигурации главной оси
TappetHysteresis	Гистерезис толкателей	LREAL	-	-	Размер гистерезиса толкателя
DeadTimeCompensation	Компенсация времени нечувствительности	LREAL	-	0	Компенсация времени нечувствительности в секундах. Ожидаемое положение шпинделя рассчитывается методом линейной экстраполяции
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функциональн ого блока еще не завершено
Error	Ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функциональн ого блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
EndOfProfile	Завершение кривого цикла	BOOL	TRUE/FALSE	FALSE	True, если положение шпинделя превышает или равно заданному циклу

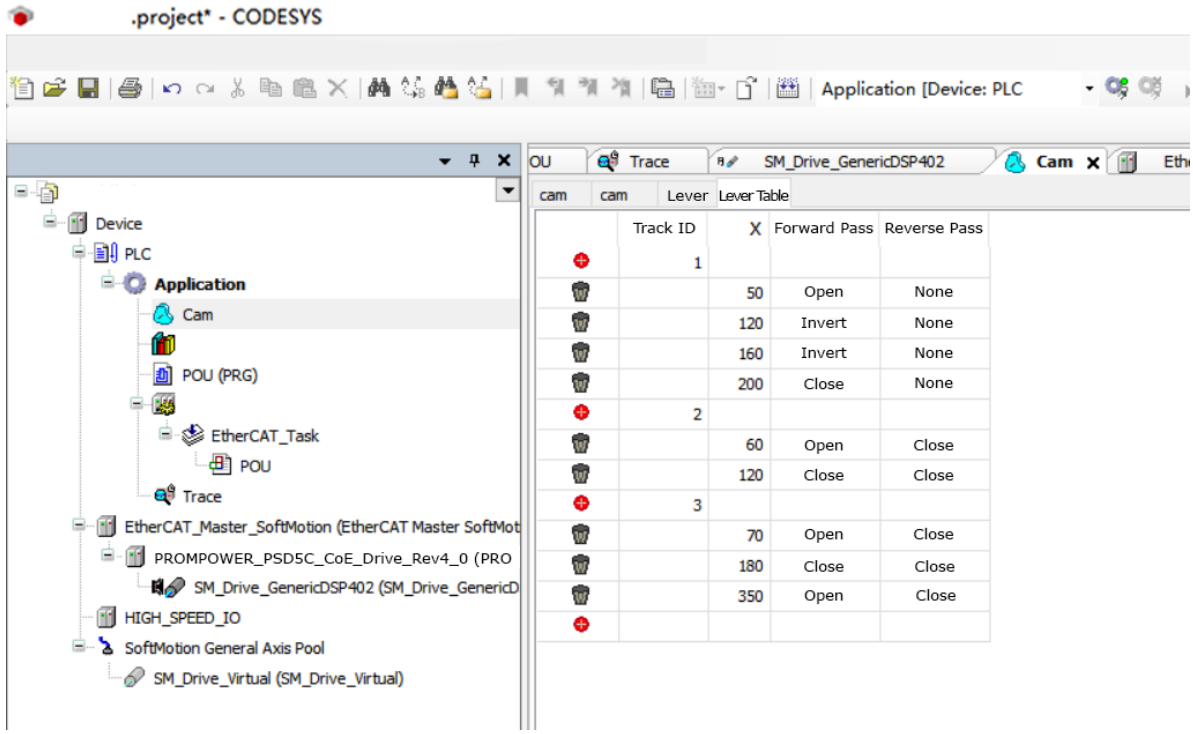
(3) Описание функции

- Сигнал Enable имеет состояние TRUE, а при отсутствии ошибок выход Busy имеет состояние TRUE для управления толкателями.
- Этот функциональный блок управления не связан с ведомой осью в электронном кулачке, и необходимо настроить только цикл шпинделя и таблицу толкателей.
- bTappet — одномерная булева структура (MAX_NUM_TAPPETS=512), а bTappet[i] соответствует выходу DeadTimeCompensation для i-го толкателя; единица измерения - секунда. При положительном значении происходит опережение сигнала толкателя, а при отрицательном - отставание от сигнала.

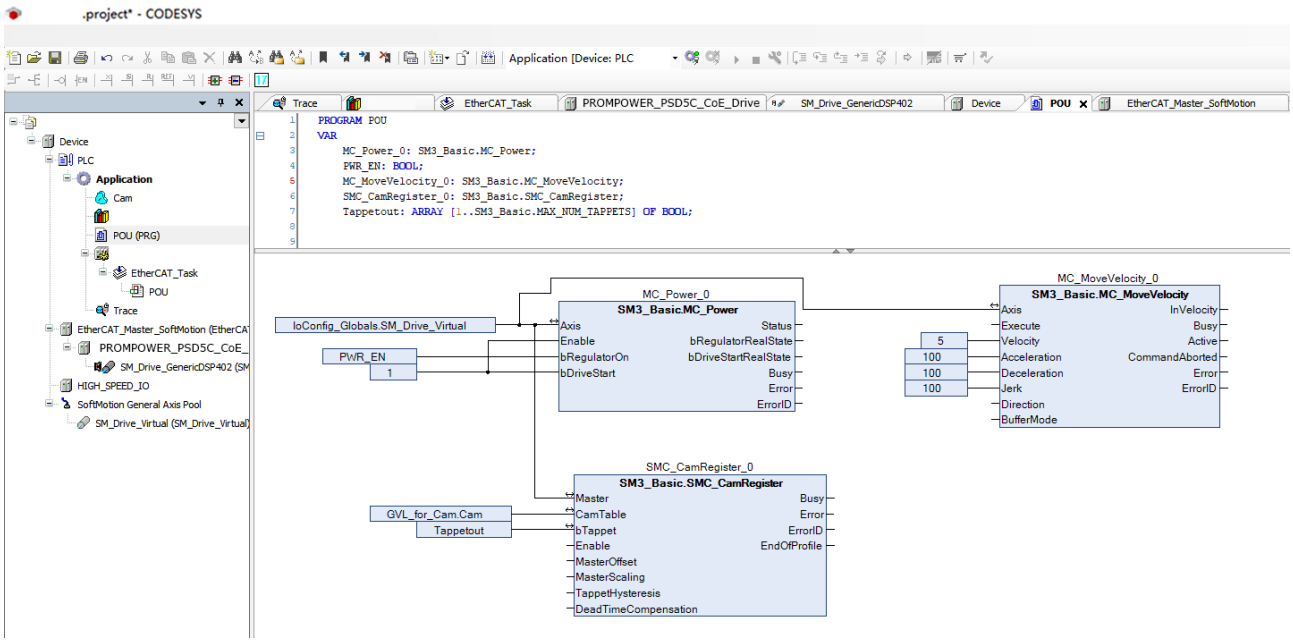
(4) Применение

Пример: Введите 3 толкателя в таблицу толкателей в составе таблицы кулачков. Установите скорость вращения шпинделя на 5, ускорение — на 100, и проведите шпиндель по траектории, заданной в таблице кулачков. Главная ось выбирается в качестве виртуальной оси, а ведомая — в качестве действительной оси.

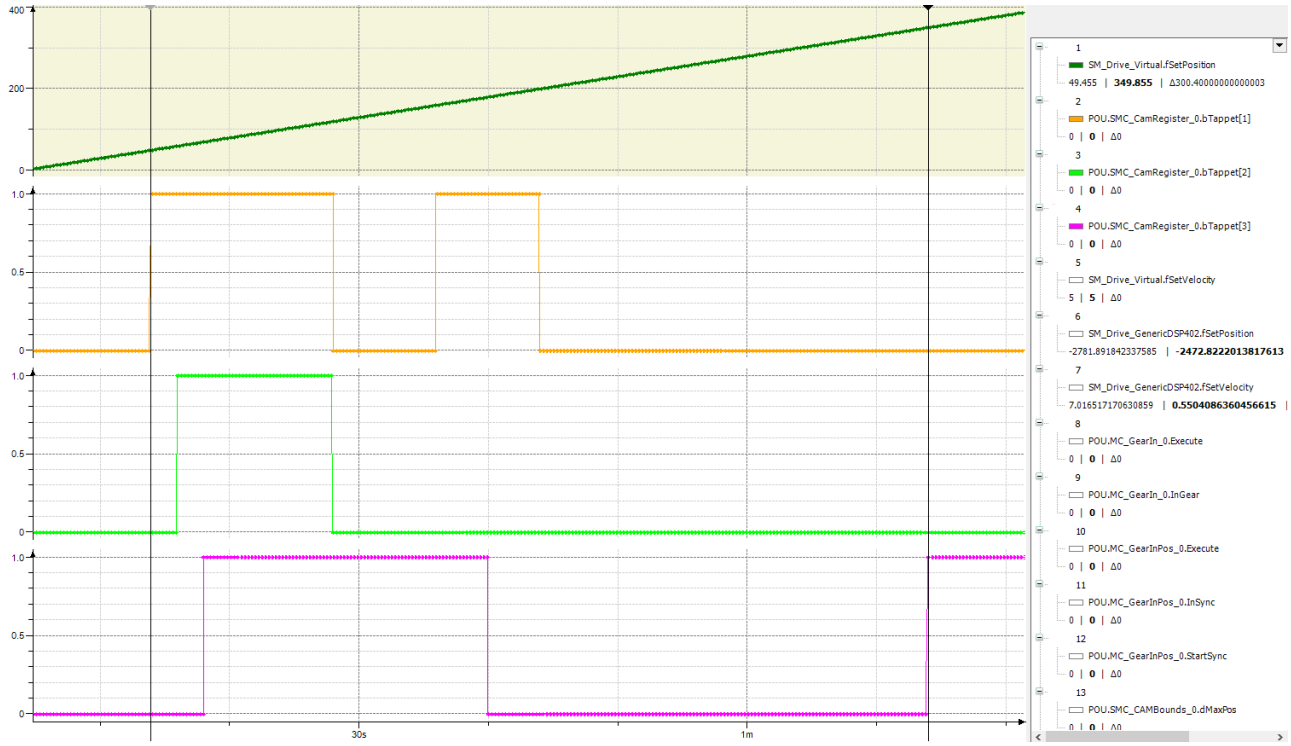
Щелкните правой кнопкой мыши по "Application" (Применение) на панели устройств проекта, выберите "Add Object" - "Cam Table" ("Добавить объект" - "Таблица кулачков"), присвойте ему имя Cam и откройте его, чтобы добавить соответствующие параметры толкателей, как показано на рисунке ниже.



Добавьте модуль включения MC_ Power в главную программу, а также модуль скорости вращения шпинделя MC_MoveVelocity. С помощью функционального блока "SMC_CamRegister" можно контролировать состояние вывода всех идентификаторов в таблице толкателей.



Через пункт 'Trace' можно просматривать все текущие выходные состояния толкателя шпинделя.

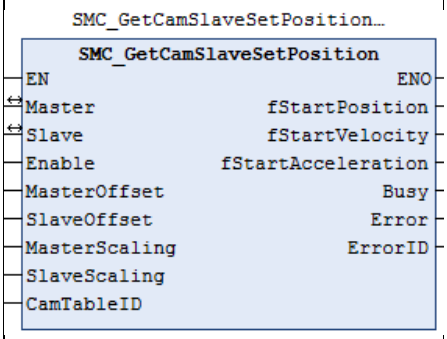


3-3-2-12. Считывание положения ведомой оси из таблицы кулачков [SMC_GetCamSlaveSetPosition]

(1) Обзор команды

Считывание информации о положении, скорости и ускорении ведомой оси из таблицы кулачков.

Считывание положения ведомой оси из таблицы кулачков [SMC_GetCamSlaveSetPosition]			
Условие выполнения	Нормальное включение/выключение триггера	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_GetCamSlaveSetPosition	Считывание положения ведомой оси из таблицы кулачков		SMC_GetCamSlaveSetPosition_0(Master:= , Slave:= , Enable:= , MasterOffset:= , SlaveOffset:= , MasterScaling:= , SlaveScaling:= , CamTableID:= , fStartPosition=> , fStartVelocity=> , fStartAcceleration=> , Busy=> , Error=> , ErrorID=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Master	Главная ось	AXIS_REF	-	-	Привязка к главной оси
Slave	Ведомая ось	AXIS_REF	-	-	Привязка к ведомой оси
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
Execute	Действительн.	BOOL	TRUE/FALSE	FALSE	Выполнение текущей команды по сигналу с нарастающим фронтом
MasterOffset	Смещение главной оси	LREAL	-	0	Смещение по таблице главной оси
SlaveOffset	Смещение ведомой оси	LREAL	-	0	Смещение по таблице ведомой оси
MasterScaling	Масштабирование главной оси	LREAL	-	1	Коэффициент масштабирования файла конфигурации главной оси

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
SlaveScaling	Ведомая ось	LREAL	-	1	Коэффициент масштабирования файла конфигурации ведомой оси
CamTableID	Таблица кулачков	MC_CAM_ID	-	-	Задание режима работы кулачка, используется вместе с MC_CamTable Выбор выходной точки CamTableI
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
fStartPosition	Положение ведомой оси	LREAL	-	0	Положение ведомой оси, определенное из таблицы кулачков и на основе имеющейся информации о шпинделе
fStartVelocity	Скорость ведомой оси	LREAL	-	0	Скорость ведомой оси, определенная из таблицы кулачков и на основе имеющейся информации о шпинделе
fStartAcceleration	Ускорение ведомой оси	LREAL	-	0	Ускорение ведомой оси, определенное из таблицы кулачков и на основе имеющейся информации о шпинделе

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Busy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
Error	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
ErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Выходное значение рассчитывается по этой команде следующим образом: Y= (cam((Cam — начальное положение поиска шпинделя кулачка +Masteroffset)* MasterScaling)+slaveoffset)* SlaveScaling, Cam — функция таблицы кулачков. Например, если начальное положение шпинделя кулачка равно 0, главная ось и ведомая ось масштабируются до 1, Masteroffset равно 100, а slaveoffset равно 0, то функциональный блок выдает положение ведомой оси, соответствующее значению в таблице кулачков при 100.
- Для этого функционального блока требуется только успешное составление таблицы кулачков для считывания положения ведомой оси и не требуются операции с главной осью и ведомой осью.

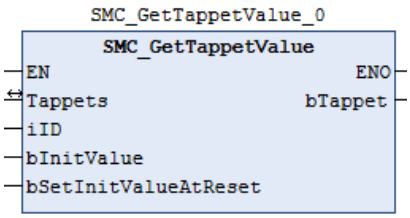
3-3-2-13. Считывание выходного значения для толкателей [SMC_GetTappetValue]

(1) Обзор команды

Для считывания текущего состояния толкателей следует использовать команду MC_CamIn.

Считывание выходного значения для толкателей [SMC_GetTappetValue]			
Условие выполнения	-	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-

Библиотека	SM3_Basic	

Команда	Название	Графическое представление	Представление на языке ST
SMC_GetTappetValue	Считывание выходного значения для толкателей		SMC_GetTappetValue_0(Tappets:= , iID:= , bInitValue:= , bSetInitValueAtReset:= , bTappet=>);

(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
Tappet	Толкатель	SMC_Tappet Data	-	-	Привязка к толкателю
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
iID	Идентификатор толкателя	INT	положительное значение, 0	0	Идентификатор толкателя
bInitValue	Начальное значение	BOOL	TRUE/FALSE	FALSE	Начальное значение толкателя, присваиваемое при первом вызове
bSetInitValueAtReset	Сброс толкателя	BOOL	TRUE/FALSE	FALSE	В состоянии true при перезапуске функционального блока CamIn выходное значение толкателя будет соответствовать начальному значению. В состоянии FALSE при перезапуске функциональног

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
					о блока CamIn значение толкателя будет сохранено.
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bTappet	Значение толкателя	BOOL	TRUE/FALSE	FALSE	Значение толкателя

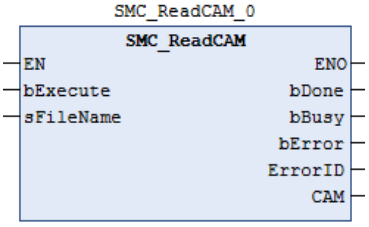
(3) Описание функции

- Этот функциональный блок должен использоваться с командой MC_CamIn.

3-3-2-14. Считывание данных из таблицы кулачков [SMC_ReadCAM]

(1) Обзор команды

Считывание данных из таблицы кулачков [SMC_ReadCAM]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_ReadCAM	Считывание данных из таблицы кулачков		SMC_ReadCAM_0(bExecute:= , sFileName:= , bDone=> , bBusy=> , bError=> , ErrorID=> , CAM=>);

(2) Связанные переменные

VAR_INPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
bExecute	Выполнение	BOOL	TRUE/FALSE	FALSE	Выполнение функционального блока при

					нарастающем фронте
sFileName	Имя файла	STRING(255)	-	-	Имя файла, содержащего описание кулачка в формате ASCII
VAR_OUTPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bDone	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если данные из таблицы кулачков были считаны
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
bError	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок
КУЛАЧОК	-	MC_CAM_REF	STRUCT(nElements : = 0, nTappets : = 0)	-	-

(3) Описание функции

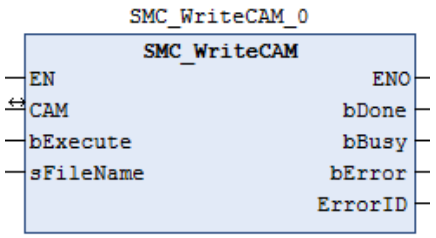
- Этот функциональный блок предназначен для загрузки кулачков во время выполнения программы и доступен для модулей MC_CamTableSelect и MC_CamIn. Загружаемый кулачок должен быть сначала создан в редакторе кулачков и сохранен в файле “*.CAM” (см. SMC_WriteCAM).
- Этот функциональный блок должен использоваться с командой MC_CamIn.
- Размер загружаемого кулачка ограничен глобальной константой gc_SMC_FILE_MAXCAMEL (количество элементов) и gc_SMC_FILE_ MAXCAMTAP (количество операций по переключению кулачка).

3-3-2-15. Внесение изменений в таблицу кулачков [SMC_WriteCAM]

(1) Обзор команды

Используется для сохранения таблицы кулачков, созданной в редакторе кулачков, в файл во время выполнения программы.

Внесение изменений в таблицу кулачков [SMC_WriteCAM]			
Условие выполнения	Запуск фронтом сигнала	Совместимые модели	PMP3xx
Прошивка	-	Программное обеспечение	-
Библиотека	SM3_Basic		

Команда	Название	Графическое представление	Представление на языке ST
SMC_WriteCAM	Внесение изменений в таблицу кулачков		SMC_WriteCAM_0(CAM:= , bExecute:= , sFileName:= , bDone=> , bBusy=> , bError=> , ErrorID=>);

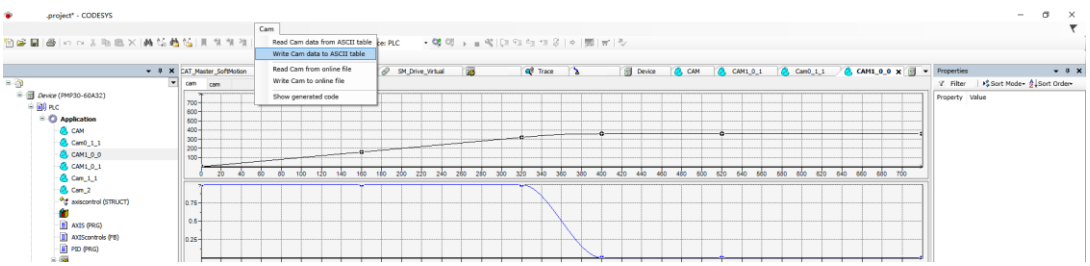
(2) Связанные переменные

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
CAM	Кулачок	MC_CAM_REF	-	-	Привязка к кулачку
VAR_INPUT	**Название**	**Тип данных**	**Эффективный диапазон**	**Начальное значение**	**Описание**
bExecute	Выполнение	BOOL	TRUE/FALSE	FALSE	Выполнение функционального блока при нарастающем фронте
sFileName	Имя файла	STRING(255)			Имя файла, содержащего описание кулачка в формате ASCII

VAR_IN_OUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
VAR_OUTPUT	Название	Тип данных	Эффективный диапазон	Начальное значение	Описание
bDone	Выполнено	BOOL	TRUE/FALSE	FALSE	True, если таблица кулачков записана в файл
bBusy	В процессе выполнения	BOOL	TRUE/FALSE	FALSE	True, если выполнение данного функционального блока еще не завершено
bError	ошибка	BOOL	TRUE/FALSE	FALSE	Ошибка при выполнении функционального блока
nErrorID	Код ошибки	SMC_ERROR	-	0	Обнаружение ошибок

(3) Описание функции

- Сохранение информации о кулачках, связанной с параметром “CAM”, в файле “sFileName” при нарастающем фронте сигнала bExecut.
- После успешного сохранения и выполнения выход сигнала Done примет состояние true.
- Объем сохраняемой таблицы кулачков ограничен памятью аппаратного обеспечения.
- Внимание: Эта функция может выполняться во время работы программы, а информацию из таблицы кулачков также можно сохранять вручную в автономном режиме.

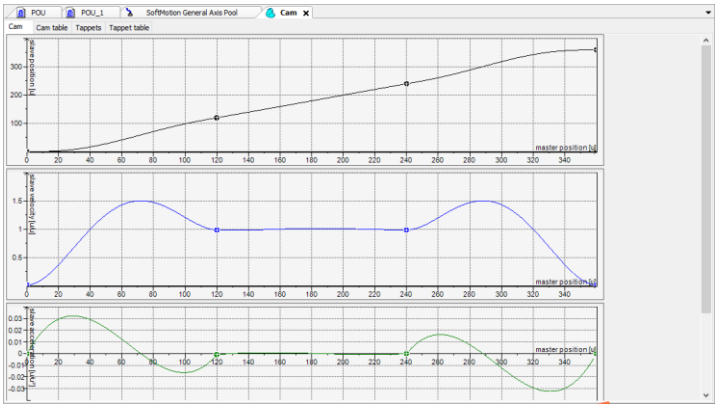


3-3-3. Применение функции для кулачков

Пример 1: Использование электронного кулачка для вращения шпинделя со скоростью 15 импульсов/с, ускорением 100 импульсов/с^2 в положительном направлении. Ведомая станция работает согласно таблице кулачков.

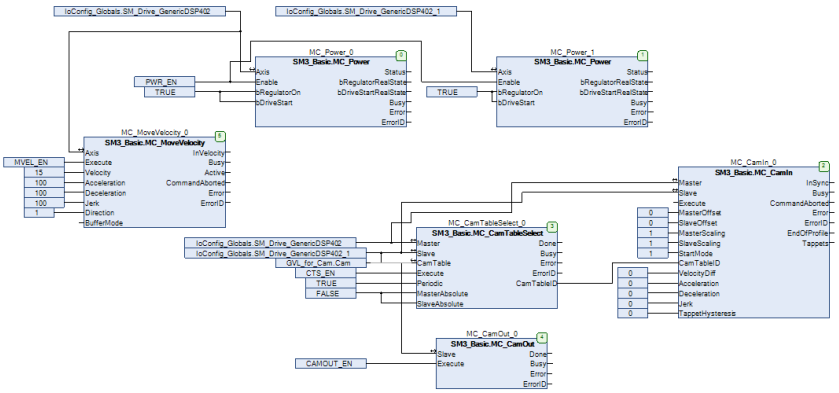
Работа с программой:

(1) Щелкните правой кнопкой мыши по "Application" (Применение) на панели инженерного оборудования, выберите "Add Object" - "Cam Table" ("Добавить объект" - "Таблица кулачков"), присвойте ему имя Cam и откройте его, чтобы добавить соответствующие параметры таблицы кулачков, как показано на рисунке ниже.



(2) Добавьте модуль включения MC_Power, модуль скорости вращения шпинделя MC_MoveVelocity в главную программу POU_1, вызвав модуль таблицы кулачков MC_CamTableSelect, настроив модуль электронных кулачков MC_CamIn и модуль отключения электронных кулачков MC_CamOut.

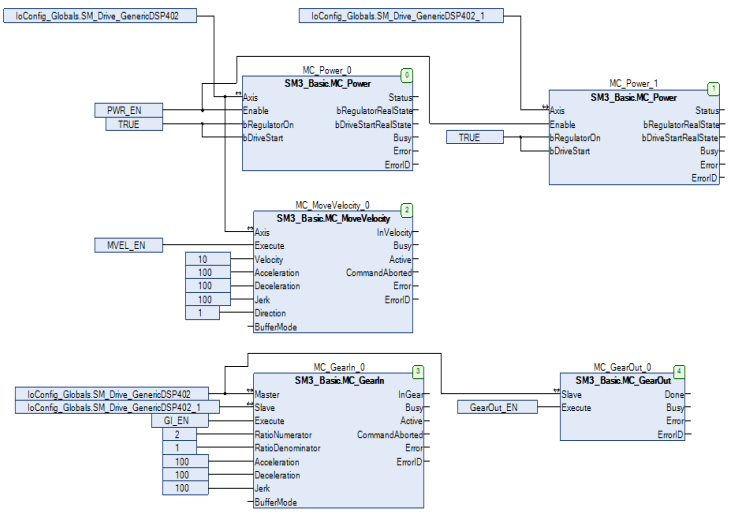
```
PROGRAM POU_1
VAR
    MC_Power_0: SM3_Basic.MC_Power;
    PWR_EN: BOOL;                  master slave enabled
    MC_Power_1: SM3_Basic.MC_Power;
    MC_CamIn_0: SM3_Basic.MC_CamIn;
    MC_CamTableSelect_0: SM3_Basic.MC_CamTableSelect;
    CTS_EN: BOOL;                  load the cam table
    MC_CamOut_0: SM3_Basic.MC_CamOut;
    CAMOUT_EN: BOOL;               unbinding cam
    MC_MoveVelocity_0: SM3_Basic.MC_MoveVelocity;
    MVEL_EN: BOOL;                 master axis speed mode
END_VAR
```



Примечание: Если во время выполнения CAMOUT ведомая ось перемещается, и после выполнения команды ведомая ось продолжает перемещение с первоначальной скоростью, остановите выполнение команды через команды MC_Stop, MC_Halt.

Пример 2: Использование электронных редукторов для вращения шпинделя со скоростью 15 импульсов/с, ускорением 100 импульсов/с^2 в положительном направлении. Ведомая станция перемещается в соответствии с передаточным числом 2 электронного редуктора. Таким образом, в этот момент скорость ведомой станции составляет 30 импульсов/с.

```
PROGRAM POU_2
VAR
    MC_Power_0: SM3_Basic.MC_Power;
    MC_Power_1: SM3_Basic.MC_Power;
    PWR_EN: BOOL;              enable
    MC_GearIn_0: SM3_Basic.MC_GearIn;
    MC_GearOut_0: SM3_Basic.MC_GearOut;
    MC_MoveVelocity_0: SM3_Basic.MC_MoveVelocity;
    MVEL_EN: BOOL;                  speed mode
    GearOut_EN: BOOL;                        disconnect the electronic cam
    GI_EN: BOOL;                     binding the electronic cam
    SMC_GroupPower_0: SM3_Robotics.SMC_GroupPower;
END_VAR
```



Приложение

Код ошибки	Причины	Описание
0	SMC_NO_ERROR	Ошибки отсутствуют
1	SMC_DI_GENERAL_COMMUNICATION_ERROR	Ошибка связи. Ведомая станция полевой шины больше не работает
2	SMC_DI_AXIS_ERROR	Ошибка оси
3	SMC_DI_FIELDBUS_LOST_SYNCRONICITY	Потеря синхронизации полевой шины
10	SMC_DI_SWLIMITS_EXCEEDED	Положение выходит за пределы диапазона настройки SWLimit
11	SMC_DI_HWLIMITS_EXCEEDED	Превышено предельное положение оборудования
12	SMC_DI_LINEAR_AXIS_OUTOFRANGE	Выход за пределы заданного положения при инкрементном линейном перемещении оси
13	SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPORTED	Состояние привода «остановка» или «быстрый останов» не поддерживается
14	SMC_DI_VOLTAGE_DISABLED	На привод не поступает питание
15	SMC_DI_IRREGULAR_ACTPOSITION	Эта ошибка больше не используется
16	SMC_DI_POSITIONLAGERROR	Чрезмерное отклонение от положения
17	SMC_DI_HOMING_ERROR	Ошибка при возврате в исходное положение
18	SMC_DI_LICENSING_ERROR	Проблемы с лицензией
20	SMC_REGULATOR_OR_START_NOT_SET	В этом состоянии оси невозможно выполнять

Код ошибки	Причины	Описание
		команды управления перемещением
21	SMC_WRONG_CONTROLLER_MODE	Ось работает в неправильном режиме контроллера
25	SMC_INVALID_ACTION_FOR_LOGICAL	Недопустимая операция логической оси
30	SMC_FB_WASNT_CALLED_DURING_MOTION	Команда на перемещение выполняется и не может быть прервана
31	SMC_AXIS_IS_NO_AXISS_REF	Ошибка типа переменных AXIS_REF
32	SMC_AXIS_REF_CHANGED_DURING_OPERATION	Переменная AXIS_REF была заменена, пока модуль активен
33	SMC_FB_ACTIVE_AXIS_DISABLED	Ось выключена во время перемещения. MC_Power.bRegulatorOn
34	SMC_AXIS_NOT_READY_FOR_MOTION	Ось не может выполнять команды на перемещение в текущем состоянии, поскольку ось не выдала сигнал, указывающий на то, что она соответствует заданному значению.
35	SMC_AXIS_ERROR_DURING_MOTION	Ошибка в приводе во время работы
40	SMC_VD_MAX_VELOCITY_EXCEEDED	Превышение макс. скорости fMaxVelocity
41	SMC_VD_MAX_ACCERATION_EXCEEDED	Превышение макс. ускорения fMaxAcceleration
42	SMC_VD_MAX_DEDRATION_EXCEEDED	Превышение макс. торможения fMaxDeceleration
50	SMC_3SH_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения

Код ошибки	Причины	Описание
51	SMC_3SH_MODE_NEEDS_HWLIMIT	Запрос режима через концевые выключатели в целях безопасности
67	SMC_MAC_TOO_MANY_TASK	Слишком много задач для генерирования осей с помощью SDO
68	SMC_MAC_ATOMIC_ADD_FAILED	Атомное сложение не удалось выполнить
69	SMC_SDO_INVALID_DATALENGTH	Считывание SDO привело к неверной длине данных (>4)
70	SMC_SCM_NOT_SUPPORTED	Режим не поддерживается
71	SMC_SCM_AXIS_IN_Error_STATE	В текущем режиме режим работы контроллера нельзя изменять
72	SMC_SCM_INTERRUPTED	Команда SMC_SetControllerMode прервана командой MC_Stop или состоянием errorstop
75	SMC_ST_WRONG_CONTROLLER_MODE	Ось работает в неправильном режиме контроллера. Отклонено и больше не возвращается функцией SMC_SetTorque
80	SMC_RAG_ERROR_DURRING_STARTUP	Ошибка при запуске группы осей
81	SMC_RAG_ERROR_AXIS_NOT_INITIALIZED	Ось еще не достигла заданного состояния
85	SMC_PP_WRONG_AXIS_TYPE	Функциональный блок не поддерживает виртуальные или логические оси
86	SMC_PP_NUMBER_OF_ABSOLUTE_BITS_INVALID	Абсолютное количество цифр недопустимо и должно быть в диапазоне от 8 до 32.

Код ошибки	Причины	Описание
90	SMC_CGR_ZERO_VALUES	Недействительное значение
91	SMC_CGR_DRIVE_POWERED	Пока привод находится в режиме управления, параметры передачи изменять нельзя.
92	SMC_CGR_INVALID_POSPERIOD	Недопустимый цикл модуля (<=0 или >половины пропускной способности шины)
93	SMC_CGR_POSPERIOD_NOT_INTEGRAL	Период модуля, выраженный в приращениях, не является целым числом, но модуль обрабатывается драйвером
110	SMC_P_FTASKCYCLE_EMPTY	Ось не содержит информации о времени цикла (fTaskCycle=0)
120	SMC_R_NO_ERROR_TO_RESET	Ось без ошибок
121	SMC_R_DRIVE_DOESNT_ANSWER	Ось не выполняет сброс ошибок
122	SMC_R_ERROR_NOT_RESETABLE	Невозможно сбросить ошибку
123	SMC_R_DRIVE_DOESNT_ANSWER_IN_TIME	Связь с осью не работает
124	SMC_R_CANNOT_RESET_COMMUNICATION_ERROR	При появлении ошибки связи ее нельзя сбросить
130	SMC_RP_PARAM_UNKNOWN	Неизвестный номер параметра
131	SMC_RP_REQUESTING_ERROR	Передача ошибки на привод. См. номер ошибки в экземпляре функционального блока ReadDriveParameter
132	SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	Нет назначенных параметров привода
133	SMC_RP_PARAM_CONVERSION_ERROR	Не удалось преобразовать значение в параметр привода/параметр привода в значение. Неизвестны параметры перемещения,

Код ошибки	Причины	Описание
		заданного через ПО
140	SMC_WP_PARAM_INVALID	Номер параметра неизвестен или не разрешен к записи
141	SMC_WP_SENDING_ERROR	См. номер ошибки в экземпляре функционального блока WriteDriveParameter
142	SMC_WP_DRIVE_PARAMETER_NOT_MAPPED	Нет назначенных параметров привода
143	SMC_WP_PARAM_CONVERSION_ERROR	Не удалось преобразовать значение в параметр привода/параметр привода в значение. Неизвестны параметры перемещения, заданного через ПО
170	SMC_H_AXIS_WASNT_STANDSTILL	Ось перемещается
171	SMC_H_AXIS_DIDNT_START_HOMING	Привод не начал возврат в исходное положение
172	SMC_H_AXIS_DIDNT_ANSWER	Привод не отвечает после завершения возврата в исходное положение
173	SMC_H_ERROR_WHEN_STOPPING	Останов из-за ошибки после сброса. Торможение настраивать необязательно
174	SMC_H_AXIS_IN_ERRORSTOP	Привод находится в состоянии останова по ошибке. Невозможно выполнить возврат в исходное положение.
180	SMC_MS_UNKNOWN_STOPPING_ERROR	Неизвестная ошибка при остановке
181	SMC_MS_INVALID_ACCDEC_VALUES	Недействительное значение скорости или ускорения
182	SMC_MS_DIRECTION_NOT_APPLICABLE	Direction=shortest не применимо

Код ошибки	Причины	Описание
183	SMC_MS_AXIS_IN_ERRORSTOP	Привод находится в состоянии останова по ошибке. Невозможно выполнить остановку
184	SMC_BLOCKING_MC_STOP_WASNT_CALLED	Не вызывается через Execute=TRUE для блокировки команды оси MC_Stop. Должна быть вызвана MC_Stop (Execute=FALSE)
185	SMC_MS_AXIS_ALREADY_STOPPING	Невозможно прервать текущий блок MC_ Stop
200	SMC_UNKNOWN_TASK_INTERVAL	Невозможно определить интервал задач для задачи шины
201	SMC_MA_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения
202	SMC_MA_INVALID_DIRECTION	Ошибка направления
226	SMC_MR_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения
227	SMC_MR_INVALID_DIRECTION	Ошибка направления
251	SMC_MAD_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения
252	SMC_MAD_INVALID_DIRECTION	Ошибка направления
276	SMC_MSI_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения
277	SMC_MSI_INVALID_DIRECTION	Ошибка направления
278	SMC_MSI_INVALID_EXECUTION_ORDER	За исключением прерывания, не вызывайте главный функциональный блок mvtFB после SMC_MoveSupplicated
300	SMC_LOGICAL_NO_REAL_AXIS	Больше не используется,

Код ошибки	Причины	Описание
		только для совместимости
301	SMC_MV_INVALID_ACCDEC_VALUES	Недействительное значение скорости или ускорения
302	SMC_MV_DIRECTION_NOT_APPLICABLE	Direction=shortest/fastest не применимо
325	SMC_PP_ARRAYSIZE	Неправильный размер массива
326	SMC_PP_STEP0MS	Время шага =t#0c
350	SMC_VP_ARRAYSIZE	Неправильный размер массива
351	SMC_VP_STEP0MS	Время шага =t#0c
375	SMC_AP_ARRAYSIZE	Неправильный размер массива
376	SMC_AP_STEP0MS	Время шага =t#0c
400	SMC_TP_TRIGGEROCCUPIED	Триггер активирован
401	SMC_TP_COULDNT_SET_WINDOW	DriveInterface не поддерживает функцию окон
402	SMC_TP_COMM_ERROR	Ошибка связи
410	SMC_AT_TRIGGERNOTOCCUPIED	Триггер не назначен
426	SMC_MCR_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения
427	SMC_MCR_INVALID_DIRECTION	Неправильное направление
451	SMC_MCA_INVALID_VELACC_VALUES	Недействительное значение скорости или ускорения
452	SMC_MCA_INVALID_DIRECTION	Неправильное направление
453	SMC_MCA_DIRECTION_NOT_APPLICABLE	Direction=fastest не применимо
475	SMC_SDL_INVALID_AXIS_STATE	SMC_ChangeDynamicLimits можно вызывать только в неподвижном состоянии, или power_off
476	SMC_SDL_INVALID_VELACC_VALUES	Недействительные значения

Код ошибки	Причины	Описание
		скорости, ускорения, торможения или скорости в толчковом режиме
600	SMC_CR_NO_TAPPETS_IN_CAM	Кулачок не установлен в точку толкателя
601	SMC_CR_TOO_MANY_TAPPETS	Идентификатор толкателей превысил значение MAX_NUM_TAPPETS
602	SMC_CR_MORE_THAN_32_ACCESSES	Количество обращений к CAM_REF превышает 32 раза
625	SMC_CI_NO_CAM_SELECTED	Кулачок не выбран
626	SMC_CI_MASTER_OUT_OF_SCALE	Выход шпинделя за пределы допустимого диапазона
627	SMC_CI_RAMPIN_NEEDS_VELACC_VALUES	Должно быть ramp_. Требуются значения скорости и ускорения для функции ramp_in
628	SMC_CI_SCALING_INCORRECT	Неправильное масштабирование переменной fEditor/TableMasterMin/Max
629	SMC_CI_TOO_MANY_TAPPETS_PER_CYCLE	Слишком много толкателей работают в пределах одного цикла
640	SMC_CB_NOT_IMPLEMENTED	Функциональный блок с указанным форматом кулачков не реализован
675	SMC_GI_RATIO_DENOM	RatioDenominator = 0
676	SMC_GI_INVALID_ACC	Недействительное значение ускорения
677	SMC_GI_INVALID_DEC	Недействительное значение торможения
678	SMC_GI_MASTER_REGULATOR_CHANGED	Состояние 'enabled/disabled' главного устройства изменено без разрешения

Код ошибки	Причины	Описание
679	SMC_GI_INVALID_JECK	Недействительное значение скорости в толчковом режиме
725	SMC_PH_INVALID_VELACCDEC	Недействительные значения скорости и ускорения/торможения
726	SMC_PH_ROTARYAXIS_PERIOD0	fPositionPeriod=0 модульная ось
750	SMC_NO_CAM_REF_TYPE	Указанный тип кулачка не является типом MC_CAM_REF
751	SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	Данные кривой не включают основные области PMP3xxtart и xEnd в таблице кулачков
752	SMC_CAM_TABLE_EMPTY_MASTER_RANGE	Главный диапазон в таблице данных о кулачках пуст
753	SMC_CAM_TABLE_INVALID_MASTER_MINMAX	Главное устройство с данными о кулачках имеет недействительные максимальные и минимальные значения
754	SMC_CAM_TABLE_INVALID_SLAVE_MINMAX	Подчиненное устройство с данными о кулачках имеет недействительные максимальные и минимальные значения
775	SMC_GIP_MASTER_DIRECTION_CHANGE	При подключении ведомой оси шпиндель изменил направление вращения
776	SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED	Параметр AvoidReversal был введен, но обратный ход ведомой оси нельзя предотвратить
777	SMC_GIP_AVOID_REVERSAL_FOR_FINITE_AXIS	Ввод параметра AvoidReversal не может ограничить установку ведомой оси

Код ошибки	Причины	Описание
778	SMC_GIP_MASTERSTARTDISTANCE_MUST_BE_ZERO_BUFFERED	Если параметр BufferMode не был прерван, значение параметра MasterStartInstance не может быть положительным
779	SMC_GIP_CANNOT_START_SYNC	Невозможно начать синхронизацию. Такая ситуация может возникнуть, когда команда GearInPos действует как буфер или команда гибридного перемещения, а главный контроллер находится в неподвижном состоянии после включения
800	SMC_BC_BL_TOO_BIG	Передаточное отношение возврата (fBacklash) слишком велико (>position periode/2)
825	SMC_QPROF_DIVERGES	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
826	SMC_QPROF_INVALID_PARAMETER	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
827	SMC_QPROF_NO_RESULT	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
828	SMC_QPROF_INVALID_NEW_LBD	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
829	SMC_QPROF_BAD_NEGOTIATION	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
830	SMC_QPROF_INVALID_INTERVAL	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить

Код ошибки	Причины	Описание
831	SMC_QPROF_NOT_ENOUGH_PHASES	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
832	SMC_TG_INTERNAL_ERROR	Внутренняя ошибка: расчет квадратичной траектории не удалось выполнить
850	SMC_SRT_NOT_STANDSTALL_OR_POWEROFF	Операции допустимы только в неподвижном состоянии или после выключения питания
851	SMC_SRT_INVALID_RAMPTYPE	Неправильный тип крутизны характеристики
852	SMC_SMT_NOT_STANDSTALL_OR_POWEROFF	Операции допустимы только в неподвижном состоянии или после выключения питания
853	SMC_SMT_INVALID_MOVEMENTTYPE_OR_POSITIONPERIOD	Неправильный тип перемещения или период позиционирования
854	SMC_SMT_AXIS_NOT_VIRTUAL	Функциональный блок относится только к виртуальным осям
1000	SMC_NO_LICENSE	Целевое значение не совместимо с ЧПУ
1001	SMC_INT_VEL_ZERO	Невозможно обработать траекторию, так как заданная скорость равна 0
1002	SMC_INT_NO_STOP_AT_END	Конечная скорость последнего объекта на траектории <>0
1003	SMC_INT_DATA_UNDERRUN	Предупреждение: Список географической информации был обработан в DataIn, но конец списка не достигнут. Причина: Невозможно задать EndOfList SMC_Interpolator очереди в

Код ошибки	Причины	Описание
		DataIn быстрее, чем функциональные блоки генерирования траектории
1004	SMC_INT_VEL_NONZERO_AT_STOP	Скорость при остановке >0
1005	SMC_INT_TOO_MANY_RECURSIONS	SMC_Interpolator избыточная рекурсия. Ошибка при перемещении, заданном через ПО
1006	SMC_INT_NO_CHECKVELOCITIES	SMC_CheckVelocity не является последним обработанным функциональным блоком, он получил доступ к данным вне очереди через poqDataIn
1007	SMC_INT_PATH_EXCEEDED	Внутренняя ошибка или численная ошибка
1008	SMC_INT_VEL_ACC_DEC_ZERO	Значения скорости и ускорения/торможения равны нулю или слишком низкие
1009	SMC_INT_DWIPOTIME_ZERO	Используйте dwIpoTime=0 для вызова функционального блока
1010	SMC_INT_JERK_NONPOSITIVE	Недействительное значение скорости в толчковом режиме, так как оно должно быть положительным
1011	SMC_INT_QPROF_DIVERGES	Внутренняя ошибка. Расчет квадратичного распределения скорости не сходится
1012	SMC_INT_INVLALID_VELOCITY_MODE	Недопустимый режим скорости
1013	SMC_INT_TOO_MANY_AXES_INTERPOLATED	Количество вставленных осей превышает допустимое число осей. Вы используете ограниченную версию
1014	SMC_INT_DEGENERATE_SEGMENT	Этот сегмент является

Код ошибки	Причины	Описание
		численно вырожденным: он очень короткий и находится в конце очереди. Его следует игнорировать
1015	SMC_HIGH_CURVATURE_SPLINE	Расчет точек интерполяции не удалось выполнить из-за слишком большой кривизны сплайновой кривой. Попробуйте изменить траекторию, чтобы избежать острых углов.
1050	SMC_INT2DIR_BUFFER_TOO_SMALL	Предупреждение: poqDataIn созданной очереди вывода слишком мал и не может гарантировать соблюдение требований к остановке
1051	SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	Траектория не полностью находится в очереди
1070	SMC_XINT_INVALID_DIRECTION	Вход направления имеет недопустимое значение
1071	SMC_XINT_NOINTERSECTION	Невозможно определить положение заданной позиции x на траектории ЧПУ
1080	SMC_WAR_INT_OUTQUEUE_TOO_SMALL	Предупреждение: poqDataIn созданной очереди вывода слишком мал и не может гарантировать соблюдение требований к остановке
1081	SMC_WAR_END_VELOCITIES_INCORRECT	Предупреждение: Конечная скорость не соответствует требованиям
1100	SMC_CV_ACC_DEC_VEL_NONPOSITIVE	Значения скорости и ускорения/торможения: неположительные
1120	SMC_CA_INVALID_ACCDEC_VALUES	Значения fGapVelocity/fGapAcceleration/F

Код ошибки	Причины	Описание
		GAPDegregation: неположительные
1130	SMC_TOK_COMPLETE_TOKEN_AT_END_OF_INPUT	Все данные введены, но все еще остались незавершенные отметки
1131	SMC_TOK_NOT_A_VALID_TOKEN	Входные данные не соответствуют ни одному типу токенов
1132	SMC_TOK_MUBILITY_INPUT	Неоднозначные входные данные, может быть несколько типов токенов
1133	SMC_TOK_STRING_TOO_LONG_FOR_TOKEN	Строка слишком длинная для хранения в токене. (Слишком длинные текст строки, имя переменной или идентификатор)
1134	SMC_TOK_INVALID_NUMLIT	Неверный цифровой текст
1150	SMC_PRS_FUNC_DECL_TOO_MORY_ARGMENTS	Слишком мало параметров задано для функции в G-коде
1151	SMC_PRS_FUNC_DECL_TOO_MANY_ARGMENTS	Слишком много параметров задано для функции в G-коде
1152	SMC_PRS_FUNC_DECL_WRONG_ARGUMENT_TYPE	Проверьте, соответствуют ли типы параметров заявленной функции. Возвращаемая ошибка
1153	SMC_PRS_LOCAL_VAR_NOT_FOUND	Невозможно найти локальную переменную в стеке
1154	SMC_PRS_INVALID_STRING	Невозможно считать строку из токена
1155	SMC_PRS_TOO_MANY_CLOSING_BRACKETS	Закрывающих скобок больше, чем открывающих
1156	SMC_PRS_TOO_MANY_OPENING_BRACKETS	Открывающих скобок больше, чем закрывающих

Код ошибки	Причины	Описание
1157	SMC_PRS_NO_SUCH_INFIX_OPERATOR	Невозможно найти инфиксный оператор
1158	SMC_PRS_NO_SUCH_PREFIX_OPERATOR	Невозможно найти префиксный оператор
1159	SMC_PRS_OPERATOR_INVALID_PRECEDENCE	Получены два оператора с равным приоритетом, но неравной ассоциативностью
1160	SMC_PRS_NOT_A_TERM	Ожидаемый срок действия
1161	SMC_PRS_EXPRESSION_INVALID_SEQUENCE	В выражении найдена недопустимая последовательность тегов
1162	SMC_PRS_TOO_MANY_TERMS	Получено больше условий, чем ожидалось
1163	SMC_PRS_STACK_OVERFLOW	Невозможно разобрать выражение, потому что стек слишком мал
1164	SMC_PRS_VAR_NAME_READY_USED	Имя параметра или переменной подпрограммы уже используется для других параметров или переменных подпрограммы
1165	SMC_PRS_INCOMPLETE_SENTENCE_IN_TOKEN_QUEUE	Входная очередь токенов анализатора G-кода не содержит полного оператора G-кода
1167	SMC_PRS_TOO_MANY_SUBPROGRAMS	Невозможно сохранить объявление подпрограммы, поскольку превышена емкость таблицы символов
1168	SMC_PRS_TOO_MANY_SUBPROGRAM_PARAMETERS	Превышено максимальное количество параметров подпрограммы
1169	SMC_PRS_SUBPROGRAM_LOOKUP_FAILED	Невозможно найти объявление подпрограммы. Возможно, имя

Код ошибки	Причины	Описание
		подпрограммы неверное, или путь поиска подпрограммы неполный
1170	SMC_PRS_VAR_NOT_FOUND	Переменная не найдена в таблице символов
1171	SMC_PRS_TOKEN_TYPE_UNKNOWN	Неизвестный тип токена
1172	SMC_PRS_GOT_NO_TERM	После G-адреса терминология отсутствует
1173	SMC_PRS_INVALID_VAR_TYPE	Неизвестный тип переменной (не является LREAL, BOOL или строкой)
1174	SMC_PRS_UNEXPECTED_TOKEN	Требуются различные типы токенов (например, операторы или идентификаторы)
1175	SMC_PRS_ID3_EXPECTED	Требуется идентификатор длиной 3 или больше
1176	SMC_PRS_ID_TOO_LONG	Идентификатор слишком длинный (больше 80 символов)
1177	SMC_PRS_GADDRESS_EXPECTED	Требуется G-адрес (например, "G", "F", "X")
1178	SMC_PRS_NWORD_EXPECTED	Требуется символ N (например, «N10»).
1179	SMC_PRS_NWORD_INVALID_SENTURE_NUMBER	Номер предложения не находится в диапазоне [0,..., 16 # FFFFFFF].
1180	SMC_PRS_NWORD_SENTENCE_NUMBER_NO_NUMBER_LITERAL	Номер предложения должен быть числовым литералом. (Например, выражения недопустимы.)
1181	SMC_PRS_USE_OF_RESERVED_KEYWORD	Идентификатор — зарезервированное ключевое слово, и его нельзя здесь использовать.

Код ошибки	Причины	Описание
1182	SMC_PRS_SUBPROGRAMS_SIGNATURE_MISMATCH	Используется другая сигнатура, или в объявлении подпрограммы была допущена ошибка чтения
1183	SMC_PRS_INITIAL_VALUE_HAS_WRONG_TYPE	Начальное значение этой локальной переменной имеет неправильный тип
1184	SMC_PRS_TOO_MANY_LOCAL_VARIABLES	Превышено максимальное количество параметров подпрограммы
1185	SMC_PRS_SUBPROG_MUST_BE_FIRST_SENTENCE	Объявление подпрограммы должно быть первым предложением в файле G-кода
1186	SMC_PRS_ONLY_ONE_subgram_PER_FILE_ALLOWED	D файле G-кода может быть только одна подпрограмма.
1187	SMC_PRS_LET_AFTER_REGULAR_SENTENCE	Объявление LET недопустимо после обычных операторов G-кода
1188	SMC_PRS_UNMATCHED_END_SUBPROGRAM	END_SUBPROGRAM. Оператор не соответствует объявлению подпрограммы
1189	SMC_PRS_UNEXPECTED_TOKENS_AFTER_SUBPROGRAM	После END_SUBPROGRAM. больше не разрешается использовать токены
1190	SMC_PRS_MISSING_END_SUBPROGRAM	Подпрограмма не завершается командой END_ SUBPROGRAM.
1200	SMC_DEC_ACC_TOO_LITTLE	Недопустимое значение ускорения
1201	SMC_DEC_RET_TOO_LITTLE	Недопустимое значение торможения
1202	SMC_DEC_OUTQUEUE_RAN_EMPTY	Недостаточно данных. Очередь прочитана и пуста

Код ошибки	Причины	Описание
1203	SMC_DEC_JUMP_TO_UNKNOWN_LINE	Невозможно выполнить переход к строке, поскольку номер строки неизвестен
1204	SMC_DEC_INVALID_SYNTAX	Недопустимый синтаксис
1205	SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	Объект не поддерживается в 3D-режиме
1206	SMC_DEC_NEGATIVE_PERIOD	Недопустимое отрицательное значение в качестве дополнительного периода оси
1207	SMC_DEC_DIMENSIONS_EXCLUSIVE_AU	Интерполяция не выполняется по обеим осям A и U. PA и PU являются взаимоисключающими
1208	SMC_DEC_DIMENSIONS_EXCLUSIVE_BV	Интерполяция не выполняется по обеим осям B и V. PB и PV являются взаимоисключающими
1209	SMC_DEC_DIMENSIONS_EXCLUSIVE_CW	Интерполяция не выполняется по обеим осям C и W. PC и PW являются взаимоисключающими
1210	SMC_DEC_DCS_NOT_ALL_OF_ABC_GIVEN	Для G54/G55/G56, если режим направления не равен SMC_ORI_CONVENTION.ADDAXES, то должны быть указаны все значения A, B, C или ни одного
1211	SMC_DEC_DCS_2D_NOT_IN_XY_PLANE	Ось Z системы координат шифратора должна соответствовать оси Z системы координат станка
1212	SMC_DEC_CIRCLE_NON_UNIFORM_SCALING	Масштабирование окружности или эллипса, имеющего команду поворота

Код ошибки	Причины	Описание
1213	SMC_DEC_ROTATION_AFFECTS_SCALING	Новый относительный поворот DCS (G55) повлияет на масштабирование
1214	SMC_DEC_DCS_NOT_ALL_OF_IJK_GIVEN	Для G54/G55/G56 должны быть указаны все значения I, J, K или ни одного
1250	SMC_IPR_LOCAL_VAR_UNKNOWN_TYPE	Невозможно считать локальную переменную из-за недопустимого типа
1251	SMC_IPR_LVALUE_WRONG_TYPE	Неверный тип переменной, которую следует записать
1252	SMC_IPR_EVAL_STACK_OVERFLOW	Невозможно оценить выражение, потому что стек оценки слишком мал
1253	SMC_IPR_NOT_A_NUMBER	Числовой член становится NaN
1254	SMC_IPR_DIVISION_BY_ZERO	Деление на ноль
1255	SMC_IPR_INVALID_SCALING_FACTORS	Неверный коэффициент масштабирования
1256	SMC_IPR_LVALUE_WRONG_TYPE	При возврате подпрограммы стек интерпретатора пуст. внутренняя ошибка
1257	SMC_IPR_INTERPRETER_STACK_OVERFLOW	Стек интерпретатора слишком мал. Слишком много локальных переменных в G-коде или слишком много активных подпрограмм
1258	SMC_IPR_INVALID_INTERPRETER_STACK_BUFFER	Размер данного буфера стека интерпретатора равен 0 или меньше 1024 байт
1259	SMC_IPR_BUFFER_TOO_SMALL	Буфер очереди вывода слишком мал
1280	SMC_DNCCS_NO_DATA	Нет никаких доступных данных

Код ошибки	Причины	Описание
1281	SMC_DNCCS_TOO_MANY_CALLSTACKS	Между текущим состоянием iObjNo и интерпретатором находится слишком много стеков вызовов
1282	SMC_DNCCS_INVALID_PROGRAM_INDEX	Неверный индекс программы
1283	SMC_DNCCS_TOO_MANY_PROGRAM	Слишком много разных программ
1284	SMC_DNCCS_WRONG_TASK	Функциональный блок вызвал неправильную задачу
1300	SMC_GCV_BUFFER_TOO_SMALL	Буфер слишком мал
1301	SMC_GCV_BUFFER_WRONG_TYPE	Неправильный тип элемента буфера
1302	SMC_GCV_UNKNOWN_IPO_LINE	Невозможно найти текущую строку интерполятора
1400	SMC_CNC_INTERNAL_ERROR	Внутренняя ошибка ЧПУ
1401	SMC_PATH_MAX_HPOINTS_EXCEEDED	Элемент траектории не может быть сохранен за пределами точек H переключения MAX_IPOSWITCHES. Настройте точку переключения, используя различные значения "O", или уменьшите количество точек H для каждого элемента траектории.
1410	SMC_TRC_G75_NOT_ALLOWED	G75 не допускается во время коррекции радиуса инструмента
1411	SMC_TRC_QUEUE_FULL_NON_CARTESIAN	Очередь заполнена, но других декартовых элементов нет
1412	SMC_TRC_SPLINE3D_5_NOT_SUPPORTED	Коррекция радиуса инструмента не поддерживает тип перемещения SPLINE3D_5
1414	SMC_TRC_PLANE_CHANGE_NOT_ALLOWED	Не допускайте изменений

Код ошибки	Причины	Описание
		плоскости во время коррекции радиуса инструмента
1450	SMC_NAV_MAX_SUBSPROGRAM_NESTING_EXCEEDED	Превышен максимальный уровень вложенности для вызова подпрограммы
1451	SMC_NAV_RETURN_FROM_MAIN	Не разрешайте основным программам G-кода возвращать операторы
1452	SMC_NAV_SUBPROGRAM_DECLARATION_NOT_FOUND	Объявление подпрограммы не найдено в файле подпрограммы ЧПУ
1453	SMC_NAV_NOT_ENOUGH_SPACE_FOR_COMPLETE_SENTENCE	Невозможно добавить следующий оператор G-кода в очередь выходных операторов из-за недостатка места
1500	SMC_NO_CNC_REF_TYPE	Данная программа ЧПУ не является программой типа SMC_ CNC_ REF
1501	SMC_NO_OUTQUEUE_TYPE	Данная очередь вывода не является очередью типа SMC_ OutQueue
1502	SMC_GEOINFO_BUFFER_MISALIGNED	Не используется 4-байтовый выровненный раздел буфера в pbyBuffer
1600	SMC_3D_MODE_NOT_SUPPORTED	Функциональный блок относится только к двухмерным траекториям
1700	SMC_SAA_SMOOTHAREA_TOO_LARGE	Слишком большой диапазон плавного хода
1701	SMC_SAA_SP_INVALID_INPUT	Неверный ввод dSmoothingPart]0..1]
1800	SMC_SA_QUEUE_NOT_IN_BUFFER	SMC_SegmentAnalyzer обнаружил, что буфер вне очереди заполнен, но не завершен. Функциональный

Код ошибки	Причины	Описание
		блок может выполняться только тогда, когда OutQueue полностью подходит для буфера
1801	SMC_SA_QUEUE_CHANGED_DURING_OP	Когда функциональный блок работает с ним, буфер вне очереди изменился
1820	SMC_OS_INVALID_PARAMETER	Недопустимое входное значение внутри dSplittingParameter
1830	SMC_BSSP_IPO_NOT_ACTIVE	Не удалось сохранить местоположение. Вставка не активна
1831	SMC_BS_SAVEDPOS_NOT_REACHED	Сохраненное местоположение пока не найдено. Возможно, выбрана другая траектория
1832	SMC_BS_NO_POS_STORED	Структура, переданная в ePos, не содержит сохраненного местоположения. SMC_BlockSearchSavePos не выполняется или выполняется неправильно
1900	SMC_INVALID_FEATURE_FLAG	Функциональный символ должен находиться в диапазоне {1,..., 31}.
1901	SMC_SMB_HFUN_NOT_SUPPORTED	Функциональный блок не поддерживает функцию h
1902	SMC_SMB_ONLY_3DMODE	Функциональный блок работает только в трехмерном режиме
1903	SMC_SMB_ERROR_COMPUTING_SPLINE	Вычислите внутреннюю ошибку сплайновой кривой
1910	SMC_SMM_INVALID_PARAM_NUMBER	Слишком большой WadAdditionalParameter
1950	SMC_INVALID_PARAMETER	Одно из входных значений

Код ошибки	Причины	Описание
		недействительно
1951	SMC_BLENDING_NOT_SUPPORTED_BY_PREVIOUS_MOVEMENT	Предыдущее перемещение не поддерживает смешивание
1952	SMC_BUFFERED_NOT_SUPPORTED_BY_PREVIOUS_MOVEMENT	Предыдущее перемещение не поддерживает буферизованные перемещения
1953	SMC_INHERIT_OWNER_ACTIVE_MOVEMENT_NOT_CALLED	Невозможно наследовать владельца, так как активное перемещение не было вызвано в этом цикле. внутренняя ошибка
1954	SMC_MOVING_WITHOUT_ACTIVE_MOVEMENT	Ось перемещается, но функциональный блок активного перемещения отсутствует. Нет последующих непрерывных перемещений одно за другим
1955	SMC_BUFFER_MODE_NOT_SUPPORTED	Заданное перемещение не поддерживает настроенный режим буферизации
1956	SMC_ERROR_IN_A_PREVIOUS_MOVEMENT	Во время предыдущего перемещения произошла ошибка
1957	SMC_MORE_THAN_ONE_MOVEMENT_PER_INSTANCE	На каждый экземпляр функционального блока может быть только один занятый буфер/смешанное перемещение
2000	SMC_RNCF_FILE_DOESNT_EXIST	Файла не существует
2001	SMC_RNCF_NO_BUFFER	Нераспределенный буфер
2002	SMC_RNCF_BUFFER_TOO_SMALL	Буфер слишком мал
2003	SMC_RNCF_DATA_UNDERRUN	Недостаточно данных. Данные считаны из буфера, он пуст
2004	SMC_RNCF_VAR_COULDNT_BE_REPLACED	Невозможно заменить

Код ошибки	Причины	Описание
		переменную-заполнитель
2005	SMC_RNCF_NOT_VARLIST	Входной pvl не указывает на объект SMC_ VARLIST
2006	SMC_RNCF_NO_STRINGBUFFER	Вход pStringBuffer не используется и не указывает на переменные типа SMC_StringBuffer
2007	SMC_RNCF_STRINGBUFFER_OVERRUN	В программах ЧПУ используется больше различных строк, чем может вместить буфер строк
2008	SMC_RNCF_SUBPROGRAM_FILE_NOT_FOUND	Невозможно найти файл для подпрограммы
2050	SMC_RNCQ_FILE_DOESNT_EXIST	Не удалось открыть файл
2051	SMC_RNCQ_NO_BUFFER	Неопределенный буфер
2052	SMC_RNCQ_BUFFER_TOO_SMALL	Буфер слишком мал
2053	SMC_RNCQ_UNEXPECTED_EOF	Неожиданный конец файла
2100	SMC_ADL_FILE_CANNOT_BE_OPENED	Не удалось открыть файл
2101	SMC_ADL_BUFFER_OVERRUN	Переполнение буфера. WriteToFile должен вызываться чаще
2200	SMC_RCAM_FILE_DOESNT_EXIST	Не удалось открыть файл
2201	SMC_RCAM_TOO_MUCH_DATA	Сохраненный кулачок слишком большой
2202	SMC_RCAM_WRONG_COMPILE_TYPE	Неправильный режим компиляции
2203	SMC_RCAM_WRONG_VERSION	Неверная версия файла
2204	SMC_RCAM_UNEXPECTED_EOF	Неожиданный конец файла
3001	SMC_WDPF_CHANNEL_OCCUPIED	Эта ошибка устарела и зарезервирована только для обеспечения совместимости.

Код ошибки	Причины	Описание
		SMC_WDPF_TIMEOUT_PREPARING_LIST
3002	SMC_WDPF_CAN_CREATE_FILE	Эта ошибка устарела и зарезервирована только для обеспечения совместимости. Не удалось создать файл
3003	SMC_WDPF_ERROR_WHEN_READING_PARAMS	Эта ошибка устарела и зарезервирована только для обеспечения совместимости. Ошибка чтения параметров
3004	SMC_WDPF_TIMEOUT_PREPARING_LIST	Эта ошибка устарела и зарезервирована только для обеспечения совместимости. Истекло время ожидания при подготовке списка параметров
5000	SMC_ENC_DENOM_ZERO	Коэффициент преобразования параметра энкодера dwRatioTechUnitsDenom named равен 0.
5001	SMC_ENC_AXISUSEDBYOTHERFB	Попытка обработки других модулей, движущихся по оси энкодера
5002	SMC_ENC_FILTER_DEPTH_INVALID	Неверная глубина фильтрации
6000	SMC_INTERNAL_ERROR_INIT_MOVEMENT	Внутренняя ошибка при инициализации нового перемещения
6001	SMC_INVALID_AXIS_TYPE	Неверный тип оси, не конечный и не модульный (внутренняя ошибка)
10000	SMC_PCCQ_POINTBUFFERTOOSMALL	Буфер pBuffer слишком мал
10001	SMC_PCCQ_INPUTBUFFERFULLBUTNOTFINALIZED	Функциональный блок должен быть применен к траектории, которая полностью подходит для буфера

Код ошибки	Причины	Описание
11000	SMC_AXIS_GROUP_WRONG_STATE	При запросе операции для группы осей была выдана ошибка
11001	SMC_AXIS_GROUP_TOO_MANY_AXES	Добавлена группа осей с количеством осей, превышающим максимально допустимое
11002	SMC_AXIS_GROUP_INVALID_DYNLIMITS	Недопустимый динамический предел для одной оси (fSWMaxVelocity/ acceleration/deceleration/jerk)
11004	SMC_AXIS_GROUP_INVALID_COORD_SYSTEM	Заданная система координат недопустима для запрашиваемой операции
11005	SMC_AXIS_GROUP_SINGLE_AXISS_ERROR	Неправильное состояние оси в группе осей
11006	SMC_MOVE_INVALID_BUFFER_MODE	Заданный буферный режим не поддерживается
11007	SMC_MOVE_INVALID_DYNAMIC_FACTOR	Динамический коэффициент не находится в диапазоне [0.. [1]
11008	SMC_MOVE_INVALID_DYNAMICS	Неверные динамические ограничения для перемещения
11009	SMC_AXIS_GROUP_AXIS_NOT_PART_OF_GROUP	Данная ось не входит в группу осей
11010	SMC_AXIS_GROUP_NOT_SUPPORTED	Запрошенная операция не поддерживается
11011	SMC_AXIS_GROUP_KINEMATICS_NOT_SET	Кинематические конфигурации еще не заданы
11012	SMC_AXIS_GROUP_WRONG_NUMBER_OF_AXES	Количество осей не равно количеству осей, необходимых для преобразования перемещения
11013	SMC_AXIS_GROUP_INTERRUPTED_BY_SINGLE_	Согласованное перемещение

Код ошибки	Причины	Описание
	AXIS	прервано перемещением одной оси
11014	SMC_AXIS_GROUP_FOLLOW_SETVALUES	Расчетные параметры отслеживания ошибок
11015	SMC_AXIS_GROUP_TOO_MANY_dependency	Одна группа осей не может быть основана на нескольких группах осей
11016	SMC_AXIS_GROUP_MUTUAL_DEPENDENCY	Группа осей A может не зависеть от другой группы осей, которая зависит от группы A
11017	SMC_AXIS_GROUP_DEPENDENCY_IN_DIFFERENT_TASK	Зависимая группа осей должна быть выполнена в той же задаче
11018	SMC_AXIS_GROUP_AXISS_IN_DIFFERENT_TASK	Оси, принадлежащие к группа осей, должны быть выполнены в той же задаче
11019	SMC_AXIS_GROUP_PCS_STILL_IN_USE	Если PCS все еще используется буферизованными командами на перемещение, функциональный блок активируется повторно
11020	SMC_AXIS_GROUP_CMD_ABORTED_DUE_TO_ERROR	Ошибка в предыдущей команде на перемещение
11021	SMC_AXIS_GROUP_INVALID_PARAMETER	Недопустимые параметры для функционального блока управления
11022	SMC_AXIS_GROUP_UNSUPPORTED_RAMPTYPE	Одна ось в группе использует неподдерживаемый тип линейного изменения характеристик. Поддерживаются только трапецеидальные и квадратичные типы

Код ошибки	Причины	Описание
11023	SMC_MOVE_INVALID_TRANSITION_PARAMETER	Недопустимый параметр преобразования
11024	SMC_AXIS_GROUP_INTERNAL_ERROR	Внутренняя ошибка в машине состояний группы осей
11025	SMC_AXIS_GROUP_CPTR_CANNOT_FOLLOW	Отслеживание CP: невозможно сохранять траекторию. Попытка снизить динамику на траектории и/или перейти в режим отслеживания CP
11026	SMC_AXIS_GROUP_CONTINUE_WRONG_POSITION	Текущее положение не позволяет продолжить операции
11027	SMC_AXIS_GROUP_CONTINUE_BUFFER_TOO_SMALL	Буфер в ContinueData слишком мал. Использование большого внешнего буфера очереди перемещений
11028	SMC_AXIS_GROUP_CONTINUE_WRONG_CHECKSUM	Четность и ошибка в данных продолжения, данные продолжения не записываются командой MC_GroupInterrupt, а изменяются позже
11029	SMC_AXIS_GROUP_IDLE_WAIT_AXES_MOVING	Ввод команды SMC_GroupWait, одновременно ось движется независимо от группы осей (перемещение одной оси или аналогичная операция)
11030	SMC_AXIS_INVERSE_TRAFO_EXCEEDING_POSLIMITS	Обратное преобразование приведет к тому, что значение оси превысит заданный предел
11031	SMC_AXIS_GROUP_CANNOT_ADD_SAME_AXIS	Запрет на повторное добавление одной и той же оси в группу осей
11032	SMC_AXIS_GROUP_TRANSFORMATION_NOT_SET	Не задавайте преобразование

Код ошибки	Причины	Описание
11033	SMC_AXIS_GROUP_NON_RESUMABLE_ERROR	Невозможно восстановить траекторию после ошибки текущей группы осей
11034	SMC_AXIS_GROUP_CONTINUE_DATA_NOT_WRITTEN	Данные продолжения для MC_GroupContinue были записаны неправильно. (MC_GroupInterrupt не вызвана/неустранимая ошибка группы осей)
11100	SMC_KERNEL_PTP_INTERNAL_ERROR	Внутренние ошибки в ядре
11101	SMC_KERNEL_PTP_INVALID_TASKCYCLETIME	Время цикла выполнения задачи не является положительным
12000	SMC_TRAFO_INVALID_PARAMETERS	Преобразованное значение параметра недействительно
12001	SMC_TRAFO_INVALID_CONSTELLATION	Запрошенное местоположение находится за пределами преобразованной рабочей зоны
12002	SMC_TRAFO_INVALID_COUNPING	Инструменты нельзя использовать одновременно с органами позиционирования станка. В качестве варианта, орган позиционирования не реализован в ISMPocationKinetics2, или этот инструмент не может обрабатывать изображения с направлением позиционирования.
12003	SMC_TRAFO_NOT_INITIALIZED	Функциональный блок реализовал |ISMkineticWithinitialization|, но еще не был инициализирован
12004	SMC_TRAFO_NO_TOOL_WITH_OFFSET_ALLOWED	Специальные ошибки в секции позиционирования Kin_4Axis_Palletzer, реализовано

Код ошибки	Причины	Описание
		ISMPositionKinetics_Offset2. Но в реальности невозможно обрабатывать смещения <>0
12005	SMC_AXIS_GROUP_TOOL_OFFSET_INCOMPATIBLE_WITH_KINEMATICS	Заданная коррекция на инструмент несовместима с используемой кинематикой
12100	SMC_CP_CACHE_FULL	Слишком мал размер кэша планировщика CP
12101	SMC_CP_EVAL_ERROR	Расчет положения на элементе траектории не удалось выполнить (внутренняя ошибка)
12102	SMC_CP_NON_CONTINUABLE_STATE	Достигнуто состояние, не являющееся непрерывным (внутренняя ошибка)
12103	SMC_CP_MAX_LENGTH_EXCEEDED	Превышена максимальная длина траектории (внутренняя ошибка)
12104	SMC_CP_ACCELERATION_TOO_HIGH	Слишком высокое ускорение на траектории (недопустимое состояние, внутренняя ошибка)
12105	SMC_CP_MAX_ITERATIONS_EXCEEDEDSMC_CP_MAX_ITERATIONS_EXCEEDED	Превышено максимальное количество итераций (внутренняя ошибка)
12106	SMC_CP_NO_TRAJECTORY	Невозможно рассчитать траекторию (внутренняя ошибка)
12107	SMC_CP_OUT_QUEUE_FULL	Заполнена очередь вывода (внутренняя ошибка)
12108	SMC_CP_QUEUE_UNDERRUN	Недостаточная работа очереди: в очереди не осталось элементов
12109	SMC_CP_INVALID_QUEUE	Неверная очередь (неверный размер или указатель)

Код ошибки	Причины	Описание
12110	SMC_CP_Blend_INTERNAL_ERROR	Внутренняя ошибка при смешивании двух перемещений CP
12111	SMC_CP_CIRCLE_COLLINEAR_POINTS	Три точки, определяющие окружность, лежат на одной прямой
12112	SMC_CP_CIRCLE_CENTER_NOT_ON_BISECTOR	Центральная точка не находится на вертикальной биссектрисе начальной и конечной точек
12113	SMC_CP_CIRCLE_RADIUS_ZERO	Нулевой радиус
12114	SMC_CP_CIRCLE_DISTANCE_LARGER_THAN_DIAMETER	Расстояние между начальной и конечной точками больше диаметра (при попытке создать окружность с помощью SMC_Circ_Mode.radius)
12115	SMC_CP_CIRCLE_START_AND_ENDPOINT_EQUAL	Одинаковые начальная и конечная точки (SMC_Circ_Mode.Radius)
12116	SMC_CP_BLENDING_DEGENERATE_SPLINE	Смешанная деградация сплайна (слишком короткий или нерегулярный)
12117	SMC_CP_ELEMENT_TOO_SHORT	Невозможно создать элемент траектории, так как он слишком короткий
12118	SMC_CP_CAN_NOT_CUT_PATH	Невозможно сократить траекторию, чтобы прервать перемещение (внутренняя ошибка)
12119	SMC_CP_INVALID_ANGULAR_VEL_ACC	Заданная угловая скорость или ускорение недопустимы (кватернионы с ненулевыми вещественными частями, внутренняя ошибка)
12120	SMC_CP_INVALID_ORIENTATION	Неверное заданное

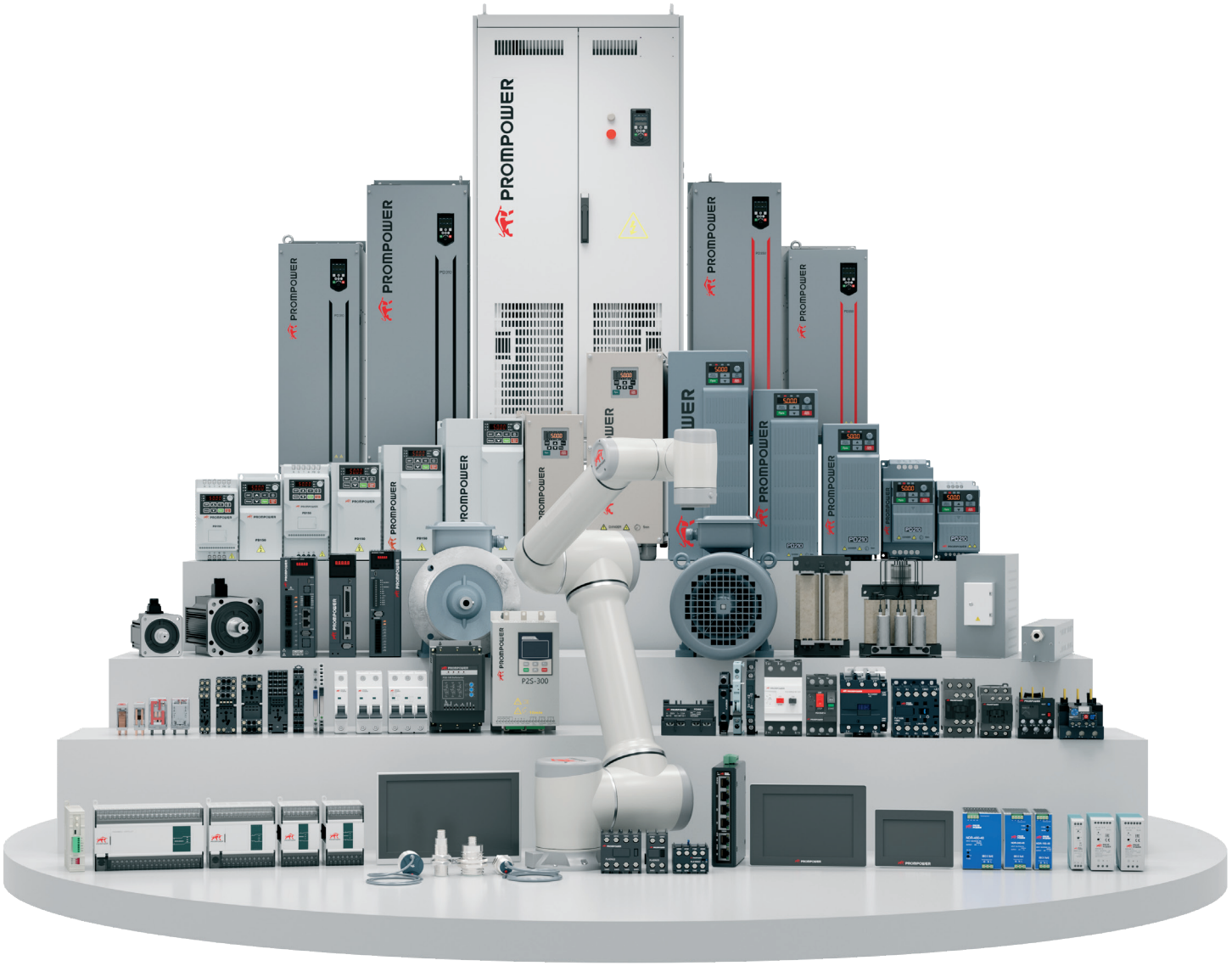
Код ошибки	Причины	Описание
		направление (неортогональная матрица или нетождественный кватернион, внутренняя ошибка)
12121	SMC_CP_TIME_BUDGET_EXCEEDED	Превышен рассчитанный заданный бюджет времени (внутренняя ошибка)
12122	SMC_AXIS_GROUP_AXIS_LIMIT_VIOLATED	Нарушены установленные для осей ограничения. Если CP движется слишком близко к сингулярности или траектория имеет очень большую кривизну, может возникнуть такая ситуация
12125	SMC_CP_CONFIGS_DIFFER	Кинематическая конфигурация начального положения отличается от конфигурации конечного положения. Перемещение CP будет проходить через сингулярности
12126	SMC_CP_BUS_TASK_NOT_CALLED	Эта ошибка устарела и зарезервирована только для обеспечения совместимости. Если используется команда CPhalt/- stop, необходимо передать некоторую информацию в задачу шины. Ожидается, что это произойдет до начала следующего цикла медленно выполняемой задачи
12127	SMC_CP_NO_ROOT_IN_INTERVAL_FOUND	Не удалось найти траекторию остановки в интервале. (внутренняя ошибка)
12128	SMC_CP_KIN_DOES_NOT_SUPPORT_AXIS_ORIENTATION_IPO	Кинематика не поддерживает режим направления перемещения по непрерывной траектории 'axis'

Код ошибки	Причины	Описание
12129	SMC_CP_AXIS_ORIENTATION_IPO_NOT_SUPPORTED_FOR_CPTR	Перемещение CP с динамической PCS (отслеживание) не поддерживает режим интерполяции направления оси
12130	SMC_CP_INVALID_PATH_ELEM	Создан недопустимый элемент траектории (внутренняя ошибка)
12131	SMC_CP_TRANSITION_NOT_SMOOTH	Преобразование между двумя элементами траектории не является непрерывным G2 и не прекратилось (внутренняя ошибка)
12132	SMC_CP_AXIS_ORIENTATION_IPO_CONFIG_DIFFERS	Был использован режим интерполяции направления оси, но между начальной и конечной положениями существует сингулярность в кинематике позиционирования
12133	SMC_CP_AXIS_ORIENTATION_IPO_OFFSET2_NOT_IMPLEMENTED	Кинематика позиционирования имеет конфигурации, но не реализует interfacesIsmPositionKinetics_Offset2. В этом случае интерполяция направления оси невозможна
12134	SMC_CP_ROTARY_AXIS_PERIOD_MISMATCH	Заданное положение оси вращения не будет достигнуто в течение командного цикла. (Например, заданное в команде положение -170°, но будет достигнуто положение 190°). Это означает, что выбранная интерполяция заданного положения несовместима с заданным положением
12135	SMC_CP_ROTARY_AXIS_RANGE_VIOLATION	Во время перемещения CP ось

Код ошибки	Причины	Описание
		вращения нарушила допустимый диапазон осей. Перемещение CP не в пределах рабочей зоны
12136	SMC_CP_COMPUTE_TARGET_DISCONTINUITY	Из-за разрыва в конечном положении (внутренняя ошибка) нельзя рассчитать заданную траекторию
12137	SMC_CP_TRACTION_NOT_SMOOTH	Траектория не является плавной, и конечное состояние фазы отличается от текущего состояния (внутренняя ошибка)
12138	SMC_CP_ERROR_CREATING_PARAM_TRANSFORM	Невозможно создать преобразование параметров (внутренняя ошибка)
12139	SMC_CP_PTP_DATA_NUMBER_OF_AXES	Ошибка при создании инвариантного к траектории элемента PTP (внутренняя ошибка)
12140	SMC_CP_NEGATIVE_PATH_VELOCITY	Ошибка при расчете новой траектории: Скорость перемещения по траектории после этапа 1 (положительное ускорение длиной dTau1) отрицательна (внутренняя ошибка)
12141	SMC_CP_TRANSITIONING_FROM_SINGLE_AXIS_MOVEMENT_NOT_SUPPORTED	Переход от перемещения одной оси к перемещению по непрерывной траектории не поддерживается
12142	SMC_CP_PLANNER_NO_PROGRESS	Нет прогресса в работе планировщика CP. Это внутренняя ошибка, которая может привести к появлению очень острых углов (например, при смешивании коротких элементов или малых угловых

Код ошибки	Причины	Описание
		расстояний)
12143	SMC_CP_INTERNAL_EVAL_CACHE_ERROR	Внутренняя ошибка в кэше оценки планировщика CP.

Профессиональное оборудование для промышленной автоматизации



Официальный дистрибьютор:

PROMPOWER

www.prompower.ru