

Программируемый логический контроллер

PROMPOWER

серии PMP20

Руководство по эксплуатации
(Программное)



**PROM
POWER**

Введение

Общие описания

В данном руководстве представлено программной части ПЛК RMP20. Внимательно прочитайте данное руководство перед использованием. Передайте это руководство конечным пользователям.

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 гарантируем 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. Краткое описание программирования.....	8
1.1 Особенности ПЛК	8
1.2 Язык программирования	10
1.3 Режим программирования	11
2. Программные компоненты	13
2.1 Краткое описание программных компонентов	13
2.2 Структура программных компонентов	18
2.3 Список программных компонентов	22
2.4 Регистры входов/выходов (X, Y)	26
2.5 Вспомогательный меркер (M, HM, SM)	28
2.6 Указатель состояния (S, HS)	29
2.7 Таймер (T, HT)	30
2.8 Счетчик (C, HC, HSC)	35
2.9 Регистр данных (D, HD, SD, HSD)	42
2.10 Flash-регистры (FD, SFD, FS)	47
2.11 Константа	48
2.12 Принцип программирования	50
3. Основные программные инструкции	54
3.1 Список основных инструкций	54
3.2 [LD], [LDI], [OUT]	58
3.3 [AND], [ANI]	59
3.4 [OR], [ORI]	60
3.5 [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF]	61

3.6	[LDD], [LDDI], [ANDD], [ANDDI], [ORD], [ORDI], [OUTD]	63
3.7	[ORB]	64
3.8	[ANB].....	66
3.9	[MCS], [MCR]	67
3.10	[ALT]	68
3.11	[PLS], [PLF].....	69
3.12	[SET], [RST]	70
3.13	[CNT], [CNT_D], [DCNT], [DCNT_D], [RST] для Счетчиков.....	72
3.14	[TMR], [TMR_A] для таймеров.....	73
3.15	[END]	74
3.16	[GROUP], [GROUPE].....	75
4.	Прикладные инструкции.....	77
4.1	Список инструкций по применению	77
4.2	Метод чтения прикладных инструкций.....	85
4.3	Управление потоком выполнения программы	88
4.4	Функция сравнения данных	101
4.5	Инструкции перемещения данных.....	108
4.6	Инструкции по работе с данными.....	130
4.7	Команды сдвига	151
4.8	Преобразование данных.....	166
4.9	Операции с числами с плавающей точкой.....	197
4.10	Команды часов реального времени	227
5.	Высокоскоростной счетчик (hsc).....	250
5.1	Обзор функций	250

5.2	Режимы HSC	251
5.3	Диапазон HSC.....	253
5.4	Подключение входа HSC	253
5.5	Назначение портов HSC	253
5.6	Настройка удвоения частоты счета импульсов фаз А и В.....	261
5.7	Команды HSC	262
5.8	Пример HSC.....	270
5.9	Прерывания HSC.....	273
6.	Функции связи.....	295
6.1	Обзор.....	296
6.2	COM-порт	296
6.3	Связь по протоколу MODBUS.....	303
6.4	Связь в свободном формате	348
6.5	Признак и регистр связи.....	365
6.6	Считывание и запись параметров последовательного порта	371
7.	Функция ПИД-регулирования.....	378
7.1	Введение в ПИД	378
7.2	Формат команды.....	379
7.3	Настройка параметров.....	381
7.4	Режим автоматической регулировки	391
7.5	Расширенный режим.....	395
7.6	Области применения.....	396
7.7	Применение	397
8.	Функциональные блоки на языке С	403

8.1	Обзор.....	403
8.2	Формат команды.....	403
8.3	Последовательность действий.....	405
8.4	Импорт и экспорт функций.....	409
8.5	Редактирование функциональных блоков	411
8.6	Пример программы.....	413
8.7	Новые функции.....	416
8.8	Библиотека функций	419
8.9	Указания по применению	438
8.10	Обеспечение качества при использовании языка С	441
8.11	Таблица функций.....	443
9.	Последовательность BLOCK	449
9.1	Описание принципа BLOCK.....	449
9.2	Вызов BLOCK.....	450
9.3	Редактирование команды в составе BLOCK.....	456
9.4	Схема запуска BLOCK.....	461
9.5	Правила редактирования команд BLOCK.....	464
9.6	Команды, связанные с BLOCK.....	466
9.7	Бит признака и регистр BLOCK.....	474
10.	Команды специальных функций	475
10.1	Широтно-импульсная модуляция [PWM].....	475
10.2	Измерение частоты [FRQM].....	480
10.3	Прерывания [EI], [DI], [IRET]	490
10.4	Управление несколькими станциями [MSC].....	499

11. Общие вопросы и ответы.....	510
Приложение.....	529
Приложение 1 Специальное вспомогательное реле.....	529
Приложение 2 Специальный регистр данных.....	541
Приложение 3 Перечень специальных FLASH регистров.....	559
Приложение 4. Таблица конфликтов ресурсов ПЛК	565
Приложение 5 Таблица настройки функций ПЛК	566

1. Краткое описание программирования

ПЛК RMP20 с программируемой логикой, обеспечивает выполнение логических операций, записанных в память контроллера в виде коммутационной программы пользователя. В этой главе Вы узнаете о возможностях ПЛК

1.1 Особенности ПЛК

1) Языки программирования

ПЛК серии RMP20 поддерживают два языка программирования – список инструкций IL и лестничную диаграмму LD, причем эти два вида языка могут образовываться друг в друга.

2) Безопасность программы

Существует возможность установить пароль для пользовательской программы для избежания кражи или её неправомерного изменения. Пароль может защитить авторские права пользователя, в то же время ограничить загрузку во избежание рисков изменения программы по ошибке.

В серии RMP20 добавлен новый регистр FS. (Для разных моделей проверьте диапазон регистров FS в программе PROMPOWER PLC Studio в мониторе данных, общий диапазон - FS0~FS47). FS используется для защиты авторских прав пользователя. Значение FS возможно записать по средствам инструкций Modbus, считать содержимое по Modbus невозможно. FS не может сравниваться с регистром, а только с константой. Значение не может быть прочитано. Регистр D, HD... может быть заменен на FS.

3) Комментарии к программе

Если вы планируете передать свою программу другим разработчикам, использовать её спустя некоторое время или описывать обширную логику с множеством функций и переменных, Вам могут быть полезными некоторые подсказки. Такими подсказками являются комментарии программ и регистров.

4) Функция смещения

Есть возможность использования подписи смещения (X3[D100], M10[D100], D0[D100]) регистрам.

Например, когда $D100 = 9$, $X3[D100] = X[3+9] = X14$; $M10[D100] = M19$, $D0[D100] = D9$.

5) Богатый набор функций

ПЛК серии PMP20 имеет достаточное количество базовых инструкций, включая базовое управление последовательностью исполнения инструкций, перемещение и сравнение данных, арифметические операции, логическое управление, управление циклом обработки, сдвиг и т.д.

ПЛК серии PMP20 также поддерживают функции прерывания, импульсного управления, осциллографа, часов реального времени, ПИД-регулятора и т.д.

6) Функциональный блок на языке C

ПЛК серии PMP20 поддерживают язык C; пользователи могут вызывать программу на языке C в представлении лестничной диаграммы. Эта функция повышает эффективность программирования.

7) Остановка ПЛК при перезагрузке

ПЛК серии PMP20 поддерживают функцию "Stop PLC when Reboot". Если во время работы ПЛК возникла серьезная проблема, функция "Stop PLC when Reboot" поможет немедленно отключить все выходы.

Кроме того, если параметры COM-порта были изменены по ошибке, эта функция также поможет подключить ПЛК к ПК.

8) Функция связи

ПЛК серии PMP20 имеет множество режимов связи, таких как Modbus-RTU, Modbus-ASCII. При изменении параметров COM-порта новые параметры будут активны без перезапуска ПЛК. Перед выполнением инструкций Modbus может быть добавлено время ожидания.

1.2 Язык программирования

1.2.1 Тип

ПЛК серии RMP20 поддерживают два типа языка программирования:

IL - Список инструкций

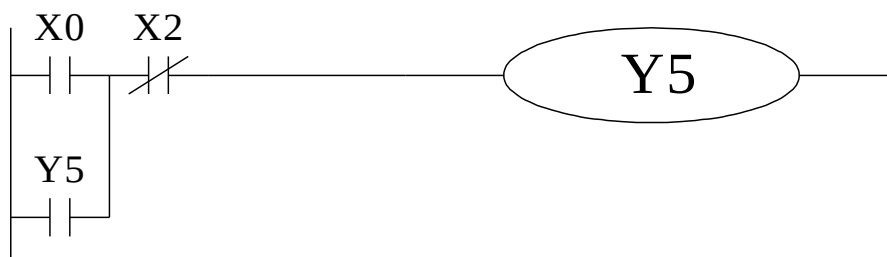
RMP20 предоставляет возможность использовать язык IL, который визуально напоминает ассемблер. Данный язык имеют форму записи функция-операнд и выполняется по шагам.

Например:

Шаг	Инструкция	Операнд
0	LD	X000
1	OR	Y005
2	ANI	X002
3	OUT	Y005

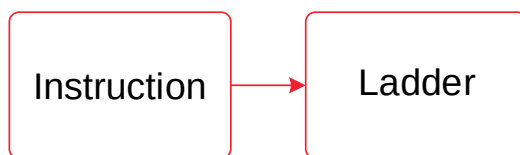
Также есть возможность использовать язык "Лестничная диаграмма" (Ladder). Этот метод использует катушки и контакты для представления управляющей программы. Лестничная диаграмма проста для понимания и может быть использована для онлайн-мониторинга исполнения логики программы ПЛК.

Например:



1.2.2 Переключение между языками

Эти два вида языка программирования могут быть преобразованы друг в друга.

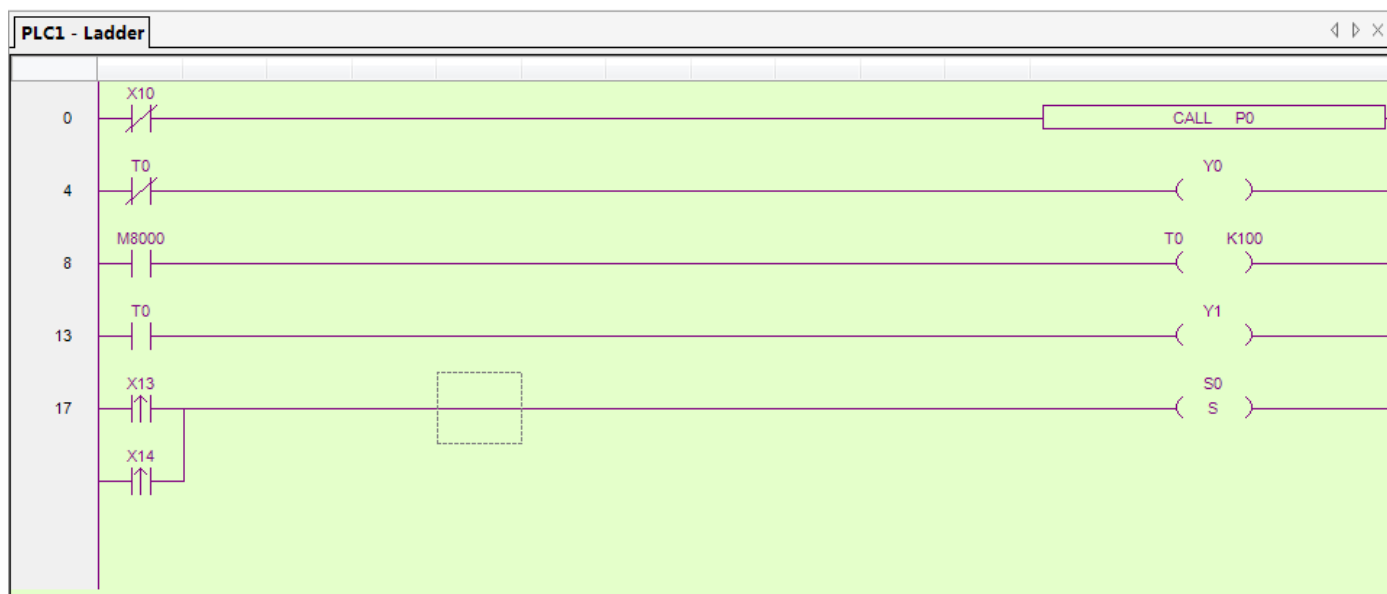


Для этого нажмите на иконку переключения в верхней области интерфейса программы PROMPOWER PLC Studio .

1.3 Режим программирования

Ввод инструкций напрямую

Ниже представлен внешний вид рабочей области PROMPOWER PLC Studio.



Конфигурация инструкции

Некоторые инструкции могут оказаться сложны в использовании, например, инструкции для настройки импульсных выходов, ПИД-регулятора и т.п. В программе PROMPOWER PLC Studio есть окно конфигурации для этих специальных инструкций. Пользователю просто нужно ввести параметры, не запоминая синтаксис сложных инструкций. Ниже представлен пример с окном конфигурации импульсных выходов.

multi section pulse output

Data start address:

D0

user params address:

D100

System params:

K1

Output:

Y0

Mode:

relative

Start execute section count:

0

Pulse Config

...

Add

Delete

Upwards

Downwards

	frequence	pulse count	wait condition	wait register	jump register

used space:

Read From PLC

Write To PLC

OK

Cancel

2. Программные компоненты

В главе 1 мы кратко познакомили Вас с языками программирования. Однако самым важным элементом в программе являются операнды. В этой главе мы опишем функции и методы использования операнд и других программных компонентов.

2.1 Краткое описание программных компонентов

В ПЛК есть множество реле, таймеров и счетчиков. Все они имеют бесчисленное количество NO контактов (нормально открытых/разомкнутых) и NC (нормально закрытых/замкнутых). Соединив контакты с катушками, Вы получите последовательную схему управления. Далее мы познакомимся с программными компонентами.

1) Регистр входа (X)

- Функция регистра входа

Входные регистры используются для приема внешнего сигнала ВКЛ/ВЫКЛ, подпись - X.

- Принцип присвоения адресов

- В каждом модуле ЦПУ адрес X имеет восьмеричную систему, например, X0~X7, X10~X17 ...
- Адрес модуля расширения: модуль 1-й начинается с X10000, модуль 2-й начинается с X10100.... К RMP20 можно подключить до 16 модулей расширения.
- Плата расширения BD: BD 1 начинается с X20000; к ПЛК на 30 точек можно подключить одну BD-плату, а к ПЛК на 60 точек - две BD-платы.
- Адрес левого сетевого модуля ED начинается с X30000 в восьмеричной системе, ПЛК серии RMP20 поддерживают левый расширенный модуль ввода/вывода ED.

- Дополнительная информация

Для входов возможно использовать цифровой фильтр. Пользователи могут изменить параметры фильтрации, записав специальный регистр SFD0, значение по умолчанию - 10 мс, диапазон изменения: 0 ~ 1000 мс.

Входные регистры, адреса которых выходят за адреса регистров физических входов, можно представить как вспомогательные меркеры.

2) Регистр выхода (Y)

- Функция регистров выходов

Регистры выхода — это интерфейс для управления внешними нагрузками, обозначение - Y.

- Принцип присвоения адресов

В каждом модуле ЦПУ адрес Y имеет восьмеричную систему, например, Y0~Y7, Y10~Y17. Адрес модуля расширения: 1-й модуль начинается с Y10000, 2-й модуль начинается с Y10100...

К RMP20 можно подключить до 16 модулей расширения.

Адреса платы BD, начинаются с X20000 – также в восьмеричной системе. ПЛК с 24-32 точками может подключить одну плату BD, ПЛК с 48-60 точками - две платы BD.

Адрес левого расширенного модуля ED, начиная с Y30000 - в восьмеричной системе. ПЛК серии RMP20 поддерживают левый расширенный модуль ввода и вывода ED.

- Дополнительная информация

Выходные регистры, адреса которых выходят за адреса регистров физических выходов, можно представить как вспомогательные меркеры.

3) Вспомогательные меркеры (M, HM)

- Функция вспомогательных меркеров

Вспомогательные меркеры — это внутренние регистры ПЛК, обозначение - M и HM.

- Принцип назначения адресов

В модулях ЦПУ адрес в десятичной форме.

- Дополнительная информация

Этот тип регистров отличается от регистров входа/выхода, они не могут управлять внешней нагрузкой и принимать внешний сигнал, а используются только в программе.

Энергонезависимые вспомогательные меркеры HM могут сохранять свое состояние (ВКЛ/ВЫКЛ), когда питание ПЛК отключено.

4) Указатель состояния шага (S, HS)

- Функция указателя состояния шага

Используются в качестве реле в представлении лестничной диаграммы, обозначение - S, HS.

- Принцип назначения адресов

В модулях ЦПУ адрес в десятичной форме.

- Дополнительная информация

Если они не используются в качестве номера операции, их можно применять как вспомогательные реле, программируя как обычные контакты/катушки. Кроме того, они могут использоваться в качестве сигнализаций при мониторинге / внешней диагностике.

5) Таймер (Т, НТ)

- Функции таймеров

Таймеры используются для накопления импульсов времени, например, 1 мс, 10 мс, 100 мс и т.д. При достижении заданного значения срабатывают выходные контакты, обозначаемые буквами Т и НТ.

- Принцип назначения адресов

В модулях ЦПУ адрес в десятичной форме.

Подробности см. в главе 2-2.

- Импульс времени

Существует три импульса таймера: 1 мс, 10 мс и 100 мс. Например, 10 мс означает накопление 10 мс импульсов.

- Режимы работы таймера

Таймер имеет два режима: таймер с накоплением означает, что даже если катушка таймера в состоянии ВЫКЛ, таймер будет сохранять текущее значение; в то время как таймер без накопления означает, что, когда значение накопления достигает установленного значения, выход срабатывает, значение сбрасывается в 0.

6) Счетчик (С, НС)

В соответствии с различными целями применения, счетчики бывают разных типов:

- Для внутреннего подсчета (для общего использования/с фиксацией при отключении питания (энергонезависимые))

16-битный счетчик: для инкрементного счета диапазон счета составляет 1~32,767

32-битный счетчик: для инкрементного счета, диапазон счета 1~2,147,483,647

Эти счетчики предназначены для внутреннего использования в программе ПЛК. Скорость срабатывания составляет один цикл сканирования или больше.

- Для высокоскоростного подсчета (с фиксацией при отключении питания(энергонезависимые))

32-битный счетчик: диапазон счета -2,147,483,648 ~ +2,147,483,647

(Однофазный счет приращения, двухфазный счет). Для специальных входных клемм.

Высокоскоростной счётчик не будет зависеть от периода сканирования ПЛК. В однофазном (инкрементном) режиме он может считать максимум 80 КГц импульсов; в двухфазном режиме - максимум 50 КГц импульсов.

- Принцип назначения адресов

В модулях ЦПУ адрес таймера в десятичной форме.

7) Регистр данных (D, HD)

- Функция регистров данных

Регистры данных используются для хранения данных, их обозначение - D и HD.

- Принцип назначения адресов

Регистры данных в ПЛК серии RMP20 имеют разрядность 16 бит (старший бит - бит знака), объединение двух регистров данных вместе позволяет обрабатывать данные разрядностью 32 бита (старший бит - бит знака).

- Дополнительная информация

Как и другие программные компоненты, регистры данных также бывают обычного типа и энергонезависимые.

8) Регистр FlashROM -памяти (FD)

- Функции Flash-регистров (FlashROM)

Flash-регистры используются для хранения данных, обозначение - FD.

- Принцип назначения адресов

В модулях ЦПУ адрес Flash-регистров имеет десятичную форму;

- Дополнительная информация

Даже при отсутствии заряда батареи, эта область может сохранять данные. Таким образом, в этой области можно хранить важные параметры. Flash-регистры можно записать около 1 000 000 раз, и на запись требуется время. Частая запись может привести к необратимым повреждениям FD.

9) Специальный секретный регистр (FS)

- Функция секретного регистра

Секретные регистры используют часть области Flash-регистров, обозначаются символом FS. Значения в регистре FS могут быть записаны, но не могут быть прочитаны, поэтому они могут использоваться для защиты прав интеллектуальной собственности пользователей.

- Принцип распределения адресов

В модуле ЦПУ регистры FS адресуются в десятичных числах.

Поскольку количество регистров FS у разных типов ПЛК может отличаться, обратитесь к "Начальным настройкам ПЛК" (PLC Initial Settings), отображаемым в онлайн-программе для ПЛК (регистры FS0-FS47).

- Дополнительная информация

Данная область хранения может запоминать данные, даже если батарея разряжена, поэтому ее можно использовать для хранения важных параметров процесса. FS можно записать около 1 000 000 раз, и каждый раз запись занимает больше времени. Частая запись приведет к необратимому повреждению FS, поэтому не рекомендуется часто записывать данные. При использовании инструкции MOV для передачи данных в FS, передача работает только при условии нарастающего фронта импульса.

Значение в регистр FS может быть записано, но не может быть прочитано (всегда возвращается в 0). Значение не может быть сравнено с регистром в программном обеспечении ПЛК, только с константой, поэтому фактическое значение регистра не может быть прочитано.

10) Константа (B) (K) (H)

“B” означает двоичную систему счисления, “K” - десятичную, “H” - шестнадцатеричную. Они используются для установки значений таймеров и счетчиков или операндов инструкций.

Например, шестнадцатеричный FF будет HFF.

2.2 Структура программных компонентов

2.2.1 Структура памяти

В ПЛК серии PMP20 существует множество регистров. Помимо регистров D, HD, FlashROM, мы также можем объединять биты в регистры.

1) Регистр данных D, HD, FD

Для общего использования - 16 бит.

Для общего использования - 32 бита (объедините два непрерывных 16-битных регистра).

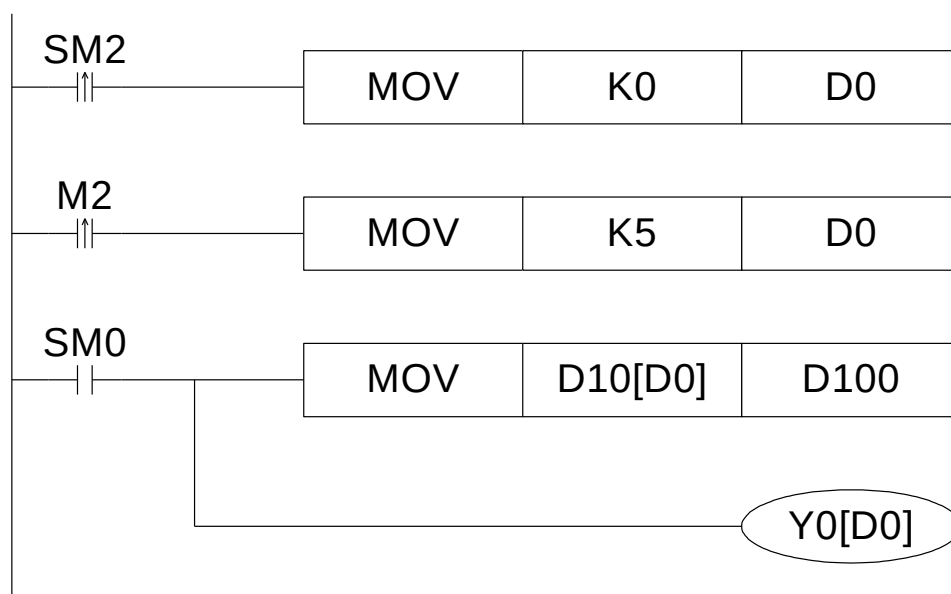
Для общего использования - 64 бита (объедините два 32-битных регистра, но адреса должны быть последовательными).

При энергонезависимом использовании нельзя изменять диапазон.

Для специального использования, занятого системой, регистры нельзя использовать для общих параметров инструкции.

Для смещенного использования:

Форма: $D_n[D_m]$, $HD_n[D_m]$, $X_n[D_m]$, $Y_n[D_m]$, $M_n[D_m]$ и т.д.



Когда $D0 = 0$, $D100 = D10$, $Y0$ находится в состоянии ВКЛ.

Когда $M2$ переключается с ВЫКЛ на ВКЛ, $D0 = 5$, тогда $D100 = D15$, $Y5$ переходит в состояние ВКЛ.

Таким образом, $D10[D0] = D[10+D0]$, $Y0[D0] = Y[0+D0]$.

Смещение слова, объединенное по битам: $DXn[Dm]$ представляет собой $DX[n+Dm]$.

Регистры со смещением: смещение может быть представлено регистром D, HD.

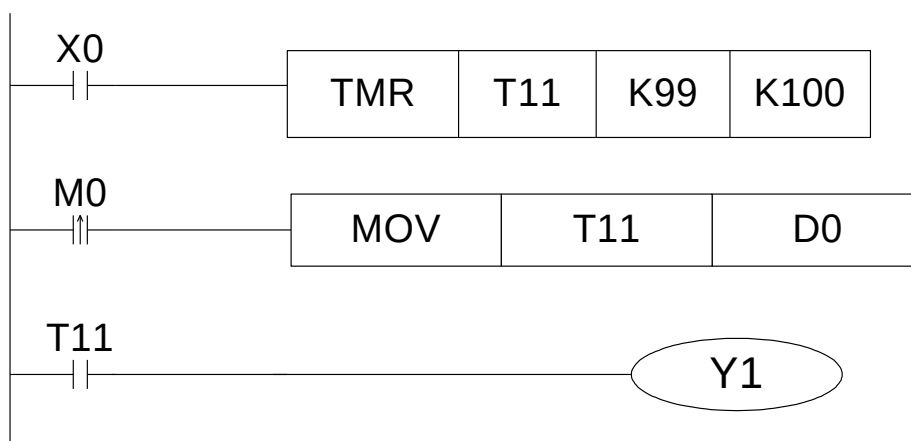
2) Таймер T, HT/Счетчик C, HC

Для общего использования - 16 бит. Представляют текущее значение таймера/счетчика;

Для общего использования - 32 бита (объедините два непрерывных 16-битных регистра).

Операнды описываются как "буква+адрес", например, T10, C11, HT10, HC11.

Например:



В приведенном выше примере MOV T11 D0: T11 представляет регистр слов; LD (контакт) T11: T11 представляет регистр битов.

3) Регистр FlashROM FD

Для энергонезависимого использования - 16 бит.

Для энергонезависимого использования - 32 бита, (объедините два непрерывных 16-битных регистра).

Регистры не могут быть использованы в качестве обычных (неспециальных/несистемных) параметров команд

4) Регистр, объединенный по битам

Для общего пользования - 16 бит, (объединить 16 бит).

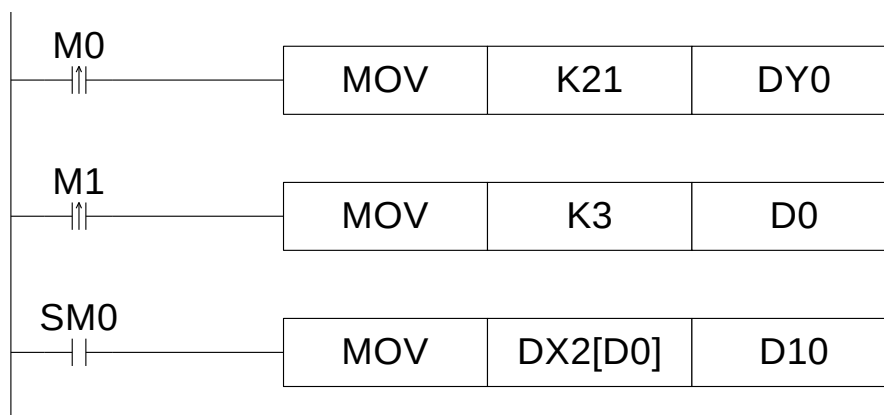
Регистры, которые можно объединить в слова, это: X, Y, M, S, T, C, HM, HS, HT, HC.

Формат: добавьте "D" перед регистром. Например, DM10, представляет собой 16-битный регистр от M10~M25.

Представление 16 бит, начиная с DXn, не выходя за пределы диапазона.

Объединенное по битам слово, не подлежит битовой адресации.

Например:



Когда M0 переключается из состояния ВЫКЛ в состояние ВКЛ, значение в слове, которое объединяет Y0~Y17, равно 21, т.е. Y0, Y2, Y4 становятся ВКЛ.

Перед активацией M1, если D0 = 0, DX2[D0] представляет слово, объединяющее X2~X21.

Если M1 переключается из состояния ВЫКЛ в состояние ВКЛ, D0 = 3, то DX2[D0] представляет собой слово, объединенное X5~X24.

2.2.2 Структура битовых регистров

К битовым программным регистрам относятся X, Y, M, S, T, C, HM, HS, HT, HC. Кроме того, бит определенного регистра также может быть использован в качестве битового регистра.

1) Реле

Входной регистр (реле) X, восьмеричная форма.

Выходной регистр (реле) Y, восьмеричная форма.

Вспомогательный меркер M, HM, S, HS; десятичная форма.

Вспомогательное меркер T, HT, C, HC, десятичная форма. Метод представления такой же, как и у других регистров, поэтому нужно определить, является ли регистр словом или битом, в соответствии с инструкцией.

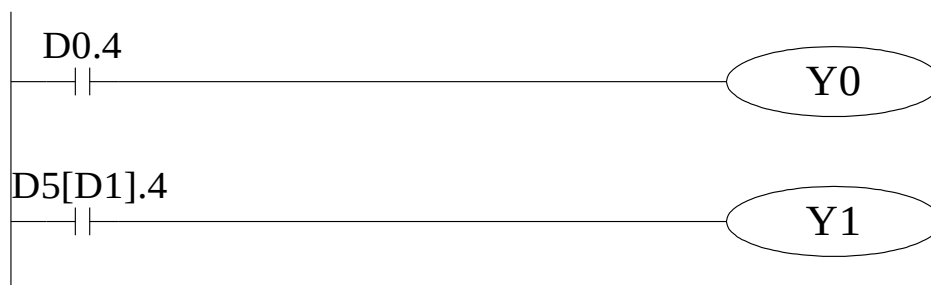
2) Бит определенного регистра

Метод представления: $D_n.m$ ($0 \leq m \leq 15$): например, $D_{10}.2$ означает второй бит D_{10} .

Метод представления бита со смещением: $D_n[D_m].x$

Бит регистра не может снова скомпоноваться в словесный регистр;

Например:



$D_{0.4}$ означает, что когда четвертый бит D_0 равен 1, установите Y_0 в состояние ВКЛ.

$D_5[D_1].4$ означает битовую адресацию со смещением, если $D_1 = 5$, то $D_5[D_1]$ означает четвертый бит D_{10} .

2.3 Список программных компонентов

2.3.1 Список регистров

Список регистров ПЛК серии RMP20:

Имя	Описание	Диапазон		Кол-во каналов ввода/вывода	
		30 каналов ввода/вывода	60 каналов ввода/вывода	30	60
X	Входы	X0~X17	X0~X43	16	36
Y	Выходы	Y0~Y15	Y0~Y27	14	24
X	Входы ^{※3}	X10000~X10077 (модуль расширения #1) X11700~X11777 (модуль расширения #16)		1024	
Y	Выходы ^{※3}	Y10000~Y10077 (модуль расширения #1) Y11700~Y11777 (модуль расширения #16)		1024	
X	Входы ^{※4}	X20000~X20077 (#1 расширение BD) X20100~X20177 (#2 расширение BD)		128	
Y	Выходы ^{※4}	Y20000~Y20077 (#1 расширение BD)		128	

		Y20100~Y20177 (#2 расширение BD)	
X	Входы ^{※5}	X30000~X30077 (#1 расширение ED)	64
Y	Выходы ^{※5}	Y30000~Y30077 (#1 расширение ED)	64
M	Внутренний меркер	M0~M69999	70000
HM		HM0~HM11999 ^{※1}	12000
SM		специального назначения SM0~SM4999 ^{※2}	5000
S	Управление потоком выполнения	S0~S7999	8000
HS		HS0~HS999 ^{※1}	1000
T	Таймер	T0~T4999	5000
HT		HT0~HT1999 ^{※1}	2000
ET		Точный таймер ET0~ET39	40
C	Счетчик	C0~C4999	5000
HC		HC0~HC1999 ^{※1}	2000
HSC		высокоскоростной счётчик HSC0~HSC39	40
D	Регистр данных	D0~D69999	70000
HD		HD0~HD24999 ^{※1}	25000
SD		специального назначения SD0~SD4999	5000
HSD		специального назначения HSD0~HSD1023 ^{※2}	1024

FD	Flash-регистр (FlashROM)	FD0~FD8191	8192
SFD	Специальный Flash-регистр	специального назначения SFD0~SFD5999※2	6000
FS	Специальный секретный регистр	FS0~FS47	48
ID※6	Основной модуль ЦПУ	ID0~ID99	100
	Модуль расширения	ID10000~ID10099 (модуль расширения #1) ID11500~ID11599 (модуль расширения #16)	1600
	Плата расширения BD	ID20000~ID20099 (#1 расширение BD) ID20100~ID20199 (#2 расширение BD)	200
	Сетевой модуль ED	ID30000~ID30099 (#1 расширение ED)	100
QD※7	Основной модуль ЦПУ	QD0~QD99	100
	Модуль расширения	QD10000~QD10099 (модуль расширения #1) QD11500~QD11599 (модуль расширения #16)	1600
	Плата расширения BD	QD20000~QD20099 (#1 расширение BD)	200

		QD20100~QD20199 (#2 расширение BD)	
	Сетевой модуль ED	QD30000~QD30099 (#1 расширение ED)	100
SEM	Специальный регистр инструкции блока последовательности BLOCK WAIT	SEM0~SEM31	32

※1 □ Область памяти - это энергонезависимая область (Примечание: энергонезависимая область ПЛК серии PMP20 не может быть изменена).

※2 Специальное использование (обслуживание без отключения питания) относится к регистрам, занимаемым системой, для особых целей.

※3 Назначение адреса ввода/вывода (восьмеричное) модуля расширения, который может использоваться в качестве вспомогательного меркера, когда модуль расширения не подключен. (к PMP20 можно подключить до 16 модулей одновременно).

※4 Адреса ввода/вывода BD (восьмеричные), могут использоваться в качестве вспомогательных меркеров, когда BD не подключен.

※5 Адреса ввода/вывода ED (восьмеричные), могут использоваться в качестве вспомогательных меркеров, когда ED не подключен.

※6 Адрес аналогового входа программного компонента, может использоваться как вспомогательный регистр, когда модуль расширения не подключен.

※7 Адрес программного компонента аналогового выхода, может использоваться как вспомогательный регистр, когда модуль расширения не подключен.

※8 Указанный выше диапазон регистров соответствует диапазону ПЛК в режиме связи X-NET. В режиме связи MODBUS некоторые из них не могут быть записаны или считаны. Конкретный допустимый диапазон показан в главе 6-2-3.

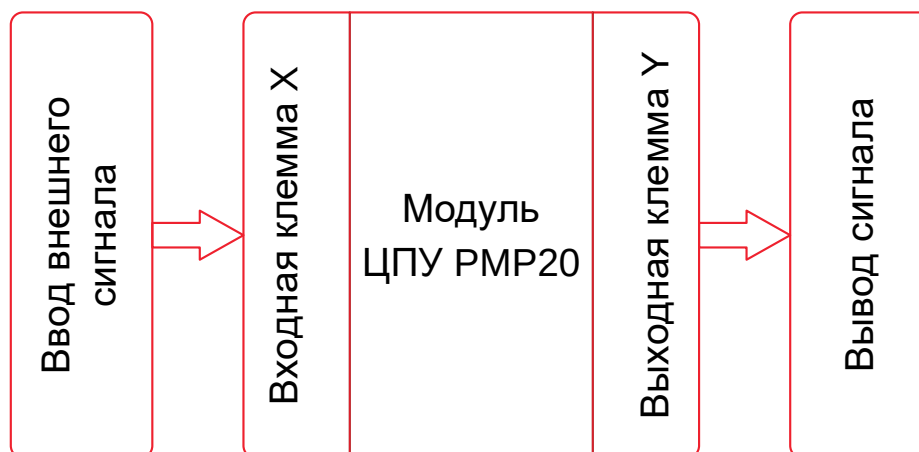
2.4 Регистры входов/выходов (X, Y)

1) Список регистров

Все входы/выходы ПЛК серии RMP20 имеют восьмеричную форму, номера каждой серии приведены ниже:

Серия	Имя	30	60	30	60
		Каналов ввода/вывода	Каналов ввода/вывода		
RMP20	X	X0~X17	X0~X43	16	36
	Y	Y0~Y15	Y0~Y27	14	24

2) Функция



Регистр входа X

Входные клеммы ПЛК используются для приема внешнего сигнала. Внутренние входные реле представляют собой опторазвязку для соединения ПЛК и входных клемм.

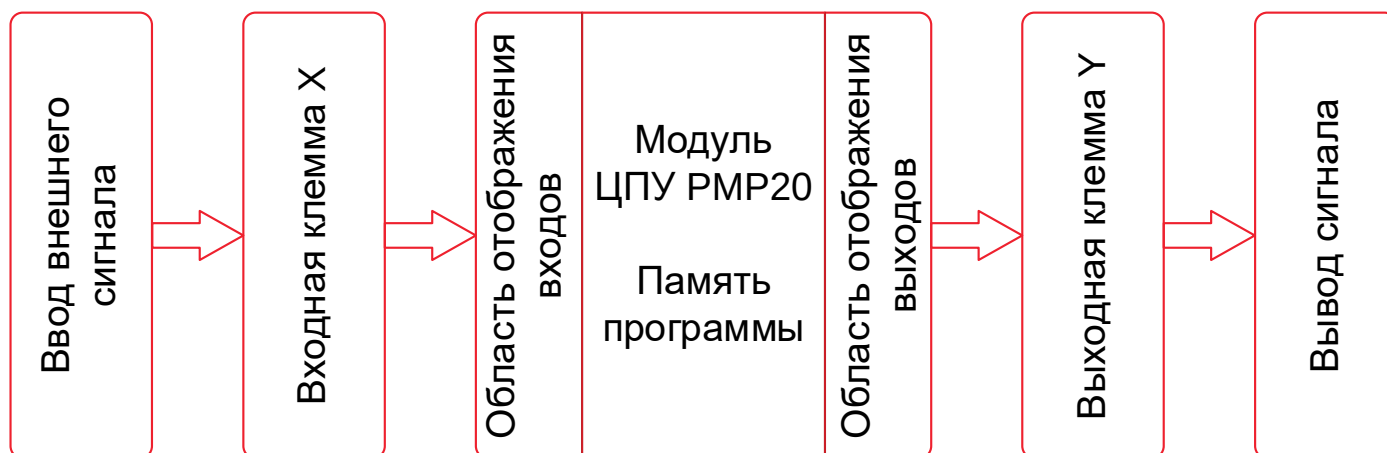
Входы, которые не подключены к внешним устройствам, могут использоваться как вспомогательные меркеры.

Выходной регистр Y

Выходные клеммы ПЛК могут использоваться для передачи сигналов на внешние нагрузки. Внутри ПЛК внешние выходные контакты реле (контакты как реле, так и транзисторов) соединяются с выходными клеммами.

Выходы, которые не подключены к внешним устройствам, могут представляться как вспомогательные меркеры.

3) Порядок исполнения



Обработка ввода

Перед тем как ПЛК выполнит программу, происходит считывание состояний ВКЛ/ВЫКЛ каждой входной клеммы в область памяти.

Когда программа работает, даже если входной сигнал изменился, содержимое в области входов не изменится до наступления следующего периода сканирования.

Обработка вывода

После выполнения всех инструкций происходит перенос состояний ВКЛ/ВЫКЛ выходов Y в область памяти выходов.

Выходные контакты будут удерживать значение в соответствии выходными программными компонентами.

2.5 Вспомогательный меркер (М, НМ, SM)

2.5.1.1 Список меркеров

Все вспомогательные меркеры в ПЛК серии RMP20 представлены в десятичной форме, см. следующую таблицу.

Серия	Имя	Диапазон		
		Нормальные меркеры	Энергонезависимые меркеры	Специальные меркеры
RMP20	M HM SM	M0~M69999	HM0-HM11999	SM0~SM4999

В ПЛК часто используются вспомогательные меркеры. Катушка этого типа аналогична катушкам выходов.

Вспомогательные реле М и НМ оснащены бесчисленными контактами с состояниями ВКЛ/ВЫКЛ. Их можно использовать свободно, но этот тип контактных может приводить в действие внешние нагрузки.

- Для общего использования

Вспомогательные реле этого типа можно использовать только как обычные вспомогательные меркеры. Т.е. если во время работы внезапно отключится питание, меркеры не сохраняют своё значение.

Обычные вспомогательные меркеры не могут быть использованы как энергонезависимые, но их диапазон может быть изменен.

- Для энергонезависимого использования

Вспомогательные меркеры для энергонезависимого использования, даже если ПЛК находится в отключенном состоянии, могут сохранять своё состояние (ВКЛ/ВЫКЛ).

Область энергонезависимого хранения не может быть изменена.

Энергонезависимые вспомогательные меркеры обычно используются для запоминания состояния перед отключением питания, затем, когда ПЛК снова включается, состояние восстанавливается.

- Для специального использования

Специальные меркеры - это некоторые реле, которые имеют особые значения или функции, начиная с SM0.

У специальных меркеров есть две функции: первая - для управления другими программными компонентами, вторая - для специальных применений.

Например: SM2 - начальный импульс, активируется только в момент запуска, SM34 - "все выходы отключены". Специальные вспомогательные меркеры не могут использоваться как обычные реле M.

Примечание:

Диапазон регистров, указанный выше, является актуальным диапазоном ПЛК (в том числе и в режиме связи X-NET). В режиме связи MODBUS некоторые регистры не могут быть прочитаны или записаны. Конкретный допустимый диапазон показан в главе 6-2-3.

2.6 Указатель состояния (S, HS)

Список адресов

Адреса указателей состояния ПЛК серии PMP20 имеют десятичную форму, адрес показан ниже.

Серия	Имя	Диапазон	
		Нормальный	Энергонезависимый
PMP20	S HS	S0~S7999	HS0~HS999

Функция

Указатель состояния (состояния шага) S и HS очень важны в лестничной диаграмме; они используются вместе с инструкцией "STL" в управлении потоком выполнения. Управление потоком выполнения может сделать программу понятной и легко модифицируемой.

- Для общего использования
После выключения питания ПЛК указатель S будет в состоянии ВЫКЛ.
- Для энергонезависимого использования
Реле HS могут сохранять состояние ВКЛ/ВЫКЛ даже при отключении питания ПЛК.

- Указатели состояния также имеют бесчисленное количество НО и НЗ контактов. Пользователи могут свободно использовать их в программе.

Примечание:

Диапазон регистров, указанный выше, является актуальным диапазоном ПЛК (в том числе и в режиме связи X-NET). В режиме связи MODBUS некоторые регистры не могут быть прочитаны или записаны. Конкретный допустимый диапазон показан в главе 6-2-3.

2.7 Таймер (Т, НТ)

Список адресов

Адреса таймеров в ПЛК серии PMP20 представлены в десятичной форме; см. следующую таблицу.

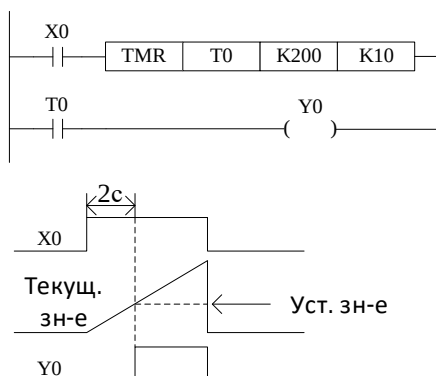
Серия	Имя	Диапазон		
		Нормальный	Энергонезависимый	Точный таймер
PMP20	Т НТ ЕТ	T0~T4999	НТ0~НТ1999	ЕТ0~ЕТ24

Функция

Таймеры накапливают импульсы длительностью 1 мс, 10 мс, 100 мс. Выходной контакт активируется, когда накопление достигает заданного значения.

Инструкция TMR предназначена для обычных таймеров. Установленное значение может быть константой (K) или регистром данных (D).

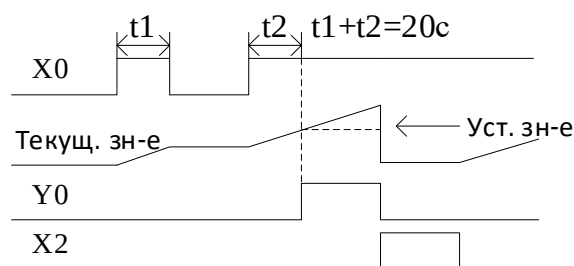
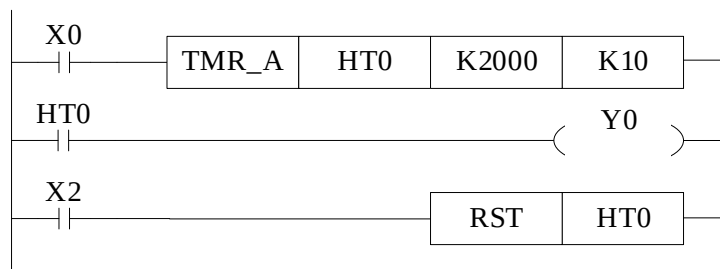
Обычный тип



Если X0 находится в состоянии ВКЛ, то T0 накапливает 10 мс импульсов (с учетом текущего значения); когда значение накопления достигает установленного значения K200, активируется выход таймера. Т.е. выход активируется через 2 с. Если X0 в состоянии ВЫКЛ, таймер сбрасывается, выход также сбрасывается.



Тип с накоплением (энергонезависимый)



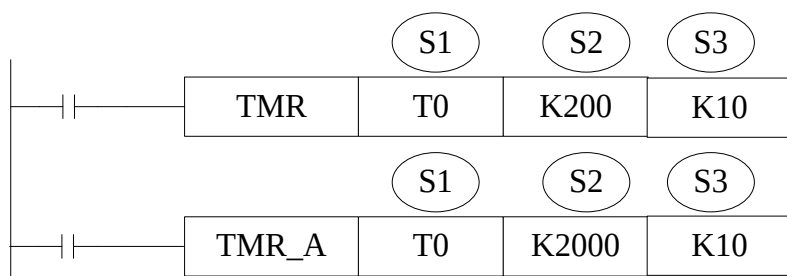
Если X0 включен (состояние ВКЛ), НТ0 накапливает 10-минутный импульс (с учетом текущего значения). Когда значение накопления достигает установленного значения K2000, активируется выход таймера.

Если во время работы таймера X0 внезапно перешел в состояние ВЫКЛ, значение таймера будет удержано. Если X0 снова ВКЛ, таймер продолжит работу.

Если ВКЛ. X2, таймер и выход будут сброшены.

Назначьте установленное значение

Формат инструкции



Не накапливается

Накопление

Сбросьте таймер и выход:



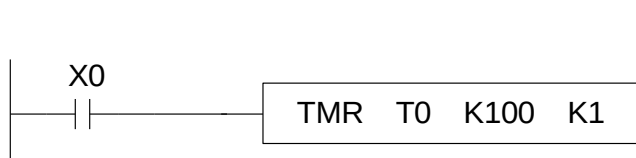
S1: таймер (T0, HT10)

S2: задайте время (например, K100)

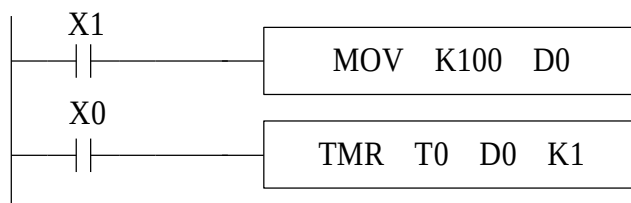
S3: единица времени (K1-1мс, K10-10мс, K100-100мс)

Зависимая от питания инструкция, значение не накапливаемое

(1) Единица времени - 1 мс, установленное время - K100, реальное время - 1 мс*100 = 0,1 с.

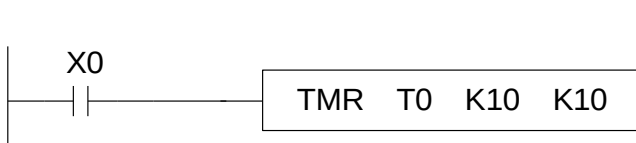


Установленное значение - постоянная K

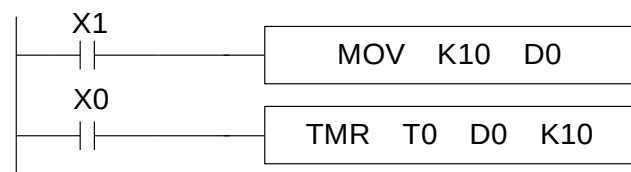


Установленное значение - регистр D

(2) Единица времени - 10 мс, установленное время - K10, реальное время - 10 мс*10 = 0,1 с.

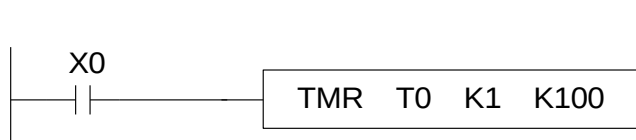


Установленное значение - постоянная K

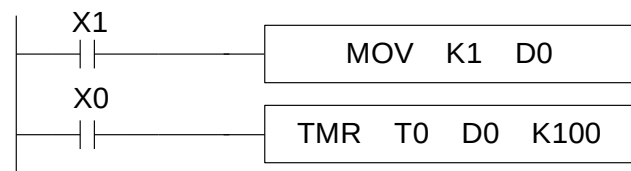


Установленное значение - регистр D

(3) Единица времени - 100 мс, установленное время - K1, реальное время - 100 мс*1 = 0,1 с.



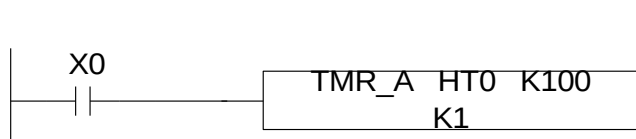
Установленное значение - постоянная K



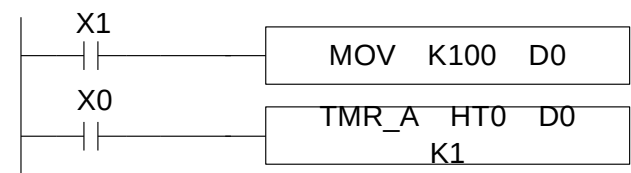
Установленное значение - регистр D

Энергонезависимая инструкция, значение накапливается

(1) Единица времени - 1 мс, установленное время - K100, реальное время - 1 мс*100 = 0,1 с.

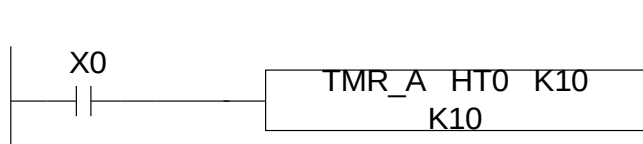


Установленное значение - постоянная K

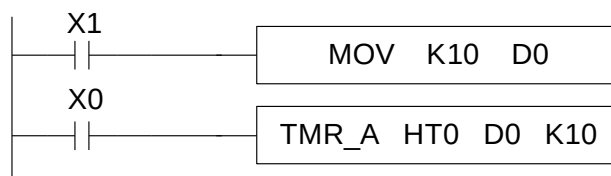


Установленное значение - регистр D

(2) Единица времени - 10 мс, установленное время - K10, реальное время - 10 мс*10 = 0,1 с.

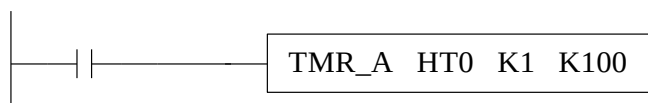


Установленное значение - постоянная K

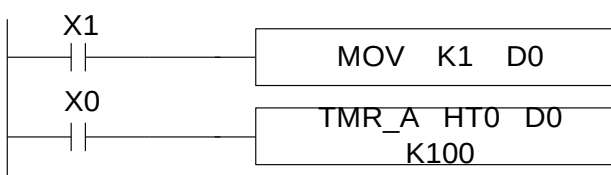


Установленное значение - регистр D

(3) Единица времени - 100 мс, установленное время - K1, реальное время - 100 мс*1=0,1 с.



Установленное значение - постоянная K



Установленное значение - регистр D

Примечания:

(1) Один и тот же таймер может использоваться как с накоплением, так и без, и его единица измерения времени также задается инструкциями как 1 мс, 10 мс или 100 мс.

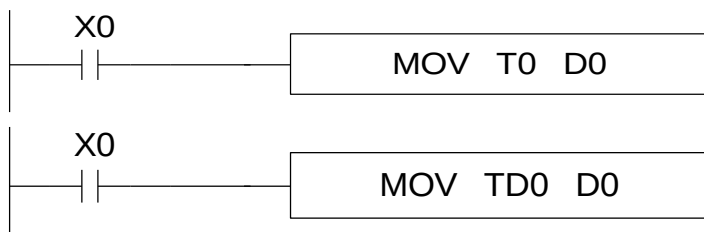
(2) Третий параметр инструкции может быть только в виде K1, K10 и K100. Не записывайте другие значения или регистры, кроме этих трех параметров. В противном случае, хотя программа может быть записана в программное обеспечение и загружена в ПЛК, инструкция по синхронизации не будет выполнена.

(3) Диапазон настройки константы K и фактическое значение настройки таймера приведены в следующей таблице.

Таймер	Диапазон K	Фактическое значение
Таймер 1 мс	1~32,767	0.001~32.767 с
Таймер 10 мс		0.01~327.67 с
Таймер 100 мс		0.1~3276.7 с

Переменная времени

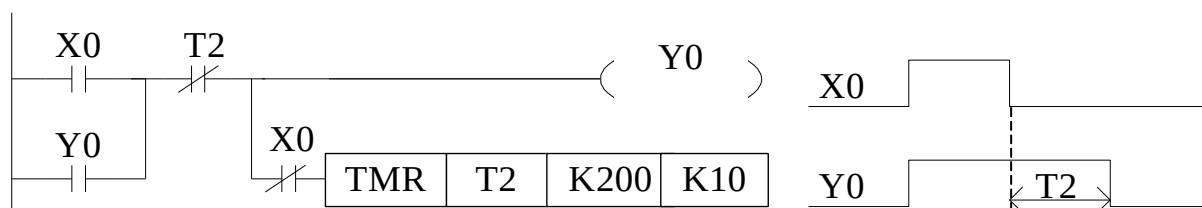
Значение времени хранится в регистре TD. Режим работы таймеров T0~T575 и HT0~HT95 - 16-разрядный линейно возрастающий. Диапазон времени составляет от 0 до 32767. Когда значение времени в TD достигнет 32767, таймер остановит отсчет времени и сохранит статус.



Эти две инструкции одинаковы. В первой инструкции T0 представляет TD0.

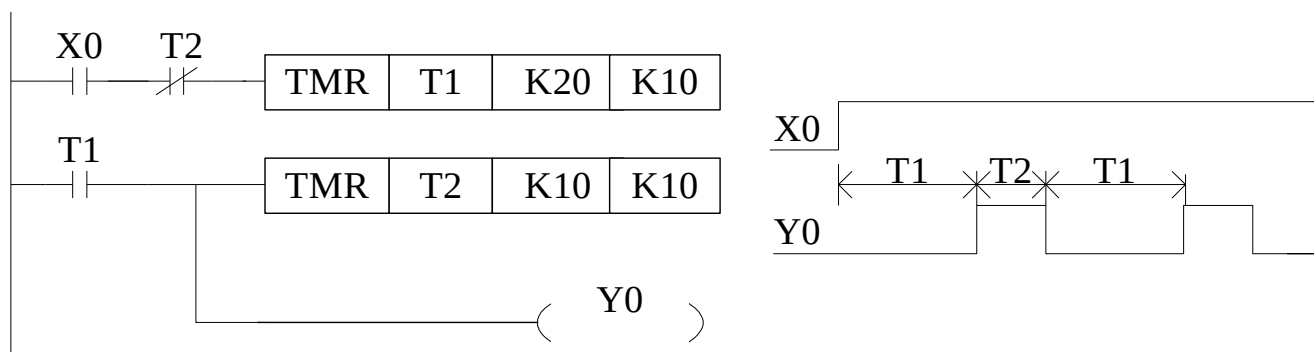
Приложение

Задержка выхода



X0 находится в состоянии ВКЛ, на выходе Y0. X0 переходит из состояния ВКЛ в состояние ВЫКЛ, задержка 2 с, а затем отключение Y0.

Мерцание



X0 - ВКЛ, Y0 - начинает мерцать (быстро переключаться из состояния ВКЛ в состояние ВЫКЛ). T1 - время ВЫКЛ; T2 - время ВКЛ.

Примечание:

Диапазон регистров, указанный выше, является актуальным диапазоном ПЛК (в том числе и в режиме связи X-NET). В режиме связи MODBUS некоторые регистры не могут быть прочитаны или записаны. Конкретный допустимый диапазон показан в главе 6-2-3.

2.8 Счетчик (C, HC, HSC)

Список адресов

Адреса счетчиков ПЛК серии RMP20 указаны в десятичной системе; подробнее см. в следующей таблице.

Серия	Имя	Диапазон		
		Нормальный	Энергонезависимый	Высокоскоростной счётчик
RMP20	C HC HSC	C0~C4999	HC0~HC1999	HSC0~HSC39

Диапазон счетчиков:

Тип счетчика	Объяснение
Счетчик 16/32 бит Возрастающий счет (счет вверх)/ убывающий счет (счет вниз)	C0~C575 HC0~HC95 (32-битный счетчик занимает два регистра, адрес счетчика должен быть четным)
Высокоскоростной счётчик	HSC0~HSC30 (HSC0, HSC2...HSC30) (каждый счетчик занимает два регистра, адрес счетчика должен быть четным)

- 1: Более подробная информация в главе 5.
- 2: Счетчики серии RMP20 могут иметь 16 или 32-битный возрастающий (вверх) и убывающий (вниз) режим счета. Один и тот же счетчик может быть использован как 16-битный или 32-битный. Режим счета с инкрементом/декрементом также выбирается переключением режима.



Особенности счетчика

Наименование	16-битный счетчик	32-битный счетчик
Направление счета	Возрастающий счет (счет вверх)/ убывающий счет (счет вниз)	Возрастающий счет (счет вверх)/ убывающий счет (счет вниз)
Установленное значение	-32,768~32,767	-2,147,483,648~+2,147,483,647
Тип уст. значения	Константа К или регистр	Константа К или пара регистров
Значение подсчета	Значение замрёт, если достигнет максимального или минимального значения	Значение замрёт, если достигнет максимального или минимального значения
Выход	Сохраняйте состояние для возрастающего счета	Сбросьте состояние для убывающего счета
Сброс	Инструкция RST для сброса счетчика и его выхода	
Регистр текущего значения счета	16-бит	32-битный

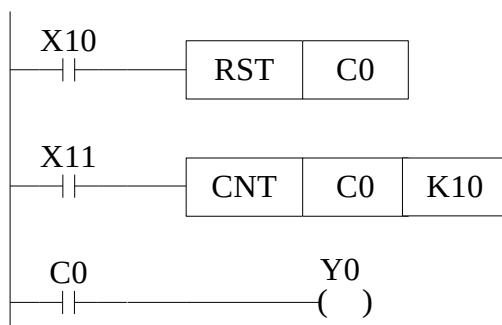
Функция

Выбор режима счетчика производится с помощью выбора соответствующих этим режимам инструкций.

16-битный общий счетчик и энергонезависимый счетчик

Диапазон установленных значений 16-битного счетчика: K1~K32,767 (десятичное значение). K0 и K1 имеют одинаковую функцию: выход счетчика будет активен при первом счете.

Если питание ПЛК отключится, общее значение счетчика будет сброшено, но значение энергонезависимого счетчика сохранится.

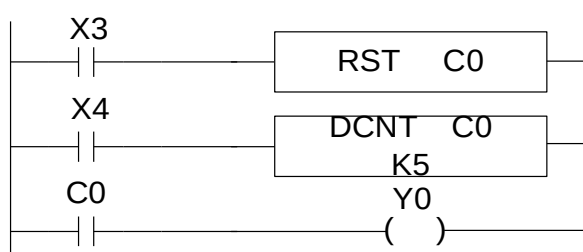


Счетчик C0 увеличивается на единицу, при активации X11. Когда значение C0 достигает 10, срабатывает выход. Если X11 снова активен, C0 продолжает счет. Если активируется X10, C0 и выход будут сброшены.

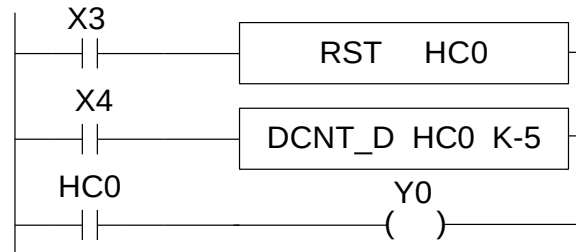
Установленное значение счетчика может быть постоянной К или принимать значение, хранимое регистром. Например, если D10 равен 123, установленное значение будет равно K123.

32-битный общий счетчик и энергонезависимый счетчик

Диапазон установленных значений 32-битного счетчика составляет $K+2,147,483,648 \sim K-2,147,483,647$ (десятичное значение). Направление счета задается с помощью инструкции.



Общий счетчик



Энергонезависимый счетчик

Если активируется X3, C0 и выход будут сброшены.

Для энергонезависимого счетчика текущее значение счетчика и состояние выхода будут сохранены после отключения питания.

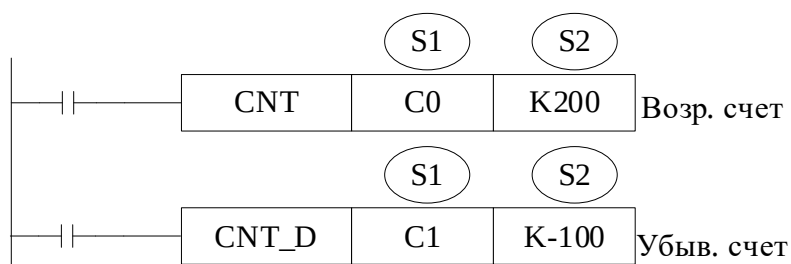
32-битный счетчик можно представить в виде 32-битного регистра.

Установленное значение счетчика

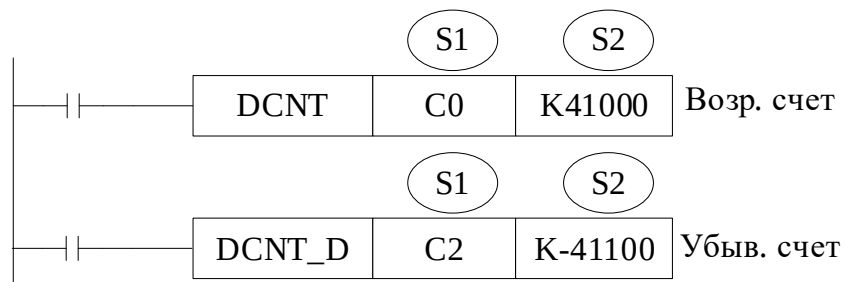
Установленное значение содержит два условия: 16-битное и 32-битное. Типы счетчиков - обычный счетчик (C) и энергонезависимый счетчик (HC).

Инструкция по подсчету

16-битный счетчик:



32-битный счетчик:



Инструкция по сбросу

16-битный счетчик:



32-битный счетчик:

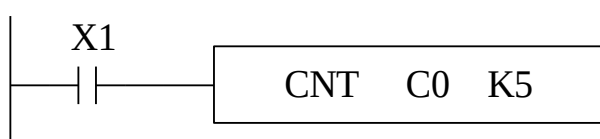


S1: Счетчик (например, C0, HC10)

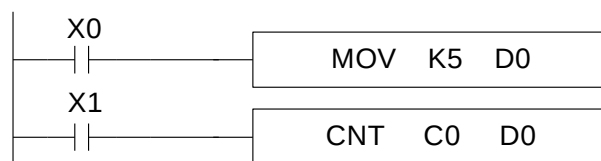
S2: Уст. значение счетчика (например, K100)

Тип счетчика устанавливается с помощью инструкции.

16-битный счетчик (обычный, возрастающий счет):

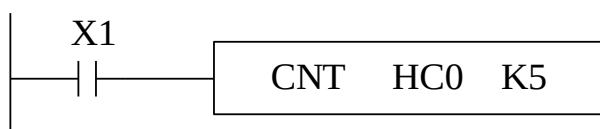


Установленное значение - постоянная K

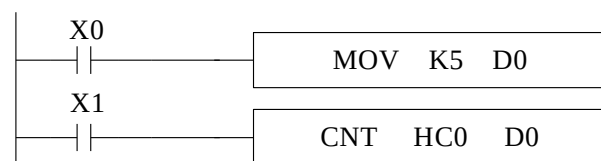


Установленное значение - регистр D

16-битный счетчик (энергонезависимый счетчик, возрастающий счет):



Установленное значение - постоянная K

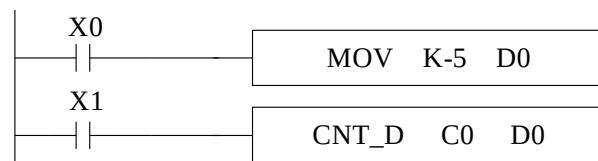


Установленное значение - регистр D

16-битный счетчик (обычный, убывающий счет):



Установленное значение - постоянная K

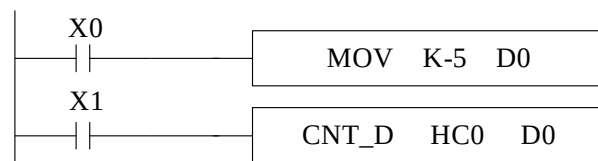


Установленное значение - регистр D

16-битный счетчик (энергонезависимый счетчик, убывающий счет):



Установленное значение - постоянная K

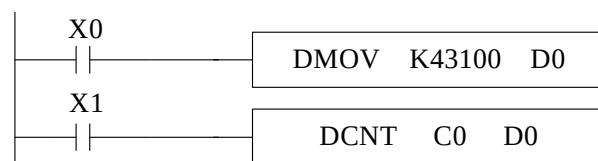


Установленное значение - регистр D

32-битный счетчик (обычный, возрастающий счет):



Установленное значение - постоянная K

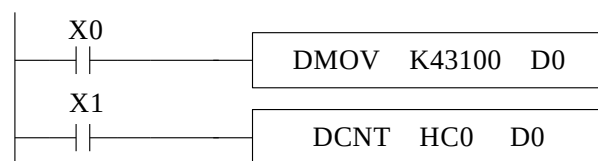


Установленное значение - регистр D

32-битный счетчик (энергонезависимый счетчик, возрастающий счет):



Установленное значение - постоянная K

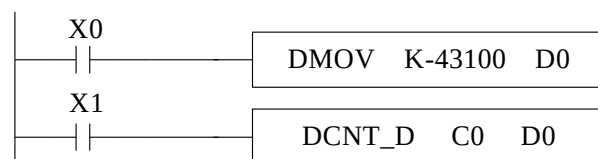


Установленное значение - регистр D

32-битный счетчик (обычный, убывающий счет):



Установленное значение - постоянная K

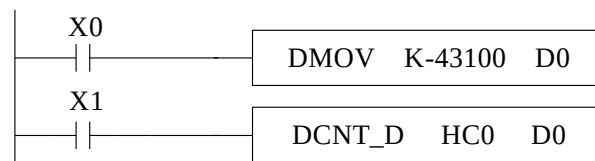


Установленное значение - регистр D

32-битный счетчик (энергонезависимый счетчик, убывающий счет):



Установленное значение - постоянная K



Установленное значение - регистр D

Примечание:

Диапазон настройки и фактическое значение настройки константы K показаны в следующей таблице.

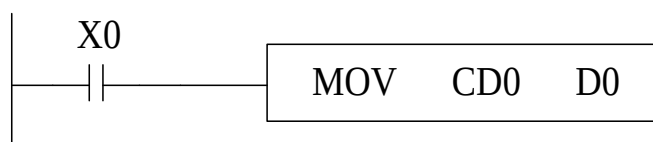
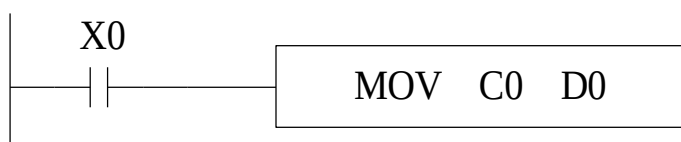
Счетчик	Диапазон настройки K	Фактический диапазон настройки
16-битный счетчик	1~32,767	1~32,767
32-битный счетчик	1~2,147,483,647	1~2,147,483,647

Значение подсчета

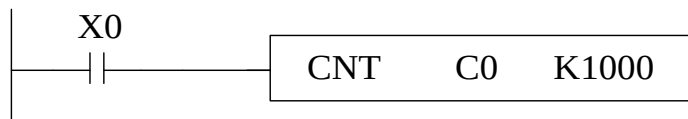
Режим подсчета - 16-битный линейный возрастающий режим счета (с инкрементом) (0~K32,767). Когда значение счета CD счетчика достигнет максимального значения K32,767, счетчик прекратит счет, и его состояние останется неизменным.

Режим подсчета - 16-битный линейный убывающий режим счета (с декрементом) (-32768-0). Когда значение счета счетчика CD уменьшится до минимального значения K-32, 768, счет прекратится, и состояние счетчика останется неизменным.

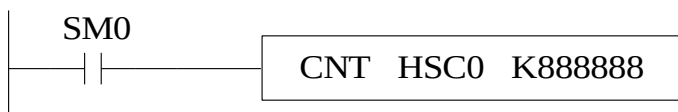
Режим подсчета - 32-битный режим линейного возрастания/убывания (-2,147,483,648 ~ +2,147,483,647). Когда значение счетчика увеличится до максимального значения K2,147,483,647, оно станет K-2,147,483,648. Когда значение счета уменьшится до минимального значения, K-2,147,483,648 станет K2,147,483,647, состояние ВКЛ/ВЫКЛ счетчика также изменится с изменением значения счета.



Две приведенные выше инструкции эквивалентны. В левой инструкции C0 обрабатывается как регистр, а в правой инструкции CD0 — это регистр данных, соответствующий таймеру C0. CD и C также эквивалентны.



Наибольшая частота, которую может считать эта инструкция, зависит от выбора параметров фильтрации и периода сканирования ПЛК. Высокоскоростной счётчик рекомендуется использовать, если частота входного сигнала превышает 25 Гц. Высокоскоростной счетчик использует операнды HSC0-HSC30.



Высокоскоростной счётчик (HSC0), когда SM0 включен, подсчитывает импульсный сигнал входной клеммы X0. Высокоскоростной счётчик не подвержен влиянию времени задержки реакции входного фильтра и времени циклического сканирования. Поэтому можно обрабатывать входные импульсы более высокой частоты. Подробнее в главе 5.

Примечание:

Диапазон регистров, указанный выше, является актуальным диапазоном ПЛК (в том числе и в режиме связи X-NET). В режиме связи MODBUS некоторые регистры не могут быть прочитаны или записаны. Конкретный допустимый диапазон показан в главе 6-2-3.

2.9 Регистр данных (D, HD, SD, HSD)

Список адресов

Регистр данных ПЛК серии PMP20 имеет десятичный формат. Ознакомьтесь со следующей таблицей.

Серия	Имя	Диапазон			
		Нормальный	Энергонезависимый	Специальный	Специальный энергонезависимый
PMP20	D HD SD HSD	D0~D69999	HD0~HD24999	SD0~SD4999	HSD0~HSD1023

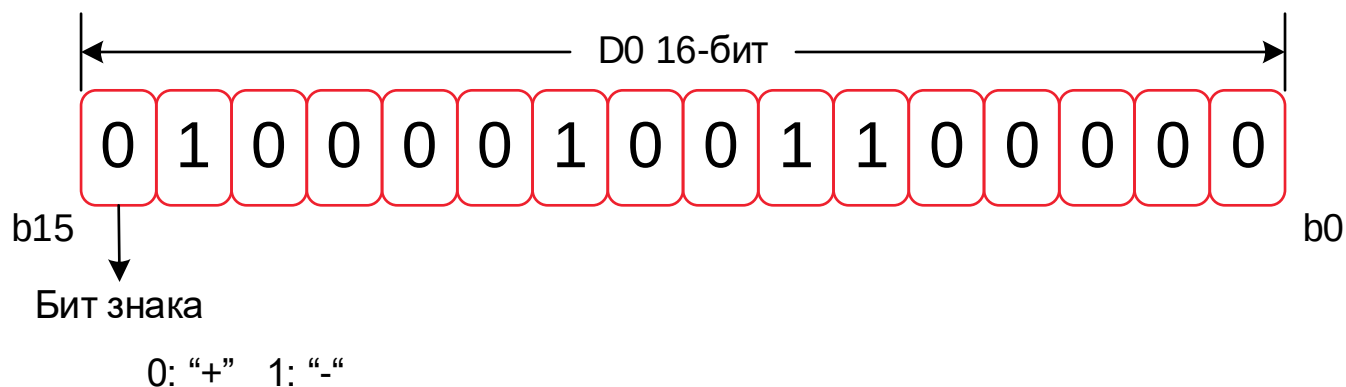
Структура

Для хранения данных используется регистр данных; Регистр данных может быть 16-битным (старший бит - бит знака) и 32-битным (32 бита содержат два регистра, старший бит - бит знака).

16 бит

Диапазон 16-битных регистров составляет -32,768 ~ +32,767.

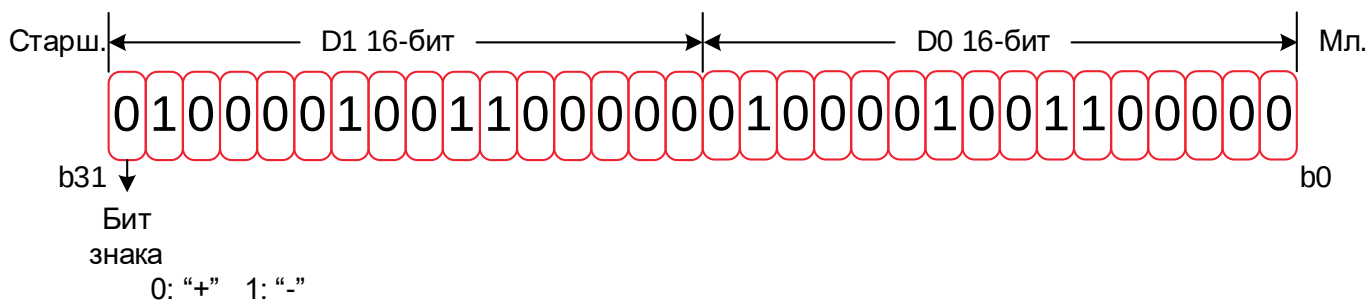
Считать и записать данные регистра можно с помощью инструкции или другого устройства, например, ЧМИ.



32 бита

32-битное значение состоит из двух непрерывных регистров. Диапазон составляет -2147483648 ~ 2147483647. Например: (D1 D0) D1 - старший 16 бит, D0 - младший 16 бит.

Для 32-битного регистра, если назначены младшие 16 бит (например, D0), следующий регистр (например, D1) автоматически станет старшим. Адрес младшего 16-битного регистра должен быть четным.



Функция

- Обычный тип

При записи нового значения в регистр, прежнее значение будет удалено.

Когда ПЛК переходит из режима RUN в STOP или из STOP в RUN, значение в регистре будет очищено.

- Энергонезависимый тип

Когда ПЛК переходит из режима RUN в STOP или выключается, значение в регистре сохраняется.

Диапазон энергонезависимых регистров не может быть изменен.

- Специальный тип

Специальный регистр используется для хранения специальных данных.

Некоторые специальные регистры инициализируются при включении питания ПЛК.

Адрес и функцию специального регистра см. в приложении.

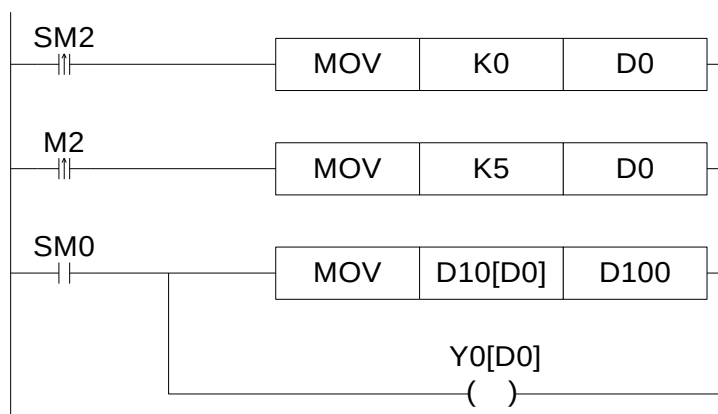
- При использовании в качестве смещения (косвенное назначение)

Регистр данных может быть использован в качестве смещения.

Формат: Dn[Dm], Xn[Dm], Yn[Dm], Mn[Dm].

Смещение слова: DXn[Dm] означает DX[n+Dm].

Значение смещения может быть установлено только в регистре D.



Когда D0 = 0, D100 = D10, Y0 находится в состоянии ВКЛ.

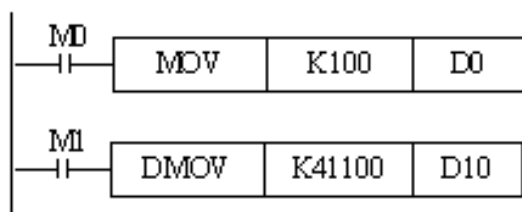
Когда M2 переходит из ВЫКЛ в состояние ВКЛ, D0 = 5, D100 = D15, Y5 - ВКЛ.

$D10[D0] = D[10+D0]$, $Y0[D0] = Y[0+D0]$.

Пример

Регистр данных D может работать со разными типами данных.

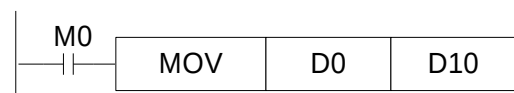
Хранение данных



Когда M0 ВКЛ, 100 записывается в D0 (16-битное значение).

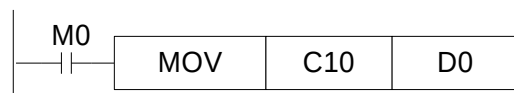
Когда M1 ВКЛ, 41100 записывается в D11, D10 (32-битное значение).

Передача данных



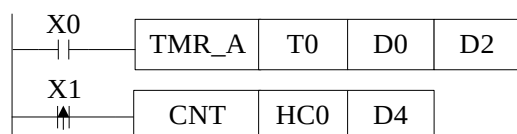
Когда M0 находится в состоянии ВКЛ, значение D10 перемещается в D0.

Считывание таймера и счетчика



Когда M0 находится в состоянии ВКЛ, значение C10 передается в D0.

Как установить значение таймера и счетчика



Когда X0 включен, активируется T0. T0 включится, когда значение D0 будет равно значению таймера, единица измерения времени - D2.

Когда X1 включен, HC0 начинает работать, HC0 примет значение ВКЛ, когда значение D4 будет равно значению счетчика.

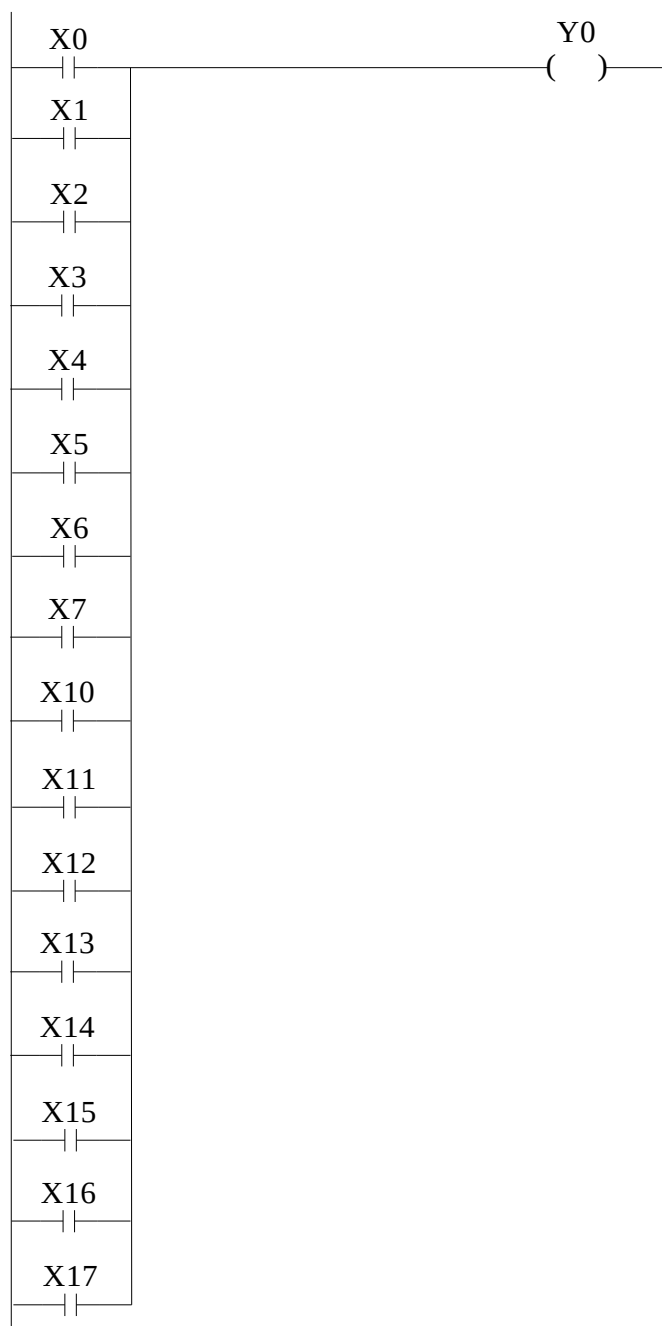
Примечание:

Диапазон регистров, указанный выше, является актуальным диапазоном ПЛК (в том числе и в режиме связи X-NET). В режиме связи MODBUS некоторые регистры не могут быть прочитаны или записаны. Конкретный допустимый диапазон показан в главе 6-2-3.

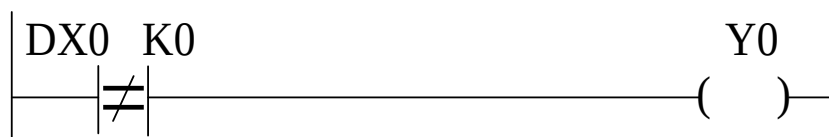
2.9.1 Слово состоящее из битов

Если один из контактов от X0 до X17 в состоянии ВКЛ, Y0 перейдет в состояние ВКЛ.

Первый метод программирования



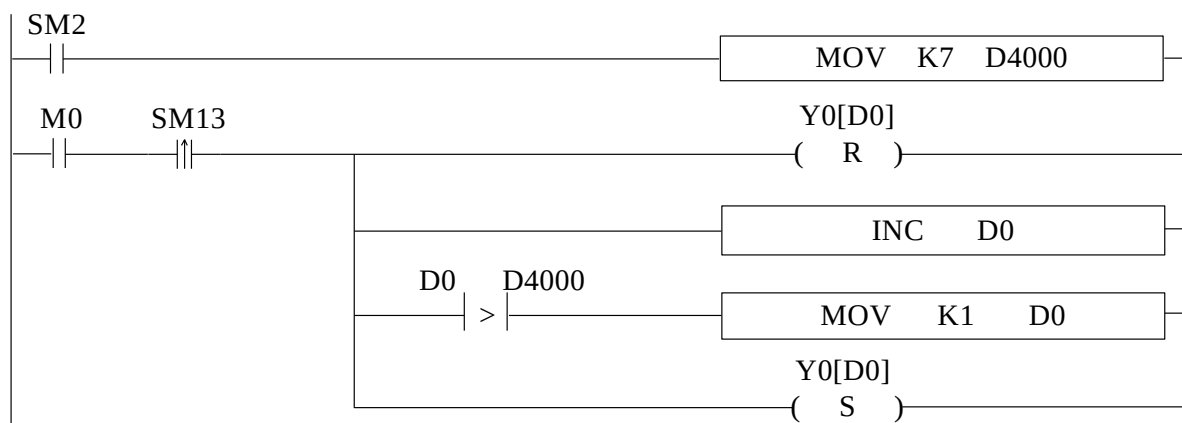
Второй метод программирования (применение слова, состоящего из битов)



2.9.2 Применение смещения

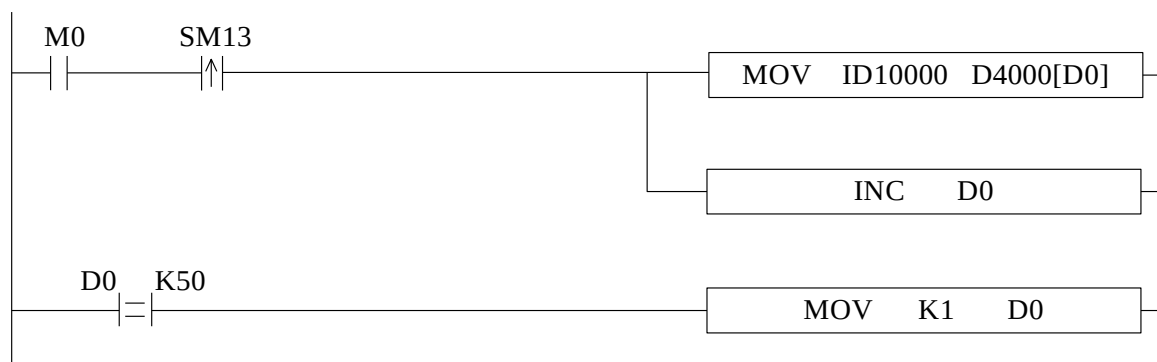
Приложение 1

Когда M0 включен, выходы с Y1 по Y7 будут включаться по очереди. D0 - это адрес смещения. При большом количестве выходных точек, M может заменить Y.



Приложение 2

Когда M0 переходит в состояние ВКЛ, происходит считывание значения ID10000 каждую секунду, далее сохраняет его в регистре, начиная с D4000 (всего 50 регистров). D0 - это адрес смещения.



2.10 Flash-регистры (FD, SFD, FS)

Регистры Flash-памяти ПЛК серии PMP20 имеют десятичную систему адресации. Порядковые номера приведены в соответствующей таблице.

Серия	Имя	Диапазон		
		Регистр пользовательских данных FLASH	Регистр системных данных FLASH	Защита чтения паролем
PMP20	FD SFD FS	FD0~FD8191	SFD0~SFD5999	FS0~FS47

Функция

- Регистр данных пользователя FLASH (FD)

Данные регистры могут хранить данные, даже если батарея разряжена, поэтому их можно использовать для хранения важных параметров.

- Регистры системных данных FLASH (SFD)

Используются для хранения системных параметров и сохранения данных при отключении питания. Область определения регистров не изменяется.

- Защита паролем для чтения регистров Flash-памяти (FS)

Часть регистра FlashROM используется для хранения данных регистров FS. Значения в регистре FS могут быть записаны, но не могут быть прочитаны, поэтому их чаще всего используют для защиты прав интеллектуальной собственности пользователей. Значение может быть произвольно установлено в регистре FS, но значение регистра не может быть прочитано (всегда возвращается в 0); и оно не может быть сравнено с регистром, только с константой, поэтому фактическое значение регистра не может быть прочитано.

Эта область памяти может сохранять данные, даже если батарея разряжена, поэтому ее можно использовать для хранения важных параметров процесса.

Примечание:

- При использовании команды MOV для передачи данных в FD, SFD и FS подходящим условием является только нарастающий фронт, даже если условием управления является нормально открытая/закрытая катушка, команда выполняется только один раз.

- (2) Регистры Flash-памяти могут быть записаны около 1 000 000 раз, и каждая запись стирает все регистры Flash-памяти, что занимает много времени. Частая запись приведет к необратимому повреждению регистров Flash-памяти, поэтому не рекомендуется часто записывать их. Не используйте в качестве условия выполнения чтения/записи катушку с изменяемым значением (например, SM11).
- (3) Когда данные передаются в один и тот же регистр Flash-памяти несколько раз, если значение в исходном регистре не изменилось по сравнению с предыдущей передачей, инструкция передачи не будет выполнена, даже если условие будет выполнено снова. Например, если значение в D0 передается в FD100, значение в D0 равно 300, когда инструкция передачи выполняется в первый раз; если условие движения устанавливается во второй раз, инструкция передачи не будет выполнена, если значение в D0 по-прежнему равно 300.
- (4) Чтобы предотвратить помехи от сигнала при передаче данных в регистры Flash-памяти, не рекомендуется использовать катушки типа SM0 и SM2 в качестве условий управления. Рекомендуется выполнять инструкции по передаче данных после включения ПЛК в течение некоторого времени.

2.11 Константа

1) Процесс обработки данных

ПЛК серии RMP20 имеет следующие 5 систем счисления.

- DEC: Десятичное число

Установленное число счетчика и таймера (константа "K"). Номер вспомогательного меркера M, HM; таймера T, HT; счетчика C, HC; состояния S, HS; регистра D, HD.

Устанавливается как значение операнда и действие применяемой инструкции (константа "K").

- HEX: Шестнадцатеричное число

Устанавливается как значения операнда и действия применяемой инструкции (константа "H")

- BIN: Двоичное число

В ПЛК все числа будут обрабатываться в двоичном формате. Но при мониторинге на устройстве все двоичные числа будут преобразованы в HEX или DEC.

- ОСТ: Восьмеричное число

Контакт и реле ввода/вывода ПЛК серии RMP20 имеют восьмеричную систему. Например, [X0-7, X10-17, ..., X70-77].

- BCD: Двоично-десятичный код

BCD использует 4 бита двоичного числа для представления десятичных чисел 0-9. BCD можно использовать в 7-сегментных светодиодах и цифровых переключателях с BCD-выходом

- Другие числа (плавающее число)

ПЛК серии RMP20 могут вычислять высокоточные числа с плавающей точкой. Они вычисляются в двоичных числах, а отображаются в десятичных.

2) Отображение

Программа ПЛК должна использовать К, Н для обработки значений. К означает десятичные числа, Н - шестнадцатеричные. Обратите внимание, что для входных/выходных регистров ПЛК используется восьмеричный адрес.

- Константа “К”

К используется для отображения десятичных чисел. К10 означает десятичное число 10. Используется для установки значения таймера и счетчика, значения операнда применяемой инструкции.

- Константа “Н”

Н используется для отображения шестнадцатеричных чисел. НА означает десятичное число 10. Используется для установки значения операнда применяемой инструкции.

- Константа “В”

В используется для отображения двоичных чисел. В10 означает десятичное число 2. Используется для установки значения операнда применяемой инструкции.

2.12 Принцип программирования

Подпись Р и I

“Р” — это обозначение программы для условия и перехода через подпрограмму.

“I” - программный знак прерывания (внешнее прерывание, прерывание по таймеру, прерывание по высокоскоростному счётчику, прерывание по точному времени...).

Адреса Р и I указываются в десятичной системе счисления. Обратитесь к следующей таблице:

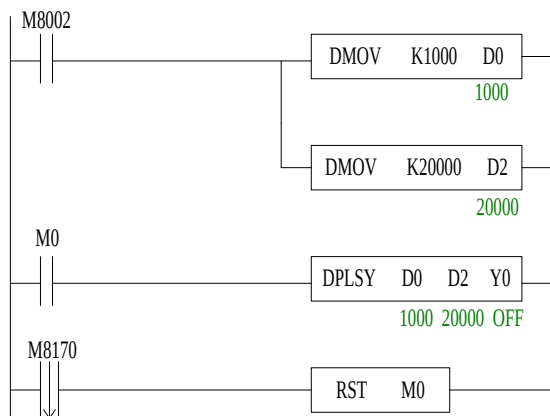
Серия	Обозначение	Адрес
PMP20	P	P0~P9999

Модель	Имя	Диапазон			
		Внешнее прерывание			Прерывание по таймеру
		Вход	Прерывание по нараст. фронту	Прерывание по убыв. фронту	
Серия PMP20 30-60 точек	I	X2	I0000	I0001	Существует 20 прерываний по таймеру. От I40** до I59**. "***" означает время прерывания таймера, единица измерения - мс.
		X3	I0100	I0101	
		X4	I0200	I0201	
		X5	I0300	I0301	
		X6	I0400	I0401	
		X7	I0500	I0501	
		X10	I0600	I0601	
		X11	I0700	I0701	
		X12	I0800	I0801	
		X13	I0900	I0901	

Подпись Р

Р обычно используется в потоке выполнения программы; он применяется вместе с CJ (переход по условию), CALL (вызов подпрограммы) и т.д.

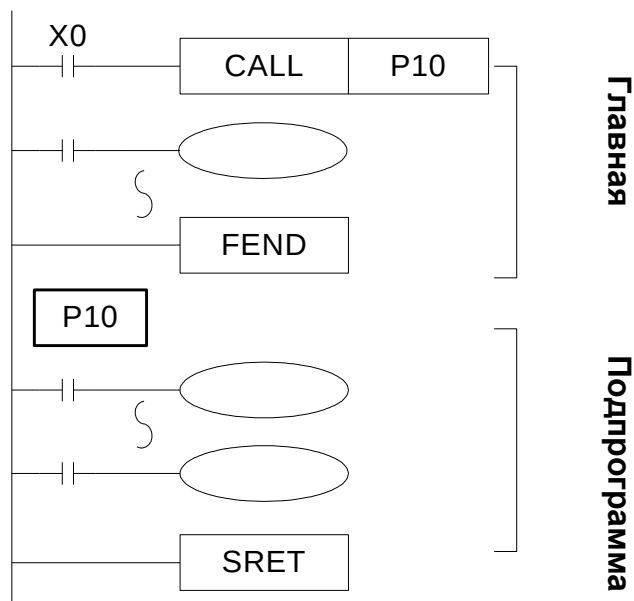
Скачок по условию CJ



Если контакт X0 ВКЛ, перейдите к программе после P1.

Если контакт X0 не ВКЛ, действие перехода не выполняется. Происходит запуск исходной программы.

Вызовите подпрограмму (CALL)



Если X0 - ВКЛ, перейдите к подпрограмме.

Если катушка не ВКЛ, запустите основную программу.

После выполнения подпрограммы вернитесь в основную программу.

Подпрограмма будет начинаться с Pn и заканчиваться SRET. CALL Pn используется для вызова подпрограммы, n - целое число в диапазоне от 0 до 9999.

Подпись I

Подпись I обычно используется в прерывании, включая внешнее прерывание, прерывание по времени и т.д. I используется вместе с IRET (возврат прерывания), EI (включить прерывание), DI (отключить прерывание).

- Внешнее прерывание

Сигнал поступает на специальные входные клеммы, на которые не влияет цикл сканирования. При подходящем значении входного сигнала, выполняется подпрограмма прерывания.

При внешнем прерывании ПЛК может обработать сигнал быстрее, чем цикл сканирования. Поэтому, I можно использовать в качестве основного приоритета при управлении последовательностью или для управления короткими импульсами.

- Прерывание времени

Прерывание по времени заключается в выполнении подпрограммы прерывания в каждое указанное время цикла прерывания. Рекомендуется использовать это прерывание для управления, отличного от управления в цикле работы ПЛК.

- Последовательность действий входных/выходных контактов/реле и задержка реакции

Вход

Перед тем как ПЛК выполнит программу, происходит считывание всех состояний ВКЛ/ВЫКЛ входных клемм в область отображения. В процессе выполнения программы, даже если входной сигнал изменился, содержимое в области отображения входа не изменится. Однако в следующем цикле сканирования изменения будут считаны.

Выход

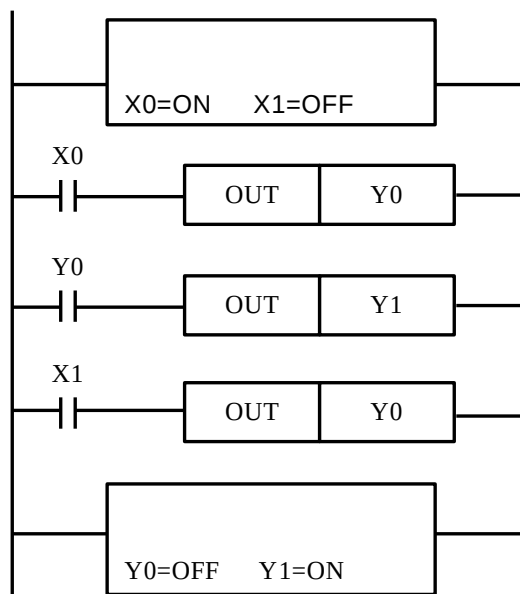
Как только все инструкции завершатся, состояние ВКЛ/ВЫКЛ области отображения выхода Y передается в область памяти выхода. Выходные клеммы будут действовать в соответствии с временем задержки срабатывания устройства.

При использовании пакетной (batch) модели ввода/вывода, время реакции и цикл операций фильтра ввода и устройства вывода также будут с задержкой.

- Отсутствие реакции на узкие импульсные сигналы на входе

Время включения/выключения входа ПЛК должно быть больше, чем время цикла. Если задержка срабатывания входного фильтра составляет 10 мс, время цикла - 10 мс, то время включения/выключения должно быть 20 мс. Таким образом, входной импульс частотой до $1\,000/(20+20) = 25$ Гц не может быть обработан (при использовании специальных функций и прикладных инструкций (таких как высокоскоростной счет, прерывание ввода, настройка входного фильтра) ситуацию можно изменить).

- Двойной выход (катушка Y)



Рассмотрите случай использования одной и той же катушки Y0 во многих положениях:

Например, X0 = ВКЛ, X1 = ВЫКЛ.

Первый Y0: X0 включен, его область отображения ВКЛ, выход Y1 также ВКЛ.

Второй Y0: поскольку вход X1 ВЫКЛ, область отображения ВЫКЛ.

Таким образом, фактический результат будет следующим: Y0 = ВЫКЛ, Y1 = ВКЛ.

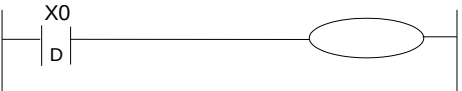
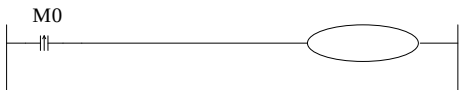
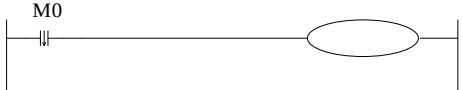
При двойном выходе (использование двойной катушки) приоритет отдается последующему.

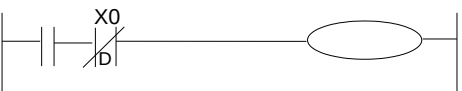
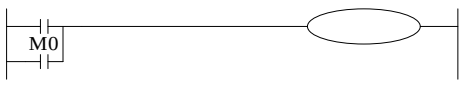
3. Основные программные инструкции

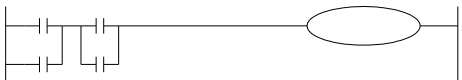
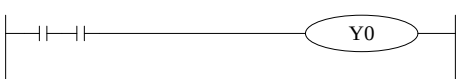
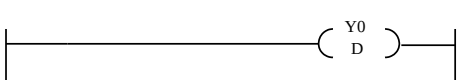
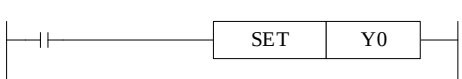
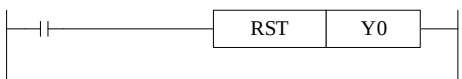
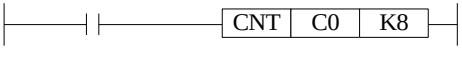
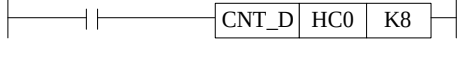
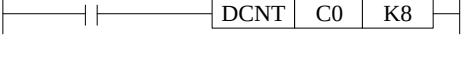
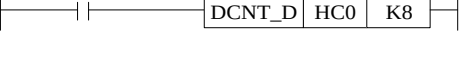
В этой главе представлены основные инструкции и их функции.

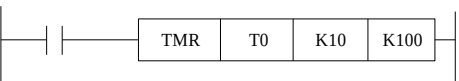
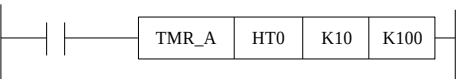
3.1 Список основных инструкций

Серия PMP20 поддерживает все основные инструкции:

Обозначение	Функция	Формат	Глава
LD	Начало логического выражения (связей) с опросом на состояние сигнала "1"		3-2
LDD	Считывание непосредственно из состояния контакта (НО)		3-6
LDI	Начало логического выражения (связей) с опросом на состояние сигнала "0"		3-2
LDDI	Считывание непосредственно из состояния контакта (НЗ)		3-6
LDP	Начало логического выражения с опросом по переднему фронту (импульс)		3-5
LDF	Начало логического выражения с опросом по заднему фронту (импульс)		3-5
AND	Логическое выражение "И" с опросом на состояние сигнала "1"		3-3
ANDD	Логическое выражение "И" с опросом из состояния контакта (НО)		3-6

Обозначение	Функция	Формат	Глава
ANI	Логическое выражение “И” с опросом на состояние сигнала “0”		3-3
ANDDI	Логическое выражение “И” с опросом из состояния контакта (НЗ)		3-6
ANDP	Логическое выражение “И” с опросом по переднему фронту (импульс)		3-5
ANDF	Логическое выражение “И” с опросом по заднему фронту (импульс)		3-5
OR	Логическое выражение “ИЛИ” с опросом на состояние сигнала “1”		3-4
ORD	Логическое выражение “ИЛИ” с опросом из состояния контакта (НО)		3-6
ORI	Логическое выражение “ИЛИ” с опросом на состояние сигнала “0”		3-4
ORDI	Логическое выражение “ИЛИ” с опросом из состояния контакта (НЗ)		3-6
ORP	Логическое выражение “ИЛИ” с опросом по переднему фронту (импульс)		3-5
ORF	Логическое выражение “ИЛИ” с опросом по заднему фронту (импульс)		3-5

Обозначение	Функция	Формат	Глава
ANB	Блок “И”, последовательное включение параллельных цепей		3-8
ORB	Блок “ИЛИ”, параллельное включение последовательных цепей		3-7
OUT	Выдача. Присвоение результата логического выражения		3-2
OUTD	Мгновенная выдача. Присвоение результата логического выражения без ожидания следующего цикла сканирования		3-6
SET	Включение		3-12
RST	Выключение		3-12
CNT	16-битный энергонезависимый возрастающий подсчет (счет вверх)		3-13
CNT_D	16-битный энергонезависимый убывающий счет (счет вниз)		3-13
DCNT	32-битный энергонезависимый возрастающий подсчет (счет вверх)		3-13
DCNT_D	32-битный энергонезависимый убывающий счет (счет вниз)		3-13

Обозначение	Функция	Формат	Глава
PLS	Создание импульса по переднему фронту		3-11
PLF	Создание импульса по заднему фронту		3-11
MCS	Ответвление на новую шину		3-9
MCR	Возврат на шину (используется в паре с MCS)		3-9
ALT	Инверсия состояния катушки		3-10
TMR	Таймер		3-14
TMR_A	Таймер (энергонезависимый)		3-14
END	Завершение программы. Процесс ввода/вывода и возврат к шагу 0		3-15
GROUP	Начало блока инструкций		3-15
GROUPE	Конец блока инструкций		3-16

3.2 [LD], [LDI], [OUT]

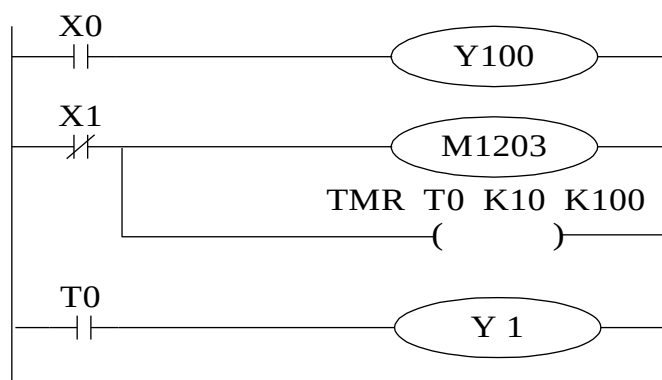
1) Обозначения и функции

Обозначение	Функция	Формат и операнды
LD	Начало логического выражения (связей) с опросом на состояние сигнала "1"	 Операнды: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
LDI	Начало логического выражения (связей) с опросом на состояние сигнала "0"	 Устройства: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
OUT	Выдача. Присвоение результата логического выражения	 Операнды: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

2) Объяснение

- Инструкции LD и LDI могут работать с ANB и использоваться в начале ветвления.
- Инструкция OUT может приводить в действие выходные реле, вспомогательные меркеры, статусы, таймеры и счетчики. Но эта инструкция не может быть использована для входных контактов.

3) Программа



```

LD      X0
OUT     Y100
LDI     X1
OUT     M1203
TMR     T0      K10 K100
LD      T0
OUT     Y1

```

3.3 [AND], [ANI]

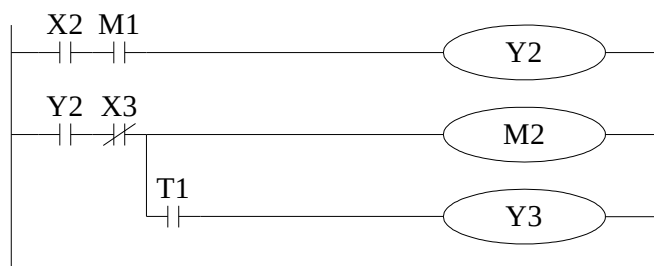
1) Обозначения и функции

Обозначение	Функция	Формат и операнды
AND	Логическое выражение "И" с опросом на состояние сигнала "1"	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ANI	Логическое выражение "И" с опросом на состояние сигнала "0"	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

2) Объяснение

- Используйте AND и ANI при последовательном соединении контактов.
- Использование команды ВКЛ через другую катушку называется "последующим" выводом (Для примера смотрите программу ниже: OUT M2 и OUT Y3). Нет ограничений на количество последовательно подключенных контактов и время последующего выхода.

3) Программа



LD	X2
AAND	M1
OUT	Y2
LD	Y2
ANI	X3
OUT	M2
AND	T1
OUT	Y3

3.4 [OR], [ORI]

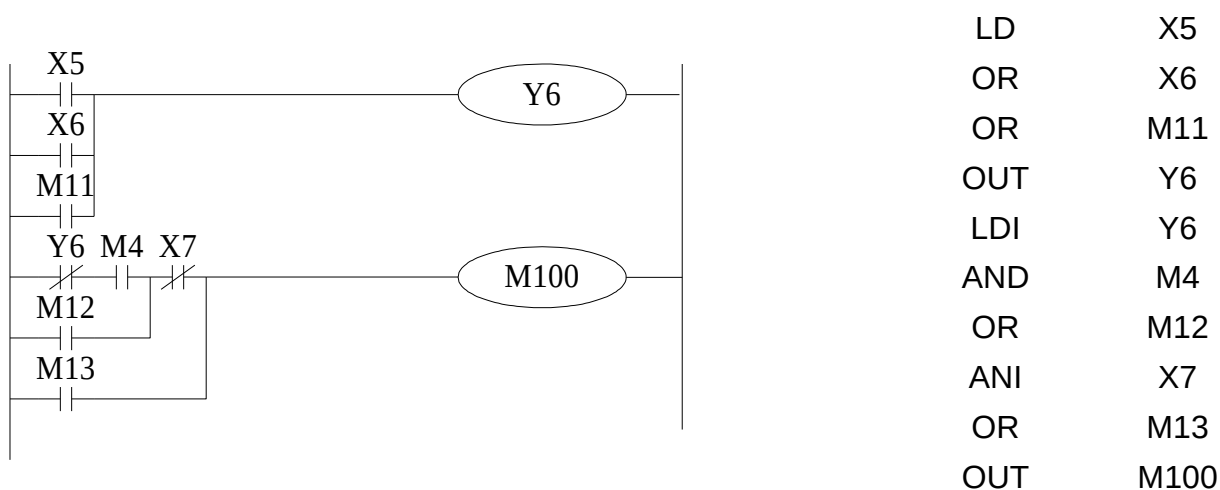
1) Обозначения и функции

Обозначение	Функция	Формат и операнды
OR	Логическое выражение "ИЛИ" с опросом на состояние сигнала "1"	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ORI	Логическое выражение "ИЛИ" с опросом на состояние сигнала "0"	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

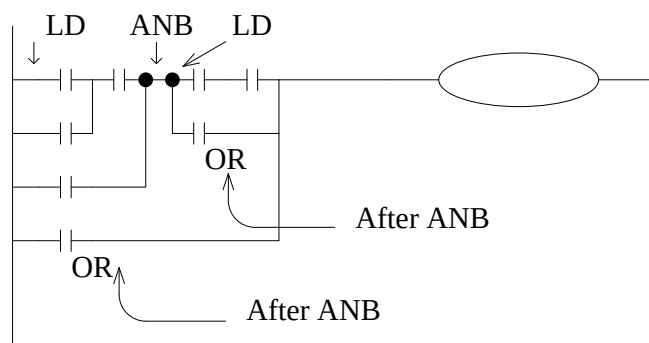
2) Объяснение

- Для параллельного соединения контактов используйте инструкции OR и ORI. Чтобы подключить блок, содержащий более одного последовательно соединенного контакта, к другому блоку параллельно, используйте инструкцию ORB;
- OR и ORI начинаются с шага инструкции, параллельно соединяясь с инструкциями LD и LDI, представленными ранее.

3) Программа



4) Связь с ANB



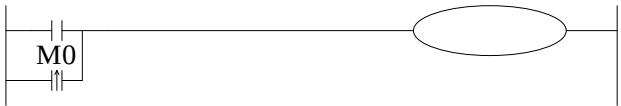
Параллельное соединение с инструкциями OR, ORI должно соединяться с инструкциями LD, LDI. Но за инструкцией ANB все равно можно добавить инструкцию LD или LDI.

3.5 [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF]

1) Обозначения и функции

Обозначение	Функция	Формат и операнды
LDP	Начало логического выражения с опросом по переднему фронту (импульс)	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
LDF	Начало логического выражения с опросом по заднему фронту (импульс)	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ANDP	Логическое выражение "И" с опросом по переднему фронту (импульс)	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ANDF	Логическое выражение "И" с опросом по заднему фронту (импульс)	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

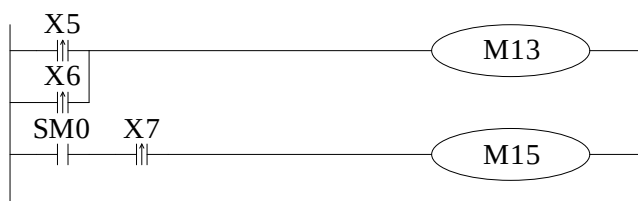


Обозначение	Функция	Формат и операнды
ORP	Логическое выражение "ИЛИ" с опросом по переднему фронту (импульс)	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
ORF	Логическое выражение "ИЛИ" с опросом по заднему фронту (импульс)	 X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

2) Объяснение

- LDP, ANDP, ORP будут ВКЛ на один период сканирования, когда придет нарастающий импульс сигнала (ВЫКЛ→ ВКЛ).
- LDF, ANDF, ORF будут включены на один период сканирования, когда придет падающий импульс сигнала (ВКЛ→ ВЫКЛ).

3) Программа



LDP X5

ORP X6

OUT M13

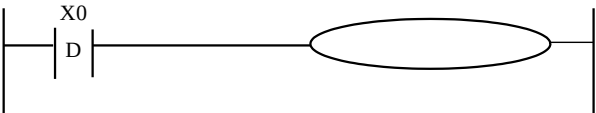
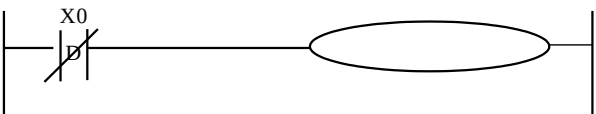
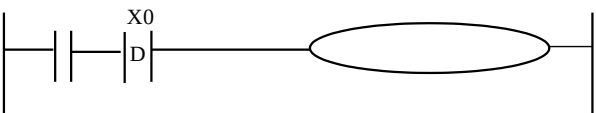
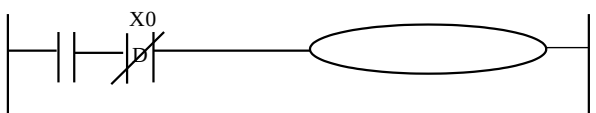
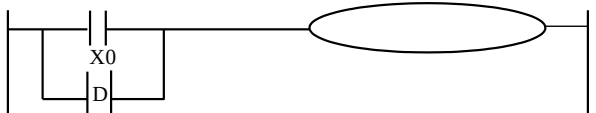
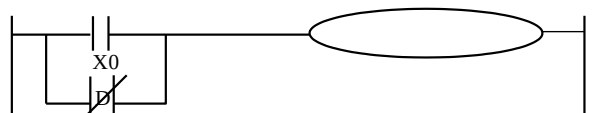
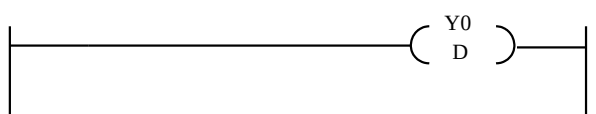
LD M8000

ANDP X7

OUT M15

3.6 [LDD], [LDDI], [ANDD], [ANDDI], [ORD], [ORDI], [OUTD]

1) Обозначения и функции

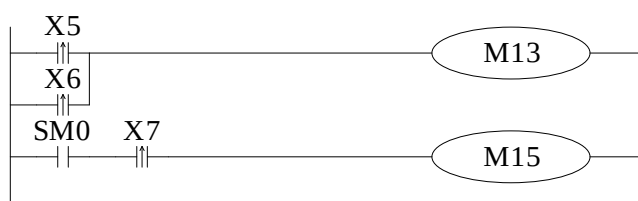
Обозначение	Функция	Формат и операнды
LDD	Считывание непосредственно из состояния контакта (НО)	 Операнд: X
LDDI	Считывание непосредственно из состояния контакта (НЗ)	 Операнд: X
ANDD	Логическое выражение "И" с опросом из состояния контакта (НО)	 Операнд: X
ANDDI	Логическое выражение "И" с опросом из состояния контакта (НЗ)	 Операнд: X
ORD	Логическое выражение "ИЛИ" с опросом из состояния контакта (НО)	 Операнд: X
ORDI	Логическое выражение "ИЛИ" с опросом из состояния контакта (НЗ)	 Операнд: X
OUTD	Мгновенная выдача. Присвоение результата логического выражения без ожидания следующего цикла сканирования	 Операнд: Y

2) Объяснение

Функции команд LDD, ANDD, ORD аналогичны командам LD, AND, OR; команды LDDI, ANDDI, ORDI аналогичны командам LDI, ANDI, ORI; но если операнд равен X, команды LDD, ANDD, ORD считывают сигнал с клемм напрямую.

OUTD и OUT - это инструкции вывода. OUTD выводит данные немедленно, когда условие выполнено, не нужно ждать следующего цикла сканирования.

3) Программа



LDD X0

LDDI X2

ORD X2

ANB

OUTD Y0

3.7 [ORB]

1) Обозначения и функции

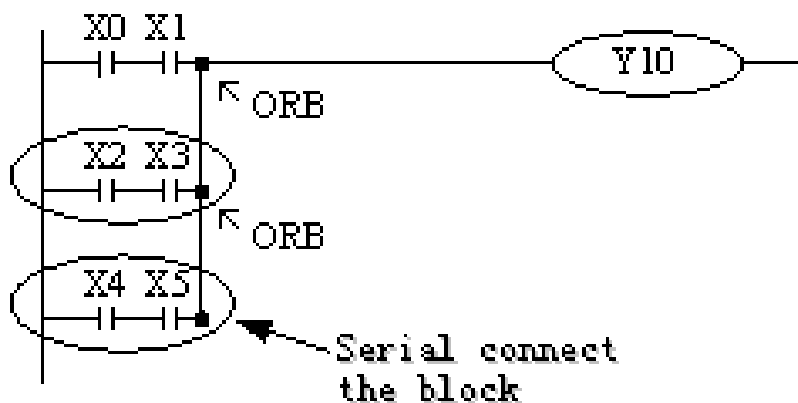
Обозначение	Функция	Формат
ORB	Блок "ИЛИ", параллельное включение последовательных цепей	 Операнд: нет

2) Объяснение

Два или более контакта называются "последовательным блоком". При параллельном соединении последовательного блока используйте LD, LDI в точке начала ветвления, используйте ORB в точке конца ветвления. Как и инструкция ANB, инструкция ORB - это независимая инструкция, которая не связана ни с каким программным компонентом.

При использовании ORB для каждой цепи нет ограничений на количество параллельных цепей.

3) Программа



Рекомендуемый метод программирования

LD	X0
AND	X1
LD	X2
AND	X3
ORB	
LD	X4
AND	X5
ORB	
OUT	Y10

Нежелательный метод программирования

LD	X0
AND	X1
LD	X2
AND	X3
LD	X4
AND	X5
ORB	
ORB	
OUT	Y10

3.8 [ANB]

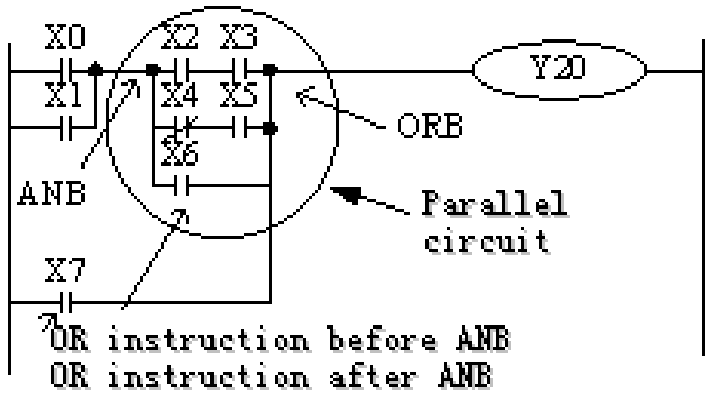
1) Обозначения и функции

Обозначение	Функция	Формат
ANB	Блок “И”, последовательное включение параллельных цепей	 Операнд: нет

Используйте ANB для последовательного соединения двух параллельных цепей. Используйте LD, LDI в начальной точке ветви; используйте ANB в конечной точке ветви.

Время использования инструкции ANB не ограничено.

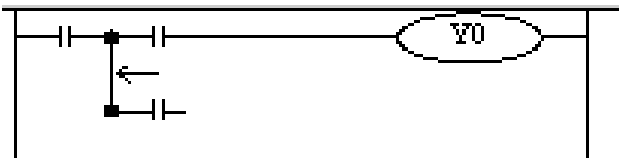
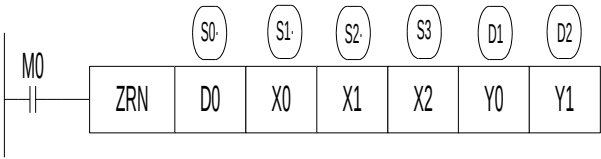
2) Программа



LD X0
OR X1
LD X2
AND X3
LDI X4
AND X5
ORB
OR X6
ANB
OR X7
OUT Y20

3.9 [MCS], [MCR]

1) Обозначения и функции

Обозначение	Функция	Формат
MCS (Ведущее устройство)	Ответвление на новую шину	 Операнд: Нет
MCR (Ведущее устройство)	Возврат на шину (используется в паре с MCS)	 Операнд: Нет

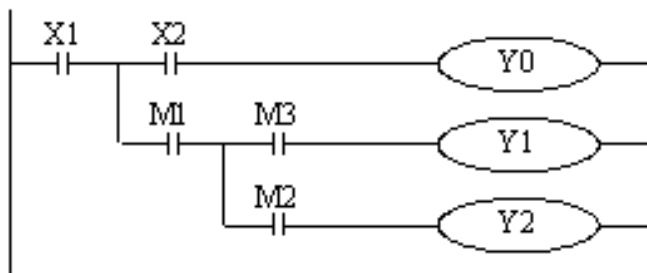
2) Объяснение

- После выполнения инструкции MCS линия шины (LD, LDI) перемещается в точку после инструкции MCS. Инструкция MCR возвращает исходную линию шины.
- Инструкции MCS, MCR следует использовать в паре.
- Линия шины может быть вложенной. Используйте инструкции MCS, MCR между инструкциями MCS, MCR. Уровень вложенности увеличивается по мере использования инструкции MCS. Максимальный уровень вложенности - десять. При выполнении инструкции MCR вернитесь на последний уровень шины.
- При использовании программы управления потоком, управление шиной может быть использовано только в том же потоке. Когда поток заканчивается, он должен вернуться на главную линию шины.

Примечание:

Инструкции MCS и MCR не могут быть записаны непосредственно в лестничной диаграмме (Представление LD) ПО ПЛК серии RMP20. Они представлены в виде горизонтальных и вертикальных линий.

3) Программа

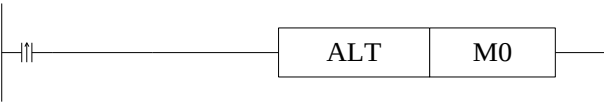


```

LD      X1
MCS
LD      X2
OUT     Y0
LD      M1
MCS
LD      M3
OUT     Y1
LD      M2
OUT     Y2
MCR
MCR
  
```

3.10 [ALT]

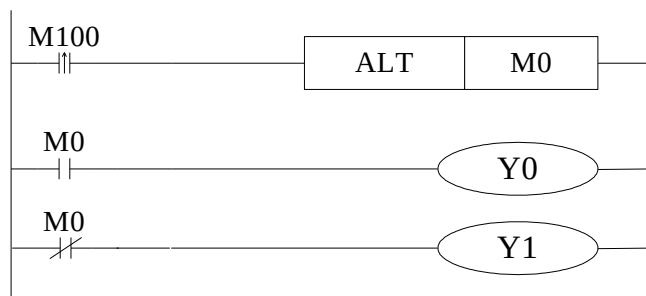
1) Обозначения и функции

Обозначение	Функция	Формат
ALT	Инверсия состояния катушки	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

2) Объяснение

После использования ALT состояние катушки меняется на противоположное (ВКЛ меняется на ВЫКЛ, ВЫКЛ меняется на ВКЛ).

3) Программа



LDP M100

ALT M0

LD M0

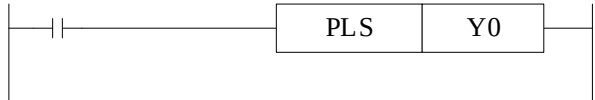
OUT Y0

LDI M0

OUT Y1

3.11 [PLS], [PLF]

1) Обозначения и функции

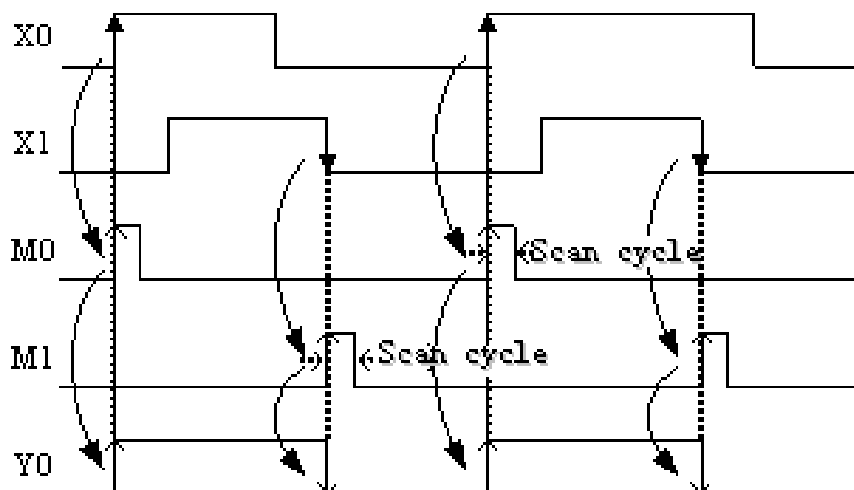
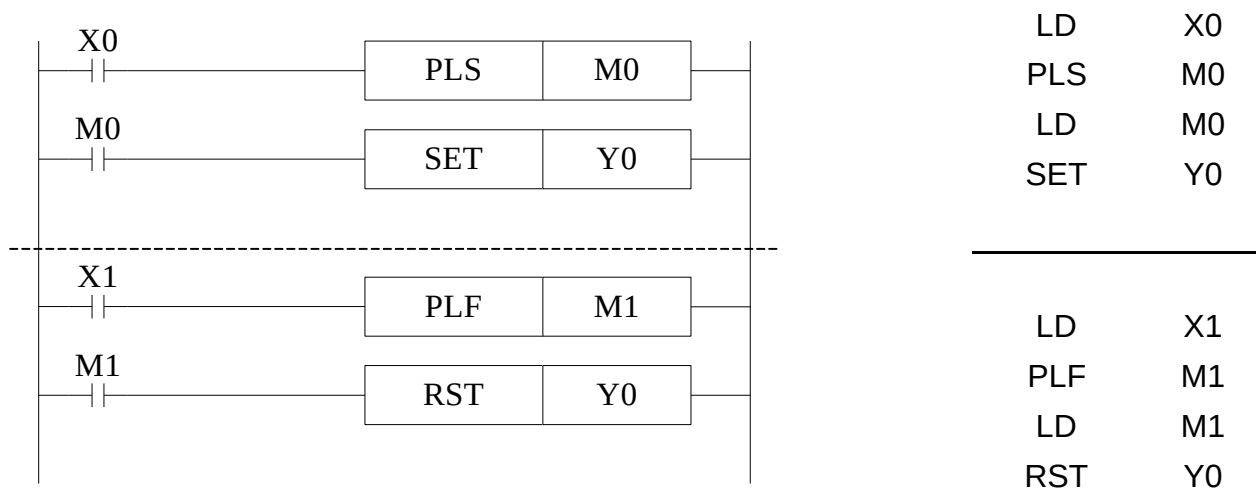
Обозначение	Функция	Формат
PLS	Создание импульса по переднему фронту	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
PLF	Создание импульса по заднему фронту	 Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

2) Объяснение

При использовании инструкции PLS: программные компоненты Y и M будут активны в течение одного периода сканирования после ВКЛ.

Для использования инструкции PLF: программные компоненты Y и M будут активны в течение одного периода сканирования после ВЫКЛ.

3) Программа



3.12 [SET], [RST]

1) Обозначения и функции

Обозначение	Функция	Формат
SET	Включение	Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m
RST	Сброс	Операнд: X, Y, M, HM, SM, S, HS, T, HT, C, HC, Dn.m

2) Объяснение

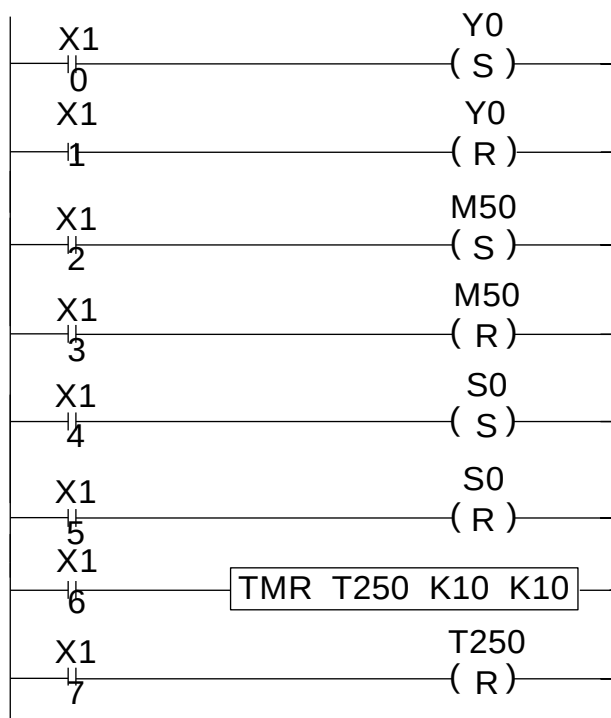
В следующей программе Y0 будет оставаться ВКЛ, даже если X10 выключится после включения. Y0 не включится, даже если X11 выключится после включения. То же самое относится к S и M.

SET и RST можно использовать много раз для одного и того же регистра. Допускается любой порядок, но эффективным является последний.

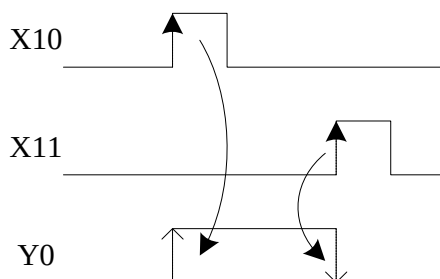
RST может использоваться для сброса счетчика, таймера и контакта.

Когда используется SET или RST, тот же самый регистр не может быть использован с OUT.

3) Программа

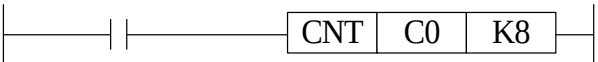
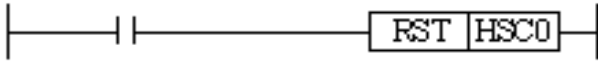


```
LD    X10
      Y0
LD    X11
RST   Y0
LD    X12
SET   M50
LD    X13
RST   M50
LD    X14
SET   S0
LD    X15
RST   S0
LD    X16
TMR   T250    K10    K10
LD    X17
RST   T250
```

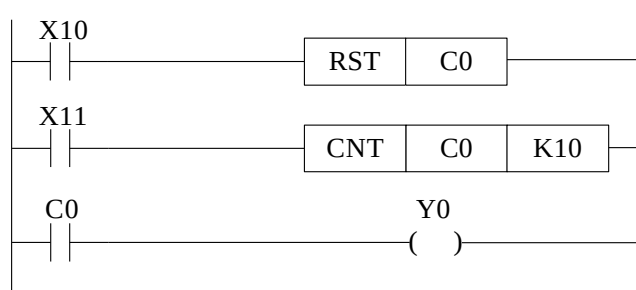


3.13 [CNT], [CNT_D], [DCNT], [DCNT_D], [RST] для Счетчиков

1) Обозначения и функции

Обозначение	Функция	Формат
CNT Выход	16 бит энергонезависимый счет с нарастанием (инкрементом/счет вверх)	 Операнд: K, D
CNT_D Выход	16 бит энергонезависимый счет с убыванием (декрементом/счет вниз)	 Операнд: K, D
DCNT Выход	32 бит энергонезависимый счет с нарастанием (инкрементом/счет вверх)	 Операнд: K, D
DCNT_D Выход	32 бит энергонезависимый счет с убыванием (декрементом/счет вниз)	 Операнд: K, D
RST Сброс	Сброс, очистка текущего значения счета	 Операнд: C, HC, HSC

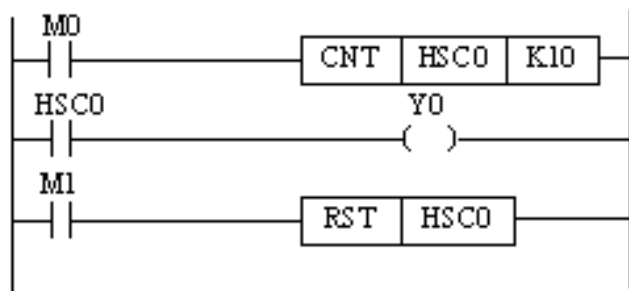
2) Программирование внутреннего счетчика



Увеличение C0 подсчитывает время перехода X11 из режима ВЫКЛ в режим ВКЛ. Когда C0 достигнет K10, C0 перейдет из состояния ВЫКЛ в состояние ВКЛ. Когда X11 переходит из состояния ВЫКЛ в состояние ВКЛ, значение тока C0 будет продолжать увеличиваться, а катушка C0 по-прежнему будет ВКЛ. Когда X10 станет ВКЛ, катушка C0 сбросится.

Энергонезависимый счетчик сохранит текущее значение и состояние счетчика при выключении питания.

3) Программирование высокоскоростного счётчика



Увеличьте время перехода от ВЫКЛ к ВКЛ (условие M0).

Когда значение счета достигнет установленного значения (значение K или D), счетная катушка -> ВКЛ.

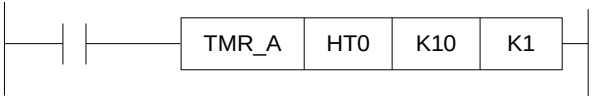
Когда M1 включится, счетная катушка HSC0 сбросится, а текущее значение станет равным 0.

3.14 [TMR], [TMR_A] для таймеров

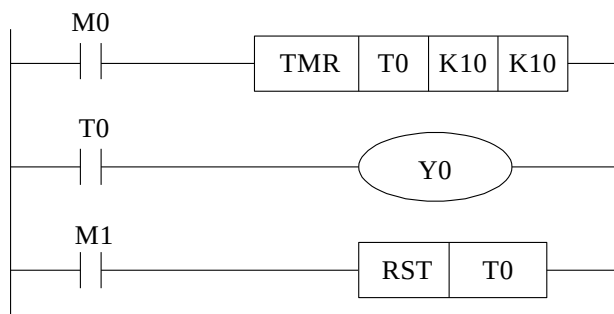
1) Обозначения и функции

Обозначение	Функция	Формат
TMR	Обычный таймер 100 мс (Датчик тактовых импульсов: 100 мс)	Операнд: K, D
TMR	Обычный таймер 10 мс (Датчик тактовых импульсов: 10 мс)	Операнд: K, D
TMR	Обычный таймер 1 мс (Датчик тактовых импульсов: 1 мс)	Операнд: K, D
TMR_A выход	Энергонезависимый таймер 100 мс (Энергонезависимый датчик тактовых импульсов: 100 мс)	Операнд: K, D
TMR_A выход	Энергонезависимый таймер 10 мс (Энергонезависимый датчик тактовых импульсов: 10 мс)	Операнд: K, D



TMR_A выход	Энергонезависимый таймер 1 мс (Энергонезависимый датчик тактовых импульсов: 1 мс)	 Операнд: C, HC, HSC
----------------	--	---

2) Программирование внутреннего таймера

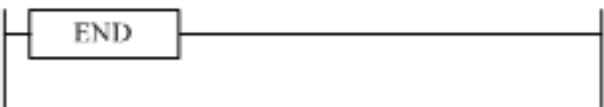


Когда M0 переходит в состояние ВКЛ, T0 начинает отсчитывать время. Когда T0 достигает K10, катушка T0 -> ВКЛ. Затем T0 продолжает отсчет времени. Когда M1 включится, T0 сбрасывается.

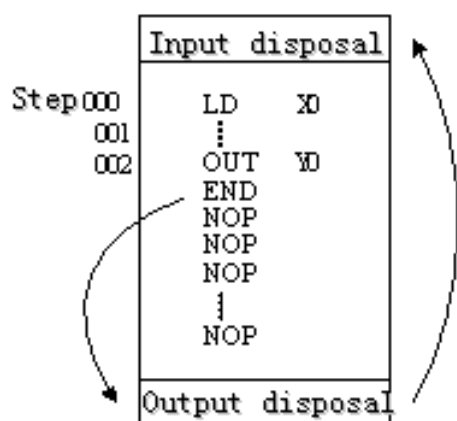
Энергонезависимый таймер сохранит текущее значение и состояние катушки счетчика при отключении питания.

3.15 [END]

1) Обозначения и функции

Обозначение	Функция	Формат: Нет
END (КОНЕЦ)	Процесс ввода/вывода и возврат к шагу 0	 Операнд: Нет

2) Объяснение



ПЛК многократно выполняет ввод, выполнение программы и вывод. Если записать инструкцию END в конце программы, то инструкции, расположенные за инструкцией END, не будут выполнены. Если в программе нет инструкции END, ПЛК выполняет завершающий шаг, а затем повторяет выполнение программы с шага 0.

При отладке вставляйте END в каждый сегмент программы, чтобы проверить действие каждой программы.

Затем, после подтверждения исправления действия предыдущего блока, удалите инструкцию END. Кроме того, первое выполнение RUN начинается с инструкции END.

При выполнении инструкции END таймер мониторинга цикла сканирования обновится (SD010).

3.16 [GROUP], [GROUPE]

1) Обозначения и функции

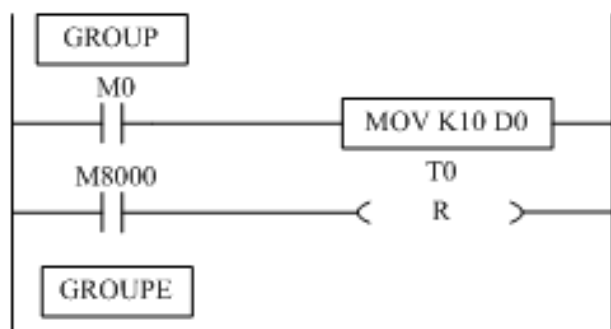
Обозначение	Функция	Формат
GROUP	Начало блока инструкций	GROUP Операнд: Нет
GROUPE	Конец блока инструкций	GROUPE Операнд: Нет

2) Объяснение

GROUP и GROUPE следует использовать в паре.

GROUP и GROUPE не имеют практического значения; они используются для оптимизации структуры программы. Поэтому добавление или удаление этих инструкций не влияет на работу программы;

Метод использования GROUP и GROUPE аналогичен использованию инструкций потока; введите инструкцию GROUP в начале части группы; введите инструкцию GROUPE в конце части группы.



Как правило, инструкции GROUP и GROUPE могут быть запрограммированы в соответствии с функцией группы.

3) Заметки по программированию

Структура и этапы работы

Общий принцип программирования:

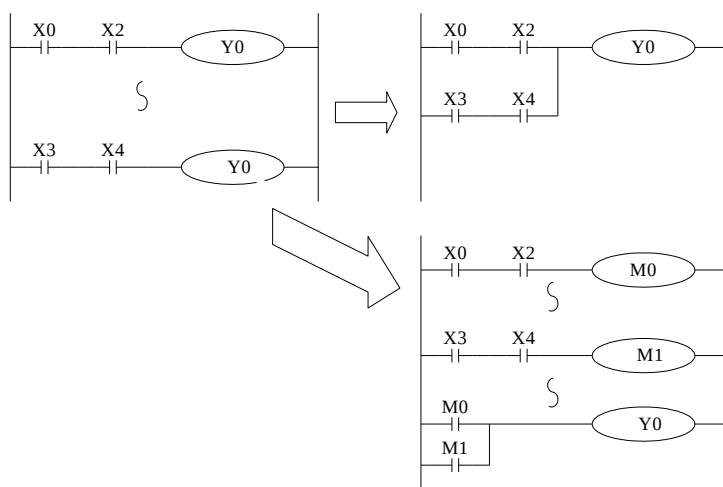
- Оформляйте программу так, чтобы в верху рабочей области было максимальное количество последовательных контактов;
- Оформляйте программу так, чтобы слева в рабочей области было максимальное количество параллельных контактов.

Последовательность выполнения программы

Стандартная последовательность выполнения инструкций: сверху вниз и слева направо. Инструкции последовательного управления также кодируются в соответствии с этой процедурой.

Активация катушки с двойным выходом

Если в управляющей программе активировать катушку дважды, то последнее действие будет в приоритете.

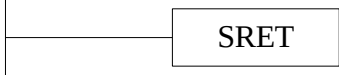
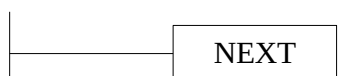


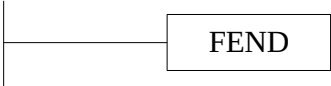
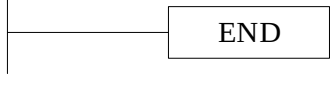
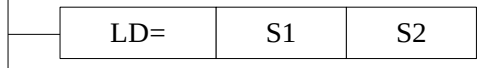
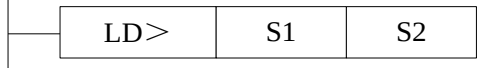
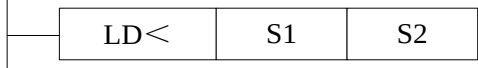
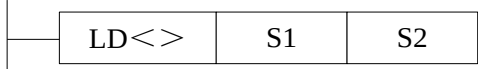
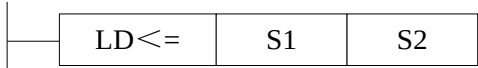
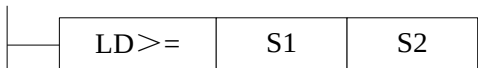
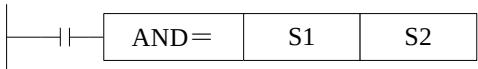
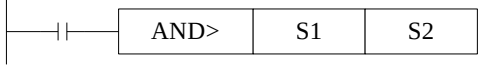
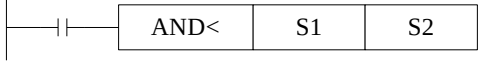
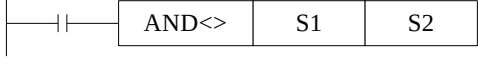
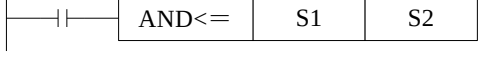
Существуют и другие методы. Например, инструкции перехода или поточные инструкции.

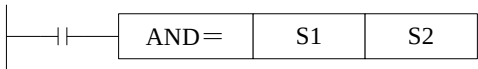
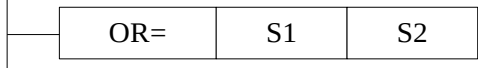
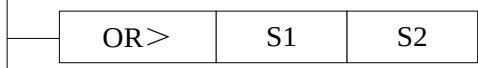
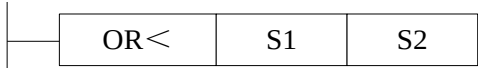
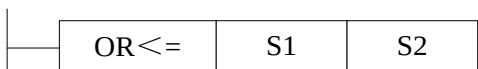
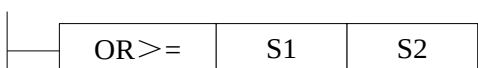
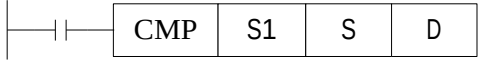
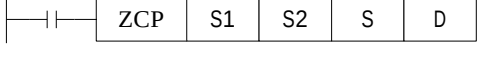
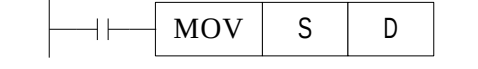
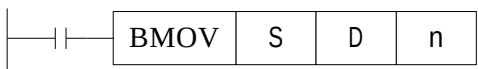
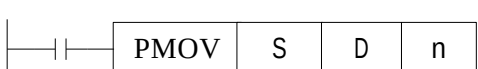
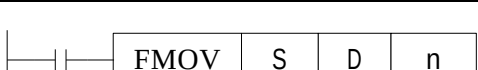
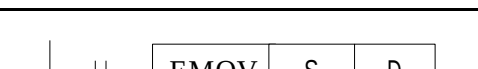
4. Прикладные инструкции

В этой главе мы описываем функции прикладных инструкций для ПЛК серии PMP20.

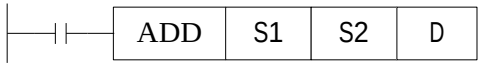
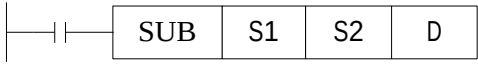
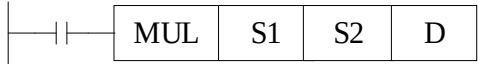
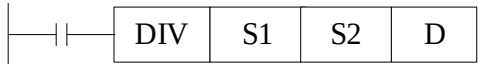
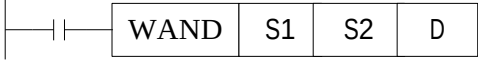
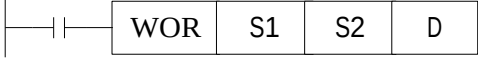
4.1 Список инструкций по применению

Обозначение	Функция	Лестничная диаграмма (LD)	Глава
Управление потоком выполнения программы			
CJ	Переход внутри программы		4-3-1
CALL	Вызов подпрограммы		4-3-2
SRET	Возврат (конец) подпрограммы		4-3-2
STL	Начало программного потока		4-3-3
STLE	Конец программного потока		4-3-3
SET	Открыть новый поток и закрыть текущий поток		4-3-3
ST	Открыть новый поток и не закрывать текущий поток		4-3-3
FOR	Начало цикла FOR-NEXT		4-3-4
NEXT	Конец цикла FOR-NEXT		4-3-4

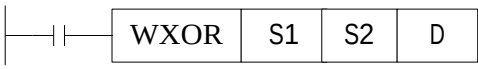
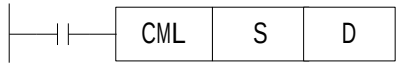
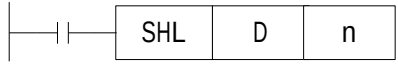
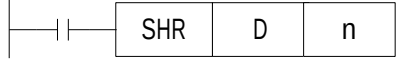
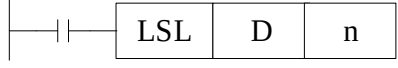
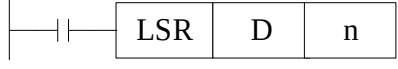
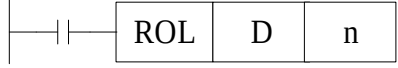
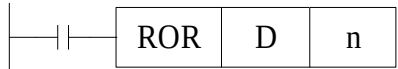
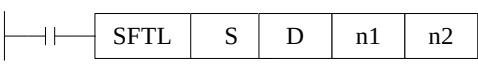
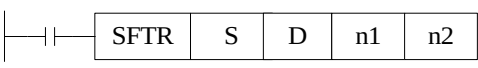
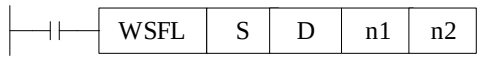
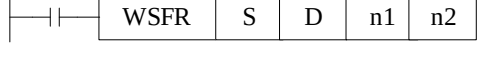
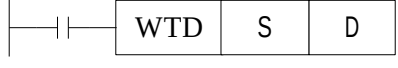
Обозначение	Функция	Лестничная диаграмма (LD)	Глава
FEND	Конец основной программы		4-3-5
END	Процесс ввода/вывода и возврат к шагу 0		4-3-5
Сравнение данных			
LD=	Активный НО контакт, если (S1) = (S2)		4-4-1
LD>	Активный НО контакт, если (S1) > (S2)		4-4-1
LD<	Активный НО контакт, если (S1) < (S2)		4-4-1
LD<>	Активный НО контакт, если (S1) ≠ (S2)		4-4-1
LD<=	Активный НО контакт, если (S1) ≤ (S2)		4-4-1
LD>=	Активный НО контакт, если (S1) ≥ (S2)		4-4-1
AND=	Логическое “И” (последовательное соединение), если (S1) = (S2)		4-4-2
AND>	Логическое “И” (последовательное соединение), если (S1) > (S2)		4-4-2
AND<	Логическое “И” (последовательное соединение), если (S1) < (S2)		4-4-2
AND<>	Логическое “И” (последовательное соединение), если (S1) ≠ (S2)		4-4-2
AND<=	Логическое “И” (последовательное соединение), если (S1) ≤ (S2)		4-4-2

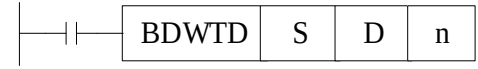
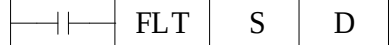
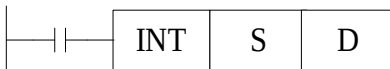
Обозначение	Функция	Лестничная диаграмма (LD)	Глава
AND>=	Логическое “И” (последовательное соединение), если $(S1) \geq (S2)$		4-4-2
OR=	Логическое “ИЛИ” (параллельное соединение), если $(S1) = (S2)$		4-4-3
OR>	Логическое “ИЛИ” (параллельное соединение), если $(S1) > (S2)$		4-4-3
OR<	Логическое “ИЛИ” (параллельное соединение), если $(S1) < (S2)$		4-4-3
OR<>	Логическое “ИЛИ” (параллельное соединение), если $(S1) \neq (S2)$		4-4-3
OR<=	Логическое “ИЛИ” (параллельное соединение), если $(S1) \leq (S2)$		4-4-3
OR>=	Логическое “ИЛИ” (параллельное соединение), если $(S1) \geq (S2)$		4-4-3
Перемещение данных			
CMP	Сравнение числовых данных		4-5-1
ZCP	Сравнение областей числовых данных		4-5-2
MOV	Передача данных		4-5-3
BMOV	Передача блоков		4-5-4
PMOV	Передача блоков (с более высокой скоростью)		4-5-5
FMOV	Передача многобайтовых данных		4-5-6
EMOV	Передача чисел с плавающей точкой		4-5-7



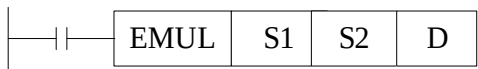
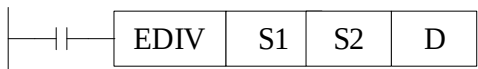
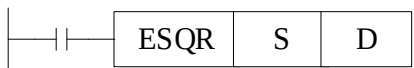
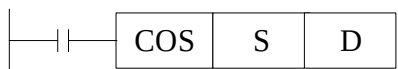
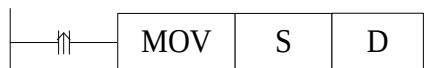
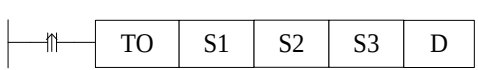
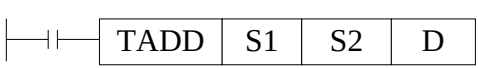
Обозначение	Функция	Лестничная диаграмма (LD)	Глава
FWRT	Запись в регистр FlashROM		4-5-8
MSET	Определение области данных		4-5-9
ZRST	Сброс области данных		4-5-10
SWAP	Обмен старшими и младшими байтами		4-5-11
XCH	Обмен данными		4-5-12
Математические операции с данными			
ADD	Сложение		4-6-1
SUB	Вычитание		4-6-2
MUL	Умножение		4-6-3
DIV	Деление		4-6-4
INC	Инкремент		4-6-5
DEC	Декремент		4-6-5
MEAN	Расчет среднего значения числа		4-6-6
WAND	Логическая связь "И"		4-6-7
WOR	Логическая связь "ИЛИ"		4-6-7



Обозначение	Функция	Лестничная диаграмма (LD)	Глава
WXOR	Логическая связь “НЕ ИЛИ”		4-6-7
CML	Пересылка с инвертированием		4-6-8
NEG	Инверсия данных		4-6-9
Сдвиг данных			
SHL	Арифметический сдвиг влево		4-7-1
SHR	Арифметический сдвиг вправо		4-7-1
LSL	Логический сдвиг влево		4-7-2
LSR	Логический сдвиг вправо		4-7-2
ROL	Ротация (циклический сдвиг) влево		4-7-3
ROR	Ротация (циклический сдвиг) вправо		4-7-3
SFTL	Сдвиг побитно двоичных данных влево		4-7-4
SFTR	Сдвиг побитно двоичных данных вправо		4-7-5
WSFL	Сдвиг данных пословно влево		4-7-6
WSFR	Сдвиг данных пословно вправо		4-7-7
Преобразование данных			
WTD	Преобразование однословного целого в двухсловное целое		4-8-1

Обозначение	Функция	Лестничная диаграмма (LD)	Глава
DWTD	Преобразование целого (32 бит) в целое (64 бит)		4-8-1
BDWTD	Групповое преобразование целых (32 бит) в целые (64 бит)		4-8-2
FLT	Преобразование целого (16 бит) в число с плавающей точкой		4-8-3
DFLT	Преобразование целого (32 бит) в число с плавающей точкой (64 бит)		4-8-3
FLTD	Преобразование целого (64 бит) в число с плавающей точкой		4-8-3
DFLTD	Преобразование целого (32-бит) в число с плавающей точкой двойной точности		4-8-4
QFLTD	Преобразование целого (64-бит) в число с плавающей точкой двойной точности		4-8-4
INT	Преобразование числа с плавающей точкой в целое число		4-8-5
DINTD	Преобразование числа с плавающей точкой двойной точности в целое двойное слово		4-8-6
QINTD	Преобразование числа с плавающей точкой двойной точности в целое число (64-бит)		4-8-6
ECON	Преобразование плавающей точки одинарной точности в плавающую точку двойной точности		4-8-7
BECON	Групповое преобразование данных с плавающей точкой одинарной точности в число с плавающей точкой двойной точности		4-8-8

Обозначение	Функция	Лестничная диаграмма (LD)	Глава
BIN	Преобразование двоично-десятичного значения (BCD) в двоичную систему		4-8-9
BCD	Преобразование двоичного значения в двоично-десятичную систему (BCD)		4-8-10
ASCI	Преобразование шестнадцатеричного кода (HEX) в ASCII		4-8-11
HEX	Преобразование ASCII в шестнадцатеричный код (HEX)		4-8-12
DECO	Декодирование данных		4-8-13
ENCO	Кодирование старших битов		4-8-14
ENCOL	Кодирование младших битов		4-8-15
GRY	Преобразование двоичного кода в код Грея		4-8-16
GBIN	Преобразование кода Грея в двоичный код		4-8-17
Операции с плавающей точкой			
ECMP	Сравнение чисел с плавающей точкой		4-9-1
EZCP	Сравнение областей данных чисел с плавающей точкой		4-9-2
EADD	Сложение чисел с плавающей точкой		4-9-3
ESUB	Вычитание чисел с плавающей точкой		4-9-4

Обозначение	Функция	Лестничная диаграмма (LD)	Глава
EMUL	Умножение чисел с плавающей точкой		4-9-5
EDIV	Деление чисел с плавающей точкой		4-9-6
ESQR	Квадратный корень числа с плавающей точкой		4-9-7
SIN	Синус		4-9-8
COS	Косинус		4-9-9
TAN	Тангенс		4-9-10
ASIN	Арксинус		4-9-11
ACOS	Арккосинус		4-9-12
ATAN	Арктангенс		4-9-13
Работа с часами реального времени			
TRD	Чтение данных часов реального времени (встроенных)		4-10-1
TWR	Запись данных часов реального времени (встроенных)		4-10-2
MOV	Считывание данных с платы часов реального времени		4-10-3
TO	Запись данных на плату часов реального времени		4-10-4
TADD	Сложение данных времени (тактовых данных)		4-10-5

Обозначение	Функция	Лестничная диаграмма (LD)	Глава
TSUB	Вычитание данных времени (тактовых данных)		4-10-6
HTOS	Преобразование единиц “часов”, “минут” и “секунд” в “секунды”		4-10-7
STOH	Преобразование единиц “секунд” в “часы”, “минуты”, “секунды”		4-10-8
TCMP	Сравнение данных времени (час, минута, секунда)		4-10-9
DACMP	Сравнение данных даты (год, месяц, день)		4-10-10

4.2 Метод чтения прикладных инструкций

В этом руководстве применяемые инструкции описаны следующим образом.

1) Описание

ADD [ADD]			
16 бит	ADD	32 бит	DADD
Условие выполнения	Нормально ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Укажите адрес данных или регистра	16 бит/32 бита, BIN
S2	Укажите адрес данных или регистра	16 бит/32 бита, BIN

D	Укажите регистр для хранения результата суммирования	16 бит/32 бита, BIN
---	--	---------------------

3) Подходящие регистры

Операнды	Словесные регистры											Программные регистры						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•	•	•	•		•	•	•										

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

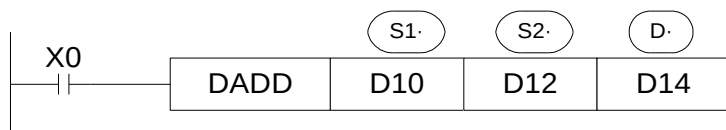
Описание

<16-битная инструкция>



$$(D10) + (D12) \rightarrow (D14)$$

<32-битная инструкция>



$$(D11D10) + (D13D12) \rightarrow (D15D14)$$

- D10 и D12 выполняют двоичное сложение, а результат сохраняется в адресе объекта.

- Старший бит каждого из данных — это бит положительного (0) и отрицательного (1) знака. Эти данные будут выполнять операцию сложения с помощью алгебры. Например, $5 + (-8) = -3$.
- Если результат вычислений равен "0", активен флаг "0". Если результат превышает 323,767 (16 бит) или 2,147,483,648 (32 бита), срабатывает флаг переноса. Если результат превышает -323,768 (16 бит) или -2,147,483,648 (32 бита), срабатывает флаг заимствования.
- При выполнении операции переноса 32 бит младшие 16 бит 32-битного регистра назначаются, адрес регистра, близкий к младшему 16-битному регистру, будет назначен старшим 16 битам 32-битного регистра. Рекомендуется выбирать четный адрес младшего 16-битного регистра.
- Источник и объект могут иметь один и тот же адрес регистра.
- В приведенном выше примере, когда X0 находится в состоянии ВКЛ, операция сложения будет выполняться в каждом периоде сканирования.

Связанный флаг

Флаг	Имя	Функция
SM20	Флаг нуля	ВКЛ: результат вычисления равен нулю ВЫКЛ: результат вычисления не равен нулю
SM21	Флаг заимствования	ВКЛ: результат вычисления превышает 32767 (16 бит) или 2147483647 (32 бита). ВЫКЛ: результат вычисления не превышает 32767 (16 бит) или 2147483647 (32 бита)
SM22	Флаг переноса	ВКЛ: результат вычисления превышает 32767 (16 бит) или 2147483647 (32 бита). ВЫКЛ: результат вычисления не превышает 32767 (16 бит) или 2147483647 (32 бита)

Примечания

- Назначение

Регистр данных ПЛК серии RMP20 — это регистр с одним словом (16 бит), одно слово данных занимает только один регистр, который используется для выполнения однословных команд. Диапазон процесса - десятичный - 327,68~327,67, или шестнадцатеричный 0000~FFFF.

Однословная объектная инструкция

D(NUM)

Команда	D(NUM)	Объект
---------	--------	--------

Двойные слова (32 бита) занимают два регистра данных; адреса двух регистров непрерывны. Диапазон процесса: десятичный - 214,748,364,8~214,748,364,7 или шестнадцатеричный 00000000~FFFFFF.

Двухсловная объектная инструкция

D(NUM+1) D(NUM)

Команда	D(NUM)
---------	--------

Объект	Объект
--------	--------

- Способ представления 32-битной инструкции

Добавьте букву "D" перед 16-битной инструкцией, чтобы представить 32-битную инструкцию.

Например:

Инструкция ADD D0 D2 D4 16 бит

DADD D10 D12 D14 32 бит

-
- | | | |
|----|---|---|
| ※1 |  | Показывает бит флага, следующий за действием инструкции. |
| ※2 |  | исходный операнд, который не меняется при выполнении инструкции. |
| ※3 | | целевой операнд, который будет меняться в процессе работы инструкции. |
| ※4 | | Основные действия инструкции, способы использования, примеры применения, функции расширения, элементы примечаний и т.д. |
-

4.3 Управление потоком выполнения программы

Обозначение	Название инструкции	Глава
CJ	Переход внутри программы	4-3-1
CALL	Вызов подпрограммы	4-3-2
SRET	Возврат (конец) подпрограммы	4-3-2
STL	Начало программного потока	4-3-3
STLE	Конец программного потока	4-3-3
SET	Открыть новый поток и закрыть текущий поток	4-3-3
ST	Открыть новый поток и не закрывать текущий поток	4-3-3
FOR	Начало цикла FOR-NEXT	4-3-4
NEXT	Конец цикла FOR-NEXT	4-3-4
FEND	Конец основной программы	4-3-5
END	Процесс ввода/вывода и возврат к шагу 0	4-3-5

4.3.1 Переход внутри программы [CJ]

1) Описание

Как инструкция по выполнению части программы, CJ сокращает рабочий цикл и позволяет избежать использования двойной катушки.

Переход внутри программы [CJ]			
16 бит	CJ	32 бит	-
Условие выполнения	Нормально ВКЛ/ВЫКЛ катушка	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

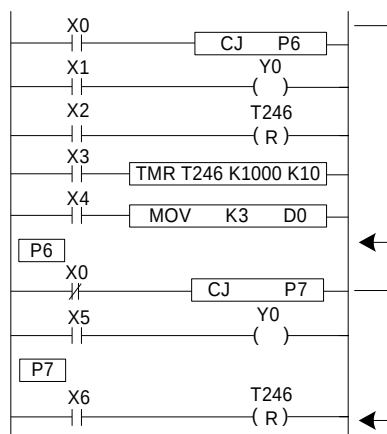
Операнды	Функция	Тип данных
Pn	Переход к цели (с указателем Nr.) P (P0~P9999)	Указатель Nr.

3) Подходящие программные компоненты

Другие	Указатель	
	P	I
	•	

Описание

На приведенном ниже графике, если X0 ВКЛ, происходит переход с первого шага на следующий за меткой P6. Если X0 ВЫКЛ, инструкция перехода не выполняется.



- На левом графике Y0 – выход с двойной катушкой, но когда X0 = ВЫКЛ, активируется X1; когда X0 = ВКЛ, активируется X5.
- CJ не может переходить от одного STL к другому STL.
- После активации таймеров T0~T575, HT0~HT795 и HSC0~HSC30, если выполняется CJ, далее идёт выполнение программы, а выход активируется.

4.3.2 Вызов подпрограммы [CALL] и возврат подпрограммы [SRET]

1) Описание

Вызов подпрограммы.

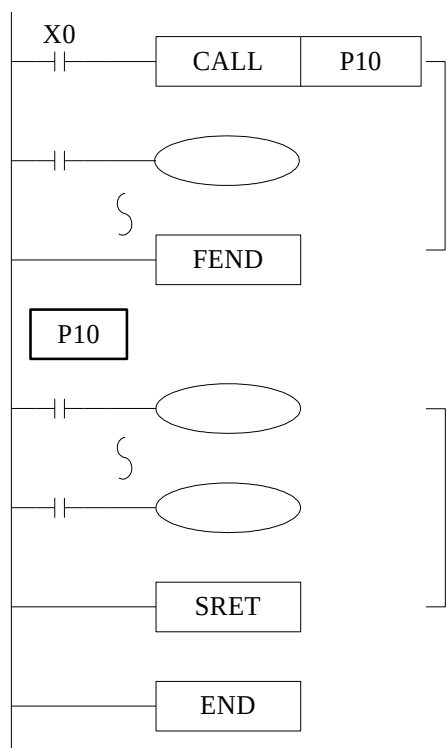
Вызов подпрограммы [CALL]			
16 бит	CALL	32 бит	-
Условие выполнения	Нормально ВКЛ/ВЫКЛ, Восходящий/нисходя- щий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Возврат подпрограммы [SRET]			
16 бит	SRET	32 бит	-
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
Pn	Переход к цели (с указателем №) P (P0~P9999)	Указатель No.

3) Подходящие программные компоненты

Другие	Указатель	
	P	I
	•	

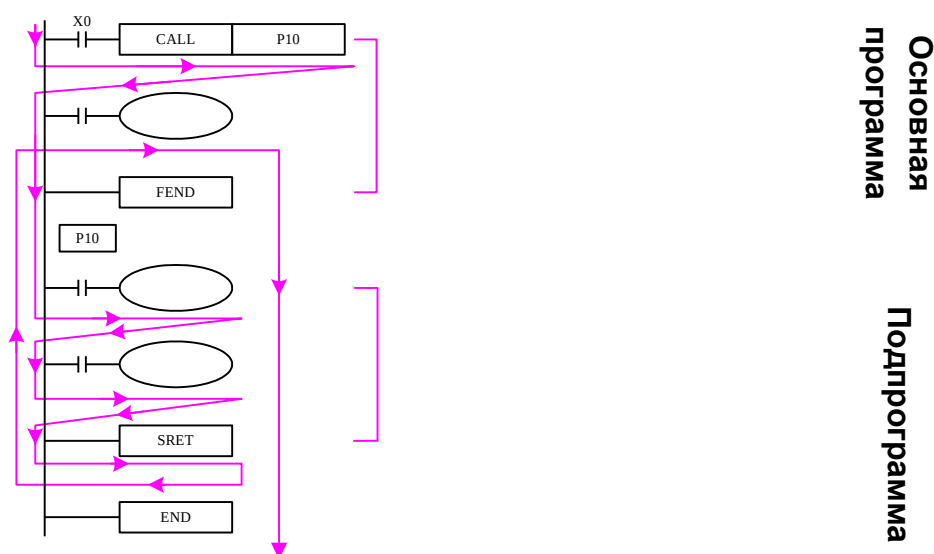


Основная программа

Подпрограмма

- Если X0 = ВКЛ, выполняется инструкция вызова и происходит переход к P10. После выполнения подпрограммы происходит возврат на исходный шаг с помощью инструкции SRET.
- Запрограммируйте метку с помощью инструкции FEND (эта инструкция будет подробнее описана дальше).
- В подпрограмме допускается 9 вызовов, поэтому всего может быть 10 вложений.
- При вызове подпрограммы все таймеры, OUT, PLS, PLF основной программы сохраняют свое состояние.
- Все OUT, PLS, PLF, таймер подпрограммы сохраняют свое состояние при возврате подпрограммы.
- Не используйте импульсы, счетчики или таймеры в подпрограммах, которые не могут быть выполнены за один период сканирования.

Диаграмма выполнения подпрограммы:



Если X0 = ВКЛ, программа выполняется так, как показано фиолетовой стрелкой. Если X0 = ВЫКЛ, инструкция CALL не будет работать; работает только основная программа.

Заметки для написания подпрограммы:

Программируйте операнд после FEND. Pn - начало подпрограммы; SRET - конец подпрограммы. CALL Pn используется для вызова подпрограммы. Диапазон n составляет от 0 до 9999.

Вызов подпрограммы может упростить программирование. Если программа будет использоваться в нескольких местах, оформите ее в виде подпрограммы и вызывайте ее.

4.3.3 Управление потоком выполнения [SET], [ST], [STL], [STLE]

1) Описание

Инструкции для указания начала, конца, открытия и закрытия потока.

Открыть указанный поток, закрыть локальный поток [SET]			
16 бит	SET	32 бита	-
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

Открыть указанный поток, не закрывать локальный поток [ST]			
16 бит	ST	32 бита	-
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Начало потока [STL]			
16 бит	STL	32 бита	-
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Управление потоком выполнения [STLE]			
16 бит	STLE	32 бита	-
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
Sn	Переход к целевому потоку выполнения S	Управление потоком выполнения.

3) Подходящие программные компоненты

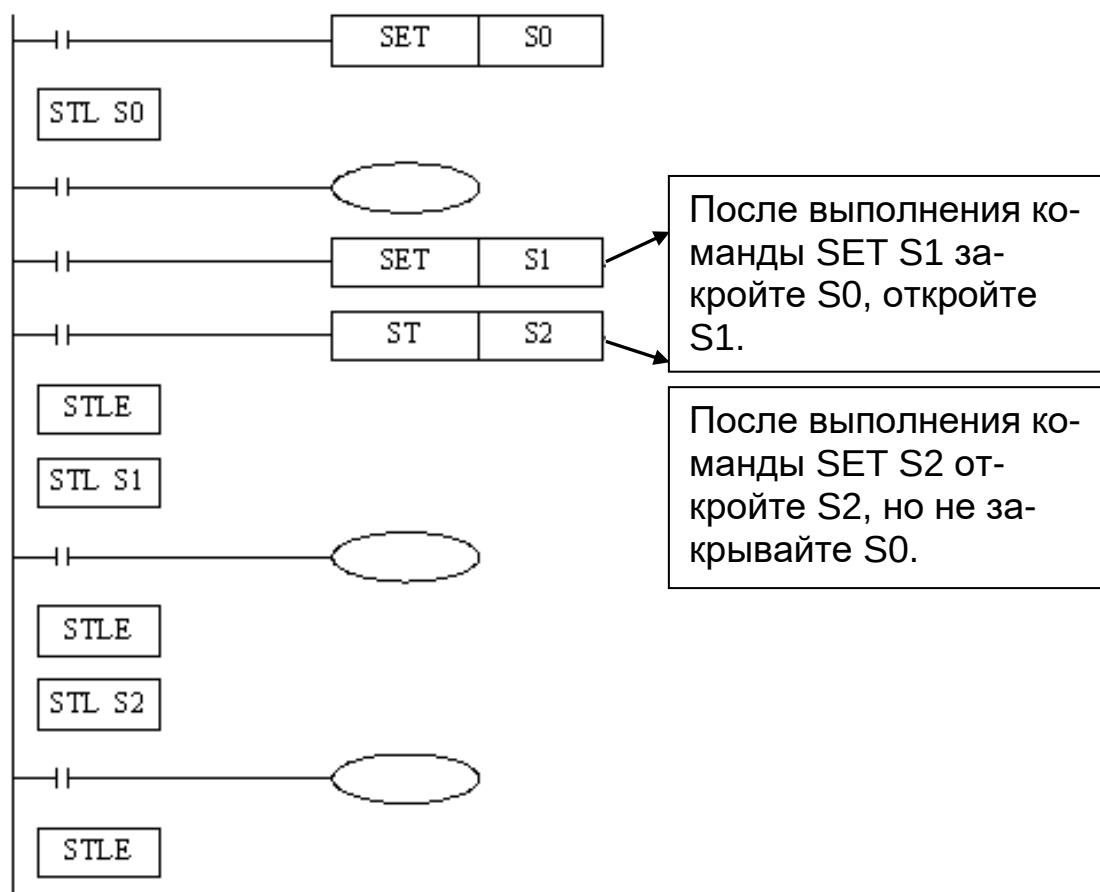
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	КН	ID	QD	X	Y	M	S	T	C	Dn.m
Sn															•			

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

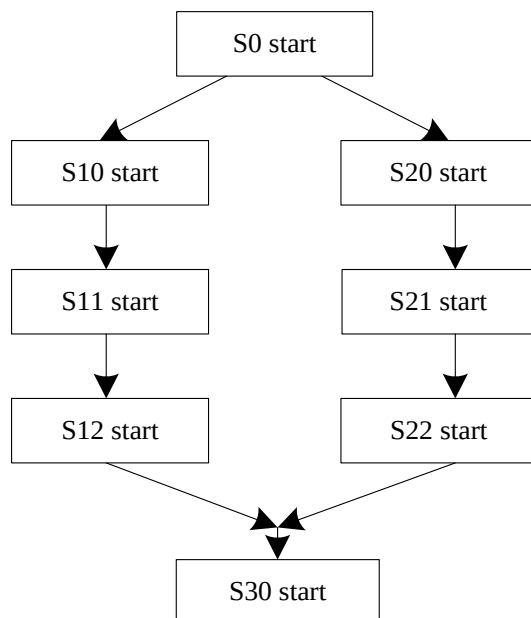
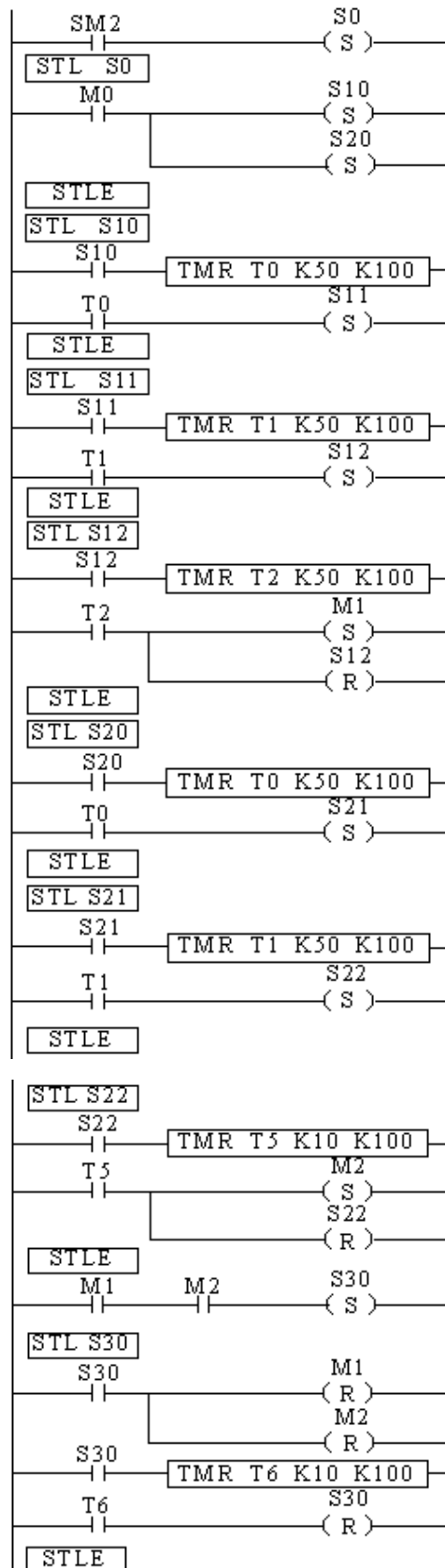
- STL и STLE следует использовать парами. STL представляет начало потока; STLE представляет конец потока.
- Каждое управление потоком выполнения является независимым. Такое управление не может быть вложенным. Нет необходимости записывать поток в виде порядка S0, S1, S2... Вы можете сделать такой порядок. Например, выполняя S10, затем S5, S0.
- После выполнения инструкции **SET Sxxx** поток, указанный этими инструкциями, становится ВКЛ.
- После выполнения инструкции **RST Sxxx** указанный поток ВЫКЛ.
- В потоке S0, SET S1 закройте текущий поток S0, откройте поток S1.
- В потоке S0, ST S2 откройте поток S2, но не закрывайте поток S0.
- Когда управление потоком выполнения переходит из состояния ВКЛ в состоянии ВЫКЛ, в потоке сбрасываются OUT, PLS, PLF и т.д.
- Инструкция ST обычно используется, когда программа должна выполнять много потоков одновременно.
- После выполнения инструкции **SET Sxxx** и перехода к следующему потоку, импульсные инструкции в предыдущем потоке будут закрыты (включая одно-сегментные, многосегментные, относительные или абсолютные, возврат к началу).



Пример

Пример 1: потоки идут по разветвлению, затем сливаются в один поток.

Схема программы:



Объяснение программы:

Когда SM2 включен, S0 -> ВКЛ. Когда M0 включен, S10 -> ВКЛ и S20 -> ВКЛ.

В ветке S10 выполняется S10, S11 и S12. M1 в состоянии ВКЛ. означает, что ветка S10 завершена.

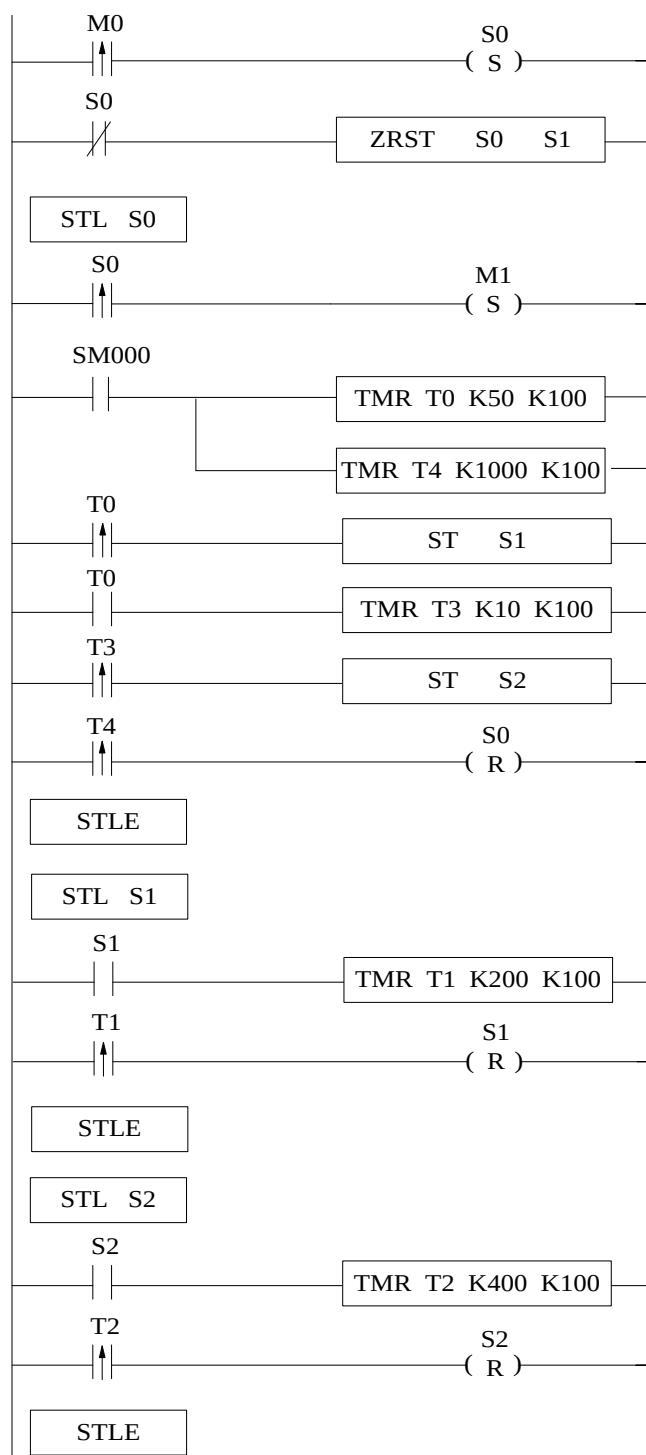
В ветке S20 выполняется S20, S21 и S22. M2 в состоянии ВКЛ. означает, что ветка S20 завершена.

Когда обе ветки S10 и S20 закончатся, установите S30. Когда S30 закончится, сбросьте S30.

Пример 2:

Управление потоком выполнения.

Когда S0 работает, S1 и S2 переходят в состояние ВКЛ.; состояние работы S1 сохраняется. По истечению некоторого времени, S0 закроется и принудительно закроются S1 и S2.



4.3.4 [FOR] и [NEXT]

1) Описание

Выполните программу в цикле между **FOR** и **NEXT** с указанным временем.

Начало цикла [FOR]			
16 бит	FOR	32 бита	-
Условие выполнения	Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Конец цикла [FOR-NEXT]			
16 бит	NEXT	32 бита	-
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Время цикла программы между FOR и NEXT	16 бит, BIN

3) Подходящие программные компоненты

Операнды	Словесные программные компоненты										Программные компоненты							
	Система								Константа	Модуль	Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
Sn	•								•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

Инструкции FOR.NEXT должны быть в паре. Вложенность разрешена, а уровень вложенности равен 8.

Программа после NEXT не будет выполнена, пока программа между FOR и NEXT не будет выполнена в течение указанного времени.

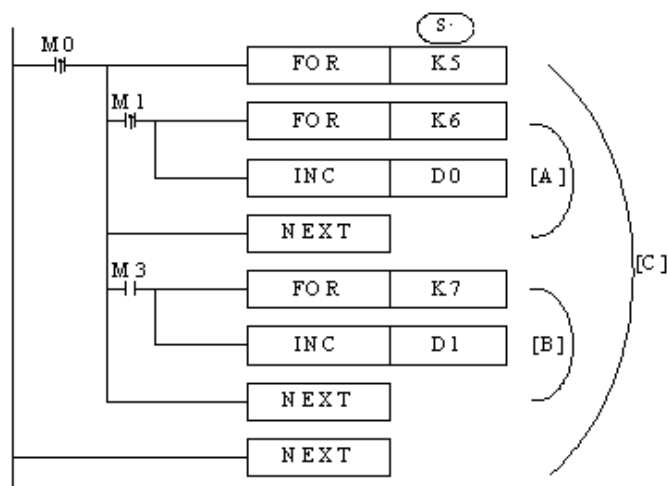
Между инструкциями FOR и NEXT, LDP и LDF действуют в течение одного раза. Каждый раз, когда M0 переходит из состояния ВЫКЛ в состоянии ВКЛ, а M1 в состоянии ВКЛ, цикл [A] выполняется $5 \times 6 = 30$ раз.

Каждый раз, когда M0 переходит из состояния ВЫКЛ в состоянии ВКЛ, а M3 в состоянии ВКЛ, цикл [B] выполняется $5 \times 7 = 35$ раз.

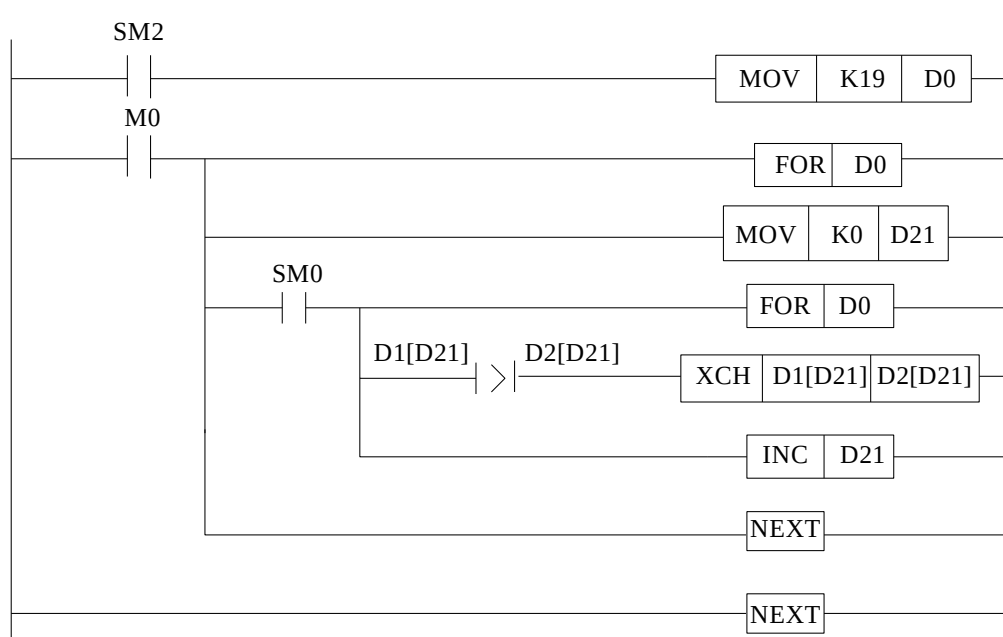
Если циклов много, цикл сканирования будет увеличен. Может возникнуть ошибка таймера мониторинга, обратите на это внимание.

Если NEXT находится перед FOR, или NEXT отсутствует, или NEXT находится за FEND, END, или FOR и номер NEXT не равен, произойдет ошибка.

Между FOR~NEXT вложение CJ не допускается. FOR~NEXT должны быть попарно в одном STL.



Пример 1: когда M0 находится в состоянии ВКЛ, FOR NEXT начинает сортировать числа в диапазоне от D1 до D20 от маленького к большому. D21 - это значение смещения. Если программа содержит большое количество сортировок, используйте язык C, чтобы сэкономить время программирования и время сканирования.



LD	SM2		//SM2 - это начальная катушка ВКЛ
MOV	K19	D0	//время выполнения цикла FOR
LD	M0		//M0 для запуска цикла FOR
MCS			//
FOR	D0		//Вложенный цикл FOR, время цикла - D0
MOV	K0	D21	//смещение начинается с 0
LD	SM0		//SM0 всегда находится в состоянии ВКЛ
MCS			//
FOR		D0	//вложенный цикл FOR, время цикла - D0
LD>	D1[D21]	D2[D21]	//если текущие данные больше следующих, ВКЛ
XCH	D1[D21]	D2[D21]	//обмен данными
LD	SM0		//M8000 всегда находится в состоянии ВКЛ.
INC	D21		//увеличиваем единицу для D21
MCR			//
NEXT			// Под второй FOR
MCR			//
NEXT			// Под первый FOR

4.3.5 [FEND] и [END]

1) Описание

FEND означает завершение основной программы, а END означает завершение программы.

Конец основной программы [FEND]			
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Конец программы [END]			
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

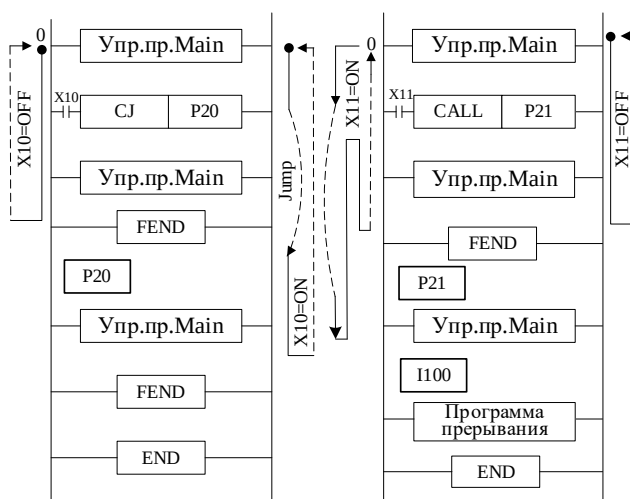
Операнды	Функция	Тип данных
Нет	-	-

3) Подходящие программные компоненты

Нет

Описание

Инструкция [FEND] представляет собой конец основной программы, ее функция аналогична END - обработка вывода/ввода, контроль обновления таймера, возврат к шагу 0 программы.



Если операнд инструкции CALL программируется за инструкцией FEND, то должна быть инструкция SRET. Если указатель прерывания программируется за инструкцией FEND, должна быть инструкция IRET.

После выполнения инструкции CALL и перед выполнением инструкции SRET, если выполнить инструкцию FEND; или выполнить инструкцию FEND после выполнения инструкции FOR и перед выполнением NEXT, произойдет ошибка.

Если Вы используете много инструкций FEND, составьте программу или под-программу между последней инструкцией FEND и инструкцией END.

4.4 Функция сравнения данных

Обозначение	Функция	Глава
LD=	LD активируется, когда $(S1) = (S2)$	4-4-1
LD>	LD активируется, когда $(S1) > (S2)$	4-4-1
LD<	LD активируется, когда $(S1) < (S2)$	4-4-1
LD<>	LD активируется, когда $(S1) \neq (S2)$	4-4-1
LD<=	LD активируется, когда $(S1) \leq (S2)$	4-4-1
LD>=	LD активируется, когда $(S1) \geq (S2)$	4-4-1
AND=※ ¹	AND активируется, если $(S1) = (S2)$	4-4-2
AND>※ ¹	AND активируется, если $(S1) > (S2)$	4-4-2
AND<※ ¹	AND активируется, если $(S1) < (S2)$	4-4-2
AND<>※ ¹	AND активируется, если $(S1) \neq (S2)$	4-4-2
AND>=※ ¹	AND активируется, если $(S1) \leq (S2)$	4-4-2
AND<=※ ¹	AND активируется, если $(S1) \geq (S2)$	4-4-2
OR=※ ¹	OR активируется, если $(S1) = (S2)$	4-4-3

OR>*1	OR активируется, если (S1) > (S2)	4-4-3
OR<*1	OR активируется, если (S1) < (S2)	4-4-3
OR<>*1	OR активируется, если (S1) ≠ (S2)	4-4-3
OR>=*1	OR активируется, если (S1) ≥ (S2)	4-4-3
OR<=*1	OR активируется, если (S1) ≤ (S2)	4-4-3

4.4.1 Сравнение LD [LD]

1) Описание

LD - это инструкция сравнения.

Сравнение LD [LD]			
16 бит	Подходящие модели	32 бита	Подходящие модели
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Сравниваемый адрес числа	16/32 бита, BIN
S2	Адрес компаранды	16/32 бита, BIN

3) Подходящие программные компоненты

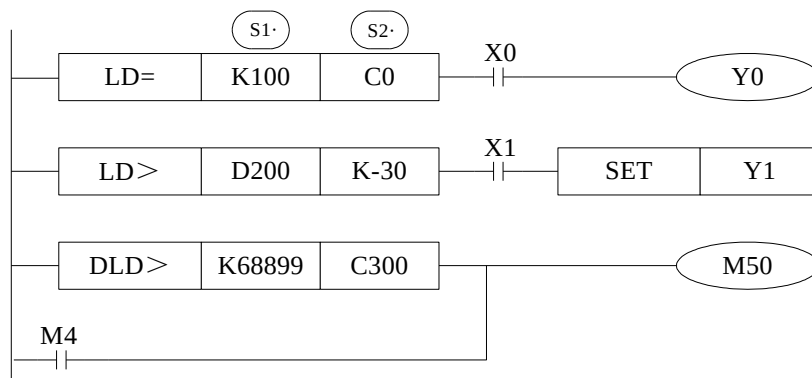
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	КН	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

16-битная инструкция	32-битная инструкция	Условие	Обратное условие
LD=	DLD=	$(S1) = (S2)$	$(S1) \neq (S2)$
LD>	DLD>	$(S1) > (S2)$	$(S1) \leq (S2)$
LD<	DLD<	$(S1) < (S2)$	$(S1) \geq (S2)$
LD<>	DLD<>	$(S1) \neq (S2)$	$(S1) = (S2)$
LD<=	DLD<=	$(S1) \leq (S2)$	$(S1) > (S2)$
LD>=	DLD>=	$(S1) \geq (S2)$	$(S1) < (S2)$



Примечания

- Если старший бит исходных данных (16 бит: b15, 32 бита: b31) равен 1, данные представляются отрицательным числом.
- Для сравнения 32-битного счетчика следует использовать 32-битную инструкцию. Если использовать 16-битную инструкцию, программа или операция будет ошибочной.

4.4.2 Последовательное сравнение [AND]

1) Описание

AND: инструкция сравнения последовательных соединений.

AND сравнение [AND]			
16 бит	Подходящие модели	32 бита	Подходящие модели
Условие выполнения	Нормально ВКЛ/ВЫКЛ катушка	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Сравниваемый адрес числа	16/32 бит, BIN
S2	Адрес	16/32 бит, BIN

3) Подходящие программные компоненты

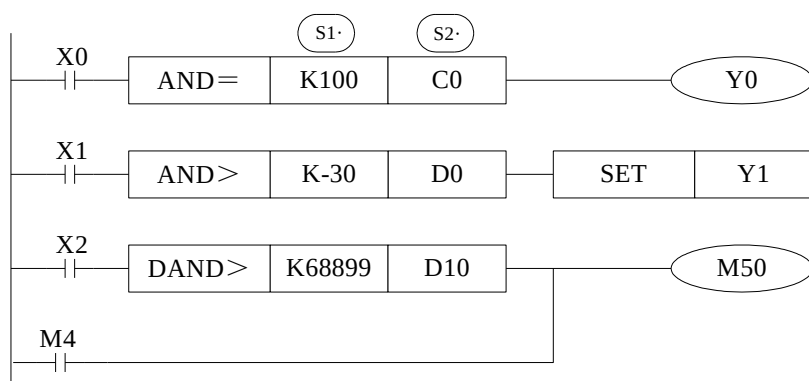
Операнды	Словесные программные компоненты										Программные компоненты							
	Система								Константа	Модуль	Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

16-битная инструкция	32-битная инструкция	Условие	Обратное условие
AND=	DAND=	$(S1) = (S2)$	$(S1) \neq (S2)$
AND>	DAND>	$(S1) > (S2)$	$(S1) \leq (S2)$
AND<	DAND<	$(S1) < (S2)$	$(S1) \geq (S2)$
AND<>	DAND<>	$(S1) \neq (S2)$	$(S1) = (S2)$
AND<=	DAND<=	$(S1) \leq (S2)$	$(S1) > (S2)$
AND>=	DAND>=	$(S1) \geq (S2)$	$(S1) < (S2)$



Примечания

Когда старший бит исходных данных (16 бит: b15, 32 бита: b31) равен 1, он представляется отрицательным числом.

Для сравнения 32-битного счетчика следует использовать 32-битную инструкцию. Если использовать 16-битную инструкцию, программа или операция будет ошибочной.



4.4.3 Параллельное сравнение [OR]

1) Описание

OR: инструкция сравнения параллельных соединений.

Параллельное сравнение [OR]			
16 бит	Как показано ниже	32 бита	Как показано ниже
Условие выполнения	-	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес сравниваемого числа	16/32 бита, BIN
S2	Адрес	16/32 бита, BIN

3) Подходящие программные компоненты

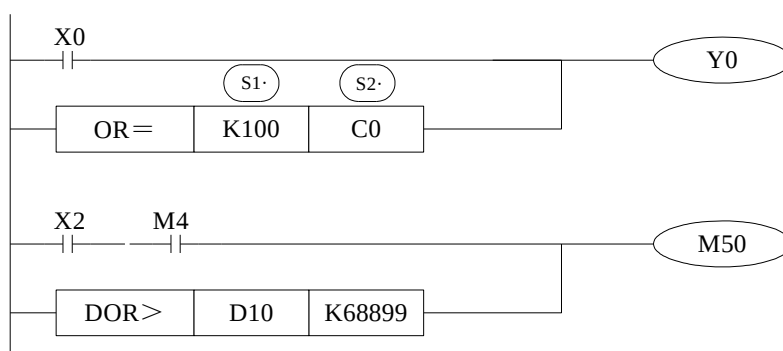
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

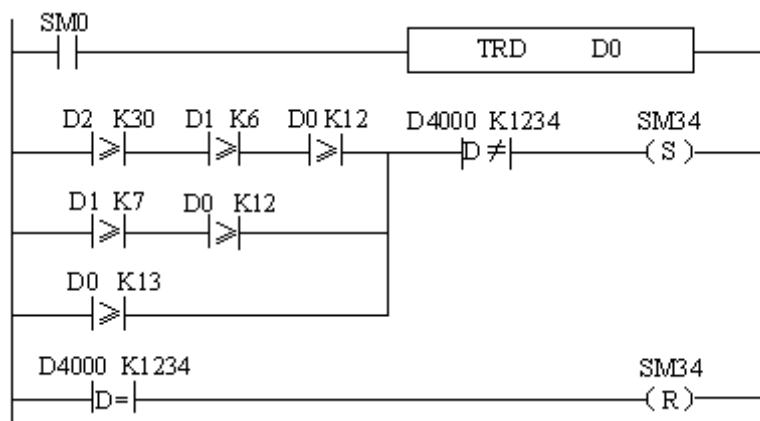
16-битная инструкция	32-битная инструкция	Условие	Обратное условие
OR=	DOR=	$(S1) = (S2)$	$(S1) \neq (S2)$
OR>	DOR>	$(S1) > (S2)$	$(S1) \leq (S2)$
OR<	DOR<	$(S1) < (S2)$	$(S1) \geq (S2)$
OR<>	DOR<>	$(S1) \neq (S2)$	$(S1) = (S2)$
OR<=	DOR<=	$(S1) \leq (S2)$	$(S1) > (S2)$
OR>=	DOR>=	$(S1) \geq (S2)$	$(S1) < (S2)$



Примечания

- Когда старший бит исходных данных (16 бит: b15, 32 бита: b31) равен 1, данные представляются в виде отрицательного числа.
- Для сравнения 32-битного счетчика следует использовать 32-битную инструкцию. Если использовать 16-битную инструкцию, программа или операция выдаст ошибку.

Пример: сброс выходов, когда наступит определенное время. В приведенной ниже программе, когда наступит дата 30 июня, 2012, все выходы будут сброшены. Пароль 1234 хранится в (D4000, D4001). Когда пароль верен, все выходы будут включены.



LD	SM0		//SM0 всегда находится в состоянии ВКЛ
TRD	D0		//Считайте значение часов реального времени
и сохраните в D0~D6			
LD>=	D2	K30	//Часы реального времени ≥ 30
AND>=	D1	K6	//Часы реального времени ≥ 6
AND>=	D0	K12	//Часы реального времени ≥ 12
LD>=	D1	K7	// или часы реального времени ≥ 7
AND>=	D0	K12	//Часы реального времени ≥ 12
ORB			//Или
OR>=	D0	K13	//Часы реального времени ≥ 13
DAND<>	D4000	K1234	//и пароль $\neq 1234$
SET	SM34		//M34 -> ВКЛ, все выходы будут отключены
DLD=	D4000	K1234	//password=1234, правильный пароль
RST	SM34		//перезагрузите M34, все выходы включены

4.5 Инструкции перемещения данных

Обозначение	Функция	Глава
CMP	Сравнение числовых данных	4-5-1
ZCP	Сравнение областей числовых данных	4-5-2
MOV	Передача данных	4-5-3
BMOV	Передача блоков	4-5-4
PMOV	Передача блоков (с более высокой скоростью)	4-5-5
FMOV	Передача многоразрядных данных	4-5-6
EMOV	Передача чисел с плавающей точкой	4-5-7
FWRT	Запись в регистр FlashROM	4-5-8
MSET	Определение области данных	4-5-9
ZRST	Сброс области данных	4-5-10
SWAP	Обмен старшими и младшими байтами	4-5-11
XCH	Обмен данными	4-5-12

4.5.1 Сравнение данных [CMP, DCMR]

1) Описание

Сравнение пары значений, вывод результата.

Сравнение данных [CMP, DCMR]			
16 бит	CMP	32 бита	DCMR
Условие выполнения	ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Укажите данные (для сравнения) или адрес программного компонента	16/32/64 бит, BIN
S	Укажите значение компаранда или адрес программного компонента	16/32/64 бит, BIN
D	Укажите адрес результата сравнения	Бит

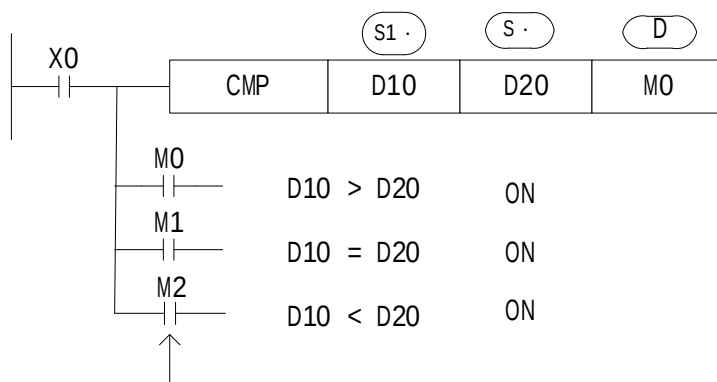
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты							
	Система								Константа	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S1	•	•	•	•	•	•	•	•	•										
S2	•	•	•	•	•	•	•	•	•										
D													•	•	•				

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



X0 -> ВЫКЛ, чтобы остановить выполнение инструкции CMP, M0~M2 сохранят исходное состояние.

- Сравниваются данные **S1**- и **S**-, выводится результат в виде трех программных компонентов.
от **D**.
- **D**, **D+ 1**, **D+ 2**: три программных компонента покажут результаты сравнения.
- Примечание: адреса операндов в инструкциях QCMP должны быть четными.

4.5.2 Сравнение областей числовых данных [ZCP, DZCP]

1) Описание

Сравните текущие данные с данными в области, выведите результат.

Сравнение областей числовых данных [ZCP, DZCP]			
16 бит	ZCP	32 бита	DZCP
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Нижняя граница области	16/32 бита, BIN
S2	Верхняя граница области	16/32 бита, BIN
S	Адрес текущих данных	16/32 бита, BIN
D	Результат сравнения	бит

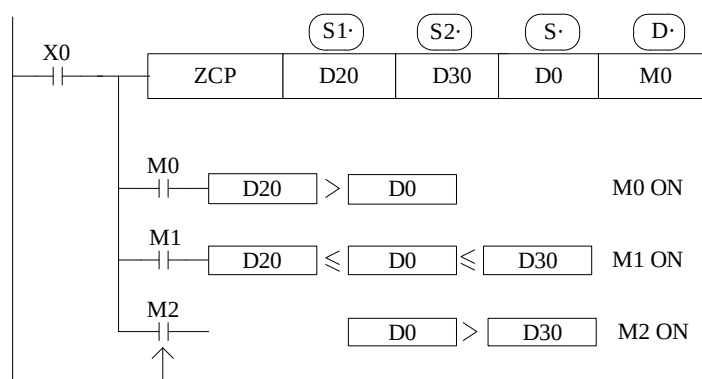
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты										Программные компоненты							
	Система								Константа	Модуль	Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
S	•	•	•	•	•	•	•	•	•									
D													•	•	•			

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



X0 = ВЫКЛ, чтобы остановить выполнение инструкции ZCP, M0~M2 сохранят исходное состояние.

- Сравниваются S- с S1 и S2, выводится три результата, начиная с D.
- D, D+1, D+2: сохранится три результата.

4.5.3 Передача данных MOV [MOV, DMOV]

1) Описание

Переместите указанные данные в другие программные компоненты.

Передача данных MOV [MOV, DMOV]			
16 бит	MOV	32 бита	DMOV
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Укажите исходные данные или адрес регистра	16 бит/32 бита/64 бита, BIN
D	Укажите адрес целевого программного компонента	16 бит/32 бита/64 бита, BIN

3) Подходящие программные компоненты

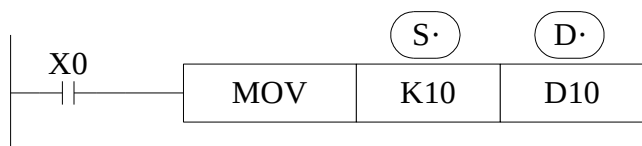
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•	•	•								
D	•		•	•		•	•	•			•							

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

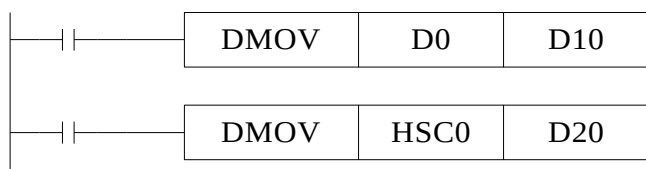
<Перемещение 16 бит данных>



- Переместите исходные данные в целевые
- Когда X0 выключен, данные не изменяются
- Переместите K10 в D10

<Перемещение 32 бит данных>

Используйте DMOV, когда значение составляет 32 бита, например, для инструкции MUL, высокоскоростного счетчика.

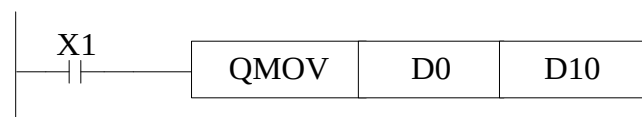


(D1, D0) → (D11, D10)

(текущее значение HSC0) → (D21, D20)

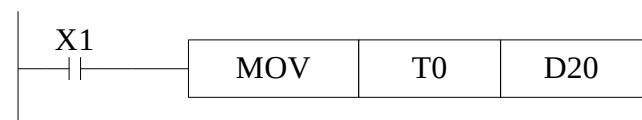
<Перемещение 64 бит данных>

Используйте QMOV, если значение 64 бита, как, например, в инструкции DMUL.



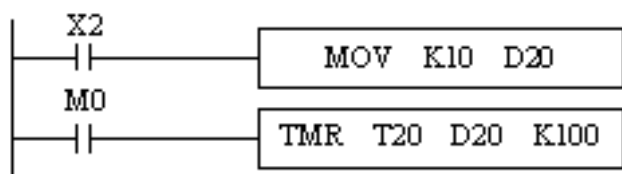
(D3, D2, D1, D0) → (D13, D12, D11, D10)

<Считывание текущего значения счетчика или таймера>.



(текущее значение T0) → (D20)

<Прямая установка значения таймера>.



(K10) (D20)

D20 = K10

4.5.4 Передача блоков [BMOV]

1) Описание

Переместите блок данных в другой программный компонент.

Передача блоков [BMOV]			
16 бит	BMOV	32 бита	-
Условие выполнения	Катушка с нормальным ВКЛ/ВЫКЛ, нарастающий/спадающий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Укажите блок исходных данных или адрес программного компонента	16 бит, BIN; бит
D	Укажите адрес целевых программных компонентов	16 бит, BIN; бит
n	Укажите номер данных перемещения	16 бит, BIN

3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•					•	•	•			
D	•		•	•		•	•	•					•	•	•			
n	•		•	•	•		•	•	•									

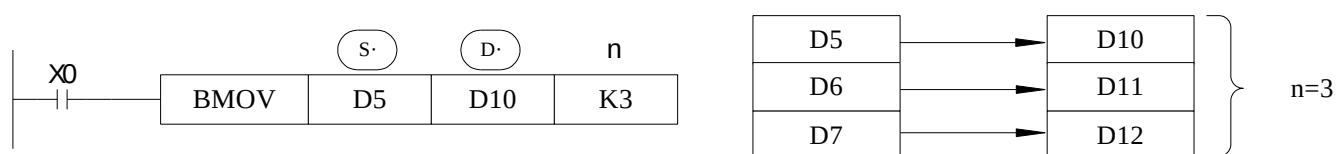
*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

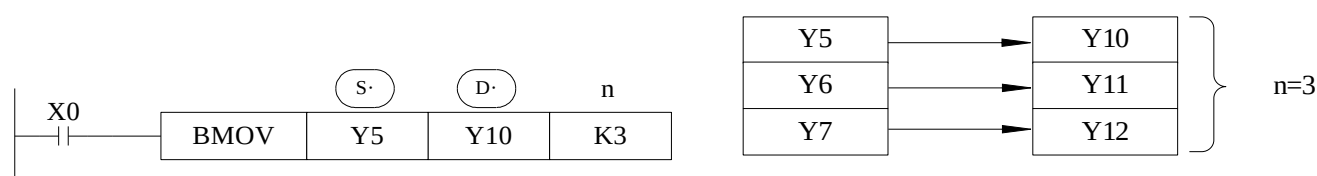
Описание

Переместите исходный блок данных в целевой блок данных. Количество данных равно n.

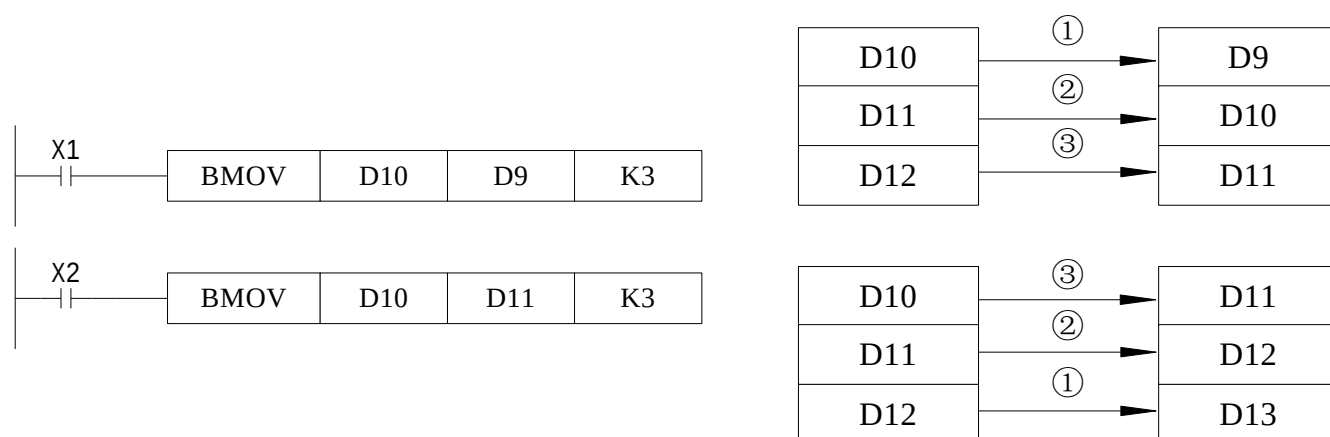
<Перемещение слов>



<Перемещение битов>



Как показано на рисунке ниже, когда диапазон номеров передачи перекрывается, чтобы предотвратить перезапись исходных данных без передачи, в соответствии с методом перекрытия номеров, эта инструкция будет выполняться в порядке ① ~ ③.



4.5.5 Передача блоков (с более высокой скоростью) [PMOV]

1) Описание

Переместите указанный блок данных в другие программные компоненты.

Передача блоков [PMOV]			
16 бит	PMOV	32 бита	-
Условие выполнения	Катушка с нормальным ВКЛ/ВЫКЛ, нарастающий/спадающий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Укажите адрес исходного блока данных или программного компонента	16 бит, BIN; бит
D	Укажите адрес целевых программных компонентов	16 бит, BIN; бит
n	Укажите количество данных	16 бит, BIN

3) Подходящие программные компоненты

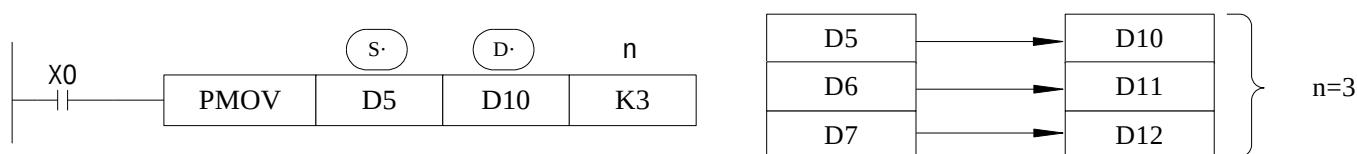
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль			Система					
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•																	
D	•																	
n	•		•	•	•		•	•	•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

Переместите исходный блок данных в целевой блок данных, количество данных равно n.



Функции PMOV и BMOV в основном одинаковы, но скорость выполнения PMOV выше.

PMOV завершается за один цикл сканирования, при выполнении PMOV закройте все прерывания.

Может появиться ошибка, если адрес источника и целевой адрес перекрываются.

4.5.6 Передача многоразрядных данных [FMOV, DFMOV]

1) Описание

Переместите указанные данные в другие программные компоненты.

Передача многоразрядных данных [FMOV, DFMOV]			
16 бит	FMOV	32 бита	DFMOV
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Укажите адрес источника данных или программного компонента	16/32 бита, BIN
D	Укажите адрес целевых программных компонентов	16/32 бита, BIN
n	Укажите объем данных для перемещения	16/32 бита, BIN

3) Подходящие программные компоненты

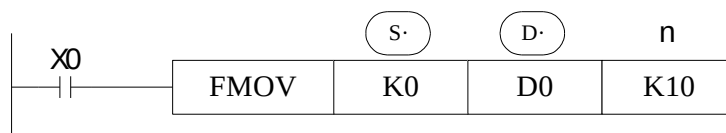
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										
n	•		•	•		•	•	•	•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

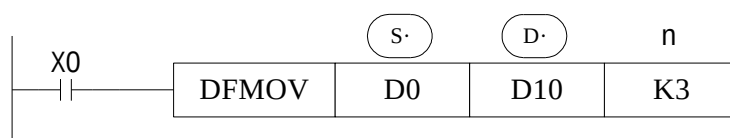
Описание

<16-битная инструкция>



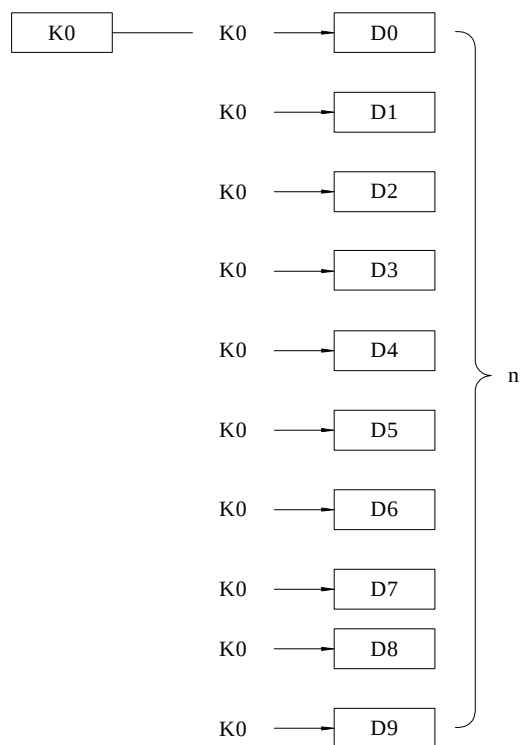
- Переместите K0 в D0~D9, скопируйте одно значение данных в целевой диапазон
- Переместите исходные данные в целевые, количество целевых данных равно n
- Если установленный диапазон превышает целевой диапазон, перейдите к возможному диапазону

<32-битная инструкция >

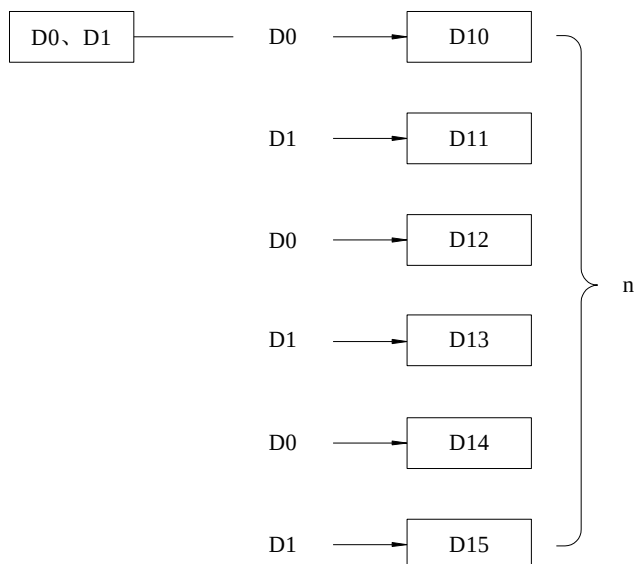


- Переместите D0.D1 в D10.D11:D12.D13:D14.D15.

<16 бит передачи данных>



<32 бита передачи данных>



4.5.7 Передача чисел с плавающей точкой [EMOV]

1) Описание

Переместите номер плавающего элемента на целевой адрес.

Передача чисел с плавающей точкой [EMOV]			
16 бит	-	32 бита	EMOV
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнд	Функция	Тип
S	Адрес исходного программного компонента	32/64 бита, BIN
D	Адрес целевого программного компонента	32/64 бита, BIN

3) Подходящие программные компоненты

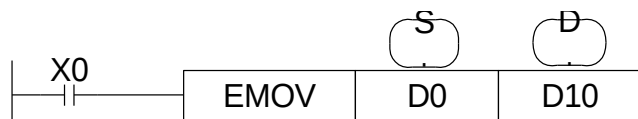
Операнды	Словесные программные компоненты											Программные компоненты							
	Система								Константа	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•			•	•	•	•	•										
D	•					•	•	•											

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

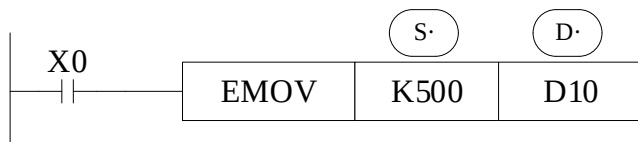
Описание

<32-битная инструкция>



$(D1, D0) \rightarrow (D11, D10)$

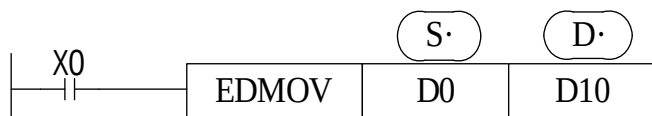
- X0 -> ВКЛ, плавающее число перемещается из (D1, D0) в (D11, D10).
- X0 -> ВЫКЛ, инструкция не работает.



$(K500) \rightarrow (D11, D10)$

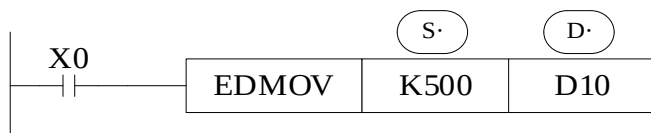
- Если константа К, Н является исходным программным компонентом, она будет преобразована в число с плавающей точкой.
- K500 будет преобразован в значение с плавающей точкой.

<64-битная инструкция>



$(D3, D2, D1, D0) \rightarrow (D13, D12, D11, D10)$

- X0 находится в состоянии ВКЛ, отправьте плавающее число из (D3, D2, D1, D0) в (D13, D12, D11, D10).
- X0 находится в состоянии ВЫКЛ, инструкция не работает.



$(K500) \rightarrow (D13, D12, D11, D10)$

- K500 будет преобразован в значение с плавающей точкой.
- Адреса операндов в инструкциях EDMOV должны быть четными.

4.5.8 Запись в регистр FlashROM [FWRT, DFWRT]

1) Описание

Запишите указанные данные в регистр FlashROM.

Запись в регистр FlashROM [FWRT, DFWRT]			
16 бит	FWRT	32 бита	DFWRT
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Данные записываются в источнике или сохраняются в программном компоненте	16/32/64 бит, BIN
D	Целевой программный компонент	16/32/64 бит
D1	Адрес целевого программного компонента	16/32/64 бит
D2	Объем записываемых данных	16/32/64 бит, BIN

3) Подходящие программные компоненты

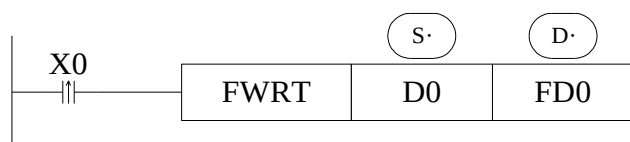
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2		•																
S		•																
D	•		•	•	•	•	•	•	•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

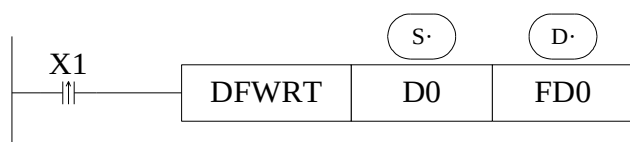
Описание

<Запись одного слова>.



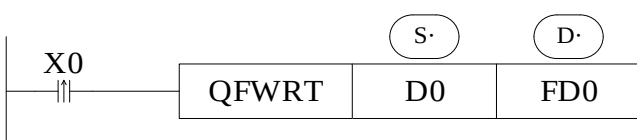
Запись значения из D0 в FD0

<Запись двойного слова >.



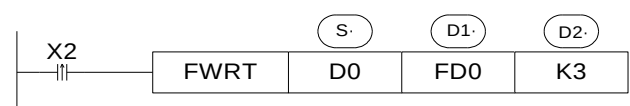
Запись значения из D0, D1 в FD0, FD1

<Запись четырех слов>.



Запись значения из D0, D1, D2, D3 в FD0, FD1, FD2, FD3

< Запись множества слов >.



Запись значения из D0, D1, D2 в FD0, FD1, FD2

-
- ※1 Инструкция FWRT может записывать данные только в регистр FlashROM. FlashROM может сохранять данные даже при выключенном питании. В нем можно хранить важные технические параметры.
 - ※2 Запись FWRT занимает много времени, около 500 мс. Частая запись не рекомендуется.
 - ※3 Ресурс записи FlashROM составляет около 1,000,000 раз. Поэтому мы предлагаем использовать сигнал по фронту (LDP, LDF и т.д.) для активации инструкции.
 - ※4 Частая запись приведет к повреждению FlashROM.
-

4.5.9 Определение области данных [MSET]

1) Описание

Установите программный компонент в определенном диапазоне.

Определение области данных [MSET]			
16 бит	MSET	32 бита	-
Условие выполнения	Обычно ВКЛ/ВЫКЛ; Падающий или нарастающий фронт импульсного сигнала	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D1	Адрес стартового программного компонента	бит
D2	Адрес конечного программного компонента	бит

3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
D1												•	•	•	•	•	•	
D2												•	•	•	•	•	•	

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Установка состояния ВКЛ в M10~M120

- Запись катушек от M10 до M120.
- **D1**, **D2** указаны как один и тот же тип программного компонента, и **D1** < **D2**.
- Когда **D1** > **D2** не будет работать зональный набор, но установите SM409, SD409 = 2.

4.5.10 Сброс области данных [ZRST]

1) Описание

Сбросьте программный компонент в определенном диапазоне.

Сброс области данных [ZRST]			
16 бит	ZRST	32 бита	-
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Падающий или нарастающий фронт импульса	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D1	Стартовый адрес программного компонента	бит, 16 бит, BIN
D2	Конечный адрес программного компонента	бит, 16 бит, BIN

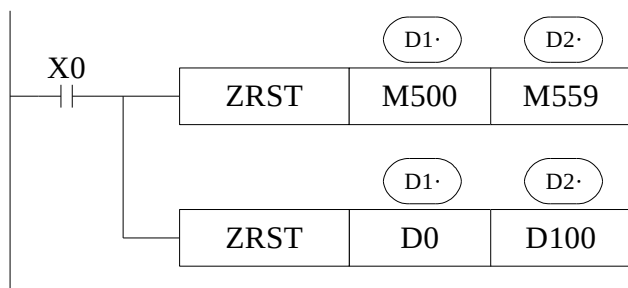
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
D1	•				•	•	•					•	•	•	•	•	•	
D2	•			•	•	•	•					•	•	•	•	•	•	

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Сброс M500~M559

Сброс D0~D100

- **D1, D2** указаны как программные компоненты одного типа, и **D1 < D2**.
- Если **D1 > D2**, сбросьте только указанный программный компонент и установите SM409, SD409 = 2.

Другие инструкции по сбросу

RST может сбросить один программный компонент. Операндом может быть Y, M, HM, S, HS, T, HT, C, HC, TD, HTD, CD, HCD, D, HD.

FMOV может переместить 0 на эти программные компоненты: DX, DY, DM, DS, T(TD), HT(HTD), C(CD), HC(HCD), D, HD.



4.5.11 Обмен старшими и младшими байтами [SWAP].

1) Описание

Поменяйте местами старший и младший байт указанного регистра.

Обмен старшими и младшими байтами [SWAP]			
16 бит	SWAP	32 бита	-
Условие выполнения	Падение или подъем фронт импульса	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного компонента	16 бит, BIN

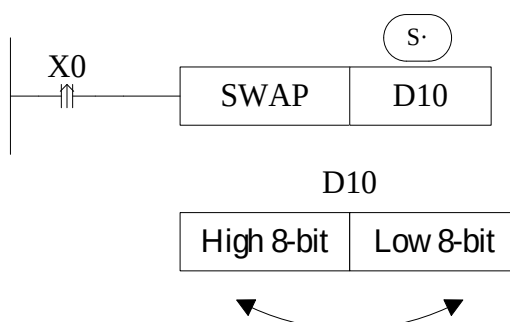
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты							
	Система								Константа	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•		•	•															

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Обменяйте старший 8-битный и младший 8-битный 16-битный регистр.
- Если эта инструкция активирована обычной катушкой ВКЛ/ВЫКЛ, она будет выполняться в каждом периоде сканирования, когда X0 находится в состоянии ВКЛ. Для активации инструкции рекомендуется использовать падающий или нарастающий импульс.

4.5.12 Обмен данными [XCH, DXCH]

1) Описание

Обмен данными в двух программных компонентах.

Обмен данными [XCH, DXCH]			
16 бит	XCH	32 бита	DXCH
Условие выполнения	Восходящий или нисходящий фронт импульса	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D1	Адрес программного компонента	16 бит/32 бита, BIN
D2	Адрес программного компонента	16 бит/32 бита, BIN

3) Подходящие программные компоненты

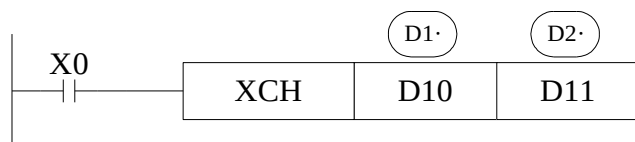
Операнды	Словесные программные компоненты											Программные компоненты							
	Система								Константа	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
D1	•		•	•		•	•	•											
D2	•		•	•		•	•	•											

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

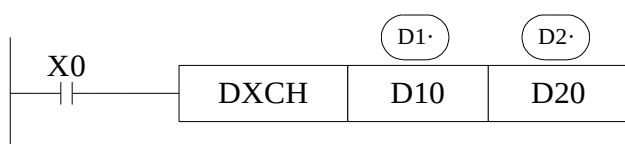
<16-битная инструкция>



До (D10) = 100 → После (D10) = 101
(D11) = 101 (D11) = 100

- Содержимое двух устройств назначения D1 и D2 меняется местами,
- Когда X0 находится в состоянии ВКЛ, инструкция будет выполняться в каждом периоде сканирования. Для активации инструкции рекомендуется подавать нисходящий или восходящий импульс.

<32-битная инструкция>



32-битная инструкция [DXCH] меняет местами значения слов D10, D11 и D20, D21

До		После
(D10) = 100		(D10) = 200
(D11) = 1 (D11D10) = 65636	→	(D11) = 10 (D11D10) = 655460
(D20) = 200		(D20) = 100
(D21) = 10 (D21D20) = 655460		(D21) = 1 (D21D20) = 65636

4.6 Инструкции по работе с данными

Обозначение	Функция	Глава
ADD	Сложение	4-6-1
SUB	Вычитание	4-6-2
MUL	Умножение	4-6-3
DIV	Деление	4-6-4
INC	Инкремент	4-6-5
DEC	Декремент	4-6-5
MEAN	Расчет среднего значения числа	4-6-6
WAND	Логическая связь “И”	4-6-7
WOR	Логическая связь “ИЛИ”	4-6-7
WXOR	Логическая связь “НЕ ИЛИ”	4-6-7
CML	Пересылка с инвертированием	4-6-8
NEG	Инверсия данных	4-6-9

4.6.1 Сложение [ADD, DADD, QADD]

1) Описание

Сложите два числа и сохраните результат.

Сложение [ADD, DADD, QADD]			
16 бит	ADD	32 бита	DADD
Условие выполнения	Нормальный ВКЛ/ВЫКЛ/падающий или нарастающий фронт импульса	Подходящие модели	PMP20

Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
--------------------------------------	---	---------------------------------------	---

2) Операнды

Операнды	Функция	Тип данных
Три операнда		
S1	Адрес данных операции сложения	16 бит/32 бита/64 бита, BIN
S2	Адрес данных операции сложения	16 бит/32 бит/64 бит, BIN
D	Адрес результата	16 бит/32 бит/64 бит, BIN
Два операнда		
D	Данные и адрес данных результата	16 бит/32 бита/64 бита, BIN
S1	Добавьте адрес данных	16 бит/32 бита/64 бита, BIN

3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты							
	Система								Константа	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m	
Три операнда																			
S1	•	•	•	•	•	•	•	•	•										
S2	•	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•											
Два операнда																			
D	•																		

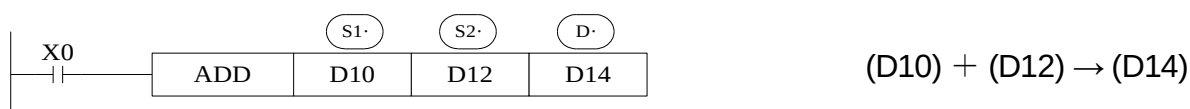
S1	•	•							•									
----	---	---	--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

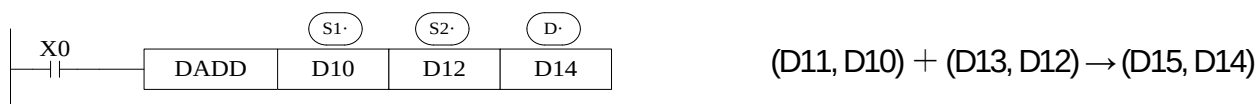
Описание

<Три операнда>



- Два операнда выполняют двоичное сложение и отправляют результат на целевой адрес. Старший бит каждого данных - это бит знака, 0 означает положительный результат, 1 - отрицательный. Все вычисления являются алгебраической обработкой ($5 + (-8) = -3$).
- Если результат вычисления равен "0", действует флаг "0". Если результат превышает 323767 (16-битная операция), или 2147483647 (32-битная операция), или 9223372036854775807 (64-битная операция), срабатывает флаг переноса. Если результат превышает -323768 (16-битная операция) или -2147483648 (32-битная операция) или -9223372036854775808 (64-битная операция), срабатывает флаг переноса (см. соответствующий флаг).
- При выполнении операции 32/64 бита, указывается младшая 16-битная сторона программного компонента слова, а следующий по номеру программный компонент будет использоваться в качестве старшей позиции. Чтобы избежать повторения ID, рекомендуется указывать программный компонент с четным номером.

Например, 32-битная форма предыдущего примера показана на следующем рисунке. В 32-битной операции адрес второго слагаемого должен начинаться с D12, поскольку первое слагаемое занимает регистры D10 и D11. Чтобы избежать повторного занятия регистров, рекомендуется нумеровать программные компоненты четными числами.

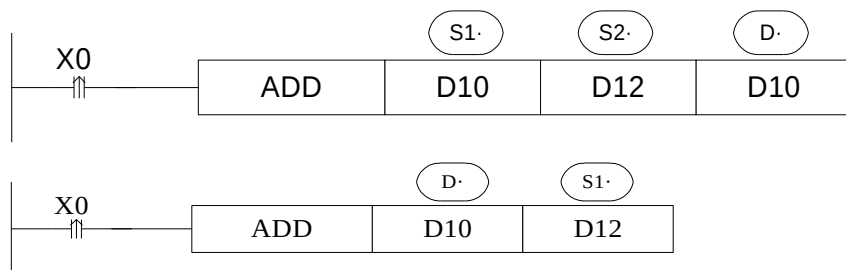


- Адрес источника и целевой адрес могут быть одинаковыми. В приведенном выше примере, когда X0 находится в состоянии ВКЛ, инструкция будет выполняться в каждом периоде сканирования.

<Два операнда>



- Два источника данных выполняют двоичное сложение и отправляют результат в целевой адрес данных. Старший бит каждого данных — это бит знака, 0 означает положительный результат, 1 - отрицательный. Все вычисления являются алгебраической обработкой $(5 + (-8) = -3)$.
- Если результат вычисления равен "0", действует флаг "0". Если результат превышает 323767 (16-битная операция), или 2147483647 (32-битная операция), или 9223372036854775807 (64-битная операция), срабатывает флаг переноса (см. раздел "Связанный флаг"). Если результат превышает -323768 (16-битная операция) или -2147483648 (32-битная операция) или -9223372036854775808 (64-битная операция), срабатывает флаг переноса (см. соответствующий флаг).
- При работе с 32/64 битами указывается младшая 16-битная сторона программного компонента слова, а следующий по номеру программный компонент будет использоваться в качестве старшей позиции. Чтобы избежать повторения ID, мы рекомендуем назначать ID устройства на четные номера.
- Примечание: адреса операндов в инструкциях QADD должны быть четными.
- В приведенном выше примере, когда X0 находится в состоянии ВКЛ, инструкция будет выполняться в каждом периоде сканирования. Для активации инструкции рекомендуется использовать фронт нарастающего или спадающего импульса.



Эти две инструкции одинаковы.

Связанный флаг

Значение флага

Флаг	Имя	Функция
SM020	Нулевой бит	ВКЛ: результат вычисления равен нулю ВЫКЛ: результат вычисления не равен нулю
SM021	Заимствованный бит	ВКЛ: результат вычисления превышает -32768 (16 бит) или -2147483648 (32 бита) или -9,223,372,036,854,775,808 (64 бита), действие флага заимствованного бита. ВЫКЛ: результат вычисления меньше -32768 (16 бит) или -2147483648 (32 бита) или -9,223,372,036,854,775,808 (64 бита)
SM022	Бит переноса	ВКЛ: результат вычисления превышает 32768 (16 бит) или 2147483648 (32 бита) или 9,223,372,036,854,775,807 (64 бита), действие бита флага. ВЫКЛ: результат вычисления меньше 32768 (16 бит) или 2147483648 (32 бита) или 9,223,372,036,854,775,807 (64 бита)

4.6.2 Вычитание [SUB]

1) Описание

Одно число вычитается из другого, результат сохраняется.

Вычитание [SUB, DSUB]			
16 бит	SUB	32 бита	DSUB
Условие выполнения	Нормально ВКЛ/ВЫКЛ/ нарастающий или спадающий фронт импульса	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
Три операнда		
S1	Адрес данных операнды	16 бит/32 бита/64 бита, BIN
S2	Адрес данных операнды	16 бит/32 бита/64 бита, BIN
D	Адрес результата	16 бит/32 бита/64 бита, BIN

Два операнда		
D	Вычитаемые данные и адрес результата	16 бит/32 бита/64 бита, BIN
S1	Вычитание адреса данных	16 бит/32 бита/64 бита, BIN

3) Подходящие программные компоненты

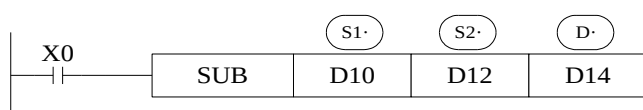
Операнды	Словесные программные компоненты										Программные компоненты							
	Система								Константа	Модуль	Система							
	D	FD	TD	CD	DX	DY	DM	DS	КН	ID	QD	X	Y	M	S	T	C	Dn.m
Три операнда																		
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										
Два операнда																		
D	•																	
S1	•	•							•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<Три операнда>



(D10) - (D12) → (D14)

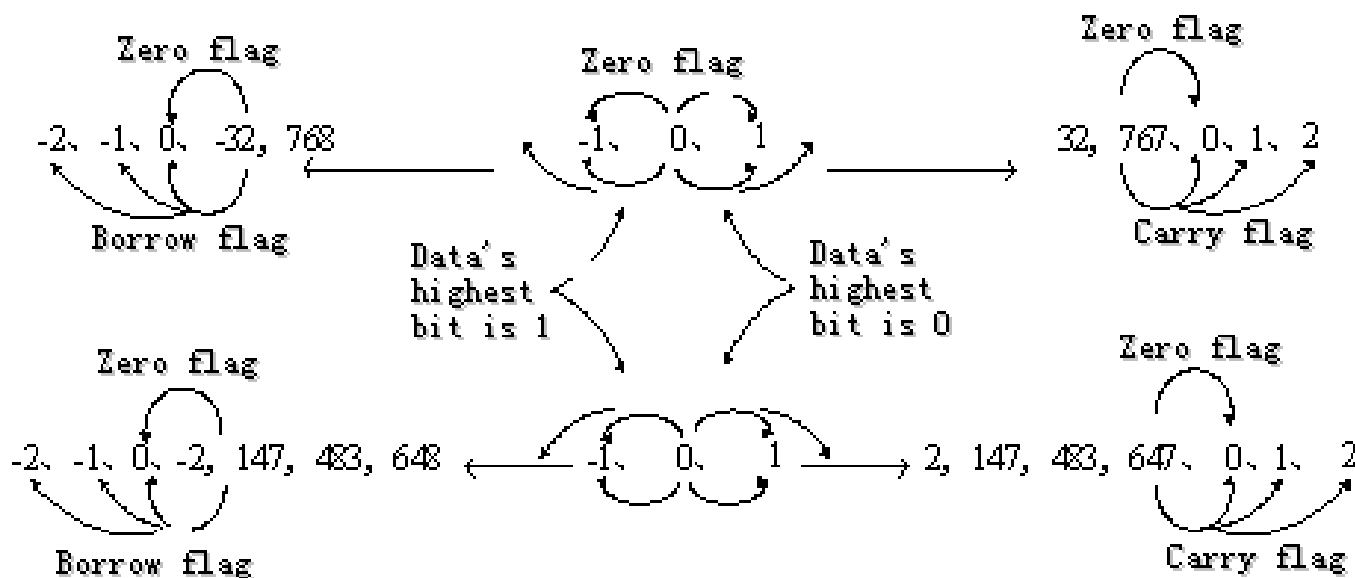
- Значение в S1 алгебраически вычитается из S2. Результат будет сохранен в программном компоненте D.

- Действие каждого флага и способ установки программных компонентов операции 32/64 бита аналогичны предыдущей инструкции ADD.
- Важно: в предыдущей программе, если X0 находится в состоянии ВКЛ, операция SUB будет выполняться каждый цикл сканирования.
- Обратитесь к главе 4-6-1, чтобы узнать про функции флагов.

<Два операнда>



- Назначьте содержимое программного компонента D, алгебраически вычтите содержимое программного компонента, назначенное S1. Результат будет сохранен в программном компоненте, назначенном D.
- Действие каждого флага и способ установки программных компонентов операции 32/64 бита аналогичны предыдущей инструкции ADD.
- Важно: в предыдущей программе, если X0 находится в состоянии ВКЛ, операция SUB будет выполняться каждый цикл сканирования. Для активации команды рекомендуется использовать фронт нарастающего или спадающего импульса.
- Обратитесь к главе 4-6-1, чтобы узнать о действии и функциях флага. Ниже показана взаимосвязь между действием флага и положительным/отрицательным значением:



Примечание: адреса операндов в инструкции QSUB должны быть четными.

4.6.3 Умножение [MUL, DMUL, QMUL]

1) Описание

Умножьте два числа, сохраните результат.

Умножение [MUL, DMUL]			
16 бит	MUL	32 бита	DMUL
Условие выполнения	Нормально ВКЛ/ВЫКЛ/ фронт им- пульса	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операнда	16 бит/32 бита/64 бита, BIN
S2	Адрес данных операнда	16 бит/32 бита/64 бита, BIN
D	Адрес результата	16 бит/32 бита/64 бита, BIN

3) Подходящие программные компоненты

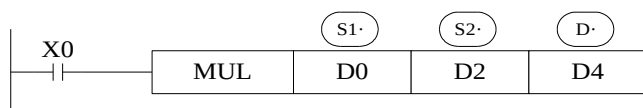
Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль			Система					
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD;
DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM;
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<16-битная операция>



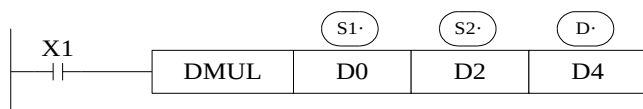
BIN BIN BIN

$(D0) \times (D2) \rightarrow (D5, D4)$

16 бит 16 бит $\rightarrow$ 32 бита

- Содержимое двух исходных устройств перемножается вместе, и результат сохраняется на устройстве назначения в формате 32 бит. Как показано на диаграмме выше: когда $(D0) = 8$, $(D2) = 9$, $(D5, D4) = 72$.
- Старший бит результата - это бит символа: положительный (0), отрицательный (1).
- В приведенном выше примере, когда X0 находится в состоянии ВКЛ, инструкция будет выполняться в каждом периоде сканирования.

<32-битная операция>



BIN; BIN; BIN.

$(D1, D0) \times (D3, D2) \rightarrow (D7, D6, D5, D4)$

32 бита 32 бита $\rightarrow$ 64 бита

- При использовании 32-битной операции результат сохраняется в целевом регистре в формате 64 бит.
- При использовании словесного устройства (word), 64-битные результаты не могут быть отслежены. В этом случае перейдите к работе с плавающими значениями.

<64-битная операция>



BIN; BIN; BIN.

$(D3, D2, D1, D0) \times (D7, D6, D5, D4) \rightarrow (D11, D10, D9, D8)$

64 бита 64 бита $\rightarrow$ 64 бита



- При 64-битной операции целевой адрес использует битовый элемент для получения 64-битного результата (занимая четыре последовательных регистра, поэтому не используйте их повторно). При использовании словесного элемента результат 64-битной операции с данными не может быть непосредственно отслежен. В этом случае рекомендуется использовать арифметику с плавающей точкой.
- Примечание: адреса операндов в инструкции QMUL должны быть четными.

4.6.4 Деление [DIV, DDIV, QDIV]

1) Описание

Разделите два числа и сохраните результат.

Деление [DIV, DDIV]			
16 бит	DIV	32 бита	DDIV
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требование к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операции деления	16 бит/32 бита/64 бита, BIN
S2	Адрес данных операции деления	16 бит/32 бита/64 бита, BIN
D	Адрес результата	16 бит/32 бита/64 бита, BIN

3) Подходящие программные компоненты

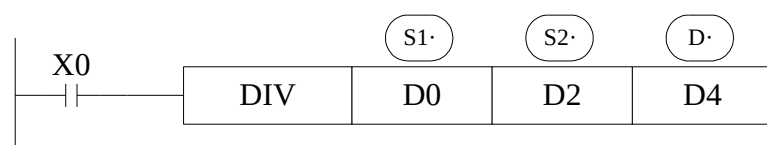
Операнды	Словесные программные компоненты										Программные компоненты							
	Система								Константа	Модуль	Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•	•	•	•	•	•									
S2	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<16-битная операция>



Делимое Делитель Частное Остаток

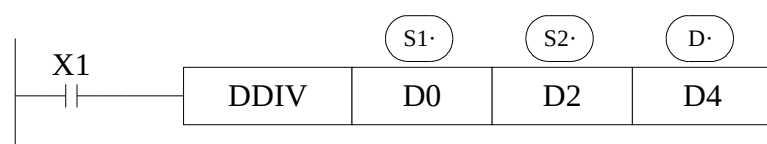
BIN BIN BIN BIN

$(D0) \div (D2) \rightarrow (D4) \dots (D5)$

16 бит 16 бит 16 бит 16 бит

- S1 назначает программный компонент делимого, S2 - программный компонент делителя, а D указывает программный компонент и следующий номер программного компонента, который нужно внести, а также остаток.
- В приведенном выше примере, если вход X0 ВКЛ, операция деления выполняется каждый цикл сканирования.

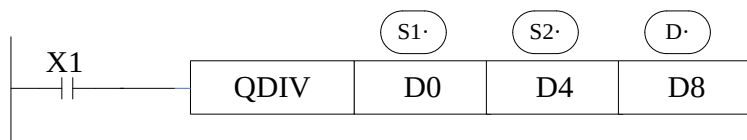
<32-битная операция>



Делимое	Делитель	Результат (частное)	Остаток
BIN	BIN	BIN	BIN
$(D1, D0) \div (D3, D2) \rightarrow (D5, D4) \dots (D7, D6)$			
32 бита	32 бита	32 бита	32 бита

- Делитель состоит из устройства, назначенного S1, и следующего за ним. Делитель составляется устройством, назначенным S2, и следующим устройством. Частное и остаток хранятся в четырех последовательных устройствах, первое из которых назначено D.
- Если значение делителя равно 0, инструкция будет ошибочной.
- Старший бит результата и остатка - это бит символа (положительный: 0, отрицательный: 1). Если делимое или делитель отрицательны, то результат будет отрицательным. Если делимое отрицательно, то и остаток будет отрицательным.

<64-битная операция>



Делимое	Делитель	Частное	Остаток
BIN	BIN	BIN	BIN
$(D3, D2, D1, D0) \div (D7, D6, D5, D4) \rightarrow (D11, D10, D9, D8) \dots (D15, D14, D13, D12)$			
64 бита	64 бита	64 бита	64 бита

- Делитель состоит из компонента, назначенного S1, и следующего за ним. Делитель составляется устройством, назначенным S2, и следующим устройством. Частное и остаток хранятся в четырех последовательных устройствах, первое из которых назначено D.
- Если значение делителя равно 0, инструкция будет ошибочной.
- Старший бит частного и остатка - это бит символа (положительный: 0, отрицательный: 1). Если делимое или делитель отрицательны, то результат будет отрицательным. Если делимое отрицательно, то и остаток будет отрицательным.
- Примечание: адреса операндов в инструкции QDIV должны быть четными.

4.6.5 Инкремент [INC, DINC, QINC] & Декремент [DEC, DDEC, QDEC]

1) Описание

Увеличьте или уменьшите число.

Инкремент [INC, DINC].			
16 бит	INC	32 бита	DINC
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Декремент [DEC, DDEC].			
16 бит	DEC	32 бита	DDEC
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требование к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D	Увеличение или уменьшение данных	16 бит/32 бита/64 бита, BIN

3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
D	•		•	•		•	•	•										

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

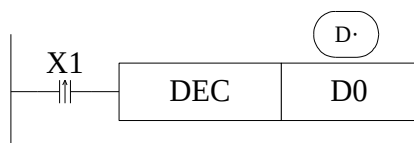
<Инкремент [INC]>



$(D0) + 1 \rightarrow (D0)$

- D увеличится на единицу, когда X0 будет ВКЛ.
- Для 16-битной операции, когда +32767 увеличится на единицу, он станет -32768; при этом будет действовать флаговый бит.
Для 32-битной после увеличения значения +2147483647 $\rightarrow$ -2147483647; действует бит флага.
При работе с 64 битами после увеличения значения +9223372036854775807 на единицу $\rightarrow$ -9223372036854775808, действует бит флага.

<Декремент [DEC]>



$(D0) - 1 \rightarrow (D0)$

- D уменьшится на единицу, когда X1 будет ВКЛ.
- При уменьшении -32767 или -2147483647 на единицу, результатом будет +32767 или +2147483647. Бит флага будет активен. Для 64-битной операции, -9223372036854775808 при уменьшении на единицу равно +9223372036854775807; будет действовать бит флага.
- Адреса операндов в инструкциях QINC и QDEC должны быть четными.

Примечание:

Когда срабатывает команда по фронту, операция автоматического сложения и вычитания выполняется для каждого срабатывания. Если срабатывает нормально открытый/нормально закрытый, операция автоматического сложения и автоматического вычитания будет выполняться в каждом периоде сканирования после проведения.

4.6.6 Расчет среднего значения числа [MEAN, DMEAN]

1) Описание

Получите среднее значение данных.

Расчет среднего значения числа [MEAN, DMEAN]			
16 бит	MEAN	32 бита	DMEAN
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требование к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Начальный адрес источника данных	16 бит/32 бита, BIN
D	Адрес результата	16 бит/32 бита, BIN
n	Количество данных	16 бит/32 бита, BIN

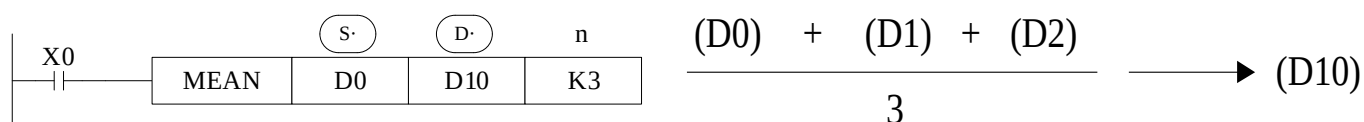
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•		•	•	•										
D	•	•	•	•		•	•	•										
n									•									

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Сохраните среднее значение исходных данных (исходную сумму разделите на исходное количество n) с остатком.
- n не может быть больше, чем количество программных компонентов, иначе произойдет ошибка.

4.6.7 Логическое И [WAND, DWAND], Логическое ИЛИ [WOR, DWOR], Логическое исключающее ИЛИ (НЕ или) [WXOR, DWXOR].

1) Описание

Выполните логические операции AND, OR, XOR для данных.

Логическое И [WAND, DWAND]			
16 бит	WAND	32 бита	DWAND
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Логическое ИЛИ [WOR, DWOR]			
16 бит	WOR	32 бита	DWOR
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Логическое НЕ ИЛИ [WXOR, DWXOR]			
16 бит	WXOR	32 бита	DWXOR
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20

Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
--------------------------------------	---	---------------------------------------	---

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операции	16 бит/32 бита, BIN
S2	Адрес данных операции	16 бит/32 бита, BIN
D	Адрес результата	16 бит/32 бита, BIN

3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты							
	Система								Константа	Модуль	Система								
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m	
S1	•	•	•	•	•	•	•	•	•										
S2	•	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•											

*Примечание:

D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<Логическое И>



$0 \& 0 = 0$ $0 \& 1 = 0$

$1 \& 0 = 0$ $1 \& 1 = 1$

<Логическое ИЛИ>



$0 \text{ или } 0 = 0$ $0 \text{ или } 1 = 1$

$1 \text{ или } 0 = 1$ $1 \text{ или } 1 = 1$

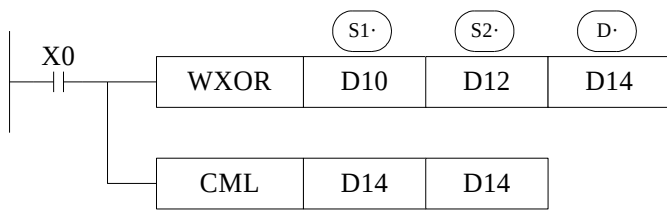
<Логическое НЕ ИЛИ>



$0 \text{ xor } 0 = 0$ $0 \text{ xor } 1 = 1$

$1 \text{ xor } 0 = 1$ $1 \text{ xor } 1 = 0$

Если использовать эту инструкцию вместе с инструкцией CML, можно также выполнить операцию XOR NOT.

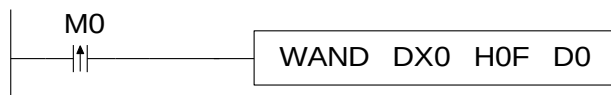


Пример 1:

16 бит данных состоят из X0~X7 и хранятся в D0.

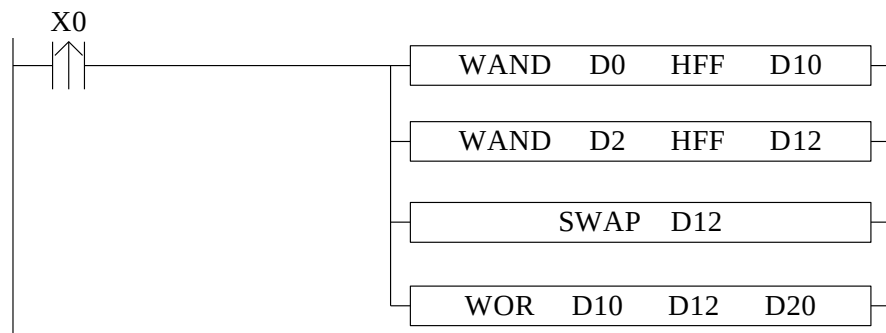


Преобразуйте состояние X0, X1, X2, X3 в код 8421 и сохраните в D0.



Пример 2:

Объедините младшие 8 бит D0 и D2 в слово.



LDP	X0	// Восходящий фронт X0
WAND	D0 HFF D10	//Логическое И, берутся младшие 8 бит D0 и сохраняются в D10
WAND	D2 HFF D12	// Логическое И, берутся младшие 8 бит D2 и сохраняются в D12
SWAP	D12	// Младшие 8 бит и старшие 8 бит меняются местами D12
WOR	D10 D12 D20	// Младшие 8 бит D10 и старшие 8 бит D12 соединяются и сохраняются в D20

4.6.8 Пересылка с инвертированием [CML, DCML]

1) Описание

Логически преобразуйте данные.

4.6.8 Пересылка с инвертированием [CML, DCML]			
16 бит	CML	32 бита	DCML
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес источника данных	16 бит/32 бита, BIN
D	Адрес результата	16 бит/32 бита, BIN

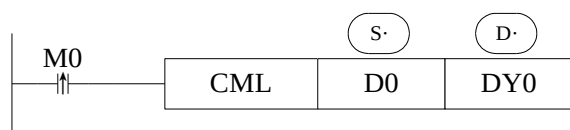
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

*Примечание:

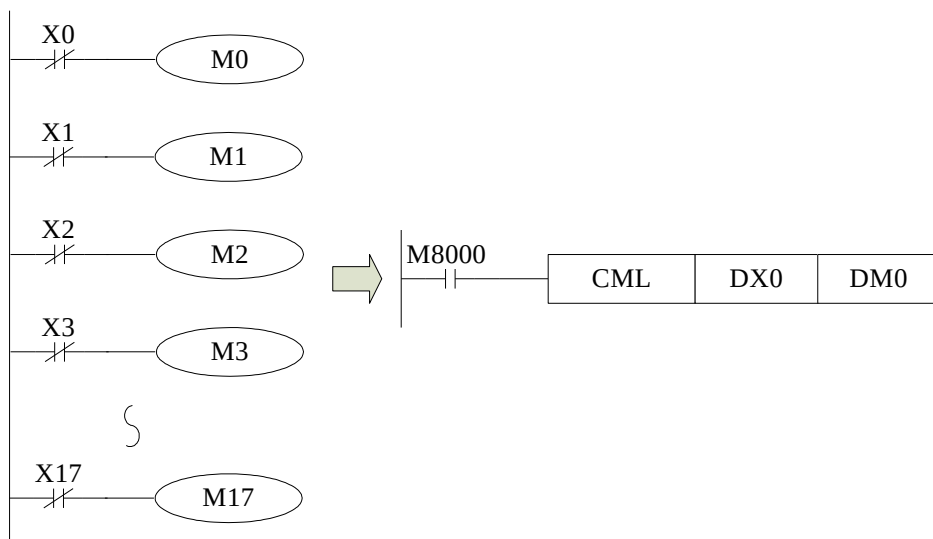
D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Каждый бит данных в источнике инвертируется ($1 \rightarrow 0$, $0 \rightarrow 1$) и отправляется по целевому адресу. Если в источнике используется константа K, она может быть автоматически преобразована в двоичное число.
- Эта инструкция предназначена для вывода логического инвертированного сигнала ПЛК.

<Чтение инвертированного значения со входов>.



- Последовательная управляющая инструкция в левой части может быть обозначена следующей инструкцией CML.

4.6.9 Инвертирование данных [NEG, DNEG]

1) Описание

Получите инвертированные данные.

Инвертирование данных [NEG, DNEG]			
16 бит	NEG	32 бита	DNEG
Условие выполнения	Обычно ВКЛ/ВЫКЛ, Восходящий/нисходящий фронт	Подходящие модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D	Адрес источника данных	16 бит/32 бита, BIN

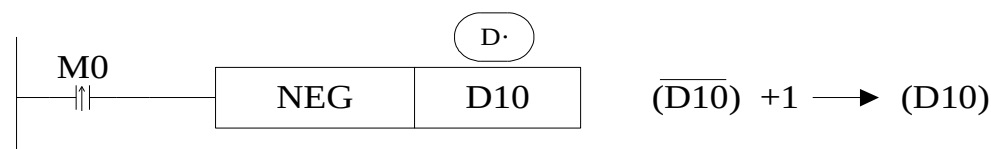
3) Подходящие программные компоненты

Операнды	Словесные программные компоненты											Программные компоненты						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
D	•		•	•		•	•	•										

*Примечание:

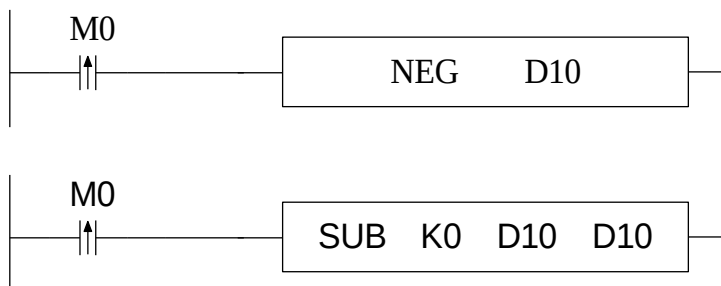
D включает D, HD; TD включает TD, HTD; CD включает CD, HCD, HSCD, HSD; DM включает DM, DHM; DS включает DS, DHS; M включает M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Преобразуйте каждый бит исходных данных (1→0, 0→1), затем прибавьте единицу и сохраните результат в адресе исходных данных.
- Например, исходные данные D10 равны 20, а при приближении нарастающего фронта M0 значение D10 становится равным -20.

Две показанные ниже команды одинаковые.



4.7 Команды сдвига

Мнемонический код	Функция	Глава
SHL	Арифметический сдвиг влево	4-7-1
SHR	Арифметический сдвиг вправо	4-7-1
LSL	Логический сдвиг влево	4-7-2
LSR	Логический сдвиг вправо	4-7-2
ROL	Поворот влево	4-7-3
ROR	Поворот вправо	4-7-3
SFTL	Битовый сдвиг влево	4-7-4
SFTR	Битовый сдвиг вправо	4-7-5
WSFL	Сдвиг слова влево	4-7-6
WSFR	Сдвиг слова вправо	4-7-7

4.7.1 Арифметический сдвиг влево [SHL, DSHL], Арифметический сдвиг вправо [SHR, DSHR]

1) Обзор

Выполнение арифметического сдвига влево/вправо для чисел.

Арифметический сдвиг влево [SHL, DSHL]			
16 бит	SHL	32 бита	DSHL
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Арифметический сдвиг вправо [SHR, DSHR]			
16 бит	SHR	32 бита	DSHR
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D	Адрес исходных данных	16 бит/32 бита, BIN
n	Количество сдвигов влево или вправо	16 бит/32 бита, BIN

3) Соответствующие программные компоненты

Опе- ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон- станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
D	•	•	•	•		•	•	•											
n									•										

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

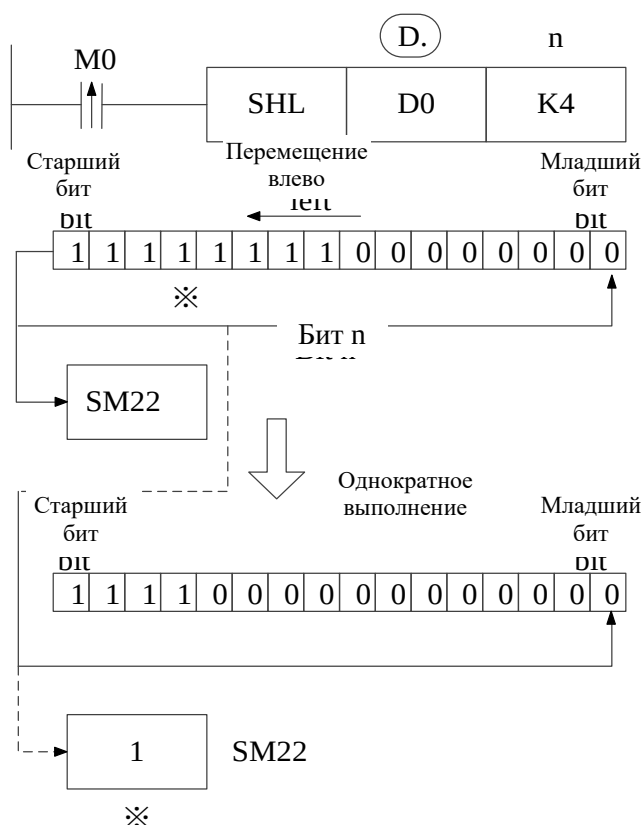
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

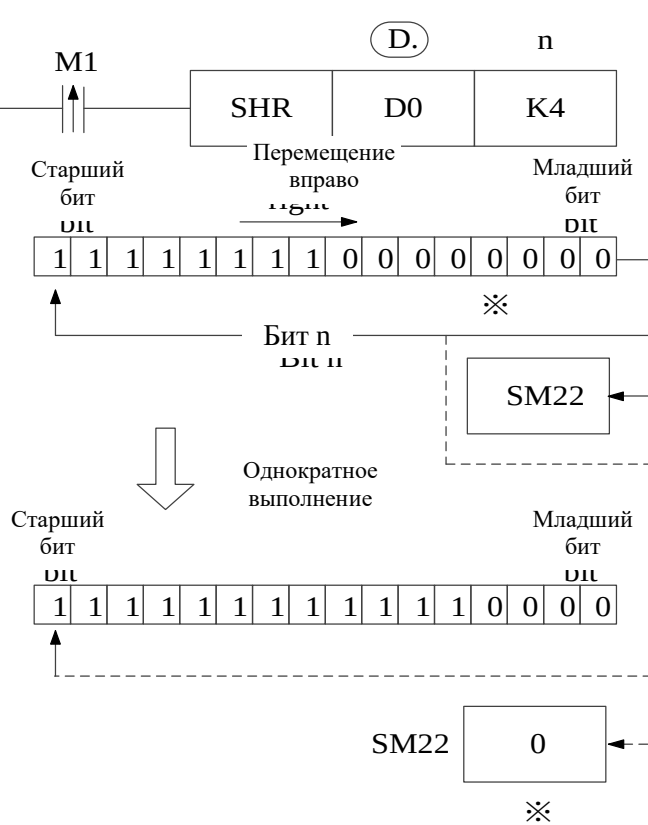
После однократного выполнения SHL младший бит заполняется 0, последний бит сохраняется во флаге переноса.

После однократного выполнения SHR старший бит остается без изменений; последний бит сохраняется во флаге переноса.

<Арифметический сдвиг влево>



<Арифметический сдвиг вправо>



4.7.2 Логический сдвиг влево [LSL], Логический сдвиг вправо [LSR]

1) Обзор

Не выполняйте логический сдвиг вправо/влево для данных.

Логический сдвиг влево [LSL, DLSL]			
16 бит	LSL	32 бита	DLSL
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Логический сдвиг вправо [LSR, DLSR]			
16 бит	LSR	32 бита	DLSR
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D	Адрес исходных данных	16 бит/32 бита, BIN
n	Количество арифметических сдвигов влево/вправо	16 бит/32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
D	•	•	•	•		•	•	•										
n									•									

4.7.3 Поворот со сдвигом влево [ROL, DROL], поворот со сдвигом вправо [ROR, DROR]

1) Обзор

Циклический сдвиг влево или вправо.

Поворот со сдвигом влево [ROL, DROL]			
16 бит	ROL	32 бита	DROL
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
Поворот со сдвигом вправо [ROR, DROR]			
16 бит	ROR	32 бита	DROR
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D	Адрес исходных данных	16 бит/32 бита, BIN
n	Количество сдвигов вправо или влево	16 бит/32 бита, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
D	•	•	•	•		•	•	•											
n									•										

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

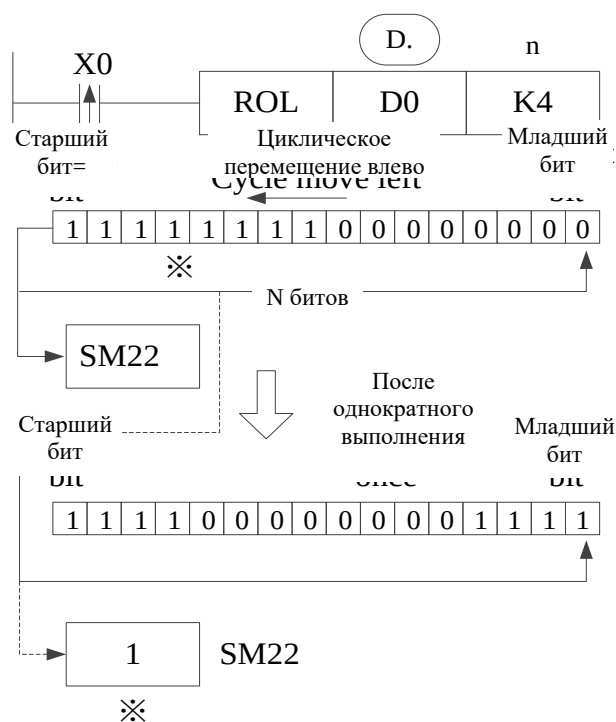
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

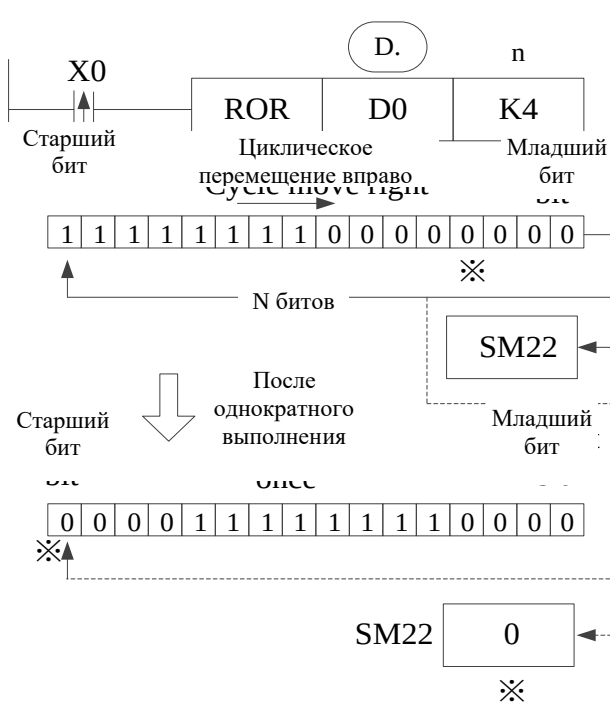
Описание

- Когда состояние X0 изменится с OFF (ВЫКЛ) на ON (ВКЛ), значение циклически переместится влево или вправо, а последний бит сохранится во флаге переноса.

<Циклический сдвиг влево>



<Циклический сдвиг вправо>



4.7.4 Битовый сдвиг влево [SFTL]

1) Обзор

Битовый сдвиг влево.

Битовый сдвиг влево [SFTL]			
16 бит	SFTL	32 бита	-
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Типы
S	Адрес заголовка исходного программного элемента	бит
D	Адрес заголовка целевого программного элемента	бит
n1	Объем исходных данных (не более 1024)	16 бит, BIN
n2	Количество сдвигов влево (не более 1024)	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S												•	•	•	•	•	•	
D													•	•	•	•	•	
n1	•	•	•	•	•	•	•	•	•									
n2	•	•	•	•	•	•	•	•	•									

*Примечание:

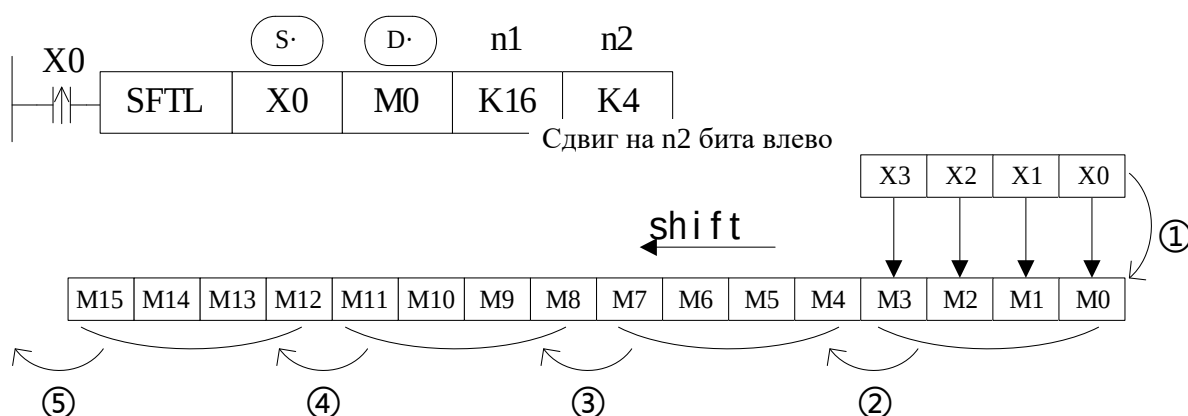
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

- Переместите n2 бита влево для объекта, который содержит n1 бит.
- Когда состояние X0 изменится с OFF на ON, команда переместит n2 бита для объекта.
- Например, если n2 равно K1, объект переместится на 1 бит влево, когда команда выполнится один раз.



- ① X3~X0 → M3~M0
- ② M3~M0 → M7~M4
- ③ M7~M4 → M11~M8
- ④ M11~M8 → M15~M12
- ⑤ M15~M12 → Переполнение

4.7.5 Битовый сдвиг вправо [SFTL]

1) Обзор

Битовый сдвиг вправо.

Битовый сдвиг вправо [SFTL]			
16 бит	SFTR	32 бита	-
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес заголовка исходного программного элемента	Бит
D	Адрес заголовка целевого программного элемента	Бит
n1	Объем исходных данных (не более 1024)	16 бит, BIN
n2	Количество сдвигов вправо (не более 1024)	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S												•	•	•	•	•	•	
D													•	•	•	•	•	
n1	•		•	•	•	•	•	•	•									
n2	•		•	•	•	•	•	•	•									

*Примечание:

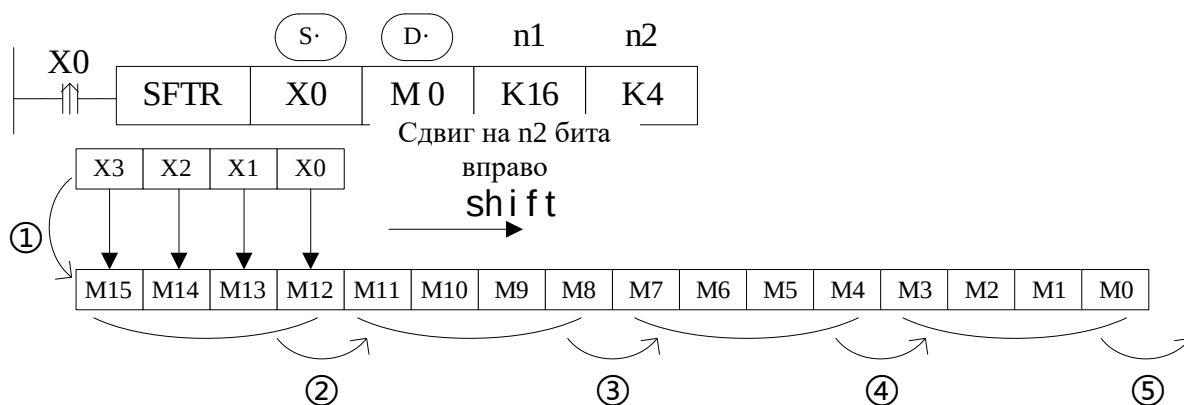
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

- Переместите n2 бита вправо для объекта, который содержит n1 бит.
- Когда состояние X0 изменится с OFF на ON, команда переместит n2 бита для объекта.
- Например, если n2 равно 1, объект переместится на 1 бит вправо, когда команда выполнится один раз.



- ① X3~X0 → M15~M12
- ② M15~M12 → M11~M8
- ③ M11~M8 → M7~M4
- ④ M7~M4 → M3~M0
- ⑤ M3~M0 → переполнение

4.7.6 Сдвиг слова влево [WSFL]

1) Обзор

Сдвиг слова влево.

Сдвиг слова влево [WSFL]			
16 бит	WSFL	32 бита	-
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес заголовка исходного программного элемента	16 бит, BIN
D	Адрес заголовка целевого программного элемента	16 бит, BIN
n1	Объем исходных данных (не более 512)	16 бит, BIN
n2	Количество сдвигов слова влево (не более 512)	16 бит, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•	•	•															
D	•	•	•	•															

n1	•	•	•	•	•	•	•	•	•									
n2	•	•	•	•	•	•	•	•	•									

*Примечание:

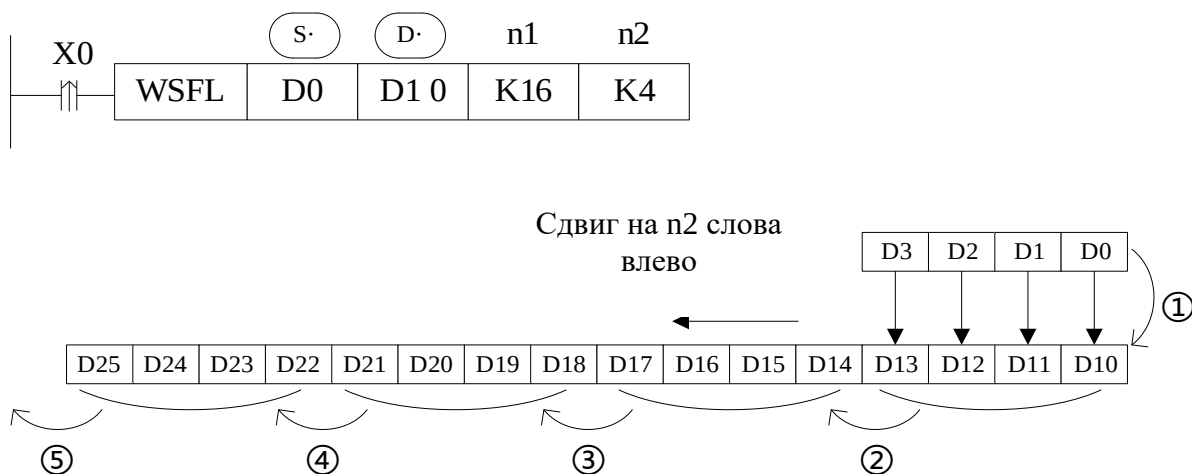
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

- Переместите n2 слова влево для объекта, который содержит n1 слов.
- Когда состояние X0 изменится с OFF на ON, команда переместит n2 слова для объекта.



- ① D3~D0 → D13~D10
- ② D13~D10 → D17~D14
- ③ D17~D14 → D21~D18
- ④ D21~D18 → D25~D22
- ⑤ D25~D22 → переполнение

4.7.7 Сдвиг слова вправо [WSFR]

1) Обзор

Сдвиг слова вправо.

Сдвиг слова вправо [WSFR]			
16 бит	WSFR	32 бита	-
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес заголовка исходного программного элемента	16 бит, BIN
D	Адрес заголовка целевого программного элемента	16 бит, BIN
n1	Объем исходных данных (не более 512)	16 бит, BIN
n2	Количество сдвигов вправо (не более 512)	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•														
D	•	•	•	•														

n1	•	•	•	•	•	•	•	•	•									
n2	•	•	•	•	•	•	•	•	•									

*Примечание:

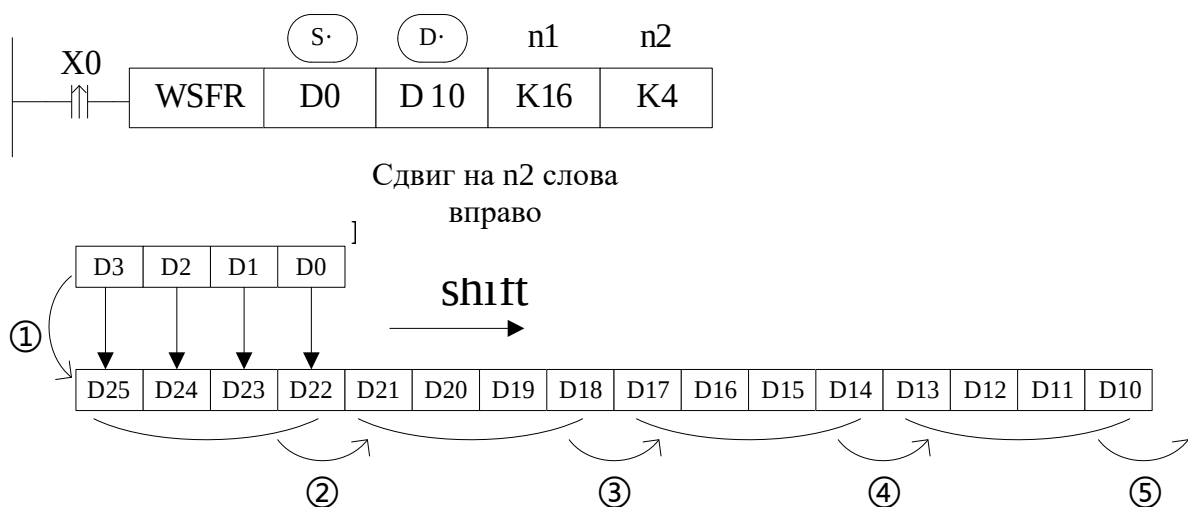
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

- Переместите n2 слова вправо для объекта, который содержит n1 слов.
- Когда состояние X0 изменится с OFF на ON, команда переместит n2 слова для объекта.



- ① D 3~D 0 → D25~D22
- ② D25~D22 → D21~D18
- ③ D21~D18 → D17~D14
- ④ D17~D14 → D13~D10
- ⑤ D13~D10 → переполнение

4.8 Преобразование данных

Мнемонический код	Функция	Глава
WTD	Преобразование целого числа из одного слова в целое число из двух слов	4-8-1
DWTD	Преобразование целого числа из двух слов в целое число из четырех слов	4-8-1
BDWTD	Пакетное преобразование 32-битных целых чисел в 64-битные целые числа	4-8-2
FLT	Преобразование 16-битного целого числа в число с плавающей точкой	4-8-3
DFLT	Преобразование 32-битного целого числа в число с плавающей точкой	4-8-3
FLTD	Преобразование 64-битного целого числа в число с плавающей точкой	4-8-3
DFLTD	Преобразование 32-битного целого числа в число с плавающей точкой двойной точности	4-8-4
QFLTD	Преобразование 64-битного целого числа в число с плавающей точкой двойной точности	4-8-4
INT	Преобразование числа с плавающей точкой в целое число	4-8-5
DINTD	Преобразование числа с плавающей точкой двойной точности в 32-битное целое число	4-8-6
QINTD	Преобразование числа с плавающей точкой двойной точности в 64-битное целое число	4-8-6
ECON	Преобразование числа с плавающей точкой одинарной точности в число с плавающей точкой двойной точности	4-8-7
BECON	Пакетное преобразование числа с плавающей точкой одинарной точности в число с плавающей точкой двойной точности	4-8-8

Мнемонический код	Функция	Глава
BIN	Преобразование BCD (двоично-кодированное десятичное число) в двоичное число	4-8-9
BCD	Преобразование двоичного числа в BCD	4-8-10
ASCII	Преобразование шестнадцатеричного числа в формат ASCII	4-8-11
HEX	Преобразование из формата ASCII в шестнадцатеричное число	4-8-12
DECO	Кодирование	4-8-13
ENCO	Кодирование с помощью старших битов	4-8-14
ENCOL	Кодирование с помощью младших битов	4-8-15
GRY	Преобразование двоичного числа в код Грея	4-8-16
GBIN	Преобразование кода Грея в двоичное число	4-8-17

4.8.1 Преобразование целого числа из одного слова в целое число из двух слов [WTD.DWTD]

1) Обзор

Преобразование целого числа из одного слова в целое число из двух слов [WTD]			
16 бит		WTD	
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	16 бит/32 бита, BIN
D	Адрес целевого программного элемента	32 бит/64 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										

*Примечание:

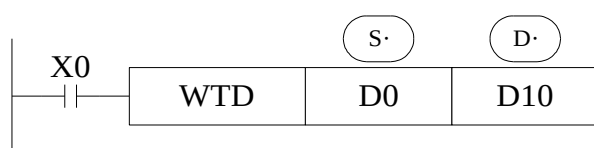
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

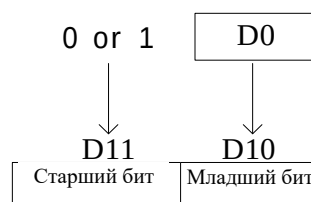
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<16-битная команда>

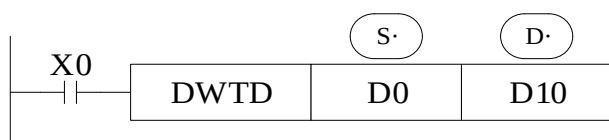


(D0) → (D11, D10)
Отдельное слово Двойное слово

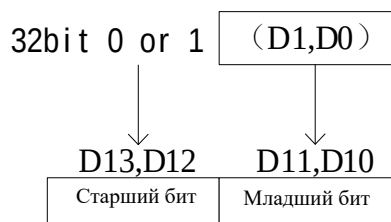


- Если отдельное слово D0 представляет собой положительное целое число, то после выполнения этой команды старший бит двойного слова D10 будет равен 0.
- Если отдельное слово D0 представляет собой отрицательное целое число, то после выполнения этой команды старший бит двойного слова D10 будет равен 1.
- Старший бит 0 и 1 является двоичным значением.

<32-битная команда>



(D1, D0) → (D13, D12, D11, D10)
 Двойное слово Четверное слово



- Если отдельное слово D0 представляет собой положительное целое число, то после выполнения этой команды старший бит четверного слова D10 будет равен 0.
- Если отдельное слово D0 представляет собой отрицательное целое число, то после выполнения этой команды старший бит четверного слова D10 будет равен 1.
- Старший бит 0 и 1 является двоичным значением.

4.8.2 Преобразование целого числа в число с плавающей точкой [FLT, DFLT, FLTD]

1) Обзор

Преобразование битного целого числа в число с плавающей точкой [FLT, DFLT, FLTD]					
16 бит	FLT	32 бита	DFLT	64 бита	FLTD
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт		Совместимые модели	PMP20	
Требования к аппаратному обеспечению	-		Требования к программному обеспечению	-	

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	16 бит/32 бита/64 бита, BIN
D	Адрес целевого программного элемента	32 бит/64 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•																
D	•																	

*Примечание:

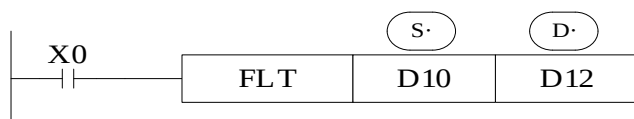
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

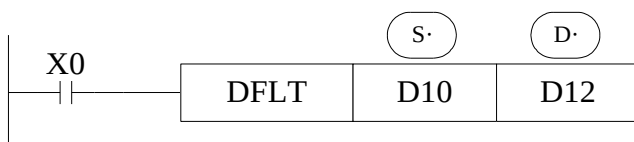
Описание

<16-битная команда>



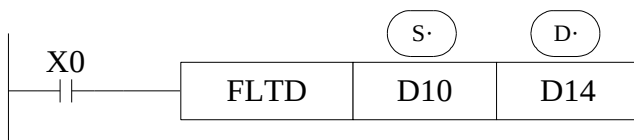
(D10 → (D13, D12)
Целое число (BIN) Двоичное число
с плавающей точкой

<32-битная команда>



(D11, D10) → (D13, D12)
Целое число (BIN) Двоичное число
с плавающей точкой

<64-битная команда>



(D11, D10) → (D13, D12)
Целое число (BIN) Двоичное число
с плавающей точкой

- Преобразуйте целое число (BIN) в двоичное число с плавающей точкой. Как константа K, H будет автоматически преобразовано с помощью команды операции с плавающей точкой, поэтому эту команду FLT использовать нельзя.
- Команда обратного преобразования — INT.
- Команда FLTD может изменить 64-битное целое число на 32-битное число с плавающей точкой.
- Операнд S команды FLTD не поддерживает константу K/H.



D0 — целое число 20, а после выполнения этой команды D10 — число с плавающей точкой 20.

Примечание:

Перед использованием команд операций с числами с плавающей точкой, например, команд EADD, ESUB, EMUL, EDIV, EMOV и ECMP, убедитесь, что все параметры работы являются числами с плавающей точкой.

4.8.3 Преобразование числа с плавающей точкой в целое число [INT, DINT]

1) Обзор

Преобразование числа с плавающей точкой в целое число [INT, DINT]			
16 бит	INT	32 бита	DINT
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	16 бит/32 бита, BIN
D	Адрес целевого программного элемента	16 бит/32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•																
D	•																	

*Примечание:

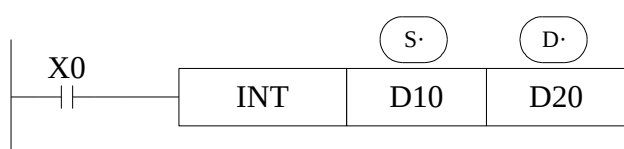
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

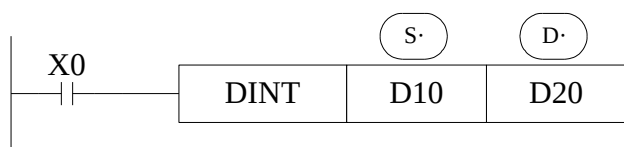
Описание

<16-битная команда>



$(D11, D10) \rightarrow (D20)$
Двоичное число с плавающей точкой Целое число (BIN)
Отбросим данные после десятичной точки

<32-битная команда>

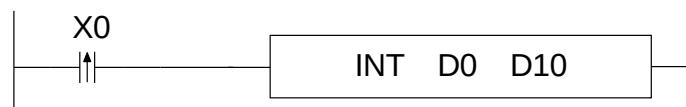


$(D11, D10) \rightarrow (D20, D21)$
Двоичное число с плавающей точкой Целое число (BIN)
Отбросим данные после десятичной точки

- Двоичное исходное число преобразуется в целое число (BIN) и сохраняется на целевом устройстве. Отбросим значение после десятичной точки.
- Команда обратного преобразования — FLT.
- Когда результат равен 0, бит признака будет иметь состояние ON.
- Результат больше указанных ниже данных, флаг переноса имеет состояние ON.

16-битная операция: -32 768~32 767

32-битная операция: -2 147 483 648~2 147 483 647



Например, если D0 является числом с плавающей точкой 130.2, то после выполнения команды INT значение D10 будет целым числом 130.

4.8.4 Преобразование BCD (двоично-кодированное десятичное число) в двоичное число [BIN]

1) Обзор

Преобразование BCD (двоично-кодированное десятичное число) в двоичное число [BIN]			
16 бит	BIN	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	BCD
D	Адрес целевого программного элемента	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										

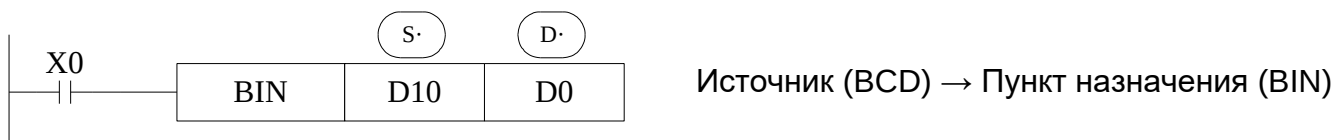
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

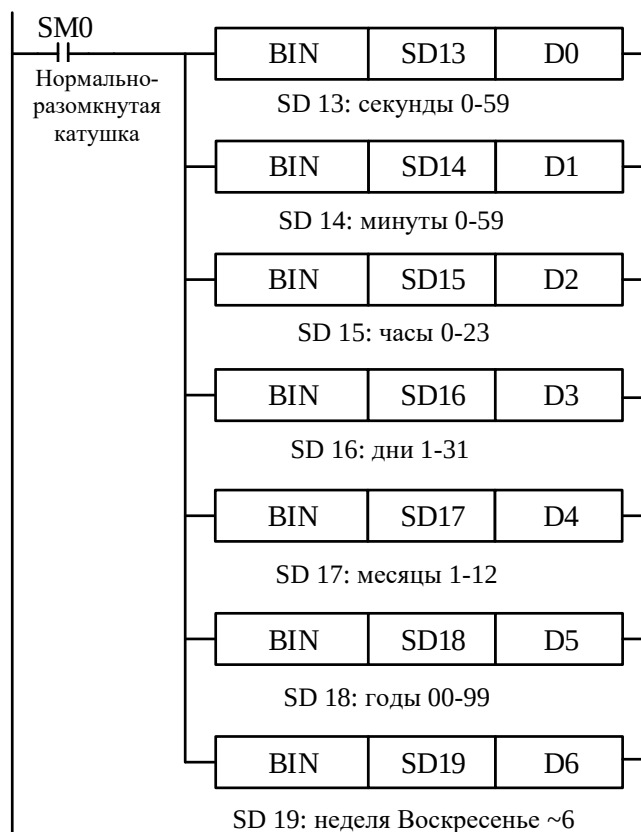
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Если исходные данные не являются кодом BCD, SM409 будет включен (ошибка операции), SD409 = 4 (возникает ошибка).
- Поскольку константа K автоматически преобразуется в двоичное число, она не подходит для этой команды.
- Например: вся информация, хранящаяся в регистре информации о времени SD13~ SD19 ПЛК, имеет код BCD, но мы привыкли использовать десятичное значение. Информацию о времени можно преобразовать из кода BCD в двоичное число:



4.8.5 Преобразование двоичного числа в BCD [BCD]

1) Обзор

Преобразование двоичных данных в код BCD.

Преобразование двоичного числа в BCD [BCD]			
16 бит	BCD	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	16 бит, BIN
D	Адрес целевого программного элемента	Код BCD

3) Соответствующие программные компоненты

Опе- ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон- станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•	•	•	•	•	•	•											
D	•		•	•		•	•	•											

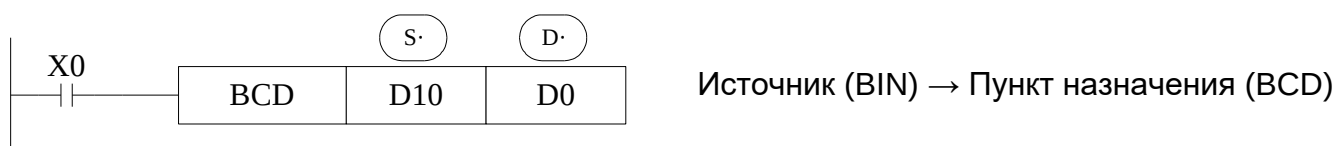
***Примечание:**

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

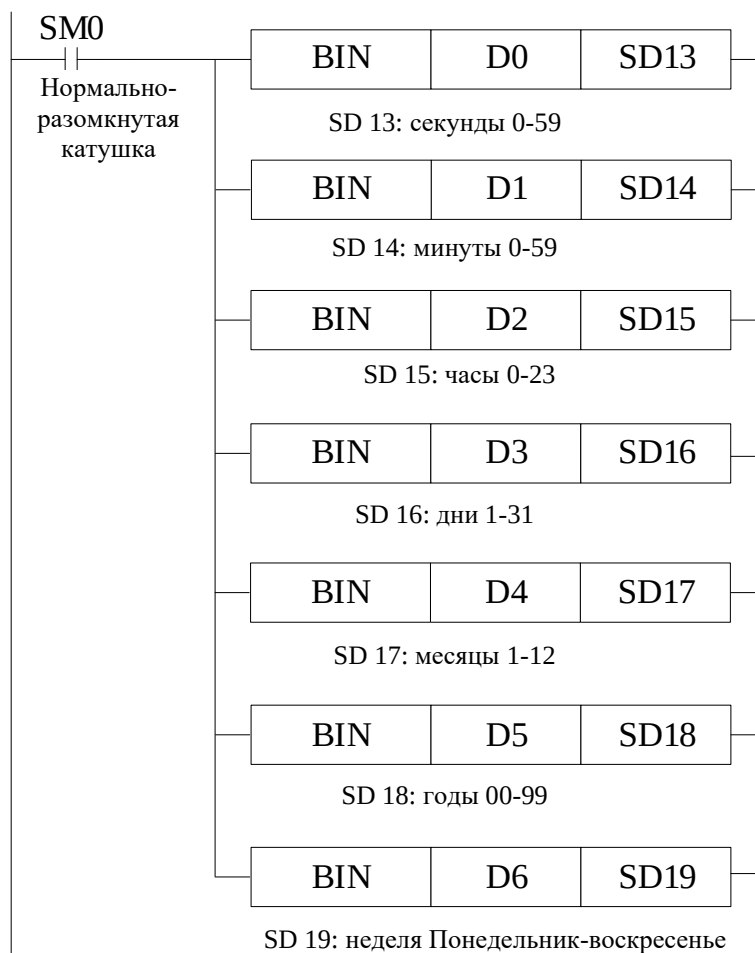
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Эта команда может изменить двоичное значение на код BCD.
- Например, регистры информации о времени SD13–SD19 ПЛК хранят код BCD, но мы привыкли использовать десятичные значения, поэтому можно использовать команду BCD для исправления информации о времени в регистрах SD13–SD19.



4.8.6 Преобразование шестнадцатеричного числа в формат ASCII [ASCII]

1) Обзор

Преобразование шестнадцатеричного числа в формат ASCII [ASCII]			
16 бит	ASCII	32 бит	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	2 бита, HEX
D	Адрес целевого программного элемента	Код ASCII
N	Преобразование количества символов	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										
n	•		•	•		•	•	•	•									

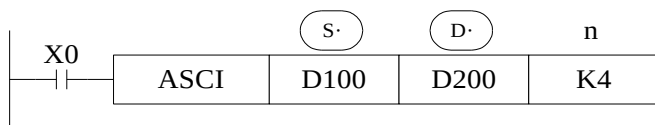
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- D преобразует исходные шестнадцатеричные данные в код ASCII и сохраняет его в S.

Символами преобразования являются n.

- D будет хранить один код ASCII.

Процесс преобразования выглядит следующим образом:

Назначьте устройство запуска:

(D100) = 0ABCH

(D101) = 1234H

(D102) = 5678H

[0] = 30H

[5] = 35H

[2] = 32H

[B] = 42H

[7] = 37H

[4] = 34H

[1] = 31H

[A] = 41H

[6] = 36H

[3] = 33H

[C] = 43H

[8] = 38H

n D	K1	K2	K3	K4	K5	K6	K7	K8	K9			
D200, младш.	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]			
D200, старш.		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]			
D201, младш.			[C]	[B]	[A]	[0]	[4]	[3]	[2]			
D201, старш.				[C]	[B]	[A]	[0]	[4]	[3]			
D202, младш.					[C]	[B]	[A]	[0]	[4]			
D202, старш.						[C]	[B]	[A]	[0]			
D203, младш.							[C]	[B]	[A]			
D203, старш.								[C]	[B]			
D204, младш.									[C]			



4.8.7 Преобразование из формата ASCII в шестнадцатеричное число [HEX]

1) Обзор

Преобразование из формата ASCII в шестнадцатеричное число [HEX]			
16 бит	HEX	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходного программного элемента	ASCII
D	Адрес целевого программного элемента	2 бита, HEX
n	Количество символов ASCII	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D	•		•	•		•	•	•										
n									•									

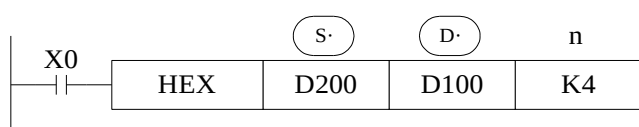
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Преобразование старших 8 бит и младших 8 бит в исходном **S** в данные типа HEX. Каждый раз 4 бита перемещаются в пункт назначения **D**.
- Количество символов преобразования назначается через n.

Процесс преобразования выглядит следующим образом:

(S)	ASCII код	HEX преобразование
D200, младш.	30H	0
D200, старш.	41H	A
D201, младш.	42H	B
D201, старш.	43H	C
D202, младш.	31H	1
D202, старш.	32H	2
D203, младш.	33H	3
D203, старш.	34H	4
D204, младш.	35H	5

n = k4

(D·) n	D102	D101	D100
1	При отсутствии изменений равно 0		...0H
2			..0AH
3			·0ABH
4			0ABCH
5		...0H	ABC1H
6		..0AH	BC12H
7		·0ABH	C123H
8		0ABCH	1234H
9	...0H	ABC1H	2345H

D200

0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

	41H?	[A]		30H?	[0]	
--	------	-----	--	------	-----	--

D201

0	1	0	0	0	0	1	1	0	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

	43H?	[C]		42H?	[B]	
--	------	-----	--	------	-----	--

D100

0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

	0		A		B		C	
--	---	--	---	--	---	--	---	--

4.8.8 Кодирование [DECO]

1) Обзор

Изменение любых данных или бита на 1.

Кодирование [DECO]			
16 бит	DECO	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес исходных данных	16 бит, BIN
D	Адрес заголовка результатов декодирования	16 бит, BIN
N	Количество битов в программном элементе декодирования	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•										
D												•	•	•	•	•		
n									•									

*Примечание:

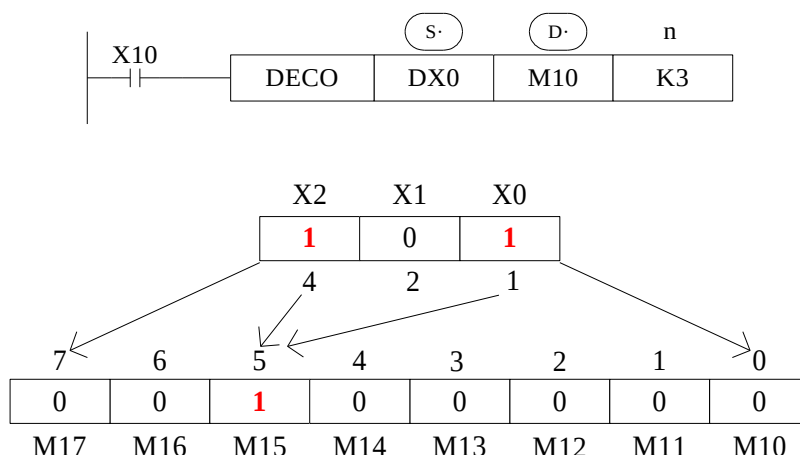
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

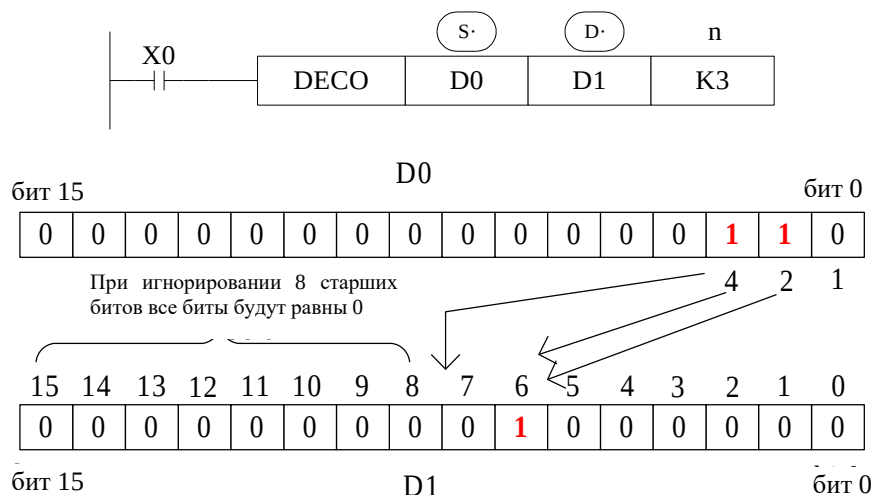
Описание

<Когда **D** является двоичной единицей > $n \leq 16$



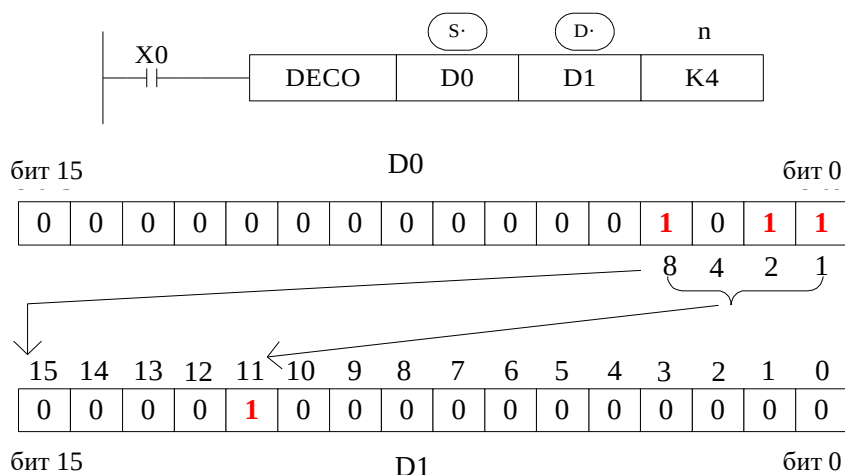
- $N = 3$, поэтому объектом декодирования являются младшие три бита в DX0, то есть $X2 \sim X0$.
- $n = 3$, поэтому результаты декодирования необходимо выразить с помощью $2^3 = 8$ бит, то есть, $M17 \sim M10$.
- Если $X2 = 1$, $X1 = 0$, $X0 = 1$, то представляемое им значение равно $4 + 1 = 5$, поэтому M15 на пятом месте от M10 изменяется на 1; если все $X2 \sim X0$ равны нулю, значение равно 0, то M10 равно 1 (M10 находится на 0-м месте).
- Если $n = 0$, команда не будет выполнена. Если n является значением вне диапазона 0 - 16, команда не будет выполнена.
- При $n = 16$, если команда декодирования представляет собой битовый программный компонент, то количество точек равно $2^{16} = 65536$.
- Если вход привода выключен, команда не будет выполняться, а декодирование выходного сигнала продолжится.

<Когда D_• является словным операндом > $n \leq 4$



- Младший n -бит ($n \leq 4$) адреса источника декодируется в целевой адрес. Если $n \leq 3$, 8 старших битов целевого адреса принимают значение 0.
- Если $n = 0$, команда не будет выполнена. Если n выходит за пределы диапазона 0 - 4, команда не будет выполнена.
- $N = 3$, поэтому объект декодирования в D0 — это бит2-бит0, а максимальное значение, которое он представляет, равно $4 + 2 + 1 = 7$.
- $N = 3$, поэтому в D1 для представления результата декодирования необходимо $2^3 = 8$ бит, то есть бит7 ~ бит0.
- Когда бит2 и бит1 оба равны 1, а бит0 равен 0, значение равно $4 + 2 = 6$, поэтому бит6 в D1 включен.

<D_• является программным компонентом слова> $n \leq 4$



- Младший n -бит ($n \leq 4$) адреса источника декодируется в целевой адрес. Если $n \leq 3$, 8 старших битов целевого адреса принимают значение 0.

- Если $n = 0$, команда не будет выполнена. Если n выходит за пределы диапазона 0 - 4, команда не будет выполнена.
- $N = 4$, поэтому объектом декодирования в D0 является бит3 ~ бит0, что представляет собой максимальное значение $8 + 4 + 2 + 1 = 15$.
- $N = 4$, поэтому в D1 для представления результата декодирования необходимо $2^4 = 16$ бит, то есть бит15 ~ бит0.
- Когда бит3, бит1 и бит0 равны 1, а бит2 равен 0, числовое значение равно $8 + 2 + 1 = 11$, поэтому бит11 в D1 включен.

4.8.9 Кодирование с помощью старших битов [ENCO]

1) Обзор

Определение старшего бита, который равен 1.

Кодирование с помощью старших битов [ENCO]			
16 бит	ENCO	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес данных кодирования	16 бит, BIN
D	Адрес результата кодирования	16 бит, BIN
N	Количество битов в результате кодирования	16 бит, BIN

3) Соответствующие программные компоненты

Опе- ранды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль			Система					
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•				•	•	•	•	•	•	
D	•		•	•		•	•	•										
n									•									

*Примечание:

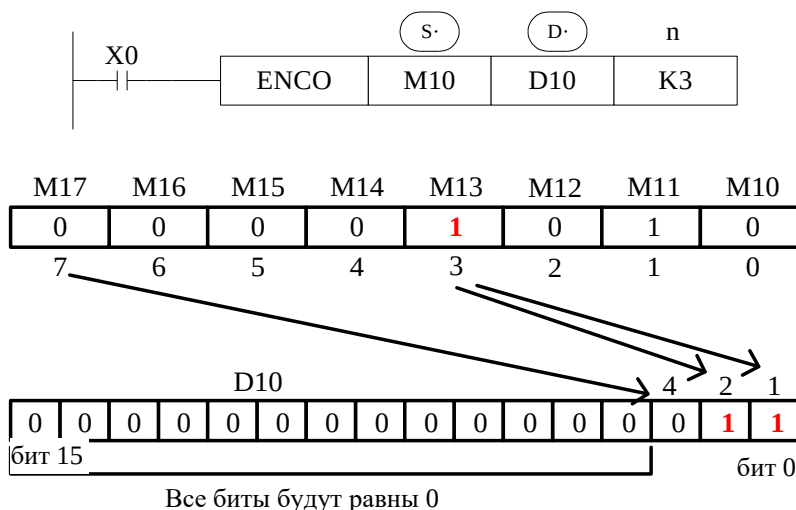
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<Когда **S** является битовым операндом > $n \leq 16$

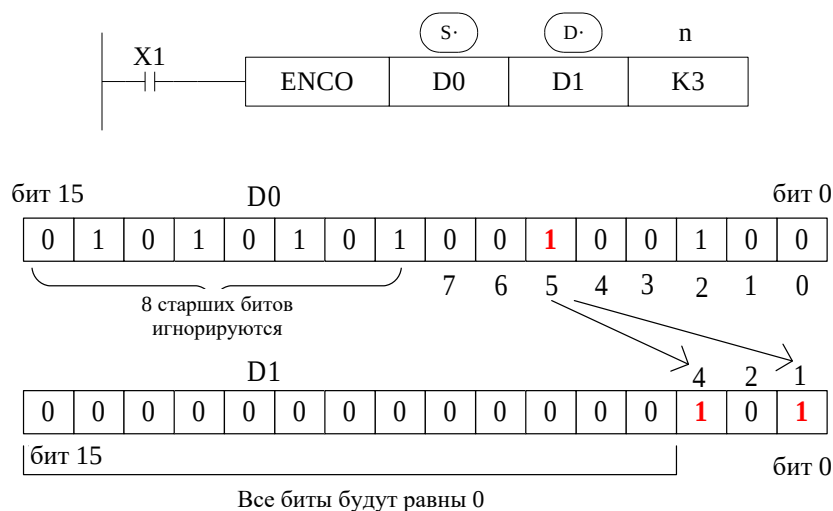


Игнорирование значения 1 для M11

- Если несколько битов в адресе источника равны 1, то младший бит игнорируется, а если адрес источника равен 0, то команда не будет выполнена.
- Если привод выключен, команда не выполняется, и выходной код остается неизменным.

- При $n = 16$, если команда кодирования является битовым элементом, ее номер точки равен $2^{16} = 65536$.
- При $N = 3$, закодированный объект имеет $2^3 = 8$ бит, которые равны M17 ~ M10, а результаты кодирования сохраняются в трех младших битах в D10, т.е. бит2 ~ бит0.
- M13 и M11 равны 1. Если M11 игнорируется, M13 кодируется, бит2-бит0 равны 3, а бит0 и бит1 — 1.

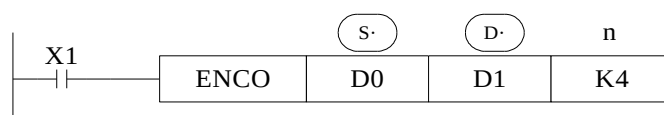
<Когда **S** является словным операндом> $n \leq 4$

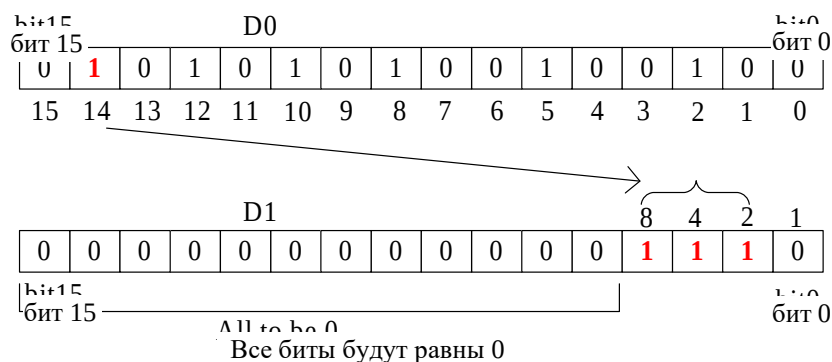


Игнорируется значение 1 бита 2

- Если несколько битов в адресе источника равны 1, то младший бит игнорируется, а если адрес источника равен 0, то команда не будет выполнена.
- Когда вход привода выключен, команда не выполняется, и выходной код не изменяется.
- Если $n \leq 3$, 8 старших битов в D0 игнорируются.
- При $n = 3$ объект кодирования имеет $2^3 = 8$ бит, то есть бит7 ~ бит0 в D0. Результат кодирования сохраняется в младших 3 битах в D1, то есть бит2 ~ бит0.
- Если бит5 и бит2 в D0 оба равны 1, бит2 игнорируется, а бит5 кодируется, бит2-бит0 равны 5, бит2 и бит0 равны 1.

<**S** является программным компонентом слова> $n \leq 4$





Значение 1 игнорируется в бит2, бит5, бит8, бит10, бит12.

- Если несколько битов в адресе источника равны 1, то младший бит игнорируется, а если адрес источника равен 0, то команда не будет выполнена.
- Когда вход привода выключен, команда не выполняется, и выходной код не изменяется.
- При $n = 4$, закодированный объект имеет $2^4 = 16$ бит, то есть бит15 ~ бит0 в D0. Результат кодирования сохраняется в младших 4 битах в D1, то есть бит3 ~ бит0.
- Старший бит со значением 1 в D0 — это бит14, при этом все младшие биты со значением 1 игнорируются, и бит14 кодируется, бит3-бит0 равны 14, бит3, бит2 и бит1 равны 1.

4.8.10 Кодирование с помощью младших битов [ENCOL]

1) Обзор

Определение положения, в котором младший бит включен.

Кодирование с помощью младших битов [ENCOL]			
16 бит	ENCOL	32 бит	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Требуется кодирование адреса программного элемента	16 бит, BIN
D	Адрес программного элемента для сохранения результата кодирования	16 бит, BIN
N	Количество битов в результате кодирования	16 бит, BIN

3) Соответствующие программные компоненты

[illegible]

*Примечание:

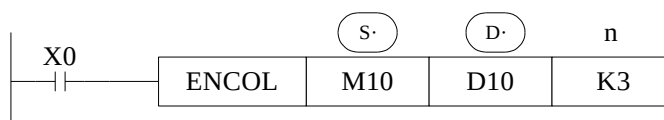
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

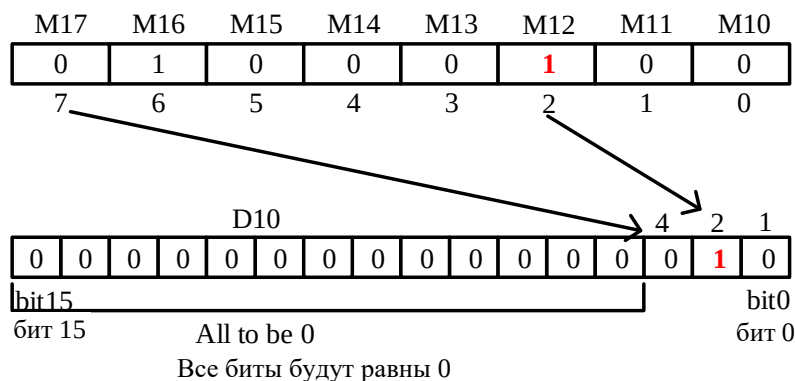
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<Когда **S** является битовым операндом> $n \leq 16$

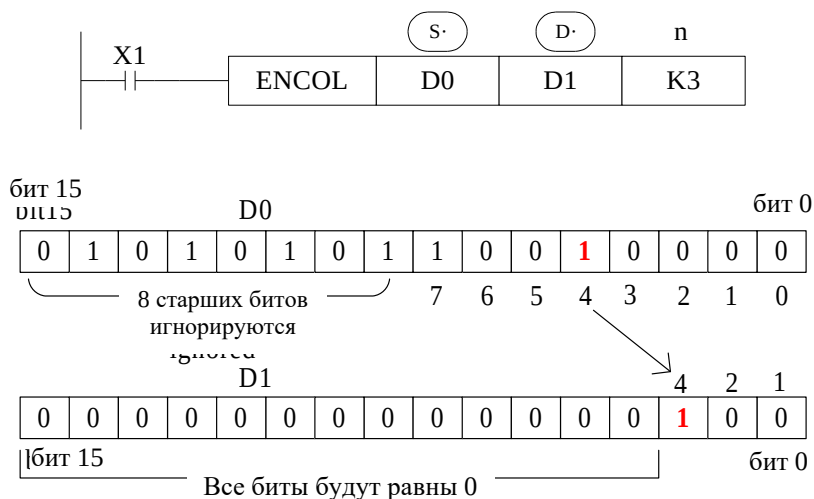




Игнорирование значения 1 для M16

- Если несколько битов в адресе источника равны 1, то старший бит игнорируется, а если адрес источника равен 0, то команда не будет выполнена.
- Если привод выключен, команда не выполняется, и выходной код остается неизменным.
- При $n = 16$, если **S** команды кодирования является битовым элементом, ее номер точки равен $216 = 65536$.
- При $n = 3$, закодированный объект имеет $2^3 = 8$ бит, которые равны M17 ~ M10, а результаты кодирования сохраняются в трех младших битах в D10, т.е. бит2 ~ бит0.
- M12 и M16 равны 1. Если M16 игнорируется, M12 кодируется, бит2-бит0 равны 2, а бит1 — 1.

<Когда **S** является словным операндом> $n \leq 4$

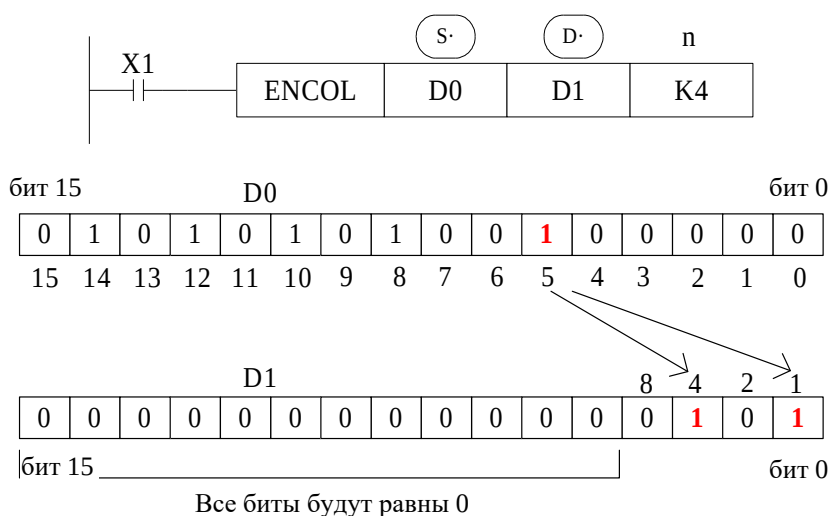


Игнорирование значения 1 для b7

- Если несколько битов в адресе источника равны 1, то старший бит игнорируется, а если адрес источника равен 0, то команда не будет выполнена.

- Когда вход привода выключен, команда не выполняется, и выходной код не изменяется.
- Если $n \leq 3$, 8 старших битов в D0 игнорируются.
- Объект кодирования имеет $2^3 = 8$ бит, то есть бит7 ~ бит0 в D0. Результат кодирования сохраняется в младших 3 битах в D1, то есть бит2 ~ бит0.
- Если бит7 и бит4 в D0 оба равны 1, бит7 игнорируется, а бит4 кодируется. Бит 2 равен 1, когда бит2-бит0 выражены в виде 4.

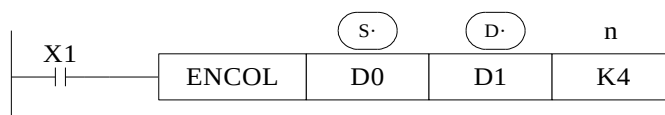
<S является программным компонентом слова> $n \leq 4$



Значение 1 игнорируется в бит14, бит12, бит10, бит8.

- Если несколько битов в адресе источника равны 1, то старший бит игнорируется, а если адрес источника равен 0, то команда не будет выполнена.
- Когда вход привода выключен, команда не выполняется, и выходной код не изменяется.
- При $n = 4$, закодированный объект имеет $2^4 = 16$ бит, то есть бит15 ~ бит0 в D0. Результат кодирования сохраняется в младших 4 битах в D1, то есть бит3 ~ бит0.
- Младший бит со значением 1 в D0 — это бит5, при этом все старшие биты со значением 1 игнорируются, и бит5 кодируется, бит3-бит0 равны 5, бит2 и бит0 равны 1.

<S является программным компонентом слова> $n \leq 4$



2) Операнды

Операнды	Функция	Тип данных
S	Требуется кодирование адреса программного элемента	16 бит/32 бита, BIN
D	Адрес программного элемента для сохранения результата кодирования	16 бит/32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова										Программные элементы бита							
	Система								Кон- станта	Модуль	Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

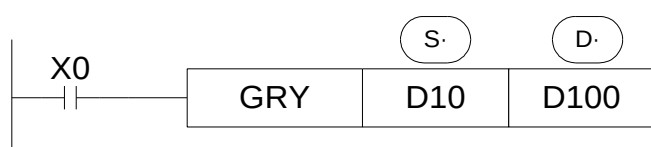
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

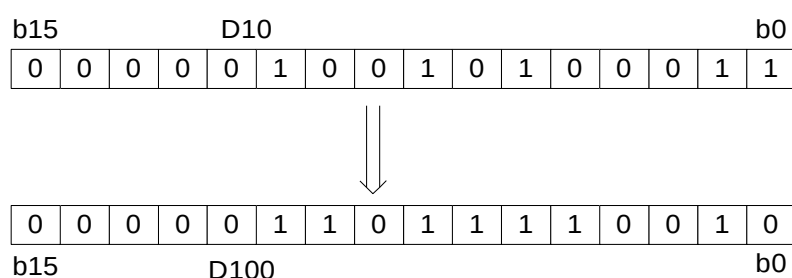
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Источник (BIN) → Пункт назначения (GRY)



Каждый бит в D10 будет выполнять операцию XOR (исключающее ИЛИ) с битом слева. Как и в соответствующем коде Грея, левый бит не изменится (левый бит равен 0); результат преобразования сохраняется в D100.

- Преобразование двоичного значения в код Грея.

- GRY имеет 32-битный режим DGRY, который может преобразовывать 32-битный код Грея.
- Диапазон **S** · составляет 0~32 767 (16-битная команда); 0~2 147 483 647 (32-битная команда).

4.8.12 Преобразование кода Грея в двоичное число [GBIN, DGBIN]

1) Обзор

Преобразование кода Грея в двоичные данные.

Преобразование кода Грея в двоичное число [GBIN, DGBIN]			
16 бит	GBIN	32 бита	DGBIN
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Требуется кодирование адреса программного элемента	16 бит/32 бита, BIN
D	Адрес программного элемента для сохранения результата кодирования	16 бит/32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•	•	•	•	•	•	•	•									
D	•		•	•		•	•	•										

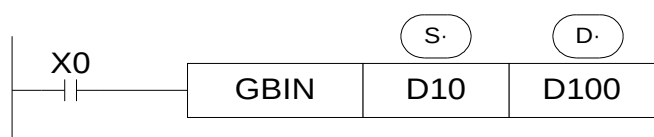
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

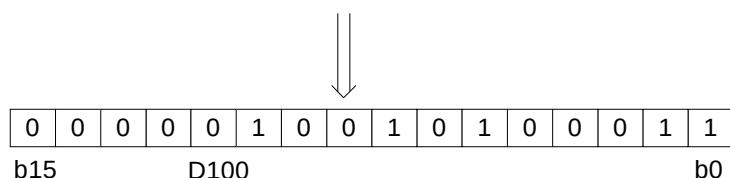
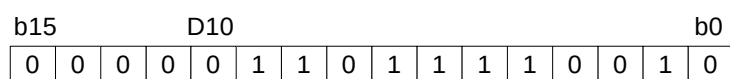
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Источник (GRY) → Пункт назначения (BIN)



Начиная со второго левого бита в D10, выполните операцию XOR для каждого бита со значением после декодирования, как со значением бита после декодирования (левый бит не изменится). Значение преобразования сохранится в D100.

- Преобразование кода Грея в двоичное значение.
- GBIN имеет 32-битный режим DBIN, который может преобразовывать 32-битное двоичное значение.



- Диапазон **S** · · составляет 0~32 767 (16-битная команда); 0~2 147 483 647 (32-битная команда).

4.9 Операции с числами с плавающей точкой

Мнемонический код	Функция	Глава
ECMP	Сравнение чисел с плавающей точкой	4-9-1
EZCP	Сравнение зон чисел с плавающей точкой	4-9-2
EADD	Сложение чисел с плавающей точкой	4-9-3
ESUB	Вычитание чисел с плавающей точкой	4-9-4
EMUL	Умножение чисел с плавающей точкой	4-9-5
EDIV	Деление чисел с плавающей точкой	4-9-6
ESQR	Квадратный корень чисел с плавающей точкой	4-9-7
SIN	Синус	4-9-8
COS	Косинус	4-9-9
TAN	Тангенс	4-9-10
ASIN	ASIN	4-9-11
ACOS	ACOS	4-9-12
ATAN	ATAN	4-9-13

4.9.1 Сравнение чисел с плавающей точкой [ECMP]

1) Обзор

Сравнение чисел с плавающей точкой [ECMP]			
16 бит	-	32 бита	ECMP
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Требуется сравнение адресов программных элементов	32/64 бита, BIN
S2	Требуется сравнение адресов программных элементов	32/64 бита, BIN
D	Результат сравнения	бит

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S1	•	•			•	•	•	•	•										

S2	•	•			•	•	•	•	•									
D												•	•	•				

*Примечание:

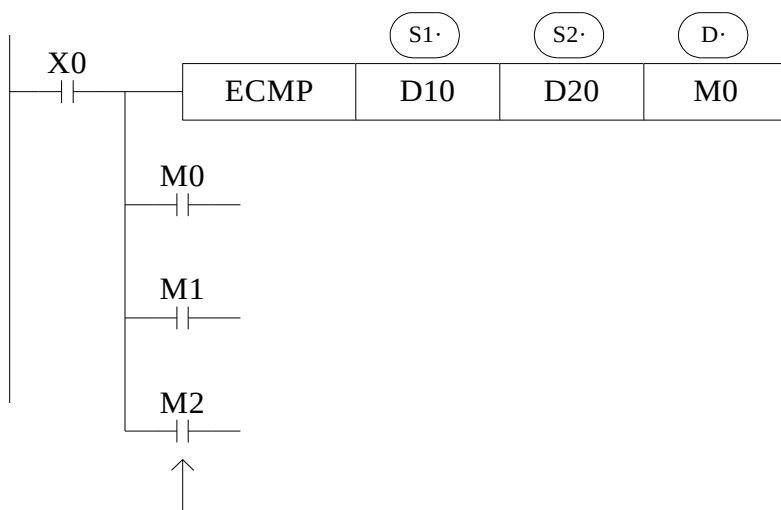
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<32-битная операция>



Когда X0 выключен, даже если ECMP не выполняется, M0~M2 сохраняют состояние, которое было до выключения X0.

(D11, D10) : (D21, D20) → M0, M1, M2

Двоичное число с
плавающей точкой

Двоичное число с
плавающей точкой

(D11, D10) > (D21 < D20)

Двоичное число с
плавающей точкой

Двоичное число с
плавающей точкой

(D11, D10) = (D21 < D20)

Двоичное число с
плавающей точкой

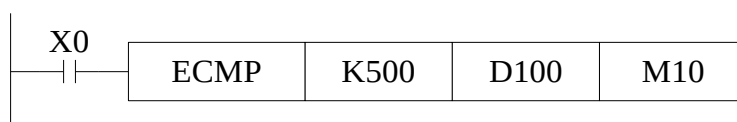
Двоичное число с
плавающей точкой

$(D11, D10) < (D21 < D20)$

Двоичное число с
плавающей точкой

Двоичное число с
плавающей точкой

- Эта команда сравнит два набора исходных данных S1 и S2. Результат сохранится в трех битах из D.
- Перед выполнением команды все данные сравнения должны быть числами с плавающей точкой (если это целое число, его можно преобразовать с помощью команд FLT); в противном случае результат выполнения будет неверным.
- Если в качестве исходных данных используется константа K или H, то значение преобразуется в число с плавающей точкой.

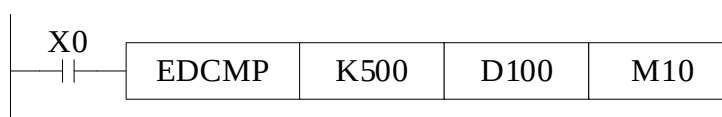


$(K500) : (D101, D100) \rightarrow M10, M11, M12$

Преобразование двоичного числа в число с плавающей точкой

Преобразование двоичного числа с плавающей точкой

<64-битная операция>



$(K500) : (D103, D102, D101, D100) \rightarrow M10, M11, M12$

Преобразование двоичного числа в число с плавающей точкой

Преобразование двоичного числа с плавающей точкой

4.9.2 Сравнение зон чисел с плавающей точкой [EZCP]

1) Обзор

Сравнение зон чисел с плавающей точкой [EZCP]			
16 бит	-	32 бита	EZCP
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Требуется сравнение адресов программных элементов	32 бита, BIN
S2	Верхний предел сравнения данных	32 бита, BIN
S3	Нижний предел сравнения данных	32 бита, BIN
D	Адрес программного элемента результата сравнения	бит

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
S3	•	•			•	•	•	•	•									
D													•	•	•			

*Примечание:

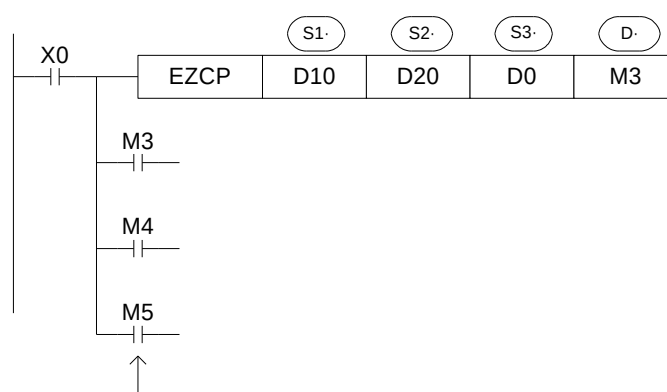
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

Сравнение исходных данных с диапазоном.



Когда X0 выключен, даже если EZCP не выполняется, M3~M5 сохраняют состояние, которое было до выключения X0.

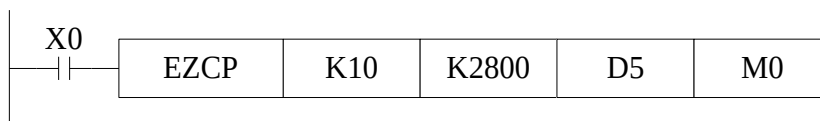
(D1, D0)	<	(D11, D10)	ON
Двоичное число с плавающей точкой		Двоичное число с плавающей точкой	

(D11, D10)	≤	(D1, D0)	≤	(D21, D20)	ON
Двоичное число с плавающей точкой		Двоичное число с плавающей точкой		Двоичное число с плавающей точкой	

(D1, D0)	>	(D21, D20)	ON
Двоичное число с плавающей точкой		Двоичное число с плавающей точкой	

- Сравните исходные данные S3 с верхним и нижним предельными значениями диапазона S1~S2.
- Результат сохранится в трех регистрах из D.

- Константы К и Н преобразуются в двоичные значения с плавающей точкой, если они являются исходными данными.



(K10) : [D6, D5] : (K2800) → M0, M1, M2

Преобразование
двоичного числа в
число с плавающей
точкой

Двоичное число с плаваю-
щей точкой

Преобразование двоичного числа в
число с плавающей точкой

Установите $S1 \leq S2$, если $S2 < S1$, установите S2 равным значению S1.

Примечание: значение для сравнения должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

4.9.3 Сложение чисел с плавающей точкой [EADD]

1) Обзор

Сложение чисел с плавающей точкой [EADD]			
16 бит	-	32 бита	EADD
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операции сложения	32/64 бита, BIN
S2	Адрес данных операции сложения	32/64 бита, BIN
D	Адрес результата	32/64 бита, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S1	•	•			•	•	•	•	•										
S2	•	•			•	•	•	•	•										
D	•					•	•	•											

*Примечание:

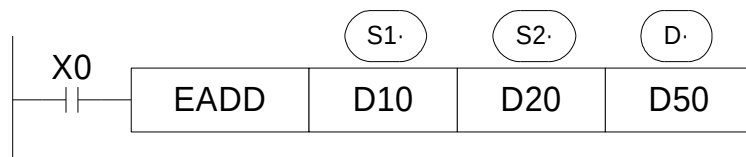
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

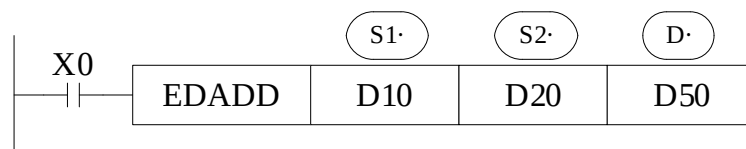
Описание

<32-битная операция>



$$(D11, D10) + (D21, D20) \rightarrow (D51, D50)$$

<64-битная операция>



$$(D13, D12, D11, D10) + (D23, D22, D21, D20) \rightarrow (D53, D52, D51, D50)$$

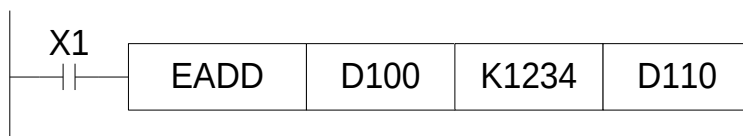
Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- Два набора исходных данных в виде двоичных чисел с плавающей точкой выполняют операцию сложения, результат будет сохранен по целевому адресу.
- Если в качестве исходных данных используется константа К или Н, то перед операцией сложения значение преобразуется в число с плавающей точкой.
- Регистры в EDADD должны начинаться с четного адреса.

<32-битная операция>



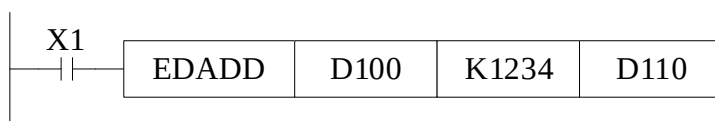
(K1234) + (D101, D100) → (D111, D110)

Преобразование двоичного
числа в число с плавающей
точкой

Двоичное число с плаваю-
щей точкой

Двоичное число с плаваю-
щей точкой

<64-битная операция>



(K1234) + (D103, D102, D101, D100) → (D113, D112, D111, D110)

Преобразование двоичного
числа в число с плавающей
точкой

Двоичное число с плаваю-
щей точкой

Двоичное число с плаваю-
щей точкой

- Исходные данные и адрес результата могут быть одинаковыми. Обратите внимание, что если X0 включен, команда будет выполняться в каждом периоде сканирования.

Примечание: значение для сложения должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

4.9.4 Вычитание чисел с плавающей точкой [ESUB]

1) Обзор

Вычитание чисел с плавающей точкой [ESUB]			
16 бит	-	32 бита	ESUB
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операции вычитания	32/64 бита, BIN
S2	Адрес данных операции вычитания	32/64 бита, BIN
D	Адрес результата	32/64 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
D	•					•	•	•										

*Примечание:

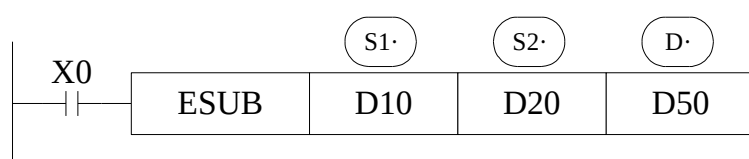
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<32-битная операция>



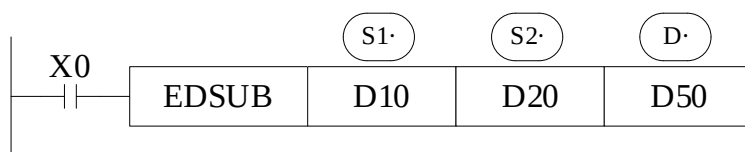
(D11, D10) — (D21, D20) → (D51, D50)

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

<64-битная операция>



(D13, D12, D11, D10) — (D23, D22, D21, D20) → (D53, D52, D51, D50)

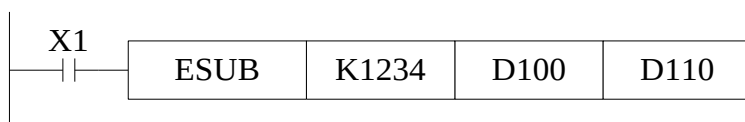
Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- Из значения двоичного числа с плавающей точкой S1 вычитается S2, результат будет сохранен по целевому адресу.
- Если в качестве исходных данных используется константа K или H, то перед операцией вычитания значение преобразуется в число с плавающей точкой.

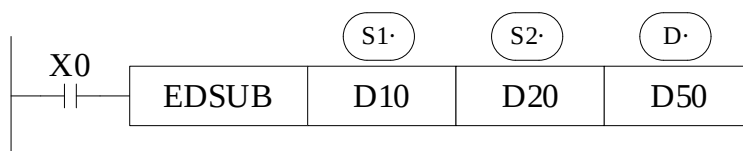
<32-битная операция>



(K1234) — (D101, D100) → (D111, D110)

Преобразование двоичного числа в число с плавающей точкой Двоичное число с плавающей точкой Двоичное число с плавающей точкой

<64-битная операция>



(D13, D12, D11, D10) — (D23, D22, D21, D20) → (D53, D52, D51, D50)

Преобразование двоичного числа в число с плавающей точкой Двоичное число с плавающей точкой Двоичное число

- Исходные данные и адрес результата могут быть одинаковыми. Обратите внимание, что если X0 включен, команда будет выполняться в каждом периоде сканирования.
- Примечание: значение операнда должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

4.9.5 Умножение чисел с плавающей точкой [EMUL]

1) Обзор

Умножение чисел с плавающей точкой [EMUL]			
16 бит	-	32 бита	EMUL
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операции умножения	32/64 бита, BIN
S2	Адрес данных операции умножения	32/64 бита, BIN
D	Адрес результата	32/64 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
D	•					•	•	•										

*Примечание:

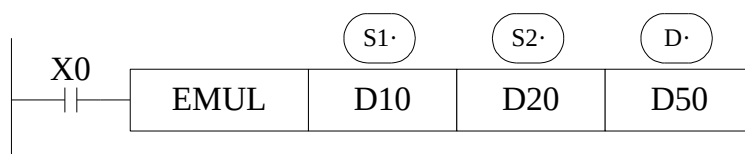
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

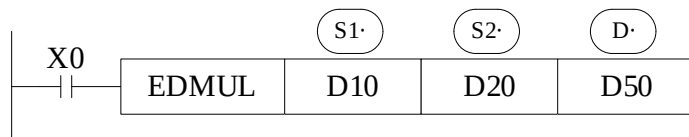
Описание

<32-битная операция>



$(D11, D10) \times (D21, D20) \rightarrow (D51, D50)$
 Двоичное число с плавающей точкой Двоичное число с плавающей точкой Двоичное число с плавающей точкой

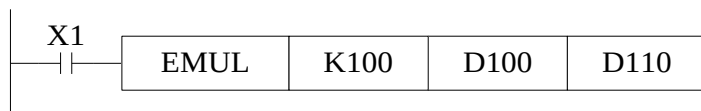
<64-битная операция>



$(D13, D12, D11, D10) - (D23, D22, D21, D20) \rightarrow (D53, D52, D51, D50)$
 Преобразование двоичного числа в число с плавающей точкой Двоичное число с плавающей точкой Двоичное число

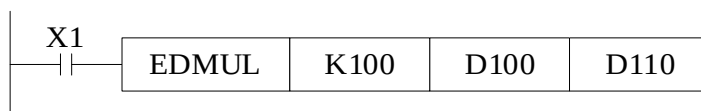
- Значение числа с плавающей точкой S1 умножается на значение числа с плавающей точкой S2. Результат умножения сохраняется в D как число с плавающей точкой.
- Если в качестве исходных данных используется константа K или H, то перед операцией умножения значение преобразуется в число с плавающей точкой.
- Регистры в EDMUL должны начинаться с четного адреса.

<32-битная операция>



$(K100) \times (D101, D100) \rightarrow (D111, D110)$
 Преобразование двоичного числа в число с плавающей точкой Двоичное число с плавающей точкой Двоичное число

<64-битная операция>



$(K00) \times (D103, D102, D101, D100) \rightarrow (D113, D112, D111, D110)$
 Преобразование двоичного числа в число с плавающей точкой Двоичное число с плавающей точкой Двоичное число

Примечание: значение операнда должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

4.9.6 Деление чисел с плавающей точкой [EDIV, EDDIV]

1) Обзор

Деление чисел с плавающей точкой [EDIV, EDDIV]			
16 бит	-	32 бита	EDIV
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
64 бита	EDDIV		
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме/ спадающий фронт/нарастающий фронт	Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме/ спадающий фронт/нарастающий фронт
Требования к аппаратному обеспечению	Версия 3.7.1 или более поздняя	Требования к программному обеспечению	Версия 3.7.4a или более поздняя

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес данных операции деления	32/64 бита, BIN
S2	Адрес данных операции деления	32/64 бита, BIN
D	Адрес результата	32/64 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•			•	•	•	•	•									
S2	•	•			•	•	•	•	•									
D	•					•	•	•										

*Примечание:

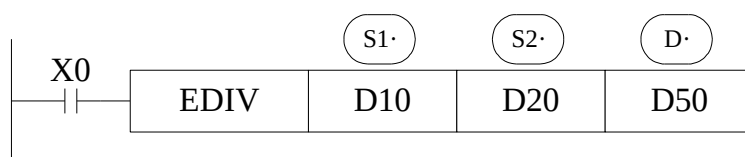
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

<32-битная операция>



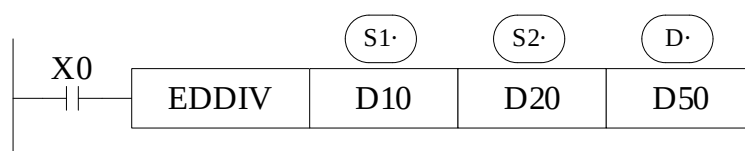
$(D11, D10) \div (D21, D20) \rightarrow (D51, D50)$

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число

<64-битная операция>



$$(D13, D12, D11, D10) \div (D23, D22, D21, D20) \rightarrow (D53, D52, D51, D50)$$

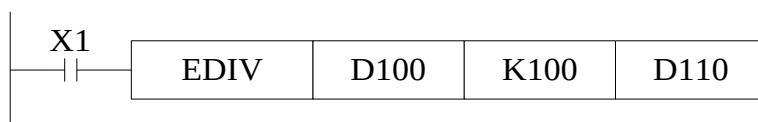
Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число

- Значение числа с плавающей точкой S1 делится на значение числа с плавающей точкой S2. Результат деления сохраняется в D как число с плавающей точкой.
- Если в качестве исходных данных используется константа K или H, то перед операцией деления значение преобразуется в число с плавающей точкой.
- Если исходные данные S2 равны 0, то при расчете появится ошибка. Команда не будет выполняться.
- Значение операнда должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

<32-битная операция>



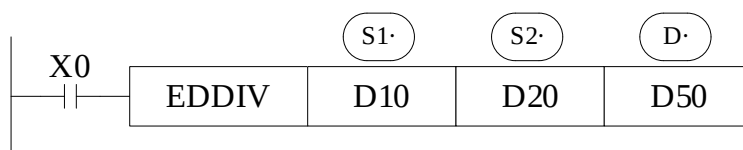
$$(D101, D100) \div (K100) \rightarrow (D111, D110)$$

Преобразование двоичного числа в число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

<64-битная операция>



$$(D13, D12, D11, D10) \div (D23, D22, D21, D20) \rightarrow (D53, D52, D51, D50)$$

Двоичное число с плавающей точкой

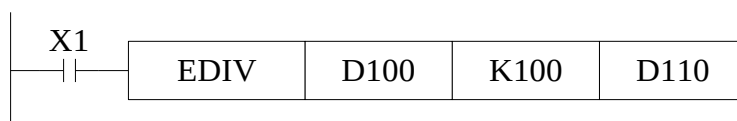
Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- Значение числа с плавающей точкой S1 делится на значение числа с плавающей точкой S2. Результат деления сохраняется в D как число с плавающей точкой.

- Если в качестве исходных данных используется константа К или Н, то перед операцией деления значение преобразуется в число с плавающей точкой.
- Если исходные данные S2 равны 0, то при расчете появится ошибка. Команда не будет выполняться.
- Значение операнда должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

<32-битная операция>



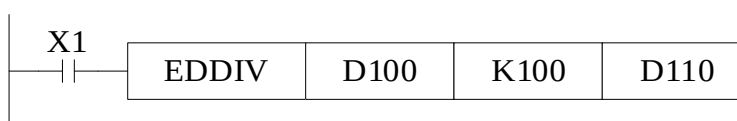
$$(D101, D100) \div (K100) \rightarrow (D111, D110)$$

Преобразование двоичного числа в число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

<64-битная операция>



$$(D103, D102, D101, D100) \div (K100) \rightarrow (D113, D112, D111, D110)$$

Преобразование двоичного числа в число с плавающей точкой

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

4.9.7 Квадратный корень чисел с плавающей точкой [ESQR]

1) Обзор

Квадратный корень чисел с плавающей точкой [ESQR]			
16 бит	-	32 бита	ESQR
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20

Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
--------------------------------------	---	---------------------------------------	---

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента нужно возвести в квадратный корень.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

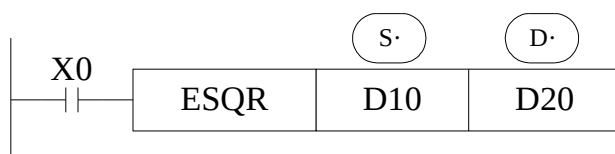
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

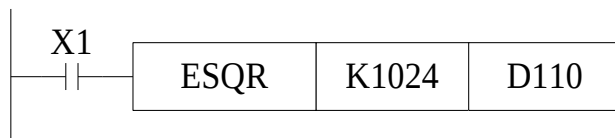


(D11, D10) → (D21, D20)

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- в квадратный корень; результат сохраняется в D.
- Если в качестве исходных данных используется константа K или H, то перед операцией значение преобразуется в число с плавающей точкой.



(K1024) → (D111, D110)

Преобразование двоичного числа в число с плавающей точкой

Двоичное число с плавающей точкой

- Если результат равен нулю, включается признак нуля.
- Только при положительных исходных данных операция будет выполнена. Если S — отрицательное число, то возникнет ошибка, и признак ошибки SM409 примет состояние ON, SD409 = 7; команда не может быть выполнена.
- Значение операнда должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

4.9.8 Синус [SIN]

1) Обзор

Синус числа с плавающей точкой [SIN]			
16 бит	-	32 бита	SIN
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента должен быть синусоидальным.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

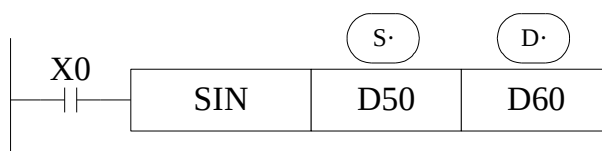
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

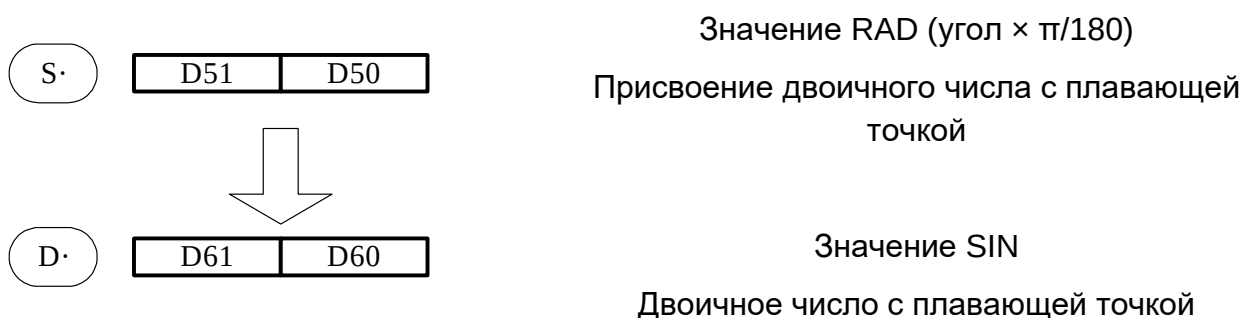


(D51, D50) RAD → (D61, D60) SIN

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- Эта команда выполняет математическую операцию SIN над значением с плавающей точкой в S (угол RAD). Результат сохраняется в D.



Примечание: значение операнда должно быть числом с плавающей точкой, в противном случае результатом будет ошибка.

4.9.9 Косинус [COS]

1) Обзор

Косинус числа с плавающей точкой [COS]			
16 бит	-	32 бита	COS
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента должен быть косинусоидальным.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Опе- ранды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

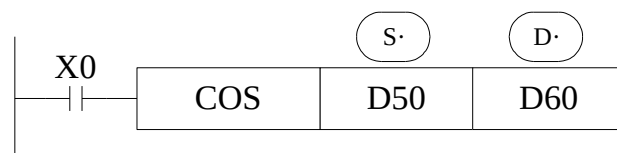
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



(D51, D50) RAD → (D61, D60) COS

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- Эта команда выполняет математическую операцию COS над значением с плавающей точкой в S (угол RAD). Результат сохраняется в D.



Примечание:

Перед выполнением команды данные в параметре S должны быть числом с плавающей точкой; в противном случае результат выполнения будет неверным.

4.9.10 TAN [TAN]

1) Обзор

TAN [TAN]			
16 бит	-	32 бита	TAN
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента должен быть тангенциальным.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

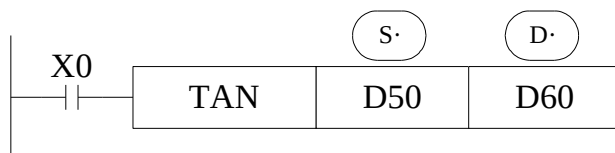
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



(D51, D50) RAD → (D61, D60) TAN

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

- Эта команда выполняет математическую операцию TAN над значением с плавающей точкой в S. Результат сохраняется в D.



Примечание:

Перед выполнением команды данные в параметре S должны быть числом с плавающей точкой; в противном случае результат выполнения будет неверным.

4.9.11 ASIN [ASIN]

1) Обзор

ASIN [ASIN]			
16 бит	-	32 бит	ASIN
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению		Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента должен быть арксинусоидальным.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Опе- ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон- станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•			•	•	•	•	•										
D	•					•	•	•											

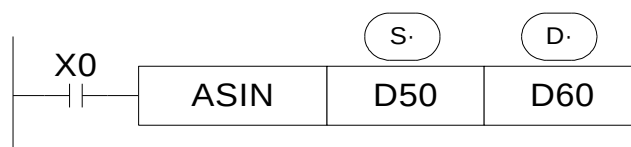
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

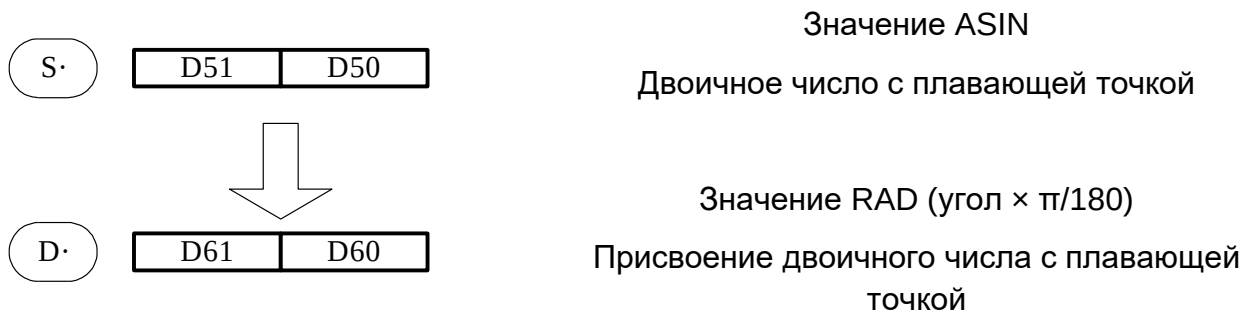


(D51, D50) ASIN → (D61, D60) RAD

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Эта команда выполняет математическую операцию ASIN над значением с плавающей точкой в S. Результат сохраняется в D.



Примечание:

Перед выполнением команды данные в параметре S должны быть числом с плавающей точкой; в противном случае результат выполнения будет неверным.

4.9.12 ACOS [ACOS]

1) Обзор

ACOS [ACOS]			
16 бит	-	32 бита	ACOS
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению		Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента должен быть арккосинусоидальным.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•	•			•	•	•	•	•										
D	•					•	•	•											

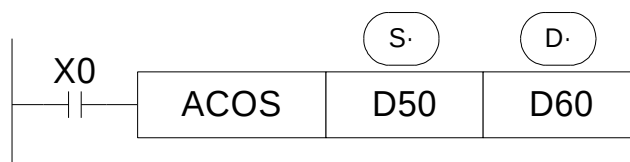
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

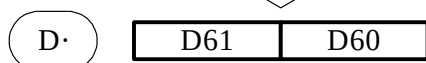
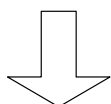
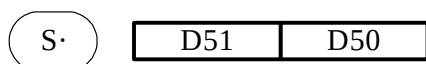


(D51, D50) ACOS → (D61, D60) RAD

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Рассчитайте значение аркосинуса (радиан), сохраните результат по целевому адресу.



Значение ACOS

Двоичное число с плавающей точкой

Значение RAD (угол × π/180)

Присвоение двоичного числа с плавающей точкой

Примечание:

Перед выполнением команды данные в параметре S должны быть числом с плавающей точкой; в противном случае результат выполнения будет неверным.

4.9.13 ATAN [ATAN]

1) Обзор

ATAN [ATAN]			
16 бит	-	32 бита	ACOS
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению		Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного элемента должен быть арктангенциальным.	32 бита, BIN
D	Адрес результата	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S	•	•			•	•	•	•	•									
D	•					•	•	•										

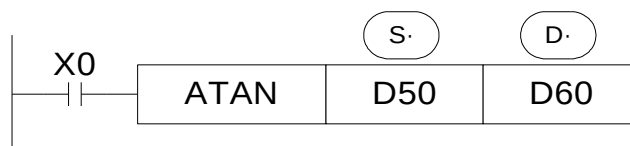
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

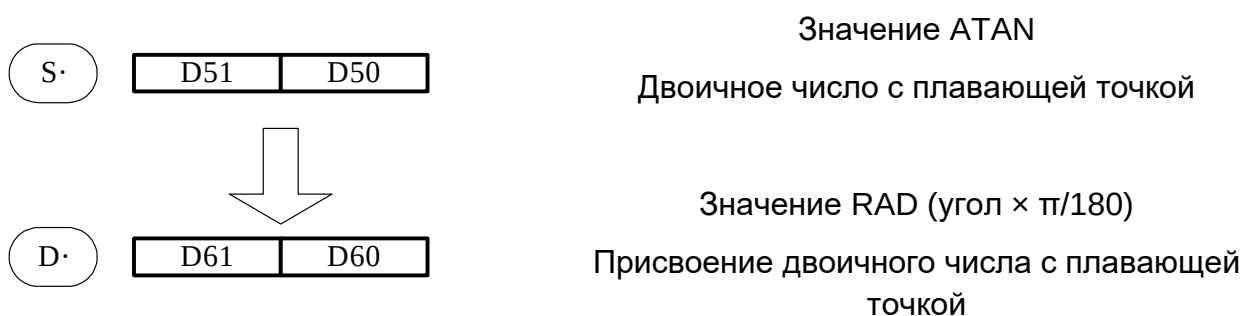


(D51, D50) ATAN → (D61, D60) RAD

Двоичное число с плавающей точкой

Двоичное число с плавающей точкой

Рассчитайте значение арктангенса (радиан), сохраните результат по целевому адресу.



Примечание:

Перед выполнением команды данные в параметре S должны быть числом с плавающей точкой; в противном случае результат выполнения будет неверным.

4.10 Команды часов реального времени

Мнемонический код	Функция	Глава
TRD	Считывание данных часов	4-10-1
TWR	Запись данных часов	4-10-2
MOV	Считывание данных платы BD точных часов	4-10-3
TO	Запись данных платы BD точных часов	4-10-4
TADD	Сложение данных часов	4-10-5

TSUB	Вычитание данных часов	4-10-6
HTOS	Преобразование часов, минут и секунд в секунды	4-10-7
STON	Преобразование секунд в часы, минуты и секунды	4-10-8
TCMP	Сравнение времени (часы, минуты, секунды)	4-10-9
DACMP	Сравнение даты (год, месяц, день)	4-10-10

※1 Для использования этих команд модель должна поддерживать функцию RTC.

Часы ПЛК серии PMP20 имеют некоторую погрешность, составляющую около ± 5 ми-

※2 нут в месяц. Часы можно регулярно калибровать с помощью ЧМИ или в программе ПЛК.

Если требуется обеспечить высокую точность определения времени, PMP-RTC-BD

※3 можно использовать совместно, при этом погрешность составит около 13 секунд в месяц.

4.10.1 Считывание данных часов [TRD]

1) Обзор

Считывание данных часов.

Считывание данных часов: [TRD]			
16 бит	TRD	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
D	Адрес регистра для сохранения данных часов	16 бит, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
D	•		•	•														

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Текущее время и дата часов реального времени считываются и сохраняются в 7 устройствах данных, указанных в адресе заголовка D.

- Считывание показаний часов реального времени ПЛК в указанном ниже формате.

Считывание данных из специального регистра данных (SD013~SD019).

Специаль-ный регистр данных	Блок	Наименование	Данные часов	→	Элемент	Наименование
	SD018	Год	0-99		D0	Год
	SD017	Месяц	1-12		D1	Месяц
	SD016	Дата	1-31		D2	Дата

	SD015	Час	0-23	→	D3	Час
	SD014	Минута	0-59	→	D4	Минута
	SD013	Секунда	0-59	→	D5	Секунда
	SD019	Неделя	0 (воскресенье) - 6 (суббота)	→	D6	Неделя

- Значение RTC (часов реального времени) имеет формат кода BCD (от SD013 до SD019).
- После считывания показаний RTC по команде TRD значение будет отображаться в десятичном формате.
- После считывания показаний RTC по команде TRD значение становится десятичным.
- После выполнения команды TRD D0 - D6 будут заняты.

4.10.2 Запись данных часов [TWR]

1) Обзор

Запись данных часов.

Запись данных часов [TWR]			
16 бит	-	32 бита	TWR
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению		Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип данных
S	Запись данных часов в регистр	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
D	•		•	•	•	•	•	•										

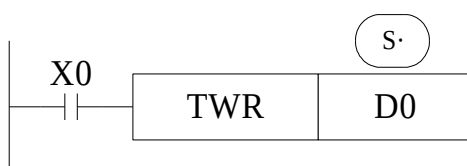
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Запись значения RTC в ПЛК.

- Запись заданного времени в часы реального времени ПЛК.
- Чтобы записать данные в часы реального времени, установите значение 7 регистров в диапазоне от D0 до D6.

Данные для настройки часов	Блок	Наименование	Данные часов	→	Блок	Наименование
	D0	Год	0-99		SD018	Год
	D1	Месяц	1-12		SD017	Месяц
	D2	Дата	1-31		SD016	Дата
	D3	Час	0-23		SD015	Час
	D4	Минута	0-59		SD014	Минута
	D5	Секунда	0-59		SD013	Секунда

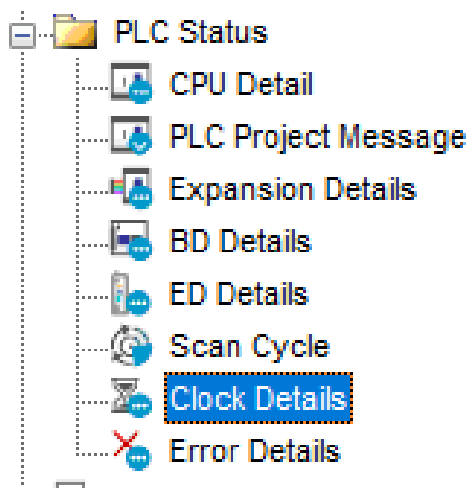
	D6	Неделя	0 (воскресенье) - 6 (суббота)	→	SD019	Неделя
--	----	--------	-------------------------------	---	-------	--------

После выполнения команды TWR время в часах реального времени немедленно изменится на новое время. Рекомендуется установить время на несколько минут позже текущего времени, а затем запустить данную команду, когда реальное время достигнет этого значения.

Примечание: при выборе режима секретной загрузки программы в программном обеспечении PROMPOWER PLC Studio настройки часов реального времени (RTC) можно изменять только с помощью команды TWR.

Есть и другой метод записи данных в RTC.

В программном обеспечении PROMPOWER PLC Studio выберите опцию Clock details (Сведения о часах) на панели проекта слева.



Затем нажмите Write into the current time (Запись текущего времени). ПК автоматически запишет текущее время в ПЛК.



Затем нажмите Write into the current time (Запись текущего времени), и ПК автоматически запишет текущее время в ПЛК.

4.10.3 Считывание данных платы BD точных часов [MOV]

1) Обзор

Считывание данных платы BD точных часов [MOV]			
16 бит	MOV	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя

2) Операнды

Операнды	Функция	Тип данных
S	Адрес программного компонента считываемых данных часов	16 бит, BIN
D	Адрес регистра для сохранения данных часов	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•																	
D	•																	

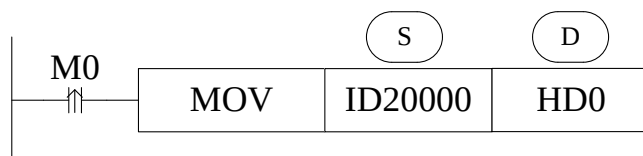
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



При включении M0 «секунды» в PMP-RTC-BD часов №1 считывается в регистр HD0 ПЛК.

- Адрес данных платы BD PMP-RTC-BD показан ниже.

Адрес платы BD №1	Адрес платы BD №2	Описание	Данные часов	Примечание
ID20000	ID20100	Секунда	0~59	Десятичная
ID20001	ID20101	Минута	0~59	Десятичная
ID20002	ID20102	Час	0~23	Десятичная
ID20003	ID20103	Дата	1~31	Десятичная
ID20004	ID20104	Месяц	1~12	Десятичная
ID20005	ID20105	Год	00~99	Десятичная
ID20006	ID20106	Неделя	0 (воскресенье) - 6 (суббота)	Десятичная

- Поскольку время в регистре ID хранится в следующем порядке: секунды, минуты, часы, дни, месяцы, годы и недели, не рекомендуется использовать команды BMOV или PMOV для считывания данных часов пакетами, если считанные данные часов используются для сравнения и расчета.

4.10.4 Запись данных платы BD точных часов [ТО]

1) Обзор

Запись данных платы BD точных часов [ТО]			
16 бит	ТО	32 бита	-
Условие выполнения	нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя

2) Операнды

Операнды	Функция	Тип данных
S1	Номер платы BD	16 бит, BIN
S2	Номер адреса заголовка программного компонента для данных часов	16 бит, BIN
S3	Количество данных часов	16 бит, BIN
D	Адрес заголовка программного компонента для данных локальных часов	16 бит, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m	
S1									•										
S2										•									
S3									•										
D	•																		

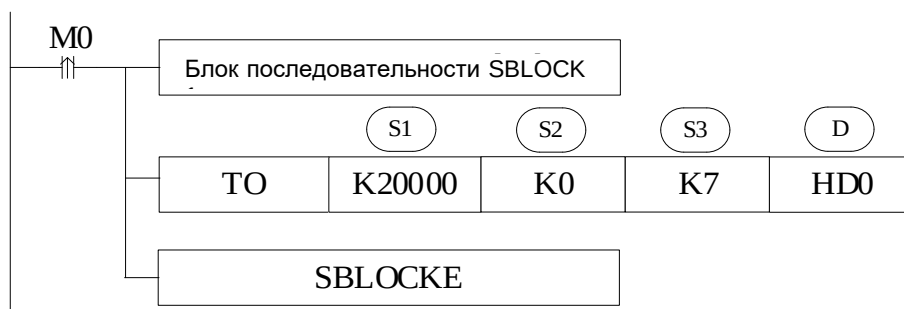
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

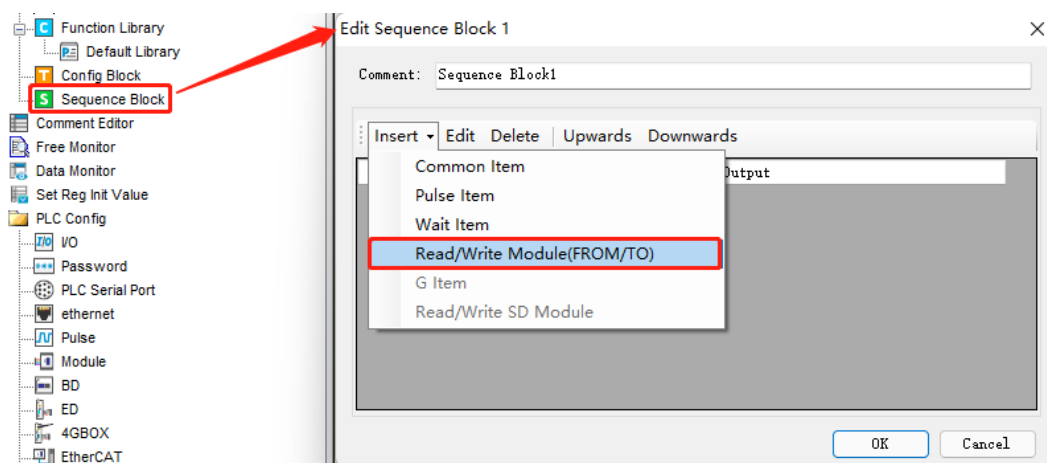


- После включения M0 запишите значения из 7 последовательных регистров, начиная с HD0, в регистр ID в PMP-RTC-BD платы BD №1 часов.

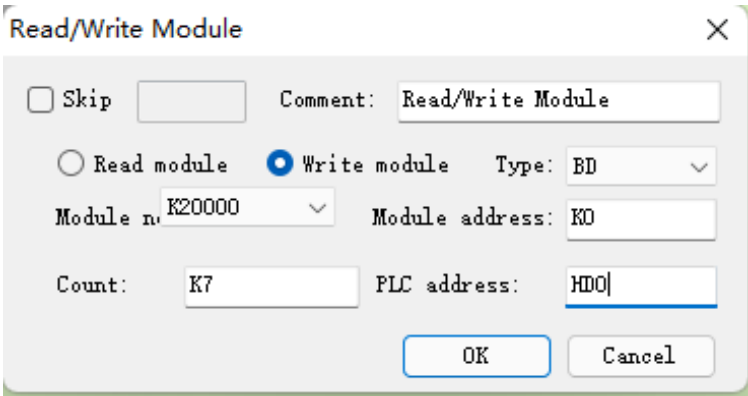
Исходные данные		Адрес платы BD №1	Описание	Данные часов	Примечание
HD0	→	ID20000 (K0)	Секунда	0~59	Десятичная
HD1	→	ID20001 (K1)	Минута	0~59	Десятичная
HD2	→	ID20002 (K2)	Час	0~23	Десятичная
HD3	→	ID20003 (K3)	Дата	1~31	Десятичная
HD4	→	ID20004 (K4)	Месяц	1~12	Десятичная
HD5	→	ID20005 (K5)	Год	00~99	Десятичная
HD6	→	ID20006 (K6)	Неделя	0 (воскресенье) - 6 (суббота)	Десятичная

- Команда TO должна быть введена в блок последовательности. Последовательность выполнения:

Во всплывающем окне выберите "Insert" - "Read/Write Module (FROM/TO)" (Вставить - Чтение/запись модуля (ИЗ/В)):



В открывшемся окне настройки конфигурации задайте следующие параметры:



Примечание: номер модуля K20000 обозначает плату BD №1, K20001 обозначает плату BD №2; адреса модулей нумеруются от K0, что соответствует ID20000, ID20001.....ID20006.

4.10.5 Сложение данных часов [TADD]

1) Обзор

Сложение данных часов [TADD]			
16 бит	TADD	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20

Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя
--------------------------------------	--	---------------------------------------	--------------------------------

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес заголовка программного элемента для данных часов (часы, минуты, секунды)	16 бит, BIN
S2	Адрес заголовка программного элемента для данных часов (часы, минуты, секунды)	16 бит, BIN
D	Адрес результата	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•																	
S2	•																	
D	•																	

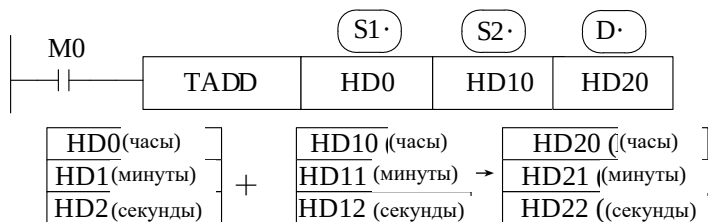
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

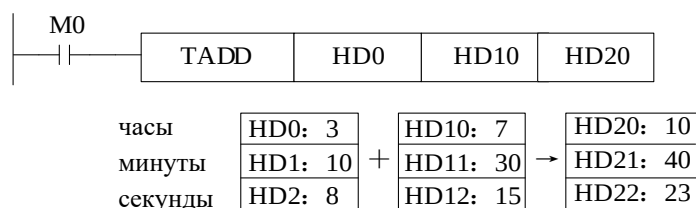


Примечание: соответствие регистров фиксировано, то есть они хранятся в следующем порядке: часы, минуты и секунды.

Если результатом операции является 0 часов, 0 минут, 0 секунд, SM20 будет включен. Операнды S1, S2 и D занимают по три регистра. Не используйте их в других целях.

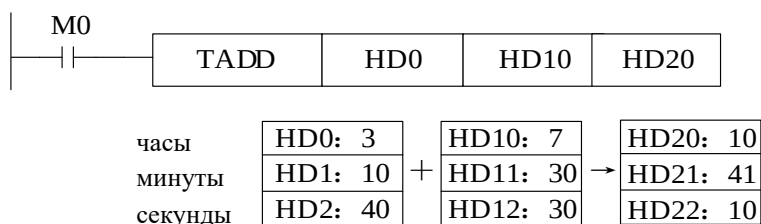
Пример 1:

<Общие условия>



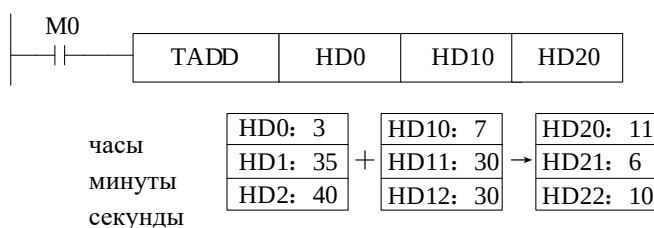
Пример 2:

<Свыше 59 секунд>



Пример 3:

<Свыше 59 минут>



Пример 4:

<Свыше 23 часов>



4.10.6 Вычитание данных часов [TSUB]

1) Обзор

Вычитание данных часов [TSUB]			
16 бит	TSUB	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя

2) Операнды

Операнды	Функция	Тип данных
S1	Адрес заголовка программного элемента для данных часов (часы, минуты, секунды)	16 бит, BIN
S2	Адрес заголовка программного элемента для данных часов (часы, минуты, секунды)	16 бит, BIN
D	Адрес результата	16 бит, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•																	
S2	•																	
D	•																	

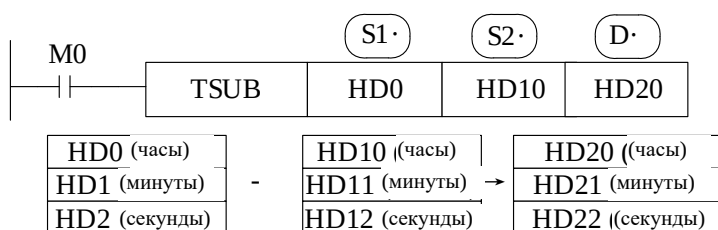
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание

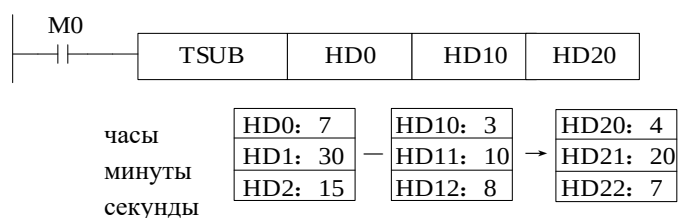


Примечание: соответствие регистров фиксировано, то есть они хранятся в следующем порядке: часы, минуты и секунды.

Если результатом операции является 0 часов, 0 минут, 0 секунд, SM20 будет включен. Операнды S1, S2 и D занимают по три регистра. Не используйте их в других целях.

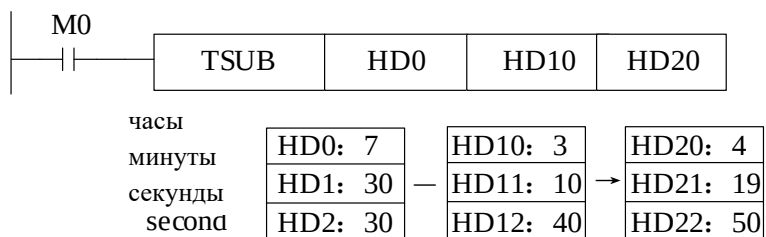
Пример 1:

<Общие условия>



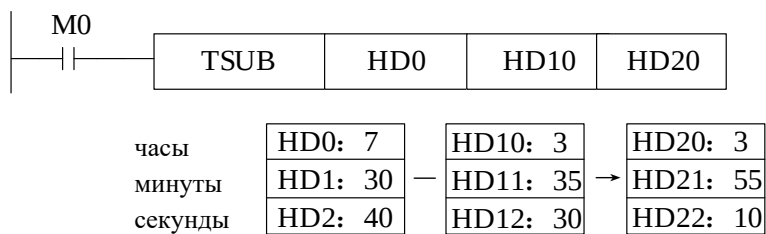
Пример 2:

<Меньше 0 секунд>



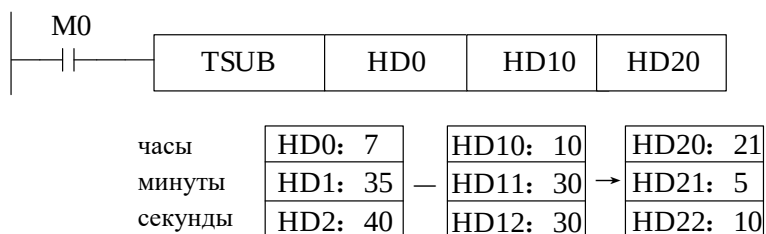
Пример 3:

<Меньше 0 минут>



Пример 4:

<Меньше 0 часов>



4.10.7 Преобразование часов, минут и секунд в секунды [HTOS]

1) Обзор

Преобразование часов, минут и секунд в секунды [HTOS]			
16 бит	HTOS	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20

Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя
--------------------------------------	--	---------------------------------------	--------------------------------

2) Операнды

Операнды	Функция	Тип данных
S	Данные часов до преобразования	16 бит, BIN
D	Данные часов после преобразования	16 бит, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m	
S	•																		
D	•																		

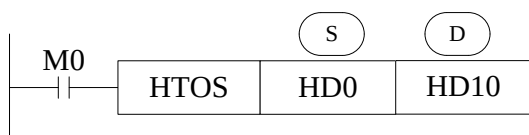
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



(HD, HD1, HD2) → (HD10, HD11)

часы, минуты, секунды → секунды

- После включения M0 данные часов (часы, минуты и секунды) в трех последовательных регистрах, начиная с HD0, преобразуются в секунды, которые сохраняются в регистре HD10 (двойное слово).
- Примечание: соответствие регистров фиксировано, то есть они хранятся в следующем порядке: часы, минуты и секунды.
- Операнды S занимают три регистра. Не используйте их в других целях.

4.10.8 Преобразование секунд в часы, минуты и секунды [STON]

1) Обзор

Преобразование часов, минут и секунд в секунды [STON]			
16 бит	-	32 бита	STON
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя

2) Операнды

Операнды	Функция	Тип данных
S	Данные часов до преобразования	16 бит, BIN
D	Данные часов после преобразования	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S	•																	
D	•																	

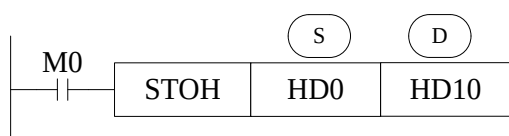
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



(HD0, HD1) → (HD10, HD11, HD12)

секунды → часы, минуты, секунды

- После включения M0 данные часов (часы, минуты и секунды) в трех последовательных регистрах, начиная с HD0, преобразуются в секунды, которые сохраняются в регистре HD10 (двойное слово).
- Примечание: соответствие регистров фиксировано, то есть они хранятся в следующем порядке: часы, минуты и секунды.
- Операнды S занимают три регистра. Не используйте их в других целях.

4.10.9 Сравнение данных часов [TCMP]

1) Обзор

Сравнение времени трех непрерывно работающих часов.

Сравнение данных часов [TCMP]			
16 бит	TCMP	32 бита	-
Условие	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Аппаратное обеспечение	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Программное обеспечение	Версия 3.5.3 или более поздняя

2) Операнды

Операнды	Функция	Модель
S1	Адрес программного компонента для часов	16 бит, BIN
S2	Адрес программного компонента для минут	16 бит, BIN
S3	Адрес программного компонента для секунд	16 бит, BIN
S4	Первый адрес в информации с часов реального времени ПЛК	16 бит, BIN
D2	Первый адрес результата сравнения	бит

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•																
S2	•	•																
S3	•	•																
S	•	•																
D													•	•				

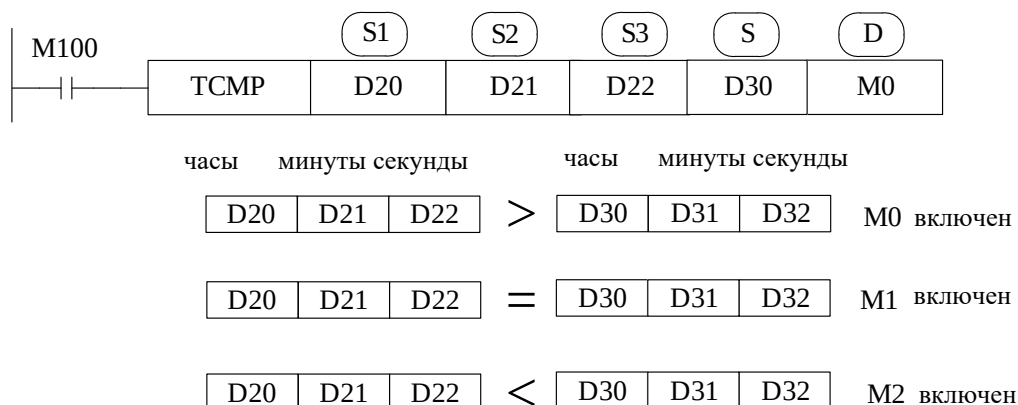
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

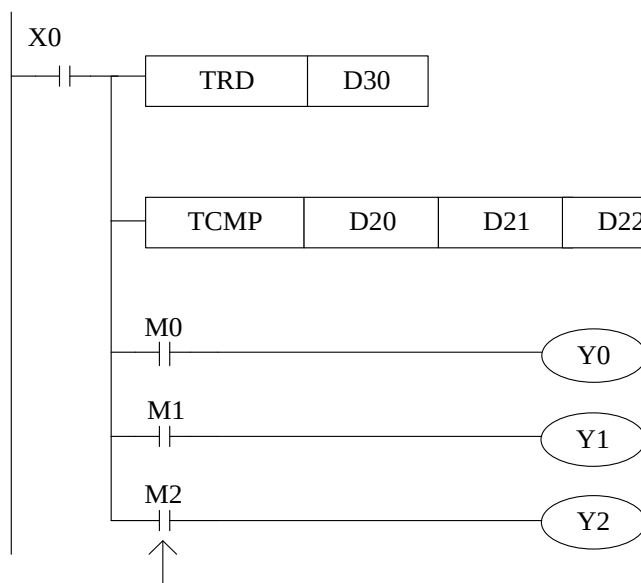
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Состояние M100 изменилось с OFF на ON, команда TCMP выполнена. Сравнение трех регистров, начиная с S4, с тремя регистрами S1, S2, S3 (год, месяц, день). Если значения регистров S1, S2, S3 больше значения регистра S4, M0 включается. Если значения регистров S1, S2, S3 равны значению регистра S4, M1 включается. Если значения регистров S1, S2, S3 меньше значения регистра S4, M2 включается.



Пример:

Текущее время — 15:32:49 7,30,2014, среда.

Таким образом, D33 = 15, D34 = 32, D35 = 49.

Если уставка времени составляет 15:32:49, D20 = 15, D21 = 32, D22 = 49, то Y1 = ON.

Если уставка времени составляет 17:32:49, D20 = 17, D21 = 32, D22 = 49, то Y0 = ON.

Если уставка времени составляет 2:32:5, D20 = 2, D21 = 32, D22 = 5, то Y2 = ON.

4.10.10 Сравнение даты (год, месяц, день) [DACMP]

1) Обзор

Преобразование часов, минут и секунд в секунды [STOH]			
16 бит	DACMP	32 бита	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	версия 3.4.6 (либо версия 3.5.3a) или более поздние версии	Требования к программному обеспечению	Версия 3.5.3 или более поздняя

2) Операнды

Операнды	Функция	Модель
S1	Адрес программного компонента для годов	16 бит, BIN
S2	Адрес программного компонента для месяцев	16 бит, BIN
S3	Адрес программного компонента для дней	16 бит, BIN
S4	Первый адрес в информации с часов реального времени ПЛК	16 бит, BIN
D2	Первый адрес результата сравнения	бит

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	•	•																
S2	•	•																
S3	•	•																
S	•	•																
D													•	•				

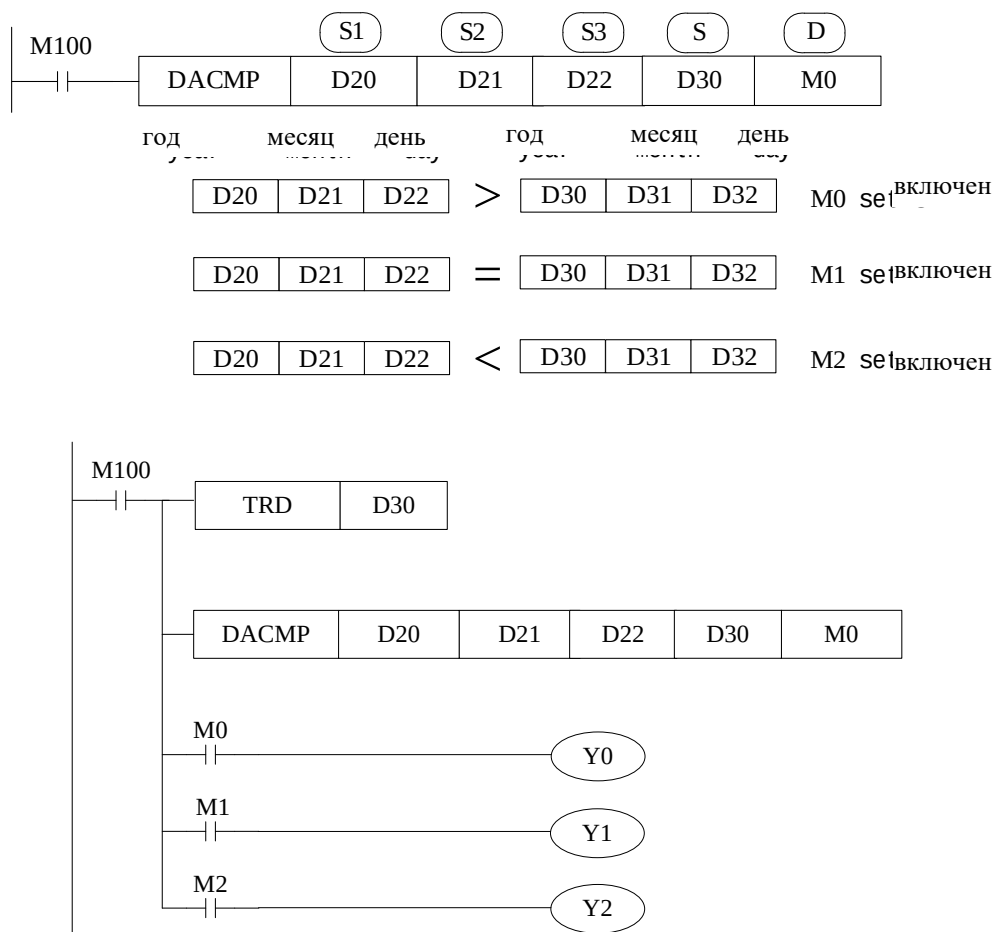
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



Пример:

Текущее время — 15:32:49 7,30,2014, среда.

Таким образом, D30 = 14, D31 = 7, D32 = 30.

Если уставка времени составляет 1,6,2015, D20 = 15, D21 = 1, D22 = 6, то Y0 = ON.

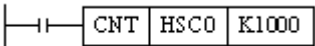
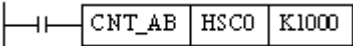
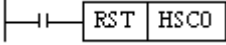
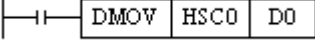
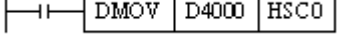
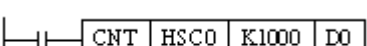
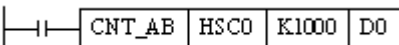
Если уставка времени составляет 7,30,2014, D20 = 14, D21 = 7, D22 = 31, то Y1 = ON.

Если уставка времени составляет 6,31,2014, D20 = 14, D21 = 6, D22 = 31, то Y2 = ON.

5. Высокоскоростной счетчик (hsc)

В этой главе описаны функции высокоскоростного счетчика, включая модель высокоскоростного счета, метод подключения, считывание/запись показаний HSC, сброс и т. д.

Перечень команд для HSC:

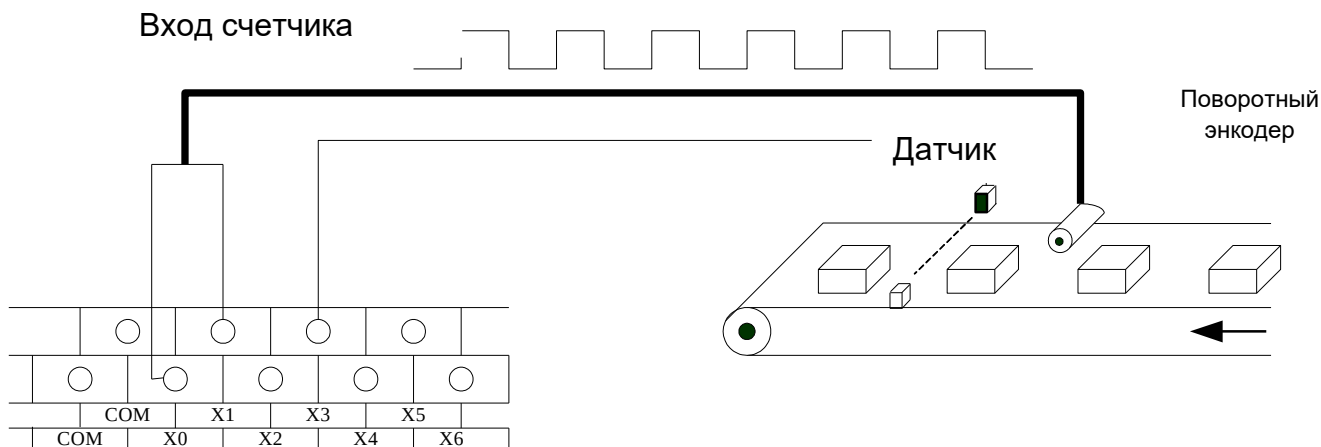
Имя команды	Функция	Команда	Глава
Считывание/запись показаний HSC			
CNT	24 сегмента в однофазном режиме отсутствуют		5-7-1
CNT_AB	24 сегмента в режиме фаз А и В отсутствуют		5-7-2
RST	Сброс HSC		5-7-3
DMOV	Считывание показаний HSC		5-7-4
DMOV	Запись показаний HSC		5-7-5
CNT	Высокоскоростной подсчет 100 сегментов в однофазном режиме (с прерыванием)		5-9-2
CNT_AB	Высокоскоростной подсчет 100 сегментов в режиме фаз А и В (с прерыванием)		5-9-3

5.1 Обзор функций

ПЛК серии PMP20 поддерживает функцию HSC (высокоскоростной счетчик), на которую не влияет цикл сканирования. Выбрав другой счетчик, проверьте высокоскоростные входные сигналы с помощью датчиков обнаружения и поворотных энкодеров. Максимальная частота проверки составляет 80 кГц.

Примечание:

- (1) Для ПЛК с NPN-подключением входа выберите энкодер с выходом с открытым коллектором NPN (OC) 24 В пост. тока; для ПЛК с PNP-подключением входа выберите энкодер с выходом с открытым коллектором PNP (OC) 24 В пост. тока.



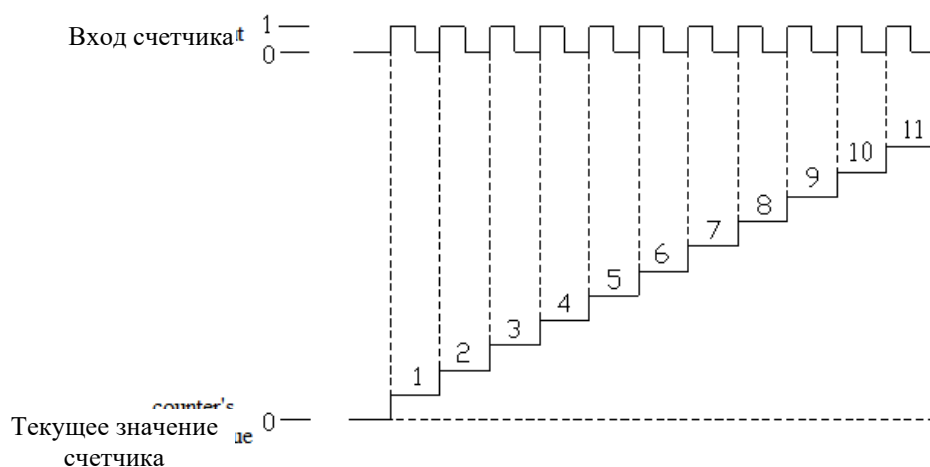
- (2) Если частота счета превышает 25 Гц, выберите высокоскоростной счетчик.

5.2 Режимы HSC

Высокоскоростной счетчик ПЛК серии RMP20 поддерживает два режима работы: Однофазный возрастающий режим и режим фаз А и В.

Однофазный возрастающий режим

В этом режиме значение счета увеличивается с каждым нарастающим фронтом импульса.

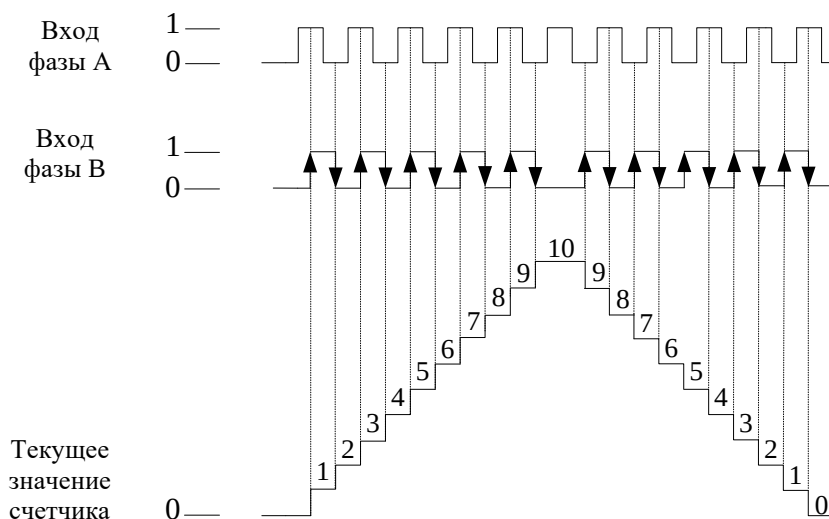


Режим фаз А и В

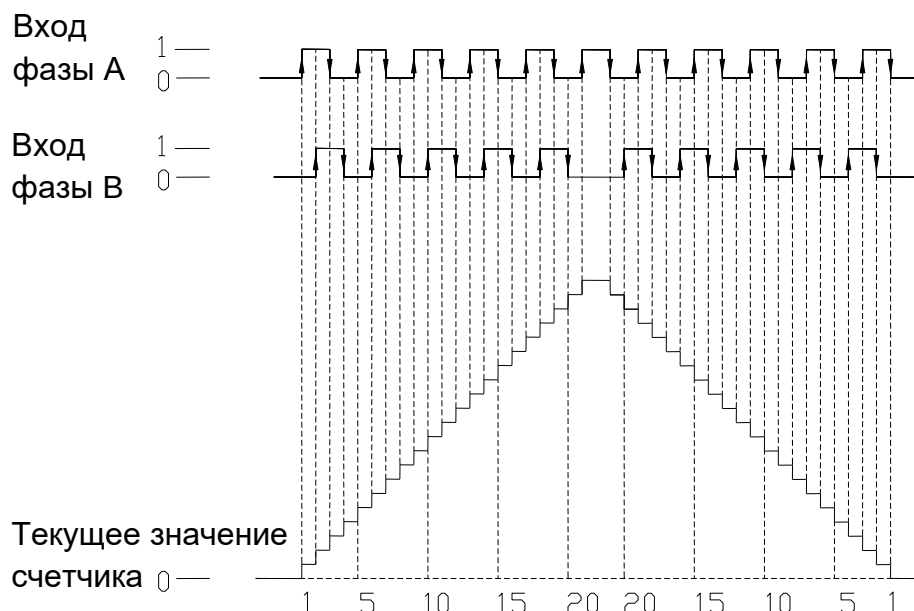
В этом режиме показания HSC увеличиваются или уменьшаются в соответствии с двумя дифференциальными сигналами (фаза А и фаза В). При умножении получаем 2-кратную частоту и 4-кратную частоту, но по умолчанию режимом счетчика является режим 4-кратной частоты.

Ниже представлены режимы 2-кратной и 4-кратной частоты:

2-кратная частота



4-кратная частота



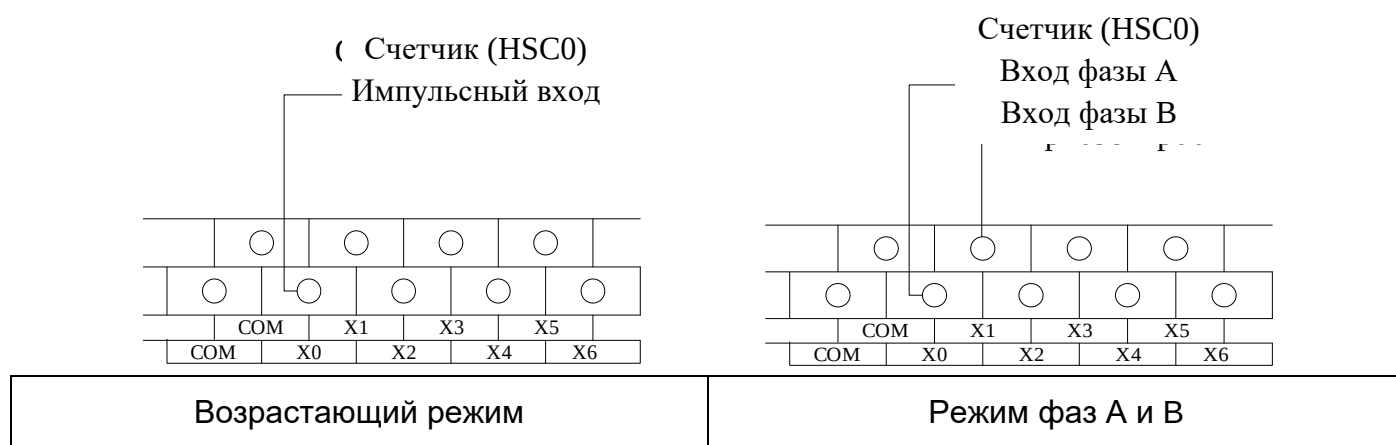
5.3 Диапазон HSC

Диапазон счета HSC: $K-2,147,483,648 \sim K+2,147,483,647$. Если значение счета выходит за пределы этого диапазона, возникает переполнение или отрицательное переполнение.

Переполнение означает, что значение счета меняется от $K + 2,147,483,647$ до $K - 2,147,483,648$, а затем счет продолжается; отрицательное переполнение означает, что значение счета меняется от $-2,147,483,648$ до $+2,147,483,647$, а затем счет продолжается.

5.4 Подключение входа HSC

Что касается подключения импульсного входа счетчика, то все зависит от модели ПЛК и модели счетчика; ниже показано несколько стандартных схем подключения входа (в качестве примера возьмем PMP20-60 HSC0):



5.5 Назначение портов HSC

1) Перечень каналов HSC ПЛК серии PMP20:

Модель ПЛК		Канал HSC	
		Возрастающий режим	Режим фаз А и В
PMP20	24/30/48/60	3	3
	30T4	4	4

Модель ПЛК		Канал HSC	
		Возрастающий режим	Режим фаз А и В
	60T4	4	4
	60T6	6	6
	60T10	10	10

2) Значение букв:

U	A	B	Z
Импульсный вход	Вход фазы А	Вход фазы В	Сбор импульсов фазы Z

Примечание: Функция подсчета сигналов фазы Z находится в процессе разработки.

В нормальном режиме входная частота X0 и X1 может достигать 80 кГц и 50 кГц соответственно в однофазном режиме и режиме фаз А и В. Другие клеммы имеют максимальную частоту 10 кГц и 5 кГц в однофазном режиме и режиме фаз А и В, соответственно.

Клеммы X можно использовать в качестве стандартных входных клемм, когда нет входных высокоскоростных импульсов. В таблице ниже 2 обозначает двойную частоту; 4 обозначает четырехкратную частоту; 2/4 означает, что можно настроить двойную и четырехкратную частоту.

PMP20-30T4/60T4												
	Возрастающий режим						Режим фаз А и В					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Макс. частота	80K	80K	80K	80K			50K	50K	50K	50K		
Четырехкратная частота							2/4	2/4	2/4	2/4		

PMP20-30T4/60T4												
	Возрастающий режим						Режим фаз А и В					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Прерывание счетчика	✓	✓	✓	✓			✓	✓	✓	✓		
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		

PMP20-60T6												
	Возрастающий режим						Режим фаз А и В					
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10
Макс. частота	80K	80K	80K	80K	80K	80K	50K	50K	50K	50K	50K	50K
Четырехкратная частота							2/4	2/4	2/4	2/4	2/4	2/4
Прерывание счетчика	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
X000	U						A					
X001							B					
X002							Z					
X003		U						A				
X004								B				
X005								Z				
X006			U						A			
X007									B			
X010									Z			
X011				U						A		
X012										B		
X013										Z		
X014					U						A	
X015											B	

X016											Z	
X017						U						A
X020												B
X021												Z

PMP20-60T10												
	Возрастающий режим											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Макс. частота	80K	80K	80K	80K	80K	80K	80K	80K	80K	80K		
Четырехкратная частота												
Прерывание счетчика	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
X000	U											
X001												
X002												
X003		U										
X004												
X005												
X006			U									

X007												
X010												
X011				U								
X012												
X013												
X014					U							
X015												
X016												
X017						U						
X020												
X021												
X022							U					
X023												
X024												
X025								U				
X026												
X027												
X030									U			
X031												
X032												
X033										U		
X034												

PMP20-60T10												
	Режим фаз А и В											
	HSC0	HSC2	HSC4	HSC6	HSC8	HSC10	HSC12	HSC14	HSC16	HSC18	HSC20	HSC22
Макс. частота	50K	50K	50K	50K	50K	50K	50K	50K	50K	50K		
Четырехкратная частота	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4	2/4		
Прерывание счетчика	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
X000	A											
X001	B											
X002	Z											
X003		A										
X004		B										
X005		Z										
X006			A									
X007			B									
X010			Z									
X011				A								
X012				B								

X013				Z								
X014					A							
X015					B							
X016					Z							
X017						A						
X020						B						
X021						Z						
X022							A					
X023							B					
X024							Z					
X025								A				
X026								B				
X027								Z				
X030									A			
X031									B			
X032									Z			
X033										A		
X034										B		
X035										Z		

5.6 Настройка удвоения частоты счета импульсов фаз А и В

Для подсчета импульсов фаз А и В двойную частоту можно настроить в специальных FLASH регистрах данных SFD321, SFD322, SFD323... SFD330, 2 обозначает двойную частоту; 4 обозначает четырехкратную частоту.

Имя регистра	Функция	Значение уставки	Значение
SFD320	Удвоение частоты HSC0	2	2, удвоение частоты
		4	4, удвоение частоты
SFD321	Удвоение частоты HSC2	2	2, удвоение частоты
		4	4, удвоение частоты
SFD322	Удвоение частоты HSC4	2	2, удвоение частоты
		4	4, удвоение частоты
SFD323	Удвоение частоты HSC6	2	2, удвоение частоты
		4	4, удвоение частоты
SFD324	Удвоение частоты HSC8	2	2, удвоение частоты
		4	4, удвоение частоты
SFD325	Удвоение частоты HSC10	2	2, удвоение частоты
		4	4, удвоение частоты
SFD326	Удвоение частоты HSC12	2	2, удвоение частоты
		4	4, удвоение частоты
SFD327	Удвоение частоты HSC14	2	2, удвоение частоты
		4	4, удвоение частоты
SFD328	Удвоение частоты HSC16	2	2, удвоение частоты
		4	4, удвоение частоты
SFD329	Удвоение частоты HSC18	2	2, удвоение частоты
		4	4, удвоение частоты

Примечание: После изменения регистра SFD необходимо перезапустить высокоскоростной счетчик (т.е. отключить и перезагрузить состояние накопителя), чтобы новая конфигурация вступила в силу!

5.7 Команды HSC

В этом разделе описывается использование команды высокоскоростного подсчета (CNT) в однофазном режиме, команды высокоскоростного подсчета (CNT_AB) в режиме фаз А и В, сброс высокоскоростного счетчика, считывание и запись значений высокоскоростного счетчика.

5.7.1 HSC в однофазном режиме [CNT]

1) Обзор

Команда HSC в однофазном режиме.

HSC в однофазном режиме [CNT]			
16-битная команда	-	32-битная команда	CNT
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S	Установка кода HSC (например, HSC0)	32 бита, BIN
D	Установка значения для сравнения (например, K100, D0)	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Может быть только HSC																	
S2	•																	

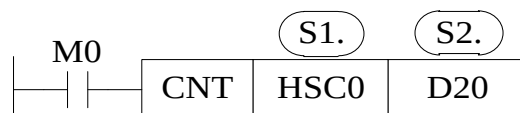
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- После включения M0 HSC0 подсчитывает сигналы X0 в однофазном режиме, сравнивает значение высокоскоростного подсчета со значением, установленным в регистре D20. Если значение высокоскоростного подсчета равно значению уставки, катушка HSC0 немедленно включается, а значение счета накапливается в HSCD0 (двойные слова).
- Если подсчет завершен, и условие включения M0 сохраняется, HSC0 останется во включенном состоянии и продолжит подсчет, а значение счета в HSCD0 продолжит накапливаться.
- Если подсчет завершен, и условие включения M0 не сохраняется, HSC0 останется включенным, а значение счета в HSCD0 останется неизменным.
- Если в процессе подсчета M0 отключается и подключается снова, значения в HSCD0 будут продолжать накапливаться после последнего значения счета.
- Если в процессе подсчета значение уставки в D20 изменяется, и текущее значение счета меньше нового значения уставки, то сравнивается новое значение уставки.
- Режим фронта высокоскоростного подсчета в однофазном режиме можно задать с помощью регистров SFD310 - SFD313 (соответствующих регистрам HSC0 - HSC6). В качестве примера рассмотрим HSC0. SFD310 принимает следующие значения:

0: подсчет по нарастающему фронту;

1: подсчет по спадающему фронту;

2: указывает на то, что подсчет ведется одновременно по нарастающему фронту и спадающему фронту.

Примечание: эта функция поддерживается только прошивкой ПЛК версии 3.4.6 и более поздних версий.

5.7.2 HSC в режиме фаз А и В [CNT_AB]

1) Обзор

Команда HSC в режиме фаз А и В.

HSC в режиме фаз А и В [CNT_AB]			
16-битная команда	-	32-битная команда	CNT_AB
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S	Установка кода HSC (например, HSC0)	32 бит, BIN
D	Установка значения для сравнения (например, K100, D0)	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Может быть только HSC																	
S2	•																	

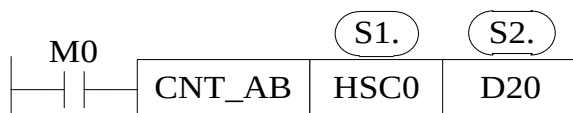
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

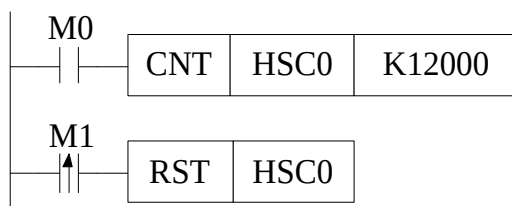
Функции и операции



- После включения M0 HSC0 подсчитывает сигналы X0, X1 в режиме фаз A и B, сравнивает значение высокоскоростного подсчета со значением, установленным в регистре D20. Если значение высокоскоростного подсчета равно значению уставки, катушка HSC0 немедленно включается, а значение счета накапливается в HSCD0 (двойные слова).
- Если условие включения M0 сохраняется, HSC0 останется во включенном состоянии и продолжит подсчет, а значение счета в HSCD0 продолжит накапливаться.
- Если условие включения M0 не сохраняется, HSC0 останется включенным, а значение счета в HSCD0 останется неизменным.
- Если в процессе подсчета M0 отключается и подключается снова, значения в HSCD0 будут продолжать накапливаться после последнего значения счета.
- Если в процессе подсчета значение уставки в D20 изменяется, и текущее значение счета меньше нового значения уставки, то сравнивается новое значение уставки.

5.7.3 Сброс HSC [RST]

Режим сброса высокоскоростного счетчика — это режим программного сброса.



Как показано выше, после включения M0 HSC0 начинает подсчитывать импульсы на входе порта X0; после переключения M1 из состояния OFF в состояние ON HSC0 сбрасывается, а значение счета в HSCD0 (двойные слова) очищается.

5.7.4 Считывание значения HSC [DMOV]

1) Обзор

Считывание значения HSC в указанный регистр.

Считывание значения HSC [DMOV]			
16-битная команда	-	32-битная команда	DMOV
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению		Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S	Установка кода HSC	32 бита, BIN
D	Установка регистра чтения/записи	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Может быть только HSC																	
S2	•																	

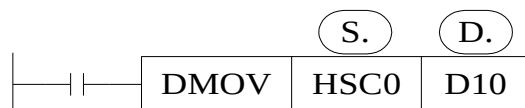
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

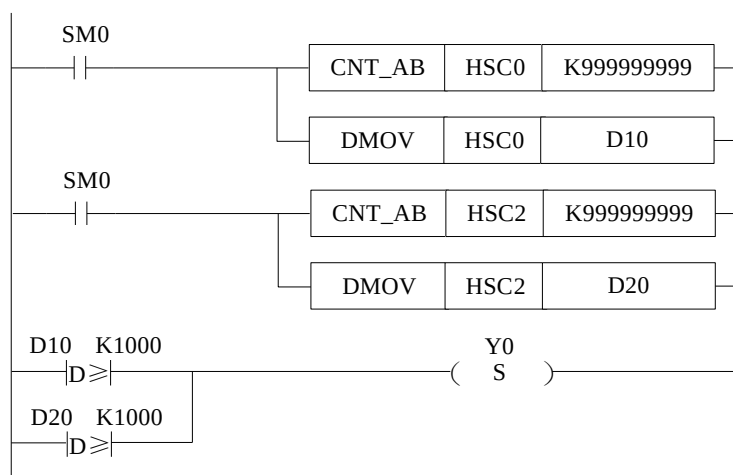
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- При достижении условия срабатывания значение высокоскоростного счетчика в накопительном регистре HSCD0 (двойные слова), соответствующее HSC0 высокоскоростного счетчика, считывается в регистр данных D10 (двойные слова).
- Высокоскоростной счетчик не может напрямую выполнять какие-либо рабочие команды или команды сравнения данных (например, DMUL, LD > и т. д.), за исключением DMOV. Такие команды могут выполняться только после считывания и записи в другие регистры.
- Так как высокоскоростной счетчик является счетчиком двойных слов, он должен использовать 32-битную команду DMOV.
- DMOV часто используется вместе с высокоскоростным счетчиком.

Пример программы:



5.7.5 Запись значения HSC [DMOV]

1) Обзор

Запись значения указанного регистра в HSC.

Запись значения HSC [DMOV]			
16-битная команда	-	32-битная команда	DMOV
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт/спадающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению		Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S	Установка кода HSC	32 бит, BIN
D	Установка регистра чтения/записи	32 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Может быть только HSC																	
S2	•																	

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- При достижении условия срабатывания значение в регистре данных с двойными словами D20 записывается в накопительный регистр HSCD0 (двойное слово), соответствующий HSC0 высокоскоростного счетчика, а исходные данные заменяются.
- Высокоскоростной счетчик не может напрямую выполнять какие-либо рабочие команды или команды сравнения данных (например, DMUL, LD > и т. д.), за исключением DMOV. Такие команды могут выполняться только после считывания и записи в другие регистры.
- Так как высокоскоростной счетчик является счетчиком двойных слов, он должен использовать 32-битную команду DMOV.
- DMOV часто используется вместе с высокоскоростным счетчиком.

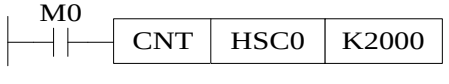
5.7.6 Отличия высокоскоростного счетчика (HSC) от стандартного счетчика

Хотя команды высокоскоростного счетчика используют "CNT" так же, как и команды стандартного счетчика, их функции совершенно разные.

Если состояние M0 изменяется один раз с OFF на ON, к значению стандартного счетчика добавляется 1.

Запущенный высокоскоростной счетчик должен находиться в нормально замкнутом состоянии при подсчете, что эквивалентно включению счетчика больших чисел, при этом значение счетчика больших чисел не изменяется. Только когда соответствующая входная клемма внешних сигналов примет сигнал, счетчик больших чисел начнет подсчет. Если на входную клемму внешних сигналов поступил сигнал, и запущенный счетчик больших чисел не находится в замкнутом состоянии, счетчик больших чисел не будет вести подсчет.

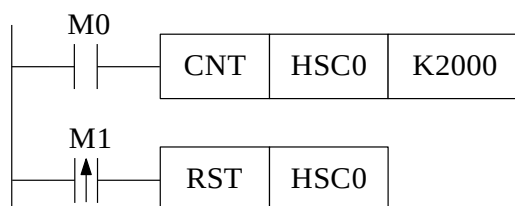
Отличия показаны в таблице ниже.

Тип счетчика	Формат команды	Функция
Стандартный счетчик		Подсчет количества выключений и включений M0; когда значение счета достигнет 2000, C0 включится.
Высокоскоростной счетчик		После включения M0 подсчитываются входные сигналы X0, и когда значение счета достигнет 2000, HSC0 включается. Во время подсчета M0 всегда должен быть включен.

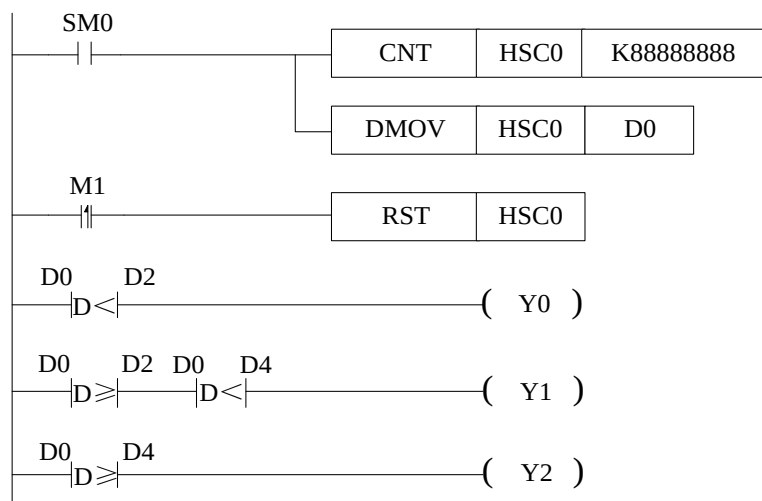
5.8 Пример HSC

Ниже в качестве примера рассмотрим PMP20-60 для иллюстрации метода программирования HSC.

Однофазный возрастающий режим



- После включения M0 HSC0 считает нарастающий фронт в выключенном состоянии до включения входа порта X0 на высокой скорости.
- При появлении нарастающего фронта M1 сбрасывается HSC0 высокоскоростного счетчика и HSCD0 (двойное слово).

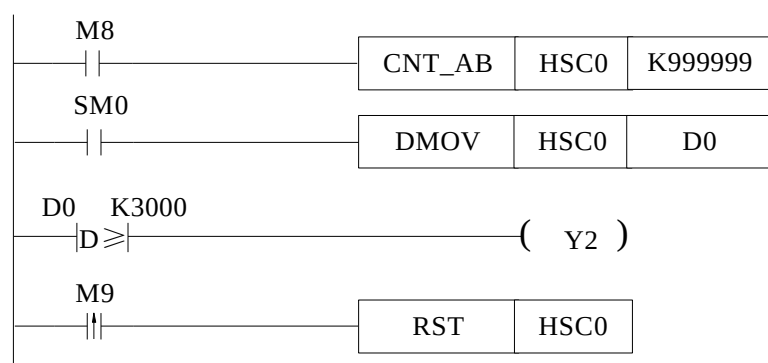


Поскольку значение высокоскоростного счетчика составляет 32 бита, все описанные здесь команды являются 32-битными, например DMOV, VDLD<, DLD≥.

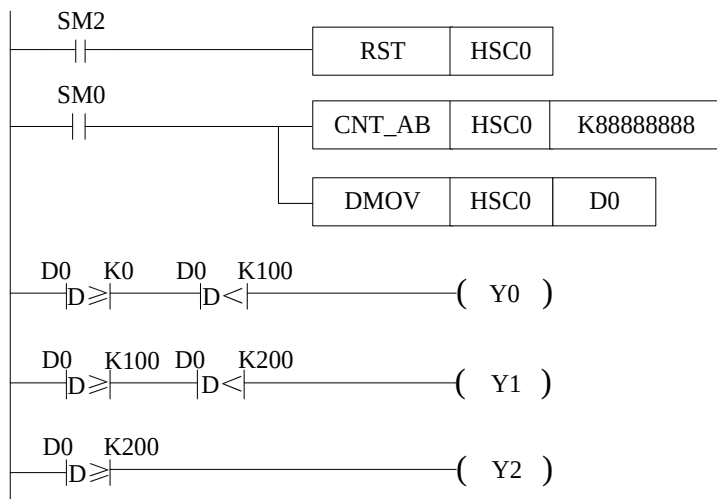


- После включения SM0 HSC0 ведет подсчет у порта X0 в однофазном инкрементном (возрастающем) режиме (значение уставки равно K888888) и считывает значение высокоскоростного счетчика в D0 (двойное слово) в реальном времени.
- Если D0 (двойные слова) меньше D2 (двойные слова), Y0 включен, если D0 (двойные слова) больше или равен D2 (двойные слова) и меньше D4 (двойные слова), Y1 включен. Если D0 (двойные слова) больше или равен D4 (двойные слова), Y2 включен.
- При появлении нарастающего фронта M1 сбрасываются HSC0 и HSCD0 (двойные слова).
- Так как высокоскоростной счетчик является счетчиком двойных слов, используйте команды, состоящие из двойных слов DLD < и DLD ≥.

Режим входных сигналов фаз А и В



- Когда M8 находится во включенном состоянии, HSC0 начинает подсчет. Входные сигналы от X0 (фаза А) и X1 (фаза В).
- После включения SM0 значение в HSCD0 (двойные слова), относящееся к HSC0, записывается в D0 (двойные слова) в режиме реального времени.
- Когда текущее значение счета превышает 3000, Y2 включается.
- При появлении нарастающего фронта M9 HSC0 и HSCD0 (двойные слова) сбрасываются.



Поскольку значение высокоскоростного счетчика составляет 32 бита, все описанные здесь команды являются 32-битными, например DMOV, DLD<, DLD≥.

- При нарастающем фронте исходного прямого импульса с катушки SM2, то есть в начале каждого цикла сканирования, HSC0 сбрасывается, а значение счета в HSCD0 очищается.
- Если катушка SM0 включена, HSC0 начинает подсчет у портов X0 и X1 в режиме фаз A и B. Уставка подсчета — K8888888. При этом значение счета в HSCD0 (двойные слова) записывается в D0 (двойные слова) в режиме реального времени.
- Если значение счета в D0 (двойные слова) больше K0 и меньше K100, выходная катушка Y0 включена; если значение счета в D0 (двойные слова) больше или равно K100 и меньше K200, выходная катушка Y1 включена; когда значение счета в D0 (двойные слова) больше или равно K200, выходная катушка Y2 включена.
- Поскольку высокоскоростной счетчик является счетчиком двойных слов, для сравнения необходимо использовать команды сравнения двойных слов DLD ≥ и DLD <.

5.9 Прерывания HSC

5.9.1 Обзор функций и панель настройки

В ПЛК серии RMP20 некоторые высокоскоростные счетчики (см. таблицу распределения входных портов высокоскоростного счетчика в главе 5-5 для каждого типа ПЛК) имеют значение уставки 32 бита в сегментах 1-100. Если разность значений высокоскоростного счета станет равна значению уставки соответствующих 100 сегментов, произойдет прерывание согласно соответствующей метке прерывания.

Если настроено значение уставки сегмента N, то должны быть метка прерывания и программа прерываний, соответствующая сегменту N. Метки прерывания, соответствующие каждому высокоскоростному счетчику, представлены в главе 5-9-4.

При использовании функции прерываний высокоскоростного счетчика команды можно писать напрямую (см. главы 5-9-2 и 5-9-3) или настраивать с помощью программной панели. Нажмите HCNT в программном обеспечении PROMPOWER PLC Studio, откроется окно, показанное ниже.

High Speed Count 24 Section Config

Single phase 100 segment high speed counting

High Speed C: HSC0

Compare Value: D500

Interrupt Address: HD100

☐ Opposite ☒ Absolute

☐ Circulate ☐ Cam

Config Value

Compare Value: 99999999

Section Num: 3

Section Num	Value
Segment1 Count Num:	500
Segment2 Count Num:	20000
Segment3 Count Num:	50000

Read From PLC Write To PLC OK Cancel

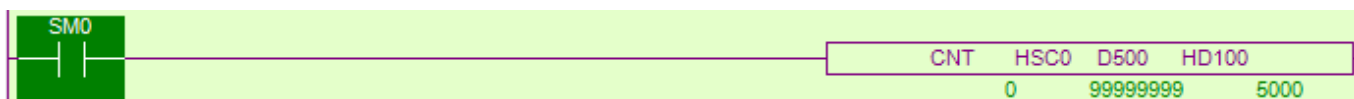
На этой панели можно настраивать параметры прерываний высокоскоростного счетчика. Для пояснения функции каждого параметра в качестве примера рассмотрим настройки, представленные на рисунке выше.

Параметр		Функция
Single phase 100 segment high speed counting	Высокоскоростной подсчет 100 сегментов в однофазном режиме	Высокоскоростной подсчет в однофазном инкрементном режиме
	Высокоскоростной подсчет 100 сегментов в режиме фаз А и В	Высокоскоростной подсчет в режиме фаз А и В
High Speed C HSC0 ▾	HSC0-HSC18 (32 бита)	Номер высокоскоростного счетчика, соответствующий входному порту высокоскоростных импульсов
Compare Value: D500	Свободное назначение	HSC0 включается, когда значение счета равно значению в регистре.
Compare Value: 99999999	Свободное назначение	Когда счет идет до значения для сравнения, HSC0 включен; значение для сравнения можно настроить здесь или ввести в регистр сравнения D500.
☐ Opposite ☒ Absolute	Relative (Относ.)	Происходит прерывание сегмента N, когда значение счета = значение счета прерывания сегмента N-1 + значение уставки сегмента N.
	Absolute (Абсол.)	Прерывание произойдет, когда значение счета станет равным значению уставки.
Interrupt Address: HD100	Свободное назначение	Значения уставки для 100 сегментов прерываний высокоскоростного счетчика сохраняются в регистрах, начиная с HD100; значения уставки сохраняются в регистрах двойных слов HD100, HD102, HD104....

☐ Circulate ☐ Cam	Interruption cycle (Цикл прерываний)	Должен применяться в относительном режиме. После завершения всех прерываний прерывания высокоскоростного счетчика могут по-прежнему генерироваться циклически.
	CAM (КУЛАЧОК)	Должен применяться в абсолютном режиме. Если значение счета равно любому значению уставки, происходит прерывание.
Section Num: 3	1-100 в качестве опции	Если введено значение 3, это означает выполнение трех прерываний высокоскоростного счетчика.
Value	Свободное назначение	Каждый сегмент соответствует значению счетчика прерываний, которое записывается в адресный блок, начиная с HD100; время прерывания зависит от относительного/абсолютного режима счета.

Подробную процедуру использования вышеуказанных параметров см. в следующих главах.

После записи в ПЛК и нажатия кнопки "ОК" настройка команды прерывания высокоскоростного счетчика будет завершена, как показано на рисунке ниже.



5.9.2 Высокоскоростной подсчет 100 сегментов в однофазном режиме [CNT]

1) Обзор

Команда HSC 100 сегментов в однофазном режиме.

Высокоскоростной подсчет 100 сегментов в однофазном режиме [CNT]			
16-битная команда	-	32-битная команда	CNT
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Настройка HSC (пример: HSC0)	32 бита, BIN
S2	Настройка значения для сравнения (например, K100, D0)	32 бита, BIN
S3	Настройка значения уставки для 100 сегментов	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Может быть только HSC																	
S2	•								•									
S3	•																	

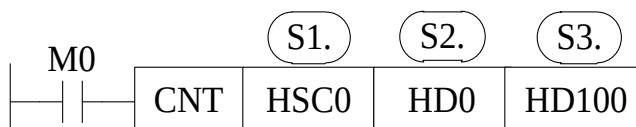
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Когда высокоскоростной счетчик HSC0 выполняет подсчет в однофазном режиме, значение высокоскоростного счета сравнивается с блоком данных, начиная с HD100 (например, HD102, HD102, HD104 и другими регистрами двойных слов). Немедленно происходит прерывание высокоскоростного счетчика при выполнении условия; каждый раздел соответствующих меток прерывания описан в главе 5-9-4.
- Во время высокоскоростного подсчета нельзя изменять значение уставки для 100 сегментов.
- Во время высокоскоростного подсчета условие включения M0 нельзя устранить. Если отключить и перезагрузить M0, прерывания не произойдет. Сначала необходимо сбросить высокоскоростной счетчик, а затем снова включить M0, чтобы вызвать прерывание.
- Если прерывание завершено в ходе одного выполнения, и необходимо снова запустить прерывание, то сначала следует сбросить высокоскоростной счетчик, а затем снова обеспечить условие включения.
- В циклическом режиме прерываний прерывания могут генерироваться последовательно до тех пор, пока M0 остается во включенном состоянии.

5.9.3 Высокоскоростной подсчет 100 сегментов в режиме фаз А и В [CNT_AB]

1) Обзор

Команда HSC 100 сегментов в режиме фаз А и В.

Высокоскоростной подсчет 100 сегментов в режиме фаз А и В [CNT_AB]			
16-битная команда	-	32-битная команда	CNT_AB
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Настройка HSC (пример: HSC0)	32 бита, BIN
S2	Настройка значения для сравнения (пример: K100, D0)	32 бита, BIN
S3	Настройка значения уставки для 100 сегментов	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	Может быть только HSC																	
S2	•								•									
S3	•																	

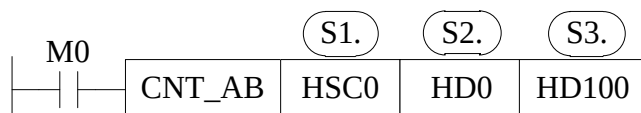
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Описание



- Когда высокоскоростной счетчик HSC0 выполняет подсчет в режиме фаз А и В, значение высокоскоростного счета сравнивается с блоком данных, начиная с HD100 (например, HD102, HD102, HD104 и другими регистрами двойных слов). Немедленно происходит прерывание высокоскоростного счетчика при выполнении условия; каждый раздел соответствующих меток прерывания описан в главе 5-9-4.
- Во время высокоскоростного подсчета нельзя изменять значение уставки для 100 сегментов.
- Во время высокоскоростного подсчета условие включения M0 нельзя устранить. Если отключить и перезагрузить M0, прерывания не произойдет. Сначала необходимо сбросить высокоскоростной счетчик, а затем снова включить M0, чтобы вызвать прерывание.
- Если прерывание завершено в ходе одного выполнения, и необходимо снова запустить прерывание, то сначала следует сбросить высокоскоростной счетчик, а затем снова обеспечить условие включения.
- В циклическом режиме прерываний прерывания могут генерироваться последовательно до тех пор, пока M0 остается во включенном состоянии.

5.9.4 Признак прерывания HSC

Признаки прерывания 100 сегментов каждого HSC приведены в таблице ниже. Например, признаки прерывания 100 сегментов HSC0 — I2000, I2001, I2002.....I2099.

HSC	Признак прерывания					
	Сегмент 1	Сегмент 2	Сегмент 3		Сегмент N	Сегмент 100
HSC0	I2000	I2001	I2002		I(2000+N-1)	I2099
HSC2	I2100	I2101	I2102		I(2100+N-1)	I2199
HSC4	I2200	I2201	I2202		I(2200+N-1)	I2299
HSC6	I2300	I2301	I2302		I(2300+N-1)	I2399
HSC8	I2400	I2401	I2402		I(2400+N-1)	I2499
HSC10	I2500	I2501	I2502		I(2500+N-1)	I2599
HSC12	I2600	I2601	I2602		I(2600+N-1)	I2699
HSC14	I2700	I2701	I2702		I(2700+N-1)	I2799
HSC16	I2800	I2801	I2802		I(2800+N-1)	I2899
HSC18	I2900	I2901	I2902		I(2900+N-1)	I2999

5.9.5 Значения уставок в абсолютном или относительном режиме

Значения уставок различаются в абсолютном и относительном режиме. Относительный/абсолютный режим можно настроить на программной панели. Режим также можно изменять с помощью специального Flash-регистра SFD330. (Примечание: (Чтобы настройки вступили в силу, необходимо ВЫКЛЮЧИТЬ и ВКЛЮЧИТЬ привод)

0: Относительный режим

1: Абсолютный режим

- **Относительный режим**

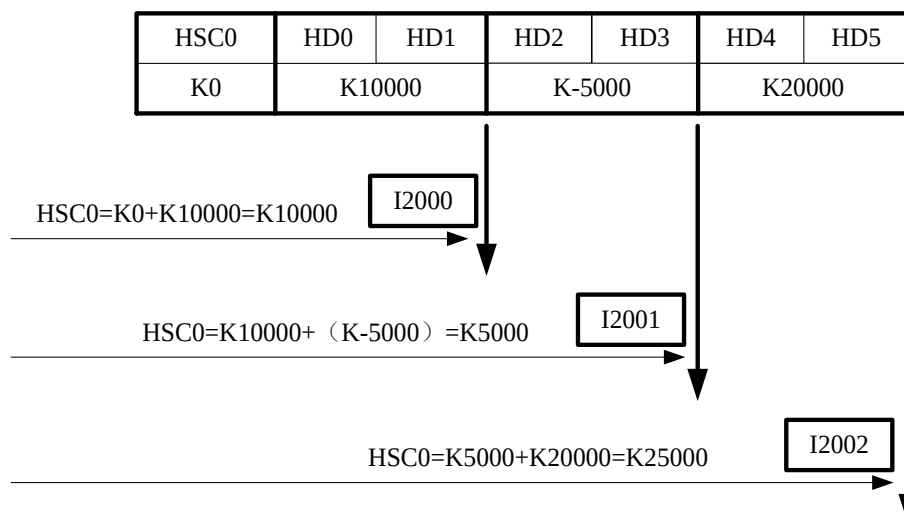
В относительном режиме значение уставки 100 сегментов высокоскоростного счетчика является относительным суммарным значением. Если значение уставки счетчика равно сумме подсчета прерываний сегмента N-1 и значения уставки сегмента N, то генерируется прерывание сегмента N.

Метки прерывания сегмента N соответствуют уставкам прерывания сегмента N. Регистр уставок прерываний N+1 зарезервирован для других целей.

Пример 1:

Текущее значение HSC0 равно 0, значение уставки сегмента 1 равно 10000, значение уставки сегмента 2 равно – 5000, значение уставки сегмента 3 равно 20000. При запуске счета, когда текущее значение счетчика равно 10000, он генерирует прерывание сегмента 1 I2000; когда текущее значение счетчика равно 5000, он генерирует прерывание сегмента 2 I2001; когда текущее значение счетчика равно 25000, он генерирует прерывание сегмента 3 I2002.

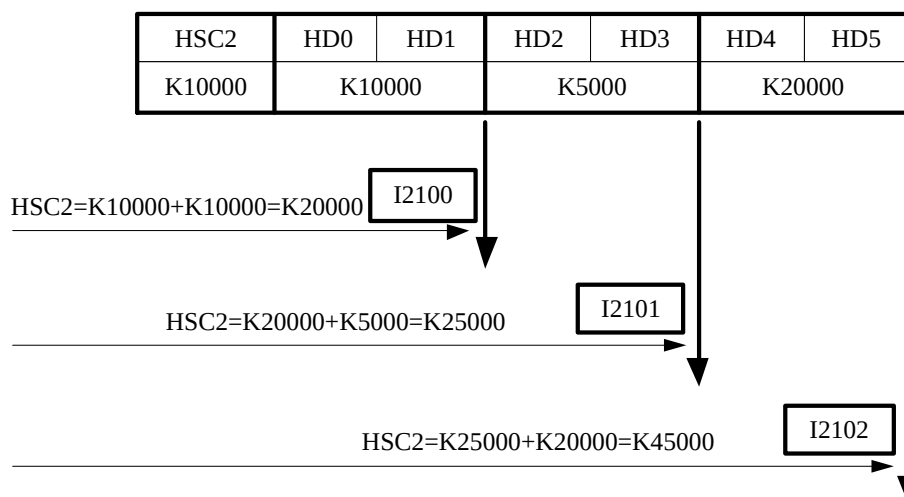
См. схему ниже:



Пример 2:

Текущее значение HSC2 равно 10000, значение уставки сегмента 1 равно 10000, значение уставки сегмента 2 равно – 5000, значение уставки сегмента 3 равно 20000. При запуске счета, когда текущее значение счетчика равно 20000, он генерирует прерывание сегмента 1 I2100; когда текущее значение счетчика равно 25000, он генерирует прерывание сегмента 2 I2101; когда текущее значение счетчика равно 45000, он генерирует прерывание сегмента 3 I2102.

См. схему ниже:

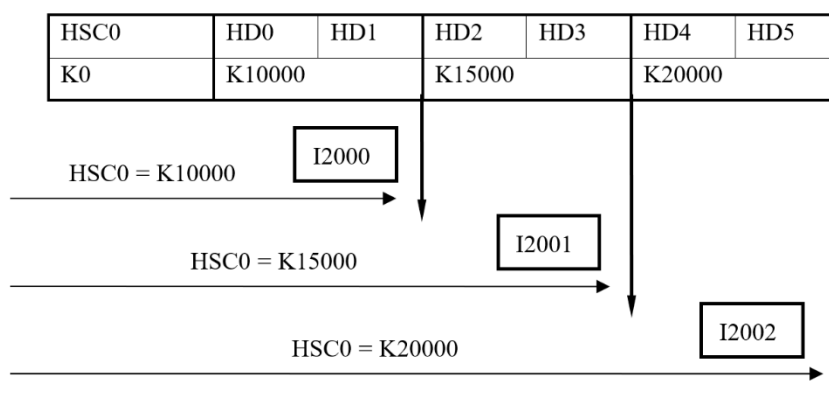


• Абсолютный режим

В абсолютном режиме прерывание происходит, когда значение счета становится равным значению уставки каждого сегмента счетчика. Метки прерывания сегмента N соответствуют уставкам прерывания сегмента N. Регистр уставок прерываний N+1 зарезервирован для других целей.

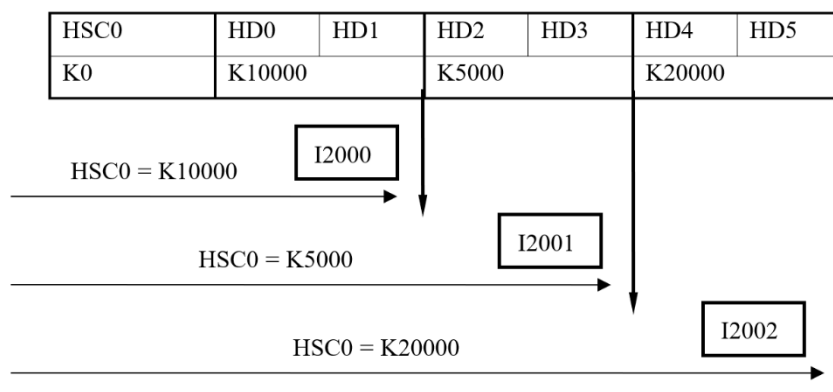
Пример 1:

Текущее значение счетчика HSC0 равно 0, значение уставки сегмента 1 равно 10000, значение уставки сегмента 2 равно 15000, а значение уставки сегмента 3 равно 20000. При начале подсчета, если текущее значение счетчика равно 10000, генерируется прерывание сегмента 1 I2000; если текущее значение счетчика равно 15000, генерируется прерывание сегмента 2 I2001; если текущее значение счетчика равно 20000, генерируется прерывание сегмента 3 I2002.



Пример 2:

Текущее значение счетчика HSC2 равно 5000, значение уставки сегмента 1 равно 10000, значение уставки сегмента 2 равно 5000, а значение уставки сегмента 3 равно 20000. При начале подсчета, если текущее значение счетчика равно 10000, генерируется прерывание сегмента 1 I2100; если текущее значение счетчика равно 5000, генерируется прерывание сегмента 2 I2101; если текущее значение счетчика равно 20000, генерируется прерывание сегмента 3 I2102.



Примечание:

При выполнении абсолютного подсчета в режиме без кулачков прерывания счетчика генерируются последовательно, т. е. прерывание сегмента 1, прерывание сегмента 2, прерывание сегмента 3... При прерывания сегмента прерывания не происходит, даже если значение счетчика снова достигнет значения уставки сегмента.

Как и в примере выше, если значение счета увеличится с 4000 до 5000 и 10000 после прерывания сегмента 1 и 2, прерывание сегмента 1 и 2 больше не произойдет, а прерывание сегмента 3 произойдет, если значение счета продолжит увеличиваться до 20000.

5.9.6 Режим прерываний циклов HSC

Режим 1: Одиночный цикл (нормальный режим)

Прерывание HSC больше не произойдет после его окончания. Следующие условия могут снова запустить прерывание.

- (1) Сброс HSC
- (2) Повторное включение HSC

Прерывание генерируется в виде следующей последовательности при выполнении одного цикла:

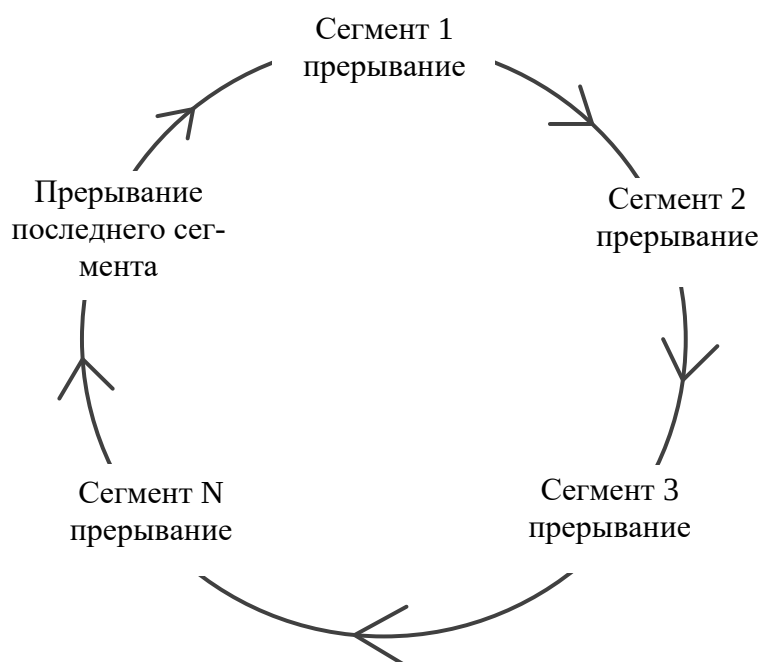


Режим 2: Непрерывный цикл

Прерывание непрерывного цикла подходит только для режима относительного подсчета. В режиме непрерывного цикла прерывание возобновится после его завершения. Этот режим особенно подходит для следующих условий:

- (1) Непрерывное перемещение вперед-назад
- (2) Генерирование прерывания цикла в соответствии с фиксированным импульсом

Прерывания непрерывного цикла (без включенной функции для кулачков) происходят в следующем порядке.



С помощью настройки SFD331 пользователи могут переключаться между режимом одиночного цикла и режимом непрерывного цикла. Подробная таблица распределения представлена ниже.

(Примечание: настройки вступят в силу после выключения и повторного включения привода)

Адрес	HSC	Настройка
Бит0	Цикл прерываний 100 сегментов HSC (HSC0)	0: одиночный цикл 1: непрерывный цикл
Бит1	Цикл прерываний 100 сегментов HSC (HSC2)	
Бит2	Цикл прерываний 100 сегментов HSC (HSC4)	
Бит3	Цикл прерываний 100 сегментов HSC (HSC6)	
Бит4	Цикл прерываний 100 сегментов HSC (HSC8)	
Бит5	Цикл прерываний 100 сегментов HSC (HSC10)	
Бит6	Цикл прерываний 100 сегментов HSC (HSC12)	
Бит7	Цикл прерываний 100 сегментов HSC (HSC14)	
Бит8	Цикл прерываний 100 сегментов HSC (HSC16)	
Бит9	Цикл прерываний 100 сегментов HSC (HSC18)	

5.9.7 Функция для кулачков, связанная с прерываниями высокоскоростного счетчика

Функция для кулачков, связанная с прерываниями высокоскоростного счетчика: После настройки всех значений уставок прерываний выбирается функция для кулачков, связанная с прерываниями высокоскоростного счетчика. Если значение высокоскоростного счета равно любому значению уставки прерывания, то соответствующее прерывание высокоскоростного счетчика (например, метка прерывания 100-сегментного высокоскоростного счетчика) выполняется немедленно. При повторном изменении значения высокоскоростного счетчика одно и то же высокоскоростное прерывание кулачка может выполняться многократно.

Функция для кулачков, связанная с прерываниями высокоскоростного счетчика, не только может полностью реализовать функцию прерывания циклической последовательности стандартного электронного кулачка, но и может генерировать несколько раз положительное и отрицательное одноточечное прерывание в одном цикле. Широко применяется в системах управления высокоскоростными намоточными и упаковочными машинами.

Примечание: Функция для кулачков подходит только для режима абсолютного подсчета.

Функцию для кулачков можно настроить с помощью панели настройки в программном обеспечении ПЛК PROMPOWER или с помощью специального Flash-регистра SFD332:

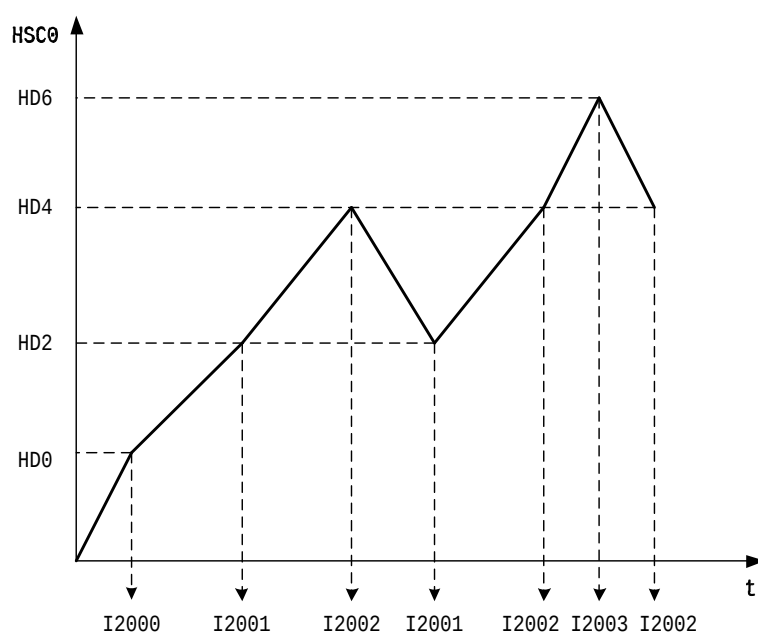
(Примечание: (Чтобы настройки вступили в силу, необходимо ВЫКЛЮЧИТЬ и снова ВКЛЮЧИТЬ привод.)

0: Функция для кулачков выключена

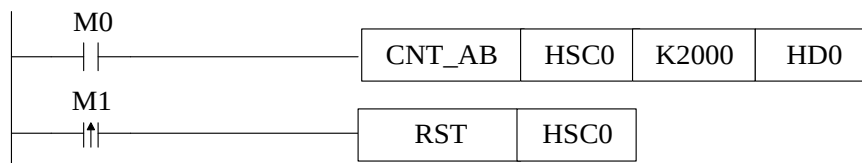
1: Функция для кулачков включена

Пример:

Четыре значения хранятся в четырех последовательных регистрах двойных слов, начиная с регистра HD0. Когда HSC0 начинает подсчет, если значение счета HSC0 равно любому из четырех регистров, то соответствующий сигнал прерывания будет сгенерирован немедленно. Это показано на рисунке ниже:



5.9.8 Прерывание с использованием примечаний и адреса параметра



LD M0 //Условие запуска HSC M0 (также условие подсчета прерываний)

CNT_AB HSC0 K2000 HD0 //Настройка адреса заголовка HSC и 100 сегментов

LDP M1 //Условие запуска сброса HSC

RST HSC0 //Сброс HSC и 100 сегментов (также сброс прерывания)

Как показано в приведенном выше примере (примечание: подпрограмма прерываний не выполняется, см. пример использования в главе 5-9-9). Регистр данных HD0 устанавливает начальный адрес области для значения уставки 100 сегментов, а затем сохраняет значение уставки 100 сегментов в виде двойного слова. Следует обратить внимание на выполнение прерываний высокоскоростного счетчика:

- Регистр после последнего сегмента не обязательно должен быть равен 0, но он должен быть зарезервирован и не может использоваться для других целей. Например, он имеет 3 сегмента, сегмент 1 — HD0, сегмент 2 — HD2, сегмент 3 — HD4, затем HD6 зарезервирован.
- Запрещено вводить значение уставки прерывания без написания программы прерываний. В противном случае, возникнут ошибки.
- Прерывания 100 сегментов высокоскоростного счетчика генерируются поочередно, то есть если первого прерывания не произошло, то и второго прерывания не произойдет.
- В процессе высокоскоростного счета, если текущее значение счета изменяется с помощью команды DMOV, ADD (DMOV K1000 HSCD0), то значение прерывания в это время не изменится. Не изменяйте значение HSCD во время работы высокоскоростного счетчика.

Некоторые параметры можно изменять в специальных Flash-регистрах, как показано в таблице ниже:

Параметр	Адрес регистра	Значение уставки
Режим счета	SFD330	0: относительный 1: абсолютный
Режим выполнения	SFD331	0: однократное выполнение 1: цикл прерываний
Функции для кулачков	SFD332	0: выключено 1: функция для кулачков включена

Вышеупомянутые параметры также можно настроить с помощью панели настройки следующим образом:

Наведите курсор мыши на команду высокоскоростного подсчета и щелкните по ней правой кнопкой мыши. В раскрывающемся меню выберите "CNT_AB Instruction Parameter Configuration" (Настройка параметров команды CNT_AB). Появится панель настройки для настройки параметров в этом окне. Это показано на рисунке ниже:

High Speed Count 24 Section Config

Single phase 100 segment high speed counting

High Speed C: HSC0

Compare Value: D10

Interrupt Address: HD0

☒ Opposite ☐ Absolute ☐ Circulate ☐ Cam

Config Value

Compare Value: 3000

Section Num: 1

Section Num	Value
Segment1 Count Num:	0

Read From PLC Write To PLC OK Cancel

5.9.9 Применение прерываний HSC

• Применение 1

Когда M0 находится во включенном состоянии, HSC0 начинает подсчет. Значение счета сохраняется по адресу, начинающемуся с HD0. При достижении значения уставки происходит прерывание. При появлении нарастающего фронта M1 выполните очистку HSC0.

Метод 1:

Настройте параметры с помощью программного обеспечения PROMPOWER PLC Studio.

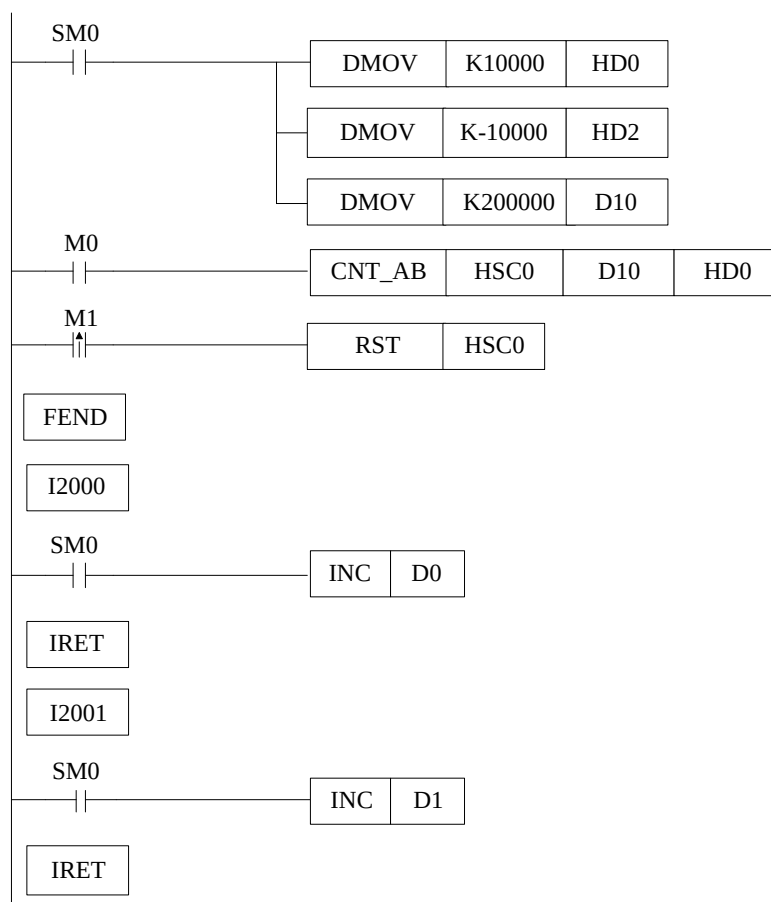
Section Num	Value
Segment1 Count Num:	10000
Segment2 Count Num:	-10000

Параметр настройки	Функция
High-speed counter (Высокоскоростной счетчик)	Выберите опцию HSC; диапазон выбора: от HSC0 до HSC18
Frequency (Частота)	Выберите удвоение частоты HSC (2 или 4)
Compare value (Значение для сравнения)	Значение может быть регистром или константой. В этом примере, когда значение счета достигает значения для сравнения, HSC0 включается. В нашем случае значение для сравнения равно 200000, и оно сохраняется в D10.

Relative and absolute (Относительный и абсолютный)	HSC работает в относительном режиме или абсолютном режиме.
Interrupt address (Адрес прерываний)	Начальные регистры для хранения уставок прерывания для 100 сегментов
Circulate (Цикличность)	Режим прерывания 100 сегментов циклический/не циклический
Cam (Кулачок)	Функция для кулачка выполняется, когда любое значение уставки прерывания 100-сегментного высокоскоростного счетчика равно значению счета.

Метод 2:

Создание программы



Команда:

LD	SM0	//SM0 — нормально-разомкнутая катушка
DMOV	K10000 HD0	//Уставка HD0 сегмента 1 — 10000
DMOV	K-10000 HD2	//Уставка HD2 сегмента 2 составляет -10000
DMOV	K200000 D10	//Установка значения для сравнения HSC
LD	M0	//Условие включения M0 для HSC

```

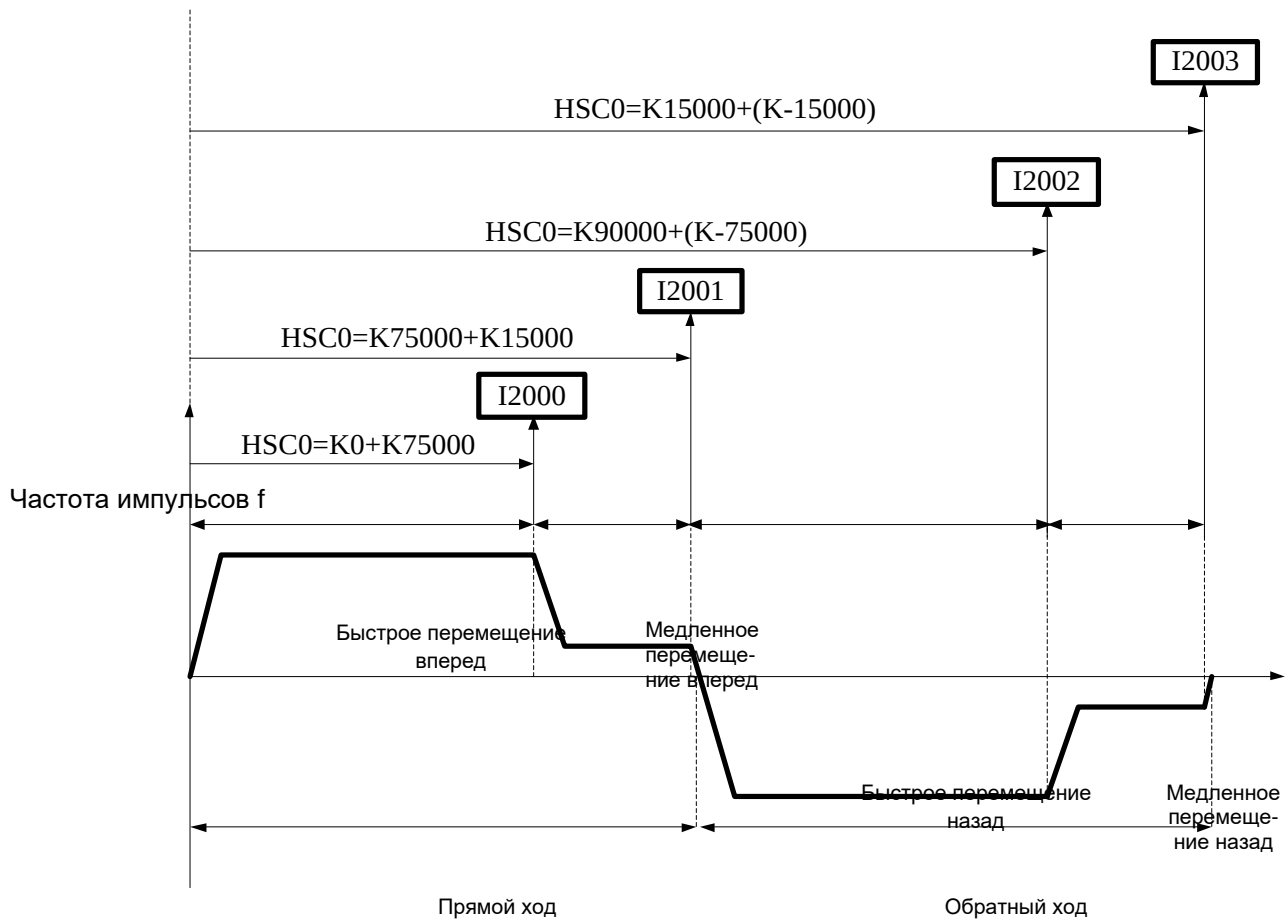
CNT_    AB HSC0 D10 HD0 //Команда прерывания HSC
LDP      M1                //Условие сброса M1 для HSC
RST      HSC0              //Сброс HSC и прерывание 100 сегментов
FEND
I2000    //Признак прерывания сегмента 1
LD       SM0              //SM0 — нормально-разомкнутая катушка
INC      D0                //D0 = D0 + 1
IRET
I2001    //Признак прерывания сегмента 2
LD       SM0              //SM0 — нормально-разомкнутая катушка
INC      D1                //D1 = D1 + 1
IRET      //Признак возврата прерывания

```

- **Применение 2: Вязальная ткацкая машина (режим непрерывного цикла)**

Принцип работы машины: Управление инвертором через ПЛК, тем самым обеспечивается управление двигателем. Посредством сигнала обратной связи от энкодера осуществляется управление вязальной ткацкой машиной и ее точное позиционирование.





Ниже описана программа ПЛК:

Y2 — выходной сигнал прямого хода;

Y3 — выходной сигнал обратного хода;

Y4 — выходной сигнал скорости 1.

HSC2: Счетчик количества перемещений вперед/назад;

HSC0: HSC в режиме фаз A и B.

High Speed Count 24 Section Config

AB phase 100 segment high speed counting

High Speed C: HSC0

Frequency: 4

Compare Value: D0

Interrupt Address: D100

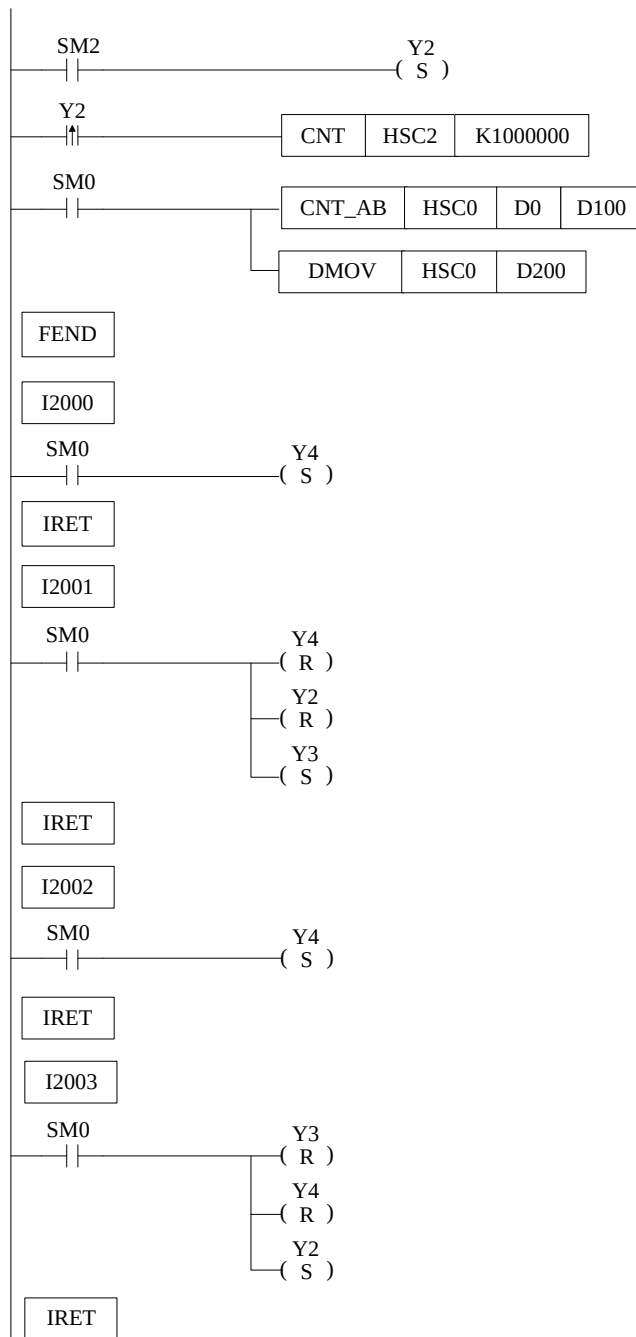
☒ Opposite ☐ Absolute ☐ Circulate ☐ Cam

Config Value

Compare Value: 1000000

Section Num: 4

Section Num	Value
Segment1 Count Num:	75000
Segment2 Count Num:	15000
Segment3 Count Num:	-75000
Segment4 Count Num:	-15000



Перечень команд:

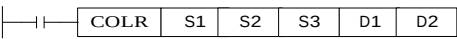
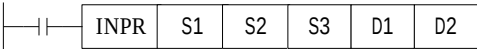
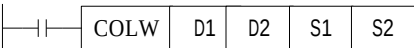
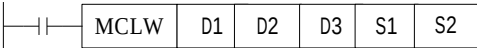
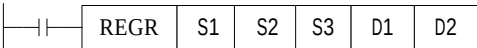
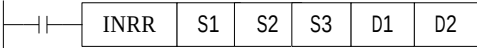
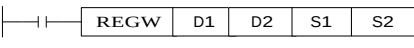
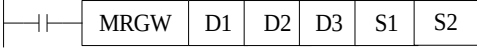
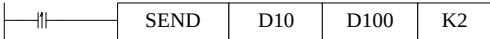
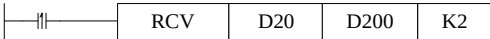
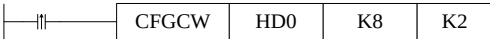
LD	SM2	//SM2 — первое включение катушки
SET	Y2	//Включение Y2 (прямой ход)
LDP	Y2	// Условие включения Y2 счетчика количества перемещений вперед/назад
CNT	HSC2 K1000000	//HSC2 начинает подсчет
LD	SM0	//SM000 — нормально-разомкнутая катушка
CNT_	AB HSC0 D0 D100	//Первый адрес 100 сегментов HSC
DMOV	HSC0 D200	//Считывание значения счета HSC0 в D200

FEND		//Конец главной программы
I2000		//Признак прерывания 1
LD	SM0	//SM0 — нормально-разомкнутая катушка
SET	Y4	//Включение Y4 (работа со скоростью 1)
IRET		//Возврат прерывания
I2001		//Признак прерывания 2
LD	SM0	//SM0 — нормально-разомкнутая катушка
RST	Y4	//Сброс Y4 (прекращение работы со скоростью 1)
RST	Y2	//Сброс Y2 (прекращение прямого хода)
SET	Y3	//Включение Y3 (обратный ход)
IRET		//Возврат прерывания
I2002		//Признак прерывания 3
LD	SM0	//SM0 — нормально-разомкнутая катушка
SET	Y4	//Включение Y4 (работа со скоростью 1)
IRET		//Возврат прерывания
I2003		//Признак прерывания 4
LD	SM0	//SM0 — нормально-разомкнутая катушка
RST	Y3	//Сброс Y3 (прекращение обратного хода)
RST	Y4	//Сброс Y4 (прекращение работы на малой скорости)
SET	Y2	//Включение Y2 (прямой ход)
IRET		//Возврат прерывания

6. Функции связи

В этой главе рассматриваются: основной принцип связи, связь по протоколу Modbus и связь в свободном формате.

Команды относительного режима

Мнемонический код	Функция	Схемные и программные компоненты	Глава
Связь по протоколу MODBUS			
COLR	Считывание данных с катушки		6-2-3
INPR	Считывание данных с входной катушки		6-2-3
COLW	Запись данных в отдельную катушку		6-2-3
MCLW	Запись данных нескольких катушек		6-2-3
REGR	Считывание данных из регистра		6-2-3
INRR	Считывание данных из входного регистра		6-2-3
REGW	Запись данных в отдельный регистр		6-2-3
MRGW	Запись данных нескольких регистров		6-2-3
Связь в свободном формате			
SEND	Передача данных		6-3-4
RCV	Прием данных		6-3-4
Считывание и запись данных последовательного порта			
CFGCR	Считывание данных последовательного порта		6-5-1
CFGCW	Запись данных последовательного порта		6-5-2



6.1 Обзор

Главные блоки ПЛК серии PMP20 отвечают требованиям к связи и сетям. Они поддерживают не только Modbus RTU, но также Modbus ASCII и полевую шину X-NET. ПЛК серии PMP20 обеспечивают несколько методов связи, с помощью которых можно взаимодействовать с устройствами (например, принтерами, контрольно-измерительными приборами и т. д.), поддерживающими протокол связи Modbus.

6.2 COM-порт

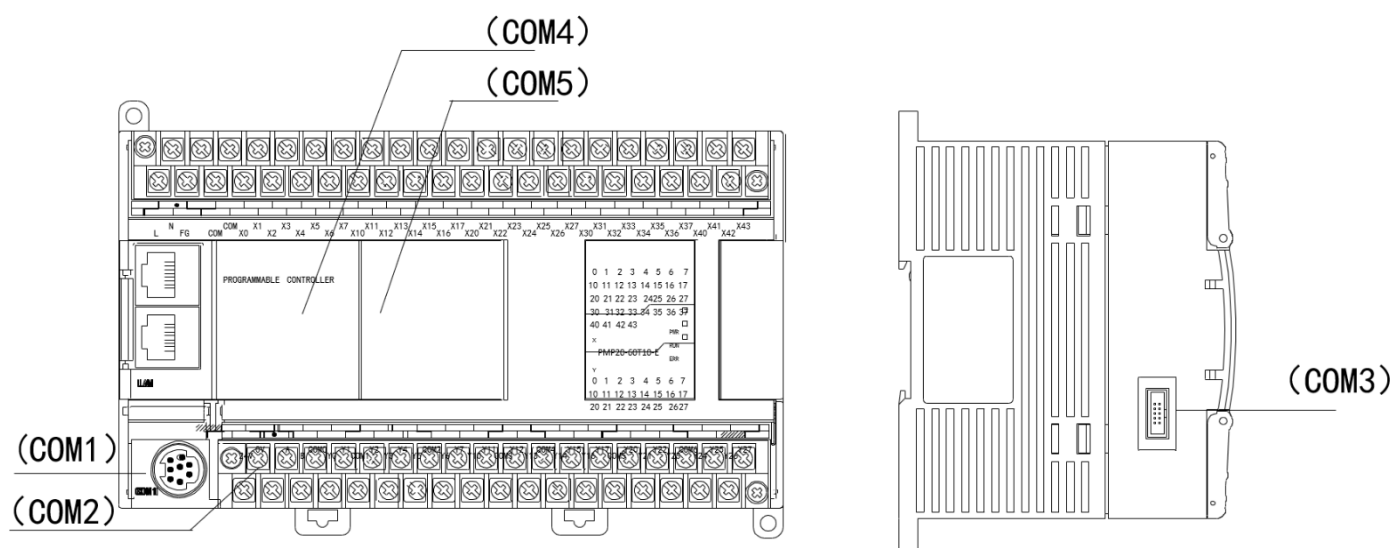
COM-порт

	USB	RJ45	COM0	COM1	COM2- RS232	COM2- RS485	COM3	COM4	COM5
PMP20	×	✓	×	✓	×	✓	✓	✓	✓

Порты связи ПЛК серии PMP20 распределяются следующим образом:

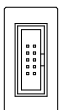
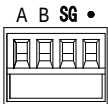
ПЛК серии PMP20 имеют несколько портов связи, например, порт Ethernet, порт RS232, порт RS485.

× не поддерживается ✓ поддерживается



Примечание: С левой стороны выходной клеммной колодки ПЛК серии PMP20 расположен порт RS232.

Названия и функции портов связи:

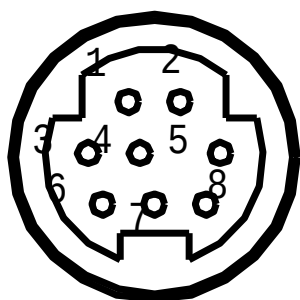
Порт	Внешний вид	Название	Протокол	Функция
COM1		Порт RS232	Modbus RTU Modbus ASCII Связь в свободном формате X-NET	Загрузите программу и подключите внешние устройства, настройте параметры порта через программное обеспечение или XNet Config Tool.
COM2- RS485	Порты A, B	Порт RS485	Modbus RTU Modbus ASCII Связь в свободном формате X-NET	Загрузите программу и подключите внешние устройства, настройте параметры порта через программное обеспечение или XNet Config Tool.
RJ45		Порт Ethernet	TCP/IP связь на основе Ethernet	Высокоскоростная стабильная загрузка/выгрузка программ и данных, удаленный мониторинг, связь с устройством TCP IP в локальной сети, настройка параметров порта с помощью программного обеспечения или XNet Config Tool
COM3		Левый порт расширения ED (для расширения порта RS232/RS485)	Modbus RTU Modbus ASCII Связь в свободном формате X-NET	Подключите внешние устройства, настройте параметры порта через программное обеспечение или XNet Config Tool.
COM4		Верхний порт расширения	Modbus RTU Modbus ASCII Связь в свободном формате X-NET	Подключите внешние устройства, настройте параметры порта через программное обеспечение или XNet Config Tool.
COM5		BD/RS232/RS485/ Порт для волоконно-оптических кабелей		

		(См. подробное описание ниже).		
--	--	--------------------------------	--	--

Примечание:

- (1) Если после изменения параметров связь между портом COM1 и ПК отсутствует, нажмите [stop PLC when reboot (остановить ПЛК при перезагрузке)] в программном обеспечении, а затем снова включите питание, чтобы устранить проблему; по возможности, лучше не изменять параметры связи COM1.
- (2) Функция связи по шине X-NET не описана в этом руководстве; см. руководство пользователя X-NET.
- (3) Связь по Ethernet не описана в настоящем руководстве; см. руководство пользователя для получения информации о связи по протоколу TCP/IP на основе Ethernet.

1. Порт RS232 (COM1)



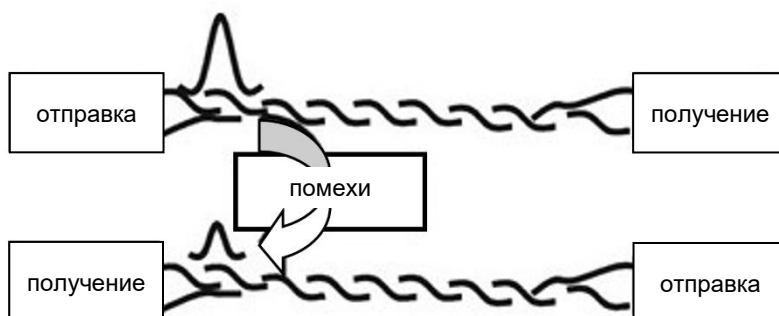
4: RxD
5: TxD
8: GND

8-контактная вилка типа Mini Din
(отверстия)

2. Порт RS485 (COM2)

Что касается порта RS485, то A — это сигнал "+", а B — сигнал "-".

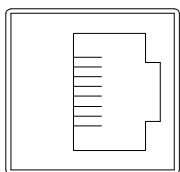
Для порта RS485 используйте кабель типа "витая пара". (см. схему ниже). Предпочтительнее использовать экранированный кабель "витая пара", один конец которого подключается к заземлению.



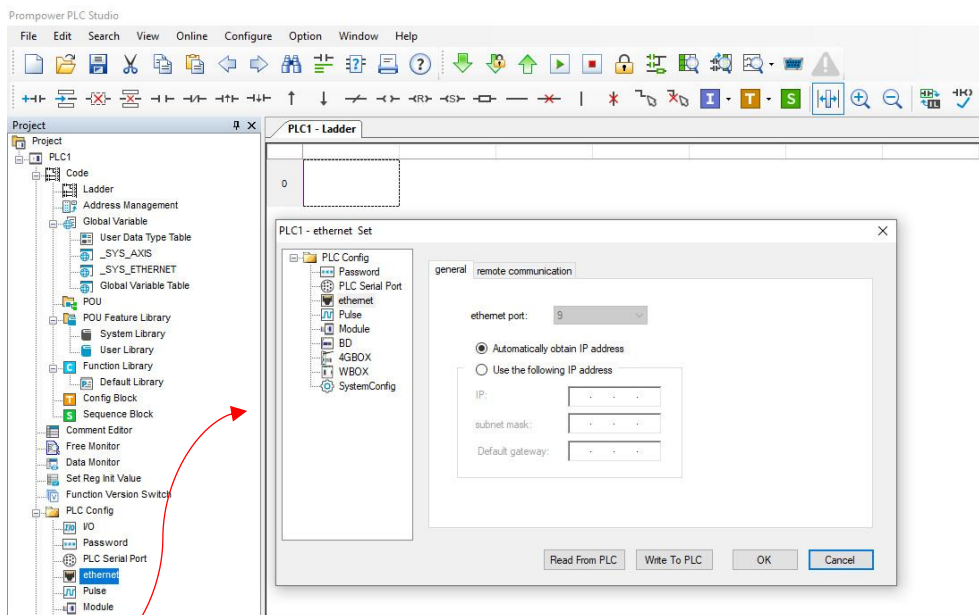
3. Порт Ethernet (RJ45)

Порт RJ45 есть только у ПЛК с поддержкой Ethernet. Этот порт поддерживает связь по протоколу TCP/IP Ethernet, он быстрее и устойчивее работает, чем USB-порт, обеспечивает более эффективный контроль данных в реальном времени, загрузка и выгрузка программ происходят быстрее. Сам режим связи через порт Ethernet имеет очевидные преимущества по сравнению с RS485 и USB. Во многих случаях пользователи могут взаимодействовать с любым ПЛК на месте эксплуатации с помощью всего лишь одного коммутатора.

Помимо применения в локальных сетях, порт Ethernet также поддерживает удаленный поиск, контроль и управление ПЛК, функции загрузки и связь с другими устройствами TCP/IP в сети через Интернет.



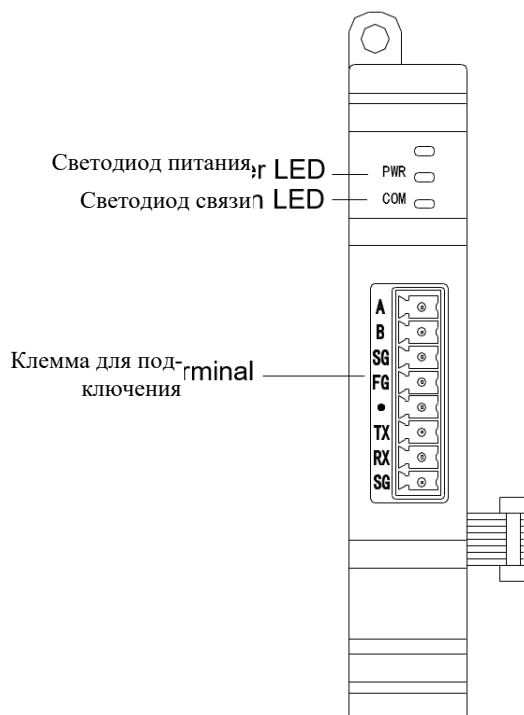
Порт RJ45 можно настроить через опцию "PLC Config-Ethernet" в программном обеспечении PROMPOWER PLC Studio или с помощью XNet Config Tool. Подробную информацию см. в соответствующем руководстве.



4. Левый порт расширения ED (COM3)

Левый порт расширения ED может использоваться для подключения платы ED при расширении портов RS232 и RS485. Платы ED бывают таких моделей как PMP-NES-ED (может расширить один порт RS232 и один порт RS485, но они не могут одновременно обеспечивать обмен данными).

PMP-NES-ED



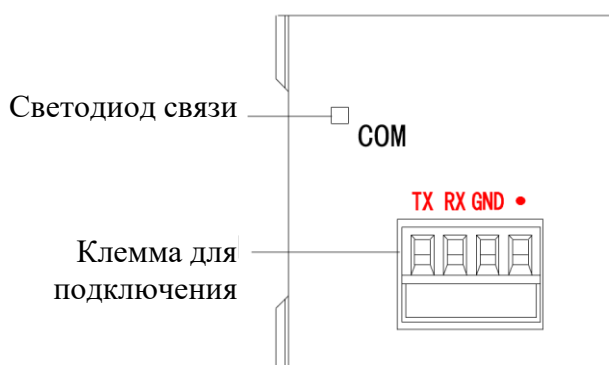
Название		Функция
Светодиод питания		Светодиод горит, когда включен модуль ED
Светодиод связи		Светодиод горит при нормальной связи с модулем ED
Клемма для подключения	A	RS485+
	B	RS485-
	SG	Заземление
	FG	Подключение к клемме заземления
	-	Не задействована
	TX	Передача данных через RS232
	RX	Прием данных через RS232

5. Верхний порт расширения BD (COM4, COM5)

Через верхний порт расширения можно подключать плату BD, которая поддерживает режим RS232 (PMP-NS-BD), режим RS485 (PMP-NE-BD) и режим волоконно-оптических кабелей (PMP-NO-BD).

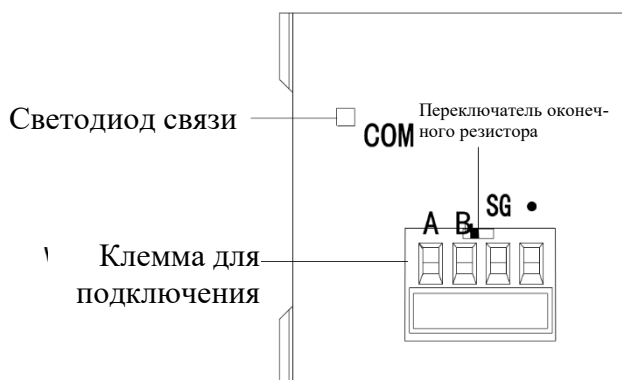
ПЛК серии PMP20 с 24/32 входами/выходами поддерживают подключение дополнительной платы BD, ПЛК серии PMP20 с 48/60 входами/выходами поддерживают подключение 2 дополнительных плат BD, ПЛК серии PMP20 с 16 входами/выходами не поддерживают подключение дополнительных плат BD.

(1) PMP-NS-BD



Название		Функция
Светодиод связи		Эта функция не поддерживается
Клемма для подключения	TX	Передача сигналов
	RX	Прием сигналов
	GND	Заземление
	•	Не задействована

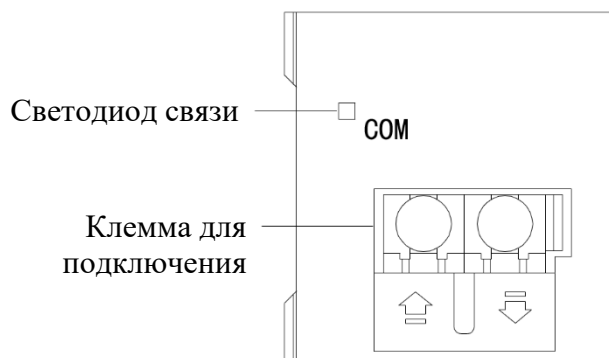
(2) PMP-NE-BD



Название		Функция
Светодиод связи		Светодиод мигает, когда связь с платой BD установлена успешно
Клемма для подключения	A	485+
	B	485-
	S	Земля логических сигналов
	•	Не задействована
Переключатель оконечного резистора		Используется для выбора оконечного резистора (120 Ом)

На плате PMP-NE-BD есть переключатель оконечного резистора. По умолчанию переключатель установлен в положение OFF (ВЫКЛ), что означает, что оконечный резистор не установлен. Если PMP-NE-BD находится в начале или конце шины, необходимо установить оконечный резистор сопротивлением 120 Ом с обеих сторон и включить переключатель (справа).

(3) PMP-NS-BD



Название		Функция
Светодиод связи		Эта функция не поддерживается
Клемма для подключения		С левой стороны — входная клемма сигналов, с правой стороны — выходная клемма сигналов

6.2.1 Параметры связи

Параметры связи

Станция	Количество станций Modbus: 1-254
Скорость передачи данных	300 бит/с-9 Мбит/с
Бит данных	5, 6, 7, 8, 9
Стоповый бит	1, 1,5, 2
Четность	положительная четность, отрицательная четность, положительная четность, не задействована, маска

Параметры по умолчанию: количество станций — 1, скорость передачи данных — 19200 бит/с, 8 битов данных, 1 стоповый бит, положительная четность.

Существует много способов настройки параметров порта связи ПЛК:

Параметры связи через Modbus можно настроить двумя способами: (1) Настройка параметров с помощью программного обеспечения для программирования; (2) настройка параметров с помощью XNet Config Tool; подробное описание см. в главе 6-2-6.

Параметры связи в свободном формате можно задать с помощью программного обеспечения для программирования; подробное описание см. в главе 6-3-2.

Параметры связи по шине X-NET можно настроить с помощью XNet Config Tool. Подробную информацию см. в руководстве по полевой шине X-NET.

Примечание:

Что касается клемм А, В на корпусе ПЛК, скорость передачи данных 1 Мбит/с и выше подходит только для режима связи через X-NET.

6.3 Связь по протоколу MODBUS

6.3.1 Обзор функции

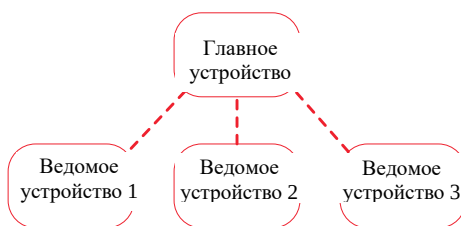
ПЛК серии РМР20 могут быть как главным устройством Modbus, так и ведомым устройством Modbus.

Режим главного устройства: Если ПЛК является главным устройством, он может взаимодействовать с другими ведомыми устройствами, имеющими протокол MODBUS-RTU или MODBUS-ASCII, посредством команд Modbus; ПЛК также может обмениваться данными с другими устройствами.

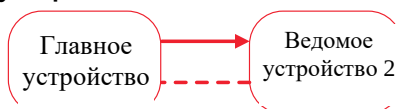
Пример: ПЛК серии РМР20 может управлять инвертором по протоколу Modbus.

Режим ведомого устройства: Если ПЛК является ведомым устройством, он может взаимодействовать только с другими главными устройствами.

Главное устройство и ведомое устройство: В сети RS485 одновременно может быть одно главное устройство и несколько ведомых устройств (см. схему ниже). Главная станция может считывать и записывать данные с любой ведомой станции. Две ведомые станции не могут обмениваться данными друг с другом. Главная станция должна записывать программу и считывать или записывать данные с одной ведомой станции; ведомая станция не имеет программы, а только отвечает главной станции (схема подключения: подключены все 485+, подключены все 485-).



В сети RS232 (см. схему ниже) одновременно может быть только одно главное устройство и одно ведомое устройство.



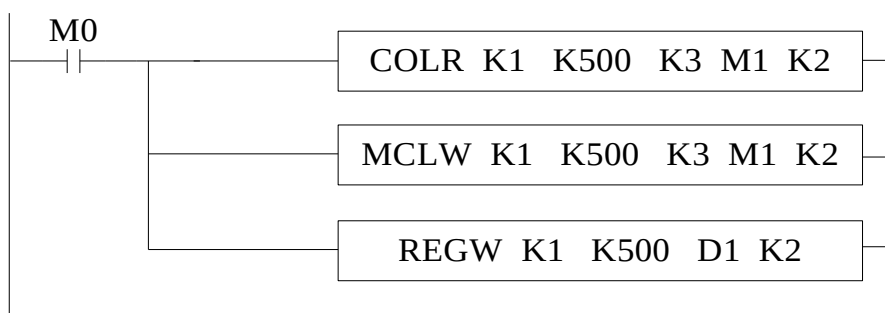
На схему нанесена пунктирная линия. Она означает, что любой ПЛК может быть главной станцией, когда все ПЛК в сети не отправляют данные. Поскольку ПЛК не имеют единого стандарта синхронизации, связь будет нарушена, если два или более ПЛК одновременно отправляют данные. Такой режим не рекомендуется.

Примечание:

1. В ПЛК серии PMP20 порты RS232 и RS485 поддерживают только полудуплексный режим.
2. В ПЛК серии PMP20 добавляется некоторое время ожидания перед следующим обменом данными, что означает, что ведомый ПЛК примет следующие данные только через некоторое время после завершения приема предыдущих данных.

6.3.2 Команды для Modbus

Пользователи ПЛК серии PMP20 могут писать команды для Modbus прямо в программе, протокольная станция будет ставить в очередь запросы Modbus, что отличается от задачи связи. Это означает, что пользователи могут использовать одно условие запуска для одновременного запуска нескольких команд Modbus. ПЛК поставит в очередь запросы Modbus в соответствии со протокольной станцией.



Примечание: В блоке последовательности ПЛК серии PMP20 отменены команды связи по Modbus и заменены на текущий режим обработки команд Modbus.

6.3.3 Адрес связи по протоколу Modbus

Код программного компонента в ПЛК соответствует идентификационному номеру Modbus; см. таблицу ниже:

Таблица адресов Modbus и внутренних программных компонентов ПЛК серии PMP20:

Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
Бит ка- тушки	М	М0-М20479	20480	0~4FFF	0~20479
	Х	X0-X77 (основной блок)	64	5000~503F	20480~20543
		X10000-X10077 (модуль №1)	64	5100~513F	20736~20799
		X10100~X10177 (модуль №2)	64	5140~517F	20800~20863
		X10200~X10277 (модуль №3)	64	5180~51BF	20864~20927
		X10300~X10377 (модуль №4)	64	51C0~51FF	20928~20991
		X10400~X10477 (модуль №5)	64	5200~523F	20992~21055
		X10500~X10577 (модуль №6)	64	5240~527F	21056~21119
		X10600~X10677 (модуль №7)	64	5280~52BF	21120~21183
		X10700~X10777 (модуль №8)	64	52C0~52FF	21184~21247

Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
		X11000~X11077 (модуль №9)	64	5300~533F	21248~21311
		X11100~X11177 (модуль №10)	64	5340~537F	21312~21375
		X11200~X11277 (модуль №11)	64	5380~53BF	21376~21439
		X11300~X11377 (модуль №12)	64	53C0~53FF	21440~21503
		X11400~X11477 (модуль №13)	64	5400~543F	21504~21567
		X11500~X11577 (модуль №14)	64	5440~547F	21568~21631
		X11600~X11677 (модуль №15)	64	5480~54BF	21632~21695
		X11700~X11777 (модуль №16)	64	54C0~54FF	21696~21759
		X20000~X20077 (плата BD №1)	64	58D0~590F	22736~22799
	Y	Y0~77 (основной блок)	64	6000~603F	24576~24639
		Y10000~Y10077 (модуль №1)	640	6100~613F	24832~24895
		Y10100~Y10177 (модуль №2)	64	6140~617F	24896~24959

Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
		Y10200~Y10277 (модуль №3)	64	6180~61BF	24960~25023
		Y10300~Y10377 (модуль №4)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (модуль №5)	64	6200~623F	25088~25151
		Y10500~Y10577 (модуль №6)	64	6240~627F	25152~25215
		Y10600~Y10677 (модуль №7)	64	6280~62BF	25216~25279
		Y10700~Y10777 (модуль №8)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (модуль №9)	64	6300~633F	25344~25407
		Y11100~Y11177 (модуль №10)	64	6340~637F	25408~25471
		Y11200~Y11277 (модуль №11)	64	6380~63BF	25472~25535
		Y11300~Y11377 (модуль №12)	64	63C0~63FF	25536~25599
		Y11400~Y11477 (модуль №13)	64	6400~643F	25600~25663
		Y11500~Y11577 (модуль №14)	64	6440~647F	25664~25727

Тип	Компо- нент	Адрес	Номера	Адрес Modbus (шестнадцатерич- ный)	Адрес Modbus (десятичный)
		Y11600~Y11677 (модуль №15)	64	6480~64BF	25728~25791
		Y11700~Y11777 (модуль №16)	64	64C0~64FF	25792~25855
		Y20000~Y20077 (плата BD №1)	64	68D0~690F	26832~26895
	S	S0~S7999	8000	7000~8F3F	28672~36671
	SM	SM0~SM4095	4096	9000~9FFF	36864~40959
	T	T0~T4095	4096	A000~AFFF	40960~45055
	C	C0~C4095	4096	B000~BFFF	45056~45151
	ET	ET0~ET39	40	C000~C027	49152~49191
	SEM	SEM0~SEM127	128	C080~C0FF	49280~49407
	HM ^{*1}	HM0~HM6143	6144	C100~D8FF	49408~55551
	HS ^{*1}	HS0~HS999	1000	D900~DCEF	55552~56551
	HT ^{*1}	HT0~HT1023	1024	E100~E4FF	57600~58623
	HC ^{*1}	HC0~HC1023	1024	E500~E8FF	58624~59647
	HSC ^{*1}	HSC0~HSC36	40	E900~E927	59648~59687
Слово реги- стра	D	D0~D20479	20480	0~4FFF	0~20479
	ID	ID0~ID99 (основной блок)	100	5000~5063	20480~20579



Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
		ID10000-ID10099 (модуль №1)	100	5100~5163	20736~20835
		ID10100-ID10199 (модуль №2)	100	5164~51C7	20836~20935
		ID10200-ID10299 (модуль №3)	100	51C8~522B	20936~21035
		ID10300-ID10399 (модуль №4)	100	522C~528F	21036~21135
		ID10400-ID10499 (модуль №5)	100	5290~52F3	21136~21235
		ID10500-ID10599 (модуль №6)	100	52F4~5357	21236~21335
		ID10600-ID10699 (модуль №7)	100	5358~53BB	21336~21435
		ID10700-ID10799 (модуль №8)	100	53BC~541F	21436~21535
		ID10800-ID10899 (модуль №9)	100	5420~5483	21536~21635
		ID10900-ID10999 (модуль №10)	100	5484~54E7	21636~21735
		ID11000-ID11099 (модуль №11)	100	54E8~554B	21736~21835
		ID11100-ID11199 (модуль №12)	100	554C~55AF	21836~21935

Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
		ID11200-ID11299 (модуль №13)	100	55B0~5613	21936~22035
		ID11300-ID11399 (модуль №14)	100	5614~5677	22036~22135
		ID11400-ID11499 (модуль №15)	100	5678~56DB	22136~22235
		ID11500-ID11599 (модуль №16)	100	56DC~573F	22236~22335
		ID20000-ID20099 (плата BD №1)	100	58D0~5933	22736~22835
	QD	QD0-QD99 (основной блок)	100	6000~6063	24576~24675
		QD10000-QD10099 (модуль №1)	100	6100~6163	24832~24931
		QD10100-QD10199 (модуль №2)	100	6164~61C7	24932~25031
		QD10200-QD10299 (модуль №3)	100	61C8~622B	25032~25131
		QD10300-QD10399 (модуль №4)	100	622C~628F	25132~25231
		QD10400-QD10499 (модуль №5)	100	6290~62F3	25232~25331
		QD10500-QD10599 (модуль №6)	100	62F4~6357	25332~25431

Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
		QD10600-QD10699 (модуль №7)	100	6358~63BB	25432~25531
		QD10700-QD10799 (модуль №8)	100	63BC-641F	25532~25631
		QD10800-QD10899 (модуль №9)	100	6420~6483	25632~25731
		QD10900-QD10999 (модуль №10)	100	6484~64E7	25732~25831
		QD11000-QD11099 (модуль №11)	100	64E8~654B	25832~25931
		QD11100-QD11199 (модуль №12)	100	654C-65AF	25932~26031
		QD11200-QD11299 (модуль №13)	100	65B0~6613	26032~26131
		QD11300-QD11399 (модуль №14)	100	6614~6677	26132~26231
		QD11400-QD11499 (модуль №15)	100	6678~66DB	26232~26331
		QD11500-QD11599 (модуль №16)	100	66DC-673F	26332~26431
		QD20000~QD20099 (плата BD №1)	100	68D0~6933	26832~26931
	SD	SD0-SD4095	4096	7000~7FFF	28672~32767

Тип	Компонент	Адрес	Номера	Адрес Modbus (шестнадцатеричный)	Адрес Modbus (десятичный)
	TD	TD0-TD4095	4096	8000~8FFF	32768~36863
	CD	CD0-CD4095	4096	9000~9FFF	36864~40959
	БДТ	ETD0-ETD39	40	A000-A027	40960~40999
	HD ^{※1}	HD0-HD6143	6144	A080-B87F	41088~47231
	HSD ^{※1}	HSD0-HSD1023	1024	B880-BC7F	47232~48255
	HTD ^{※1}	HTD0-HTD1023	1024	BC80-C07F	48256~49279
	HCD ^{※1}	HCD0-HCD1023	1024	C080-C47F	49280~40303
	HSCD ^{※1}	HSCD0-HSCD39	40	C480-C4A7	50304~50343
	FD ^{※2}	FD0-FD8191	8192	C4C0-E4BF	50368~58559
	SFD ^{※2}	SFD0-SFD5999	6000	E4C0-FC2F	58560~64559
	FS ^{※2}	FS0-FS47	48	F4C0-F4EF	62656~62703

Примечание:

1: Область сохранения данных после выключения питания отмечена знаком ※1, а область flash-регистра отмечена знаком ※2.

2: Адрес в таблице выше используется, когда ПЛК является нижним компьютером, и для связи используется протокол Modbus RTU или MODBUS ASCII, а общий верхний компьютер: SCADA/ЧМИ/ПЛК.

3: Если верхний компьютер — ПЛК, программируйте в соответствии с протоколом Modbus RTU или MODBUS ASCII.

4: Если верхний компьютер — SCADA или ЧМИ, возможны две ситуации: в первом случае установлен драйвер PROMPOWER. Программу можно написать напрямую, используя внутренние программные компоненты ПЛК (Y0 / M0). Во

втором случае выбирают Modbus RTU или Modbus ASCII, если драйвер PROMPOWER отсутствует, а затем используются адреса в таблице выше для задания переменных данных.

5: Точка ввода и точка вывода являются восьмеричными. Рассчитайте восьмеричный адрес MODBUS для точки ввода и точки вывода, например: MODBUS, соответствующий Y0, имеет адрес H6000, Modbus, соответствующий Y10, имеет адрес H6008 (не H6010), а Modbus, соответствующий Y20, имеет адрес H6010 (не H6020).

6: Если адрес Modbus превышает 32767, его необходимо выразить в шестнадцатеричном формате, добавив перед адресом «0». Пример: Адрес MODBUS HD0 равен 41088 в десятичном формате (больше 32767), и 41088 нельзя записать в программное обеспечение, поэтому его необходимо выразить в шестнадцатеричном формате как H0A080.

7: Расчет адреса Modbus для X и Y. Взяв в качестве примера X, расчет адреса Modbus для Y выполняется так же как для X.

X0: 20480 X10: 20480 + 8 X20: 20480 + 16 X30: 16384 + 24...

X10000: 20736 X10010: 20736 + 8 X10020: 20736 + 16...

X10200: 20800 X10210: 20800 + 8 X10220: 20800 + 16...

6.3.4 Формат данных Modbus

Режим передачи данных по протоколу Modbus:

Существуют два режима передачи данных: RTU и ASCII. Режим определяет последовательную передачу битового содержимого в домене сообщений, метод упаковки и расшифровки информации. Режим передачи данных (и параметры порта) для всех устройств в последовательных каналах Modbus должен быть один и тот же.

Структура данных Modbus-RTU

1. Режим RTU

В режиме Modbus RTU (удалённый терминал) сообщение содержит два 4-битных шестнадцатеричных символа в каждом 8-битном байте. Этот режим имеет очень высокую плотность данных и более высокую пропускную способность,

чем Modbus ASCII. Каждое сообщение должно отправляться в виде непрерывного потока символов. Область проверки кадра в режиме RTU: циклический контроль избыточности (CRC).

Описание кадра режима RTU:

Станция Modbus	Код функции	Данные	CRC	
1 байт	1 байт	0-252 байта	2 байта	
			Младший байт CRC	Старший байт CRC

Формат:

START	Входной сигнал отсутствует ≥ 10 мс
Address (station no.)	Адрес связи: 8-битный двоичный адрес
Function	Код функции: 8-битный двоичный адрес
DATA (n - 1)	Содержимое данных: N*8-битных данных, $N \leq 8$, макс. 8 байтов
.....	
DATA 0	
CRC CHK Low	Код контроля CRC
CRC CHK High	16-битный код контроля CRC состоит из двух 8-битных двоичных комбинаций
END	Входной сигнал отсутствует ≥ 10 мс

2. Адрес Modbus

00H: Все ПЛК PROMPOWER обеспечивают широковещательную передачу — ведомые станции не отвечают.

01H: Связь с адресом 01H ПЛК.

0FH: Связь с адресом 15H ПЛК.

10H: Связь с адресом 16H ПЛК, и т.д. До 254 (FEH).

3. Функции и данные

Код функции	Функция	Команды для Modbus
01H	Считывание данных с катушки	COLR
02H	Считывание данных с входной катушки	INPR (ПЛК PROMPOWER не поддерживается)
03H	Считывание данных из регистра	REGR
04H	Считывание данных из входного регистра	INRR
05H	Запись данных в катушку	COLW
06H	Запись данных в регистр	REGW
10H	Запись данных нескольких регистров	MRGW
0FH	Запись данных нескольких катушек	MCLW

(1) Возьмем в качестве примера код функции 06H (запись данных в отдельный регистр) и представим формат данных.

Например: верхний компьютер записывает данные в H0002 (D2) ПЛК.

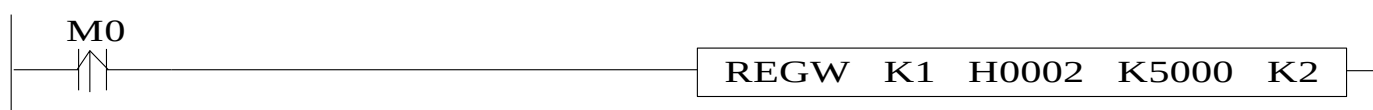
Режим RTU:

Формат запроса		Формат ответа	
Идентификатор	01H	Идентификатор	01H
Код функции	06H	Код функции	06H
Идентификатор регистра	00H	Идентификатор регистра	00H
	02H		02H
Содержимое данных	13H	Содержимое данных	13H
	88H		88H
CRC CHECK High	25H	CRC CHECK High	25H
CRC CHECK Low	5CH	CRC CHECK Low	5CH

Пояснение:

1. Address (Адрес) — это номер станции ПЛК.
2. Function code (Код функции) — это код чтения/записи протокола Modbus-RTU.
3. Register address (Адрес регистра) — это адрес Modbus ПЛК, см. главу 6-2-3.
4. Data content (Содержимое данных) — это значение в D2.
5. CRC CHECK High / CRC CHECK Low — это старший и младший бит в значении контроля CRC.

Если два ПЛК серии PMP20 взаимодействуют друг с другом, запишите K5000 в D2.



M0 — условие запуска (нарастающий фронт). Если связь не удалось установить, команда повторит попытку дважды. Если и на третий раз связь не установлена, связь завершается.

Связь между протоколом REGW и Modbus RTU (остальные команды те же).

REGW	Код функции 06H
K1	Номер станции
H0002	Адрес Modbus
K5000	Содержимое данных 1388H
K2	Последовательный порт ПЛК

Полные данные о связи:

01H 06H 00H 02H 13H 88H (система автоматически выполняет проверку CRC).

Если контролировать данные последовательного порта 2 с помощью инструмента отладки последовательного порта, то данные будут следующими:

01 06 00 02 13 88 25 5C.

Примечание: команда не различает десятичные, шестнадцатеричные, двоичные, восьмеричные значения и т. д. Например, B10000, K16 и H10 — это одно и то же значение, поэтому представленные ниже команды одинаковы.

REGW K1 B111110100 D1 K2

REGW K1 K500 D1 K2

REGW K1 H1F4 D1 K2

(2) Код функции 01H/02H: Считывание данных с катушки/Считывание данных с входной катушки

Например, считывание адреса катушки 6000H (Y0). При этом Y0 и Y1 включены.

Режим RTU:

Формат запроса		Формат ответа	
Адрес	01H	Адрес	01H
Код функции	01H/02H	Код функции	01H/02H
Адрес катушки	60H	Номера байтов	01H
	00H		
Номер катушек	00H	Содержимое данных	03H
	02H		
CRC CHECK Low	A3H	CRC CHECK Low	11H
CRC CHECK High	CBH	CRC CHECK High	89H

Поскольку статус Y0 и Y1 — ON (ВКЛ), содержимое данных — 03H (0000 0011).

(3) Код функции 03H: Считывание данных из регистра

Например, считывание данных из двух регистров, начиная с 03E8H (D1000, D1001).

Режим RTU:

Формат запроса		Формат ответа	
Адрес	01H	Адрес	01H

Код функции	03H	Код функции	03H
Адрес регистра	03H	Номера байтов	04H
	E8H		
Номер регистра	00H	Содержимое данных	12H
			2EH
	02H		04H
			E8H
CRC CHECK Low	44H	CRC CHECK Low	9DH
CRC CHECK High	7BH	CRC CHECK High	CCH

При этом данные, считанные из D1000 и D1001, представляют собой 122EH (4654) и 04E8H (1256).

(4) Код функции 05H: Запись данных в отдельную катушку

Например, задание адреса катушки 6000H (Y0).

Режим RTU:

Формат запроса		Формат ответа	
Адрес	01H	Адрес	01H
Код функции	05H	Код функции	05H
Адрес катушки	60H	Адрес катушки	60H
	00H		00H
Содержимое данных (сначала младший байт, затем старший байт)	FFH	Содержимое данных	FFH
	00H		00H
CRC CHECK Low	92H	CRC CHECK Low	92H

CRC CHECK High	3AH	CRC CHECK High	3AH
----------------	-----	----------------	-----

Примечание: При записи данных в отдельную катушку статусу ON соответствует 00FFH, статусу OFF соответствует 0000H; младший байт находится перед старшим байтом содержимого данных.

(5) Код функции 0FH: Запись данных нескольких катушек

Например, запись данных 16 катушек начинается с адреса 6000H (Y0).

Режим RTU:

Формат запроса		Формат ответа	
Адрес	01H	Адрес	01H
Код функции	0FH	Код функции	0FH
Адрес катушки	60H	Адрес катушки	60H
	00H		00H
Номер катушек	00H	Номер катушек	00H
	10H		10H
Номера байтов	02H	-	-
Содержимое данных (сначала младший байт, затем старший байт)	03H		
	01H		
CRC CHECK Low	43H	CRC CHECK Low	4AH
CRC CHECK High	16H	CRC CHECK High	07H

Содержимое данных — 0103H, двоичный формат — 0000 0001 0000 0011, запишите соответствующие Y17~Y0, чтобы Y0, Y1, Y10 были включены.

Примечание: при записи содержимого данных младший байт располагается перед старшим байтом.

(6) Код функции 10H: Запись данных нескольких регистров

Например, запишите данные 3 регистров, начиная с адреса 0000H (D0).

Режим RTU:

Формат запроса		Формат ответа	
Адрес	01H	Адрес	01H
Код функции	10H	Код функции	10H
Адрес регистра	00H	Адрес регистра	00H
	00H		00H
Номер регистра	00H	Номер регистра	00H
	03H		03H
Номера байтов	06H	-	-
Содержимое данных	00H		
	01H		
	00H		
	02H		
	00H		
	03H		
CRC CHECK Low	3AH	CRC CHECK Low	3AH
CRC CHECK High	81H	CRC CHECK High	81H

После выполнения значениями в D0, D1, D2 будут 1, 2, 3.

Примечание: номер байта = номер регистра × 2.

Структура данных Modbus-ASCII

1. Режим ASCII

Для режима Modbus ASCII (американский стандартный код для обмена информацией) в последовательных каналах каждый 8-битный байт передается как два символа ASCII. Если каналы связи и устройства не подходят для временного монитора режима RTU, обычно используют режим ASCII.

Примечание: для одного байта требуется два символа, поэтому режим ASCII более эффективен, чем режим RTU.

Пример: Байт 0X5B будет закодирован как два символа: 0x35 и 0x42 (Код ASCII 0x35 = "5", 0x42 = "B").

Область проверки кадра в режиме ASCII: продольный контроль по избыточности (LRC).

Описание кадра режима ASCII:

Метка начала	Номер Modbus	Код функции	данные	LRC	Метка окончания	
1 символ	2 символа	2 символа	0~252*2 символа	2 символа	2 символа	
0x3A					0x0D	0x0A

Формат:

STX (3AH)	Метка начала = 3AH
Старший бит кода адреса	Положение связи (номер): Состоит из 2 кодов формата ASCII
Младший бит кода адреса	
Старший бит кода функции	Код функции (команда): Состоит из 2 кодов формата ASCII
Младший бит кода функции	
Идентификатор начала команды	Стартовый бит команды: Состоит из 4 кодов формата ASCII
Идентификатор начала команды	
Идентификатор начала команды	
Идентификатор начала команды	

Длина данных	Длина от начала до конца: Состоит из 4 кодов формата ASCII
Длина данных	
Длина данных	
Длина данных	
Старший бит контроля LRC	Код контроля LRC: Состоит из 2 кодов формата ASCII
Младший бит контроля LRC	
Старший бит END	Метка окончания: END Hi = CR(0DH), END Lo = CR(0AH)
Младший бит END	

2. Адрес связи

00H: Все ПЛК PROMPOWER обеспечивают широковещательную передачу — ведомые станции не отвечают.

01H: Связь с адресом 01H ПЛК.

0FH: Связь с адресом 15H ПЛК.

10H: Связь с адресом 16H ПЛК.

И так далее, до 254 (FEH).

3. Функции и данные

Код функции	Функция	Соответствующая команда Modbus
01H	Считывание данных с катушки	COLR
02H	Считывание данных с входной катушки	INRR
03H	Считывание данных из регистра	REGR
04H	Считывание данных из входного регистра	INRR
05H	Запись данных в отдельную катушку	COLW
06H	Запись данных в отдельный регистр	REGW

10H	Запись данных нескольких регистров	MRGW
0FH	Запись данных нескольких катушек	MCLW

Возьмем в качестве примера код функции 06H (запись данных в отдельный регистр) и представим формат данных (другие функции аналогичны этой):

Например: верхний компьютер записывает данные K5000 (H1388) в ПЛК H0002 (D2).

Режим ASCII:

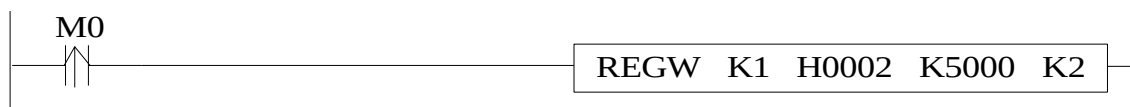
Метка начала	3AH
Идентификатор	30H
	31H
Код функции	30H
	36H
Старший байт идентификатора регистра	30H
	30H
Младший байт идентификатора регистра	30H
	32H
Старший байт содержимого данных	31H
	33H
Младший байт содержимого данных	38H
	38H
LRC	35H
	43H
Метка окончания	0DH
	0AH

Описание:

1. Address (Адрес) — это номер станции ПЛК.
2. Function code (Код функции) — это код чтения/записи протокола Modbus-ASCII.

3. Register ID (Идентификатор регистра) — это идентификатор связи по Modbus ПЛК, см. главу 7-2-2.
4. Data content (Содержимое данных) — это значение в D2.
5. LRC CHECK Low / CRC CHECK High — это младший и старший бит значения контроля CRC.

Если два ПЛК серии PMP20 взаимодействуют друг с другом, запишите K5000 в D2.



M0 — условие запуска (нарастающий фронт). Если при обмене данными между ПЛК PROMPOWER и Modbus связь не удалось установить, команда повторит попытку дважды. Если и на третий раз связь не установлена, связь завершается.

REGW	Код функции 06H
K1	Номер станции
H0002	Идентификатор Modbus
K5000	Содержимое данных — 1388H
K2	Последовательный порт связи ПЛК

Связь между REGW и протоколом ASCII (остальные команды те же):

Полная строка данных:

3АН 30Н 31Н 30Н 36Н 30Н 30Н 30Н 32Н 31Н 33Н 38Н 38Н 35Н 43Н (система автоматически выполняет контроль CRC)

Если контролировать данные последовательного порта 2 с помощью инструмента отладки последовательного порта, то данные будут следующими:

3АН 30Н 31Н 30Н 36Н 30Н 30Н 30Н 32Н 31Н 33Н 38Н 38Н 35Н 43Н 0DH
0АН

Примечание: Команда не различает десятичные, шестнадцатеричные, двоичные значения и т. д. Например, B10000, K16 и H10 — это одно и то же значение, поэтому представленные ниже команды одинаковы.

REGW K1 B111110100 D1 K2

REGW K1 K500 D1 K2

REGW K1 H1F4 D1 K2

6.3.5 Команды связи

К командам Modbus относятся: считывание/запись данных в катушку, считывание/запись данных в регистр. Подробная информация представлена ниже.

Подробное описание команд. Определение операнда в командах:

1. Удаленная станция связи и номер последовательного порта.

Например, один ПЛК позволяет подключить 3 инвертора. ПЛК должен записывать и считывать параметры инвертора. Номера станций инверторов — 1, 2 и 3. Таким образом, номера устройств удаленной связи — 1, 2 и 3.

2. Начальный идентификатор удаленного регистра/катушки:

Назначение номера удаленной катушки/регистра: начальный идентификатор катушки/регистра для считывания и записи данных в ПЛК, обычно используется с «назначенным номером катушки/регистра».

Пример: ПЛК считывает выходную частоту инвертора (H2103), выходной ток (H2104), напряжение шины (H2105), затем начальный идентификатор удаленного регистра/катушки — H2103, назначенный номер катушки — K3.

3. Локальный адрес катушки/регистра приема/передачи: катушка/регистр в ПЛК, используемая для обмена данными с нижним компьютером.

Пример: Запись в катушку M0: запись статуса M0 по назначенному адресу в нижнем компьютере

Запись регистра D0: запись значения D0 по назначенному адресу

Считывание данных катушки M1: считывание содержимого в нижнем компьютере по назначенному адресу M1

Считывание данных регистра D1: считывание содержимого в нижнем компьютере по назначенному адресу D1

4. Условия для связи

Условия для связи по протоколу Modbus: нормально-разомкнутая/нормально-замкнутая катушка и нарастающий/спадающий фронт. При срабатывании катушки в нормально-разомкнутом/нормально-замкнутом состоянии команды Modbus всегда выполняются. При обмене данными между несколькими ведомыми станциями или при большом объеме трафика может возникнуть задержка связи. Вибрирующая катушка может быть условием запуска. При запуске по нарастающему/спадающему фронту команды Modbus будут выполнены только один раз, и только при появлении следующего нарастающего/спадающего фронта команды Modbus будут выполнены снова.

Считывание данных с катушки [COLR]

1) Обзор

Считывание состояния катушки указанной станции в локальное устройство.

Считывание данных с катушки [COLR]			
16-битная команда	COLR	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Задание номера удаленной станции связи	16 бит, BIN
S2	Задание начального адреса удаленной катушки	16 бит, BIN
S3	Задание количества катушек	16 бит, BIN

D1	Задание начального адреса локальной катушки	биты
D2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

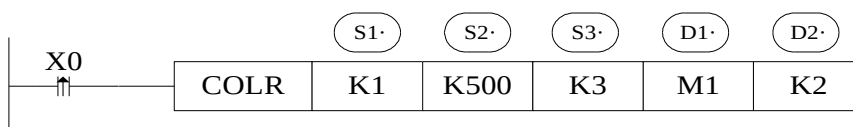
Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Идентификатор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•		•	•				•									
S2	•	•		•	•				•									
S3	•	•		•	•				•									
D1												•	•	•	•	•	•	
D2									K									

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; M включает в себя M, HM, SM; S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Считывание данных с катушки, код функции Modbus 01H

- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнды S3: K1~K2000, макс. количество катушек — 2000.
- Если X0 включен, команда COLR будет выполняться. После начала выполнения команды признак считывания и записи Modbus SM160 (последовательный порт 2) включается; после завершения выполнения SM160 (последовательный порт 2) выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

Считывание данных с входной катушки [INPR]

1) Обзор

Считывание состояния входной катушки указанной станции в локальное устройство.

Считывание данных с входной катушки [INPR]			
16-битная команда	INPR	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, нарастающий фронт	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Задание номера удаленной станции связи	16 бит, BIN
S2	Задание номера начального адреса удаленной катушки	16 бит, BIN
S3	Задание номера катушки	16 бит, BIN
D1	Задание номера начального адреса локальных катушек приема	бит
D2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Опе- ранды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден- тифи- катор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•		•	•				•									
S2	•	•		•	•				•									
S3	•	•		•	•				•									
D1												•	•	•	•	•	•	
D2									К									

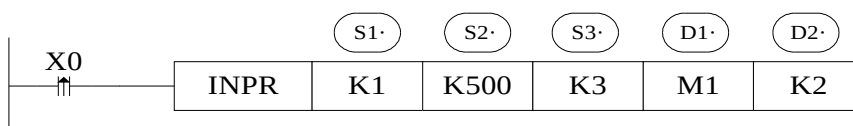
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Считывание данных с входной катушки, код функции Modbus 02H.
- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнд S3: K1~K2000, макс. количество входных катушек — 2008.

- Если X0 включен, выполняется команда INPR, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.
- Эта команда не может считывать данные с входной катушки ПЛК PROMPOWER.

Запись данных в отдельную катушку [COLW]

1) Обзор

Запись данных с катушки локального устройства в катушку с номером удаленной станции.

Запись данных в отдельную катушку [COLW]			
16-битная команда	COLW	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по фронту	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
D1	Задание номера удаленной станции связи	16 бит, BIN
D2	Задание начального адреса удаленной катушки	16 бит, BIN
S1	Задание начального адреса локальной катушки	бит
S2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m
D1	•	•		•	•				•									
D2	•	•		•	•				•									
S1												•	•	•	•	•	•	
S2									К									

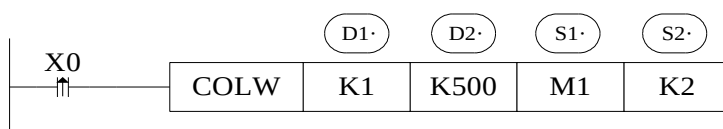
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Запись данных в отдельную катушку, код функции Modbus 05H.
- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Если X0 включен, выполняется команда COLW, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

Запись данных нескольких катушек [MCLW]

1) Обзор

Запись данных нескольких катушек локального устройства в катушку с номером удаленной станции.

Запись данных нескольких катушек [MCLW]			
16-битная команда	MCLW	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по фронту	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
D1	Задание номера удаленной станции связи	16 бит, BIN
D2	Задание начального адреса удаленной катушки	16 бит, BIN
D3	Задание номера катушки	16 бит, BIN
S1	Задание начального адреса локальных катушек	бит
S2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Идентификатор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•					•									
S2	•	•	•	•					•									
S3	•	•	•	•					•									
D1												•	•	•	•	•	•	
D2									К									

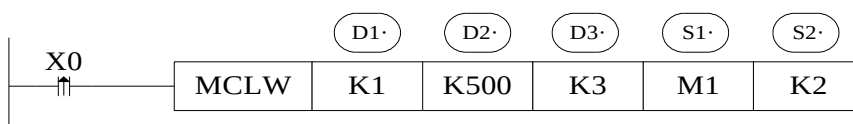
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Запись данных нескольких катушек, код функции Modbus — 0FH.
- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнд D3: макс. количество катушек — 1976.
- Если X0 включен, выполняется команда MCLW, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

Считывание данных из регистра [REGR]

1) Обзор

Считывание данных из регистра номера удаленной станции в локальное устройство.

Считывание данных из регистра [REGR]			
16-битная команда	REGR	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по фронту	Совместимые модели	PMP20

Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-
--------------------------------------	---	---------------------------------------	---

2) Операнды

Операнды	Функция	Тип
S1	Задание номера удаленной станции связи	16 бит, BIN
S2	Задание начального адреса удаленного регистра	16 бит, BIN
S3	Задание номера регистра	16 бит, BIN
D1	Задание начального адреса локального регистра	16 бит, BIN
D2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m	
S1	•	•	•	•					•										
S2	•		•	•					•										
S3	•	•	•	•					•										
D1	•																		
D2									К										

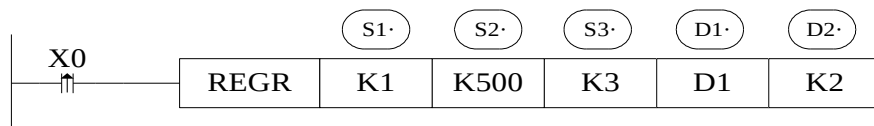
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Считывание данных из регистра, код функции Modbus 03H.
- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнд S3: макс. количество регистров — 125.
- Если X0 включен, выполняется команда REGR, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

Считывание данных из входного регистра [INRR]

1) Обзор

Считывание данных из входного регистра номера удаленной станции в локальное устройство.

Считывание данных из входного регистра [INRR]			
16-битная команда	INRR	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по фронту	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Задание номера удаленной станции связи	16 бит, BIN
S2	Задание начального адреса удаленного регистра	16 бит, BIN
S3	Задание номера регистра	16 бит, BIN

D1	Задание начального адреса локального регистра	16 бит, BIN
D2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден- тифи- катор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•					•									
S2	•		•	•					•									
S3	•	•	•	•					•									
D1	•																	
D2									К									

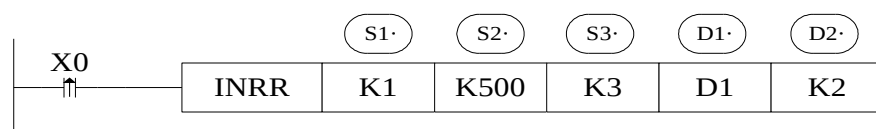
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Считывание данных из входного регистра, код функции Modbus 04H.

- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнд S3: макс. количество регистров — 125.
- Если X0 включен, выполняется команда INRR, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

Запись данных в отдельный регистр [REGW]

1) Обзор

Запись данных из регистра локального устройства в указанный регистр номера удаленной станции.

Запись данных в отдельный регистр [REGW]			
16-битная команда	REGW	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по фронту	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
D1	Задание номера удаленной станции связи	16 бит, BIN
D2	Задание начального адреса удаленного регистра	16 бит, BIN
S1	Задание начального адреса локального регистра	16 бит, BIN
S2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден- тифи- катор	QD	X	Y	M	S	T	C	Dn.m
D1	•	•	•	•					•									
D2	•	•	•	•					•									
S1	•																	
S2									K									

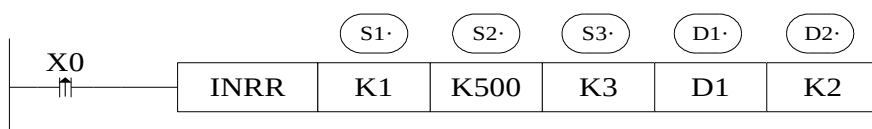
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Считывание данных из входного регистра, код функции Modbus 04H.
- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнд S3: макс. количество регистров — 125.
- Если X0 включен, выполняется команда INRR, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически от-

правлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

Запись данных нескольких регистров [MRGW]

1) Обзор

Запись данных нескольких регистров локального устройства в регистры номеров удаленных станций.

Запись данных нескольких регистров [MRGW]			
16-битная команда	MRGW	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по фронту	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
D1	Задание номера удаленной станции связи	16 бит, BIN
D2	Задание начального адреса удаленного регистра	16 бит, BIN
D3	Задание номера регистра	16 бит, BIN
S1	Задание начального адреса локальных регистров	16 бит, BIN
S2	Задание номера последовательного порта	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Идентификатор	QD	X	Y	M	S	T	C	Dn.m
D1	•	•	•	•					•									
D2	•	•	•	•					•									
D3	•	•	•	•					•									

S1	•																	
S2									K									

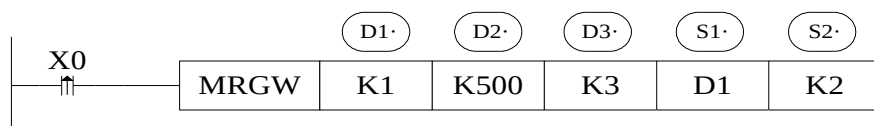
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



- Запись данных нескольких регистров, код функции Modbus — 10H.
- Последовательный порт: K0~K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- Операнд D3: макс. количество регистров — 123.
- Если X0 включен, выполняется команда MRGW, признак считывания и записи Modbus SM160 (последовательный порт 2) включается, и после завершения выполнения SM160 выключается. Если произошла ошибка связи, и установлено количество повторных отправок, сообщение будет автоматически отправлено повторно. Пользователи могут проверить соответствующие регистры, чтобы определить причину ошибки. Результат выполнения команд считывания и записи Modbus последовательного порта 2 хранится в SD160.

6.3.6 Настройка последовательного порта Modbus

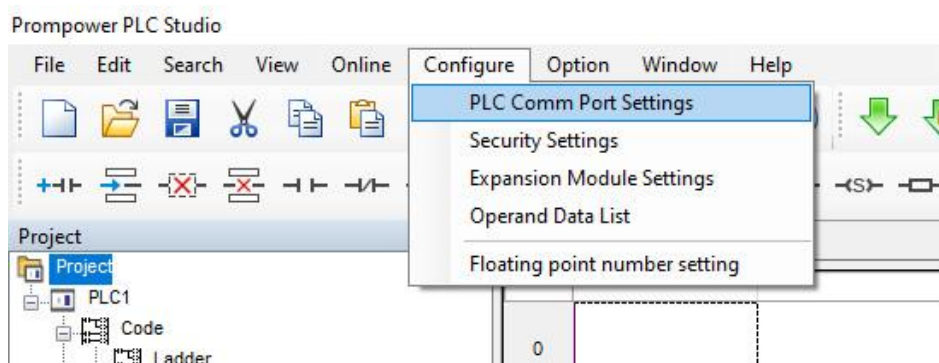
Параметры связи через Modbus можно настроить двумя способами:

- 1) Настройка параметров с помощью программного обеспечения для программирования;
- 2) Настройка параметров с помощью XNetConfigTool

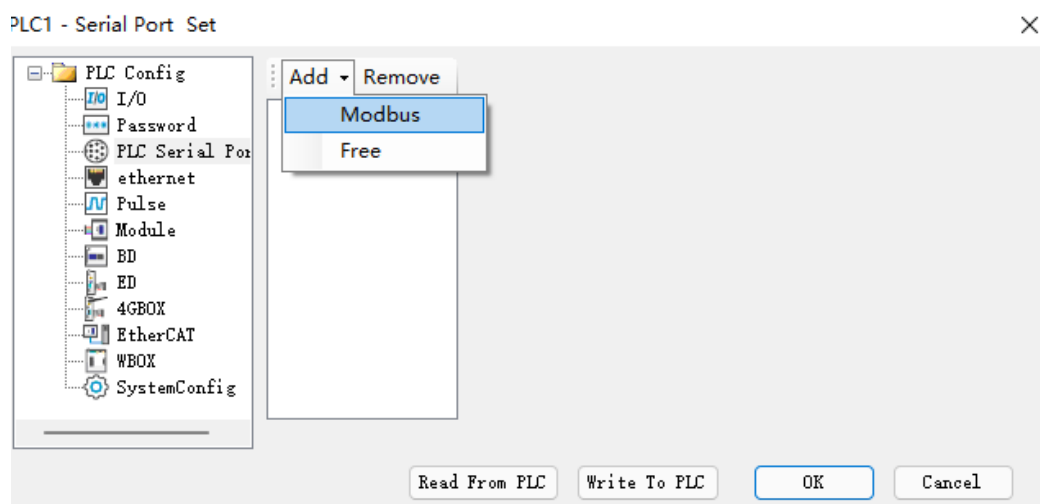
1. Настройка параметров с помощью программного обеспечения для программирования

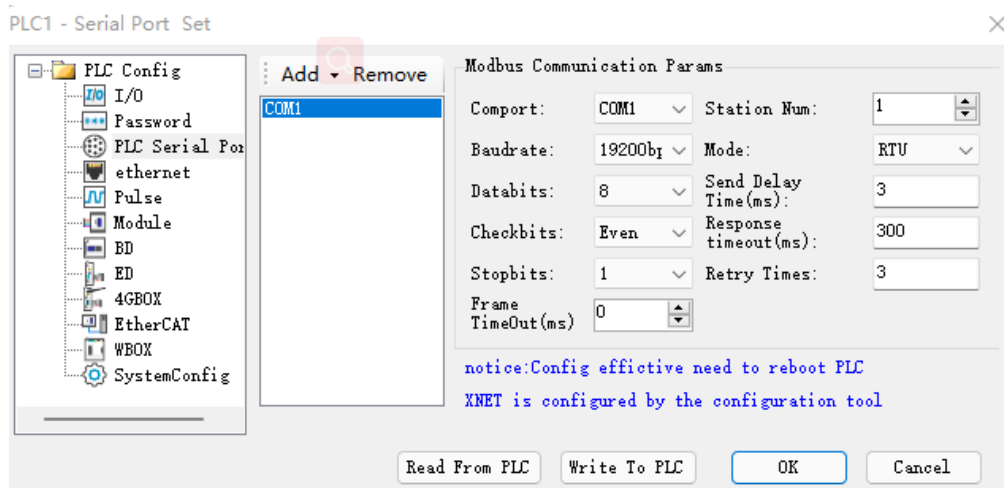
Если для настройки параметров последовательного порта ПЛК используется программное обеспечение для программирования, то версия ниже 3.4 должна использовать режим связи через XNET, а версия выше 3.4 дополнительно может использовать режим связи через Modbus (порт RS232).

- (1) Откройте программное обеспечение для программирования, выберите configure/PLC com port settings (Настроить/Настройки COM-порта ПЛК). См. рисунок ниже:



- (2) Нажмите Add (Добавить), отобразятся два режима: режим Modbus и режим связи в свободном формате (Free). Выберите режим Modbus, как показано на рисунке ниже.





Port No. (Номер порта): это порты ПЛК. COM1 — порт 1 (RS232), COM2 — порт 2 (RS485), COM3 — порт 3 (левый порт расширения ED), COM4 — порт 4 (верхний порт 1 расширения BD), COM5 — порт 5 (верхний порт 2 расширения BD).

The baud rate, data bit, parity bit, stop bit (Скорость передачи данных, бит данных, бит четности, стоповый бит) должны быть такими же, как у устройства СВЯЗИ.

Station number (Номер станции): если ПЛК является главным устройством, то номером станции по умолчанию будет 1, если ПЛК является ведомым устройством, необходимо задать другой номер станции.

Two communication modes (Два режима связи): RTU, ASCII.

Delay before sending (Задержка перед отправкой): время ожидания перед отправкой данных из ПЛК; ПЛК серии RMP20 после получения данных от ведомой станции должен выждать определенное время перед получением следующих данных связи, в противном случае могут возникнуть описанные выше проблемы.

Reply overtime (ms) (Превышено время ответа (мс)): это время, в течение которого ПЛК не может получить ответ после отправки запроса и ждет повторной отправки.

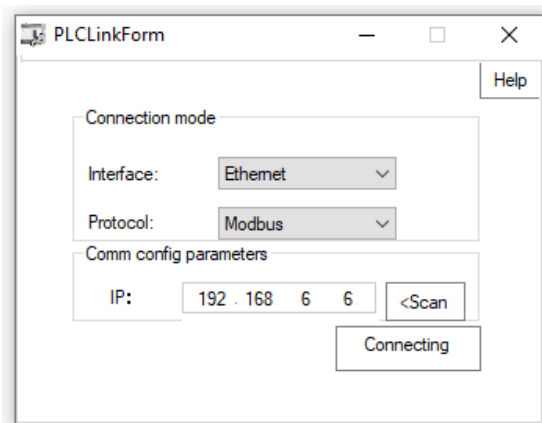
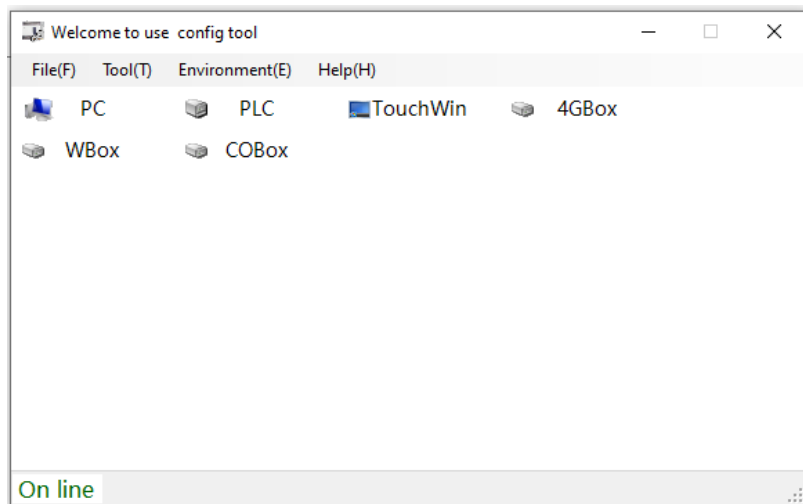
Retry times (Количество повторных попыток): Это количество раз, когда ПЛК не может получить ответ, и для каждого ответа требуется время таймаута ответа.

(3) После ввода настроек нажмите Write to PLC (Записать в ПЛК), затем отключите питание ПЛК и включите его снова, чтобы настройки вступили в силу.

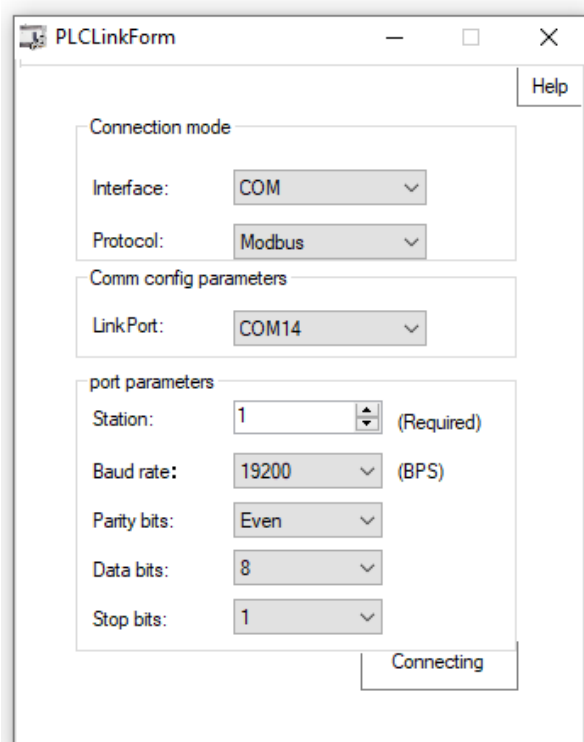
2. Настройка параметров с помощью XNet Config Tool

Инструмент XNet Config Tool для версии 1.6.309 и более поздних версий также можно настроить через порт RS232.

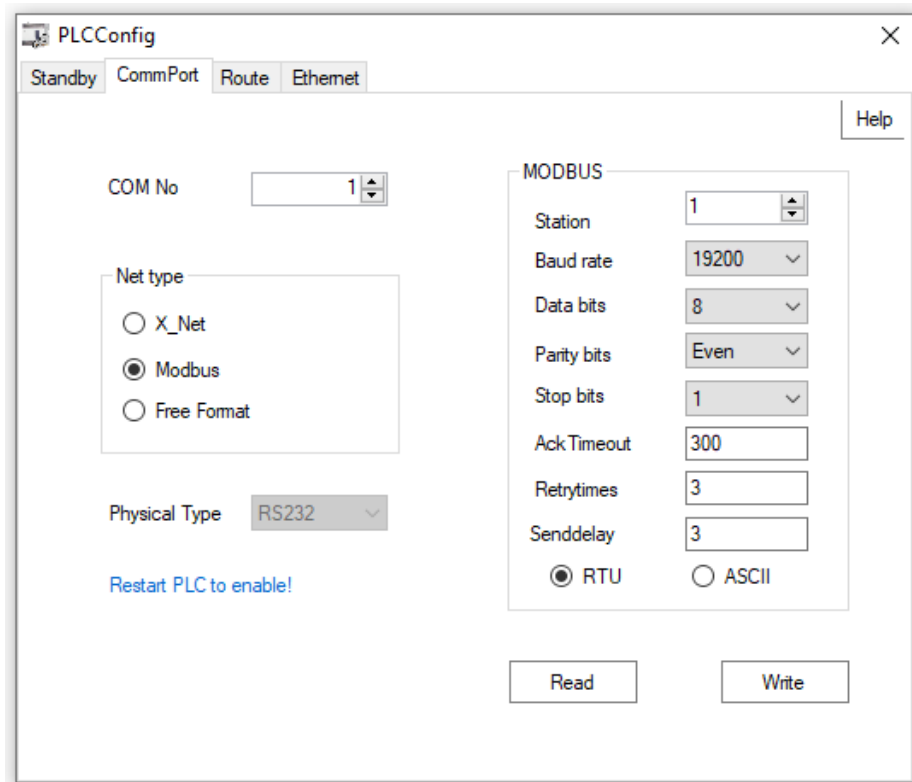
(1) Откройте XNet Config Tool, нажмите PLC (ПЛК).



(2) Выберите COM-интерфейс и COM-порт, связывающие ПК и ПЛК, нажмите “Connecting” (Подключение), а затем выберите вкладку “ComPort”:



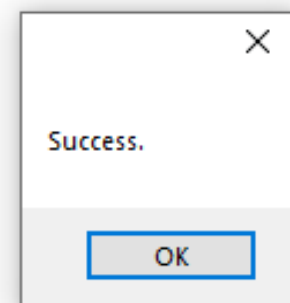
(3) См. окно ниже.



Последовательный порт: К1 ~ К5. Порт1 (RS232), Порт2 (RS485) или Порт2-RS232 (RS232) или Порт2-RS485 (RS485), Порт3 (левый порт расширения), Порт4 (верхний порт расширения 1), Порт5 (верхний порт расширения 2).

Здесь мы можем настроить режим связи и параметры каждого порта связи.

(4) После завершения настройки параметров COM-порта нажмите writeconfig. Появится сообщение “write configuration success” (Запись конфигурации успешно выполнена).



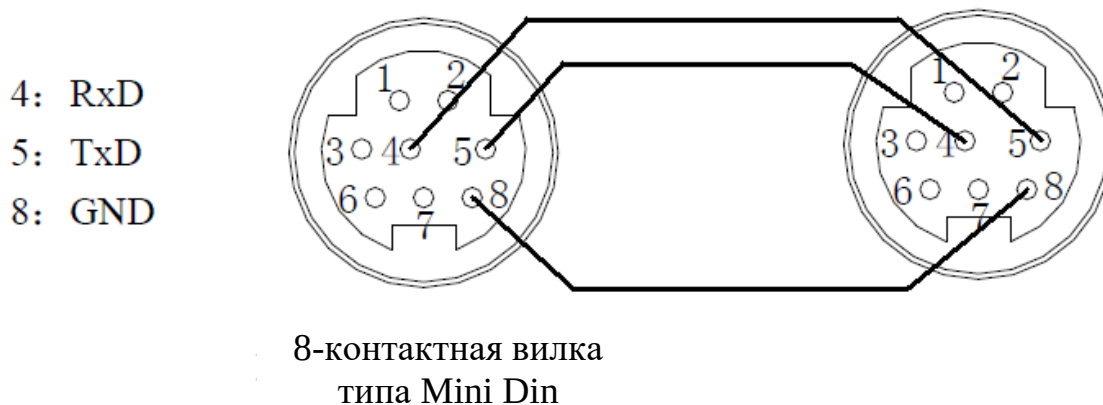
(5) Закройте XNet Config Tool, отключите питание ПЛК и включите его снова, чтобы настройки вступили в силу.

6.3.7 Применение связи через Modbus

Существуют два метода подключения:

Подключение через порт 232

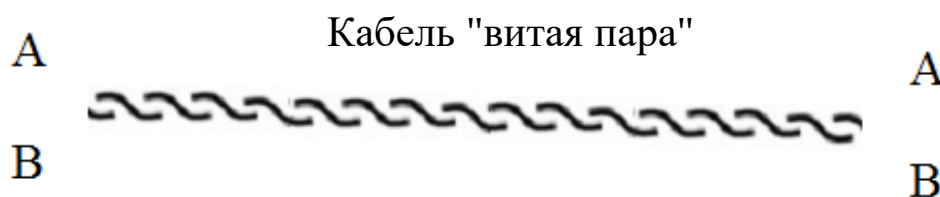
Схема порта COM2*1



Примечание:

1. Порт COM2 только с *1 имеет контакты RS232.
2. У ПЛК серии RMP20 порт RS232 не поддерживает дуплексный режим работы, поэтому передача данных может происходить только в одном направлении.
3. Расстояние связи через RS232 небольшое (около 13 м); RS485 подходит для связи на большом расстоянии.

Подключение через порт 485



Подключение всех клемм А, подключение всех клемм В. А — RS485+, В — RS485-.

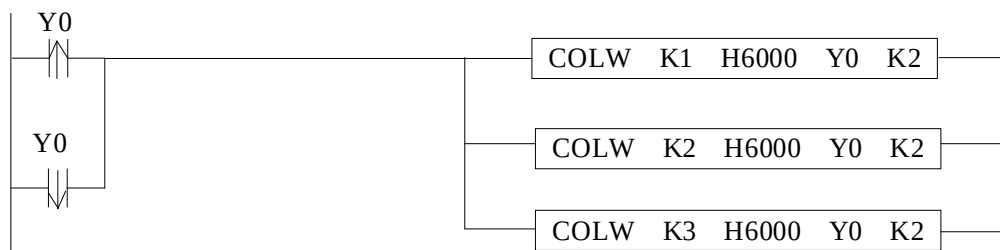
Область применения:

Один ПЛК серии RMP20 управляет тремя аналогичными ПЛК, ведомые ПЛК выполняют команды главного ПЛК (включается Y0 главного ПЛК, затем включается Y0 ведомого ПЛК; выключается Y0 главного ПЛК, затем выключается Y0 ведомого ПЛК).

Обязательные условия:

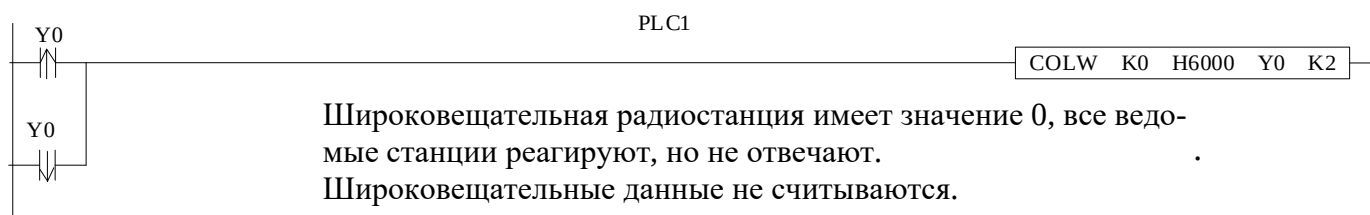
Включение-выключение Y0 дает устройствам связи достаточно времени для реагирования. Кроме того, три ведомых ПЛК могут быть не настолько синхронными (не полностью синхронными).

Метод 1 Стандартная программа



В качестве примера рассмотрим последовательный порт 2, поэтому соответствующий признак связи относится к последовательному порту 2. Другие последовательные порты описаны в Приложении 1. Последовательный порт, см. приложение 1.

Метод 2 Использование функции широковещательной передачи



При изменении состояния Y0 главного устройства оно будет передавать это состояние всем ведомым устройствам. Синхронизация трех ПЛК лучше, чем метод 1.

6.3.8 Применение

Пример 1:

Ниже приведены программы для считывания и записи данных связи по протоколу Modbus между 1 главной станцией и 3 ведомыми станциями.

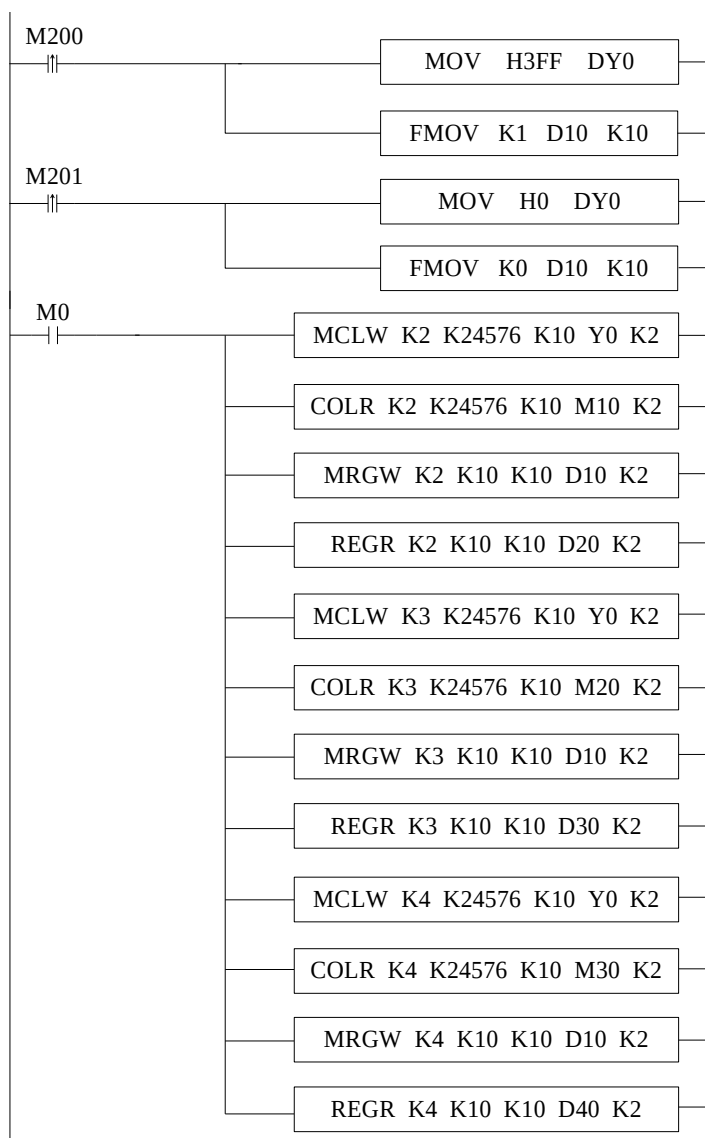
Работа с программой:

- (1) Запишите состояние Y0~Y11 главного ПЛК в Y0~Y11 ведомого ПЛК 2.
- (2) Считайте состояние Y0~Y11 ведомого ПЛК 2 в M10~M19 главного ПЛК.

- (3) Запишите состояние D10~D19 главного ПЛК в D10~D19 ведомого ПЛК 2.
- (4) Считайте состояние D10~D19 ведомого ПЛК 2 в D20~D29 главного ПЛК.
- (5) Такая же процедура для ПЛК 3 и 4.

Пользователи могут записывать команды Modbus-RTU прямо в пользовательские программы. Стек протоколов будет ставить в очередь запросы связи по протоколу Modbus-RTU. Связь является другой задачей. В главной программе пользователи могут записывать несколько команд связи Modbus-RTU вместе и запускать их одновременно с помощью одного и того же условия запуска. ПЛК запускает эти команды связи. Команды выстраиваются в очередь в соответствии с протокольной станцией Modbus-RTU, благодаря чему не будет ошибок связи при одновременном выполнении нескольких команд связи.

Программа:



// По нарастающему фронту M200 включите Y0~Y11 главного ПЛК, D10~D19 принимают значение 1, по нарастающему фронту M201 отключите Y0~Y11 главного ПЛК, сбросьте D10~D19.

// Запишите Y0~Y11 главного ПЛК в Y0~Y11 ведомого ПЛК 2, считайте Y0~Y11 ведомого ПЛК 2 в M10~M19 главного ПЛК. Запишите D10~D19 главного ПЛК в D10~D19 ведомого ПЛК 2. Считайте D20~D29 ведомого ПЛК 2 в D20~D29 главного ПЛК.

6.4 Связь в свободном формате

6.4.1 Режим связи в свободном формате

Связь в свободном формате — это передача данных в виде блоков данных, ограниченных размером кэша ПЛК. Максимальный объем отправляемых за один раз данных составляет 256 байт.

При так называемой связи в свободном формате, т.е. связи по специальному протоколу, в настоящее время многие интеллектуальные устройства на рынке поддерживают связь через RS232 или RS485, но протоколы, используемые различными продуктами, отличаются, например: ПЛК PROMPOWER использует стандартный протокол Modbus-RTU, некоторые производители регуляторов температуры используют специальные протоколы. Если применяется ПЛК PROMPOWER, то для связи с регулятором температуры необходимо использовать связь в свободном формате для передачи данных в полном соответствии с протоколом производителя прибора.

Условия для связи в свободном формате:

1. Порт0 (RS232), Порт1 (RS232), Порт2 (RS485) или Порт2-RS232 (RS232) или Порт2-RS485 (RS485), Порт3 (левый порт расширения), Порт4 (верхний порт расширения 1), Порт5 (верхний порт расширения 2). Все они поддерживают связь в свободном формате. Поскольку для связи в свободном формате требуются изменения параметров связи, порт 1 использовать не рекомендуется.
2. Скорость передачи данных: 300 бит/с - 3 Мбит/с, 4,5 Мбит/с - 9 Мбит/с (поддержка специальных моделей).
3. Формат данных должен совпадать с настройками нижнего устройства. Возможно несколько вариантов:

Бит данных: 5 битов (поддержка специальных моделей), 6 битов (поддержка специальных моделей), 7 битов, 8 битов, 9 битов.

Бит четности: отсутствует, отрицательная четность, положительная четность, не задействована, маска.

Стоповый бит: 1 бит, 1,5 бита, 2 бита.

4. Стартовый символ: 1 байт, символ завершения: 1 байт.



Пользователи могут настраивать стартовый символ/символ завершения. После настройки стартового символа/символа завершения ПЛК автоматически добавляет стартовый символ/символ завершения при отправке данных и автоматически удаляет этот символ при получении данных.

Фактически, инициатор и терминатор можно рассматривать как начало и конец кадра данных в протоколе. Таким образом, если связь нижнего устройства имеет стартовый символ и символ завершения, их можно задать в программном обеспечении или записать в протоколе.

5. Режим связи: 8 бит, 16 бит.

Если для связи выбран 8-битный буфер, старшие байты регистров недействительны. Для отправки и получения данных ПЛК использует только младшие байты регистров.

Если для связи выбран 16-битный буфер, ПЛК отправит все данные регистра, сначала отправив данные младшего байта, затем данные старшего байта.

При передаче младших и старших байтов одного 16-битного регистра в другой 16-битный регистр для связи следует выбрать 16-битные буферы, а количество байтов связи должно быть равно 2. Если значение, хранящееся в 16-битном регистре, занимает только младшие байты, для связи можно выбрать 8-битный буфер. Количество байтов связи — 1. Обычно при обмене данными объем данных не превышает младшего байта регистра (HFF), поэтому для обмена данными нам потребуется только 8-битный буфер по умолчанию в программном обеспечении.

6. Таймаут: тайм-аут кадра (мс), тайм-аут ответа (мс).

Кадр: Строка данных:

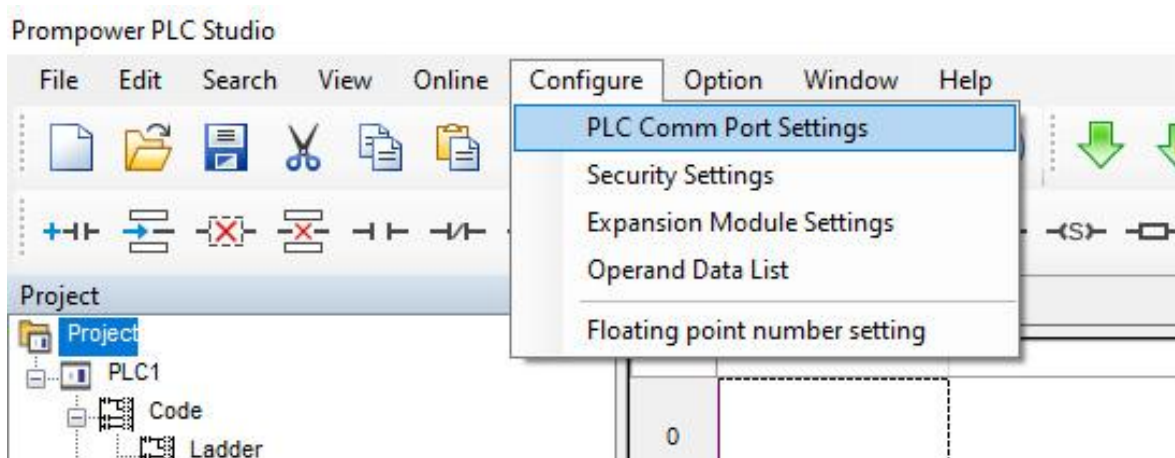
Тайм-аут кадра: это временной интервал между двумя кадрами данных, принятых ПЛК, который гарантирует, что ПЛК может распознать время окончания приема кадра. Обычно тайм-аут используется для оценки того, был ли получен кадр данных в ПЛК или нет. Если интервал между двумя кадрами данных превышает тайм-аут кадра, это означает конец одного кадра данных связи.

Тайм-аут ответа: это время, в течение которого ПЛК не может получить ответ после отправки запроса и ждет повторной отправки. Если время ответа превышает 300 мс, то при обмене данными по умолчанию ПЛК ждет ответа другой стороны в течение 300 мс. Если время ответа истекло, запрос будет повторно отправлен.

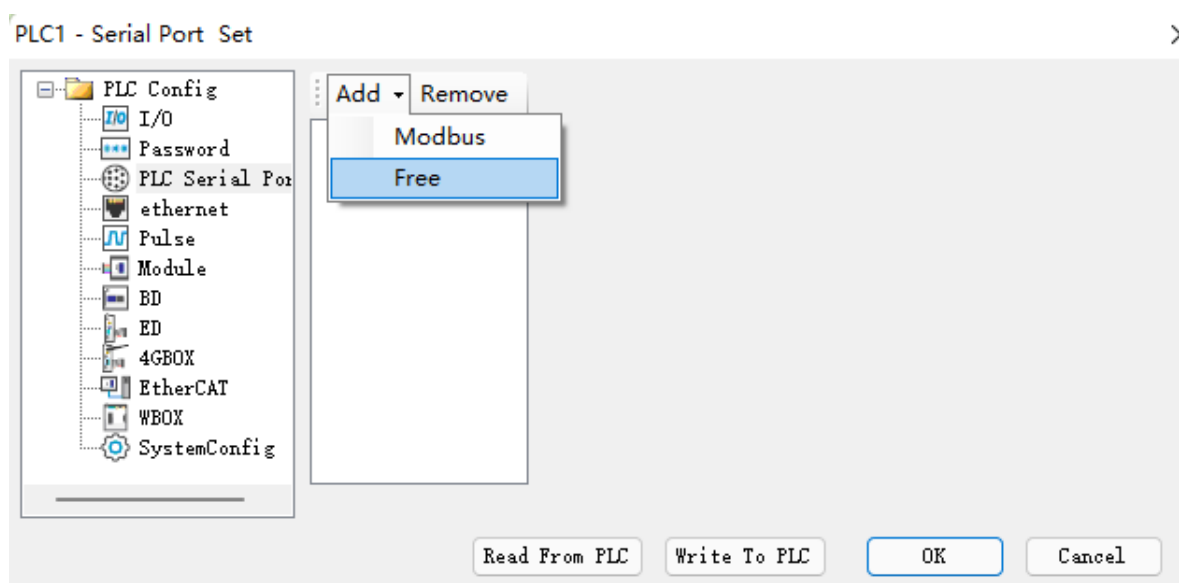
Чтобы сократить время связи, можно настроить два вышеуказанных параметра в соответствии со скоростью передачи данных.

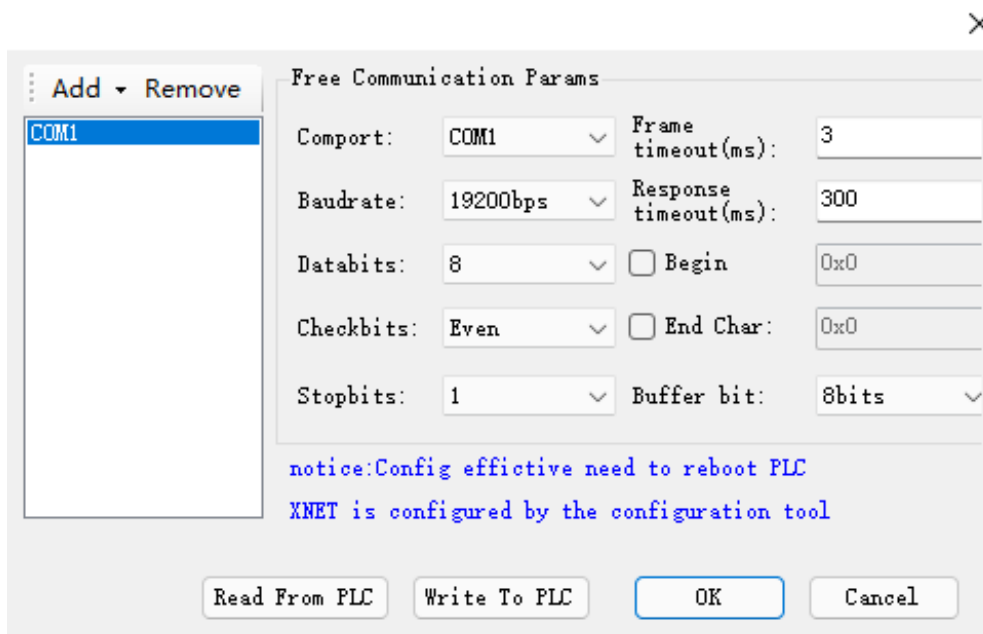
6.4.2 Настройка последовательного порта

- (1) Используйте кабель RS232 для подключения ПЛК к компьютеру. Программное обеспечение должно переключиться в режим связи через XNet.
- (2) Откройте программное обеспечение для программирования, выберите configure/PLC comm port settings (Настроить/Настройки COM-порта ПЛК). См. рисунок ниже:



- (3) Нажмите Add (Добавить), отобразятся два режима: режим Modbus и режим связи в свободном формате (Free). Выберите режим Free, как показано на рисунке ниже.





Port No. (Номер порта): Это порты ПЛК. COM1 — порт 1 (RS232), COM2 — порт 2 (RS485), COM3 — порт 3 (левый порт расширения ED), COM4 — порт 4 (верхний порт 1 расширения BD), COM5 — порт 5 (верхний порт 2 расширения BD).

Frame timeout (ms) (Тайм-аут кадра (мс)): это временной интервал между двумя кадрами данных, отправленными ПЛК, который гарантирует, что получатель распознает время окончания приема кадра.

Response timeout (ms) (Тайм-аут ответа (мс)): это время, в течение которого ПЛК не может получить ответ после отправки запроса и ждет повторной отправки.

Другие параметры последовательного порта можно настроить в соответствии с параметрами нижнего устройства.

- (4) После ввода настроек нажмите Write to PLC (Записать в ПЛК), затем отключите питание ПЛК и включите его снова, чтобы настройки вступили в силу.

6.4.3 Применение

Когда нужно использовать связь в свободном формате?

В качестве примера рассмотрим ситуацию, описанную в предыдущем разделе: ПЛК PROMPOWER обменивается данными с терморегулятором, и у терморегулятора есть собственный протокол связи, который требует, чтобы считываемая температура отправлялась в виде четырех символов: "R", "T", "CR". Значение этих символов:

Символ	Значение
:	Начало данных
R	Считывание
T	Температура
CR	Ввод, конец данных

ПЛК должен отправить ASCII-код, состоящий из указанных выше символов на прибор, чтобы считать текущее значение температуры, измеренное прибором. Значения кода ASCII (шестнадцатеричные) для каждого символа можно получить, запросив таблицу кодов ASCII.

Символ	Значение кода ASCII
:	3A
R	52
T	54
CR	0D

Очевидно, что в описанной выше ситуации использование команд MODBUS не позволяет осуществлять связь, и в таком случае нужно использовать связь в свободном формате. Подробная процедура будет использована в качестве примера для программирования образца программы в последующих главах.

6.4.4 Команды связи в свободном формате

Отправка данных [SEND]

1) Обзор

Запись локальных данных на удаленную станцию по указанному адресу.

Отправка данных [SEND]			
16-битная команда	SEND	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, Запуск по нарастающему фронту	Совместимые модели	PMP20
Аппаратное обеспечение	Версия 3.2.3 и более поздние версии	Программное обеспечение	Версия 3.2.2 и более поздние версии

2) Операнд

Операнды	Функция	Тип
S1	Начальный адрес локальных данных	16 бит, BIN
S2	Отправляемое количество байтов	16 бит, BIN
n	Номер порта связи	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Идентификатор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•														
S2	•	•	•	•					•									
n	•								K									

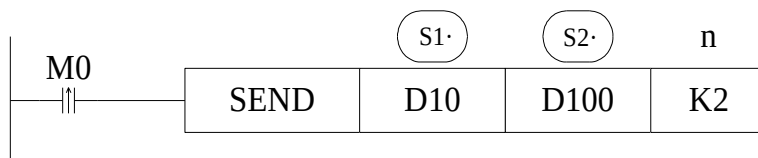
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

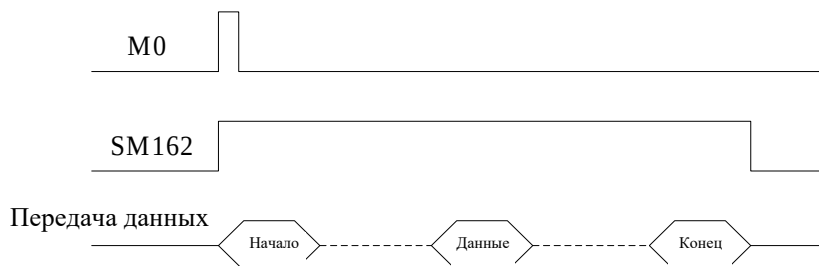
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Команды отправки данных: по нарастающему фронту M0 данные отправляются один раз.
- Порт связи. Диапазон: K0 ~ K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- В процессе передачи данных включается признак отправки SM162 (порт связи 2).



- Если номер буфера имеет размер 8 бит, отправляются только данные младшего байта, поэтому D100 = количеству отправленных регистров; например, для отправки данных младшего байта в D10-D17, D100 следует установить на 8.
- Если номер буфера имеет размер 16 бит, будут отправлены данные старшего и младшего байта, поэтому D100 = количеству отправленных регистров × 2. Например, при отправке данных старшего и младшего байтов в D10-D17, D100 следует установить на 16, и при отправке младший байт будет стоять перед старшим байтом.

Прием данных [RCV]

1) Обзор

Запись данных с номером указанной удаленной станции в локальное устройство.

Прием данных [RCV]			
16-битная команда	RCV	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, Запуск по нарастающему фронту	Совместимые модели	PMP20
Аппаратное обеспечение	Версия 3.2.3 и более поздние версии	Программное обеспечение	Версия 3.2.2 и более поздние версии

2) Операнды

Операнды	Функция	Тип
S1	Начальный адрес локальных данных	16 бит, BIN
S2	Количество принятых байтов или адрес программного компонента	16 бит, BIN
n	Номер порта связи	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Идентификатор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•														
S2	•	•	•	•					•									
n	•								K									

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

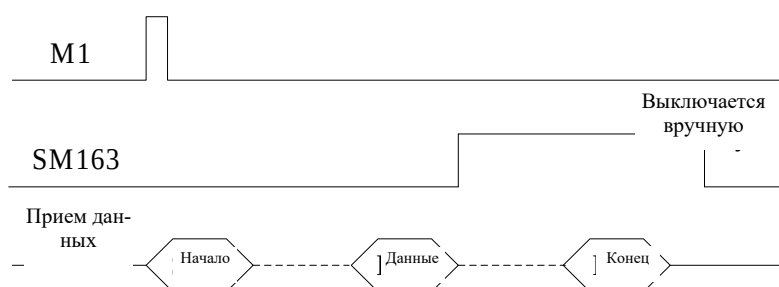
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Команды для приема данных: по нарастающему фронту M1 данные принимаются один раз.
- Порт связи. Диапазон: K0 ~ K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- После приема данных включается признак «принято» SM163 (порт связи 2).



- Если номер буфера имеет размер 8 бит, то принятые данные сохраняются только в младших байтах, поэтому D200 = количество принимаемых байтов × 2. Например, чтобы принять 8 байтов данных, сохраненных в младших байтах восьми регистров D20-D27 по очереди, D200 следует установить на 16.
- Если номер буфера имеет размер 16 бит, то принятые данные сохраняются в полном регистре, поэтому D200 = количество принимаемых байтов. Например, чтобы принять 8 байтов данных, сохраненных в четырех регистрах D20-D23 по очереди, D200 следует установить на 8. При приеме данных младшие байты будут стоять перед старшими байтами.

Освобождение последовательного порта [RCVST]

1) Обзор

Освобождение указанного последовательного порта.

Освобождение последовательного порта [RCVST]			
16-битная команда	RCVST	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, Запуск по нарастающему фронту	Совместимые модели	PMP20

Аппаратное обеспечение	Версия 3.2.3 и более поздние версии	Программное обеспечение	Версия 3.2.2 и более поздние версии
------------------------	-------------------------------------	-------------------------	-------------------------------------

2) Операнды

Операнд	Функция	Тип
n	Номер порта связи	16 бит, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m
n	•								К									

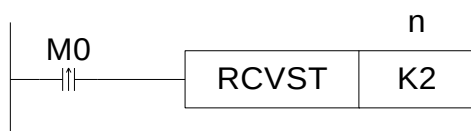
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

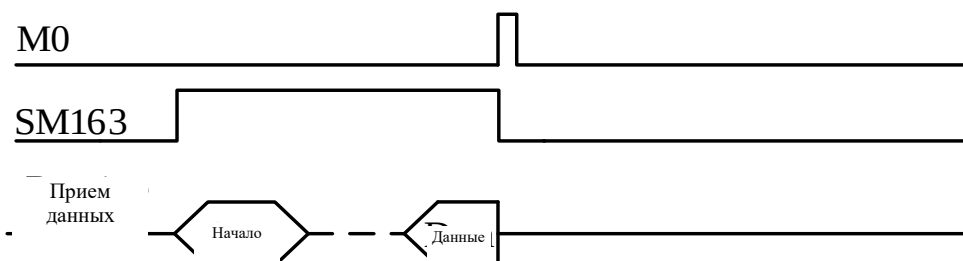
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Команды освобождения последовательного порта, по нарастающему фронту M0 выполняется один раз.
- Порт связи. Диапазон: K0 ~ K5. K0: Порт0 (RS232), K1: Порт1 (RS232), K2: Порт2 (RS485), K3: Порт3 (левый порт расширения), K4: Порт4 (верхний порт расширения 1), K5: Порт5 (верхний порт расширения 2).
- После освобождения последовательного порта выключается признак «принято» SM163 (порт связи 2).

- В режиме связи в свободном формате, если тайм-аут отсутствует, или установлен слишком большой тайм-аут, занятые ресурсы последовательного порта можно немедленно освободить для других операций связи с помощью команд RCVST.



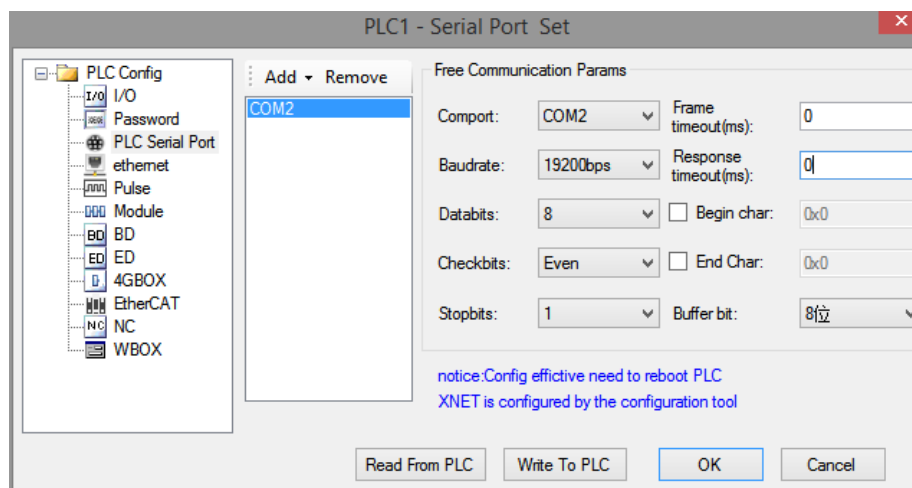
6.4.5 Пример связи в свободном формате

Пример 1:

В главе 6-3-3 мы привели пример связи между ПЛК RROMPOWER и терморегулятором, объяснив, почему следует использовать связь в свободном формате. Ниже приведен пример.

Последовательность действий:

- Сначала подключим оборудование. В данном случае мы используем последовательный порт 2 ПЛК для связи, то есть контакт 485 + на приборе подключен к клемме А выходного порта ПЛК, а контакт 485- на приборе подключен к клемме В выходного порта ПЛК.
- Настроим параметры последовательного порта ПЛК в соответствии с параметрами связи терморегулятора. Заданы следующие параметры. После настройки параметров питание можно выключить и снова включить.



3. Создайте программу согласно описанию в главе 6-3-3.

Считывание данных о температуре: « : » «R» «T» «CR»

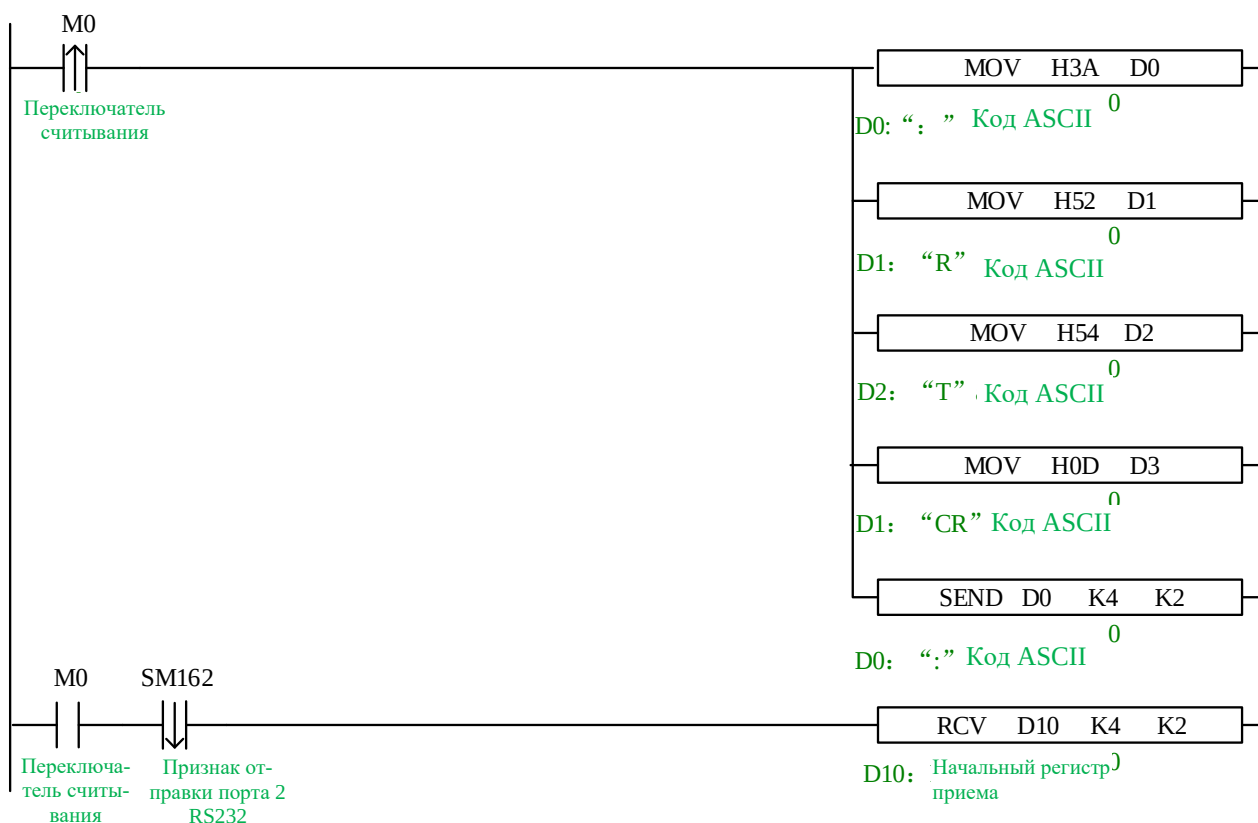
« : » ----- начало данных

«R» ----- считывание

«T» ----- температура

«CR» ----- ввод, конец данных

Программа:



При попытке установления связи между ПЛК и другими интеллектуальными устройствами рекомендуется использовать инструмент отладки последовательного порта для определения формата данных связи, то есть протокола. Преимущества этого метода: инструмент отладки последовательного порта легко трансформируется и универсален в использовании; после того, как инструмент отладки последовательного порта определит, что связь может быть успешной, программа ПЛК будет записана в соответствии с полученным форматом данных. Это дает в два раза лучший результат при вдвое меньших усилиях.

Фактически, протокол Modbus-RTU можно рассматривать как особый вид свободного протокола. Соотношение между ними похоже на соотношение между

эллипсом и окружностью. Можно попытаться использовать свободный формат, чтобы реализовать функции команд Modbus.

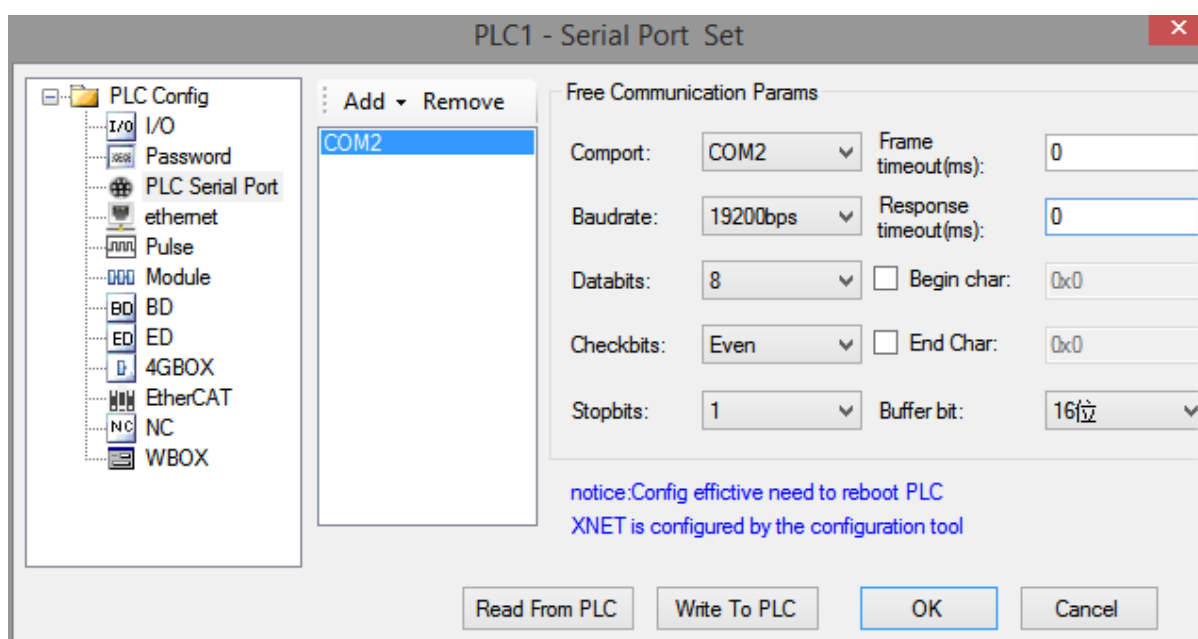
Пример 2:

Значения пяти регистров ПЛК серии RMP20 отправляются в D1-D5 другого ПЛК.

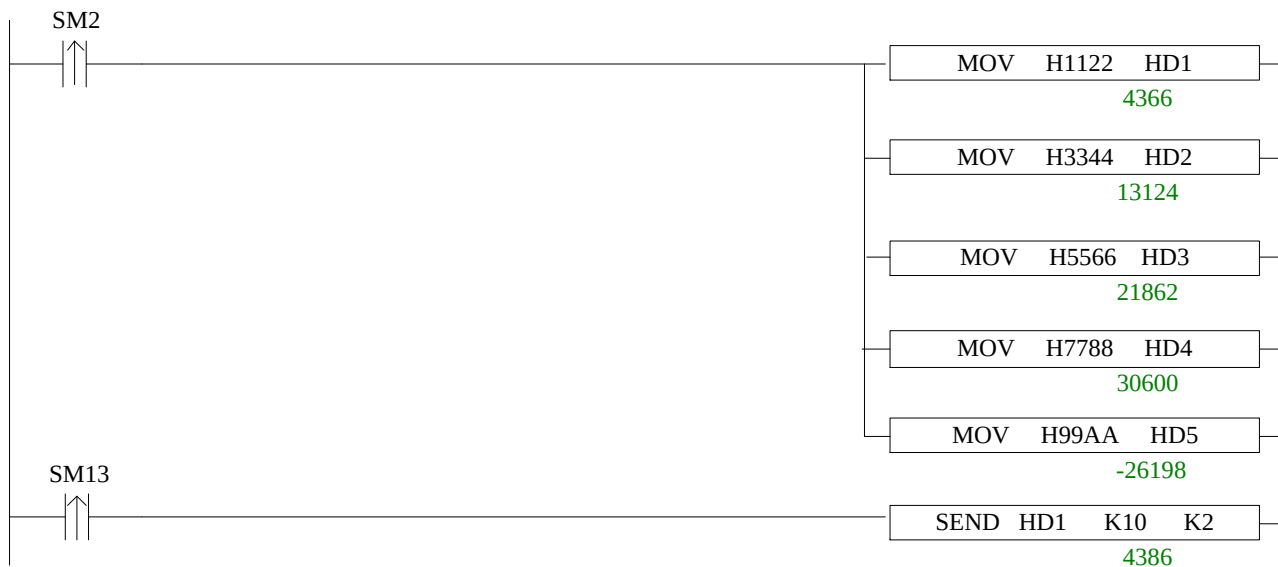
Если пользователь понимает принцип связи по Modbus, он может использовать режим связи по протоколу Modbus-RTU, пока не напишет команды «Запись данных нескольких регистров (MRGW)» в хосте. Мы сделаем это в режиме связи в свободном формате.

Последовательность действий:

1. Сначала подключим оборудование. В данном случае мы используем последовательный порт 2 ПЛК для связи, то есть соединяем клеммы А и В двух ПЛК.
2. Задайте одинаковые параметры последовательного порта для двух ПЛК. Заданы следующие параметры. После настройки параметров питание можно выключить и снова включить.



3. Программа для ПЛК серии RMP20:



Программа для другого ПЛК:



Иногда данные пользовательской связи хранятся в нескольких регистрах в виде кода ASCII. Пользователям необходимо извлечь это значение, сохранить его в регистре и отобразить на ЧМИ. Для этого клиенты часто используют команды HEX (преобразование ASCII в шестнадцатеричный формат). Но команды HEX сложны в использовании и понимании. Зачастую, мы не используем эту команду. Взаимосвязь между значениями представлена в таблице сравнения кодов ASCII.

Таблица кодов ASCII:

Значение ASCII	Управля- ющий символ	Значение ASCII	Управля- ющий символ	Значение ASCII	Управля- ющий символ	Значение ASCII	Управля- ющий символ
0	NUT	32	(пробел)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	D
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f

Значение ASCII	Управляющий символ	Значение ASCII	Управляющий символ	Значение ASCII	Управляющий символ	Значение ASCII	Управляющий символ
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	°	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	об
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	-	127	DEL

Пример 3:

Регулятор давления обменивается данными с ПЛК в режиме связи в свободном формате для осуществления сбора данных. На регуляторе давления отображается значение -0,7814 МПа. Значение, полученное ПЛК, сохраняется в D0, и поочередно сохраняются семь регистров. Однако значение комбинации из семи регистров нужно будет извлечь и сохранить в D46 в десятичном формате.

С помощью функции контроля данных ПЛК можно контролировать коды ASCII в регистрах D0~D6 следующим образом:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	-	0	.	7	8	1	4			
D10										
D20										
D30										
D40										

Dec Bin Hex Unsigned **ASCII**

После перехода на десятичный формат откроется показанное ниже окно:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
D0	45	48	46	55	56	49	52	0	0	0
D10	0	0	0	0	0	0	0	0	0	0
D20	0	0	0	0	0	0	0	0	0	0
D30	0	0	0	0	0	0	0	0	0	0
D40	0	0	0	0	0	0	0	0	0	0

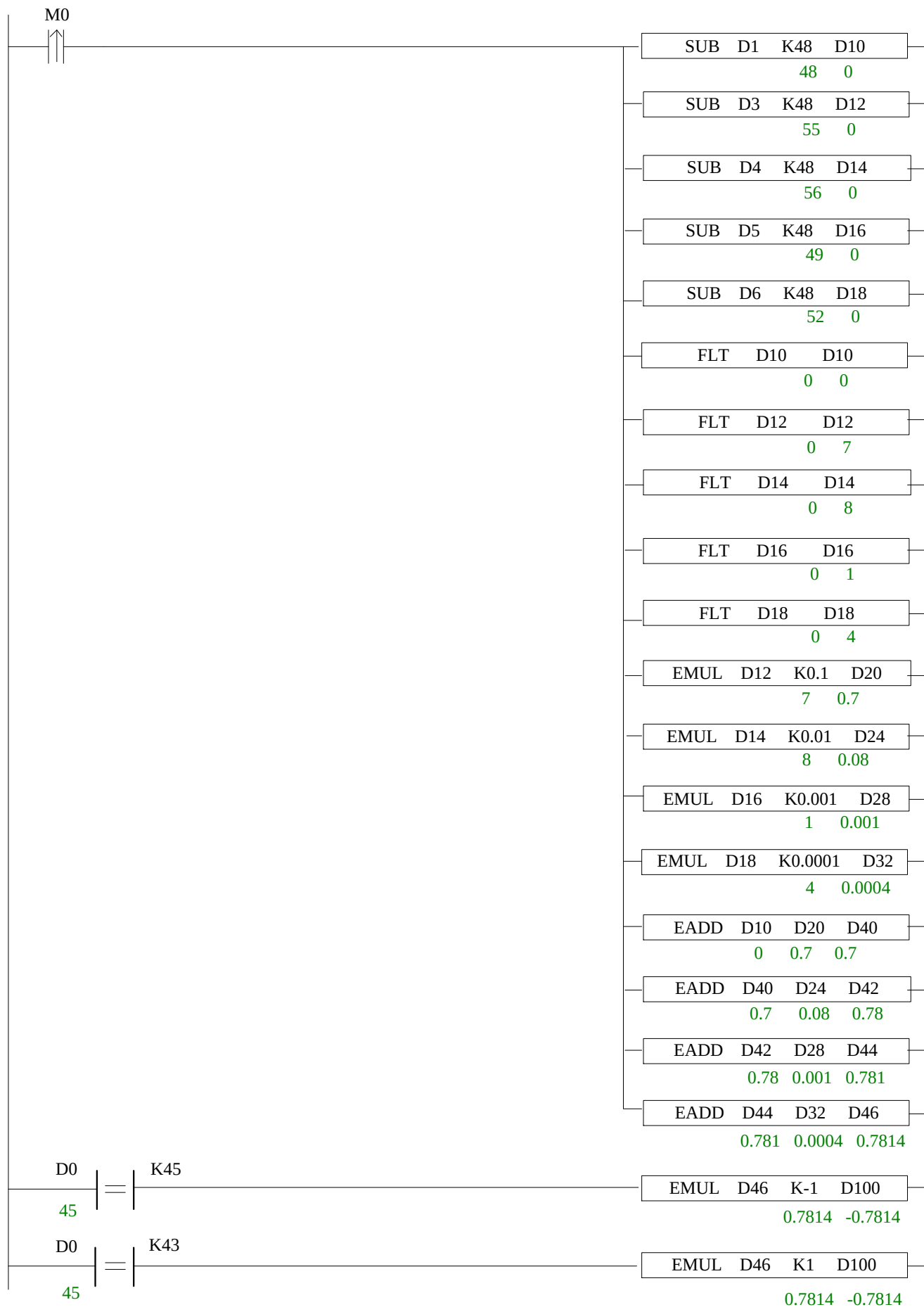
Dec Bin Hex Unsigned ASCII

Сравнивая взаимосвязь между кодами ASCII и десятичными значениями, можно установить правило, согласно которому существует 48 различий между кодами ASCII в D1, D3, D4, D5, D6 и десятичными значениями. Окончательные десятичные значения получаются путем вычитания значений в регистрах из K48 и умножения полученного результата на 10. Формула имеет следующий вид:

$$D46 = (D1 - 48) \cdot 1 + (D3 - 48) \cdot 0.1 + (D4 - 48) \cdot 0.01 + (D5 - 48) \cdot 0.001 + (D6 - 48) \cdot 0.0001$$

D0 — это символный бит. Просмотрев таблицу, можно сделать вывод, что когда D0 = K45, это отрицательное значение, а когда D0 = K43, это положительное значение.

Лестничная диаграмма представлена ниже:



6.5 Признак и регистр связи

Признак связи

Последовательный порт	Адрес регистра	Функция	Описание
Порт 0	SM140	Признак выполнения команды считывания и записи Modbus	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM142	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM143	Признак приема данных при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы
Порт 1	SM150	Признак выполнения команды считывания и записи Modbus	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM152	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM153	Признак приема данных при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы
Порт 2	SM160	Признак выполнения команды считывания и записи Modbus	Включен, когда начинается выполнение команды

Последовательный порт	Адрес регистра	Функция	Описание
			Выключен, когда выполнение команды завершено
	SM162	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM163	Признак приема данных при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы
Порт 3	SM170	Признак выполнения команды считывания и записи Modbus	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM172	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM173	Признак приема данных при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы
Порт 4	SM180	Признак выполнения команды считывания и записи Modbus	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM182	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено



Последовательный порт	Адрес регистра	Функция	Описание
	SM183	Признак приема данных при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы
Порт 5	SM190	Признак выполнения команды считывания и записи Modbus	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM192	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM193	Признак приема данных при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы

Регистры связи

	№	Функция	Описание
Порт 0	SD140	Результаты выполнения команды считывания и записи Modbus	0: правильно 100: ошибка при приеме данных 101: превышено время приема 180: ошибка CRC 181: ошибка LRC 182: ошибка номера станции 183: переполнение пересылочного буфера 400: ошибка кода функции 401: ошибка адреса

	№	Функция	Описание
			402: ошибка длины 403: ошибка в данных 404: ведомая станция занята 405: ошибка памяти (Ошибка стирания FLASH-памяти)
	SD141	Результат связи по шине X-Net	0: правильно 1: превышено время связи 2: ошибка памяти 3: ошибка CRC при приеме данных
	SD142	Результат отправки при связи в свободном формате	0: правильно 410: переполнение буфера при связи в свободном формате
	SD143	Результат получения при связи в свободном формате	0: правильно 410: переполнение при отправке длины данных 411: прием коротких данных 412: прием длинных данных 413: ошибка при приеме данных 414: превышено время приема 415: стартовый символ отсутствует 416: стоповый символ отсутствует
	SD144	Количество полученных данных при связи в свободном формате	Считается в байтах, не включает в себя стартовый символ и стоповый символ
			
	SD149		
Порт 1	SD150	Результаты выполнения	0: правильно 100: ошибка при приеме данных



	№	Функция	Описание
		команды считывания и за- писи Modbus	101: превышено время приема 180: ошибка CRC 181: ошибка LRC 182: ошибка номера станции 183: переполнение пересылочного бу- фера 400: ошибка кода функции 401: ошибка адреса 402: ошибка длины 403: ошибка в данных 404: ведомая станция занята 405: ошибка памяти (Ошибка стира- ния FLASH-памяти)
	SD151	Результат связи по шине X- Net	0: правильно 1: превышено время связи 2: ошибка памяти 3: ошибка CRC при приеме данных
	SD152	Результат отправки при связи в свободном формате	0: правильно 410: переполнение буфера при связи в свободном формате
	SD153	Результат получения при связи в свободном фор- мате	0: правильно 410: переполнение при отправке длины данных 411: прием коротких данных 412: прием длинных данных 413: ошибка при приеме данных 414: превышено время приема 415: стартовый символ отсутствует 416: стоповый символ отсутствует

	№	Функция	Описание
	SD154	Количество полученных данных при связи в свободном формате	Считается в байтах, не включает в себя стартовый символ и стоповый символ
			
	SD159		
Порт 2	SD160	Результаты выполнения команды считывания и записи Modbus	0: правильно 100: ошибка при приеме данных 101: превышено время приема 180: ошибка CRC 181: ошибка LRC 182: ошибка номера станции 183: переполнение пересылочного буфера 400: ошибка кода функции 401: ошибка адреса 402: ошибка длины 403: ошибка в данных 404: ведомая станция занята 405: ошибка памяти (Ошибка стирания FLASH-памяти)
	SD161	Результат связи по шине X-Net	0: правильно 1: превышено время связи 2: ошибка памяти 3: ошибка CRC при приеме данных
	SD162	Результат отправки при связи в свободном формате	0: правильно 410: переполнение буфера при связи в свободном формате



	№	Функция	Описание
	SD163	Результат получения при связи в свободном формате	0: правильно 410: переполнение при отправке длины данных 411: прием коротких данных 412: прием длинных данных 413: ошибка при приеме данных 414: превышено время приема 415: стартовый символ отсутствует 416: стоповый символ отсутствует
	SD164	Количество полученных данных при связи в свободном формате	Считается в байтах, не включает в себя стартовый символ и стоповый символ
			
	SD169		
Порт 3	SD170-SD179		
Порт 4	SD180-SD189		
Порт 5	SD190-SD199		

6.6 Считывание и запись параметров последовательного порта

Параметры связи можно изменять через панель настройки последовательного порта. Изменение параметров также можно выполнить через команду считывания [CFGCR] параметров последовательного порта и команду записи [CFG CW] параметров последовательного порта.

6.6.1 Считывание параметров последовательного порта [CFGCR]

1) Обзор

Считывание параметров последовательного порта в указанные локальные регистры.

Считывание параметров последовательного порта [CFGCR]			
16-битная команда	CFGCR	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, Запуск по нарастающему фронту	Совместимые модели	PMP20
Аппаратное обеспечение	-	Программное обеспечение	Версия 3.4 и более поздние версии

2) Операнды

Операнды	Функция	Тип
D	Начальный адрес локального регистра	16-битный, BIN
S1	Считывание номеров параметров последовательного порта	16-битный, BIN
S2	Номер последовательного порта:	16-битный, BIN

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m	
D	•																		
S1	•	•							•										
S2	•								К										

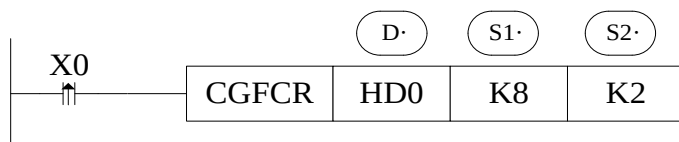
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Оператор S1: Количество регистров, используемых для считывания параметров последовательного порта, — как правило, 8 (у ПЛК серии RMP20 — 9).
- Оператор S2: Диапазон последовательных портов: K0 ~ K5. K0: Порт0, K1: Порт1, K2: Порт2 или Порт2-RS232 или Порт2-RS485, K3: Порт3, K4: Порт4, K5: Порт5.
- Считывание 8 параметров последовательного порта 2 в HD0~HD7. Имена и определения конкретных параметров приведены в разделе 6-5-3.

6.6.2 Запись параметров последовательного порта [CFGCW]

1) Обзор

Запись значения указанного локального регистра в определенный последовательный порт.

Запись параметров последовательного порта [CFGCW]			
16-битная команда	CFGCW	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, Запуск по нарастающему фронту	Совместимые модели	RMP20
Аппаратное обеспечение	-	Программное обеспечение	Версия 3.4 и более поздние версии

2) Операнды

Операнды	Функция	Тип
S1	Начальный адрес локального регистра	16-битный, BIN
S2	Запись номеров параметров последовательного порта	16-битный, BIN

S3	Номер последовательного порта:	16-битный, BIN
----	--------------------------------	----------------

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m
S1	•																	
S2	•	•							•									
S3	•								К									

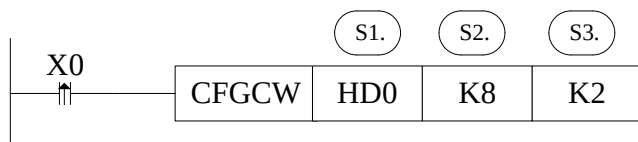
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Оператор S2: Количество регистров, используемых для записи параметров последовательного порта, — как правило, 8 (у ПЛК серии PMP20 — 9).
- Оператор S3: Диапазон последовательных портов: K0 ~ K5. K0: Порт0, K1: Порт1, K2: Порт2 или Порт2-RS232 или Порт2-RS485, K3: Порт3, K4: Порт4, K5: Порт5.
- Запись параметров HD0~HD7 в последовательный порт 2. Имена и определения конкретных параметров приведены в разделе 6-5-3.



6.6.3 Имена и настройки параметров последовательного порта

Исходя из того, что HD0-HD14 соответствуют параметрам последовательного порта, имена и настройки параметров, представленные регистрами, показаны в таблице ниже.

Адрес пара- метра	Имя и уставки параметров				
	Связь через MODBUS (HD0 = 1)	Связь в свобод- ном формате (HD0 = 2)	Связь по шине X-NET		Связь по Ethernet (HD0 = 3)
			OMMS (HD0 = 3)	TBN (HD0 = 3)	
HD0	Тип сети 1: MODBUS; 2: связь в свободном формате; 3: X-NET; 4: Протокол MODBU-TCP				
HD1	MODBUS станция № 1-254	Скорость передачи данных, см. в Таблице 1	Идентификатор сети 0~32767	Идентификатор сети 0~32767	Идентифика- тор сети IP-адрес, 2 старших байта
HD2	Режим пере- дачи 0: RTU 128: ASCII	Формат кадра, см. таблицу 2	Номер станции 0~100	Номер станции 0~100	Номер стан- ции IP-адрес, 2 младших байта
HD3	Скорость пере- дачи данных, см. таблицу 1	Характеристики связи в свободном формате бит7: 1: со стартовым символом 0: стартовый сим- вол отсутствует бит6: 1: со стоповым символом 0: стоповый сим- вол отсутствует	Тип физического уровня 1: PHY_RS485 2: PHY_SOF (Однонаправленная волоконно- оптическая кольцевая сеть) 3: PHY_OFPP (Оптоволоконная точечная сеть) 4: PHY_RS232 5: PHY_RS422 6: PHY_TTL (силовая сеть с TTL)		
HD4		Формат кадра см. Таблицу 2	Стартовый символ	Тип канального уровня 0: TBN 1: HDN 2: CCN 3: PPFD 4: PPU 5: Ethernet	

Адрес параметра	Имя и уставки параметров				
	Связь через MODBUS (HD0 = 1)	Связь в свободном формате (HD0 = 2)	Связь по шине X-NET		Связь по Ethernet (HD0 = 3)
			OMMS (HD0 = 3)	TBN (HD0 = 3)	
HD5	число повторных попыток 0~5	Стоповый символ	Свойства OMMS 128: Поддержка периодической связи, в противном случае не поддерживается	Скорость передачи данных, см. таблицу 1	Маска подсети, 2 старших байта
HD6	Тайм-аут ответа 0~65535	Тайм-аут кадра 0~255	Скорость передачи данных OMMS, см. таблицу 1	Время цикла токена 1-60000 (мс)	Маска подсети, 2 младших байта
HD7	Задержка перед отправкой 0~255	Тайм-аут ответа 0~65535 (0 — бесконечное ожидание)	Перечень ведомых станций OMMS Каждый бит каждого байта в массиве указывает, доступна ли ведомая станция (главная станция действует, т.е. количество станций — 1).	Макс. количество станций 1-100	Адрес шлюза, 2 старших байта
HD8	-	-	-	-	Адрес шлюза, 2 младших байта

Примечание:

Таблица не содержит «буферных цифр» в режиме связи в свободном формате, поэтому «буферные цифры» не могут считываться и записываться с помощью команд CFGCR и CFGCW, но их можно считывать и записывать с помощью команд MOV. Адрес «буферных цифр» указан в Приложении 3.

Таблица 1: Скорость передачи данных

Значение	Скорость передачи данных	Значение	Скорость передачи данных	Значение	Скорость передачи данных	Значение	Скорость передачи данных
1	300 бит/с	7	19200 бит/с	13	256000 бит/с	19	1000000 бит/с
2	600 бит/с	8	28800 бит/с	14	288000 бит/с	20	1200000 бит/с
3	1200 бит/с	9	38400 бит/с	15	384000 бит/с	21	1500000 бит/с
4	2400 бит/с	10	57600 бит/с	16	512000 бит/с	22	2400000 бит/с
5	4800 бит/с	11	115200 бит/с	17	576000 бит/с	23	3000000 бит/с
6	9600 бит/с	12	192000 бит/с	18	768000 бит/с		

Таблица 2: Формат кадра

Стоповый бит		Бит четности			Длина бита данных		
Бит7	Бит6	Бит5	Бит4	Бит3	Бит2	Бит1	Бит0
00: 1		000: нет			000: 5		
01: 1.5		001: нечетность			001: 6		
10: 2		010: четность			010: 7		
		011: не задействовано			011: 8		
		100: Маска			100: 9		

7. Функция ПИД-регулирования

В этой главе мы познакомимся с применением команд ПИД-регулирования в ПЛК серии RMP20, включая: вызов команд, настройку параметров, элементы, на которые нужно обращать внимание, примеры программ, и т. д.

7.1 Введение в ПИД

Команды ПИД-регулирования и функция автоматической регулировки были добавлены в основные блоки ПЛК серии RMP20. С помощью функции автоматической регулировки пользователи могут задать оптимальные период выборки и параметры ПИД-регулирования, а также повысить точность управления.

Команды ПИД-регулирования оказались очень удобными для пользователей.

Выходные данные могут быть в форматах D, HD и в виде количества включений-выключений Y. Пользователь может легко настроить их при программировании.

С помощью функции автоматической регулировки пользователи могут задать оптимальные период выборки и параметры ПИД-регулирования, а также повысить точность управления.

Пользователь может выбирать положительное или отрицательное действие через соответствующие настройки в ПО. Положительное действие используется для регулирования нагрева, а отрицательное действие — для регулирования охлаждения.

ПИД-регулирование разделяет основные блоки с расширениями, что повышает универсальность этой функции.

ПЛК серии RMP20 поддерживают два метода автоматической регулировки: метод ступенчатого отклика и метод критических колебаний.

Для объекта терморегулирования:

Метод ступенчатого отклика: автоматическая регулировка ПИД-регулятора начнется, когда текущая температура регулируемого объекта станет равна температуре окружающей среды.

Метод критических колебаний: автоматическая регулировка ПИД-регулятора может начаться при любой температуре.

7.2 Формат команды

1) Обзор

Выполнение команд ПИД-регулирования с данными в указанных регистрах.

ПИД-регулирование [PID]			
16-битная команда	ПИД	32-битная команда	-
Условия выполнения	нормально-разомкнутая/нормально-замкнутая катушка, триггер	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	версия 3.2 или более поздняя версия

2) Операнды

Операнды	Функция	Тип
S1	Настройка адреса заданного значения (SV)	16 бит, BIN
S2	Настройка адреса проверенного значения (PV)	16 бит, BIN
S3	Настройка начального адреса параметров регулирования	16 бит, BIN
D	Адрес результата операции (MV) или выходного порта	16 бит, BIN; бит

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Константа	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Идентификатор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•							•									
S2	•	•																
S3	•	•																
D	•	•											•	•	•	•	•	

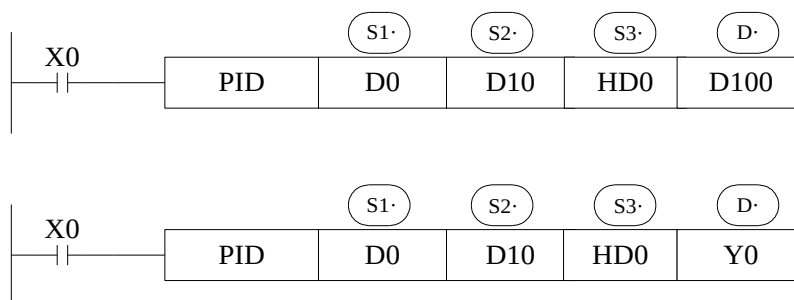
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

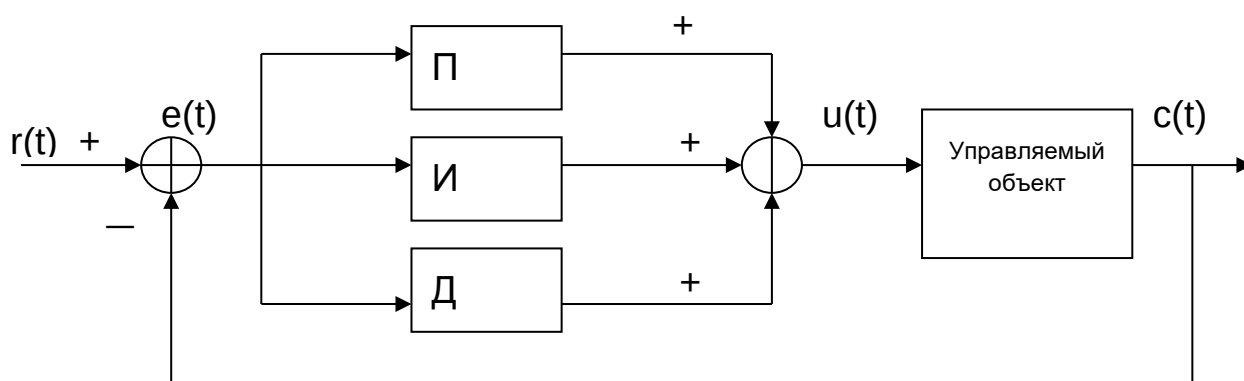
S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Эта команда будет занимать S3~S3+69, поэтому не используйте их в качестве общих регистров данных.
- Эта команда выполняется при наступлении каждого интервала времени выборки.
- Что касается результата операции, регистры данных используются для хранения выходных значений ПИД-регулятора; точки вывода используются для вывода коэффициента заполнения в форме ON/OFF (ВКЛ/ВЫКЛ).
- Правила ПИД-регулирования представлены ниже:

П: пропорциональное, И: интегральное, Д: дифференциальное



Аналоговая система ПИД-регулирования

$$e(t) = r(t) - c(t) \quad (1-1)$$

$$u(t) = K_p[e(t) + 1/T_i \int e(t)dt + T_D de(t)/dt] \quad (1-2)$$

Здесь $e(t)$ — значение смещения, $r(t)$ — значение уставки, $c(t)$ — фактическое выходное значение, а $u(t)$ — контрольное значение.

В функции (1-2) K_p — коэффициент пропорциональности, T_i — коэффициент времени интегрирования, а T_D — дифференциальный коэффициент времени.

Результат операции:

1. Аналоговый выход: цифровая форма $MV = u(t)$, диапазон по умолчанию — 0-4095.
2. Цифровой выход: $Y = T \cdot [\text{верхний предел выходного сигнала } MV / PID]$. Y — время включения выходов в пределах цикла регулирования. T — цикл регулирования, равный времени выборки. Значение верхнего предела выходного сигнала ПИД-регулирования по умолчанию — 4095.

7.3 Настройка параметров

Пользователи могут вызывать функцию ПИД-регулирования прямо в программном обеспечении PROMPOWER PLC Studio и задавать соответствующие параметры в окне (см. рисунок ниже). Подробную информацию см. в руководстве пользователя PROMPOWER PLC Studio. Пользователи также могут записывать параметры в указанные регистры с помощью команд MOV перед операцией ПИД-регулирования.

PID Instruction Parameter Config

Target Value (SV): D0 Measure Value (PV): D10 Parameter: HD0 Output: Y0

Parameter Config

☒ Manual ☐ Auto

Sampling Time: 0 ms

Proportion Gain (Kp): 0 %

Integration Time (Ti): 0 *100ms

Differential Time (Td): 0 *10ms

PID Computation Scope: 0

PID Control Death Band: 0

Self Study Periodic Value: 0

Self Study Method: Step Response

Self Study PID Control Mode: PID Control

Mode Config

☒ Common Mode ☐ Advanced Mode

Input Filter Constant (a): 0 %

Differential Increase (KD): 50 %

Output Upper Limit Value: 4095

Output Lower Limit Value: 0

Direction Config

☒ Negative Movement ☐ Positive Movement

Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Parameter Range: HD0 - HD69

Overshoot Config

☒ Enable Overshoot ☐ Disable Overshoot

Each time adjust the increase: 100 %

Current target value resident Count: 15

Suggestion value

Read From PLC Write To PLC OK Cancel

Режим автоматической регулировки:

PID Instruction Parameter Config

Target Value (SV): D0 Measure Value (PV): D10 Parameter: HD0 Output: Y0

Parameter Config

☐ Manual ☒ Auto

Sampling Time: 0 ms

Proportion Gain (KP): 0 %

Integration Time (TI): 0 *100ms

Differential Time (TD): 0 *10ms

PID Computation Scope: 0

PID Control Death Band: 0

Self Study Periodic Value: 0

Self Study Method: Step Response

Self Study PID Control Mode: PID Control

Mode Config

☒ Common Mode ☐ Advanced Mode

Input Filter Constant (a): 0 %

Differential Increase (KD): 50 %

Output Upper Limit Value: 4095

Output Lower Limit Value: 0

Direction Config

☒ Negative Movement ☐ Positive Movement

Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Overshoot Config

☒ Enable Overshoot ☐ Disable Overshoot

Each time adjust the increase: 100 %

Current target value resident Count: 15

Suggestion value

Read From PLC Write To PLC OK Cancel

Parameter Range: HD0 - HD69

В программном обеспечении версии 3.2 и более поздних версий можно выбрать режим автоматической регулировки: метод ступенчатого отклика или метод критических колебаний.

7.3.1 Регистры и их функции

В таблице ниже представлены идентификаторы относительных параметров команд ПИД-регулирования.

Идентификатор	Функция	Описание	Примечание
S3	Период выборки	Все параметры должны быть заданы независимо от режима (ручной или автоматический)	32 бита без знака, Единица измерения: мс
S3+2	Настройка режима	бит0: 0: отрицательное действие; 1: положительное действие	

Идентификатор	Функция	Описание	Примечание
		Бит1-бит6 не используются бит7: 0: ручной режим ПИД-регулятора; 1: автоматическая регулировка ПИД-регулятора бит8: 1: признак успешного выполнения автоматической регулировки бит9-бит10: метод автоматической регулировки 00: ступенчатый отклик 01: критические колебания бит11-бит12: не используются бит13-бит14 в режиме автоматической регулировки ПИД-регулятора (действует в режиме критических колебаний) 00: ПИД-регулирование 01: ПИ-регулирование 10: П-регулирование бит15: 0: нормальный режим; 1: расширенный режим	
S3+3	Пропорциональное усиление (Kp)	0~32767 [%]	
S3+4	Время интегрирования (TI)	0~32767 [единица измерения: 100 мс]	0 принимается за отсутствие интегрального регулирования
S3+5	Дифференциальное время (TD)	0~32767 [единица измерения: 10 мс]	0 принимается за отсутствие дифференциального регулирования
S3+6	Рабочая зона ПИД-регулирования	0~32767	Ширина диапазона регулирования при ПИД-регулировании

Идентификатор	Функция	Описание	Примечание
S3+7	Зона нечувствительности	0~32767	Выходное значение ПИД-регулирования не изменяется в зоне нечувствительности
S3+8	Коэффициент фильтрации температуры выборки	0~100 [%]	Фильтрация входной температуры выборки в расширенном режиме, 0 — входной фильтр отсутствует
S3+9	Дифференциальное усиление (KD)	0~100 [%]	Только для расширенного режима (значение по умолчанию в нормальном режиме составляет 50%), 0 — дифференциальное усиление отсутствует
S3+10	Верхнее предельное значение выходного сигнала	0~32767	
S3+11	Нижнее предельное значение выходного сигнала	0~32767	
S3+12	Изменение температуры блока соответствует изменению значения AD	Значение AD полной шкалы · (0,3~1%) Значение по умолчанию — 10.	16 бит, без знака Только для метода ступенчатого отклика при ПИД-регулировании
S3+13	Превышение при автоматической регулировке ПИД-регулятора	0: превышение разрешено 1: превышение запрещено (попытка уменьшить превышение)	Только для метода ступенчатого отклика при ПИД-регулировании
S3+14	Текущее заданное значение корректируется в процентах каждый раз на этапе конечного перехода при автоматической регулировке	Нельзя откорректировать	16 бит, без знака Только для метода ступенчатого отклика при ПИД-регулировании
S3+15	Кратность превышения заданного значения на этапе конечного перехода при автоматической регулировке при ограничении превышения		Только для метода ступенчатого отклика при ПИД-регулировании, Значение по умолчанию — 15.

Идентификатор	Функция	Описание	Примечание
S3+16	Тип и статус ПИД-регулирования	Бит0~бит1: 00: Ручной режим 01: Режим ступенчатого отклика 10: Режим критических колебаний Бит 8: 0: Состояние ручного регулирования 1: Завершение автоматической регулировки, переход в состояние ручного регулирования	Внутренние параметры использования системы только для контроля
S3+17	Макс. выходное значение ПИД-регулирования	0~32767	Внутренние параметры использования системы только для контроля
S3+18	Мин. выходное значение ПИД-регулирования	0~32767	Внутренние параметры использования системы только для контроля
S3+19	Последний период выборки	0~период выборки (единица измерения: мс)	16 бит, без знака. Внутренние параметры использования системы только для контроля
S3+20	Фактическая продолжительность периода выборки	Значение примерно равно периоду выборки	32 бит, без знака. Внутренние параметры использования системы только для контроля
S3+22	Последний раз, когда пользователь настраивал заданную температуру	Значение до изменения заданной температуры	Внутренние параметры использования системы только для контроля
S3+23	-	-	Параметр зарезервирован
Ниже приводится общий адрес (делится на настройку режима ступенчатого отклика, режима критических колебаний и ручное регулирование)			
Режим ступенчатого отклика (параметры только для чтения, только для контроля)			
S3+24	Фактическая продолжительность выборки	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы

Иден-тифи-катор	Функция	Описание	Примечание
S3+26	Рабочий сегмент автоматической регулировки ПИД-регулятора	0: Этап подготовки 1~2: сбор параметров автоматической регулировки 3: Расчет параметров ПИД-регулирования	Внутренние параметры использования системы
S3+28	Продолжительность автоматической регулировки рабочих параметров ПИД-регулирования	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+30	Накопление в режиме реального времени двух точек перегиба	Сброс и повторный расчет времени после достижения точки перегиба в диапазоне 0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+32	Вариация элементов выборки точки перегиба	Выборочная разность между двумя точками перегиба -2147483648~2147483647	Внутренние параметры использования системы
S3+34	Интервал выборки точки перегиба ЕК	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+36	Время от автоматической регулировки при ПИД до точки перегиба	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+38	Последняя температура выборки	-32767~32767	Внутренние параметры использования системы
S3+39	Время от операции автоматической регулировки ПИД-регулятора до точки перегиба	-32767~32767 (единица измерения: мс)	Внутренние параметры использования системы
S3+40	Начальное значение выборки операции автоматической регулировки ПИД-регулятора	-32767~32767	Внутренние параметры использования системы
S3+41	Количество подходов к точке перегиба	0~65535	Внутренние параметры использования системы



Идентификатор	Функция	Описание	Примечание
	во время автоматической регулировки		
S3+42	Время бездействия	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+44	Температура остановки	Температура в конце автоматической регулировки Диапазон: -32767~32767	Внутренние параметры использования системы
Режим критических колебаний (параметры только для чтения, только для контроля)			
S3+24	Режим ПИД-регулирования	0: ПИД-регулирование 1: ПИ-регулирование 2: П-регулирование	16 бит, без знака, внутренние параметры использования системы
S3+25	Текущий сегмент автоматической регулировки	0: Этап подготовки 1: Запуск автоматической регулировки 2~3: Сбор параметров автоматической регулировки 4: Расчет параметров ПИД-регулирования	16 бит, без знака, внутренние параметры использования системы
S3+26	Температура автоматического регулирования возникает у нескольких пиковых значений	0: первое пиковое значение 1: второе пиковое значение	16 бит, без знака, внутренние параметры использования системы
S3+27	Минимальная температура выборки	-32767~32767	Внутренние параметры использования системы
S3+28	Максимальная температура выборки	-32767~32767	Внутренние параметры использования системы
S3+30	Период выборки минимальной температуры выборки	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+32	Период выборки максимальной температуры выборки	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы



Иден-тифи-катор	Функция	Описание	Примечание
S3+34	Суммарное время автоматической регулировки	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
Режим ручного регулирования (параметры только для чтения, только для контроля)			
S3+24	Текущая заданная температура	-32767~32767	Внутренние параметры использования системы
S3+25	Требуется обновление заданной температуры	0: не требуется 1: требуется	16 бит, без знака, внутренние параметры использования системы
S3+26	Количество раз, когда нужно достичь заданной температуры	0~65535	Внутренние параметры использования системы
S3+27	Верхний предел ПИД-регулирования рабочего диапазона	-32767~32767	Внутренние параметры использования системы
S3+28	Нижний предел ПИД-регулирования рабочего диапазона	-32767~32767	Внутренние параметры использования системы
S3+30	Время возникновения высокого напряжения, когда ПИД-регулятор использует Y как точку вывода	0~4294967296 (единица измерения: мс)	Внутренние параметры использования системы
S3+32	Температура выборки после последней фильтрации	Отфильтрованное значение температуры в последний период выборки (сначала необходимо задать константу входного фильтра в расширенном режиме)	Значение с плавающей точкой, внутренние параметры использования системы
S3+34	Последнее отклонение температуры		Значение с плавающей точкой, внутренние параметры использования системы
S3+36	Значение последней интегральной составляющей	Цифровое значение, соответствующее значению U_i последнего периода выборки	Значение с плавающей точкой, внутренние параметры использования системы



Идентификатор	Функция	Описание	Примечание
S3+38	Значение последней дифференциальной составляющей	Цифровое значение, соответствующее значению U_d последнего периода выработки	Значение с плавающей точкой, внутренние параметры использования системы
S3+40	Последнее выходное значение ПИД-регулятора		Значение с плавающей точкой, внутренние параметры использования системы

Примечание:

При изменении режима автоматической регулировки на режим ручной регулировки значение в исходном адресе $S3 + 24 \sim S3 + 40$ будет перезаписано значением в режиме ручной регулировки.

7.3.2 Описание параметров

Направление перемещения

Положительное перемещение: Выходное значение MV будет увеличиваться с увеличением измеренного значения PV, обычно используемого для управления охлаждением.

Отрицательное перемещение: Выходное значение MV будет уменьшаться с увеличением измеренного значения PV, обычно используемого для управления нагревом.

Настройка режима

Общий режим

Диапазон регистров параметров: $S3 \sim S3+69$, а диапазон $S3 \sim S3+7$ должен задаваться пользователями

Регистры $S3+8 \sim S3+69$ заняты системой, пользователи их не могут использовать.

Расширенный режим

Диапазон регистров параметров: S3~S3+69, из которых S3~S3+7 и S3+8~S3+11 должны быть настроены пользователями; S3+16~S3+69 заняты системой, пользователи их не могут использовать.

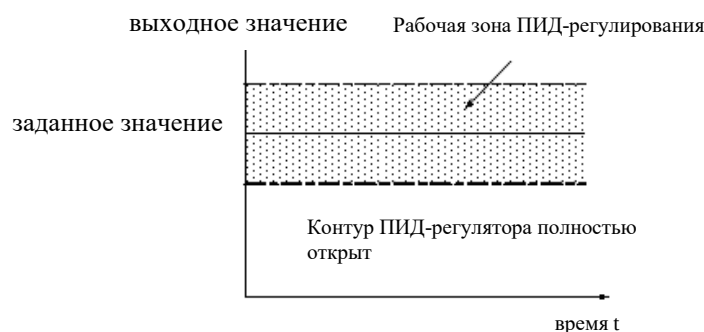
Период выборки [S3]

Система выбирает текущие значения через определенный промежуток времени и сравнивает их с выходным значением. Этот временной интервал является периодом выборки **T**. Нет требований к **T** во время вывода **DA**; значение **T** должно быть больше одного периода сканирования ПЛК во время вывода импульсов с порта. Значение **T** должно в 100-1000 раз превышать продолжительность периода сканирования ПЛК.

Рабочая зона ПИД-регулирования [S3+6]

В начале контур ПИД-регулятора полностью открыт и близок к заданному значению с самой высокой скоростью (значение по умолчанию 4095), и параметры K_p , T_I , T_D вступят в силу при входе в диапазон вычисления ПИД-регулирования.

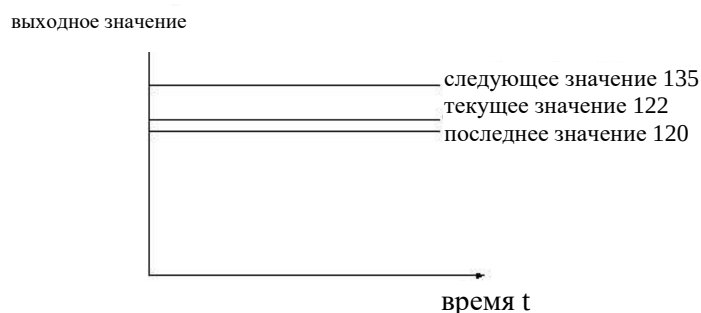
См. схему ниже:



Если заданное значение равно 100, то рабочая зона ПИД-регулирования равна 10, а фактическая рабочая зона ПИД-регулирования находится в диапазоне от 90 до 110.

Зона нечувствительности [S3+7]

Если измеренное значение в течение длительного времени изменилось незначительно, а ПИД-регулятор все еще находится в рабочем режиме, то проводить регулирование бессмысленно. Эту ситуацию можно преодолеть, настроив зону нечувствительности. См. схему ниже:



Предположим: значение в зоне нечувствительности составляет 10. Тогда на приведенном выше графике разность составляет всего 2, если сравнивать текущее значение с последним значением. ПИД-регулирование не будет выполняться; разность составляет 13 (больше, чем значение 10 в зоне нечувствительности), если сравнивать текущее значение со следующим значением, и это значение разности будет больше, чем значение в зоне нечувствительности. ПИД-регулирование будет выполняться со значением 135.

7.4 Режим автоматической регулировки

Если нет информации, как настроить параметры ПИД-регулятора, можно выбрать режим автоматической регулировки, в котором автоматически будут выбраны оптимальные параметры регулирования (период выборки, пропорциональное усиление **K_p**, интегральное время **T_i**, дифференциальное время **T_D**).

Режим автоматической регулировки подходит для следующих регулируемых объектов: температура, давление. Он не подходит для регулирования уровня и расхода жидкости.

Автоматическая регулировка — это процесс извлечения параметров ПИД-регулятора. Иногда в режиме автоматической регулировки не удастся найти оптимальные параметры за один раз. Поэтому автоматическую регулировку придется выполнить несколько раз. В нормальных условиях вибрация во время этого процесса не возникает. После того, как в конце автоматической регулировки будут найдены оптимальные параметры, перейдите в ручной режим ПИД-регулирования. Если объект управления ведет себя неустойчиво во время ручного ПИД-регулирования, и им нельзя управлять при постоянном заданном значении, причина может быть в неправильной регулировке параметров. Поэтому необходимо повторно отрегулировать параметры ПИД-регулирования для обеспечения устойчивости регулирования.

Метод ступенчатого отклика:

Пользователи могут настроить цикл выборки на 0 в начале процесса автоматической регулировки, а затем вручную изменить это значение с учетом практических потребностей после завершения процесса автоматического регулирования.

Метод ступенчатого отклика:

Перед выполнением автоматической регулировки система должна находиться в неконтролируемом устойчивом состоянии. Возьмем, к примеру, температуру: измеренная температура должна соответствовать температуре окружающей среды.

Метод критических колебаний:

В начале процесса автоматической регулировки необходимо задать период выборки. Для системы с медленным откликом: 1000 мс. Для системы с быстрым откликом: 10-100 мс.

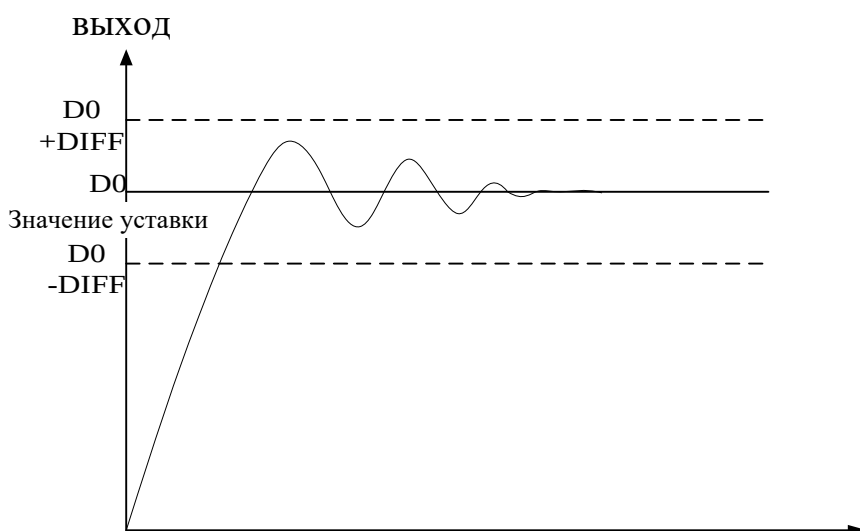
Метод критических колебаний:

Система может запустить процесс автоматической регулировки в любом состоянии. Текущая температура объекта не обязательно должна совпадать с температурой окружающей среды.

Два разных метода и схема ПИД-регулирования:

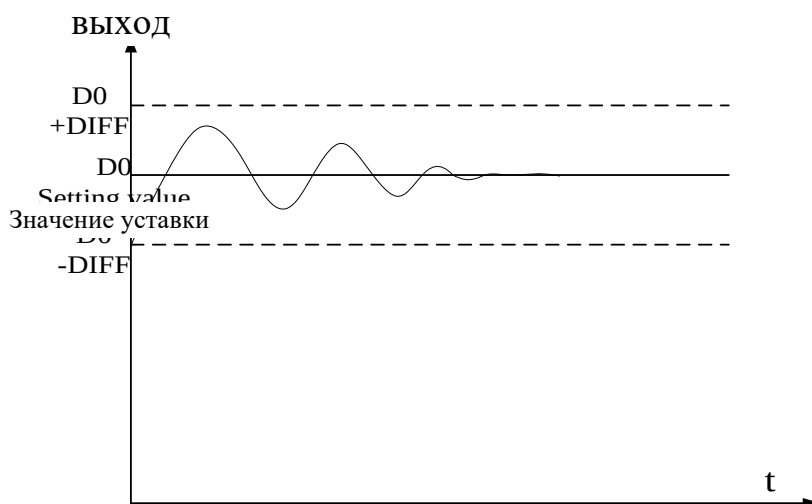
(1) Метод ступенчатого отклика

Убедитесь, что текущая температура равна температуре окружающей среды.



(2) Метод критических колебаний

Температурой начала автоматической регулировки может быть любое значение.



Чтобы войти в режим автоматической регулировки, установите бит7 (**S3 + 2**) на 1 и включите рабочий режим ПИД-регулятора. Если бит8 (**S3 + 2**) изменился на 1, это указывает на то, что автоматическая регулировка была успешно выполнена.

Значение периода автоматической регулировки ПИД-регулятора [S3 + 12]

Задайте это значение в команде S3 + 12 во время автоматической регулировки. Это значение определяет производительность автоматической регулировки. В большинстве случаев это значение является результатом AD, соответствующим одному стандартному испытанному блоку. По умолчанию выставляется значение 10. Рекомендуемый диапазон настройки: результат AD полной шкалы $\times 0,3\sim 1\%$.

Пользователям не нужно изменять это значение. Однако если система подвергается сильному внешнему воздействию, данное значение следует немного увеличить, чтобы избежать неправильной оценки положительного и отрицательного перемещения. Если это значение слишком большое, период ПИД-регулирования (период выборки), полученный в процессе автоматической регулировки, будет слишком длинным. Поэтому не устанавливайте это значение слишком большим.

Если у пользователей нет опыта, используйте значение по умолчанию 10, установите
 ※1 период выборки ПИД-регулятора (период управления) на 0 мс, затем запустите автоматическую регулировку.

Настройка разрешения на превышение при автоматической регулировке ПИД-регулятора [S3 + 13]

Если установлено значение 0, превышение разрешено, и система может постоянно изучать оптимальные параметры ПИД-регулятора. Однако в процессе автоматической регулировки обнаруженное значение может быть ниже или выше заданного значения, поэтому следует учитывать коэффициент безопасности.

Если установлено значение 1, превышение запрещено. Для объектов, к безопасной работе которых предъявляются строгие требования, например, сосуды под давлением, настройте [S3 + 13] на 1, чтобы испытательное значение не сильно превышало заданное значение.

В процессе, если бит8 [S3 + 2] изменился с 0 на 1, это означает, что автоматическая регулировка прошла успешно, и получены оптимальные параметры. Если бит8 [S3 + 2] по-прежнему равен 0, когда бит7 [S3 + 2] меняется с 1 на 0, это означает, что автоматическая регулировка завершена, но параметры не оптимальные, и их необходимо изменить вручную.

Текущее заданное значение корректируется в процентах каждый раз на этапе конечного перехода при автоматической регулировке [S3 + 14]

Этот параметр действует при условии, что для [S3 + 13] установлено значение 1.

При ПИД-регулировании после автоматической регулировки может возникнуть небольшое превышение. Рекомендуется уменьшить этот параметр, чтобы можно было контролировать превышение. Если это значение слишком мало, может возникать задержка ответа. Значение по умолчанию — 100%, что означает, что параметр не действует. Рекомендуемый диапазон — 50-80%.

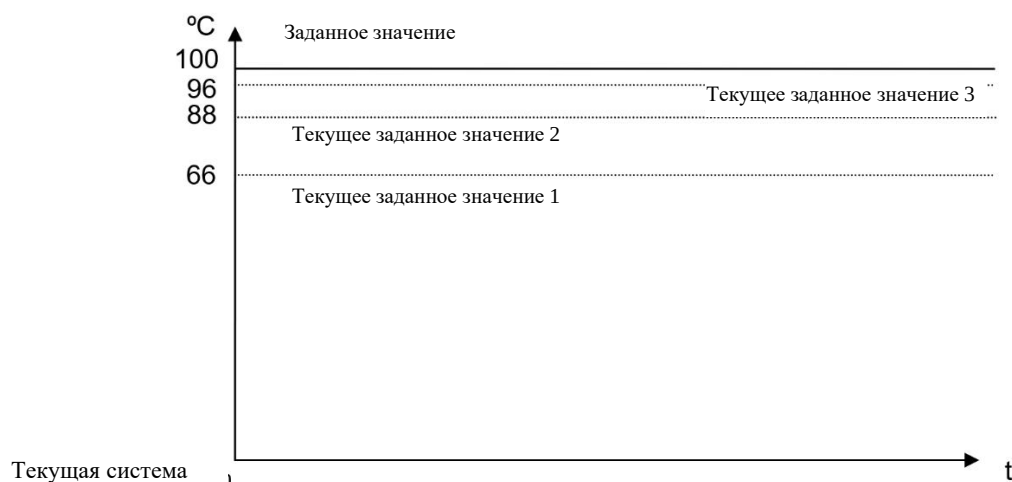
Расшифровка обозначений на схеме:

Процент корректировки текущего заданного значения составляет 2/3 (S3 + 14 = 67%), исходная температура системы составляет 0 °C, заданная температура составляет 100 °C. Корректировка текущей заданной температуры показана на схеме ниже:

Следующее текущее заданное значение = текущее заданное значение + (конечное заданное значение - текущее заданное значение) × 2/3.

Таким образом, последовательность изменения текущей заданной температуры следующая: 66 °C, 88 °C, 96 °C, 98 °C, 99 °C, 100 °C.





Кратность превышения заданного значения на этапе конечного перехода при автоматической регулировке во время ограничения превышения [S3 + 15]

Этот параметр действует при условии, что для [S3 + 13] установлено значение 1.

При ПИД-регулировании сразу после автоматической регулировки может возникнуть небольшое превышение. Чтобы предотвратить превышение, рекомендуется правильно увеличивать этот параметр. Но произойдет задержка ответа, если данное значение слишком большое. По умолчанию задано значение — в 15 раз. Рекомендуемый диапазон — от 5 до 20.

7.5 Расширенный режим

Пользователи могут задать некоторые параметры в расширенном режиме, чтобы получить оптимальный эффект ПИД-регулирования. Войдите в расширенный режим, установите бит 15 [S3 + 2] на 1 или настройте его в программном обеспечении PROMPOWER PLC Studio.

Константа входного фильтра [S3 + 8]

Она сглаживает значение выборки. Значение по умолчанию — 0%, что означает, что фильтр отсутствует.

Дифференциальное усиление [S3 + 9]

Процесс фильтрации нижних частот сглаживает резкое изменение выходного значения. Значение по умолчанию — 50%; эффект сглаживания будет еще очевиднее, если увеличить это значение. Пользователям необязательно менять данное значение.

Значение верхнего предела и нижнего предела [S3 + 10], [S3 + 11]

Пользователи могут выбрать диапазон аналогового выхода, установив это значение.

Значение по умолчанию: выход нижнего предела = 0

Верхний предел = 4095

7.6 Области применения

В условиях непрерывного вывода данных система, эффективность которой будет снижаться с изменением значения обратной связи, может выполнять автоматическую регулировку, например, температуры или давления. Она не подходит для регулировки уровня потока или жидкости.

Если превышение разрешено, система получит оптимальные параметры ПИД-регулятора в результате автоматической регулировки.

Если превышение запрещено, то параметры ПИД-регулятора, полученные в результате автоматической регулировки, будут соответствовать заданному значению. Это означает, что другое заданное значение приведет к другим параметрам ПИД-регулятора, которые не являются оптимальными параметрами системы и приведены только для справки.

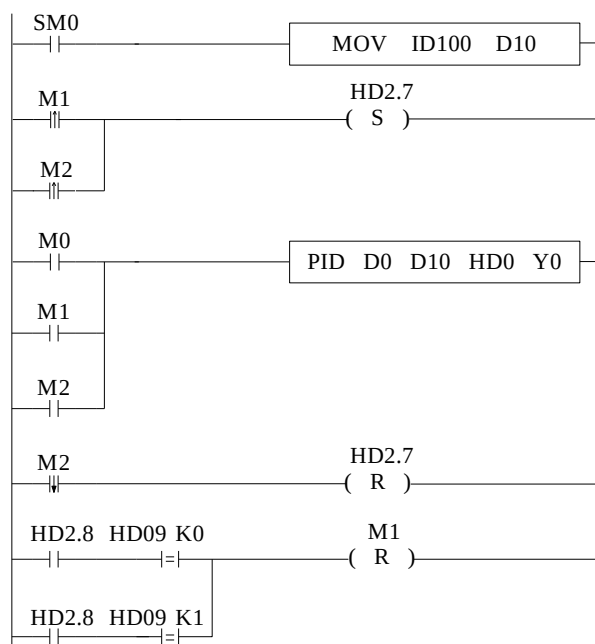
Если функция автоматической регулировки не поддерживается, пользователи могут задать параметры ПИД-регулятора, опираясь на свой практический опыт. Пользователям придется изменить параметры при отладке. Ниже для справки приведены некоторые экспериментальные значения систем регулирования:

- Система регулирования температуры: P (%) 2000 ~ 6000, I (минуты) 3 ~ 10, D (минуты) 0,5 ~ 3
- Система регулирования потока: P (%) 4000 ~ 10000, I (минуты) 0,1 ~ 1
- Система регулирования давления: P (%) 3000 ~ 7000, I (минуты) 0,4 ~ 3
- Система регулирования уровня жидкости: P (%) 2000 ~ 8000, I (минуты) 1 ~ 5

7.7 Применение

Пример 1:

Программа ПИД-регулирования представлена ниже:



// перемещение содержимого ID100 в D10
// режим автоматической регулировки, или переход в режим автоматической регулировки после завершения автоматической регулировки
// запуск ПИД-регулятора, D0 — заданное значение, D10 — измеренное значение, из HD0 — область параметров ПИД-регулирования; вывод результата ПИД-регулирования с помощью Y0
// завершение ПИД-регулирования, выход из режима автоматической регулировки ПИД-регулятора
// Если автоматическая регулировка выполнена успешно, и превышение разрешено, управляющий бит автоматической регулировки закроется, автоматическая регулировка завершится. Если автоматическая регулировка выполняется в ручном режиме, и превышение запрещено, закрыть управляющий бит автоматической регулировки.

Примечания по функциям программных элементов

HD2.7: Бит автоматической регулировки

HD2.8: Признак успешного выполнения автоматической регулировки

M0: Нормальный режим ПИД-регулирования

M1: Управление автоматической регулировкой

M2: Вход в режим ПИД-регулирования после автоматической регулировки

Последовательность действий:

1. Отправьте фактическое значение температуры в накопительный регистр ПИД-регулятора.
2. Задайте возможное значение для периодов выборки П, И, Д.
3. Включите управляющий бит автоматической регулировки M1, чтобы запустить автоматическую регулировку ПИД-регулятора.

4. M1 будет сброшен после завершения автоматической регулировки.
5. Включите программный элемент M0, используйте параметры ПИД-регулятора, полученные во время автоматической регулировки.
6. Если эффект ПИД-регулирования невысокий при использовании автоматической регулировки параметров ПИД-регулятора, можно вручную отрегулировать параметры ПИД-регулятора.

Примечание: данная программа ПИД-регулирования температуры ПЛК подходит практически для всех устройств регулирования температуры.

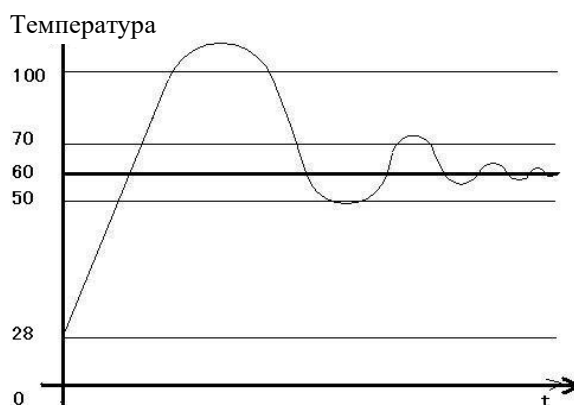
Пример 2:

Регулирование заданной температуры 60°C в режиме ступенчатого отклика.

Превышение разрешено:

1. Заданная температура составляет 60 °C (600).
2. Настройка параметров

3. Кривая результатов



Пояснение:

Заданная температура составляет 60 градусов, диапазон расчета ПИД-регулятора составляет 10 градусов, мертвая зона ПИД-регулятора составляет 0,2 градуса, значение изменения периода автоматической регулировки —10. Если ПИД-регулирование выполняется при нормальной температуре воздуха, выходная клемма ПИД-регулятора будет обеспечивать нагрев в диапазоне от 28 до 100 градусов, затем выходная клемма отключится, а температура продолжит расти до 110 градусов (максимальная температура). Затем температура будет падать до 60 градусов, выход снова обеспечит нагрев до 70 градусов, после чего отключится. Температура немного увеличится, а потом снова будет падать. Этот процесс повторится. В конечном итоге, температура будет колебаться вблизи заданной температуры.

Примечание:

1. Когда температура достигнет 100 градусов, и нагрев прекратится, стартовый бит D4002.7 ПИД-регулятора не сбросится сразу, а перед его сбросом будет некоторая задержка.
2. Когда температура достигнет 100 градусов, и нагрев прекратится, бит успешного выполнения автоматической регулировки D4002.8 ПИД-регулятора сразу включится.
3. При запуске функции расчета ПИД-регулятора ПЛК автоматически задает период выборки (около 2500). Этот параметр будет заменен на оптимальный период выборки ПИД-регулятора после прекращения нагрева при температуре 100 градусов.
4. При запуске функции расчета ПИД-регулятора ПЛК автоматически настроить параметры ПИД-регулятора (P (П) = 4454, I (И) = 926, D (Д) = 2317). Эти параметры будут заменены на оптимальное значение ПИД-регулятора после прекращения нагрева при температуре 100 градусов.
5. Когда температура достигнет 100 градусов, и нагрев прекратится, стартовый бит D4002.7 ПИД-регулятора не сбросится сразу, а перед его сбросом будет некоторая задержка. При этом температура выборки превышает заданную температуру. Если снова включить режим автоматической регулировки ПИД-регулятора, ПЛК получит все параметры ПИД-регулятора со значением 0. Включите ПИД-регулятор после того, как температура упадет до уровня нормальной температуры воздуха.

6. Если стартовый бит автоматической регулировки ПИД-регулятора и бит успешного выполнения автоматической регулировки сохраняются при отключении питания, настройте или сбросьте их, чтобы избежать ошибки расчета при следующем запуске ПЛК.
7. Конечная температура нагрева составит 110 градусов, если превышение разрешено. Если заданная температура превышена на 50 градусов, превышение будет слишком сильным.
8. После начала работы ПИД-регулятора выход нагревает объект с 28 до 60 градусов, затем выход принудительно прекращает нагрев, чтобы исключить превышение, но это прерывает процесс автоматической регулировки ПИД-регулятора.
9. Расширение диапазона расчета ПИД-регулятора может устранить перегрев.

Превышение запрещено:

1. Заданная температура составляет 60 °C (600).
2. Соответствующие настройки параметров:

PID Instruction Parameter Config

Target Value (SV): D4500 Measure Value (PV): D0 Parameter: D4000 Output: Y0

Parameter Config

☐ Manual ☒ Auto

Sampling Time: 100 ms

Proportion Gain (KP): 0 %

Integration Time (TI): 0 *100ms

Differential Time (TD): 0 *10ms

PID Computation Scope: 1000

PID Control Death Band: 20

Self Study Periodic Value: 10

Self Study Method: Step Response

Self Study PID Control Mode: PID Control

Mode Config

☒ Common Mode ☐ Advanced Mode

Input Filter Constant (a): 0 %

Differential Increase (KD): 50 %

Output Upper Limit Value: 4095

Output Lower Limit Value: 0

Direction Config

☒ Negative Movement ☐ Positive Movement

Negative Movement: Along with the increase of the measures definite value PV, outputvalue MV will also reduce. It's usually used in heat up control.

Positive Movement: Along with the increase of the measures definite value PV, outputvalue MV will also increase. It's usually used in cool control.

Overshoot Config

☐ Enable Overshoot ☒ Disable Overshoot

Each time adjust the increase: 100 %

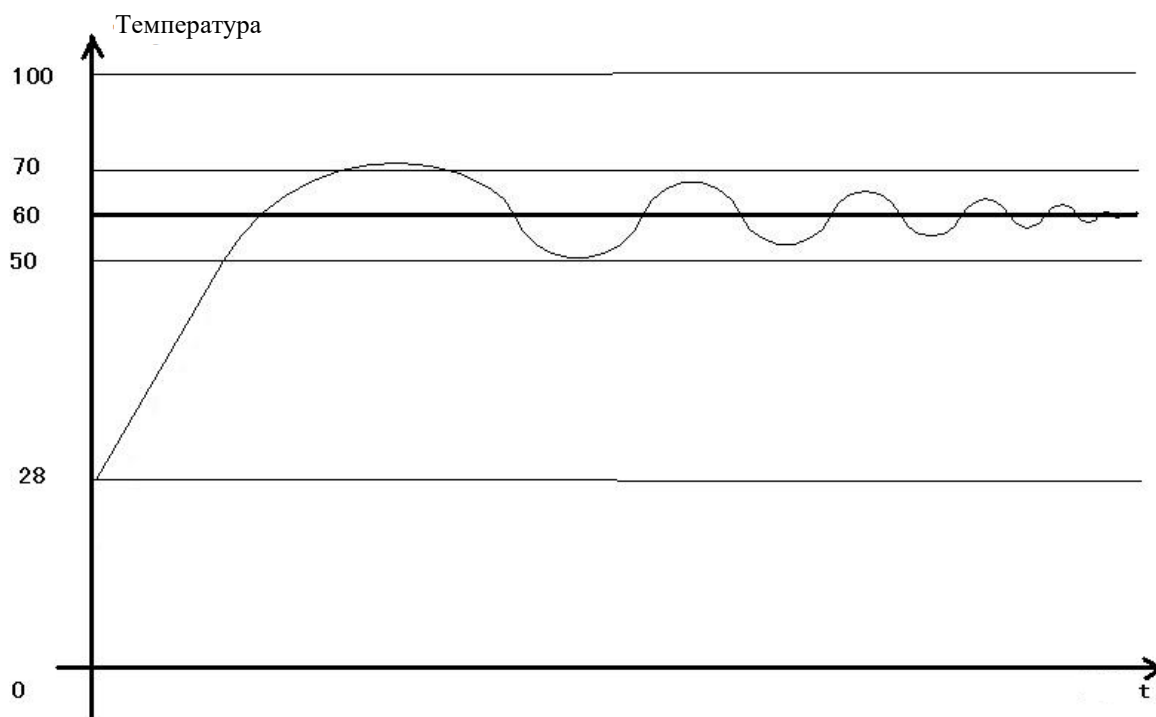
Current target value resident Count: 15

Suggestion value

Parameter Range: D4000 - D4069

Read From PLC Write To PLC OK Cancel

3. Кривая результатов



Пояснение:

Заданная температура составляет 60 градусов, диапазон расчета ПИД-регулятора составляет 10 градусов, мертвая зона ПИД-регулятора составляет 0,2 градуса, значение изменения периода автоматической регулировки —10. Если ПИД-регулирование выполняется при нормальной температуре воздуха, выходная клемма ПИД-регулятора будет обеспечивать нагрев в диапазоне от 28 до 48 градусов, затем выходная клемма отключится, а температура продолжит расти до 70 градусов (максимальная температура). Затем температура будет падать до 60 градусов, выход снова обеспечит нагрев до 62 градусов, после чего отключится. Температура немного увеличится (примерно до 64 градусов), а потом снова будет падать. Этот процесс повторится. В конечном итоге, температура будет колебаться вблизи заданной температуры. Погрешность составляет $\pm 0,25$ градуса.

Примечание:

1. Когда температура достигнет 48 градусов, и нагрев прекратится, стартовый бит D4002.7 ПИД-регулятора не сбросится сразу, а перед его сбросом будет некоторая задержка.
2. Когда температура достигнет 48 градусов, и нагрев прекратится, бит успешного выполнения автоматической регулировки D4002.8 ПИД-регулятора сразу выключится. Он не включится даже после успешной автоматической регулировки.

3. При запуске функции расчета ПИД-регулятора ПЛК автоматически задает период выборки (около 2500). Этот параметр будет заменен на оптимальный период выборки ПИД-регулятора после прекращения нагрева при температуре 48 градусов.
4. При запуске функции расчета ПИД-регулятора ПЛК автоматически настроить параметры ПИД-регулятора (P (П) = 4454, I (И) = 926, D (Д) = 2317). Эти параметры будут заменены на оптимальное значение ПИД-регулятора после прекращения нагрева при температуре 48 градусов.
5. Когда температура достигнет 48 градусов, и нагрев прекратится, стартовый бит D4002.7 ПИД-регулятора не сбросится сразу, а перед его сбросом будет некоторая задержка. При этом температура выборки превышает заданную температуру. Если снова включить режим автоматической регулировки ПИД-регулятора, ПЛК получит все параметры ПИД-регулятора со значением 0. Включите ПИД-регулятор после того, как температура упадет до уровня нормальной температуры воздуха.
6. Если стартовый бит автоматической регулировки ПИД-регулятора и бит успешного выполнения автоматической регулировки сохраняются при отключении питания, настройте или сбросьте их, чтобы избежать ошибки расчета при следующем запуске ПЛК.
7. Конечная температура нагрева составит 70 градусов, если превышение разрешено. Если заданная температура превышена на 10 градусов, превышение будет слишком сильным.
8. Расширение диапазона расчета ПИД-регулятора может устранить перегрев.

8. Функциональные блоки на языке C

В этой главе мы рассмотрим характеристики функциональных блоков на языке C, их редактирование, вызов соответствующих команд, области применения и т. д. Также представлен перечень общих функций.

8.1 Обзор

ПЛК серии RMP20 поддерживают написание функциональных блоков на языке C в программном обеспечении ПЛК PROMPOWER и их вызов при необходимости. ПЛК этой серии поддерживают практически все функции языка C, а также глобальные переменные, что повышает конфиденциальность программы. При этом ПЛК может вызывать множество мест и различных файлов, что значительно повышает эффективность работы программистов.

8.2 Формат команды

1) Обзор

Вызов функционального блока на языке C в указанном месте.

Вызов функционального блока на языке C [NAME_C]			
16-битная команда	NAME_C	32-битная команда	-
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме, запуск по нарастающему фронту/спадающему фронту	Совместимые модели	RMP20
Аппаратное обеспечение		Программное обеспечение	

2) Операнды

Операнды	Функция	Тип
S1	Название функционального блока на языке C, заданное пользователем	Строка
S2	Соответствующий начальный идентификатор слова W в функции языка C	16 бит, BIN

S3	Соответствующий начальный идентификатор бита В в функции языка C	бит, BIN
----	---	----------

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден- тифи- катор	QD	X	Y	M	S	T	C	Dn.m
S1																		
S2	•																	
S3														•				

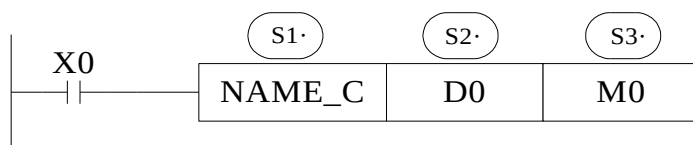
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



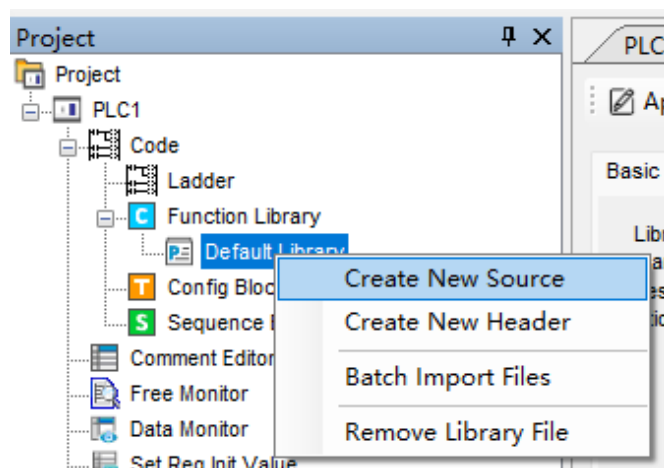
S1 — имя функции. Оно состоит из цифр, букв и подчеркиваний. Первый символ не может быть числом, а длина имени должна быть <=9 символов в формате ASCII.

Имя может совпадать с собственными командами ПЛК, например, LD, ADD, SUB, PLSR и т. д. Имя не может совпадать с именами функциональных блоков, существующих в текущем ПЛК.

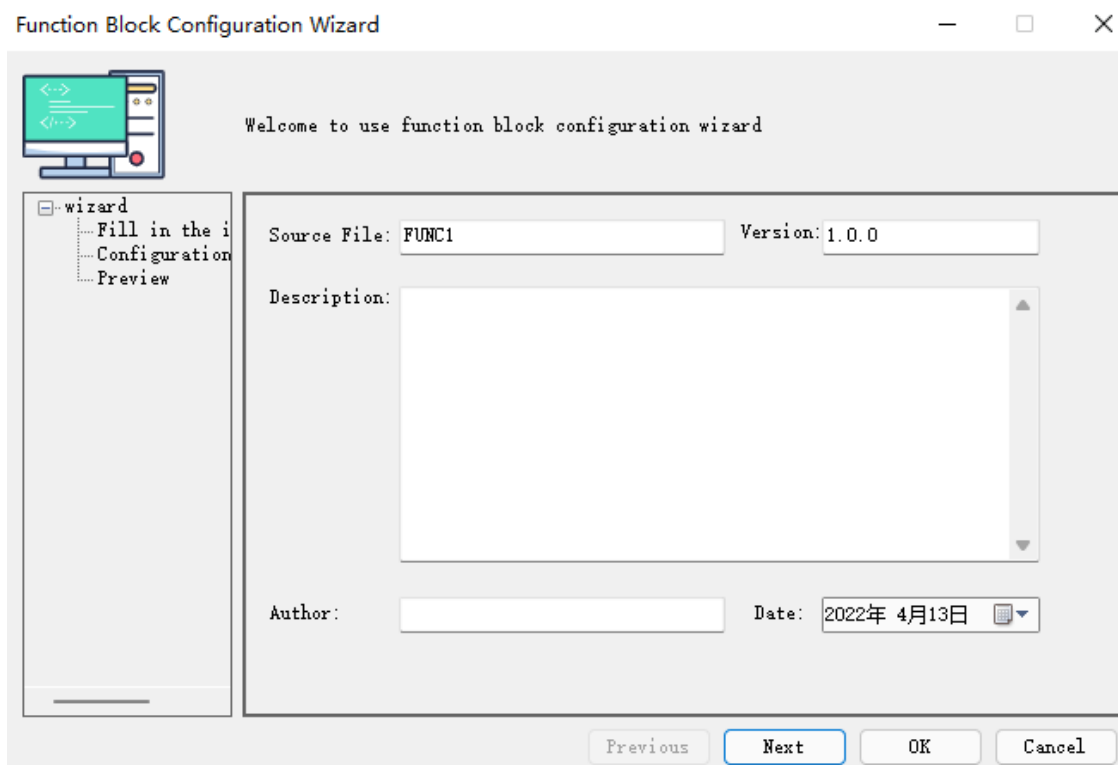


8.3 Последовательность действий

1. Откройте инструмент редактирования ПЛК, на панели инструментов “Project” (Проект) слева, выберите “Func Block” (Функциональный блок), щелкните по нему правой кнопкой мыши и выберите “Add New Func Block” (Добавить новый функциональный блок).



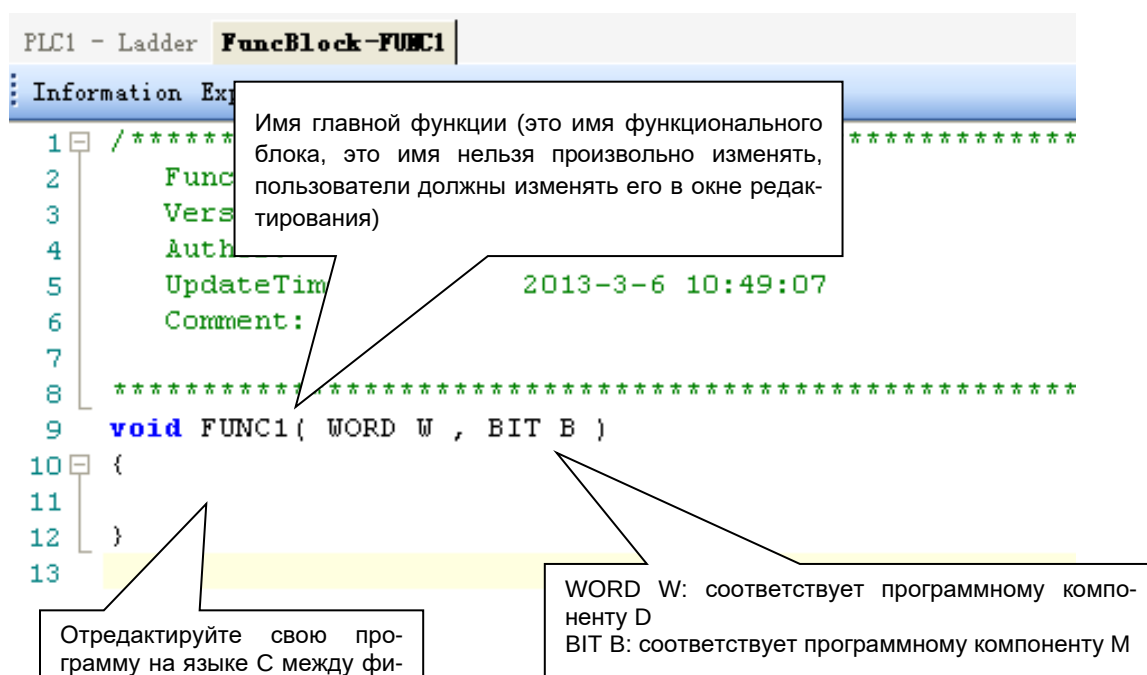
2. В показанном ниже окне введите информацию по вашей функции.



Имя функционального блока — это имя, которое мы используем для вызова функционального блока BLOCK. Например, ниже представлена схема FUNC1.



3. После создания нового функционального блока станет доступным показанное ниже окно редактирования.



- Способ **функциональный блок** на лестничной диаграмме, то переданные компоненты D (HD) и M (HM) являются начальными идентификаторами W и B. Возьмем в качестве примера приведенный выше график, начнем с D0 и M0, тогда W[0] будет D0, W[10] будет D10, B[0] будет M0, B[10] будет M10. Если параметрами на лестничной диаграмме являются HD0, HM0, то W[0] = HD0, B[0] = HM0. Если параметрами на лестничной диаграмме являются D100, HM100, то W[0] = D100, B[0] = HM100. Таким образом, начальные адреса компонентов Word и Bit задаются в программе ПЛК пользователем.

Примечание: локальная переменная, заданная внутри функции C, не может содержать более 100 слов.

- Параметр W: является программным компонентом Word, используется в виде группы данных. Например, W[0] = 1; W[1] = W[2] + W[3]. В программе используйте программные компоненты в соответствии со стандартными правилами языка C.
- Параметр B: является программным компонентом Bit, используется в виде группы данных. Поддержка SET и RESET. Пример: B[0] = 1; B[1] = 0; и присвоение, например, B[0] = B[1].
- Операция с двойным словом: добавьте D перед W. Например, DW[10] = 100000, это указывает на присвоение двойному слову W[10]W[11]. Операция с двойным словом: Поддержка задания переменной с плавающей точкой в

функции и выполнения операций с плавающей точкой. (Пример: регистр с плавающей точкой D0 (двойное слово) означает FW[0], FW[0] = 123,456)

- Задание других программных элементов на языке C:

После создания функционального блока в главной функции по умолчанию задается `#define SysRegAddr_HD_D_HM_Mi`. Если нужно использовать вход (X) и выход (Y), следует добавить X, Y в макроопределение по умолчанию `"#define SysRegAddrHD_D_HM_M"`, что будет иметь вид `"#define SysRegAddrHD_D_HM_M_X_Y"`. Например, установите состояние X0 для катушки M0, B[0] = X[0]; установите состояние Y0 для катушки M10, B[10] = Y[0]. (Примечание: Соответствующие X и Y выражены в десятичной, а не в восьмеричной системе счисления на языке C).

Аналогичным образом, программы на языке C одинаковы для памяти без отключения питания S, счетчика C, таймера T, регистра счетчика CD, регистра таймера TD, регистра D (HD) и катушки M (HM) и т. д. Макроопределение — `"#define SysRegAddr_S_C_T_CD_TD_D_M"`. Для памяти с отключением питания HS, счетчика HC, таймера HT, регистра счетчика HCD, регистра таймера HTD и т. д. макроопределение будет иметь вид `"#define SysRegAddr_HS_HC_HT_HCD_HTD"`.

Примеры: W[0] = CD[0]; W[1] = TD[0]; B[1] = C[0]; B[2] = T[0].

Примечание: поддерживаются все типы программных компонентов, кроме SEM.

- После создания функционального блока в главной функции по умолчанию задается `#define SysRegAddr_HD_D_HM_M`.

```
7
8  *****
9  void FUNC1( WORD W , BIT B )
10 {
11     #define SysRegAddr_HD_D_HM_M
12
13
14 }
15
```

Рекомендуется использовать его как локальное макроопределение, то есть внутри тела функции.

- Библиотека функций: Пользовательский функциональный блок может напрямую использовать функции и константы, заданные в библиотеке функций.

Функции и константы, содержащиеся в библиотеке функций, представлены в главе 8-10.

- Другие поддерживаемые типы данных:

BOOL	//логическая величина BOOL
INT8U	//8-битное целое число без знака
INT8S	//8-битное целое число со знаком
INT16U	//16-битное целое число без знака
INT16S	//16-битное целое число со знаком
INT32U	//32-битное целое число без знака
INT32S	//32-битное целое число со знаком
FP32	//число с плавающей точкой одинарной точности
FP64	//число с плавающей точкой двойной точности

Примеры:

```
#define DHD*(INT32S*)&HD // DHD обозначает двойное слово HD
```

```
#define FFW*(FP64*)&D // FFW обозначает числа с плавающей точкой двойной точности
```

```
#define DDW*(long long*)&D // DDW обозначает регистр из четырех слов
```

Пояснение:

DHD — 32-битное целое число со знаком DHD[0] представляет собой 32-битный знаковый целочисленный регистр временного хранения данных после выключения питания, состоящий из HD0 и HD1.

Заданные макросы:

```
#define true 1
```

```
#define false 0
```

```
#define TRUE 1
```

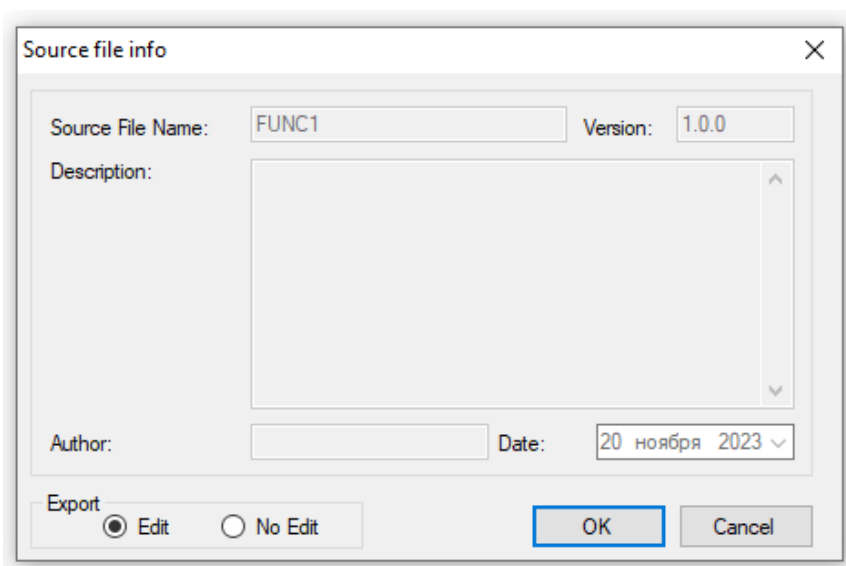
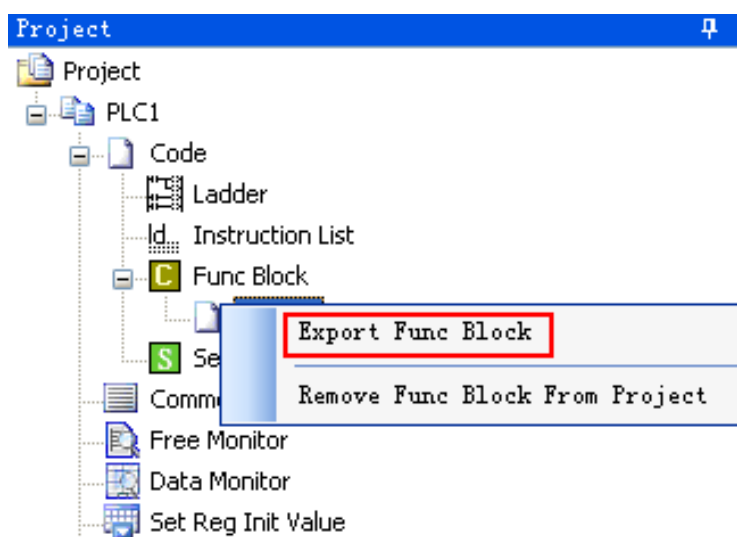
```
#define FALSE 0
```

- Для экспорта файлов заголовков нет редактируемой опции, остальные опции аналогичны опциям для исходных файлов.
- В языке C существуют два правила для создания ссылок на файлы заголовков: `#include "xx.h"` и `#include <xxx.h>`. При использовании файла заголовка в проекте для ПЛК в исходном файле следует использовать `#include "xxx.h"`.
- Не используйте макроопределение `#define SysRegAddr` в файле заголовка; это макроопределение неэффективно в файле заголовка, его можно использовать только в исходном файле.

8.4 Импорт и экспорт функций

1. Экспорт

- (1) Функция: Экспортируйте функцию как файл, после чего другую программу ПЛК можно импортировать для использования.

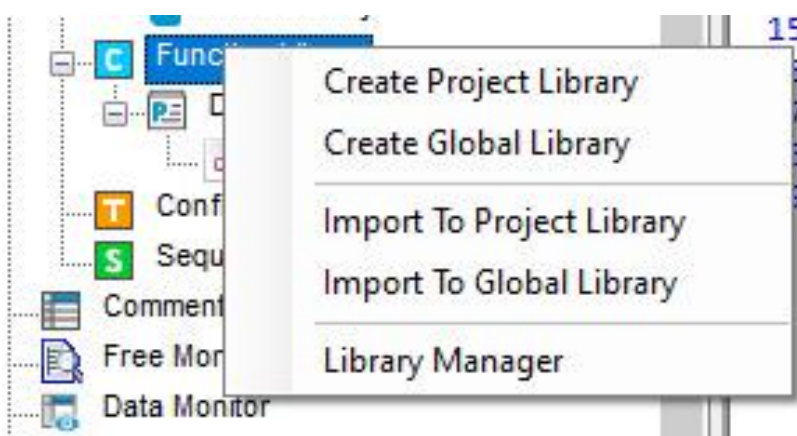


(2) Формат экспорта

- a) Edit (Редактирование): Экспорт исходных кодов и их сохранение в виде файла. При повторном импорте файл можно редактировать.
- b) No edit (Без редактирования): Экспорт исходных кодов не выполняется. После импорта файла его нельзя будет редактировать. Модели с поддержкой Ethernet и модели без поддержки Ethernet нельзя вместе использовать. Перед экспортом модели достаточно будет ее изменить.

2. Импорт

Функция: Импорт существующего файла **Func Block** для использования в программе ПЛК.



Выберите **Func Block**, щелкните правой кнопкой мыши по 'Import To Project Library' (Импорт в библиотеку проекта), выберите нужный файл и нажмите OK.

8.5 Редактирование функциональных блоков

Пример:

Добавьте D0 и D1 в регистры ПЛК, после чего присвойте значение элементу D2.

- (1) На панели инструментов 'Project' (Проект) создайте функциональный блок. Назовем его ADD_2, а затем отредактируем программу на языке C.
- (2) После редактирования отметьте галочкой опцию 'compile' (Компилирование).

```
PLC1 - Ladder FuncBlock-ADD_2
┌ Information Export Compile
7 |           W [2] =W [0] +W [1]
8 | *****
9 | void ADD_2( WORD W , BIT B )
10| { W [2] =W [0] +W [1]
11|
12| }
13|
└ Information(1)
  Error List Output
1. ...\\.\tmp\PrjFuncB\ADD_2.c: In function 'ADD_2':
  ...\\.\tmp\PrjFuncB\ADD_2.c:6:1: error: expected ';' before 'asm'
```

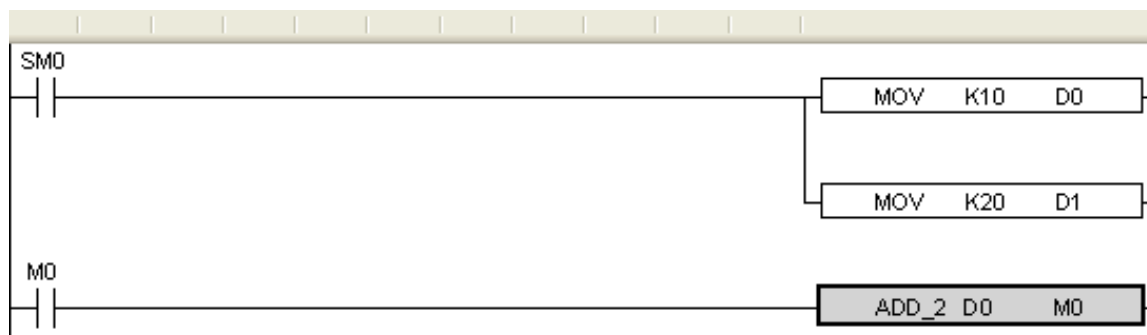
Перечень информации

Используя информацию, отображаемую в форме вывода, можно найти и исправить грамматическую ошибку в программе на языке C. Здесь мы видим, что в программе нет знака ';' после `W[2] = W[0] + W[1]`.

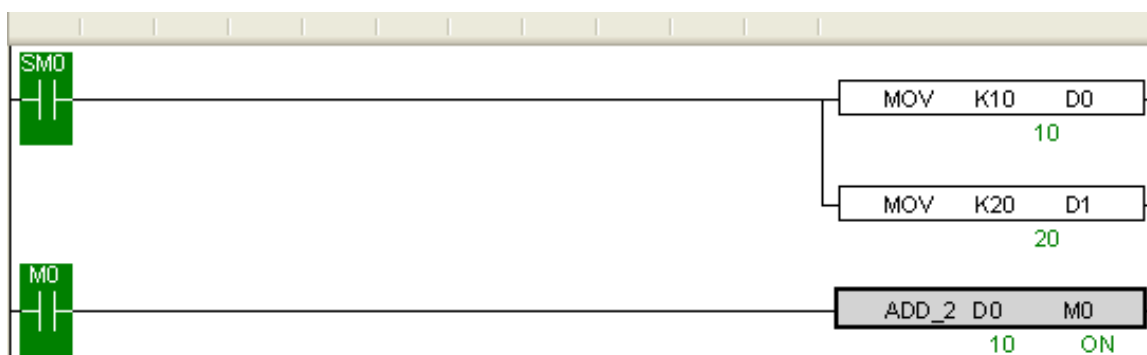
Скомпилируйте программу еще раз после внесения в нее изменений. В перечне информации можно подтвердить, что в программе нет грамматических ошибок.

```
┌ Information Export Compile
6 | Comment:
7 |           W [2] =W [0] +W [1]
8 | *****
9 | void ADD_1( WORD W , BIT B )
10| { W [2] =W [0] +W [1];
11|
12| }
└ Information
  Error List Output
```

- (3) Напишите программу ПЛК, присвойте значения 10 и 20 регистрам D0, D1 по отдельности, затем вызовите функциональный блок ADD_2. См. рисунок ниже.



- (4) Загрузите программу в ПЛК, запустите ПЛК и настройте M0.



- (5) С помощью опции Free Monitor (Свободный контроль) на панели инструментов можно проконтролировать, что D2 меняется на 30, это означает, что присвоение выполнено успешно.



Свободный контроль

PLC1-FreeMOnitor1				
Moni-Window ▾ Add Edit Delete Delete All Upword Downword Top Bottom				
Name	Value	Type	Map-Address...	Comment
D2	30	INT	SWord	

8.6 Пример программы

Если ПЛК должен проводить сложные вычисления (включая вычисления с плюсом и минусом), то результаты вычислений будут использоваться много раз. Функция языка С проста в использовании.

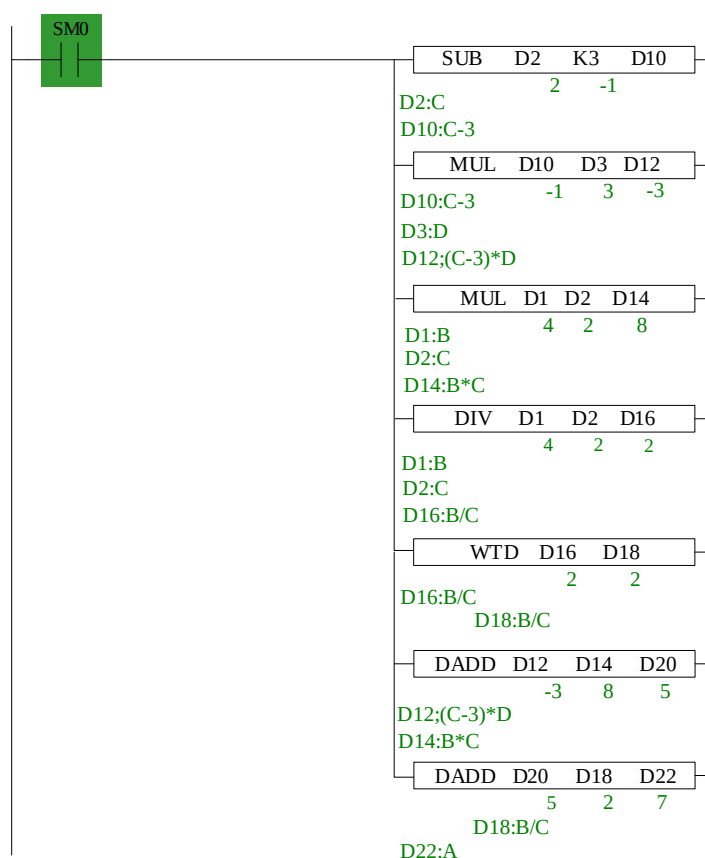
Пример 1:

Вычисление $a = b/c + b \cdot c + (c - 3) \cdot d$

Метод 1: использование лестничной диаграммы

- Получение результата $c - 3$
- Получение результата трех уравнений умножения
- Получение суммы

Лестничная диаграмма поддерживает только два исходных операнда, а для получения результата требуется много шагов.

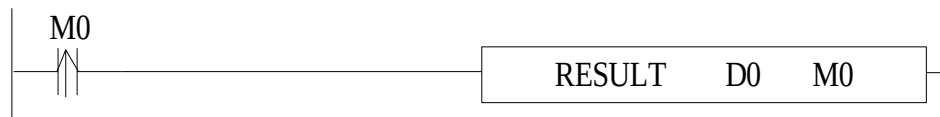


Примечание:

1. Результатом MUL является Dword, результат сохраняется в D14~D15.

2. Результатом DIV является частное D16 и остаток D17. Если D17 имеет значение, точность расчетов снизится. Для повышения точности используйте формат с плавающей точкой.
3. Частное D16 — это значение слова, а при положительном расчете все данные следует изменить на Dword. Конечный результат хранится в D22~D23.

Метод 2: использование языка C



Результат	Имя функции
D0	В функции W [0] = D0, W [1] = D1... Если D0 = D32, то W [0] = D32, W [1] = D33... Если S2 = HD32, то W [0] = HD32, W [1] = HD33...
M0	В функции B [0] = M0, B [1] = M1... Если S2 = M32, то B [0] = M32, B [1] = M33... Если S2 = HM32, то B [0] = HM32, B [1] = HM33...

Программа на языке C:

```

9  void RESULT( WORD W , BIT B )
10 {
11  long int a,b,c,d;;
12  b=W[1];
13  c=W[2];
14  d=W[3];
15  a=b/c+b*c+(c-3)*d;
16  DW[4]=a;
17  }
  
```

Метод 2 позволяет упростить программу.

Приведенная выше функция языка C похожа на лестничную диаграмму метода 1, точность которого невысокая. Если требуется высокая точность вычислений, используйте вычисления с плавающей точкой.

Пример 2:

Вычислите четность CRC через функциональный блок.

Правила вычисления CRC:

- (1) Настройте 16-битный регистр (регистр CRC) = FFFF H

- (2) Примените логическую операцию XOR (исключающее ИЛИ) для первого 8-битного байтового сообщения и регистра CRC с 16 младшими битами.
- (3) Выполните сдвиг вправо на 1 бит регистра CRC, заполните нулями старший бит.
- (4) Проверьте сдвинутое вправо значение. Если оно равно 0, сохраните новое значение, полученное в шаге 3, в регистре CRC. Если оно не равно 0, выполните операцию XOR для значения регистра CRC с A001 H, а затем сохраните результат в регистре CRC.
- (5) Повторяйте шаги 3 и 4, пока не будут вычислены все 8 бит.
- (6) Повторяйте шаги (2)-(5), затем вычислите следующее 8-битное сообщение. Пока все сообщения не будут вычислены, результатом будет код четности CRC в регистре CRC.

Отредактируйте программу функционального блока на языке С. См. рисунок ниже:

```

9  void CRC_CHECK( WORD W , BIT B )
10 {
11     int i,j,m,n;
12     unsigned int reg_crc=0xffff,k;
13
14     for( i = 0 ; i < W[0] ; i++ )
15     {
16         reg_crc^=W[i+1];
17         for(j=0;j<8;j++)
18         {
19             if (reg_crc&0x01)
20                 reg_crc=(reg_crc>>1)^0xa001;
21             else
22                 reg_crc=reg_crc>>1;
23         }
24     }
25
26     m=W[0]+1;
27     n=W[0]+2;
28     k=reg_crc&0xff00;
29     W[n] = k>>8;
30     W[m]=reg_crc&0xff;
31 }

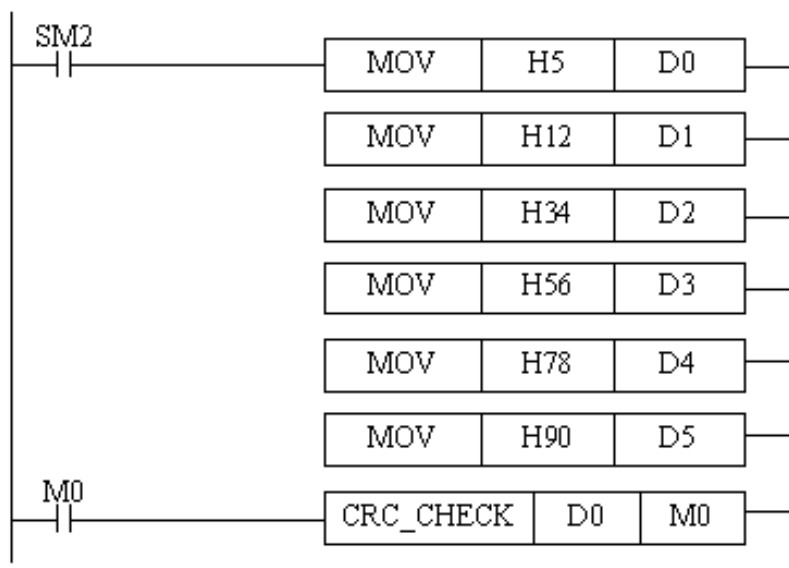
```

Отредактируйте релейно-контактную программу ПЛК.

D0: Проверьте количество байтов данных.

D1~D5: Проверьте содержимое данных.

См. схему ниже:

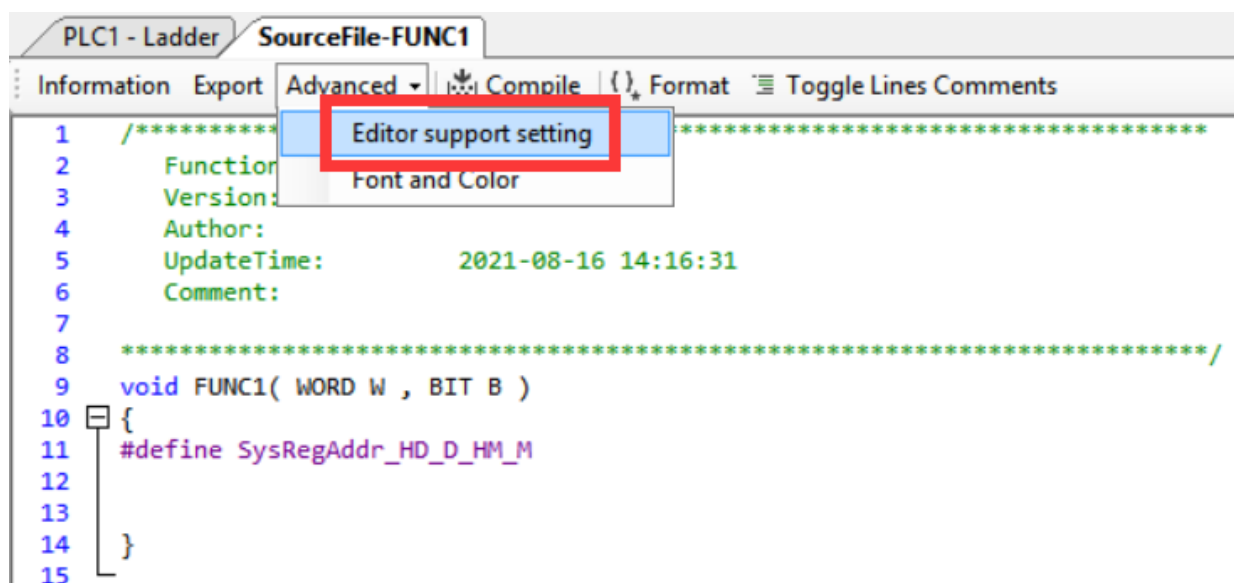


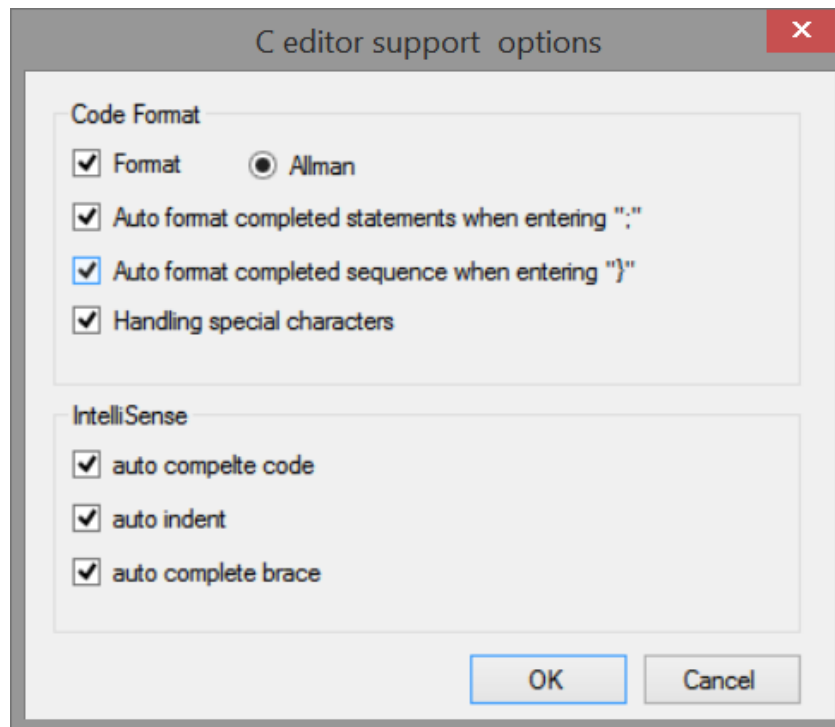
Загрузите программу в ПЛК, затем запустите (RUN) ПЛК, настройте M0. Через опцию Free Monitor (Свободный контроль) можно определить, что значения в D6 и D7 являются самым старшим и самым младшим битом четности CRC.

8.7 Новые функции

(1) Формат

Перейдите в меню Advanced/editor support setting (Дополнительно/Дополнительные настройки редактора), чтобы открыть окно C editor support options (Дополнительные опции редактора языка C).





(2) Автоматическое форматирование локального кода

- Auto format completed statements when entering ";" (Автоматическое форматирование завершенных операторов при вводе ";")

Когда пользователь введет символ ";", оператор текущей строки будет отформатирован.

- Auto format completed sequence when entering "}" (Автоматическое форматирование завершенных операторов при вводе "}")

Когда пользователь введет символ "}", содержимое в "}" будет отформатировано.

(3) Работа со специальными символами

Символы полной ширины, введенные пользователем в редактор, необходимо преобразовать в символы половинной ширины, поскольку они не распознаются компилятором.

(4) Auto complete code (Автоматическое завершение кода)

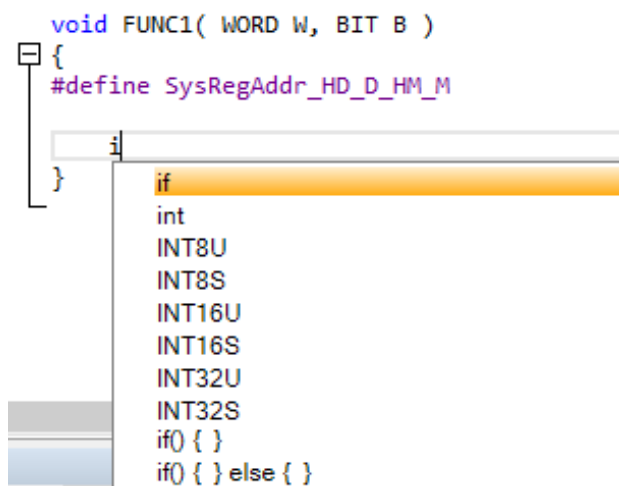
Когда пользователь вводит символы, функция подсказки кода выводит определенные подсказки, помогающие пользователю вводить и завершать код.

- Submit (Отправка)

Когда пользователь нажмет Enter или ";", текущий отредактированный код будет отправлен в анализатор на анализ, и будет сгенерирован список подсказок по коду.

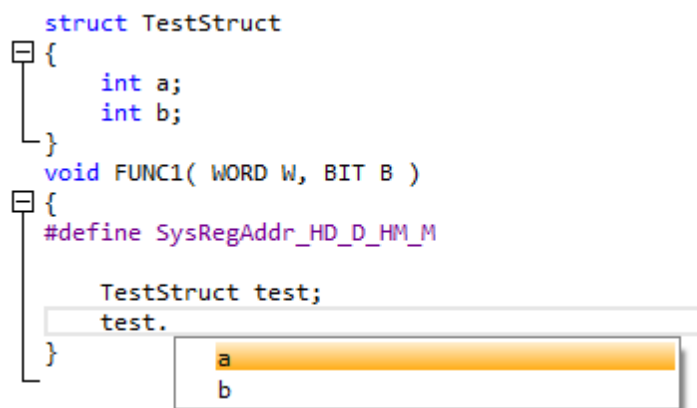
➤ Prompt (Подсказка)

Когда пользователь вводит символы, автоматически всплывает подсказка по коду.



➤ Access tips for member variables (Советы по доступу к переменным членам)

Когда пользователь вводит "." "or" - ">", функция подсказки кода подскажет пользователю члены структуры или тип группы элементов определенной переменной, как показано на рисунке ниже.



➤ Auto indent (Автоматический отступ)

Оптимизирована функция автоматического отступа редактора, которая теперь в большей степени соответствует пожеланиям конкретных пользователей.

➤ Auto complete brace (Автоматическое заключение в скобки после завершения)

Когда пользователь вводит "(" [{"", это автоматически поможет ему сгенерировать соответствующую скобку ")" "]" "}".

(5) Comment/uncomment (Выбор комментария/отмена выбора комментария)

Выбор/отмена выбора комментария в строке.

Горячая комбинация клавиш — Ctrl +/.

(6) Библиотека функций

См. главу 8-8.

8.8 Библиотека функций

Библиотека содержит функции шифрования, инкапсуляции, экспорта и импорта функциональных блоков на языке С.

8.8.1 Новые функции

8.8.1.1 Классификация библиотек

Библиотека функций делится на библиотеку проекта и глобальную библиотеку.

Библиотека проекта: Функции в библиотеке проекта пользователя сохраняются в проекте и могут использоваться напрямую.

Глобальная библиотека: Функции в глобальной библиотеке пользователя сохраняются в локальном каталоге для удобства пользователя.

8.8.2 Основные функции

8.8.2.1 Открытие и сохранение файла

Запустите программное обеспечение PROMPOWER PLC Studio, запустите пустой проект или откройте любой существующий проект, чтобы просмотреть библиотеку функций.

Примечания:

Библиотека функций делится на библиотеку проекта и глобальную библиотеку. Библиотека по умолчанию (т.е. библиотека проекта) добавляется в пустой проект по умолчанию.

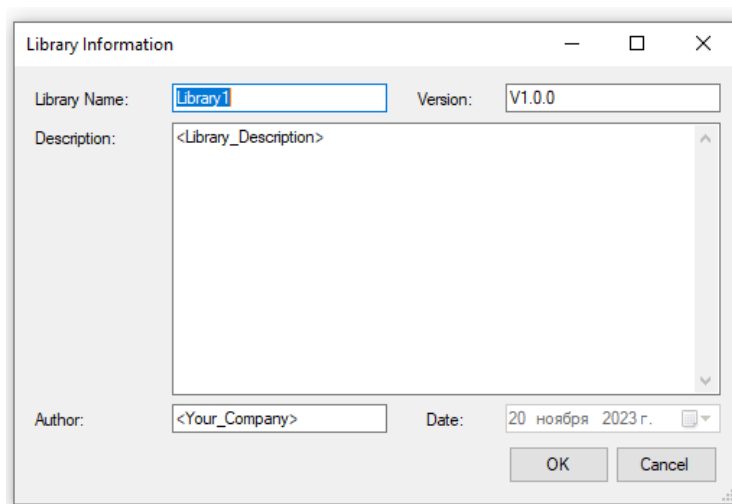
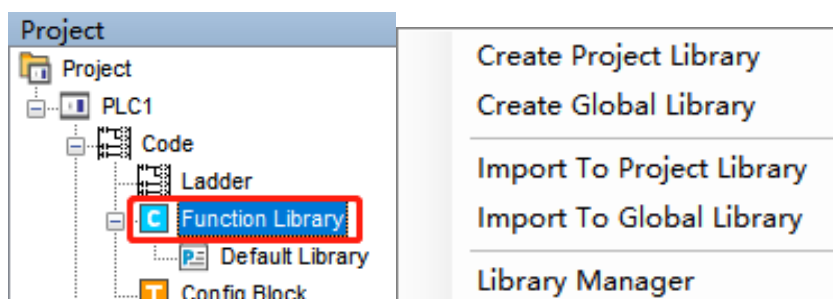
Если открыть проект старой версии в новой версии программы, его функции добавятся в библиотеку по умолчанию.

Если проект новой версии открыть в старой версии программы, функции в библиотеке по умолчанию сохранятся, а остальные функции нельзя будет проанализировать.

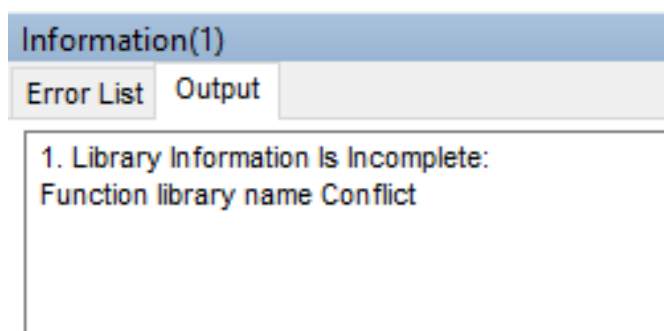
8.8.3 Новая версия

8.8.3.1 Создание библиотеки проекта

Выберите "Function library" (Библиотека функций) на панели инструментов "Project" (Проект) в левой части окна, щелкните правой кнопкой мыши и выберите "Create Project Library" (Создать библиотеку проекта). После этого можно будет отредактировать имя, версию, описание, автора и другую информацию о библиотеке проекта во всплывающем окне, показанном на рисунке ниже:

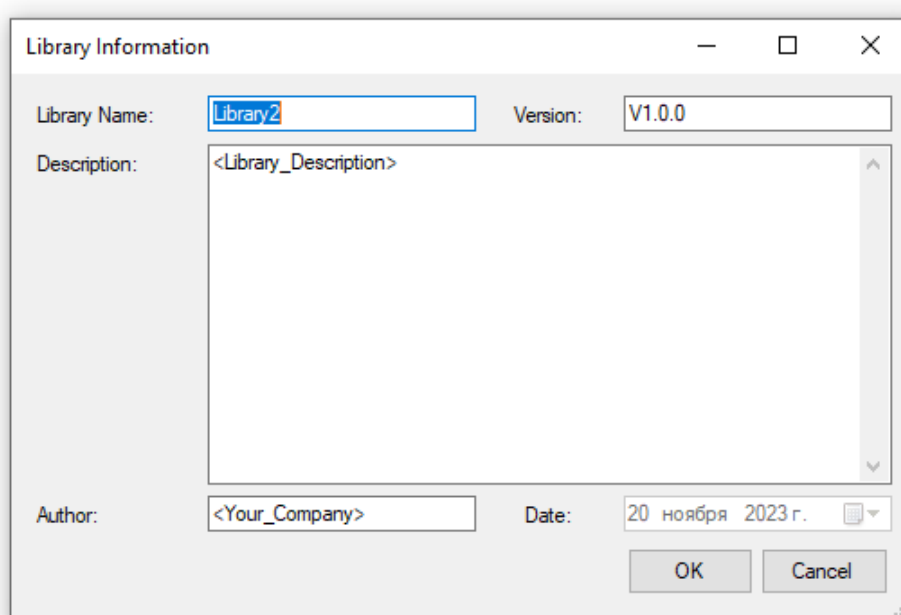
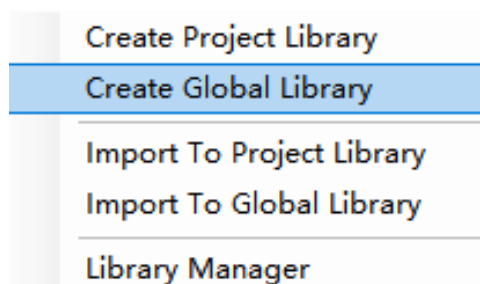


Примечание: если имя библиотеки совпадает с именем любой библиотеки в текущей библиотеке, в информационном окне появится следующее сообщение:

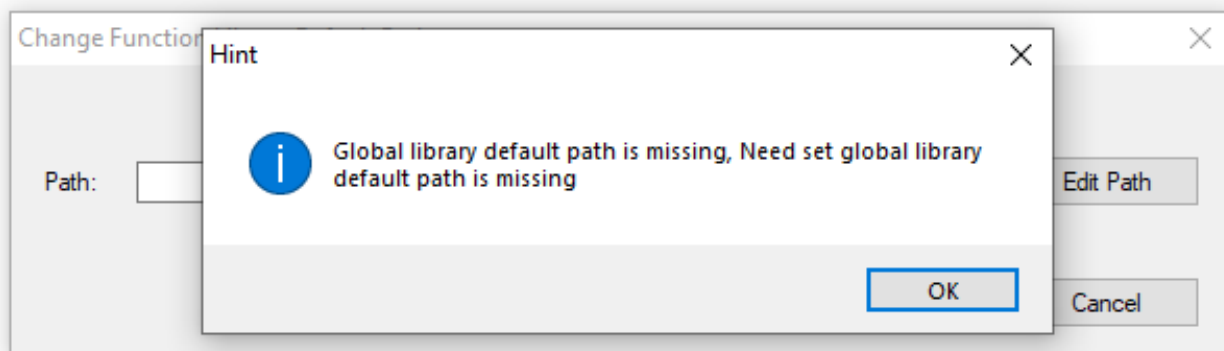


8.8.3.2 Создание глобальной библиотеки

Выберите "Function library" (Библиотека функций) на панели инструментов "Project" (Проект) в левой части окна, щелкните правой кнопкой мыши и выберите "Create Global Library" (Создать глобальную библиотеку). После этого можно будет отредактировать имя, версию, описание, автора и другую информацию о глобальной библиотеке во всплывающем окне, показанном на рисунке ниже:



Примечание: если каталог глобальной библиотеки не установлен, появится подсказка, показанная на рисунке ниже, и откроется окно настройки каталога глобальной библиотеки:

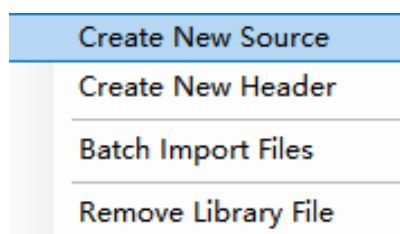


После выбора пути откроется окно нового файла библиотеки, и будет заполнена информация о библиотеке (название, версия, описание, автор). Если имя библиотеки совпадает с любым именем библиотеки в информационном окне, то появится следующее сообщение:

2. Library Information Is Incomplete:
Function library name Conflict

8.8.3.3 Создание нового источника

На панели инструментов "Project" (Проект) в левой части окна выберите библиотеку проекта или глобальную библиотеку, в которую необходимо добавить исходный файл в "Function library" (Библиотека функций), щелкните правой кнопкой мыши и выберите "New source file" (Новый исходный файл), чтобы отредактировать имя, версию, описание, автора и другую информацию исходного файла во всплывающем окне, показанном на рисунке ниже:



Function Block Configuration Wizard

Welcome to use function block configuration wizard

Source File: FUNC1 Version: 1.0.0

Description:

Author: Date: 2022年 7月 4日

Previous Next OK Cancel

Нажмите "Next" (Далее) после заполнения данных, чтобы настроить параметры:

Function Block Configuration Wizard

Welcome to use function block configuration wizard

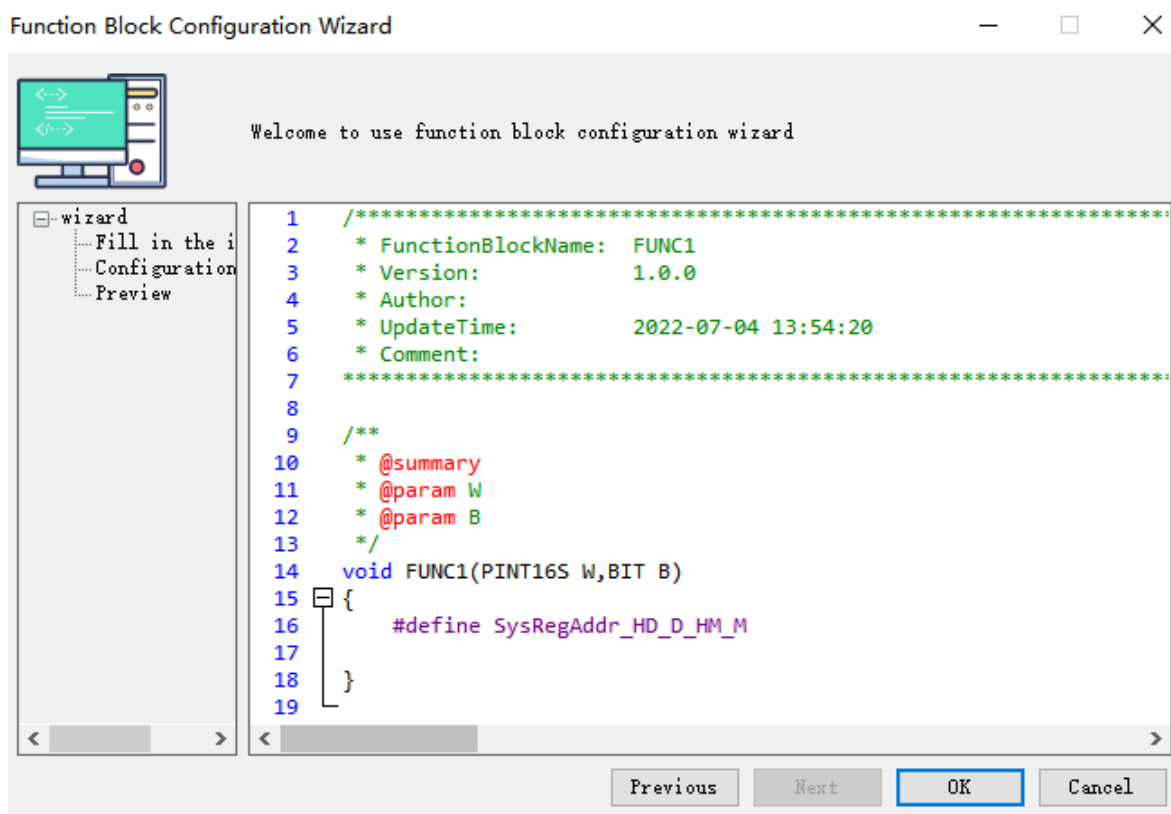
Parameter Name Parameter Type Parameter Mode Parameter Occupied Count Parameter Comment Support Devices List

Parameter Name	Parameter Type	Parameter Mode	Parameter Occupied Count	Parameter Comment	Support Devices List
W	PINT16S	In, Out	unknown		D HD
B	BIT	In, Out	unknown		M HM

Previous Next OK Cancel

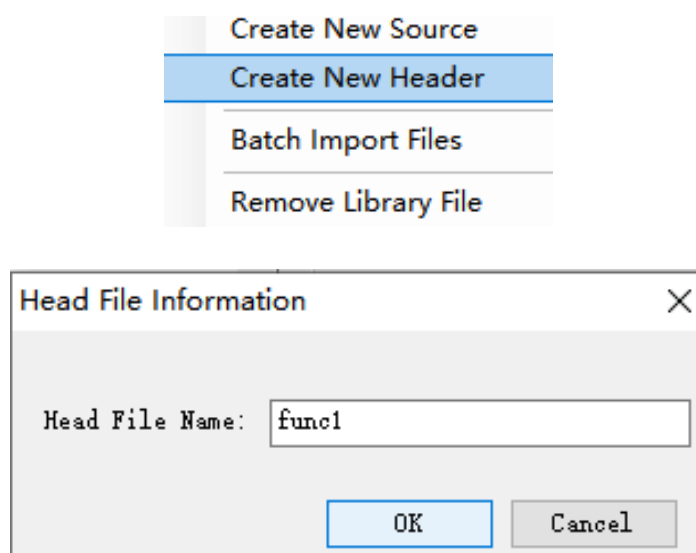
После настройки параметров нажмите "Next" (Далее), чтобы отобразить окно предварительного просмотра исходного файла. В случае проблем нажмите

"previous" (Назад), чтобы сбросить параметры. Если процедура прошла успешно, нажмите "OK", чтобы закончить добавление исходного файла.



8.8.3.4 Создание нового заголовка

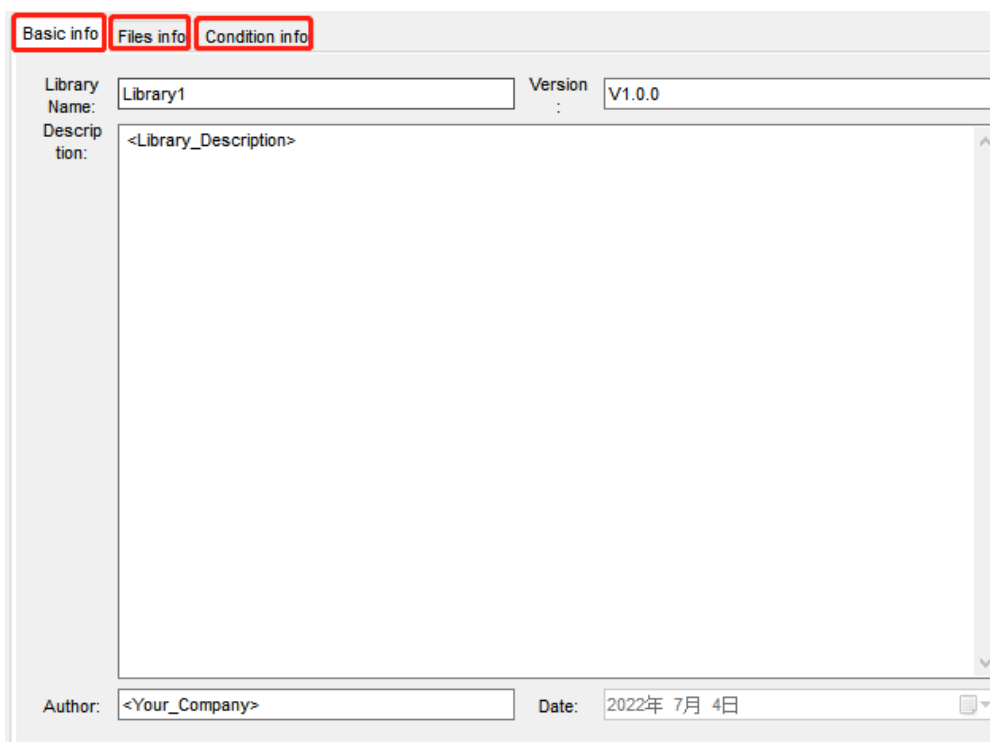
На панели инструментов "Project" (Проект) в левой части окна выберите библиотеку проекта или глобальную библиотеку, в которую необходимо добавить исходный файл в "Function library" (Библиотека функций), щелкните правой кнопкой мыши и выберите "New header file" (Новый файл заголовка), чтобы отредактировать имя файла заголовка во всплывающем окне, показанном на рисунке ниже:



8.8.4 Редактирование

8.8.4.1 Редактирование информации в библиотеке

Нажмите "Project library" (Библиотека проекта) или "Global library" (Глобальная библиотека) на панели проекта в левой части окна, чтобы отредактировать информацию. Во всплывающем окне с информацией о библиотеке можно будет просматривать и редактировать основную информацию/информацию о файлах/информацию об ограничениях, установленных для данной библиотеки:



Basic info Files info Condition info

Library Name: Library1 Version: V1.0.0

Description: <Library_Description>

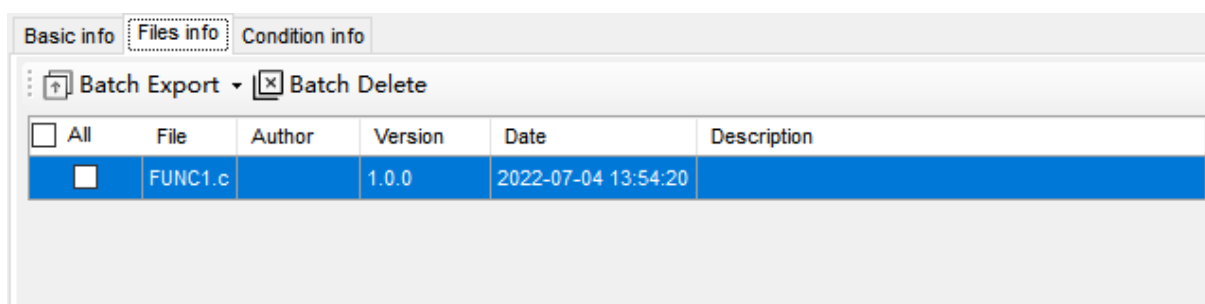
Author: <Your_Company> Date: 2022年 7月 4日

1) Basic information (Основная информация)

Library name (Имя библиотеки): Имя библиотеки может состоять только из букв и цифр.

Version (Версия): Формат версии библиотеки — "V primary. Secondary. Revision".

2) Files information (Информация о файлах)

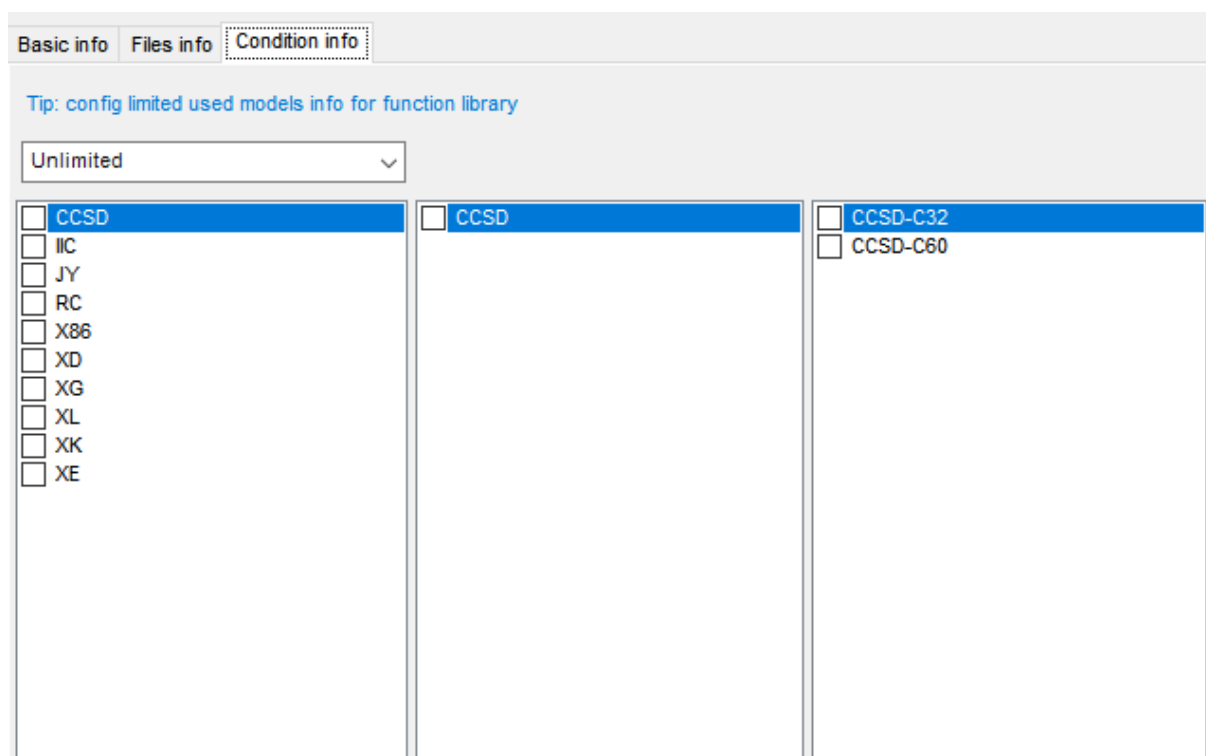


Batch Export ▾ Batch Delete					
☐ All	File	Author	Version	Date	Description
☐	FUNC1.c		1.0.0	2022-07-04 13:54:20	

- Добавьте исходный файл/файл заголовка в выбранную библиотеку функций, после чего в окне с информацией о файле появится основная информация о файле.
- В настройках импортированного файла указано, может ли пользователь его редактировать.
- Файлы, экспортированные в пакетном режиме, можно редактировать либо не редактировать.
- После пакетного удаления приложения удалите ссылку на файл библиотеки в проекте ПЛК.

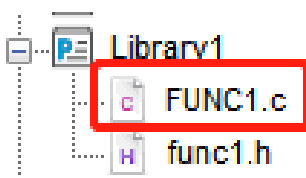
3) Condition information (Информация об условиях)

Модели из черного списка использовать нельзя, можно использовать только модели из белого списка.

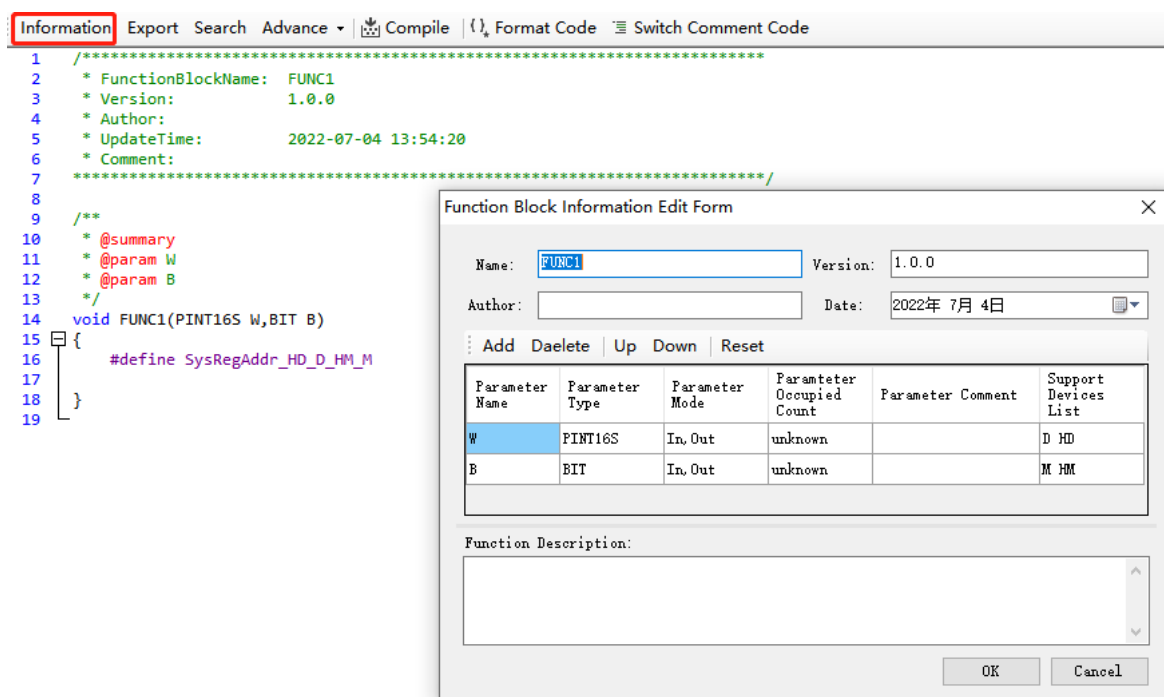


8.8.4.2 Source file information (Информация об исходном файле)

Нажмите Source file (Исходный файл), чтобы отредактировать информацию на панели проекта:



Во всплывающем окне Source file (Исходный файл) щелкните по Information (Информация), чтобы изменить информацию об исходном файле. Сигнатура функции исходного файла будет изменена, и код будет соответствующим образом изменен.



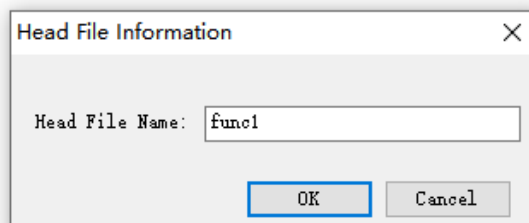
8.8.4.3 Header file information (Информация о файле заголовка)

Нажмите Header file (Файл заголовка), чтобы отредактировать информацию на панели проекта:



Во всплывающем окне Header (Заголовок) щелкните по Information (Информация), чтобы изменить информацию о файле заголовка. Сигнатура функции файла заголовка будет изменена, и код будет соответствующим образом изменен.

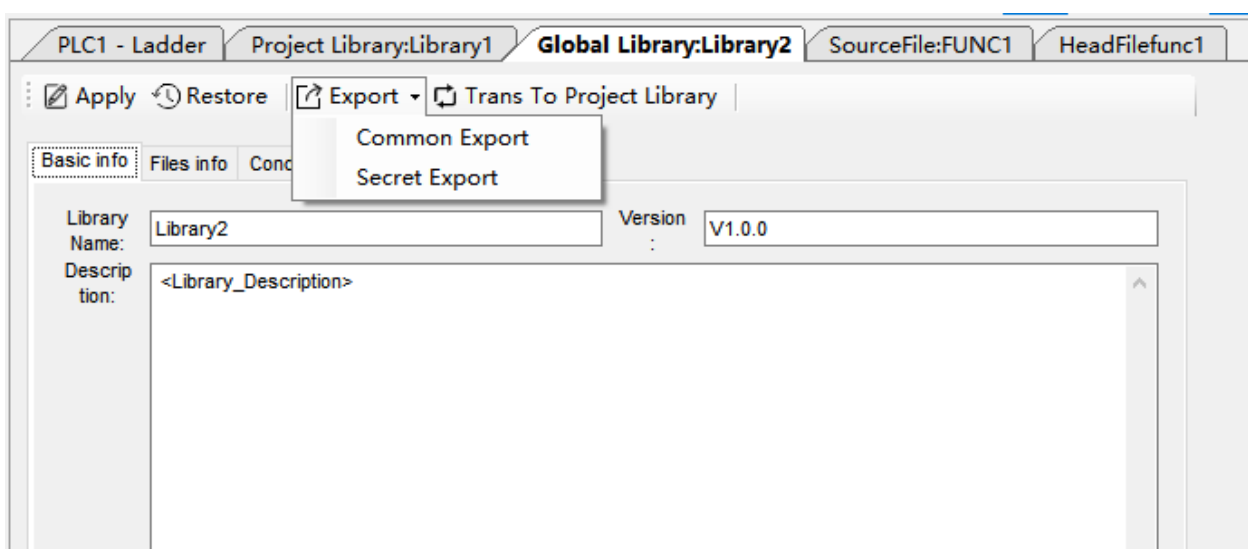
```
Information Export Search Advance ▾ |  Compile |  Format Code |  Switch Comment Code  
1 #ifndef _FUNC1_H  
2 #define _FUNC1_H  
3  
4 #endif
```



8.8.5 Экспорт

8.8.5.1 Экспорт библиотеки функций

Нажмите "Project library" (Библиотека проекта) или "Global library" (Глобальная библиотека) на панели проекта в левой части окна, чтобы отредактировать информацию. Во всплывающем окне с информацией о библиотеке нажмите "Export" (Экспорт):

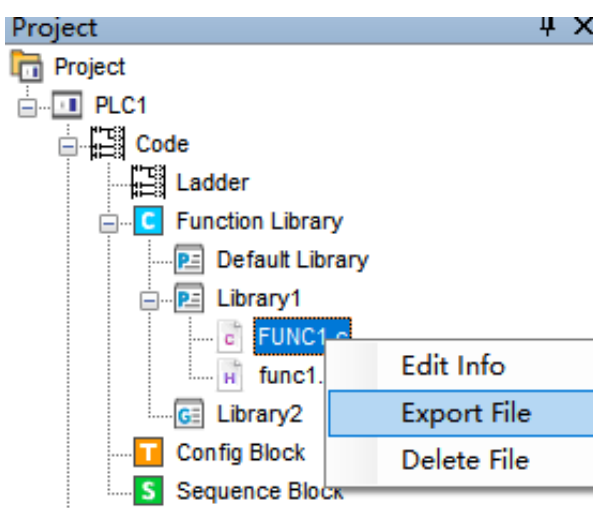


Normal export (Обычный экспорт): Если файл библиотеки является редактируемой библиотекой, экспортируйте его как редактируемую библиотеку; если файл библиотеки является нередактируемой библиотекой, экспортируйте его как не-редактируемую библиотеку.

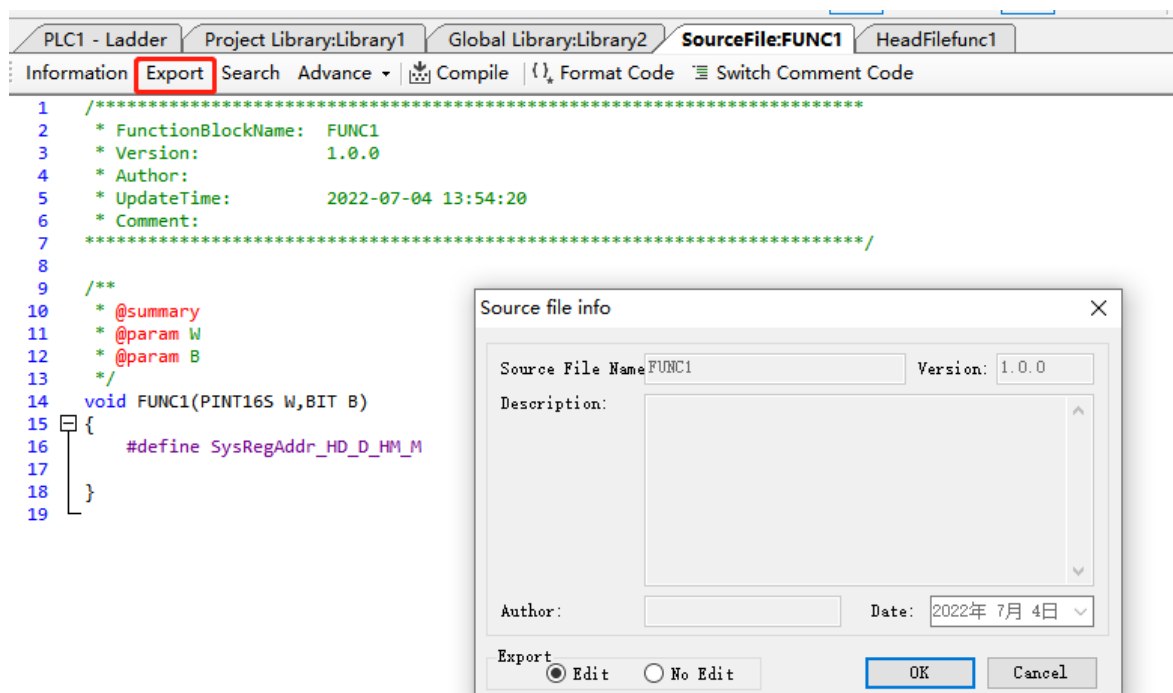
Encrypted export (Зашифрованный экспорт): Если файл библиотеки доступен для редактирования, исходный файл в файле библиотеки компилируется и экспортируется как нередактируемая библиотека; если файл библиотеки недоступен для редактирования, сохраните файл библиотеки напрямую.

8.8.5.2 Экспорт исходного файла/файла заголовка

На панели проекта щелкните правой кнопкой мыши по исходному файлу/файлу заголовка, который нужно экспортировать, и выберите Export file (Экспортировать файл)



Либо в столбце проекта в левой части окна можно выбрать исходный файл/файл заголовка, который нужно экспортировать, а затем щелкнуть по "export" (Экспорт) в окне редактирования в правой части:



Выберите режим экспорта (редактируемый или нередатируемый) во всплывающем окне с информацией о файле.

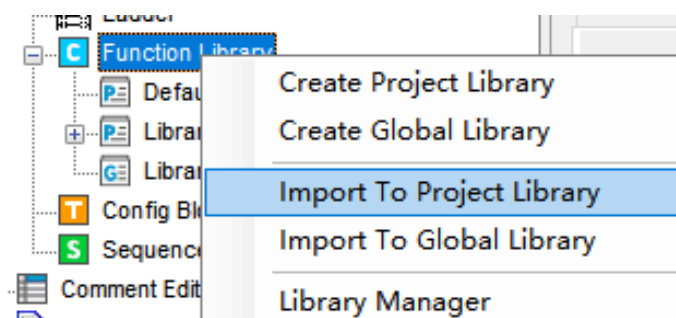
После ввода настроек нажмите ОК и выберите путь сохранения файла.

Выбрав путь, нажмите ОК, чтобы завершить экспорт.

8.8.6 Импорт

8.8.6.1 Импорт библиотеки функций

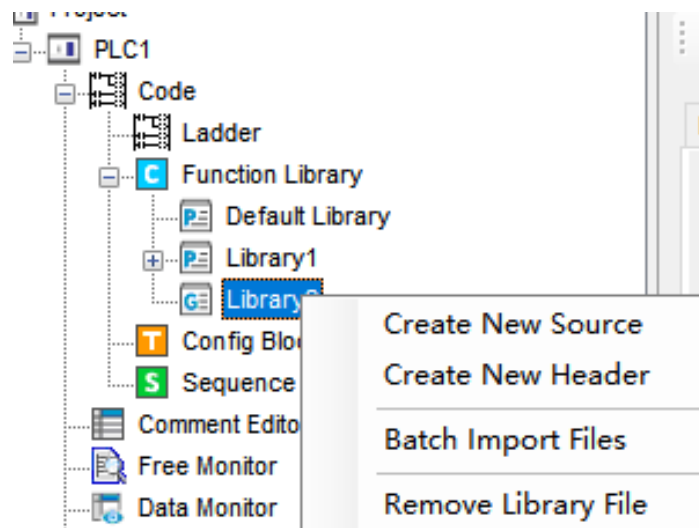
Выберите "Function Library" (Библиотека функций) на панели инструментов "Project" (Проект) в левой части окна, щелкните правой кнопкой мыши и выберите "Import to Project Library" (Импорт в библиотеку проекта) или "Import to Project Library" (Импорт в глобальную библиотеку):



Во всплывающем окне "Select function library file" (Выбор файла библиотеки функций) выберите файл и нажмите "Open" (Открыть), чтобы завершить импорт.

8.8.6.2 Импорт функциональных файлов

Щелкните правой кнопкой мыши по "Project Library" (Библиотека проекта) или "Global Library" (Глобальная библиотека) на панели проекта в левой части окна, чтобы импортировать файлы, и выберите "Batch import files" (Пакетный импорт файлов):

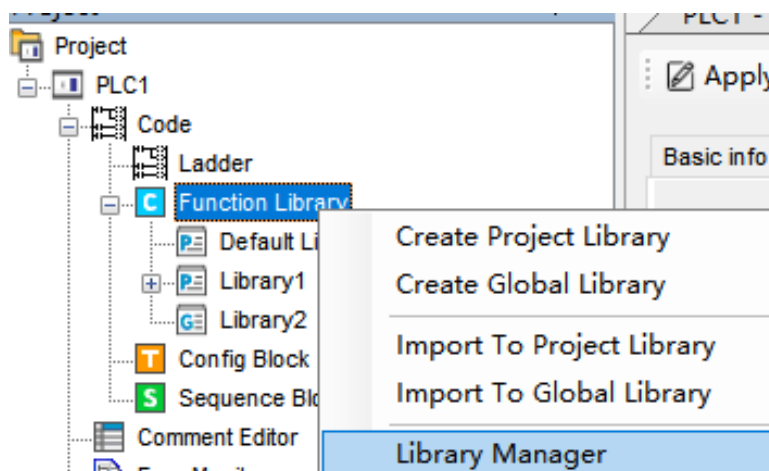


В окне "Select file" (Выбрать файл) выберите функциональный файл, который необходимо импортировать, и нажмите "Open" (Открыть), чтобы завершить импорт файла.

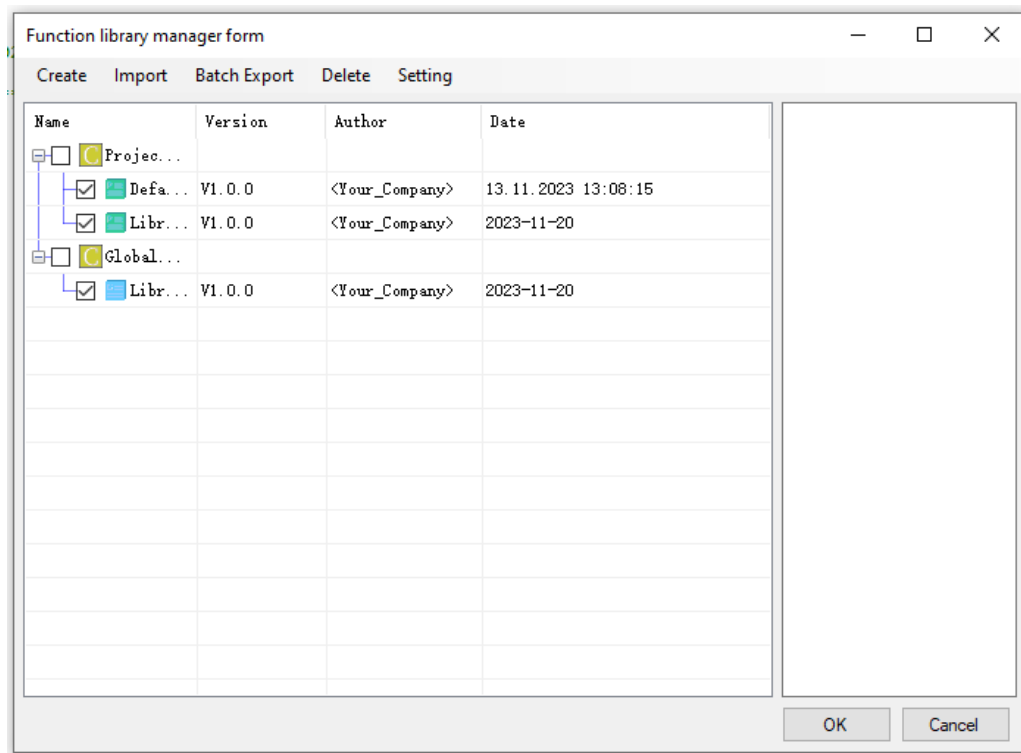
8.8.7 Другие функции

8.8.7.1 Диспетчер библиотек

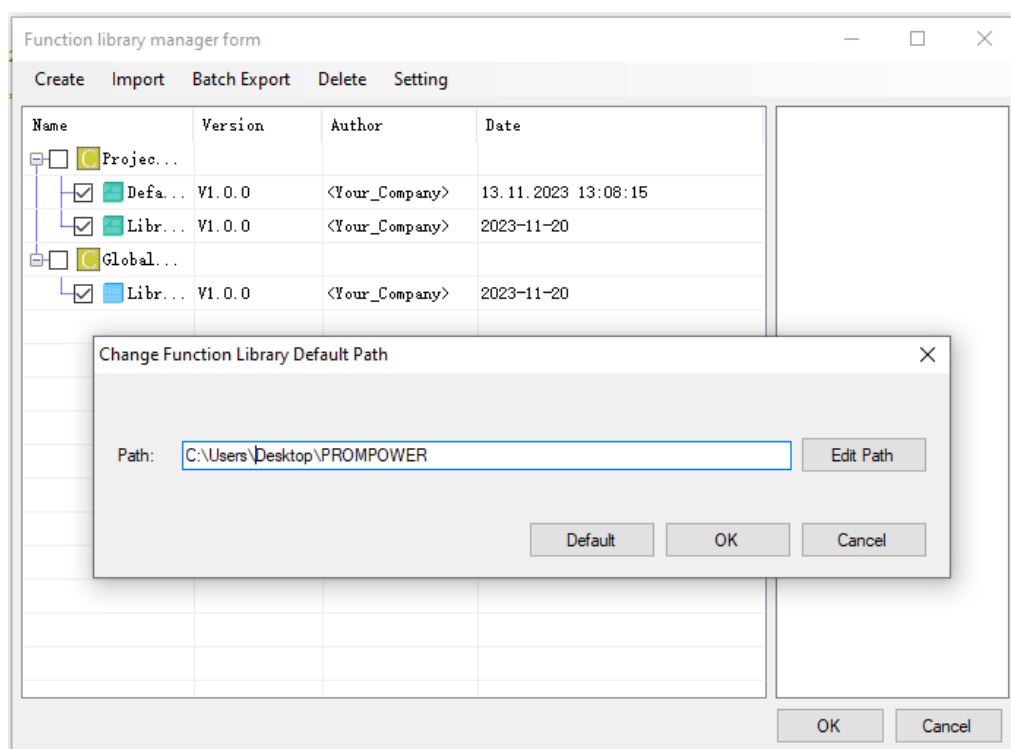
Выберите "Function Library" (Библиотека функций) на панели инструментов "Project" (Проект) в левой части окна, щелкните правой кнопкой мыши и выберите "Library management" (Управление библиотекой):



Во всплывающем окне "Function Library management window" (Окно управления библиотекой функций) можно создавать, импортировать, удалять (в том числе удаление файлов библиотеки, на которые имеются ссылки в проекте) и настраивать библиотеку функций. Проверив библиотеку функций в окне управления, можно активировать функцию генерирования, а затем вызвать ее в проекте.

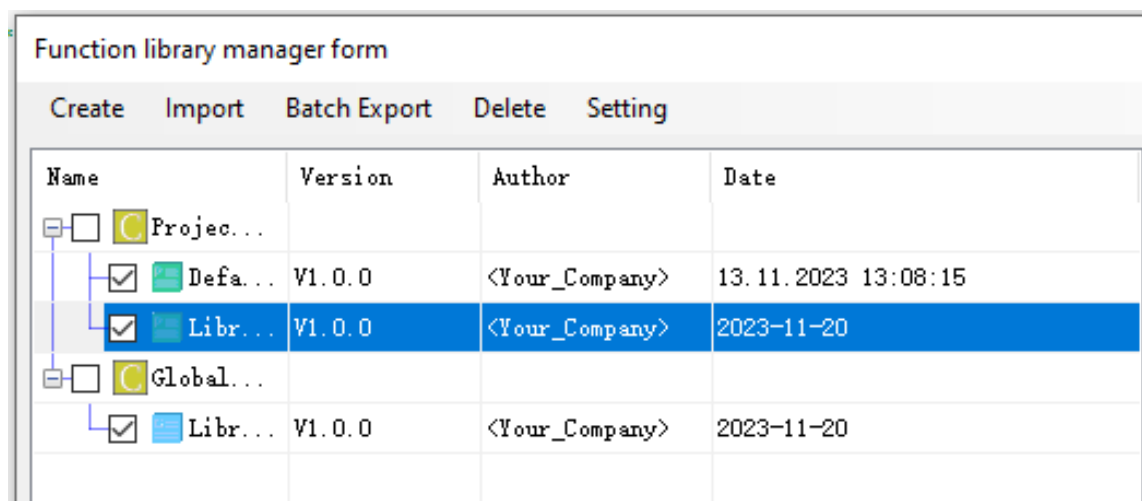


Нажмите Settings (Настройки), чтобы изменить каталог глобальной библиотеки:



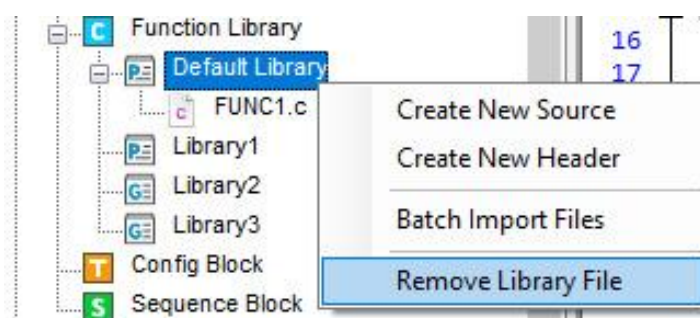
8.8.7.2 Удаление файла библиотеки

В окне "Function Library management window" (Окно управления библиотекой функций), описанном в предыдущей главе, отметьте галочкой соответствующий файл библиотеки и нажмите Delete (Удалить), чтобы удалить файл библиотеки в текущем проекте.



8.8.7.3 Перемещение файла библиотеки

Щелкните правой кнопкой мыши по "Project Library" (Библиотека проекта) или "Global Library" (Глобальная библиотека) на панели проекта в левой части окна, чтобы импортировать файл, и выберите "Remove Library file" (Переместить файл библиотеки):

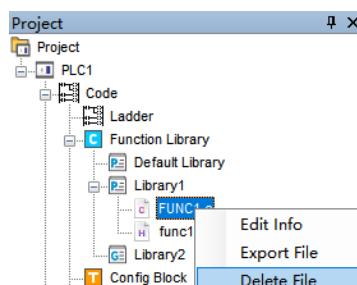


Примечание: Перемещение файла библиотеки означает отмену применения файла из текущего проекта без его удаления.

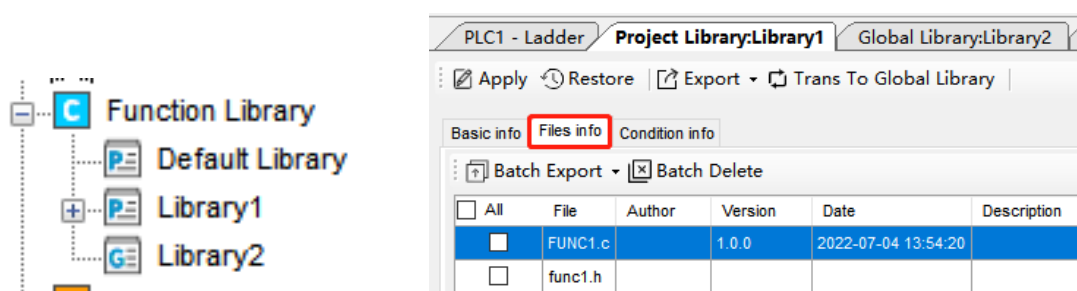
8.8.7.4 Удаление исходного файла/файла заголовка

Удалить исходный файл/файл заголовка можно двумя способами:

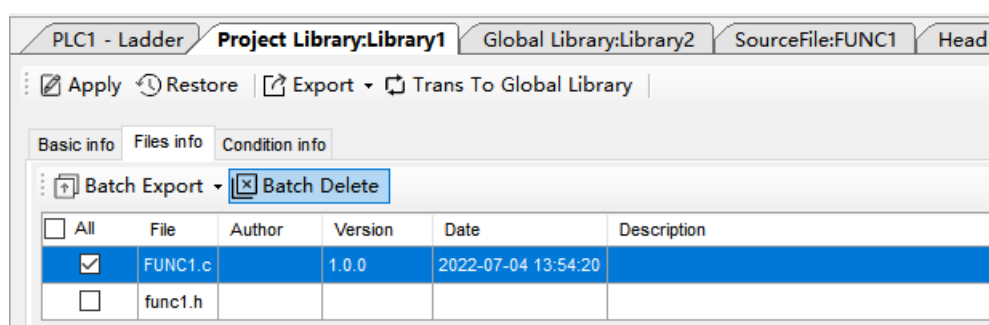
Способ 1: На панели проекта щелкните правой кнопкой мыши по исходному файлу/файлу заголовка, который нужно экспортировать, и выберите Delete file (Удалить файл):



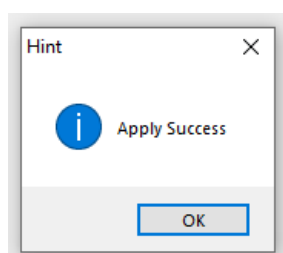
Метод 2: Чтобы удалить файл, выберите библиотеку функций на панели проекта в левой части окна:



Отметьте галочками удаляемые файлы и нажмите "Batch delete" (Пакетное удаление):

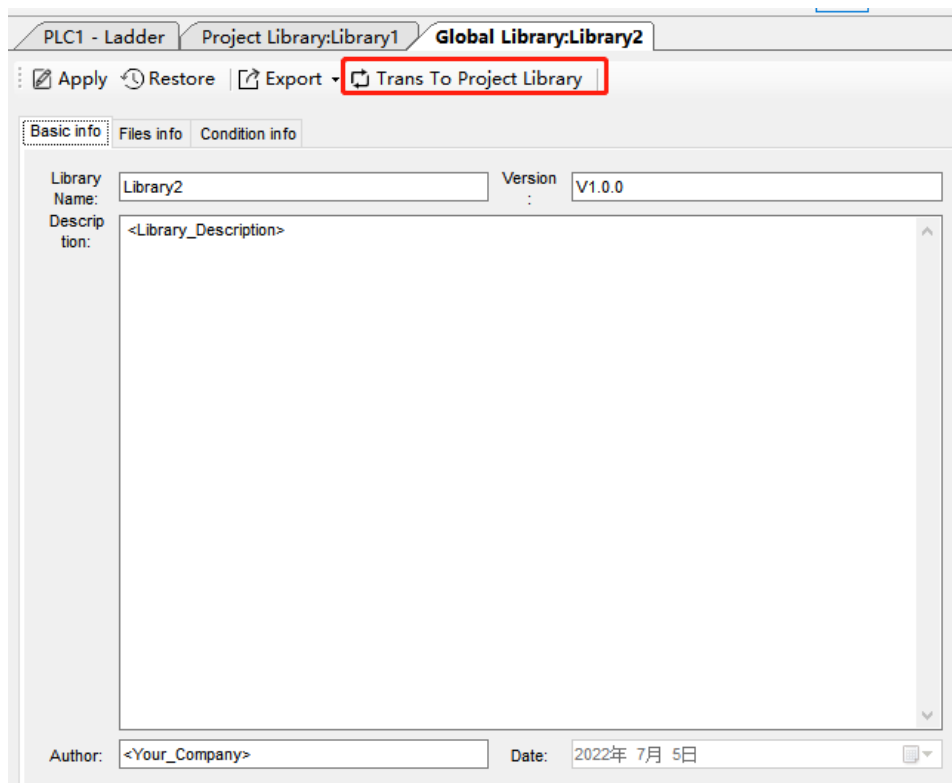
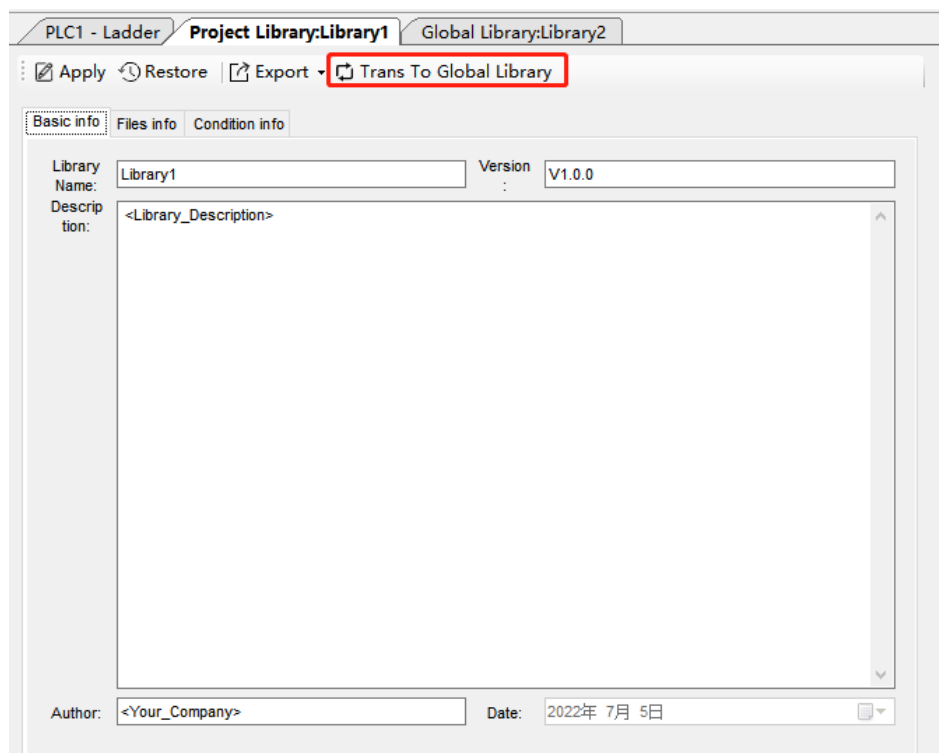


Нажмите "Apply" (Применить), после чего появится подсказка в виде сообщения "Successfully applied" (Успешно применено). Файл удален.



8.8.7.5 Преобразование

«Глобальную библиотеку» и «библиотеку проекта» можно преобразовывать друг в друга, а также можно открыть окно редактирования библиотеки функций (подробную процедуру см. в главе 8-8-7-4, метод 2).



8.8.7.6 Компилирование

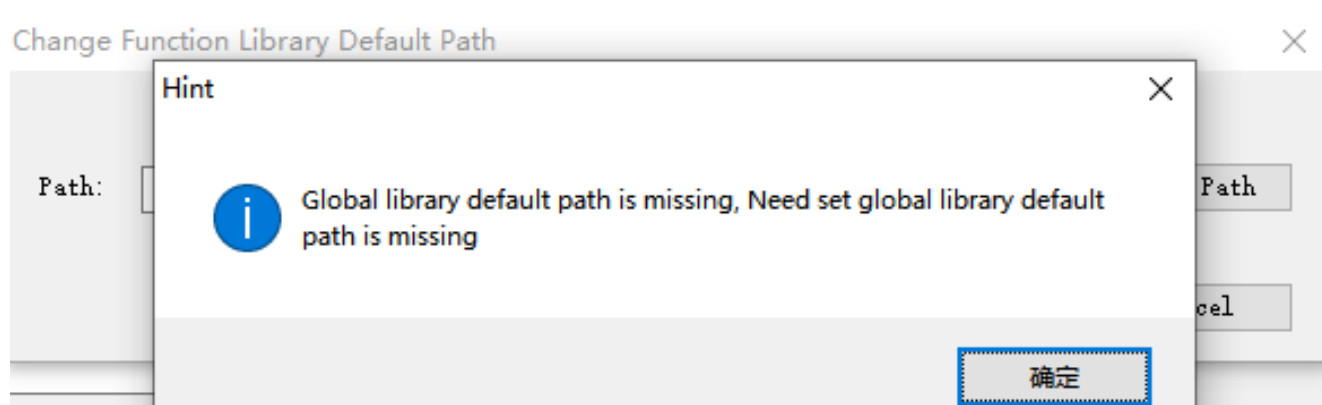
Щелкните по исходному файлу на панели проекта в левой части окна и нажмите "Compile" (Компилировать) в окне редактирования справа.



8.8.7.7 Настройка каталога глобальной библиотеки

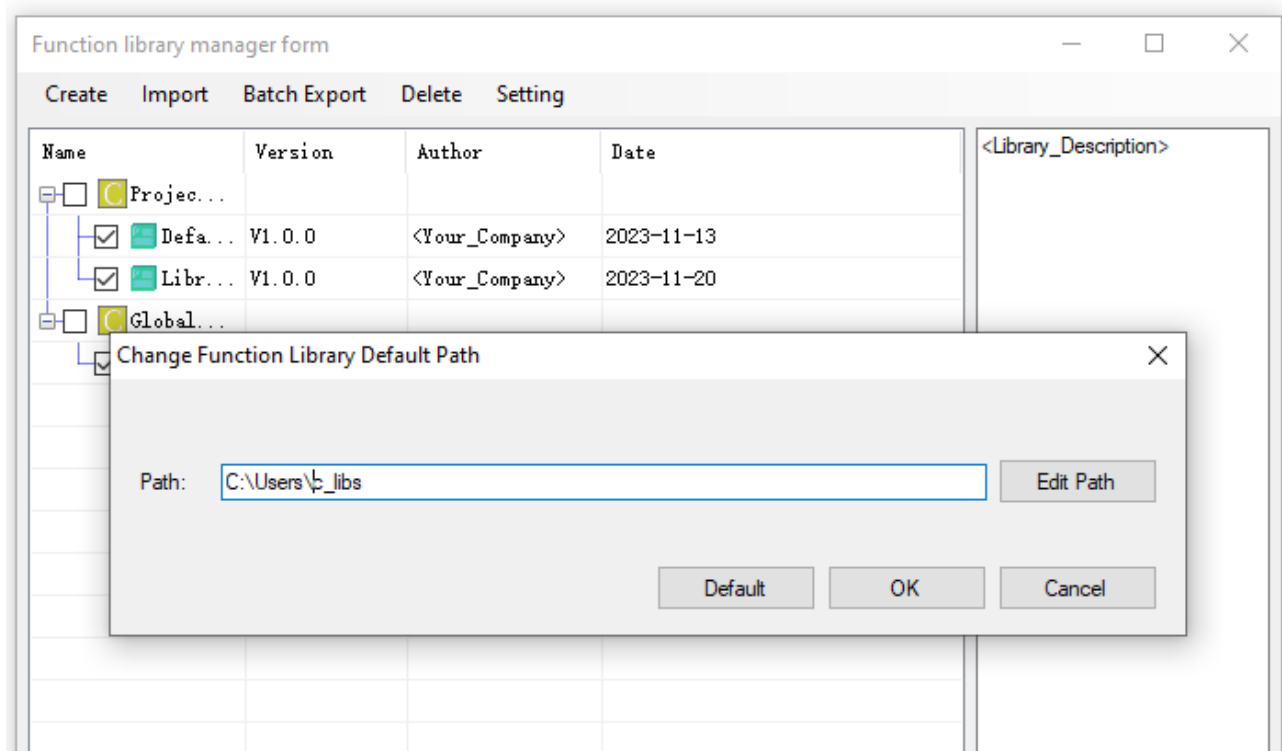
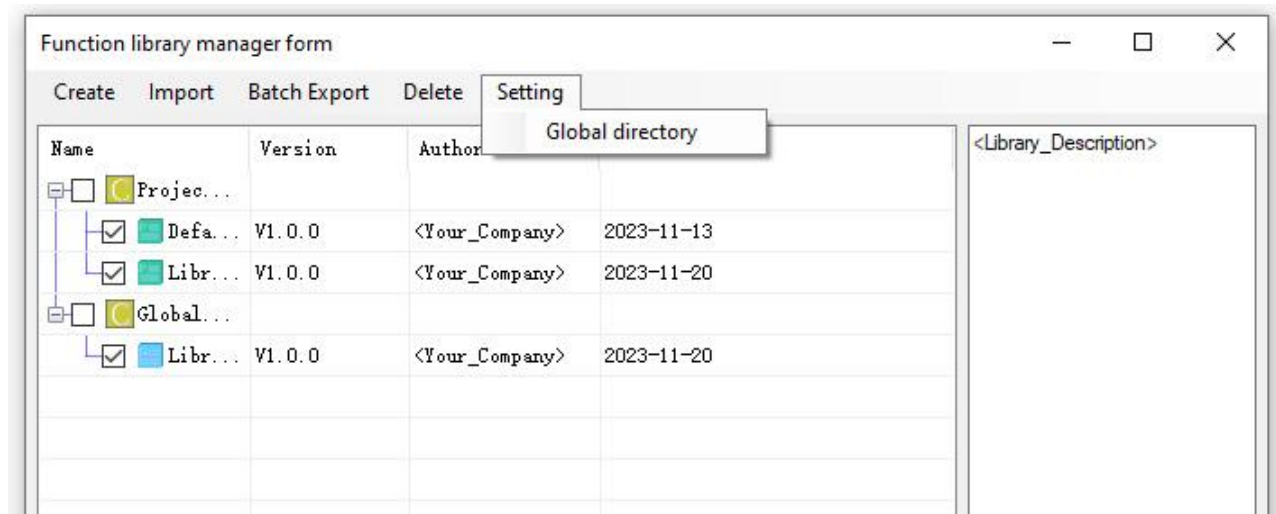
Существуют три метода настройки глобальной библиотеки:

Метод 1: Откройте окно управления библиотекой (см. главу 8-8-7-1 «Диспетчер библиотек» для получения подробной информации). Если каталог глобальной библиотеки не настроен, появится запрос на настройку каталога глобальной библиотеки.



Метод 2: Если каталог глобальной библиотеки не был настроен во время создания новой глобальной библиотеки, появится тот же запрос, что и для метода 1. Путь можно задать во всплывающем окне "Change Function Library Default Path" (Изменить путь к библиотеке функций по умолчанию).

Метод 3: Откройте окно "Library Manager" (Диспетчер библиотек) (см. главу 8-8-7-1 «Диспетчер библиотек») и выберите "Settings" > "Global Library Directory" (Настройки > Каталог глобальной библиотеки), как показано ниже:



8.9 Указания по применению

- В одном файле функционального блока Func Block можно записать множество функций, и они могут вызывать друг друга.
- Все файлы функциональных блоков независимы друг от друга — функцию в другом функциональном блоке нельзя вызвать.
- Файлы функционального блока Func Block могут вызывать функции библиотеки языка C в виде функции float, например, sin, cos, tan.
- ПЛК серии RMP20 поддерживают локальные и глобальные переменные. Поэтому блок, написанный на языке C, более универсальный и удобный.
- Рекомендации по использованию глобальных переменных:
 - ① Для хранения данных глобальных переменных используйте область программных компонентов вместо обычной памяти.
 - Область программных компонентов ПЛК может использоваться как пространство глобальных переменных, при этом безопасность гарантирована.
 - ② Пример использования

В качестве примера рассмотрим тип FP64:

```
1  /*****
2      FunctionBlockName:  FUNC1
3      Version:           1.0.0
4      Author:
5      UpdateTime:       2020/1/3 10:30:47
6      Comment:
7
8  *****/
9  void Test();
10  FP64 * GlobalV;          declaration
11
12  void FUNC1( WORD W , BIT B )
13  {
14      #define SysRegAddr_HD_D_HM_M
15
16      GlobalV = (FP64*)&W[0];  initialization
17
18      Test();
19  }
20  void Test()
21  {
22      #define SysRegAddr_HD_D_HM_M
23      FP64 value = GlobalV[0];  using
24      *(FP64*)&HD[0] = value;
25  }
26
```

Как показано на рисунке выше, глобальный указатель GlobalV объявляется вне функции, а затем инициализируется в главной функции, чтобы указывать на

пространство программного компонента. Первый адрес пространства — это адрес, где находится $W[0]$. Наконец, значение переменной можно получить, используя указатель в других функциях.

В качестве примера рассмотрим тип "структура":

```
1  #ifndef _STRUCT_H
2  #define _STRUCT_H
3
4  typedef struct
5  {
6      INT16U V;
7      FP64 S;
8  } ExStruct;
9
10 #endif
```

Объявление структуры

```
1  /*****
2      FunctionBlockName:  STRUCT
3      Version:            1.0.0
4      Author:
5      UpdateTime:         2020/1/3 10:58:49
6      Comment:
7  *****/
8
9  #include "struct.h"
10 void test();
11 ExStruct* ST;
12 void STRUCT( WORD W, BIT B )
13 {
14
15     #define SysRegAddr_HD_D_HM_M
16
17     ST = ((ExStruct*)&W[0]);
18
19     ST->V = 10;
20     ST->S = 100.001;
21
22     Test(ST);
23 }
24 void Test(ExStruct* ex)
25 {
26     #define SysRegAddr_HD_D_HM_M
27
28     *(INT16U*)&HD[0] = ex->V;
29     *(FP64*)&HD[2] = ex->S;
30 }
```

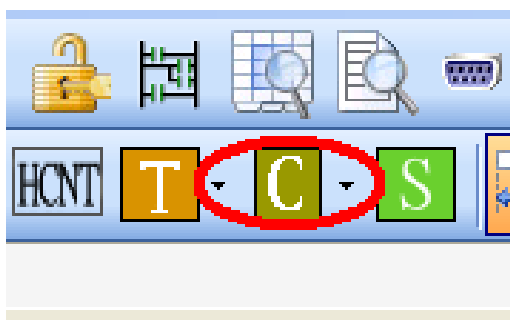
the header file contained declaration

initialization

using

Пример использования глобальной переменной типа «структура»

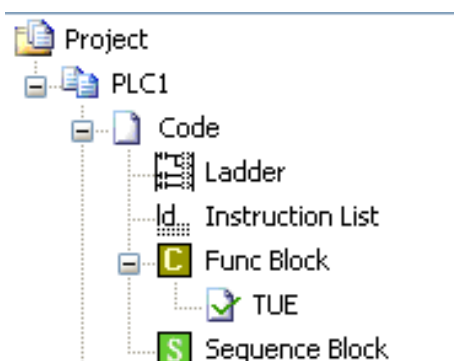
Программное обеспечение PROMPOWER PLC Studio версии 3.3 и более поздних версий имеют библиотеку функций на языке C:



В этом функциональном блоке пользователь может напрямую вызывать функцию на языке C:

C	S				Ld m0	
C	TCA	Calculation	area of a circle			
C	TCC	Circumference	calculation			
C	TCRC	CRC	Check			
C	TDSL	Input data (short)	from big to small order			
C	TDSS	Input data (short)	from small to large order			
C	TECA	Calculation	area of a circle			
C	TECC	Circumference	calculation			
C	TEEX	Exponentiation	calculation			
C	TEL10	Natural	logarithm			
C	TELO	Natural	logarithm			
C	TEPTH	Known two right-angle	sides and the hypotenuse demanded			
C	TEPTR	Known one right-angle	side and hypotenuse need to demand the other right-angle side			
C	TEQE	Quadratic equation	(float)			
C	TESUM	Sum of memory 32-bit	floating data			
C	TETP	The product of memory	data (float)			
C	TEUE	Quadratic equation	(float)			
C	TEX	Exponentiation	calculation			
C	TFA	Factorial	solving			
C	TITF	Inverse trigonometric	functions			
C	TQE	Quadratic equation	(short)			
C	TSUM	Sum of memory 32-bit	integer data			
C	TTP	The product of memory	data (short)			
C	TUE	Quadratic equation	(short)			

Пример: Выберите TUE, после чего имя функции отобразится на панели проекта:



Пользователь может в любое время вызвать ее в окне редактирования лестничной диаграммы.

8.10 Обеспечение качества при использовании языка C

(1) Для катушки задано второе макроопределение

Некоторые пользователи дополнительно расширили тип программного компонента после его задания, как показано в следующем коде:

```
#define SysRegAddr_HD_D_HM_M_X_Y
```

```
#define OUT Y[1]
```

```
OUT = 100;
```

Второе макроопределение катушек, например Y, не допускается, поскольку чтение и запись данных катушки осуществляется не просто указателем, а через функцию. Иначе компилятор не сможет обработать данные, что приведет к ошибке.

(2) Использование значения катушки в качестве условия оценки

Пользователь использует значение катушки в качестве условия оценки оператора if (если), как показано в следующем коде:

```
if(X[0])D[0] = 10;
```

Этот метод записи приведет к ошибке во время компиляции, поскольку компилятор допустил ошибку во время внутренней обработки. Рекомендуется изменить строку следующим образом:

```
if(X[0])
```

```
D[0] = 10;
```

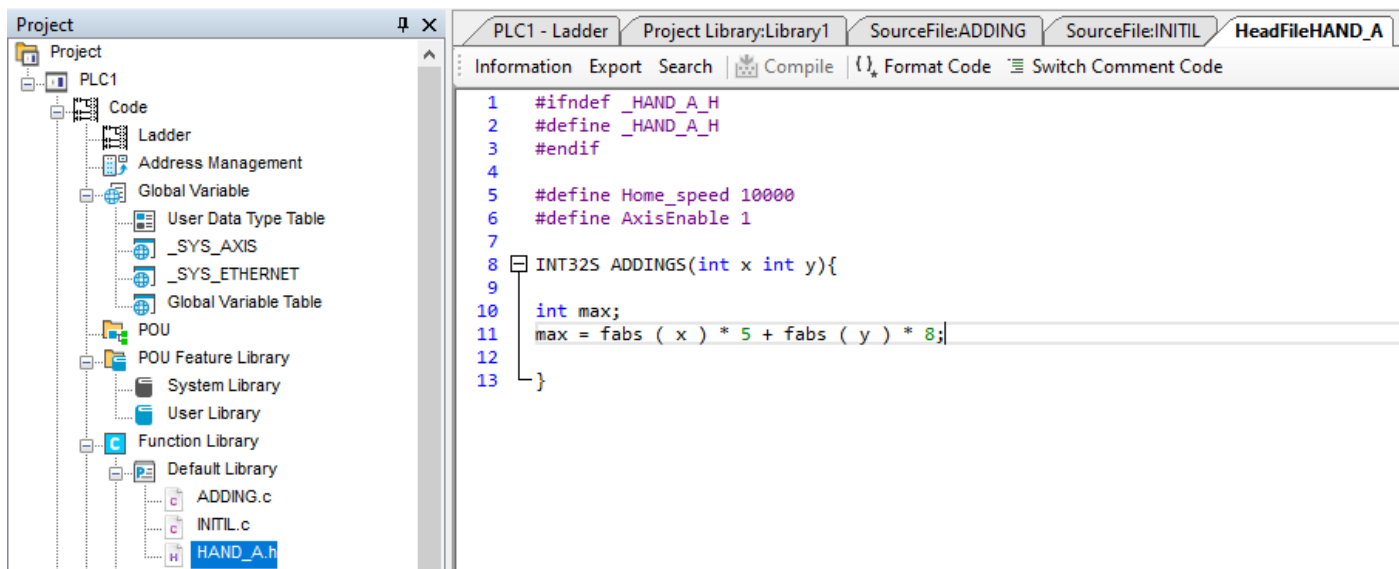
(3) Использование DM

В настоящий момент DM[0] не поддерживается. Поддерживаются только операции с двойными словами DW и FW.

(4) Во время компиляции появляется сообщение об ошибке, и цвет макроопределения меняется на черный.

Это явление вызвано наличием в коде символов полного угла. Символы полного угла можно удалить с помощью форматирования.

(5) Функция языка C в файле заголовка не имеет функции компиляции.



В файле заголовка нет функции компиляции. Скомпилировать можно только исходный файл. Файл заголовка не может быть скомпилирован отдельно.

(6) Когда два исходных файла вызывают файл заголовка, нужно написать объявление только в одном исходном файле.

Оба исходных файла записаны и скомпилированы правильно, но программа загрузки неверна.

Использование `#include "xxx.h"` вне функции можно рассматривать как глобальное включение этого файла заголовка. Компиляция отдельного исходного файла не вызывает проблем.

Функцию файла заголовка можно понимать следующим образом: компилятор заменяет `#include "xxx.h"` переменными и функциями, объявленными в файле заголовка во время предварительной обработки кода.

Однако в процессе загрузки компилируются и связываются несколько исходных файлов. После предварительной обработки оба исходных файла будут содержать объявления переменных и функций в файле заголовка. Во время связывания будут возникать повторяющиеся ошибки объявления, и PROMPOWER PLC Studio будет отображаться как ошибка связывания.

Рекомендация:

Правильно включайте файл заголовка, если необходимо использовать его содержимое, а не просто вслепую включайте файл заголовка непосредственно за пределами функции.

```

1  /*****
2  * FunctionBlockName:  ADDING
3  * Version:           1.0.0
4  * Author:
5  * UpdateTime:        2023-11-20 16:12:01
6  * Comment:
7  *****/
8
9
10 #include "HAND_A.h"
11
12 void ADDING( Word W, BIT B)
13 {
14 #define SysRegAddr_HD_D_HM_M
15 // #include "HAND_A.h"
16
17 INT32S MUL_A(INT32S r);
18
19 DW[10] = ADDINGS(-10, 30);
20 DW[14] = MUL_A(1);
21
22 }
23
24 INT32S MUL_A(INT32S r) {
25
26 INT32S mas;
27 mas = ADDINGS ( -10, 30 ) + r;
28
29 }

```

```

1  /*****
2  * FunctionBlockName:  INITIL
3  * Version:           1.0.0
4  * Author:
5  * UpdateTime:        2023-11-20 16:12:32
6  * Comment:
7  *****/
8
9
10 #include "HAND_A.h"
11
12 void INITIL(PINT16S W,PBIT B)
13 {
14 #define SysRegAddr_HD_D_HM_M
15
16 DW[10] = ADDINGS(-10, 30);
17
18 }

```

8.11 Таблица функций

Библиотека функций по умолчанию

Константа	Данные	Описание
_LOG2	(двойн.) 0.693147180559945309417232121458	Логарифм из 2
_LOG10	(двойн.) 2.3025850929940459010936137929093	Логарифм из 10
_SQRT2	(двойн.) 1.41421356237309504880168872421	Радикал из 2
_PI	(двойн.) 3.1415926535897932384626433832795	PI
_PIP2	(двойн.) 1.57079632679489661923132169163975	PI/2
_PIP2x3	(двойн.) 4.71238898038468985769396507491925	PI*3/2

Строковая функция	Описание
<code>void * memchr(const void *s, int c, size_t n);</code>	Возвращает первую позицию c среди n слов перед позицией s
<code>int memcmp(const void *s1, const void *s2, size_t n);</code>	Сравнивает первые n слов позиций s1 и s2
<code>void * memcpy(void *s1, const void *s2, size_t n);</code>	Копирует n слов из позиции s2 в позицию s1 и возвращает s1
<code>void * memset(void *s, int c, size_t n);</code>	Заменяет n слов, начиная с позиции s на слово c и возвращает в позицию s
<code>char * strcat(char *s1, const char *s2);</code>	Подключает строку ct за строкой s
<code>char * strchr(const char *s, int c);</code>	Возвращает позицию первого слова c в строке s
<code>int strcmp(const char *s1, const char *s2);</code>	Сравнивает строки s1 и s2
<code>char * strcpy(char *s1, const char *s2);</code>	Копирует строку s1 в строку s2

Математическая функция двойной точности	Математическая функция одинарной точности	Описание
<code>double acos(double x);</code>	<code>float acosf(float x);</code>	Обратная косинусная функция
<code>double asin(double x);</code>	<code>float asinf(float x);</code>	Обратная синусная функция
<code>double atan(double x);</code>	<code>float atanf(float x);</code>	Обратная функция тангенса
<code>double atan2(double y, double x);</code>	<code>float atan2f(float y, float x);</code>	Значение обратного тангенса параметра (y/x)
<code>double ceil(double x);</code>	<code>float ceilf(float x);</code>	Возвращает наименьшее двойное целое число, которое больше или равно параметру x
<code>double cos(double x);</code>	<code>float cosf(float x);</code>	Косинусоидная функция
<code>double cosh(double x);</code>	<code>float coshf(float x);</code>	Функция гиперболического косинуса, $\cosh(x) = (e^x + e^{-x})/2$



Математическая функция двойной точности	Математическая функция одинарной точности	Описание
double exp(double x);	float expf(float x);	Экспонента (e^x) натуральных данных
double fabs(double x);	float fabsf(float x);	Абсолютное значение параметра x
double floor(double x);	float floorf(float x);	Возвращает наибольшее двойное целое число, которое меньше или равно параметру x
double fmod(double x, double y);	float fmodf(float x, float y);	Если y не равно нулю, возвращает остаток плавающего значения параметра x/y
double frexp(double val, int *_far *exp);	float frexpf(float val, int *_far *exp);	Разбивает плавающие данные x на мантиссу и экспоненту $x = m \cdot 2^{\text{exp}}$, возвращает мантиссу m , сохраняет логарифм в exp .
double ldexp(double x, int exp);	float ldexpf(float x, int exp);	x умножить на (два в степени n) равно $x \cdot 2^n$.
double log(double x);	float logf(float x);	Логика натурального логарифма
double log10(double x);	float log10f(float x);	logarithm ($\log_{10}x$)
double modf(double val, double *pd);	float modff(float val, float *pd);	Разбить плавающие данные x на целую часть и десятичную часть, возвращает десятичную часть, сохраняет целую часть в параметре ip .
double pow(double x, double y);	float powf(float x, float y);	Значение степени параметра y (x^y)
double sin(double x);	float sinf(float x);	Синусоидальная функция
double sinh(double x);	float sinhf(float x);	Функция гиперболического синуса, $\sinh(x) = (e^x - e^{-x})/2$
double sqrt(double x);	float sqrtf(float x);	Квадратный корень параметра x
double tan(double x);	float tanf(float x);	Функция тангенса.
double tanh(double x);	float tanhf(float x);	Функция гиперболического тангенса $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$

Метод использования функций в таблице:

float asinf(float x);

float asinf: float означает, что возвращаемое значение имеет формат float.

float x: float означает, что формальный параметр функции имеет формат float. На практике, необязательно использовать формат float. См. строку 14 в примере ниже:

```
9 void ZHENGXIAN( WORD W , BIT B )
10 {
11 int a;
12 float x,y,z;
13 x=FW[0];
14 y=asinf(x);
15 z=180*y/3.14159;
16 a=(int) z;
17 W[2]=a;
18 }
```

Библиотека специальных функций для операций с Flash-регистрами.

Специальная функция для операций с Flash-регистрами	Описание
flash_copy (void *dst, void *src, size_t len);	Функция для копирования данных во Flash-регистр. DST: начальный адрес целевого регистра, в который происходит копирование; SRC: адрес исходных данных; Len: количество скопированных байтов;
flash_move (void *dst, void *src, size_t len);	Байты данных, скопированных во Flash-регистр. Если целевая область и исходная область перекрывают друг друга, то функция flash_Move может гарантировать, что байты перекрывающейся области будут скопированы в целевую область до того, как исходная строка будет перезаписана, при этом исходное содержимое будет изменено после копирования. Однако, когда целевая область не перекрывается исходной областью, происходит то же самое, что при использовании функции flash_copy.

Специальная функция для операций с Flash-регистрами	Описание
	DST: начальный адрес целевого регистра, в который происходит копирование; SRC: адрес исходных данных; Len: количество скопированных байтов;
flash_set_int8 (void* dst, int8 data);	Выполняется определенное назначение Flash-реgistру. DST: начальный адрес целевого регистра; Data: различные типы данных;
flash_set_int16 (void* dst, int16 data);	
flash_set_int32 (void* dst, int32 data);	
flash_set_int64 (void* dst, int64 data);	
flash_set_float32(void* dst, float32 data);	
flash_set_float64(void* dst, float64 data);	

Чтобы проиллюстрировать использование функций в таблице функций, в качестве примера рассмотрим копирование данных и назначение Flash-регистра:

Пример 1: Копирование данных во Flash-регистр FD100

```
flash_copy ( void *dst, void *src, size_t len );
```

Void в функции flash_copy обозначает тип параметра. На практике, необязательно записывать void. См. строку 13 в примере ниже:

```

9  void FUNC1( WORD W , BIT B )
10 {
11  #define SysRegAddr_HD_D_HM_M_FD_SFD
12  char a[8] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h'};
13  flash_copy ( &FD[100], &a, sizeof(a) );//使用sizeof(a)计算a的长度;
14  |
15  |
16  }
```

Пример 2: Задание значения во Flash-регистре

```
flash_set_int16 ( void* dst, int16 data );
```

Преимущества offlash_set_int16 по сравнению с flash_copy:

Использование flash_copy для задания значения во Flash-регистре. Очень неудобно использовать.

```
int temp_val = 1000;
```

```
flash_copy(&FD[1000], &temp_val, sizeof(temp_val));
```

Использование flash_set: flash_set_int32(&FD[1000], 1000);

См. строки 13-18 в примере ниже:

```
9 void FUNC1( WORD W , BIT B )
10 {
11     #define SysRegAddr_HD_D_HM_M_FD_SFD
12     //flash_set系列函数的使用示例
13     flash_set_int8 ( &FD[104], 8 );
14     flash_set_int16 ( &FD[106], 16 );
15     flash_set_int32 ( &FD[108], 32 );
16     flash_set_int64 ( &FD[112], 64 );
17     flash_set_float32 ( &FD[120], 32.32 );
18     flash_set_float64 ( &FD[122], 64.64 );
19 }
20
21
```

Примечание:

- (1) Функция Flash_move требует соответствующей версии прошивки ПЛК нижнего компьютера (версия прошивки: 3.7.2, дата прошивки: 20210528).
- (2) Flash-регистр можно записывать около 1000000 раз, и при каждой записи приходится стирать весь Flash-регистр, что занимает много времени. Частая запись может привести к необратимому повреждению Flash-регистра. Следовательно, частая запись не рекомендуется. В качестве условий управления с осторожностью используйте включенное питание и вибрирующую катушку (например, SM0, SM11).

9. Последовательность BLOCK

В настоящей главе описываются команды блока последовательности и их применение.

Команда блока последовательности:

Мнемонический код	Функция	Лестничная диаграмма	Глава
Блок последовательности			
SBSTOP	Приостановка BLOCK		9-6-1
SBGOON	Перейти к выполнению BLOCK		9-6-1

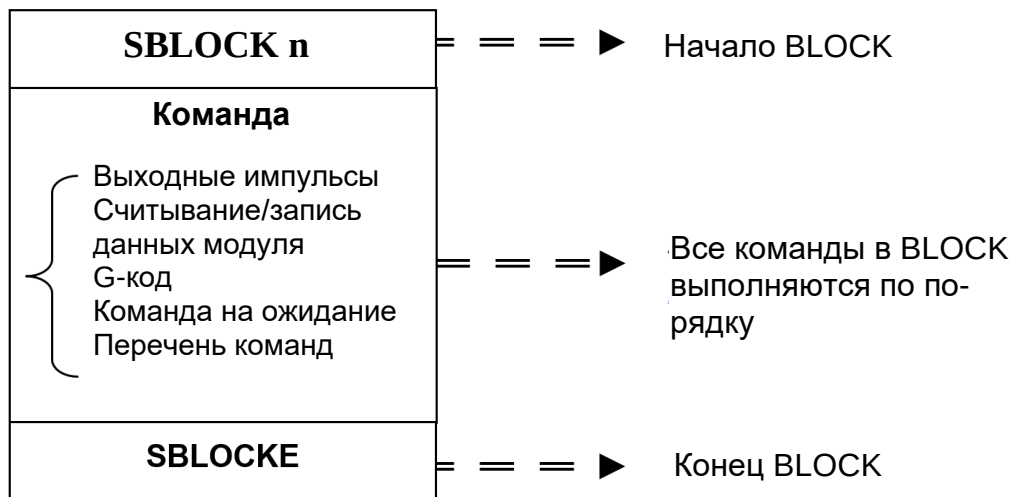
9.1 Описание принципа BLOCK

Блок последовательности, кратко именуемый BLOCK, представляет собой программный блок для реализации некоторых функций. Все команды в блоке выполняются по порядку, что является самым большим отличием от общих процессов.

BLOCK начинается с SBLOCK и заканчивается SBLOCKE; программисты могут записывать команды в BLOCK. Если один блок последовательности BLOCK содержит несколько команд вывода импульсов (или других команд), то команды вывода импульсов будут выполняться в порядке поступления. При этом следующая команда вывода импульсов не будет выполняться, пока не завершилась текущая команда.

ПЛК серии PMP20 поддерживают несколько блоков BLOCK^{*1}.

Полная структура блока BLOCK показана ниже:



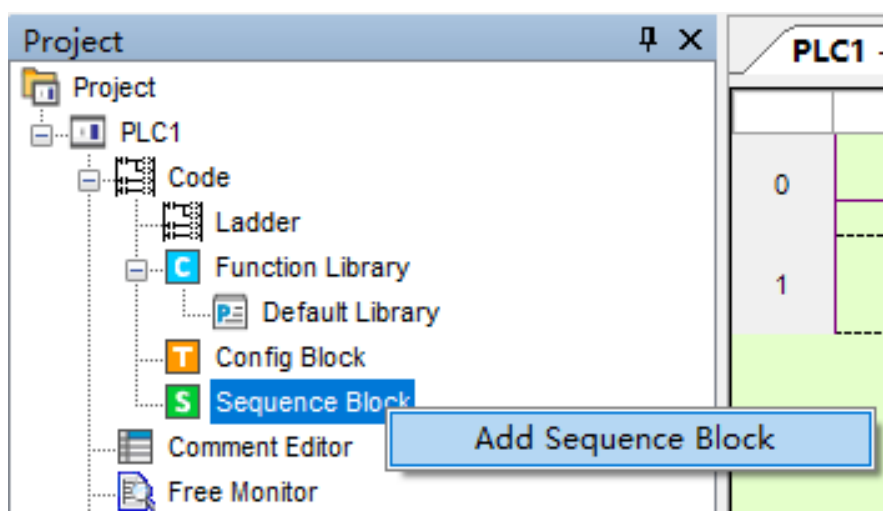
-
- Прошивка версий ниже 3.4.5: ПЛК серии PMP20 поддерживают использование до восьми блоков BLOCK.
- ※1 Прошивка версии 3.4.5 и более поздних версий: ПЛК серии PMP20 могут записывать до 100 блоков BLOCK, но одновременно могут работать только с 8 из них.
- Если блоки BLOCK запускаются замыканием нормально разомкнутой катушки, команды в блоках будут выполняться сверху вниз по очереди. После выполнения последней команды выполнение BLOCK немедленно возобновится сверху вниз. При исчезновении условия запуска выполнение BLOCK не прекратится немедленно.
- ※2 Последнее сканирование завершится, и блок остановится после выполнения невыполненной программы.
- Если блоки BLOCK запускаются нарастающим фронтом импульса с катушки, то последовательная функция BLOCK будет выполняться один раз сверху вниз и не будет выполняться циклически.
- ※3
-

9.2 Вызов BLOCK

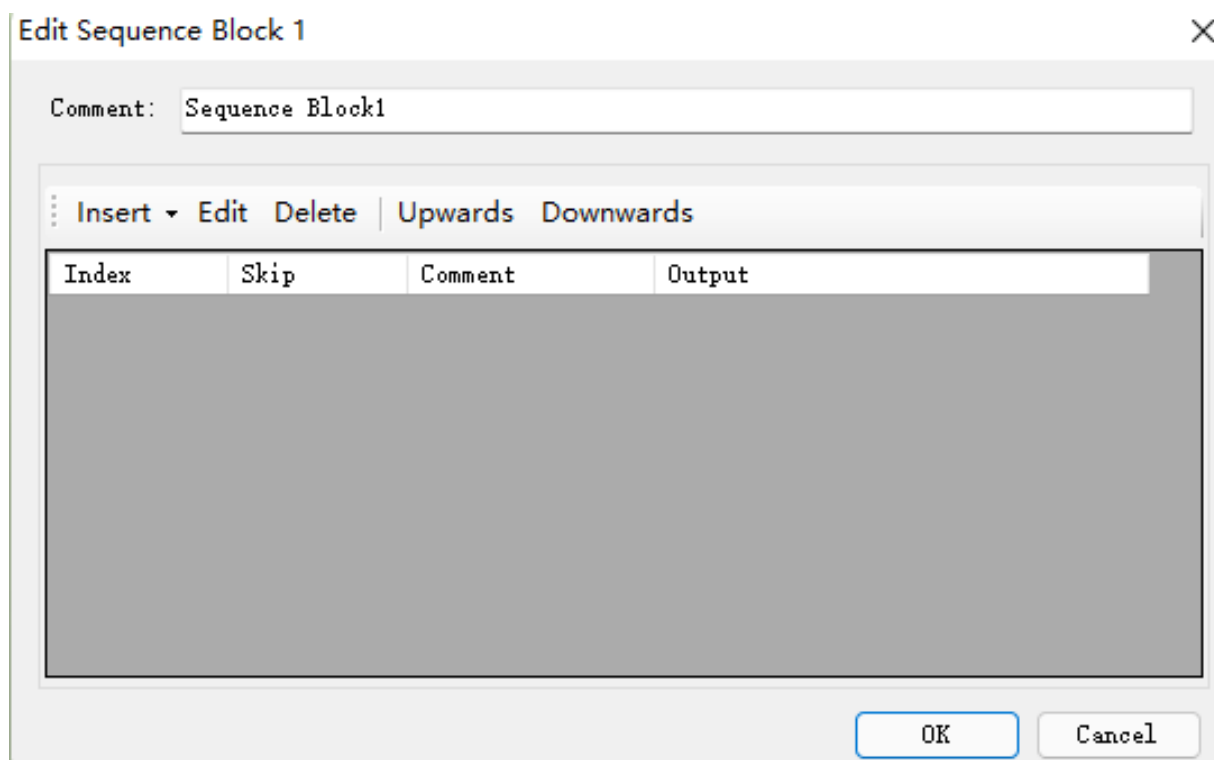
В одном программном файле можно вызывать множество блоков BLOCK. Ниже описан способ добавления BLOCK в программу.

9.2.1 Добавление BLOCK

Откройте ПО PROMPOWER PLC Studio, щелкните правой кнопкой мыши по элементу Sequence block (Блок последовательности) на панели проекта:

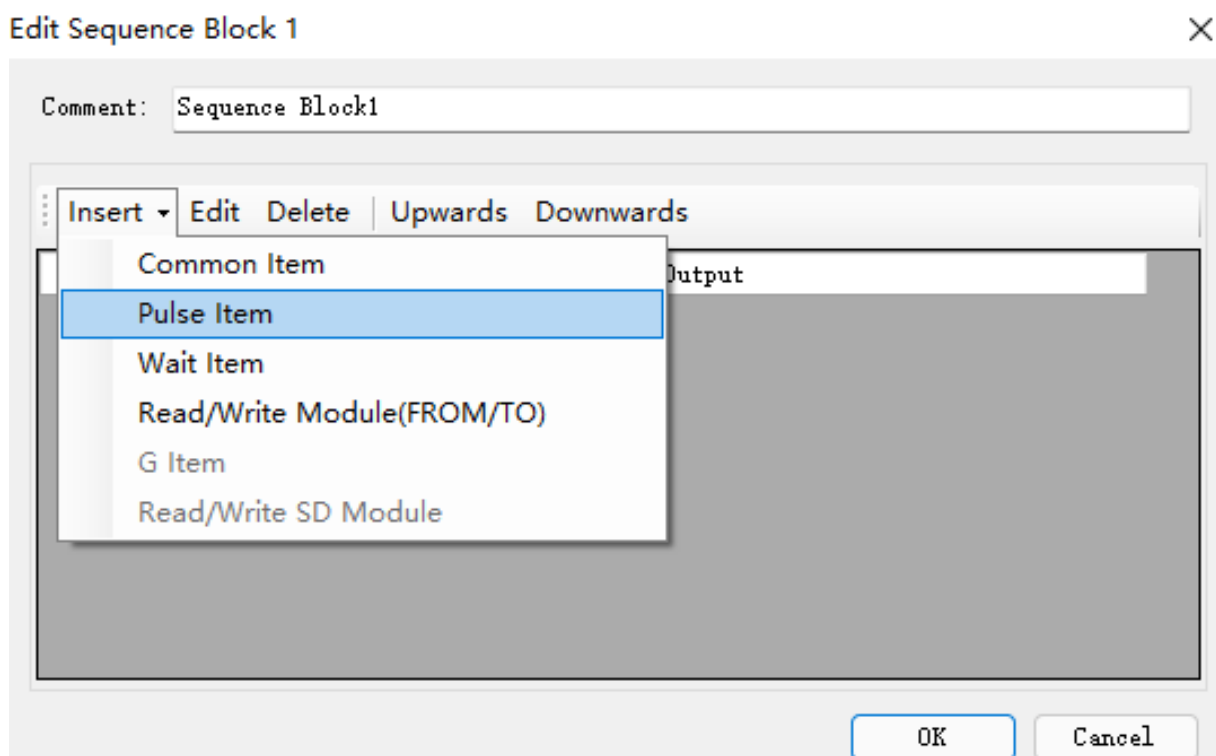


Выберите команду Add sequence block (Добавить блок последовательности), после чего откроется показанное ниже окно.

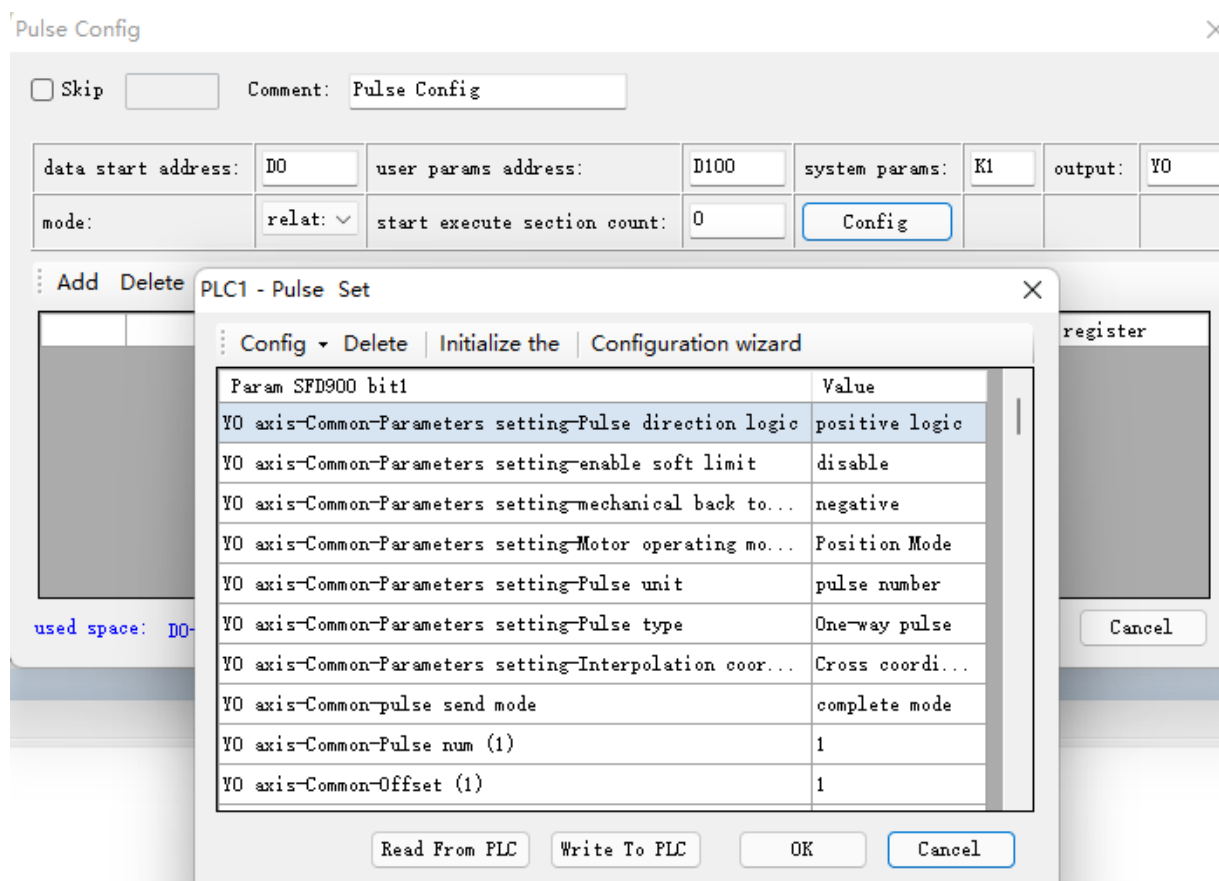


Можно отредактировать BLOCK в этом окне. Опция Upwards/Downwards (Вверх/вниз) используется для изменения положения команд в блоке.

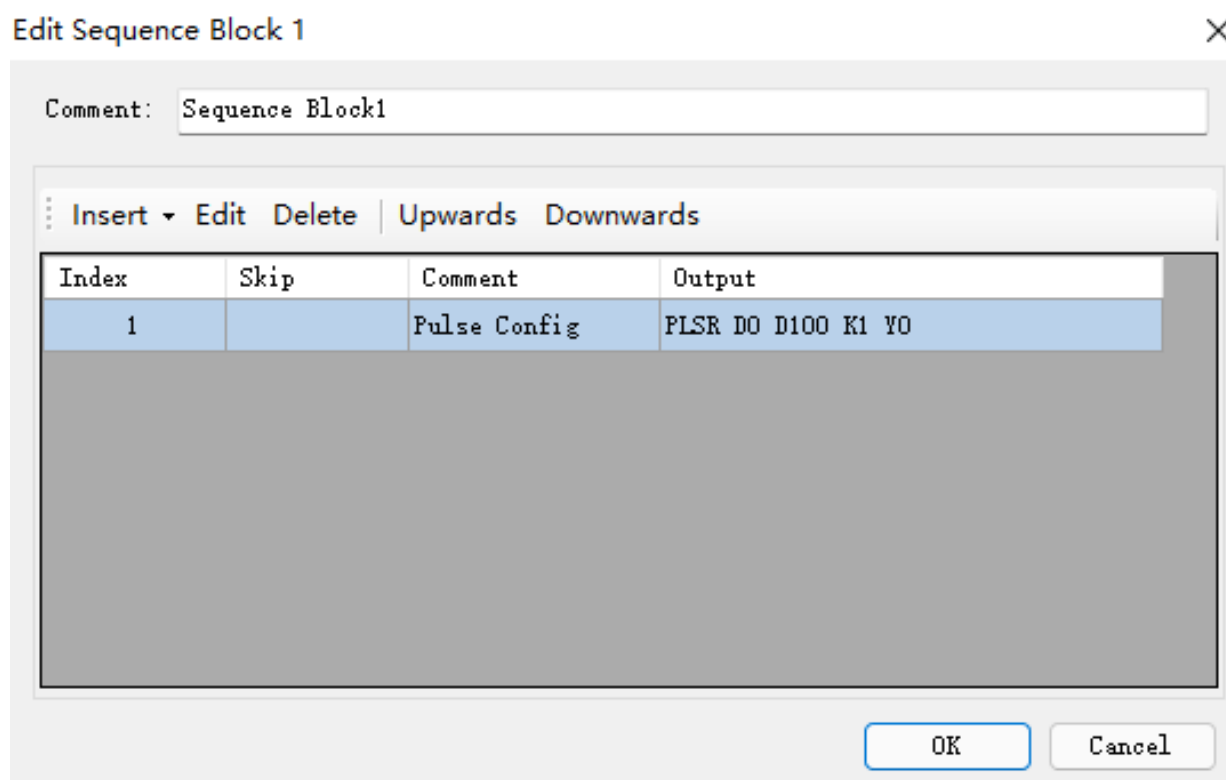
Нажмите кнопку 'Insert' (Вставить), после чего некоторые команды будут выведены в виде списка под меню:



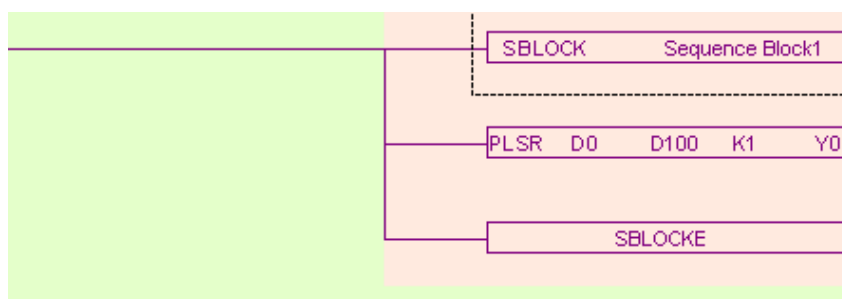
Рассмотрим 'Pulse Item' (Импульс) в качестве примера:



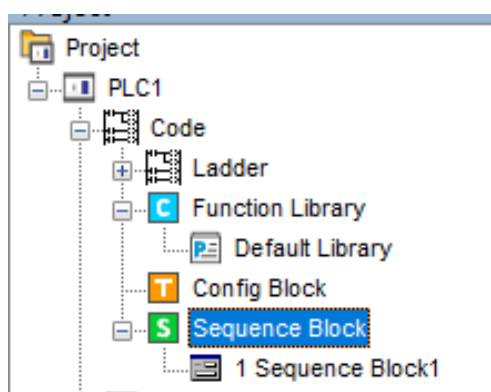
Нажав 'OK', можно вывести информацию о конфигурации:



Нажмите 'OK', после чего показанные ниже команды будут добавлены в лестничную диаграмму.

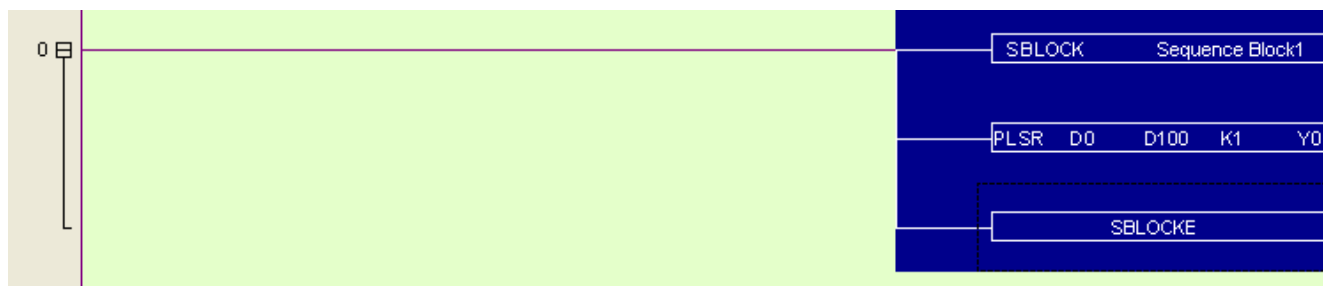


А в левой части панели проекта добавится новый блок последовательности:

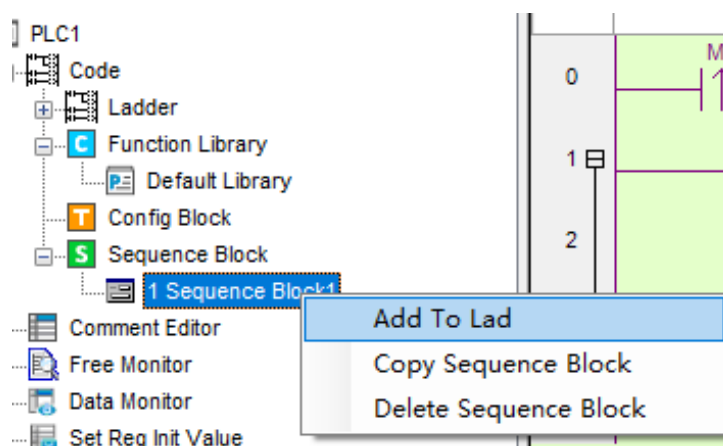


9.2.2 Перемещение BLOCK

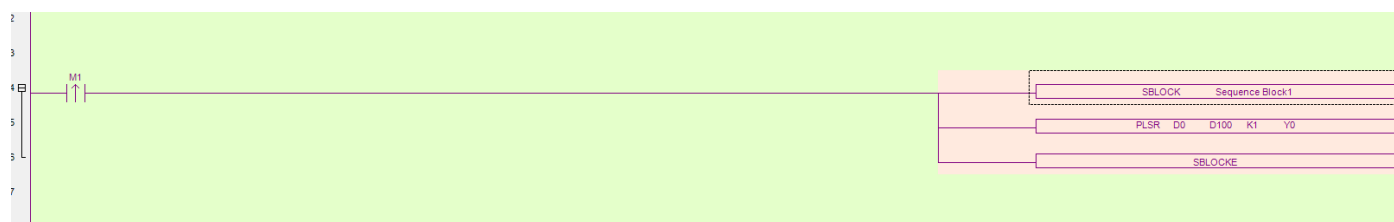
Чтобы переместить BLOCK в другое место, нужно выбрать исходный блок BLOCK и удалить его (выбрать select all (Выбрать все), а затем удалить):



Переместите курсор в новое место, затем щелкните правой кнопкой мыши по BLOCK и выберите 'Add to lad' (Добавить в лестничную диаграмму):

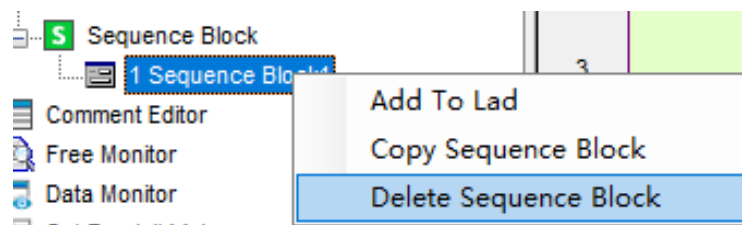


После этого блок BLOCK будет перемещен в новое место:



9.2.3 Удаление BLOCK

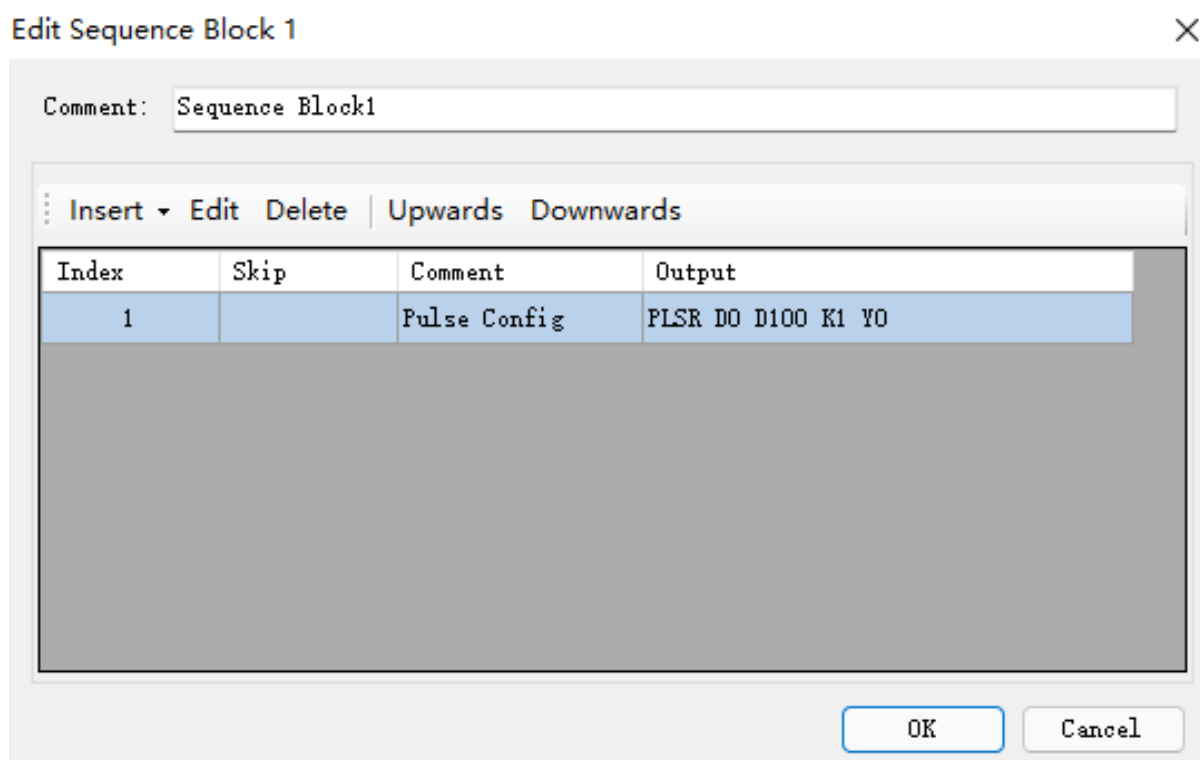
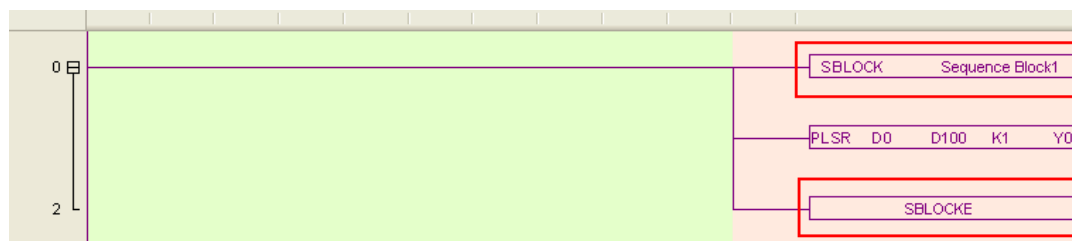
Можно выбрать вызванный BLOCK и удалить его. Чтобы полностью удалить BLOCK, щелкните правой кнопкой мыши по функциональному блоку и выберите 'Delete sequence block' (Удалить блок последовательности). После этой операции больше нельзя будет вызвать этот блок BLOCK:



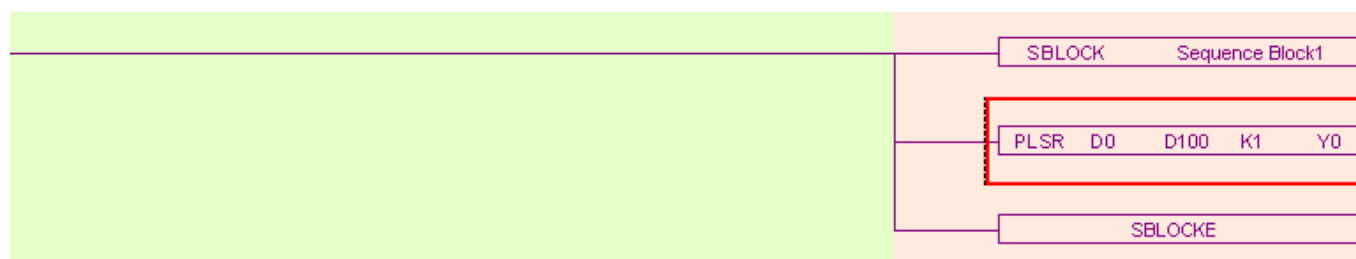
9.2.4 Изменение BLOCK

Существуют два способа изменения BLOCK.

(A) Дважды щелкните по начальному/конечному сегменту, чтобы изменить весь BLOCK:



(В) Дважды щелкните по средней части, чтобы ее изменить:



Pulse Config

☐ Skip Comment: Pulse Config

data start address:	D0	user params address:	D100	system params:	K1	output:	Y0
mode:	relat: ▾	start execute section count:	0	Config			

⋮ Add Delete Upwards Downwards

	frequence	pulse count	jump register

used space: D0-D9, D100-D103

Read From PLC Write To PLC OK Cancel

9.3 Редактирование команды в составе BLOCK

9.3.1 Command item (Команда)

Выберите 'Command item' (Команда), чтобы отредактировать программу:

Edit Sequence Block 1

Comment: Sequence Block1

⋮ Insert ▾ Edit Delete Upwards Downwards

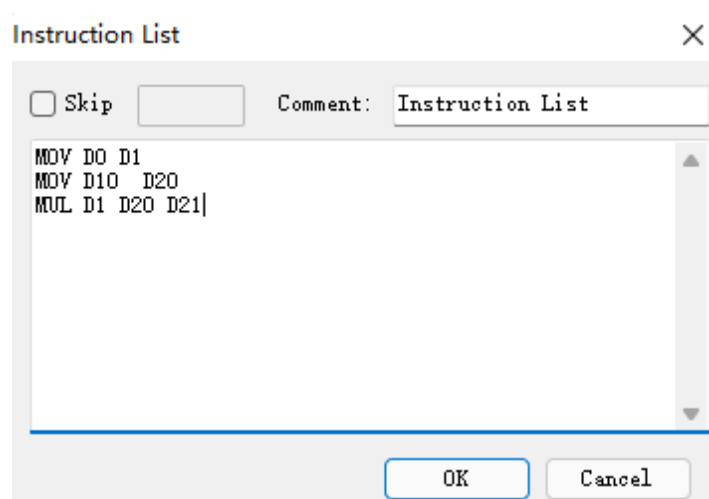
- Common Item
- Pulse Item
- Wait Item
- Read/Write Module(FROM/TO)
- G Item
- Read/Write SD Module

Output

LSR D0 D100 K1 Y0

OK Cancel

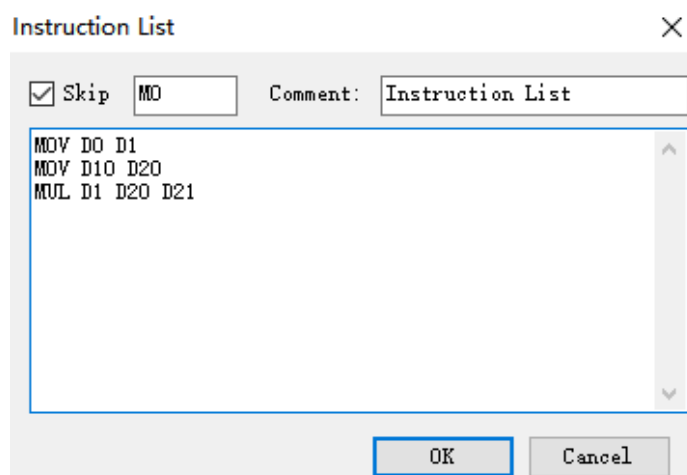
После выбора 'Command item' (Команда) откроется окно 'Instruction list' (Перечень команд):



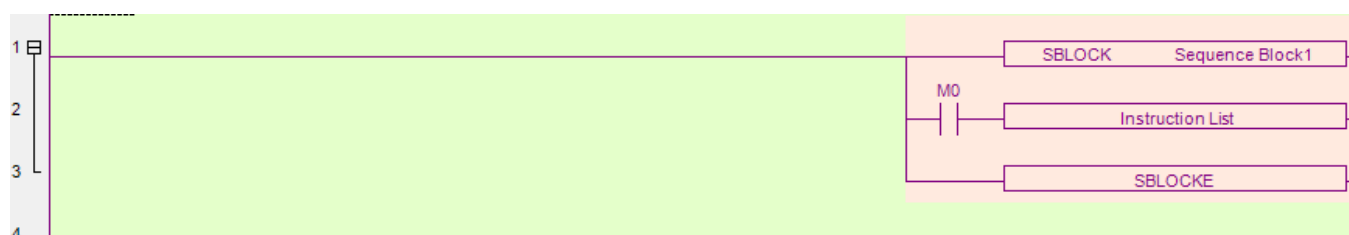
Пользователи могут добавлять команды в этом окне.

Skip (Пропустить): для управления остановкой и выполнением команд. Если в окне выбрана опция Skip (Пропустить), и введена катушка управления, то при включенной катушке управления команда не будет выполнена. Если эта опция не выбрана, то по умолчанию команда будет выполняться.

Comment (Комментарий): для изменения примечания к команде.

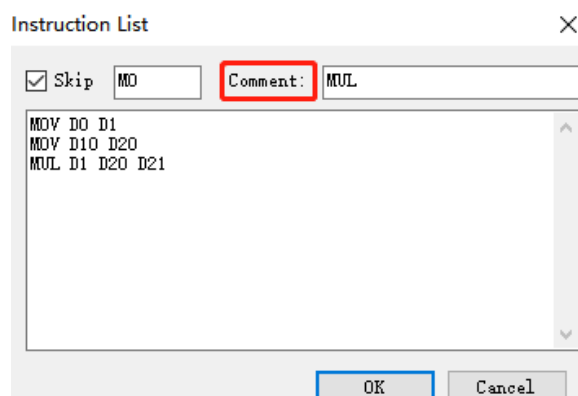


Нажмите 'OK', релейно-контактная программа примет следующий вид:



Примечание: можно добавить несколько команд в один BLOCK и использовать опцию Skip (Пропустить) в качестве условия выполнения каждой команды.

Как показано на рисунке выше, сегмент команды не развернут на лестничной диаграмме, но его описание можно изменить в соответствии с функцией сегмента, как показано на рисунке ниже:

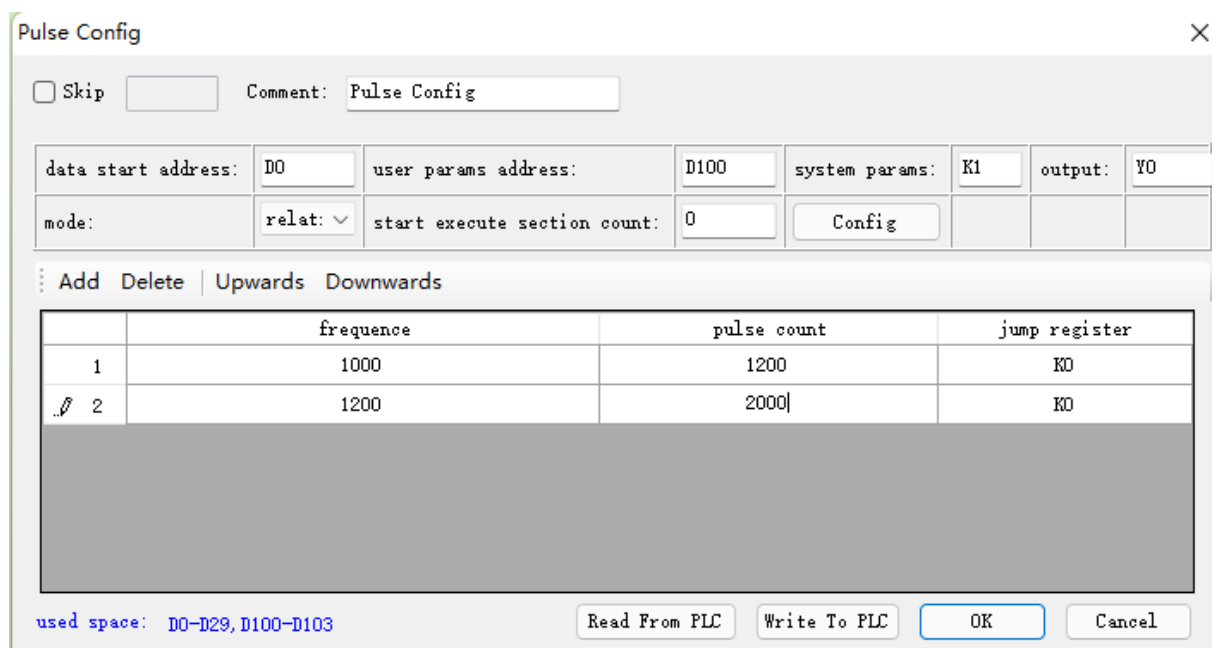


Измененная фраза блока также соответствующим образом изменилась:

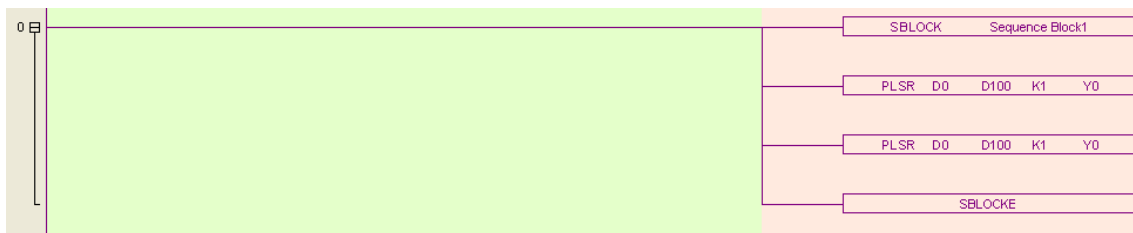


9.3.2 Pulse Item (Импульс)

Аналогичным способом откройте окно 'Pulse item' (Импульс):



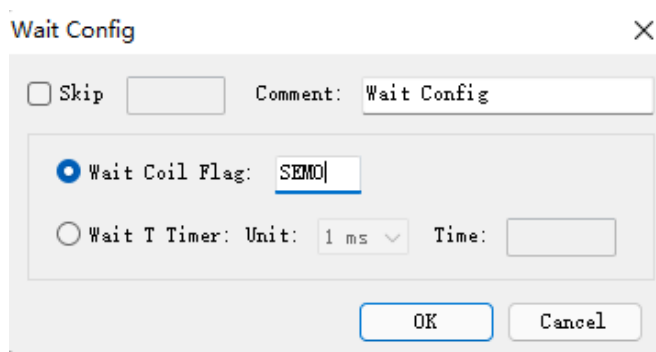
Добавим два командных импульса в показанный ниже BLOCK:



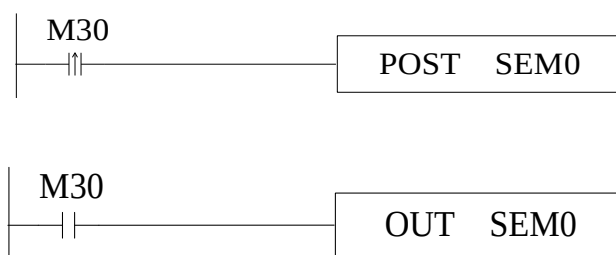
9.3.3 Wait Item (Ожидание)

'Wait Item' (Ожидание): ожидание признака катушки или бита таймера. Аналогичным способом откройте окно 'Wait Item' (Ожидание). Доступно два режима ожидания: бит признака и ожидание таймера.

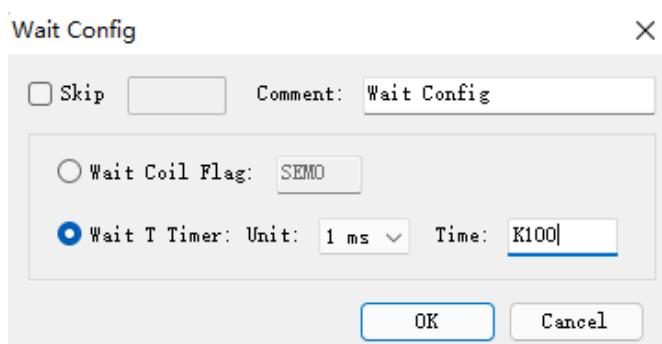
(A) Бит признака



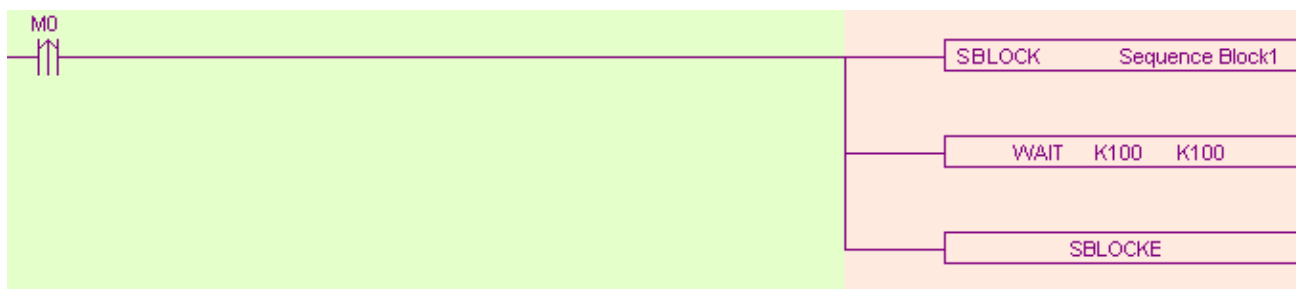
Соответствующая лестничная диаграмма SEM показана ниже:



(B) Ожидание таймера



(C) Соответствующая лестничная диаграмма:



Примечание:

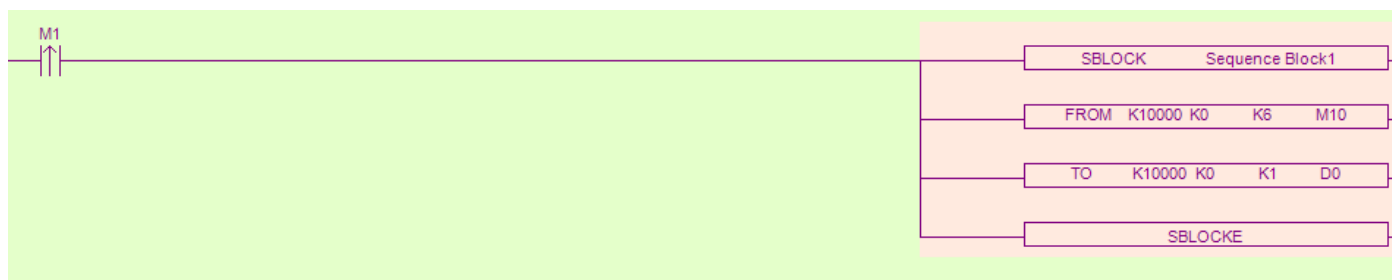
Не добавляйте стандартную катушку после команды WAIT в последовательности BLOCK для ПЛК серии PMP20, а добавьте бит SEM специального сигнала для ПЛК серии PMP20 (SEM0~SEM31); SEM нельзя контролировать путем установки или сброса. Его можно установить только командой POST и сбросить командой WAIT SEM. Либо можно вывести через команду OUT. Разница между ними заключается в том, что команда POST должна быть запущена фронтом импульса, чтобы сохранить состояние SEM; команда OUT должна быть запущена нормально разомкнутой катушкой, а SEM сбрасывается после отключения условия запуска.

9.3.4 Команда считывания/записи данных модуля (FROM/TO)

Этот элемент используется для считывания и записи данных между ПЛК и модулями, а панель управления выглядит следующим образом:

1#read

Команду FROM\TO можно выбрать из раскрывающегося списка:



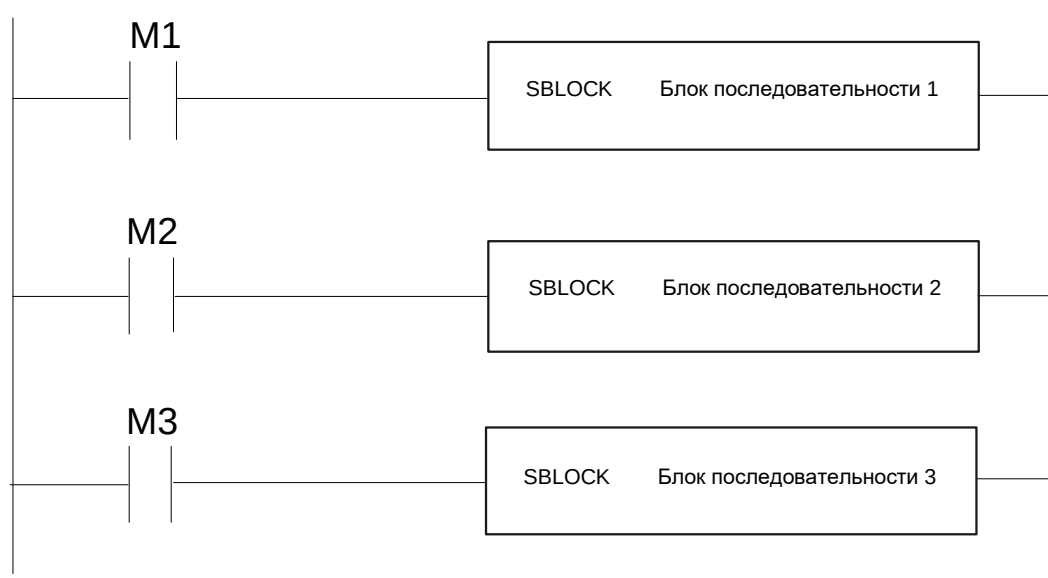
Примечание:

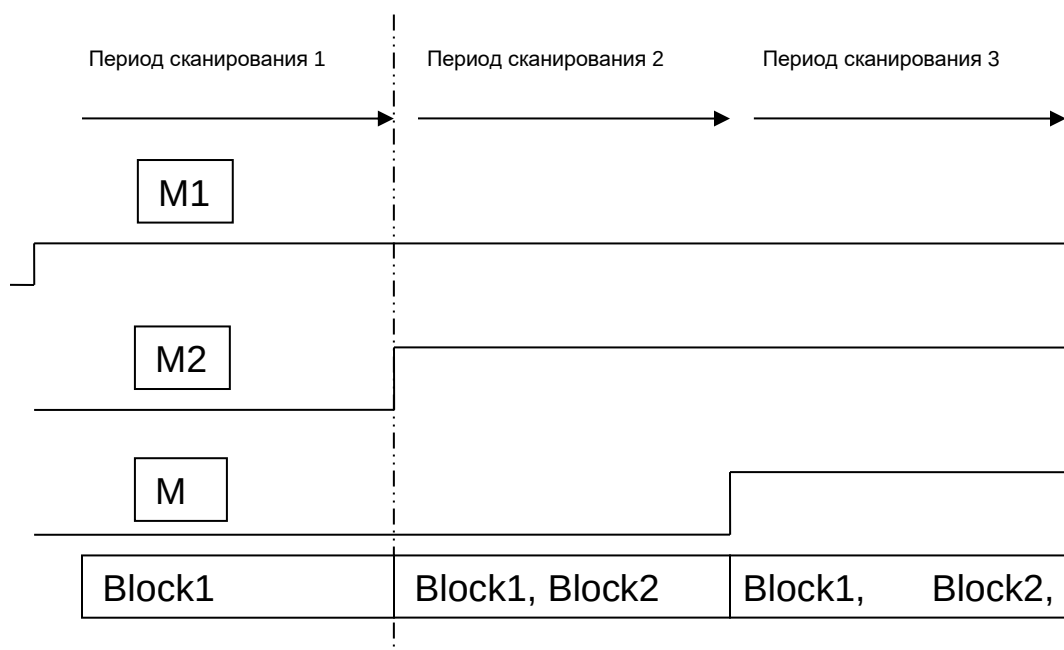
Как показано на рисунке выше, в программном обеспечении версии 3.4 и более поздних версий, когда номер модуля находится в диапазоне K0~K15, соответствующая лестничная диаграмма будет отображаться как K10000~K10015.

9.4 Схема запуска BLOCK

1. Если блоков много, они запускаются как обычная программа. Блок запускается, когда присутствует необходимое условие.

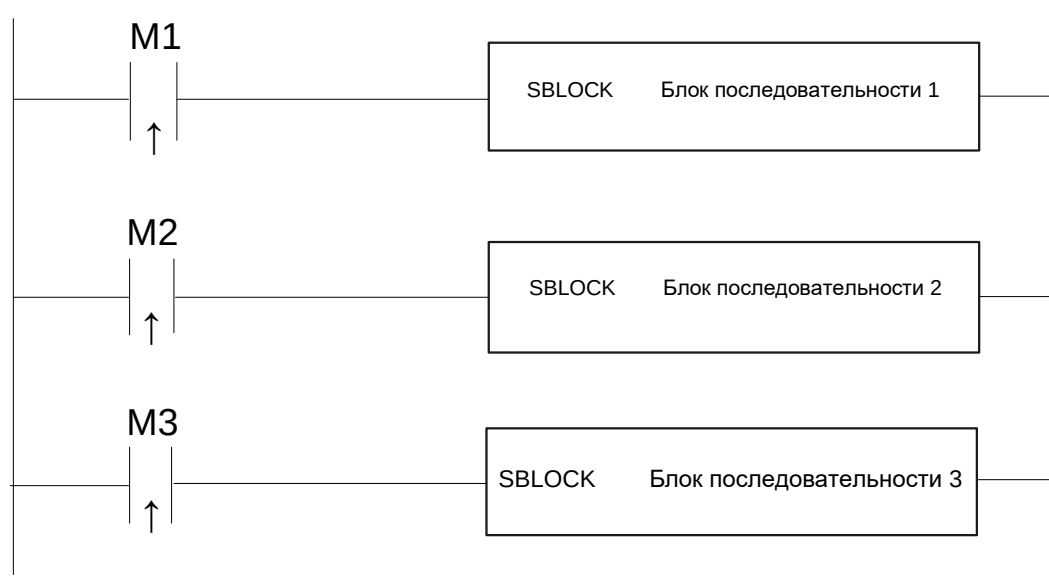
(A) Условием запуска является включенная/выключенная в нормальном режиме катушка.





Примечание: Если программа в BLOCK не выполнена, и условие запуска M исчезло, выполнение BLOCK не прекратится немедленно. Последнее сканирование завершится, и блок остановится после выполнения оставшейся части программы.

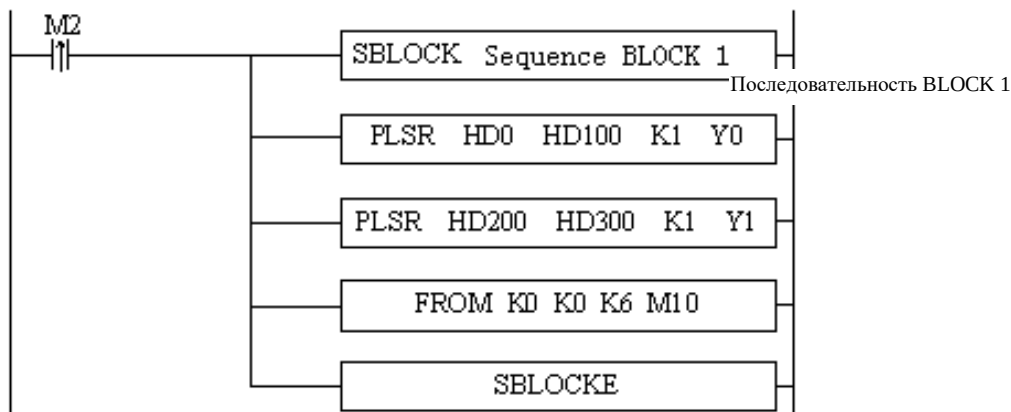
(B) Условием запуска является нарастающий или спадающий фронт импульса



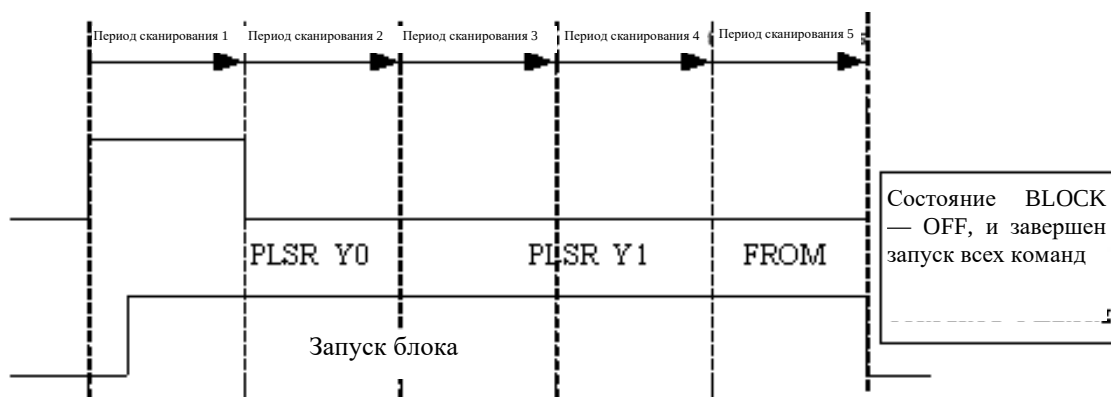
Если состояние M1, M2, M3 изменилось с OFF на ON, все эти блоки будут запущены одновременно.

2. Команды в блоке запускаются по очереди в зависимости от времени сканирования. Они запускаются одна за другой, когда состояние — ON.

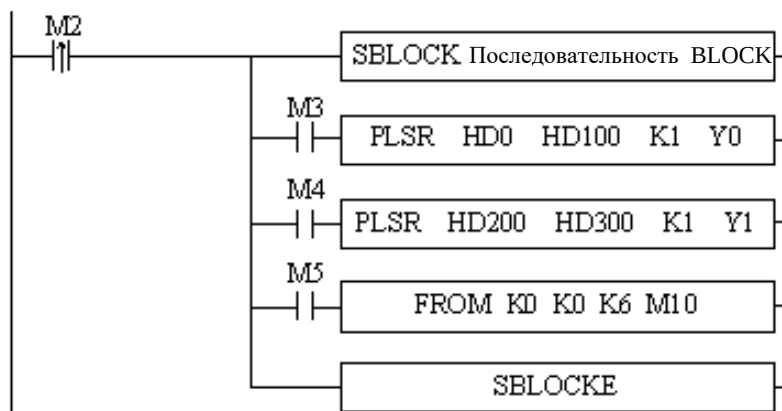
(A) Без условия SKIP (ПРОПУСК)



Ниже представлена последовательность запуска команд в блоке 1:



(В) С условием SKIP (ПРОПУСК)



Пояснение:

А) Если M2 включен, блок 1 запущен.

В) Все команды в блоке запускаются по очереди.

С) M3, M4, M5 — это признаки условия SKIP (ПРОПУСК). Когда они включены, эта команда не будет запускаться.

Д) Когда M3 выключен, и никакие другие команды не используют этот импульс Y0, будет запущен PLSR HD0 HD100 K1 Y0 ; в противном случае PLSR HD0

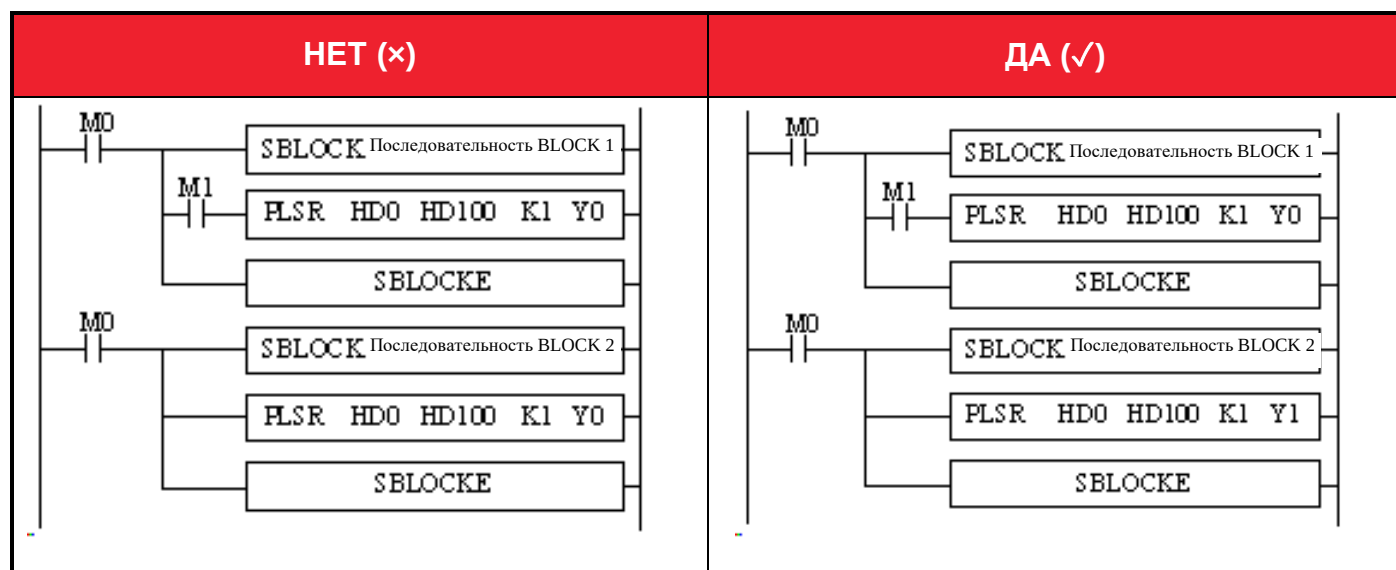
HD100 K1 Y0 будет запущен после того, как будет освобожден другими командами.

Е) После завершения отправки импульса Y0 проверьте M4. Если M4 выключен, проверьте блок Y1, если M4 включен, проверьте M5. Если M5 выключен, связь с модулем будет работать.

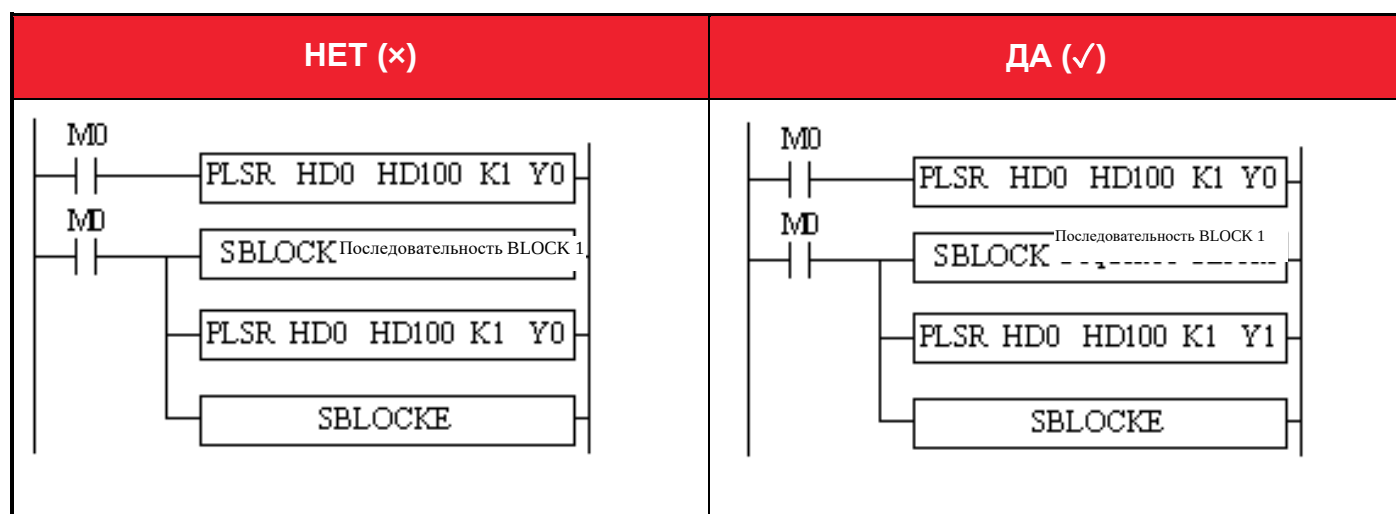
9.5 Правила редактирования команд BLOCK

В блоке BLOCK команды следует редактировать согласно определенным стандартам.

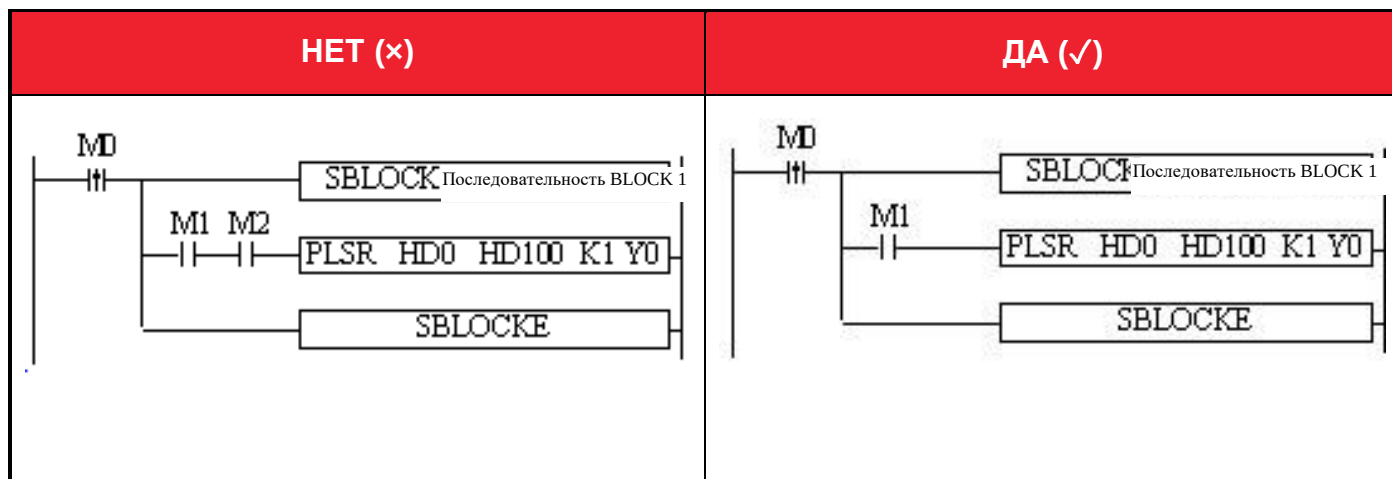
- (1) Не используйте ту же самую выходную импульсную клемму в другом BLOCK.



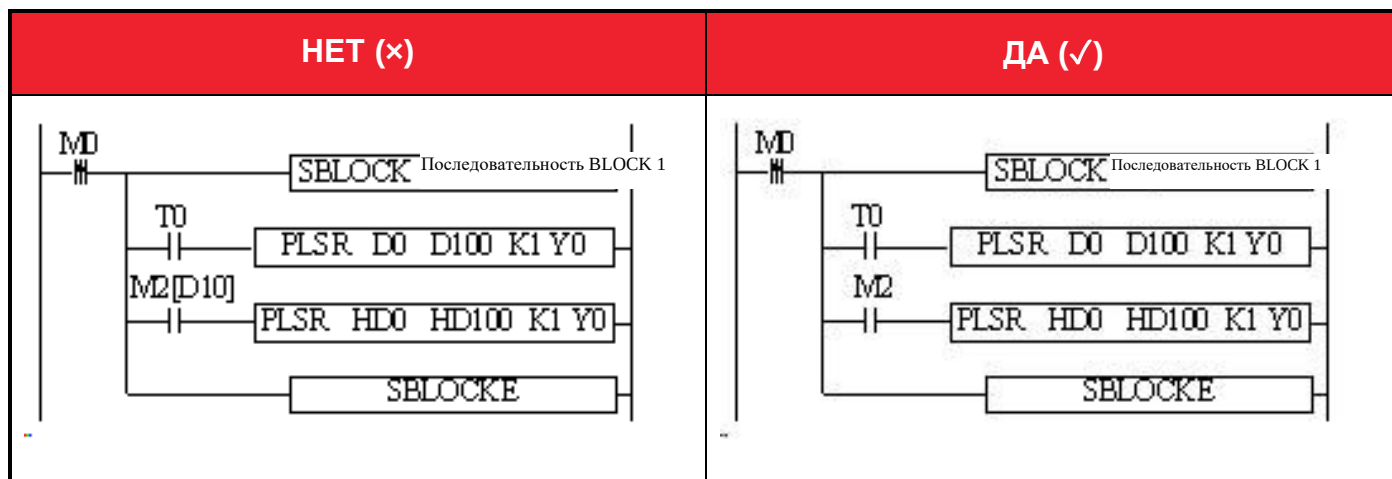
- (2) Не используйте ту же самую выходную импульсную клемму в BLOCK и главной программе.



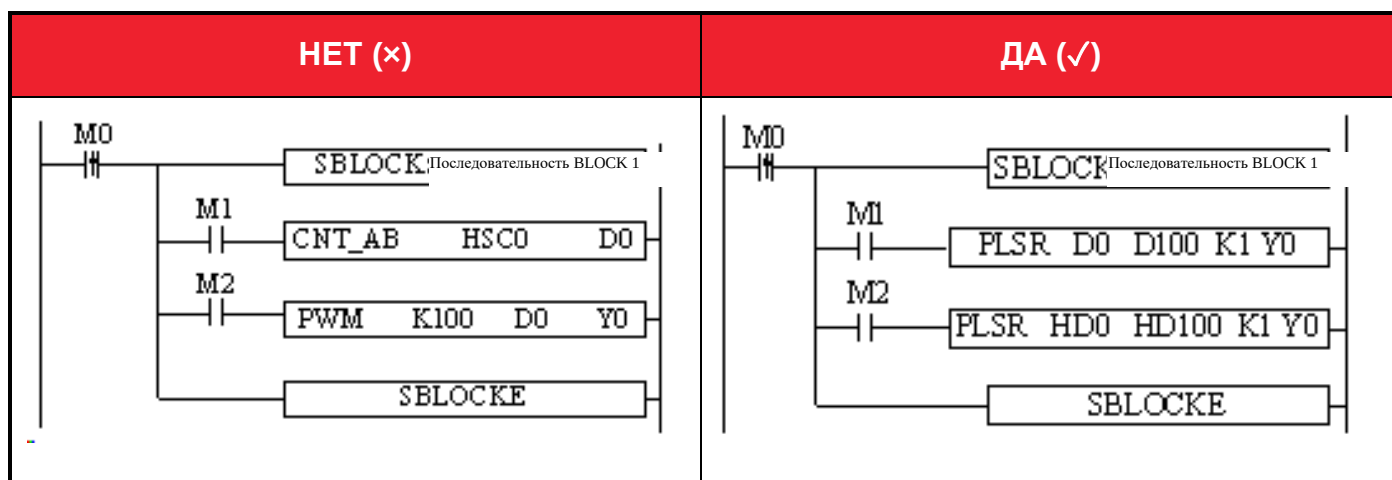
(3) Для каждой команды BLOCK может быть только одно условие SKIP.



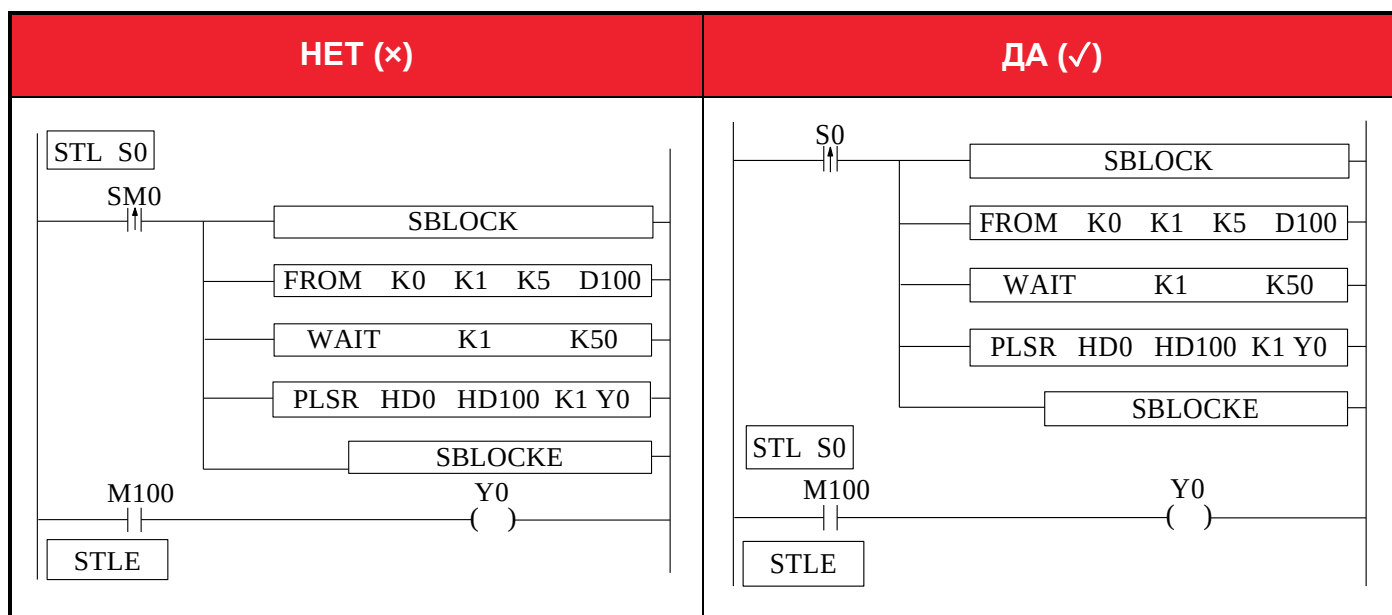
(4) Условие SKIP может использовать только M, X, но не может использовать другую катушку или регистр.



(5) Командами вывода не могут быть команды CNT_AB(CNT), PWM.



- (6) BLOCK не рекомендуется помещать в STL, так как если один STL закончится, а BLOCK не закончится, то возникнут большие проблемы.



- (7) Тип метки Kind не может быть использован в блоке.

Знак P, I не может быть использован в блоке. Их даже можно добавить в блок, но по факту они не работают.

9.6 Команды, связанные с BLOCK

9.6.1 Описание команд

Прекращение выполнения BLOCK [SBSTOP]

1) Обзор

Прекращение выполнения команд в блоке.

[SBSTOP]			
16 бит	SBSTOP	32 бита	-
Условие	НО, НЗ катушка и фронт импульса	Подходящие типы	PMP20
Аппаратное обеспечение		Программное обеспечение	V3.2

2) Операнды

Операнд	Функция	Тип
S1	Номер BLOCK	16 бит, BIN
S2	Режим остановки BLOCK	16 бит, BIN

3) Соответствующие компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден- тифи- катор	QD	X	Y	M	S	T	C	Dn.m
S1	•								•									
S2									•									

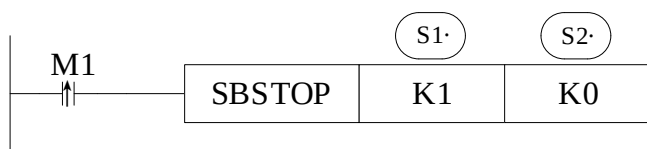
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

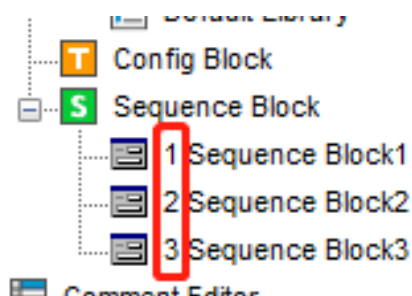
DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



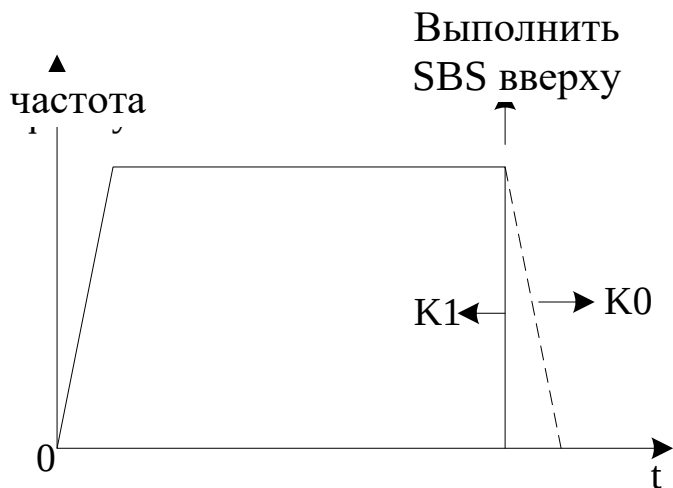
S1 — номер блока последовательности.. Номер блока уникален, и его нельзя изменять. Его можно посмотреть в левой панели проектирования.



S2 — режим остановки BLOCK, операнд: K0, K1, K2.

K0: медленная остановка BLOCK. Если выводится импульс, BLOCK остановится после завершения вывода импульса.

K1: немедленная остановка BLOCK; остановка выполнения всех команд в BLOCK.



K2: Деструктивный медленный останов BLOCK, то есть при отправке импульса выполняется условие SBSTOP, затем импульс замедляется по наклонной, без использования команды SBGOON, поэтому остальные команды не будут выполнены. После выполнения этой команды BLOCK можно перезапустить. (Примечание: Режим K2 поддерживается только прошивками ПЛК версий 3.4.2 и более поздних версий).

Продолжение выполнения BLOCK [SBGOON]

1) Обзор

Эта команда является противоположностью команды SBSTOP. Она необходима для продолжения выполнения BLOCK.

[SBGOON]			
16 бит	SBGOON	32 бита	-
Условие	Фронт импульса	Подходящие типы	PMP20
Аппаратное обеспечение	-	Программное обеспечение	V3.2

2) Операнды

Операнд	Функция	Тип
S1	Номер BLOCK	16 бит, BIN
S2	В этом режиме продолжается выполнение BLOCK	16 бит, BIN

3) Соответствующие компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон- станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден- тифи- катор	QD	X	Y	M	S	T	C	Dn.m
S1	•								•									
S2									•									

*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функция



S2 — режим, в котором продолжается выполнение BLOCK. Операнд: K0, K1.

K0: продолжение выполнения команд в BLOCK.

Например, если в прошлый раз вывод импульсов прекратился, SBGOON продолжит вывод оставшихся импульсов.

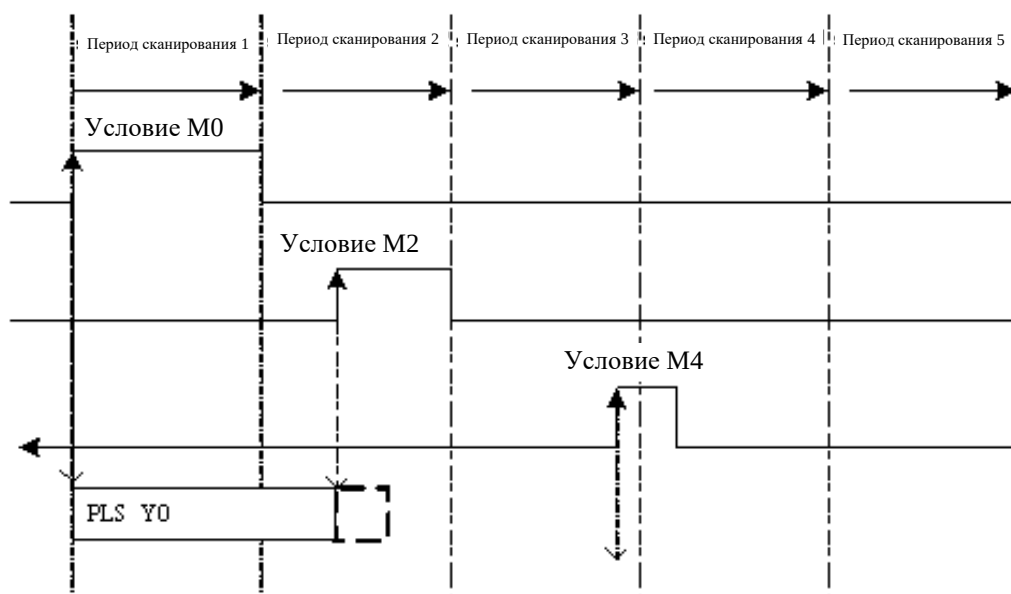
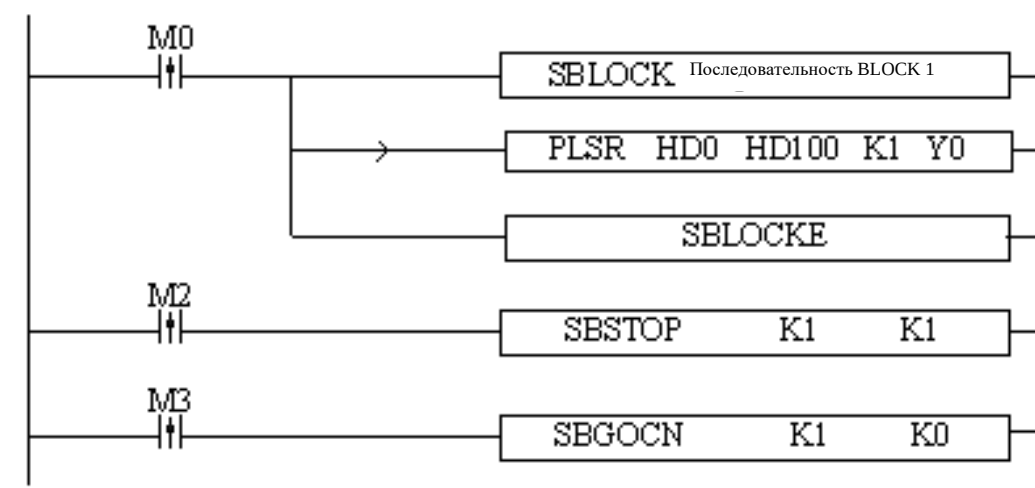
K1: продолжить выполнение BLOCK, но отказаться от команд, не выполненных в прошлый раз.

Например, если в прошлый раз импульс не завершился, SBGOON не продолжит выдачу этого импульса, а перейдет к следующей команды в BLOCK.

Эта команда применяется только к командам PLSR в BLOCK и может отправлять только оставшиеся импульсы для команд интерполяции, которые нельзя пропустить.

9.6.2 Временная последовательность команд

SBSTOP(K1 K1) + SBGOON(K1 K1)

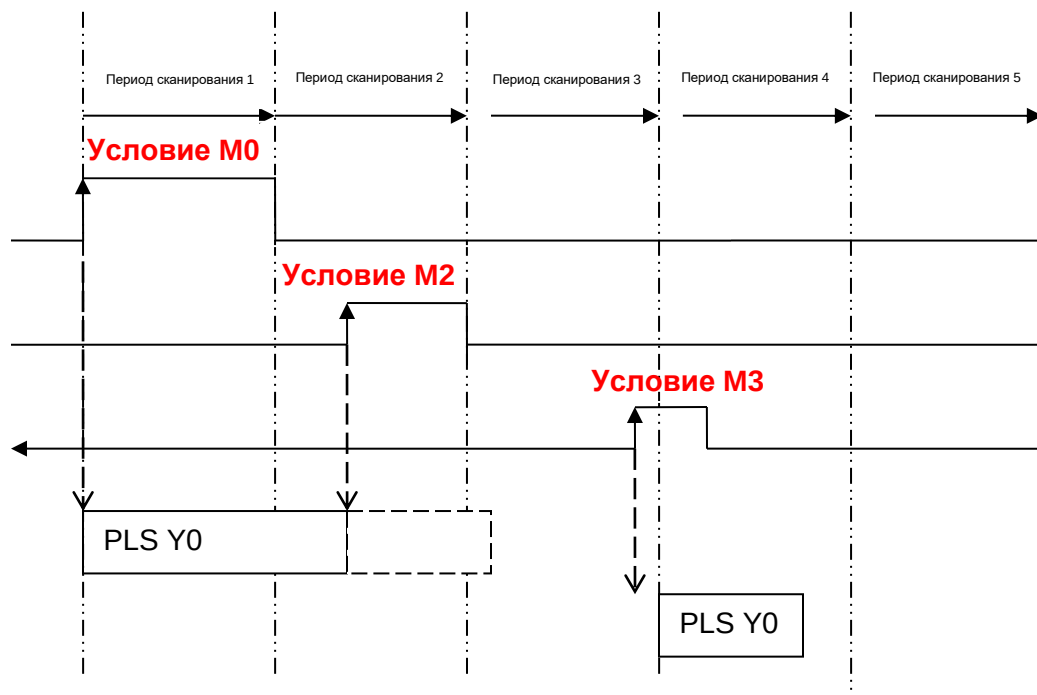
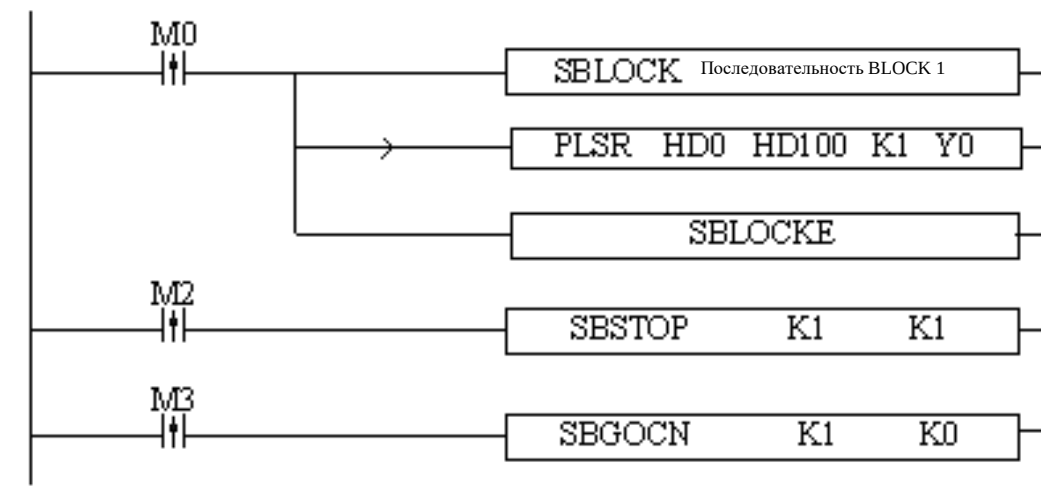


Если M0 меняется с OFF на ON, запустите “PLSR HD0 HD100 K1 Y0” в BLOCK для вывода импульсов.

Если M2 меняется с OFF на ON, выполнение BLOCK сразу прекратится.

Если M4 меняется с OFF на ON, отказ вывода остальных импульсов.

SBSTOP(K1 K1) + SBGOON(K1 K0)

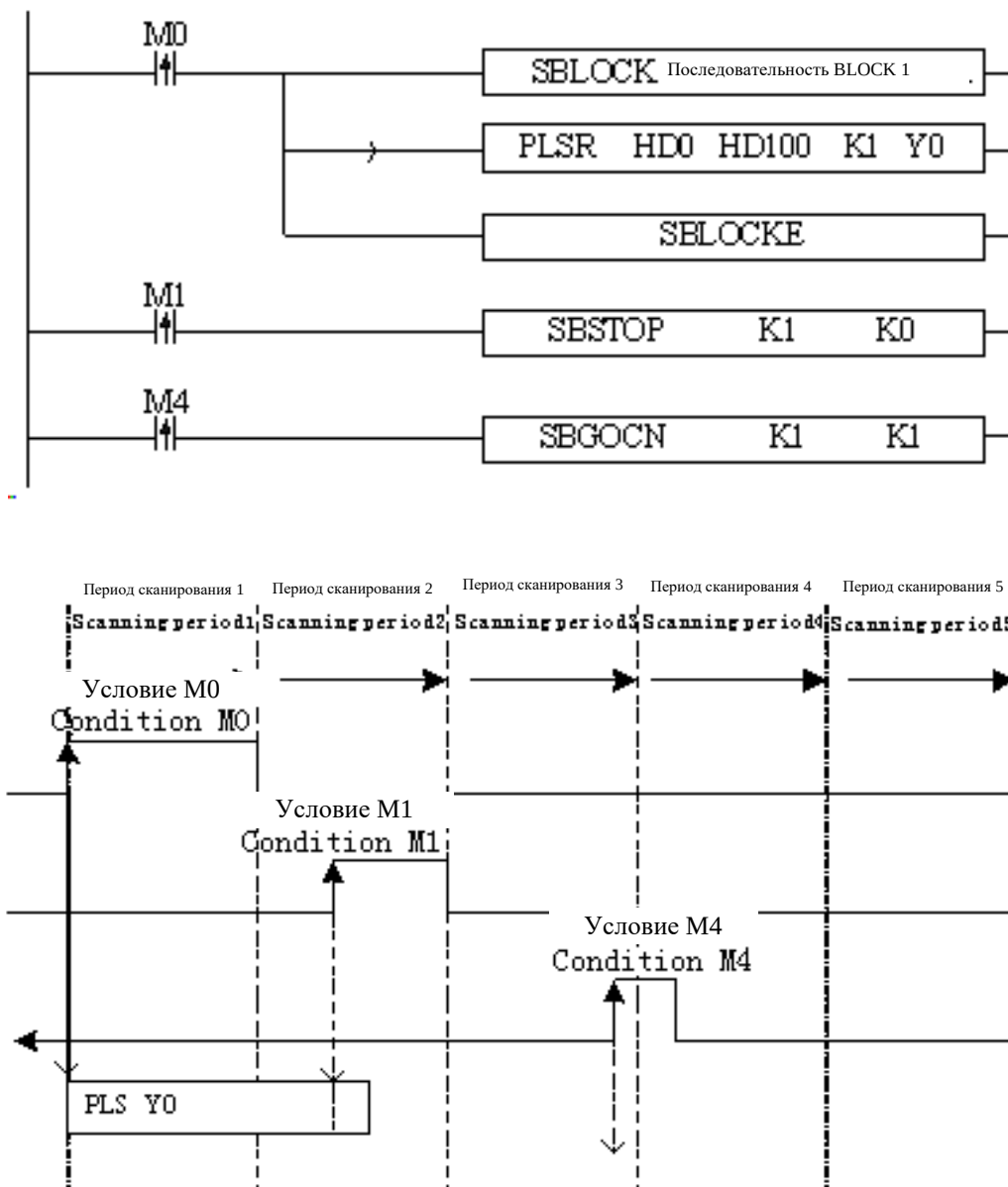


Если M0 меняется с OFF на ON, запустите ‘PLSR HD0 HD100 K1 Y0’ в BLOCK для вывода импульсов.

Если M2 меняется с OFF на ON, выполнение BLOCK прекратится, вывод импульсов сразу прекратится.

Если M3 меняется с OFF на ON, произойдет вывод остальных импульсов.

SBSTOP(K1 K0) + SBGOON(K1 K1)

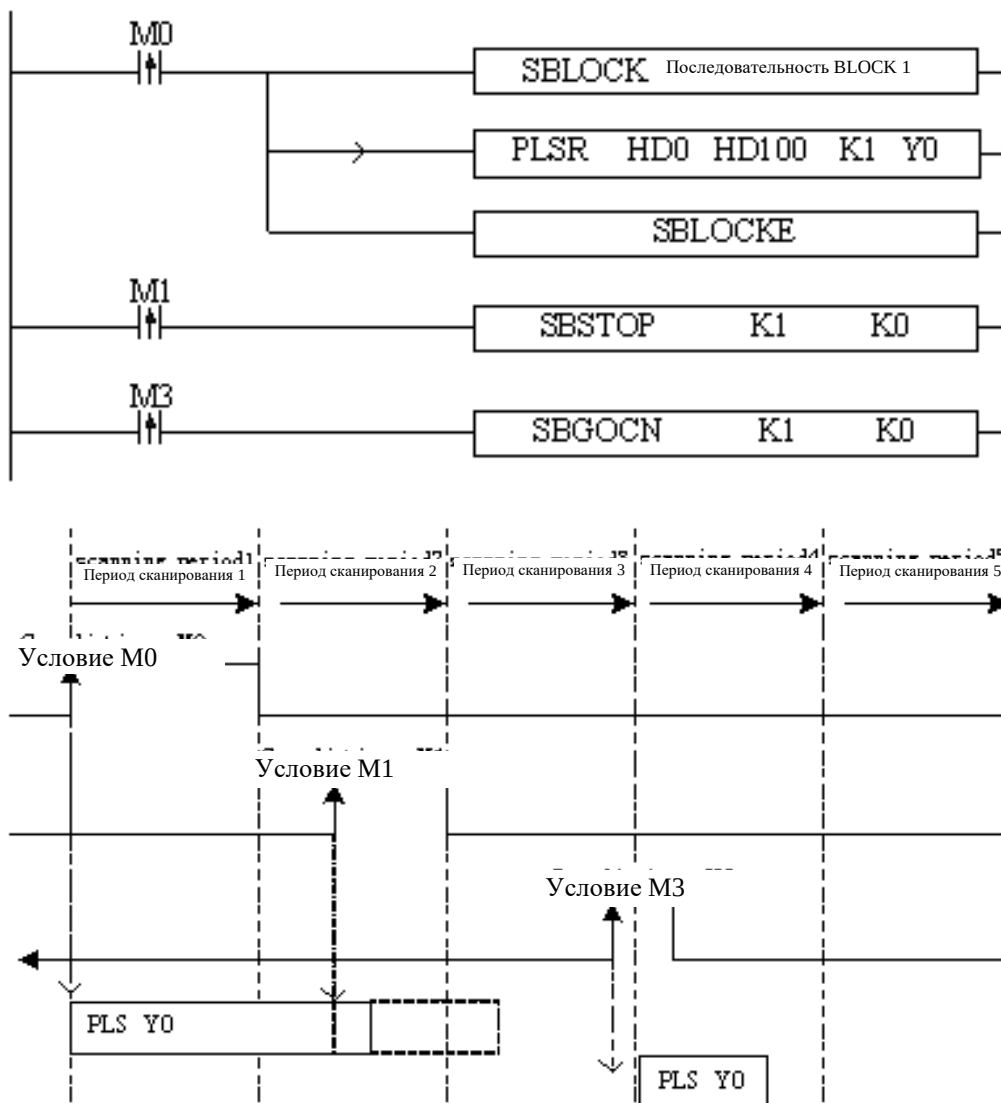


Если M0 меняется с OFF на ON, запустите 'PLSR HD0 HD100 K1 Y0' в BLOCK для вывода импульсов.

Если M1 меняется с OFF на ON, выполнение BLOCK прекратится, импульс будет медленно останавливаться с наклоном.

Если M4 меняется с OFF на ON, отказ вывода остальных импульсов.

SBSTOP(K1 K0) + SBGOON(K1 K0)



Если M0 меняется с OFF на ON, запустите 'PLSR HD0 HD100 K1 Y0' в BLOCK для вывода импульсов.

Если M1 меняется с OFF на ON, выполнение BLOCK приостановится, импульс будет медленно останавливаться с наклоном.

Если M3 меняется с OFF на ON, произойдет вывод остальных импульсов.

Обратите внимание, что после остановки SBSTOP импульса с наклоном могут остаться некоторые импульсы; в этом случае, если снова запустить SBGOON K1 K0, то будут выведены остальные импульсы.

9.7 Бит признака и регистр BLOCK

1. Бит признака BLOCK:

Адрес	Функция	Описание
SM300	Признак выполнения BLOCK1	1: выполняется 0: не выполняется
SM301	Признак выполнения BLOCK2	
SM302	Признак выполнения BLOCK3	
.....		
.....		
SM399	Признак выполнения BLOCK100	

2. Регистр признака BLOCK:

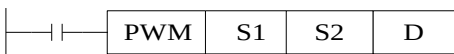
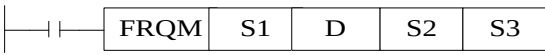
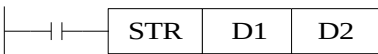
Адрес	Функция	Описание
SD300	Команда выполнения BLOCK1	BLOCK будет использовать это значение во время контроля
SD301	Команда выполнения BLOCK2	
SD302	Команда выполнения BLOCK3	
.....		
.....		
SD399	Команда выполнения BLOCK100	

Если используется GBLOCK, он будет занимать SM399 и SD399.

10. Команды специальных функций

В этой главе рассматриваются PWM (шиотно-импульсная модуляция), FRQM, точная синхронизация, прерывания и т. д.

Перечень команд специальных функций:

Мнемонический код	Функция	Схемные и программные компоненты	Глава
Широтно-импульсная модуляция, частотное детектирование			
PWM	Выходной импульс с заданной скважностью и частотой		10-1
FRQM	Измерение частоты фиксированных импульсов		10-2
Время			
STR	Точное время		10-3
Прерывание			
EI	Разрешить прерывание		10-4-1
DI	Запретить прерывание		10-4-1
IRET	Возврат прерывания		10-4-1

10.1 Широтно-импульсная модуляция [PWM]

1) Обзор

Команда для выполнения широтно-импульсной модуляции (ШИМ)

Широтно-импульсная модуляция (ШИМ) [PWM]			
16-битная команда	-	32-битная команда	PWM
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Ввод значения скважности или идентификационного номера программного компонента	32 бита, BIN
S2	Ввод значения выходной частоты или идентификационного номера программного компонента	32 бита, BIN
D	Ввод порта выходных импульсов	бит

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m
S1	•	•	•	•					•									
S2	•	•	•	•					•									
D													•					

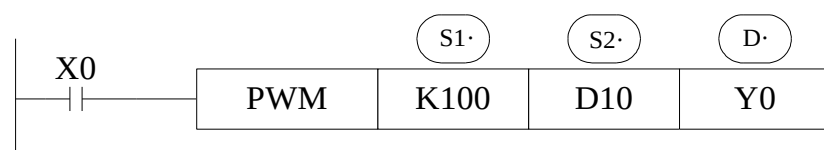
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



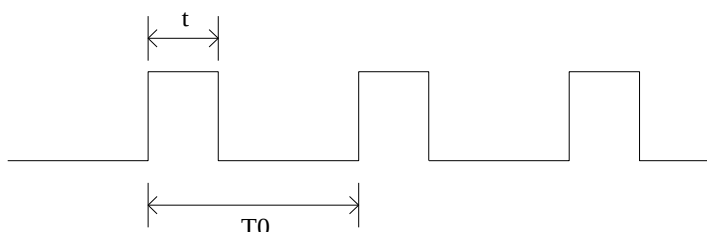
Скважность n: 1-65535

Выходной импульс f : 1-100 кГц

Для выхода ШИМ ПЛК серии RMP20 потребуется клемма транзисторного типа:

Модель ПЛК	Клемма ШИМ
RMP20-30T -60T	Y0, Y1
RMP20-30T4 -60T4 -60T6 -60T10	Y0, Y1, Y2, Y3

- Скважность для выхода **ШИМ** = $n/65535 \times 100\%$.
- Для ШИМ единицей измерения является 0,1 Гц, поэтому при настройке частоты S2 заданное значение в 10 раз превышает фактическую частоту ($10f$).
- Например: чтобы задать частоту 72 кГц, установите значение 720000 в S2.
- Когда X0 включен, выводится ШИМ-волна; когда X0 выключен, вывод прекращается. При выводе ШИМ-волны импульсы не накапливаются.



На левом графике:

$$T0 = 1/f$$

$$t/T0 = n/65535$$

Примечание: При использовании команды PWM необходимо подключить усиленный резистор сопротивлением 1 кОм между выходной клеммой и общей клеммой.

Пример



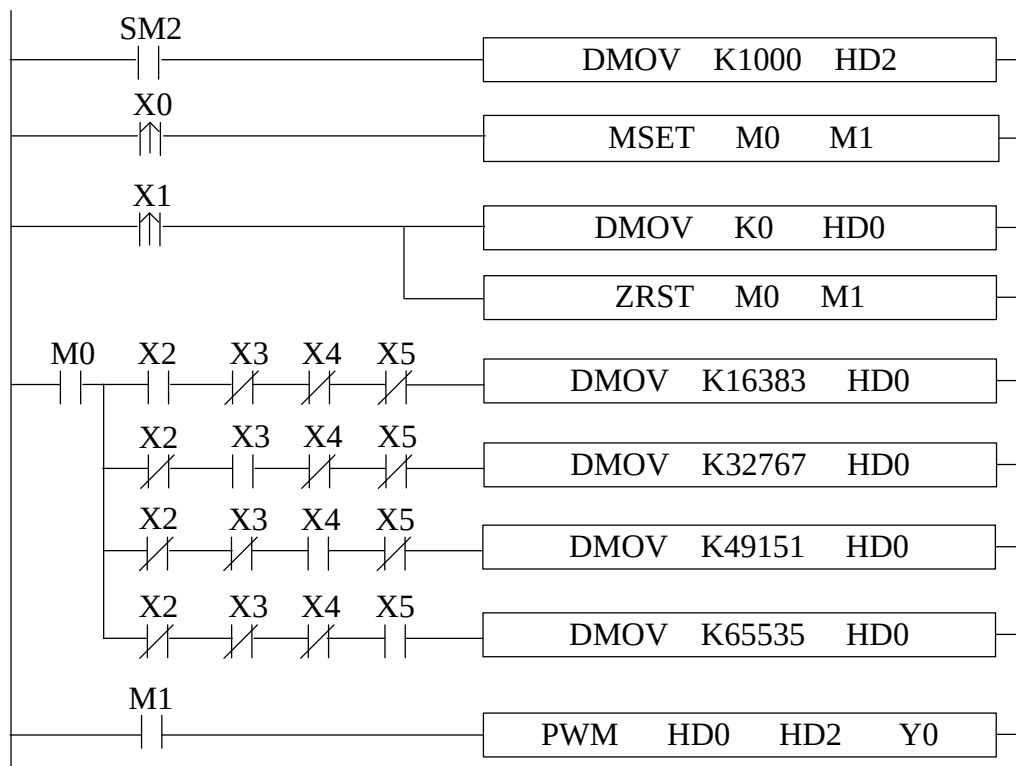
Есть светодиод, работающий от источника постоянного тока 24 В. Необходимо отрегулировать яркость светодиода. Чтобы уменьшить потери мощности волнового коллектора, включите выключатель в тот момент, когда он выключен, а затем снова выключите его. Этот процесс повторится. Подключите транзистор между источником питания и светодиодом. Импульсный сигнал будет поступать с клеммы базы транзистора. Ток между базой и излучателем импульсный. Входное напряжение светодиода пропорционально коэффициенту заполнения. Входное напряжение светодиода меняется при изменении коэффициента заполнения. Существует множество способов изменить его значение. Стандартный способ — широтно-импульсная модуляция (ШИМ), которая предполагает только изменение времени сохранения включенного состояния, но не изменение частоты включения.

В этом примере рассмотрено применение технологии ШИМ для регулировки яркости свечения светодиода. Контроллер может принимать управляющий сигнал ШИМ напряжением 24 В. Диапазон яркости: 25%, 50%, 75%, 100%. Яркость регулируется в зависимости от коэффициента заполнения ШИМ.

Описание компонентов:

Компонент ПЛК	Описание	Метка
X0	Кнопка Start (Пуск), при нажатии включается X0	
X1	Кнопка Stop (Стоп), при нажатии включается X1	
X2	Кнопка 25% brightness (яркость 25%), при нажатии включается X2	
X3	Кнопка 50% brightness (яркость 50%), при нажатии включается X3	
X4	Кнопка 75% brightness (яркость 75%), при нажатии включается X4	
X5	Кнопка 100% brightness (яркость 100%), при нажатии включается X5	
HD0	Регистр коэффициента заполнения ШИМ	
HD2	Регистр частоты ШИМ	По умолчанию: 100 Гц

Программа:



Описание программы:

1. HD0 будет управлять напряжением светодиода. Напряжение = $24 \cdot \text{HD0} / 65535$, выходная частота импульсов 100 Гц.
2. Нажмите кнопку Start (Пуск), X0 включится, M0, M1 выключатся, и начнется регулировка яркости светодиода.
3. X2 включен ON, HD0 = 16383, $\text{HD0} / 32768 = 0,25$, яркость светодиода — 25%.
4. X3 включен ON, HD0 = 32767, $\text{HD0} / 32768 = 0,5$, яркость светодиода — 50%.
5. X4 включен ON, HD0 = 49151, $\text{HD0} / 32768 = 0,75$, яркость светодиода — 75%.
6. X5 включен ON, HD0 = 65535, $\text{HD0} / 32768 = 1$, яркость светодиода — 100%.
7. Нажмите кнопку выключения, X1 включится, HD0 будет сброшен, отключите состояние триггера ШИМ, напряжение светодиода составит 0 В.

10.2 Измерение частоты [FRQM]

1) Обзор

Измерение частоты.

Измерение частоты [FRQM]			
16-битная команда	-	32-битная команда	FRQM
Условие выполнения	ВКЛ/ВЫКЛ катушки в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
S1	Количество импульсов выборки	16 бит, BIN
S2	Точность отображения	16 бит, BIN
D	Результаты измерений	32 бита, BIN
S3	Входная клемма импульсов	бит

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные элементы бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m	
S1	•	•	•	•					•										
S2	•	•	•	•					•										
D													•						

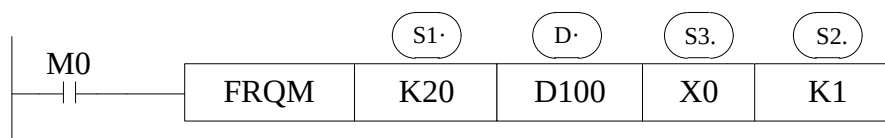
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- Количество импульсов выборки можно регулировать в зависимости от частоты: чем выше частота, тем больше будет импульсов выборки.
- Результаты измерений, единица измерения — Гц.
- Разрешение дисплея: 1, 10, 100, 1000, 10000.
- Когда M0 включен, FRQM получает 20 импульсов с X0 и фиксирует период выборки. Результат деления числа выборок на период выборки будет сохранен в D100. Процесс измерений повторится. Если частота измерения меньше диапазона измерения, результат равен 0.
- Точность измерения составляет 0,001%.

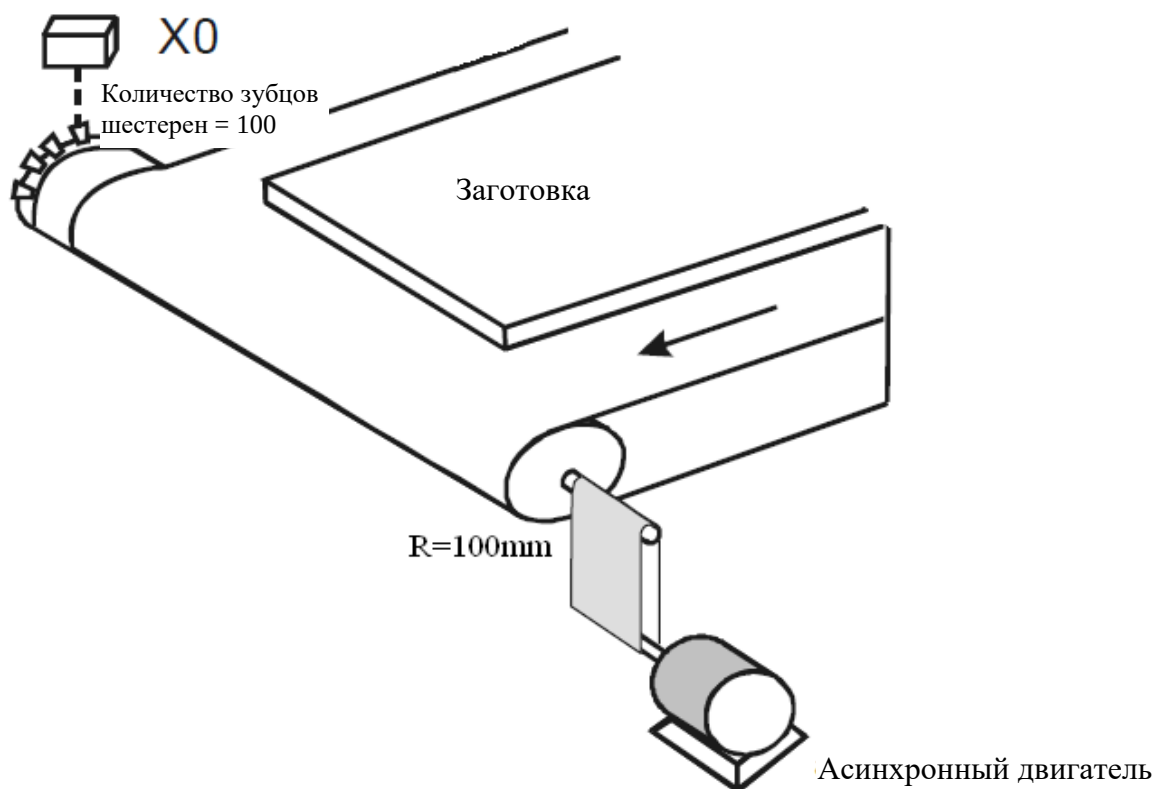
Входная клемма импульсов для FRQM:

Модель		Клемма X	Макс. частота (Гц)
Серия PMP20	24/30/60 I/O	X0, X3	80K
		X6	10K
	30T4/60T4/60T6/60T10	X0, X3, X6, X11	80K

Пример

Асинхронный двигатель приводит в действие конвейер, перемещающий заготовку. Необходимо отображать скорость перемещения заготовки в режиме реального времени. Диаметр вала трансмиссии — 100 мм, количество зубцов шестерен на валу трансмиссии — 100, единица измерения скорости — м/мин.

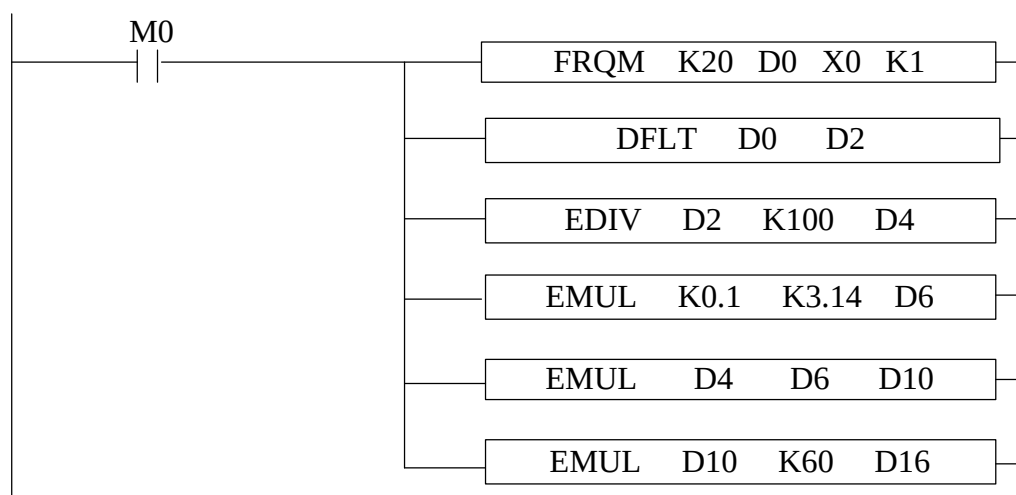
Бесконтактный выключатель



Описание компонентов:

Компонент ПЛК	Управляющие компоненты	Метка
X0	Бесконтактный выключатель для подсчета количества зубцов шестерен	
M0	Запускающий сигнал	
D16	Регистр скорости (число с плавающей точкой)	

Программа:



Описание программы:

1. Включите запускающий сигнал M0, чтобы запустить программу измерения частоты.
2. Преобразуем частоту в число с плавающей точкой, затем разделим ее на 100 (число зубьев за один оборот), получим число оборотов вала в секунду (число с плавающей точкой).
3. Рассчитаем диаметр вала трансмиссии и сохраним это значение в регистре D6 (число с плавающей точкой), затем рассчитаем расстояние передачи в секунду и сохраним его в регистре D10 (число с плавающей точкой).
4. Расстояние передачи в секунду, умноженное на 60, дает скорость (м/мин).

10.2.1 Точная синхронизация [STR]

1) Обзор

Считывание данных и прекращение точной синхронизации после выполнения команды на точную синхронизацию.

Точная синхронизация [STR]			
16-битная команда	-	32-битная команда	STR
Условие выполнения	Запуск по фронту импульса	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Тип
D1	Номер таймера	бит
D2	Ввод значения таймера или идентификационного номера программного компонента	32 бита, BIN

3) Соответствующие программные компоненты

Операнды	Программные элементы слова											Программные элементы бита						
	Система								Кон-станта	Модуль		Система						
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m
D																•		
D1																•		
D2	•	•	•	•					•									

*Примечание:

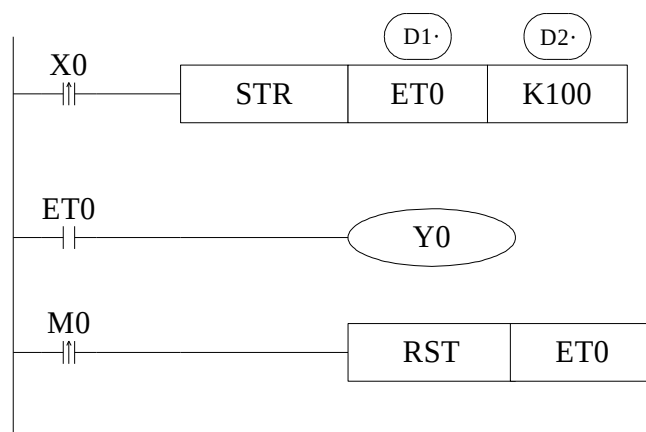
D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции

<Точная синхронизация>, <Сброс точной синхронизации>



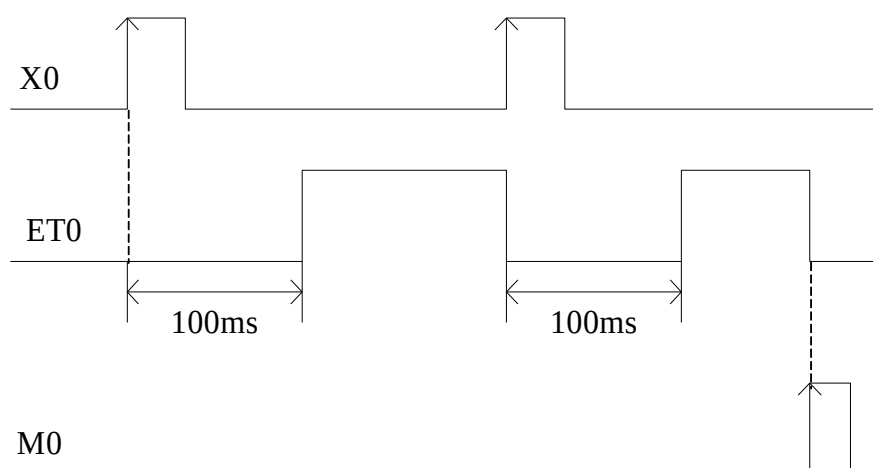
D1·: Номер таймера.

Диапазон: ET0~ET30 (ET0, ET2, ET4.....все номера должны быть четными)

D2·: Значение синхронизации

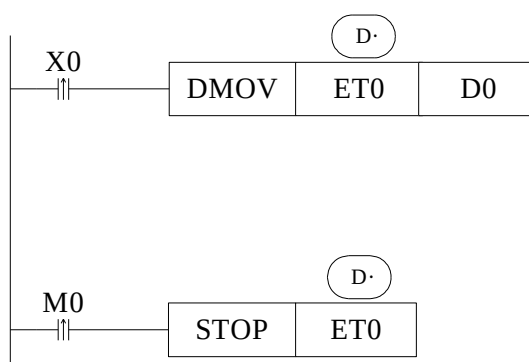
- Точный таймер работает с шагом в 1 мс.

- Точный таймер — 32 бита, диапазон счета — от 0 до +2 147 483 647.
- Во время выполнения STR таймер сбросится до начала синхронизации.
- Когда состояние X0 изменится с OFF на ON, ET0 запустит синхронизацию. ET0 будет сброшен и сохранит свое значение 100, когда время накопления достигнет 100 мс. Если X0 снова перейдет из состояния OFF в состояние ON, таймер T600 перейдет из состояния ON в состояние OFF и перезапустится. Когда время накопления достигнет 100 мс, T600 снова сбросится. См. схему ниже:



Если предварительным условием STR является нормально разомкнутая/замкнутая катушка, точный таймер немедленно включится по достижении заданного времени и сбросит показания времени, а затем будет циклически выполнять возвратно-поступательное движение.

<Считывание данных точной синхронизации>, <Остановка точного таймера>



- Когда состояние X0 изменится с OFF на ON, немедленно переместите текущее значение точной синхронизации в D0; на него не повлияет цикл сканирования.

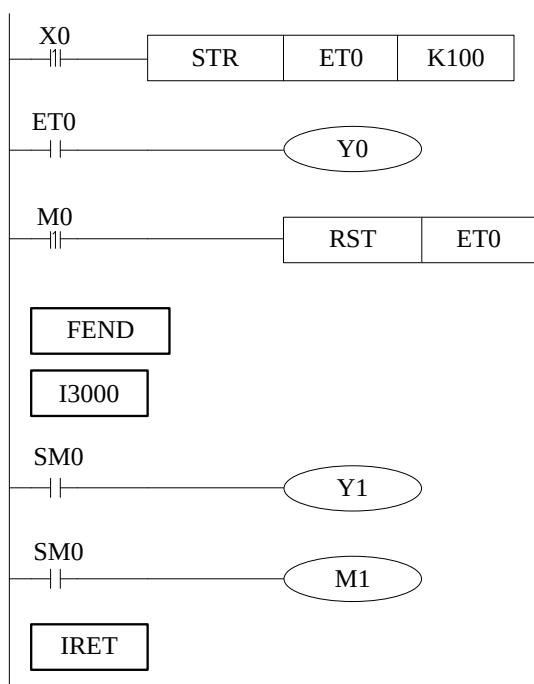
- Когда состояние M0 изменится с OFF на ON, немедленно выполните команду STOP, остановите точную синхронизацию и обновите значение счетчика в ETD0. На него не повлияет цикл сканирования.

Прерывание точной синхронизации

- Когда точное время достигнет значения счетчика, будет сгенерирован признак прерывания, и будет выполнена подпрограмма прерывания.
- Точная синхронизация может начаться после прерывания точной синхронизации.
- Как показано в таблице ниже, каждый точный таймер имеет свой признак прерывания:

Признак прерывания, соответствующий таймеру:

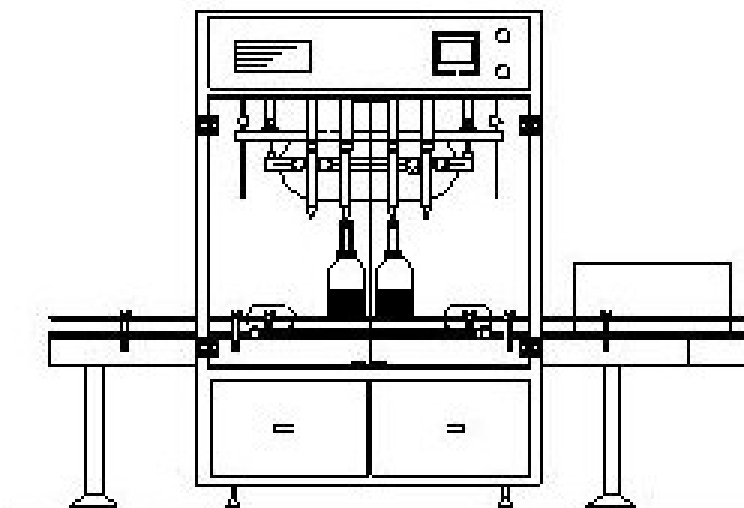
Номер таймера	Признак прерывания	Номер таймера	Признак прерывания
ET0	I3000	ET10	I3005
ET2	I3001	ET12	I3006
ET4	I3002		
ET6	I3003	ET22	I3011
ET8	I3004	ET24	I3012



Когда состояние X0 изменится с OFF на ON, ET0 запустит синхронизацию. ET0 будет сброшен, когда время накопления достигнет 100 мс. При этом генерируется прерывание, программа переходит к признаку прерывания I3000 и выполняет подпрограмму.

Пример 1

Разливочная машина контролирует точность разлива, регулируя время открытия жидкостного клапана (в данном случае оно составляет 3000 мс). Для повышения точности разлива время открытия жидкостного клапана можно регулировать с помощью функции точной синхронизации.

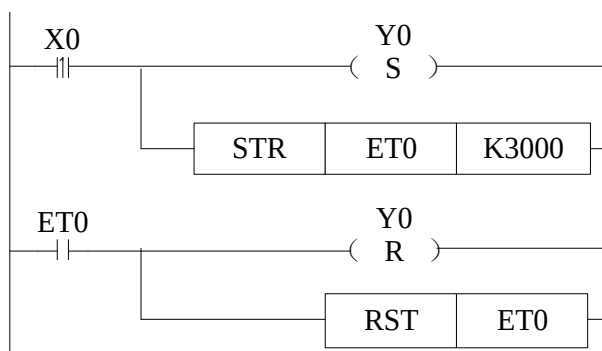


Разливочная машина

Описание компонентов:

Компонент ПЛК	Управляющие компоненты	Примечание
X0	Кнопка Start (Пуск), при нажатии включается X0	
ET0	Точный таймер	
Y0	Регулирование жидкостного клапана, после открытия клапана включается Y0 Y0 выключается после закрытия клапана	

Программа:

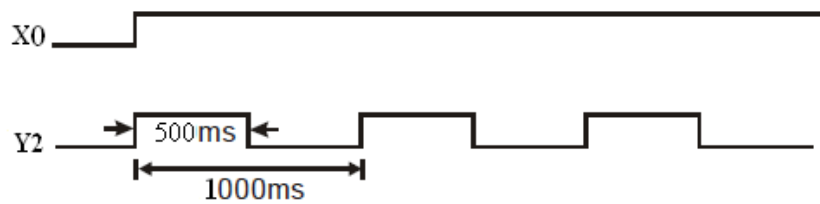


Описание программы:

- 1. Когда X0 включен, жидкостный клапан Y0 сразу открывается, и точный таймер ET0 сразу включается.
- 2. По истечении заданного времени жидкостный клапан Y0 закрывается, а точный таймер ET0 выключается.

Пример 2

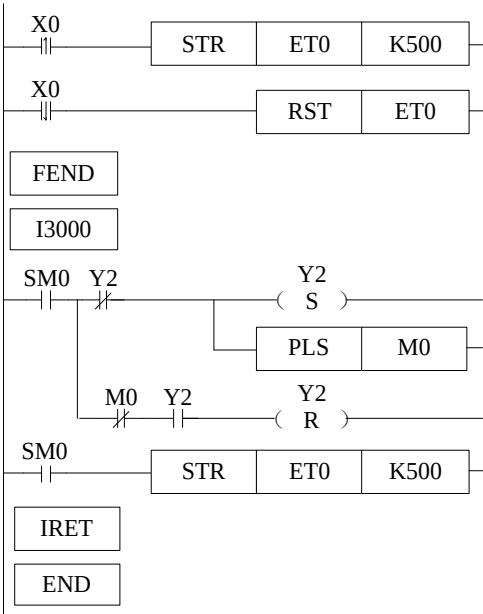
Прерывание точного таймера может вызвать импульсную волну. Время включения Y2 составляет 500 мс, период следования импульсов — 1000 мс.



Описание компонентов:

Компонент ПЛК	Управляющие компоненты	Примечание
X0	Кнопка Start (Пуск), при нажатии включается X0	
Y2	Выходная импульсная клемма	
M0	Внутренняя вспомогательная катушка	
ET0	Точный таймер	

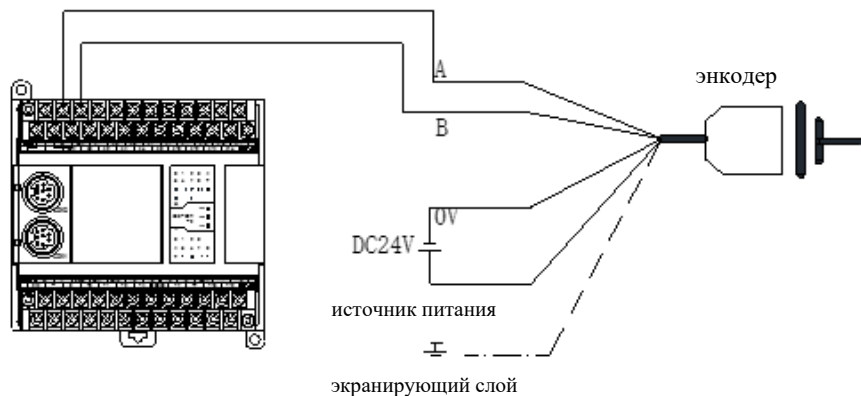
Программа:



Описание программы:

1. Когда X0 включен, срабатывает прерывание точного таймера, и Y2 будет выдавать импульсную волну.
2. Когда X0 выключен, прерывание точного таймера отключится, и Y2 прекратит выдавать импульсы.

Пример 3

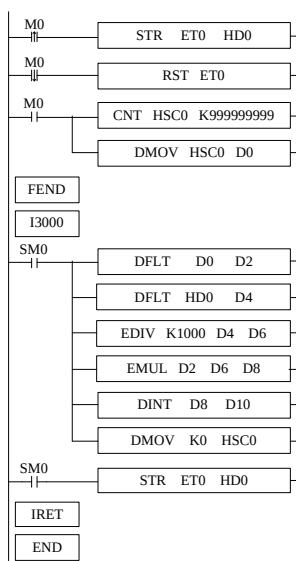


Поскольку FRQM рассчитывает время для фиксированного количества импульсов, мы изменим способ расчета количества импульсов за фиксированное время.

Описание компонентов:

Компонент ПЛК	Управляющие компоненты	Примечание
M0	Кнопка Start (Пуск), при нажатии включается X0	
ET0	Точный таймер	
HD0	Значение настройки точного таймера (единица измерения: мс)	
HSC0	Высокоскоростной счетчик	
D10	Измеренная частота (единица измерения: с)	

Программа:



Описание программы:

1. Настройте регистр периода выборки высокоскоростного счетчика HD0, единица измерения — мс.
2. Включите M0, чтобы запустить прерывание точного таймера и высокоскоростной счетчик, вычислить частоту.
3. Диапазон частот — 0-80 кГц, точность — 0,005%.

10.3 Прерывания [EI], [DI], [IRET]

ПЛК серии RMP20 поддерживают функцию прерываний, включая внешние прерывания и временные прерывания. Функция прерываний позволяет работать с некоторыми специальными программами. На эту функцию не влияет цикл сканирования.

10.3.1 Внешние прерывания

Входные клеммы X можно использовать для ввода внешних прерываний. Каждая входная клемма соответствует одному внешнему прерыванию. Нарастающий фронт/спадающий фронт входа может запустить прерывание. Подпрограмма прерываний записывается после главной программы (после FEND). После запуска прерывания главная программа немедленно останавливается, и начинается выполнение соответствующей подпрограммы. После завершения выполнения подпрограммы выполнение главной программы продолжится.



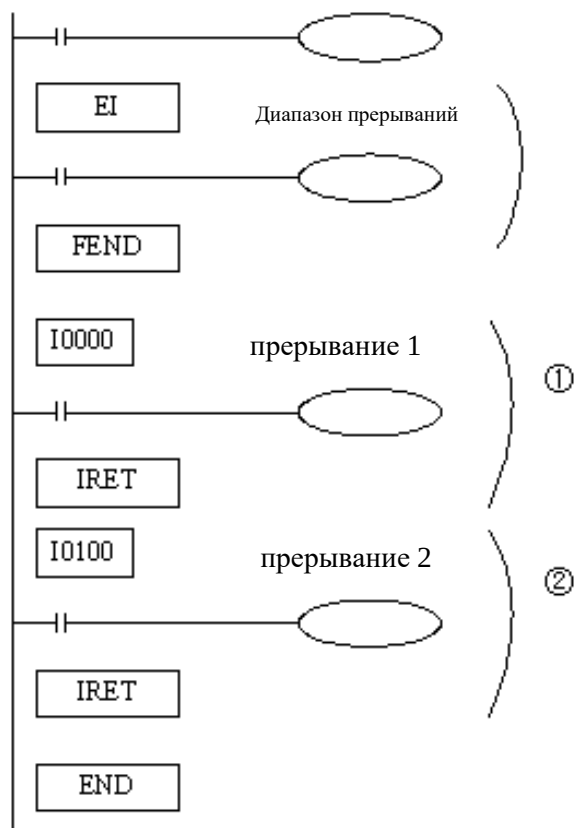
Примечание: ПЛК серии RMP20 поддерживают запуск по нарастающему фронту и спадающему фронту.

Назначение портов внешних прерываний

ПЛК серии RMP20, 16 вводов/выводов				ПЛК серии RMP20, 24-64 ввода/вывода			
Входная клемма	Указатель №		Отключение команды прерываний	Входная клемма	Указатель №		Отключение команды прерываний
	Прерывание по нарастающему фронту	Прерывание по спадающему фронту			Прерывание по нарастающему фронту	Прерывание по спадающему фронту	
X2	I0000	I0001	SM050	X2	I0000	I0001	SM050
X3	I0100	I0101	SM051	X3	I0100	I0101	SM051
X4	I0200	I0201	SM052	X4	I0200	I0201	SM052
X5	I0300	I0301	SM053	X5	I0300	I0301	SM053
X6	I0400	I0401	SM054	X6	I0400	I0401	SM054
X7	I0500	I0501	SM055	X7	I0500	I0501	SM055
ПЛК серии RMP20, 10 вводов/выводов				X10	I0600	I0601	SM056
Входная клемма	Указатель №		Отключение команды прерываний	X11	I0700	I0701	SM057
	Прерывание по нарастающему фронту	Прерывание по спадающему фронту		X12	I0800	I0801	SM058
X2	I0000	I0001	SM050	X13	I0900	I0901	SM059
X3	I0100	I0101	SM051	Примечание: Если катушка запрета прерываний включена, внешние прерывания не будут выполняться.			
X4	I0200	I0201	SM052				

Команды прерываний

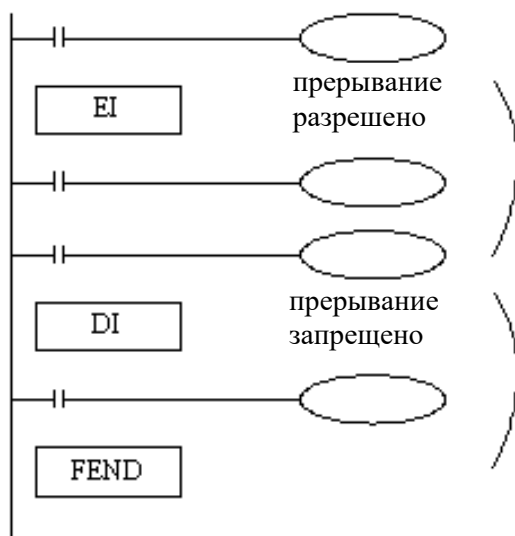
Разрешить прерывание [EI], Запретить прерывание [DI], Возврат прерывания [IRET]



- Если использована команда EI для разрешения прерываний, то при сканировании программы, когда состояние входа прерываний меняется с OFF (ВЫКЛ) на ON (ВКЛ), будет выполняться подпрограмма ①, ②. Происходит возврат в исходную главную программу.
- Указатель прерываний (I****) должен быть после команды FEND.
- ПЛК обычно находится в состоянии, когда прерывания разрешены.

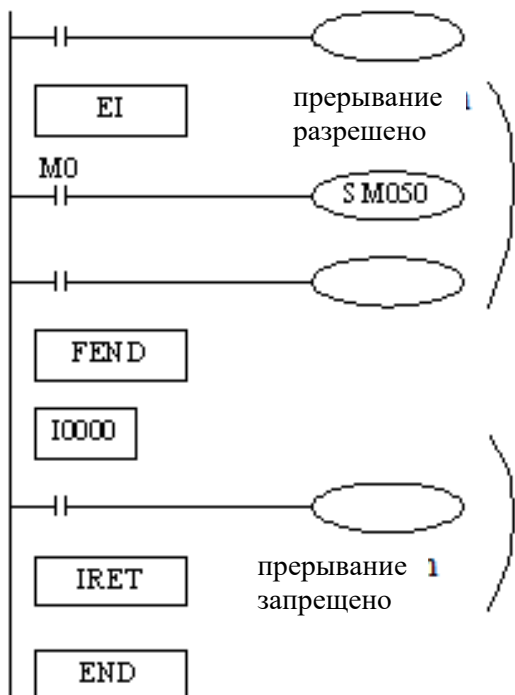
Примечание: В подпрограмме прерываний можно записать только простые команды, например, команды настройки, сброса, передачи и выполнения операций, и эти команды могут быть выполнены в цикле сканирования. Другие команды, например, команды отправки импульсов, синхронизации (за исключением точной синхронизации), связи и другие команды, которые должны выполняться непрерывно, не поддерживаются.

Ограничение диапазона прерываний



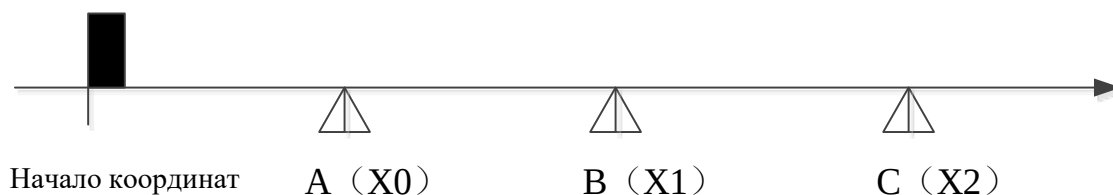
- Запрограммировав команду DI, можно настроить область запрета прерываний.
- Разрешить ввод прерываний между EI~DI.
- Если запрет прерываний не требуется, программируйте только с EI, а программирование с DI не требуется.

Запретить прерывание



- Для каждого прерывания на входе предусмотрены специальные реле (SM50~SM69), позволяющие запретить прерывание.
- В показанной слева программе, если M0 используется для включения SM50, то следует запретить прерывание 0.

Пример 1



Положения А, В, С неизвестны. Скорость каждого из трех сегментов отличается. Можно выполнить команду PLSF и вызвать внешнее прерывание. Можно установить три бесконтактных выключателя в положения А, В, С и подключить сигнал к входным клеммам X0, X1, X2 ПЛК (предположим, что X0, X1, X2 являются клеммами внешних прерываний, а соответствующие идентификаторы прерывания по нарастающему фронту — I0000, I0100, I0200). Клеммы внешних прерываний ПЛК описаны в разделе «Определение клемм внешних прерываний». Импульсная клемма — Y0, клемма направления — Y2. Чтобы улучшить точность изменения скорости, для времени ускорения и торможения задано значение 0. Скорость изменится по внешнему прерыванию.

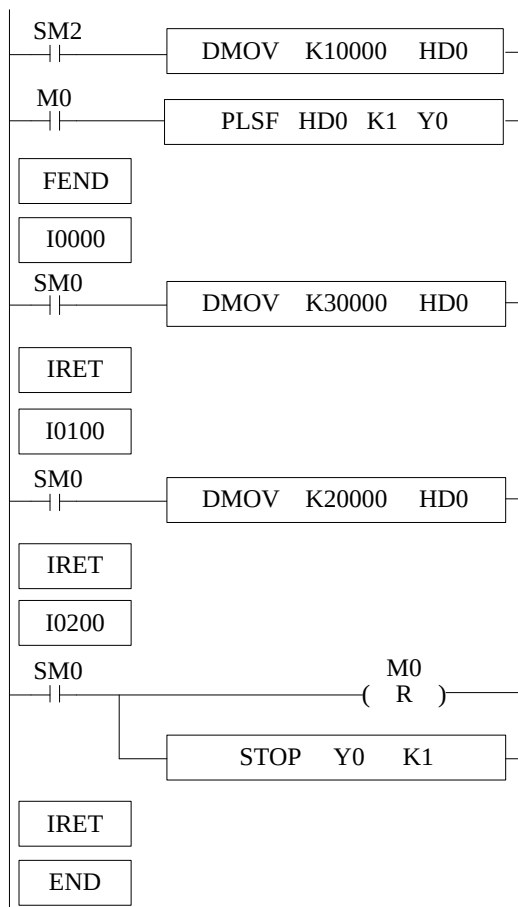
Сегмент	Значение уставки частоты (Гц)	Количество импульсов
Начало координат ---- А	10000	999999999
А ---- В	30000	999999999
В ----- С	20000	999999999
Время ускорения и торможения	0	

Примечание: Поскольку количество импульсов для каждого сегмента неизвестно, количество импульсов должно быть достаточно большим, чтобы обеспечить перемещение объекта к бесконтактному выключателю. Команда STOP запустится по внешнему прерыванию, когда объект достигнет положения С.

Описание компонентов:

Компонент ПЛК	Управляющие компоненты	Метка
M0	Кнопка Start (Пуск), при нажатии на нее PLSF будет передавать импульсы	
HD0	Регистр частоты импульсов PLSF	

Программа:

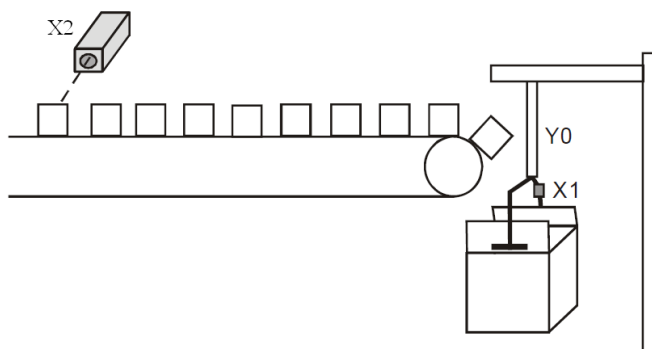


Описание программы:

1. Когда SM2 включен, установите HD0 на 10000, включите M0, и команда PLSF будет передавать импульсы с частотой 10000 Гц, объект переместится из начала координат в точку A.
2. Когда объект достигнет точки A, X0 сразу включится, сработает внешнее прерывание I0000, HD0 установится на 30000, объект будет перемещаться из точки A в точку B со скоростью 30000 Гц.
3. Когда объект достигнет точки B, X1 сразу включится, сработает внешнее прерывание I0100, HD0 установится на 20000, объект будет перемещаться из точки B в точку C со скоростью 20000 Гц.
4. Когда объект достигнет точки C, X2 сразу включится, сработает внешнее прерывание I0200, M0 выключится, отправка импульсов сразу же прекратится.

Пример 2

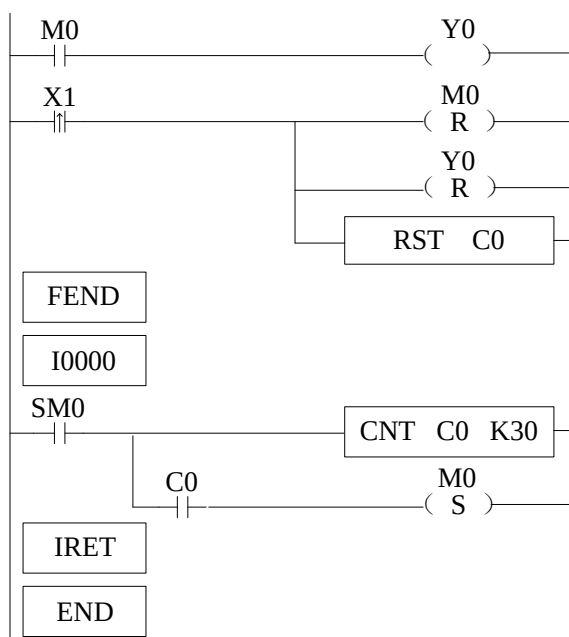
Ниже показана схема работы упаковочной машины. Робот упакует изделия, когда обнаружит 30 изделий, а после завершения упаковки робот и счетчик будут сброшены. Для повышения производительности скорость транспортировки изделий установлена очень высокой, датчик X2 обнаруживает изделие за 8 мс, время фильтрации входной клеммы ПЛК составляет 10 мс. Стандартный счетчик не сможет обнаружить изделия. Внешние прерывания можно использовать для подсчета количества изделий.



Описание компонентов:

Компонент ПЛК	Управляющие компоненты	Примечание
X2	Фотоэлектрический датчик подсчета изделий, X2 включается при обнаружении изделия	
X1	Датчик завершения работы робота, X1 включается, когда работа завершена	
C0	16-битный счетчик	
Y0	Робот	

Программа:



Описание программы:

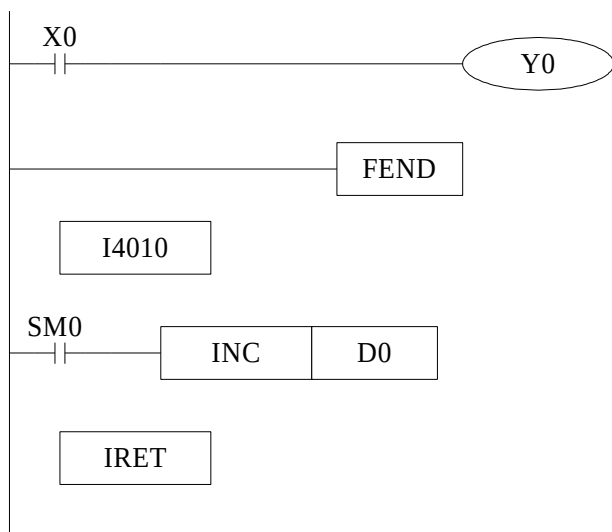
1. В программе внешних прерываний подсчитайте количество изделий на входе X2, когда X2 равен 30, включите M0.
2. В главной программе Y0 управляется в зависимости от состояния M0.
3. После завершения работы робота X1 однократно переключается из состояния ВЫКЛ в состояние ВКЛ, RST срабатывает, Y0 и C0 сбрасываются, M0 выключается. Упаковочная машина переходит в режим ожидания.

10.3.2 Временные прерывания

Функции и операции

Если цикл выполнения главной программы очень длинный, когда приходится работать со специальной программой или время от времени выполнять конкретную программу, когда программа сканирует в режиме управления последовательностью, временные прерывания могут оказаться очень полезными. Такие

прерывания не зависят от цикла сканирования ПЛК, подпрограмма временных прерываний выполняется каждые N мс.



- По умолчанию временные прерывания имеют открытое состояние, как и другие подпрограммы прерывания. Временные прерывания должны записываться после главной программы, начинаться с I40xx и заканчиваться на IRET.
- Существуют 20 каналов временных прерываний: I40**~I59** (**— это время прерывания; единица измерения — мс). Пример: I4010 означает выполнение после первого временного прерывания каждые 10 мс.

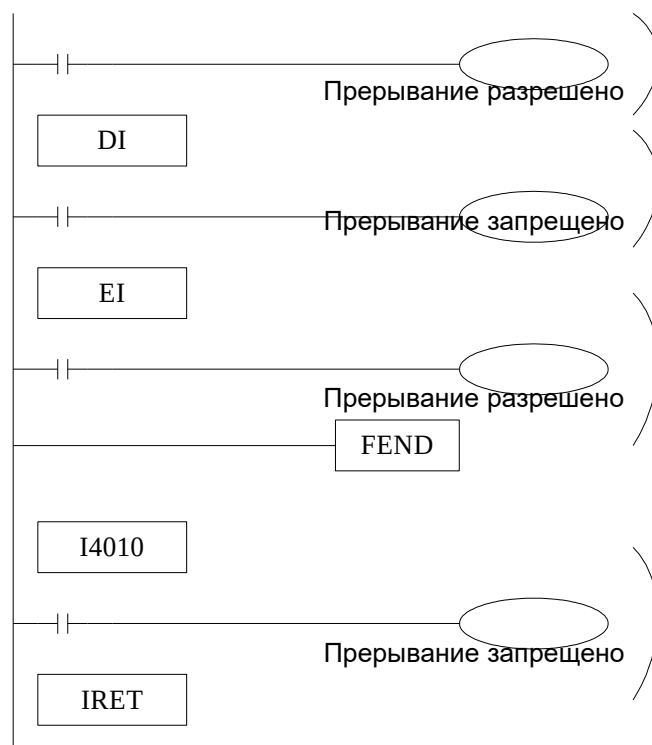
Номер прерывания

Номер прерывания	Команда запрета прерываний	Номер прерывания	Команда запрета прерываний	Описание
I40**	SM070	I50**	SM080	** обозначает продолжительность временных прерываний, диапазон — 1~99, единица измерения — мс. Примечание: 1. Только временное прерывание I59** имеет продолжительность 100 мкс. 2. Функция выбора на основе времени поддерживается только прошивкой ПЛК версии 3.4.6 или более поздней.
I41**	SM071	I51**	SM081	
I42**	SM072	I52**	SM082	
I43**	SM073	I53**	SM083	
I44**	SM074	I54**	SM084	
I45**	SM075	I55**	SM085	
I46**	SM076	I56**	SM086	
I47**	SM077	I57**	SM087	
I48**	SM078	I58**	SM088	
I49**	SM079	I59**	SM089	

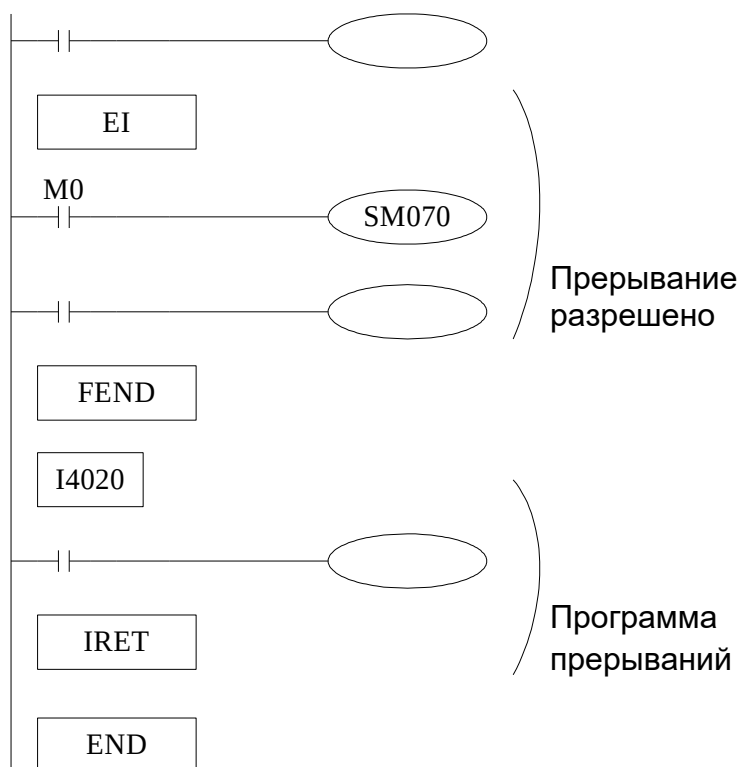
Временные прерывания ПЛК серии PMP20:

Ограничение диапазона прерываний

- Временное прерывание обычно имеет статус «разрешено».
- С помощью команд EI, DI можно разрешать и запрещать прерывания. Как показано на рисунках ниже, все временные прерывания запрещены в диапазоне между DI и EI и разрешены за пределами диапазона DI~EI.



Прерывание запрещено



- Для первых временных прерываний ЗСН предусмотрены специальные реле (SM070~SM079).
- В показанном слева примере, если M0 используется для включения SM070, то следует запретить временные прерывания.

10.4 Управление несколькими станциями [MSC]

1) Обзор

Получите значение энкодера в соответствии с входным сигналом запуска, рассчитайте и сохраните входное и выходное значение для заготовки на каждой станции, сравните сохраненное значение для каждой заготовки на каждой станции с текущим значением энкодера и выведите результат сравнения.

Управление несколькими станциями [MSC]			
16 бит	-	32 бита	MSC
Условие выполнения	ВКЛ/ВЫКЛ в нормальном режиме	Совместимые модели	PMP20
Требования к аппаратному обеспечению	-	Требования к программному обеспечению	-

2) Операнды

Операнды	Функция	Модель
S1	Ввод номера адреса программного компонента точки ввода командного триггера	бит
S2	Ввод номера высокоскоростного счетчика	32 бита, BIN
S3	Ввод количества станций и заготовок, и номера первого адреса регистра времени фильтрации	16 бит, BIN
S4	Ввод номера первого адреса регистра с контрольным значением и значением отклонения	32 бита, BIN
D1	Ввод номера первого адреса регистра, в котором хранится заданное значение и значение для сравнения	16/32 бита, BIN
D2	Ввод номера адреса программного компонента с результатами вывода	бит

3) Соответствующие программные компоненты

Опе-ранды	Программные элементы слова											Программные эле-менты бита							
	Система								Кон-станта	Модуль		Система							
	D	FD	TD	CD	DX	DY	DM	DS	К/Н	Иден-тифи-катор	QD	X	Y	M	S	T	C	Dn.m	
S1												•							
S2	Только HSC																		
S3	•																		
S4	•																		
D1	•																		
D2													•	•					

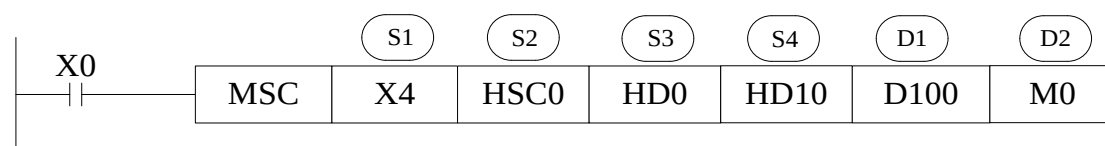
*Примечание:

D включает в себя D, HD; TD включает в себя TD, HTD; CD включает в себя CD, HCD, HSCD, HSD.

DM включает в себя DM, DHM; DS включает в себя DS, DHS; M включает в себя M, HM, SM.

S включает в себя S, HS; T включает в себя T, HT; C включает в себя C, HC.

Функции и операции



- S1: это точка ввода командного триггера, которая позволяет выбрать точку ввода внешнего прерывания или стандартную точку ввода, запустить команду по нарастающему и спадающему фронту и получить значение с энкодера.
- S2: это номер высокоскоростного счетчика, который используется для ввода сигнала энкодера. Режим высокоскоростного счетчика, соответствующий однофазному инкрементному режиму.

- S3: три 16-битных регистра (одно слово) постоянно заняты для настройки количества станций, количества заготовок и времени фильтрации. Рекомендуется использовать регистр временного хранения данных после выключения питания.

Регистры распределены следующим образом:

S3: задает количество станций, записывается как n , диапазон: 1-32;

S3+1: задает максимальное количество обрабатываемых заготовок, записывается как m , диапазон: 1-64;

S3+2: задает время фильтрации, диапазон: 0-32767, единица измерения: мс. Этот параметр можно использовать для предотвращения ошибок, вызванных механическим дрожанием. Если для времени фильтрации задано значение 0, фильтрация не будет выполняться. Если значение меньше 0, оно считается равным 0. Если допустить, что время фильтрации равно t , а точка ввода командного триггера — X4, захват входного сигнала осуществляется следующими способами:

Нарастающий фронт: после выключения X4 и сохранения этого состояния в течение как минимум t мс, первый обнаруженный нарастающий фронт будет являться запускающим сигналом.

Спадающий фронт: после включения X4 и сохранения этого состояния в течение как минимум t мс, первый обнаруженный спадающий фронт будет являться запускающим сигналом.

- S4: Постоянно заняты $3n$ 32-битных регистра (двойные слова), которые используются для настройки контрольного значения, значения отклонения на участке загрузки заготовок и значения отклонения на участке выгрузки заготовок на каждой станции. Каждый параметр непрерывно занимает 2 регистра. Рекомендуется использовать регистр временного хранения данных после выключения питания. Адреса регистров распределены следующим образом:

Название	Станция 1	Станция 2		Станция n
Контрольное значение (двойное слово)	S4	S4+2		S4+($n-1$) $\times$ 2
Значение отклонения на участке загрузки заготовок (двойное слово)	S4+2 n	S4+2 n +2		S4+(2 $n-1$) $\times$ 2

Значение отклонения на участке выгрузки заготовок (двойное слово)	$S4+4n$	$S4+4n+2$		$S4+(4n-1) \times 2$
---	---------	-----------	-------	----------------------

- Если контрольное значение для станции равно 0, это означает, что станция не работает.
- Для калибровки положения в основном используются значения отклонения на участке загрузки заготовок и отклонения на участке выгрузки заготовок. Если значение энкодера для заготовки, входящей и выходящей из соответствующей станции, не соответствует уставке во время фактической эксплуатации, его можно откалибровать, отрегулировав значение отклонения на участке загрузки заготовок и значение отклонения на участке выгрузки заготовок. Например, контрольное значение для станции 1 равно 1000. Это означает, что заготовка поступает на станцию 1 после запуска по нарастающему фронту X4 через 1000 значений высокоскоростного счетчика. Если во время фактической эксплуатации заготовка поступает на станцию 1 только с 990 значениями высокоскоростного счетчика, то значение отклонения на участке выгрузки заготовок можно установить на уровне -10.
- D1: непрерывно занимает 2n 16-битных регистра (одно слово), а $2m \times n$ 32-битных регистра (двойное слово) используются для хранения заданных значений перемещения заготовки вперед, заданных значений следящего управления, значений для сравнения входа и значений для сравнения выхода на каждой станции. Адреса регистров распределены следующим образом:

Название	Станция 1	Станция 2		Станция n
Заданное значение перемещения вперед (слово)	D1	D1+1		D1+(n-1)
Заданное значение следящего управления (слово)	D1+n	D1+(n+1)		D1+(2n-1)
Значение для сравнения заготовки 1 на участке загрузки заготовок (двойное слово)	D1+2n	D1+2n+2		D1+2n+2(n-1)
Значение для сравнения заготовки 1 на участке выгрузки заготовок (двойное слово)	D1+4n	D1+4n+2		D1+4n+2(n-1)
.....				
Значение для сравнения на участке загрузки заготовки m (двойное слово)	D1+4mn-2n	D1+4mn-2n+2		D1+4mn-2

Значение для сравнения на участке выгрузки заготовки m (двойное слово)	$D1+4mn$	$D1+4mn+2$		$D1+4mn+2(n-1)$
--	----------	------------	-------	-----------------

Примечание: D1 занимает большой объем памяти. Проверьте, достаточно ли места в регистре. Если его недостаточно, ПЛК будет только сохранять данные в рабочей области и не будет генерировать сигналы тревоги и подсказки.

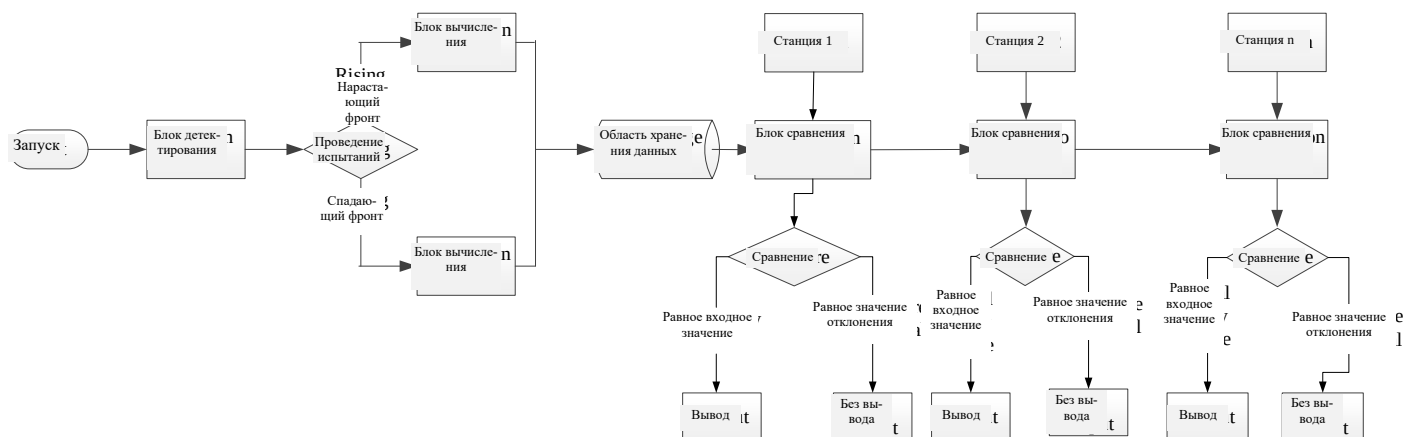
- Если значение для сравнения на входе и значение для сравнения на выходе станции равны 0, это означает, что операция сравнения для станции не выполняется.
- Заданное значение перемещения вперед будет автоматически увеличиваться на 1 при запуске по каждому нарастающему и спадающему фронту входного сигнала (если время фильтрации > 0 , дождитесь пока оно не истечет, а затем увеличивайте значение на 1), при этом применяется метод циклического накопления. Например, если максимальное количество обрабатываемых заготовок составляет $m = 10$, то заданное значение перемещения вперед будет циклически изменяться на 0, 1, 2, 3... 19, 0, 1, 2, 3... 19 (исходное значение равно 0). Поскольку заданное значение перемещения вперед будет увеличиваться на 1 как при нарастающем, так и при спадающем фронте, максимальное заданное значение перемещения вперед составит $2*m$.

Примечание: Следующее заданное значение будет оцениваться перед добавлением 1 к заданному значению перемещения вперед. Если значение после прибавления 1 равно следующему заданному значению, то заданное значение перемещения вперед не будет накапливаться, а вместо него будет записано значение для сравнения.

- К заданному значению перемещения вперед будет автоматически добавляться 1, когда заготовка входит и выходит из станции. Как правило, после того, как заготовка прошла через станцию, следующее заданное значение для соответствующей станции будет четным.
- Значение для сравнения на входе автоматически рассчитывается и сохраняется в области данных D1, когда соответствующая заготовка вызывает нарастающий фронт входного сигнала. Значение для сравнения на входе станции обычно составляет:
- Значение для сравнения для заготовки m , поступающей на станцию $n =$ значение количества захватов заготовки m (при нарастающем фронте) + контрольное значение станции n + значение отклонения для заготовки, поступающей на станцию n .

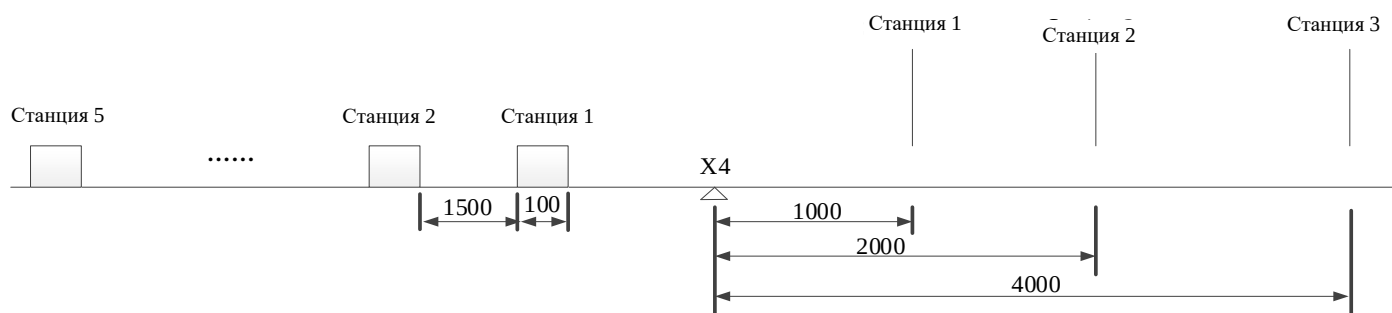
- Значение для сравнения на участке выгрузки автоматически рассчитывается и сохраняется в области данных D1, когда спадающий фронт входного сигнала вызван соответствующей заготовкой. Значение для сравнения на выходе станции обычно составляет:
- Значение для сравнения для заготовки m , выходящей из станции n = значение количества захватов заготовки m (при спадающем фронте) + контрольное значение станции n + значение отклонения для заготовки, выходящей из станции n .
- D2: непрерывно занимает n катушек (соответствует числу n станций), и только выходы катушек Y и M можно выбирать, чтобы определить, поступает ли соответствующая заготовка на станцию и выходит ли из нее. После выполнения команды каждая станция будет оценивать, входит ли соответствующая заготовка на станцию и выходит ли она из нее, в соответствии с заданным значением для сравнения согласно заданному значению следящего управления. Когда значение счета в реальном времени для соответствующей заготовки $\geq$ введенного значения для сравнения, соответствующая точка вывода включается, а заданное значение следящего управления автоматически увеличивается на 1; когда значение счета в реальном времени для соответствующей заготовки выходит за пределы значения для сравнения, соответствующая точка вывода выключается, а заданное значение следящего управления автоматически увеличивается на 1, но не превышает заданное значение перемещения вперед.
- Ограничений на количество раз использования команд MSc нет, но если в программе нужно задействовать один и тот же высокоскоростной счетчик, то каждую команду необходимо поместить в отдельный процесс, и одновременно может выполняться только одна команда.
- Перед выполнением команды проверьте, не переполнен ли используемый высокоскоростной счетчик (об этом можно судить по биту признака переполнения высокоскоростного счетчика sm130, и т. д.), и выполните соответствующую обработку.
- При отключении и повторном подключении предварительного условия выполнения команд MSC значения в областях хранения D1 и D2 будут сброшены на 0 и отключены.



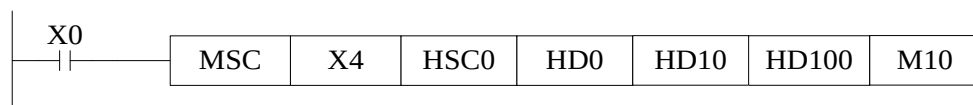


Пример процедуры

Например, пять заготовок необходимо обработать на трех станциях. Входной сигнал триггера - X4, точка ввода сигнала энкодера - X0 (соответствующий высокоскоростной счетчик - HSC0), ширина каждой заготовки - 100, расстояние между заготовками - 1500, расстояние между заготовками - 1000, расстояние между заготовками - 1000, расстояние между заготовками - X4,2000, расстояние между заготовками - 4000.



Процедура выглядит следующим образом:



Назначение адреса результатов вывода:

Адрес программного компонента	Описание функции
X4	Точка ввода триггера
HSC0	Точка подключения высокоскоростного счетчика, прием сигнала энкодера
HD0	Количество станций, HD0 = 3 в этом примере
HD1	Максимальное количество заготовок, которое можно обработать, HD1 = 4 в этом примере
HD2	Время фильтрации, HD2 = 300 мс в этом примере
HD10 (двойное слово)	Контрольное значение для станции 1, HD10 = 1000 в этом примере
HD12 (двойное слово)	Контрольное значение для станции 2, HD12 = 2000 в этом примере
HD14 (двойное слово)	Контрольное значение для станции 3, HD14 = 4000 в этом примере
HD16 (двойное слово)	Заготовка станции 1 отклоняется и принимает значение 0
HD18 (двойное слово)	Заготовка станции 2 отклоняется и принимает значение 0
HD20 (двойное слово)	Заготовка станции 3 отклоняется и принимает значение 0
HD22 (двойное слово)	Значение отклонения заготовки на участке выгрузки заготовок из станции 1 равно 0
HD24 (двойное слово)	Значение отклонения заготовки на участке выгрузки заготовок из станции 2 равно 0
HD26 (двойное слово)	Значение отклонения заготовки на участке выгрузки заготовок из станции 3 равно 0

Название	Станция 1	Станция 2	Станция 3
Заданное значение перемещения вперед (одно слово)	HD100	HD101	HD102
Заданное значение следящего управления (одно слово)	HD103	HD104	HD105
Значение для сравнения заготовки 1 на участке загрузки заготовок (двойное слово)	HD106	HD108	HD110
Значение для сравнения заготовки 1 на участке выгрузки заготовок (двойное слово)	HD112	HD114	HD116
Значение для сравнения заготовки 2 на участке загрузки заготовок (двойное слово)	HD118	HD120	HD122



Значение для сравнения заготовки 2 на участке выгрузки заготовок (двойное слово)	HD124	HD126	HD128
Значение для сравнения заготовки 3 на участке загрузки заготовок (двойное слово)	HD130	HD132	HD134
Значение для сравнения заготовки 3 на участке выгрузки заготовок (двойное слово)	HD136	HD138	HD140
Значение для сравнения заготовки 4 на участке загрузки заготовок (двойное слово)	HD142	HD144	HD146
Значение для сравнения заготовки 4 на участке выгрузки заготовок (двойное слово)	HD148	HD150	HD152
Признак вывода	M10	M11	M12

Результаты выполнения программы:

При допущении, что значение высокоскоростного счетчика, когда заготовка 1 запускает нарастающий фронт X4, равно 1000, заданное значение перемещения вперед, заданное значение следящего управления, значение для сравнения заготовки на участке загрузки заготовок и значение для сравнения заготовки на участке выгрузки заготовок для каждой станции представлены в таблице ниже:

Параметр		Станция 1	Станция 2	Станция 3
Заготовка 1	Заданное значение перемещения вперед	Нарастающий фронт X4:1	Нарастающий фронт X4:1	Нарастающий фронт X4:1
		Спадающий фронт X4: 2	Спадающий фронт X4: 2	Спадающий фронт X4: 2
	Заданное значение следящего управления	Нарастающий фронт M10 1	Нарастающий фронт M11 1	Нарастающий фронт M12 1
		Спадающий фронт M10: 2	Спадающий фронт M11: 2	Спадающий фронт M12: 2
	Значение для сравнения заготовки на участке загрузки заготовок	HD106=2000	HD108=3000	HD110=5000
	Значение для сравнения	HD112=2100	HD114=3100	HD116=5100

	заготовки на участке выгрузки заготовок			
Заготовка 2	Заданное значение перемещения вперед	Нарастающий фронт X4:3	Нарастающий фронт X4:3	Нарастающий фронт X4:3
		Спадающий фронт X4:4	Спадающий фронт X4:4	Спадающий фронт X4:4
	Заданное значение следящего управления	Нарастающий фронт M10 3	Нарастающий фронт M11 3	Нарастающий фронт M12 3
		Спадающий фронт M10: 4	Спадающий фронт M11: 4	Спадающий фронт M12: 4
	Значение для сравнения заготовки на участке загрузки заготовок	HD118=3600	HD120=4600	HD122=6600
	Значение для сравнения заготовки на участке выгрузки заготовок	HD124=3700	HD126=4700	HD128=6700
Заготовка 3	Заданное значение перемещения вперед	Нарастающий фронт X4:5	Нарастающий фронт X4:5	Нарастающий фронт X4:5
		Спадающий фронт X4:6	Спадающий фронт X4:6	Спадающий фронт X4:6
	Заданное значение следящего управления	Нарастающий фронт M10:5	Нарастающий фронт M11:5	Нарастающий фронт M12:5
		Спадающий фронт M10: 6	Спадающий фронт M11: 6	Спадающий фронт M12: 6
	Значение для сравнения заготовки на участке загрузки заготовок	HD130=5200	HD132=6200	HD134=8200
	Значение для сравнения заготовки на участке выгрузки заготовок	HD136=5300	HD138=6300	HD140=8300
Заготовка 4	Заданное значение перемещения вперед	Нарастающий фронт X4:7	Нарастающий фронт X4:7	Нарастающий фронт X4:7
		Спадающий фронт X4:0	Спадающий фронт X4:0	Спадающий фронт X4:0
		Нарастающий фронт M10:7	Нарастающий фронт M11:7	Нарастающий фронт M12:7



	Заданное значение следящего управления	Спадающий фронт M10: 0	Спадающий фронт M11: 0	Спадающий фронт M12: 0
	Значение для сравнения заготовки на участке загрузки заготовок	HD142=6800	HD144=7800	HD146=9800
	Значение для сравнения заготовки на участке выгрузки заготовок	HD148=6900	HD150=7900	HD152=9900
Заготовка 5	Заданное значение перемещения вперед	Нарастающий фронт X4:1	Нарастающий фронт X4:1	Нарастающий фронт X4:1
		Спадающий фронт X4:2	Спадающий фронт X4:2	Спадающий фронт X4:2
	Заданное значение следящего управления	Нарастающий фронт M10:1	Нарастающий фронт M11:1	Нарастающий фронт M12:1
		Спадающий фронт M10: 2	Спадающий фронт M11: 2	Спадающий фронт M12: 2
	Значение для сравнения заготовки на участке загрузки заготовок	HD106=8400	HD108=9400	HD110=11400
	Значение для сравнения заготовки на участке выгрузки заготовок	HD112=8500	HD114=9500	HD116=11500

Примечание: после отключения и повторного подключения X0 все данные в приведенной выше таблице будут сброшены на 0.

11. Общие вопросы и ответы

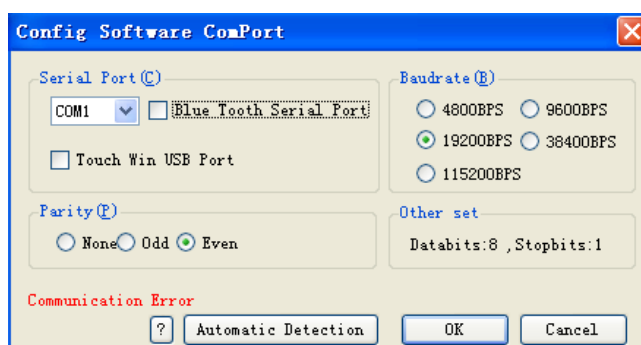
В этой главе представлены общие вопросы и ответы, касающиеся ПЛК серии PMP20.

Вопрос 1: Как подключить ПЛК к ПК?

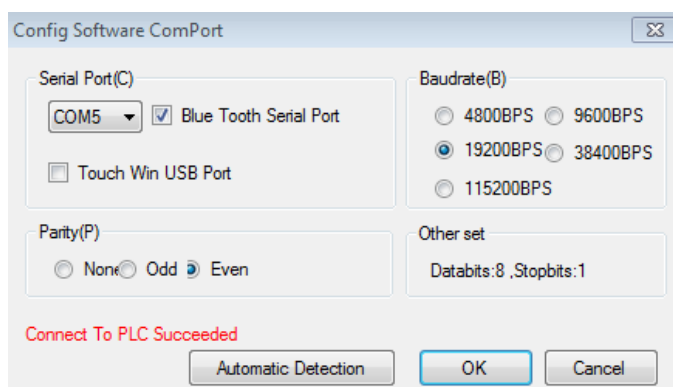
Ответ 1: ПЛК серии PMP20 поддерживают порт COM1 (RS232), порт COM2 (RS485) и порт Ethernet (RJ45) для подключения.

1. Подключение через порт COM1 (RS232) и ПК

Если у вас настольный ПК, можно использовать специальные кабели DVP или XVP нашей компании для подключения ПК к ПЛК (обычно через PORT1). Стандартный коммерческий настольный компьютер имеет 9-контактный последовательный порт. После правильного подключения DVP включите ПЛК, нажмите Config Software ComPort , откроется показанное ниже окно:



Выберите последовательный порт связи в соответствии с фактическим последовательным портом вашего ПК; скорость передачи данных должна быть 19200 бит/с, выберите четность, 8 битов данных, 1 стоповый бит; можно также нажать кнопку Check (Проверить) прямо в окне, и параметры связи будут выбраны самим ПЛК. В левом нижнем углу окна отобразится сообщение 'Connect to PLC succeed' (Подключение к ПЛК выполнено успешно), как показано на рисунке ниже:



Это означает, что ПЛК успешно подключен к ПК!

Подключение к ноутбуку с 9-контактным последовательным портом выполняется так же как для настольного ПК.

Если на ноутбуке нет 9-контактного последовательного порта, можно использовать USB-переходник для соединения между ПЛК и USB-портом ноутбука. Обязательно установите программное обеспечение для USB-переходника (рекомендуется специальный USB-переходник COM-USB от PROMPOWER, программное обеспечение для USB-переходника можно скачать на официальном сайте PROMPOWER)!

2. Подключение через порт COM2 (RS485) и ПК

Если компьютер имеет 9-контактный последовательный порт, то можно подключать ПК к ПЛК (обычно через порт COM2) через последовательный преобразователь RS485 и кабель XVP. Если компьютер имеет только интерфейс USB, его можно подключить через кабель USB-RS485.

После правильного подключения включите ПЛК, нажмите Config Software ComPort,  откроется показанное ниже окно:

Communication configuration

New Edit Delete Move-Up Move-Down

Name	Connection status	Status	Belonging	Description	Connect Info
COM_Modbus_1	Not connected	in use	Global	Station number: 1, serial port: COM3, baud r...	
Ethernet_Xnet_1	Not connected		Global	Ethernet, specified address connection, IP a...	
Ethernet_Modbus_Default	Not connected		Global	Modbus-TCP connection, device IP address: 19...	

Communication configuration

Communication Name: COM_Modbus_1

Connection mode selection

Interface Type: COM

CommProtocol: Modbus

Communication parameter configuration

Automatic Detection

Station No: 1

Baudrate(B): 4800BPS 9600BPS 19200BPS 38400BPS 115200BPS

Serial Port(C): COM3 (USB)

Blue Tooth Serial Port

Parity(P): None Odd Even

Other set: Databits:8, Stopbits:1

Auto-connect on exit

Comm-Test OK Cancel

Выберите последовательный порт связи в соответствии с фактическим последовательным портом вашего ПК; скорость передачи данных должна быть 19200 бит/с, выберите четность, 8 битов данных, 1 стоповый бит; можно также нажать кнопку Check (Проверить) прямо в окне, и параметры связи будут выбраны самим ПЛК. В левом нижнем углу окна отобразится сообщение 'Connect to PLC succeed' (Подключение к ПЛК выполнено успешно), как показано на рисунке ниже.

3. Подключение через порт RJ45

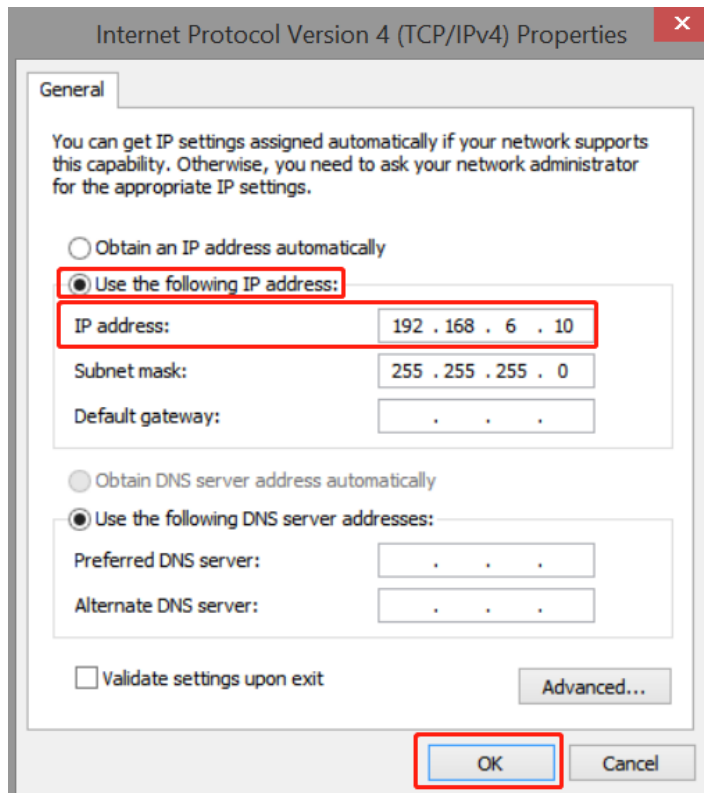
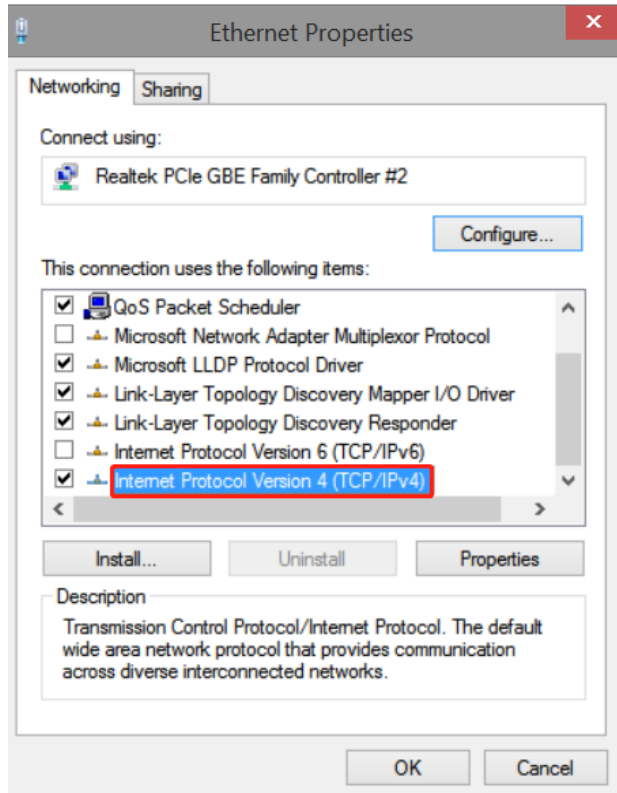
1) Настройка компьютера

После подключения сетевого кабеля откройте "control panel" → "network and Internet" → "network connection" (Панель управления → Сеть и Интернет → Сетевые подключения).

Найдите успешно выполненное подключение к сети Ethernet. Щелкните правой кнопкой мыши по Ethernet и нажмите Properties (Свойства). Откроется окно свойств Ethernet. Выполните следующие действия:

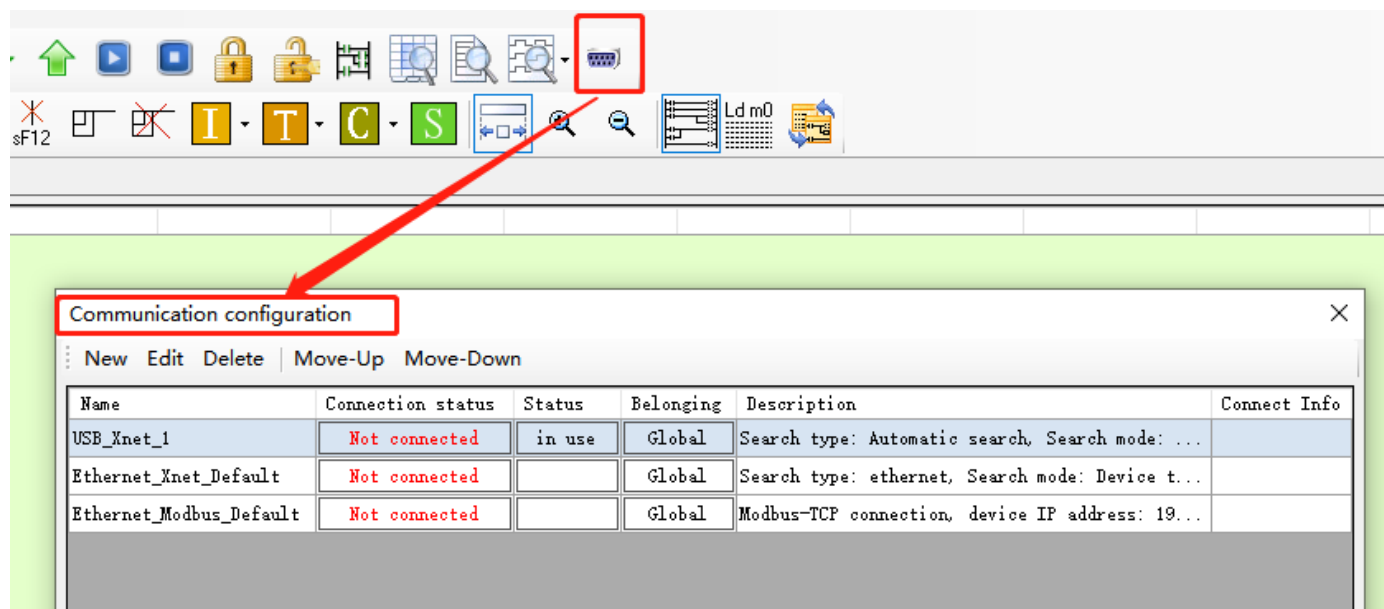
- (1) Дважды щелкните "Internet Protocol Version 4 (TCP/IPV4)" (Протокол Интернета версии 4 (TCP/IPV4)).
- (2) Выберите "use the following IP address" (Использовать следующий IP-адрес).
- (3) Настройте IP-адрес: 192.168.6.xxx, "xxx" можно ввести произвольно (кроме 6).

Примечание: Последнюю цифру адреса компьютера и IP-адрес устройства ПЛК нельзя повторно вводить.



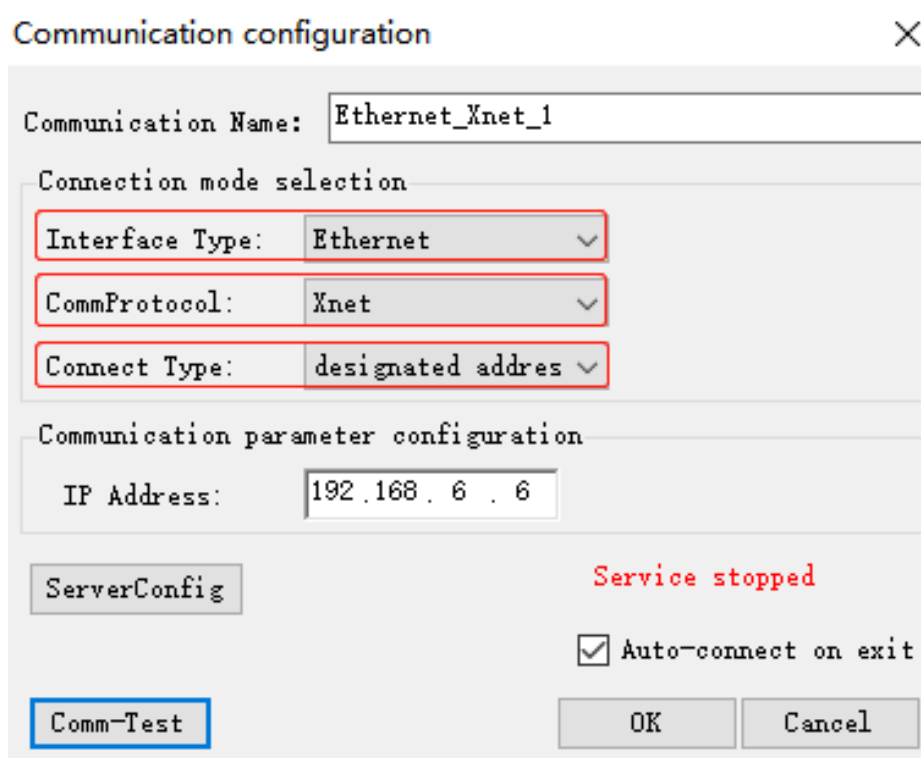
2) Настройка ПЛК

После проверки подключения и настроек Ethernet откройте инструмент программирования PROMPOWER PLC STUDIO → щелкните по Communication configuration (Настройка связи) → дважды щелкните по Ethernet-Xnet.



Выполните настройку, как показано на рисунке ниже:

Выберите протокол Xnet, IP-адрес — это IP-адрес вашего ПЛК. Нажмите [Comm-Test] (Проверка соединения), появится сообщение 'Connect to PLC succeed' (Подключение к ПЛК выполнено успешно).



Нажмите ОК после настройки и выберите " in use" (используется) для соответствующего статуса.

Вопрос 2: ПК не может подключиться к ПЛК через порт RS232, отображается статус «офлайн»

Ответ 2:

Возможные причины:

Случайно изменены параметры связи через PORT1 в ПЛК (не изменяйте параметры связи через Port1, иначе это может привести к сбою соединения между ПК и ПЛК!).

Поврежден кабель связи PORT1, подключенный к ПЛК.

Используется кабель связи, не одобренный компанией PROMPOWER.

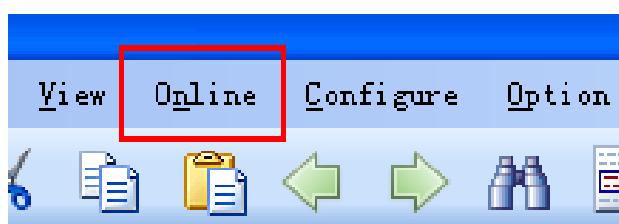
Методы устранения:

Используйте оригинальный кабель XVP от компании PROMPOWER для подключения ПК к ПЛК.

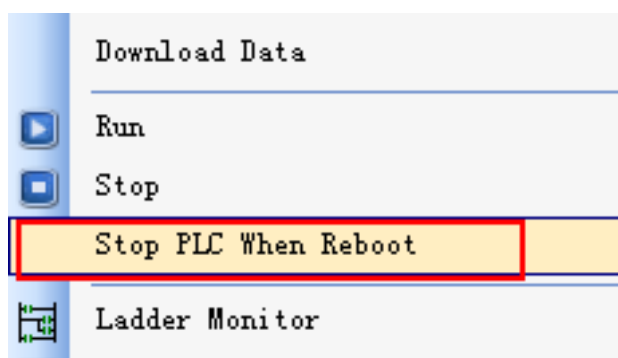
Убедившись, что соединительный кабель является специальным кабелем XVP от PROMPOWER, можно использовать его для подключения настольного ПК с 9-контактным последовательным портом к ПЛК. Если настольный ПК подключается правильно, замените кабель с USB-переходником на более производительный или заново установите драйвер последовательного порта для USB-переходника.

Если ПЛК не подключается к настольному компьютеру, можно использовать функцию stop PLC when reboot (остановить ПЛК при перезагрузке), чтобы остановить ПЛК и восстановить заводские настройки ПЛК. Выполните следующие действия.

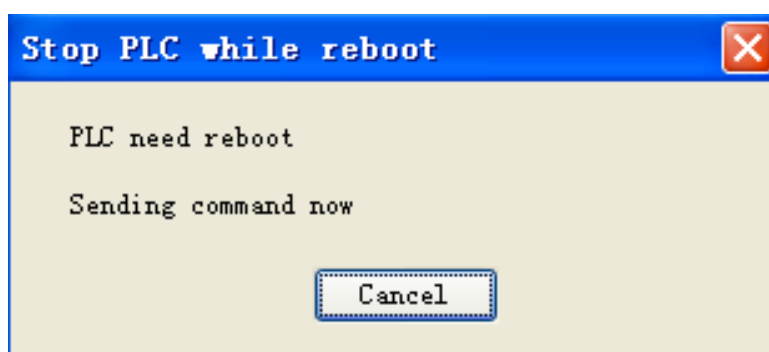
Включите ПЛК и подключите ПЛК с помощью кабелей DVP, затем нажмите кнопку 'online' (Онлайн) в меню программного обеспечения для редактирования настроек ПЛК.



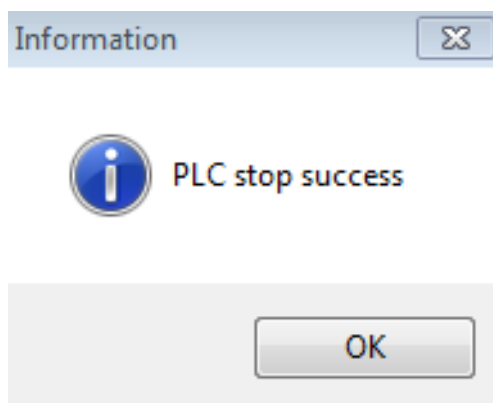
В раскрывающемся меню выберите 'Stop when PLC reboot' (Остановить ПЛК при перезагрузке).



Откроется показанное ниже окно.



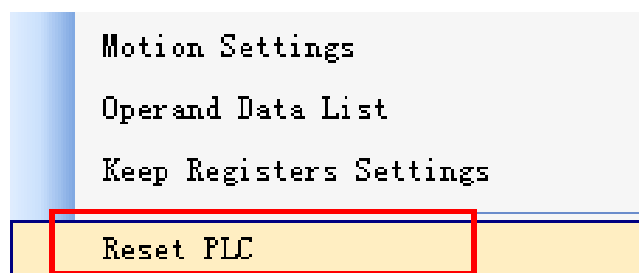
Отключите питание ПЛК на 2–3 секунды и включите его снова, после чего обычно появляется окно 'PLC has been stopped successfully' (ПЛК успешно остановлен). Если окно не появляется после включения питания, попробуйте еще несколько раз, пока не появится окно с информацией об успешной остановке.



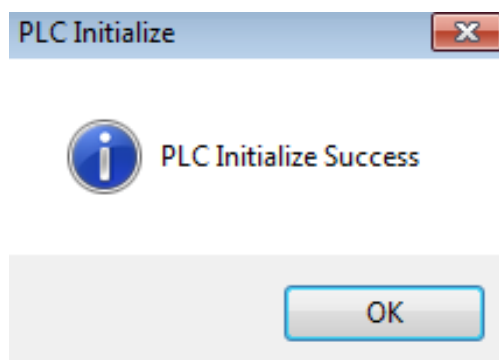
Затем нажмите кнопку 'configure' (Настроить).



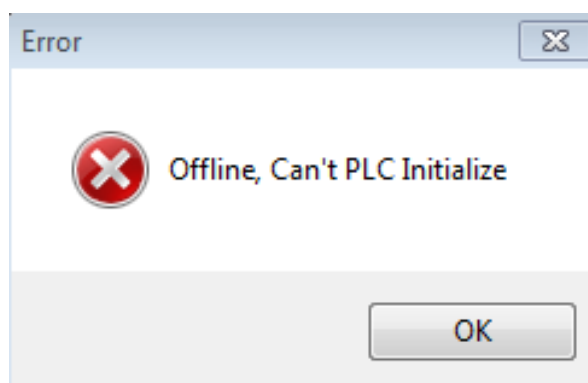
В раскрывающемся меню нажмите кнопку 'Reset PLC' (Сбросить ПЛК).



Появится информационное окно 'Reset PLC' (Сбросить ПЛК), что означает, что все шаги процедуры 'Stop when PLC reboot' (Остановить ПЛК при перезагрузке) выполнены.



Если инициализацию ПЛК не удалось выполнить после нескольких попыток, или после нажатия кнопки 'Reset PLC' (Сбросить ПЛК) появилось показанное ниже окно:



В обоих случаях используйте инструмент обновления системы ПЛК для обновления системы ПЛК. После обновления ПЛК и ПК будут успешно подключены (для получения дополнительной информации по обновлению системы см. пункт «Вопрос 3»).

Если обновление настольного компьютера с 9-контактным последовательным портом не удастся выполнить, весьма вероятно, что поврежден порт связи ПЛК, в этом случае обратитесь к производителю или торговому представителю.

Вопрос 3: Функции программного компонента Bit

Ответ 3:

Непрерывные 16 катушек состоят из слова, например: Слово DM0 состоит из 16 катушек (битов) M0~M15:

DM0:

M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

Можно использовать бит в регистре напрямую.

Пример 1:



Когда состояние M100 изменилось с OFF на ON, M0 M1 включены, а M2—M15 выключены.

Другой режим — битовая операция фиксированного регистра. Пример: D0.0 — первый бит из 16 бит в регистре D0. Аналогично, D0.1 — это второй бит и т. д., как показано ниже:

D0:

D0.15	D0.14	D0.13	D0.12	D0.11	D0.10	D0.9	D0.8	D0.7	D0.6	D0.5	D0.4	D0.3	D0.2	D0.1	D0.0
-------	-------	-------	-------	-------	-------	------	------	------	------	------	------	------	------	------	------

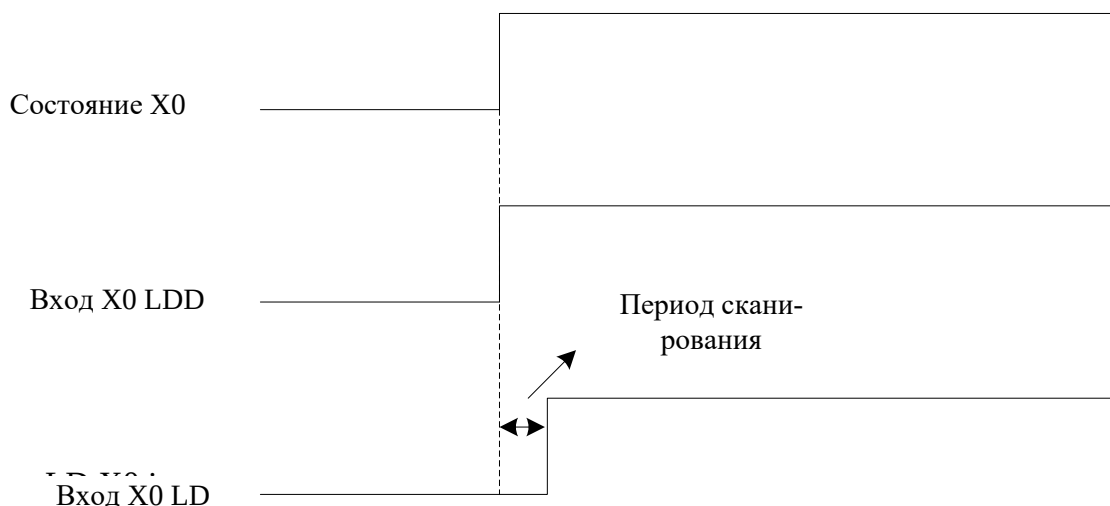
Аналогичным образом, можно использовать бит в регистре D0.

Вопрос 4: Для чего нужны команды выполнения LDD/OUTD, и т.д.?

Ответ 4:

Когда ПЛК выполняет программу, состояние точки ввода будет отображаться в регистре изображений. С этого момента ПЛК будет обновлять состояние входа в начале каждого цикла сканирования. Если используется команда LDD, то состояние точки ввода не нужно будет отображать в регистре изображений; то же самое касается и точки вывода (OUTD).

Команда LDD/OUTD обычно применяется в тех случаях, когда требуется немедленное обновление вводов/выводов, что позволяет избежать воздействия цикла сканирования на состояние ввода и вывода.



Точка ввода X0, схема последовательности ввода команд LDD и LD.

Вопрос 5: Почему выходной светодиод продолжает мигать при использовании команды ALT?

Ответ 5:

ALT и многие команды вычислений выполняются каждый период сканирования, если соблюдены условия (например, условием является нормальное включение катушки). Рекомендуемым условием является нарастающий фронт или спадающий фронт.

Вопрос 6: Почему катушки M и Y иногда не выводят сигналы?

Ответ 6:

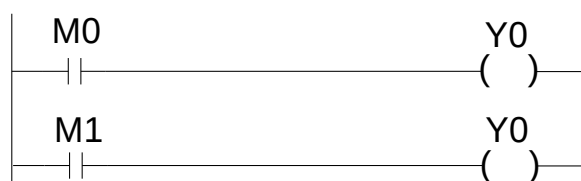
Как правило, выводятся две команды:

1. Команда OUT
2. Команда SET

Катушка будет продолжать выводить сигналы, если нет команды RST.

Одна катушка M или Y в программе, как правило, использует один и тот же способ вывода. В противном случае, катушка вообще не сможет выводить сигналы.

Пример:



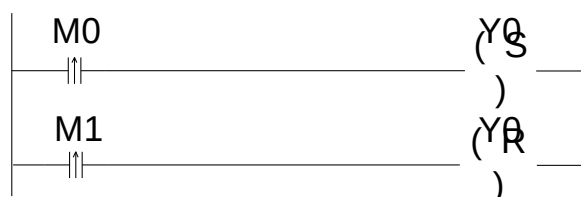
M0 включена, M1 выключена, Y0 не может выводить сигналы.

M0 выключена, M0 включена, Y0 выводит сигналы.

Причина: две разные катушки управляют одной и той же выходной катушкой.



Y0 будет включена в течение одного периода сканирования.



M0 включена, Y продолжит выводить сигналы.

M1 включена, Y0 выключена

Вопрос 7: Проверка и замена батареи таблеточного типа на печатной плате ПЛК.

Ответ 7:

Номинальное напряжение батареи таблеточного типа составляет 3 В. Напряжение можно измерить мультиметром. Если значение в регистре сохранения параметров при отключении питания очень большое, это означает, что батарея разряжена. Замените на новую батарею таблеточного типа. Можно использовать SM5 и SD5 для определения заряда батарей таблеточного типа, чтобы обеспечить их своевременную замену. Подробную информацию см. в Приложении 1 и Приложении 2.

Вопрос 8: Настройки связи в программном обеспечении SCADA

Ответ 8:

Если в программном обеспечении SCADA не выбран ПЛК серии PMP20, выберите протокол Modbus-RTU и осуществляйте связь через порт RS485. См. Руководство по эксплуатации ПЛК серии PMP20, глава 6.

Вопрос 9: Связь по протоколу MODBUS

Ответ 9:

Прежде всего, убедитесь, что клеммы А и В на ПЛК правильно подключены к клеммам связи RS485 других устройств. Параметры порта PORT 2 ПЛК можно изменять следующими способами:

Способ 1: Настройка с помощью команды настройки параметров конфигурации

Список команд представлен в главе 6 «Функции связи» настоящего руководства.

Настройки параметров связи для различных устройств, как правило, отличаются, поэтому важно выбрать правильный режим настройки частоты устройств связи, четко указать соответствующий адрес связи MODBUS и код функции, а для некоторых устройств связи перед отображением заданной частоты требуется ввести специальный рабочий сигнал.

Способ 2: Настройка через панель управления (метод настройки см. в главе 6 «Функции связи» настоящего руководства).

Вопрос 10: Светодиодные индикаторы ПЛК серии PMP20 (PWR/RUN/ERR)

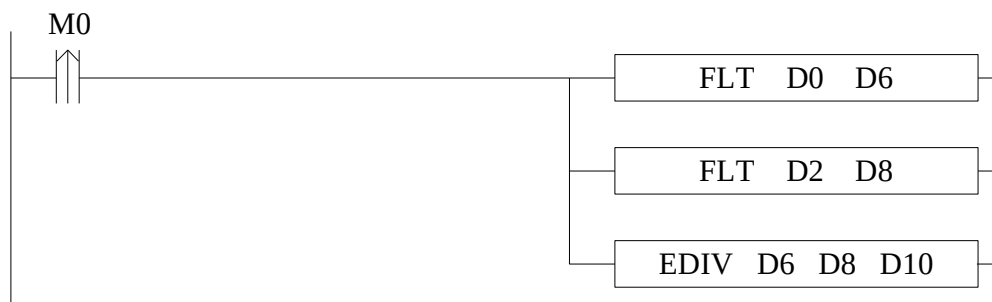
A10:

Светодиодные индикаторы	Неисправность	Метод устранения
Светодиод PWR мигает, остальные светодиоды выключены.	1. Короткое замыкание на плате ввода-вывода 2. Нагрузка слишком высока для питания 24 В 3. Не нажата кнопка RUN для выполнения программы	Проверьте клемму ввода/вывода на наличие короткого замыкания. Проверьте, не слишком ли высокая нагрузка для источника питания 24 В. Убедитесь, что в ПЛК выполняется программа. При необходимости обращайтесь к нам за помощью.
Все три светодиодных индикатора не горят	1. Короткое замыкание на входе питания ПЛК 2. Плата питания ПЛК повреждена	Проверьте вход питания ПЛК. При необходимости обращайтесь к нам за помощью.
Горят светодиоды PWR и ERR.	1. Неустойчивое напряжение на входе ПЛК 2. В программе есть мертвый цикл 3. Неисправности в системе ПЛК	Проверьте напряжение питания, убедитесь, что в программе нет мертвого цикла. Обновите аппаратное обеспечение ПЛК. При необходимости обращайтесь к нам за помощью.

Вопрос 11: Неправильный результат при выполнении операций с плавающей точкой

Ответ 11:

Преобразуйте целое число в число с плавающей точкой. Пример: EDIV D0 D2 D10. Если значение D0 и D2 является целым числом, будет выведена ошибка (D10). Преобразуйте целое число в число с плавающей точкой с помощью представленной ниже команды.

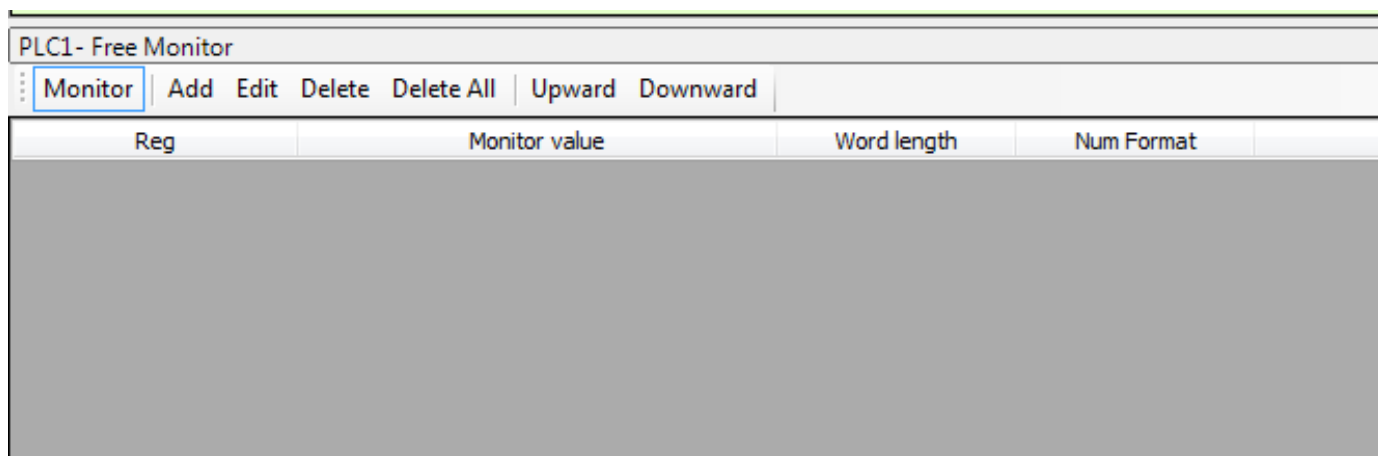


Вопрос 12: Почему числа с плавающей точкой превращаются в беспорядочный код в окне контроля лестничной диаграммы в режиме онлайн?

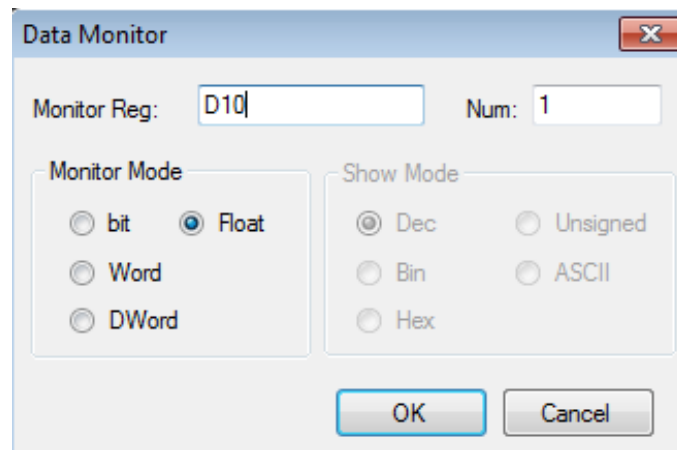
Ответ 12:

Поскольку числа с плавающей точкой нельзя отобразить в окне контроля лестничной диаграммы в режиме онлайн, отслеживайте такие числа через функцию Free Monitor (Свободный контроль).

Откройте ПО PROMPOWER PLC Studio и нажмите online/free monitor (Контроль в режиме онлайн/свободный контроль). Откроется показанное ниже окно:



В этом окне нажмите “Add” (Добавить). Откроется показанное ниже окно. Выберите режим контроля “float” (Числа с плавающей точкой). Введите имя D10 для регистра контроля. Затем нажмите ОК.



Вопрос 13: Почему возникают ошибки данных после использования команд DMUL?

Ответ 13:

Команда операций DMUL представляет собой операцию 32 бита*32 бита = 64 бита, а результат занимает 4 слова, например: EMUL D0 D2 D10, два множителя, оба 32-битные (D1, D0) и (D3, D2), результат — 64 бита (D13, D12, D11, D10), поэтому D10~D13 будут заняты. Если в дальнейшем использовать эти регистры данных, появится ошибка.

Вопрос 14: Почему точка вывода перестает работать после непродолжительной работы ПЛК ?

Ответ 14:

Возможно, выходная клемма ослаблена, проверьте.

Вопрос 15: Почему модуль расширения не работает, хотя индикатор питания включен?

Ответ 15:

Вероятно, модульные выводы и контакты ПЛК или ЦП плохо подключены. Сравните ЦП и модуль расширения перекрестным способом, чтобы выявить неисправности.

Вопрос 16: Почему сигнал поступает, но высокоскоростной счетчик не работает?

Ответ 16:

Для работы высокоскоростного счетчика нужно не только подавать высокоскоростные импульсы на вход высокоскоростного счетчика ПЛК, но и написать соответствующую программу высокоскоростного подсчета с функциональными командами. Подробную информацию см. в главе 5 настоящего руководства.

Вопрос 17: Каковы преимущества языка С по сравнению с лестничной диаграммой?

Ответ 17:

- (1) ПЛК серии RMP20 поддерживают практически все функции языка С. Когда речь идет о сложных математических операциях, преимущества языка С становятся еще более очевидными.
- (2) Повышение защиты (конфиденциальности) программы (в расширенном режиме хранения файлов язык С не может выгружаться).
- (3) Функциональный блок на языке С можно вызывать во многих местах иррегулярных файлов, что значительно повышает эффективность работы программистов.

Вопрос 18: Что такое выходные клеммы А, В ПЛК?

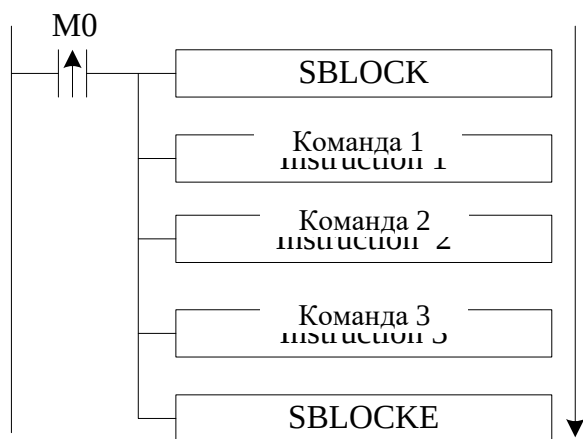
Ответ 18:

Выходные клеммы А, В ПЛК представляют собой клеммы RS485 порта PORT2 на ПЛК.

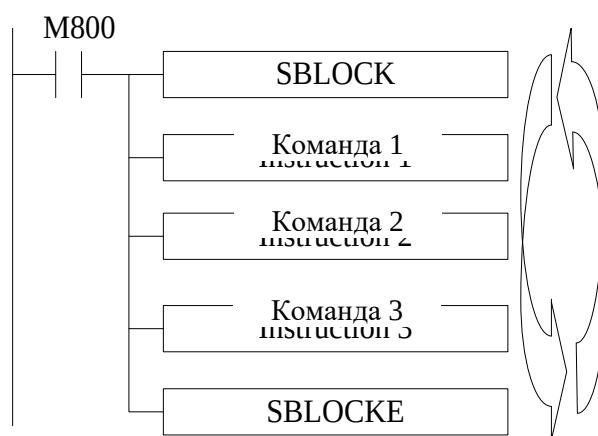
Вопрос 19: В чем разница между условиями запуска функции последовательности BLOCK: запуск по нарастающему фронту и запуск в нормально-замкнутом состоянии?

Ответ 19:

Запуск по нарастающему фронту: при таком запуске функциональный блок выполняется сверху вниз. Запуск в нормально-замкнутом состоянии: при таком запуске блок выполняется сверху вниз, затем происходит возврат вверх и повторное выполнение до тех пор, пока нормально-замкнутое состояние не изменится. Цикл остановится после завершения выполнения последнего блока.



Выполнение сверху вниз, выполнение команд по очереди



Выполнение сверху вниз, циклическое выполнение команды

Вопрос 20: Какие режимы загрузки поддерживаются в ПЛК серии RMP20 и каковы их характеристики?

Ответ 20:

ПЛК серии RMP20 поддерживают три режима загрузки:

Общий режим загрузки

В этом режиме можно легко загрузить программу с компьютера на ПЛК или выгрузить программу из ПЛК на компьютер. Этот режим очень удобно использовать при отладке оборудования.

Режим загрузки с защитой паролем

Для ПЛК можно установить пароль. При загрузке программы из ПЛК в компьютер нужно будет ввести правильный пароль. В опции меню Advanced password (Дополнительные настройки пароля) можно проверить, что задана функция "download the program needs to be decrypted first" (Загружаемая программа должна быть сначала расшифрована) (Примечание: Эта операция опасная, так как ПЛК будет заблокирован, если вы забудете пароль!). Этот режим загрузки подходит пользователям, для которых важно защитить программу устройства и иметь возможность вызывать ее в любое время.

Секретный режим загрузки

В этом режиме программу с компьютера можно загружать в ПЛК, независимо от того, каким способом пользователь выгрузил программу из ПЛК в компьютер. Пользовательскую программу можно загружать конфиденциально, что потре-

бует меньше внутренних ресурсов ПЛК, значительно увеличит объем программы на ПЛК и может обеспечить более высокую скорость загрузки. Но после этого режима загрузки программу уже невозможно будет откатить в предыдущее состояние.

Вопрос 21: Какие методы обеспечения конфиденциальности используются в ПЛК серии RMP20?

Ответ 21:

В ПЛК PROMPOWER есть три способа обеспечения конфиденциальности: (1) импорт и экспорт загруженных файлов; (2) секретный режим загрузки; (3) режим загрузки с защитой паролем.

Импорт и экспорт загруженных файлов: После сохранения программы ПЛК таким способом пользователи смогут загружать и использовать программу, но они не могут просматривать и редактировать ее.

Секретный режим загрузки: После секретной загрузки в ПЛК программа и данные не будут выгружены в ПЛК, что означает «программы не существует».

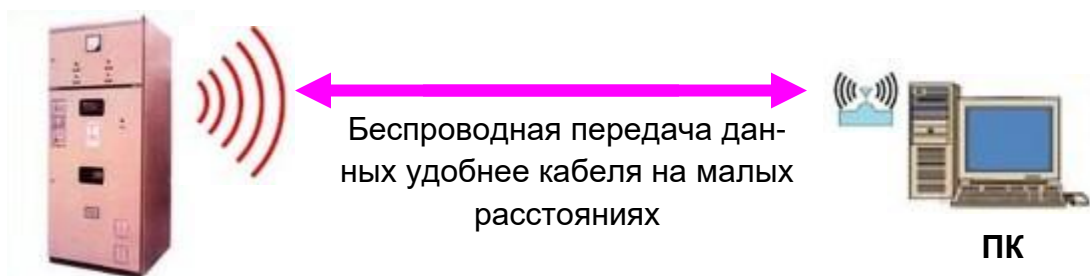
Режим загрузки с защитой паролем: При загрузке программы под паролем в ПЛК нужно ввести правильный пароль; если выбрать опцию "download the program needs to be decrypted first" (Загружаемая программа должна быть сначала расшифрована), вам также нужно будет ввести правильный пароль при загрузке новой программы в ПЛК. В этом режиме нельзя изменять время ПЛК, а конфиденциальность повышается.

Вопрос 22: В чем преимущество замены кабеля загрузки данных DVP на Bluetooth в ПЛК серии RMP20?

Ответ 22:

Функция Bluetooth в ПЛК серии RMP20 позволяет выполнять загрузку и выгрузку программ ПЛК, мониторинг и онлайн-моделирование программного обеспечения конфигурации Twin. Bluetooth можно использовать вместо кабеля при передаче данных.

Примечание: COM-Bluetooth подходит только для ПЛК PROMPOWER.



ПЛК серии PMP20 и COM-Bluetooth в шкафу управления

Вопрос 23: Для чего нужна функция косвенной адресации в ПЛК серии PMP20?

Ответ 23:

Суффикс смещения после катушек и регистров данных (например, X3[D100], M10[D100], D0[D100]) дает возможность реализовать функцию косвенной адресации; например, D100=9, X3[D100] представляет X14, M10[D100] представляет M19, D0[D100] представляет D9. Эта функция обычно используется во множестве операций с битами и регистрами, а также во время хранения.

Вопрос 24: Как ПЛК серии PMP20 подключается к сети?

Ответ 24:

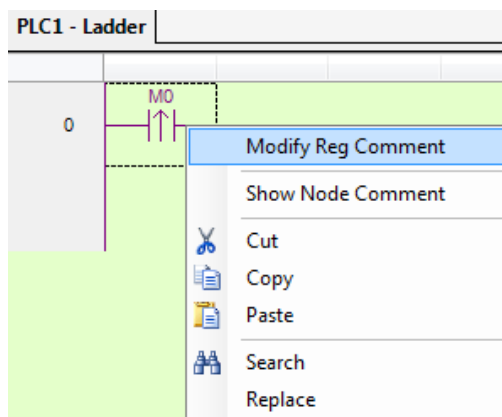
ПЛК серии PMP20 можно подключить к сети с помощью модулей расширения PROMPOWER T-BOX, G-BOX, W-BOX, S-BOX, A-BOX или плат расширения BD, которые имеют собственные характеристики связи. Подробную информацию см. в руководстве пользователя для модуля связи или платы BD.

Вопрос 25: Как добавлять примечания к программным элементам и строкам в программе PROMPOWER PLC Studio?

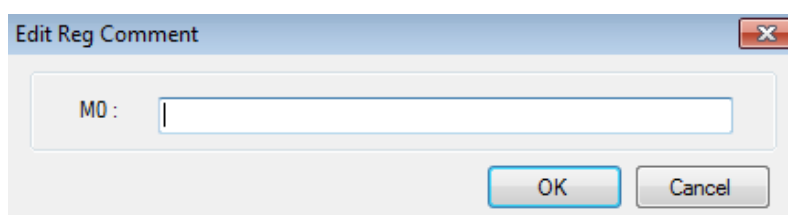
Ответ 25:

Примечания к программным элементам

Откройте программное обеспечение PROMPOWER PLC Studio, наведите указатель мыши на соответствующий программный элемент и щелкните по нему правой кнопкой мыши. Откроется следующее меню:

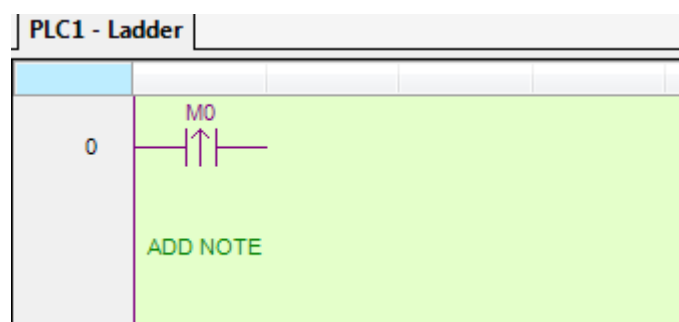
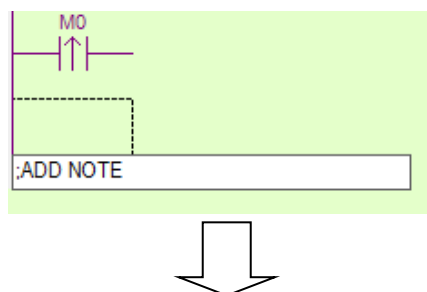


Нажмите “Modify reg comment” (Изменить комментарий по регистру), чтобы добавить примечания к элементу в показанном ниже окне:



Примечания к строкам

Примечание к строке начинается с “;”. Дважды щелкните по строке, затем введите точку с запятой и содержимое примечания.



Вопрос 26: Предусмотрена ли функция часов? Почему часы показывают неточное время?

Ответ 26:

Функция часов в ПЛК серии RMP20 является дополнительной опцией. Если вы хотите приобрести ПЛК с функцией часов, пожалуйста, уточните это при размещении заказа. В стандартной заводской комплектации ПЛК не поддерживает функцию часов.

При использовании ПЛК с функцией часов проверьте, является ли значение в регистрах SD13-SD19 десятичным. Если нет, нужно будет преобразовать его в десятичное значение с помощью команд BIN или TRD.

Часы ПЛК серии RMP20 имеют некоторую погрешность. Она составляет около ± 5 минут в месяц. Часы можно калибровать с помощью ЧМИ или в программе ПЛК.

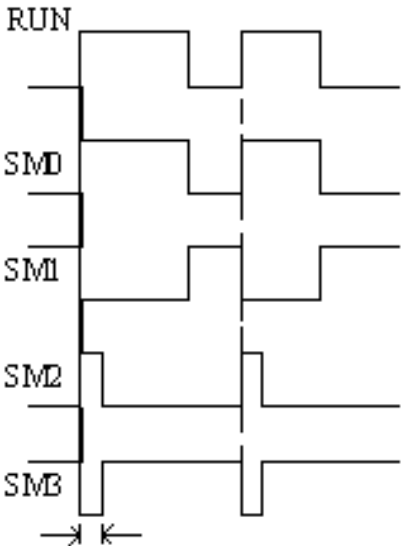
Приложение

Специальные программные компоненты

В этом приложении описаны функции специальных программных элементов ПЛК серии PMP20, регистры данных, FlashROM, и представлены таблицы распределения адресов модулей расширения.

Приложение 1 Специальное вспомогательное реле

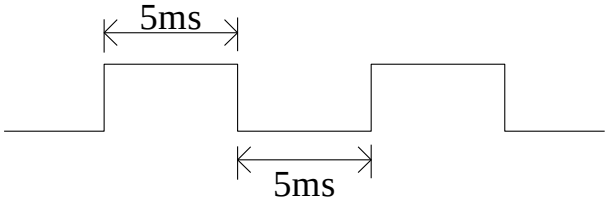
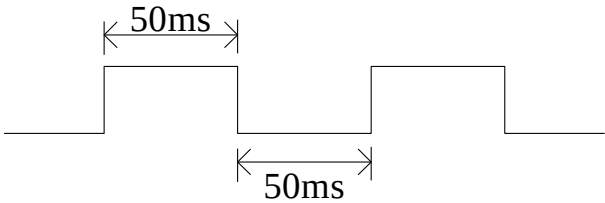
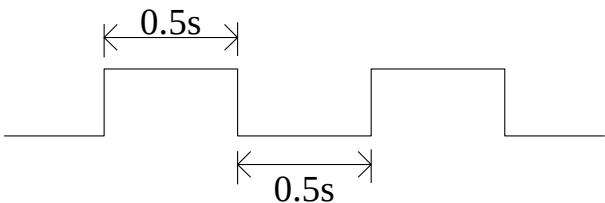
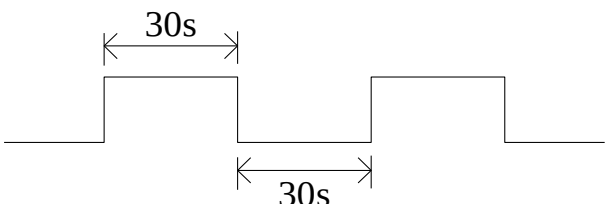
Начальное состояние (SM0-SM7)

Идентификатор	Функция	Описание	
SM000	Катушка включена во время работы ПЛК	 Цикл сканирования	SM000 поддерживает включенное состояние во время работы ПЛК
SM001	Катушка выключена во время работы ПЛК		SM001 поддерживает выключенное состояние во время работы ПЛК
SM002	Катушка с начальным положительным импульсом		SM002 включен в первом цикле сканирования
SM003	Катушка с начальным отрицательным импульсом		SM003 выключен в первом цикле сканирования
SM004	Ошибка во время работы ПЛК	Если SM4 находится во включенном состоянии, это указывает на наличие ошибки в работе ПЛК. (Прошивки версии 3.4.5 и более поздних версий поддерживают эту функцию ПЛК)	
SM005	Катушка выдает сигнал разряда батареи	Если напряжение батареи упадет ниже 2,5 В, SM5 включится (в этом случае замените батарею как можно скорее, иначе данные не сохранятся).	



SM007	Признак занятости при загрузке в режиме онлайн	Бит признака занятости при загрузке во флэш-память в режиме онлайн
-------	--	--

Тактовые импульсы (SM11-SM14)

Идентификатор	Функция	Описание
SM011	Частотный цикл 10 мс	
SM012	Частотный цикл 100 мс	
SM013	Частотный цикл 1 с	
SM014	Частотный цикл 1 мин	

Маркировка (SM20-SM22)

Иден-тифи-катор	Функция	Описание
SM020	Нулевой бит	SM020 включен, когда результат операции «плюс/минус» равен 0
SM021	Бит займа	SM021 включен, когда результатом операции «минус» является переполнение данными
SM022	Бит переноса	SM022 включен, когда результатом операции «плюс» является переполнение данными

Режим ПК (SM30-M34)

Иден-тифи-катор	Функция	Описание
SM030	Инициализация ПЛК	Возврат к заводским настройкам
SM032	Сброс регистра сохранения параметров	Когда SM032 включен, распределение памяти ON/OFF для НМ, НS и текущие значения НТ, НС, НD будут сброшены.
SM033	Сброс пользовательской программы	Когда SM033 включен, все программы пользователя ПЛК будут сброшены..
SM034	Запрет всех выходных данных	Когда SM034 включен, все внешние контакты ПЛК будут отключены.

Лестничная диаграмма

Иден-тифи-катор	Функция	Описание
SM040	Процесс выполняется	Включен, когда процесс выполняется



Запрет прерываний (SM50-SM90)

Иден-тифи-катор	Адрес	Функция	Описание
SM050	I0000/I0001	Запрет прерываний на входе 0	После выполнения команды EI прерывание на входе не может работать независимо от M, даже если прерывание разрешено. Например: когда SM050 включен, I0000/I0001 запрещены.
SM051	I0100/I0101	Запрет прерываний на входе 1	
SM052	I0200/I0201	Запрет прерываний на входе 2	
SM053	I0300/I0301	Запрет прерываний на входе 3	
SM054	I0400/I0401	Запрет прерываний на входе 4	
.....			
SM069	I1900/I1901	Запрет прерываний на входе 19	
SM070	I40**	Запрет временных прерываний 0	После выполнения команды EI временное прерывание не может работать независимо от M, даже если прерывание разрешено.
SM071	I41**	Запрет временных прерываний 1	
SM072	I42**	Запрет временных прерываний 2	
SM073	I43**	Запрет временных прерываний 3	
SM074	I44**	Запрет временных прерываний 4	
.....			
SM089	I59**	Запрет временных прерываний 19	
SM090		Запрет всех прерываний	Запрет всех прерываний

Высокоскоростной кольцевой счетчик (SM99)

Адрес	Функция	Примечание
SM099	Высокоскоростной кольцевой счетчик включен	SM99 включен, SD99 добавляет по одному каждые 0,1 мс, цикл от 0 до 32767

Работа высокоскоростного счетчика завершена (SM100-SM109)

Адрес	Функция	Примечание
SM100	Признак завершения счета HSC0 (100 сегментов)	
SM101	Признак завершения счета HSC2 (100 сегментов)	
SM102	Признак завершения счета HSC4 (100 сегментов)	
SM103	Признак завершения счета HSC6 (100 сегментов)	
SM104	Признак завершения счета HSC8 (100 сегментов)	
SM105	Признак завершения счета HSC10 (100 сегментов)	
SM106	Признак завершения счета HSC12 (100 сегментов)	
SM107	Признак завершения счета HSC14 (100 сегментов)	
SM108	Признак завершения счета HSC16 (100 сегментов)	
SM109	Признак завершения счета HSC18 (100 сегментов)	

Направление работы высокоскоростного счетчика (SM110-SM119)

Адрес	Функция	Примечание
SM110	Признак направления HSC0	
SM111	Признак направления HSC2	
SM112	Признак направления HSC4	
SM113	Признак направления HSC6	
SM114	Признак направления HSC8	
SM115	Признак направления HSC10	
SM116	Признак направления HSC12	
SM117	Признак направления HSC14	
SM118	Признак направления HSC16	
SM119	Признак направления HSC18	

Ошибка высокоскоростного счетчика (SM120-SM129)

Адрес	Функция	Примечание
SM120	Признак ошибки HSC0	
SM121	Признак ошибки HSC2	
SM122	Признак ошибки HSC4	
SM123	Признак ошибки HSC6	
SM124	Признак ошибки HSC8	
SM125	Признак ошибки HSC10	

SM126	Признак ошибки HSC12	
SM127	Признак ошибки HSC14	
SM128	Признак ошибки HSC16	
SM129	Признак ошибки HSC18	

Признак переполнения высокоскоростного счетчика (SM130-SM139)

Адрес	Функция	Примечание
SM130	Признак переполнения HSC0	
SM131	Признак переполнения HSC2	
SM132	Признак переполнения HSC4	
SM133	Признак переполнения HSC6	
SM134	Признак переполнения HSC8	
SM135	Признак переполнения HSC10	
SM136	Признак переполнения HSC12	
SM137	Признак переполнения HSC14	
SM138	Признак переполнения HSC16	
SM139	Признак переполнения HSC18	

Связь (SM140-SM193)

	Адрес	Функция	Примечание
Последовательный порт 0	SM140	Признак выполнения команды Modbus	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM141	Признак выполнения команды X-NET	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM142	Признак отправки при связи в свободном формате	Включен, когда начинается выполнение команды Выключен, когда выполнение команды завершено
	SM143	Признак завершения получения при связи в свободном формате	Включен при получении кадра данных или тайм-ауте приема данных Требуется выключение пользовательской программы
Последовательный порт 1	SM150	Признак выполнения команды Modbus	Аналогично SM140
	SM151	Признак выполнения команды X-NET	Аналогично SM141
	SM152	Признак отправки при связи в свободном формате	Аналогично SM142
	SM153	Признак завершения получения при связи в свободном формате	Аналогично SM143
Последовательный порт 2	SM160	Признак выполнения команды Modbus	Аналогично SM140

	SM161	Признак выполнения команды X-NET	Аналогично SM141
	SM162	Признак отправки при связи в свободном формате	Аналогично SM142
	SM163	Признак завершения получения при связи в свободном формате	Аналогично SM143
Последовательный порт 3	SM170	Признак выполнения команды Modbus	Аналогично SM140
	SM171	Признак выполнения команды X-NET	Аналогично SM141
	SM172	Признак отправки при связи в свободном формате	Аналогично SM142
	SM173	Признак завершения получения при связи в свободном формате	Аналогично SM143
Последовательный порт 4	SM180	Признак выполнения команды Modbus	Аналогично SM140
	SM181	Признак выполнения команды X-NET	Аналогично SM141
	SM182	Признак отправки при связи в свободном формате	Аналогично SM142
	SM183	Признак завершения получения при связи в свободном формате	Аналогично SM143
Последовательный порт 5	SM190	Признак выполнения команды Modbus	Аналогично SM140
	SM191	Признак выполнения команды X-NET	Аналогично SM141
	SM192	Признак отправки при связи в свободном формате	Аналогично SM142
	SM193	Признак завершения получения при связи в свободном формате	Аналогично SM143

Функция последовательности BLOCK (SM300-SM399)

Иденти- фикатор	Функция	Описание
SM300	Признак выполнения BLOCK1	SM300 будет включен во время выполнения block1
SM301	Признак выполнения BLOCK2	SM301 будет включен во время выполнения block2
SM302	Признак выполнения BLOCK3	SM302 будет включен во время выполнения block3
SM303	Признак выполнения BLOCK4	SM303 будет включен во время выполнения block4
SM304	Признак выполнения BLOCK5	SM304 будет включен во время выполнения block5
SM305	Признак выполнения BLOCK6	SM305 будет включен во время выполнения block6
.....		
SM396	Признак выполнения BLOCK97	SM396 будет включен во время выполнения block97
SM397	Признак выполнения BLOCK98	SM397 будет включен во время выполнения block98
SM398	Признак выполнения BLOCK99	SM398 будет включен во время выполнения block99
SM399	Признак выполнения BLOCK100	SM399 будет включен во время выполнения block100

Проверка ошибок (SM400-SM415)

Идентификатор	Функция	Описание
SM400	Ошибка ввода/вывода	Светодиод ERR продолжает гореть, ПЛК не работает и не выдает импульсы. Проверить при включенном питании
SM401	Ошибка связи с модулем расширения	
SM402	Ошибка связи с BD	
.....		
SM405	Пользовательская программа отсутствует	Неправильная проверка внутреннего кода
SM406	Ошибка в пользовательской программе	Неправильная проверка кода реализации или таблицы настройки
SM407	Ошибка при проверке SSFD	Светодиод ERR продолжает гореть, ПЛК не работает и не выдает импульсы. Проверить при включенном питании
SM408	Ошибка памяти	Стирание из/запись во флэш-память невозможны
SM409	Ошибка вычисления	
SM410	Переполнение при смещении	Смещение превышает диапазон программных элементов
SM411	Переполнение FOR-NEXT	Сброс при включении питания, либо пользователи могут выполнить сброс вручную.
SM412	Заполнение неправильными данными	Если при смещении регистра происходит переполнение данными, возвращаемое значение будет значением SM372
SM413	Ошибка контрольной суммы при шифровании	
SM414	Ошибка в данных FLASH	

SM415	Бит признака ошибки часов реального времени RTC	Не удалось выполнить проверку времени и даты в часах реального времени
-------	---	--

Сообщения об ошибках (SM450-SM463)

Идентификатор	Функция	Описание
SM450	Проверка ошибок системы	
SM451	Признак прерывания при неисправности аппаратного обеспечения	
.....		
SM453	Ошибка карты памяти SD	
SM454	Питание отключено	
SM455	Ошибка данных в памяти при выключении питания	
SM456	Бит признака ошибки при загрузке в режиме онлайн	
.....		
SM460	Идентификатор модуля расширения не совпадает	
SM461	Идентификатор модуля BD/ED не совпадает	
SM462	Превышено время связи с модулем расширения	
SM463	Превышено время связи с модулем BD/ED	
SM464	Переполнение буфера данных при связи с модулем расширения	
SM465	Переполнение буфера данных при связи с модулем BD/ED	



Модули расширения, состояние BD (SM500)

Идентификатор	Функция	Описание
SM500	Считывание данных о состоянии модуля завершено	

Приложение 2 Специальный регистр данных

Батарея (SD5-SD7)

Идентификатор	Функция	Описание
SD005	Регистр батареи	Отображается 100, если напряжение батареи составляет 3 В, а если напряжение батареи ниже 2,5 В, будет отображаться 0. Это означает, что требуется немедленная замена на новую батарею, иначе данные будут потеряны при отключении питания ПЛК.

Тактовые импульсы (SD10-SD019)

Идентификатор	Функция	Описание
SD010	Текущий цикл сканирования	100 мкс, единица измерения — мкс
SD011	Минимальное время сканирования	100 мкс, единица измерения — мкс
SD012	Максимальное время сканирования	100 мкс, единица измерения — мкс
SD013	Секунда (тактовый импульс)	0-59 (код BCD)
SD014	Минута (тактовый импульс)	0-59 (код BCD)
SD015	Час (тактовый импульс)	0-23 (код BCD)
SD016	День (тактовый импульс)	1-31 (код BCD)

SD017	Месяц (такты́ый импульс)	1-12 (код BCD)
SD018	Год (такты́ый импульс)	2000-2099 (код BCD)
SD019	Неделя (такты́ый импульс)	0 (воскресе́нье) - 6 (суббота) (код BCD)

Признак (SD020-SD031)

Иден-тифи-катор	Функция	Примечание
SD020	Модель/тип	
SD021	Модель (младшие 8 бит), серия (старшие 8 бит)	
SD022	Совместимая версия системы (младшие биты), версия системы (старшие биты)	
SD023	Совместимая версия модели (младшие биты), версия модели (старшие биты)	
SD024	Информация о модели	
SD025	Информация о модели	
SD026	Информация о модели	
SD027	Информация о модели	
SD028	Совместимая версия программного обеспечения	
SD029	Совместимая версия программного обеспечения	
SD030	Совместимая версия программного обеспечения	
SD031	Совместимая версия программного обеспечения	

Лестничная диаграмма (SD040)

Иден-тифи-катор	Функция	Описание
SD40	Признак выполнения процесса S	

Иден-тифи-катор	Функция	Описание
SD99	Высокоскоростной кольцевой счетчик	Когда SM99 включен, SD99 добавляет по единице каждые 0,1 мс, цикл от 0 до 32767

Лестничная диаграмма (SD099)

Высокоскоростной счетчик (SD100-SD109)

Иден-тифи-катор	Функция	Описание
SD100	Текущий сегмент (номер сегмента n)	HSC00
SD101	Текущий сегмент (номер сегмента n)	HSC02
SD102	Текущий сегмент (номер сегмента n)	HSC04
SD103	Текущий сегмент (номер сегмента n)	HSC06
SD104	Текущий сегмент (номер сегмента n)	HSC08
SD105	Текущий сегмент (номер сегмента n)	HSC10
SD106	Текущий сегмент (номер сегмента n)	HSC12
SD107	Текущий сегмент (номер сегмента n)	HSC14
SD108	Текущий сегмент (номер сегмента n)	HSC16
SD109	Текущий сегмент (номер сегмента n)	HSC18

Ошибка высокоскоростного счетчика (SD120-SD129)

Идентификатор	Функция	Примечание
SD120	Информация об ошибке HSC0	
SD121	Информация об ошибке HSC2	
SD122	Информация об ошибке HSC4	
SD123	Информация об ошибке HSC6	
SD124	Информация об ошибке HSC8	
SD125	Информация об ошибке HSC10	
SD126	Информация об ошибке HSC12	
SD127	Информация об ошибке HSC14	
SD128	Информация об ошибке HSC16	
SD129	Информация об ошибке HSC18	

Связь (SD140-SD199)

	Идентификатор	Функция	Примечание
Последовательный порт 0	SD140	Результаты выполнения команды команды считывания и записи Modbus	0: правильно 100: ошибка при приеме данных 101: превышено время приема 180: ошибка CRC 181: ошибка LRC 182: ошибка станции 183: переполнение пересылочного буфера 400: ошибка кода функции 401: ошибка адреса 402: ошибка длины 403: ошибка в данных 404: ведомая станция занята 405: ошибка памяти (Ошибка стирания FLASH-памяти)
	SD141	Результат связи по шине X-Net	0: правильно 1: превышено время связи 2: ошибка памяти 3: ошибка CRC при приеме данных

	Иден-тифи-катор	Функция	Примечание
	SD142	Результат отправки при связи в свободном формате	0: правильно 410: переполнение пересылочного буфера при связи в свободном формате
	SD143	Результат получения при связи в свободном формате	0: правильно 410: переполнение при отправке длины данных 411: прием коротких данных 412: прием длинных данных 413: ошибка при приеме данных 414: превышено время приема 415: стартовый символ отсутствует 416: стоповый символ отсутствует
	SD144	Количество полученных данных при связи в свободном формате	В байтах, стартовые и стоповые символы отсутствуют символа
			
	SD149		
Последовательный порт 1	SD150	Результаты выполнения команды считывания и записи Modbus	0: правильно 100: ошибка при приеме данных 101: превышено время приема 180: ошибка CRC 181: ошибка LRC 182: ошибка станции 183: переполнение пересылочного буфера 400: ошибка кода функции 401: ошибка адреса 402: ошибка длины 403: ошибка в данных 404: ведомая станция занята 405: ошибка памяти (Ошибка стирания FLASH-памяти)
	SD151	Результат связи по шине X-Net	0: правильно 1: превышено время связи 2: ошибка памяти 3: ошибка CRC при приеме данных
	SD152	Результат отправки при связи в свободном формате	0: правильно 410: переполнение пересылочного буфера при связи в свободном формате

	Иден-тифи-катор	Функция	Примечание
	SD153	Результат получения при связи в свободном формате	0: правильно 410: переполнение при отправке длины данных 411: прием коротких данных 412: прием длинных данных 413: ошибка при приеме данных 414: превышено время приема 415: стартовый символ отсутствует 416: стоповый символ отсутствует
	SD154	Количество полученных данных при связи в свободном формате	В байтах, стартовые и стоповые символы отсутствуют
			
	SD159		
Последовательный порт 2	SD160	Результаты выполнения команды считывания и записи Modbus	0: правильно 100: ошибка при приеме данных 101: превышено время приема 180: ошибка CRC 181: ошибка LRC 182: ошибка станции 183: переполнение пересылочного буфера 400: ошибка кода функции 401: ошибка адреса 402: ошибка длины 403: ошибка в данных 404: ведомая станция занята 405: ошибка памяти (Ошибка стирания FLASH-памяти)
	SD161	Результат связи по шине X-Net	0: правильно 1: превышено время связи 2: ошибка памяти 3: ошибка CRC при приеме данных
	SD162	Результат отправки при связи в свободном формате	0: правильно 410: переполнение пересылочного буфера при связи в свободном формате
	SD163	Результат получения при связи в свободном формате	0: правильно 410: переполнение при отправке длины данных 411: прием коротких данных

	Иден- тифи- катор	Функция	Примечание
			412: прием длинных данных 413: ошибка при приеме данных 414: превышено время приема 415: стартовый символ отсутствует 416: стоповый символ отсутствует
	SD164	Количество полученных данных при связи в свободном формате	В байтах, стартовые и стоповые сим- волы отсутствуют символа
			
	SD169		
Последо- ватель-	SD170 ~ SD179		
Последо- ватель-	SD180 ~ SD189		
Последо- ватель-	SD190 ~ SD199		

Функция последовательности BLOCK (SD300-SD399)

Иден-тифи-катор	Функция	Описание
SD300	Выполнение команды BLOCK1	Это значение будет использоваться при контроле BLOCK
SD301	Выполнение команды BLOCK2	Это значение будет использоваться при контроле BLOCK
SD302	Выполнение команды BLOCK3	Это значение будет использоваться при контроле BLOCK
SD303	Выполнение команды BLOCK4	Это значение будет использоваться при контроле BLOCK
SD304	Выполнение команды BLOCK5	Это значение будет использоваться при контроле BLOCK
SD305	Выполнение команды BLOCK6	Это значение будет использоваться при контроле BLOCK
.....		
SD396	Выполнение команды BLOCK97	Это значение будет использоваться при контроле BLOCK
SD397	Выполнение команды BLOCK98	Это значение будет использоваться при контроле BLOCK
SD398	Выполнение команды BLOCK99	Это значение будет использоваться при контроле BLOCK
SD399	Выполнение команды BLOCK100	Это значение будет использоваться при контроле BLOCK

Проверка ошибок (SD400-SD413)

Иден-тифи-катор	Функция	Примечание
SD400		
SD401	Ошибка связи с модулем расширения №	Это означает, что в модуле № произошла ошибка
SD402	Ошибка связи с модулем BD/ED №	

SD403	Тип ошибки FROM/TO	
SD404	Тип ошибки PID	
SD405	Пользовательская программа отсутствует	
SD406	Тип ошибки в пользовательской программе	
SD407	Тип ошибки SSDF	
SD408	Тип ошибки стирания FLASH-памяти	
SD409	Код ошибки вычисления	1: ошибка при делении на 0 2: MRST, MSET Адрес переднего операнда меньше адреса заднего операнда 3: Биты данных ENCO, DECO команд кодирования и декодирования превышают предел. 4: Ошибка в коде BDC 7: Ошибка в знаке радикала
SD410	Номер регистра смещения D, когда смещение пересекает границу	
SD411		
SD412	Заполнение неправильными данными (16 младших битов)	
SD413	Заполнение неправильными данными (16 старших битов)	
SD414	Тип ошибок данных во Flash-регистре	
SD415	Тип ошибки часов реального времени RTC	1: Низкое напряжение питания часов реального времени, данные должны быть перезаписаны. 2: RTC перезаписывает данные, а микросхема часов не реагирует на сигнал ACK 3: Запись недопустимых данных о времени и дате

Проверка ошибок (SD450-SD465)

Иден-тифи-катор	Функция	Описание
SD450	1: Работа сторожевого таймера (по умолчанию: 200 мс) 2: Не удалось использовать блок управления 3: Недействительный адрес посещения	
SD451	Тип ошибки аппаратного обеспечения: 1: Ошибка регистра 2: Ошибка шины 3: Ошибка из-за неправильного использования	
SD452	Ошибка аппаратного обеспечения	
SD453	Ошибка карты памяти SD	
SD454	Время отключения питания	
SD455		
SD456		
SD460	Идентификатор модуля расширения не совпадает	
SD461	Идентификатор модуля BD/ED не совпадает	
SD462	Превышено время связи с модулем расширения	
SD463	Превышено время связи с модулем BD/ED	
SD464	Переполнение данных связи с номером модуля расширения	
SD465	Переполнение данных связи с номером модуля BD/ED	

Модули расширения, состояние BD (SD500-SD516)

Идентификатор	Функция	Описание
SD500	Номер модуля Модули расширения: #10000~10015 BD: #20000~20001 ED: #30000	
SD501 ~516	Модуль расширения, состояние BD/ED	16 регистров

Информация о модуле (SD520-SD823)

Идентификатор	Функция	Описание	Примечание
SD520-SD535	Информация о модуле расширения	Модуль расширения 1	Каждый модуль расширения, BD, ED занимает 16 регистров
.....			
SD760-SD775	Информация о модуле расширения	Модуль расширения 16	
SD776-SD791	Информация о модуле BD	Модуль BD 1	
SD792-SD807	Информация о модуле BD	Модуль BD 2	
SD808-SD823	Информация о модуле ED	Модуль ED 1	

Информация об ошибках модуля расширения

Иден-тифи-катор	Функция	Описание	
SD860	Количество ошибок при считывании данных из модуля		Модуль расширения 1
SD861	Типы ошибок при считывании данных из модуля	Ошибка адреса модуля. Ошибка длины данных, полученных модулем. Ошибка четности CRC модуля, когда ПЛК принимает данные. Ошибка идентификатора модуля. Ошибка при превышении времени работы модуля.	
SD862	Количество ошибок при записи данных в модуль		
SD863	Типы ошибок при записи данных в модуль		
SD864	Количество ошибок при считывании данных из модуля		Модуль расширения 2
SD865	Типы ошибок при считывании данных из модуля	Ошибка адреса модуля. Ошибка длины данных, полученных модулем. Ошибка четности CRC модуля, когда ПЛК принимает данные. Ошибка идентификатора модуля. Ошибка при превышении времени работы модуля.	
SD866	Количество ошибок при записи данных в модуль		
SD867	Типы ошибок при записи данных в модуль		
.....			

Иден-тифи-катор	Функция	Описание	
SD920	Количество ошибок при считывании данных из модуля		Модуль расширения 16
SD921	Типы ошибок при считывании данных из модуля	Ошибка адреса модуля. Ошибка длины данных, полученных модулем. Ошибка четности CRC модуля, когда ПЛК принимает данные. Ошибка идентификатора модуля. Ошибка при превышении времени работы модуля.	
SD922	Количество ошибок при записи данных в модуль		
SD923	Типы ошибок при записи данных в модуль		
SD930	Продолжительность считывания/записи текущей ED	100 мкс, единица измерения: мкс	Модуль ED 1
SD931	Минимальная продолжительность считывания/записи текущей ED		
SD932	Максимальная продолжительность считывания/записи текущей ED		
SD933	Длительность ошибки считывания/записи ED		
SD934	Общее количество считываний/записи ED (16 младших битов)		

Иден-тифи-катор	Функция	Описание	
SD935	Общее количество считываний/записи ED (16 старших битов)		
SD936	Продолжительность считывания/записи текущей BD1	100 мкс, единица измерения: мкс	Модуль BD 1
SD937	Минимальная продолжительность считывания/записи BD1		
SD938	Максимальная продолжительность считывания/записи BD1		
SD939	Количество ошибок считывания/записи BD1		
SD940	Общее количество считанных и записанных BD1 (16 младших битов)		
SD941	Общее количество считанных и записанных BD1 (16 старших битов)		
SD942	Продолжительность считывания/записи текущей BD2		Модуль BD 2
SD943	Минимальная продолжительность считывания/записи BD2 , считывание типа ошибок модуля		
SD944	Максимальная продолжительность считывания/записи BD1		

Иден-тифи-катор	Функция	Описание	
SD945	Количество ошибок считыва-ния/записи BD1		
SD946	Общее количество считанных и записанных BD2 (16 младших битов)		
SD947	Общее количество считанных и записанных BD2 (16 старших битов)		

Информация о версии (SD990-SD993)

Иден-тифи-катор	Функция	Описание	Примечание
SD990	Дата создания версии про-шивки	16 младших битов	
SD991	Дата компиляции версии про-шивки	16 старших битов	
SD992	Дата компиляции версии вен-тильной матрицы, программ-руемой пользователем	16 младших битов	
SD993	Дата компиляции версии вен-тильной матрицы, программ-руемой пользователем	16 старших битов	

Специальная функция (HSD50-HSD60)

Идентификатор	Функция	Описание
HSD50	Сохранение времени записи данных после сбоя питания ^{※1}	Одно слово, единица измерения: 1 мс
HSD51	Обнаружение сбоя питания	Время работы ЦП после сбоя питания, единица измерения: 100 мкс
HSD52	Последнее время работы ПЛК (16 младших битов)	Двойное слово, единица измерения: 1 с
HSD53	Последнее время работы ПЛК (16 старших битов)	
HSD54	Текущее время работы ПЛК (16 младших битов)	Двойное слово, единица измерения: 1 с
HSD55	Текущее время работы ПЛК (16 старших битов)	
HSD58	Количество стертых записей во Flash-регистре	

Регистрация ошибок (HSD80-HSD179)

Идентификатор	Функция	Описание
HSD79	Значение индекса списка ошибок	
HSD80-HSD84	Сообщение об ошибках изделия 1	
HSD85-HSD89	Сообщение об ошибках изделия 2	
HSD90-HSD94	Сообщение об ошибках изделия 3	

HSD95-HSD99	Сообщение об ошибках изделия 4	(1) Для этой функции потребуется программное обеспечение для программирования версии 3.5.3 (20190326) и более поздних версий. () Режим перемещений Н поддерживает до 20 сообщений об ошибках, а режим перемещений С поддерживает до 4 сообщений об ошибках.
HSD100-HSD104	Сообщение об ошибках изделия 5	
HSD105-HSD109	Сообщение об ошибках изделия 6	
HSD110-HSD114	Сообщение об ошибках изделия 7	
HSD115-HSD119	Сообщение об ошибках изделия 8	
HSD120-HSD124	Сообщение об ошибках изделия 9	
HSD125-HSD129	Сообщение об ошибках изделия 10	
HSD130-HSD134	Сообщение об ошибках изделия 11	
HSD135-HSD139	Сообщение об ошибках изделия 12	
HSD140-HSD144	Сообщение об ошибках изделия 13	
HSD145-HSD149	Сообщение об ошибках изделия 14	
HSD150-HSD154	Сообщение об ошибках изделия 15	
HSD155-HSD159	Сообщение об ошибках изделия 16	
HSD160-HSD164	Сообщение об ошибках изделия 17	

HSD165-HSD169	Сообщение об ошибках изделия 18	
HSD170-HSD174	Сообщение об ошибках изделия 19	
HSD175-HSD179	Сообщение об ошибках изделия 20	
HSD180-HSD184	Сообщение об ошибках изделия 21	В режиме перемещений Н количество сообщений об ошибках увеличено до 76 (прошивка версии 3.7.2 и более поздних версий)
HSD185-HSD189	Сообщение об ошибках изделия 22	
.....		
HSD450-HSD454	Сообщение об ошибках изделия 75	
HSD455-HSD459	Сообщение об ошибках изделия 76	

Примечания:

✱ 1: HSD50 — «Сохранение времени записи данных после сбоя питания» в версии 3.7.2 и более поздних версиях.



Приложение 3 Перечень специальных FLASH регистров

Специальный FLASH регистр данных SFD

Примечание: * Работает только после повторного включения питания ПЛК

Фильтрация входных импульсов

Иден-тифи-катор	Функция	Описание
SFD0*	Время фильтрации входных импульсов	
SFD2*	Время работы сторожевого таймера, значение по умолчанию 200 мс	

Настройка специальной функции

Иден-тифи-катор	Функция	Примечание
SFD3*	Настройка специальной функции (значение по умолчанию — 0x0000)	Бит0: Обработка исключений регистра памяти при отключении питания. 0: Система выполняет сброс. 1: Обработка не выполняется. Бит1: Выполнение пользовательской программы в подпрограмме внешних прерываний. 0: Выполнение в задаче 1: Выполнение в режиме прерываний (в этом режиме пользовательская подпрограмма прерываний не может содержать функциональный блок на языке C). Этот режим обычно используется, когда требуется большое количество внешних сигналов в режиме реального времени. Бит2: повышение приоритета внешних прерываний. 0: Не повышается 1: Повышается (до максимума)

Отображение вводов

Идентификатор	Функция	Описание	
SFD10*	I00 соответствует X**	Входная клемма 0 соответствует номеру X**	0xFF указывает на неисправность клеммы,

			0xFE указывает на незадействованную клемму
SFD11*	I01 соответствует X**		
SFD12*	I02 соответствует X**		
.....			
SFD73*	I77 соответствует X**	Значение по умолчанию 77 (восьмеричное)	

Отображение выводов

Идентификатор	Функция	Описание	
SFD74*	O00 соответствует Y**	Выходная клемма 0 соответствует номеру Y**, значение по умолчанию — 0.	0xFF указывает на неисправность клеммы, 0xFE указывает на незадействованную клемму
.....			
SFD137*	O77 соответствует Y**	Значение по умолчанию 77 (восьмеричное)	

Признак входа

Идентификатор	Функция	Описание	
SFD138*	Признак I00	Признак входной клеммы 0	0: положительная логика другие значения: отрицательная логика
SFD139*	Признак I01		
.....			
SFD201*	Признак I77		



Высокоскоростной счетчик

Идентификатор	Функция	Описание
SFD310	HSC0 Настройка фронта запуска счетчика в однофазном режиме	0: Подсчет по нарастающему фронту 1: Подсчет по спадающему фронту 2: Подсчет выполняется по нарастающему и спадающему фронту
SFD311	HSC2 Настройка фронта запуска счетчика в однофазном режиме	0: Подсчет по нарастающему фронту 1: Подсчет по спадающему фронту 2: Подсчет выполняется по нарастающему и спадающему фронту
SFD312	HSC4 Настройка фронта запуска счетчика в однофазном режиме	0: Подсчет по нарастающему фронту 1: Подсчет по спадающему фронту 2: Подсчет выполняется по нарастающему и спадающему фронту
SFD313	HSC6 Настройка фронта запуска счетчика в однофазном режиме	0: Подсчет по нарастающему фронту 1: Подсчет по спадающему фронту 2: Подсчет выполняется по нарастающему и спадающему фронту
SFD320	HSC0 Превышение частоты (раз)	2: частота в 2 раза выше 4: частота в 4 раза выше (в режиме работы счетчика для фаз А и В)
SFD321	HSC2 Превышение частоты (раз)	То же самое
SFD322	HSC4 Превышение частоты (раз)	То же самое
SFD323	HSC6 Превышение частоты (раз)	То же самое
SFD324	HSC8 Превышение частоты (раз)	То же самое
SFD325	HSC10 Превышение частоты (раз)	То же самое
SFD326	HSC12 Превышение частоты (раз)	То же самое



Идентификатор	Функция	Описание
SFD327	HSC14 Превышение частоты (раз)	То же самое
SFD328	HSC16 Превышение частоты (раз)	То же самое
SFD329	HSC18 Превышение частоты (раз)	То же самое
SFD330	Выбор бита HSC в абсолютном и относительном режиме (24 сегмента)	бит0 соответствует HSC0, бит1 соответствует HSC2, и т.д., бит9 соответствует HSC18 0: относ. 1: абсолютн.
SFD331	Прерывание для 24 сегментов при высокоскоростном подсчете	бит0 соответствует HSC0, бит1 соответствует HSC2, и т.д., бит9 соответствует HSC18 0: одиночный 1: циклический
SFD332	Функции для кулачков	бит0 соответствует HSC0, бит1 соответствует HSC2, и т.д., бит9 соответствует HSC18 0: функция для кулачков не поддерживается 1: функция для кулачков поддерживается

Конфигурация модуля расширения

Идентификатор	Функция	Описание
SFD340	Состояние конфигурации модуля расширения (#1#2)	Состояние конфигурации модулей расширения 1 и 2
SFD341	Состояние конфигурации модуля расширения (#3#4)	Состояние конфигурации модулей расширения 3 и 4

Идентификатор	Функция	Описание
.....		
SFD347	Состояние конфигурации модуля расширения (#15#16)	Состояние конфигурации модулей расширения 15 и 16
SFD348	Состояние конфигурации модуля BD (#1#2)	Состояние конфигурации модулей BD 1 и 2
SFD349	Состояние конфигурации модуля ED (#1)	Состояние конфигурации модуля ED 1
SFD350	Конфигурация модуля расширения	Конфигурация модуля расширения 1
:		
SFD359		
SFD360	Конфигурация модуля расширения	Конфигурация модуля расширения 2
:		
SFD369		
⋮	⋮	
SFD500	Конфигурация модуля расширения	Конфигурация модуля расширения 16
:		
SFD509		
SFD510	Конфигурация модуля BD	Конфигурация модуля BD 1
:		
SFD519		
SFD520	Конфигурация модуля BD	Конфигурация модуля BD 2

Идентификатор	Функция	Описание
⋮		
SFD529		
SFD530	Конфигурация модуля ED	Конфигурация модуля ED 1
⋮		
SFD539		

Связь

Идентификатор	Функция	Примечание
SFD600	Количество бит в буфере связи в свободном формате через порт COM1	0: 8 бит 1: 16 бит
SFD610	Количество бит в буфере связи в свободном формате через порт COM2	0: 8 бит 1: 16 бит
SFD620	Количество бит в буфере связи в свободном формате через порт COM3	0: 8 бит 1: 16 бит
SFD630	Количество бит в буфере связи в свободном формате через порт COM4	0: 8 бит 1: 16 бит
SFD640	Количество бит в буфере связи в свободном формате через порт COM5	0: 8 бит 1: 16 бит

Приложение 4. Таблица конфликтов ресурсов ПЛК

Во время эксплуатации ПЛК могут возникать конфликты из-за одновременного использования некоторых ресурсов. В этом разделе перечислены ресурсы, которые могут вызывать конфликты в каждой модели ПЛК. Это в основном касается высокоскоростного счетчика, точной синхронизации и импульсного выхода.

Точная синхронизация	Высокоскоростной подсчет				Выходные импульсы
PMP20-24/30/48/60					
ET0	-	-	-	-	-
ET2				HSC6	
ET4			HSC4		
ET6		HSC2			
ET8	HSC0				
ET10					Y3
ET12					Y3
ET14					Y2
ET16					Y2
ET18					Y1
ET20					Y1
ET22					Y0
ET24					Y0

✖1 Эту таблицу следует читать по горизонтали. Любые два ресурса в каждой строке нельзя использовать одновременно. В противном случае, могут возникнуть конфликты.

Приложение 5 Таблица настройки функций ПЛК

Это приложение предназначено для проверки конфигурации каждой модели. С помощью этой таблицы можно легко определить тип изделий.

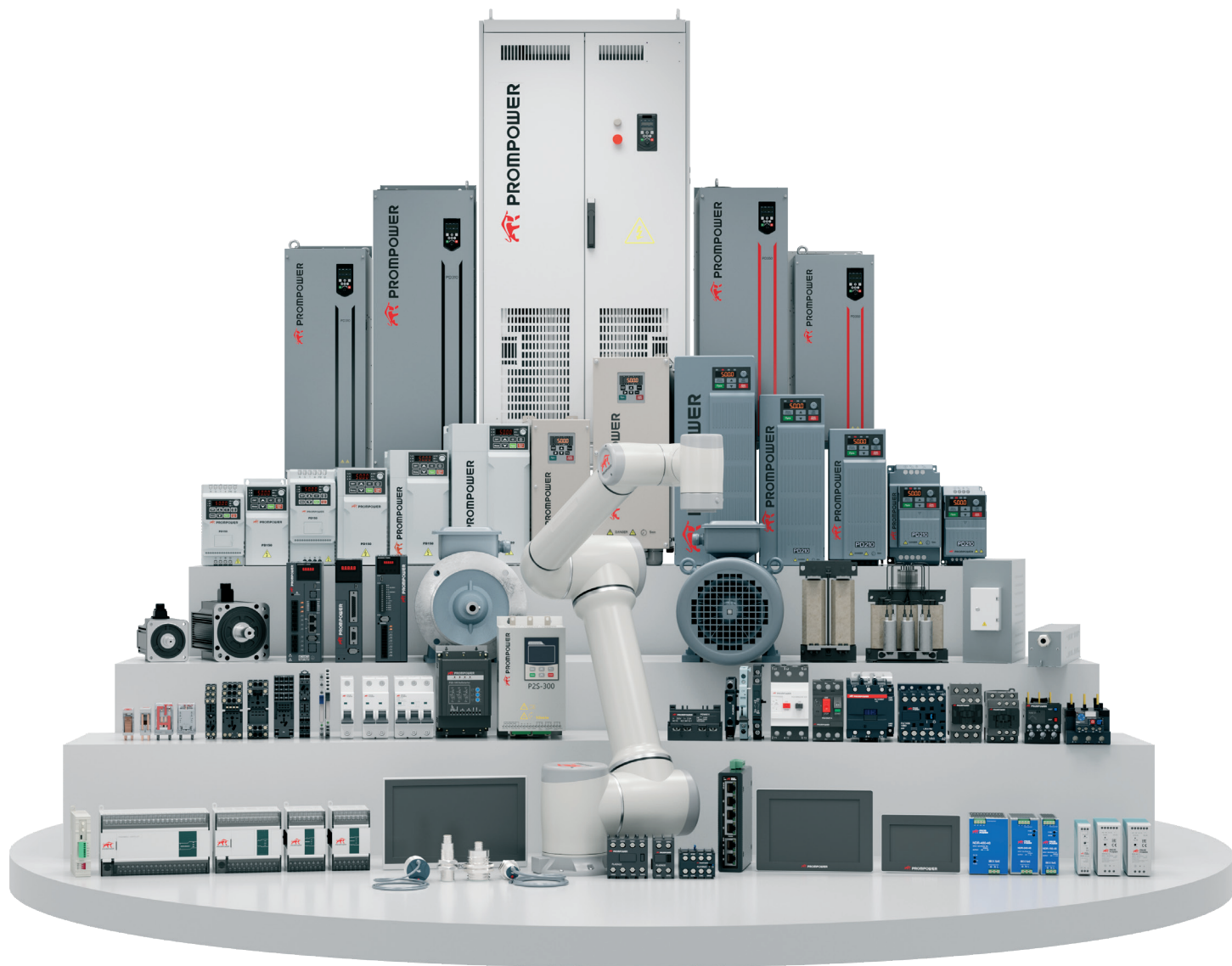
○ Выбираемый элемент × Не поддерживается ✓ Поддерживается

Серия	U S B	2 3 2	4 8 5	Ether net	Модуль рас- ширения	B D	E D	O C	HSC	Выходные им- пульсы		Внешнее пре- рывание
									Дифферен- циальный	Нор- маль- ный	Диф- фе- рен- циаль ный	
PMP20												
PMP20-30	×	1	1	2	16	1	1	3	×	2	×	10
PMP20-30T4	×	1	1	2	16	1	1	4	×	4	×	10
PMP20-60T6	×	1	1	2	16	2	1	6	×	6	×	10

Примечание:

1: Все модели в стандартной комплектации поддерживают функцию часов.

Профессиональное оборудование для промышленной автоматизации



Официальный дистрибьютор:



 **PROMPOWER**
www.prompower.ru