Серия ПЛК FX MELSEC

Программируемые контроллеры

Руководство по обучению

Учебный стенд
FX-TRN-KIT-R

Артикул xxxxx
15 02 2008
FXTRN0573023-A
Версия A

Версии

* Номер руководства указан в нижнем левом углу передней части обложки.

Дата выпуска	Номер руководства	Версия
12/2007		Версия A – Вновь созданное руководство (в значительной степени созданное на основе Руководства по обучению MEAU FX)

Введение

Данное руководство по обучению было создано для сопровождения учебного стенда FX-TRN-KIT-R. Пожалуйста, внимательно изучите руководство, чтобы полностью ознакомиться с функциями и характеристиками ПЛК серии FX3U и самого учебного стенда. Это поможет вам в будущем правильно использовать контроллеры серии FX3U и сопутствующие устройства.

Содержание

ГЛАВА 1 – Введение и обзор

Добро пожаловать на учебный семинар по контроллерам серии FX. Семинар предназначен для разработчиков и специалистов по системам управления, ответственных за разработку прикладных программ с использованием программируемых логических контроллеров (ПЛК) серии FX. Кроме традиционного обучения продуктам, ориентированного на аппаратную часть, часть семинара посвящена написанию программ, для того, чтобы заложить основу для успешного и быстрого цикла их разработки и отладки.

Цели главы: Завершив эту главу, слушатели смогут…

- Понять цели данного учебного семинара.
- Понять цели каждой главы.
- Перечислить предпосылки и определить целевую аудиторию семинара.

Материалы: Руководство по обучению FX-TRN-KIT-R

Обзор: Эта глава является введением к учебному семинару, описывает его структуру, дает перечень глав, их последовательность и содержание. Если у вас имеются дополнительные потребности или вам необходимо рассмотреть темы и вопросы, не охваченные в учебном семинаре, пожалуйста, сообщите об этом преподавателю учебного семинара или ближайшему провайдеру продуктов Мицубиси Электрик.

1.1 Цели учебного семинара

После завершения этого семинара **слушатели смогут**…

- Ориентироваться в имеющихся версиях аппаратных средств серии FX и понимать области их применения.

- Использовать программное обеспечение GX Developer в сочетании с учебным стендом FX-TRN-KIT-R, чтобы разрабатывать, проверять, отлаживать и внедрять программы на языке символических релейных диаграмм для приложений управления станками.

- Понимать структуру и базовые операции специальных функциональных модулей (SFM), плат расширения левой шины FX3U (BD) и специальных адаптеров (ADP), используемых для таких операций, как аналого-цифровое (АЦ) и цифро-аналоговое (ЦА) преобразование, работа в сетях, а также быстродействующая обработка.

1.2 Что необходимо для семинара

Этот курс предназначен для лиц с базовыми знаниями электроники и с определенным уровнем общих знаний в области управления производственными процессами. Также желателен опыт работы с кодом языка релейных диаграмм ПЛК или любым языком программирования.

1.3 Продолжительность учебного семинара

Это **3-дневный** учебный семинар.

1.4 Описание учебного семинара

ГЛАВА 1 – Введение к учебному семинару
Введение к учебному семинару и краткое описание глав.

ГЛАВА 2 – Обзор аппаратной части
Обсуждается структура аппаратной части программируемого логического контроллера серии FX, типы и характеристики центральных процессоров (ЦП), модулей ввода и вывода, специальных функциональных модулей, плат расширения, специальных адаптеров, а также прочие аппаратные компоненты, связанные с серией FX.

ГЛАВА 3 – Программирующее оборудование
Рассмотрены аппаратные средства, программное обеспечение, и соединения, необходимые для подключения ноутбука или ПК к ПЛК серии FX, а также альтернативы программированию с помощью ноутбука и ПК.

ГЛАВА 4 – Системы счисления
Обсуждаются четыре различные системы счисления, которые используются системами ПЛК серии FX: двоичная, восьмеричная, шестнадцатеричная и двоично-кодированная десятичная (BCD).

ГЛАВА 5 – Цифровые данные в ПЛК
Объясняется, как целые числа и нецелые числа обрабатываются ПЛК.

ГЛАВА 6 – Системные операнды
Рассмотрены устройства ПЛК серии FX, например, операнды входов X, операнды выходов Y и маркеры M, которые используются в инструкциях программы.

ГЛАВА 7 – Адресация
Рассмотрены правила адресации на правой и левой шине FX3U, включая ограничения на максимальное количество входов-выходов.

ГЛАВА 8 – Конструкция учебного стенда
Анализируется и объясняется аппаратный учебный стенд, на котором будут запускаться программы, написанные на этих курсах.

ГЛАВА 9 – Типы команд ПЛК
Объясняется 3 основных типа команд программирования ПЛК серии FX и их области применения.

ГЛАВА 10 – Базовые команды
Объясняются контакты, катушки и прочие базовые команды языка релейных диаграмм.

ГЛАВА 11 – Разработка и редактирование программ
Рассмотрен процесс запуска проекта, написание программы простой релейной диаграммы, передача данных на ПЛК и мониторинг операций ПЛК.

ГЛАВА 12 – Таймеры и счетчики
Подробно рассмотрены эти два важных операнда. Также в главе содержится демонстрационное приложение с использованием таймеров.

ГЛАВА 13 – Прикладные команды
Рассмотрено много расширенных команд: команды манипулирования данными, арифметические команды, команды сравнения, команды преобразования, логические операции, методы использования левой шины FX3U, и команды TO/FROM. Глава содержит множество примеров приложений с использованием прикладных команд.

ГЛАВА 14 – Диагностические операнды
Рассмотрены специальные маркеры и регистры данных, которые могут помочь в поиске неисправностей и написании программ. Кроме того, глава содержит краткое описание диагностических возможностей GX Developer и образец диагностического кода языка релейных диаграмм.

ГЛАВА 15 – Документация и распечатка
Рассмотрены типы документации, которая может быть включена в релейную диаграмму GX Developer, а также различные возможности распечатки проекта GX Developer.

1.5 Обзор серии продукции

Мицубиси предлагает **модульные** и **микро** контроллеры. Ниже показаны две из выпускаемых моделей.

Серия Q

Серия FX

В этом курсе мы рассмотрим ПЛК серии FX. Программирование контроллеров серии FX и Q имеет много общего. Просто серия FX включает входы, выходы, электропитание и процессор в одном корпусе!

ГЛАВА 2 – Обзор аппаратной части серии ПЛК серии FX

В этой главе обсуждается структура аппаратной части программируемого логического контроллера (ПЛК) серии FX, включая обзор различных типов ЦП, модулей ввода-вывода и других связанных аппаратных компонентов и устройств.

Цели главы: Завершив эту главу, слушатели смогут…

- Объяснить общие компоненты ПЛК.
- Описать различные модели в серии FX.
- Описать характеристики имеющихся модулей ввода-вывода.
- Перечислить основные факторы, которые необходимо рассматривать при определении аппаратной части.

Материалы: Руководство по обучению FX-TRN-KIT-R
Каталог ПЛК серии FX

2.1 Что такое программируемый логический контроллер?

Программируемый логический контроллер ПЛК – это управляемый микропроцессором компьютер, специально разработанный для **управления производственной установкой в режиме реального времени**.

Все ПЛК включают три уровня: **УРОВЕНЬ ВВОДА**, **УРОВЕНЬ ОБРАБОТКИ**, и **УРОВЕНЬ ВЫВОДА**.

- ***УРОВЕНЬ ВВОДА***

  Уровень ввода включает встроенные входные контактные клеммы или входные клеммы расширения, к которым подключаются входные устройства приложения, например, конечные переключатели, транзисторные датчики и т.д. Когда входное напряжение достигает заданного уровня, вход активизируется. Активный вход может быть считан ПЛК.

- ***ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР***

  Центральный процессор является главным компонентом ПЛК, который выполняет и управляет всеми процессами в системе. Каждая модель серии FX имеет свой тип ЦП. Поэтому модель FX (и ЦП) необходимо выбирать в зависимости от требований приложения.

- ***УРОВЕНЬ ВЫВОДА***

  Уровень вывода включает встроенные выходные контактные клеммы и выходные клеммы расширений, которые управляются ЦП. Анализируя условия на входах, ЦП включает соответствующие выходы, активизируя такие устройства машины, как лампы, сирены, реле, соленоиды или двигатели.

Эти три уровня ПЛК управляются написанной пользователем программной, называемой релейной диаграммой, соответствующей структуре аппаратных цепей. Связь между входами и выходами управляется логическими элементами в релейной диаграмме. Вместо аппаратных цепей мы теперь работаем с логическими элементами программы, что намного упрощает модификации и усовершенствования машины.

2.2 ПЛК серии FX

Все ПЛК серии FX имеют ряд общих характеристик:

1) Встроенный блок электропитания. Большинство ПЛК этой серии имеют встроенный блок питания, для которого требуется электросеть100-240 В переменного тока. Несколько ПЛК имеют версию с электропитанием от линии постоянного тока.

2) Встроенный ввод-вывод. Каждый главный блок имеет определенное количество входов и выходов, в зависимости от выбранной модели. Обычно на входы подается сигнал постоянного тока, хотя некоторые модели имеют также входы переменного тока. Для всех моделей (кроме FX0S) предлагаются версии с релейным и транзисторным выходом; в некоторых имеются симисторные выходы.

3) Типичные команды. Хотя более совершенные ПЛК включают расширенные наборы команд, все ПЛК поддерживают по меньшей мере систему типичных команд, включающую 20 базовых и 35 прикладных команд.

4) Дополнительные встроенные специальные функциональные возможности. Например, все ПЛК серии FX включают встроенный высокоскоростной счетчик (или несколько высокоскоростных счетчиков).

История серии FX

ПЛК серии F впервые были представлены на рынке в 1981 г. Впоследствии серия F несколько раз совершенствовалась и модернизировалась; были выпущены модели F1, F1J и F2. На смену F2 были выпущены ПЛК серии FX. Эти модели включали FX, FX0, FX1, FX2, FX0S, FX0N и FX2C, а также предлагаемые в настоящее время FX1S, FX1N, FX2N, FX2NC и FX3U. **К 2006 г. всемирные продажи ПЛК Мицубиси серии F превысили 7 миллионов устройств!**

Ниже мы детально рассмотрим имеющиеся в настоящее время модели ПЛК серии FX.

FX1S

ПЛК серии FX1S отличаются небольшой занимаемой площадью, подобно FX0S, но имеют намного больше возможностей. FX1S включает больше адресов ввода-вывода (до 30), больше внутренних операндов, и обеспечивает позиционирование с выходом для вывода серий импульсов до 100 кГц и встроенными командами позиционирования. К передней панели можно подключить небольшой интерфейс оператора (FX1N-5DM), чтобы контролировать и изменять состояние таймеров, счетчиков и регистров данных. В FX1S может использовать одна плата расширения, но не расширительная шина.

FX1N

FX1N обеспечивает среднее количество точек ввода-вывода, аналогично FX0N, но отличается более широкими возможностями и расширяемостью. FX1N имеет больше внутренних операндов, чем FX0N (1536 маркеров (M) и 235 счетчиков), и возможности позиционирования, эквивалентные FX1S. FX1N также включает ряд плат расширения, которые можно добавить для обеспечения дополнительного ввода-вывода, а также позволяет подключать коммуникационные модули FX0N. Может быть подключена правая расширительная шина со специальными модулями, обеспечивающими дополнительные возможности ввода-вывода, аналоговые модули, позиционирования или связи. К FX1N также можно подключить FX1N-5DM.

FX2N

В настоящее время FX2N – один из наиболее мощных процессоров Мицубиси. Он является полностью расширяемым и может управлять до 256 точками ввода-вывода. ПЛК включает 3072 маркеров, 256 таймеров, 234 счетчика, 8000 регистров данных, и до 21 высокоскоростного счетчика. Для этой серии имеются многочисленные специальные модули, включая модули для Profibus, CC-Link, AS-Interface и ввода-вывода, а также модуль электронного кулачкового переключателя. Кроме специальных модулей на правой расширительной шине, этот универсальный ПЛК может быть расширен с помощью плат расширения, которые подключаются к передней панели ПЛК. Таким образом можно ввести в ПЛК второй порт программирования, порт RS-485, порт RS-232, или даже шлюз Ethernet. С этим ПЛК могут использоваться FX0N и модули серий FX0N и FX.

FX2NC

Аналогичный FX2N во всех отношениях, кроме отсутствия встроенных часов реального времени, FX2NC использует распределенный ввод-вывод с разъемным соединением, а не клеммы ввода-вывода. В результате получен эффективный, расширяемый, сверхкомпактный главный блок ПЛК, занимающий менее 1/3 пространства наименьшего блока FX2N. Платы расширения заменены модулями специальных адаптеров, а к правой расширительной шине все еще можно подключать специальные модули.

FX3U

FX3U – последний и самый мощный процессор Мицубиси. Этот высокорасширяемый ПЛК может управлять до 384 входами-выходами со стандартными 64 000 шагами в памяти для хранения программы. Он включает в четыре раза больше внутренних операндов, чем FX2N, более быстрый ЦП, а также новую расширительную шину с левой стороны базового блока. Все специальные модули правой расширительной шины, имеющиеся для FX2N, совместимы с этой серией. Для того, чтобы по максимуму задействовать три последовательных соединения на этом ПЛК, можно использовать новые коммуникационные платы и специальные адаптеры, включая новую USB коммуникационную плату. Новые опции работы с сетями включают полномасштабный модуль Ethernet, ведущие и ведомые модули Profibus и специальный адаптер Modbus. Выходы для вывода серий импульсов на главном блоке FX3U позволяют управляют трехосевой системой позиционирования, расширяемой до четырехосевой при использовании новых быстродействующих специальных адаптеров. Наконец, FX3U можно расширить, используя модуль двухосевого позиционирования, в котором применяется SSCNET III – собственная оптоволоконная сеть позиционирования Мицубиси.

В этом курсе обучения мы будем работать с ПЛК серии FX3U. Однако, большая часть учебного материала применима ко всей серии FX программируемых логических контроллеров.

2.3 Аппаратные компоненты

Главные блоки.

Главный блок FX3U

Главный блок FX2NC

Главный блок содержит:

1) ЦП. Именно он обеспечивает вычислительную мощность ПЛК; считывает входы, производит логические и математические вычисления и записывает значения в выходы.

2) Встроенный блок питания. Он подает электропитание на ЦП, входы, и ограниченное количество подключенных расширительных блоков, специальных функциональных модулей и специальных адаптеров.

3) Встроенные входы. Они могут работать с сигналами постоянного или переменного тока, в зависимости от выбранной модели. Наибольшие главные блоки серии FX имеют 64 входа.

4) Встроенные выходы. Они могут быть релейными, транзисторными или симисторными, в зависимости от выбранной модели. Наибольший главный блок серии FX имеет 64 выхода. Для серии FX1S и FX1N отношение количества входов к числу выходов составляет 3:2 или 4:3. В сериях FX2N, FX2NC и FX3U это отношение равно 1:1.

5) Порт программирования В качестве протокола связи данный порт использует RS-422. ПЛК может быть запрограммирован через этот порт, или вместо него можно подключить интерфейс человек-машина (HMI).

6) Порты для подключения принадлежностей. Они позволяют подсоединить модули памяти или платы расширения к главному блоку.

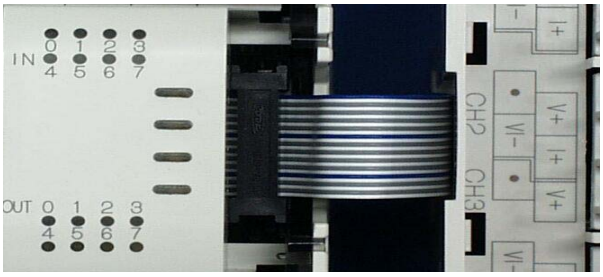
Модули расширения с собственным блоком питания

Как отмечалось выше, ввод-вывод серии FX является расширяемым. Один из способов увеличить количество точек ввода-вывода – использование модулей расширения с собственным блоком питания. Эти модули включают:

1) Встроенный блок питания. Он может использоваться как альтернатива электропитанию главного блока для других расширительных блоков и специальных модулей. Имеются версии, работающие от сети переменного и постоянного тока.

2) Встроенные входы. Имеются входные модули <30 В= и <264 В ≈ . Модуль может включать 16 или 24 входа, в зависимости от выбранной модели.

3) Встроенные выходы. Число выходов совпадает с числом входов (16 или 24). Имеются релейные, транзисторные и симисторные выходы.



Подключение входов-выходов к главному блоку

Блоки расширения без собственного блока питания

Другой способ увеличить количество точек ввода-вывода в системе серии FX – использование блоков расширения без собственного блока питания. Не имея блока питания, они получают электропитание от главного блока или модуля расширения с блоком питания. Поэтому имеется ограничение на количество добавляемых расширительных блоков (в дополнение к ограничению на назначение входов-выходов). Позже мы рассмотрим, как вычислить количество таких блоков.

Блоки расширения без собственного блока питания включают 8 или 16 входов-выходов. Каждая клемма является либо входом, либо выходом, но не обоими. Входы работают от 24 В пост. тока, и имеются все три типа выходов. Если имеющиеся модели не удовлетворяют требования системы, можно использовать модули ПЛК серии FX0N, а также ПЛК серии FX2NC (с помощью конвертора FX2N-CNV-IF).

Замечание: Правильно используйте терминологию: **модуль расширения** содержит **блок питания**, в то время как **блок расширения не содержит.**

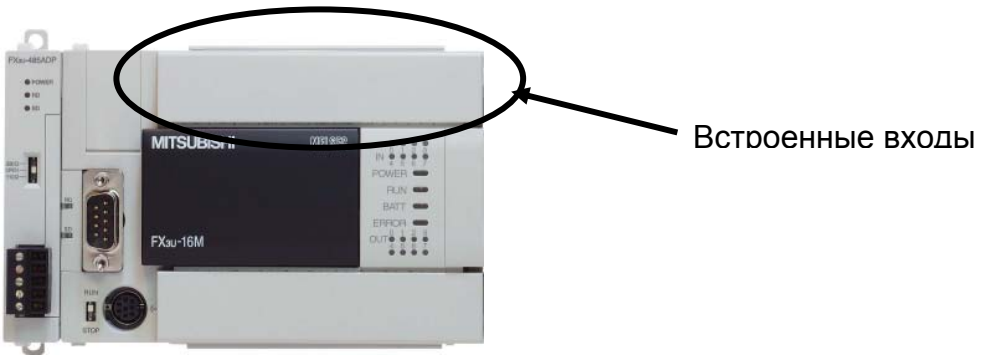
2.4 Входы

Входные устройства являются интерфейсом между ПЛК и машинами. Главный блок ПЛК серии FX имеет несколько встроенных входов. Если их недостаточно, можно подключить входные модули расширения и блоки расширения.

Имеются два различных типа входов…

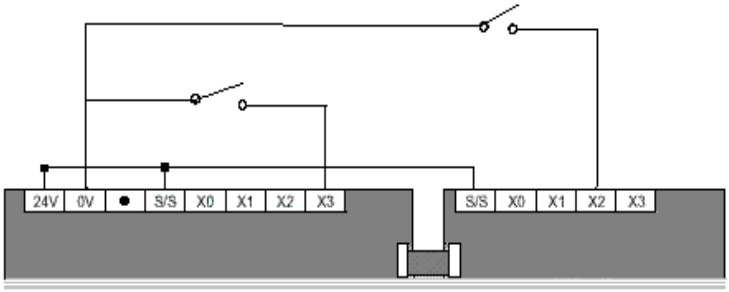
- Входы постоянного тока
  - Быстрый отклик
  - В 90% новых конструкций используется этот тип входа
  - Входное напряжение 24 В постоянного тока
  - Отрицательная или положительная логика

- Входы переменного тока
  - Медленная реакция
  - Простое подключение к устройствам переменного тока
  - Входное напряжение 120 В переменного тока



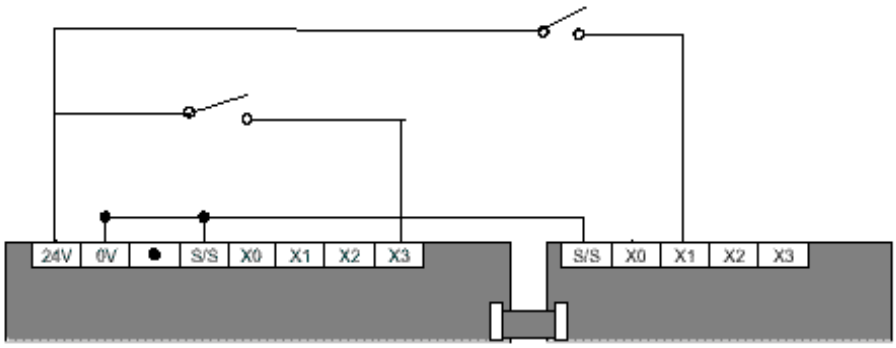
ОТРИЦАТЕЛЬНАЯ или ПОЛОЖИТЕЛЬНАЯ логика относится к уровню напряжения, который активизирует вход.

ОТРИЦАТЕЛЬНАЯ логика: вход активизируется при подключении к **ЗЕМЛЕ**. Клемму S/S необходимо вручную соединить с +24 В=.



Используйте ОТРИЦАТЕЛЬНУЮ логику для датчиков типа **NPN**

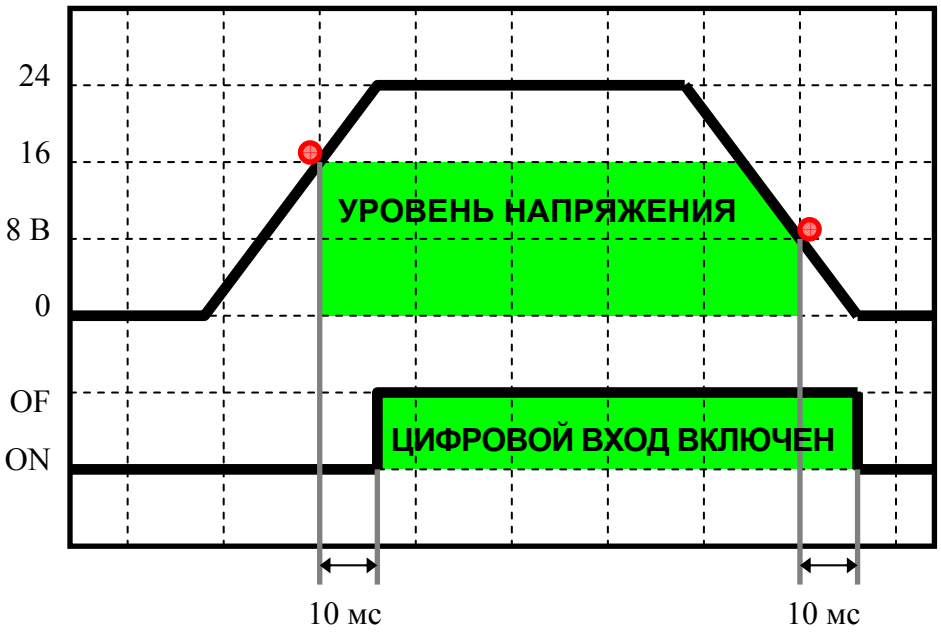
ПОЛОЖИТЕЛЬНАЯ логика: вход активизируется при подключении к **+24 В=**. Клемму S/S необходимо вручную соединить с ЗЕМЛЕЙ.



Используйте ПОЛОЖИТЕЛЬНУЮ логику для датчиков типа **PNP**

Уровни активизации входа изменяются в зависимости от типа модуля. Как правило, вход активизируется при уровне сигнала 2/3 и затем становится неактивным на уровне 1/3.

Например, при напряжении питания +24 В= вход станет активным, когда входное напряжение достигнет 16 В=, и затем перейдет в неактивное состояние, когда напряжение упадет до 8 В=.



Стандартное устройство ввода имеет **входной фильтр с постоянной времени 10 мс**. Это делается преднамеренно для устранения дребезга контактов. Когда контакты переключателя закрываются, механические контакты немного дребезжат, что в течение короткого промежутка времени приводит к быстрым флуктуациям входного напряжения. За эти 10 мс контакты переключателя прекращают дребезжать и стабилизируются.

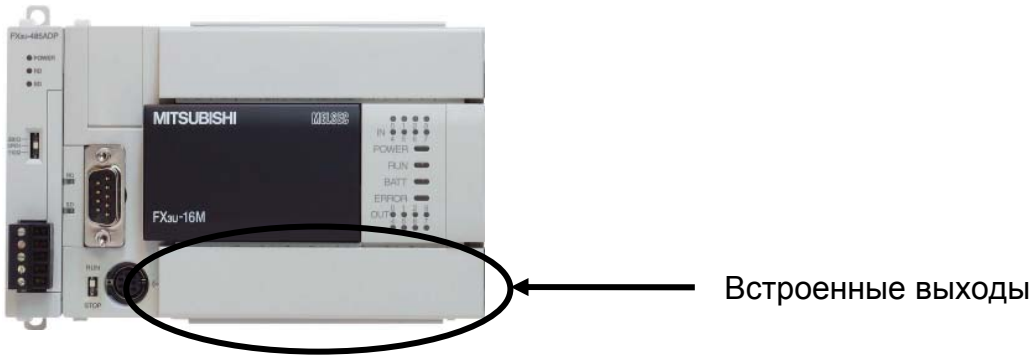
Возможно регулировать постоянную входного фильтра (см. раздел 14.2 данного Руководства).

Как правило, если требуется обрабатывать более 20 импульсов в секунду, должен использоваться модуль высокоскоростного счетчика или встроенные высокоскоростные входы. Более полную информацию см. в разделе 12.2 данного Руководства.

Главный блок ПЛК будет содержать 8, 16, 24, 32, 40, или 64 встроенных физических входов. Модули расширения обеспечивают 16 или 24 дополнительных точек ввода. Блоки расширения обеспечивают 8 или 16 дополнительных точки ввода. Все точки ввода могут переключаться между режимами отрицательной или положительной логики.

2.5 Выходы

Выходные устройства позволяют ПЛК управлять и взаимодействовать с другими машинами и оборудованием. Главный блок ПЛК серии FX имеет несколько встроенных выходов. Если их недостаточно, можно подключить выходные модули расширения и блоки расширения.



Имеются три различных типа выходов …

- Релейные выходы
- Симисторные выходы
- Транзисторные выходы

РЕЛЕЙНЫЕ ВЫХОДЫ

Реле – это сухие контакты; при активизации выхода входной контакт отключается от общего провода. Это наиболее распространенный тип использования выходной клеммы. Могут коммутироваться нагрузки до 2 А, 100 В ... 240 В переменного тока или 30 В постоянного тока с максимальным током 8 А через общий провод. Большинство базовых блоков и модулей расширения имеют четыре выхода на общий провод.

СИМИСТОРНЫЕ ВЫХОДЫ

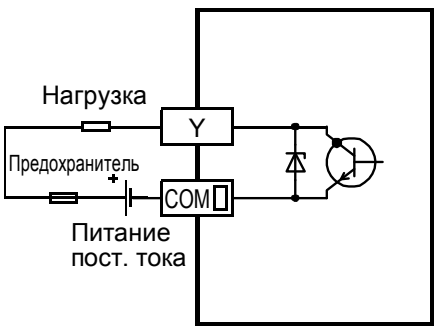
Симисторы – это твердотельные переключатели переменного тока. При активизации выхода клемма соединяет нагрузку с источником питания переменного тока. Коммутируется нагрузка до 0,3 А при 240 В≈ на точку, с максимальным током 0,8 А через общий провод. Каждый базовый блок или модуль расширения имеет до четырех выходов на общий провод.

ТРАНЗИСТОРНЫЕ ВЫХОДЫ

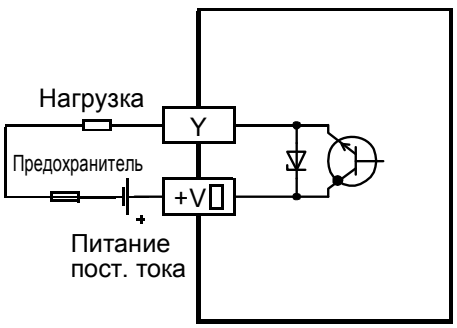
Быстрый отклик – основная характеристика такого выхода. Транзисторы являются твердотельными переключателями постоянного тока. При активизации выхода клемма соединяет нагрузку с источником питания постоянного тока. Коммутируется нагрузка до 0,5 А на точку, с максимальным током 0,8 А через общий провод. Могут коммутироваться напряжения от 5 до 30 В=. Большинство базовых блоков и модулей расширения имеют четыре выхода на общий провод. Транзисторные выходы требуются для быстродействующих выходов для вывода серий импульсов из главного блока ПЛК.

Подключение транзисторного выхода:

Выходы в режиме отрицательной логики



Выходы в режиме положительной логики



2.6 Аналоговые модули и специальные адаптеры

Во всех обсуждавшихся выше аппаратных устройствах использовался дискретный ввод-вывод: входы или выходы могли находиться во включенном (ON) или выключенном (OFF) состояниях. Это допустимо, если все входы в системе ПЛК являются выключателями или простыми датчиками, и выходы необходимо просто включать и выключать. Однако, если ПЛК серии FX должен контролировать или регулировать температуру или другой вход/выход с изменяющимся значением напряжения, требуется специальный аналоговый функциональный модуль (SFM) или специальный адаптер (ADP). Аналоговые платы расширения (BD) для FX1S, FX1N и FX2N не описаны в этом документе.

Аналоговые модули для правой шины

Имеются 3 типа аналоговых специальных функциональных модулей: аналоговые входные модули, аналоговые выходные модули и комбинированные модули аналогового входа-вывода. Все они используются с точками ввода/вывода, которые имеют больше состояний, чем просто "Включен" или "Выключен". Примеры аналоговых входов – считывание скорости или давления. Примером аналогового выхода может служить регулируемая скорость вращения двигателя.

Поставляются входные модули с 2, 4 или 8 каналами аналогового ввода. Это **FX2N-2AD**, **FX2N-4AD** и **FX2N-8AD**, соответственно. Выходные модули могут иметь 2 или 4 канала. Это **FX2N-2DA** и **FX2N-4DA**, соответственно. Имеются 2 комбинированных модуля: **FX0N-3** имеет 2 канала ввода и 1 канал вывода, **FX2N-5** имеет 4 канала ввода и 1 канал вывода. Серия FX3U также включает два аналоговых модуля, **FX3U-4AD** и **FX3U-4DA**, которые аналогичны FX2N-4AD и FX2N-4DA, но поддерживают более высокое разрешение и повышенное быстродействие с ПЛК серии FX3U.

Все модули измеряют или подают на выход ток или напряжение в различных диапазонах: обычно от –20 мА до +20мА, от 4 до 20 мА, или от –10В до +10 В, как задано при программировании. В зависимости от типа модуля данные принимаются либо с ПЛК, либо из каналов аналогового ввода. Если данные поступают с ПЛК, то исходные цифровые данные интерпретируются как уровень выводимого тока или напряжения, т.е. производится цифро-аналоговое преобразование (ЦАП). Если аналоговое значение считывается из канала ввода (аналогового входа), то производится аналого-цифровое преобразование (АЦП).

Пример программирования

Программист ПЛК хочет обнаружить небольшие изменения в давлении, чтобы управлять процессом смешивания реагентов. В его распоряжении датчик с диапазоном 0 ... 300 фунтов/кв. дюйм (PSI), генерирующий напряжение в диапазоне от –10 до +10 В. Модуль FX2N-4AD интерпретирует значение –10 В как число -2000, и +10 В как +2000.

Имея эту информацию, программист знает, что при давлении 0 PSI датчик посылает напряжение –10В, что приводит к значению -2000, которое считывается ПЛК. При давлении 150 PSI датчик генерирует 0 В; в результате ПЛК считывает значение 0. При давлении 300 PSI напряжение равно 10 В, и ПЛК считает 2000. Программист должен знать, каким образом аналоговое значение давления преобразуется в цифровое значение ПЛК, и написать программу, чтобы масштабировать и/или использовать это значение.

Температурные входные модули

Эти модули аналогичны другим аналоговым модулям, имеющим 4 входных канала, но отличаются типом подключаемых входных устройств. В **FX2N-4AD-TC**для измерения температуры используются термопарные датчики, которые вырабатывают изменяющееся напряжение. Поддерживаются термопары типа J и K. В **FX2N-4AD-PT** используются платиновые датчики температуры (PT-100 RTD). Эти датчики позволяют обнаруживать очень небольшие изменения температуры (например, 0,2°C – 0,3°C, 0,36°F – 0,54°F). Входы модуля **FX2N-8АЦ** можно также конфигурировать под типы термоэлементов K, J или T.

Аналоговые адаптеры левой шины FX3U

Одной из новых особенностей FX3U является высокоскоростная расширительная шина на левой стороне ПЛК. На этой расширительной шине с левой стороны базового блока можно использовать максимум 4 любых аналоговых специальных адаптера (ADP) из имеющихся 4-х типов аналоговых ADP. Ассортимент ADP включает 4-канальный входной ADP, 4-канальный выходной ADP, 4-канальный термопарный входной ADP, и 4-канальный входной ADP для датчиков PT-100 RTD. Эти ADP для левой расширительной шины не требуют команд TO/FROM (которые обсуждаются ниже). Данные с аналоговых входов записываются непосредственно в специальные регистры данных в ПЛК.

2.7 Высокоскоростные модули ввода-вывода и позиционирования. Специальные адаптеры

Обычные счетчики в ПЛК зависят от времени цикла релейной диаграммы в силу двух обстоятельств: обновления условий на входе, используемом как счетный вход, и обновления накопленного значения используемого счетчика. Это может оказаться слишком медленным для высокоскоростных счетных приложений. Соответственно, главные блоки ПЛК серии FX включают встроенные высокоскоростные счетчики с различными максимальными частотами счета, в зависимости от модели ПЛК. Высокоскоростные выходы для вывода серий импульсов на главном блоке (SFM), или ADP можно использовать для приложений позиционирования, где необходимо точно управлять таким оборудованием, как инверторы и серводвигатели. Используя специальные функциональные модули, можно реализовать приложения позиционирования с дополнительными возможностями интерполирования. ПЛК серии FX имеет много расширяемых опций высокоскоростного ввода/вывода и управления позиционированием.

Высокоскоростной модуль счетчика

Модуль **FX2N-1HC** предоставляет подключенному ПЛК серии FX один дополнительный высокоскоростной счетчик с частотой до 50 кГц с возможностью выбора входов 5, 12 или 24 В=. Он позволяет реализовать два однофазных входа, один прямой и один реверсивный, или один двухфазный вход. Он также включает два встроенных транзисторных выхода, которые могут управляться независимо с помощью внутренних команд сравнения.

Модули позиционирования по одной оси

Модуль **FX2N-1PG** (PG означает Pulse Generator = генератор импульсов) создает выход для вывода серий импульсов, который может использоваться в приложениях позиционирования. Этот модуль принимает входные сигналы 24 В= и генерирует серии импульсов с частотой до 100 кГц. FX2N-1PG присоединяется к правой расширительной шине.

Модуль **FX2N-10PG** может работать с входными сигналами 5 В= или 24 В= и генерировать серии импульсов с частотой до 1 МГц. Этот модуль также присоединяется к правой расширительной шине.

Для обоих модулей позиционирования PG характеристики выходов, вырабатывающих серии импульсов, например, частота и состояние ON/OFF, могут управляться программой ПЛК или параметрами, установленными до работы.

FX2N-10GM – одноосевой модуль позиционирования, который может использоваться либо на правой расширительной шине, либо независимо, из ПЛК серии FX на частоте до 200 кГц. Для обоих модулей позиционирования GM специализированное программное обеспечение FX-PCS-VPS/WIN-E позволяет пользователям программировать модули отдельно от ПЛК в визуальной пошаговой среде.

Модули позиционирования по двум осям

FX2N-20GM – двухосевой модуль позиционирования, который может использоваться либо на правой расширительной шине, либо независимо из ПЛК серии FX на частоте до 200 кГц. FX2N-20GM также обладает возможностями линейной и круговой интерполяции. Для обоих модулей позиционирования GM специализированное программное обеспечение FX-PCS-VPS/WIN-E позволяет пользователям программировать модули отдельно от ПЛК в визуальной пошаговой среде.

Модуль **FX3U-20SSC-H** позволяет реализовать в ПЛК серии FX3U высокопроизводительное управление позиционированием. Он предназначен для использования серия серводвигателей Mitsubishi MR-J3-B с 10 Мбит/с оптоволоконной сетью связи SSCNETIII. Сервоусилители MR-J3-B можно разместить на расстоянии до 50 метров от ПЛК, используя специализированные оптоволоконные готовые к использованию кабели. Этот модуль управляет до двумя осями на частоте до 50 МГц и может быть настраиваться и использоваться с собственным специализированным программным обеспечением FX Configurator-FP. Модуль может выполнять обширную таблицу операций и обладает возможностями линейной и круговой интерполяции. Он присоединяется к правой расширительной шине.

Высокоскоростной входной адаптер левой шины FX3U

Специальный адаптер **FX3U-4HSX-ADP** подключается к левой расширительной шине FX3U и увеличивает существующую частоту высокоскоростного счета 100 кГц главного блока FX3U до 200 кГц. FX3U-4HSX-ADP не требует команд TO/FROM (см. ниже). Вместо этого он непосредственно направляет высокоскоростные входы в X0-X3 или X4-X7 (в зависимости от адреса адаптера). На ЦП FX3U можно использовать максимум два высокоскоростных входных адаптера, что обеспечивает до восьми высокоскоростных входов. Эти специальные адаптеры должны располагаться непосредственно с левой стороны ПЛК или подключаться к левой стороне высокоскоростного выходного адаптера FX3U-2HSY-ADP.

Высокоскоростной выходной адаптер левой шины FX3U

Модуль **FX3U-2HSY-ADP** подключается к левой расширительной шине FX3U и расширяет существующие возможности высокоскоростного импульсного выхода транзисторных главных блоков FX3U (3 оси, 100 кГц) до 200 кГц и 4 осей. FX3U-2HSY-ADP не требует команд TO/FROM (см. ниже). Вместо этого он непосредственно направляет высокоскоростные выходы в Y0, Y1, Y4, и Y5 или Y2, Y3, Y6, и Y7 (в зависимости от адреса адаптера). На ЦП FX3U можно использовать максимум два высокоскоростных выходных адаптера, что обеспечивает до четырех высокоскоростных выходов. Эти специальные адаптеры должны располагаться непосредственно с левой стороны ПЛК или подключаться к левой стороне высокоскоростного входного адаптера FX3U-4HSX-ADP.

2.8 Модули связи, платы расширения и специальные адаптеры

Во многих приложениях от системы ПЛК требуется больше, чем просто контролировать входы и управлять выходными устройствами. Может оказаться необходимым собирать данные с или пересылать данные на другое периферийное устройство, ПЛК или ПК. Одного порта программирования может быть недостаточно.

В серии FX имеется несколько опций, которые могут использоваться для расширения коммуникационных возможностей и повышения гибкости системы FX. Модули для соединения с сетью будут обсуждаться в последующих разделах данного Руководства. В этом разделе рассмотрим только модули, платы расширения и специальные адаптеры, которые используются для расширения возможностей последовательной передачи данных в ПЛК.

ПЛК серии FX3U поддерживают добавление двух последовательных портов через левую шину. Если применяется коммуникационная BD плата, можно использовать только один дополнительный коммуникационный ADP модуль. Если применяется плата FX3U-CNV-BD, можно использовать два дополнительных коммуникационных ADP модуля. Коммуникационные устройства BD и ADP для FX1S, FX1N, FX2N и FX2NC не отражены в этом документе.

Адаптер и модуль интерфейса связи RS-232

Чтобы добавить порт связи RS-232 к ПЛК серии FX3U, необходимо подключить **FX3U-232ADP**, **FX3U-232ADP-MB**, **FX2N-232IF** или **FX3U-232-BD**.

Для FX1S, FX1N, FX2N и FX2NC будет работать **FX2NC-232ADP**. Однако чтобы использовать FX2NC-232ADP в ПЛК, отличных от FX2NC, требуется FX*N-CNV-BD. Имеются также FX1N и FX2N 232-BD.

FX3U-232ADP можно подключить к левой стороне ПЛК через левую расширительную шину. Для этого модуля может потребоваться специальный код языка релейных диаграмм и/или настройка параметров в ПЛК, чтобы конфигурировать порт связи. Данные могут передаваться и приниматься с помощью команды RS или RS2.

FX3U-232ADP-MB обладает той же функциональностью, что и FX3U-232ADP, но с добавленной поддержкой Modbus-связи при использовании с ПЛК серии FX3U с версией микропрограммного обеспечения 2.40 или выше. Возможности RS-232 Modbus в серии FX позволяют ПЛК серии FX3U быть главной станцией для одной подчиненной станции (с номером 0x01-0x10), или подчиненной станцией в любой существующей сети Modbus.

FX2N-232IF подключается к правой шине ПЛК так же, как любой другой специальный функциональный модуль. Для инициализации модуля и настройки его параметров требуется программирование ПЛК с использованием команд TO/FROM (см. ниже). Модуль можно конфигурировать, чтобы он автоматически преобразовывал данные между принимаемым/передаваемым ASCII-форматом и бинарным или двоично-десятичным кодом данных, которые используются в ПЛК. С этим модулем не могут использоваться команды RS и RS2. Он не считается одним из двух дополнительных последовательных портов, упомянутых выше.

RS-232 коммуникационная плата – FX3U-232-BD

Эта плата подключается к порту, расположенному на левой стороне ПЛК серии FX3U. В отличие от с FX2N-232IF и FX3U-232ADP, при подключении в этом месте не требуется никакого дополнительного пространства в системе ПЛК.

Как и для 232ADP, для данной платы может понадобиться специальный код языка релейных диаграмм и/или настройка параметров в ПЛК, чтобы конфигурировать порт связи. Если используется открытый протокол, для передачи и приема данных требуется команда RS или RS2.

RS-422 коммуникационная плата – FX3U-422-BD

Серия HMI (Интерфейс человек-машина) Мицубиси обычно подключается к ПЛК серии FX через порты программирования. Если программатор необходимо соединить с программой ПЛК без разъединения HMI, то простейшие способы достижения этой цели – прозрачный режим (через HMI) или коммуникационная плата. Плата просто добавляет второй порт программирования на ПЛК, не требуя никакого кода на языке релейных диаграмм или настройки параметров. Учтите, что HMI может взаимодействовать с ПЛК через этот порт и через встроенный порт программирования. Имеются также FX1N и FX2N 422-BD.

USB коммуникационная плата – FX3U-USB-BD

Эта плата добавит стандартный гнездовой порт USB Mini-B на переднюю панель ПЛК FX3U, позволяя подсоединить ПК к ПЛК через любой стандартный USB гнездовой порт типа A. Необходимые драйверы для ПК и высокоскоростной 3-метровый кабель USB 2.0 прилагаются к плате.

2.9 Сетевые модули, платы расширения и специальные адаптеры

ПЛК серии FX существует несколько возможностей работы с сетями, которые можно добавить к главному блоку. Они позволяют ПЛК работать в гораздо большей системе, чем это было бы возможно с главным блоком ПЛК FX. Могут иметь место ограничения на количество и комбинацию подключаемых модулей (см. более подробную информацию в руководствах пользователя).

RS-485 коммуникационная плата и интерфейсный адаптер

Коммуникационная плата **FX3U-485-BD** подключается к порту, расположенному на левой стороне ПЛК FX3U, и предоставляет несколько опций для работы с последовательными сетями. Один из примеров – работа с сетями N:N. Это многоабонентская сеть, основанная на принципе ведущий-ведомый. Она позволяет подключать до 8 станций к одной сети. Прочие имеющиеся сети связи включают Parallel Link, Computer Link, Inverter Communication, и Non-Protocol Communication, использующие команды языка релейных диаграмм для организации связи между ПЛК FX3U и ассортиментом периферийных устройств. Эта плата занимает один последовательный порт FX3U.

FX3U-485ADP обладает теми же функциональными возможностями, что и FX3U-485-BD, но сигнал большей амплитуды позволяет увеличить длину сети. Этот адаптер также занимает один последовательный порт FX3U.

FX3U-485ADP-MB обладает той же функциональностью, что и FX3U-485ADP, но с добавленной поддержкой Modbus-связи при использовании с ПЛК серии FX3U с версией микропрограммного обеспечения 2.40 или выше. Возможности RS-485

Modbus в серии FX позволяют ПЛК серии FX3U быть главной станцией в сети, включающей до 16 подчиненных станций (с номером 0x01-0x10), или подчиненной станцией в любой существующей сети Modbus. Этот адаптер также занимает один последовательный порт FX3U.

Для FX1S, FX1N, FX2N, и FX2NC, имеется адаптер для подключения **FX2NC-485ADP** поRS-485, который требует использования платы преобразователя FX*N-CNV-BD для подключения к ПЛК (кроме FX2NC). Имеются также FX1N и FX2N 485-BD. Все они занимают один последовательный порт.

Ведущий модуль AS-I – FX2N-32ASI-M

AS-I – экономичная электромеханическая система соединения, предназначенная для работы по двухпроводному кабелю, и передающая данные и электропитание на расстояние до 100 м или больше, при использовании ретрансляторов. Она особенно удобна на нижних уровнях автоматизации предприятия, где простые – часто бинарные – полевые устройства (например, выключатели) должны работать в автономной сети автоматизации локального участка, управляемого ПЛК или ПК. AS-интерфейс может считаться наилучшей цифровой заменой традиционных архитектур кабельной разводки.

Этот модуль позволяет ПЛК FX2N, FX2NC или FX3U использовать AS-I сеть, чтобы управлять и контролировать до 31 полевых устройств.

Дополнительную информацию о AS-I см. на сайте www.as-interface.com

I/O Link – FX2N-16LNK-M

I/O Link – распределенная система ввода-вывода для ПЛК Мицубиси серии FX. Модули ввода-вывода можно поместить на расстоянии до 200 м от основной системы и адресовать как стандартные устройства ввода-вывода согласно ограничениям ЦП. Каждый ведущий модуль I/O Link может управлять до 128 точками ввода/вывода на до 16 станций.

Ведущий модуль CC-Link – FX2N-16CCL-M

Как правило, CC-Link является силовой сетью удаленного ввода-вывода, хотя ее можно также использовать для соединения локальных станций. В CC-Link используются кабели на витой паре и поддерживается дальность 1200 м с максимальной скоростью передачи данных 10 Мбит/с на 100 м. CC-Link также обладает быстрой скоростью обновления (3.9 мс для 64-х станций) и высокой емкостью ввода-вывода (2048 точек).

Этот модуль позволяет ПЛК FX1N, FX2N, FX2NC или FX3U быть главным устройством в сети CC-Link, содержащей до 15 подчиненных станций.

Интерфейсный модуль CC-Link – FX2N-32CCL

FX2N-32CCL позволяет ПЛК FX1N, FX2N, FX2NC или FX3U подключаться к сети CC-Link в качестве удаленной станции. Главное устройство CC-Link управляет ПЛК серии FX, записывая данные в буферную память FX2N-32CCL, который затем, в свою очередь, передает данные на и из ПЛК.

Интерфейсный модуль CC-Link/LT – FX2N-64CL-M

CC-Link/LT – высокоскоростная сеть удаленного ввода-вывода, использующая специализированную кабельную сеть длиной до 500 м, обновляя до 1024 точек ввода-вывода за 2 мс.

Этот модуль позволяет ПЛК FX1N, FX2N, FX2NC или FX3U быть главным устройством в сети CC-Link/LT, управляя до 256 точками ввода-вывода.

Ведущий модуль Profibus – FX3U-64DP-M

Данный модуль работает только с серией ПЛК FX3U и является полнофункциональным ведущим модулем для сетей Profibus DP-V1. С этим модулем FX3U может быть главным устройством в сети Profibus с 64 подчиненными станциями, поддерживая максимальную скорость 12 Мбит/с на расстояниях до 100 м.

Profibus является сетью с открытым протоколом, поддерживающей многие устройства различных компаний. Дополнительную информации о Profibus см. на сайте www.profibus.org.

Profibus Interface Modules – FX3U-32DP & FX0N-32NT-DP

Оба этих модуля позволяют подключенному ПЛК серии FX быть ведомым в сети Profibus, однако FX3U-32DP позволяет ПЛК FX3U связываться, используя Profibus DP-V1, а FX0N-32NT-DP позволяет ПЛК FX1N, FX2N, FX2NC или FX3U связываться, используя Profibus DP-V0. FX3U-32DP позволяет передавать данные до 144 байтов, используя циклическую связь, или 140 байтов, используя нециклическую связь. FX0N-32NT-DP позволяет передавать данные до 40 байтов, используя только циклическую связь.

Интерфейсный модуль PROFIBUS – FX2N-32DP-IF

FX2N-32DP-IF используется для замены ЦП в ПЛК серии FX, позволяя конфигурировать расширительные блоки (ввода-вывода) и специальные модули серии FX как устройства удаленного ввода-вывода в сети Profibus, используя Profibus DP-V0.

Интерфейсные модули Ethernet – FX2NC-ENET-ADP & FX3U-ENET

FX2NC-ENET-ADP можно подключить к ПЛК FX1S, FX1N, FX2N, и FX2NC. Для всех, кроме FX2NC потребуется FX*N-CNV-BD. FX2NC-ENET-ADP – простой шлюз последовательный порт – Ethernet для ПЛК серии FX. GX Developer v8.25 и выше поддерживают Ethernet соединение с ПЛК серии FX через этот модуль. Для ранних версий программного обеспечения необходимы дополнительные драйверы.

FX3U-ENET работает только с ПЛК серии FX3U и является полнофункциональным модулем Ethernet. FX3U-ENET поддерживает до 8 одновременных соединений с ПЛК серии FX3U для связи ПЛК-ПЛК или ПЛК-ПК, включая мониторинг, а также пересылку/загрузку параметров и релейных диаграмм. Кроме того, при подключении к серверу электронной почты FX3U-ENET имеет обширные возможности передачи данных и текста по электронной почте.

2.10 Прочие аппаратные модули

Дисплейный модуль – FX3U-7DM & FX3U-7DM-HLD

Дисплейный модуль FX3U-7DM можно установить непосредственно на переднюю панель ПЛК серии FX3U, предоставив оператору текстовый дисплей 4 строки x 16 символов. Он позволяет оператору просматривать часы ПЛК, просматривать или изменять адреса данных в ПЛК, проверять ошибки, показывать состояние ПЛК, а также отображать пользовательские текстовые сообщения. Модуль также может быть защищен паролем. Установочный адаптер продается отдельно, что позволяет смонтировать устройство через вырез в шкафу или дверце.

2.11 Источники питания

Все ПЛК серии FX имеют встроенный блок электропитания. Блок электропитания принимает напряжения 85-264 В≈ или 24 В= (12В= на выбранных моделях FX1N).

Блок электропитания FX генерирует два типа питания шины: 5 В= и 24 В=. Значение тока для каждого блока питания изменяется в зависимости от модели и размера ПЛК. Более подробные сведения см. в руководстве по аппаратному обеспечению рассматриваемого продукта.

Шина 5 В= обеспечивает электропитанием ЦП и расширительные блоки. Блок питания имеет ограниченную мощность. Это ограничивает количество подключаемых расширительных блоков.

Напряжение 24 В= также подается на клеммы 0 В и 24 В на ПЛК. Это электропитание можно использовать для дискретных расширений ввода-вывода, а также

для питания принадлежностей, например, датчиков и HMI. Будьте внимательны, не превышайте номинальную мощность блока питания.

Как вычислить мощность, потребляемую от блока питания

Как отмечалось выше, встроенный блок питания может поддерживать только определенное количество блоков расширения и специальных функциональных модулей.

К ПЛК серии FX3U можно подключить дополнительный блок питания – **FX3U-1PSU-5V**, обеспечивающий шине дополнительную мощность. Он создает 1 А в шине 5 В= и 300 мА в шине 24 В=. Рекомендуется использовать этот модуль после дискретных расширительных блоков ввода-вывода, поскольку они потребляют высокую мощность в шине 24 В=, превышающую возможности встроенного блока питания. В одной системе FX3U можно использовать до двух дополнительных источников питания.

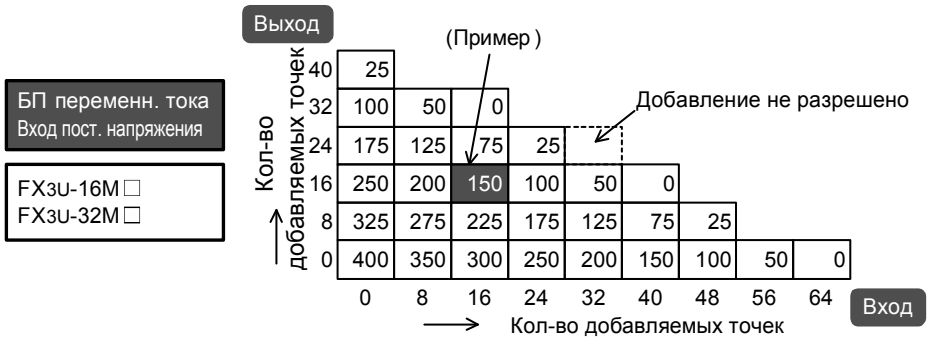
Ниже описана процедура, позволяющая определить допустимые конфигурации системы.

1) Подсчитайте полное количество модулей ввода и вывода без собственного источника питания, которые должны быть **добавлены** к главному блоку. Учтите, что когда используется модуль расширения с блоком питания, все подключенные к нему точки ввода-вывода рассчитываются как отдельная система. Каждые 16 дополнительных точек ввода требуют 100 мА от шины +24 В=; каждые 16 дополнительных точек вывода требуют 150 мА от шины +24 В=. См. таблицу 1 или 2, где указан тип ПЛК, или таблицу 3 или 4, где приведен тип модуля расширения с блоком питания. Задайте нужные значения в колонке ввода и колонке вывода. Найденное число является остаточным током в шине +24 В=.
2) См. таблицу 5, чтобы определить доступный ток в шине 5 В=. В таблицах A1 и A2 приведены главные блоки FX3U, а в таблице D1 – модули расширения с собственным блоком питания. При необходимости возможности шины 5 В постоянного тока FX3U можно расширить, подключив FX3U-1PS-5V.
3) Учитывайте все дискретные расширительные блоки ввода-вывода, специальные модули, опциональные платы расширения и специальные адаптеры в системе. Для каждого найдите номинальный потребляемый ток для шин 5 В= и 24 В= в таблице 6. Если в таблице указано внешнее значение 24 В=, то устройство имеет клеммы для подключения к внешнему источнику электропитания. В этом случае устройство не добавляется в сводку внутреннего потребляемого тока в шине 24 В=, если к выходным клеммам 24 В= на ПЛК подведен внешний источник электропитания.
4) Сложите значения, чтобы вычислить полный потребляемый ток, и вычтите его из полного поставляемого тока источника питания тока.

ПРИМЕЧАНИЕ Эти таблицы приведены в главе 6 Руководства по аппаратной части FX3U.

ТАБЛИЦА 1: FX3U (блоки с 16 или 32 точками ввода-вывода)

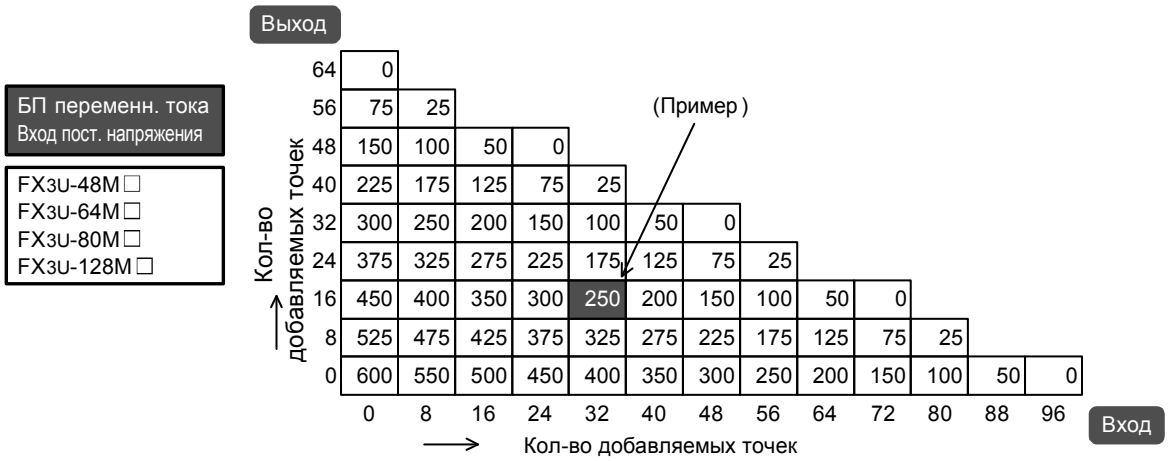
FX3U-16MR/ES, FX3U-16MT/ES, FX3U-16MT/ESS, FX3U-32MR/ES, FX3U-32MT/ES, FX3U-32MT/ESS



(Пример) Когда блок расширения, содержащий 16 точек ввода и 16 точек вывода, подключен к FX3U-16/32M□, из шины питания 24 В= потребляется ток 150 мА или меньше.

ТА

БЛИЦА 2: FX3U (48 точек ввода-вывода или больше)



(Пример) Когда блок расширения, содержащий 32 точки ввода и 16 точек вывода, подключен к FX3U-48~128M□, из шины питания 24 В= потребляется ток 250 мА или меньше.

ТАБЛИЦА 3: FX2N-32E*

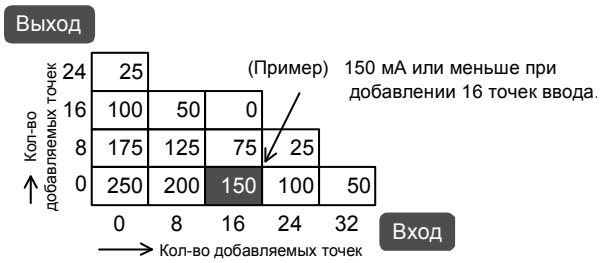


ТАБЛИЦА 4: FX2N-48E*

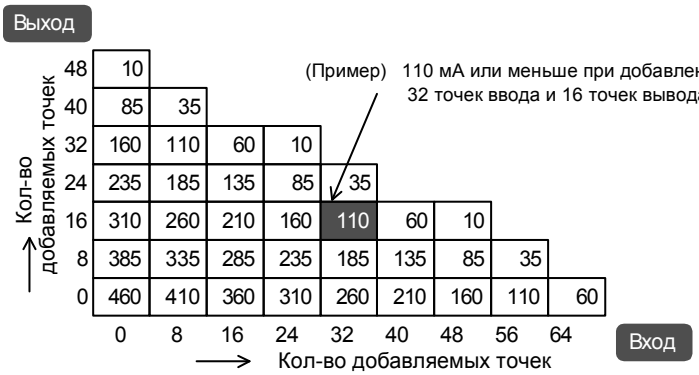


ТАБЛИЦА 5: Нагрузочная способность для шин 24 В= и 5 В=

Модули ЦП FX3U (с собственным блоком питания переменного тока)

№	Тип	Входы/выходы		Выходной ток (мА)	
		Кол-во точек ввода-выхода [адресов]	Входы/выходы [адреса]	Электро-питание 5 В=	Шина питания 24 В=
БП переменного напряжения/вход 24В=/релейный выход					
A1	FX3U-16MR/ES	16	8/8	500	400
	FX3U-32MR/ES	32	16/16		
	FX3U-48MR/ES	48	24/24		600
	FX3U-64MR/ES	64	32/32		
	FX3U-80MR/ES	80	40/40		
	FX3U-128MR/ES	128	64/64		
БП переменного напряжения/вход 24В=/транзисторный выход					
A1	FX3U-16MT/ES	16	8/8	500	400
	FX3U-16MT/ESS	16	8/8		
	FX3U-32MT/ES	32	16/16		
	FX3U-32MT/ESS	32	16/16		
	FX3U-48MT/ES	48	24/24		600
	FX3U-48MT/ESS	48	24/24		
	FX3U-64MT/ES	64	32/32		
	FX3U-64MT/ESS	64	32/32		
	FX3U-80MT/ES	80	40/40		
	FX3U-80MT/ESS	80	40/40		
	FX3U-128MT/ES	128	64/64		
	FX3U-128MT/ESS	128	64/64		

Модули ЦП FX3U (с собственным блоком питания постоянного тока)

№	Тип	Входы/выходы		Выходной ток (мА)	
		Кол-во точек ввода-выхода [адресов]	Входы/выходы [адреса]	Электро-питание 5 В=	Нагрузочная способность для внутр. 24В=
БП постоянного напряжения/вход 24В=/релейный выход					
A2	FX3U-16MR/DS	16	8/8	500	400*1
	FX3U-32MR/DS	32	16/16		
	FX3U-48MR/DS	48	24/24		600*2
	FX3U-64MR/DS	64	32/32		
	FX3U-80MR/DS	80	40/40		
БП постоянного напряжения/вход 24В=/транзисторный выход					
A2	FX3U-16MT/DS	16	8/8	500	400*1
	FX3U-16MT/DSS	16	8/8		
	FX3U-32MT/DS	32	16/16		
	FX3U-32MT/DSS	32	16/16		
	FX3U-48MT/DS	48	24/24		600*2
	FX3U-48MT/DSS	48	24/24		
	FX3U-64MT/DS	64	32/32		
	FX3U-64MT/DSS	64	32/32		
	FX3U-80MT/DS	80	40/40		
	FX3U-80MT/DSS	80	40/40		

Модули расширения FX2N с собственным блоком питания

№	Тип	Входы/выходы		Выходной ток (мА)	
		Кол-во точек ввода-выхода [адресов]	Входы/выходы [адреса]	Электропитание 5 В=	Шина питания 24 В=
D1	FX2N-32ER-ES/UL	32	16/16	690	250
	FX2N-32ET-ESS/UL	32	16/16		
	FX2N-32ER	32	16/16		
	FX2N-32ES	32	16/16		
	FX2N-32ET	32	16/16		
	FX2N-48ER-ES/UL	48	24/24		460
	FX2N-48ET-ESS/UL	48	24/24		
	FX2N-48ER	48	24/24		
	FX2N-48ES	48	24/24		
	FX2N-48ET	48	24/24		
	FX2N-48ER-DS	48	24/24		–
	FX2N-48ET-DSS	48	24/24		
	FX2N-48ER-D	48	24/24		
	FX2N-48ET-D	48	24/24		

ТАБЛИЦА 6: Устройства расширения

Опциональные платы расширения FX3U

№	Тип	Кол-во занятых точек ввода-выхода [адресов]	Потребляемый ток (мА)	
			5 В=	Внутр. 24 В=
B1	FX3U-232-BD	–	20	–
	FX3U-422-BD	–	20*1	–
	FX3U-485-BD	–	40	–
	FX3U-USB-BD	–	15	–
	FX3U-CNV-BD	–	–	–

Адаптеры левой шины FX3U

№	Тип	Кол-во занятых точек ввода-выхода [адресов]	Потребляемый ток (мА)		
			5 В=	Внутр. 24 В=	Внешн. 24 В=
C1	FX3U-4HSX-ADP	–	30	30	0
	FX3U-2HSY-ADP	–	30	60	0
C2	FX3U-4AD-ADP	–	15	0	40
	FX3U-4DA-ADP	–	15	0	150
	FX3U-4AD-PT-ADP	–	15	0	50
	FX3U-4AD-TC-ADP	–	15	0	45
C3	FX3U-232ADP	–	30	0	0
	FX3U-485ADP	–	20	0	0

Блоки расширения FX2N без собственного блока питания

№	Тип	Кол-во точек ввода-выхода	Потребляемый ток (мА)		
			5 В=	Внутр. 24 В=	Внешн. 24 В=
D2	**Типы для добавления точек ввода/вывода**				
	FX2N-8ER-ES/UL	**16**	**–**	**125**	**0**
	FX2N-8ER	**16**	**–**	**125**	**0**
	Типы для добавления точек ввода				
	FX2N-8EX-ES/UL	**8**	**–**	**50**	**0**
	FX2N-8EX	**8**	**–**	**50**	**0**
	FX2N-8EX-UA1/UL	**8**	**–**	**50**	**0**
	FX2N-16EX-ES/UL	**16**	**–**	**100**	**0**
	FX2N-16EX	**16**	**–**	**100**	**0**
	FX2N-16EX-C	**16**	**–**	**100**	**0**
	FX2N-16EXL-C	**16**	**–**	**100**	**0**
	Типы для добавления точек вывода				
	FX2N-8EYR-ES/UL	**8**	**–**	**75**	**0**
	FX2N-8EYT-ESS/UL	**8**	**–**	**75**	**0**
	FX2N-8EYR	**8**	**–**	**75**	**0**
	FX2N-8EYT	**8**	**–**	**75**	**0**
	FX2N-8EYT-H	**8**	**–**	**75**	**0**
	FX2N-16EYR-ES/UL	**16**	**–**	**150**	**0**
	FX2N-16EYT-ESS/UL	**16**	**–**	**150**	**0**
	FX2N-16EYR	**16**	**–**	**150**	**0**
	FX2N-16EYS	**16**	**–**	**150**	**0**
	FX2N-16EYT	**16**	**–**	**150**	**0**
	FX2N-16EYT-C	**16**	**–**	**150**	**0**

Специальные функциональные модули

№	Тип	Кол-во точек ввода/занятых точек вывода	Потребляемый ток (мА)		
			5 В=	Внутр. 24 В=	Внешн. 24 В=
E1	FX**3U**-4AD	8	110	0	90
	FX**3U**-4DA	8	120	0	160
	FX**3U**-20SSC-H	8	100	0	220
E2	FX**2N**-2AD	8	20	50[*6]	0
	FX**2N**-2DA	8	30	85[*6]	0
	FX**2N**-4AD	8	30	0	55
	FX**2N**-4DA	8	30	0	200
	FX**2N**-4AD-TC	8	30	0	50
	FX**2N**-4AD-PT	8	30	0	50
	FX**2N**-8AD	8	50	0	80
	FX**2N**-5A	8	70	0	90
	FX**2N**-2LC	8	70	0	55
	FX**2N**-1HC	8	90	0	0
	FX**2N**-1PG(-E)	8	55	0	40
	FX**2N**-10PG	8	120	0	70[*1]
	FX**2N**-232IF	8	40	0	80
	FX**2N**-16CCL-M	8[*2]	0	0	150
	FX**2N**-32CCL	8	130	0	50
	FX**2N**-64CL-M	8[*3]	190	Питается от источника питания для CC-Link/LT	
	FX**2N**-16LNK-M	0[*4]	200	0	90
	FX**2N**-32ASI-M	8[*5]	150	0	70
E3	FX**0N**-3A	8	30	90[*6]	0

Специальные функциональные модули (Продолжение)

№	Тип	Кол-во точек ввода/занятых точек вывода	Потребляемый ток (мА)		
			5 В=	Внутр. 24 В=	Внешн. 24 В=
E3	FX2N-10GM	8	–	–	5
	FX2N-20GM	8	–	–	10
	FX2N-1RM(-E)-SET	8	–	–	5

Дисплейный модуль FX3U

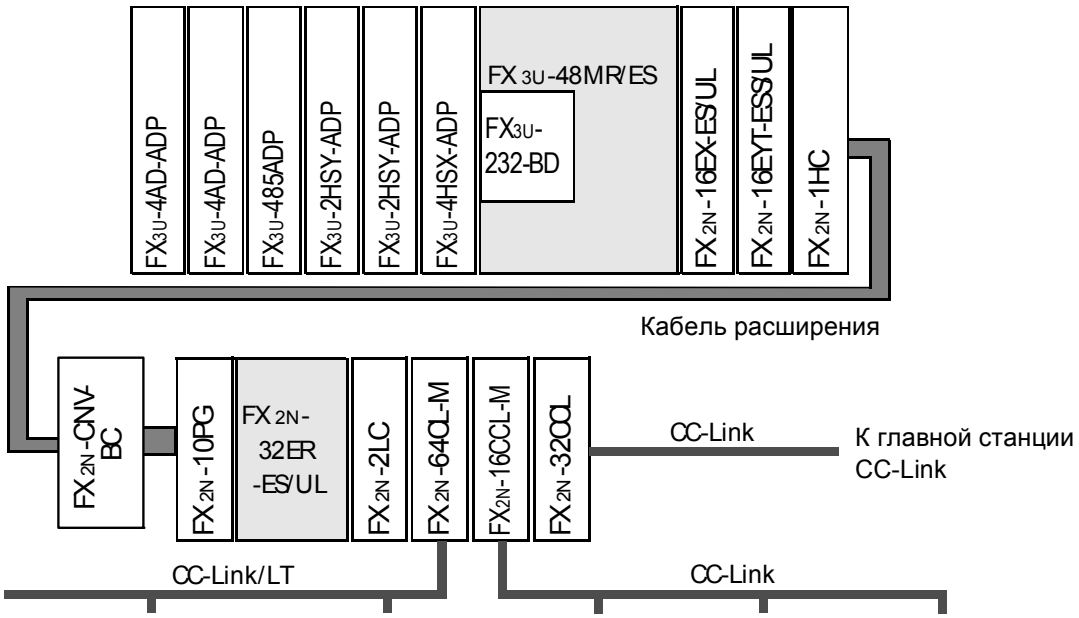
№	Тип	Кол-во точек ввода/занятых точек вывода	Потребляемый ток (мА)		
			5 В=	Внутр. 24 В=	Внешн. 24 В=
G1	FX3U-7DM	–	20	0	0

Таблица для упражнения 2.12

	Классификация	Количество подключенных модулей	Тип	Кол-во точек ввода-выхода [адресов]	Нагрузочная способность встроенного блока питания	
					Электропитание 5 В= [мА]	Шина питания 24 В= [мА]
				1-1	1-2	1-3
Со встроенным блоком питания	A Главный блок	1	FX3U-48MR/ES	48	500	600

	Классификация	Количество подключенных модулей	Тип	Кол-во (занятых) точек ввода-выхода [адресов]	Вычисление тока, потребляемого от встроенного блока питания	
					Электропитание 5 В= [мА]	Внутр. 24 В= [мА]
Введите продукты, подключенные к главному блоку.	B: **Плата расширения**	**1**	**FX3U-**	–		–
	C **Специальный адаптер**	**10**	**FX3U-**	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
			FX3U-	–		
	D2 **Блок расширения ввода-вывода**	–	**FX2N-**		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
			FX2N-		–	
	E **Специальное функциональное устройство / блок**	**8**	**FX0N/FX2N/FX3U-**			
			FX0N/FX2N/FX3U-			
			FX0N/FX2N/FX3U-			
			FX0N/FX2N/FX3U-			
			FX0N/FX2N/FX3U-			
			FX0N/FX2N/FX3U-			
			FX0N/FX2N/FX3U-			
			FX0N/FX2N/FX3U-			
	G **Дисплейный модуль**	**1**	**FX3U-7DM**	–		–
				2 -1	2 -2	2 -3
Рассчитайте суммарное значение.						

2.12 Упражнение Расчет электропитания



Используйте таблицы на предыдущих страницах, а также обсуждавшиеся выше этапы, чтобы определить потребляемую мощность указанной выше системы.

1. Является ли эта конфигурация допустимой? _____

2. Если нет, почему? _____

3. Если нет, как можно ее исправить? _____

_____

_____

2.13 Типы памяти

Встроенная внутренняя память ПЛК серии FX не расширяема. Однако для каждой модели имеются кассеты памяти, которые могут использоваться вместо внутренней памяти для хранения программы, иногда даже если кассета памяти имеет большую емкость, чем внутренняя память ПЛК. Кассеты памяти могут также повышать мобильность программы. Программа сохраняется в кассете, и когда кассета подключается к другому ПЛК, релейная диаграмма в кассете обрабатывается ЦП вместо сохраняемого экземпляра в памяти для хранения программы ПЛК. С помощью опции загрузчика программ специальные карты памяти позволяют считывать карту памяти или перезаписывать программу в ПЛК.

Отметим, что поскольку ПЛК использует кассету вместо памяти для хранения программы, кассета памяти **не является кумулятивной** с программой памяти ПЛК.

В зависимости от типа ПЛК, выделяют четыре типа кассет памяти: **RAM**, **EPROM**, **EEPROM** и **FLROM**. Каждый тип имеет свои преимущества и недостатки. Вид используемой памяти определяется типом ПЛК, приложением и необходимым уровнем безопасности.

RAM (Оперативная память)

Оперативная память является энергозависимой, т.е. для сохранения программы в памяти необходима батарея. В программу, сохраняемую в оперативной памяти, несложно вносить изменения. Оперативная память позволяет изменять программу в режиме online.

EPROM (Стираемое программируемое постоянное запоминающее устройство)

Память типа EPROM является энергонезависимой, т.е. память для хранения программы сохраняется без батареи. В программу, сохраняемую в EPROM, сложно вносить изменения, потому что для ее стирания необходим УФ свет. Требуется программатор EPROM. Изменения в режиме online НЕ разрешены.

EEPROM (Электрически стираемое программируемое ПЗУ)

Память типа EEPROM также является энергонезависимой, т.е. программа сохраняется без подключения батареи. Однако, в память типа EEPROM несложно вносить изменения, потому что она стирается электрически. Изменения в режиме online разрешены в FX1N и выше.

FLROM (Flash ROM)

Память типа FLROM также является энергонезависимой – для сохранения программы не требуется батарея. Память типа FLROM работает точно так же, как EEPROM, она является электрически стираемой и может многократно перезаписываться. Память типа Flash ROM используется только в FX3U.

В таблице ниже показана емкость внутренней памяти, типы кассет памяти, и значения емкости кассет памяти, имеющихся для каждой модели ПЛК серии FX.

Тип ПЛК	Внутренняя	RAM	EPROM	EEPROM	FLROM
FX1S	2 K	-	-	2K с загрузчиком	-
FX1N	8 K	-	-	8 K с загрузчиком	-
FX2N	8 K	16 K	16 K	4 K, 8 K, 16 K	-
FX2NC	8 K	-	-	4 K, 16 K, часы реальн. времени*	-
FX3U	64 K	-	-	-	16 K, 64 K, 64 K с загрузчиком*

Примечания

- В FX2NC не имеется внутренних часов реального времени. Если требуется функция часов реального времени, необходимо подключить модуль памяти с часами реального времени.
- FX-RAM-8 и FX-ROM-8 фактически хранят 16 кШагов памяти.
- FX3U-FLROM-64L должен использоваться для обеспечения функции загрузчика программ на FX3U.

Выбирая тип памяти, задайте себе несколько вопросов:

1) Каков тип используемого ПЛК и какие опции имеются?
2) Должна ли сохраняться программа даже при потере батареи питания?
3) Должна ли программа легко изменяться?

ГЛАВА 3 – Программирующее оборудование

В этой главе обсуждаются требования к аппаратным средствам и программному обеспечению для программирования ПЛК. Слушателям будет показано, как соединить систему. Также рассмотрены альтернативы программированию с помощью ноутбука и ПК.

Цели главы: Завершив эту главу, слушатели смогут…

- Перечислить аппаратные средства, необходимые для программирования ПЛК с помощью ноутбука или ПК.
- Описать альтернативы использованию ноутбука или ПК для программирования.
- Описать, как подключить систему ПЛК к ноутбуку или ПК.
- Перечислить типы программного обеспечения, используемые для программирования ПЛК.

Материалы: Руководство по обучению FX-TRN-KIT-R

3.1 Ручные программаторы

Большинство программистов используют ноутбук или ПК для изменения или доступа к своей программе ПЛК. Однако вследствие факторов экономичности, безопасности или удобства возможны и другие варианты.

FX-10P-E и **FX-20P-E** могут быть полезны в ситуациях, когда необходимо лишь изменить таймер, добавить контакт или создать и модифицировать адреса. Эти карманные модули с ЖК дисплеем подключаются непосредственно к ПЛК и питаются от ПЛК. Они позволяют программировать изменения и производить мониторинг.

FX-10P-E – этот модуль позволяет вносить изменения в программу только в режиме online. Дисплей содержит 2 строки по 16 символов. Он может контролировать программу и изменять состояние операндов (принудительно устанавливая биты).

FX-20P-E – этот модуль является более эффективной версией FX-10P-E с подсвечиваемым дисплеем 4 строк по 16 символов. Он поддерживает программирование в режиме online и offline с помощью специализированной кассеты памяти FX-20P-MXF*, т.е. может как считывать, так и записывать программы ПЛК . До 16 кШагов памяти может быть запрограммировано в режиме offline.

Учтите, что как FX-10P-E, так и FX-20P-E поддерживают функциональность ПЛК только до FX2N. Таким образом, могут иметь место ограничения на емкость памяти и набор команд.

3.2 Программное обеспечение для программирования

Имеются несколько пакетов программ, которые используются для программирования и отладки всех ПЛК Мицубиси. Из них чаще всего используются три пакета – GX Developer, GX IEC Developer и GX Simulator. Некоторым пользователям могут быть знакомы предыдущие пакеты программ, включая MEDOC (приложение DOS), FX-WIN, и GPP-Win (предыдущая версия GX Developer). На этих курсах мы будем иметь дело только с последней версией GX Developer.

GX Developer – программа, работающая под Windows (95, 98, NT, 2000, XP), которая используется для программирования всех серий ПЛК Мицубиси, включая серии FX, Q, A и Motion A. Это программное обеспечение заменяет предыдущий пакет для Windows, GPP-WIN, имеет большое число отладочных и диагностических возможностей, а также простые сетевые настройки. Пакет также включает возможности импорта, позволяющие использовать программы, написанные с использованием старых версий программного обеспечения.

GX IEC Developer – программа, работающая под Windows (95, 98, NT, 2000, XP) с возможностями, во всех отношениях аналогичными GX Developer, но использующая стандарты программирования IEC 1131.3. Это программное обеспечение строит дерево проекта и программирует, используя встроенные и/или пользовательские функциональные блоки. Преимущество функциональных блоков заключается в возможности простого повторного использования многих экземпляров одинакового функционального блока во всех программах, что сокращает время разработки и повышает производительность.

GX Developer-FX – программа, работающая под Windows (95, 98, NT, 2000, XP) на базе программного пакета GX Developer. Это программное обеспечение имеет все особенности GX Developer, но поддерживает только ПЛК серии FX и их функциональные возможности. Данный программный пакет предлагается по цене со скидкой по сравнению с полным пакетом GX Developer, как экономичное решение для людей, которым не требуется программировать весь ассортимент моделей ПЛК Мицубиси.

GX IEC DEVELOPER-FX – программа, работающая под Windows (95, 98, NT, 2000, XP) на базе программного пакета GX Developer. Это программное обеспечение имеет все особенности GX IEC Developer, но поддерживает только ПЛК серии FX и их функциональные возможности. Как и GX Developer-FX, этот пакет предлагается по цене со скидкой, также как экономичное решение для тех, кому не требуется программировать весь ассортимент моделей ПЛК Мицубиси.

GX Simulator – программа, работающая под Windows (95, 98, NT, 2000, XP) не используется для программирования ПЛК; она помогает отлаживать программу. Это программное обеспечение эмулирует ПЛК на ПК, позволяя проверять работу пакетов GX Developer или GX IEC Developer без необходимости загрузки на реальный ПЛК. GX Simulator способен моделировать дискретный и аналоговый ввод-вывод, последовательную и сетевую связь, а также специальные функциональные модули. Программы могут частично выполняться, или частично пропускаться, или выполняться шаг за шагом.

3.3 Обзор GX Developer

В этом разделе рассматриваются важные обстоятельства, о которых следует помнить, устанавливая и используя GX Developer. Раздел не содержит полное описание особенностей GX Developer. Вы будете знакомиться с ними в ходе нашего курса.

УСТАНОВКА

ВАЖНО: Перед установкой GX Developer не забудьте удалить любую предыдущую версию GX Developer или GPP-WIN, используя утилиту "Установка или удаление программ" в Панели управления Windows . НЕ удаляйте каталоги и не пытайтесь переустановить. В ходе деинсталляции не стираются никакие созданные ранее программы ПЛК.

Когда Windows удаляет программу, она показывает сообщение, указывая, что некоторые элементы не могут быть удалены, и предлагает удалить их вручную. Это созданные ранее программы ПЛК, и пользователь может удалить их вручную по своему усмотрению.

Установочный CD с GX Developer должен включать меню автоматического запуска, которое активизируется при установке диска в ПК. Если этого не произошло, запустите программу, дважды щелкнув на файле autorun.exe, который находится в корневом каталоге установочного CD. Или откройте меню "Пуск" Windows, выберите "Выполнить" и напечатайте или найдите D:\autorun.exe, где D:\ – CD-дисковод ПК.

Очень важно прочитать запрос и ответить, потому что это является единственной возможностью установить опцию "Import from MELSEC MEDOC" (Импортировать из MELSEC MEDOC). Получив это предложение, пользователь должен щелкнуть на каждом флажке, чтобы установить его, иначе в будущем программист не сможет импортировать программы MEDOC без переустановки GX Developer.

Версия GX Developer

Версию установленного программного обеспечения можно найти в GX Developer, перейдя в справки Help и выбрав информацию о продукте Product Information. Если номер изделия (после номера версии в окне информации о продукте "Product information") начинается с SW2-SW5, то программное обеспечение первоначально имело название GPP-WIN. Начиная с SW6, программное обеспечение было переименовано в GX Developer.

ОСОБЕННОСТИ

Множественные окна

В GX Developer можно открывать несколько окон. Таким образом, можно одновременно открыть различные окна, показывающие разные блоки кода и различные мониторы.

Импорт из других форматов

Программы, написанные в MEDOC, GPPA, и FX-WIN, могут быть импортированы в GX Developer.

Сохраняются настройки рабочего пространства

Команды Save и Save As сохраняют последнее состояние программы, включая все открытые окна и их позиции. Таким образом поддерживается желательная рабочая среда; вам не нужно каждый раз создавать ее, открывая программу.

Монитор элементов релейной диаграммы

Новая возможность GX Developer – Монитор элементов релейной диаграммы (Entry Ladder Monitor) – позволяет программисту копировать элементы из различных блоков программы на один экран для упрощения мониторинга.

Монитор локальных операндов

Эта новая возможность GX Developer позволяет программисту контролировать состояния локальных операндов, (*используется только с QCPU)*

ПРЕДОСТЕРЕЖЕНИЯ

Импорт из MEDOC

В процессе импорта GX Developer записывает временный файл. Если адресат импорта защищен от записи или не имеет достаточно свободного пространства, импорт сорвется. Перед импортом рекомендуется скопировать исходный файлы на жесткий диск .

Импорт документации

Документация не будет импортироваться, если она включает иностранные символы (подобные тильде ~).

Копирование и вставка

Копирование и вставка между GX Developer и другими приложениями Windows невозможны, за исключением комментариев. Комментарии можно копировать между списком комментариев и любой программой электронных таблиц. Код языка релейных диаграмм можно копировать и вставлять между сессиями GX Developer.

Множественные проекты

В текущей сессии GX Developer можно открыть только один проект. Если требуется копирование и вставка между несколькими проектами GX Developer, необходимо открыть несколько сессий GX Developer. Это можно сделать из меню File, командой "Start New GX Developer session" (Начать новую сессию GX Developer), или второй раз запустив GX Developer из меню "Пуск".

Файлы, доступные только для чтения

GX Developer не может открывать файлы, доступные только для чтения. Если проект был архивирован на CD, CD является носителем, доступным только для чтения, и все файлы на CD будут доступны только для чтения. Если скопировать эти файлы на жесткий диск, они все еще будут иметь атрибут "только для чтения". Пользователи должны модифицировать эти файлы, разрешив считывание/запись на вкладке атрибутов в Windows, лишь после этого GX Developer увидит папки как проект. Это относится к каталогу проекта, всем подкаталогам и всем файлам.

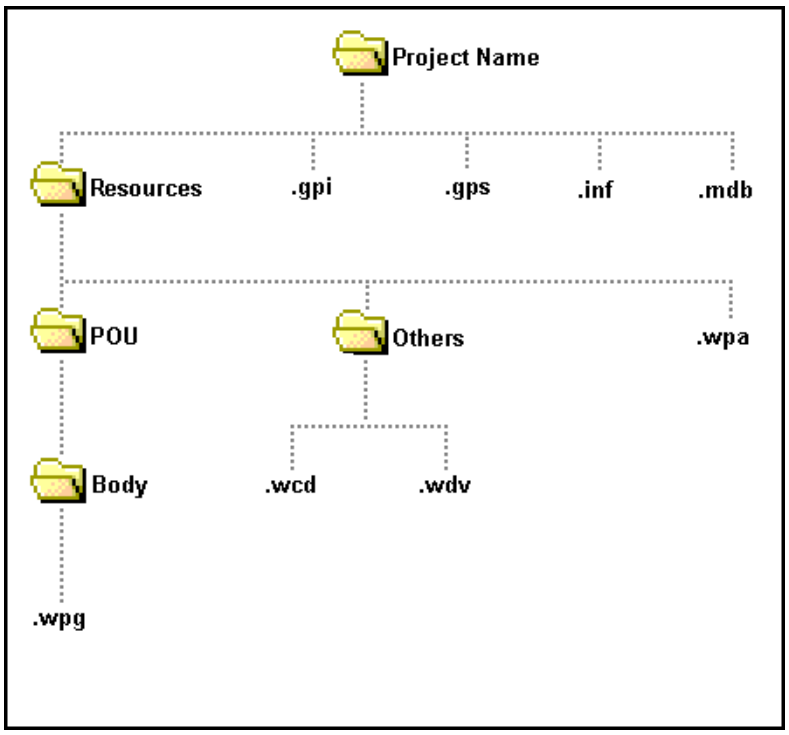
Архивирование программы для распределения

Проект GX Developer имеет специальный формат папок и размещения файлов. Хотя есть возможность вручную обновлять этот формат, отправляя программу по электронной почте, желательно сделать это командой Save As и архивировать zip весь каталог. Это сохранит формат проекта.

Утилиты дефрагментации диска

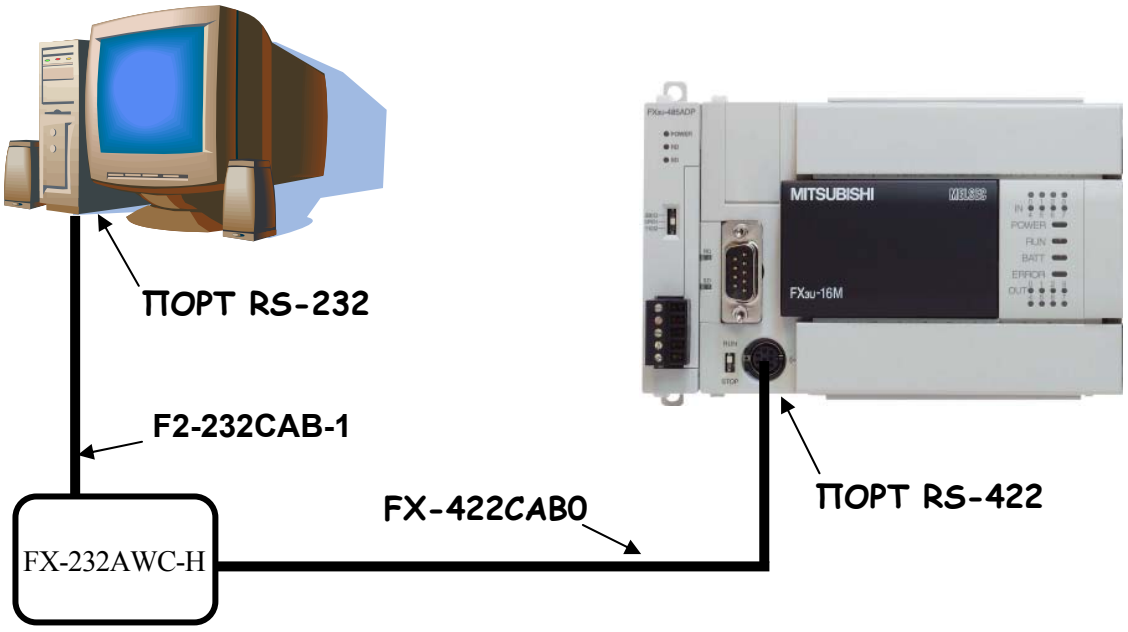
Использование утилита дефрагментация диска 3^{их} компаний может повредить лицензию GX Developer. Каталог GPPW необходимо исключить из процесса дефрагментации диска. Утилита дефрагментация, входящая в состав Windows, может выполняться без ограничений.

3.4 Формат файлов



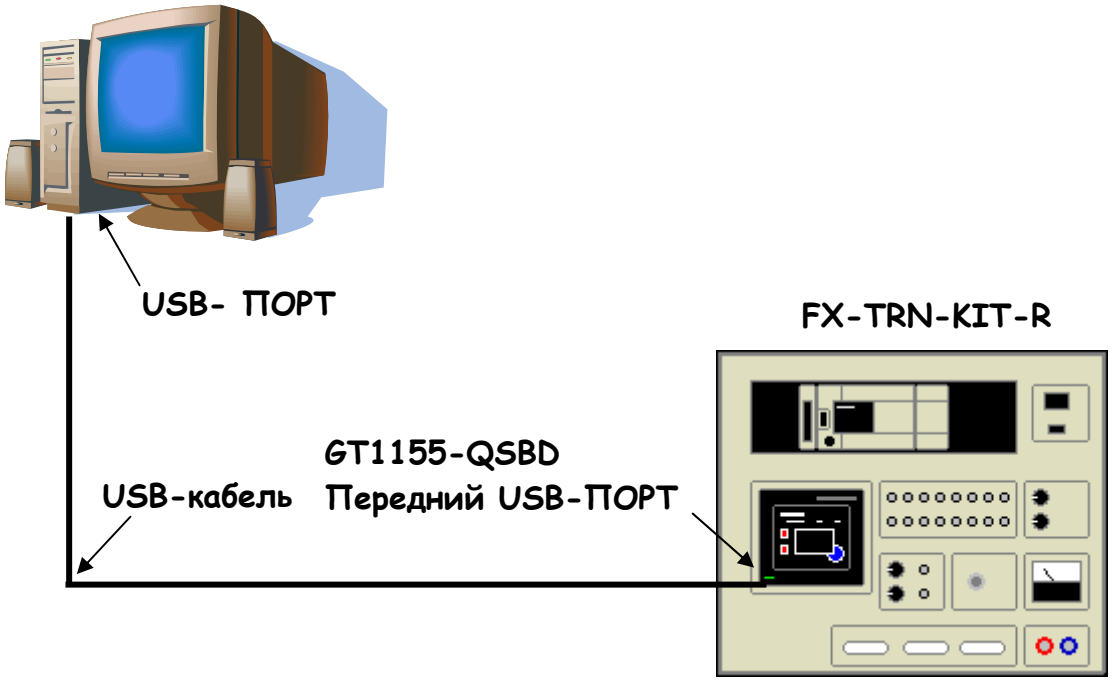
3.5 Подключение аппаратных средств

Круглый 8-выводной порт на модуле ЦП ПЛК используется для связи RS-422. Большинство персональных компьютеров имеют только порт связи RS-232 (если вообще имеют таковой). Поэтому для подключения ПК к ПЛК рекомендуются модули F2-232CAB-1, FX-232AWC-H и FX-422CAB0. FX-232AWC-H является преобразователем интерфейса RS-422 => RS-232, F2-232CAB-1 подключает преобразователь интерфейса в стандартный порт RS-232 порт ПК на удалении до 3 м, и FX-422CAB0 подключает преобразователь интерфейса к ПЛК серии FX на удалении до 1,5 м.



Если в ПК отсутствует последовательный порт RS232, имеются несколько опций.

- К ПЛК серии FX3U можно подключить **FX3U-USB-BD**, чтобы ПК со стандартным USB портом типа A можно было непосредственно соединить с ПЛК, используя поставленный или пользовательский USB-кабель (штекер A <=> мини-штекер B).
- Для ПЛК FX1S, FX1N, FX2N, FX2NC и FX3U может использоваться преобразователь интерфейса USB => последовательный порт, например, **FX-USB-AW**. Преобразователь подключается непосредственно в порт RS-422 ПЛК серии FX; он также поставляется с USB-кабелем.
- При использовании интерфейсов HMI Мицубиси, например, серии GOT1000, имеется прозрачный режим, позволяющий подключить USB-кабель со штекерами A <=> мини B к HMI, который уже соединен с ПЛК. Это обеспечивает доступ к проекту HMI, а также ПЛК по одному кабелю. Прозрачный режим GOT1000 поддерживается в GX Developer версии 8.22Y или выше; на нашем курсе он будет стандартным методом соединения.



ГЛАВА 4 – Системы счисления

В ПЛК используется несколько систем счисления, помимо десятичной системы (с основанием 10). Понимание этих систем критически важно для успешного программирования.

Цели главы: Завершив эту главу, слушатели смогут…

- Назвать различные системы счисления.
- Описать, как представляются числа в различных системах.
- Переходить между системами счисления.

Материалы: Руководство по обучению FX-TRN-KIT-R
Учебный стенд FX-TRN-KIT-R (опционально)

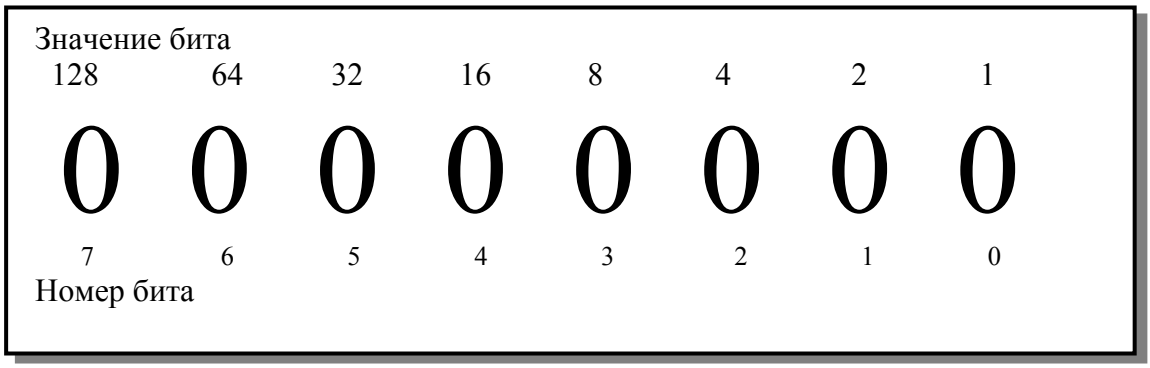
4.1 Двоичные числа

В **двоичной** системе счисления, которая сокращенно обозначается как **BIN**, каждая цифра называется двоичной единицей, или для краткости ***битом***. Двоичная система имеет основание 2. Это означает, что для каждая цифра может принимать только два возможных значения. Каждая цифра может иметь только значение ‘0’ или ‘1’.

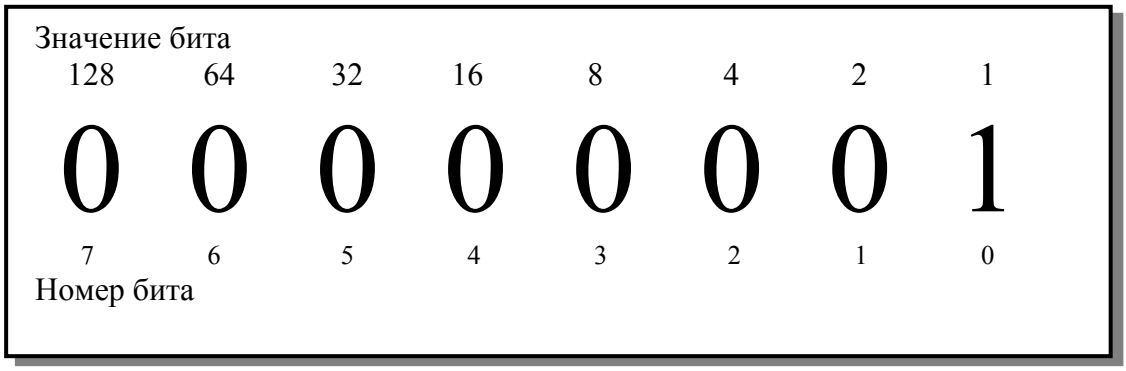
Группа из **4 битов** называется ***ПОЛУБАЙТОМ***
Группа из **8 битов** называется ***БАЙТОМ***
Группа из **16 битов** называется ***СЛОВОМ***

Позиция бита в байте или слове определяет его *значение*. Начиная с правой стороны, бит номер 0 представляет значение ‘1’. По мере того, как бит перемещается влево, значение бита удваивается с каждой позицией. Бит 1 имеет значение 2, бит 2 имеет значение 4, бит 3 имеет значение 8, и т.д.

Следующий пример показывает соответствующие значения в БАЙТЕ:



Чтобы перейти из двоичной системы счисления в десятичную, просто сложите значения битов, которые имеют значение ‘1’, как показано ниже.



Двоичное слово		**Десятичное значение**
0000 0001	……	1
0000 0010	……	2
0000 0100	……	4
0000 1000	……	8
0000 0011	……	3
0000 0101	……	5
0000 0110	……	6

Используя 4 бита, можно представить значения от 0 до 15 …

0000 0000	……	0	
0000 1111	……	15	(8+4+2+1=15)

4.2 Шестнадцатеричные числа

Шестнадцатеричная система, которая сокращенно обозначается как **HEX**, – это система счисления с основанием 16, и каждая цифра в ней имеет 16 возможных значений. Каждая цифра представляет число от 0 до 15. Значения больше 9 представляются буквенными символами.

Цифры от 0 до 9 имеют тот же вид, что и в десятичной системе, а затем используются символы от A до F .

ДЕСЯТИЧНОЕ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16-РИЧНОЕ	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Шестнадцатеричная система также считается быстрым методом записи двоичных чисел.
Каждая шестнадцатеричная цифра представляет 4 бита (или полубайт) данных.

ДВОИЧНОЕ	0000	0010	0011	0100	1000	1001	1010	1011	1111
16-РИЧНОЕ	0	2	3	4	8	9	A	B	F

4.3 Восьмеричные числа

Восьмеричная система, которая сокращенно обозначается как **ОСТ**, это система счисления с основанием 8, и каждая цифра в ней имеет 8 возможных значений. Каждая цифра представляет число от 0 до 7.

В десятичной системе, когда счет проходит 9, 19, и т.д., счет перезапускается с 0 и следующий разряд увеличивается на 1 (т.е. после 9 идет 10, после 19 идет 20).

Таким же образом, когда в восьмеричной системе счет проходит 7, счет перезапускается с 0 и увеличивается следующий разряд. Таким образом после 7 идет 10, после 17 идет 20.

ДЕСЯТИЧНОЕ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8-РИЧНОЕ	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20	21	22

Восьмеричное число также является **сокращенным** методом записи **двоичного**. Каждая восьмеричная цифра представляет 3 бита данных.

ДВОИЧНОЕ	000	010	011	100	001	101	110	111
8-РИЧНОЕ	0	2	3	4	1	5	6	7

Сравнив схемы для шестнадцатеричной и восьмеричной систем, несложно переходить между ними.

Преобразуем шестнадцатеричное число 349AFh в восьмеричный формат

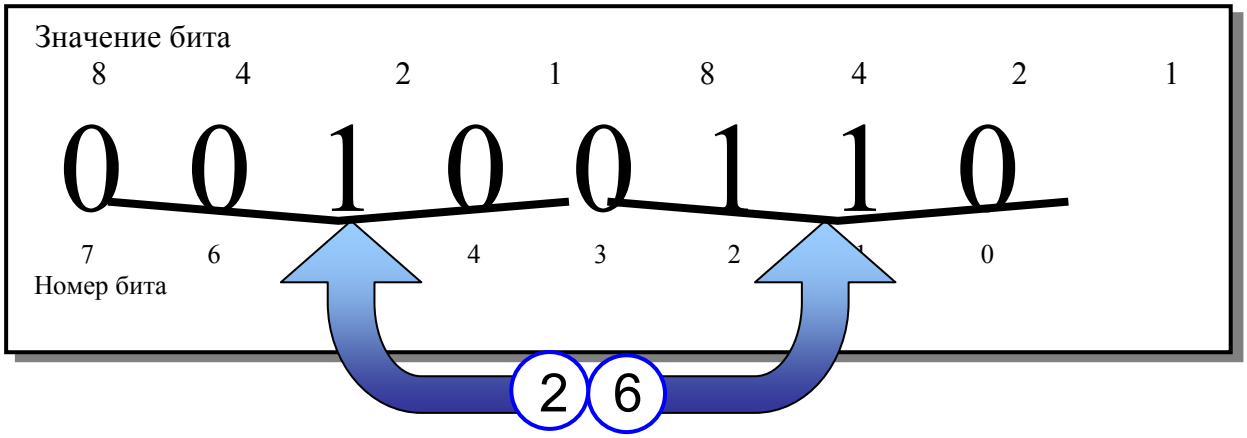
ШЕСТНАДЦАТЕРИЧНОЕ	3	4	9	A	F		
ДВОИЧНОЕ	0011	0100	1001	1010	1111		
(Перегруппировано по 3)	00	110	100	100	110	101	111
ВОСЬМЕРИЧНОЕ	0	6	4	4	6	5	7

Таким образом, шестнадцатеричное число 0x349AF равно 0644657 в восьмеричной системе

4.4 Двоично-кодированное десятичное число

Двоично-кодированное десятичное число, или **ДВОИЧНО-ДЕСЯТИЧНЫЙ КОД (BCD)**, использует те же цифры, что и десятичное число (от 0 до 9), но в том же формате, как в двоичном числе. При преобразовании из ДВОИЧНО-ДЕСЯТИЧНЫЙ КОДА в двоичный, каждая десятичная цифра разделяется на 4 бита (или полубайт).

Преобразуем десятичное число 26 в двоично-кодированный формат



Формат BCD был разработан для использования с десятичными устройствами ввода-вывода, типа дисковых переключателей и семисегментных индикаторов. Десятичные устройства ввода-вывода считают только от 0 до 9, и для чего требуется 4 бита.

ДЕСЯТИЧНОЕ	2	9	12	30
BCD	0000 0010	0000 1001	0001 0010	0011 0000

Различие между двоичным и двоично-десятичным кодом очевидно при преобразовании из десятичной системы.

Преобразуем десятичное число 12 в бинарный формат: биты 3 (значение 8) и 2 (значение 4) имеют значение ‘1’.
Преобразуем двоично-десятичное число 12 в бинарный формат: биты 4 (значение 8) и 1 (значение 4) имеют значение ‘1’.

ПЛК серии FX включают специализированные команды для преобразования между форматами BCD и BIN.

Команда **BCD** осуществляет преобразование из **BIN** в **BCD**.
Команда **BIN** осуществляет преобразование из **BCD** в **BIN**.

4.5 Упражнение Преобразование систем счисления

В этом упражнении преобразуйте следующие числа в заданную систему счисления.

❶ Преобразуйте десятичное число 2 в HEX= BIN= OCT=

❷ Преобразуйте десятичное число 10 в HEX= BIN= OCT=

❸ Преобразуйте десятичное число 16 в HEX= BIN= OCT=

❹ Преобразуйте десятичное число 28 в HEX= BIN= OCT=

❺ Преобразуйте десятичное число 6 в BCD=

❻ Преобразуйте десятичное число 16 в BCD=

❼ Преобразуйте десятичное число 35 в BCD=

Ответы можно проверить, используя интерфейс GOT учебного стенда FX-TRN-KIT-R. За дополнительной информацией об учебном стенде FX-TRN-KIT-R обращайтесь к главе 8 данного Руководства.

ГЛАВА 5 – Цифровые данные в ПЛК

В большинстве приложений ПЛК потребуется обрабатывать данные, будь то манипулирование значениями счетчиков и таймеров, считывание данных из специальных функциональных модулей и обработка информации, или выполнение математических вычислений высокого уровня. Важно, чтобы программист понимал, как ПЛК распознает и обрабатывает различные типы данных, с которыми он может столкнуться.

Цели главы: Завершив эту главу, слушатели смогут…

- Описать, как ПЛК обрабатывает целочисленные и нецелые числа.

Материалы: Руководство по обучению FX-TRN-KIT-R

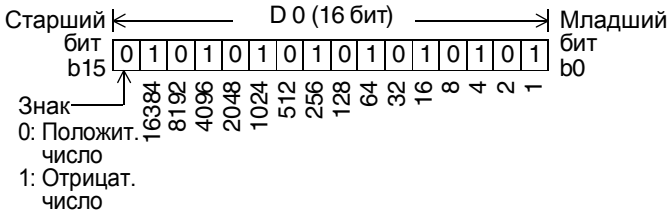
5.1 Обработка целочисленных данных

Очень важно отметить, что ПЛК обрабатывает только целочисленные данные, и по умолчанию числа распознаются только как целые; 1, 10, -2, и т.д. ПЛК не распознает нецелые числа, типа 3,14159. Попытка ввести подобное число приведет к ошибке. Если выполняется математическая операция, например, 5 делится на 3 (ответ: 1.667), ПЛК опустит дробную часть и выдаст ответ 1. Остаток, в этом примере 2, хранится в регистре данных, следующем за регистром назначения математической команды.

16-битовые числа

Целые числа в ПЛК являются 16-битовыми, если иное не объявлено при программировании. Вспомним главу, в которой рассматривались двоичные числа. Это означает, что область числовых значений для целых чисел составляет: 0000 0000 0000 0000 – 1111 1111 1111 1111. После преобразования в десятичный формат это означает, что целочисленный диапазон для ПЛК составляет 0 – 65 535.

Фактически целочисленный диапазон лежит в пределах от -32 768 до + 32 767, потому что крайний левый бит (бит 15) используется ПЛК как знаковый разряд. Этот бит также известен как как старший значащий бит, или MSB. Если он равен 1, то число является отрицательным, если 0 – положительным. Таким образом, реальное максимальное положительное число составляет 0111 1111 1111 1111 что равно 32 767. Если программа увеличивает целочисленное значение и переходит через 32 767, ПЛК считает, что значение сбросилось до –32 768. Если программа уменьшает целочисленное значение и переходит через -32 767, ПЛК считает, что значение сбросилось до 32 768. Таким образом для ПЛК число 1000 0000 0000 0000 равно -32 768. Почему?



Для представления отрицательных чисел ПЛК использует формат, известный как дополнительный код. Дополнительный код несложно вычислить:

❶ Замените все "1" на "0" и все "0" на "1". Это новое число известно как дополнение.

❷ Добавьте к числу 1

0111 1111 1111 1111	равно 32,767
1000 0000 0000 0000	является дополнением (знаковый разряд не включен в дополнение, но должен быть равным 1, чтобы это число было отрицательным)
+ 1	добавляется 1
1000 0000 0000 0001	равно –32 767.

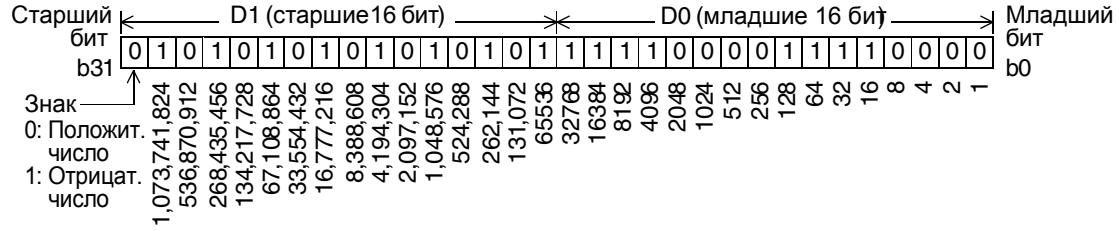
1000 0000 0000 0000 на 1 меньше чем –32 767, поэтому равно –32 768.

Команда **NEG** может использоваться для вычисления дополнительного кода 16-битовых или 32-битовых данных и изменения знака чисел.

32-битовые числа

Как отмечалось выше, по умолчанию целые числа являются 16-битовыми. Некоторые команды позволяют использовать 32-битовые числа. При этом ПЛК распознает два 16-битовых регистра как один большой регистр. Бит 15 больше не считается старшим значащим битом. Теперь ПЛК рассматривает бит 31 как старший значащий бит. Это позволяет ПЛК показывать целочисленные значения
в диапазоне от -2,147,483,648 до 2,147,483,647

Используя 32-битовые команды, важно отметить что число занимает как регистр источника/назначения, так и следующий регистр. Учтите это, когда будете писать программы релейных диаграмм – перезапись второго регистра может иметь непредсказуемые последствия для данных.



5.2 Обработка нецелочисленных данных

Как упоминалось выше, по умолчанию при обработке дробных значений дробная часть отбрасывается. Это ограничение можно обойти, используя числа с плавающей запятой, описанные в разделе 5.1 Руководства по программированию FX3U.

Имеются два формата для показа нецелых чисел: экспоненциальный формат и плавающая запятая.

Экспоненциальный формат

В экспоненциальном формате используются два регистра для хранения мантиссы и экспоненты. Мантисса включает первые четыре значащих разряда числа, а экспонента показывает позицию десятичной точки. Этот формат не может использоваться в вычислениях, но полезен для показа данных.

Пример: 1,238,900 можно показать как 1238 x 10^3. 1238 является мантиссой, а 10^3, который указывает, что десятичная точка находится в 3 позиции вправо, является экспонентой. Учтите, что 9 отбрасывается и не округляется.

Число между 0 и 1000 или 0 и –1000 представляется отрицательной экспонентой. Экспонента показывает, на сколько разрядов слева от младшего разряда мантиссы находится позиция десятичной точки.

Пример: 0.00123 можно показать как 123 x 10^{-5}.

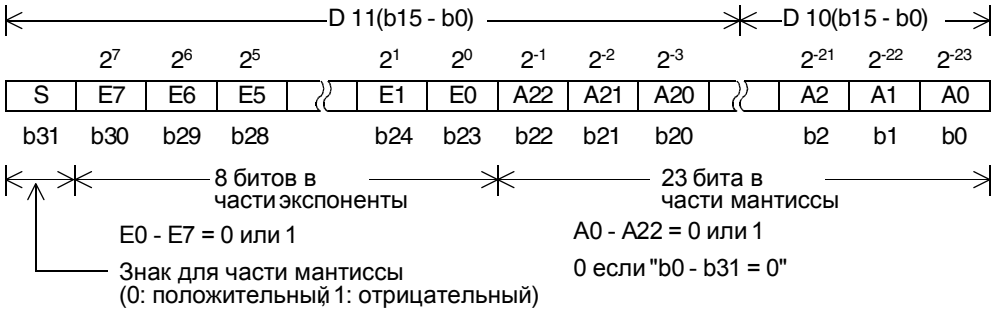
Этот формат позволяет показывать числа, выходящие за пределы стандартного 32-битового диапазона. Диапазон чисел составляет теперь от 9999 x 10^{35} до –9999 x 10^{35}. Компромисс заключается в потере точности – используются только 4 значащих разряда.

Метод хранения числа в экспоненциальном формате: мантисса хранится в 16-битовом регистре, а экспонента – в следующем 16-битовом регистре. В рассмотренном выше примере, если 1 238 900 было необходимо сохранить в D0 и .00123 необходимо сохранить в D2, регистры данных будут иметь следующий вид:

D0	1238
D1	3
D2	123
D3	-5

Плавающая запятая

Как и в экспоненциальном формате, здесь используются два последовательных 16-битовых регистра. Мантисса занимает все 16 битов первого 16-битового регистра и первые 7 битов второго 16-битового регистра. Экспонента занимает последние 9 битов второго 16-битового регистра, а бит 31 служит знаковым разрядом.



Представление числа с плавающей запятой в виде мантиссы и экспоненты соответствует специальному формату, рекомендованному I.E.E.E. (Институтом инженеров по электротехнике и радиоэлектронике). Дополнительная информация содержится в разделе 5.1.3 Руководства по программированию FX3U.

Основное преимущество этого формата – его более высокая точность по сравнению с экспоненциальным форматом. Например, число π (3.1415926…) представляется как 3.141592 (7 значащих цифр) в формате с плавающей запятой, и как
3142×10^{-3} в экспоненциальном формате.

ГЛАВА 6 – Системные операнды

Чтобы написать программу для ПЛК, необходимо знать операнды, которые используются в командах. В этом разделе представлен обзор; более детальная информация будет приведена в следующих главах.

Цели главы: Завершив эту главу, слушатели смогут…

- Назвать и описать операнды, используемые в релейной диаграмме для ПЛК серии FX.

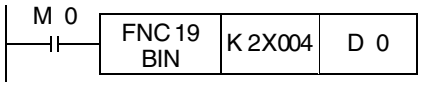
Материалы: Руководство по обучению FX-TRN-KIT-R

Общий вопрос, возникающий при обсуждении системных операндов – это число используемых операндов каждого типа. Оно изменяется в зависимости от модели ПЛК серии FX. Смотри руководство пользователя соответствующего ПЛК серии FX.

X – Физические входы

X – так обозначаются битовые операнды, которые используются для назначения физических входов.

Все битовые операнды в ПЛК, включая X, Y, M, и C, могут группироваться с шагом 4 (т.е. 4, 8, 12, 16 … 32) для использования с командами релейных диаграмм. Чтобы сгруппировать биты вместе, задайте число шагов, а также значение K (десятичной константы) как префикс к битовому операнду, например K4X000 указывает биты X000-X017.



Двухразрядные данные в двоично-десятичном коде (BCD), представленные в X004 - X013, преобразуются в двоичный код, а затем передаются в D0.

Y – Физические выходы

Y – так обозначаются битовые операнды, которые используются для назначения физических выходов.

M – Маркеры

M – так обозначаются маркеры – внутренние битовые операнды, которые могут использоваться для любой необходимой функции. Когда флаг выхода M включается или устанавливается, соответствующий M-операнд (контакт) активизируется или устанавливается.

В пакете GX Developer есть возможность конфигурировать фиксированные маркеры, буферизованные батареей. Буферизованный маркер сохраняет состояние своего контакта (неактивное 0 или активное 1) при переключении ПЛК в режим STOP или отключении электропитания.

Имеется группа маркеров со специальными функциями. Они занимают область адресов **M8000-M8511**. Их значения полностью описаны в главе 36 Руководства по программированию 36 FX3U**.**

S – Маркеры состояния

S – так обозначаются маркеры – внутренние битовые операнды, которые используются при программировании на языке STL, чтобы указать на активность состояний или блоков логического кода релейной диаграммы. Если программирование на языке STL не используется, эти биты могут использоваться, как M-биты.

В пакете GX Developer есть возможность конфигурировать фиксированные S-маркеры, буферизованные батареей. Буферизованный маркер сохраняет состояние своего контакта (неактивное 0 или активное 1) при переключении ПЛК в режим STOP или отключении электропитания.

Использование программирования на языке STL в сочетании с командой IST (Начальное состояние) заставляет некоторые маркеры состояния выполнять специальные операции. Например, S0 является начальным состоянием ручного режима, и S2 – начальным состоянием автоматического режима.

S-маркеры могут также использоваться как "**Сигнализаторы**". В методах программирования, описанных в главе 4.4 Руководства по программированию FX3U, маркеры **S900-S999** могут использоваться как пользовательские индикаторы ошибок.

T – Таймеры

T – так обозначаются операнды внутренних таймеров. По умолчанию таймеры используют шаг времени 100 мс, 10 мс или 1 мс в зависимости от адреса операнда. Большинство таймеров, в зависимости от адреса, **являются нефиксируемыми**, т.е. не сохраняют достигнутое фактическое значение времени после отключения управляющей логической связи. В ПЛК FX2N, FX2NC и FX3U, таймеры с адресами T246 и выше **являются фиксируемыми**. Это означает, что таймер будет сохранять достигнутое фактическое значение времени, пока оно не будет сброшено. Пока вход (катушка) таймера включен, таймер считает заданные временные шаги, увеличивая свое значение. Когда значение счета достигает заданного значения, устанавливается выход (контакт) таймера.

Все таймеры являются 16-битовыми, т.е. максимальное значение таймера и уставка времени равны +32767.Допустимые значения уставки времени определяются с помощью десятичной константы K и значения регистра D, которые умножаются на временной шаг таймера. Например, таймер с шагом 100 мс имеет максимальную уставку времени 3276.7 секунд.

В GX Developer есть возможность назначать область адресов таймеров, буферизованных батареей. Буферизованный таймер будет сохранять значения достигнутого времени при переключении ПЛК в режим STOP или отключении электропитания – при условии, что остается активной логическая связь, управляющая его входом. Иначе значение достигнутого времени и выход таймера сбросятся.

Таймеры будут подробно рассмотрены в главе 12 этого Руководства по обучению.

C – Счетчики

C – так обозначаются операнды внутренних счетчиков. При каждом поступлении сигнала "1" на вход (катушку)счетчика его значение увеличивается или уменьшается на единицу (в зависимости от адреса счетчика и настройки прямого/обратного счета). Когда значение счета достигает задан-

ного значения, устанавливается выход (контакт) счетчика. По умолчанию, все счетчики сохраняют значение счета до сброса.

Счетчики могут быть 16-битовыми или 32-битовыми, т.е. максимальное значение счетчика и уставки счета находится в диапазоне от -32 768 до +32 767 (для 16-битовых) или от -2 147 483 648 до 2 147 483 647 (для 32-битовых). Допустимые задаваемые значения счета определяются десятичной константой K и значениями регистра D.

Имеются три типа счетчиков: 16-битовые счетчики прямого счета, 32-битовые реверсивные счетчики, и 32-битовые быстродействующие реверсивные счетчики. В категории быстродействующих счетчиков имеются 1-фазные и 2-фазные счетчики.

В GX Developer есть возможность назначить область адресов счетчиков, буферизованных батареей. Буферизованные счетчики будут сохранять значение счета при переключении ПЛК в режим STOP или отключении электропитания. Иначе значение счета и выход (контакт) счетчика сбросятся.

Счетчики будут подробно рассмотрены в главе 12 этого Руководства по обучению.

D – Регистры данных и регистры файлов

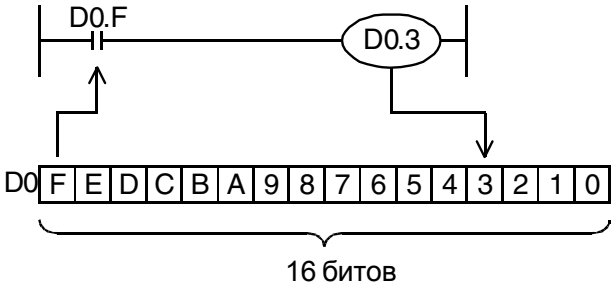
D – адреса D являются регистрами данных (словными операндами), которые могут использоваться для любой необходимой функции. Все регистры D являются 16-битовыми, т.е. принадлежат диапазону значений от -32768 до 32767. В релейной диаграмме можно также выполнять 32-битовые операции. В таких случаях два последовательных регистра D используются вместе и распознаются ПЛК как один длинный регистр, способный хранить числовые значения от -2 147 483 648 до 2 147 483 647.

В GX Developer есть возможность назначить область адресов регистров D, буферизованных батареей. Это означает, что регистры сохранят свои значения при переключении ПЛК в режим STOP или отключении электропитания.

Регистры файлов являются регистрами D, которые сохраняются в памяти для хранения программы кассеты памяти, а не в памяти данных ПЛК. Они должны объявляться в разделе Parameters (Параметры) GX Developer блоками по 500; при работе программы возможно только копирование значений из регистров файлов в регистры D. Каждый блок 500 объявленных регистров файлов уменьшает количество шагов программы на 500. Учтите, что для использования регистров файлов необходимо подключить кассету памяти.

Регистры данных в области адресов **D8000-D8511** являются специальными регистрами для диагностики ПЛК и выполнения специальных функций. Все их функции описаны в главе 36 Руководства по программированию FX3U .

ПЛК серии FX3U способны осуществлять непосредственный доступ к битам в регистрах D. Разместив десятичную точку между адресом регистра D и адресом бита, можно идентифицировать состояние отдельного бита в регистре, используя любую команду с битовым операндом. В качестве примера приведем команду D100.0, которая опрашивает бит 0 (младший значащий бит) в регистре D100.



R и ER – расширенные регистры и регистры файлов (только в FX3U)

R – адреса R являются расширенными регистрами – это расширенный набор буферизованных регистров данных, который имеется только в ПЛК серии FX3U. Их можно использовать для любой необходимой функции, а также специальной функции регистрации данных, которая поддерживается только для регистров R и ER. Более подробные сведения о функциях регистрации см. в главе 33 Руководства по программированию FX3U.

ER – адреса ER являются расширенными регистрами файлов, которые соотносятся с R-регистрами таким же образом, как регистры файлов с регистрами D. Для работы с регистрами ER необходимо, чтобы к ПЛК серии FX3U была подключена кассета памяти; они поддерживают специализированные команды регистрации данных в ПЛК серии FX3U. Регистры ER должны инициализироваться в релейной диаграмме блоками по 2048 адресов; возможно только копировать значения из регистров ER в регистры R. Более подробные сведения см. в главе 33 Руководства по программированию FX3U.

V и Z – Индексные регистры

Регистры V и регистры Z являются индексными; оба типа можно использовать с 16-битовыми командами, в то время как только Z-тип можно использовать с 32-битовыми командами (занимающими V-регистр с тем же адресом). Значения, сохраненные в индексном регистре, используются как смещения для указанного операнда. Чтобы получить смещение, делается ссылка на смещение с адресом индексного регистра в качестве суффикса к операнду. Например, если V0 имеет значение 2, то D10V0 означает D10 + 2, т.е. D12. Если Z2 имеет значение 8, то Y001Z2 означает Y011. Помните, что адреса регистров X и Y являются восьмеричными, так что значение смещения преобразуется в восьмеричный формат и прибавляется к базовому адресу. Поскольку 8 в десятичном формате соответствует 10 в восьмеричном, адрес увеличился на 10.

Если регистры V или Z используются без операнда или с постоянным значением префикса, они функционируют так же, как регистры D.

Индексные регистры полезны для написания короткого кода на языке релейных диаграмм, который может использоваться для доступа к адресам многих различных операндов, не перегружая время цикла.

P – Указатели

P – адреса P обозначают указатели, которые используются с командами условного перехода (CJ) и вызова подпрограмм (CALL), позволяя изменять процесс выполнения программы. Команды CJ и CALL заставляют цикл программы выполнять различные блоки релейной диаграммы, либо переходя в другое место в той же программе, либо вызывая подпрограммы, которые должны выполняться в течение стандартного цикла.

Мы рассмотрим указатели в разделе 13.22 данного Руководства.

I – Прерывания

I – адреса I обозначают прерывания, которые используются для выполнения процессов при выполнении стандартной последовательности цикла. При запуске программы немедленно выполняется код релейных диаграмм, назначенный прерыванию, независимо от хода обработки других блоков кода релейных диаграмм. Прерывания могут вызываться физическими входами, счетчиками, или в указанные временные интервалы.

K, H, и E – Числовые константы

K, H и **E** используются для указания на числовые константы. ПЛК не распознает собственно числовые значения, вместо этого необходимо, чтобы значения имели префиксы, объявляющие, к какому типу числовых значений они относятся. Префикс "K" объявляет, что константа будет десятичной. Префикс "H" объявляет шестнадцатеричную константу. Префикс "E" объявляет вещественную константу, типа 3,14159. Это означает, что потребуется использовать функции плавающей запятой.

U□\G□ – Доступ к буферной памяти (только в FX3U)

U□\G□ – эта система адресации позволяет командам релейных диаграмм непосредственно опрашивать ячейки буферной памяти в подключенном специальном функциональном модуле (SFM). Число после "U" указывает адрес SFM (0-7), а число после "G" – адрес ячейки буферной памяти в SFM. Эта возможность имеется только в FX3U.

Номер модуля (U) ..... 0 - 7
Адрес ячейки памяти (\G) ..... 0 - 32766

ГЛАВА 7 – Адресация

Для управления и контроля ввода-вывода и устройств в программе ПЛК необходимо знать адрес управляемого устройства. То же относится и к внутренней памяти специальных функциональных модулей. В этой главе объясняется, как определяются адреса специальных функциональных модулей, специальных адаптеров и точек ввода-вывода в системе.

Цели главы: Завершив эту главу, слушатели смогут…

- Правильно адресовать точку дискретного входа или выхода.
- Определить адрес специального функционального модуля и специального адаптера.
- Описать ограничения ввода-вывода системы FX3U.

Материалы: Руководство по обучению FX-TRN-KIT-R

7.1 Адресация правой шины

1) Адресация входов и выходов выполняется в восьмеричной системе (X000-X007, X010-X017, и т.д.).

2) Адресация как входов, так и выходов начинается с 0 (X000 и Y000).

3) Адресация является последовательной.

4) Специальные функциональные модули (SFM) имеют адреса 0 – 7. Первый SFM на правой стороне ПЛК – это SFM 0; далее идет SFM 1, и т.д.

5) К главному блоку можно подключить максимум 8 SFM.

6) SFM не влияют на адресацию модулей ввода-вывода, и наоборот.

7) ПЛК серии FX3U не могут иметь более 128 встроенных входов и 128 выходов. Можно расширить число адресов ввода-вывода до 384, используя сетевой ввод-вывод (по CC-Link и/или AS-I).

8) Каждый SFM использует 8 адресов ввода-вывода, которые вычитаются из максимального количества встроенных адресов ввода-вывода. Таким образом, FX3U с одним SFM имеет максимум 248 встроенных точек ввода-вывода. Порознь максимально возможное количество входов все еще равно 128, и максимальное количество выходов все еще равно 128, пока не превышено ограничение на 248 суммарных адресов ввода-вывода.

7.2 Адресация левой шины FX3U55

ПЛК серии FX3U имеют дополнительную расширительную шину на левой стороне ПЛК. Эта шина используется с платами расширения (BD) и специальными адаптерами (ADP), которые могут применяться, чтобы повысить функциональность ПЛК.

Имеются четыре различных типа специальных адаптеров.

- Специальные адаптеры аналогового ввода-вывода (максимум 4 на ПЛК)
- Специальные адаптеры с высокоскоростным импульсным входом (максимум 2 на ПЛК)
- Специальные адаптеры с высокоскоростным импульсным выходом (максимум 2 на ПЛК)
- Специальные адаптеры последовательной передачи данных (максимум 2 на ПЛК или 1, когда одновременно используется BD последовательной передачи данных)

Аналоговые специальные адаптеры

Данные и параметры аналогового входа-вывода хранятся непосредственно в устройстве памяти ПЛК. ADP адресуются от ПЛК начиная с левой стороны ПЛК.

- 1ый ADP использует M8260-M8269 и D8260-D8269
- 2ой ADP использует M8270-M8279 и D8270-D8279
- 3ий ADP использует M8280-M8289 и D8280-D8289
- 4ый ADP использует M8290-M8299 и D8290-D8299

ADP с высокоскоростным импульсным входом

ADP с высокоскоростным импульсным входом адресуются от ПЛК начиная с левой стороны ПЛК.

- 1ый ADP использует X000, X001, X002, X006 (C235-C237, C244)
- 2ой ADP использует X003, X004, X005, X007 (C238-C239, C245)

Учтите, что это те же адреса устройств, которые используются первыми восемью физическими входами на ПЛК. Должны использоваться только клеммы высокоскоростного входа ADP или клеммы встроенного ввода-вывода. Не подводите кабели к обеим клеммам. Прочие ограничения см. в Руководстве по программированию FX3U.

ADP с высокоскоростным импульсным выходом

ADP с высокоскоростным импульсным выходом адресуются от ПЛК начиная с левой стороны ПЛК.

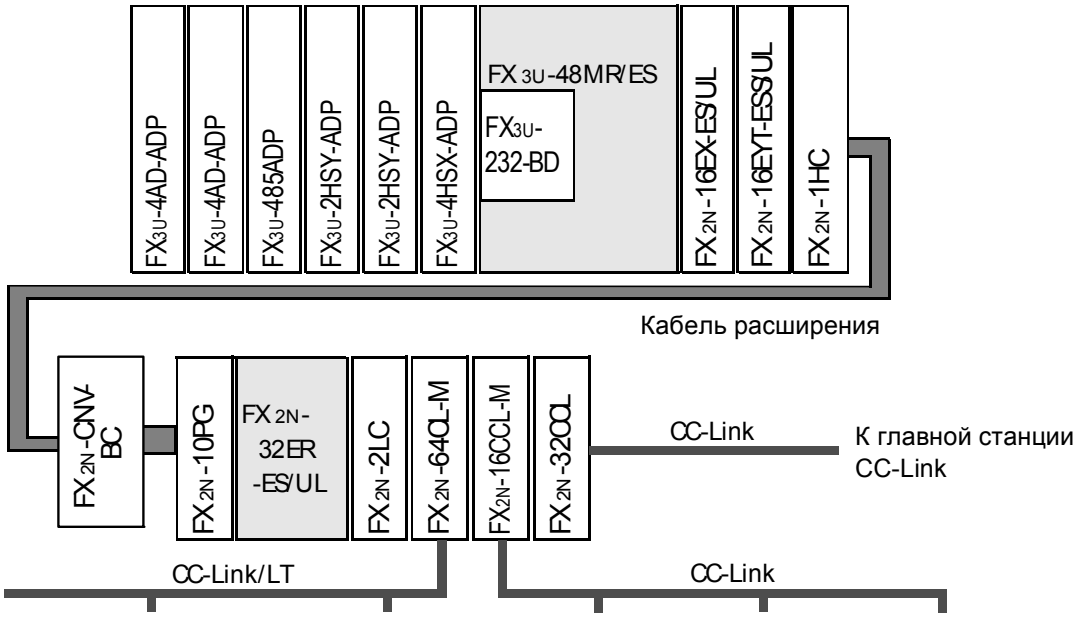
- 1ый ADP использует Y000, Y001, Y004, Y005
- 2ой ADP использует Y002, Y003, Y006, Y007

Учтите, что это те же адреса устройств, которые используются первыми восемью физическими выходами на ПЛК. Должны использоваться только клеммы высокоскоростного выхода ADP или клеммы встроенного ввода-вывода . Не подводите кабели к обеим клеммам. Прочие ограничения см. в Руководстве по программированию FX3U.

ADP последовательной связи

Порты последовательной связи ADP последовательной передачи данных адресуются как Канал 1 (Ch1) и Канал 2 (Ch2), от ПЛК начиная с левой стороны ПЛК. При использовании BD последовательной передачи данных BD адресуется как Ch1, что оставляет только Ch2 для одного ADP последовательной передачи данных. Информация по программированию для последовательной передачи данных приведена в Руководстве по передаче данных в серии FX.

7.3 Пример адресации



FX3U-48MR/ES X000-X007, X010-X017, X020-X027
Y000-Y007, Y010-Y017, Y020-Y027

Левая шина: От ПЛК начиная с левой стороны ПЛК

Плата FX3U-232-BD не адресуется

FX3U-4HSX-ADP	X000, X001, X002, X006
FX3U-2HSY-ADP	Y000, Y001, Y004, Y005
FX3U-2HSY-ADP	Y002, Y003, Y006, Y007

FX3U-485-ADP не адресуется

FX3U-4AD-ADP	M8260-M8269, D8260-D8269
FX3U-4AD-ADP	M8270-M8279, D8270-D8279

Правая шина: От ПЛК начиная с правой стороны ПЛК

FX2N-16EX-ES/UL	X030-0X37, X040-X047
FX2N-16EYT-ESS/UL	Y030-Y037, Y040-Y047
FX2N-1HC	SFM 0
FX2N-10PG	SFM 1
FX2N-32ER-ES/UL	X050-X570, X060-0X67 Y050-Y057, Y060-Y067
FX2N-2LC	SFM 2
FX2N-64CL-M	SFM 3
FX2N-16CCL-M	SFM 4
FX2N-32CCL	SFM 5

7.4 Упражнение Адресация ПЛК

1) Система ПЛК включает FX3U-64MR (32/32 адреса ввода-вывода), один FX3U-USB-BD, один FX3U-4AD-ADP, один 8-точечный входной модуль, два 16-точечных выходных модуля, два SFM, один 16-точечный входной модуль и еще один SFM. Нарисуйте систему и определите адресацию.

2) ДОПУСТИМЫ ЛИ СЛЕДУЮЩИЕ СИСТЕМЫ? ПОЧЕМУ?

A. Главный блок с 64 точками ввода-вывода (32/32 адреса ввода-вывода), четыре 8-точечных входных модуля, шесть 8-точечных выходных модулей и девять SFM.

B. Главный блок с 128 точками ввода-вывода (64/64 адреса ввода-вывода), два SFM, два модуля расширения на 48 точек ввода-вывода (24/24 ввода-вывода в каждом), один 16-точечный входной модуль.

C. Главный блок с 128 точками ввода-вывода (64/64 адреса ввода-вывода), три SFM, два модуля расширения на 48 точек ввода-вывода (24/24 адреса ввода-вывода в каждом), один 16-точечный входной модуль.

D. Главный блок с 80 точками ввода-вывода (40/40 адресов ввода-вывода), два модуля расширения на 48 точек ввода-вывода (24/24 адреса ввода-вывода в каждом), четыре 16-точечных выходных модуля, один 8-точечный входной модуль.

E. Главный блок FX3U с 64 точками ввода-вывода (32/32 адреса ввода-вывода), три FX3U-4AD-ADP, один FX3U-4DA-ADP, один FX3U-4AD-PT-ADP, два FX3U-4HSX-ADP, два FX3U-485-ADP, один FX3U-USB-BD.

ГЛАВА 8 – Конструкция учебного стенда

Теперь, получив все необходимые знания, мы займемся изучением и настройкой аппаратных средств, если это уже не было сделано раньше. В данной главе объясняется состав учебного стенда и приводится краткий обзор его характеристик.

Цели главы: Завершив эту главу, слушатели смогут…

- Назвать различные части учебного стенда.
- Узнать, как части работают вместе.

Материалы: Руководство по обучению FX-TRN-KIT-R
Учебный стенд FX-TRN-KIT-R

8.1 Адресация

Этот демонстрационный стенд включает **FX3U-32MT/ESS**, **FX3U-485-BD**, **FX3U-4AD-PT-ADP**, **FX2N-5A**, и **GT1155-QSBD**.

FX3U имеет 16 входов с адресами X000-X007 и X010-X017 и 16 выходов с адресами Y000-Y007 и Y010-Y017.

FX2N-5A – аналоговый модуль, включающий 4 аналоговых входа и 1 аналоговый выход. Это первый специальный функциональный модуль на правой стороне главного блока, поэтому он имеет адрес SFM 0.

FX3U-485-BD занимает место платы расширения BD в главном блоке и использует канал последовательной передачи данных 1 (Ch1).

FX3U-4AD-PT-ADP – первый аналоговый модуль слева от ПЛК. Поэтому в нем используются адреса устройств M8260-M8269 и D8260-D8269.

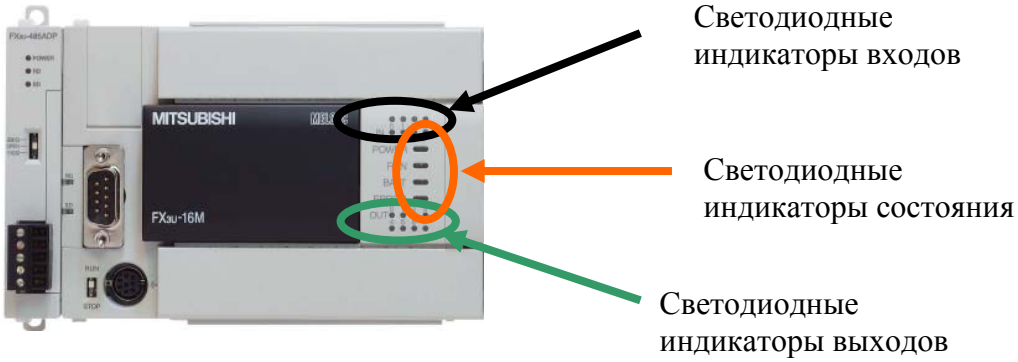
8.2 Индикаторы

ПЛК имеет светодиодные индикаторы входов и выходов, а также светодиодные индикаторы состояния на передней панели главного блока.

Имеется два набора светодиодных индикаторов ввода-вывода, один набор для входов, один набор для выходов. Когда активизируется (включается) X000, загорается светодиод 0 во входной секции. Когда активизируется (включается) Y000, загорается светодиод 0 в выходной секции, и т.д.

Имеются 4 индикатора на правой стороне передней панели главного блока – светодиодные индикаторы состояния.

- 1ый светодиод указывает, что на ПЛК подано электропитание.
- 2ой светодиод включается, когда ПЛК находится в режиме RUN и выключается – в режиме STOP.
- 3ий светодиод включается при низком напряжении батареи.
- 4ый светодиод имеет две цели:
    - мигание указывает на существование ошибки в программе.
    - непрерывное свечение указывает на проблему ЦП, например, вынимание кассеты памяти из включенного ПЛК, который находится в режиме RUN.



8.3 Интерфейс оператора

Упражнения в этом курсе будут включать использование графического терминала оператора (GOT) как средство визуальной поддержки и интерфейс ПЛК. В GOT загружаются экраны для визуализации операций ПЛК, мониторинга значений устройств, а также имитации различных упражнений по программированию, запланированных для этого курса обучения. Только некоторые главы и разделы данного курса затрагивают интерфейс GOT, они будет специально отмечены в этом Руководстве.

Попробуйте перейти к экрану "Глава 8; Конструкция учебного стенда" используя сенсорный экран GOT, чтобы привыкнуть к интерфейсу и структуре GOT.

Учебный стенд FX-TRN-KIT-R также включает аппаратные **переключатели цифровых входов**, **регуляторы аналоговых входов**, **переключатели и регуляторы высокоскоростных входов**, и **температурный датчик типа PT** для пользователей, чтобы соединять их с ПЛК. Канал аналогового вывода 1 на FX2N-5A выводится на **измеритель аналогового выхода**, который будет обсуждаться в разделе 13.15 этого Руководства. Все аппаратные входы и выходы можно переподключить вручную к другим периферийным устройствам и оборудованию, используя **вспомогательные входы и выходы**. Для этого см. блок-схему и монтажную схему, приложенные к учебному стенду.

ГЛАВА 9 – Типы команд ПЛК

Чтобы написать релейную диаграмму для ПЛК, необходимо ознакомиться с инструкциями, входящими в систему команд ПЛК. Этот раздел обзорный; более детальная информация будет приведена в следующих главах.

Цели главы: Завершив эту главу, слушатели смогут…

- Описать три различных типа команд.

Материалы: Руководство по обучению FX-TRN-KIT-R

Более подробную информацию по всем инструкциям можно найти в Руководстве по программированию серии FX для соответствующей модели ПЛК.

Номер руководства	Рассмотренная серия ПЛК
JY992D48301	FX, FX0, FX0S, FX0N, FX2C
JY992D88101	FX1S, FX1N, FX2N, FX2NC
JY997D16601	FX3U

9.1 Базовые команды

К этой категории относятся команды для четырех базовых битовых операндов (X, Y, M, и S), таймеров (T) и счетчиков (C), за исключением высокоскоростных счетчиков. Они включают функции LD, LDI, OUT, SET, RST и PLS. Из подобных команд обычно состоит большая часть программы.

9.2 Инструкции релейных диаграмм

Инструкции релейных диаграмм используются при программировании релейных диаграмм (STL). Этот стиль программирования аналогичен программированию последовательных функциональных схем (SFC), в котором составляется блок-схема системных операций, но в STL блок-схема составляется и фактически строится в коде языка релейных диаграмм. При программировании в коде языка релейных диаграмм используется распространенная команда контакт STL, позволяющая проверять, активно ли состояние. Для индикации состояний используются маркеры S.

Программирование на языке STL не входит в программу этого курса обучения. Однако полное объяснение и примеры можно найти в главе 34.2 Руководства по программированию FX3U.

9.3 Прикладные команды

Эти команды позволяют ПЛК выполнять сложные манипуляции с данными, математические операции и операции связи. Примеры использования прикладных команд – преобразование BIN-BCD , тригонометрические функции с плавающей запятой, сортировка данных и связь с инвертором. Большинство прикладных команд для ПЛК серии FX работают на уровне 16-битовых или 32-битовых слов.

ГЛАВА 10 – Базовые команды

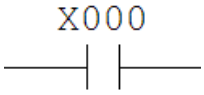
Базовые команды включают инструкции управления битом, таймером и счетчиком; как правило, из них состоит большая часть релейной диаграммы. Они используются для подтверждения состояния входов, управления выходами, сдвига битов и выполняют основную часть управления для вложенных цепей и других конструкций в коде языка релейных диаграмм.

Цели главы: Завершив эту главу, слушатели …

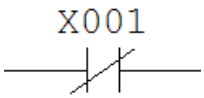
- Смогут назвать наиболее распространенные базовые команды.
- Узнают о формате этих инструкций и их функциях.

Материалы: Руководство по обучению FX-TRN-KIT-R
Руководство по программированию серии FX3U – Базовые и прикладные команды
Учебный стенд FX-TRN-KIT-R

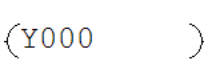
10.1 Символы



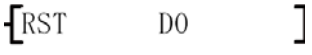
НОРМАЛЬНО ОТКРЫТЫЙ КОНТАКТ – КОМАНДА ЗАГРУЗКИ – этот "контакт" активен, когда соответствующий операнд "катушка" имеет значение "1". Мнемоника команды – **LD**, сокращение от LOAD (ЗАГРУЗИТЬ). Символ занимает 1 шаг памяти для хранения программы.



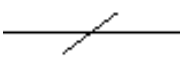
НОРМАЛЬНО ОТКРЫТЫЙ КОНТАКТ – ЗАГРУЗИТЬ ИНВЕРСНО – этот "контакт" активен, когда соответствующий операнд "катушка" имеет значение "0". Мнемоника команды **LDI**, сокращение от LOAD INVERSE. Символ занимает 1 шаг памяти для хранения программы.



КОМАНДА ВЫВОДА Этот символ всегда присутствует в правой стороне релейной диаграммы и представляет выходную "катушку" – команду вывода, или присвоение результата логической операции. Указанный операнд установлен (включен), до тех пор, пока действуют все условия его включения. При включении "катушки" устанавливается "контакт" – битовый флаг выхода с тем же адресом. Мнемоника команды – **OUT**, сокращение от OUTPUT (ВЫВОД). Символ занимает 1 шаг памяти для хранения программы, кроме ситуаций, когда используются с таймером или счетчиком, когда может занять до 5 шагов.



СКОБКИ. – как правило, этот символ используется для прикладных команд. Однако имеются несколько базовых команд, которые также используют скобки. Обычно они находятся в правой стороне релейной диаграммы. С символом скобок связаны многие команды и мнемоника. Для каждой из них может потребоваться несколько шагов в памяти для хранения программы, в зависимости от команды.



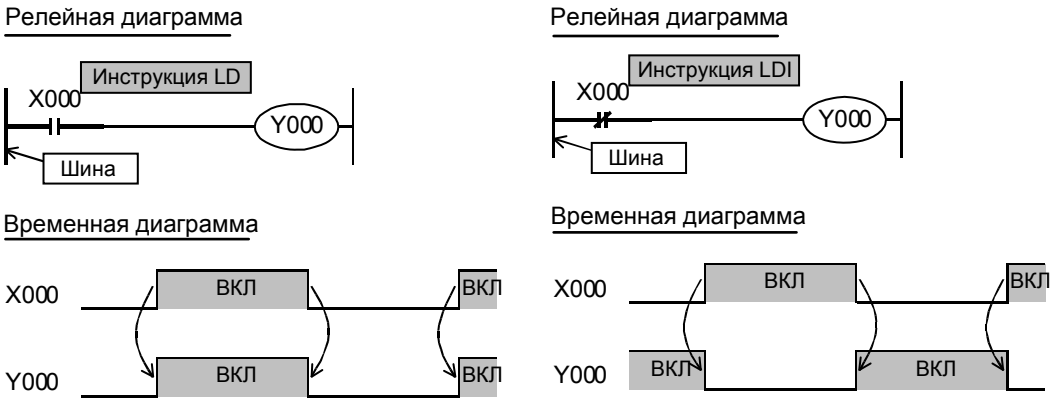
ИНВЕРТИРОВАТЬ. Этот команда обращает результат логической операции, который был действителен перед ее выполнением. Если результат логической операции был "1", то после инверсии он становится равным "0", и наоборот.

Важно усвоить изложенные выше концепции перед тем, как двигаться дальше. Показанные выше символы представляют инструкции в неактивном состоянии. Интуитивно понятно, что активный контакт позволяет протекать электрическому току.

Например, выключатель освещения обычно находится в выключенном (неактивном) состоянии, не позволяя течь току (контакты переключателя разомкнуты), пока кто-нибудь его не включит (установит, активизирует). При этом начинает течь ток (контакты переключателя замкнуты) и освещение ВКЛЮЧАЕТСЯ. Выключатель освещения может рассматриваться как аналог нормально разомкнутого контакта.

Нормально замкнутый контакт является его противоположностью во всех отношениях. Ток течет, пока выключатель ВКЛЮЧЕН. Типичный пример – аварийный выключатель (аварийный останов). Выключатель позволяет течь току, пока оператор не нажмет на него (активизирует, установит) в аварийной обстановке. Выключатель активизируется и останавливает протекание тока.

Пример применения: Нормально разомкнутый и нормально замкнутый контакты



10.2 Основы релейных диаграмм

Структура и порядок "контактов" и "катушек" определяют, как обрабатывается звено (при горизонтальном представлении схемы) кода релейных диаграмм.

Если две или большее количество инструкций помещены последовательно (одна за другой, как показано ниже), то они работают согласно логической операции "И". Для того, чтобы "ток" мог течь через звено, все контакты должны быть активными. В этом примере Y000 включается, только если одновременно X000 активен, X001 неактивен и X002 активен.

Если две или несколько инструкций помещены параллельно (одна под другой, как показано ниже), то они работают согласно логической операции "ИЛИ". Условия "ИЛИ " обеспечивают несколько путей для протекания "тока". В этом примере Y000 включается, если X000 активен, или X001 неактивен, или X002 активен.

Условия И/ИЛИ можно комбинировать, создавая сложные логические звенья.

Замечание Звено должно имеют входную команду (подобную LD или LDI), и выходную команду (подобную OUT), чтобы завершить схему. Если выходная команда всегда должна быть включена, имеется специальный маркер с адресом M8000, перед которым помещается инструкция LD. Этот бит установлен, пока ПЛК находится в режиме RUN. Не разрешается подключать флаг выхода или команду в квадратных скобках непосредственно к левой вертикальной питающей шине релейной схемы.

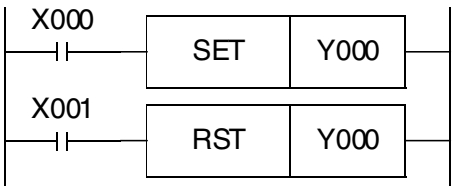
Вы можете визуально смоделировать работу этих базовых команд. См. экран "Раздел 10.2; Стандартные команды" на интерфейсе GOT учебного стенда FX-TRN-KIT-R. Используя тумблеры цифровых входов и сенсорный экран GOT, включайте контакты в моделируемом коде языка релейных диаграмм.

10.3 Распространенные команды

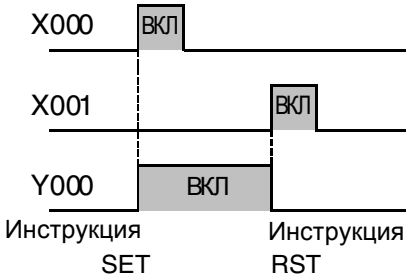
SET – Устанавливает бит
RST – Сбрасывает бит

Команда установки (**SET**) фиксирует указанный битовый операнд в состоянии "1" Команда сброса (**RST**) сбрасывает операнд. Когда команда RST используется со словным операндом, значение словного операнда устанавливается равным 0.

Релейная диаграмма



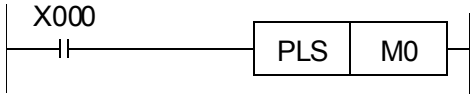
Временная диаграмма



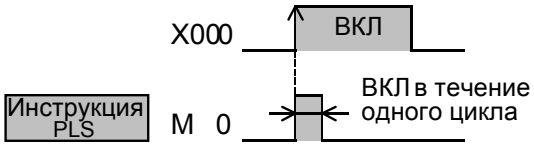
PLS – Установка операнда при возрастающем фронте
PLF – Установка операнда при нисходящем фронте

Команда (**PLS**) устанавливает операнд при возрастающем фронте сигнала на входе, в то время как команда (**PLF**) устанавливает операнд при падающем фронте сигнала на входе. Обе команды устанавливают операнд только на время **одного цикла** релейной диаграммы.

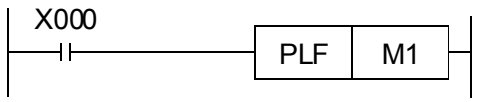
Релейная диаграмма



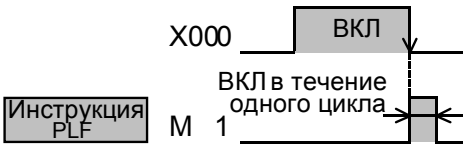
Временная диаграмма



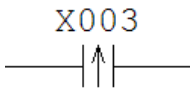
Релейная диаграмма



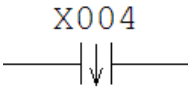
Временная диаграмма



Другой метод обработки фронтов на входе заключается в выполнении логических операций в зависимости от фронта сигнала. Эти сигналы активизируют операнд только на один цикл программы, аналогично PLS и PLF.



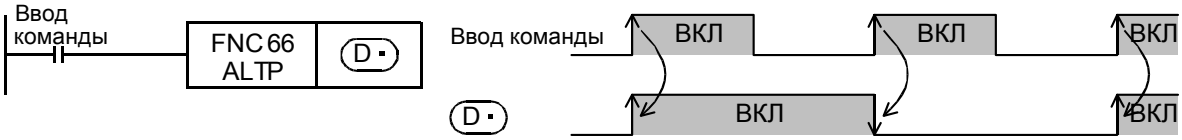
LDP. Команда загрузки при возрастающем фронте операнда (**LDP**) устанавливает выход на один цикл при включении указанного операнда. Вместо использования X003 в качестве входа и затем команды PLS для маркера M, эта команда генерирует импульс на один цикл для последующих инструкций, не занимая дополнительный битовый операнд.



LDF. Команда загрузки при ниспадающем фронте операнда (**LDP**) устанавливает выход на один цикл при выключении указанного операнда. Вместо использования X004 в качестве входа и затем команды PLF для маркера M, эта команда генерирует импульс на один цикл для последующих инструкций, не занимая дополнительный битовый операнд.

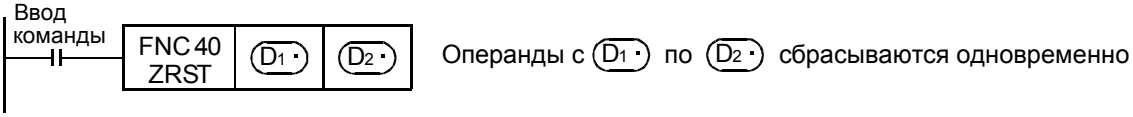
ALT – Триггерная функция

Команда триггерной функции (**ALT**) переключает состояние указанного битового операнда. Если битовый операнд активен, то команда ALT делает его неактивным, и наоборот.



ZRST – Сброс областей операндов

Команда сброса области операндов (**ZRST**) позволяет сбрасывать область адресов операндов, а не отдельные операнды, как это делает RST. В качестве операндов могут использоваться биты, слова, таймеры или счетчики, для которых указываются начальный и конечный адреса. Когда ZRST используется со словными операндами, значения словного операнда устанавливаются на 0. Второй аргумент команды (конечный адрес операнда) должен быть больше, чем первый аргумент (начальный адрес операнда).



Вы можете визуально смоделировать работу некоторых из этих распространенных команд. См. экран "Раздел 10.3; Распространенные команды" на интерфейсе GOT учебного стенда FX-TRN-KIT-R. Используйте сенсорный экран GOT, чтобы активизировать и подавать импульсы на входы в моделированной системе.

10.4 Упражнение Основы релейных диаграмм

1) X001 включает и устанавливает Y003. Каково поведение Y003, когда X001 выключается?

2) Как символ обычно используется, чтобы представить стандартный аварийный останов в релейной диаграмме? (Аварийный останов являются обычно выполняется аппаратно, но часто по различным причинам на него ссылаются в других частях программы.)

4) Перечислите символы распространенных базовых программ (например, ) и опишите их функции.

4) Перечислите распространенные базовые команды и распространенные команды (например, **PLS**) и опишите их функции.

5) Что необходимо для завершения звена схемы?

6) На главном конвейере датчик (вход X002) проверяет наличие определенных упаковок. Обнаружив ее, датчик активизируется и включает толкач (выход Y007). Толкач остается включенным, пока упаковка не будет обнаружена на боковом конвейере датчиком (X003). Когда Y007 выключается, толкач автоматически втягивается. Напишите код релейной диаграммы, реализующей эту схему.

ПРИМЕЧАНИЕ Когда упаковка сдвинута, она покидает область обнаружения датчика (X002) перед тем, как будет обнаружена датчиком бокового конвейера (X003).

Для визуального моделирования приложений, описанных в этом упражнении, перейдите на экран "Раздел 10.4; Упражнение – Основы релейных диаграмм" на интерфейсе GOT учебного стенда FX-TRN-KIT-R. Этот экран предназначается для помощи студентам в визуализации приложений, описанных в упражнении, упрощая написание кода релейных диаграмм. Учтите, что код релейных диаграмм ПЛК фактически не используется при моделировании на интерфейсе GOT.

ГЛАВА 11 – Разработка и редактирование программ

Теперь настало время поработать над предметом, который мы пока что не изучали. В этой главе мы обзорно рассмотрим запуск GX Developer, написание простой программы, загрузим ее на ПЛК и проверим ее работу, а попутно исследуем некоторые инструменты в программе GX Developer.

Цели главы: Завершив эту главу, слушатели смогут…

- Запускать GX Developer.
- Вводить инструкции, чтобы написать небольшую релейную диаграмму.
- Переносить релейную диаграмму между ПЛК и ноутбуком или ПК.
- Редактировать программу в режиме онлайн.
- Проверять программу.
- Изменять значения операндов на ПЛК с помощью программного обеспечения.
- Контролировать значения операндов на ПЛК в регистрах данных.

Материалы: Руководство по обучению FX-TRN-KIT-R
Учебный стенд FX-TRN-KIT-R

11.1 Запуск программы GX Developer

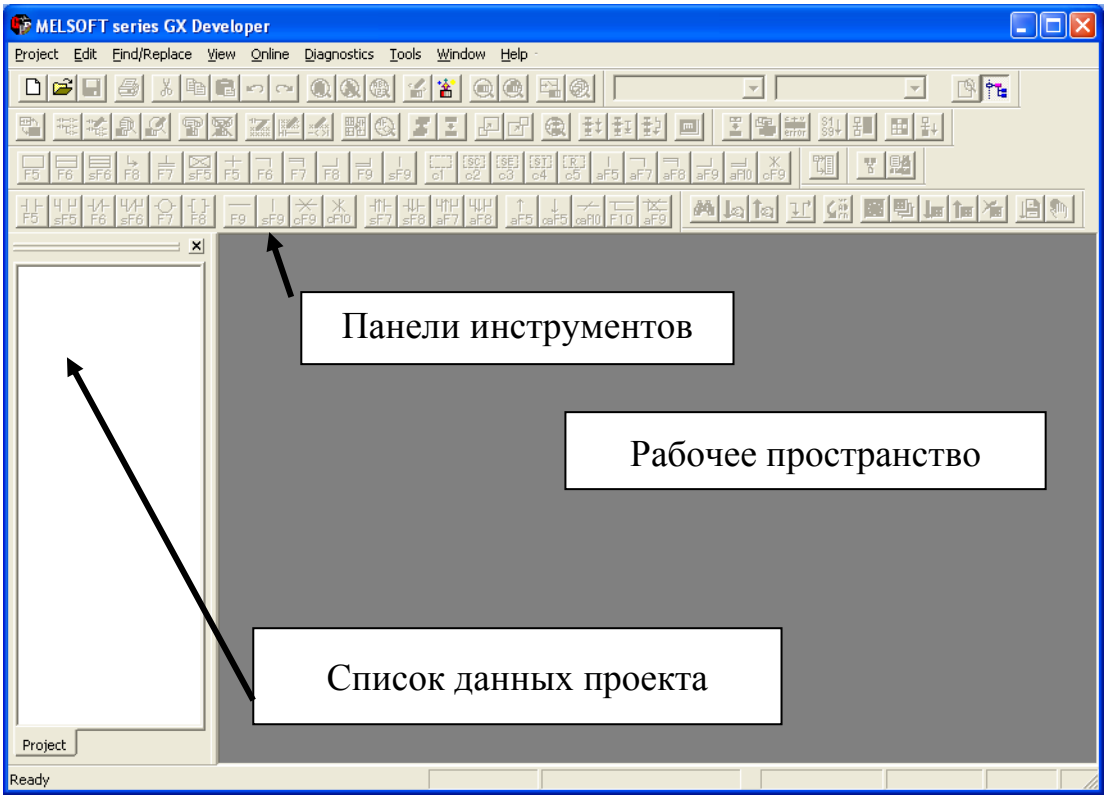
GX Developer – это работающая под **WINDOWS** программа для программирования/мониторинга релейных диаграмм. Для обеспечения связи между ПК и ПЛК используются **последовательный порт** или виртуальный последовательный порт (USB, Ethernet, Transparent Mode, и т.д.) .

GX Developer работает под Windows 95, 98, NT, 2000 и XP.

GX Developer можно запустить одним из двух способов:

1) дважды щелкнув на иконке программы, если она присутствует;
2) выбрав программу из меню Пуск. По умолчанию задан путь Пуск → Программы → MELSOFT Application → GX Developer.

После запуска появится экран, аналогичный одному из приведенных на следующей странице:



Темно-серая область – это рабочее пространство, где будут появляться все рабочие окна. Большая часть открытых панелей инструментов редко используются. Их можно закрыть, расширив рабочее пространство:

1) Перейдите в меню View и выберите "Toolbar".
2) Выберите "Standard" и "LD symbol", и снимите выделение с остальных.
3) Перейдите в меню View и найдите "Project Data List" (Список данных проекта).
4) Снимите выделение с "Project Data List" до того момента, пока он не понадобится.

11.2 Создание нового проекта

1) Новый проект можно создать тремя способами:
   - В меню Project выберите "New"
   - Нажмите Ctrl+N
   - Щелкните на иконке нового документа на инструментальной панели (значок чистого листа бумаги)
2) Выберите серию ПЛК из поля со списком. Для этого курса выберите FXCPU.
3) Выберите тип ПЛК из поля со списком. Для этого курса выберите FX3U(C).
4) Определять сначала имя пути и имя проекта, а также краткое название необязательно. На данном этапе их определять не нужно. Эта информация может быть задана позже, при сохранении проекта.
5) Нажмите "OK".

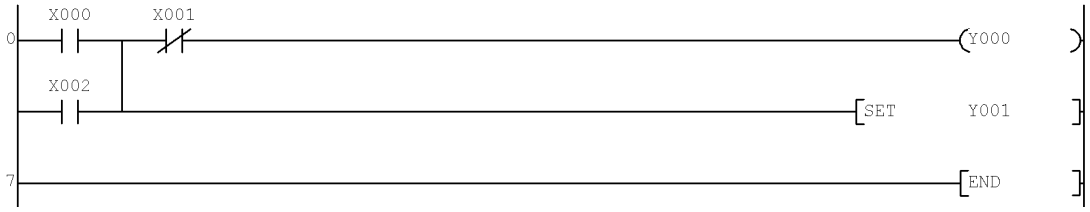
11.3 Редактирование релейных диаграмм

Следуя предложенной процедуре, создадим простую релейную диаграмму.

1) Щелкните на иконке (символе) нормально разомкнутого (NO) контакта
2) Напечатайте "X0" в появившемся диалоговом окне и щелкните на OK
3) Дважды щелкните в окне размещения (курсор, или синий квадрат по умолчанию)
4) В левой части диалогового окна имеется поле со списком. Выберите символ нормально замкнутого контакта (NO). Напечатайте "X1" в текстовом поле справа и нажмите OK.
5) Нажмите кнопку F7. Напечатайте "Y0" в текстовом поле и нажмите OK.

ПРИМЕЧАНИЕ Кнопка F7 может не работать в зависимости от выбранной раскладки клавиатуры. В меню инструментальных средств Tools, в пункте настройки клавиатуры Customize Keys можно выбрать один из трех стандартных наборов клавиатурных сокращений. 7 – это кнопка в формате MEDOC для флага выхода. В формате GPPQ и GPPA этой кнопкой является F7. Клавиатура на вновь установленной программе GX Developer по умолчанию имеет формат GPPQ. Его можно изменить на любую раскладку клавиатуры, наиболее знакомую и удобную для пользователя.

6) Не щелкая и не перемещая окно размещения, напечатайте "OR X2" и нажмите кнопку "Enter".
7) Перейдите в меню Edit и выберите символ релейной диаграммы "Ladder Symbol" затем "Application Instruction"(Применить инструкцию). Напечатайте "SET Y1" в текстовом поле и нажмите OK.



Шаги 1-7 показывают, как создать законченное звено кода релейных диаграмм, используя каждый из пяти способов ввода символов. Затем следуйте предложенной процедуре чтобы редактировать код языка релейных диаграмм.

8) Щелкните на символе "LD X2" (вторая строка).
9) Перейдите в меню Edit и выберите "Delete Line" (Удалить линию). Звено исчезает.
10) Возвратитесь в меню Edit и выберите "Undo" (Отменить). Звено вновь появляется.
11) Щелкните правой кнопкой на X000. Выберите "Delete Row" (Удалить строку) и нажмите "Yes". X000 и X002 исчезают.
12) Щелкните правой кнопкой на X001. Выберите "Delete Row" (Удалить строку) и нажмите "Yes". X001 исчезает.
13) Щелкните правой кнопкой на звене и выберите "Undo". X1 вновь появляется.
14) Попробуйте снова выбрать "Undo".

Обратите внимание, что команда Undo стала серой. Имеется только один уровень отмены.

Команды "Insert Rung" и "Insert Row " добавляют пространство для нового звена или нового контакта. Используйте "Insert Row", чтобы поместить X0 и X2 назад на их соответствующие места.

Но две линии больше не соединяются.

Соедините линии и звенья вместе, используя горячую клавишу "Free-draw Line" (Свободная линия, F10) или "Draw Vertical Line" (Нарисовать вертикальную линию, Shift+F9). Теперь две линии должны быть объединены вместе в одно звено.

Замечание что звено представлено серым. Это означает, что звено является только предварительным кодом релейных диаграмм и пока не установлено в программу. Перейдите в меню Convert и выберите "Convert" (Преобразовать), или преобразуйте релейную диаграмму, щелкнув правой кнопкой на рабочем пространстве или нажав кнопку F4. Серый цвет исчезает. Если предварительный код релейных диаграмм имеет ошибки или не образует законченной схемы, преобразование не будет выполнено и выдаст сообщение об ошибке. Программное обеспечение не позволит сохранить или загрузить программы, содержащие не преобразованный код.

Щелкните на меню View и выберите "Instruction List" (Список инструкций). Релейная логическая диаграмма исчезает и заменяется на сокращения и адреса:

```
0   LD        X000
1   OR        X002
2   MPS
3   ANI       X001
4   OUT       Y000
5   MPP
6   SET       Y001
7   END
```

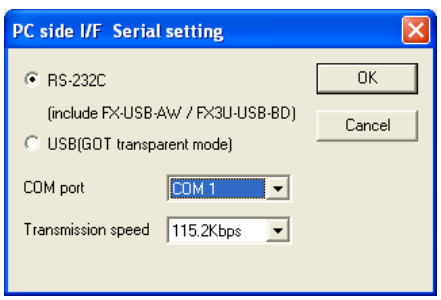
Это релейная диаграмма, написанная в формате списка инструкций – это формат программы, который действительно понимается ПЛК. Релейную диаграмму можно показать снова, возвратившись в меню View. Вместо "Списка инструкций", теперь снова показана "Релейная диаграмма".

Сохраните программу, щелкнув на иконке "Save" в инструментальной панели, или нажав Ctrl+S. Введите имя проекта FXPROG1. Щелкните на "Yes" в диалоговом окне, чтобы сохранить проект.

11.4 Передача программы

В программе GX Developer ПК связывается с ПЛК через порты аппаратного интерфейса (COM-порты), которые автоматически присваиваются ПК и Windows. Каждый компьютер будет иметь различные номера COM-портов для каждого устройства связи в ПК (RS-232, параллельного порта, USB, и т.д.). Чтобы проверить номера COM-портов, пользователь должен перейти на вкладку "Оборудование" элемента "Система" в "Панели управления" Windows . Однако, в этом курс обучения используется прозрачный режим GOT, чтобы избежать нумерации всех COM-портов. Это делается с помощью процесса, называемого "Передачей настроек".

1) В GX Developer перейдите в меню Online и выберите "Transfer Setup".
2) Должно открыться большое диалоговое окно с большим количеством различных кнопок и опций. Первая кнопка вверху слева, с названием "Serial" (Последовательный), должна быть выделена. Дважды щелкните на ней.
3) Для FX3U в этом диалоговом окне используются следующие настройки по умолчанию: "RS-232C", "COM 1", и "115.2 кбит/с" . Вместо них выберите "USB (GOT transparent mode)".
4) Опции "COM port" и "Transmission speed" исчезают. Нажмите "OK", чтобы зарегистрировать новые настройки связи.
5) Соедините ПК с передним USB портом

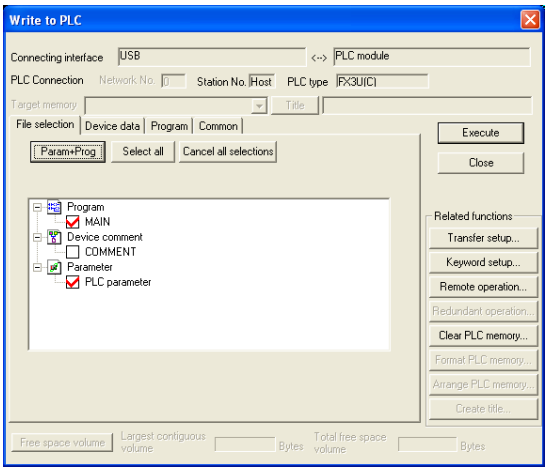


GOT (если он еще не соединен) и нажмите "Connection test" (в правой стороне большого диалогового окна), чтобы проверить соединение.

6) Если связь успешна, не забудьте выбрать "OK" в большом диалоговом окне, чтобы сохранить новые настройки связи.

Теперь можно передавать данные между ПК и ПЛК.

Чтобы передать программу на ПЛК, ПЛК должен быть в режиме STOP. Это можно сделать, установив переключатель с ключом на ПЛК в позицию STOP, или дистанционно, используя программное обеспечение. Если ПЛК находится в режиме RUN, можно вручную выполнить "Remote Stop" (Дистанционный останов), но если попытаться загрузить программу на ПЛК в режиме RUN, то программное обеспечение автоматически вызывает "Дистанционный останов" ПЛК и затем загружает программу. После завершения загрузки программы завершена будет вызван "Дистанционный запуск" ПЛК.



1) Перейдите в меню Online и выберите "Write to PLC".
2) Нажмите "Param+Prog" или вручную выберите "Program-Main" и "Parameter -> PLC parameter" из диалогового окна.
3) Нажмите "Execute" и затем выберите "Yes" для продолжения.

Появляется индикатор выполнения, отражая процесс загрузки программы, после чего всплывает другое диалоговое окно, указывающее, что загрузка закончена. Нажмите "OK". Возвратите ПЛК в режим RUN – переключателем или дистанционно.

Две другие опции в меню Online – Read from PLC" (Считать с ПЛК) и "Verify with PLC" (Сравнить с ПЛК).

"Read from PLC" выгружает программу из ПЛК в ПК и показывает ее в GX Developer. Это позволяет внести изменения в незащищенную программу в ПЛК, не имея исходного проекта GX Developer.

4) Выберите "Read from PLC" из меню Online.
5) Нажмите "Param+Prog" (Параметры + программа) или вручную выберите "Program-Main" и "Parameter -> PLC parameter" из диалогового окна.
6) Нажмите "Execute" (Выполнить) и затем выберите "Yes" для продолжения.
7) Сделав это, нажмите "OK", и закройте диалоговое окно.

"Verify with PLC" сравнивает программу, открытую в программе GX Developer с программой в ПЛК. Это особенно полезно в среде, где несколько служащих могут вносить изменения в программу. Это защищает программиста от случайной перезаписи несохраненных изменений, сделанных сотрудниками или им самим.

8) Выберите "Read from PLC" из меню Online.
9) Нажмите "Param+Prog" (Параметры + программа) или вручную выберите "Program-Main" и "Parameter -> PLC parameter" из диалогового окна.
10) Нажмите "Execute" (Выполнить) и затем выберите "Yes" для продолжения.

GX Developer выполнит сравнение и перечислит все несогласующиеся элементы.

11) Закройте экраны проверки, либо нажав кнопку X в правом верхнем углу окна, или выбрав "Close" в крайнем слева меню.

Спецификация диапазона шагов

Для FX3U с загрузкой не должно быть никаких проблем, для ПЛК, отличных от FX3U, может оказаться, что даже очень небольшие программы долго загружаются. Причина заключается в том, что независимо от количества логических шагов в релейной диаграмме, GX Developer всегда загружает по меньшей мере 8000 шагов для FX1N и FX2N с более низкой скоростью передачи данных, чем для FX3U. Можно значительно ускорить процесс, но необходимо быть внимательным, чтобы гарантировать загрузку всего кода, включая команду END.

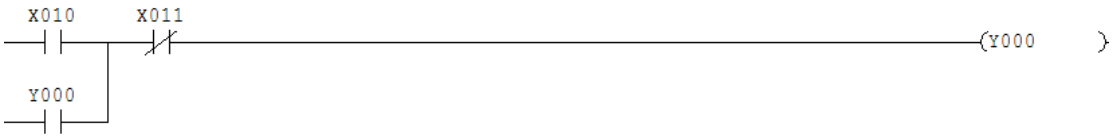
1) Определите номер конечного шага рядом со звеном с командой END.
2) Перейдите в меню Online и выберите "Write to PLC".
3) Как обычно, выберите "Programs-Main" и "Parameter-PLC parameter" .
4) Щелкните на вкладке "Program".
5) Измените "Range type" на "Step Range".
6) Соответственно отрегулируйте диапазон в колонках "Start" и "End".
7) Нажмите "Execute" и затем выберите "Yes" для продолжения.

11.5 Редактирование в режиме онлайн

Итак, мы написали небольшую программу и загрузили ее на ПЛК. Поскольку программа изменялась только на компьютере, эта процедура называется **Редактированием в режиме офлайн**. Если компьютер подключен к ПЛК, есть возможность изменять программу непосредственно в ПЛК, что избавляет от необходимости повторно загружать ее в ПЛК или переводить ПЛК в режим STOP. Это называется **Редактированием в режиме онлайн**. Учтите, что некоторые типы памяти ПЛК не поддерживают такую возможность, как отмечалось в разделе 2.13.

Монитор (Режим записи)

1) Перейдите в "Monitor" в меню Online.
2) Выберите "Monitor (Write Mode)".
3) Измените звено так, чтобы оно было похоже на показанное ниже.



4) Преобразуйте звено, чтобы записать изменения в ПЛК.

При этом новая программа существует в ПЛК, но не в ПК. Поэтому перед выходом из "Monitor (Write Mode)", не забудьте сохранить проект.

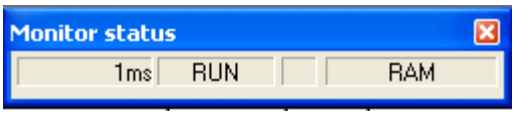
Учтите, что имеется опция "Convert (Online Change)" (Преобразовать (Изменить в режиме онлайн) которая эффективно преобразует и записывает релейную диаграмму в ПЛК; при этом ПЛК находится в режиме RUN. Ее можно найти в меню Convert, или нажав Shift+F4, находясь в "Write Mode" (Режим записи). Различие между данной опцией и "Monitor (Write Mode)" заключается в том, что значения операндов не будут показаны с релейной диаграммой в "Режиме записи".

Текущее звено называется схемой **Самоблокирующейся защелки**; она очень распространена. X010 – контакт без фиксации типа кнопки, включающей станок.

Без ответвления (вторая строка, OR Y000) станок работал бы, только пока нажата кнопка. Но в этой схеме при включении Y000 звено остается активным даже после выключения X010. Чтобы выключить станок, необходимо нажать кнопку останова X011.

11.6 Мониторинг выполнения программы

В GX Developer есть возможность просматривать выполнение блоков в выполняемой релейной диаграмме и контролировать значения битовых и словных операндов. Этот процесс называется **Мониторингом** программы.



1) Перейдите в "Monitor" в меню Online.
2) Выберите "Monitor Mode" (Режим мониторинга) или нажмите F3.

Появится небольшое окно, указывающее режим ПЛК (RUN или STOP) и максимальное время цикла, зарегистрированное к настоящему времени для релейной диаграммы.

Обратите внимание, что X010 и Y000 не подсвечены, а X011 подсвечен. Это указывает на активность вводов, или на то, когда активна катушка выхода. Подсвеченный контакт установлен (проводит ток), а подсвеченная катушка активизирована. Пока инструкции LD X010 и LD Y000 должны быть неактивны, потому что они являются нормально разомкнутыми контактами. С другой стороны, инструкция LDI X011, которая является нормально замкнутым контактом, должна быть активной.

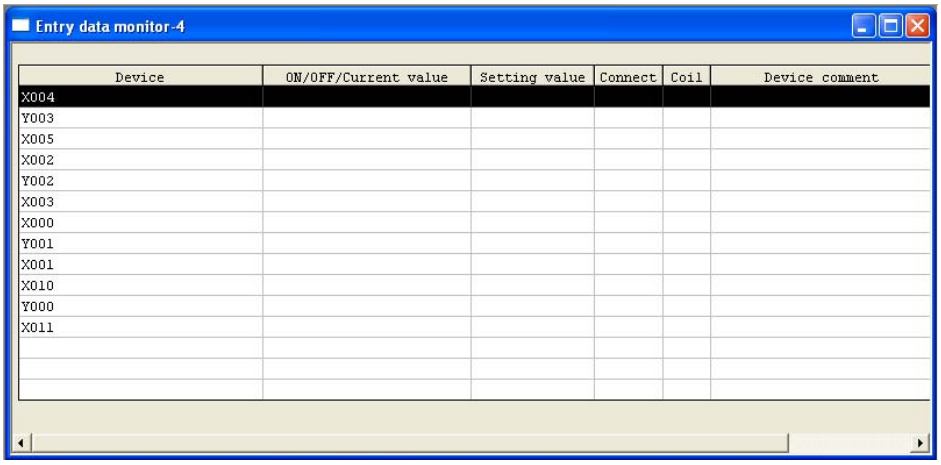
Включите X010, переключив соответствующий аппаратный переключатель цифрового входа. Когда переключатель переходит в состояние ON (ВКЛ), бит в программе подсвечивается. Переключите X011 в состояние OFF (ВЫКЛ). Обратите внимание, что бит больше не подсвечивается. Переключите X011 и X010 так, чтобы оба были подсвечены. Если все контакты (условия на входе) в звене подсвечены, звено является **истинным.** Когда звено является истинным, включается выход звена. Обратите внимание, что Y000 подсвечен на экране и что горит светодиод "Output 0" на FX3U.

Монитор элементов релейной диаграммы

Эта возможность позволяет контролировать несколько несмежных звеньев.

1) Перейдите в "Write Mode" в меню Edit или нажмите F2.
2) Выделите, скопируйте, и вставьте звено три раза в одну релейную диаграмму (всего 4 звена, 8 линий).
3) Измените адреса операндов контактов и катушек (т.е. X000, X001, Y001 и X002, X003, Y002, и т.д.) чтобы создать четыре различных звена.
4) Выполните преобразование "Convert (Online Change)".
5) Переключите релейную диаграмму в режим мониторинга "Monitor Mode".
6) Перейдите в "Monitor" в меню Online, и выберите "Entry Data Monitor" (Мониторинг данных элементов).
7) Выберите "Tile Horizontally" (сверху вниз) из меню Window .
8) В нижнем окне (окне кода релейных диаграмм) выделите 4^{e} звено.
9) Щелкнув на выбранном звене и не отпуская кнопку мыши, переместите его в верхнее окно
10) Повторите шаги 8 и 9 для 1го звена и 3го звена.

11) Щелкните в верхнем окне (Entry Data Monitor), чтобы сделать его активным, и разверните его.
12) Переключите окно в "Monitor Mode" (Режим мониторинга). Учтите, что каждое окно необходимо переключить в "Режим мониторинга" индивидуально.
13) Переключайте соответствующие тумблеры и наблюдайте за результатами.



11.7 Принудительная установка битов и изменение регистров

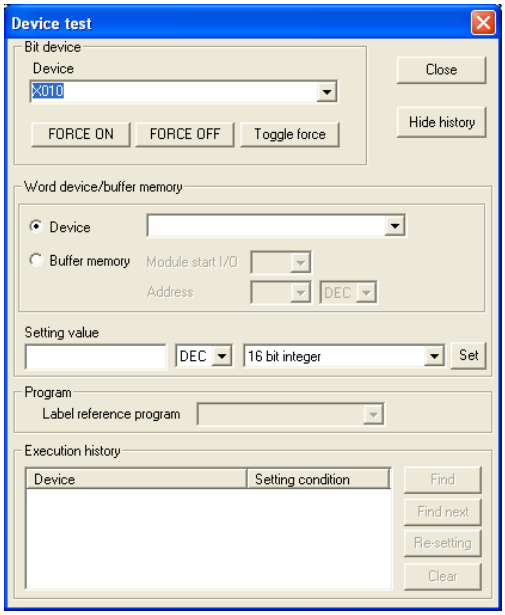
Зачастую целесообразно выполнять блоки кода ПЛК при написании программы. Это позволяет программисту проверять части кода, пока программа достаточно невелика и просто вносить изменения. Используя ПЛК и GX Developer, это можно сделать без помощи выключателей или других устройств. Этот метод называется **Принудительной установкой**.

1) Переключите ПЛК в режим RUN.
2) Используя релейную диаграмму из последнего раздела, перейдите в "Monitor (Write Mode)" в программе GX Developer.
3) Убедитесь, что аппаратные тумблеры для X010 и X011 находятся в положении OFF. Затем, удерживая клавишу SHIFT, дважды щелкните на X010.
4) Увидев изменение, сохраняйте X010 в состоянии OFF и, удерживая клавишу SHIFT, дважды щелкните на Y000. Что получилось?

Обратите внимание, что для X010, аппаратного входа, можно установить принудительное значение только временно. В ПЛК серии FX для аппаратных входов можно принудительно установить значение ON или OFF только на один цикл программы, после чего состояние аппаратного входа будет перезаписано программой. Аппаратные выходы, которые используются в релейной диаграмме, также можно установить принудительно на один цикл. Когда физические выходы (Y) не используются в релейной диаграмме, для них можно устанавливать принудительные значения без ограничений. Однако, когда физические входы (X) не подключены, их значения также можно устанавливать принудительно на один цикл. Адреса физических входов и выходов, клеммы которых не связаны с ПЛК физически, как X020 и Y020 в этом курсе обучения, можно устанавливать принудительно без ограничений. Внутренние биты, подобные M-маркерам, также могут принудительно устанавливаться, пока ими не управляет релейная диаграмма в ПЛК.

Это самый простой способ включать/выключать контакты и реле без внешних переключателей. Однако принудительная установка не рекомендуется, когда ПЛК подключен к работающей системе. Не предусмотрено никакого диалогового окна, предупреждающего о возможных изменениях, и принудительные изменения операндов могут привести к опасным результатам.

Другой способ принудительно задать значения операндов – использовать окно "Device Test".



1) Перейдите в "Monitor" в меню Online, и выберите "Entry Data Monitor".
2) Если не закрыт "Entry Data Monitor" из последнего раздела, дважды щелкните на X010. В противном случае, дважды щелкните в колонке "Device", напечатайте в диалоговом окне "X10" и щелкните "Register". Затем закройте диалоговое окно и дважды щелкните на X010.
3) Когда появится окно Device Test, щелкните на "Force ON".

Из этого окна также можно изменять числовые значения регистров данных.

4) В окне Device Test, напечатайте D0 в текстовом поле "Device" в секции "Word device/buffer memory".
5) Введите 10 в текстовое поле "Setting Value".
6) Щелкните на "Set", чтобы записать значение 10 в D0.

Как проверить, что значение 10 было записано в регистр данных D0.

7) Перейдите в "Monitor" в меню Online, и выберите "Device batch".
8) Введите D0 в текстовое поле "Device".
9) Щелкните на кнопке "Start Monitor".

11.8 Упражнение Контакты и катушки

Найдите проект 1 в приложении. Выполняя этот проект, студенты смогут практиковаться во вводе и управлении логикой релейных диаграмм.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 11.8; Упражнение – Контакты и катушки" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование приложения должно пройти без проблем. Этот экран предназначен для помощи студентам в визуализации приложений, он позволяет управлять даже простейшими кодами языка релейных диаграмм.

ГЛАВА 12 – Таймеры и счетчики

Таймеры и счетчики – стандартные части программы ПЛК. В этой главе будут рассмотрены различные типы таймеров и счетчиков, имеющийся в ПЛК серии FX, а также их программирование. Упражнения помогут пользователям в понимании концепций.

Цели главы: Завершив эту главу, слушатели смогут…

- Описать различные типы таймеров
- Узнать о наличии таймеров и счетчиков в ПЛК
- Описать формат команд для таймеров и счетчиков
- Описать ограничения для таймеров и счетчиков
- Перечислить типы уставок, имеющиеся для таймеров и счетчиков
- Написать программу, использующую таймеры

Материалы: Руководство по обучению FX-TRN-KIT-R
Руководство по программированию серии FX3U – Базовые и прикладные команды
Учебный стенд FX-TRN-KIT-R

12.1 Таймеры

Наличие

- ПЛК серии FX3U имеют 512 таймеров.
- ПЛК FX1N, FX2N, и FX2NC имеют 256 таймеров.
- ПЛК FX1S имеют 64 таймера.

Типы

Шаг времени зависит от используемого адреса операнда таймера.

- 100 мс (0.1 секунды);
- 10 мс (0.01 секунды);
- 1мс (0.001 секунды)

Шаг времени	Адреса операндов таймера на ПЛК		
	FX1S	FX1N/FX2N/FX2NC	FX3U
100 мс	0-62	0-199	0-199
10 мс	32-62*	200-245	200-245
1 мс (фиксируемый)	-	246-249	246-249
100 мс (фиксируемый)	-	250-255	250-255
1 мс	63	-	256-511

*M8028 можно установить для работы с 31 таймером с временным шагом от 100 мс до 10 мс.

Уставка времени

Уставка времени – это временной интервал, в течение которого таймер работает перед тем, как значение таймера прекращает увеличиваться и активизируется выход (контакт) таймера. Уставка задается кратной шагу времени таймера. Таким образом таймер T0 с уставкой 50 работает 5 секунд (50 x .1 секунды = 5 секунд).

Уставка времени должна быть целым числом в диапазоне от 1 до 32767, потому что все таймеры являются 16-битовыми регистрами и могут считать только в прямом направлении. Значения уставки могут быть либо значениями K (десятичных констант), либо словными операндами, типа регистра D. Когда уставка времени задается словными операндами, длительность таймера можно регулировать с интерфейса HMI или в релейной диаграмме, изменяя значение словного операнда.

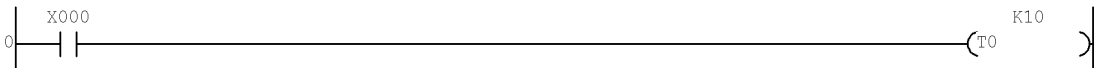
Сброс (фиксируемого таймера)

Если таймер является нефиксируемым, то достигнутое значение времени таймера сбрасывается на 0 и выход (контакт) таймера становится неактивным после того, как состояние на входе звена лестничной диаграммы будет неактивным. Чтобы сбросить значение достигнутого времени фиксируемого таймера к 0 и деактивировать контакт таймера, необходимо использовать команду RST.

Нефиксируемые таймеры теряют значение достигнутого времени и состояние контакта при переходе ПЛК в режим STOP или отключении электропитания. Фиксируемые таймеры сохраняют как значение достигнутого времени, так и состояние контакта при переходе ПЛК в режим STOP или отключении электропитания.

Программирование

Таймеры программируются с использованием команды OUT; задаются два аргумента – сначала адрес операнда таймера, а затем уставка времени. Результирующий код релейной диаграммы показан ниже.



В этом примере, когда X000 активен, T0 работает 1 секунду, (10 x 100 мс) перед тем, как активизируется его контакт.

12.2 Счетчики

Наличие

- ПЛК FX2N, FX2NC и FX3U имеют 256 счетчиков.
- ПЛК FX1N имеют 256 счетчиков.
- ПЛК FX1S имеют 45 таймеров.

Тип счетчика	Адреса операндов счетчика на ПЛК		
	FX1S	FX1N	FX2N/FX2NC/FX3U
16 бит	0-15	0-15	0-99
16-бит (фиксируемый)	16-31	16-199	100-199
32-бит (реверсивный)	-	200-219	200-219
32-бит (реверсивный, фиксируемый)	-	220-234	220-234
Высокоскоростные счетчики*	235-254**	235-255	235-255

* Хотя все счетчики C235 – C255 (21 точка) являются высокоскоростными, соответствующие физические входы (X0-X7) делятся между ними. Один физический вход не может использоваться более чем одним высокоскоростным счетчиком одновременно. Более подробные сведения см. в конце этого раздела Руководства по обучению и Разделе 4.7 Руководства по программированию FX3U.

** В FX1S не имеется C239, C240, C243, C245, C248, C250 или C253.

16-битовый счетчик

Уставка

Уставка – это количество переключений условий на входе счетчика из неактивного в активное состояние перед тем, как значение счета прекратит увеличиваться и активизируется выход (контакт) счетчика. Таким образом, если C0 имеет входное условие X000 и уставка = 5, то X000 должен 5 раз переключиться из состояния ВЫКЛ в состояние ВКЛ, перед тем, как выход C0 станет активным.

Для 16-битовых счетчиков уставка должна быть целочисленным значением в диапазоне от 1 до 32767, т.е. они могут считать только в прямом направлении. Значения уставки могут быть либо значениями K (десятичных констант), либо словными операндами, типа регистра D. Когда уставка задается словными операндами, предельное значение счета можно регулировать с интерфейса HMI или в релейной диаграмме, изменяя значение словного операнда.

Аккумулированное значение счета 16-битового счетчика никогда не превышает уставки. Став активным, контакт счетчика остается в активном состоянии, пока не будет сброшен. Даже если значение счета регулируется вручную, контакт счетчика не перейдет в неактивное состояние.

Направление счета

16-битовый счетчик может считать только в прямом направлении.

Сброс

Пока ПЛК находится в режиме RUN, аккумулированное значение счета 16-битового счетчика сбрасывается на 0, и выход (контакт) счетчика становится неактивным, только когда используется команда RST.

16-битовые счетчики C100-C199 являются буферизованными, и поэтому сохраняют значения счета и состояние контакта даже при переключении ПЛК в режим СТОП или отключении электропитания. 16-битовые счетчики C0-C99 теряют значения счета и состояние контакта при переходе ПЛК в режим STOP или отключении электропитания (если они не были объявлены как буферизованные батареей в параметрах ПЛК), после чего значение счета и состояние контакта сохраняются.

Ограничения

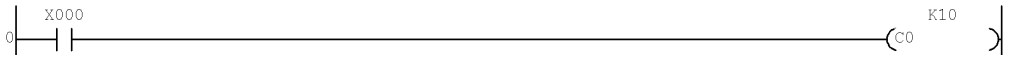
Отрицательные значения уставки недопустимы для 16-битовых счетчиков.

Единственный способ заставить 16-битовый счетчик считать в обратном порядке – уменьшать значение счета в коде релейных диаграмм перед тем, как значение счета достигнет соответствующей уставки. После того как контакт счетчика активизируется, единственный способ перевести его в неактивное состояние – использовать команду RST. Значение счета будет считаться в обратном порядке, но контакт счетчика останется активным.

Ручная настройка значение счета вверх возможна, но контакт счетчика не станет активным. Конечный счет до уставки должен производиться в результате перехода условия на входе звена счетчика из неактивного состояния в активное.

Программирование

16-битовые счетчики программируются с использованием команды OUT; задаются два аргумента – сначала адрес операнда счетчика, а затем уставка. Результирующий код релейной диаграммы показан ниже.



В этом примере каждый раз, когда X000 переключается из неактивного состояния в активное, значение счета C0 увеличивается. Когда значение счета = 10, его контакт активизируется.

32-битовый счетчик

Уставка

Уставка – это предельное значение счета, которое должно превысить аккумулированное значение счета (считая от значения ниже уставки до значения выше уставки) перед тем, как контакт счетчика станет активным. Таким образом, если используется C235 с уставкой 5000, значение счета C235 (запускаемое X000) должно перейти от 4999 к 5000 перед тем, как контакт C235 станет активным.

Для 32-битовых счетчиков уставка должна быть целочисленным значением между
-2 147 483 648 и 2 147 483 647, т.е. что они могут считать в прямом и обратном направлении. Значения уставки могут быть либо значениями K (десятичных констант), либо 32-битовыми словными операндами, типа двух смежных регистров D. Когда уставка задается словными операндами, предельное значение счета можно регулировать с интерфейса HMI или в релейной диаграмме, изменяя значение словного операнда.

Аккумулированное значение счета 32-битового счетчика никогда не превышает уставки. Став активным, контакт счетчика остается в активном состоянии, пока не будет сброшен, или значение счета не будет отсчитываться вниз, переходя значение уставки. Если значение счета регулируется вручную, переходя значение уставки, контакт счетчика не станет неактивным.

Направление счета

32-битовые счетчики могут считать в прямом и обратном направлении. Направление счета для 32-битовых счетчиков основано на состоянии специальных маркеров M8200-M8255 с адресом операнда, соответствующим адресу 32-битового счетчика. Если маркер неактивен, соответствующий счетчик считает вверх. Если маркер активен, соответствующий счетчик считает вниз.

Например, направление 32-битового счетчика C201 определяется M8201.

Сброс

Пока ПЛК находится в режиме RUN, аккумулированное значение счета 32-битового счетчика может быть сброшено на 0 и выход (контакт) счетчика может переключиться в неактивное состояние с использованием команды RST.

32-битовые счетчики C220-C234 являются буферизованными, и поэтому сохраняют значения счета и состояние контакта даже при переключении ПЛК в режим СТОП или отключении электропитания. 32-битовые счетчики C200-C219 теряют значения счета и состояние контакта при переходе ПЛК в режим STOP или отключении электропитания (если они не были объявлены как буферизованные батареей в параметрах ПЛК), после чего как значение счета, так и состояние контакта сохраняются.

Ограничения

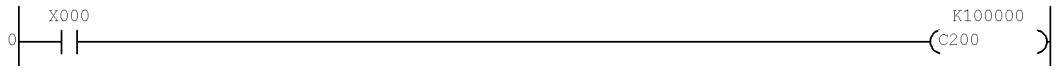
Хотя 32-битовый счетчик может иметь отрицательные значения уставки, все же контакт счетчика активизируется только при счете от меньшего числа до уставки.

Например, если счетчик C200 имеет уставку -10, то, когда счетчик считает в обратном порядке от -9 до -10, контакт счетчика не станет активным. Если счетчик считает в обратном порядке к -11, а затем в прямом направлении до -10, то контакт счетчика станет активным.

Ручная настройка значения счета возможна, но контакт счетчика не станет активным. Конечный счет до уставки должен производиться в результате перехода условия на входе звена счетчика из неактивного состояния в активное.

Программирование

32-битовые счетчики программируются с использованием команды OUT; задаются два аргумента – сначала адрес операнда счетчика, а затем уставка. Результирующий код релейной диаграммы показан ниже.



В этом примере каждый раз, когда X000 переключается из неактивного состояния в активное, значение счета C200 уменьшается. Когда значение счета переходит от 99 999 к 100,000, контакт счетчика активизируется.

Высокоскоростные счетчики

Обзор

Высокоскоростные счетчики требуются в ПЛК для счета серий импульсов входных сигналов (примерно 20 в секунду или выше); ПЛК обрабатывает их в релейной диаграмме. Эти входные сигналы могут приходить от периферийных устройств, например, ультразвуковых датчиков или от устройств позиционирования, вырабатывающих серии импульсов.

Технические данные

Высокоскоростные счетчики в ПЛК серии FX обладают всеми свойствами стандартного 32-битового реверсивного фиксируемого счетчика, но в качестве входных условий они используют фиксированные физические входы и комбинации физических входов. Кроме того, эти высокоскоростные счетчики можно разделить по категориям атрибутов, включая **аппаратные счетчики** или **программные счетчики**, и **1-фазные** или **2-фазные**. Число имеющихся высокоскоростных входов и допустимый частотный диапазон входов зависят от используемой модели ПЛК серии FX. Более подробные сведения о технических данных высокоскоростных входов см. в соответствующем руководстве пользователя.

Ограничения

Ограничения для аппаратного счетчика зависят только от технических характеристик аппаратных средств в ПЛК, которые остаются неизменным в серии в модели ПЛК. Например, ПЛК серии FX3U содержит шесть высокоскоростных встроенных физических входов, работающих в 1-фазном режиме в диапазоне до 100 кГц, соответствующих адресам C235-C240 и два входа для диапазона до 10 кГц, соответствующих адресам C244-C245. Как отмечалось в разделе 2.7 этого Руководства по обучению, максимальная входная частота может быть увеличена до 200 кГц с использованием FX3U-4HSX-ADP.

С другой стороны, программные счетчики используют прерывания программы для обработки входных импульсных сигналов, поэтому ограничения зависят от нескольких характеристик системы ПЛК (т.е. подключенных специальных адаптеров), и релейной диаграммы (т.е. количества используемых высокоскоростных команд). В FX3U, каждый адрес C241-C243 использует один высокоскоростной встроенный физический вход для 1-фазного счета, и другой для сброса значения высокоскоростного счетчика. Максимальная частота для этих счетчиков требует вычисления согласно таблице в разделе 4.7.10 Руководства по программированию FX3U.

Программирование

Высокоскоростные счетчики программируются с использованием команды OUT; задаются два аргумента – сначала адрес счетчика, а затем уставка. Различие между этими счетчиками и прочими битовыми счетчиками заключается в том, что входные условия в коде релейных диаграмм используются для выбора высокоскоростного счетчика, а не для увеличения/уменьшения значения счета. Результирующий код релейной диаграммы показан ниже.

В этом примере пока X010 активен, каждый раз, когда X000 переключается из неактивного состояния в активное, значение счета C235 увеличивается или уменьшается. Когда значение счета переходит от 99 999 к 100,000, контакт счетчика активизируется.

В данном курсе FX3U будет использован с аппаратными счетчиками C235 и C236, соответствующими высокоскоростным физическим входам X000 и X001, соответственно. Эти высокоскоростные входы можно имитировать, используя переключатели и регуляторы высокоскоростных входов на учебном стенде FX-TRN-KIT-R.

12.3 Примеры программ

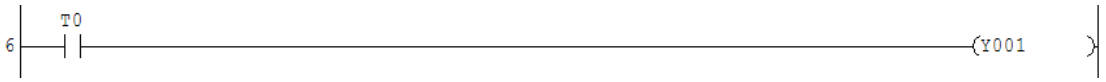
Создайте новую программу, введите следующее звено таймера и запишите его в ПЛК.

Разрабатывая логику, старайтесь думать, используя слова **И** и **ИЛИ**, например:

Когда X012 **И** X013 включены, **ИЛИ** когда X14 включен, таймер T0 работает.

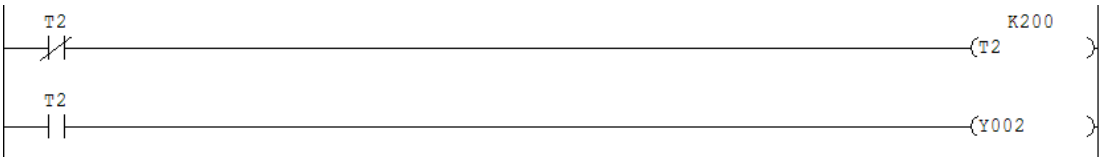
Переведите переключатель X014 в позицию ON и проверьте значение таймера в "Режиме мониторинга". Переведите переключатель X014 в позицию OFF и обратите внимание, что значение таймера сбрасывается на 0. Таймеры T0-T199 имеют временной шаг 100 мс, и T200-T245 имеют временной шаг 10 мс. Уставка K40 означает множитель 40 x 100 мс = 4 секунды. После того, как таймер достигнет 4 секунд, контакт T0 станет активным.

Добавьте следующий код релейных диаграмм после предыдущего звена и запишите его в ПЛК.



Переключите X012 **И** X013 в позицию ON. Когда T0 достигает 40, контакт T0 станет активным, включив выход Y001.

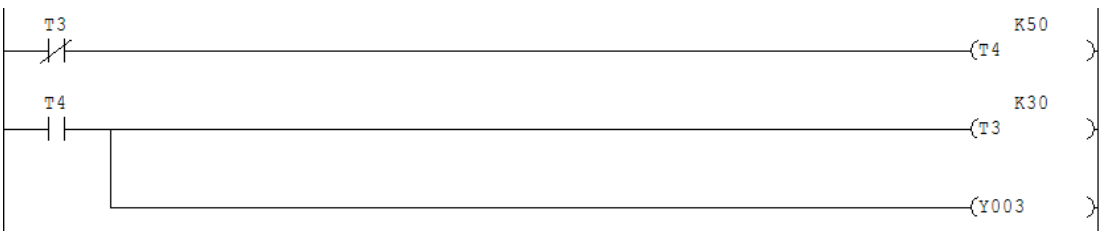
Это импульсный таймер.



Когда ПЛК включается, активизируется нормально замкнутый (NC) контакт T2. Это запускает таймер T2 до 20 секунд (200 x 100 мс). Когда таймер достигает 20 секунд, контакт T2 проводит ток, питая катушку Y002. Поскольку в этом примере используется нормально замкнутый контакт как входное условие для T2, то когда контакт T2 проводит, нормально замкнутый контакт становится неактивным, автоматически сбрасывая значение таймера и его контакт. Когда таймер сбрасывается, контакт T2 снова становится неактивным, заставляя нормально замкнутый контакт проводить, снова запуская таймер.

В результате нормально разомкнутый контакт T2 будет активный в течение 1 цикла релейной диаграммы каждые 20 секунды, генерируя регулярный выходной импульс.

Это триггерная схема.

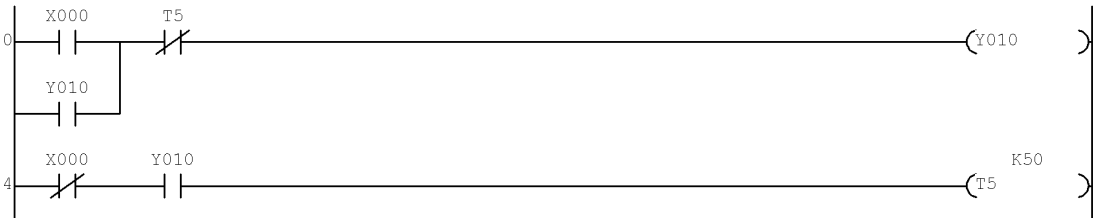


Первоначально нормально замкнутый контакт T3 проводит, запуская 5 секундный таймер T4. Когда контакт T4 активизируется, он запускает 3-секундный тай-

мер T3. Через 3 секунды активизируется контакт T3, разрывая нормально замкнутый контакт T3, что затем сбрасывает T4 и T3.

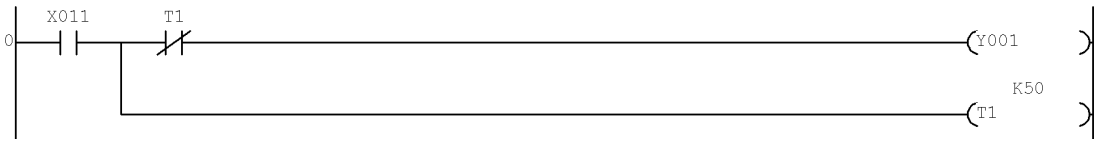
В результате Y003 будет сброшенным 5 секунд и затем установленным 3 секунды.

Другим полезным хронирующим устройством является таймер с задержкой отключения.



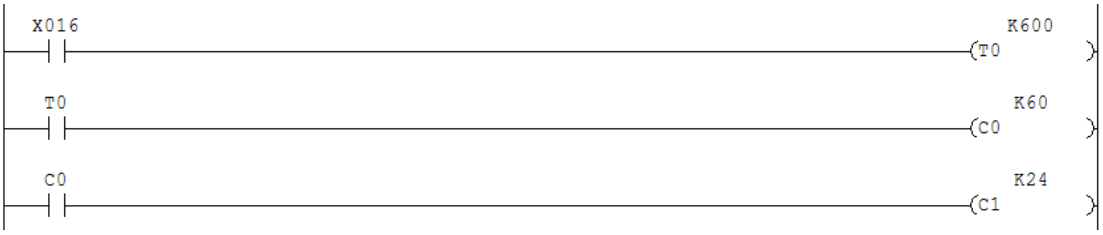
В этом примере X000 первоначально запускает Y010, когда T5 неактивен. Когда Y010 включен, он остается включенным независимо от X000 благодаря ветви. После отключения X000 нормально замкнутый контакт X000 и нормально разомкнутый контакт Y010 активны, запуская T5 за секунды перед тем, как нормально замкнутый контакт T5 сбрасывает первое звено наряду с Y010.

Иногда выход необходимо включить на заданное количество времени, независимо от того, как долго остается активным входное условие. Обычно это называется одновибратором.



В данном примере X011 первоначально устанавливает Y001. Одновременно начинает работать таймер T1. Когда активизируется выход T1, он сбрасывает Y001. Y001 не установится повторно до тех пор, пока не сбросится X011 и не сбросится таймер T1.

Быстрое вычисление показывает, что наибольший временной интервал, который может отрабатываться таймером, составляет 32 767 x 0.1 сек / 60 = 54,36 минут. Как быть, если необходимо создать таймер с большей временной уставкой? Одно из решений – использовать комбинацию совместно работающих таймеров и счетчиков, как показано ниже:



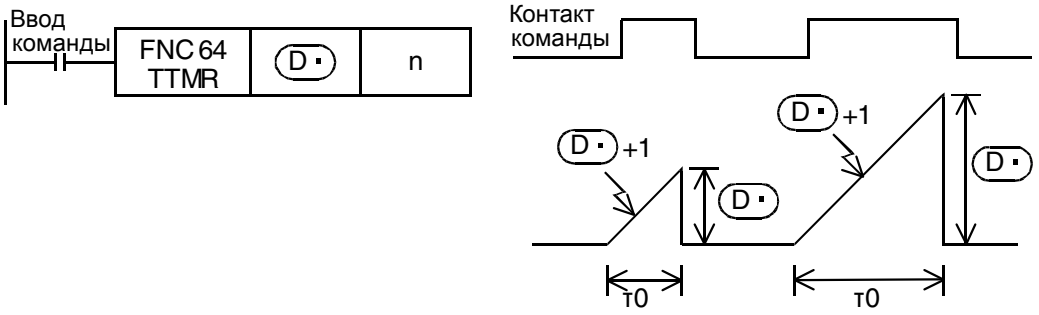
Пока X016 включен, T0 запускается на 1 минуту. Через 1 минуту T0 приводит к увеличению C0. После 60 приращений (1 час) C0 активизируется, увеличивая C1. Через 24 часа C1 станет активным и установит счетчик дней.

Замечание: **Показанная выше программа не закончена.** Что необходимо добавить, чтобы она работала должным образом?

12.4 Дополнительные команды таймера

Рассмотри еще две удобные команды таймера в системе команд ПЛК серии FX – Таймер обучения (TTMR) и Счетчик рабочих часов (HOUR).

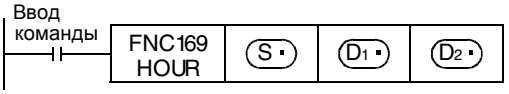
Команда TTMR измеряет количество времени, когда входные условия активны, используя временной шаг 100 мс. Он увеличивает это значение, преобразуя его во второе умножением на повышающий коэффициент, и сохраняет это увеличенное значение в указанный словный операнд (D·). Текущее измеренное значение с единицах 100 мс постоянно записывается в следующий последовательный словный операнд (D· +1). После того как значения записаны в устройстве памяти ПЛК, их можно использовать как значения уставки для других таймеров, и т.д.



Время нажатия и удерживания Время нажатия и удерживания

Команда берет количество накопленных секунд и умножает его на повышающий коэффициент 10^n, где n равно 0, 1 или 2. Чтобы сохранить реальное количество секунд, необходимо выбрать коэффициент K0 (10^1 = 1). Чтобы задать уставку для таймера 100 мс, используйте повышающий коэффициент K1. При этом берется число секунд и умножается на 10, чтобы получить число шагов по 100 мс. Для таймера 10 мс число секунд необходимо умножить на 100, или 10^2 (n = K2).

HOUR – встроенная команда счетчика рабочих часов. Команда считает число секунд, в течение которых ее входные условия были активны (D1· +1) и хранит число часов в указанном словном операнде (D·). После того, как будет достигнута уставка часов (S·) , устанавливается указанный выходной битовый операнд (D2·) .



Когда полное время, в течение которого входные условия команды активны, превышает время, сохраненное в (S·) (D2·) устанавливается равным "1".
Текущее значение, меньшее, чем один час, сохраняется в (D1·)+1 (единицы: секунды).

(S·) : Время, после которого (D2·) устанавливается равным "1".
: Задает значение в часах.
(D1·) : Текущее значение в часах.
(D1·) + 1 : Текущее значение меньше, чем один час (единицы: секунды).
(D2·) : Назначение выхода тревоги.
Он включается, когда текущее значение (D1·) превышает время, заданное в (S·)

Рекомендуется размещать операнды, используемые для хранения текущего значения в часов и секунд, в области энергонезависимых адресов ПЛК, чтобы они не терялись при переключении ПЛК в режим STOP или отключении электропитания. Эта функция может быть закодирована как DHOUR с использованием 32-битовых регистров, для хранения более длинных интервалов времени.

12.5 Упражнение Таймеры и счетчики

Найдите проект 2 в приложении. Выполняя этот проект, студенты смогут практиковаться в программировании таймеров и счетчиков, а также наблюдать за поведением таймеров и счетчиков.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 12.5; Упражнение – Таймеры и счетчики" на интерфейсе GOT. Этот экран поможет студентам визуализировать операции таймеров и счетчиков для облегчения дальнейшего понимания.

12.6 Упражнение Управление конвейером

Напишите программу, которая использует:

X010 как кнопку "Пуск" (контакт без фиксации – команда генерации импульса)
X011 как кнопку "Останов" (контакт без фиксации – команда генерации импульса)
M0 как фиксирующийся контакт
(остается установленным весь цикл, сбрасывается после отключения последнего конвейера)

Когда нажимается кнопка "Пуск", последовательно включаются выходы Y0 – Y7. Эти выходы соответствуют 8 конвейерам, которые должны включаться последовательно. Каждый конвейер включается через 1 секунду после включения предыдущего конвейера. Когда все конвейеры проработали 5 секунд, выключите конвейеры в обратном порядке по одному с интервалом в одну секунду. Кнопка "Останов" выключает все конвейеры и M0.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 12.6; Упражнение – Управление конвейером" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование приложения должно пройти без проблем. Этот экран поможет студентам визуализировать приложение, описанное в данном упражнении, для упрощения программирования и отладки.

ГЛАВА 13 – Прикладные команды

Прикладные команды являются "расширенными" инструкциями для ПЛК серии FX. Эти команды позволяют ПЛК выполнять сложные манипуляции с данными, математические операции и операции связи. Большинство прикладных команд работают на уровне 16-битовых или 32-битовых слов.

Цели главы: Завершив эту главу, слушатели смогут…

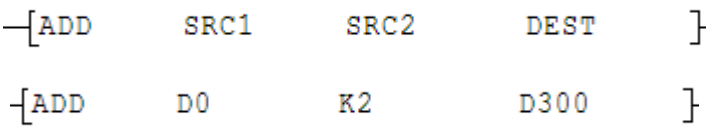
- Назвать наиболее распространенные прикладные команды.
- Описать формат этих инструкций и их функции.
- Написать несколько программ, используя эти инструкции.

Материалы: Руководство по обучению FX-TRN-KIT-R
Учебный стенд FX-TRN-KIT-R

13.1 Общий формат

В прикладных командах в программе GX Developer всегда используется символ скобок. Число аргументов изменяется в зависимости от команды.

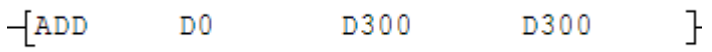
Пример



Эта команда прибавляет содержимое регистра источника 1 (D0) к содержимому регистра источника 2 (K2) и помещает результат в регистр назначения (D300).

Также возможно большую часть времени использовать регистр источника как регистр назначения:

Пример



В этом случае регистр источника 2 и регистры назначения являются одним регистром. Если до выполнение команды D0 содержит 9, а D300 – 200, то после выполнения команды 9 прибавляется к 200 и результат (209) сохраняется в D300.

По умолчанию прикладные команды являются 16-битовыми инструкциями. Если требуются манипуляции с 32-битовыми данными, пользователь должен добавить "D" как префикс к команде.

Пример: **MOV** передает 16 битов данных
DMOV передает 32 бита данных

Большинство этих инструкций выполняются один раз каждый цикл, пока активны входные условия. Иногда это нежелательно. Допустим, пользователь хочет увеличивать регистр данных на 1 (с помощью INC) при включении входа. Команда INC будет увеличивать регистр данных один раз каждый цикл, пока установлен вход. Этот может происходить сотни раз каждую секунду.

Чтобы избежать этого, пользователь должен применять импульсные инструкции, аналогичные описанным в разделе 10.3 данного Руководства по обучению. Чтобы преобразовать прикладную команду в импульсную версию, добавьте к команде суффикс "P".

Примеры

На каждом цикле разделить D0 на D1 и поместить результат в D20.

Выполняет деление только один раз, когда входные условия станут активными.

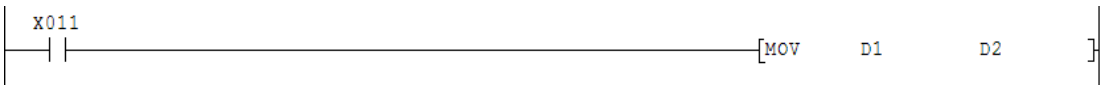
13.2 Команды передачи данных

Команды передачи данных следует поместить в конце звена (с правой стороны). Команда передачи данных выполняется, когда активны входные условия.

Команда передачи данных	16-битовые данные	32-битовые данные	Плавающая запятая
Регистр => Регистр	MOV	DMOV	DEMOV
Блок => Блок	BMOV	-	-
1 Регистр => Много регистров	FMOV	DFMOV	-

MOV – Передача данных
DMOV – Передача 32-битовых данных
DEMOV – Передача данных с плавающей запятой

Команда передачи (**MOV**) "перемещает", или фактически копирует, данные из регистра источника в регистр назначения (после выполнения команды оба регистра содержат одинаковые данные). Эта команда имеет два варианта: **DMOV** – используется для 32-битовых значений, и **DEMOV** – используется с 32-битовыми значениями с плавающей запятой.



В этом примере при включении X011 значение в регистре данных D1 копируется в D2.

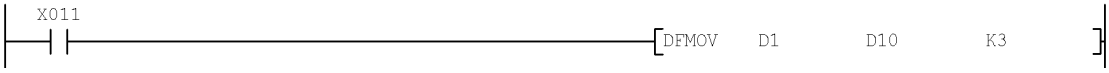
BMOV – Передача блока

Команда передачи блока данных (**BMOV**) "перемещает" (копирует) указанное число от словных операндов, начиная с регистра источника, в одинаковое число словных операндов, начиная с регистра назначения. Эта команда не работает с 32-битовыми данными, поскольку число "передаваемых" слов может просто удвоиться.

В этом примере при включении X011 три регистра данных начиная с D1 копируются в три регистра данных начиная с D7.

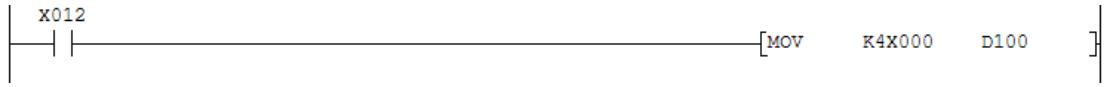
FMOV – Передача одинаковых данных в несколько операндов
DFMOV – Передача одинаковых 32-битовых данных в несколько операндов

Команда передачи одинаковых данных в несколько операндов (**FMOV**) "перемещает" (копирует) указанное значение данных из источника в указанное количество словных операндов, начиная с регистра назначения. Команда **DFMOV** использует 32-битовые числа, так что источник относится к двум последовательным словным операндам, и назначение – к 2х число указанных слов начиная с регистра назначения.



В этом примере при включении X011 регистры данных D1 и D2 копируются в D10 и D11, D12 и D13, а также D14 и D15.

Для любой из инструкций передачи данных можно указать группу битовых операндов, используя префикс из "K" и числа между 1 и 8.



В этом примере при включении X012 16 битов (4 полубайта по 4 бита в каждом) начиная с X000 копируются в 16-битовый регистр данных D100.

Значение после "K" определяет, какое количество полубайтов будет использовано в операции.

K1 = 1 полубайт (4 бита)
K2 = 2 полубайта (8 битов)
K3 = 3 полубайта (12 битов)
K4 = 4 полубайта (16 битов)
K5 = 5 полубайтов (20 битов)
K6 = 6 полубайтов (24 бита)
K7 = 7 полубайтов (28 битов)
K8 = 8 полубайтов (32 бита)

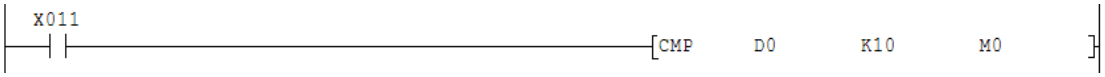
13.3 Команды сравнения

Команды сравнения данных следует поместить в конце звена (с правой стороны). Команда сравнения данных выполняется, когда активны входные условия.

Команда сравнения	16-битовые данные	32-битовые данные	Плавающая запятая
Значение => Значение	CMP	DCMP	DECMP
Значение => Диапазон	ZCP	DZCP	DEZCP

CMP – Сравнение
DCMP – Сравнение 32-битовых значений
DECMP – Сравнение значений с плавающей запятой

Команда сравнения (**CMP**) сравнивает одно 16-битовое значение со вторым 16-битовым значением и записывает результаты сравнения в группу 3-битовых операндов, которые могут быть Y-выходами, M-маркерами или S-маркерами. 3 бита результата представляют ситуации, когда данные источника 1 больше, чем равны, или меньше, чем данные источника 2, соответственно.



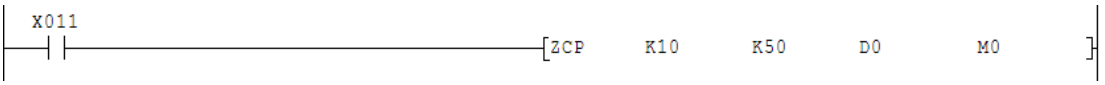
В этом примере при включении X011 биты M0, M1 и M2 показывают результаты сравнения следующим образом:

M0 включается, если значение в D0 больше, чем 10.
M1 включается, если значение в D0 равняется 10.
M2 включается, если значение в D0 меньше, чем 10.

Эта команда имеет два варианта: **DCMP**, который сравнивает два 32-битовых значения, и **DECMP**, который сравнивает два 32-битовых значения с плавающей запятой.

ZCP – Сравнение числовых областей данных
DZCP – Сравнение числовых областей 32-битовых данных
DEZCP – Сравнение числовых областей данных с плавающей запятой

Команда сравнения (**ZCP**) сравнивает одно 16-битовое значение с указанной областью, и записывает результаты сравнения в группу 3-битовых операндов, которые могут быть Y-выходами, M-маркерами или S-маркерами. 3 бита результата представляют ситуации, когда данные источника 3 меньше чем, в пределах, или больше, чем область, указанная данными источника 1 и 2.



В этом примере при включении X011 биты M0, M1 и M2 показывают результаты сравнения областей следующим образом:

M0 включается, если значение в D0 меньше, чем 10.
M1 включается, если значение в D0 находится между 10 и 50, включительно.
M2 включается, если значение в D0 больше, чем 50.

Учтите, что значение данных источника 1 должно быть меньше значения данных источника 2. В противном случае ПЛК все еще примет команду, но она не будет функционировать должным образом.

Эта команда имеет два варианта: **DZCP**, который сравнивает 32-битовое значение с
32-битовой областью, и **DEZCP**, который сравнивает 32-битовое значение с плавающей запятой с 32-битовой областью с плавающей запятой.

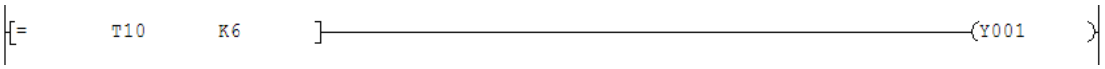
СРАВНЕНИЕ ДАННЫХ В РАМКАХ ЛОГИЧЕСКИХ ОПЕРАЦИЙ

В отличие от всех обсуждавшихся инструкций сравнения, инструкции сравнения данных в рамках логических операций можно поместить в любом месте звена. Они функционируют таким же образом, как любое входное условие или контакт. Эти инструкции имеются только на ПЛК FX1S, FX1N, FX2N(C) и FX3U. Данные команды отсутствуют на более старых ПЛК серии FX.

В этом примере катушка Y001 включается, когда значение в D0 меньше, чем 10.

Команда сравнения данных в рамках логических операций	16-битовые данные	32-битовые данные
Больше, чем	LD>	LDD>
Больше, чем или равно	LD>=	LDD>=
Равно	LD=	LDD=
Меньше, чем или равно	LD<=	LDD<=
Меньше, чем	LD<	LDD<
Не равно	LD<>	LDD<>

Инструкции сравнения данных в рамках логических операций позволяют просматривать накопленное значение таймеров или счетчиков, чтобы включать различные выходы в разное время или при значении счета.



В этом примере, когда значение таймера T10 достигает 6, включается выход Y001.

16-битовое и 32-битовое сравнения

Рассмотренные выше инструкции сравнения, сравнения числовых областей данных, и сравнения данных в рамках логических операций, по умолчанию являются 16-битовыми. Как описано в начале раздела, если требуется 32-битовое сравнение, необходимо добавить "D" к команде. Например, **DCMP**, **DZCP**, **LDD=**, **LDD>**, и т.д.

Очень важно помнить об этом при сравнении значений, например, высокоскоростных счетчиков. Такие значения счета являются 32-битовыми по умолчанию, и не будет функционировать должным образом со стандартной 16-битовой командой сравнения.

13.4 Упражнение Автомобильная парковка

Напишите программу для управления автомобильной парковкой:

X010 указывает прибытие автомобиля.
X011 указывает выезд автомобиля.
Y000 – знак, который включается, указывая, что парковка заполнена.
C200 (реверсивный 32-битовый счетчик) отслеживает число автомобилей в парковке.
D0 хранит максимальное количество автомобилей.

Когда автомобиль въезжает, количество автомобилей увеличивается. Когда автомобиль выезжает, текущее состояние счетчика уменьшается на 1. Когда парковка заполнена, включается знак "Парковка заполнена". В этом упражнение предположим, что автомобили въезжают и выезжают из парковки не в строго заданные моменты времени, но как в реальной жизни, что необходимо учитывать.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.4; Упражнение – Автомобильная парковка" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование приложения должно пройти без проблем. Используйте сенсорный экран GOT, чтобы задать для начального значения D0 значение 10. Позже используйте различные значения D0 для облегчения дальнейшего понимания. Этот экран поможет студентам визуализировать приложение, описанное в данном упражнении, для упрощения программирования и отладки.

13.5 Упражнение Управление конвейером, часть 2

Перепишите программу управления конвейером (разд. 12.6), используя ТОЛЬКО ОДИН таймер с командами сравнения для управления выходами.

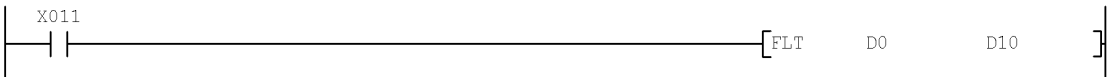
После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.5; Упражнение – Управление конвейером 2" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование приложения должно пройти без проблем. Используйте сенсорный экран GOT, чтобы выбрать таймер для мониторинга. Этот экран поможет студентам визуализировать приложение, описанное в данном упражнении, для упрощения программирования и отладки.

13.6 Команды преобразования

Команды преобразования следует поместить в конце звена (с правой стороны). Команда преобразования данных выполняется, когда активны входные условия. Имеются инструкции, позволяющие преобразовывать данные в и из большинства числовых форматов и ASCII строк. Инструкции, которые будут рассмотрены в этом курсе обучения, преобразуют целые числа в числа с плавающей запятой и обратно.

FLT – Целое число в число с плавающей запятой
DFLT – 32-битовое целое число в число с плавающей запятой

Команда **FLT** осуществляет преобразование 16-битового целого числа в 32-битовое значение с плавающей запятой. Команда **DFLT** осуществляет преобразование 32-битового целого числа в 32-битовое значение с плавающей запятой.



В этом примере при включении X011 целочисленное значение, сохраненное в D0, преобразуется в значение с плавающей запятой и сохраняется в D10 и D11.

Замечание Помните, что числа с плавающей запятой можно также представлять как константы, используя префикс "E", как описано в главе 6 данного Руководства по обучению.

INT – Число с плавающей запятой в 16-битовое целое число
DINT – Число с плавающей запятой в 32-битовое целое число

Команда **INT** осуществляет преобразование 32-битового числа с плавающей запятой в 16-битовое целочисленное значение. Преобразуется только целочисленная часть . Любые дробные значения будут потеряны. **Команда** DINT осуществляет преобразование 32-битового числа с плавающей запятой к 32-битовое целое число.

В этом примере при включении X010 значение с плавающей запятой, сохраненное в D10 и D11, преобразуется в 16-битовое целочисленное значение и сохраняется в D1. Учтите, что если значение с плавающей запятой, которое используются в команде INT, является целым числом, имеющим разрядность более 16-битов, команда не будет выполняться. Значение в регистре назначения останется.

13.7 Команды приращения и отрицательного приращения

Команды приращения и отрицательного приращения следует помещать в конце звена (с правой стороны). Команда выполняется, когда активны входные условия.

INC – Приращение
DINC – 32-битовое приращение
DEC – Отрицательное приращение
DDEC – 32-битовое отрицательное приращение

Команды приращения и отрицательного приращения (**INC** и **DEC**) просто прибавляют или вычитают 1 из значения в 16-битовом регистре данных. Эти инструкции выполняются почти с удвоенной скоростью по сравнению с командами ADD или SUB и не связаны с ограничениями счетчика. Инструкции можно закодировать для работы с 32-битовыми числами как **DINC** и **DDEC**. Учтите, что все названные инструкции будут, вероятнее всего, использоваться как импульсные инструкции (**INCP**, **DECP**, **DINCP** и **DDECP**).

В этом примере при включении X011 регистр D1 увеличивается на 1 каждый цикл.

В этом примере при включении X011 регистр D1 уменьшается на 1 каждый цикл.

13.8 Упражнение INC и DEC

Найдите проект 3 в приложении. Проект предназначен для демонстрации различия между INC/DEC и INCP/DECP.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.8; Упражнение – INC и DEC" на интерфейсе GOT. Если код релейных диаграмм введен правильно, отличия между INC/INCP и DEC/DECP должны проявиться немедленно. Этот экран поможет студентам визуализировать инструкции, которые используются в проекте, для облегчения дальнейшего понимания.

13.9 Арифметические команды

Арифметические команды следует поместить в конце звена (с правой стороны). Команда выполняется, когда активны входные условия.

Арифметическая команда	16-битовые данные	32-битовые данные	Плавающая запятая
Сложение	ADD	DADD	DEADD
Вычитание	SUB	DSUB	DESUB
Умножение	MUL	DMUL	DEMUL
Деление	DIV	DDIV	DEDIV
Квадратный корень	SQR	DSQR	DESQR

Обратите внимание, что все 32-битовые инструкции, а также все инструкции с плавающей запятой снабжены префиксом "D", поскольку все они

оперируют с 32-битовыми числами. Инструкции, которые также снабжены префиксом "E", указывают на формат с плавающей запятой. Учтите, что благодаря природе умножения и деления, 16-битовые инструкции производят 32-битовый результат, а 32-битовые инструкции производят 64-битовый результат. Более подробные сведения см. в Руководстве по программированию FX3U.

Целочисленный квадратный корень требует несколько более подробного объяснения. Значение рассчитанного квадратного корня будет целым числом; все дробные значения игнорируются, если не используется режим с плавающей запятой. Если разряды десятичной дроби игнорируются, устанавливается специальный маркер M8021.

В этом примере каждый цикл при включении X011 к значению в регистре D0 прибавляется 30, и результат помещается в D10.

13.10 Упражнение Бинарная математика

Найдите проект 4 в приложении. Выполняя этот проект, студенты смогут практиковаться в использовании арифметических команд.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.10; Упражнение – Бинарная математика" на интерфейсе GOT. Если код релейных диаграмм введен правильно, арифметический операции должны точно выполняться в зависимости от входов X010-X013. Этот экран поможет студентам визуализировать арифметические команды в обеих 16-битовой (или 32-битовой) десятичной и бинарной форме для облегчения дальнейшего понимания.

13.11 Упражнение Автомобильная парковка, часть 2

Это модификация программы для программы автомобильной парковки из раздела 13.4. Добавьте код релейных диаграмм для подсчета общего количества автомобилей, въехавших на стоянку за день, и сохраните это значение в D10. Также добавьте код релейных диаграмм для подсчета количества денег, которые должны быть в кассе в конце дня, предполагая, что каждый автомобиль платит $4.50 за въезд, и сохраните это значение в D12-D13 (отметим, что это дробное значение). Также необходимо добавить кнопку сброса, которая позволит менеджеру сбрасывать количество автомобилей и полную выручку в конце дня. Кнопка сброса должна быть M10. Как и в прошлый раз, в этом упражнении предположим, что автомобили въезжают и выезжают из парковки не в строго заданные моменты времени, но как в реальной жизни, что необходимо учитывать.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.11; Упражнение – Автомобильная парковка 2" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование приложения должно пройти без проблем. Используйте сенсорный экран GOT, чтобы задать для начального значения D0 значение 10 и кнопку M10 для сброса количества автомобилей и полной выручки. Позже используйте различные значения D0 для облегчения дальнейшего понимания. Этот экран поможет студентам визуализировать приложение, описанное в этом упражнении, для упрощения программирования и отладки.

13.12 Упражнение Управление конвейером, часть 3

Это модификация программы для программы управления конвейером из раздела 13.4. Добавьте код релейных диаграмм, чтобы задать число циклов, которое система конвейеров отработает (D0) непрерывно. Число циклов должно иметь значение между 5 и 15. В противном случае, значение за пределами диапазона 5 – 15 не позволит системе конвейеров начать работу. В этом упражнении предположим, что D0 не изменяется в ходе работы, но учтите, что в реальной ситуации необходимо учитывать такую возможность.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.12; Упражнение – Управление конвейером 3" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование приложения должно пройти без проблем. Используйте сенсорный экран GOT, чтобы регулировать значение регистра данных D0. Этот экран поможет студентам визуализировать приложение, описанное в этом упражнении, для упрощения программирования и отладки.

13.13 Быстродействующая обработка

Как обсуждалось в разделе 12.2, FX3U имеет встроенные высокоскоростные счетчики и встроенные высокоскоростные входы и выходы (см. краткое описание в разделе 2.7).

Высокоскоростные инструкции счетчика

Как уже отмечалось, высокоскоростные счетчики являются либо аппаратными счетчиками, которые выполняются независимо от цикла ПЛК, либо программными счетчиками, которые выполняются как прерывания, когда они считают, также за пределами цикла. Таким образом, преимущество высокоскоростных счетчиков заключается в том, что они функционируют за пределами цикла.

Важно помнить, что высокоскоростные счетчики являются 32-битовыми счетчиками, поэтому при использовании значений высокоскоростного счета необходимы 32-битовые инструкции типа "D".

Хотя имеется возможность использовать с высокоскоростными счетчиками стандартные инструкции SET, RST и инструкции сравнения, эти инструкции являются зависимыми от цикла, а потому ограничивают преимущества высокоскоростных счетчиков. Чтобы полностью реализовать преимущества высокоскоростных счетчиков, используйте следующие высокоскоростные инструкции обработки счетчиков.

HSCS – Установка с помощью высокоскоростного счетчика

Инструкция установки с помощью высокоскоростного счетчика (**HSCS**) функционирует, как стандартная команда SET. Когда указанный высокоскоростной счетчик достигает заданного значения, устанавливается указанный битовый операнд. Эта команда использует прерывания и является независимой от цикла.

В данном примере инструкция устанавливает Y010 (на "1"), когда значение высокоскоростного счетчика C255 равно 100.

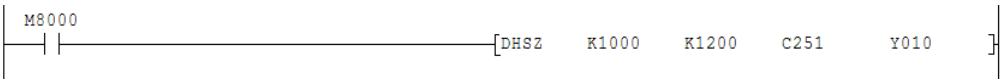
HSCR – Сброс с помощью высокоскоростного счетчика

Инструкция сброса с помощью высокоскоростного счетчика (**HSCR**) функционирует как стандартная команда RST. Когда указанный высокоскоростной счетчик достигает заданного значения, сбрасывается указанный битовый операнд. Эта команда использует прерывания и является независимой от цикла.

В данном примере инструкция сбрасывает Y010 (на "0"), когда значение высокоскоростного счетчика C255 равно 200.

HSZ – Высокоскоростное сравнение областей

Команда высокоскоростного сравнения областей (**HSZ**) функционирует, как стандартная команда ZCP. Значение указанного высокоскоростного счетчика сравнивается с диапазоном значений и результаты сравнения хранятся в 3-х последовательных битовых операндах. Учтите, что команда HSZ может использоваться в релейной диаграмме только один раз .



В этом примере результаты имеют следующий вид:

Y010 включается, когда C251 < 1000.
Y011 включается, когда 1000 <= C251 <= 1200.
Y012 включается, когда C251 > 1200.

Заметим, что DHSCS, DHSCR, и DHSZ используются как операции вместо HSCS, HSCR, и HSZ. Помните, что высокоскоростные счетчики являются 32-битовыми операндами, так что перед командой должен находиться префикс "D".

Инструкции высокоскоростного ввода и вывода

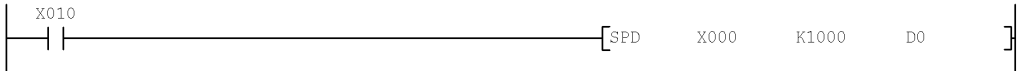
Как уже отмечалось, ПЛК серии FX включают встроенные высокоскоростные физические входы, а некоторые модели имеют встроенные высокоскоростные физические выходы. FX3U имеет максимальную частоту высокоскоростного входа 100 кГц для всех моделей. Модели FX3U транзисторного типа также поддерживают максимальную частоту высокоскоростного выхода 100 кГц. Спецификации ввода-вывода для FX3U можно расширить, используя адаптеры высокоскоростного ввода и вывода.

В этом курсе обучения мы рассмотрим только две наиболее распространенных базовых команды. Подробнее о быстродействующей обработке и инструкциях позиционирования см. в главах 13, 20, 24 и 32 Руководства по программированию FX3U.

SPD – Распознание скорости

Команда распознавания скорости (**SPD**) измеряет частоту входных импульсов, полученных клеммой высокоскоростного входа за указанный временной интервал (с интервалами 1 мс) и помещает результат в 16-битовый словный операнд.

Имеется также команда **DSPD**, которая измеряет частоту сигнала высокоскоростного входа и помещает результат в 32-битовый словный операнд.



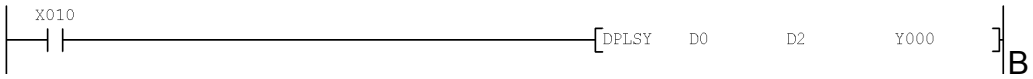
В этом примере при включении X010 частота импульсов за 1 секунду (1000 x 1 мс, или 1 Гц) на высокоскоростном входе X000 каждый цикл записывается в D0.

Учтите, что адрес высокоскоростного физического устройства ввода (т.е. X000) для этой команды не может одновременно использоваться для любого другого высокоскоростного счетчика или команды быстродействующей обработки.

PLSY – Вывод импульсов Y

Команда вывода импульсов Y (**PLSY**) выходит серию импульсов указанной длины и частоты (Гц) через клемму высокоскоростного выхода. Когда для длины серии импульсов указан 0, выводится непрерывная последовательность импульсов. Эта команда может использоваться для позиционирования серии импульсов с серводвигателями и инверторами.

Имеется также команда **DPLSY**, которая выведет 32-битовые сигналы на 32-битовых частотах.



В этом примере при включении X010 серия импульсов 32-битовой длины, хранящейся в D2-D3 и 32-битовой частоты, хранящейся в D0-D1, выводится на клемму высокоскоростного выхода Y000.

Учтите, что входные условия для команды PLSY/DPLSY должны быть активными на весь период, когда выводится серия импульсов. После отключения управляющей логической связи серия импульсов остановится независимо от того, было выведено указанное число импульсов, или нет. Кроме того, после завершения вывода серии импульсов, для повторного выполнения того же звена, входные условия должен быть выключены, а затем снова включены, чтобы вывести следующую серию импульсов. Чтобы выяснить, действительно ли завершена команда PLSY/DPLSY, используйте специальный маркер M8029, который включается, когда были выведены все импульсы в серии импульсов. За дополнительной информацией о M8029 обращайтесь к Руководству по программированию FX3U.

13.14 Упражнение Быстродействующий ввод-вывод

Напишите релейную диаграмму, которая моделирует энкодер с разрешением 3 000 000 импульсов (0 – 3 000 000), используя высокоскоростной счетчик C235, считающий серию импульсов на входе X000. Используйте сравнения, чтобы быть уверенным, что C235 остается в пределах своего диапазона разрешения, и используйте переключатель X010, чтобы переключать направление энкодера (счет вверх или вниз). Одновременно сравните значение энкодера в области от 1 000 000 до 2 000 000 с результатами, записанными в Y010-Y012. Имитируемый датчик подключен к входной клемме X001; частоту входной серии импульсов в Гц необходимо измерить и сохранить в D0. Y010 запускает непрерывную серию импульсов на выходной клемме Y000 с частотой из D0. Y011 запускает непрерывную серию импульсов на выходной клемме Y001 с частотой 1000 Гц. Y012 запускает серию 100 000 импульсов на выходной клемме Y002 с частотой из D0.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.14; Упражнение – Высокоскоростной ввод/вывод" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование должно пройти без проблем. Используйте высокоскоростные входы для X000 и X001 и цифровой вход X010 на учебном стенде, чтобы моделировать входы энкодера и датчика. Этот экран поможет студентам визуализировать имитируемый энкодер, датчик, и систему высокоскоростного выхода, описанную в данном упражнении, для облегчения дальнейшего понимания инструкций быстродействующей обработки и для упрощения программирования и отладки.

13.15 Левая шина FX3U

ПЛК серии FX3U имеют левую шину, связанную непосредственно с ЦП. Эта шина используется с платами расширения (BD) и специальными адаптерами (ADP) для реализации различных функций. Важно понимать, что основное различие между BD и ADP левой шины и специальными функциональными модулями (SFM) правой шины заключается в том, BD и ADP левой шины не содержат никакой внутренней памяти. Для устройств, использующих левую шину, вся обработка выполняется через память операндов и ЦП FX3U. Специальные маркеры (M8000 и выше) и специальные регистры данных (D8000 и выше) используются для специализированных инструкций и целей. Следующие два раздела будут содержать демонстрационные проекты, показывающие использование левой шины. Использование правой шины мы рассмотрим позже, в разделе 13.18 – Команды TO/FROM. Полный список специальных адресов памяти операндов, функции и цели см. в главе 36 Руководства по программированию FX3U.

13.16 Упражнение Температурный датчик

Это упражнение содержит пример использования специальной памяти операндов, ассоциированных с BD и ADP левой шины FX3U, и демонстрирует некоторые измерительные возможности температурного датчика.

В этом упражнении FX3U-4AD-PT-ADP адресуется как 1^{й} аналоговый адаптер, подключенный к FX3U, и Канал-1 подключен к температурному датчику PT100 на учебном стенде FX-TRN-KIT-R. Технические данные ADP приведены в Руководстве по аналоговым устройствам FX3U. Специальный маркер M8260 используется для переключения измеренной температуры между градусами Цельсия и Фаренгейта, и специальный регистр данных D8264 хранит время усреднения измерения температуры. Измерение температуры из Канала-1 первого аналогового адаптера хранится в D8260 в целочисленной форме, но с точностью значения с одним десятичным разрядом после запятой. Состояние ошибки для 1го аналогового адаптера хранится в D8268, и код модели хранится в D8269.

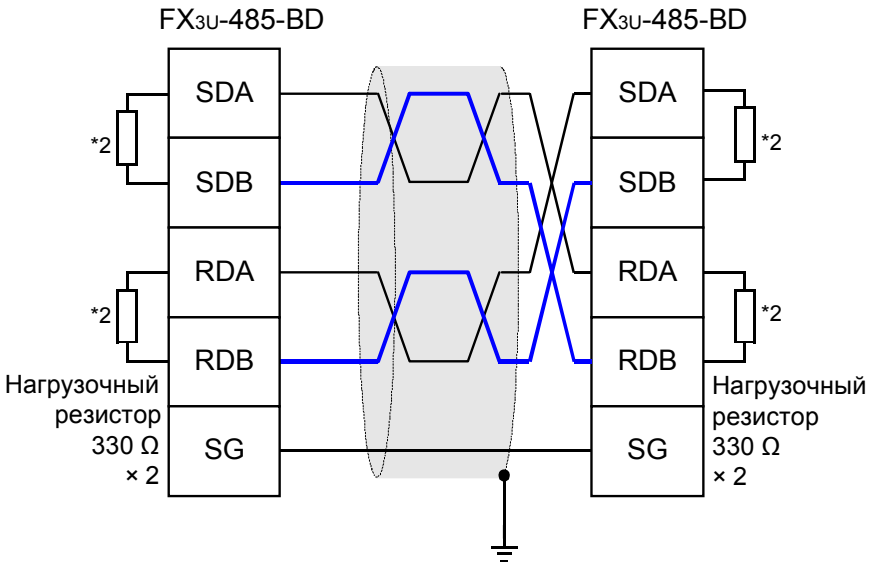
Перейдите на экран "Раздел 13.16; Упражнение – Температурный датчик" на интерфейсе GOT. Температура будет выводиться на экран интерфейса GOT. Используйте сенсорный экран GOT, чтобы переключаться между градусами Цельсия и Фаренгейта, и регулировать время усреднения измерения температуры (D8264). Учтите, что операции с ADP выполняются автоматически, поэтому не требуется никакого кода релейных диаграмм. Этот экран показывает студентам простоту адаптера температурного датчика, а также прочих аналоговых адаптеров, для облегчения дальнейшего понимания.

13.17 Упражнение Parallel Link

Для выполнения этого упражнения потребуются два учебных стенда FX-TRN-KIT-R, а также предоставленные пользователем кабели витой пары. Если любой из перечисленных компонентов отсутствует, пожалуйста пропустите этот раздел. Дополнительную информацию о Parallel Link и всех возможностях последовательной передачи данных в FX3U, см. в Руководстве по передаче данных в FX3U.

Найдите проект 5 в приложении. Выполняя этот проект, студенты смогут практиковаться в использовании специальной памяти операндов, ассоциированной с BD и ADP левой шины в FX3U и показать пример встроенной возможности работы с сетями последовательной передачи данных в FX. Эта релейная диаграмма также показывает пример использования индексного регистра.

Используйте следующую монтажную схему установки(скопированную из Руководства по передаче данных в FX3U), в которой два учебных стенда FX-TRN-KIT-R, расположенные на расстоянии 50 м друг от друга, соединены кабелями витой пары. Для устройств RS-485 в ПЛК серии FX3U не требуются согласующие резисторы.



Заземление класса D (сопротивление: 100 Ω или меньше)

Parallel Link является последовательным соединением между главным/подчиненным устройствами. Оно связывает два однотипных ПЛК серии FX (т.е. FX3U и FX3U) и используется для автоматического обновления специализированных системных операндов на обоих ПЛК. Это означает, что главный ПЛК на соединении будет постоянно записывать область битовых и словных операндов в связанном ПЛК, и считывать область битовых и словные операндов из этого ПЛК. При этом специализированная область битовых и словных операндов будет одинаковой на обоих ПЛК. Для этой иллюстративной программы будет использован регулярный режим Parallel Link, который обновляет M800-M999 и D490-D509. Более детальное описание Parallel Link и его технические данные приведены в Руководстве по передаче данных в FX3U.

В релейной диаграмме проекта 5 для Parallel Link не требуется задание каких-либо специальных параметров связи, поэтому для специального регистра данных D8120 должно быть задано значение по умолчанию 0. Кроме того, Parallel Link является саморегенерующейся, т.е., как только главное устройство успешно обнаруживает подчиненное, Parallel Link активи-

зируется. Единственный дополнительный код релейных диаграмм, необходимый для соединения, должен установить специальный маркер M8070 на главной станции и M8071 на подчиненной станции. В рассматриваемой иллюстративной программе M0 будет использован для переключения между главным и подчиненным устройствами. Учтите, что на учебном стенде FX-TRN-KIT-R используется FX3U-485-BD, т.е. применяется канал последовательной передачи данных 1 (Ch1). На других системах, если должен использоваться Ch2, необходимо установить M8178 в дополнение к специальному маркеру на главном или подчиненном устройстве . Релейная диаграмма проекта 5 создает подключение Parallel Link и управляет клеммами физического выхода Y000-Y003 на обоих ПЛК, определяя простую логику с нормально разомкнутым и нормально замкнутым контактом. Также регистры данных D490-D491 и D500-D501 обновляются между ПЛК, когда Parallel Link активна, что показывается как перемещение X/Y-координат на интерфейсе GOT. С помощью индексных регистров Z0 и V0 одну релейную диаграмму можно использовать как на главном, так и на подчиненном стендах, применяя смещение адресов маркера и регистра данных адреса. За дополнительной информацией об индексных регистрах обращайтесь к Руководству по программированию FX3U.

После того как релейная диаграмма была введена GX Developer и записана в оба ПЛК на учебных стендах, перейдите на экраны "Раздел 13.16; Упражнение – Parallel Link" на ОБОИХ интерфейсах GOT. Переключайтесь между главным устройством (нормально разомкнутый контакт M0) и подчиненным (нормально замкнутый контакт M0), нажимая кнопки "Master" и "Slave" на сенсорном экране интерфейса GOT или двойную стрелку. Если код релейных диаграмм введен правильно и подключение выполнено правильно, Parallel Link станет активной, как только главное устройство соединится с подчиненным. На интерфейсе GOT это показано зеленым индикатором, представляющим M8072. Используйте X010-X017 на каждом стенде, чтобы активизировать и увеличивать соответствующие маркеры и регистры данных на каждой станции. Обратите внимание, что битовые операнды и словные операнды обновляются автоматически между двумя станциями. Учтите, что только три звена кода языка релейных диаграмм в этом иллюстрационном проекте относятся к Parallel Link. Данный экран поможет студентам визуализировать простоту Parallel Link и опции работы с сетями последовательной передачи данных для облегчения дальнейшего понимания.

13.18 Команды TO/FROM

Как отмечалось в предыдущих разделах, основное различие между BD и ADP левой шины и специальными функциональными модулями (SFM) правой шины заключается в том, что BD и ADP левой шины не содержат никакой внутренней памяти. Внутренняя память в SFM называется буферной памятью (BFM), из которой SFM выполняет все свои функции. Инструкции **TO** и **FROM** позволяют ПЛК иметь доступ к BFM.

BFM можно рассматривать как большую группу 16-битовых словных операндов, в которых каждое слово или бит слова соответствуют различному параметру или команде. Чтобы ПЛК мог иметь доступ к любому из этих параметров или дать команду SFM, он должен писать в или считывать BFM. 16-битовые словные операнды, из которых состоит BFM, имеют собственные адреса. Эти адреса увеличиваются, начиная просто с BFM #0.

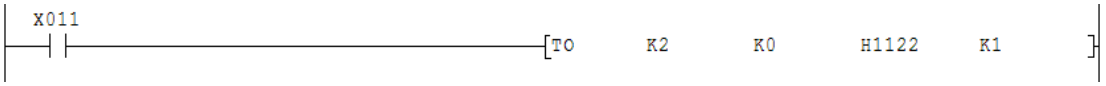
Замечание Для программирования функционального модуля SFM необходимо иметь список содержимого буферной памяти BFM для этого SFM.

Без списка содержимого BFM невозможно знать, какие параметры нужно устанавливать, распределение параметров по адресам BFM, и адреса BFM, запускающие конкретные команды.

Инструкция TO используется для записи 16-битовых данных в буферную память BFM в модуле SFM. Как правило, эти данные являются параметром или командой, которая будет управлять функциями модуля.

Формат команды:

Пример:



В этом примере каждый цикл при включении X011 одно слово данных (K1), значение константы (H1122), записывается в BFM #0 (K0), в 3^{ем} модуле SFM, подключенном к ПЛК (K2).

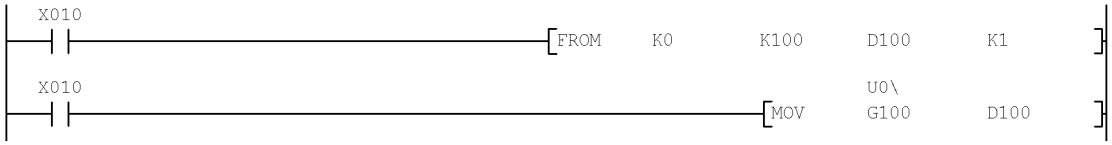
Команда FROM используется для считывания 16-битовых данных из BFM модуля SFM в память операндов ПЛК. Эти данные могут быть преобразованным значением аналогового входа, или состоянием модуля, и т.д. Данные из буферной памяти BFM копируются в указанные операнды в ПЛК, где они могут в дальнейшем обрабатываться ПЛК.

Формат команды:

Учтите, что имеются также 32-битовые версии инструкций TO и FROM – **DTO** и **DFROM**, которые записывают или считывают два адреса буферной памяти BFM одновременно. При использовании 32-битовых версий важно отметить, что отсчет операндов команды производится в 32-битовых приращениях, а не в 16-битовых, как в командах TO и FROM.

U□\G□ – Доступ к буферной памяти (только в FX3U)

Как отмечалось в главе 6, U□\G□ также могут использоваться вместо адреса операнда для доступа к отдельным 16-битовым словным операндам в буферной памяти BFM при использовании других прикладных команд в FX3U. Например, следующие два звена являются эквивалентными.



В обоих звеньях этого примера, каждый цикл при включении X010 считывается BFM #100 из 1го модуля SFM (SFM 0) и сохраняется в D100 в ПЛК. Учитывайте, что имеется различие во времени обработки между двумя методами. Более подробные сведения см. в Руководстве по программированию FX3U.

13.19 Упражнение Управление аналоговыми значениями

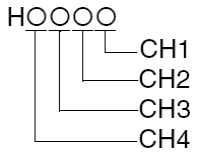
Найдите проект 6 в приложении. Выполняя этот проект, студенты смогут практиковаться в использовании инструкций TO и FROM для правой шины ПЛК серии FX. Кроме того, здесь демонстрируются некоторые возможности управления аналоговыми значениями.

В этом упражнении FX2N-5A адресуется как 1й модуль SFM (SFM 0), подключенный к FX3U, с входным каналом-1 (CH1) и входным каналом-2 (CH2), соединенным к двум **регуляторам аналоговых входов** на учебном стенде FX-TRN-KIT-R. Оба регулятора аналоговых входов имеют приблизительный диапазон ввода от 0 до +10 В=. Канал вывода-1 (CH1) подключен к **Измерителям аналогового выхода** на учебном стенде FX-TRN-KIT-R. Измеритель аналогового выхода показывает диапазон напряжений от 0 до 10 В= (помните, что отрицательные напряжения не показываются). FX2N-5A имеет максимальное напряжение аналогового входа и выхода от -10 до +10 В= с выбираемыми диапазонами цифровых значений. Например, диапазон цифровых значений от -32 000 до +32 000 означает, что цифровое значение -32 000 соответствует -10 В=, 0 представляет 0В=, а +32 000 представляет +10 В=, и т.д. Более подробное объяснение технических данных FX2N-5A приведено в Руководстве пользователя FX2N-5A.

Ниже показан пример типичного описания BFM, в данном случае BFM #0; он взят из Руководства пользователя FX2N-5A. Описание всех функциональных возможностей буферной памяти приведено в Руководство пользователя FX2N-5A.

Подробное описание буферной памяти

BFM 0 - спецификация режима входа (СЧИТЫВАНИЕ/ЗАПИСЬ)
BFM 0 указывает режим входов CH1 - CH4. BFM состоит из 4-разрядного шестнадцатеричного кода, где каждому входному каналу присвоен один разряд. Область значений для каждого разряда - шестнадцатеричное значение в диапазоне 0 - F.
Старший разряд соответствует входу ch4, а младший разряд соответствует входу ch1.
BFM#0



Разряды имеют следующие значения:
0: Режим входа напряжения (-10 ... +10 В) (Диапазон отображения -32000 ... +32000)
1: Режим токового входа (4 - 20 мА) (Диапазон отображения 0 ... +32000). Если значение тока меньше 2 мА, тревога по ошибке диапазона будет установлена в BFM 28
2: Режим токового входа (-20 ... +20 мА) (Диапазон отображения -32000 ... +32000)
3: Режим входа напряжения (-100 ... +100 мВ) (Диапазон отображения -32000 ... +32000)
4: Режим входа напряжения (-100 ... +100 мВ) (Диапазон отображения -2000 ... +2000)
5: Режим индикации вольтметра (-10 В ... + 10 В) (Диапазон отображения -10000 ... +10000)
6: Режим индикации амперметра (4 - 20 мА) (Диапазон отображения 2000 ... +20000 = 2 мА ... 20 мА). Если значение тока меньше 2 мА, тревога по ошибке диапазона будет установлена в BFM 28
7: Режим индикации амперметра (-20 мА ... +20мА) (Диапазон отображения -20000 ... +20000)
8: Режим индикации вольтметра(-100 мВ ... + 100 мВ) (Диапазон отображения -10000 ... +10000)
9: Функция масштабирования (-10 ... +10 В) (максимальный диапазон отображения -32768 ... +32767); по умолчанию = -32640 ... +32640
A: Функция масштабирования для режима токового входа (-20 ... +20 мА) (максимальный диапазон отображения -32768 ... +32767); по умолчанию = -32640 ... +32640
B: Функция масштабирования для режима входа напряжения (-100 ... +100 мВ) (максимальный диапазон отображения -32768 ... +32767); по умолчанию = -32640 ... +32640
F: канал заблокирован, канал всегда возвращает 0.
C ... E: недопустимы; модуль автоматически восстановит последнюю допустимую настройку.

Такие характеристики входа, как настройки смещения и коэффициента передачи, автоматически изменяются с настройками BFM 0. Измерение режима в BFM 0 будет также воздействовать на настройки BFM 41 - 44 (данные смещения) и BFM 51 - 54 (данные коэффициента передачи) или настройки BFM 200 к 239 (данные функции масштабирования). Перед изменением данных смещения/коэффициента передачи или масштабирования данных данные, необходимо задать опции режима в BFM 0. В противном случае данные смещения/коэффициента передачи или функции масштабирования будет перезаписана данными по умолчанию выбранного режима.
Состояние выхода за пределы допустимого диапазона (BFM 28), существовавшие перед изменением режима, не будут автоматически сбрасываться при изменении режима.
Блокирование канала приведет к увеличению частоты просмотра других каналов. Значением по умолчанию BFM 0 является H0000.

Значение BFM 0 хранится энергонезависимым образом во внутреннем EEPROM. Имеется функция безопасности, защищающая внутренний EEPROM от потери данных при случайной записи одного значение во всю память BFM 0.

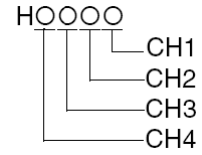
В релейной диаграмме проекта 6 BFM #0 будет инициализироваться значением H55, которое указывает на диапазон аналогового входа от -10 до +10 В=, соответствующий области цифровых значений от -10 000 до +10 000. BFM #1 будет инициализироваться значением по умолчанию 0, который указывает на диапазон аналогового выхода от -10 до +10 В=, соответствующий диапазону цифровых значений от -32 000 до +32 000. В BFM #2 и BFM #3 хранятся значения времени усреднения CH1 и CH2, соответственно. BFM #6 и BFM #7 содержат усредненные данные аналогового входа из CH1 и CH2, соответственно. BFM #14 и BFM #15 обрабатывают значение аналогового выхода. BFM #14 содержит выходное значение цифро-аналогового преобразователя (ЦАП), но когда используется функция прямого управления (BFM #23), BFM #15 содержит отрегулированное значение выхода ЦАП. В этом упражнении будут использоваться различные функции прямого управления, чтобы регулировать значение аналогового выхода. См. приведенное ниже описание BFM #23, взятое из Руководства пользователя FX2N-5A.

BFM 23 Задание параметров для прямого управления между каналом ввода и каналом вывода (СЧИТЫВАНИЕ/ЗАПИСЬ)

В BFM 23 пользователь может задать обратную связь прямого управления между всеми 4-мя каналами аналогового ввода и аналогового вывода.
Формат BFM 23 представляет собой 4-разрядное шестнадцатеричное значение, и каждый разряд представляет работу одного входного канала.

Значения шестнадцатеричных разрядов имеют следующий смыл:
H0: Соответствующий аналоговый входной канал никак не воздействует на значение аналогового выхода.
H1: Среднее входное значение (BFM 6 - BFM 9) соответствующего аналогового входного канала прибавляется к значению аналогового выхода в BFM 14.
H2: Непосредственное входное значение (BFM 10 - BFM 13) соответствующего аналогового входного канала прибавляется к значению аналогового выхода в BFM 14.
H3: Среднее входное значение (BFM 6 - BFM 9) соответствующего аналогового входного канала вычитается из значения аналогового выхода в BFM 14.
H4: Непосредственное входное значение (BFM 10 - BFM 13) соответствующего аналогового входного канала вычитается из значения аналогового выхода в BFM 14.
H5 - HF: Соответствующий аналоговый входной канал никак не воздействует на значение аналогового выхода, однако, будет установлен флаг ошибки прямого управления выходом (бит 15) в BFM 29.

BFM#0



Пример: Значение в BFM 23 установлено равным H1234.
Значение выхода (BFM 15) = BFM 14(TO) - BFM 10 - BFM 7 + BFM 12 + BFM 9 Если по меньшей мере один шестнадцатеричный разряд в BFM 23 установлен равным шестнадцатеричному числу между 1 и 4, то после вычисления значения цифрового выхода в BFM 15, к этим цифровым данным применяется вычисление настройки смещения/коэффициента передачи или вычисление функции масштабирования применяется, чтобы обеспечить реальный аналоговый выход. Если функция прямого управления выключена для всех каналов, BFM 14 будет рассматриваться как значение аналогового выхода.
Настройки BFM 25 будут также воздействовать на значения функции прямого управления.

В этом упражнении в D0 будет храниться среднее входное значение аналого-цифрового преобразования (АЦП) из CH1, а в D1 – среднее входное значение АЦП из CH2. Время усреднения аналогового входа для CH1 и CH2 будет управляться D2 и D3, соответственно. Значение выхода ЦАП

будет храниться в D10, отрегулированное значение аналогового выхода (BFM #15) – в D11. Функция прямого управления будет управляться M0-M7 (поскольку используются только два канала аналогового ввода). X000 и X001 предназначены для управления функцией прямого управления CH1, в то время как X004 и X005 – для CH2. Когда индивидуально включаются X000 или X004 , соответствующие значения цифровых входов из CH1 и CH2 будут автоматически прибавляться к значениям выхода ЦАП в SFM и записываться в BFM #15. Когда в дополнение к X000 или X005 включаются X001 или X005, значения цифровых входов из CH1 и CH2 автоматически вычитаются из значений выхода ЦАП в SFM и записываются в BFM #15. Таким образом, аналоговый выход FX2N-5A непосредственно регулируется аналоговыми входами на FX2N-5A и цифровыми входами на FX3U. После инициализации X010 активизирует все инструкции FROM, и X011 активизирует все инструкции TO. Это демонстрирует различие между данными в ПЛК и данными в буферной памяти BFM модуля SFM.

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.19; Упражнение – Управление аналоговыми значениями" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование должно пройти без проблем. Используйте сенсорный экран GOT для регулирования времени усреднения аналогового входа (D2-D3) и значения выхода ЦАП (D10). Используйте X000-X001 и X004-X005, чтобы переключаться между функциями прямого управления. Помните, что измеритель аналогового выхода на FX-TRN-KIT-R может показывать только значения в диапазоне от 0 до +10 В=. Если код релейных диаграмм введен правильно, вы сможете увидеть, как данные на экране интерфейса GOT отличаются от аналоговых входов и выходов на учебном стенде, когда не активизированы инструкции TO и/или FROM. Регуляторы аналоговых входов и измеритель аналогового выхода используют данные непосредственно из модуля SFM, тогда как GOT и переключатели цифровых входов воздействуют на ПЛК. Этот экран поможет студентам визуализировать возможное использование инструкций TO и FROM для облегчения дальнейшего понимания, а также продемонстрирует некоторые имеющиеся опции управления аналоговыми значениями.

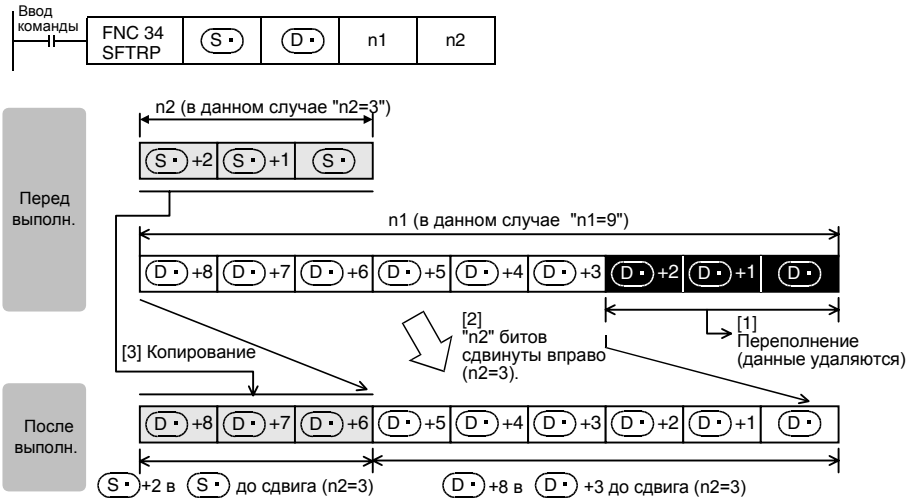
13.20 Команды сдвига

ПЛК серии FX содержат многочисленные инструкции для сдвига битов в словных операндах или слов в массивах словных операндов. Эти инструкции полезны для неарифметической обработки данных и организации данных.

SFTL – Побитовый сдвиг влево
SFTR – Побитовый сдвиг вправо

Инструкции побитового сдвига влево (**SFTL**) и побитового сдвига вправо (**SFTR**) сдвигают биты в слове (или нескольких словах) "влево" или "вправо", заменяя сдвинутые биты на биты из источника. Инструкции включают 4 параметра: расположение данных, сдвигаемых в регистр (ы), первый адрес сдвигаемых данных, длину сдвигаемых данных и число битов, на которое данные сдвигаются влево или вправо.

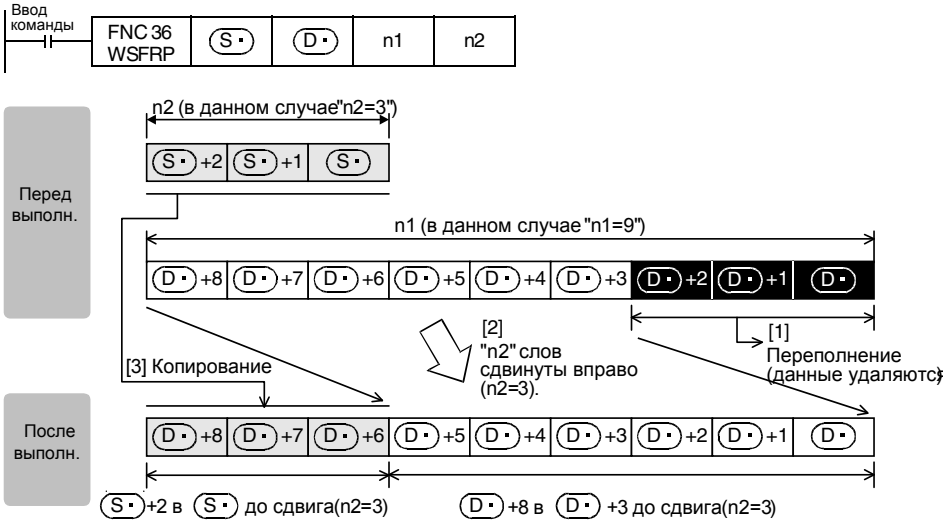
Для "n1" битов (сдвиг регистра влево), начиная с (D·), "n2" битов сдвинуты вправо ([1] и [2] показаны ниже).
После сдвига "n2" битов, начиная с (S·), сдвигаются в "n2" битов, начиная с (D·) +n1-n2 (см. [3] ниже).



WSFL – Пословный сдвиг влево
WSFR – Пословный сдвиг вправо

Инструкции пословного сдвига влево (**WSFL**) и пословного сдвига вправо (**WSFR**) сдвигают слова данных таким же образом, как инструкции побитового сдвига. Как и инструкции побитового сдвига, они включают 4 параметра: расположение данных, сдвигаемых в регистр (ы), первый адрес сдвигаемых данных, длину сдвигаемых данных, и число слов, на которое данные сдвигаются влево или вправо.

Для "n1" словных операндов, начиная с (D·), "n2" слов сдвинуты вправо ([1] и [2] показаны ниже).
После сдвига "n2" слов, начиная с (S·) сдвигаются в "n2" слов, начиная с (D·) +n1-n2 (см. [3] ниже).



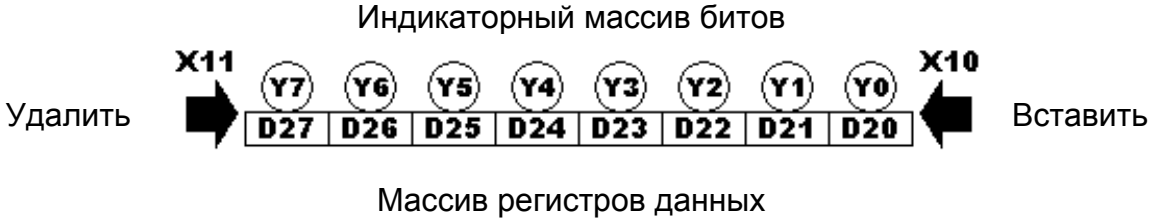
13.21 Упражнение Обработка аналоговых значений

Это упражнение основано на релейной диаграмме из проекта 6: Управление аналоговыми значениями.

Сначала отредактируем код релейной диаграммы из проекта 6 так, чтобы инструкции FROM и TO выполнялись каждый цикл (пока ПЛК находится в режиме RUN). Затем, используя команды побитового и пословного сдвига, соберем в D20-D27 кадр последовательной передачи данных, включающий отрегулированное значение аналогового выхода (D11), который можно переслать на другой контроллер или ПК.

Кадр, или массив регистров данных , должен состоять из восьми слов данных (восьми регистров данных). Каждое значение аналогового выхода должно быть вставлено в массив, начиная с D20. Запустите вставку каждого элемента данных с помощью X010. В то же время каждый регистр данных (D20-D27) в массиве имеет соответствующий битовый операнд (Y000-Y007) для указания, что в регистр данных были загружены правильные данные. Это также должно запускаться с помощью X010 с включением, начиная с Y000. Чтобы удалить значение регистра данных и его соответствующий бит, просто вставьте данные из пустого регистра данных (например, D5) начиная с D27, и также отрегулируйте массив битов, начиная с Y007. Запустите удаление данных с помощью X011. Когда весь индикаторный массив битов заполнен, кадр последовательной передачи данных готов к пересылке. Однако, для этой иллюстративной программы, просто сбросьте весь массив битов и массив регистров данных, когда они заполнятся.

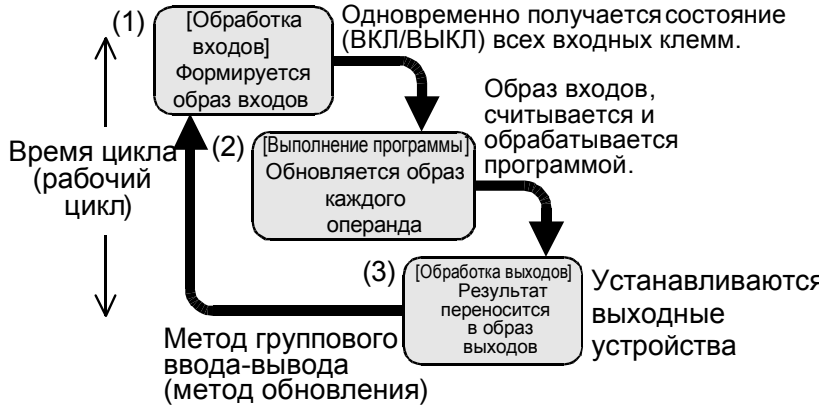
Ниже приведена иллюстрация системы.



После того как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 13.21; Упражнение – Обработка аналоговых значений" на интерфейсе GOT. Если код релейных диаграмм введен правильно, моделирование должно пройти без проблем. Все опции управления аналоговыми значениями остаются из проекта 6 в разделе 13.15 кроме индикаторов FROM и TO . Используйте X010-X011 на учебном стенде, чтобы вставлять и удалять данные из массива индикаторных битов и регистров данных. Этот экран поможет студентам понять инструкции сдвига, визуализовав приложение, описанное в данном упражнении, для упрощения программирования и отладки.

13.22 Управление процессом выполнения программы

По умолчанию ЦП ПЛК обрабатывает релейную диаграмму сверху вниз и слева направо. В начале каждого цикла программы все значения на физических входах в системе считываются в отображение в памяти. Затем программа обрабатывается, внося модификации в отображение в памяти. Когда достигнута команда END, выходы обновляются на базе отображения в памяти. Затем процесс повторяется.



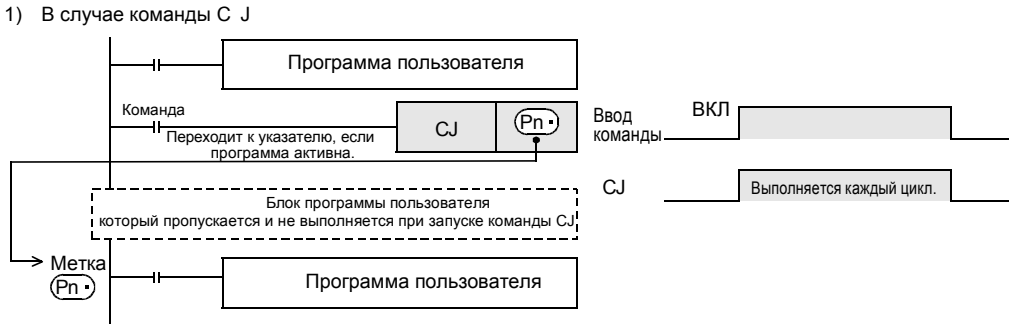
Иногда этот процесс выполнения программы необходимо изменить, в зависимости от приложения. Существует несколько методов управления процессом выполнения программы, включая переходы и вызовы подпрограмм.

Эти инструкции используют упомянутые выше адреса указателей. Чтобы установить адрес указателя, дважды щелкните на левом поле релейной диаграммы (слева от вертикальной шины) и введите адрес ячейки памяти с префиксом P. При обращении к этой ячейке памяти при переходе или вызове подпрограммы он опрашивается по его P-адресу операнда.

ПРИМЕЧАНИЕ Адрес P63 зарезервирован. Он является переходом к инструкции END. Не пишите код для P63 в релейной диаграмме, или произойдет ошибка.

CJ – Условный переход

Инструкция условного перехода (**CJ**) позволяет пропустить (не обрабатывать) блок кода релейной диаграммы в зависимости от его входных условий. Когда входные условия активны и команда выполняется, весь код релейной диаграммы между командой CJ и заданным указателем не выполняется.

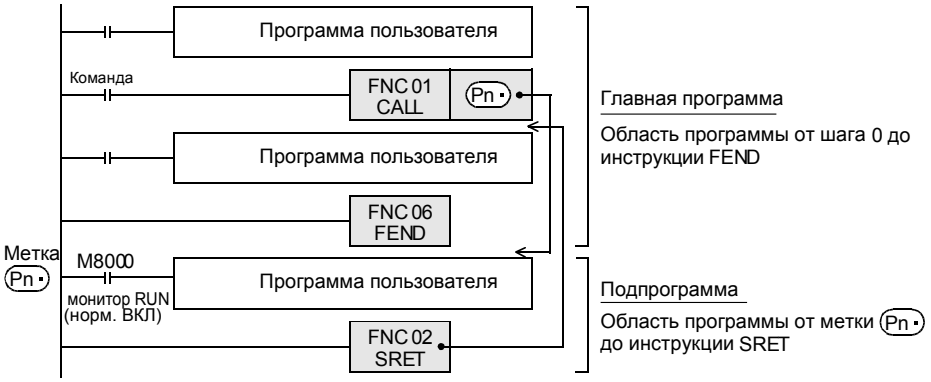


CALL – Вызов подпрограммы
SRET – Конец подпрограммы
FEND – Конец области программы

Команда вызова подпрограммы (**CALL**) позволяет выполнить указанный блок кода релейной диаграммы (подпрограмму) в ходе стандартной обработки, когда входные условия подпрограммы станут активными. В отличие от команды CJ выполнение главной релейной диаграммы возобновляется с точки, из которой была вызвана подпрограмма, после завершения подпрограммы. Удалив условно повторяемый код релейной диаграммы из главной релейной диаграммы (выполняемой каждый цикл) и используя вместо него подпрограммы, можно сократить время цикла, поскольку код релейной диаграммы для подпрограммы обрабатывается только при выполнении команды CALL.

Подпрограммы кодируются в конце главной релейной диаграммы ПЛК. Операции главной релейной диаграммы останавливаются на инструкции окончания основной программы (**FEND**), и код релейной диаграммы для подпрограммы записывается между командами FEND и END. Подпрограмма идентифицируется по адресу указателя в начальной точке и команде конца подпрограммы (**SRET**) в конце подпрограммы. Когда входные условия активны, команд CALL выполняет код релейной диаграммы начиная с указателя данной подпрограммы, пока не будет достигнута команда SRET. По достижении команды SRET выполнение программы возвраща-

ется к главной релейной диаграмме и продолжается с шага после выполнения команды CALL.



Перейдите на экран "Раздел 13.22; Процесс выполнения программы" на интерфейсе GOT. В этом моделировании сравниваются инструкции перехода (CJ) и вызова подпрограммы (CALL). Экран поможет студентам визуализировать различие в инструкциях, регулирующих процесс выполнения программы, для облегчения дальнейшего понимания.

ГЛАВА 14 – Диагностические операнды

ПЛК серии FX имеет ряд специализированных маркеров и регистров данных, в которых сохраняется информация, включая коды операционных ошибок ПЛК.
Цели главы: Завершив эту главу, слушатели смогут…

- Идентифицировать наиболее распространенные специальные маркеры и регистры данных, которые помогают в поиске неисправностей.
- Обрабатывать информацию, показанную наиболее распространенными специальными маркерами и регистрами данных.
- Писать небольшие программы, которые могут предоставлять диагностические функции.
- Использовать диагностику с помощью GX Developer для выявления ошибок.

Материалы: Руководство по обучению FX-TRN-KIT-R
Учебный стенд FX-TRN-KIT-R

14.1 Специальные M-маркеры

Специальный маркер M8000 и выше зарезервированы для специальных системных целей. В ПЛК FX1S, FX1N, FX2N и FX2NC к этой области принадлежат M8000-M8255. В FX3U к этой области принадлежат M8000-M8511. Специальные M-маркеры имеют различные системные функции. Полный список специальных M-маркеров можно найти в главе 36 Руководства по программированию FX3U или в меню Справка GX Developer.

Специализированные M-маркеры могут быть полезными как для программирования, так и для отладки. Приведем некоторые из типичных диагностических специальных M-маркеров перечислены ниже.

M8004	Указывает, что обнаружена ошибка
M8006	Разряженная батарея
M8060-M8068	Флаги обнаружения ошибок
M8064	Ошибка параметров
M8065	Синтаксическая ошибка программы
M8066	Ошибка программирования
M8067-M8068	Операционная ошибка (т.е. деление на 0)

Специальные маркеры M8060-M8068 имеют соответствующие специальные регистры данных D8060-D8068, которые содержат номер ошибки. Они будут рассмотрены в следующем разделе.

Следующие специальные M-маркеры не являются диагностическими маркерами, но очень полезны при программировании (некоторые уже использовались в различных учебных проектах):

M8000	Всегда имеет значение "1" (для ПЛК в режиме RUN)
M8001	Всегда имеет значение "0" (для ПЛК в режиме RUN)
M8002	Импульс инициализации (нормально разомкнутый контакт)
M8003	Импульс инициализации (нормально замкнутый контакт)
M8011	10 мс тактовый импульс
M8012	100 мс тактовый импульс
M8013	1 с тактовый импульс
M8014	1 мин тактовый импульс

Перейдите на экран "Раздел 14.1; Специальные M-маркеры" на интерфейсе GOT. Некоторые из указанных выше специальных маркеров контролируются на интерфейсе GOT.

14.2 Специальные D-регистры

Специальные регистры данных D8000 и выше зарезервированы для специальных системных целей. В ПЛК FX1S, FX1N, FX2N, и FX2NC к этой области принадлежат D8000-D8255, в FX3U – D8000-D8511. Специальные регистры D имеют различные системные функции. Полный список специальных D-регистров можно найти в главе 36 Руководства по программированию FX3U или в меню Справка GX Developer.

Некоторые из типичных диагностических специальных D-регистров перечислены ниже.

D8004	Адрес первого активного маркера ошибки (т.е. значение 8064 указывает, что M8064 активен)
D8006	Аварийный уровень разряда батареи
D8010	Текущее время цикла программы
D8061-D8068	Коды ошибок
D8064	Код ошибки параметра
D8065	Код синтаксической ошибки программы
D8066	Код ошибки программирования
D8067-D8068	Код операционной ошибки
D8069	Номер шага ошибки

Следующие специальные D-регистры не являются диагностическими регистрами, но очень полезны при программировании или отладке.

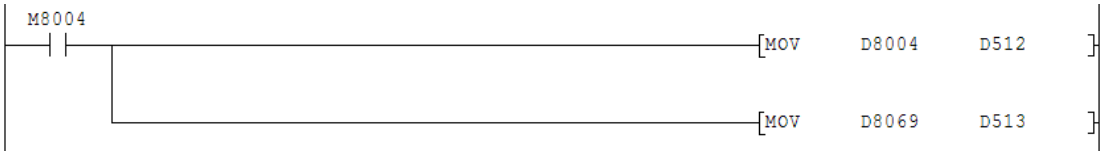
D8001	Тип ПЛК и версия микропрограммного обеспечения
D8005	Напряжение батареи
D8013-D8019	Часы реального времени
D8013	Секунды
D8014	Минуты
D8015	Часы (24-часовой формат)
D8016	День
D8017	Месяц
D8018	Год (2 цифры 00-99)
D8019	День недели (0-6, Воскресенье-Суббота)
D8020	Входной фильтр (0-60 в мс, где 0 = минимально возможный аппаратный фильтр, зависящий от адреса операнда)

Перейдите на экран "Раздел 14.2; Специальные регистры D" на интерфейсе GOT. Некоторые из указанных выше специальных регистров данных контролируются на интерфейсе GOT.

14.3 Удобные цепи поиска неисправностей

Показанная ниже цепь указывает, что напряжение батареи ПЛК является низким. Бит тревоги включает выход, который может активизировать сигнальную лампу или сирену и т.д.

Когда возникает ошибка, схема сохраняет код ошибки и шаг ошибки. Если словные операнды назначения цепи были объявлены как фиксированные в параметрах ПЛК, или если словные операнды принадлежат диапазону D512-D7999, то информация об ошибках будет сохранена даже при переключении ПЛК в режим СТОП или отключении электропитания.



14.4 Использование часов реального времени

Все ПЛК серии FX имеют возможность использования часов реального времени. Начиная с FX2NC для этой возможности требуется кассета памяти с часами реального времени. Данные для часов реального времени хранятся в специальных регистрах данных D8013-D8019, как обсуждалось выше. Имеется множество команд, которые могут использоваться для доступа к данным в часах реального времени; некоторые из них показаны ниже. Более подробное описание этих инструкций см. в главе 21 Руководства по программированию FX3U.

Если регистры D8013-D8019 должны быть модифицированы командами, кроме описанных ниже, часы реального времени должны быть остановлены. Чтобы остановить часы, включите специальный маркер M8015. Когда M8015 снова сбрасывается, данные из D8013 – D8019 записываются во внутренние часы реального времени ПЛК. Если часы не остановлены перед регулировкой, изменения не возымеют эффект.

Если требуется отображение 4 разрядов года, можно написать код релейной диаграммы, чтобы прибавить ‘2000’ к регистру года D8018, изменив индикацию до 4 разрядов, но фактически не модифицируя значение годов в RTC. Это необходимо делать каждый раз, когда ПЛК переключается из режима STOP в режим RUN, поэтому эта процедура должна запускаться маркером M8002.

TRD – Считать время суток и дату

Команда считывания времени (**TRD**) используется для считывания данных часов реального времени ПЛК из специальных регистров данных D8013-D8019 в другие словные операнды в устройстве памяти ПЛК. Команда копирует все 7 регистров в 7 последовательных регистров, следующих за регистром назначения словного операнда.

TWR – Передать время суток и дату в контроллер

Команда передачи времени в контроллер(**TWR**) используется для перезаписи данных часов реального времени ПЛК в специальных регистрах данных D8013-D8019 значениями из других словных операндов в устройстве памяти ПЛК. Команда копирует 7 последовательных регистров, начиная со словного регистра источника, в 7 специальных D-регистров, который формируют данные часов реального времени ПЛК (D8013-D8019).

TCMP – Сравнение данных часов

Команда сравнения времени (**TCMP**) сравнивает три произвольных словных операнда источника, соответствующих часам, минутам, и секундам, с данными трех последовательных словных операндов для часов, минут и

секунд, соответственно (учтите, что этот порядок противоположен регистрам RTC D8013-D8015). Результаты записываются в группу 3-битовых операндов, которые могут быть Y-выходами, M-маркерами или S-маркерами. 3 бита результата представляют ситуации, когда последовательные данные словного операнда источника меньше чем, равны, или больше чем данные словного операнда произвольного источника, соответственно.

TZCP – Сравнение данных часов с диапазоном

Команда сравнения данных часов с диапазоном (**TZCP**) сравнивает один набор из трех последовательных словных операндов источника с двумя наборами из трех последовательных словных операндов, которые содержат верхний и нижний пределы для сравниваемых данных часов. Результаты записываются в группу 3-битовых операндов, которые могут быть Y-выходами, M-маркерами или S-маркерами. 3 бита результата представляют ситуации, когда набор последовательных данных словного операнда источника меньше чем, равен, или больше чем диапазон данных часов, соответственно.

TADD – Сложение данных часов
TSUB – Вычитание данных часов

Инструкции сложения и вычитания времени (**TADD** и **TSUB**) позволяют складывать или вычитать два набора из трех последовательных словных операндов источника данных времени и записывать результат в третий набор 3-х последовательных словных операндов назначения. Эти словные операнды содержат данные часов, минут, и секунд, соответственно, как и все прочие инструкции обработки времени.

XCH – Обмен данными

Команда обмена данными (**XCH**) не связана непосредственно с инструкциями обработки времени, но может быть полезна при их использовании. Команда XCH обменивает данные между двумя 16-битовыми словными операндами; при этом значения каждого словного операнда обменивается и оба словных операнда перезаписываются. 32-битовая версия этой команды **DXCH** выполняет ту же функцию с 32-битовыми словными операндами.

14.5 Упражнение Декретное время

ПЛК не имеет какой-либо встроенной опции для учета декретного (летнего) времени. Если необходимо переходить на летнее время, следует написать код релейной диаграммы, управляющий часами ПЛК. Ниже показан американский стандарт для декретного времени.

> В первое воскресенье апреля в 2 часа до полудня перевести часы на один час вперед
> В последнее воскресенье октября в 2 часа до полудня перевести часы на один час назад.

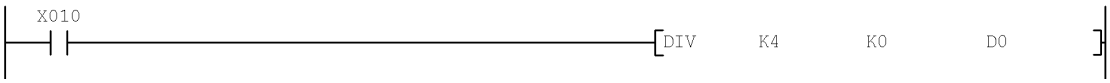
Напишите релейную диаграмму для ПЛК, переводящую часы реального времени вперед весной и назад осенью. Будьте внимательны, часы необходимо перевести назад только один раз!

После того, как релейная диаграмма была введена в GX Developer и записана в ПЛК на учебном стенде, перейдите на экран "Раздел 14.5; Упражнение – Декретное время" на интерфейсе GOT. Существует много способов выполнить это упражнение, некоторые из них нелегко имитировать.

Предусмотренный экран GOT позволяет студентам вручную остановить часы реального времени, попеременно используя специальный маркер M8015, отрегулировать соответствующие регистры данных часов, и затем снова запустить часы реального времени. Учтите, что D8019, представляющий день недели, можно не регулировать, так как часы реального времени вычисляет день недели в зависимости от имеющихся прочих данных часов. Также учтите, что любые введенные неправильные дата и время будут автоматически отвергнуты, когда M8015 возвратится в состояние "0". Этот экран поможет студентам отлаживать их релейные диаграммы для приложения, описанного в упражнении, а также демонстрирует один от способов использования GOT как удобного и полезного инструмента отладки.

14.6 Диагностика с помощью GX Developer

Создайте новую программу и введите звено, как показано ниже:

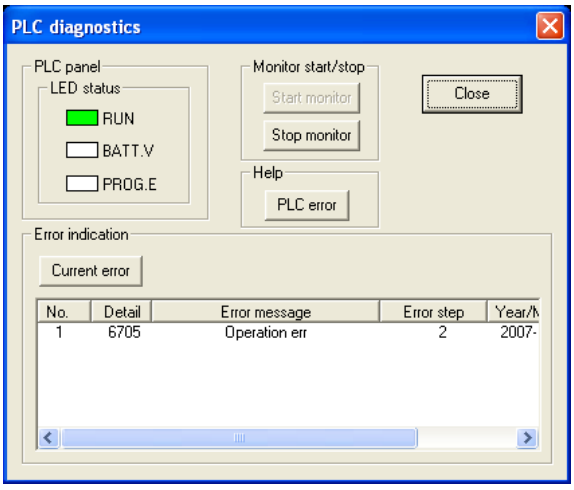


Исследуем приведенное выше звено. Что случится, когда включится X010? Фиксированная десятичная константа 4 будет разделена на 0, и результат сохранится в D0-D1. Эта операция является недопустимой, поскольку в результате получается бесконечное число.

Запишите программу в ПЛК на учебном стенде и включите X010.

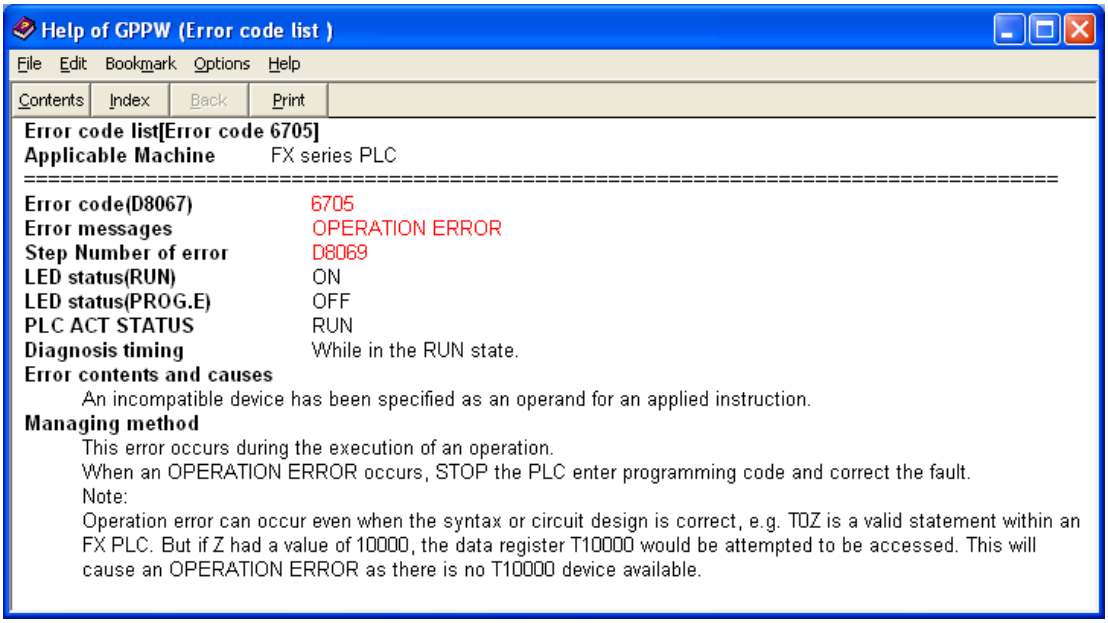
Для поиска ошибки, выполните следующие шаги:

1. Перейдите в меню Diagnostics.
2. Выберите PLC Diagnostics. Появляется окно, аналогичное показанному ниже:



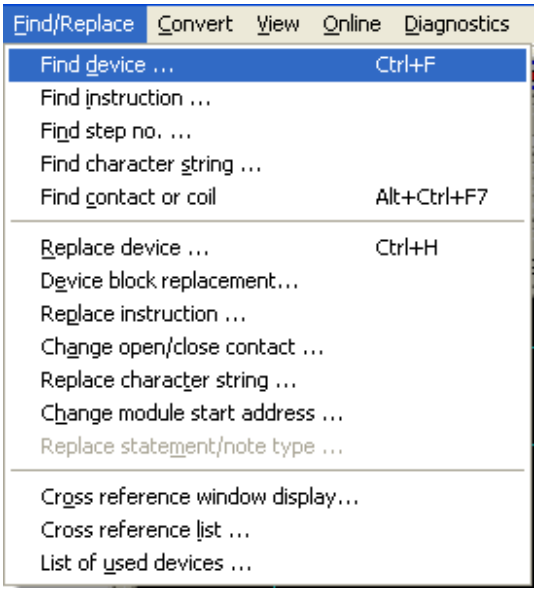
Номер ошибки 6705 указывает на ошибку, которая появляется в специальном регистре данных D8067. На шаге 2 возникает ошибка и сохранятся в D8069. Сообщение об операционной ошибке указывает на код ошибки 8067, который хранится в D8004.

3. Выделите ошибку, щелкнув на ней
4. Нажмите "PLC Error" в разделе окна Help.
5. Должен появиться следующий экран:



Этот экран объясняет возможные причины состояния ошибки и возможные решения. Хотя деление на 0 не задано здесь явно, оно может быть инициировано обращением к "несовместимому операнду" (т.е. K0 как делителю).

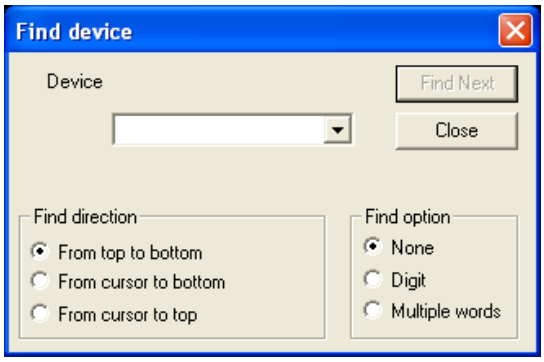
14.7 Меню Find/Replace (Найти/Заменить)



GX Developer предоставляет несколько различных методов для локализации операндов или инструкций. Меню Find/Replace включает разнообразные инструменты для поиска или модификации проектов GX Developer. Функции поиска будут искать в программе все включения операндов, команд, или строк. Функции замены позволяют заменять операнды, инструкции, открытые/закрытые контакты, и другие опции. Функции перекрестных ссылок "Cross reference" и использованных операндов "used devices" покажут все вхождения указанного операнда в программе, или все операнды, которые используются в программе.

Справа показан список опций в меню Find/Replace.

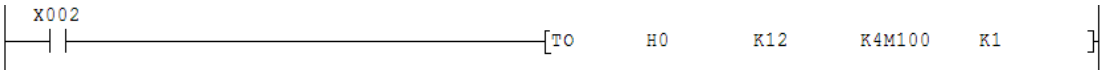
Find Device (Найти операнд) позволяет находить указанный адрес операнда независимо от команды. Программа будет последовательно искать каждое вхождение операнда во всей программе . В этом окне имеется несколько опций. Опции направления поиска "Find direction" в нижней левой части окна позволяют задать направление и зону поиска.



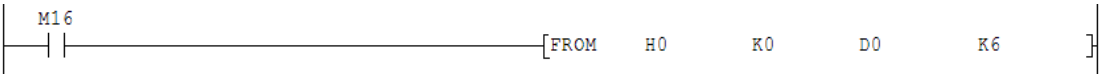
В нижней правой части окна, имеется также блок опций поиска "Find option" ; он описан ниже.

None (Нет) заставляет функцию поиска операнда специально искать введенный адрес операнда.

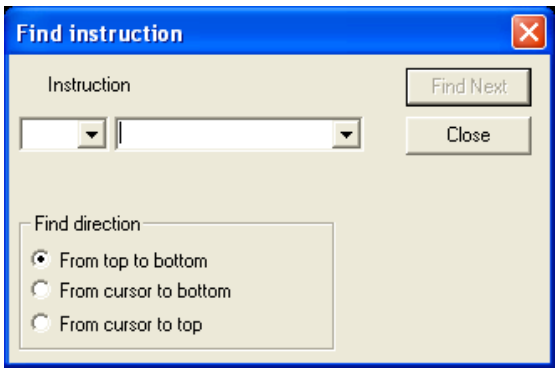
Digit (Число) позволяет функции поиска операнда искать адрес операнда в слове битов. Например, поиск маркера M110 с опцией "Digit" обнаружит, что он используется как часть от следующей инструкции TO.



Multiple Words (Несколько слов)расширяет поиск, включая любой словный операнд, который используется инструкциями с операндами, включающими несколькими слов. Поиск регистра данных D5 с опцией "Multiple word" обнаружит, что он используется как часть следующей команды FROM.

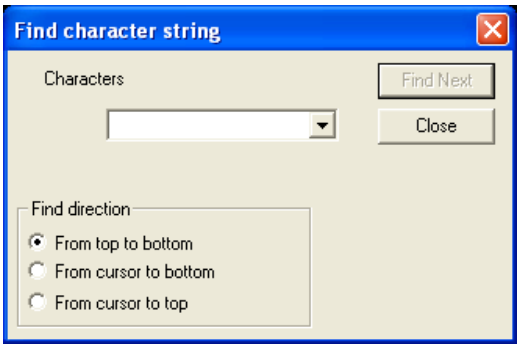


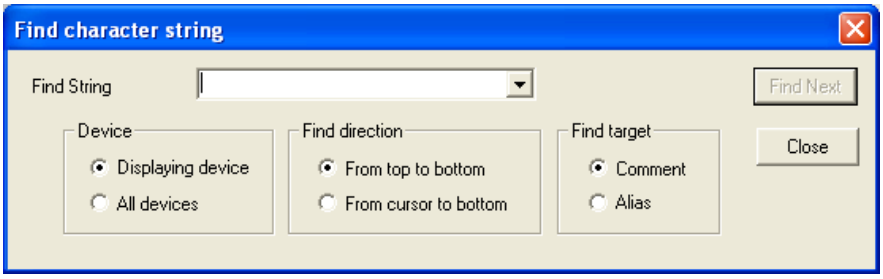
Find Instruction(Найти команду) позволяет последовательно найти все вхождения конкретной команды. Из первого поля со списком выбирается символ релейной диаграммы, а из второго поля со списком – искомая команда. Функции из блока "Find direction" (Направление поиска) уже обсуждались выше.



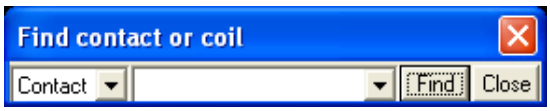
Find Step Number (Найти номер шага) позволяет найти определенный номер шага в релейной диаграмме. Эта опция может быть полезной с обсуждавшейся выше диагностической информацией по ошибкам ПЛК для быстрого обнаружения ошибок в программе. Вводя номера шагов числа непосредственно с цифрового блока стандартной клавиатуры вы также автоматически введете их в функцию "Найти номер шага".

Find Character String (Найти строку символов) позволяет найти текстовые строки в проекте GX Developer. Если активным окном является окно релейной диаграммы, функция будет искать текстовые строки в релейной диаграмме. Если активным окном является список комментариев, поиск будет производиться в комментариях. Другие блоки опций применяются при поиске в комментариях ("Device" (Операнд), "Find direction" (Направление поиска), и "Find target" (Цель поиска), позволяя изменять критерии и методы поиска.

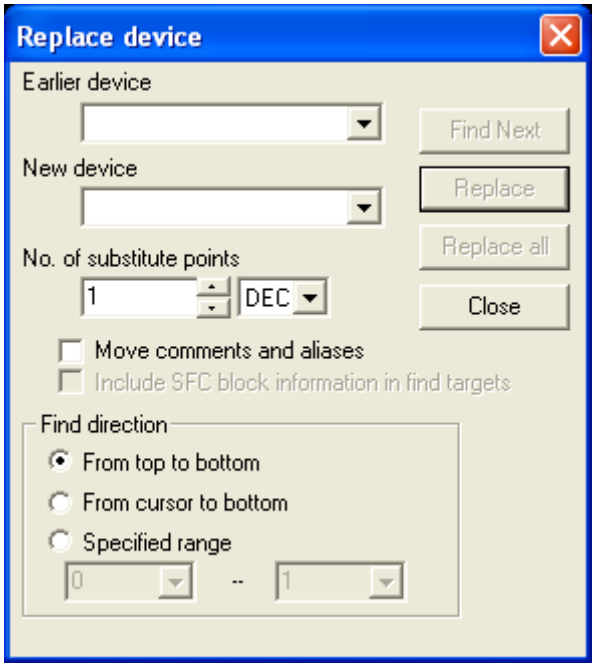




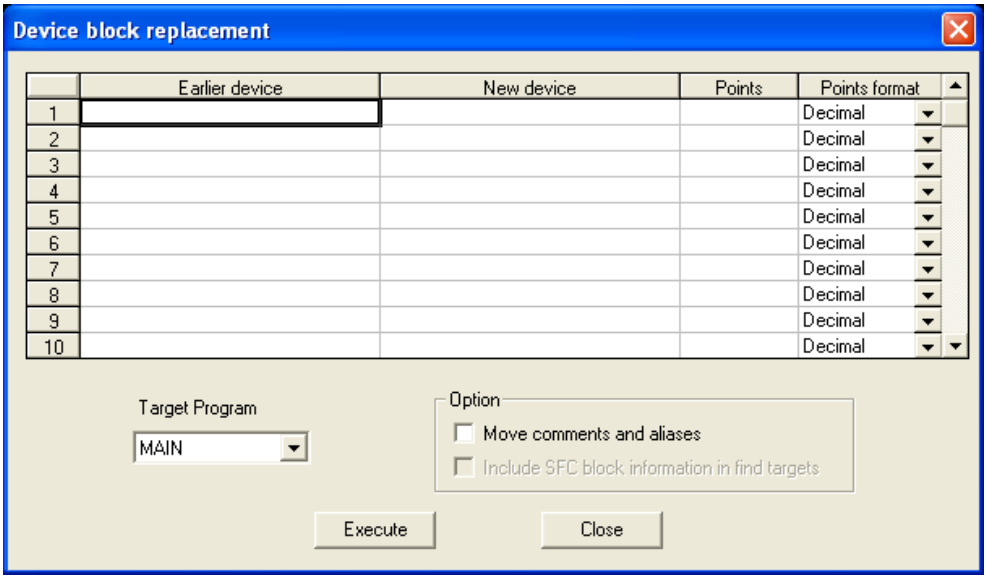
Find Contact or Coil (Найти контакт или флаг) найдет все вхождения в релейной диаграмме, где указанный адрес операнда используется как контакт или катушка. В первом поле со списком выберите Contact (Контакт) или Coil (Катушка), затем в правом поле со списком введите адрес битового операнда. Это полезно, когда адрес битового операнда был запрограммирован много раз как контакт в релейной диаграмме, но только в одном месте как катушка.



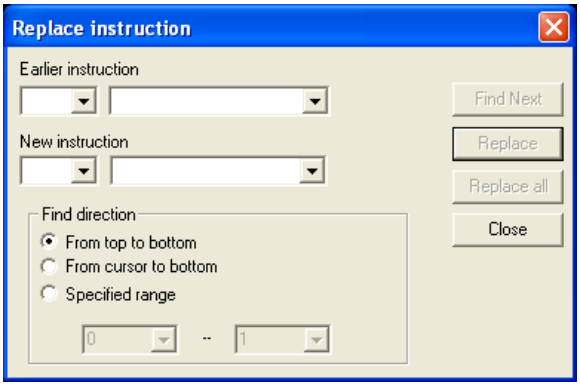
Replace Device (Заменить операнд)позволяет найти и заменить адрес операнда или область адресов операндов в релейной диаграмме. Адрес отдельного операнда может быть указанный для "Earlier device" (Исходного операнда) и "New device" (Нового операнда). Опция "No. of substitute points" (Количество точек замены) задает, какое количество последовательных адресов операндов следует заменить. Флажок для "Move comments and aliases" (Передать комментарии и псевдонимы) позволяет присвоить адресу заменяемого операнда комментарии и псевдонимы, присвоенные адресу заменяющего операнда. Функции из блока "Find direction" (Направление поиска) уже обсуждались выше.



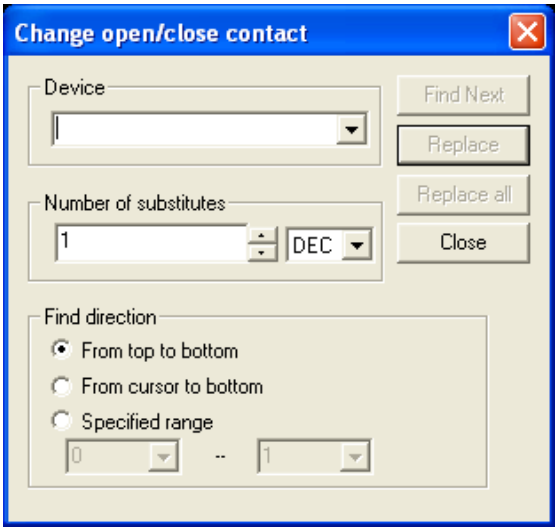
Device Block Replacement (Заменить блок операндов) аналогична функции замены операнда Replace Device, но позволяет одновременно заменять диапазоны адресов операндов. Кроме того, можно одновременно заменять группы таймеров, счетчиков, входов, и регистров данных.



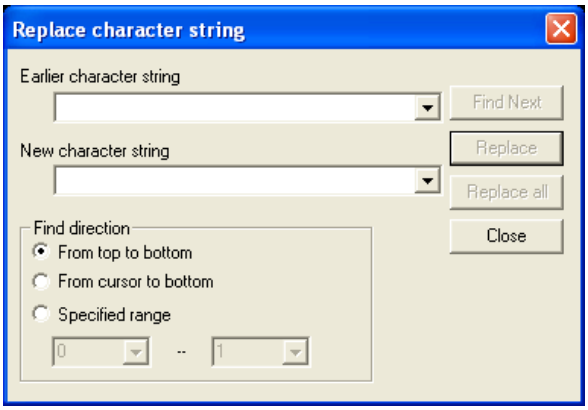
Replace Instruction (Заменить инструкцию) позволяет заменять один тип команд на другой. Поле "Earlier instruction" (Исходная команда) содержит тип заменяемой команды, а "New instruction" (Новая команда) задает тип заменяющей команды. Это может быть полезным при замене всех вхождений INC на INCP в релейной диаграмме.

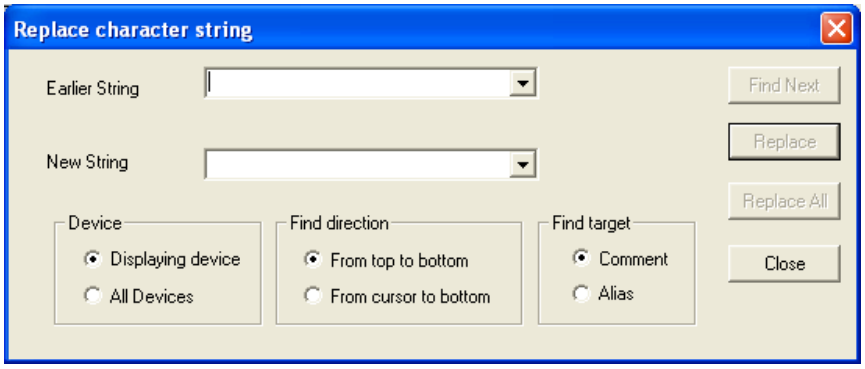


Change Open/Close Contact (Заменить разомкнутый контакт на замкнутый) позволяет заменить все вхождения разомкнутого контакта на замкнутый для отдельного адреса битового операнда, и наоборот. Возможно одновременно инвертировать состояния всех вхождений адреса операнда. Это полезно, если тип входа, предусмотренный для станка, изменяется или не соответствует коду релейной диаграммы, например, когда программа, которая использует нормально замкнутые сигналы останова, используется со станком с нормально разомкнутыми выключателями.

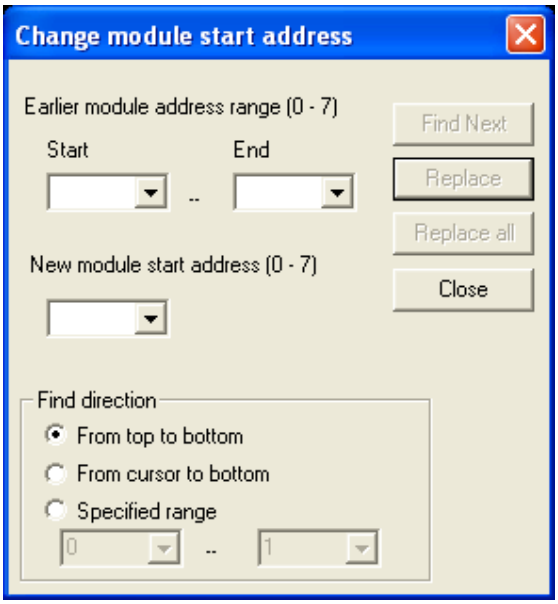


Replace Character String (Заменить строку символов) позволяет заменять текстовые строки. Эта функция аналогична обсуждавшейся выше функции "Find character string". Функция ищет вхождения исходной строки символов "Earlier character string" в релейной диаграмме, списке комментариев, или списке операндов, и заменяет их на новую строку символов "New character string".

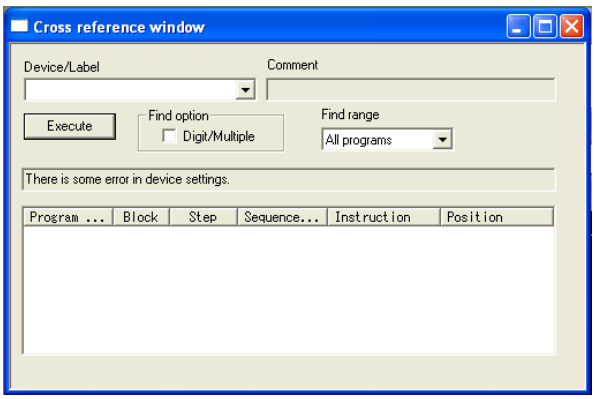




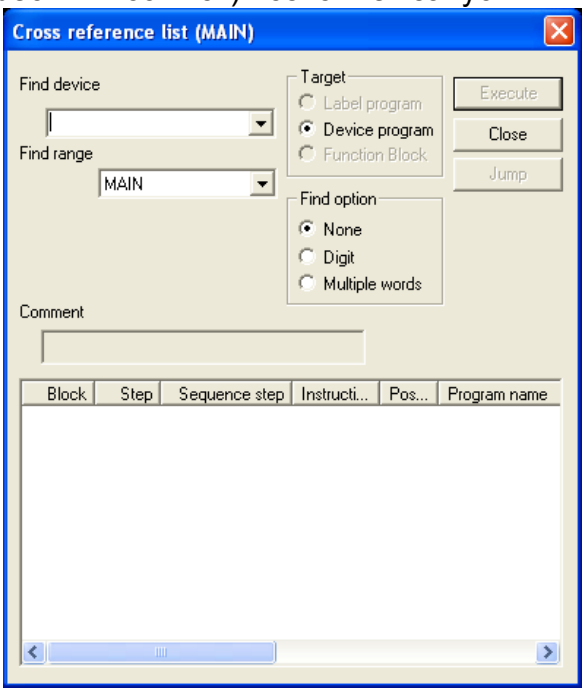
Change Module Start Address (Заменить начальный адрес модуля) позволяет заменять адрес специального функционального модуля SFM. Функция ищет в программе и заменяет все вхождения адреса модуля SFM в инструкциях типа TO или FROM. Задав различные значения в полях "Start" (Начальный) и "End" (Конечный) области адресов, можно сразу изменить код релейной диаграммы для области адресов SFM. Введите адреса модуля SFM "Start" и "End", и новый стартовый адрес модуля SFM. Инструкции, использующие старую область адресов модуля SFM, будут модифицированы с использованием области адресов модуля SFM того же размера, начиная с "New module start address" (Нового начального адреса модуля), поочередно или все вместе.



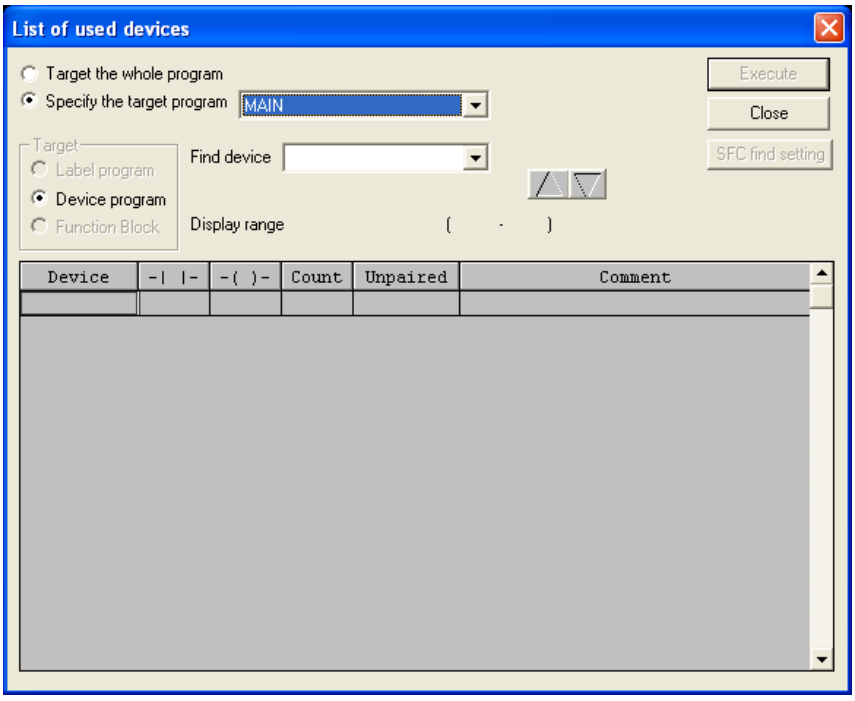
Cross Reference Window (Окно перекрестных ссылок) позволяет запустить многооконную версию списка перекрестных ссылок. Этот многооконный экран можно отображать в плиточном и каскадном режимах, минимизировать или перемещать относительно других окон на рабочем пространстве. Его не нужно закрывать для доступа к другим окнам.



Cross Reference List (Список перекрестных ссылок) позволяет запустить всплывающую версию списка перекрестных ссылок. Это всплывающее окно останется поверх всех других окон; его необходимо закрыть для доступа к другим окнам.



List of Used Devices (Список использованных операндов позволяет показать список операндов, используемых в релейной диаграмме. Список можно конфигурировать, чтобы показывать описание для одной программы (в серии FX поддерживается только одна программа на ПЛК) или всех программ в ПЛК (для ПЛК Q-серии). Адрес данных вводится в окно Find device (Найти операнд); нажав кнопку "Execute", вы получите список, начиная с указанного адреса операнда. Имеются две колонки, в которых будет показана звездочка (*), если соответствующий адрес данных использовался в качестве операнда входа или выхода. Входы включают данные источника для прикладных команд, а выходы включают операнды назначения прикладных команд. "ERR" появляется в колонке "Unpaired", если адрес операнда используется как вход или выход, но не и то, и другое. В зависимости от адреса операнда и его использования в программе, это не обязательно является ошибкой.



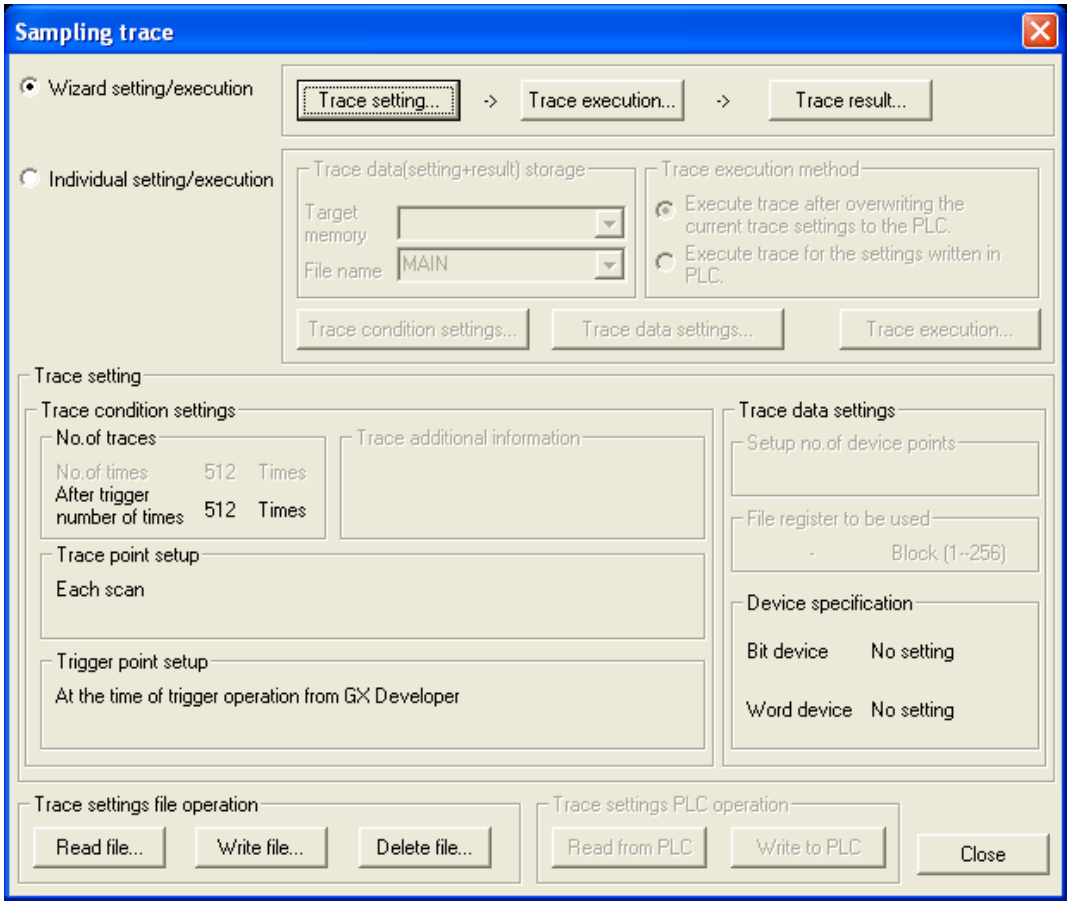
14.8 Трассировка данных

GX Developer включает инструмент отладки, называемый "Trace" (Трассировка). Этот инструмент позволяет программисту изображать состояние значений в ПЛК через какое-то время. Инструмент выполняется в ПЛК, так что он не ограничен способом связи между ПК и ПЛК.

ПЛК серии FX поддерживает трассировку до 512 образцов данных. Образцы могут быть основаны на базе циклов ПЛК или на регулируемом интервале от 10 мс до 2000 мс, с шагом 10 мс. Образцы могут запускаться программным обеспечением или с помощью значения битового или словного операнда в ПЛК. Функция трассировки данных в ПЛК серии FX может регистрировать до 10 битовых операндов и 3 словных операнда одновременно.

GX Developer включает встроенный мастер, помогающий в конфигурировании прослеживаемых данных. Настройки можно также задавать вручную. Задав настройки трассировки, их можно сохранить в файл на ПЛК, который затем может быть считан из ПЛК. Результаты трассировки можно выгрузить после завершения трассировки, а также удалить с ПЛК.

После того, как данные завершенной трассировки выгружены из ПЛК, их можно просмотреть в программе GX Developer или вывести в .CSV файл, который может быть просмотрен или представлен графически с помощью Microsoft Excel.



ГЛАВА 15 – Документация и распечатка

В сравнительно простых программах, наподобие использованных в этом курсе обучения, и написанных самим программистом, довольно просто искать ошибки при возникновении каких-либо проблем. Однако, представьте себе программу, состоящую из 4000 шагов, которую написал 2 года тому назад сотрудник, уволившийся из компании. Без документации может оказаться почти невозможным устранить проблемы в этой программе. Вот почему документирование является очень важным этапом в создании программы.

Цели главы: Завершив эту главу, слушатели смогут…

- Описать 4 типа документации.
- Добавить документацию к программе.
- Описать различные опции для распечатки программы

Материалы: Руководство по обучению FX-TRN-KIT-R

GX Developer предлагает 4 типа документации: комментарии, текстовые вставки, надписи и псевдонимы (иногда называемые "Метками операндов").

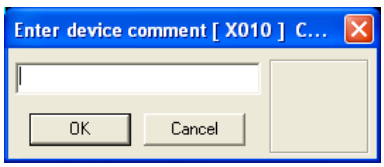
Замечание Как отмечалось выше, можно добавлять и изменять операнды и инструкции, дважды щелкнув на звене или операнде. Эта функция не будет работать, когда выбран инструмент документации. Однако, имеются и другие методы добавления и модификации кода релейной диаграммы.

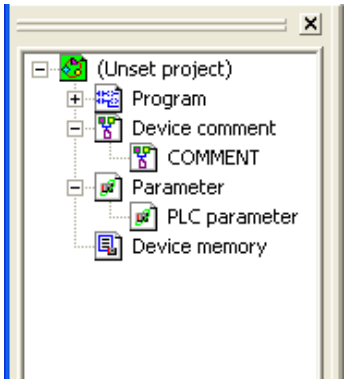
15.1 Комментарии

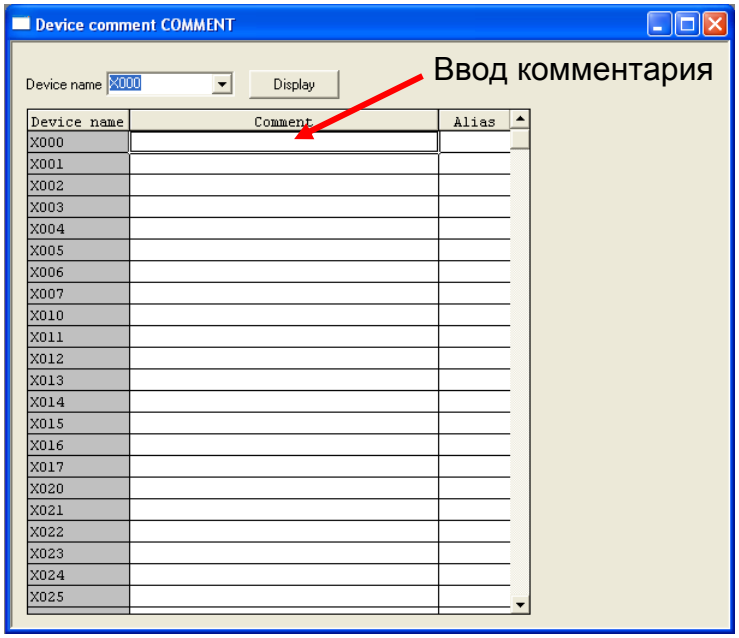
Комментарии присваиваются операнду, чтобы обеспечить название или описание. Типичными комментариями для операнда ввода являются: кнопка "Старт", "Загрузить блок связанных переменных", и т.д. Типичными комментариями для катушки или внутреннего бита являются: "Сигнальная лампа", "Включить двигатель 1", "Процесс разрешен" и т.д. Комментарии могут иметь 4 строки по 8 символов, 3 строки по 5 символов или 2 строки по 8 символов. Это устанавливается в пункте формата комментария "Comment format" в меню View.

По умолчанию тем имеются только опции "4 * 8 символов" и "3 * 5 символов". Чтобы использовать "2 * 8 символов", перейдите в меню Tools и выберите "Options". Щелкните на вкладке "Whole data" и выберите "16" из поля со списком "Common device comment". Когда вы снова выберете пункт "Comment format", там будут опции "2 * 8 символов" и "3 * 5 символов".

Для добавления комментария выберите "Comment" в пункте "Documentation" в меню Edit, или щелкните на горячей клавише  в инструментальной панели. Выбрав инструмент создания комментария, дважды щелкните на любом операнде в релейной диаграмме и введите комментарий для операнда в открывшемся диалоговом окне. Альтернативно, в списке данных проекта "Project Data List" раскройте элемент "Device comment" и дважды щелкните на появившемся элементе "COMMENT" . Этот откроет окно списка комментариев, которые можно скопировать и вставить для использования в любой программе электронных таблиц, например, Microsoft Excel.





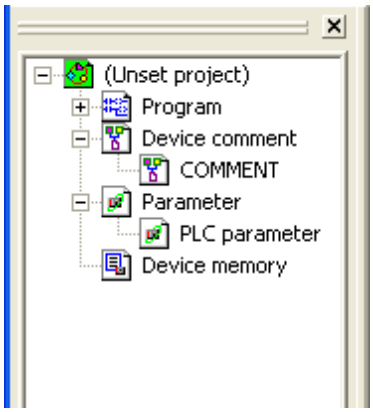


Комментарии являются единственной формой документации, которую можно загрузить в ПЛК. Если из ПЛК выгружается программа, содержащая комментарии, то эти комментарии будут видны и новому программисту. С другой стороны, текстовые вставки и надписи хранятся в файле проекта GX Developer. Если другой программист желает получить к ним доступ, потребуется вывести на экран исходный проект GX Developer.

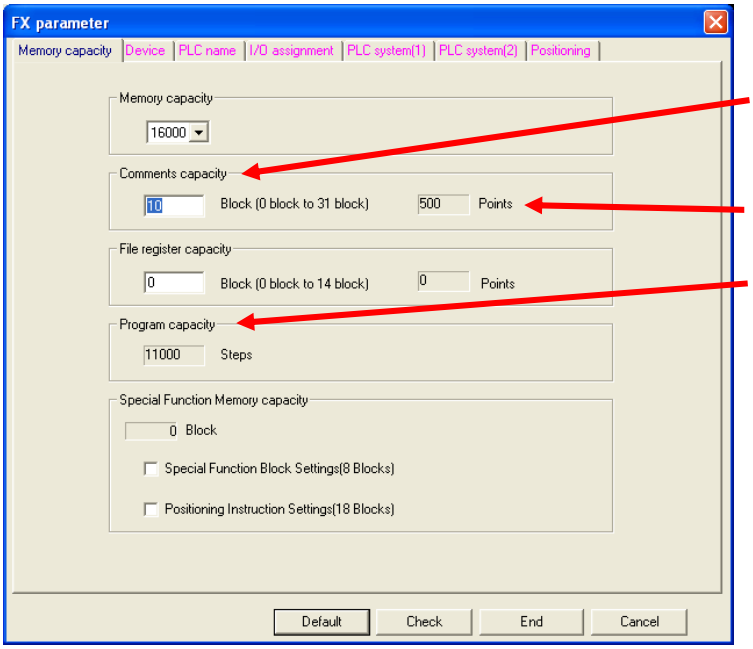
Важное замечание: ПЛК сохранит только первые 16 символов каждого комментария. Остальные символы в комментарии будут отброшены.

Если программист хочет документировать программу, комментируя ее по мере написания, есть возможность открывать диалоговое окно комментария после написания каждой команды. Перейдите в меню Tools, выберите "Options", и выберите опцию "Continues during command write" (Продолжать во время написания команды) в секции "Comment input" (Ввод комментария) в окне "Options". При вводе инструкций в программу GX Developer будет предлагать ввести комментарий для каждого операнда в команде.

Выделение памяти под комментарии



Чтобы загрузить комментарии в ПЛК, для них необходимо выделить область в памяти ПЛК для хранения программы. Она называется "Comments capacity" и задается в параметрах ПЛК на вкладке "Memory capacity". Доступ к параметрам ПЛК в Project Data List можно получить, раскрыв элемент "Parameter" и дважды щелкнув на появившемся элементе "PLC Parameter". Область комментариев назначается как число "блоков", каждый блок увеличивает емкость памяти для комментарией на 50 комментариев к операндам, но уменьшает память для хранения программ на 500 шагов. Помните, что комментарий для X000 учитывается как один операнд, даже если X000 появляется в программе 15 раз.



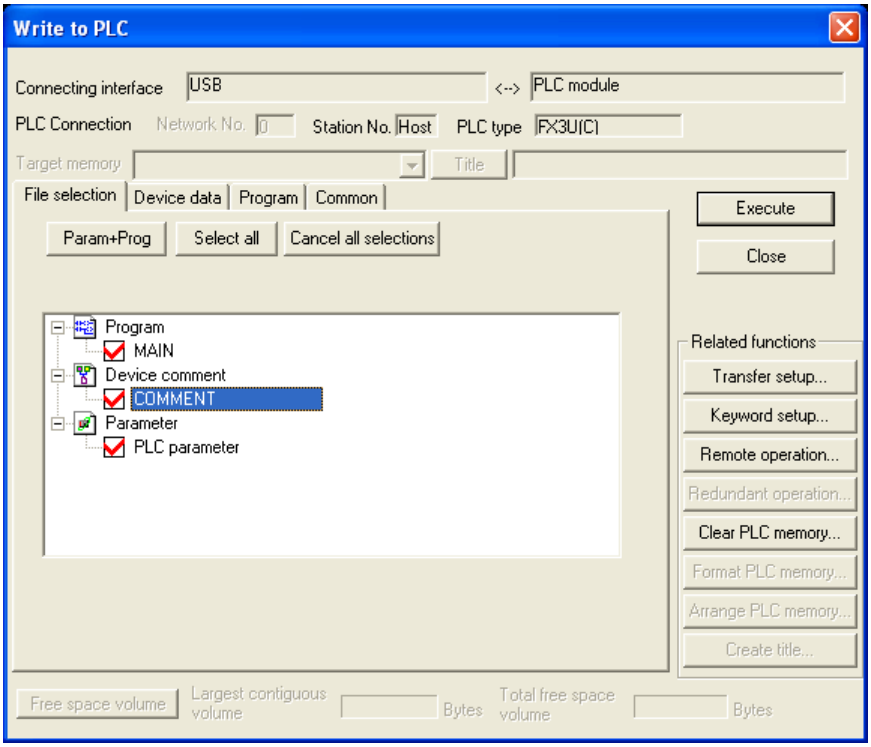
Выделенное число блоков комментариев

Количество комментариев

Оставшаяся память для хранения программы

Выгрузка/загрузка комментариев

Процедура выгрузки и загрузки комментариев несложна и выполняется непосредственно в окне "Program Upload/Download". Перед выгрузкой или загрузкой программы выберите опцию "COMMENT", аналогичное опции "MAIN" для релейной диаграммы, и выберите опцию "PLC parameter" для параметров ПЛК. Затем, выберите "Execute", чтобы выгрузить или загрузить программу, параметры и комментарии.

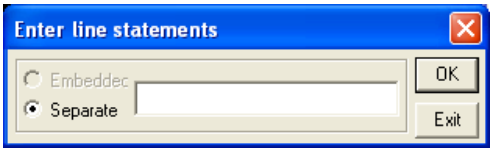


15.2 Текстовые вставки

Текстовые вставки, также известные как комментарии к схеме или звену, используются для описания всего звена. Типичная текстовая вставка может иметь следующий вид: "Этот звено ждет, пока счетчик не достигнет 20, и открывает ворота".

Каждая текстовая вставка может включать до 64 символов. К одному звену можно прикрепить несколько текстовых вставок, чтобы обеспечить подробное описание назначения звена. Текстовые вставки появляются над звеном поперек левой вертикальной шины.

Чтобы добавить текстовые вставки, выберите "Statement" из элемента "Documentation" в меню Edit, или щелкните на горячей клавише  в инструментальной панели. Выбрав инструмент текстовой вставки, дважды щелкните на любой части звена и введите примечание для этого звена в открывшемся диалоговом окне. Или просто нажмите точку с запятой в любой момент, когда звено находится в режиме записи "Write mode" и введите текстовую вставку после точки с запятой. Дважды щелкните на текстовой вставке, чтобы снова ее модифицировать, или щелкните на текстовой вставке и нажмите кнопку "Delete", чтобы ее удалить. Когда к звену добавляется несколько текстовых вставок, новые текстовые вставки появятся под старыми.

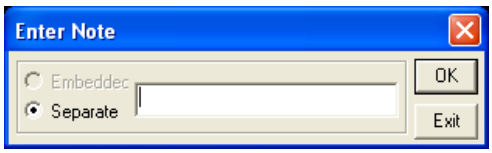


15.3 Надписи

Надписи, которые также называют комментариями к катушкам, появляются над катушками или прикладными командами вывода в правой части звена.

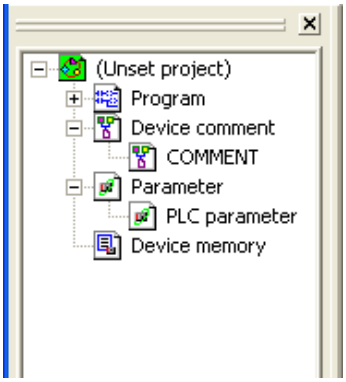
Надписи можно использовать для любого дополнительного описания, но их цель – предоставить информацию о выходе звена, над которым они помещаются. К катушке или команде выхода можно подключить только одну надпись. Однако в одном звене могут использоваться несколько катушек и инструкций выхода, что позволяет подключить много надписей к одному звену. Надписи могут включать до 32 символов.

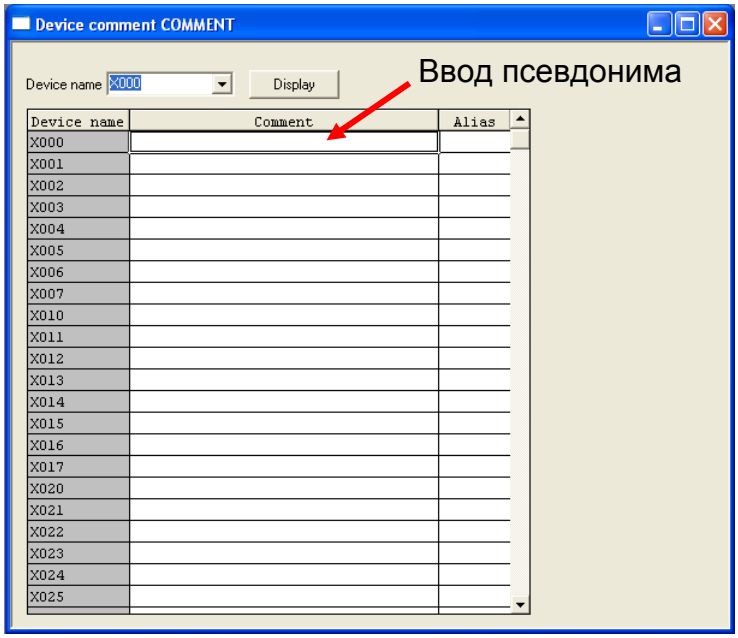
Чтобы добавить надпись, выберите "Note" из элемента "Documentation" в меню Edit, или щелкните на горячей клавише  в инструментальной панели. Выбрав инструмент надписи, дважды щелкните на любой катушке или команде выхода звена и введите надпись для этого выхода в открывшемся диалоговом окне. Или просто напечатайте точку с запятой после катушки или команды выхода, и затем напечатайте надпись в открывшемся диалоговом окне.



15.4 Псевдонимы

Псевдонимы, которые также называются метками операндов, могут быть показаны вместо адреса операнда (т.е. VACPUMP вместо Y000). Псевдоним может включать не больше 8 символов без пробелов. Псевдонимы вводятся в колонку "Alias" описанного выше окна списка комментариев.





Чтобы заменить адреса операндов их псевдонимами, когда активно окно релейной диаграммы, перейдите в меню View и выберите "Alias". Все адреса операндов с псевдонимами будут заменены. Учтите, что есть возможность использовать псевдоним как адрес операнда при вводе новых инструкций; Просто напечатайте апостроф перед вводом псевдонима для необходимого адреса операнда.

15.5 Просмотр документации

Чтобы просмотреть документацию проекта GX Developer, перейдите в меню View и выберите тип просматриваемой документации. Можно одновременно показывать комментарии, текстовые вставки, надписи и псевдонимы.

15.6 Распечатка

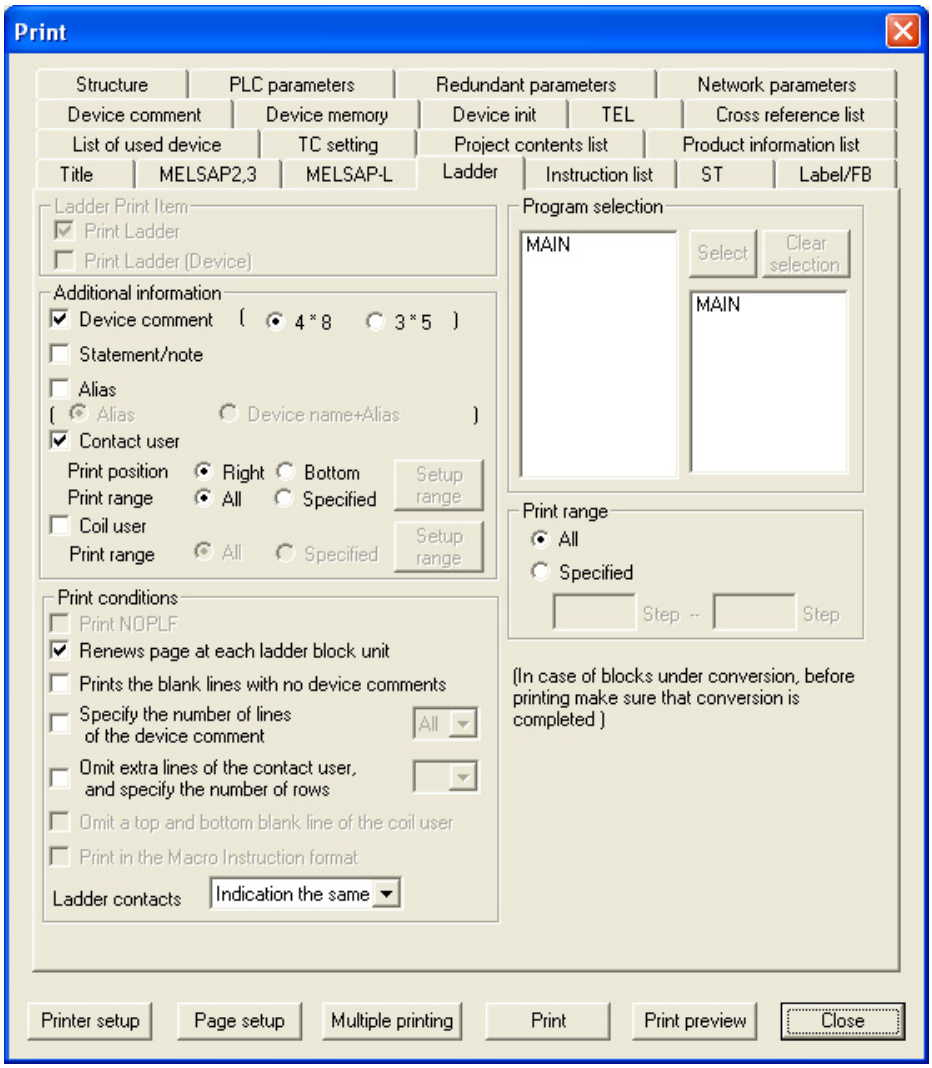
GX Developer включает очень гибкие возможности распечатки. Их можно вызвать, щелкнув на горячей клавише  в инструментальной панели, или выбрав "Print" из меню Project. Программист может выбрать объем и характер распечатки программы по собственному желанию. Перечислим некоторые из имеющихся опций распечатки:

1. Title – Печатает титульный лист. Название может содержать 9 строк по 64 символа. Оно автоматически включает распечатку даты.
2. Ladder – Печатает релейную диаграмму. Отдельные части релейной диаграммы также можно выбрать для распечатки.
3. TC setting – Печатает просто информацию о таймерах и/или счетчиках.
4. Печатает все операнды с комментариями и соответствующие комментарии и/или псевдонимы.
5. List of Used Devices – Печатает описание операндов, которые используются в программе. Также можно выбрать диапазоны операндов для распечатки.

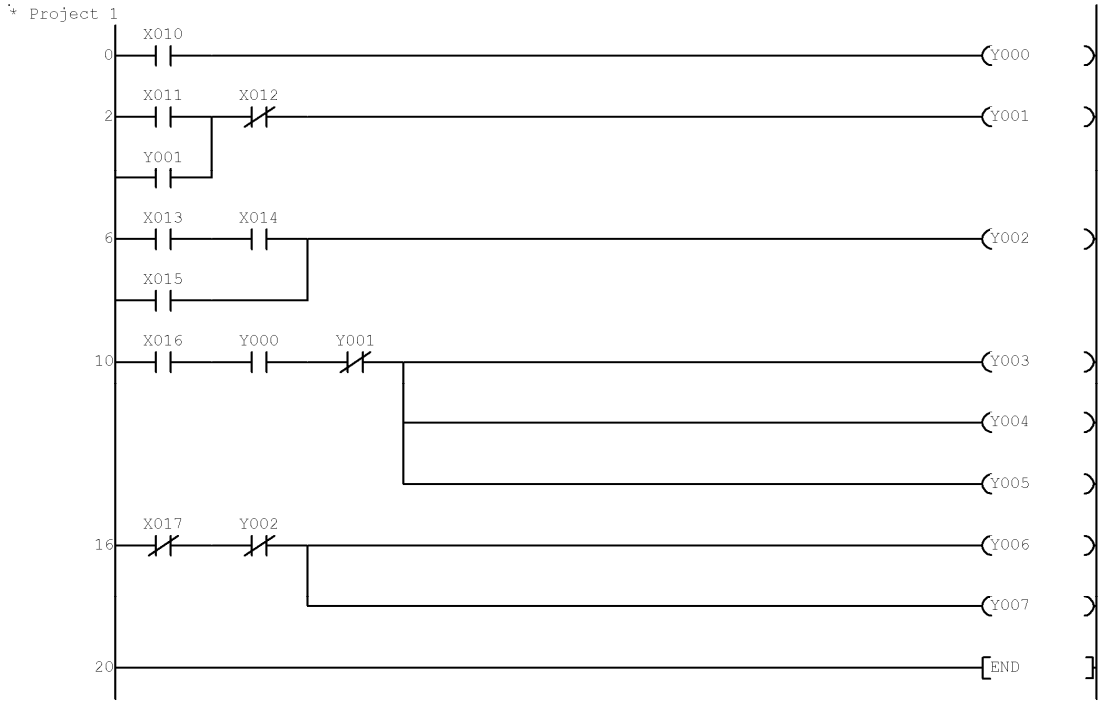
6. Device Memory – Печатает содержимое данных памяти операндов "Device memory"; обычно это считывание всех регистров данных в ПЛК. Ее можно также записать в ПЛК, чтобы инициализировать значения операндов.
7. Parameters – Печатает указанные параметры ПЛК.
8. Cross reference list – Печатает все контакты и/или катушки, используемые в программе. Также можно выбрать диапазоны операндов для распечатки.

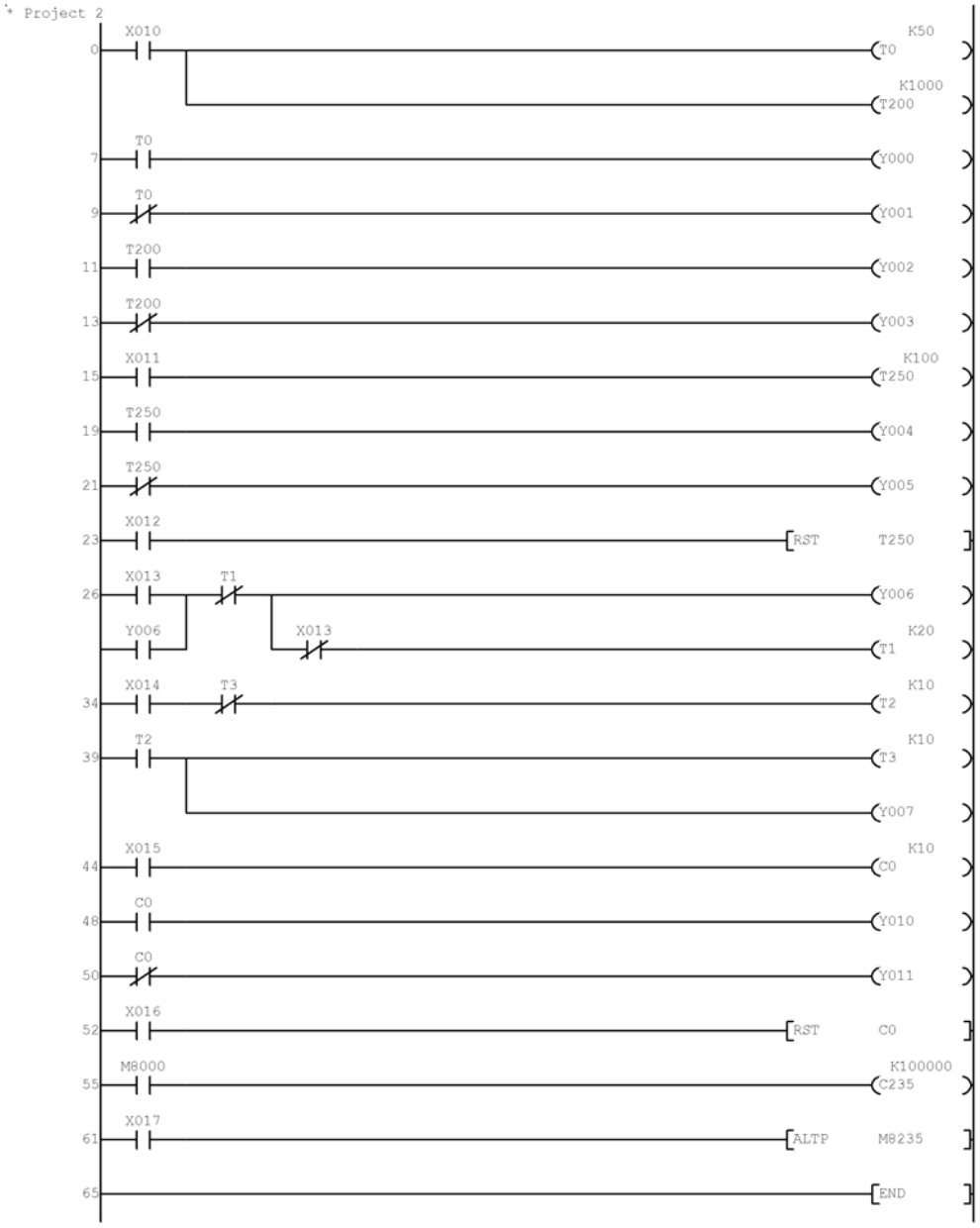
Хотя каждую вкладку можно конфигурировать и печатать отдельно, имеется опция одновременной печати нескольких вкладок данных. Она осуществляется кнопкой "Multiple printing" в нижней части окна Print. Эта функция позволяет пользователям выбрать, какие информационные вкладки печатать, а также задать порядок, в котором они будут напечатаны. Номера страниц будут непрерывными во всей распечатке.

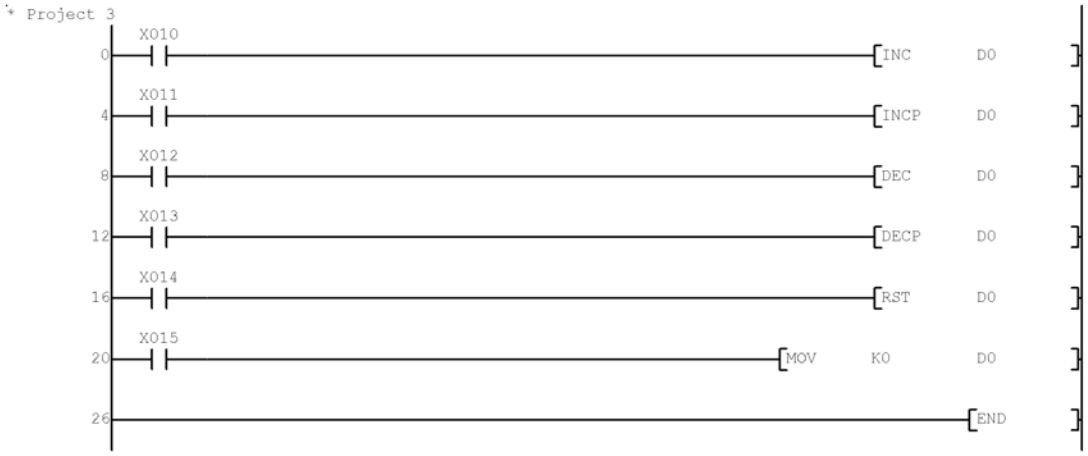
Кнопка параметров станицы "Page setup" позволяет настроить формат страницы. Эта процедура включает поля, верхние и нижние колонтитулы, номера страниц, размер и ориентацию бумаги.

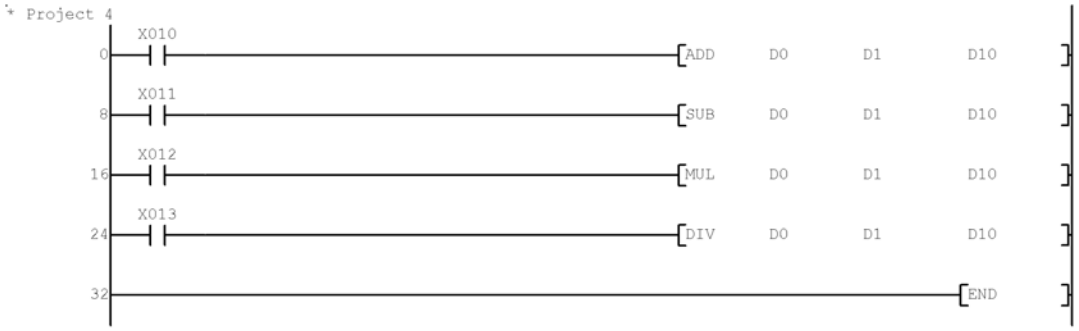


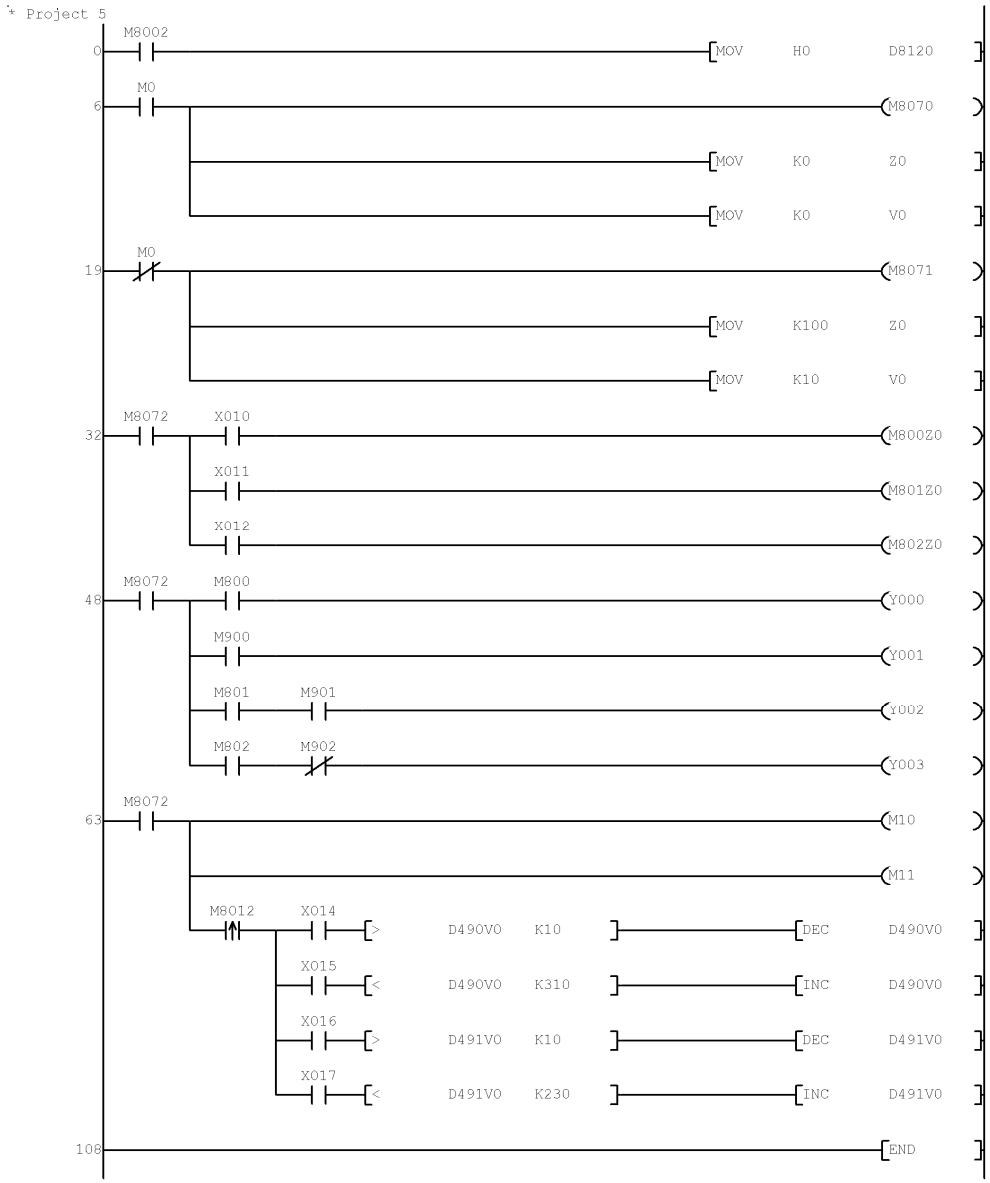
ПРИЛОЖЕНИЕ

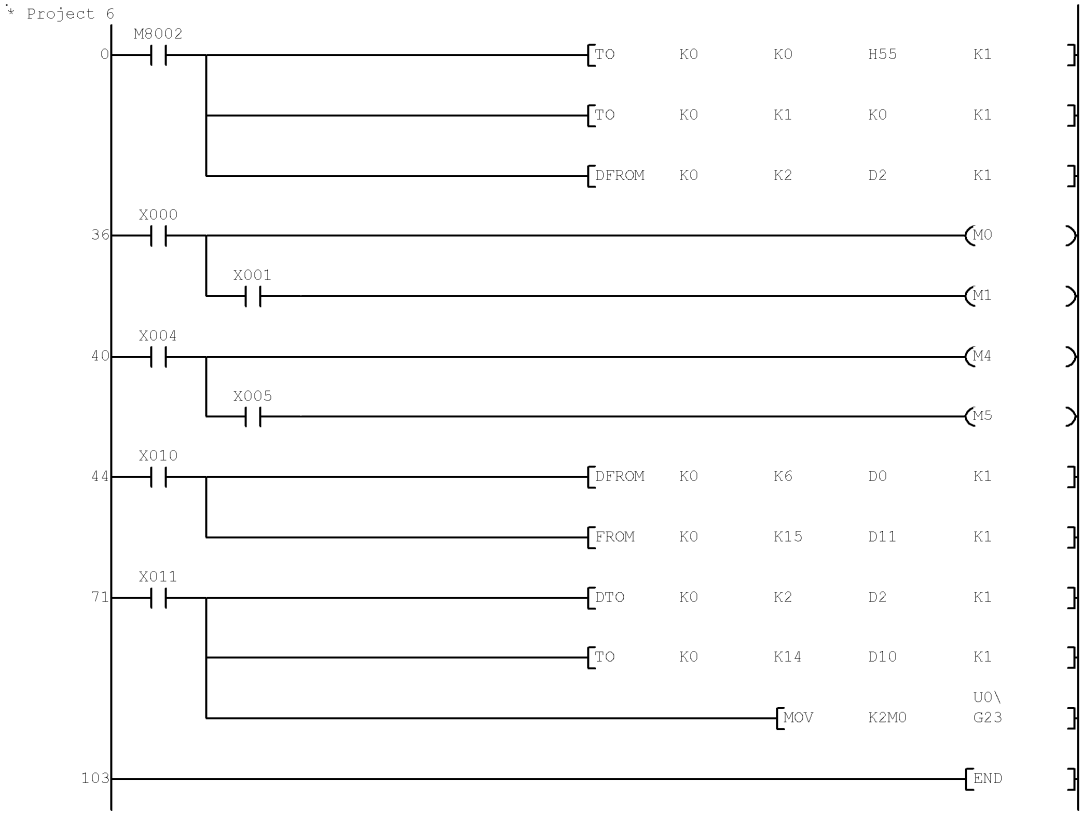

* Project 1
X010
0
Y000
X011
X012
2
Y001
Y001
X013
X014
6
Y002
X015
X016
Y000
Y001
10
Y003
Y004
Y005
X017
Y002
16
Y006
Y007
20
END


* Project 2
X010
K50
T0
K1000
T200
T0
Y000
T0
Y001
T200
Y002
T200
Y003
X011
K100
T250
T250
Y004
T250
Y005
X012
RST
T250
X013
T1
Y006
Y006
X013
K20
T1
X014
T3
K10
T2
T2
K10
T3
Y007
X015
K10
C0
C0
Y010
C0
Y011
X016
RST
C0
M8000
K100000
C235
X017
ALTP
M8235
END


Project 3
X010
0
INC D0
X011
4
INCP D0
X012
8
DEC D0
X013
12
DECP D0
X014
16
RST D0
X015
20
MOV K0 D0
26
END


Project 4
X010
0
ADD D0 D1 D10
X011
8
SUB D0 D1 D10
X012
16
MUL D0 D1 D10
X013
24
DIV D0 D1 D10
32
END


* Project 5
0
M8002
MOV H0 D8120
6
M0
M8070
MOV K0 Z0
MOV K0 V0
19
M0
M8071
MOV K100 Z0
MOV K10 V0
32
M8072
X010
M800Z0
X011
M801Z0
X012
M802Z0
48
M8072
M800
Y000
M900
Y001
M801
M901
Y002
M802
M902
Y003
63
M8072
M10
M11
M8012
X014
> D490V0 K10
DEC D490V0
X015
< D490V0 K310
INC D490V0
X016
> D491V0 K10
DEC D491V0
X017
< D491V0 K230
INC D491V0
108
END


Project 6
M8002
0
TO K0 K0 H55 K1
TO K0 K1 K0 K1
DFROM K0 K2 D2 K1
X000
36
M0
X001
M1
X004
40
M4
X005
M5
X010
44
DFROM K0 K6 D0 K1
FROM K0 K15 D11 K1
X011
71
DTO K0 K2 D2 K1
TO K0 K14 D10 K1
MOV K2M0 U0\G23
103
END

MITSUBISHI ELECTRIC EUROPE B.V. /// РОССИЯ /// Москва /// Космодамианская наб., 52, стр. 5
Тел.: +7 495 721 20 70 /// Факс: +7 495 721 20 71 /// automation@mitsubishielectric.ru /// www.mitsubishi-automation.ru

MITSUBISHI ELECTRIC
FACTORY AUTOMATION

Mitsubishi Electric Europe B.V. /// FA - European Business Group /// Gothaer Straße 8 /// D-40880 Ratingen /// Germany
Tel.: +49(0)2102-4860 /// Fax: +49(0)2102-4861120 /// info@mitsubishi-automation.com /// www.mitsubishi-automation.com

Тех. параметры могут быть изменены /// 06.2008