

GX IEC Developer

Система программирования
и документации

**Руководство
по курсу обучения**

Об этом руководстве

Содержащиеся в этом руководстве тексты, изображения и примеры служат только для разъяснения установки, работы и применения среды программирования GX IEC Developer.

Если у вас возникнут вопросы по программированию и эксплуатации упоминаемых в данном руководстве программируемых логических контроллеров, свяжитесь с вашим дилером или с одним из региональных партнеров по сбыту (см. последнюю страницу обложки). Актуальную информацию и ответы на часто задаваемые вопросы вы можете найти на сайте Mitsubishi www.mitsubishi-automation.ru.

MITSUBISHI ELECTRIC EUROPE B.V. сохраняет за собой право в любое время и без специального уведомления вносить изменения в данное руководство или технические спецификации.

**Руководство по обучению
Программирование в среде GX IEC Developer
Артикул: 211677**

Версия	Изменения / дополнения / исправления
А 12/2008 pdp	Первое издание

Указания по безопасности

Кому адресовано это руководство

Это руководство предназначено исключительно для знающих, имеющих специальное образование специалистов-электриков, которые знакомы со стандартами по безопасности техники автоматизации. Проектирование, подключение, ввод в эксплуатацию, обслуживание и проверка приборов должны выполняться только квалифицированными электриками, имеющими признанное специальное образование, которые знакомы со стандартами и нормативами по безопасности техники автоматизации.

Использование согласно назначению

Программируемые логические контроллеры предназначены только для тех областей применения, которые описаны в этом руководстве. Обращайте внимание на соблюдение всех указанных в руководстве характеристик. Вся продукция разработана, изготовлена, проверена и задокументирована с соблюдением норм безопасности. Любая модификация аппаратуры или программного обеспечения либо несоблюдение предупреждений, содержащихся в этом руководстве или нанесенных на сам прибор, могут привести к серьезным травмам, повреждению оборудования или материальному ущербу. Разрешается использовать только дополнительные или расширительные приборы, рекомендуемые фирмой MITSUBISHI ELECTRIC. Любое иное использование оборудования, выходящее за рамки вышеуказанного, считается использованием не по назначению.

Предписания, относящиеся к безопасности

При проектировании, установке, вводе в эксплуатацию, техническом обслуживании и проверке приборов должны соблюдаться предписания по технике безопасности и охране труда, относящиеся к специфическому случаю применения. Особенно следует обращать внимание на указанные ниже предписания. Этот список не претендует на полноту охвата, однако пользователь несет ответственность за знание и соблюдение соответствующих нормативов.

- Предписания электротехнического союза Германии (VDE)
 - VDE 0100
(Правила возведения силовых электроустановок с номинальным напряжением до 1000 В)
 - VDE 0105
(Эксплуатация силовых электроустановок)
 - VDE 0113
(Электроустановки с электронными компонентами оборудования)
 - VDE 0160
(Оборудование силовых электроустановок и электрических компонентов оборудования)
 - VDE 0550/0551
(Правила установки трансформаторов)
 - VDE 0700
(Безопасность электрических приборов, предназначенных для домашнего пользования и подобных целей)
 - VDE 0860
(Правила безопасности для электронных приборов и их принадлежностей, работающих от сети и предназначенных для домашнего пользования и подобных целей)

- Правила противопожарной безопасности
- Правила предотвращения несчастных случаев
 - VBG No. 4 (Электроустановки и электрические компоненты оборудования)

Предупреждения об опасности в данном руководстве

В данном руководстве специальные указания, имеющие значение для безопасной эксплуатации устройств, отмечены следующим образом:



ОПАСНО:

Предупреждения об опасности для здоровья и возможности травмирования персонала. Означает, что неприятие соответствующих мер предосторожности опасно для жизни и здоровья пользователя.



ВНИМАНИЕ:

Предупреждения об опасности для сохранности оборудования и имущества. Означает предупреждение по возможному повреждению применяемых устройств или имущества, если не придерживаться соответствующих мероприятий по безопасности.

Общие предупреждения об опасностях и профилактические меры безопасности

Нижеследующие предупреждения об опасностях следует рассматривать как общие правила обращения с программируемым контроллером в сочетании с другими приборами. Эти указания должны безусловно соблюдаться при проектировании, монтаже и эксплуатации управляющих устройств.



ВНИМАНИЕ:

- **Соблюдайте предписания по технике безопасности и охране труда, относящиеся к конкретному случаю применения. Перед тем как выполнять монтаж, работать с электропроводкой и открывать блоки, компоненты и устройства, необходимо отключить все источники электропитания.**
- **Блоки, компоненты и приборы должны устанавливаться в защищенных от прикосновения корпусах с соответствующими крышками и защитными устройствами.**
- **Если приборы подключаются к сети постоянной проводкой, в оборудование здания должен быть встроены выключатель для отключения от сети по всем полюсам и предохранитель.**
- **Регулярно проверяйте токоведущие кабели и провода, которыми соединены приборы, на отсутствие дефектов изоляции или мест обрыва. При обнаружении неисправностей в соединениях следует сразу обесточить приборы и отключить их, а затем заменить дефектный кабель.**
- **Перед вводом в эксплуатацию проверьте, совпадает ли допустимый диапазон сетевого напряжения с местным сетевым напряжением.**
- **Для установок с системами приводов позиционирования недостаточно использовать устройства защиты от остаточных токов согласно DIN VDE 0641, часть 1-3, в качестве единственной защиты при косвенных прикосновениях. Для таких установок должны быть приняты дополнительные или иные меры защиты.**
- **Устройства аварийного выключения в соответствии со стандартом EN 60204 / IEC 204 VDE 0113 должны оставаться работоспособными во всех рабочих режимах программируемого контроллера. Деблокировка устройства аварийного выключения не должна вызывать неконтролируемого или неопределенного повторного запуска.**
- **Чтобы обрыв провода или жилы на сигнальной стороне не мог привести к неопределенным состояниям в системе управления, в аппаратуре и программном обеспечении должны быть приняты соответствующие профилактические меры безопасности.**
- **При использовании всех модулей в установке следует всегда строго соблюдать расчетные электрические и физические параметры.**

СОДЕРЖАНИЕ

1	Обзор курса и требования	
1.1	Модульный ПЛК для обучения.....	1-1
2	Аппаратные средства	
2.1	Общее введение в ПЛК	2-1
2.1.1	История и разработка.....	2-1
2.1.2	Базовые технические требования к ПЛК	2-1
2.1.3	Сравнение ПЛК и релейных систем	2-2
2.1.4	Программирование	2-2
2.1.5	Панели оператора	2-2
2.2	Что такое программируемый контроллер?	2-3
2.3	Выполнение программы в контроллере	2-5
2.4	Семейство MELSEC FX.....	2-7
2.5	Выбор контроллера	2-8
2.6	Конструкция контроллеров	2-9
2.6.1	Входные и выходные контуры.....	2-9
2.6.2	Описание базовых блоков MELSEC FX1S.....	2-9
2.6.3	Описание базовых блоков MELSEC FX1N.....	2-10
2.6.4	Описание базовых блоков MELSEC FX2N.....	2-10
2.6.5	Описание базовых блоков MELSEC FX2NC	2-11
2.6.6	Описание базовых блоков MELSEC FX3U	2-11
2.7	Подключение.....	2-12
2.7.1	Блок питания	2-12
2.7.2	Подключение входов	2-13
2.7.3	Подключение выходов.....	2-14
2.8	Расширение диапазона цифровых входов-выходов	2-16
2.8.1	Платы расширения.....	2-16
2.8.2	Компактные расширительные модули	2-16
2.8.3	Модульные блоки расширения	2-17
2.9	Расширение для специальных функций	2-18
2.9.1	Аналоговые модули.....	2-19
2.9.2	Модуль и адаптеры высокоскоростного счетчика	2-21
2.9.3	Модули позиционирования	2-22
2.9.4	Сетевые модули для Ethernet	2-23
2.9.5	Сетевые модули для PROFIBUS/DP	2-24
2.9.6	Сетевые модули для CC-Link	2-26

2.9.7	Сетевой модуль для DeviceNet.....	2-27
2.9.8	Сетевой модуль для CANopen.....	2-27
2.9.9	Сетевой модуль для AS-интерфейса	2-28
2.9.10	Интерфейсные модули и адаптеры	2-29
2.9.11	Коммуникационные адаптеры	2-30
2.9.12	Адаптеры аналогового ввода уставок	2-31
2.10	Конфигурация системы	2-32
2.10.1	Подключение специальных адаптеров (только для FX3U)	2-33
2.10.2	Базовые правила конфигурации системы.....	2-35
2.10.3	Краткие справочные таблицы.....	2-36
2.11	Назначение входов-выходов	2-38
2.11.1	Концепция назначения	2-38
2.11.2	Адрес специального функционального модуля.....	2-39

3 Программирование

3.1	Концепции стандарта IEC61131-3	3-1
3.2	Структура программного обеспечения и определение терминов	3-2
3.2.1	Определение терминов в IEC61131-3	3-2
3.2.2	Системные переменные	3-9
3.2.3	Системные метки	3-10
3.3	Языки программирования.....	3-11
3.3.1	Текстовые редакторы.....	3-11
3.3.2	Графические редакторы	3-13
3.4	Типы данных	3-15
3.4.1	Простые типы данных.....	3-15
3.4.2	Сложные типы данных	3-15
3.4.3	Таймеры и счетчики MELSEC	3-20

4 Разработка проекта

4.1	Запуск GX IEC Developer	4-2
4.2	Прикладная программа.....	4-4
4.2.1	Пример: Управление карусельным столом.....	4-4
4.2.2	Создание нового проекта	4-6
4.2.3	Создание нового программного модуля "POU"	4-8
4.2.4	Назначение глобальных переменных	4-9
4.2.5	Программирование тела программного модуля.....	4-14
4.2.6	Создание новой задачи	4-30
4.2.7	Документирование программы.....	4-34
4.2.8	Проверка и построения кода проекта	4-36
4.2.9	Иллюстрация: Направляемый режим ввода релейной диаграммы	4-37

4.3	Процедуры загрузки проекта	4-38
4.3.7	Подключение с помощью периферийных устройств	4-38
4.3.8	Конфигурация коммуникационного порта	4-38
4.3.9	Загрузка проекта	4-42
4.4	Мониторинг проекта	4-44
4.4.7	Раздельный / Многооконный мониторинг	4-45
4.4.8	Настройка видимости монитора	4-47
4.5	Список перекрестных ссылок	4-48
4.6	Диагностика ПЛК	4-51
4.7	Документация проекта	4-52

5 Пример программы

5.1	QUIZMASTER - ВЕДУЩИЙ ТЕЛЕВИКТОРИНЫ	5-1
5.1.1	Метод	5-2
5.1.2	Quizmaster - Принцип работы	5-5
5.1.3	Описание программы Quizmaster	5-5

6 Функции и функциональные блоки

6.1	Функции	6-1
6.1.1	Пример: Создание функции	6-1
6.1.2	Обработка чисел формата REAL (с плавающей запятой)	6-11
6.2	Создание функционального блока	6-15
6.3	Режимы выполнения функциональных блоков	6-23
6.3.1	Выполнение макрокоманды	6-24
6.3.2	Enable / Enable Output (EN/ENO)	6-24

7 Расширенные функции мониторинга

7.1	Контроль входных данных (EDM)	7-1
7.1.1	Настройка EDM	7-3
7.1.2	Ограничения монитора	7-6
7.1.3	Переключение логических переменных	7-7
7.2	Мониторинг заголовков	7-8
7.3	Основные возможности в режиме мониторинга	7-9
7.4	Мониторинг с использованием "бланка перевода" Mitsubishi	7-11
7.5	Модификация значений переменных из тела программного модуля	7-12
7.6	Мониторинг "экземпляров" функциональных блоков	7-13

8	Принудительная установка цифровых входов и выходов	
9	Функция Device Edit	
10	Режим онлайн	
10.1	Режим изменения в реальном времени	10-1
10.2	Изменение программы в режиме онлайн	10-4
11	Типы блоков данных (DUT)	
11.1	Пример использования DUT	11-2
11.2	Автоматическое заполнение, переменные	11-5
11.3	Назначение переменных DUT функциональным блокам	11-8
12	Массивы	
12.1	Обзор	12-1
12.2	Пример массива: Одномерный массив	12-3
13	Работа с библиотеками	
13.1	Пользовательские библиотеки	13-1
13.1.1	Пример - Создание новой библиотеки	13-1
13.1.2	Открытие библиотеки	13-3
13.1.3	Перемещение программного модуля "Функциональный блок" в открытую библиотеку	13-4
13.2	Специальное замечание о библиотеках	13-7
13.3	Импорт библиотек в проекты	13-8
13.3.1	Импорт функционального блока из библиотеки Мицубиси	13-8
13.3.2	Пример: Импорт функционального блока из библиотеки Mitsubishi	13-11
13.3.3	Контекстная справка библиотечного функционального блока:	13-14
14	Безопасность	
14.1	Пароль	14-1
14.1.1	Настройка пароля	14-1
14.1.2	Изменение уровня безопасности	14-2
14.1.3	Модификация парольного доступа к программному модулю	14-3

15	Последовательная функциональная схема - SFC	
15.1	Что такое SFC?	15-1
15.2	Элементы SFC	15-2
15.2.1	Переходы SFC	15-2
15.2.2	Начальный шаг	15-2
15.2.3	Конечный шаг	15-2
15.3	Примеры конфигурации SFC	15-4
15.4	Действия SFC	15-5
15.5	Сложные переходы	15-7
15.6	Экран программы SFC в режиме монитора	15-8
16	Список инструкций IEC	
16.1	Пример списка инструкций IEC (IL)	16-1
16.1.1	Некоторые полезные подсказки	16-1
16.2	Смешивание IEC IL и MELSEC IL в программных модулях	16-2
17	Структурированный текст IEC	
17.1	Операторы языка структурированного текста	17-1
17.2	Пример программы языка структурированного текста	17-2
18	Связь по PROFIBUS/DP	
18.1	Конфигурирование сети PROFIBUS/DP	18-1
19	Связь по Ethernet	
19.1	Конфигурирование параметров Ethernet модуля FX3U	19-1
19.1.1	Конфигурирование ПЛК (с использованием ПК для начальной настройки)	19-2
19.2	Конфигурирование ПК для Ethernet	19-8
19.3	Конфигурирование GX IEC Developer для доступа контроллера в Ethernet	19-9
19.4	Установка интерфейса человек - машина (HMI)	19-12
19.5	Связь через MX Component	19-15

A	Приложение	
A.1	Специальные маркеры	A-1
A.1.1	Диагностическая информация состояния ПЛК (M8000 - M8009)	A-2
A.1.2	Таймеры и часы реального времени (M8011 - M8019).....	A-2
A.1.3	Режим работы ПЛК (M8030 - M8039).....	A-3
A.1.4	Обнаружение ошибок (M8060 - M8069)	A-3
A.1.5	Платы расширения (предназначенные для FX1S и FX1N).....	A-4
A.1.6	Аналоговый специальный адаптер для FX3U (M8260 - M8299).....	A-4
A.2	Специальные регистры	A-5
A.2.1	Диагностическая информация состояния ПЛК (D8000 - D8009)	A-5
A.2.2	Информация цикла сканирования и часы реального времени (D8010 - D8019)	A-6
A.2.3	Режим работы ПЛК (D8030 - D8039)	A-6
A.2.4	Коды ошибок (D8060 - D8069)	A-7
A.2.5	Платы расширения (предназначенные для FX1S и FX1N).....	A-7
A.2.6	Аналоговый специальный адаптер для FX3U (D8260 - D8299).....	A-7
A.3	Список кодов ошибок	A-8
A.3.1	Коды ошибок 6101 - 6409	A-8
A.3.2	Коды ошибок 6501 - 6510	A-9
A.3.3	Коды ошибок 6610 - 6632	A-10
A.3.4	Коды ошибок 6701 - 6710	A-11
A.4	Количество занятых точек ввода-вывода и потребление тока.....	A-12
A.4.1	Интерфейсные адаптерные платы и платы коммуникационных адаптеров.....	A-12
A.4.2	Средство разработки приложений, преобразователь интерфейса, дисплейный модуль и GOT.....	A-13
A.4.3	Специальные адаптеры.....	A-13
A.4.4	Блоки расширения	A-13
A.4.5	Специальные функциональные модули	A-14
A.5	Глоссарий функциональных компонентов	A-15

1 Обзор курса и требования

Этот курс специально разработан как введение в ПЛК Mitsubishi серии FX с использованием программного пакета **GX IEC Developer** версии 8.

Содержание курса подобрано таким образом, чтобы ознакомить с функциональными возможностями продуктов Mitsubishi серии FX, а также с системой программирования GX Developer. В первом разделе рассматривается аппаратная конфигурация и работа ПЛК, а последующая часть курса охватывает использование системы программирования Mitsubishi, проиллюстрированное на рабочих примерах.

Предполагается, что студенты обладают достаточными практическими знаниями по операционной среде Microsoft Windows.

1.1 Модульный ПЛК для обучения

Существуют различные модели учебных стендов для контроллеров Mitsubishi серии FX. Большинство задач в данном руководстве основано на использовании установок, предложенных в этих тренировочных системах. Используемые в курсе примеры представлены для следующей конфигурации:

- 6 переключателей, имитирующих цифровые входы: X0-X5
- Переменный синхронизирующий вход (1-100 Гц и 0,1- 10 кГц): X7
- 6 светодиодных индикаторов цифровых выходов: Y0-Y5
- 1 специальный функциональный блок FX2N-5A с 4 аналоговыми входами и 1 аналоговым выходом
- 1 специальный адаптер датчика температуры FX3U-4AD-PT-ADP



Таким образом, изменив соответствующие адреса, можно адаптировать другие тренировочные имитаторы к примерам программ, приведенным в этом документе по обучению.

2 Аппаратные средства

2.1 Общее введение в ПЛК

2.1.1 История и разработка

Основанная Ричардом Морли Bedford Associates представила первый программируемый логический контроллер в 1968 году. Этот ПЛК известен как *Modular Digital Controller*, от которого получила название компания MODICON.

Программируемые логические контроллеры были разработаны на замену большим панелям управления на базе реле. Эти системы не обладали достаточной гибкостью и требовали значительного ремонта или замены проводов, если возникала необходимость в изменении последовательности управления.

Разработка микропроцессоров в середине 1970-х позволила программируемым логическим контроллерам взяться за более сложные задачи и масштабные функции, поскольку возросло быстродействие процессора. Сейчас ПЛК лежат в основе функций управления в системах, часто интегрированных со SCADA (*Supervisory Control And Data Acquisition*), интерфейсом человек-машина (HMI – *Human Machine Interfaces*), экспертными системами и графическими интерфейсами пользователя (*Graphical User Interfaces (GUI)*). Требования к ПЛК были расширены, включая обеспечение возможности контроля, обработки данных и управления.

2.1.2 Базовые технические требования к ПЛК

- Легко программируемый и перепрограммируемый на предприятии, чтобы обеспечить последовательность операций, изменяемый.
- Легко обслуживаемый и ремонтируемый – предпочтительно с использованием 'съёмных' карт или модулей.
- Способен противостоять жесткой окружающей среде, механическим и электрическим условиям, имеющимся на предприятии.
- Меньше, чем его эквиваленты на основе реле и "дискретных твердотельных" компонентов.
- Рентабелен по сравнению с системами на основе реле и "дискретных твердотельных" компонентов.

2.1.3 Сравнение ПЛК и релейных систем

Характеристика	ПЛК	Реле
Цена за функцию	Низкая	Низкая – Если эквивалентная релейная программа использует более 10 реле
Физический размер	Очень компактный	Большой
Скорость работы	Быстрая	Медленная
Защищённость от электрических помех	Хорошая	Прекрасная
Конструкция	Легко программируемая	Проводка – отнимает время
Расширенные команды	Да	Нет
Изменение последовательности управления	Очень простое	Очень сложное – требуется изменение проводки
Техническое обслуживание	Отличное. ПЛК отказывают редко	Плохое – требуется регулярное техническое обслуживание реле

2.1.4 Программирование

Релейная логическая схема

ПЛК должны обслуживать инженеры и электрики. Для их поддержки был разработан язык программирования релейных схем. Релейная логика основана на символах реле и контактов, которые инженеры использовали в схемах соединений электрических панелей управления.

Документации для ранних программ ПЛК не существовало, или она была очень бедной, обеспечивая лишь простую адресацию или базовые комментарии, что создавало трудности при сопровождении больших программ. Все значительно улучшилось с разработкой сред программирования ПЛК, таких, как работающий под Windows® программный пакет **GX Developer** от Мицубиси (он будет детально рассмотрен в этом документе).

До последнего времени не было формального стандарта программирования для ПЛК. Введение в 1998 году стандарта **IEC 61131-3** обеспечивает более формальный метод для кодирования. Компания Мицубиси Электрик разработала пакет для программирования, "**GX-IEC Developer**". Это позволяет использовать IEC-совместимое кодирование.

2.1.5 Панели оператора

Ранние программируемые логические контроллеры взаимодействовали с оператором почти тем же способом, что и релейные панели управления, через кнопки и выключатели для управления и индикаторные лампы для индикации.

С появлением персонального компьютера (ПК) в 1980-е стала возможной разработка интерфейса для оператора на базе компьютера, сначала просто через системы диспетчерского управления и сбора данных (Supervisory Control And Data Acquisition – SCADA), а позже через специализированные пульта управления оператора, известные как панели оператора (Human Machine Interfaces – HMI). Сейчас, как правило, можно видеть ПЛК полностью интегрированными с такими продуктами, что создает удобные в использовании решения для систем управления.

Мицубиси предлагает широчайший ассортимент HMI и SCADA продуктов, используемых для различных интерфейсов оператора.



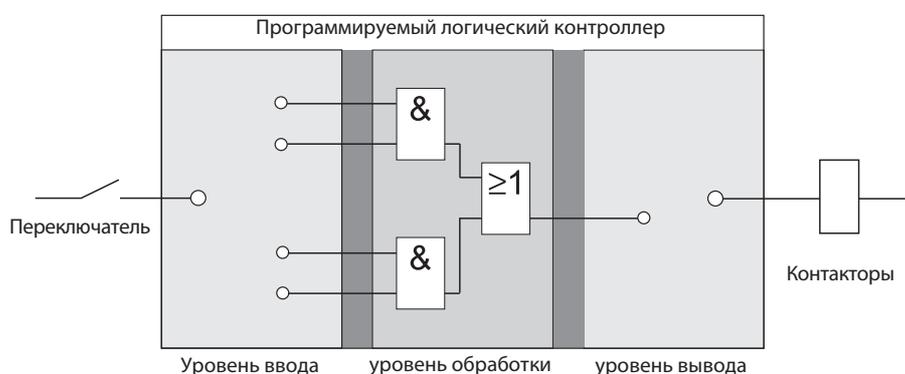
Повсеместно можно найти панели оператора встроенные в системы управления на основе ПЛК, обеспечивающие функциональность интерфейса оператора.

2.2 Что такое программируемый контроллер?

В отличие от системы управления, функционирование которой зависит только от электрического монтажа, работа программируемого контроллера определяется его программой. Конечно, для соединения с внешним миром электропроводка нужна и программируемому контроллеру. Однако его принципиальное отличие заключается в том, что содержание памяти для хранения программы можно в любое время изменить и тем самым приспособить программу к различным задачам управления.

При работе программируемых контроллеров данные вводятся, обрабатываются, а затем снова выводятся как результаты обработки. Этот процесс подразделяется на следующие уровни:

- уровень ввода,
- уровень обработки и
- уровень вывода.



Уровень ввода

Уровень ввода служит для того, чтобы передавать на уровень обработки управляющие сигналы, поступающие от выключателей, кнопок или датчиков.

Сигналы этих компонентов возникают по ходу процесса управления и подаются на входы контроллера в виде логических состояний. С уровня ввода уже предварительно подготовленные сигналы передаются на уровень обработки.

Уровень обработки

На уровне обработки сигналы, принятые и подготовленные на уровне ввода, обрабатываются хранящейся в памяти программой. При этом между ними устанавливаются логические взаимосвязи. Память для программы на уровне обработки свободно программируется. Имеется возможность в любое время изменить ход обработки, изменив или заменив хранящуюся в памяти программу.

Уровень вывода

На выходном уровне результаты, полученные при обработке входных сигналов программой, влияют на подключенные к выходам устройства, например, контакторы, сигнальные лампы, электромагнитные клапаны и т. п.

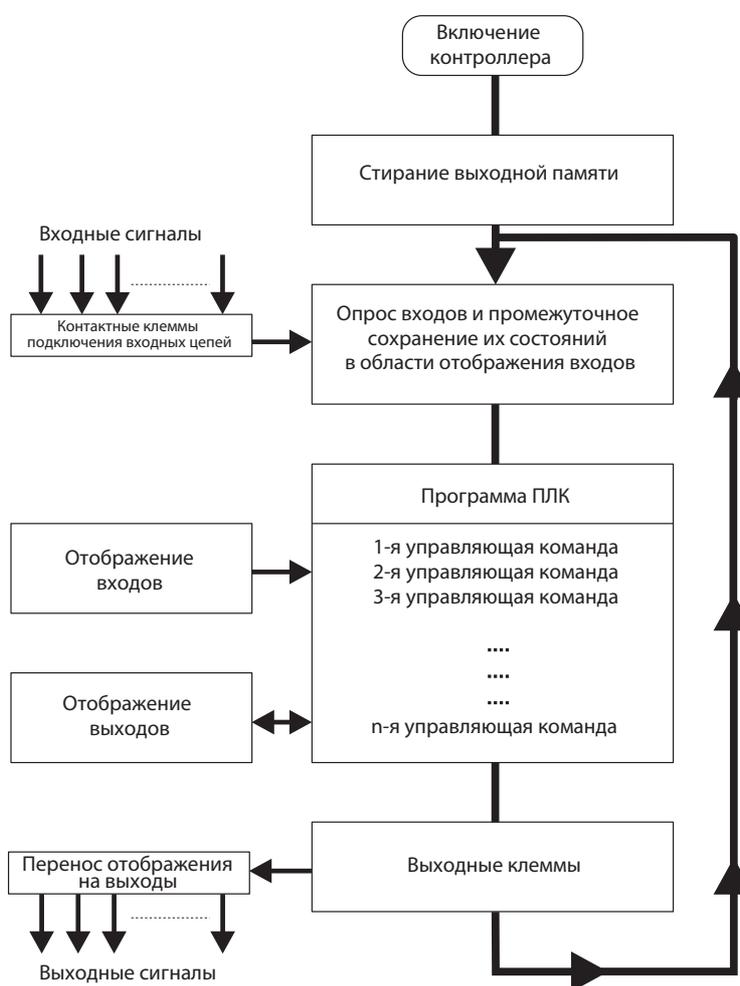
2.3 Выполнение программы в контроллере

Программируемый контроллер работает по заданной программе, которая, как правило, создается вне контроллера, а затем передается в контроллер и хранится в его памяти. Для программирования важно знать, как контроллер обрабатывает программу.

Программа состоит из череды отдельных команд, определяющих функционирование контроллера. Контроллер одну за другой обрабатывает управляющие команды в запрограммированной последовательности. Выполнение всей программы постоянно повторяется, т. е. происходит ее циклическое выполнение. Время, необходимое для выполнения программы, называется временем цикла программы.

Отображение процесса

При обработке программы контроллер обращается не непосредственно к входам и выходам, а к их отображению:



Отображение входов

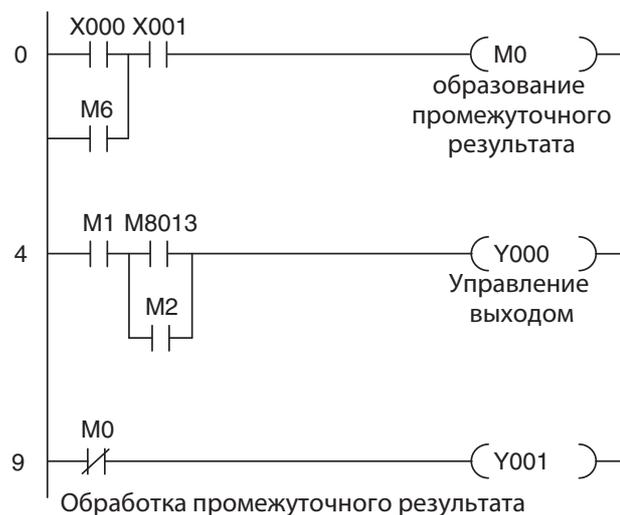
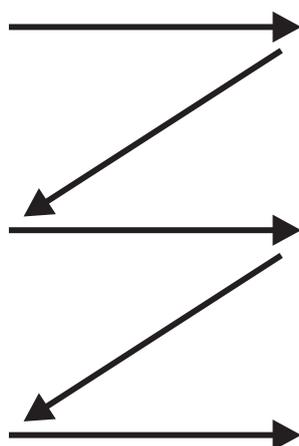
В начале программного цикла состояния входов опрашиваются и сохраняются в промежуточной памяти: создается так называемое отображение входов.

Выполнение программы

Во время последующего прохождения программы контроллер обращается к состояниям входов, хранящимся в области отображения. Поэтому изменения сигналов на входах распознаются лишь при **следующем** программном цикле!

Программа обрабатывается сверху вниз, в последовательности ее ввода. Промежуточные результаты можно использовать уже в том же программном цикле.

Выполнение программы



Отображение выходов

Результаты логических операций, относящиеся к выходам, передаются в выходную буферную память (область отображения выходов). В выходной буферной памяти отображение выходов сохраняется до очередной перезаписи. После присвоения значений выходам программный цикл повторяется.

Обработка сигналов в программируемом контроллере в отличие от системы управления, запрограммированной путем электромонтажа

В случае системы управления, запрограммированной путем электромонтажа, программа задана типом функциональных звеньев и соединениями между ними (электропроводкой). Все процессы управления выполняются одновременно (параллельно). Любое изменение состояний входных сигналов сразу вызывает изменение состояний выходных сигналов.

В программируемом контроллере, если во время выполнения программы изменились состояния входных сигналов, эти изменения могут быть учтены лишь при следующем программном цикле. Этот недостаток в значительной степени компенсируется малыми значениями времени программного цикла. Время цикла программы зависит от количества и типа управляющих команд.

2.4 Семейство MELSEC FX

MELSEC означает MITSUBISHI ELECTRIC SEQUENCER. Компактные малые контроллеры MELSEC серии FX являются экономичным решением для небольших и средних задач управления и позиционирования в промышленности и технике зданий. Эти контроллеры имеют от 10 до 256 встроенных входов и выходов.

Если в установке понадобится сделать изменения, все контроллеры серии FX (кроме FX1S) можно расширять и наращивать в зависимости от потребностей.

Имеются также возможности привязки к сетям обмена данными. Таким образом, контроллеры семейства FX могут поддерживать связь с другими программируемыми контроллерами, а также с регулирующими системами и интерфейсами. Для этого контроллеры можно, во-первых, встраивать в сети МИЦУБИСИ в качестве локальных станций и, во-вторых, применять в качестве подчиненных устройств в открытых сетях (например, PROFIBUS/DP).

Кроме того, семейство MELSEC FX предоставляет возможность создания многоточечных и одноранговых сетей.

Для пользователя, желающего решить сложные задачи управления и при этом использовать многие особые функции (например, аналогово-цифровое и цифро-аналоговое преобразование, сетевая коммуникация), оптимальным выбором являются контроллеры FX1N, FX2N и FX3U, имеющие возможность модульного расширения.

Контроллеры всех типов являются составной частью большого семейства MELSEC FX и совместимы между собой.

Технические данные	FX1S	FX1N	FX2N	FX2NC	FX3U
Макс. количество встроенных адресов входов-выходов	30	60	128	96	128
Расширяемость (максимальное количество входов-выходов)	34	132	256	256	384
Память для программы (шагов)	2000	8000	16000	16000	64000
Время цикла на логическую инструкцию (μс)	0.55 – 0.7	0.55 – 0.7	0.08	0.08	0.065
Количество инструкций (стандартных команд / команд состояния шага / особых команд)	27 / 2 / 85	27 / 2 / 89	27 / 2 / 107	27 / 2 / 107	27 / 2 / 209
Макс. число подключаемых специальных модулей	—	2	8	4	8 справа 10 слева

2.5 Выбор контроллера

Базовые блоки семейства MELSEC FX имеются в различных исполнениях, различающихся по электропитанию и типу выходов. Имеется выбор между приборами с электропитанием 100...240 В пер. тока или 24 В пост. тока, или 12...24 В пост., тока, а также между вариантами выходов

Серия	Входы-выходы	Тип	Кол-во входов	Кол-во выходов	Источник питания	Тип выхода
FX1S	10	FX1S-10 M□-□□	6	8	=24 В или ~100-240 В	Транзисторные или релейные
	14	FX1S-14 M□-□□	8	6		
	20	FX1S-20 M□-□□	12	8		
	30	FX1S-30 M□-□□	16	14		
FX1N	14	FX1N-14 M□-□□	8	6	=24 В или ~100-240 В	Транзисторные или релейные
	24	FX1N-24 M□-□□	14	10		
	40	FX1N-40 M□-□□	24	16		
	60	FX1N-60 M□-□□	36	24		
FX2N	16	FX2N-16 M□-□□	8	8	=24 В или ~100-240 В	Транзисторные или релейные
	32	FX2N-32 M□-□□	16	16		
	48	FX2N-48 M□-□□	24	24		
	64	FX2N-64 M□-□□	32	32		
	80	FX2N-80 M□-□□	40	40		
	128	FX2N-128 M□-□□	64	64		
FX2NC	16	FX2NC-16 M□-□□	8	8	=24 В	Транзисторные или релейные
	32	FX2NC-32 M□-□□	16	16		
	64	FX2NC-64 M□-□□	32	32		
	96	FX2NC-96 M□-□□	48	48		
FX3U	16	FX3U-16 M□-□□	8	8	=24 В или ~100-240 В	Транзисторные или релейные
	32	FX3U-32 M□-□□	16	16		
	48	FX3U-48 M□-□□	24	24		
	64	FX3U-64 M□-□□	32	32		
	80	FX3U-80 M□-□□	40	40		
	128	FX3U-128 M□-□□	64	64	~100-240 В	Транзисторные или релейные

При конфигурации системы необходимо учитывать следующие критерии:

- Требования к источнику питания
 - Напряжение питания: =24 В или ~100-240 В
- Требования к входам-выходам
 - Сколько сигналов должно приниматься (т. е. от внешних выключателей, кнопок и датчиков)?
 - Какие и сколько функций должны коммутироваться?
 - Какие нагрузки коммутируются выходами? Если требуется коммутировать высокие нагрузки, следует применять релейные выходы. Для быстрых, бестриггерных процессов переключения используются транзисторные выходы.
- Специальные функциональные модули
 - Число модулей в системе
 - Требования к внешним источникам питания

2.6 Конструкция контроллеров

Все контроллеры, в принципе, имеют одинаковую конструкцию. В разделе 2.5.7 дан обзор наиболее важных функциональных элементов и узлов.

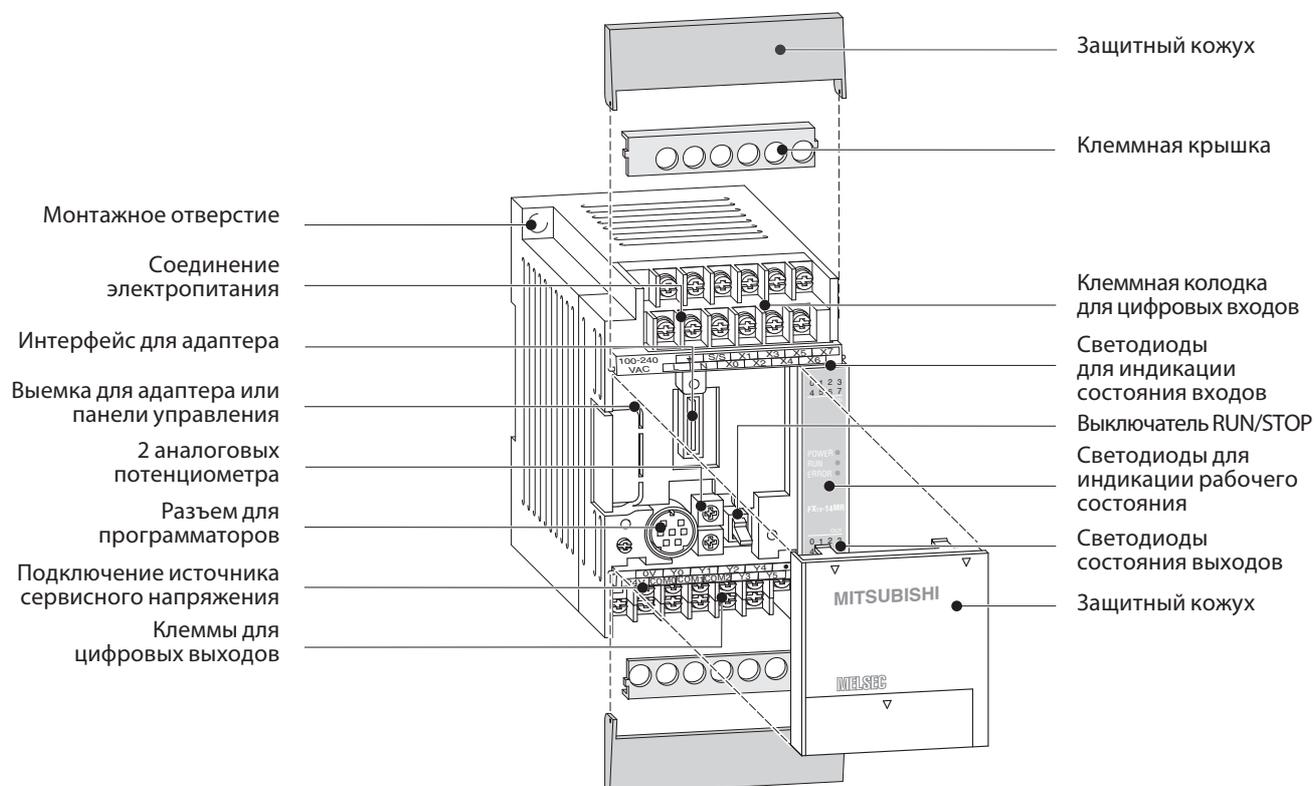
2.6.1 Входные и выходные контуры

Входные контуры выполнены в виде бесконтактных входов. Для изоляции электрических контуров в контроллере используется гальваническая развязка с помощью оптического соединителя. **Выходные контуры** выполнены либо в виде релейных, либо в виде транзисторных выходов. Для изоляции электрических контуров в контроллере в случае транзисторных модулей также используется гальваническая развязка с помощью оптопар.

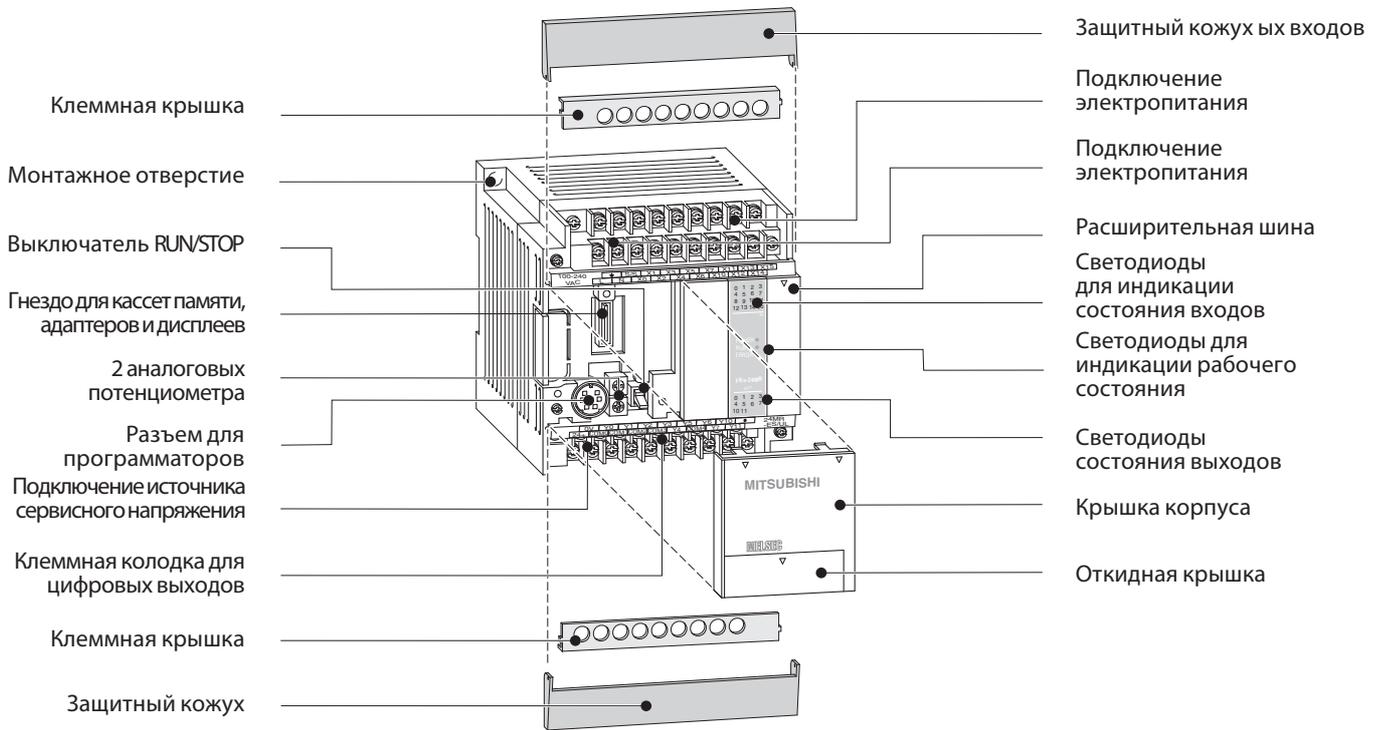
Для всех цифровых входов необходимо определенное коммутируемое входное напряжение (например, постоянное 24 В). Это напряжение можно снимать с встроенного блока питания контроллера. Если коммутируемое напряжение на входе ниже указанного номинального значения (24 В), вход не обрабатывается.

Максимальный выходной ток составляет: в случае релейных модулей 2 А при переменном напряжении 250 В и омической нагрузке; в случае транзисторных модулей 0.5 А при постоянном напряжении 24 В и омической нагрузке.

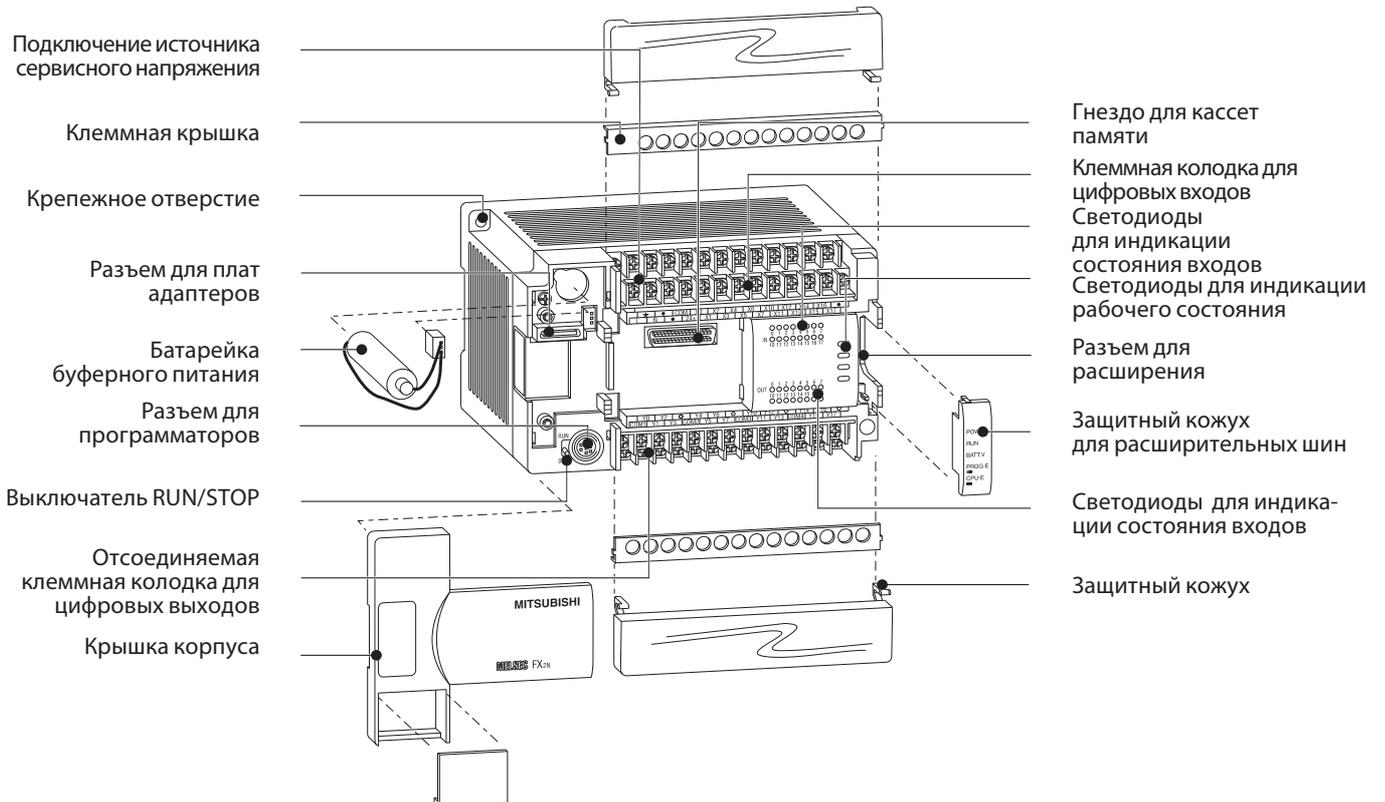
2.6.2 Описание базовых блоков MELSEC FX1S



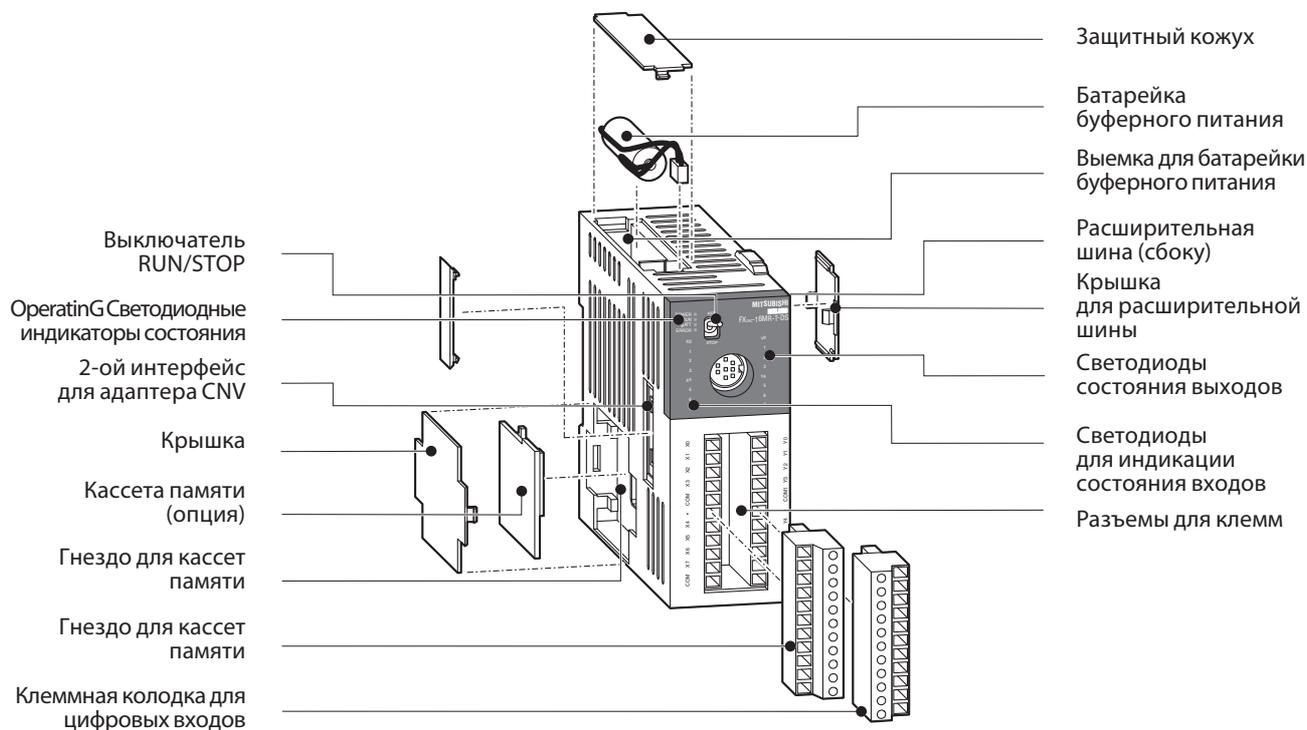
2.6.3 Описание базовых блоков MELSEC FX1N



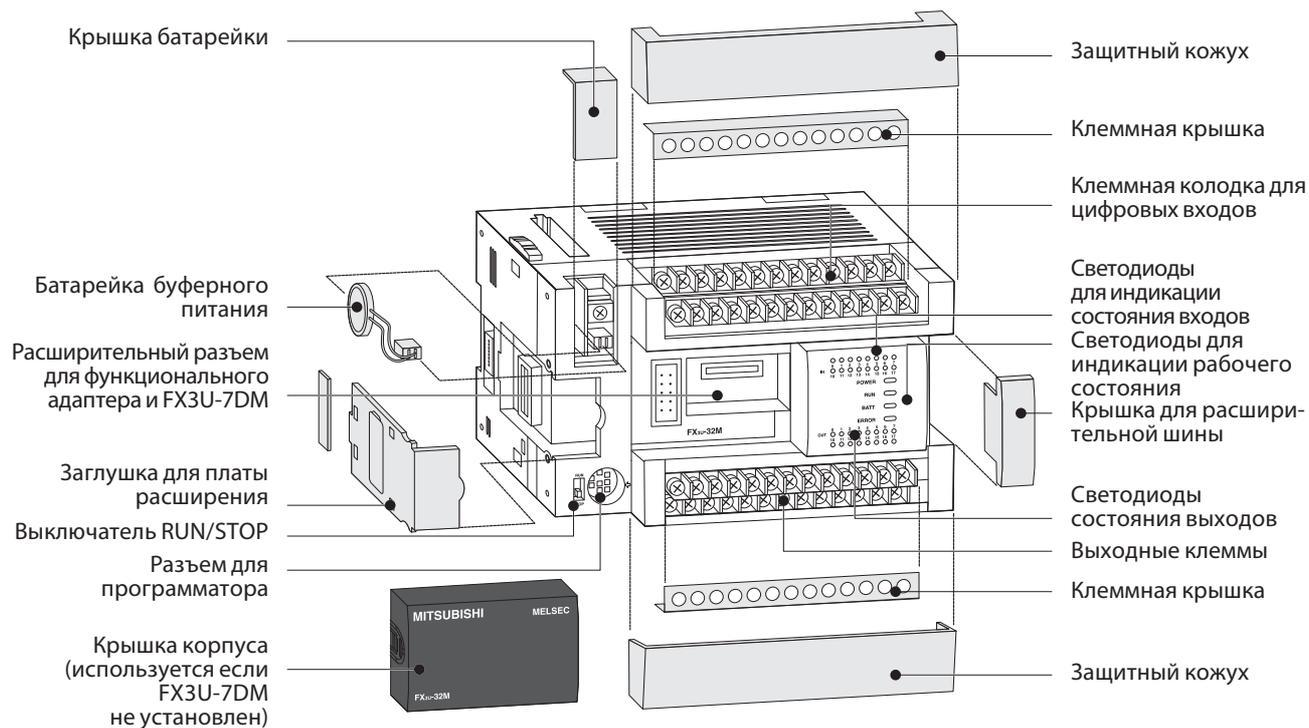
2.6.4 Описание базовых блоков MELSEC FX2N



2.6.5 Описание базовых блоков MELSEC FX2NC



2.6.6 Описание базовых блоков MELSEC FX3U



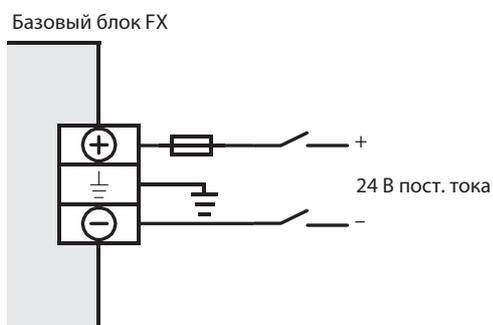
2.7 Подключение

2.7.1 Блок питания

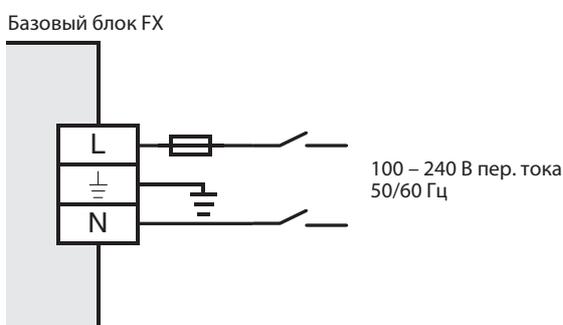
Технические данные источников питания

Параметры	Модули для источника питания пост. тока		Модули для источника питания пер. тока
	12 – 24 В пост. тока	24 В пост. тока	100 – 240 В пер. тока
Номинальное напряжение	12 – 24 В пост. тока	24 В пост. тока	100 – 240 В пер. тока
Диапазон напряжений	10.2 – 26.4 В пост. тока	20.4 – 26.4 В пост. тока	85 – 264 В пер. тока
Допустимое время исчезновения напряжения	5 мс		20 мс

Подключение блоков с источником питания пост. тока



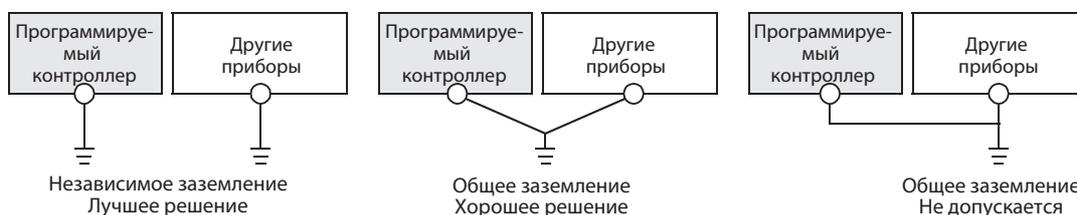
Подключение блоков с источником питания пер. тока



Заземление

ПЛК необходимо заземлять.

- Сопротивление заземления должно быть 100 Ω или ниже.
- Точка заземления должна располагаться близко к ПЛК. Провода заземления должны быть как можно короче.
- Для наилучших результатов следует выполнить независимое заземление. Если независимое заземление не делается, выполните "совместное заземление", согласно следующему рисунку.

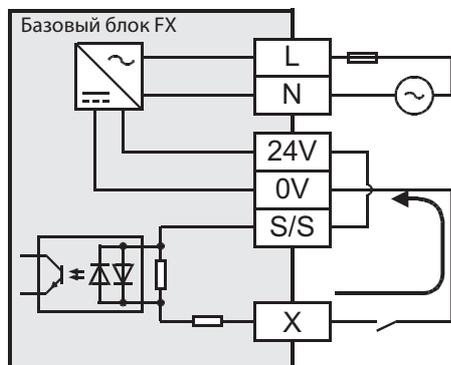


- Сечение проводов заземления должно быть не менее 2 мм².

2.7.2 Подключение входов

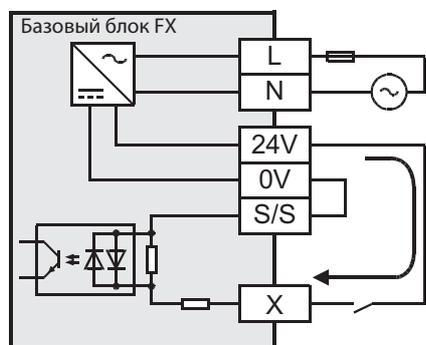
Подключение устройств, работающих как приемник или источник тока

Базовые блоки серии FX можно использовать с коммутационными устройствами, работающими как приемник или источник тока. Выбор осуществляется различными соединениями клемм "S/S".



Если вход работает в режиме отрицательной логики, то клемма S/S соединена с клеммой 24 В источника сервисного напряжения или, когда используется главный блок с собственным блоком питания постоянного тока, с положительным полюсом источника питания.

Вход, работающий в режиме отрицательной логики, означает, что контакт, подключенный к входу (X), или датчик с NPN транзисторным выходом с открытым коллектором соединяет вход ПЛК с отрицательным полюсом источника питания.



Если вход работает в режиме положительной логики, то клемма S/S соединена с клеммой 0 В источника сервисного напряжения или, когда используется главный блок с собственным блоком питания постоянного тока, с отрицательным полюсом источника питания.

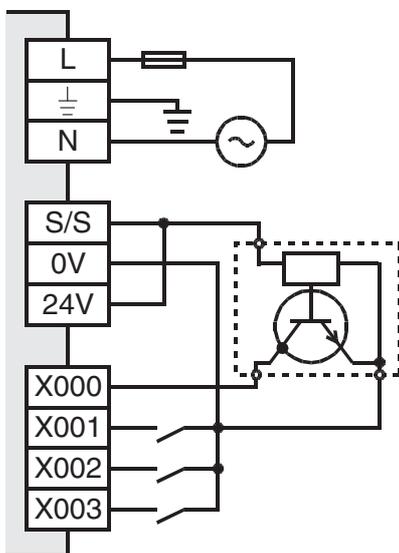
Вход, работающий в режиме положительной логики, означает, что контакт, подключенный к входу (X), или датчик с PNP транзисторным выходом с открытым коллектором соединяет вход ПЛК с положительным полюсом источника питания.

Все входы базового блока или расширительного модуля могут использоваться как входы приемника или источника тока, но не допускается смешивать входы приемника и источника тока в одном модуле. Однако отдельные модули в одном ПЛК могут быть установлены как входы приемника или источника тока, поскольку базовый блок и модули расширения с питаемыми входами/выходами индивидуально установлены во входной режим приемника или источника тока.

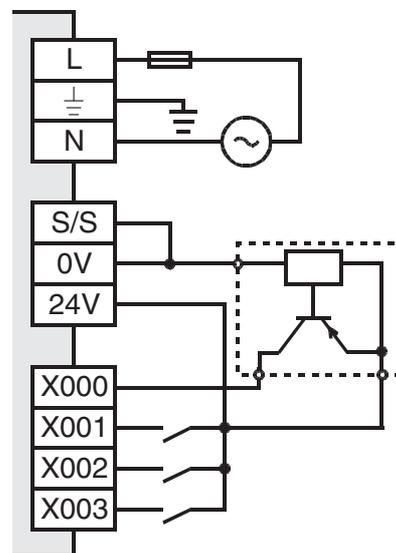
Примеры для типов входов

Базовые блоки, питаемые от переменного тока

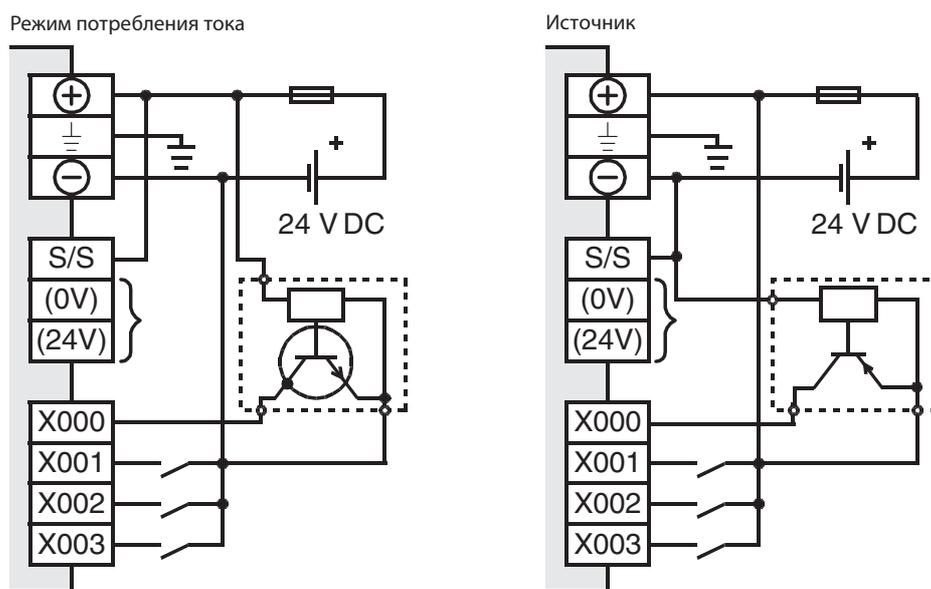
Режим потребления тока



Источник



Базовые блоки, питаемые от постоянного тока



2.7.3 Подключение выходов

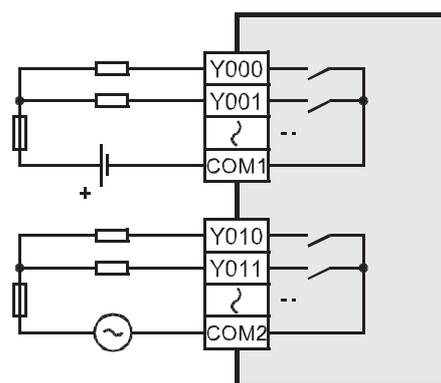
В случае FX3U-16M□ каждый выход можно подключать отдельно. Для главных блоков FX3U-32□M – FX3U-128M□ выходы объединены в группы по 4 или 8 выходов. Каждая группа имеет общий контакт для напряжения нагрузки. Клеммы маркированы "COM□" для главных блоков с релейными выходами или транзисторными выходами, работающими в режиме приемника тока, и "+V□" для главных блоков с транзисторными выходами, работающими в режиме источника тока. "□" обозначает номер группы выходов, например "COM1".

Поскольку группы выходов изолированы друг от друга, один главный блок может коммутировать несколько напряжений с различными потенциалами. Главные блоки с релейными выходами могут коммутировать даже переменное и постоянное напряжение.

Базовый блок FX3U с релейными выходами

Первая группа выходов используется для подключения постоянного напряжения.

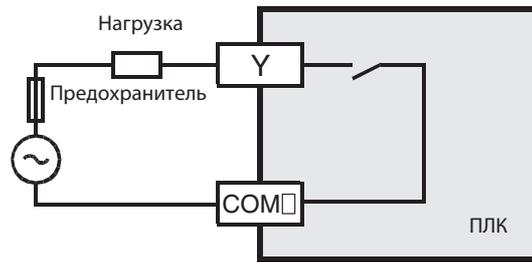
Вторая группа реле управляет нагрузки, питаемые переменным током.



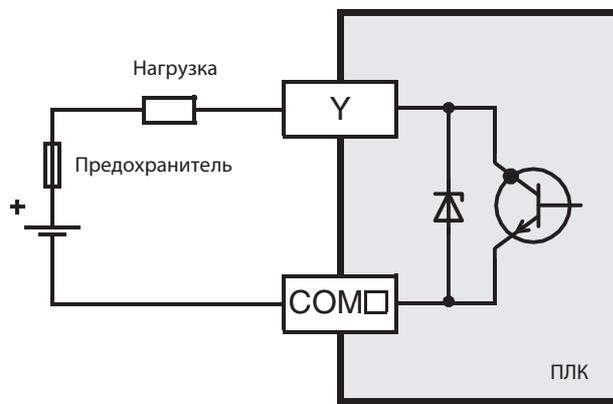
Выбор типа выхода приемника или источника тока осуществляется выбором соответствующего базового блока. Оба типа имеются с питания переменного или постоянного напряжения. Тип выхода задается в обозначении типа модели: базовые блоки с кодом "MT/□S" обеспечивают транзисторные выходы в режиме приемника тока (например, FX3U-16MT/ES), тогда как базовые блоки с кодом "MT/□SS" обеспечивают транзисторные выходы в режиме источника тока (например, FX3U-16MT/ESS).

Примеры подключения выходов

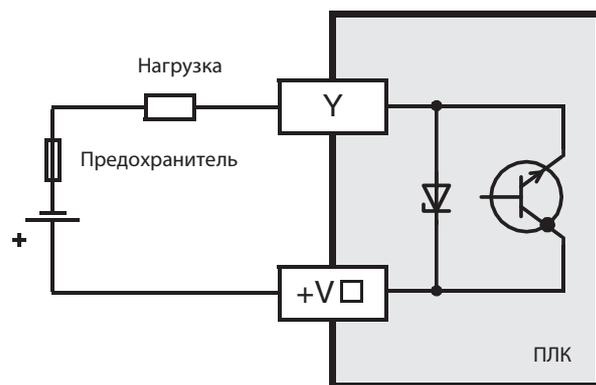
Релейный выход



Транзисторный выход (приемник)



Транзисторный выход (источник)

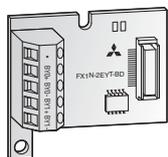


2.8 Расширение диапазона цифровых входов-выходов

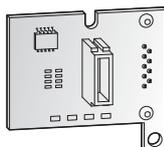
Для ПЛК семейства MELSEC FX имеется несколько способов обеспечения базового блока дополнительными входами и выходами.

2.8.1 Платы расширения

FX1N-2EYT-BD с двумя цифровыми выходами



Сторона с разъемом



Для небольшого количества входов-выходов (2 – 4) плату адаптера расширения можно установить непосредственно в базовом блоке FX1S или FX1N. Поэтому платы адаптеров расширения не требуют дополнительного места для установки.

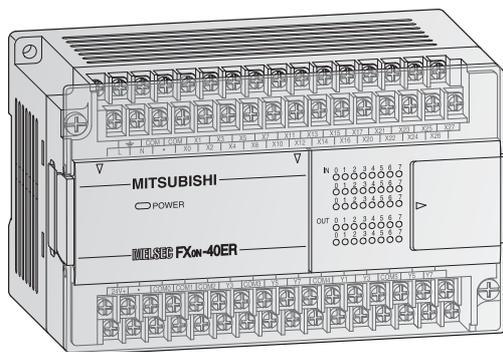
Состояние дополнительных входов и выходов отражается в специальных реле в ПЛК (см. раздел А.1.5). В программе эти реле используются вместо операндов X и Y.

Обозначение	Количество входов-выходов			Тип выхода	Источник питания	FX1S	FX1N	FX2N FX2NC	FX3U
	Полное	Кол-во входов	Кол-во выходов						
FX1N-4EX-BD	4	4	—	—	От базового блока	●	●	○	○
FX1N-2EYT-BD	2	—	2	Транзисторный					

● : Плату расширения можно использовать с базовым блоком этой серии.

○ : Плату расширения нельзя использовать с этой серией.

2.8.2 Компактные расширительные модули



Компактные расширительные модули включают собственный блок питания. Встроенный источник сервисного напряжения (24 В пост. тока) расширительных модулей с собственным блоком питания переменного тока можно использовать для питания внешних устройств. Можно выбирать релейный или транзисторный (источник тока) тип выхода.

Компактные расширительные модули серии FX0N

Обозначение	Количество входов-выходов			Тип выхода	Источник питания	FX1S	FX1N	FX2N FX2NC	FX3U
	Полное	Кол-во входов	Кол-во выходов						
FX0N-40ER/ES-UL	40	24	16	Релейный	100 – 240 В переменн. тока	○	●	○	
FX0N-40ER/DS	40	24	16	Релейный	24 В пост. тока				
FX0N-40ET/DSS	40	24	16	Транзисторный					

● : Расширительный модуль можно использовать с базовым блоком этой серии.

○ : Расширительный модуль нельзя использовать с этой серией.

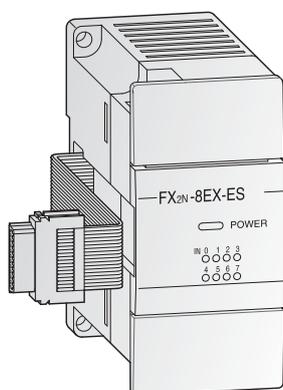
Компактные расширительные модули серии FX2N

Обозначение	Количество входов-выходов			Тип выхода	Источник питания	FX1S	FX1N	FX2N FX2NC	FX3U
	Полное	Кол-во входов	Кол-во выходов						
FX2N-32ER-ES/UL	32	16	16	Релейный	100-240 В переменн. тока	○	●	●	●
FX2N-32ET-ESS/UL	32	16	16	Транзисторный					
FX2N-48ER-ES/UL	48	16	16	Релейный	24 В пост. тока				
FX2N-48ET-ESS/UL	48	24	24	Транзисторный					
FX2N-48ER-DS	48	24	24	Релейный					
FX2N-48ET-DSS	48	24	24	Транзисторный					

● : Расширительный модуль можно использовать с базовым блоком этой серии.

○ : Расширительный модуль нельзя использовать с этой серией.

2.8.3 Модульные блоки расширения



Модульные блоки расширения не оснащены встроенными блоками питания, но имеют очень компактные размеры. Имеются модульные блоки расширения серии FX2N с 8 или 16 точками входа-выхода. Можно выбирать релейный или транзисторный (источник тока) тип выхода.

Обозначение	Количество входов-выходов			Тип выхода	Источник питания	FX1S	FX1N	FX2N FX2NC	FX3U
	Полное	Кол-во входов	Кол-во выходов						
FX2N-8ER-ES/UL	16*	4	4	Релейный	100 – 240 В переменн. тока	○	●	●	●
FX2N-8EX-ES/UL	8	8	—	—					
FX2N-16EX-ES/UL	16	16	—	—					
FX2N-8EYR-ES/UL	8	—	8	Релейный	24 В пост. тока				
FX2N-8EYT-ESS/UL	8	—	8	Транзисторный					
FX2N-16EYR-ES/UL	16	—	16	Релейный					
FX2N-16EYT-ESS/UL	16	—	16	Транзисторный					

* Блок расширения FX2N-8ER-ES/UL занимает 16 точек входа/выхода ПЛК. Четыре вход и четыре выхода заняты и не могут использоваться.

2.9.1 Аналоговые модули

Без дополнительных модулей базовые блоки семейства MELSEC FX могут обрабатывать только цифровые входные и выходные сигналы (например, данные ВКЛ/ВЫКЛ). Поэтому для ввода и вывода аналоговых сигналов необходимы дополнительные аналоговые модули.

Тип модуля	Обозначение	К-во каналов	Область	Разрешение	FX1S	FX1N	FX2N FX2NC	FX3U	
Модули аналоговых входов	Адаптерная плата	FX1N-2AD-BD	2	Напряжение: 0 – 10 В пост. тока	2.5 мВ (12 бит)	●	●	○	○
				Ток: 4 мА – 20 мА пост. тока	8 мкА (11 бит)				
	Специальный адаптер	FX3U-4AD-ADP	4	Напряжение: 0 – 10 В пост. тока	2.5 мВ (12 бит)	○	○	○	●
				Ток: 4 мА – 20 мА пост. тока	10 мкА (11 бит)				
	Специальные функциональные модули	FX2N-2AD	2	Напряжение: 0 – 5 В пост. тока 0 В – 10 В пост. тока	2.5 мВ (12 бит)	○	●	●	●
				Ток: 4 мА – 20 мА пост. тока	4 мкА (12 бит)				
		FX2N-4AD	4	Напряжение: -10 В ... +10 В пост. тока	5 мВ (со знаком, 11 бит)	○	●	●	●
				Ток: 4 мА – 20 мА пост. тока -20 мА ... + 20 мА пост. тока	10 мкА (со знаком, 10 бит)				
		FX2N-8AD*	8	Напряжение: -10 В ... +10 В пост. тока	0.63 мВ (со знаком, 14 бит)	○	●	●	●
				Ток: 4 мА – 20 мА пост. тока -20 мА ... + 20 мА пост. тока	2.50 мкА (со знаком, 13 бит)				
FX3U-4AD	4	Напряжение: -10 В ... +10 В пост. тока	0.32 мВ (со знаком, 16 бит)	○	○	○	●		
		Ток: 4 мА – 20 мА пост. тока -20 мА ... + 20 мА пост. тока	1.25 мкА (со знаком, 15 бит)						
Модули аналоговых выходов	Адаптерная плата	FX1N-1DA-BD	1	Напряжение: 0 – 10 В пост. тока	2.5 мВ (12 бит)	●	●	○	○
				Ток: 4 мА – 20 мА пост. тока	8 мкА (11 бит)				
	Специальный адаптер	FX3U-4DA-ADP	4	Напряжение: 0 – 10 В пост. тока	2.5 мВ (12 бит)	○	○	○	●
				Ток: 4 мА – 20 мА пост. тока	4 мкА (12 бит)				
	Специальный функциональный блок	FX2N-2DA	2	Напряжение: 0 – 5 В пост. тока 0 В – 10 В пост. тока	2.5 мВ (12 бит)	○	●	●	●
				Ток: 4 мА – 20 мА пост. тока	4 мкА (12 бит)				
		FX2N-4DA	4	Напряжение: -10 В ... +10 В пост. тока	5 мВ (10 бит)	○	●	●	●
				Ток: 0 мА – 20 мА пост. тока 4 мА – 20 мА пост. тока	20 мкА (со знаком, 11 бит)				
		FX3U-4DA	4	Напряжение: -10 В ... +10 В пост. тока	0.32 мВ (со знаком, 16 бит)	○	○	○	●
				Ток: 0 мА – 20 мА пост. 4 мА – 20 мА пост.	0.63 мкА (15 бит)				

* Специальный функциональный блок FX2N-8AD подходит для измерения напряжения, тока и температуры

Тип модуля	Обозначение	К-во каналов	Область	Разрешение	FX1S	FX1N	FX2N FX2NC	FX3U	
Комбинированные модули аналогового ввода/вывода	Специальные функциональные модули	2 входа	Напряжение: 0 - 5 В пост. тока 0 В - 10 В пост. тока	40 мВ (8 бит)	○	●	●	●	
			Ток: 4 мА - 20 мА пост. тока	64 мкА (8 бит)					
		1 выход	Напряжение: 0 - 5 В пост. тока 0 В - 10 В пост. тока	40 мВ (8 бит)					
			Ток: 4 мА - 20 мА пост. тока	64 мкА (8 бит)					
	FX2N-5A	4 входа	Напряжение: -100 мВ ... +100 мВ пост. тока -10 В ... +10 В пост. тока	50 мкВ (со знаком, 11 бит) 0.312 мВ (со знаком, 15 бит)	○	●	●	●	
			Ток: 4 мА - 20 мА пост. тока -20 мА ... + 20 мА пост. тока	10 мкВ/1,25 мкА (со знаком, 14 бит)					
1 выход	Напряжение: -10 В ... +10 В пост. тока	5 мВ (со знаком, 12 бит)							
	Ток: 0 мА - 20 мА пост. тока	20 мкА (10 бит)							
Модули обработки температуры	Специальный адаптер	FX3U-4AD-PT-ADP	4	Резистивный термометр Pt100: -50 °C ... 250 °C	0.1 °C	○	○	○	●
		FX3U-4AD-TC-ADP	4	Термопара тип K: -100 °C ... 1000 °C	0.4 °C	○	○	○	●
	Специальные функциональные модули	FX2N-8AD*	8	Термопара тип K: -100 °C ... 1200 °C	0.1 °C				
				Термопара тип J: -100 °C ... 600 °C	0.1 °C				
				Термопара тип T: -100 °C ... 350 °C	0.1 °C				
	FX2N-4AD-PT	4	Резистивный термометр Pt100: -100 °C ... 600 °C	0.2 ... 0, °C	○	●	●	●	
	FX2N-4AD-TC	4	Термопара тип K: -100 °C ... 1200 °C	0.4 °C	○	●	●	●	
			Термопара тип J: -100 °C ... 600 °C	0.3 °C					
Модуль регулирования температуры (Специальные функциональные модули)	FX2N-2LC	2	Например, с термопарой тип K: -100 °C ... 1300 °C	0.1 °C или 1 °C (зависит от используемого температурного зонда)	○	●	●	●	
			Резистивный термометр Pt100: -200 °C ... 600 °C						

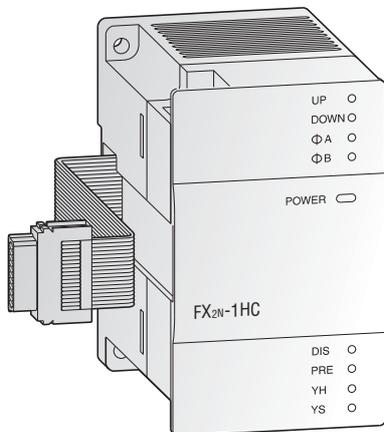
* Специальный функциональный блок FX2N-8AD подходит для измерения напряжения, тока и температуры.

- Адаптерную плату, специальный адаптер или специальный функциональный модуль можно использовать с базовым блоком или модулем расширения этой серии.
- Адаптерную плату, специальный адаптер или специальный функциональный модуль нельзя использовать с этой серией.

2.9.2 Модуль и адаптеры высокоскоростного счетчика

FX2N-1HC

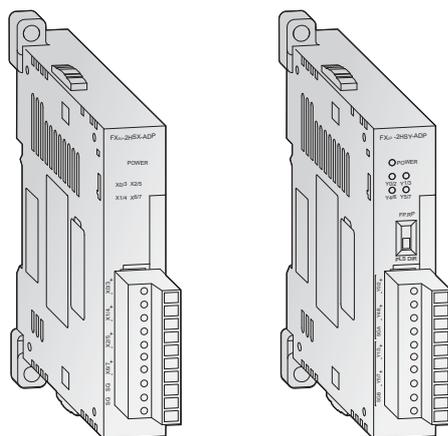
Кроме внутренних высокоскоростных счетчиков MELSEC FX, модуль высокоскоростного счетчика FX2N-1HC обеспечивает пользователю внешний счетчик. Он подсчитывает 1- или 2-фазные импульсы с частотой до 50 кГц. Счетный диапазон 16 или 32 бит.



Два встроенных транзисторных выхода могут переключаться независимо друг от друга с помощью внутренних функций сравнения. Поэтому простые задачи позиционирования можно также реализовать экономично. Кроме того, FX2N-1HC может использоваться как кольцевой счетчик.

FX3U-4HSX-ADP и FX3U-2HSY-ADP

Эти адаптерные модули поддерживают прямую обработку данных приложений позиционирования.



FX3U-4HSX-ADP (крайний левый) обеспечивает четыре высокоскоростных счетных входа до 200 кГц, а FX3U-2HSY-ADP (слева) дает два канала счетных выходов до 200 кГц.

Обзор модулей высокоскоростных счетчиков/адаптеров

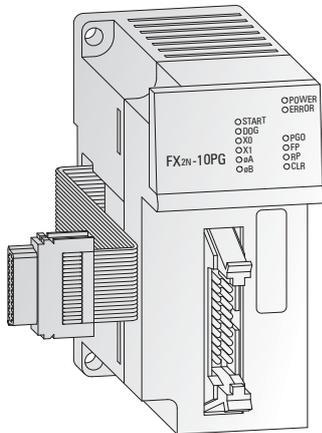
Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальный модуль	FX2N-1HC	1-кан. высокоскоростной счетчик	○	○	●	●
Специальный адаптер	FX3U-4HSX-ADP	Вход дифференциального линейного электропривода (высокоскоростной счетчик)	○	○	○	●
	FX3U-2HSY-ADP	Вход дифференциального линейного электропривода (выход позиционирования)				

- Специальный адаптерный или специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный адаптерный или специальный функциональный модуль нельзя использовать с этой серией.

2.9.3 Модули позиционирования

FX2N-1PG-E, FX2N-10PG

Модули позиционирования FX2N-1PG-E и FX2N-10PG представляют собой многофункциональные одноосевые модули позиционирования для управления либо шаговыми приводами, либо сервоприводами (с помощью внешнего регулятора) с использованием последовательности импульсов.



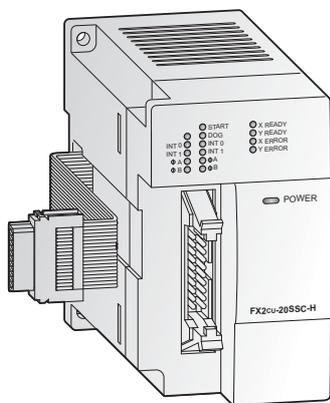
Данные модули, используемые в составе контроллеров серии FX MELSEC, обеспечивают высокую точность позиционирования. Конфигурация параметров позиционирования выполняется непосредственно с помощью программы ПЛК.

FX2N-1PG-E обеспечивает выход с открытым коллектором 100 кГц, тогда как FX2N-10PG оснащен выходом дифференциального линейного электропривода 1 МГц.

В распоряжении пользователя широкий набор функций для ручного и автоматического режимов.

FX3U-20SSC-H

Использование модуля SSCNET* FX3U-20SSC-H в комбинации с программируемым контроллером FX3U - экономически целесообразное решение для высокоточного, высокоскоростного позиционирования. Опволоконный кабель SSCNET типа plug-and-play уменьшает время установки и увеличивает дальность передачи сигнала управления для операций позиционирования в широком спектре приложений.



Параметры сервоприводов и данные позиционирования для FX3U-20SSC-H легко задаются с помощью базового блока FX3U и персонального компьютера. Для установки параметров, контроля и тестирования имеется простое программное обеспечение FX Configurator-FP.

* SSCNET: Servo System Controller Network

Обзор модулей позиционирования

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальные функциональные модули	FX2N-1PG-E	Импульсный выход для независимого 1-осевого управления	○	○	●	●
	FX2N-10PG		○	○	●	●
	FX3U-20SSC-H	Одновременное 2-осевое (независимо 2оси) управление (применимо в SSCNET III)	○	○	○	●

- Специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный функциональный модуль нельзя использовать с этой серией.

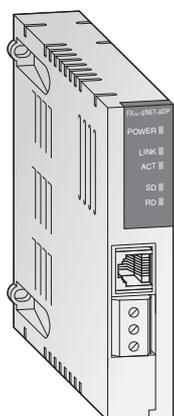
2.9.4 Сетевые модули для Ethernet

ETHERNET является наиболее широко распространенной сетью для соединения информационных процессоров, таких как персональные компьютеры и рабочие станции. При подключении интерфейса ETHERNET к ПЛК, относящуюся к производству управленческую информацию можно быстро передавать на персональные компьютеры или рабочие станции. ETHERNET является платформой для очень широкого диапазона протоколов обмена данными. Комбинация Ethernet и достаточно широко распространенного протокола TCP/IP обеспечивает высокоскоростную передачу данных между системами контроля технологического процесса и ПЛК серии MELSEC. TCP/IP предоставляет логическую связь "точка-точка" между двумя станциями ETHERNET.

В программном обеспечении GX Developer имеются функциональные блоки или конфигурационные подпрограммы для ПЛК, что ускоряет и упрощает создание конфигурации для одного или нескольких соединений TCP/IP.

FX2NC-ENET-ADP

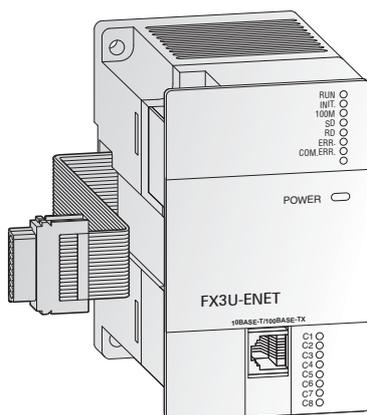
Коммуникационный адаптер FX2NC-ENET-ADP - это интерфейс Ethernet (10BASE-T) для серий FX1S, FX1N, FX2NC и FX2N*.



FX2NC-ENET-ADP позволяет загружать, скачивать, контролировать и тестировать программы через Ethernet с персонального компьютера (с установленными GX Developer или MX Component и виртуальным драйвером COM порта).

* При подключении этого адаптерного модуля к ПЛК FX1S или FX1N требуется коммуникационный адаптер FX1N-CNV-BD. При подключении этого адаптерного модуля к ПЛК FX2N требуется коммуникационный адаптер FX2N-CNV-BD.

FX3U-ENET



Коммуникационный модуль FX3U-ENET обеспечивает FX3U прямым соединением с сетью Ethernet.

FX3U-ENET установленный к ПЛК FX3U обеспечивает полную загрузку/выгрузку программ, всеобъемлющий мониторинг, а также быстрый и легкий обмен данными с системой визуализации процессов. Модуль также поддерживает одноранговое соединение и протокол MC. Его легко настроить с помощью программного обеспечения FX Configurator-EN.

Обзор сетевых модулей для Ethernet

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальные функциональные модули	FX2NC-ENET-ADP	Сетевые модули Ethernet	●	●	●	○
	FX3U-ENET		○	○	○	●

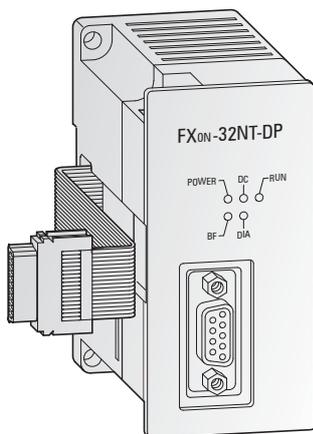
2.9.5 Сетевые модули для PROFIBUS/DP

Сеть PROFIBUS/DP устанавливает связь между ведущим модулем и децентрализованными ведомыми модулями со скоростью передачи данных до 12 Мбит/с. С ПЛК MELSEC в качестве ведущего, PROFIBUS/DP обеспечивает быстрое и простое соединение датчиков и исполнительных механизмов, даже от различных производителей.

ПЛК MELSEC, используемый как ведомый в сети PROFIBUS/DP, может выполнять задачи децентрализованного управления и одновременно обмениваться данными с ведущим устройством PROFIBUS/DP.

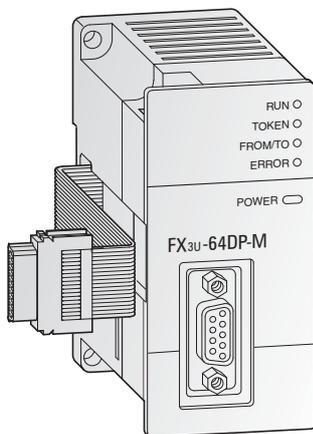
Для того, чтобы сократить стоимость, в PROFIBUS/DP используется технология RS485 с экранированным 2-проводным кабелем.

FX0N-32NT-DP



FX0N-32NT-DP позволяет интегрировать ПЛК семейства MELSEC FX ведомым в существующую сеть PROFIBUS /DP. Он связывает систему с ведущим ПЛК в сети PROFIBUS/DP для эффективного и надежного обмена данными.

FX3U-64DP-M

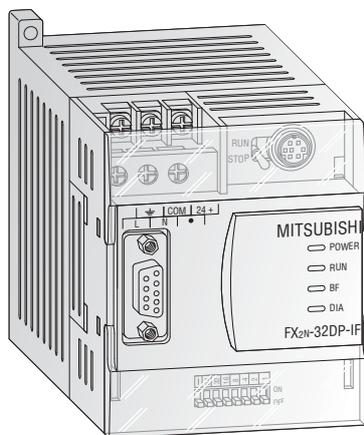


С FX3U-64DP-M ПЛК MELSEC FX3U может действовать как мастер класса 1 в сети PROFIBUS/DP. Это дает возможность ЦП FX3U организовать обмен данными по сети PROFIBUS/DP для реализации задач децентрализованного управления.

Ведущий PROFIBUS/DP FX3U легко настроить с помощью программного обеспечения GX Configurator-DP.

FX2N-32DP-IF

Станция удаленного ввода-вывода FX2N-32DP-IF представляет собой чрезвычайно компактный модуль связи, обеспечивающий подключение модулей ввода-вывода с количеством точек ввода-вывода до 256 и/или альтернативно до 8 специальных функциональных модулей.



В станции удаленного ввода-вывода базовый блок FX не устанавливается. FX2N-32DP-IF связывает подключенные модули ввода/вывода или специальные функциональные модули с ведущим ПЛК в сети PROFIBUS/DP. С ПЛК FX3U и FX3U-64DP-M в качестве ведущего модуля PROFIBUS/DP, можно создать достаточно эффективную систему удаленного ввода-вывода, используя только компоненты серии FX.

Данные PROFIBUS такие как скорость передачи или данные ввода-вывода можно контролировать непосредственно с помощью программного обеспечения или на ручном программаторе FX-20P-E. Такие возможности облегчают диагностику ошибок непосредственно на станции удаленного ввода-вывода.

Обзор модулей PROFIBUS/DP

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальные функциональные модули	FX0N-32NT-DP	Ведомый PROFIBUS/DP	●	●	●	●
	FX3U-64DP-M	Ведущий PROFIBUS/DP	○	○	○	●
—	FX2N-32DP-IF	Станция удаленного ввода-вывода PROFIBUS/DP	Электропитание: 100-240 В переменн. тока		Совместимы с ведущими PROFIBUS/DP	
	FX2N-32DP-IF-D		Электропитание: 24 В пост. тока			

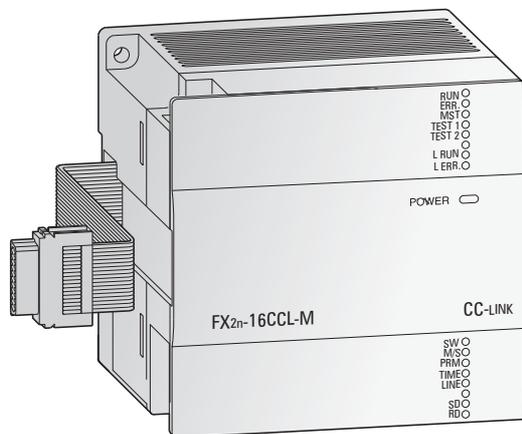
- Специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный функциональный модуль нельзя использовать с этой серией.

2.9.6 Сетевые модули для CC-Link

Ведущий модуль CC-Link FX2N-16CCL-M

Сетевые модули CC-Link управляют и контролируют децентрализованные модули ввода-вывода.

Ведущий модуль CC-Link FX2N-16CCL-M - это специальный блок расширения, делающий ПЛК серии FX ведущей станцией системы CC-Link.



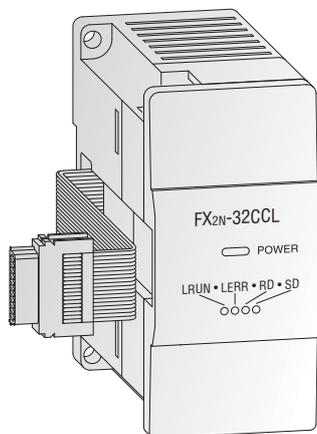
Настройка всех модулей в сети осуществляется непосредственно ведущим модулем.

В качестве децентрализованных станций ввода/вывода к ведущей станции можно подключить до 15 удаленных станций и удаленных интерфейсов устройств. Эти удаленные станции могут включать до 7 модулей ввода-вывода и до 8 интеллектуальных модулей. К одному базовому блоку FX1N или FX2N можно подключить два ведущих модуля.

Максимальное расстояние передачи составляет 1200 м без повторителя.

Коммуникационный модуль CC-Link FX2N-32CCL

Коммуникационный модуль FX2N-32CCL обеспечивает пользователю подключение к сети CC-Link с системой ПЛК верхнего уровня в качестве ведущего ЦП. Это предоставляет ему доступ ко всем ПЛК MELSEC и преобразователям частоты, подключенным к данной сети, а также к дополнительным продуктам других производителей, а также дополнительных продуктов других поставщиков.



Таким образом, сеть может быть расширена дискретными модулями входа-выхода серии FX до 256 точек входа-выхода.

Связь между базовым блоком и внутренней памятью FX2N-32CCL обрабатывается инструкциями FROM/ TO.

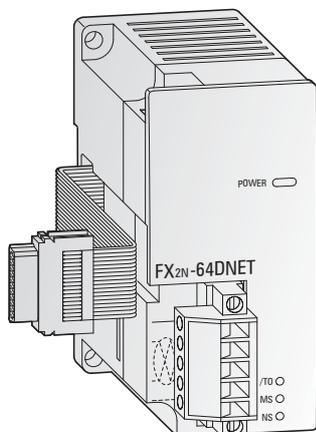
Обзор сетевых модулей для CC-Link

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальные функциональные модули	FX2N-16CCL-M	Ведущий для CC-Link	○	●	●	●
	FX2N-32CCL	Удаленная станция для CC-Link	○	●	●	●

- Специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный функциональный модуль нельзя использовать с этой серией.

2.9.7 Сетевой модуль для DeviceNet

DeviceNet представляют экономичное решение для сетевой интеграция оконечного оборудование низкого уровня. В одну сеть можно интегрировать до 64 устройств, включая ведущее. Для обмена данными используется кабель с двумя экранированными витыми парами.



Ведомый модуль DeviceNet FX2N-64DNET может использоваться для подключения контроллеров FX2N и FX3U к сети DeviceNet.

FX2N-64DNET может связываться с ведущим по связи ведущий/ведомый (используя соединение ввода-вывода ведущий/ведомый) и с другими узлами, поддерживая соединение UCMM по типу клиент/сервер.

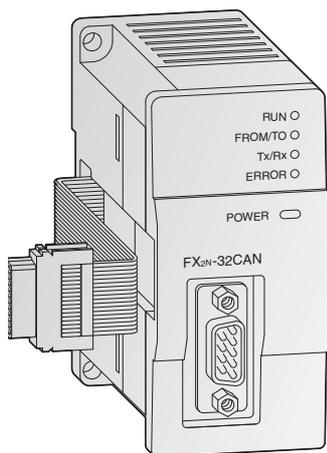
Связь между базовым блоком и внутренней памятью FX2N-64DNET обрабатывается инструкциями FROM/ TO.

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальный модуль	FX2N-64DNET	Ведомый модуль DeviceNet	○	○	●	●

- Специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный функциональный модуль нельзя использовать с этой серией.

2.9.8 Сетевой модуль для CANopen

CANopen – это "открытый" протокол Controller Area Network (CAN), определенный в стандарте EN50325-4. CANopen предлагает экономичные сетевые связи с отказоустойчивой сетевой структурой, где можно быстро и просто интегрировать компоненты от различных производителей. Сети CANopen используются для подключения датчиков, исполнительных механизмов и контроллеров в различных приложениях. Для шины используется недорогие кабели типа витая пара.



Коммуникационный модуль FX2N-32CAN делает возможным подключение ПЛК FX1N, FX2N или FX3U к уже существующей сети CANopen.

Помимо способности функционировать в режиме реального времени и высокой скорости передачи данных до 1Мбит/с, характерной особенностью модуля CANopen является высокая надежность передачи данных и простое конфигурирование сети. До 120 слов данных может отправляться или приниматься как объектов данных процесса (30 PDO). Число слов, которое может быть передано в каждом направлении устанавливается в диапазоне от 1 до 120.

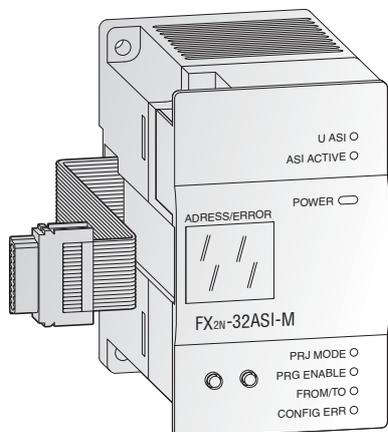
Как и для всех специальных функциональных модулей, связь с внутренней памятью модуля осуществляется с помощью простых инструкций FROM/TO.

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальный модуль	FX2N-32CAN	модуль CANopen	○	●	●	●

- Специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный функциональный модуль нельзя использовать с этой серией.

2.9.9 Сетевой модуль для AS-интерфейса

Интерфейс датчика исполнительного механизма (AS интерфейс или ASi) является международным стандартом для самого низкого уровня полевой шины. Сеть удовлетворяет разнонаправленным требованиям, достаточно гибкая и очень простая в установке. ASi удобен для управления датчиками, исполнительными механизмами и блоками ввода-вывода.



FX2N-32ASI-M служит ведущим модулем для подключения контроллеров FX1N/FX2N и FX3U к системе с AS интерфейсом. Может управлять до 31 ведомыми блоками с максимум 4 входами и 4 выходами.

Для сообщений состояния и диагностики имеется 7-сегментный дисплей.

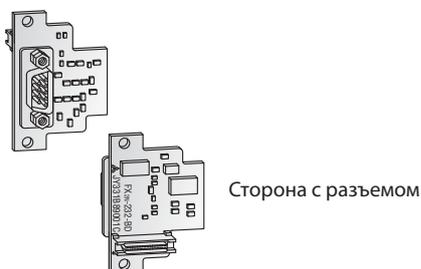
Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Специальный модуль	FX2N-32ASI-M	Ведущий для AS-i системы	○	●	●	●

- Специальный функциональный модуль можно использовать с базовым блоком или устройством расширения этой серии.
- Специальный функциональный модуль нельзя использовать с этой серией.

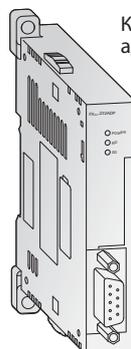
2.9.10 Интерфейсные модули и адаптеры

Для последовательной передачи данных имеется широкий ассортимент интерфейсных модулей/адаптеров. Ниже показаны только некоторые примеры, а в следующей таблице приведены все имеющиеся интерфейсы.

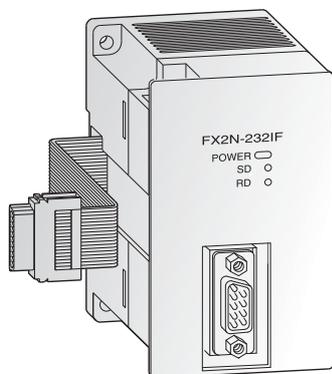
Адаптерная плата интерфейса RS232C FX2N-232-BD



Коммуникационный специальный адаптер FX3U-232ADP (интерфейс RS232C)



Интерфейсный модуль FX2N-232IF



Интерфейсный модуль FX2N-232IF обеспечивает интерфейс RS232C для последовательной передачи данных с MELSEC FX2N, FX2NC с FX3U.

Связь с ПК, принтерами, модемами, сканерами штрих-кодов и т.п. осуществляется программой ПЛК. Переданные и полученные данные хранятся в собственной буферной памяти FX2N-232IF.

Обзор интерфейсных модулей и адаптеров

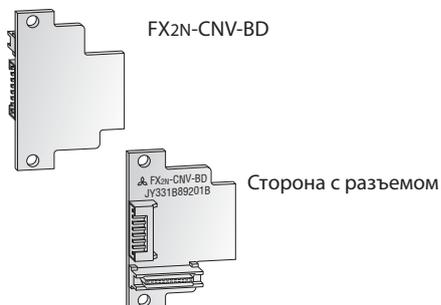
Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Адаптерные платы	FX1N-232-BD	Интерфейсы RS232C	●	●	○	○
	FX2N-232-BD		○	○	●	○
	FX3U-232-BD		○	○	○	●
Специальный адаптер	FX2NC-232ADP*		●	●	●	○
	FX3U-232ADP		○	○	○	●
Специальный модуль	FX2N-232IF		○	○	●	●
Адаптерные платы	FX1N-422-BD		Интерфейсы RS422	●	●	○
	FX2N-422-BD	○		○	●	○
	FX3U-422-BD	○		○	○	●
Адаптерные платы	FX1N-485-BD	Интерфейсы RS485	●	●	○	○
	FX2N-485-BD		○	○	●	○
	FX3U-485-BD		○	○	○	●
Специальный адаптер	FX2NC-485ADP*		●	●	●	○
	FX3U-485ADP		○	○	○	●
Адаптерная плата	FX3U-USB-BD	Интерфейс USB	○	○	○	●

① Для FX2NC-232ADP и FX2NC-485ADP требуется интерфейсный адаптер FX2N-CNV-BD или FX1N-CNV-BD при подключении к базовому блоку FX1S, FX1N или FX2N.

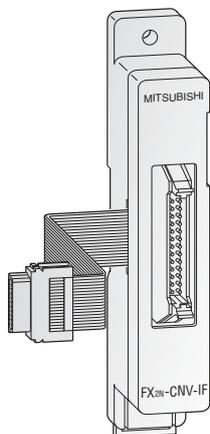
2.9.11 Коммуникационные адаптеры

Платы коммуникационных адаптеров

Коммуникационные адаптеры (FX□□-CNV-BD) устанавливаются непосредственно в базовый блок. Они требуются для подключения специальных адаптеров (FX□□-□□□ADP) с левой стороны базовых блоков.



FX2N-CNV-IF



Интерфейс FX2N-CNV-IF позволяет подключать стандартные блоки расширения и специальные функциональные модули старых серий FX к ПЛК FX2N.

Обзор коммуникационных адаптеров

Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Адаптерные платы	FX1N-CNV-BD	Коммуникационные адаптеры для подключения специальных адаптеров	●	●	○	○
	FX2N-CNV-BD		○	○	●	○
	FX3U-CNV-BD		○	○	○	●
Адаптер	FX2N-CNV-IF	Коммуникационный адаптер для подключения модулей серии FX	○	○	●	○

- Адаптер можно использовать с базовым блоком или устройством расширения этой серии.
- Адаптер нельзя использовать с этой серией.

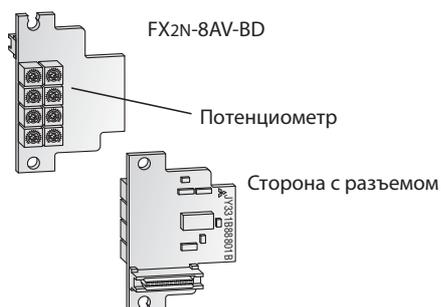
2.9.12 Адаптеры аналогового ввода уставок

Адаптеры аналогового ввода установок позволяют задать 8 аналоговых значений установок. Аналоговые значения (0 – 255) потенциометра считываются в контроллер и используются как установленные по умолчанию значения для таймеров, счетчиков и регистров данных программ ПЛК пользователей.

Значение каждого потенциометра также можно считывать как 11 позиционный поворотный переключатель (позиции 0 – 10).

Опрос заданного значения выполняется в программе ПЛК с помощью специальной инструкции VRRD. Позиция поворотного переключателя считывается с помощью инструкции VRSC.

Адаптеры аналогового ввода устанавливаются в слоте расширения базового блока. Для работы не требуется дополнительного электропитания.



Тип модуля	Обозначение	Описание	FX1S	FX1N	FX2N FX2NC	FX3U
Адаптерные платы	FX1N-8AV-BD	Аналоговые установочные адаптеры	●	●	○	○
	FX2N-8AV-BD		○	○	●	○

- Адаптерную плату можно использовать с базовым блоком или устройством расширения этой серии.
- Адаптерную плату нельзя использовать с этой серией.

2.10 Конфигурация системы

Базовая система ПЛК серии FX может состоять из автономного базового блока, функциональность и диапазон ввода-вывода которого могут быть расширены с помощью модулей ввода-вывода и специальных функциональных модулей. Обзор возможных опций приведен в разделах 2.8 и 2.9.

Базовые блоки

Имеющиеся базовые блоки обладают различными конфигурациями ввода-вывода от 10 до 128 точек, но могут быть расширены до 384 точек, в зависимости от выбранной линейки FX.

Платы расширения

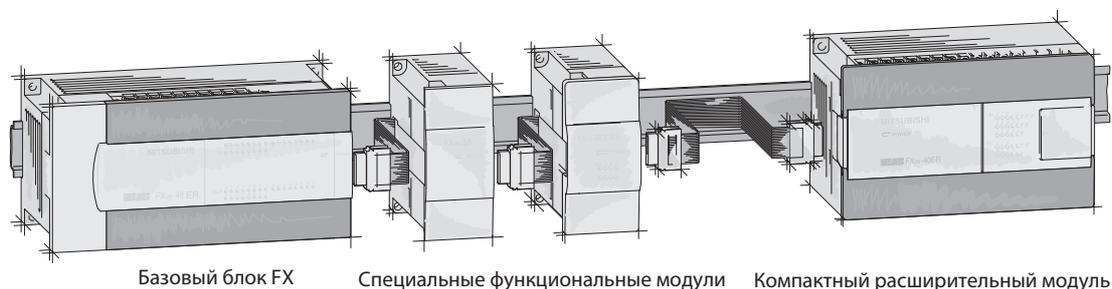
Платы адаптеров расширения могут быть установлены непосредственно в базовый блок, а поэтому не требуют дополнительного места для установки. Для небольшого количества входов-выходов (2 - 4) платы адаптеров расширения можно установить непосредственно в контроллере FX1S или FX1N. Предусмотрены платы адаптеров для интерфейсов, служащие для оснащения ПЛК серии FX дополнительными интерфейсами RS232 или RS485.

Расширительные модули ввода-вывода

Модульные блоки расширения без блоков питания и компактные расширительные модули с блоками питания могут подключаться к ПЛК FX1N, FX2N и FX3U. Для модульных блоков расширения с питанием от базового блока, необходимо рассчитать потребление мощности, так как шина постоянного напряжения 5 В может поддерживать только ограниченное количество входов-выходов расширения.

Специальные функциональные модули / специальные адаптеры

Для ПЛК FX1N, FX2N и FX3U имеется широкий ряд специальных функциональных модулей. Эти модули обеспечивают сетевую функциональность, управление аналоговыми значениями, импульсные выходы и температурные входы (дополнительную информацию см. в разделе 2.9).



Возможности расширения

ПЛК	Количество модулей слева от базового блока	Количество плат в порте для плат расширения базового блока	Количество модулей справа от базового блока
FX1S	Модули FX0N-485ADP и FX0N-232ADP можно устанавливать в комбинации с коммуникационным адаптером FX1N-CNV-BD.	1 (код продукта FX□□-□□□-BD)	—
FX1N			До 2 специальных функциональных модулей серии FX2N.
FX2N			До 8 специальных функциональных модулей серии FX2N.
FX2NC	Модули FX0N-485ADP и FX0N-232ADP можно установить непосредственно слева. Адаптер не требуется.		До 4 специальных функциональных модулей серии FX2N.
FX3U	До 10 адаптеров серии FX3U можно непосредственно установить с левой стороны базового блока.		До 8 специальных функциональных модулей серии FX2N или FX3U.

Ниже приведены различия между базовым блоком, расширительным модулем и блоком расширения:

- Базовый блок включает до 4 компонентов, т.е. блок питания, входы, выходы и ЦП.
- Расширительный модуль включает 3 компонента, т.е. блок питания, входы и выходы.
- Блок расширения включает 1 или 2 компонента, т.е. входы и/или выходы.

Из этого видно, что блок расширения не имеет блока питания. Поэтому он получает требуемое питание либо от базового блока, либо от расширительного модуля.

Следовательно, необходимо определить, какое количество блоков без питания можно подключить, не превышая нагрузочную способность 'встроенного' блока питания.

2.10.1 Подключение специальных адаптеров (только для FX3U)

До 10 специальных адаптеров можно установить непосредственно с левой стороны базового блока FX3U. Соблюдайте следующие правила.

Специальные адаптеры высокоскоростного ввода/вывода

К базовому блоку можно подключить до двух специальных адаптеров высокоскоростного ввода FX3U-4HSX-ADP и до двух специальных адаптеров высокоскоростного вывода FX3U-2HSY-ADP.

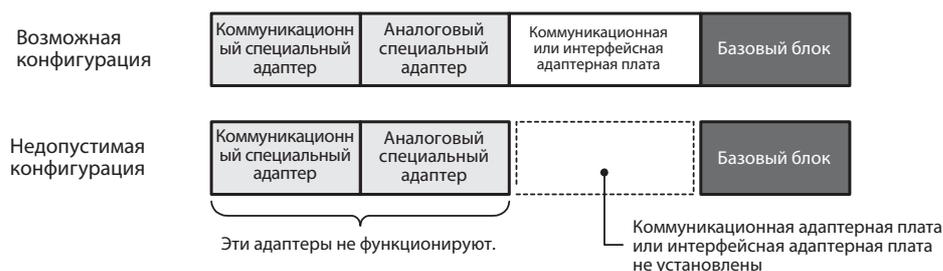
Подключите все специальные адаптеры высокоскоростного ввода/вывода до подключения других специальных адаптеров, когда они используются в комбинации. Специальный адаптер высокоскоростного ввода/вывода можно устанавливать слева от коммуникационного или аналогового специального адаптера.

Когда подключаются только специальные адаптеры высокоскоростного ввода/вывода, адаптеры можно использовать без установки коммуникационной или интерфейсной адаптерной платы в базовом блоке.



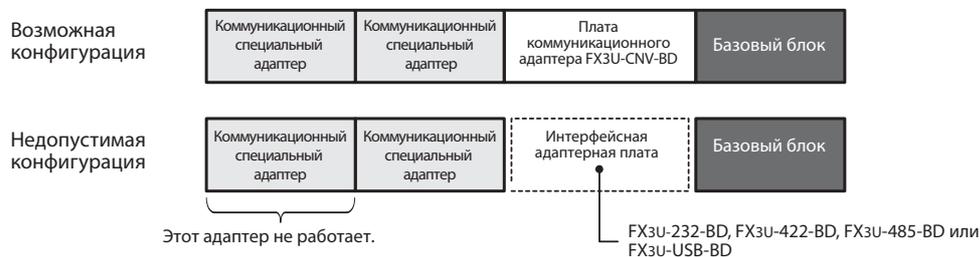
Комбинация аналоговых и коммуникационных специальных адаптеров

Аналоговые и коммуникационные специальные адаптеры должны использоваться с установленными в базовом блоке коммуникационной адаптерной платой или интерфейсной адаптерной платой.



Комбинация коммуникационных специальных адаптеров и интерфейсной адаптерной платы

Когда вместо коммуникационной адаптерной платы FX3U-CNV-BD устанавливается интерфейсная адаптерная плата FX3U-232-BD, FX3U-422-BD, FX3U-485-BD, или FX3U-USB-BD, может использоваться один коммуникационный специальный адаптер FX3U-232ADP или FX3U-485ADP.



Комбинация высокоскоростных ввода/вывода, аналоговых и коммуникационных специальных адаптеров

При использовании этих адаптеров установите специальные адаптеры высокоскоростного ввода/вывода слева от базового блока. Специальные адаптеры высокоскоростного ввода/вывода нельзя устанавливать после коммуникационного/аналогового специального адаптера.



Сводка

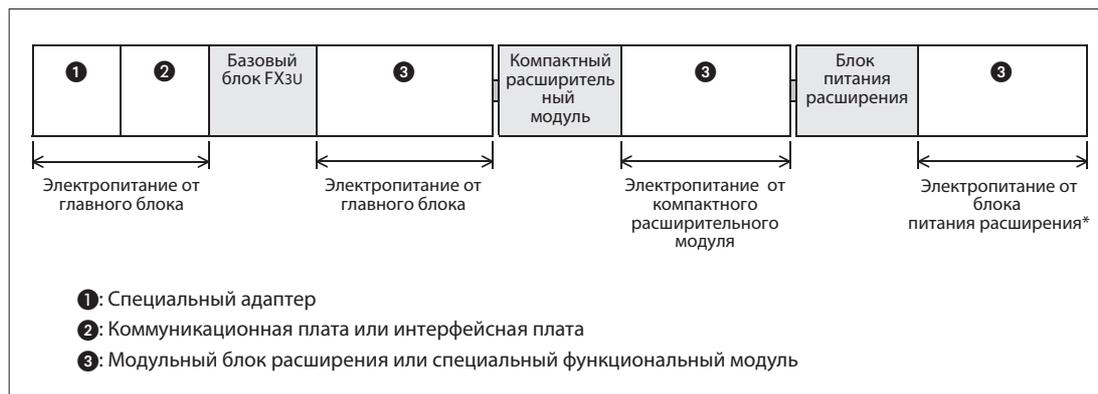
Установленная коммуникационная адаптерная плата или интерфейсная адаптерная плата	Количество подключаемых специальных адаптеров			
	Коммуникационный специальный адаптер	Аналоговый специальный адаптер	Специальный адаптер высокоскоростного ввода	Специальный адаптер высокоскоростного вывода
Адаптерная плата не установлена	Эти специальные адаптеры подключить нельзя.			2
FX3U-CNV-BD	2	4	2	2
FX3U-232-BD FX3U-422-BD FX3U-485-BD FX3U-USB-BD	1	4	2	2

2.10.2 Базовые правила конфигурации системы

При конфигурировании системы с расширительными модулями или специальными функциональными модулями необходимо учитывать следующие соображения:

- Потребляемый ток от задней шины 5 В пост. тока
- Потребляемый ток 24 В пост. тока
- Общее количество точек ввода и вывода не должно превышать максимальное количество вводов-выводов.

На следующем рисунке показано распределение питания на примере FX3U.



* Когда блок расширения входов подключается после блока питания расширения, этот блок расширения входов питается от базового блока или модуля расширения входов/выходов с блоком питания, который устанавливается между базовым блоком и блоком питания расширения.

Расчет энергопотребления

Питание подается на каждое подсоединенное устройство от встроенного блока питания главного блока, модуля расширения входов/выходов с блоком питания или - только для FX3U - блока питания расширения.

Имеется три типа встроенных источников питания

- 5 В пост. тока
- 24 В пост. тока (для внутреннего использования)
- Шина сервисного питания 24 В пост. тока (только в базовых блоках с питанием от переменного тока).

В следующей таблице приведена нагрузочная способность встроенных источников питания:

Модель		Встроенный блок питания 5 В пост. тока	Встроенный блок питания 24 В пост. тока (источник внутреннего / сервисного напряжения)
Базовые блоки	FX1N	Подходит для питания всех подключенных модулей	400 мА
	FX2N	290 мА	250 мА (FX2N-16M□, FX2N-32M□) 460 мА (все другие базовые блоки)
	FX3U	500 мА	400 мА (FX3U-16M□, FX3U-32M□) 600 мА (все другие базовые блоки)
Компактный расширительный модуль	FX2N	690 мА	250 мА (FX2N-32E□) 460 мА (FX2N-48E□)

Когда добавляются только блоки расширения ввода-вывода, можно использовать краткую справочную таблицу.

Когда добавляются также специальные функциональные модули, вычислите потребляемый ток, чтобы проверить, что встроенный блок питания может подавать полный ток, потребляемый дополнительными модулями. Более подробную информацию об энергопотреблении см. в разделе А.4.

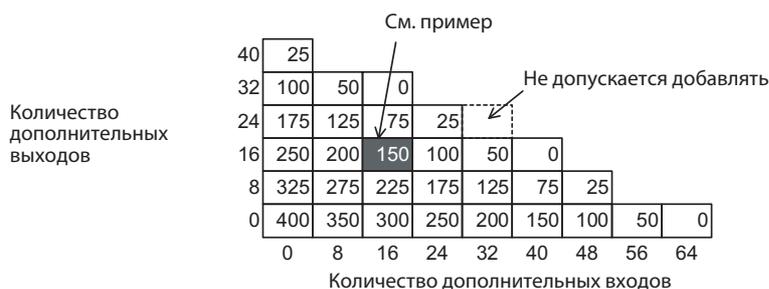
2.10.3 Краткие справочные таблицы

Когда к базовому блоку добавляются только блоки расширения ввода-вывода без встроенного питания, можно использовать краткую справочную таблицу. Следующие примеры допустимы для базовых блоков серии FX3U.

Базовые блоки, питаемые от переменного тока

В следующих кратких справочных таблицах, значение на пересечении количества добавляемых входных точек (горизонтальная ось) с количеством добавляемых выходных точек (вертикальная ось) указывает остаточную емкость блока питания.

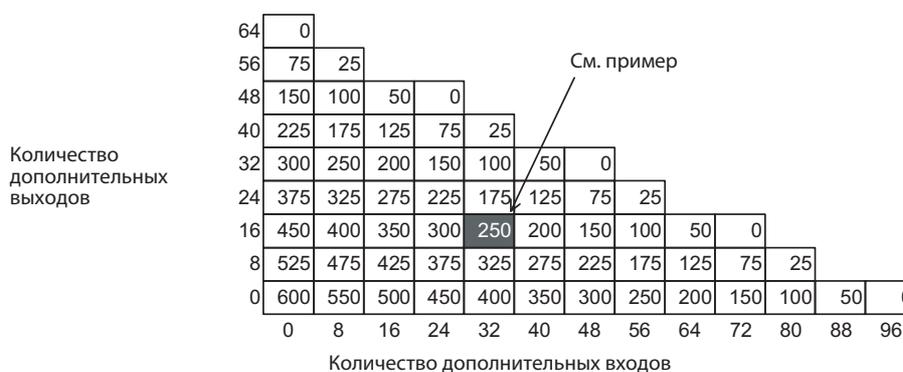
Для FX3U-16MR/ES, FX3U-16MT/ES, FX3U-16MT/ESS, FX3U-32MR/ES, FX3U-32MT/ES или FX3U-32MT/ESS:



● Пример

Когда блок расширения, содержащий 16 точек ввода и 16 точек вывода подключается к базовому блоку FX3U-16M□ или FX3U-32M□, остаточный ток из шины питания 24 В пост. тока составляет 150 мА.

Для FX3U-48MR/ES, FX3U-48MT/ES, FX3U-48MT/ESS, FX3U-64MR/ES, FX3U-64MT/ES, FX3U-64MT/ESS, FX3U-80MR/ES, FX3U-80MT/ES, FX3U-80MT/ESS, FX3U-128MR/ES, FX3U-128MT/ES или FX3U-128MT/ESS:



● Пример

Когда блок расширения, содержащий 32 точек ввода и 16 точек вывода, подключается к базовому блоку с собственным питанием переменного тока на 48, 64, 80 или 128 входов-выходов, шина питания 24 В пост. тока может обеспечить еще максимальный ток 250 мА на другие устройства.

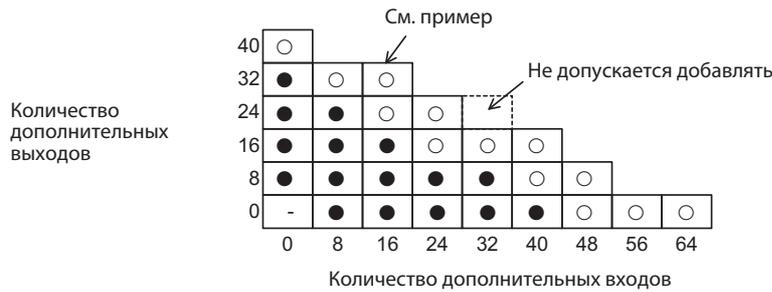
Подтвердите допустимый ток шины питания 24 В пост. тока от значения, показанного в краткой справочной таблице. Эта остаточная нагрузочная способность блока питания (ток) может быть использована для питания внешних нагрузок (датчиков или т.п.). Когда подключаются специальные функциональные модули, необходимо учитывать, могут ли они питаться от остаточной нагрузочной способности блока питания.

Базовые блоки, питаемые от постоянного тока

Главные блоки с питанием от постоянного тока имеют ограничения по расширяемым точкам ввода-вывода поскольку они нуждаются во встроенных источниках сервисного напряжения.

В следующих таблицах показаны расширяемые блоки до знака ○, где желаемые входы (горизонтальная ось) и выходы (вертикальная ось) пересекаются. Система расширяется до знака ● когда напряжение питания составляет 16.8 В - 19.2 В.

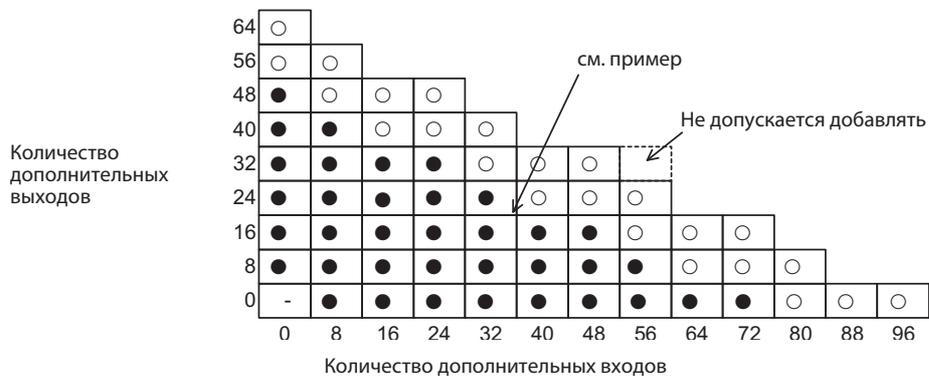
Для FX3U-16MR/DS, FX3U-16MT/DS, FX3U-16MT/DSS, FX3U-32MR/DS, FX3U-32MT/DS или FX3U-32MT/DSS:



● Пример

При добавлении 16 входов к базовому блоку с собственным питанием постоянного тока с 16 или 32 входами-выходами, расширяется максимум 32 выхода. Когда добавляется 16 входов при напряжении питания 16.8 В - 19.2 В, расширяется максимум 16 выходов.

Для FX3U-48MR/DS, FX3U-48MT/DS, FX3U-48MT/DSS, FX3U-64MR/DS, FX3U-64MT/DS, FX3U-64MT/DSS, FX3U-80MR/DS, FX3U-80MT/DS или FX3U-80MT/DSS:



● Пример

При добавлении 32 входов к базовому блоку с собственным питанием постоянного тока с 48, 64 или 80 входами-выходами, расширяется максимум 40 выходов. Когда добавляется 32 входов при напряжении питания 16.8 В - 19.2 В, расширяется максимум 24 выхода.

2.11 Назначение входов-выходов

Назначение входов и выходов в ПЛК семейства MELSEC FX фиксировано и не может изменяться.

Когда после установки модулей/блоков расширения с питаемыми входами-выходами подается питание, главный блок автоматически присваивает номера входов-выходов (X/Y) модулям/блокам.

Поэтому не требуется задавать номера входов-выходов с помощью параметров.

Специальным функциональным блокам/модулям номера входов-выходов не назначаются.

2.11.1 Концепция назначения

Номера входов-выходов (X/Y) представляются в восьмеричном формате

Входы и выходы ПЛК семейства MELSEC FX подсчитываются в восьмеричной системе счисления. Это числовая система с основанием 8, в ней используются цифры от 0 до 7.

В следующей таблице приведено сравнение некоторых десятичных и восьмеричных чисел:

Десятичное	Восьмеричное
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20
:	:

Восьмеричные числа назначаются как номера входов-выходов (X/Y), как показано ниже.

- X000 - X007, X010 - X017, X020 - X027....., X070 - X077, X100 - X107...
- Y000 - Y007, Y010 - Y017, Y020 - Y027....., Y070 - Y077, Y100 - Y107...

Номера для добавленных модулей/блоков ввода-вывода

В добавленном расширительном модуле/блоке ввода-вывода с собственным блоком питания назначаются номера входов и номера выходов, следующие за номерами входов и номерами выходов, данными в предыдущем устройстве. Последняя цифра присвоенного номера должна начинаться с 0.

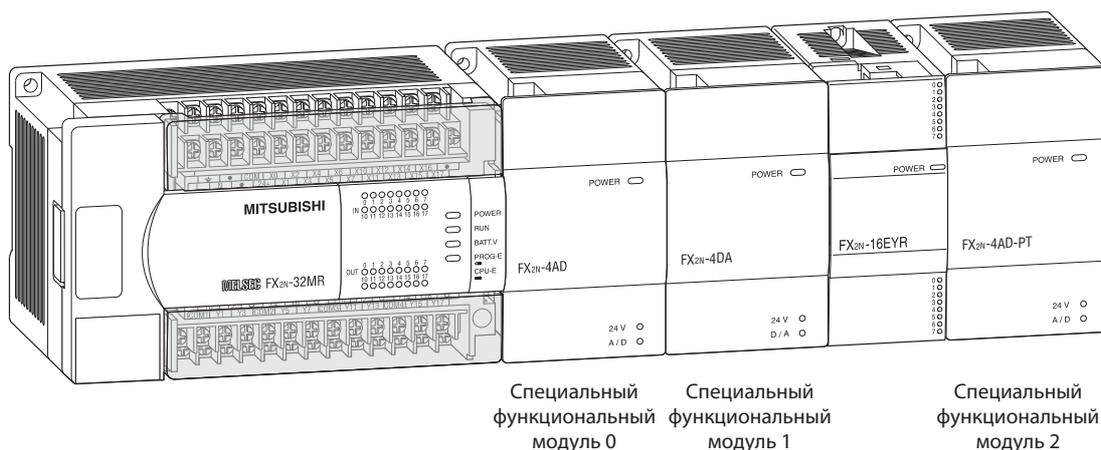
Например, когда последний номер на предыдущем устройстве Y43, номер выхода, назначенный следующему устройству, начинается с Y50.



* Входы от X044 до X047 и выходы от Y024 до Y027 заняты FX2N-8ER-ES/UL, но они не могут использоваться.

2.11.2 Адрес специального функционального модуля

Поскольку к одному базовому блоку можно подключать несколько специальных функциональных модулей, каждый модуль должен иметь уникальный идентификатор, чтобы к нему можно было обращаться для передачи или считывания данных. Для этого каждый специальный модуль автоматически получает номер из диапазона от 0 до 7 (к контроллеру можно подключить максимум 8 специальных модулей). Номера присваиваются непрерывно, и нумерация начинается с модуля, который первым подключается к контроллеру.



В следующих продуктах адреса специальных функциональных модулей **не** назначаются:

- Блоки расширения ввода-вывода с собственным блоком питания (например, FX2N-32ER-ES/UL или FX2N-48ET-ESS/UL)
- Блоки расширения ввода-вывода (например, FX2N-16EX-ES/UL или FX2N-16EYR-ES/UL)
- Коммуникационный адаптер (например, FX3U-CNV-BD)
- Интерфейсный адаптер (например, FX3U-232-BD)
- Специальный адаптер (например, FX3U-232ADP)
- Блок питания расширения FX3U-1PSU-5V

3 Программирование

3.1 Концепции стандарта IEC61131-3

IEC 61131-3 - международный стандарт для программ ПЛК, установленный Международной электромеханической комиссией (IEC). Он определяет языки программирования и структурирующие элементы, которые используются для написания программ ПЛК.

Эта система позволяет создавать структурированные программы, используя высокую степень модуляризации. Такой подход повышает эффективность, позволяя повторно использовать проверенные программы и подпрограммы, и уменьшает количество ошибок программирования.

Благодаря методам структурного программирования IEC61131-3 облегчает процедуры поиска ошибок, позволяя независимо проверять операционные элементы программы.

Одно из важных преимуществ IEC61131-3 заключается в том, что он помогает процедурам управления проектом и контроля качества. В частности, структурные методы IEC61131-3 помогают внедрению процессов валидации в ПЛК. Фактически, в некоторых отраслях промышленности сегодня считается обязательным использовать этот подход структурного программирования. Он повсеместно применяется в фармацевтической и нефтехимической промышленности, где некоторые процессы могут рассматриваться как критические для обеспечения безопасности.

Иногда считают, что метод программирования IEC требует дополнительной работы для создания конечного кода. Однако общепринято, что преимущества структурированного подхода над "неструктурированными" и "открытыми" методами программирования обеспечивают IEC61131-3 заслуживающее внимания преимущество.

PLCopen



PLCopen - независимая организация поставщиков и производителей, которая была создана для продвижения и дальнейшего использования IEC61131-3 пользователями систем управления производственными процессами. Эта организация определила 3 уровня соответствия конструкции и реализации систем требованиям IEC61131-3.

Организацией PLCopen были установлены:

- процедура аккредитации
- аккредитованные испытательные организации
- разработка тестирующего программного обеспечения, доступного организациям-членам
- определенная процедура сертификации
- члены с сертифицированными продуктами

Это обеспечивает соответствие на текущий момент и в будущем.

Сертификация PLCopen



61131-3



Пакет GX-IEC Developer от Mitsubishi полностью совместим с "**Базовым уровнем IL**" (списком инструкций) и "**Базовым уровнем ST**" (языком структурированного текста) PLCopen и полностью сертифицирован на соответствие этим стандартам.

3.2 Структура программного обеспечения и определение терминов

В следующем разделе будут определены основные термины, используемые в GX IEC Developer :

- ПРОГРАММНЫЕ МОДУЛИ (POU)
- ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ
- ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ
- ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ И ФУНКЦИОНАЛЬНЫЕ БЛОКИ
- ЗАДАЧНЫЙ ПУЛ
- РЕДАКТОРЫ ТЕКСТОВ ПРОГРАММ:
 - Список инструкций
 - Лестничная диаграмма
 - Функциональная блок-схема
 - Последовательная функциональная схема
 - Структурированный текст
 - Список инструкций MELSEC

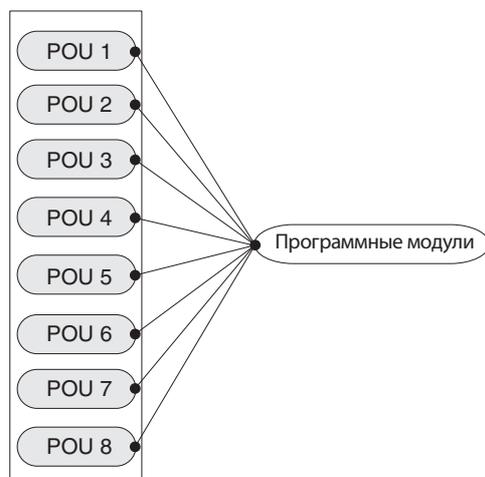
3.2.1 Определение терминов в IEC61131-3

Проекты

Проект содержит программы, документацию и параметры, необходимые для приложения.

POU - Программный модуль

Подход структурного программирования заменяет прежнюю неупорядоченную совокупность отдельных инструкций на четкую организацию программы в программных модулях. Эти модули называются программными модулями (ПМ); они формируют основание нового подхода к программированию систем ПЛК.



Программные модули (ПМ) используются для решения всех задач программирования.

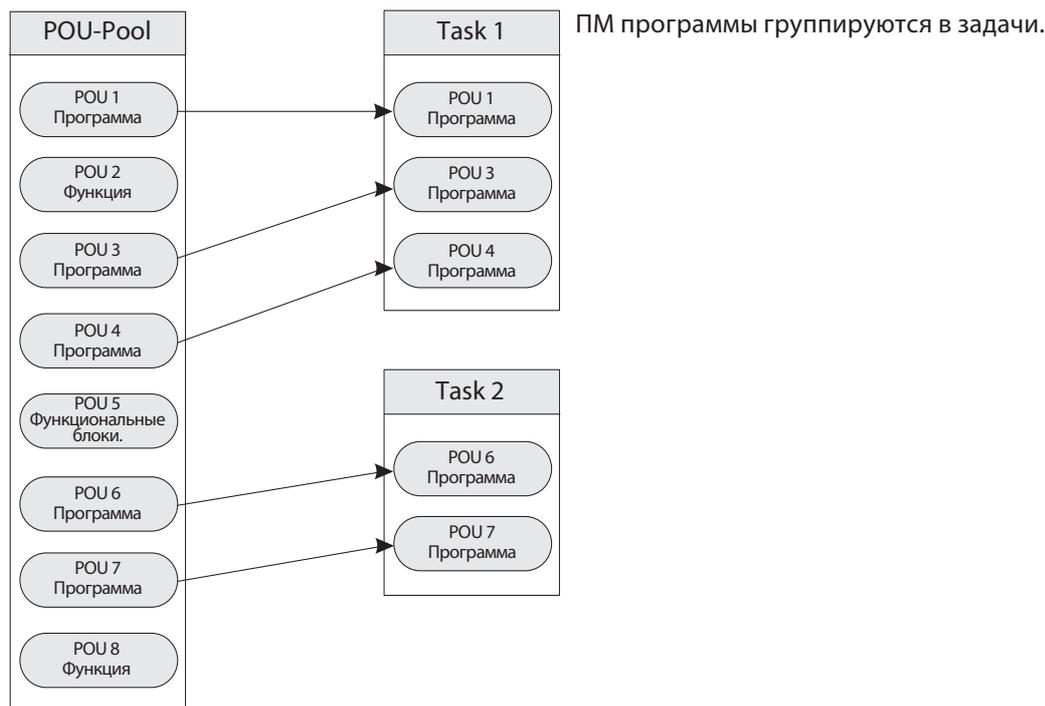
Имеются три различных класса ПМ, которые классифицируются на основе их функциональности:

- Программы
- Функции
- Функциональные блоки

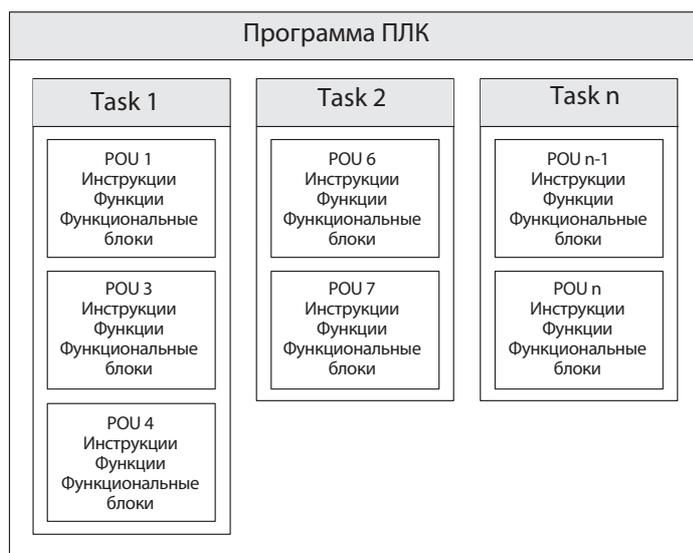
ПМ, объявленные как функциональные блоки, могут рассматриваться как команды программирования в их собственном смысле; их также можно использовать в каждом модуле ваших программ.

Конечная программа компилируется из ПМ, которые вы определяете как программы. Этот процесс выполняется управлением Задачами в задачном пуле. ПМ программы объединяются в группы, называемые "Задачами".

Задачи



В свою очередь все задачи группируются, образуя реальную программу ПЛК.



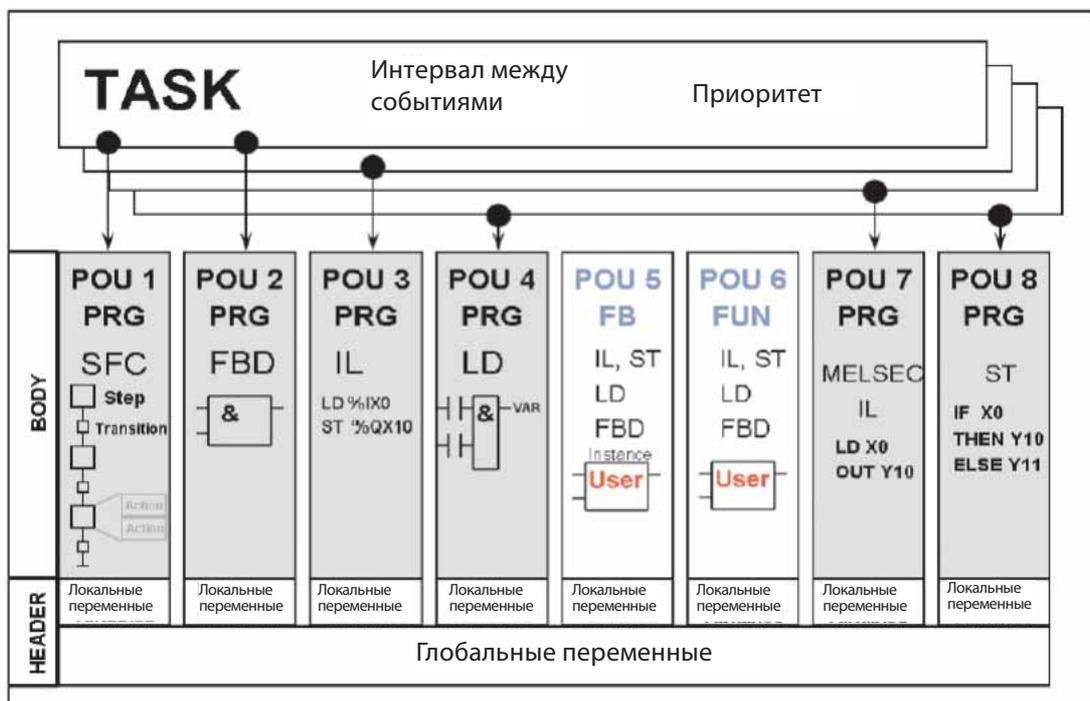
Большинство программ ПЛК состоит из областей кода, которые выполняют специфические задачи. Они могут формировать часть одной большой программы или быть написанными в виде подпрограмм с инструкциями управления программой, позволяющими выбирать текущую подпрограмму, например, CALL, CJ и т.д.



В приведенной выше программе GX IEC Developer полагает, что каждая подпрограмма в программе которая выполняет специфическую задачу, является ПМ или программным модулем.

Каждый ПМ можно записать, используя любой из поддерживаемых редакторов, т.е. LD, IL, FBD, SFC, ST, как показано ниже:

Полная конфигурация проекта, иллюстрирующая интеграцию программных модулей с использованием SFC, FBD, IL, LD и MELSEC IL и ST форматов программ.

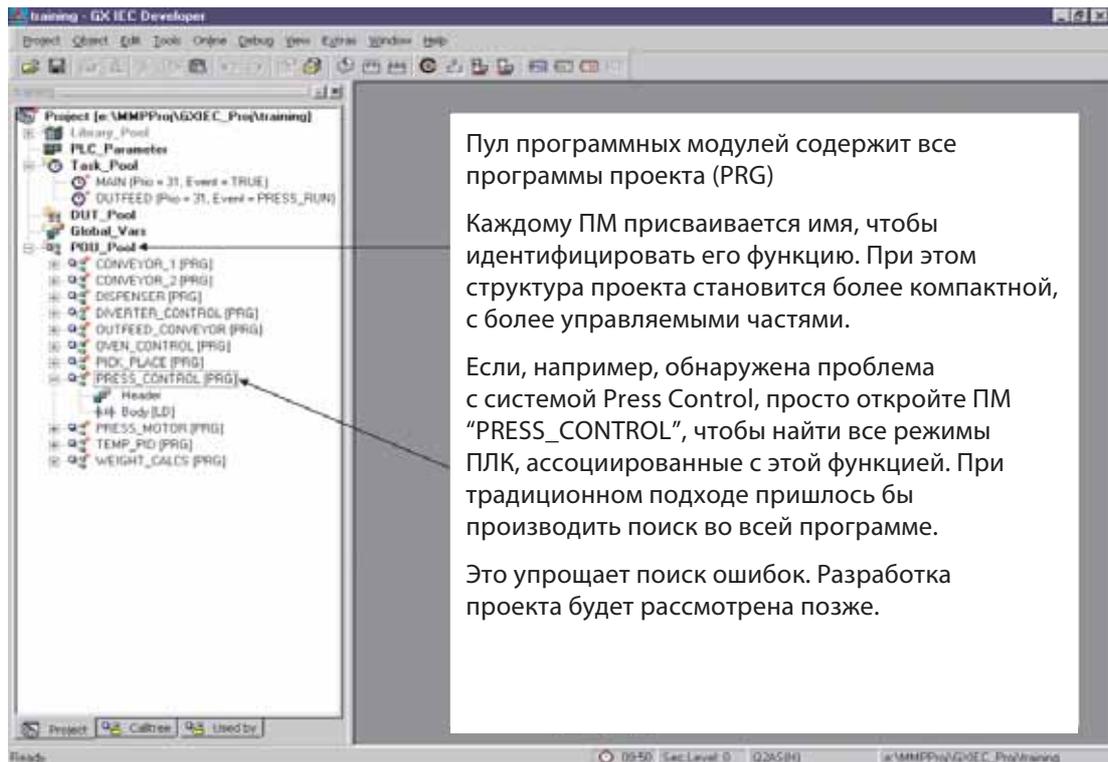


Пул программных модулей

Проект будет состоять из многих ПМ, каждый из которых обеспечивает специализированную функцию управления и содержится в пуле программных модулей. Каждый ПМ может быть написан в любом из редакторов IEC. Поэтому в любом конкретном проекте можно выбрать наилучший язык для необходимой функции. Компилятор транслирует проект в код, который может понять ПЛК, но интерфейс пользователя остается в написанном виде.

Это позволяет писать сложные взаимодействующие подпрограммы в виде лестничных ПМ, а для сложных вычислений или алгоритмов может лучше подойти один из текстовых редакторов или редакторов блоков описания файлов.

Эта среда обеспечивает гибкость, расширяя выбор разработчика/пользователя.



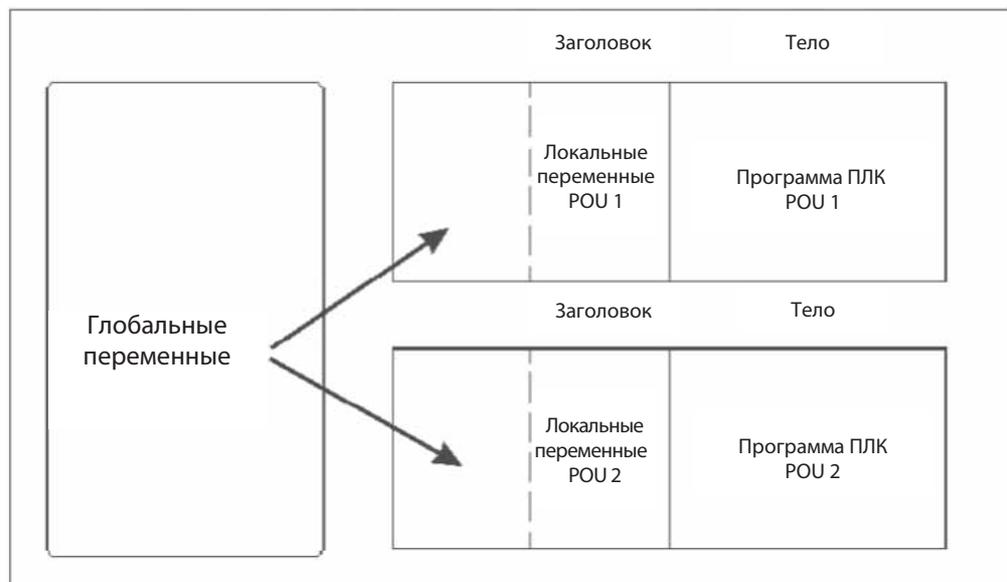
Состав ПМ

Каждый программный модуль (ПМ) состоит из:

- Заголовка и
- тела

Переменные, которые используются в ПМ, объявляются в заголовке.

Тело содержит фактическую программу ПЛК, написанную на одном из различных языков.



Определение переменных - ГЛОБАЛЬНЫХ и ЛОКАЛЬНЫХ

- Переменные

Перед созданием программы необходимо решить, какие переменные потребуются в каждом конкретном программном модуле. Каждый ПМ содержит список локальных переменных, которые определяются и объявляются для использования только в конкретном ПМ. Глобальные переменные можно использовать во всех ПМ в программе; они объявляются в отдельном списке.

- Локальные переменные

Когда элементы программы объявлены как локальные переменные, GX IEC Developer автоматически использует некоторые из своих системных переменных, как соответствующие операнды хранения в конкретном ПМ. Эти переменные используются исключительно каждым ПМ и недоступны для любой другой подпрограммы в проекте.

- Глобальные переменные

Глобальные переменные можно рассматривать как переменные "общего доступа"; они являются интерфейсом к физическим устройствам ПЛК. Они доступны всем ПМ и ссылаются на реальный физический вход-выход ПЛК или указанные внутренние устройства в ПЛК. Внешние устройства HMI и SCADA могут взаимодействовать с пользовательской программой, используя глобальные переменные.

Переменные IEC61131-3 и переменные MELSEC

GX IEC Developer поддерживает создание программ с использованием как объявления символических переменных (символических имен), так и абсолютных адресов Mitsubishi (X0, M0 и т.д.), назначенных элементам программы.

Использование объявления символических переменных соответствует IEC 61131-3.

Если используется объявление символических переменных, то затем для имен тегов должны задаваться перекрестные ссылки на реальные адреса ПЛК.

Список локальных переменных

Чтобы конкретный ПМ мог получить доступ к глобальной переменной, она должна быть объявлена в его списке локальных переменных (LVL), в заголовке ПМ.

LVL может включать как глобальные переменные, так и локальные переменные.

Локальную переменную можно рассматривать как промежуточный результат, то есть если программа выполняет вычисления в пять стадий, используя три значения и заканчивая выдачей одного результата, то при традиционном подходе программист создал бы программу, вырабатывающую несколько промежуточных результатов, хранящихся в регистрах данных перед тем, как результат будет сохранен в регистре конечного результата.

Вероятно, что эти промежуточные результаты не используются ни для какой цели, а только сохраняются, и только окончательный результат используется в другом месте.

В GX IEC Developer промежуточные результаты могут быть объявлены как локальные переменные и в этом случае, только исходные три числа и результат объявляются как глобальные переменные.

Список глобальных переменных

Список глобальных переменных (GVL) предоставляет интерфейс для всех имен, которые относятся к реальным адресам ПЛК, т.е. регистры данных ввода-вывода и т.д.

GVL имеется и может быть считан всеми ПМ созданными в проекте.

Задачный пул и диспетчер задач

Если мы теперь рассматриваем свои подпрограммы как ПМ, написанные для каждой функции и заданного имени, мы можем создать задачу для каждого из назначенных ПМ.

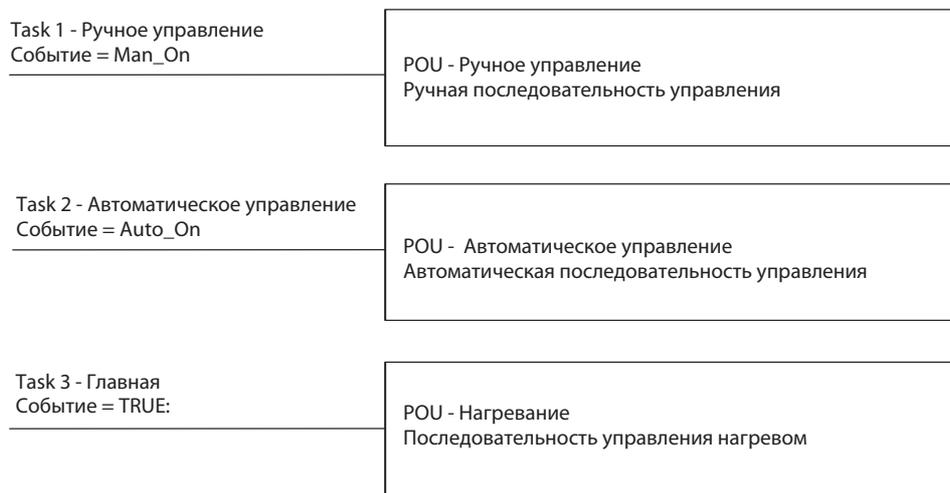
У каждой задачи могут быть различные рабочие состояния, или события.

- Событие: обращение к заданию, например, при прерывании или в процессе выполнения цикла (TRUE)
- Интервал: выполнение через заданные промежутки времени
- Приоритет: определение последовательности обработки заданий

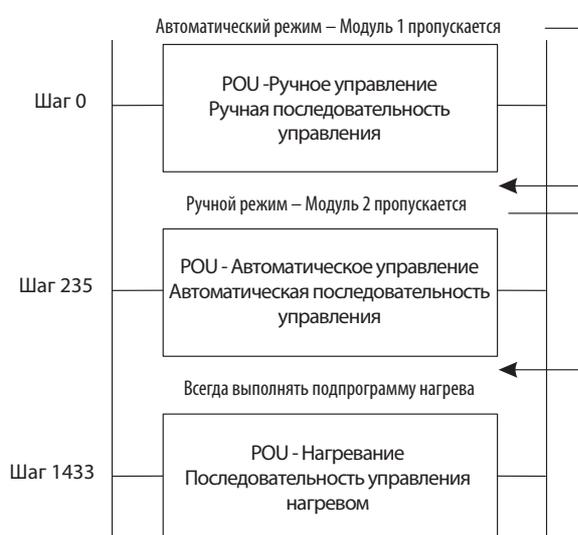
На следующем рисунке показаны различные настройки условий выполнения для трех заданий:

- Task 1 выполняется, только когда тег под названием "Man_On" является истинным.
- Task 2 выполняется, только когда тег под названием "Auto_On" является истинным.
- Task 3 выполняется все время (это обозначается как Event = True).

Эти имена тегов могут быть объявлены как глобальные переменные и присвоены битовым операндам ПЛК (они могут быть адресами, например, X0).



Для наглядности условий выполнения можно сравнить изображенную программу с программой, в которой все операторы расположены друг под другом, однако определенные участки программы пропускаются.



Инструкции перехода (например, CJ) используются, чтобы пропустить программные модули 1 и 2, когда они не используются. Подпрограмма управления нагревом всегда должна выполняться.

Программные модули 1 и 2 можно сравнить с событийно управляемыми заданиями, в то время, как программный модуль 3 обрабатывается всегда (событие = TRUE, см. выше).

Когда GX IEC Developer транслирует программу в машинный код, он вставляет в код фактические инструкции перехода, чтобы соответствовать заданным условиям выполнения.

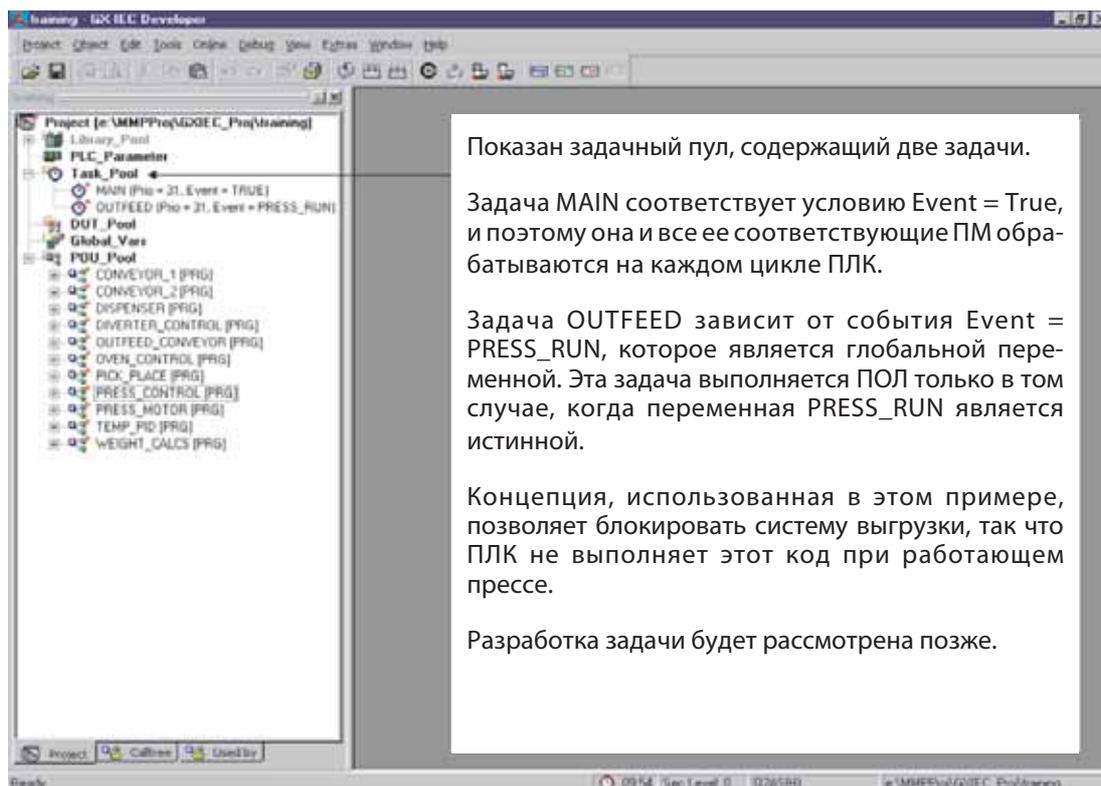
Когда GX IEC Developer компилирует проект, он автоматически вставляет в программу команды ветвления программы в соответствии с управляемыми событиями задачами.

Задаче может быть присвоен больше, чем один ПМ. Обычно задача, где Событие = Истинно, должна содержать все ПМ, которые необходимо выполнять каждый цикл ПЛК. ПМ с конкретным именем может быть назначен только в одной задаче в любом проекте.

ПРИМЕЧАНИЕ

Любой ПМ, не присвоенный задачам, НЕ ДОЛЖЕН ПЕРЕСЫЛАТЬСЯ В ПЛК при передаче программы. Не забывайте - это относится к загрузке по умолчанию. Задачи могут быть расположены по приоритетам - по времени или на основании прерываний.

Задачный пул содержит все назначенные задачи в проекте.



Диспетчер задач позволяет пользователю эффективно управлять циклом ПЛК, гарантируя, что будут выполняться только подпрограммы, которые требуются в цикле. Он также обеспечивает простой метод присвоения конкретных подпрограмм событиям, а также временным или приоритетным прерываниям.

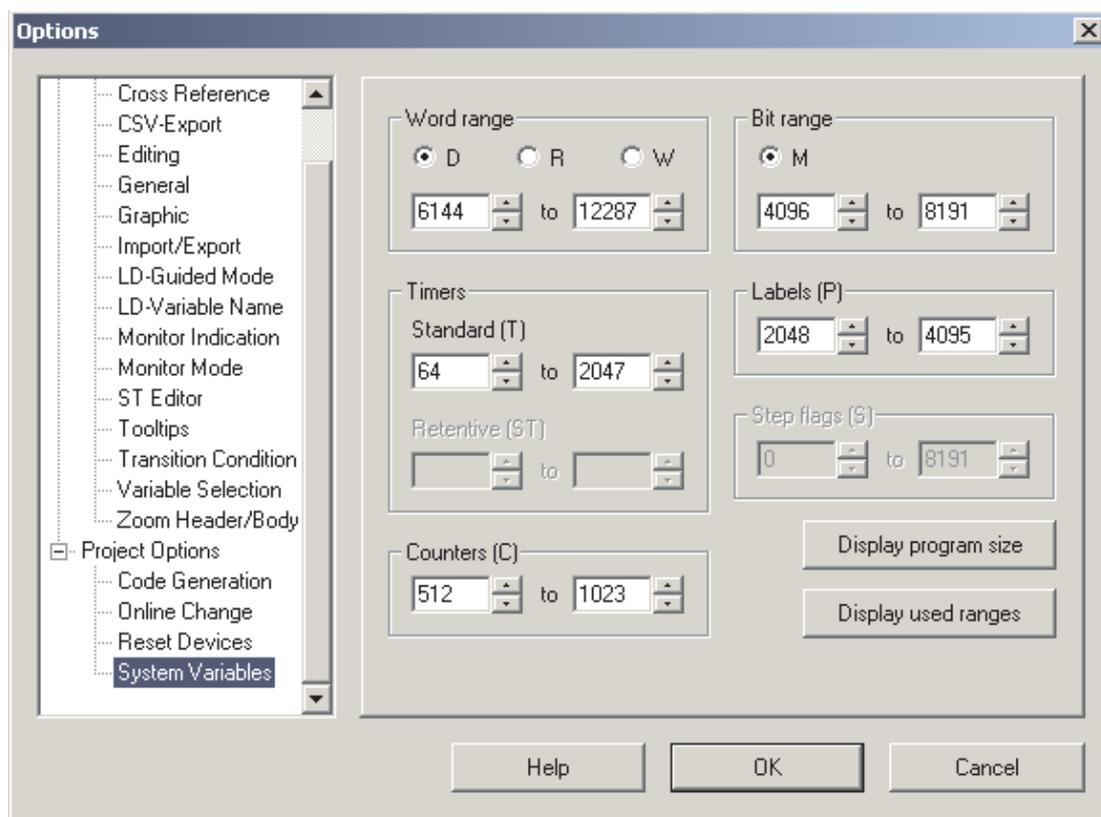
При этом программный инженер должен думать только о содержании программы, а не о том, правильны ли команды перехода и согласуются ли они с правилами.

У станков/процессов, составленных из стандартных частей, может быть отдельный ПМ, написанный для каждой части. Полный станок может состоять из многих ПМ.

Для каждого варианта станка поставщик может присваивать администратору задач только ПМ, уместные для данного станка, поскольку только присвоенные ПМ будут переданы в ПЛК при загрузке.

3.2.2 Системные переменные

Здесь можно редактировать диапазоны операндов, которые GX IEC Developer назначил системным переменным. Для этого используйте команду **Options** в меню **Extras**:



Диапазоны системных переменных для реального проекта. Доступны, если открыт Q/QnA проект.

- **Word range** - Словная область
 - D: D-операнды используются как словные системные переменные.
 - R: R-операнды используются как словные системные переменные.
 - W: W-операнды используются как словные системные переменные.
 - Диапазон: В зависимости от типа ПЛК, как определено в параметрах.
- **Timers** - Таймеры
 - Standard (T) - Стандартные (T) - Диапазон: В зависимости от типа ПЛК, как определено в параметрах.
 - Retentive (ST) - Фиксируемые - Диапазон: В зависимости от типа ПЛК, как определено в параметрах.
- **Counters (C)** - Счетчики
 - Диапазон: В зависимости от типа ПЛК, как определено в параметрах.
- Битовая область
 - M: M-операнды используются как битовые системные переменные.
 - Диапазон: В зависимости от типа ПЛК, как определено в параметрах.

- **Метки (P)**

Диапазон: В зависимости от типа ПЛК, как определено в соответствующем CNF-файле.

- **Step flags (S) - Флаги шагов**

Диапазон: В зависимости от типа ПЛК, как определено в соответствующем TYP-файле.

- **Display program size - Отображение размера программы**

Сводка используемого размера программы, показанная в отдельном диалоговом окне. Если программа не компилирована, в диалоговом окне вместо размера программы показан символ "?". Если для этого ЦП недоступны SFC или SUB программы, соответствующая опция будет недоступна.

- **Display used ranges - Отображение используемых диапазонов**

Сводка используемых диапазонов системных переменных, показанная в отдельном диалоговом окне.



3.2.3

Системные метки

Системные метки, показанные в списке системных переменных в главе 3.2.2, используются GX IEC Developer для внутреннего управления проектом. GX IEC Developer снабжает системными метками следующие объекты:

- Сетевые метки
- Задача, управляемые событиями (не "EVENT = TRUE")
- Пользовательские функциональные блоки (одну на функциональный блок - за исключением макро-кода)
- Системные таймеры (используются диспетчером задач для интервальных задач и локальных таймеров.)

3.3 Языки программирования

GX IEC Developer предоставляет отдельные редакторы для следующих языков программирования, которые могут использоваться для программирования тела ваших программ:

Текстовые редакторы

- Список инструкций (IEC и MELSEC)
- Язык структурированного текста

Графические редакторы

- Релейная диаграмма
- Функциональная блок-схема
- Последовательная функциональная схема

За исключением языка последовательных функциональных схем, все редакторы делят программы ПЛК на секции, которые называются "цепями". Этим цепям могут назначаться имена (метки), которые могут включать максимум 8 символов и завершаются двоеточием (:). Эти цепи нумеруются последовательно и могут использоваться как адресаты для команд перехода.

3.3.1 Текстовые редакторы

Список инструкций (IL)

Рабочая область списка инструкций (IL) - это простой текстовый редактор, в который непосредственно вводятся инструкции.

Список инструкций состоит из последовательности операторов или инструкций. Каждая инструкция должна содержать оператор (функцию) и один или более операндов. Каждая инструкция должна начинаться на новой строке. К каждой инструкции вы можете также добавлять опциональные метки, модификаторы и комментарии.

Используются два различных типа списков инструкций:

- Список инструкций IEC

Списки инструкций IEC вводятся и редактируются точно таким же образом, как списки инструкций MELSEC. Однако необходимо учитывать следующие различия в программировании:

- Цепи MELSEC в IEC IL

Вы можете включать цепи MELSEC в списки инструкций IEC, что обеспечивает доступ к системным инструкциям MELSEC.

- Сумматор

Сумматор - система управления результатами, известная из языков высокого уровня. Результат каждой операции сохраняется в битовом сумматоре непосредственно после выполнения инструкции. Сумматор всегда содержит результат операции последней выполненной инструкции. Вы не должны программировать любые входные условия (условия выполнения) для операций; выполнение всегда зависит от содержимого сумматора.

Дополнительная информация о списке инструкций IEC содержится в главе 16.

- Список инструкций MELSEC

Списки инструкций MELSEC вводятся и редактируются точно таким же образом, как списки инструкций IEC. Однако вы можете использовать только систему команд MELSEC; стандартное программирование IEC невозможно.

MELSEC	LD	X0
	CJ	P_20
	LD	X1
	POU	Y0
P_20 MELSEC	LD	X2
	OUT	Y1

Пример сети MELSEC

Структурированный текст

Структурированный текст - это полезный инструмент. Он особенно нравится программистам, пришедшим из мира ПК. При тщательном программировании и обдумывании способа работы ПЛК они будут довольны этим редактором.

Редактор структурированного текста совместим с IEC 61131-3, выполнены все требования стандарта.

```
(*Example showing Structured Text*)
Y00:=X00;
Y01:=X01 AND X02 OR X03;
M0:=(M1 AND (M2 OR M3)) OR X04;
```

Пример структурированного текста

Пример программирования на языке структурированного текста приведен в главе 17.

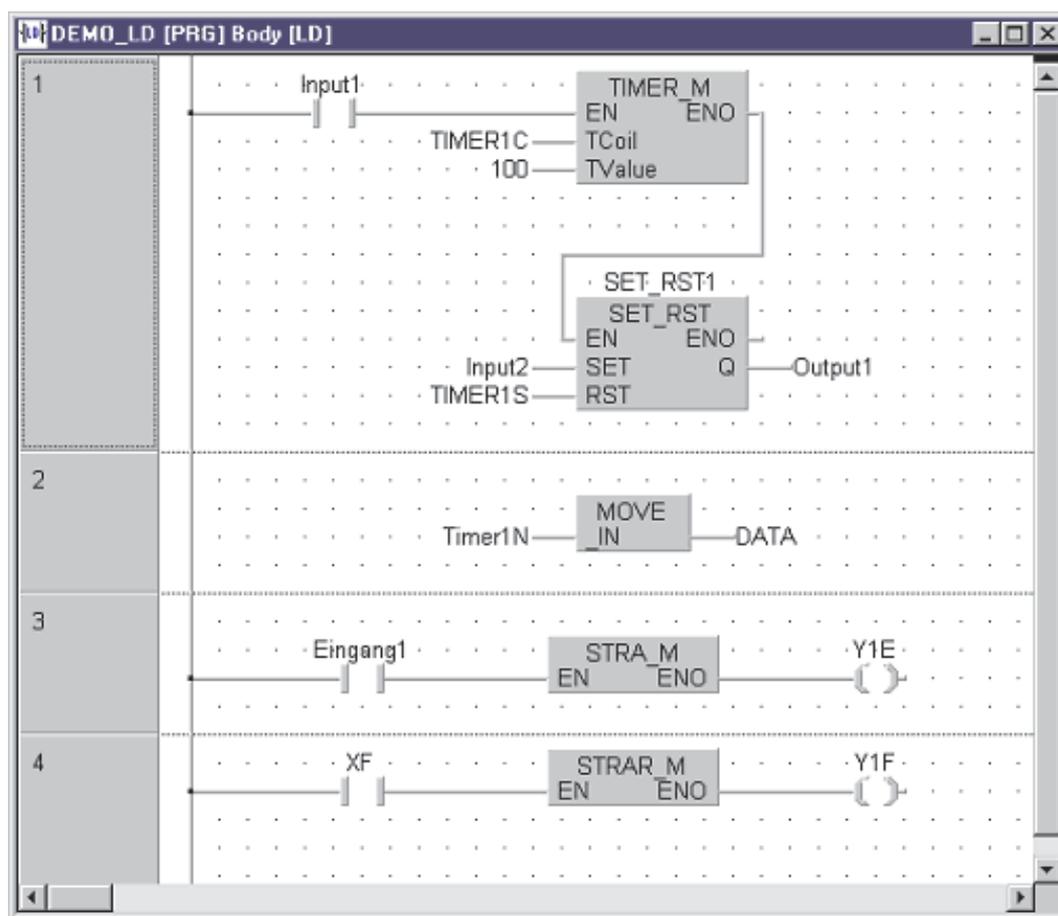
3.3.2 Графические редакторы

Релейная диаграмма

Релейная диаграмма состоит из входных контактов (устройств и прерывателей), выходных катушек, функциональных блоков и функций. Эти элементы соединяются горизонтальными и вертикальными линиями, чтобы создать цепи. Цепи всегда начинаются на сборной шине (шине электропитания) слева.

Функции и функциональные блоки показываются как блоки в диаграмме. Кроме стандартных входных и выходных параметров, некоторые блоки также включают булев вход (EN = включить) и выход (ENO = включить ВЫХОД). Состояние входа всегда соответствует состоянию выхода.

Пример релейной диаграммы:

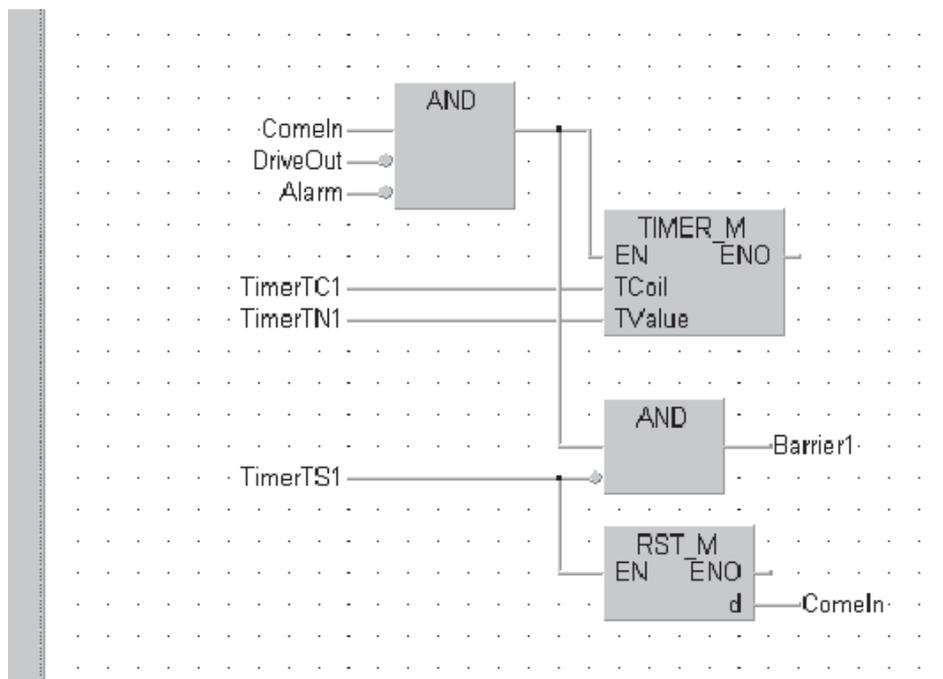


Функциональная блок-схема

Все инструкции выполняются с использованием блоков, которые соединены друг с другом горизонтальными и вертикальными соединительными элементами. Шины электропитания отсутствуют.

Кроме стандартных стандартных входных и выходных параметров, некоторые блоки также включают булев вход (EN = включить) и выход (ENO = включить ВЫХОД). Состояние входа всегда соответствует состоянию выхода.

Пример функциональной блок-схемы:

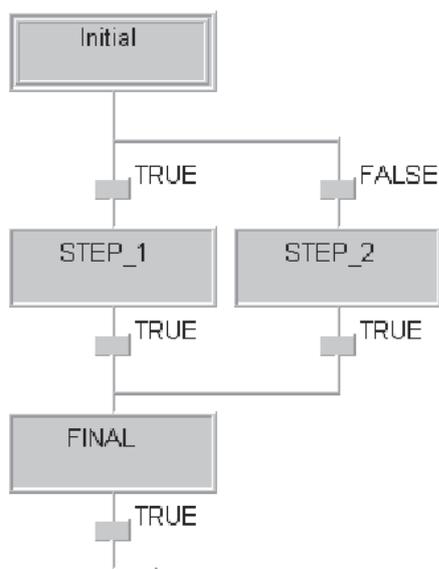


Последовательная функциональная схема

Последовательная функциональная схема является одним из графических языков. Его можно рассматривать как структурирующий инструмент, позволяющий четко и ясно представить последовательное выполнение процессов.

Единственный возможный программный модуль в SFC - это программа.

Последовательная функциональная схема включает два базовых элемента - шаги и переходы. Последовательность состоит из последовательности шагов, каждый шаг отделяется от следующего переходом. В любой момент может быть активным только один шаг. Следующий шаг не активизируется прежде, чем не закончено выполнение предыдущего шага и не удовлетворены условия перехода.



Пример последовательной функциональной схемы

3.4 Типы данных

GX IEC Developer поддерживает следующие типы данных.

3.4.1 Простые типы данных

Тип данных		Диапазон значений		Размер	Допустимые операнды / ПЛК
BOOL	Булев	Битовый операнд	0 (Ложный), 1 (Истинный)	1 бит	X, Y, M, B
INT	Целочисленный	Регистр	-32768 ... +32767	16 бит	D, W, R
DINT	Двойной целочисленный		-2 147 483 648 ... 2 147 483 647	32 бита	
WORD	Строка битов	K4M0	0 - 65 535	16 бит	X, Y, M, B
DWORD		K8M0	0 - 4 294 967 295	32 бита	
REAL	Значение с плавающей запятой	7 разрядов		32 бита	FX2N, FX3U
STRING	Строка символов	20 символов (по умолчанию)		32 бита	FX3U
TIME	Значение времени	-T#24d0h31m23s64800ms ... T#24d20h31m23s64700 ms		32 бита	Все контроллеры серии FX

3.4.2 Сложные типы данных

МАССИВЫ

Массив является областью или матрицей переменных конкретного типа.

Например, **ARRAY [0..2] OF INT** - одномерный массив трех целочисленных элементов (0,1,2). Если начальный адрес массива - D0, то массив состоит из D0, D1 и D2.

Идентификатор	Адрес	Тип	Длина
Motor_Volts	D0	ARRAY	[0...2] OF INT

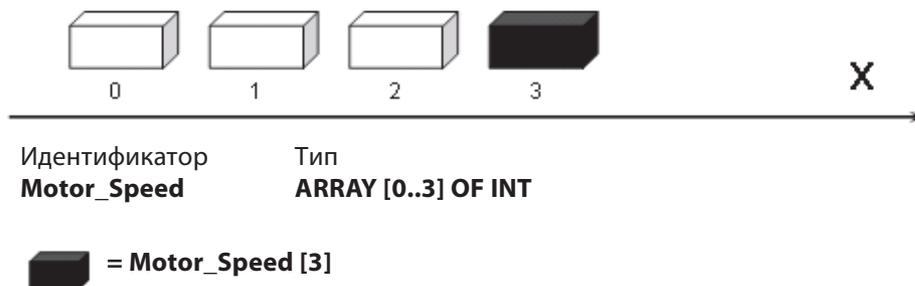
В программе элементы программы могут использовать: Motor_Volts[1] и Motor_Volts[2], как объявления. В данном примере это означает, что адресуются D1 и D2 .

Массивы могут иметь до трех размерностей, например: ARRAY [0...2, 0...4] содержит три элемента в первой размерности и пять во второй.

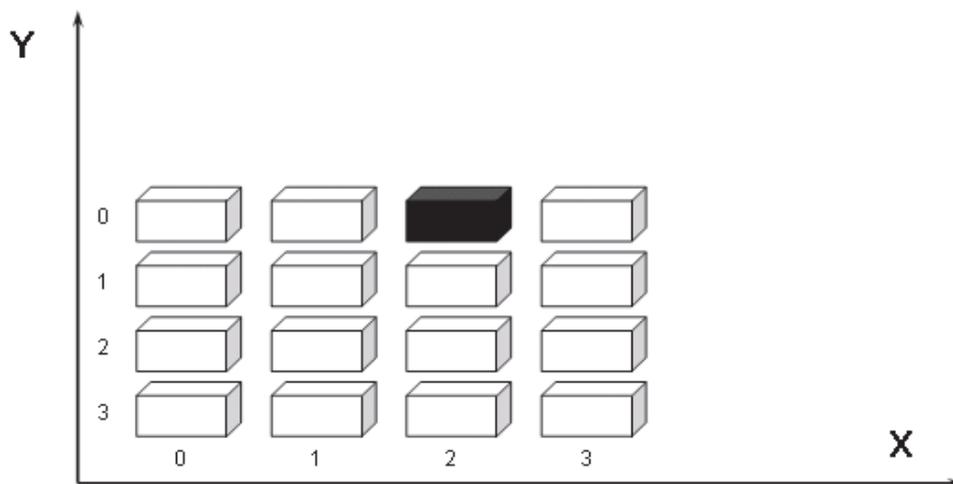
Массивы могут обеспечить удобный способ "индексации" имен тегов, т.е. одно объявление в таблице локальных или глобальных переменных может иметь доступ ко многим элементам.

Следующие диаграммы иллюстрируют графическое представление трех типов массивов.

Одномерный массив



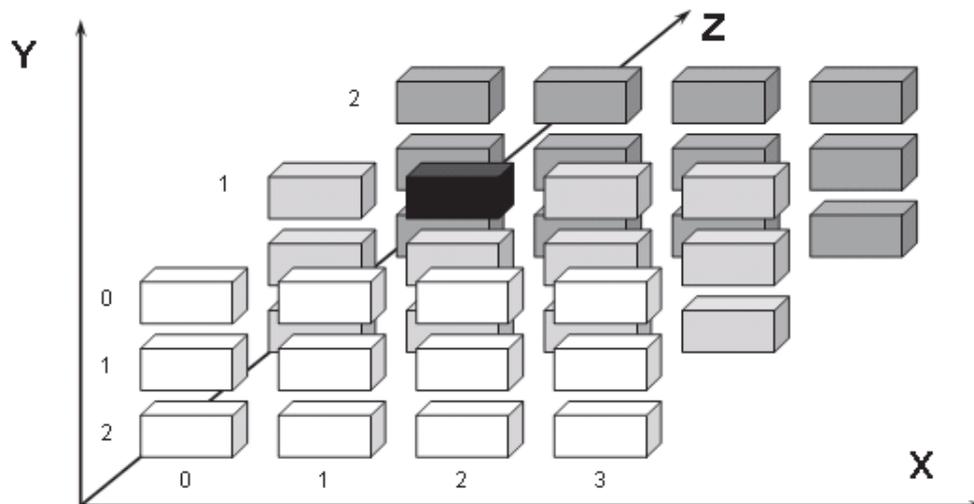
Двумерный массив



Идентификатор **Motor_Volts** Тип **ARRAY [0..3, 0..3] OF INT**

 = **Motor_Volts [2, 0]**

Трёхмерный массив



Идентификатор **Motor_Current** Тип **ARRAY [0..3, 0..2, 0..2] OF INT**

 = **Motor_Strom [1, 0, 1]**

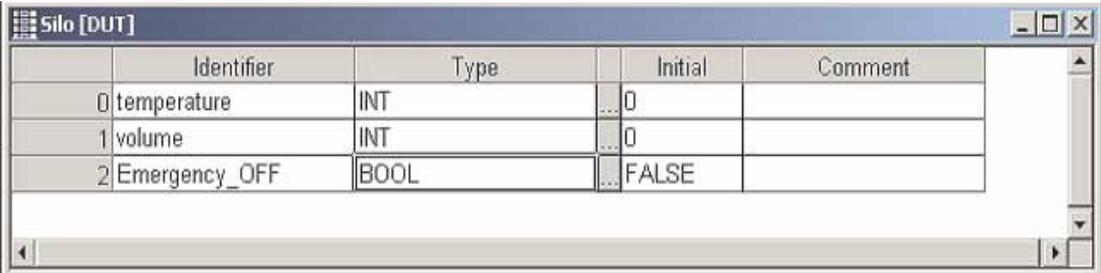
Типы блоков данных(DUT)

Можно создавать пользовательские типы блоков данных (DUT). Это может быть полезным для программ, которые содержат общие части, например, при управлении шестью идентичными бункерами. Поэтому может быть создан тип блока данных, названный "Silo" (Бункер), состоящий из структур различных элементов, например, INT, BOOL и т.д.

Заканчивая список глобальных переменных, можно использовать идентификаторы типа Silo. Это означает, что предварительно заданную группу "Silo" можно использовать с элементами, определенными для каждого бункера, что сокращает время разработки и позволяет многократно использовать DUT.

Пример использования DUT

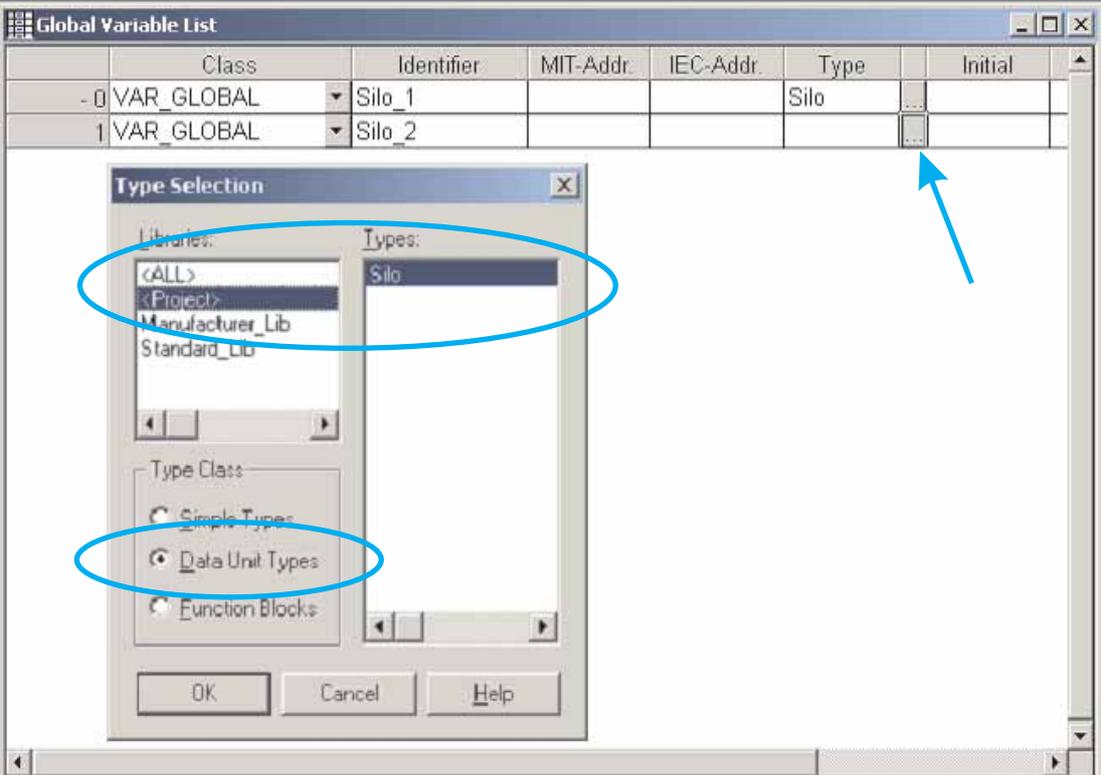
Следующий пример показывает создание типа данных, названного Silo. Набор переменных Silo содержит две переменные INT и одну переменную типа BOOL.



Identifier	Type	Initial	Comment
0 temperature	INT	0	
1 volume	INT	0	
2 Emergency_OFF	BOOL	FALSE	

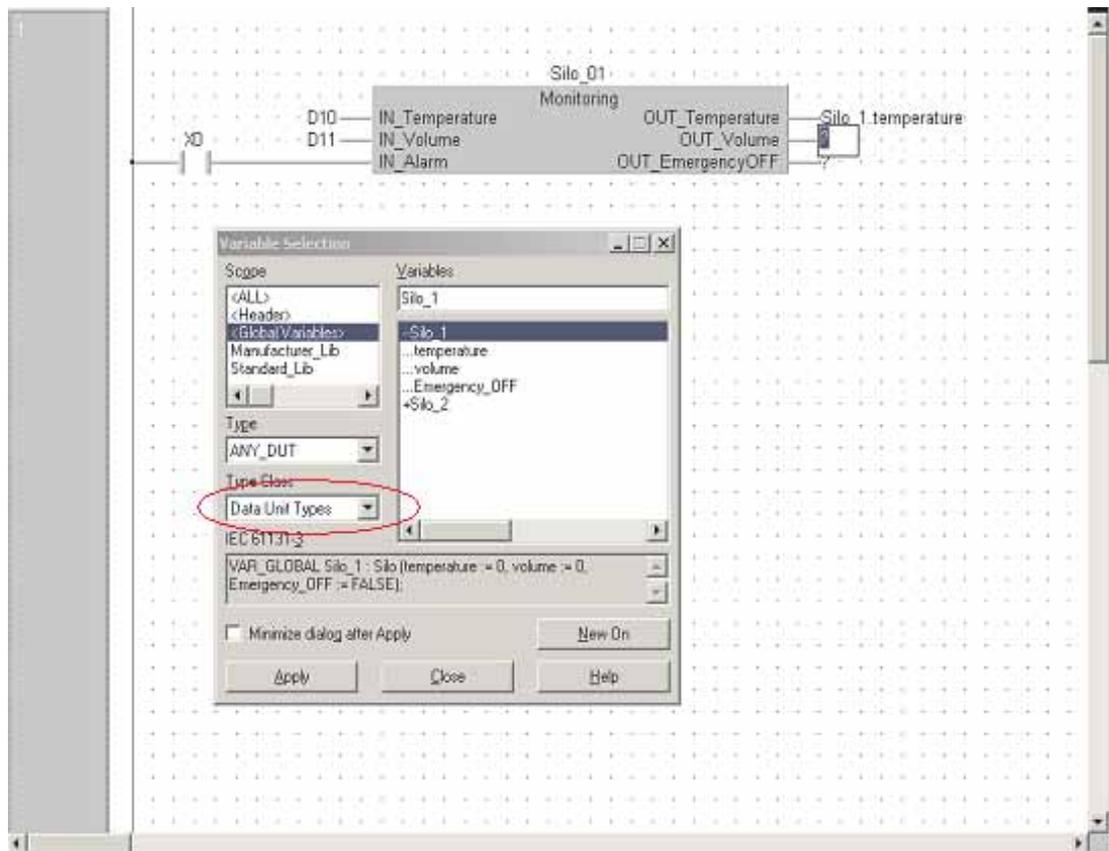
Как объявлять DUT

Дважды щелкните на **Global_Vars** в окне Project Navigator и введите следующие строки в таблицу объявления глобальных переменных.

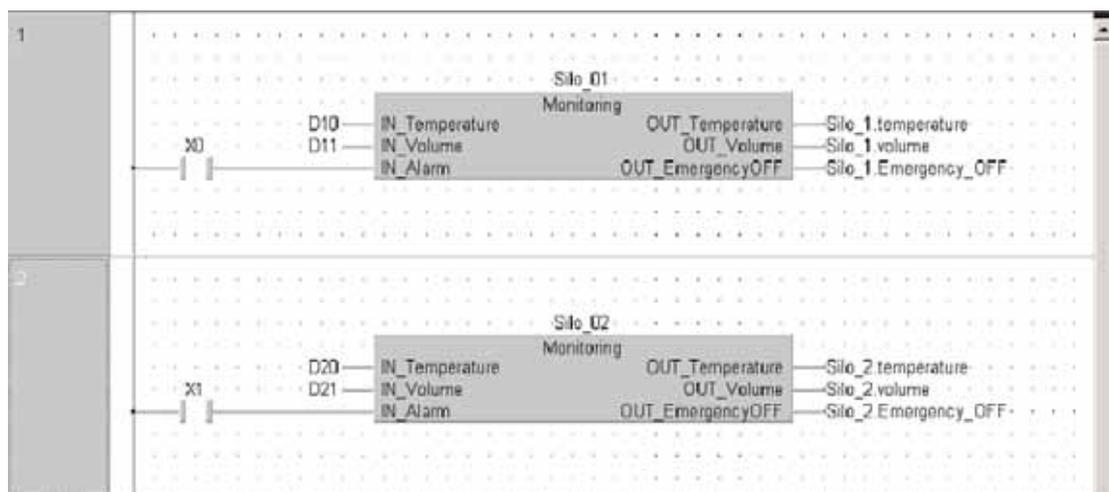


Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
VAR_GLOBAL	Silo_1			Silo	
VAR_GLOBAL	Silo_2			Silo	

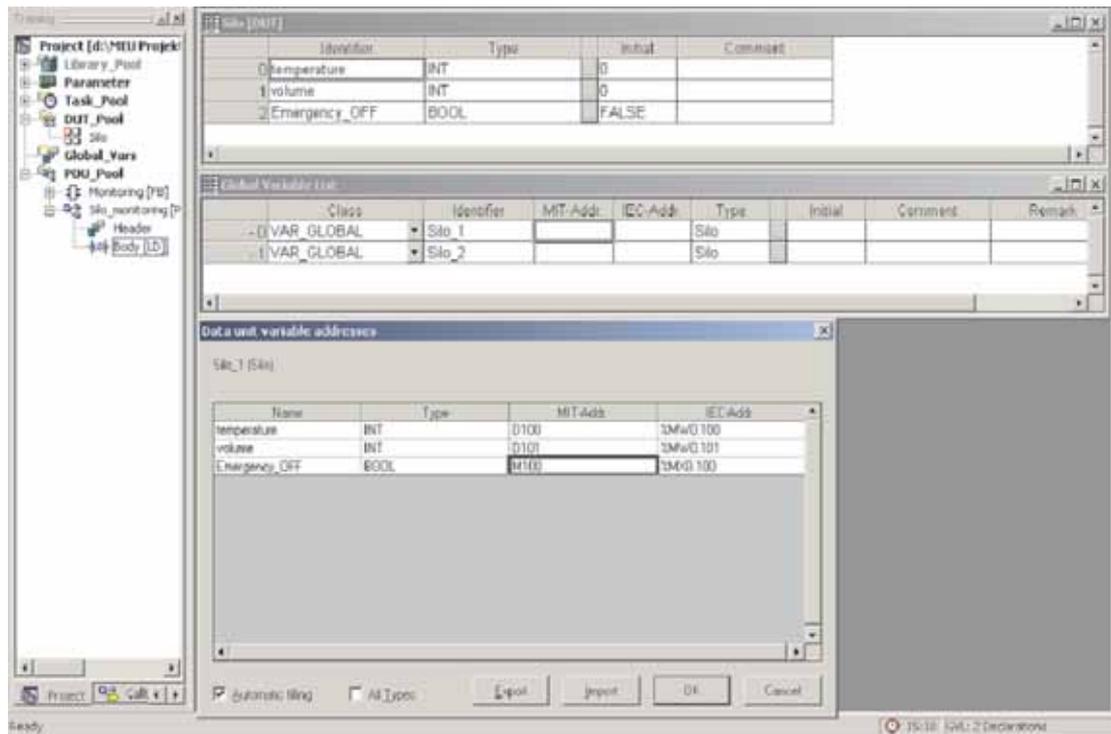
Переменные хранятся в списке глобальных переменных. Структура обеих переменных, Silo_1 и Silo_2, идентична, поэтому для ссылки на отдельную переменную каждого DUT вам нужен лишь префикс в их именах с именем соответствующей глобальной переменной.



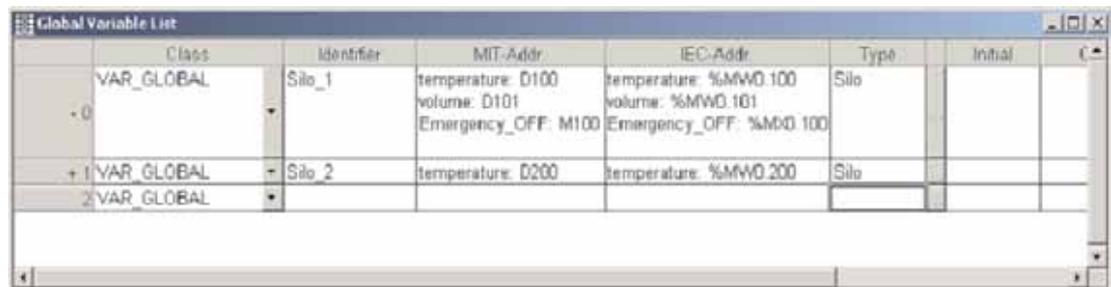
В этом примере функциональный блок типа "Monitoring" был запрограммирован для того, чтобы присвоить значение регистра и Булева входа для элементов DUT. Затем для двух бункеров были созданы два отдельных экземпляра (Silo_01 и Silo_02) этих функциональных блоков.



Список глобальных переменных GVL был расширен, чтобы определить адреса для всех элементов типов блоков данных. Не определенные адреса обрабатываются системой.



Чтобы просмотреть все определения сразу (если имеется несколько определений), элементы DUT в GVL можно расширить, дважды щелкнув в поле номера строки.



Дополнительные примеры применения элементов DUT содержатся в главе 11.

3.4.3 Таймеры и счетчики MELSEC

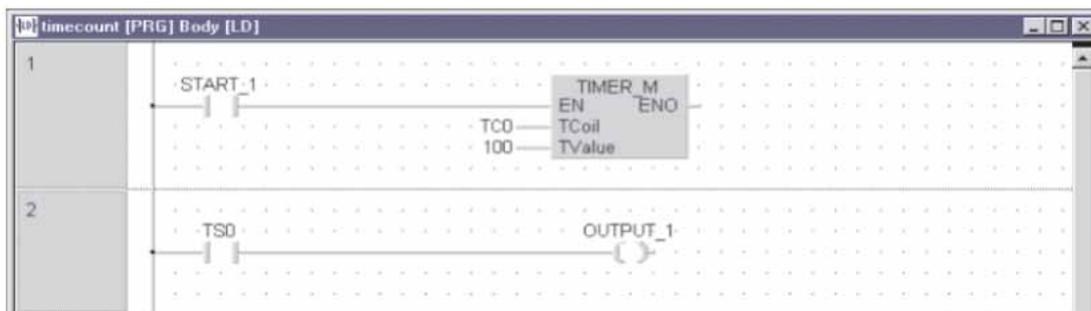
При программировании стандартных Таймеров/Счетчиков должны соблюдаться правила IEC:

Программирование катушки таймера/счетчика: **TCn / CCn**

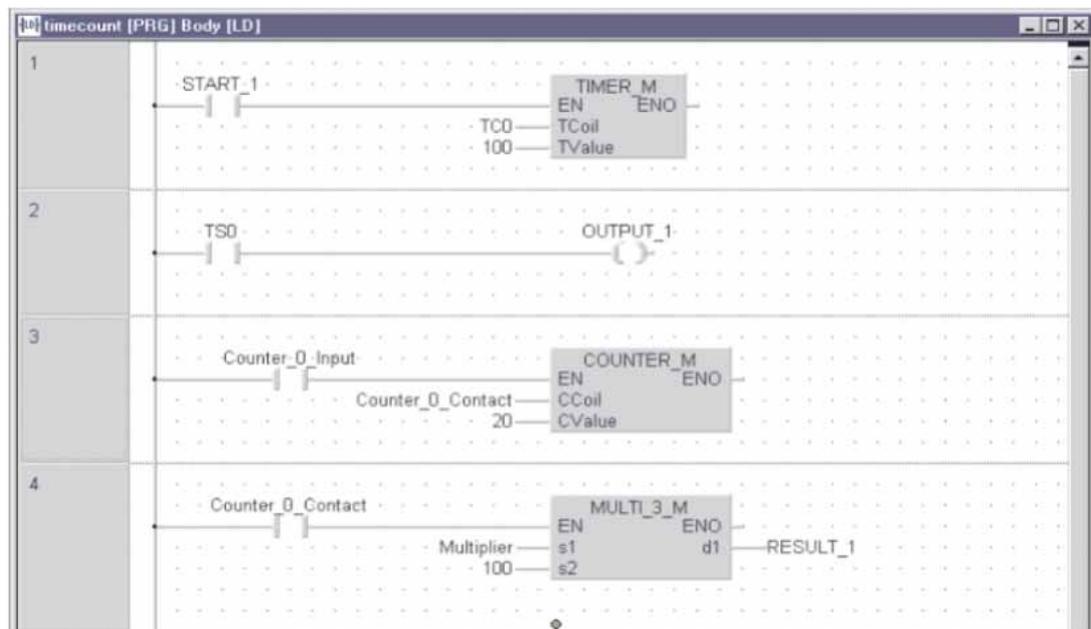
Программирование контакта таймера/счетчика: **TSn / CSn**

Программирование значения таймера/счетчика: **TNn / CNn**

В следующем примере T0 превращается в TC0 и TS0. В этом случае были использованы адреса Mitsubishi, поэтому очень важно проверить использование системной переменной по умолчанию T/C:



В следующем примере счетчик был запрограммирован с использованием идентификаторов, которые должны быть объявлены в таблицах глобальных и локальных переменных:



4 Разработка проекта

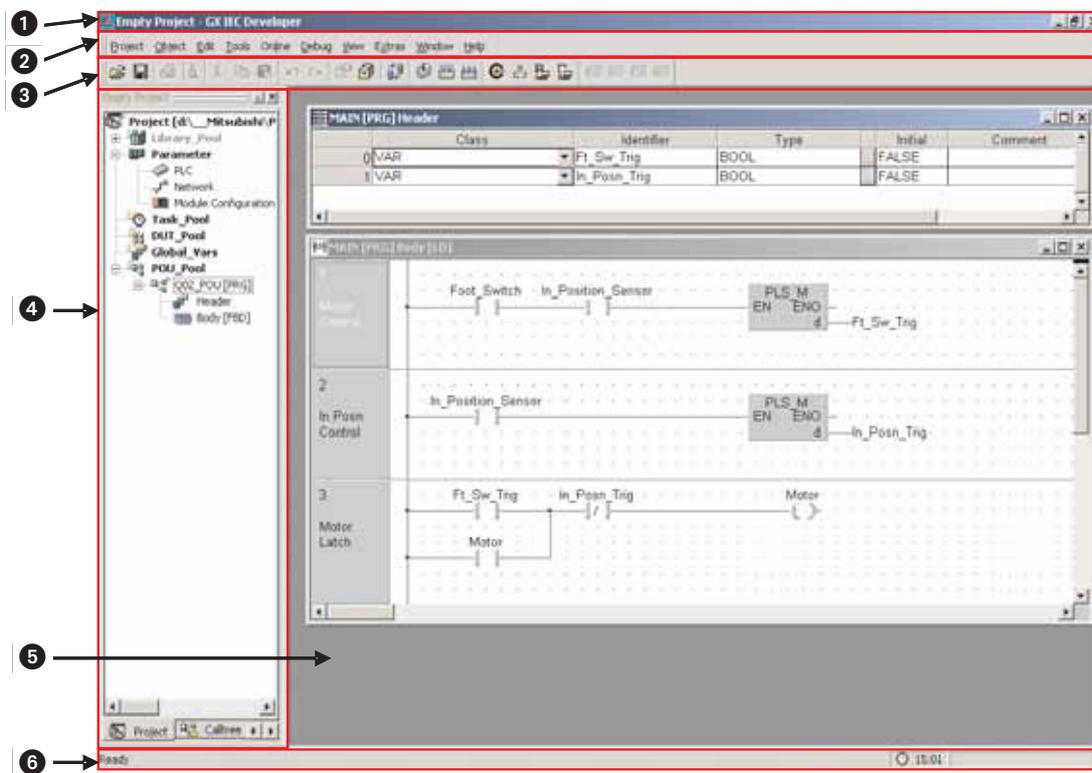
В следующем разделе мы разработаем наш первый проект, сначала используя редактор лестничных диаграмм.

Обсуждаемые вопросы

- Использование Project Navigator
- Использование списка глобальных переменных (GVL) с идентификаторами
- Объявление переменных в заголовке программы
- Разработка программ с помощью редактора релейных диаграмм IEC
- Программирование таймеров/счетчиков IEC
- Комментирование и документация
- Загрузка и мониторинг

4.1 Запуск GX IEC Developer

После запуска GX-IEC Developer под Windows появится следующее окно:



* На этой иллюстрации проект уже открыт, чтобы лучше проиллюстрировать вид экрана программы. Открыв GX IEC Developer, обычно вы должны открыть существующий проект или создать новый.

1 Строка заголовка приложения

Строка заголовка приложения дает вам имя открытого проекта.

2 Строка меню

Строка меню обеспечивает доступ ко всем меню и командам, которые используются для управления GX IEC Developer. Когда вы выбираете один из элементов в строке, щелкая мышью, раскрывается меню опций. Опции, отмеченные стрелкой, содержат подменю, которые при щелчке показываются с дополнительными опциями. Выбор команд обычно открывает диалоговое окно или поле ввода.

Структура меню GX IEC Developer является контекстно-зависимой, изменяясь в зависимости от того, что вы в делаете в программе. Команды, показанные светло-серым, в настоящее время недоступны.

3 Панель инструментов

Значки панели инструментов одним щелчком мыши дают вам прямой доступ к чаще всего используемым командам. Панель инструментов является контекстно-зависимой, показывая различные наборы значков в зависимости от того, что вы в делаете в программе.

4 Окно Project Navigator

Project Navigator - это центр управления GX IEC Developer. Окно Project Navigator не отображается, пока вы не открыли существующий проект или не создали новый.

5 Редактор (Тело)

В этой области можно редактировать программные модули (POU). Каждый ПМ состоит из заголовка и тела.

– Заголовок

Заголовок является встроенной частью программного модуля (POU). Это место, где должны объявляться переменные, которые будут использованы в программном модуле.

– Тело

Тело является встроенной частью программного модуля (POU). Оно содержит элементы кода и синтаксис реальной программы, функционального блока или функции.

6 Строка состояния

Эта строка, показанная в нижней части экрана, дает полезную информацию о текущем состоянии вашего проекта. Отображение строки состояния может быть разрешено или запрещено, и вы также можете конфигурировать отдельные опции отображения согласно текущим потребностям.

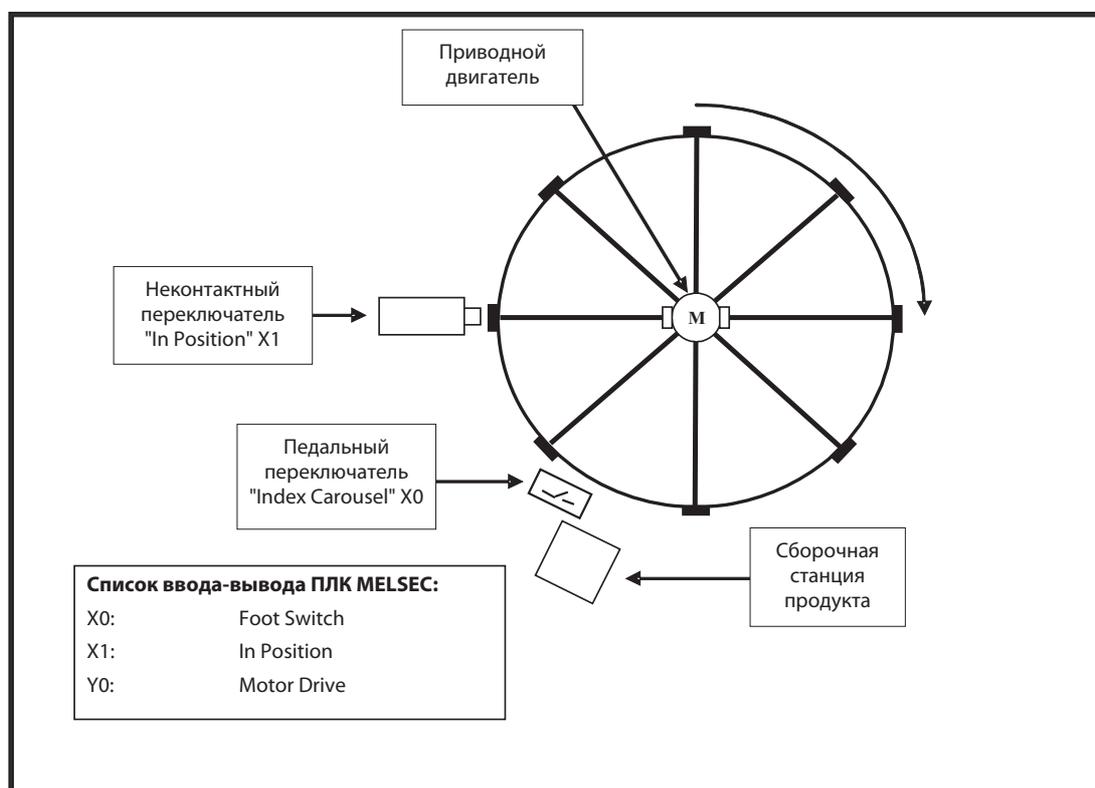
4.2 Прикладная программа

4.2.1 Пример: Управление карусельным столом

Следующая прикладная программа будет использована как пример создания простой программы с использованием инструментов GX IEC Developer.

Последовательность рабочих операций

- ① Кратковременно нажмите на pedalный выключатель, чтобы повернуть карусель на определённый угол.
- ② Карусель вращается - датчик позиции "В позиции" (In-Position) выключается, когда карусель начинается вращаться.
- ③ Датчик позиции "In-Position" включается, когда карусель достигает определенной позиции.
- ④ Соберите продукт
- ⑤ Повторите процесс (Возвратитесь к ①.)



Имеется ряд проблем, которые необходимо разрешить, проектируя программу ПЛК для этого приложения. Использование стандартной старстопной схемы невозможно без модификации в связи со следующими затруднениями:

- Педаальный выключатель может срабатывать произвольно. Нажав выключатель, оператор может забыть отпустить его - при этом стол может непрерывно вращаться, проходя мимо позиции сборки.
- Когда срабатывает "In-Position" X1, он остается установленным; это препятствует повторному повороту стола.

Следовательно, конструкция должна содержать блокировки для предотвращения описанных выше неправильных операций. Альтернативный подход к конструкции связан с использованием "импульсной логики" с помощью конфигураций IEC или MELSEC, "запускаемых фронтом".

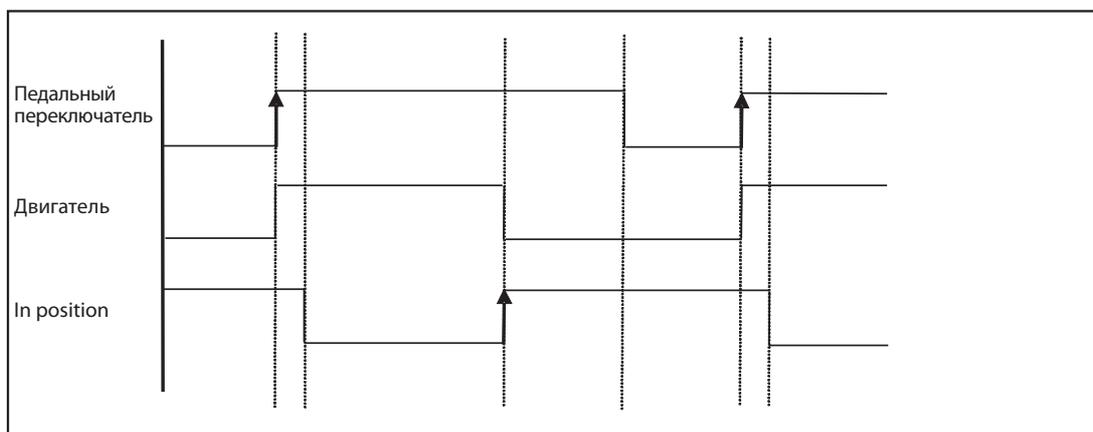
В этом приложении наиболее целесообразно использовать команду MELSEC "PLS" (Установка операнда при возрастающем фронте). Она используется здесь вместо инструкции IEC R_TRIG (триггер, тактируемый нарастающим фронтом), которая также может применяться.

Следующая схема иллюстрирует порядок формирования последовательностей команд управления каруселью. Отметим, что положительный фронт педального выключателя включает двигатель, независимо от того, включен ли датчик "In Position".

Когда стол начинает вращаться, датчик "In position" выключается несколько позже. Двигатель продолжает вращать карусельный конвейер, пока не будет обнаружен положительный фронт от датчика "In position"; он выключает двигатель. Учтите, что педальный выключатель продолжает удерживаться нажатым.

Двигатель может начать вращение, только когда педальный выключатель отпускается и нажимается повторно. Следовательно, двигатель включается снова по нарастающему фронту педального выключателя.

Временная диаграмма логики управления каруселью:

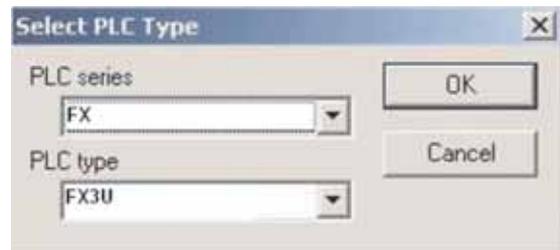


4.2.2 Создание нового проекта

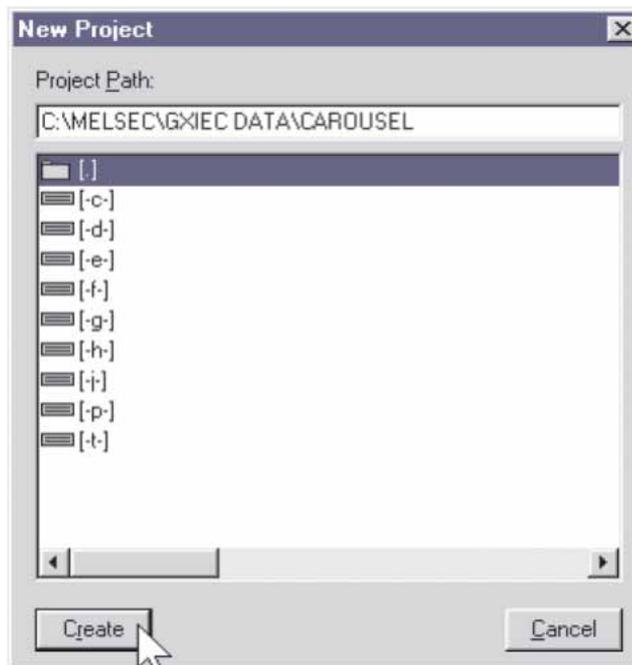
- ① В меню **Project** выберите **New**.



- ② Выберите соответствующий тип ПЛК из списка:

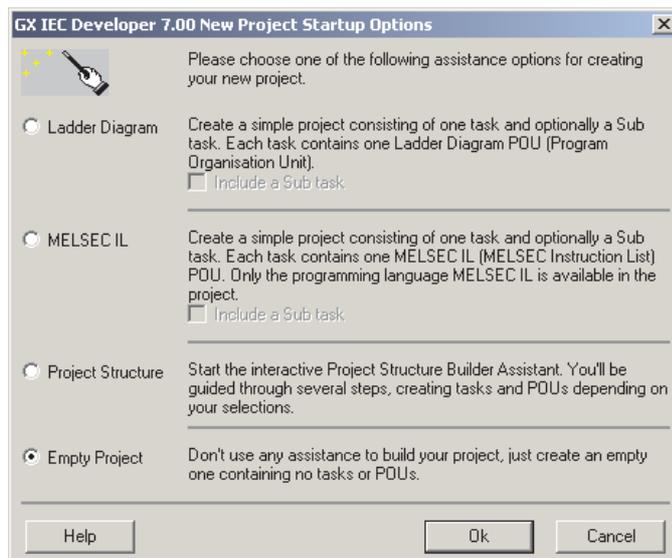


- ③ Задайте название проекта в поле Project Path. В этом случае используйте "\GX-IEC DATA\CAROUSEL" и щелкните на **Create**, как на следующей иллюстрации:



Мастер

Будет показан мастер запуска проекта Project Startup Wizard :

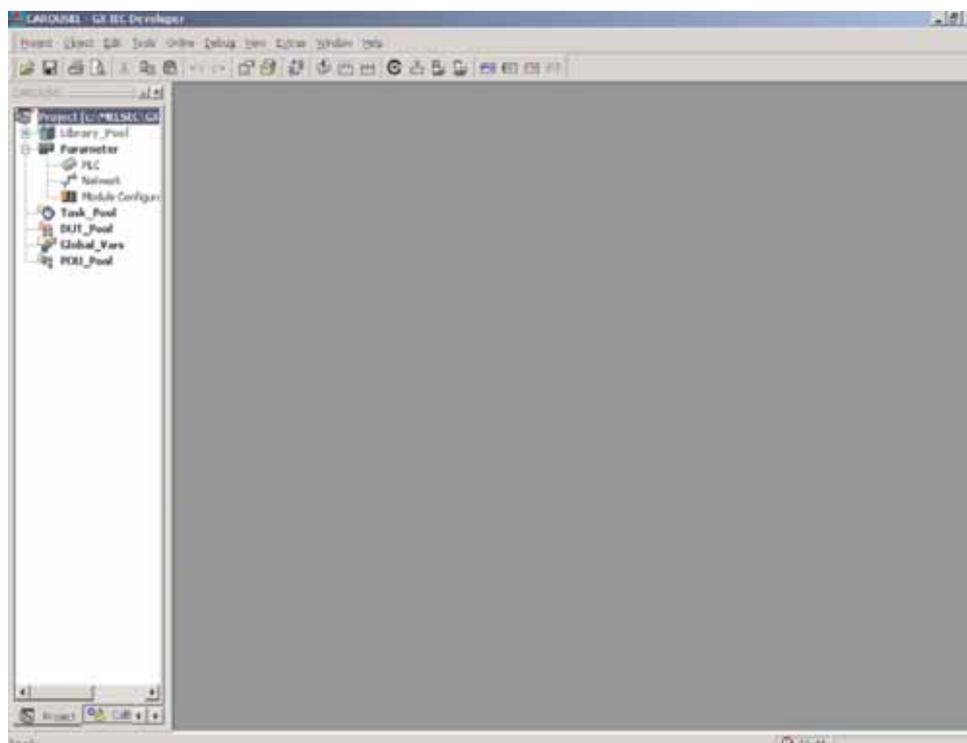


Мастер предоставляет быстрый способ начинать проекты. Он создает базовые стартовые структуры для простых проектов.

Выберите опцию **Empty Project** и щелкните на **OK**.

Это эффективно запрещает Мастеру создавать любые элементы проекта. Конечно, при желании можно использовать Мастер, однако, чтобы полностью исследовать основные функции GX-IEC Developer, для целей курса обучения цели мы создадим программу вручную.

Показан экран проекта дисплея, представленный ниже:

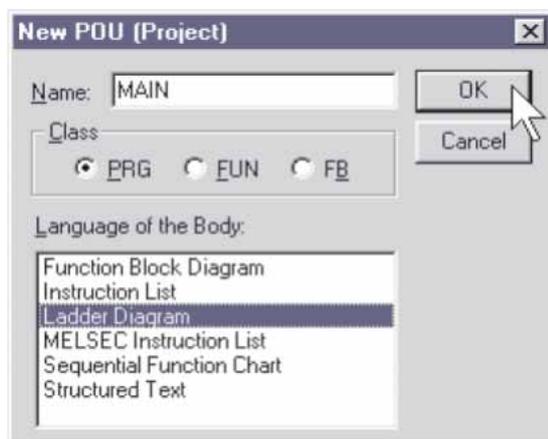


Это основной экран проекта.

Навигационное окно проекта в левой части экрана позволяет пользователю быстро переходить к любой части проекта, дважды щелкнув на выбранном элементе.

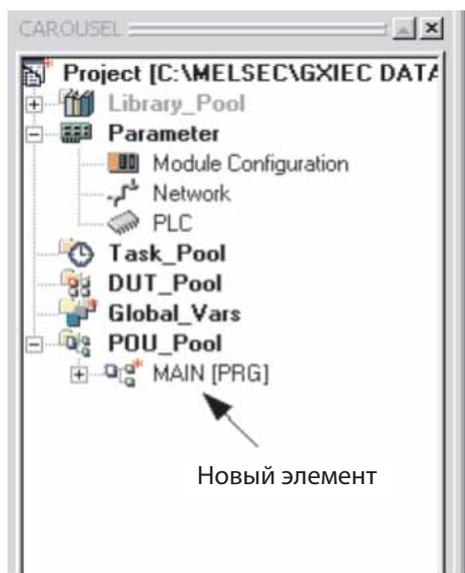
4.2.3 Создание нового программного модуля "POU"

- ① Щелкните на кнопке "New POU"  (или щелкните правой кнопкой на пуле программных модулей **POU Pool**) на панели инструментов. Необходимо ввести спецификации нового ПМ следующим образом:

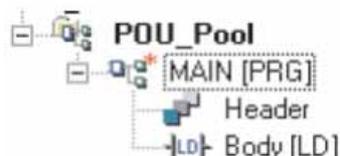


Имя POU будет "MAIN", и его необходимо задать как релейную диаграмму **Ladder Diagram** типа **PRG** (программа).

- ② Щелкните на **OK**; проект будет добавлен в пул программных модулей POU Pool, что показано в "навигационном окне проекта":



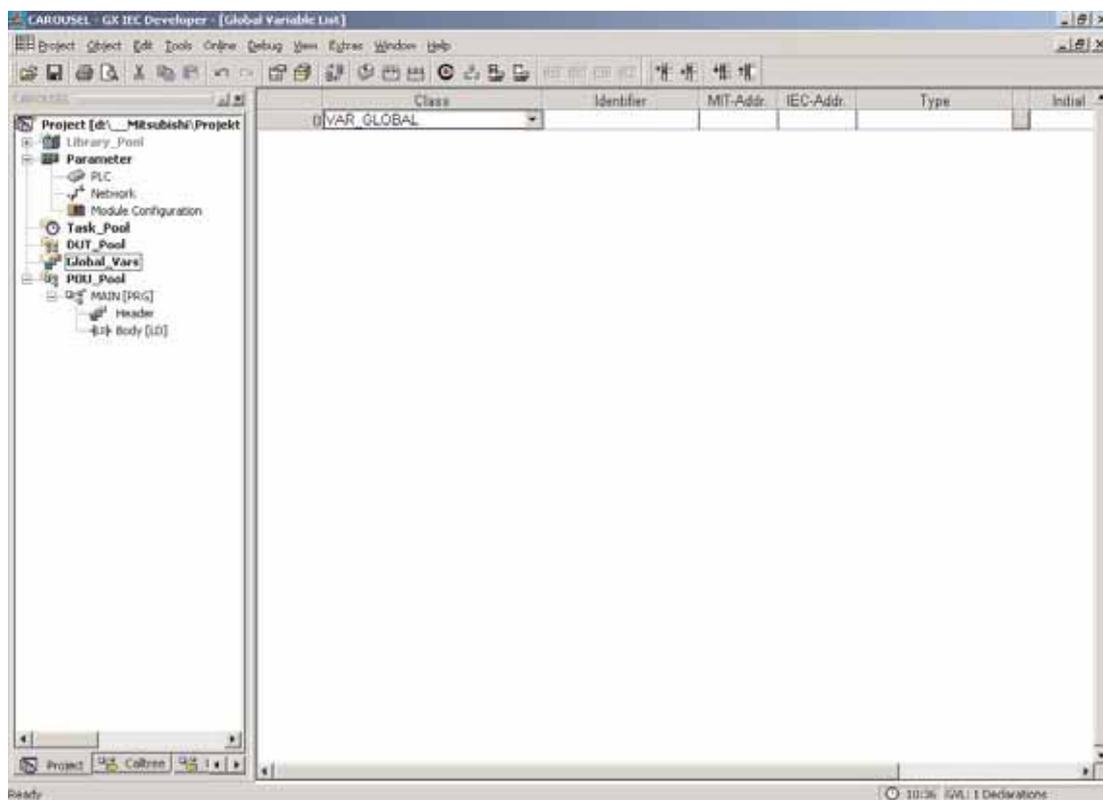
- ③ Дважды щелкните на значке программы **MAIN** или щелкните на символе в POU Pool, чтобы показать структуру каталога и элементы заголовка **Header** и тела **Body**:



4.2.4 Назначение глобальных переменных

Прежде, чем можно будет создать любой код программы, необходимо определить и присвоить все предварительно определенные физические входы и выходы ПЛК, включая любые совместно используемые переменные, которые должны применяться в проекте.

Дважды щелкните указателем мыши на **Global_Vars**, чтобы открыть редактор для Глобальных переменных. Он называется списком глобальных переменных - GVL.



Глобальные переменные являются ссылками на физические устройства ПЛК.

Как отмечалось выше, если должны применяться соглашения IEC, то вместо дискретных адресов в нашей программе должны использоваться символические идентификаторы (имена). Поэтому данные адреса необходимо объявить в списке глобальных переменных (GVL). Необходимо ввести идентификатор, используя его адрес в ПЛК (используя обозначения Mitsubishi или IEC) и его тип, например; является ли этот операнд "битовым" или "словным". После завершения этот список может использоваться всеми созданными программными модулями ROU.

Объявление переменных

Как можно заметить по списку полей GVL, у каждой переменной есть следующий набор элементов:

- **Class** - Класс

Класс присваивает переменной специфическое свойство, определяющее, как эта переменная должна использоваться в проекте

- **Identifier** - Идентификатор

Каждой переменной присваивается символический адрес, то есть имя. Оно определяется как идентификатор и состоит из строки от алфавитно-цифровых символов и символов "подчеркивания". Идентификатор должен всегда начинаться с буквы или символа подчеркивания. Пробелы и символы математических операторов (например, +,-,*) не допускаются.

- **MIT-Addr** - Адрес Mitsubishi

Это абсолютный адрес, на который производится ссылка в ПЛК.

- **IEC-Addr** - Адрес IEC

Адрес в синтаксисе IEC.

- **Type** - Тип

Относится к типу данных, т.е. BOOL, INT, REAL, WORD и т.д.

- **Initial** - Начальное значение

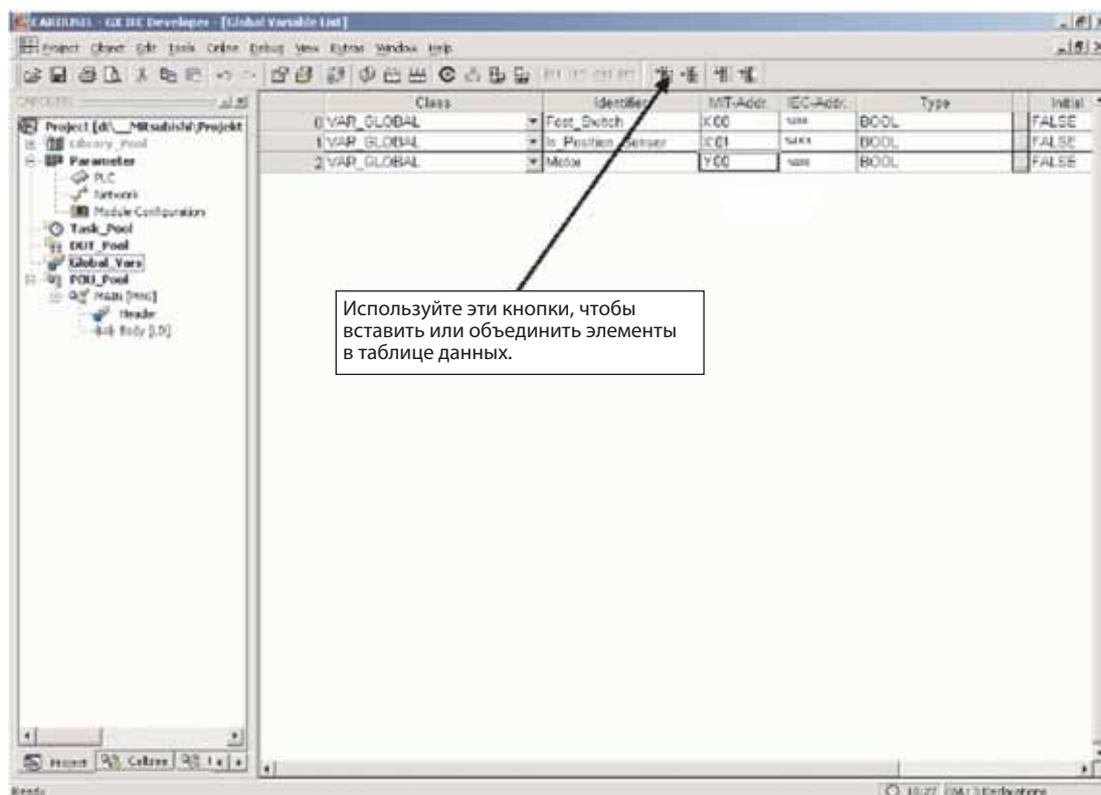
Начальные значения автоматически устанавливаются системой и не могут изменяться пользователем.

- **Comment** - Комментарий

Для каждой переменной можно добавить комментарий до 64 символов

Если в программе не предполагается использовать символические идентификаторы, но только адреса Mitsubishi, то затем нет никакой необходимости заполнять Список глобальных переменных (GVL). Однако программа потеряет совместимость с IEC61131-3.

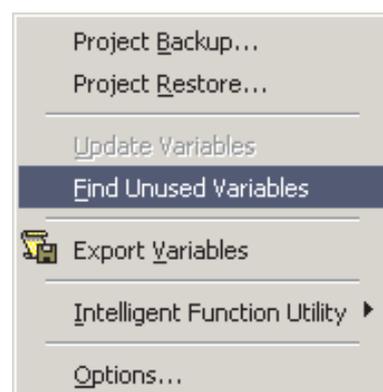
Заполните таблицу, как показано на следующей иллюстрации. Переменная "Type Selection" автоматически распознается и помещается GX IEC Developer после ввода "Address", но ее можно ввести вручную или изменить, щелкнув на стрелке выбора типа в области поля **Type**. Когда введен адрес Mitsubishi, система автоматически осуществляет преобразование и вводит эквивалент IEC.



Выше мы задали глобальные переменные для проекта.

Найти неиспользованные переменные

Используя функцию **Extra** -> **Find Unused Variables**, вы можете найти и удалить все неиспользованные глобальные и локальные переменные, которые объявлены, но не используются в проекте. Неиспользуемые глобальные и локальные переменные будут обнаружены во всем проекте, за исключением пользовательских библиотек.



ПРИМЕЧАНИЕ

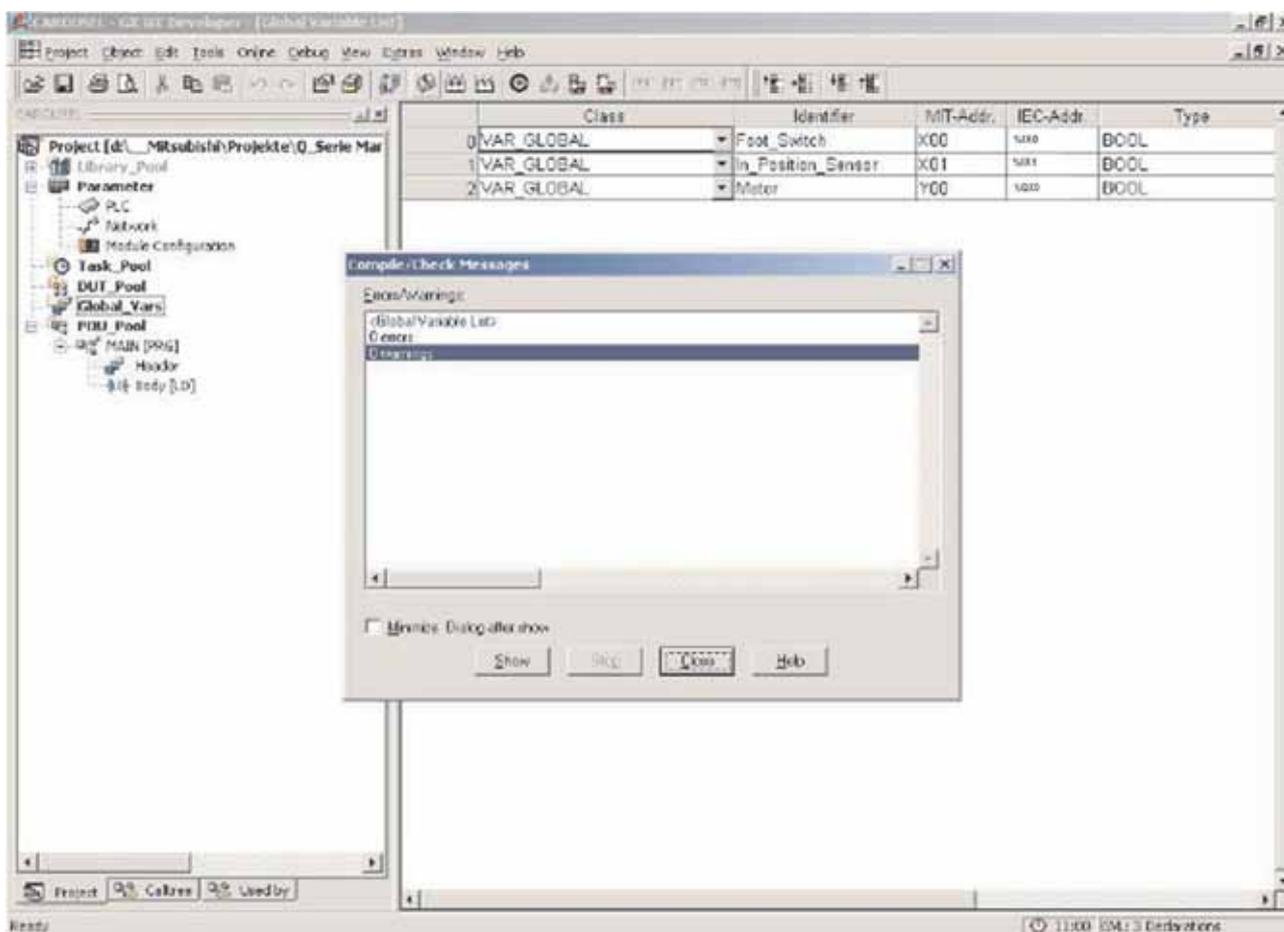
Обнаружение неиспользованных переменных можно выполнить, если проект был создан и с тех пор не изменялся. В противном случае будет показано предупреждающее сообщение.

ПРИМЕЧАНИЯ

Список глобальных переменных включает возможность "Приращения новых объявлений". Если GVL содержит элементы, например, для ряда клапанов - от "Valve_1" до "Valve_n" - то после того, как введена первая строка для Valve_1 и новые строки объявляются либо с помощью значков панели инструментов, либо с использованием "Shift+Enter", то производится автоматическое увеличение как идентификатора, так и поля адреса. Эта возможность разрешена по умолчанию. Если она не требуется, ее можно отключить через меню **Extras (Options → Editing)**, которое будет описано ниже. Можно выбрать все или некоторые программные модули, а также удалить все или выбранные переменные. При этом все неиспользованные глобальные переменные в программных модулях удаляются. Эта возможность будет обсуждаться позже.

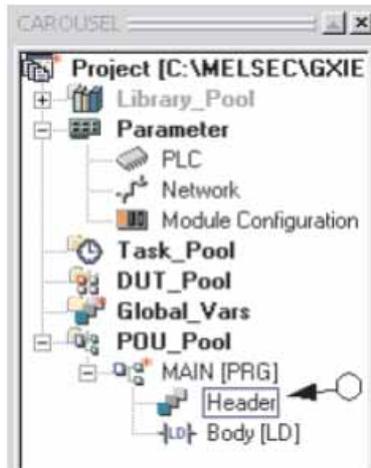
Для всех типов ЦП FX2N, FX3U, Q & AnA(S) или выше полностью поддерживаются значения IEC типа REAL (с плавающей запятой).

Завершив ввод данных в GVL, щелкните на кнопке "Check"  , как показано ниже:

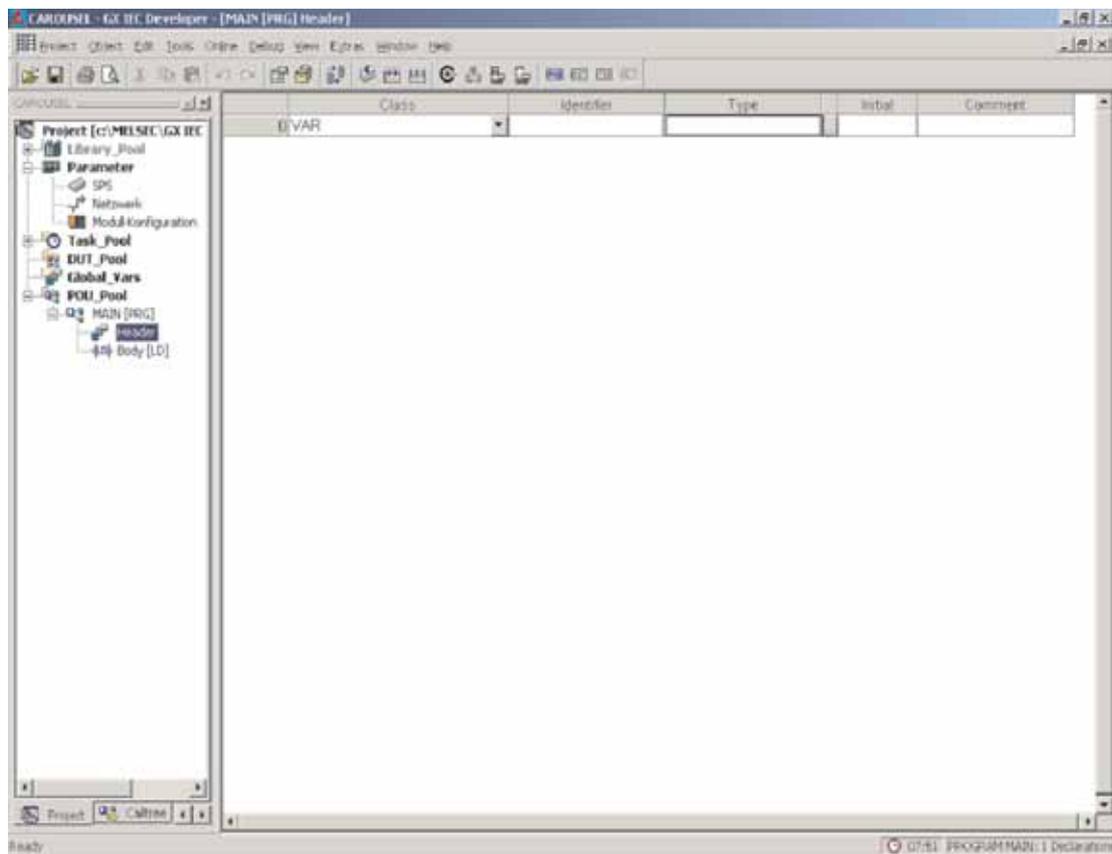


Открытие заголовка POU

Из окна Project Navigation дважды щелкните на **Header** на **MAIN** в пуле POU.



Появится следующий экран:

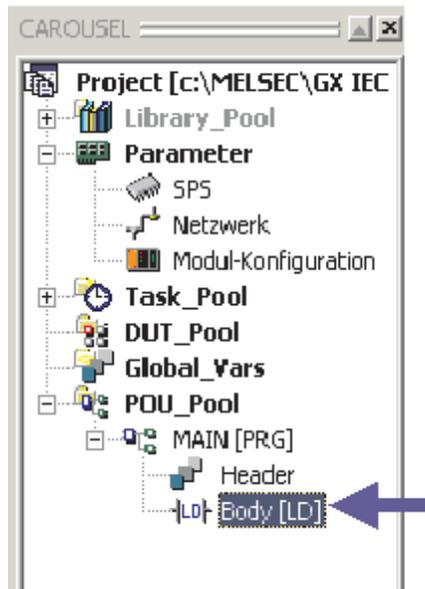


Закройте этот экран заголовка программного модуля.

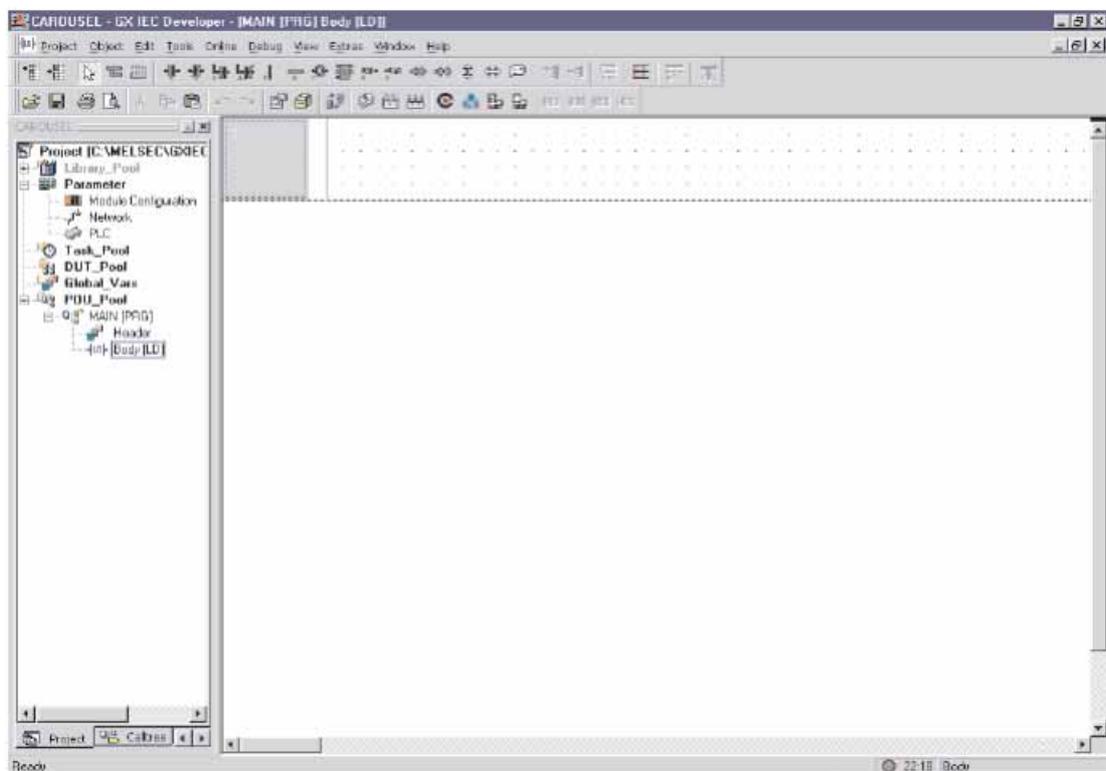
4.2.5 Программирование тела программного модуля

Фактическая программа ПЛК хранится в теле программного модуля.

- ① Чтобы открыть Редактор релейных диаграмм, дважды щелкните на пункте **Body** в пуле программных модулей POU_Pool в навигационном окне проекта:



Появится следующее окно:

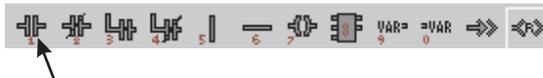


- ② Поместив указатель на границу окна, щелкните и перетащите ее вниз, чтобы увеличить вертикальный размер цепи:



Использование панели символов релейной диаграммы

- ③ Переведя редактор в выбор режима "Selection Mode", выберите "Нормально разомкнутый" контакт из панели инструментов:

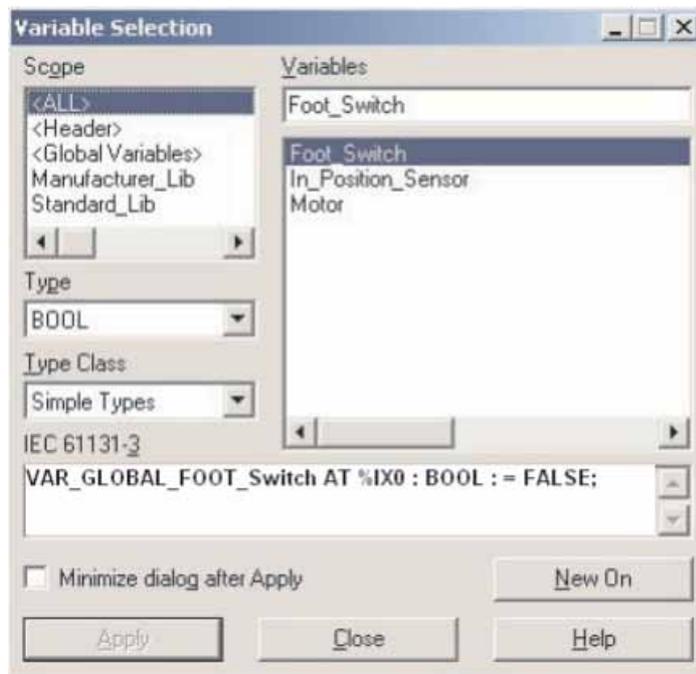


- ④ Перемещая указатель мыши по рабочей области, щелкните, чтобы задать позицию помещения в окне:



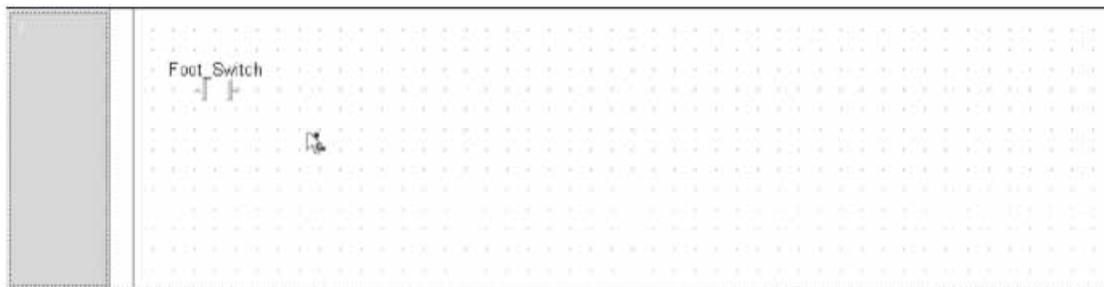
Выбор переменных из заголовка программного модуля

- ① Нажмите кнопку "F2" на клавиатуре или щелкните на кнопке  на панели инструментов, чтобы вызвать окно выбора переменных. Будет показан экран, приведенный ниже:



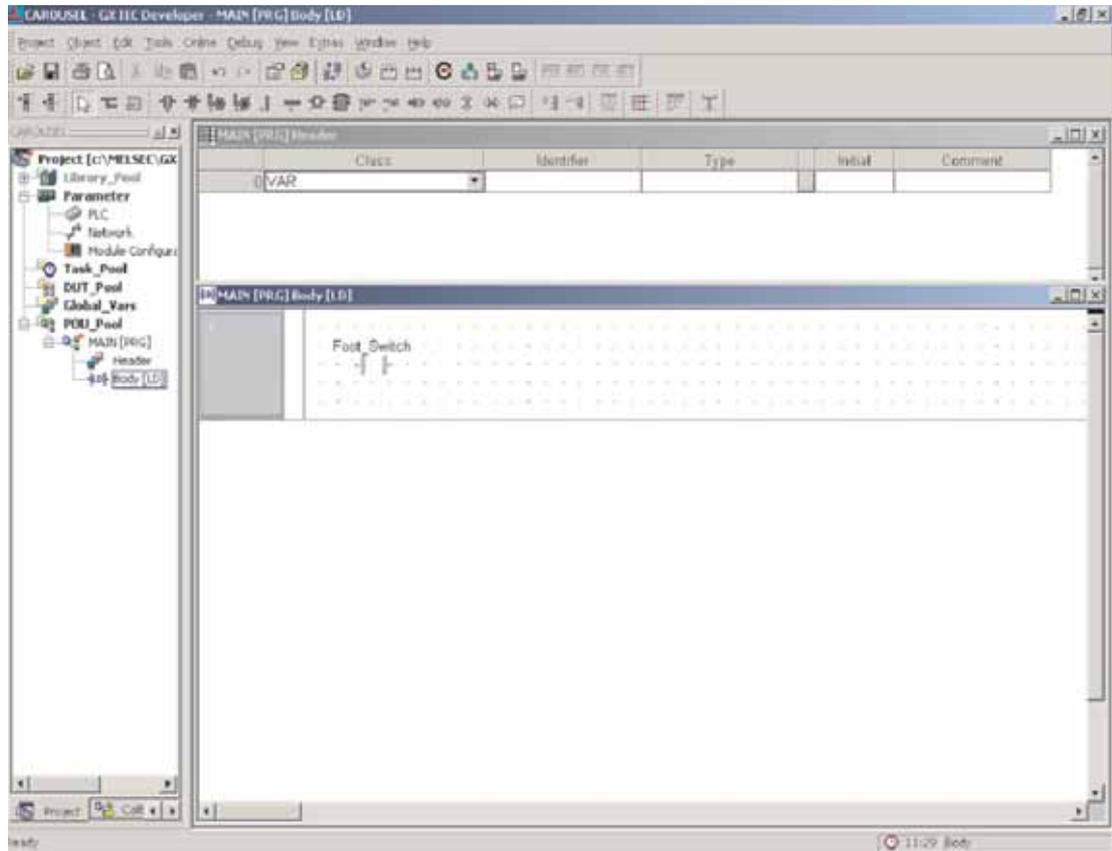
Учтите, что в области **Scope** диалогового окна должен быть выбран текущий заголовок "Header".

- ② Щелкните на "Foot_Switch", чтобы выделить эту переменную, и щелкните на кнопке **Apply**. Затем закройте окно выбора переменных **Variable Selection**.



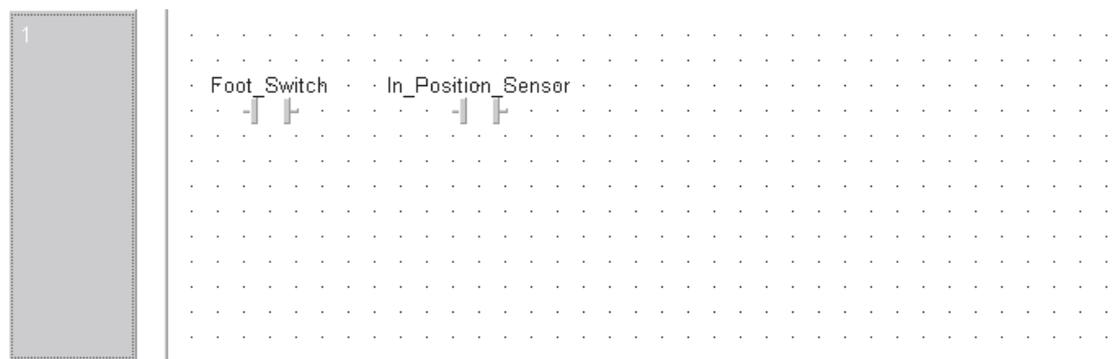
Альтернативный метод задания переменной: Редактирование на разделенном экране

Можно разделить экран для просмотра программного модуля релейной диаграммы и заголовка, открыв так заголовок, так и и релейную диаграмму, и выбрав "Tile Horizontally."



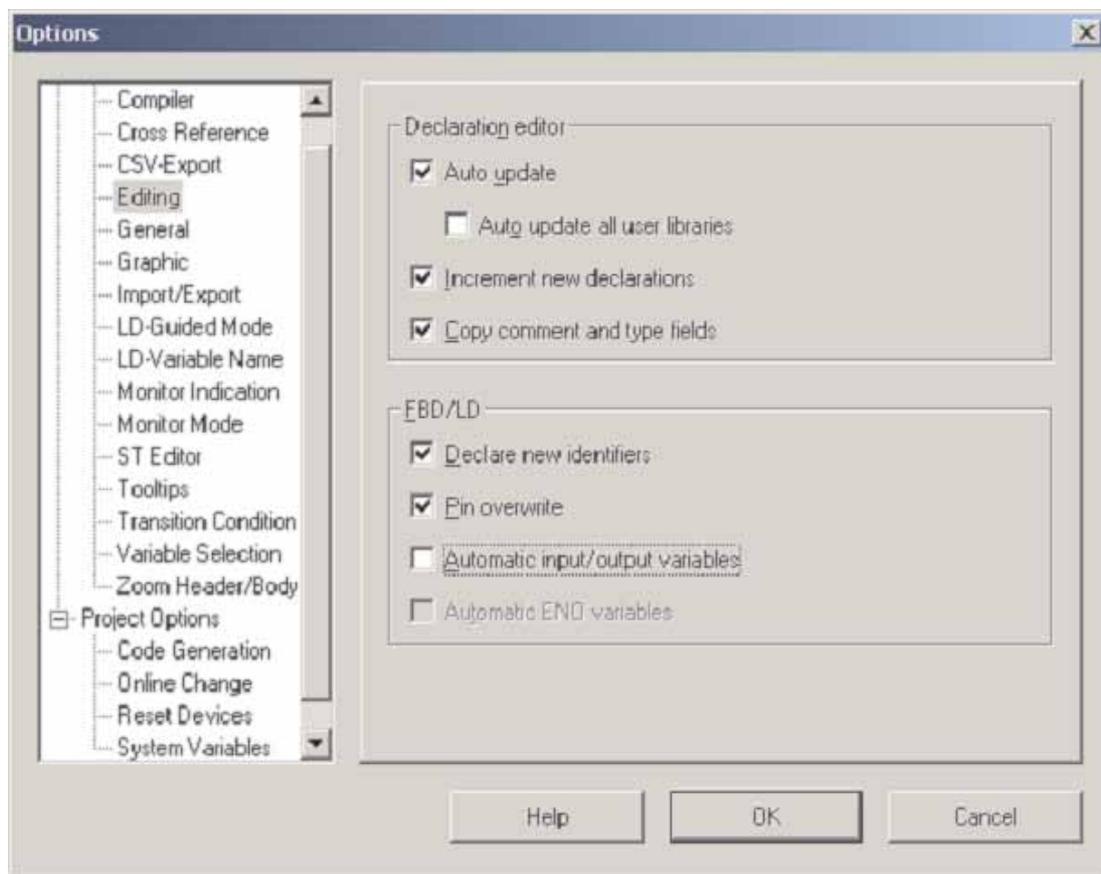
Продолжайте редактировать проект "Carousel"

Введите нормально разомкнутый контакт датчика "In_Position_Sensor" в положении, показанном на текущем экране, так же, как показано ниже:



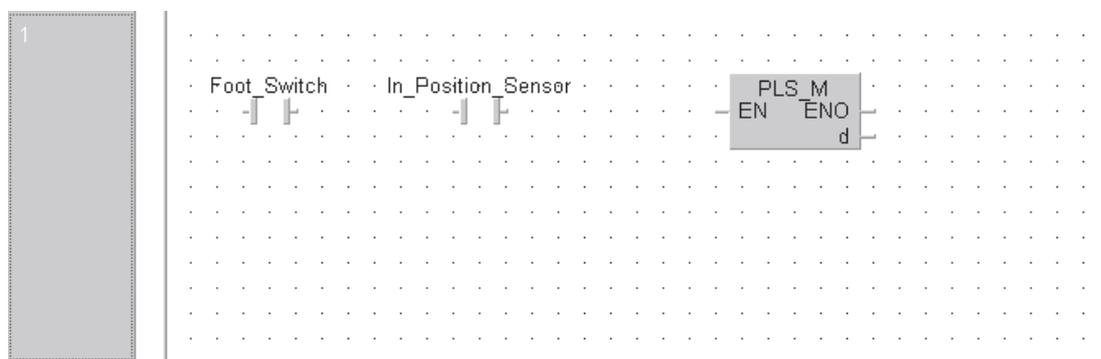
Ввод команды функционального блока в релейную диаграмму

Перед продолжением мы рекомендуем, чтобы в оставшейся части этого учебного курса возможность **Automatic input/output variables** была заблокирована, "Disabled" т.е. эта опция не была выбрана. Эта опция находится в меню **Extras**; используйте **Options** и выберите **Editing**, как показано ниже:



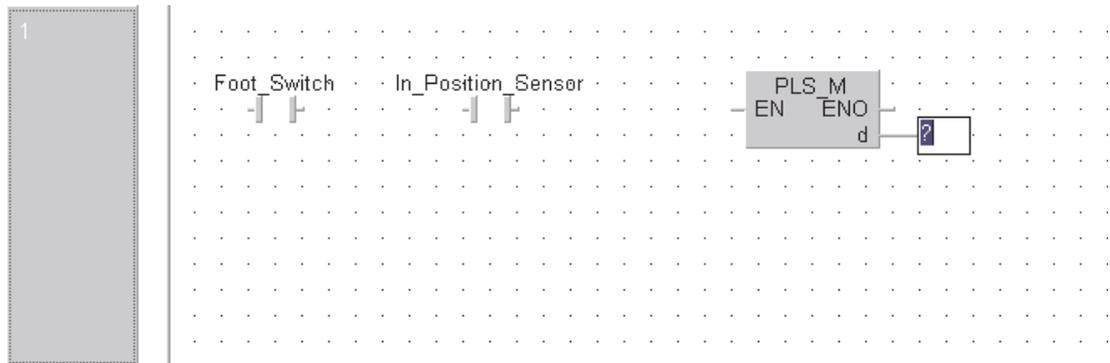
Команда функционального блока MELSEC, "PLS_M", будет добавлена в программу как выходная функция.

- Щелкните на кнопке выбора функции / функционального блока  на панели инструментов. В **Operator type** щелкните на **Functions** и напечатайте "PLS_M" в **Operators** в окне запроса; получим:



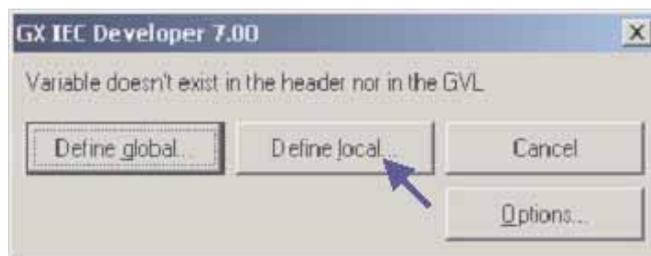
Назначение переменной инструкции

- ② Щелкните на запрос выходной переменной в инструментальной панели . Щелкните на адресате "d", выходной функции PLS_M, чтобы поместить поле запроса переменной.

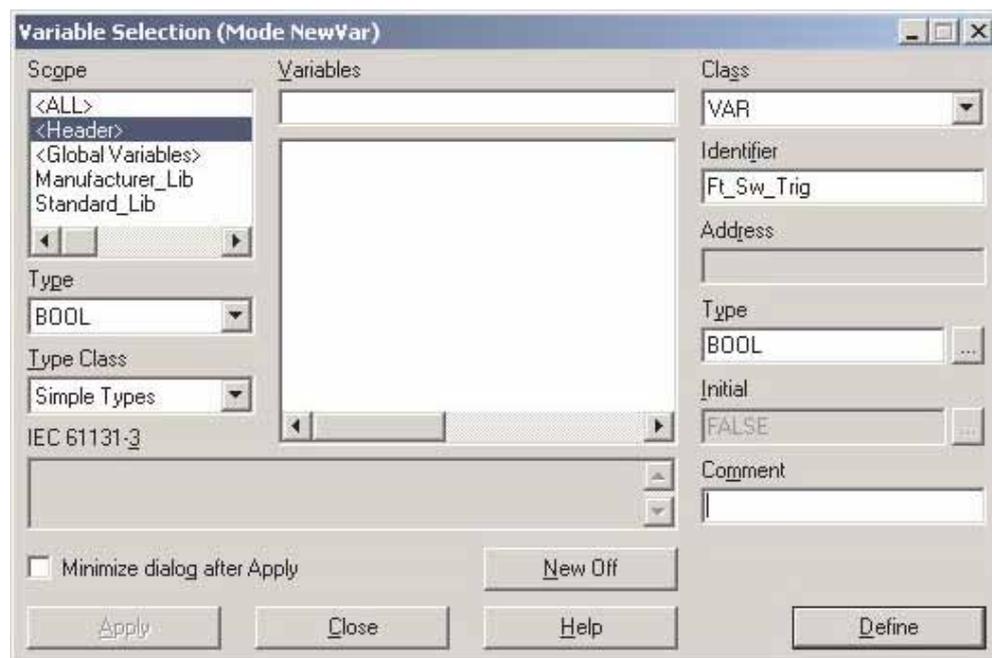


- ③ Введите имя переменной Ft_Sw_Trig в пустое окно "?".

Выводится следующий запрос, если переменная отсутствует в списке локальных переменных "LVL" (локальном заголовке) или в списке глобальных переменных "GVL":



- ④ Щелкните на **Define Local**, чтобы определить новую локальную переменную "LVL". Открывается окно выбора переменной Variable Selection, предлагающее определить новую переменную:

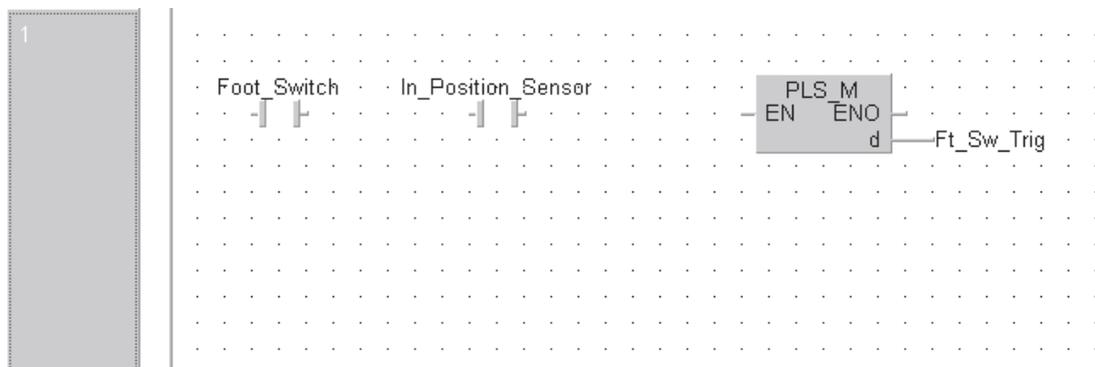


- ⑤ Щелкните на **Define**, чтобы ввести новую переменную в LVL (Локальный заголовок).

ПРИМЕЧАНИЕ

- | Чтобы подтвердить эту операцию, проверьте локальный заголовок!!

Дисплей должен иметь следующий вид:



В заключение необходимо составить схему релейной логики, соединив элементы следующим образом.

- ⑥ Щелкните правой кнопкой мыши в любом месте в окне редактирования и отмените выбор функции **Auto connect**.

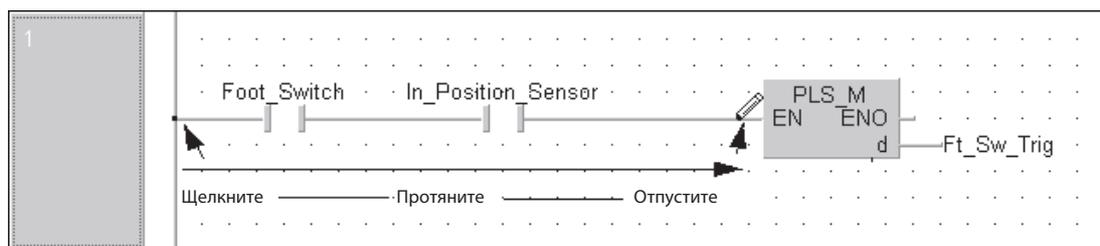


- ⑦ Аналогичным образом, щелкните, чтобы выбрать **Interconnect Mode**.



Обратите внимание - теперь указатель изменил вид, превратившись в небольшой значок карандаша.

- ⑧ Щелкните на левой точке релейной диаграммы, "протяните" указатель мыши вдоль схемы и отпустите его на входе "EN" функции "PLS_M", как показано ниже:



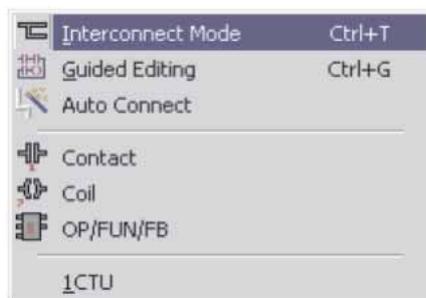
Теперь схема закончена.

Изменение режима курсора

Перед тем, как продолжать работу с нашим примером, необходимо понять работу курсора и различные режимы редактирования, которые имеются в программном пакете.

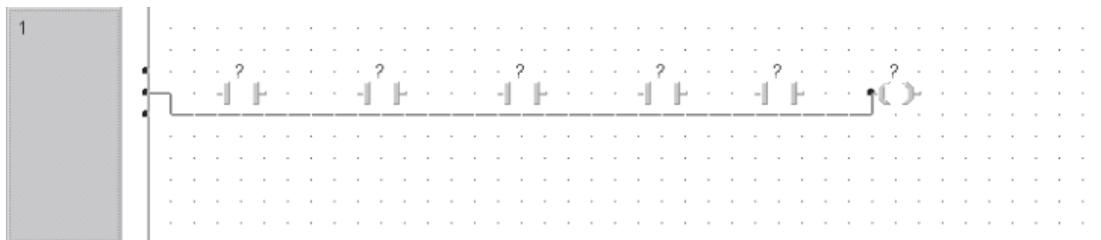
Следующий текст приведен только для иллюстрации.

Находясь на экране редактирования релейной диаграммы, щелкните правой кнопкой мыши - всплывает небольшое окно выбора, как показано ниже. Щелчок на **Auto connect** включает/отключает эту возможность; вы также можете переключаться между ручкой и стрелкой, не пользуясь значками панели инструментов.

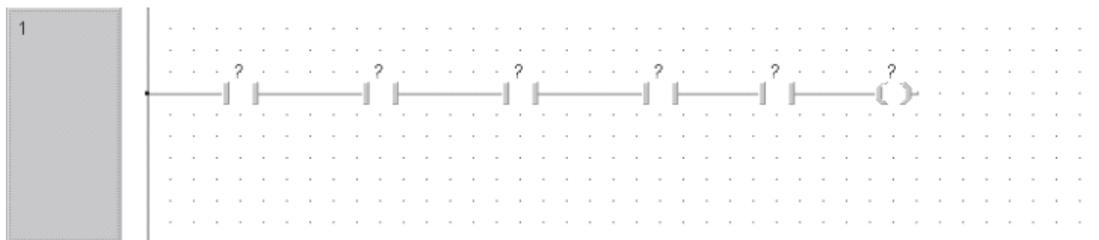


Предосторожности при использовании редактора релейных диаграмм

Как можно видеть на экране ниже, благодаря тому, что **Auto connect** формирует соединение между двумя точками, для ряда контактов он пытается провести следующую соединительную линию. С выбранной опцией **Auto Connect**, единственный способ соединить эти контакты - сделать соединение между каждой отдельной парой:



Ручка позволяет затем пройти через все контакты, от сборной шины до катушки. В редакторе релейных диаграмм мы предлагаем активизировать опцию **Auto connect** при помещении элементы на тело программного модуля или соединении параллельных элементов. Однако, эту опцию следует отключить при подключении ряда контактов, как показано на следующем экране, или вставке контакта в существующую схему.

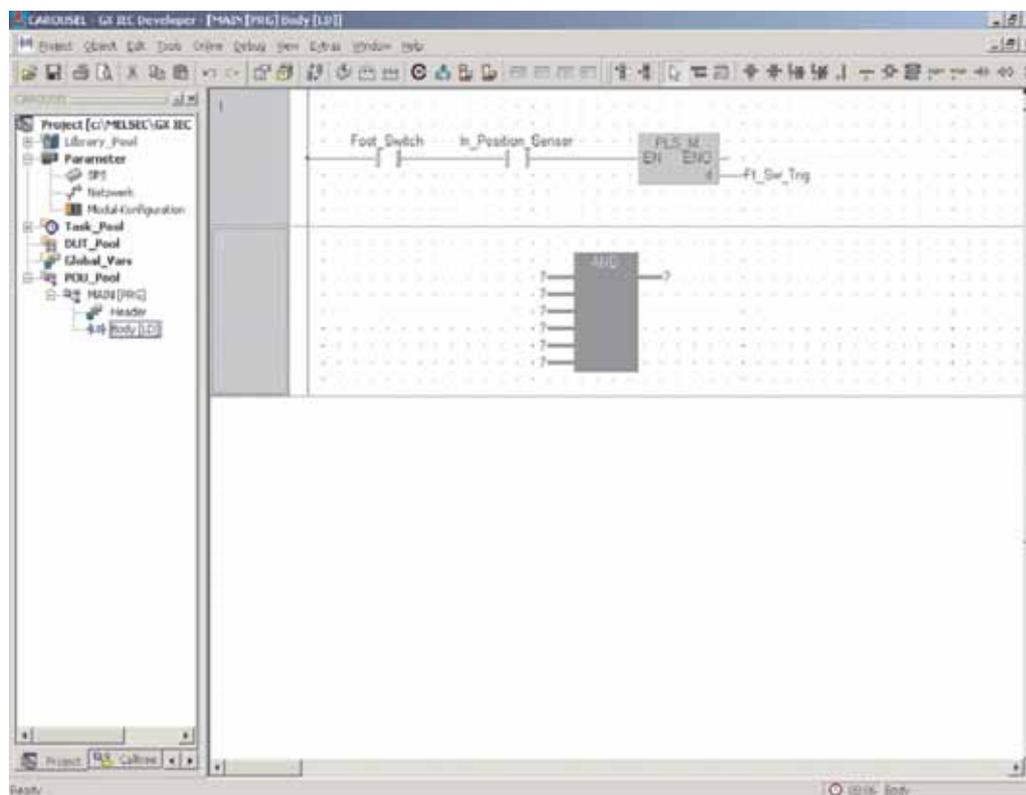


Для соединения нескольких элементов или при добавлении элементов в существующие сети **Auto Connect** должен быть отключен.

Количество входных переменных функций, например, оператор MUL (умножение), можно увеличивать или уменьшать. Для этого щелкните на функции и затем нажмите на соответствующую кнопку на панели инструментов.

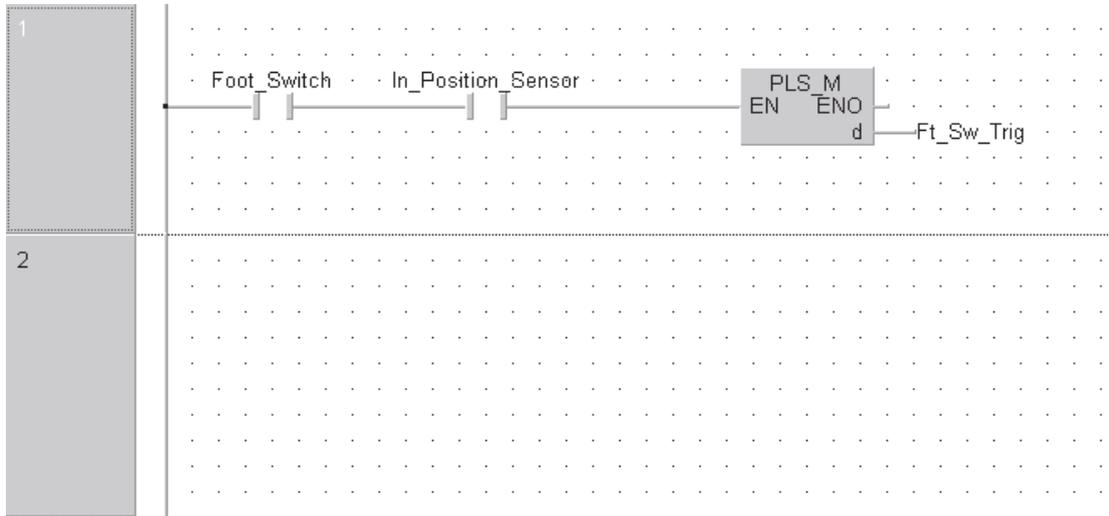


При использовании функций с несколькими входами, например, MUL, число входов можно увеличить/уменьшить, используя специальную панель инструментов с соответствующими значками. Это также можно сделать, поместив курсор на нижний край функции, удерживая левую кнопку мыши, и затем перетащив курсор, как показано ниже:

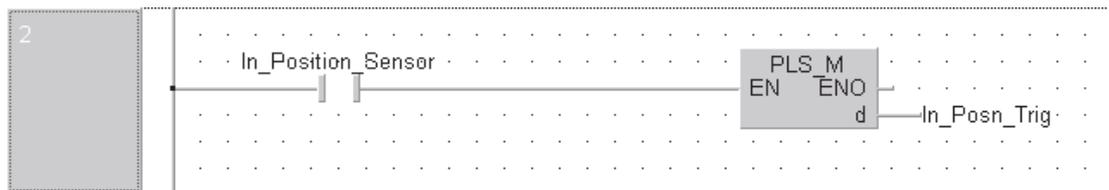


Создание новой цепи программы

- ① Чтобы создать цепь под текущей, щелкните на кнопке "Вставить после" . Появится пустое пространство под цепь:



- ② Введите вторую цепь с тем же форматом, что для описанной выше, со следующими атрибутами:

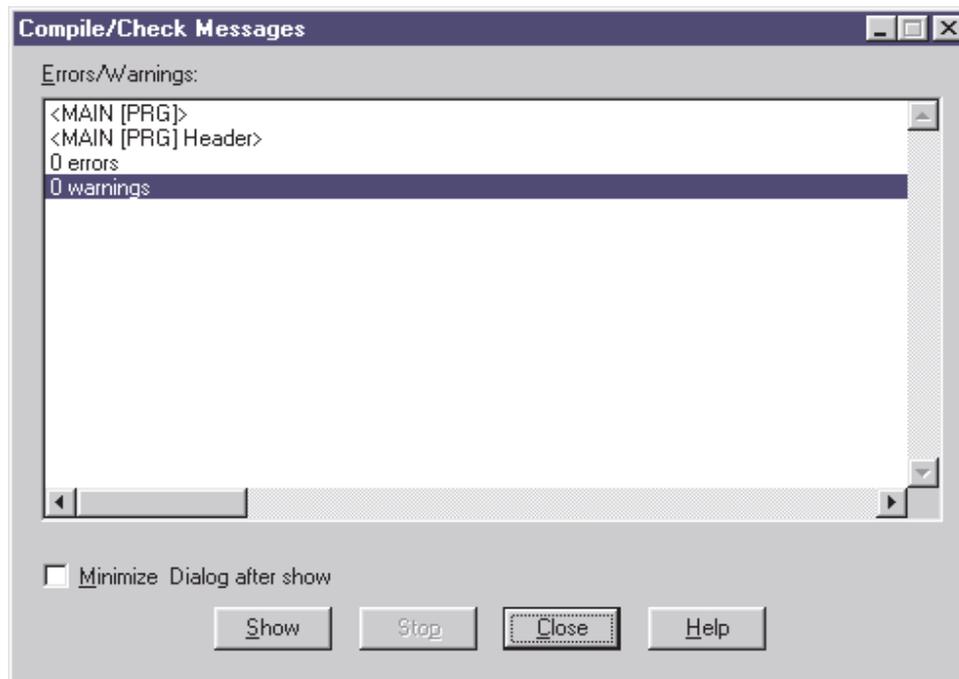


- ③ Наконец, введите следующую цепь, как показано:



Проверка введенной программы

Введя три цепи щелкните на кнопке Проверить  и, если все в порядке, будет показано следующее диалоговое окно:



Добавление новых программных модулей - счетчики и таймеры

Продолжим работу с примером "Управление каруселью". Теперь добавим дополнительные подпрограммы, чтобы проиллюстрировать использование таймерных функций и функций счета.

- Счет числа операций (Счетчик партий продуктов)
- Создайте дополнительный программный модуль, чтобы реализовать функцию счета партий.

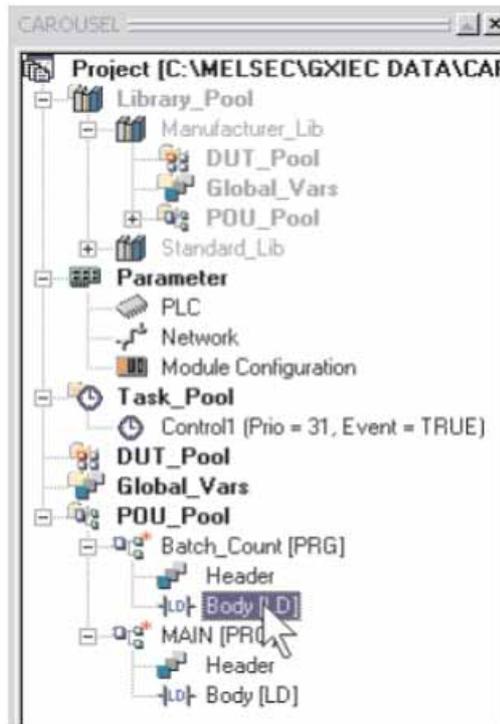
Задача:

Теперь в проект будет добавлен дополнительный программный модуль, чтобы считать количество включений двигателя, т.е. счетчик партий продуктов.

Когда подчитывается десять продуктов, ПЛК будет выдавать пульсирующий выходной сигнал с периодом 1 секунду, пока не будет нажата кнопка, сбрасывающая счетчик партий.

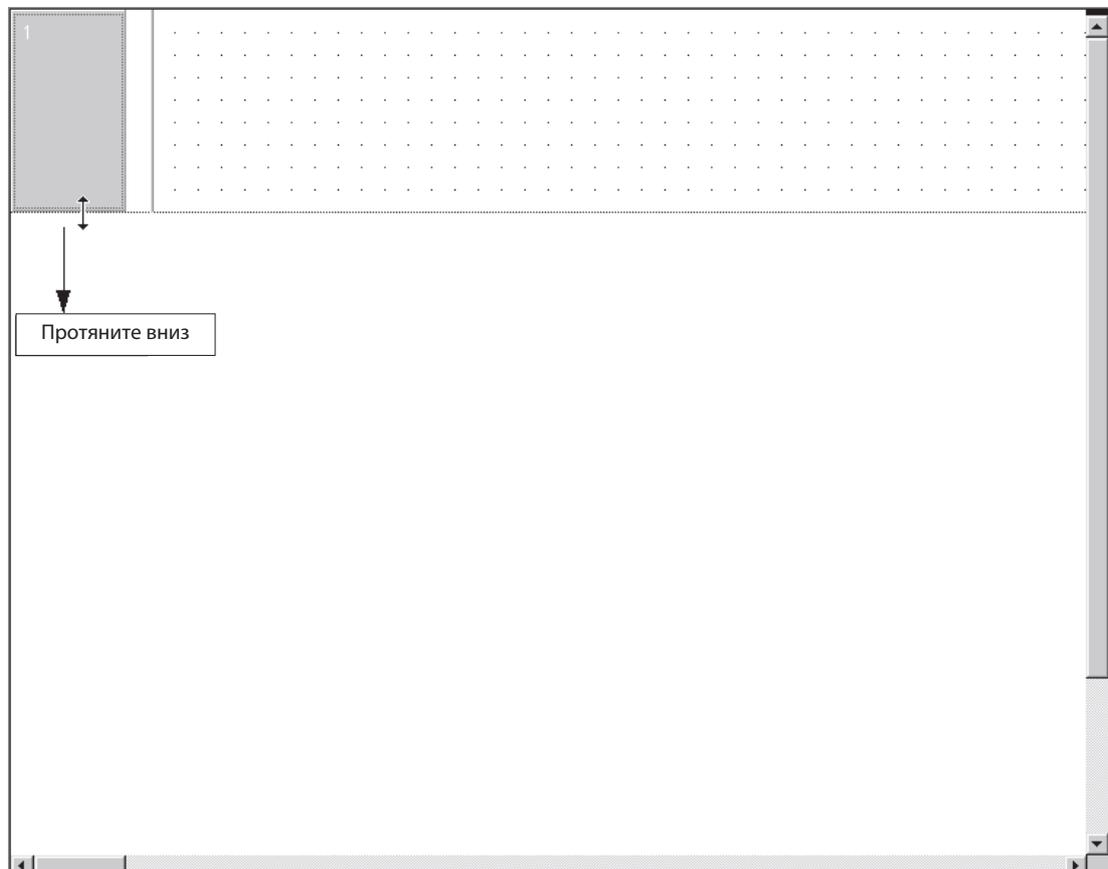
Введите следующий программный модуль подпрограммы релейной диаграммы, используя редакторы "свободных форм", как показано:

① Создайте новый программный модуль, щелкнув на кнопке .



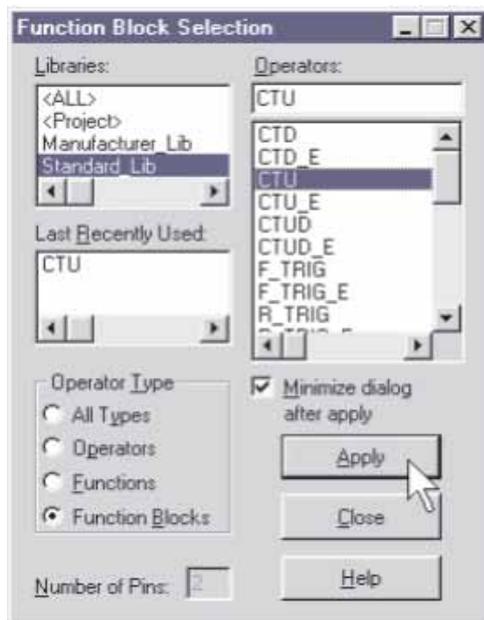
② Выберите тело нового программного модуля, открыв только что созданный элемент в навигационном окне проекта.

Как отмечалось выше, размер схемы релейной логики можно изменять, перемещая указатель мыши к нижней границе заголовка схемы и "протягивая" ее вниз, чтобы увеличить вертикальный размер:

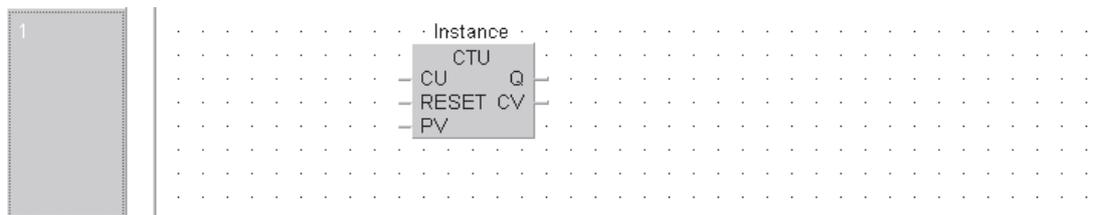


Счетная функция

Используя редактор в режиме "выбора", введите инструкцию CTU (Счет вверх) в схему релейной логики:



Поместите функциональный блок IEC на пустую схему релейной логики:

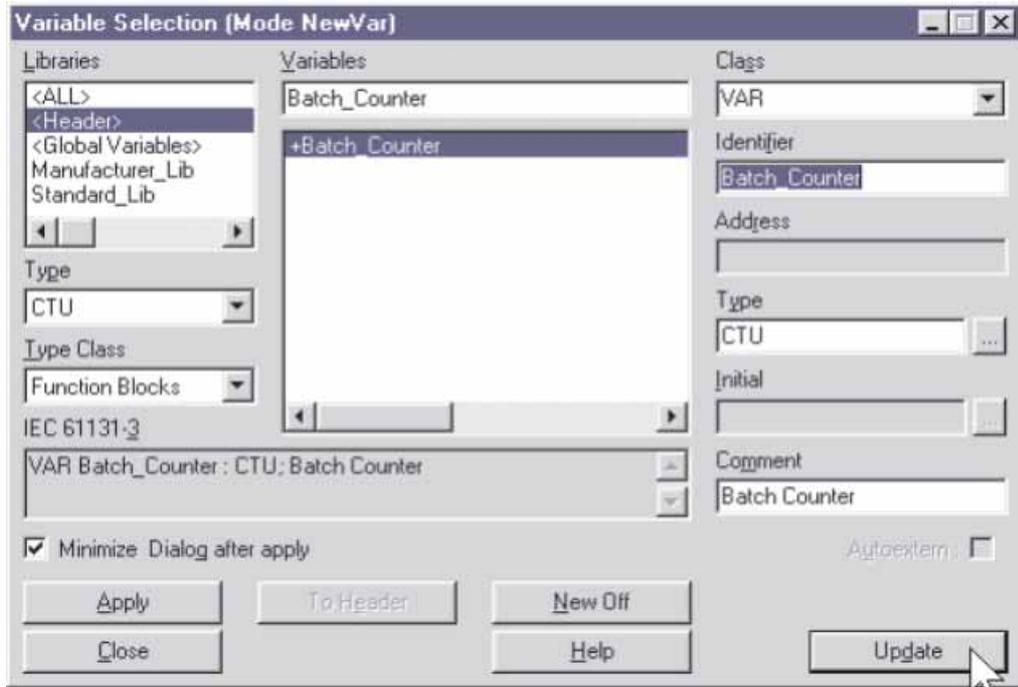


Экземпляры функциональных блоков

Функциональные блоки можно вызывать только как "**Экземпляры**." Процесс создания копии функционального блока выполняется в заголовке программного модуля, в котором должен использоваться экземпляр. В этом заголовке функциональный блок будет объявлен как переменная и результирующему экземпляру присваивается имя. Возможно объявить несколько экземпляров с различными именами из одного функциональных блоков в одном программном модуле. Затем экземпляры вызываются в теле программного модуля и "**фактические**" параметры передаются в "**формальные**" параметры. Каждый экземпляр можно использовать неоднократно.

Ввод функционального блока IEC CTU

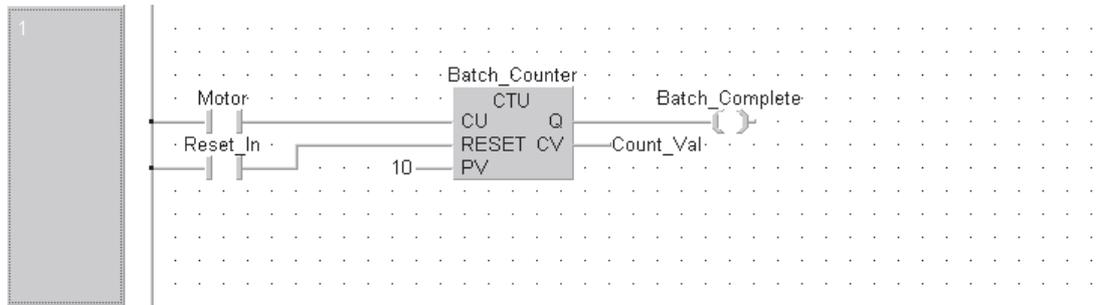
- ① Чтобы создать новое имя для этого экземпляра функционального блока CTU в данном программном модуле, щелкните на имени переменной **Instance** над функциональным блоком CTU. Нажмите F2, чтобы открыть диалоговое окно выбора переменной **Variable selection**. Заполните результирующее окно, как показано на следующей странице.



Щелкните на **Apply**, затем на **Update**, и имя переменной изменится, как показано слева.



Продолжайте, чтобы ввести программу, как описано выше, чтобы получить следующий экран:



Вводя значения PV и CV, используйте кнопки переменных , соответственно.

Добавление элементов в GVL

Заметим, в частности: "Reset_In" (глобальный) - это новый вход, отображенный из булева адреса MELSEC X02 или IEC %IX2. Для этого необходимо новый следующий элемент в GVL:

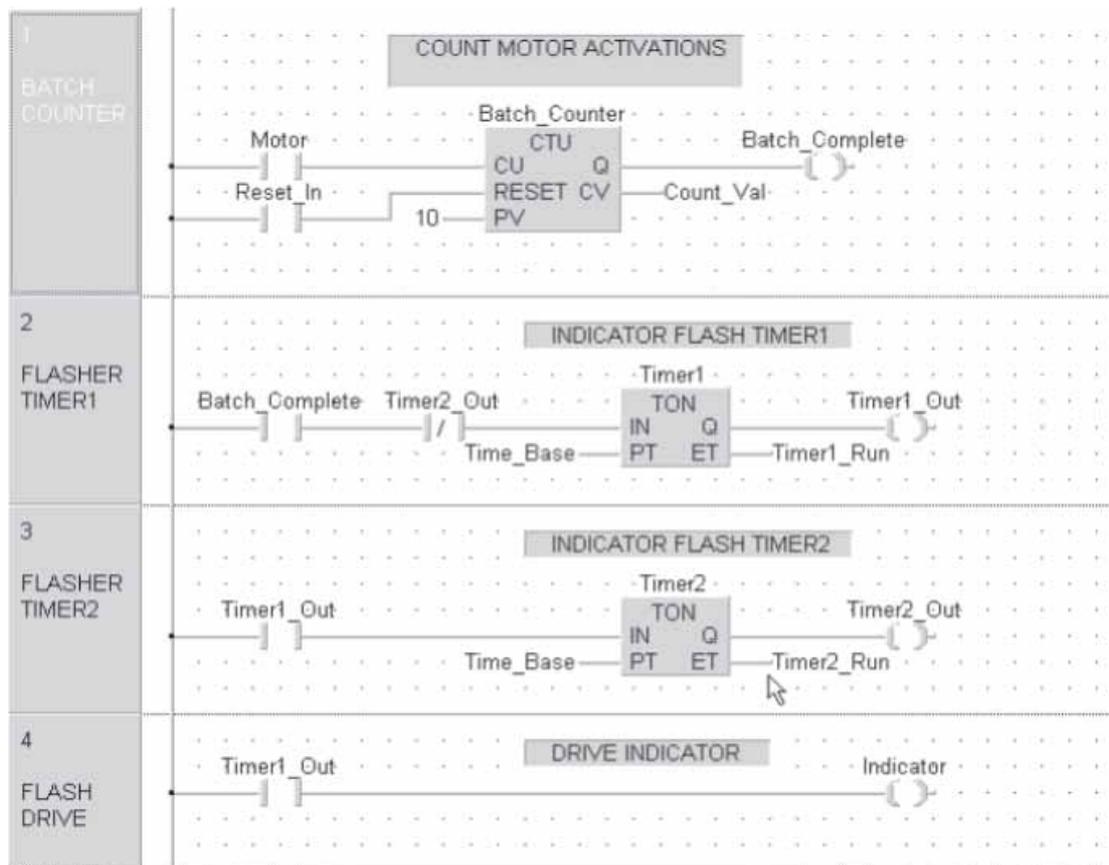
	Class	Identifier	MIT-Addr	IEC-Addr	Type	Initial
0	VAR GLOBAL	Foot Switch	X00	%X16	BOOL	FALSE
1	VAR GLOBAL	In Position Sensor	X01	%X17	BOOL	FALSE
2	VAR GLOBAL	Reset_In	X02	%X18	BOOL	FALSE
3	VAR GLOBAL	Motor	Y00	%Q32	BOOL	FALSE

	Class	Identifier	Type	Initial	Comment
0	VAR	Batch_Counter	CTU	...	Batch Counter
1	VAR	Batch_Complete	BOOL	...FALSE	Batch Complete
2	VAR	Batch_Complete1	BOOL	...FALSE	
3	VAR	Count_Val	INT	...0	

Закончив ввод всех новых элементов, щелкните на кнопке проверки , затем на кнопке "Rebuild All" , чтобы проверить и собрать проект.

Таймерная функция

Создайте следующие цепи релейной диаграммы под подпрограммой счета партий в программном модуле Batch_Count, как показано ниже:



После завершения редактирования список глобальных переменных GVL должен иметь следующий вид:

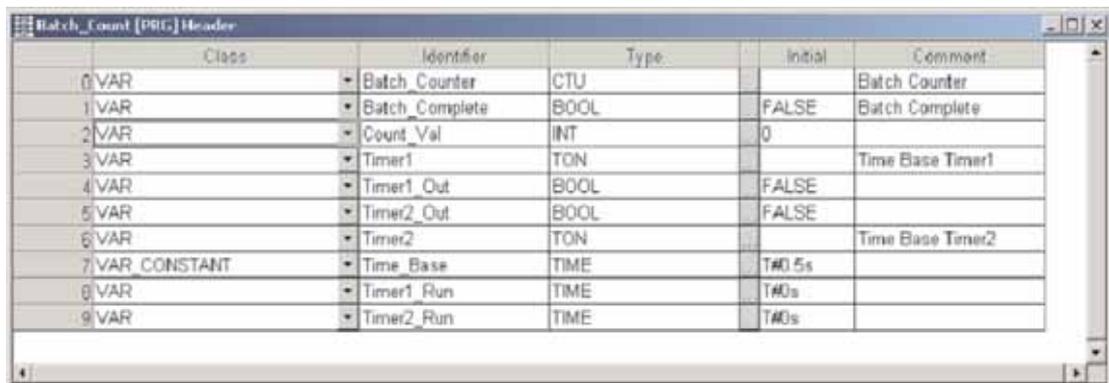
	Class	Identifier	MIT-Addr	IEC-Addr	Type	Initial
0	VAR_GLOBAL	Foot Switch	X00	%IX16	BOOL	...FALSE
1	VAR_GLOBAL	In_Position_Sensor	X01	%IX17	BOOL	...FALSE
2	VAR_GLOBAL	Reset_In	X02	%IX18	BOOL	...FALSE
3	VAR_GLOBAL	Motor	Y02	%QX32	BOOL	...FALSE
4	VAR_GLOBAL	Indicator	Y21	%QX33	BOOL	...FALSE

Теперь заголовок (LVL) для обсуждавшейся выше программы "Batch_Count" имеет следующий вид:

	Class	Identifier	Type	Initial	Comment
0	VAR	Batch_Counter	CTU	...	Batch Counter
1	VAR	Batch_Complete	BOOL	FALSE	Batch Complete
2	VAR	Count_Val	INT	0	
3	VAR	Timer1	TON	...	Time Base Timer1
4	VAR	Timer1_Out	BOOL	FALSE	
5	VAR	Timer2_Out	BOOL	FALSE	
6	VAR	Timer2	TON	...	Time Base Timer2
7	VAR_CONSTANT	Time_Base	TIME	T#0.5s	
8	VAR	Timer1_Run	TIME	T#0s	
9	VAR	Timer2_Run	TIME	T#0s	

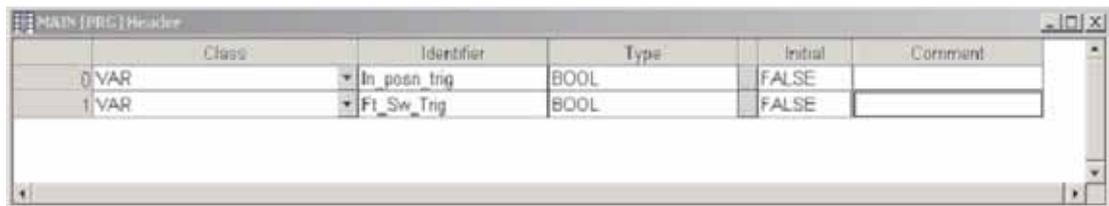
Закончив ввод всех новых элементов, щелкните на кнопке проверки , затем на кнопке "Rebuild All" , чтобы проверить и ассемблировать проект.

Заголовок для программного модуля "Batch_Count"



	Class	Identifier	Type	Initial	Comment
0	VAR	Batch_Counter	CTU	...	Batch Counter
1	VAR	Batch_Complete	BOOL	FALSE	Batch Complete
2	VAR	Count_Val	INT	0	
3	VAR	Timer1	TON	...	Time Base Timer1
4	VAR	Timer1_Out	BOOL	FALSE	
5	VAR	Timer2_Out	BOOL	FALSE	
6	VAR	Timer2	TON	...	Time Base Timer2
7	VAR_CONSTANT	Time_Base	TIME	T#0.5s	
8	VAR	Timer1_Run	TIME	T#0s	
9	VAR	Timer2_Run	TIME	T#0s	

Заголовок для программного модуля "MAIN":

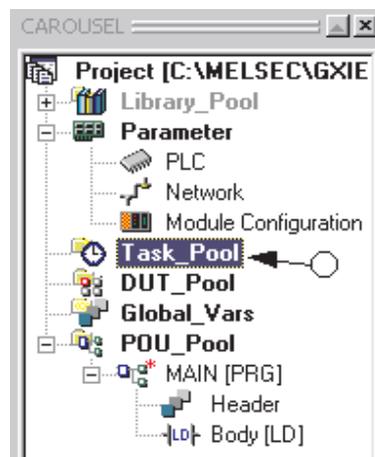


	Class	Identifier	Type	Initial	Comment
0	VAR	In_posn_trig	BOOL	FALSE	
1	VAR	Ft_Sw_Trig	BOOL	FALSE	

4.2.6 Создание новой задачи

Для того, чтобы программные модули "MAIN" и "Batch_Count" были ассемблированы и выполнены в ПЛК, они должны быть заданы как допустимые задачи в задачном пуле **Task Pool**.

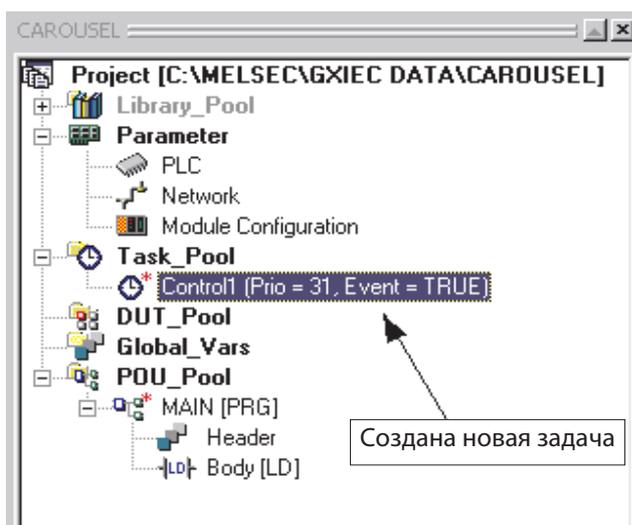
- ① Щелкните один раз, чтобы выделить значок **TASK_Pool** в навигационной области проекта.



- ② Затем щелкните на кнопке задач  на панели инструментов. Альтернативно, щелкните правой кнопкой на значке задачного пула в навигационном окне проекта и выберите из меню опцию **New Task**.
- ③ Введите имя новой задачи ("Control1") в окне запроса.



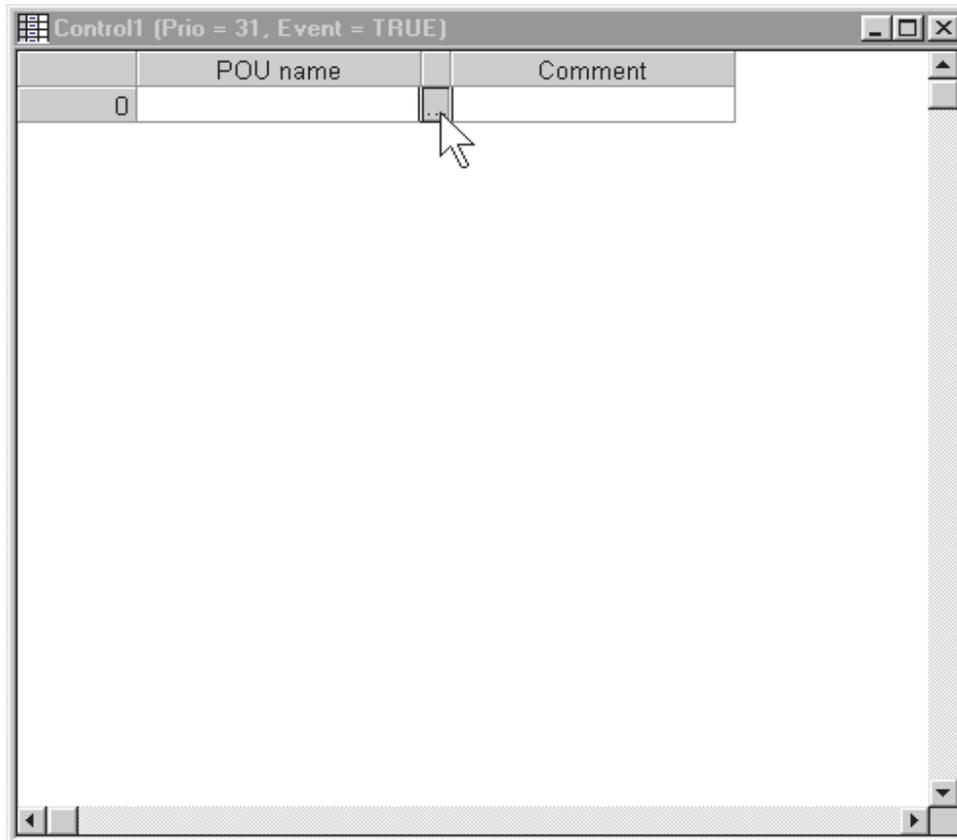
- ④ Щелкните на кнопке **OK** и в окне Project Navigation теперь показана только что созданная задача "Control1":



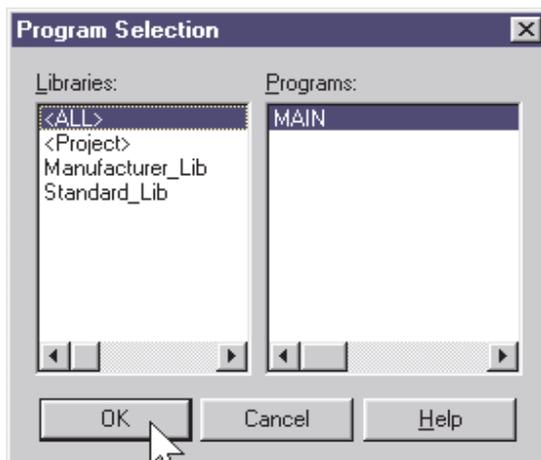
Назначение программного модуля задаче

Только что созданная задача "Control1" должна теперь ссылаться на программный модуль.

- 1 Дважды щелкните на значке задачи "Control1" Task в навигационном окне проекта; будет показано окно "список событий задачи":



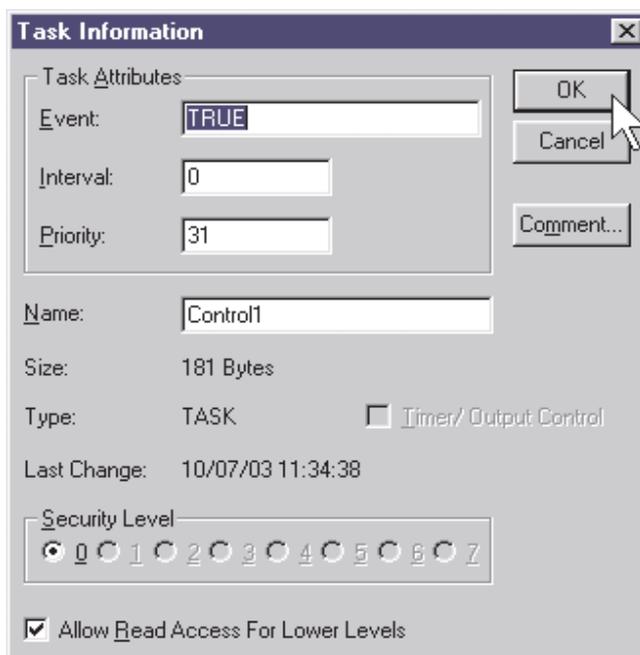
- 2 Щелкните на среднем многоточии "просмотр вариантов", как показано выше. Будет показано следующее диалоговое окно:



- 3 Выберите MAIN и щелкните на **OK**, чтобы закончить операцию назначения.

Свойства задачи

Чтобы показать свойства задачи, щелкните правой кнопкой мыши на требуемом элементе задачного пула (т.е. Control1) и выберите **Properties** из меню. Будет показано следующее окно настроек задачи:



- Атрибуты задачи

- **Event** = TRUE: Всегда выполнять
- **Interval** = 0: Устанавливать на нуль, потому что событие **Event** всегда истинно.
- **Priority** = 31: 31 является наимизшим приоритетом, который сканируется последним.

Перед продолжением неплохо сохранить ("SAVE") проект; щелкните на кнопке Save .

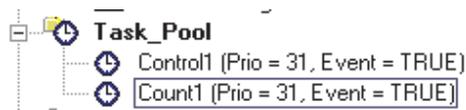
Создание новой задачи для программного модуля "Batch-Count"

Программный модуль "Batch-Count" также должен вызываться задачей в "задачном пуле".

- ① Чтобы создать новую задачу, щелкните правой кнопкой на значке **Task_Pool** значок в окне Project Navigation (PNW) и выберите **New Task** из представленного меню. Альтернативно, следуйте предыдущей процедуре, щелкните один раз на значке Task_Pool, чтобы выделить его на PNW и щелкните на значке "New Task"  на панели инструментов.
- ② Введите имя "Count1" в окно запроса, как показано:



Под предыдущей задачей "Control1" в задачном пуле появится новая задача:



- ③ Дважды щелкните на значке новой задачи "Count1" в PNW.
- ④ Присвойте остающийся программный модуль этой задаче:

	POU name	Comment
	Batch_Count	...

Закончив, щелкните на кнопке проверки , затем на кнопке "Rebuild All" , чтобы проверить и ассемблировать проект.

Сохраните проект, используя кнопку сохранения . Проект теперь закончен и поэтому должен быть передан в ПЛК.

4.2.7 Документирование программы

Заголовок звена

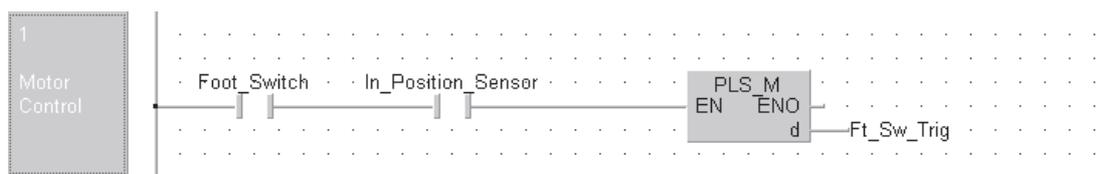
Использование заголовка звена является опциональным; он позволяет идентифицировать звено программы описательным названием длиной до 22 символов. Это может помочь в работе с проектами с большим количеством звеньев.

- 1 Выбрав звено 1, щелкните на кнопке заголовка **Network Header**  или дважды щелкните указателем мыши в области заголовка звена область и введите следующие данные ТОЛЬКО в поле названия название **Title** ТОЛЬКО - на этом этапе оставьте поле метки **Label** **незаполненным** – оно имеет другую функцию:



Введите комментарий в поле Title. Пожалуйста, на этом этапе ничего не вводите в поле Label – это поле имеет специальное назначение и пока не должно заполняться.

- 2 Щелкните на кнопке **OK**, и в левой части экрана будет показан заголовок звена:



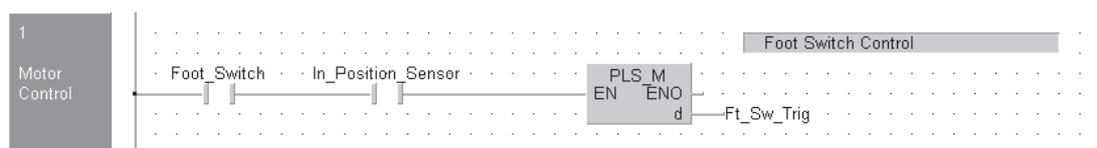
Учтите, что для названия может потребоваться предварительное форматирование (дополнительные пробелы), в зависимости от установленного разрешения экрана, чтобы текст правильно читался с автоматическим переходом слов на следующую строку и вписывался в доступное горизонтальное пространство (макс. 22 символа).

Комментарии к звену

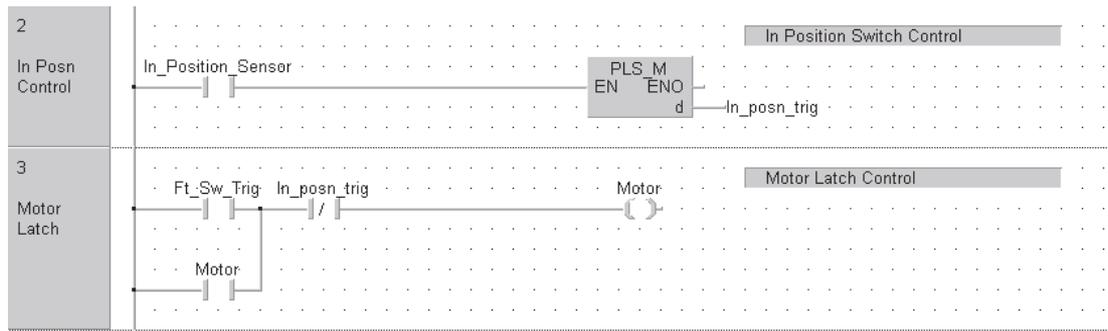
Комментарии позволяют добавлять описатели с фактически ручным текстом в любом месте в звене релейной диаграммы. Они жизненно важны, чтобы обеспечить описание работы программы.

- 1 Чтобы создать комментарий, нажмите "кнопку комментария"  на панели инструментов.

- 2 Указатель мыши изменится на ; щелкните левой кнопкой мыши в точке, куда необходимо поместить комментарий, напечатайте необходимый текст и нажмите :



Продолжайте, чтобы закончить документацию программы следующим образом:



Изменение позиции комментария

Когда курсор находится в "Выборе режима", можно захватить и перемещать комментарии в области схемы релейной логики. Для этого щелкните на левой части диалогового окна комментария и удерживайте кнопку мыши. Переместите комментарий в любое место на экране и отпустите кнопку мыши.

Удаление комментария

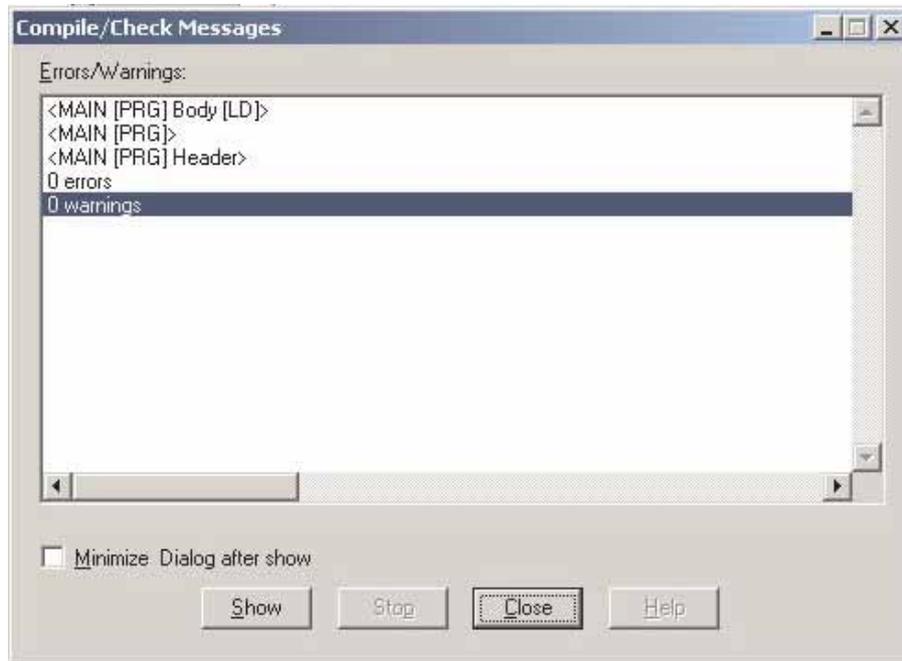
Щелкните один раз на комментарии, чтобы выделить его, и нажмите кнопку.

Вырезание / копирование комментария

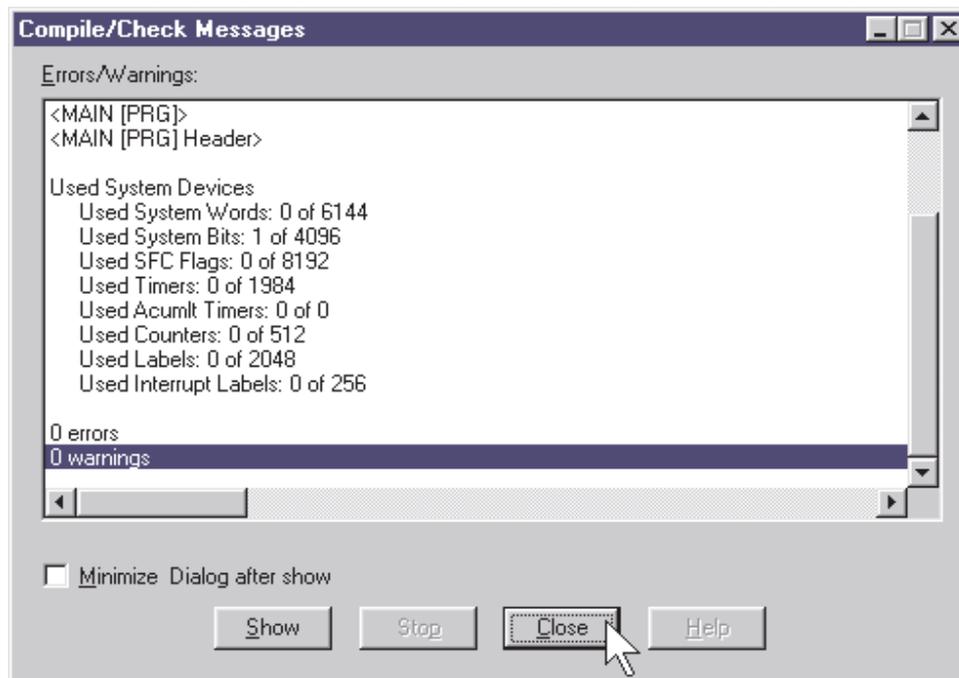
Для дублирования комментариев щелкните на левом конце исходного комментария, чтобы выделить его. Используйте процедуру вырезания/копирования - вклеивания Windows и щелкните мышью еще раз, чтобы установить конечную позицию комментария в другом звене.

4.2.8 Проверка и построения кода проекта

- ① Когда релейная диаграмма закончена и задача задана в задачном пуле, еще раз нажмите кнопку "Check"  на панели инструментов, чтобы проверить программу на наличие ошибок; должно быть показано следующее диалоговое окно:



- ② Щелкните на кнопке "Build"  или на кнопке "Rebuild All"  на панели инструментов и, если все в порядке, будут показаны следующие сообщения компилятора:

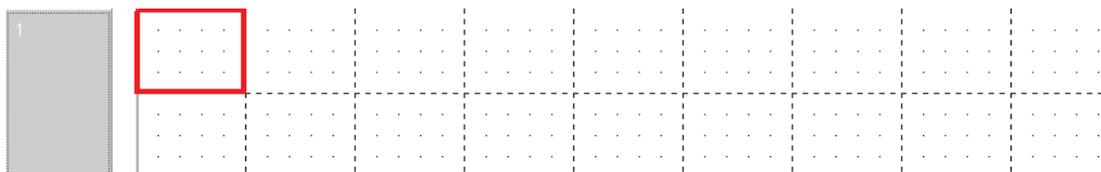


- ③ Щелкните на **Close**, чтобы выйти из этого экрана.

4.2.9 Иллюстрация: Направляемый режим ввода релейной диаграммы

В дополнение к ручным методам ввода релейных диаграмм, в GX-IEC Developer версии 6 имеется продвинутая возможность монитора направляемого метода ввода релейной диаграммы, который может использоваться для помощи при вводе релейной диаграммы. Этот метод ввода может оказаться полезным тем, кто желает перейти к GX-IEC Developer и уже знаком с пакетом MEDOC и GX-Developer Mitsubishi.

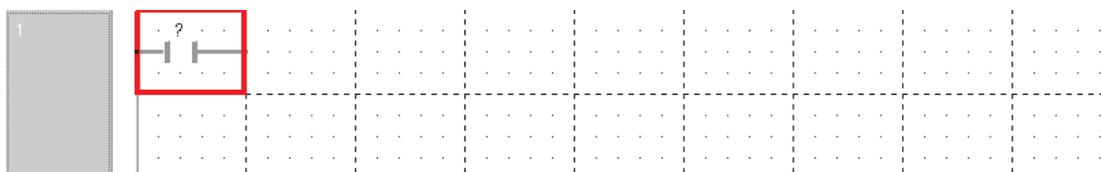
- ① Войдите в режим **Guided Entry Monitor**, нажав кнопку  на панели инструментов. В области редактирования появится следующая матрица:



- ② Используйте следующие кнопки на панели инструментов, чтобы выбирать символы релейной диаграммы. Можно нажать кнопку соответствующего числа, чтобы выбирать соответствующий символ с клавиатуры, что устраняет необходимость в использовании мыши:



- ③ Выберите символ "нормально разомкнутого" контакта "1", и будет показано следующее:



Программа может продолжать ввод с использованием кнопки "F2" на клавиатуре или путем щелчка на кнопке  на панели инструментов. Будет открываться описанное выше окно выбора переменных.

4.3 Процедуры загрузки проекта

4.3.7 Подключение с помощью периферийных устройств

Следующие инструкции описывают, как проект загружается в ПЛК серии FX. Чтобы подключить контроллер серии FX к ПК, используется конвертор SC 09, преобразующий последовательные сигналы общего режима RS232, идущие "в и из" компьютера, в последовательный дифференциальный формат RS 422, необходимый для ПЛК.



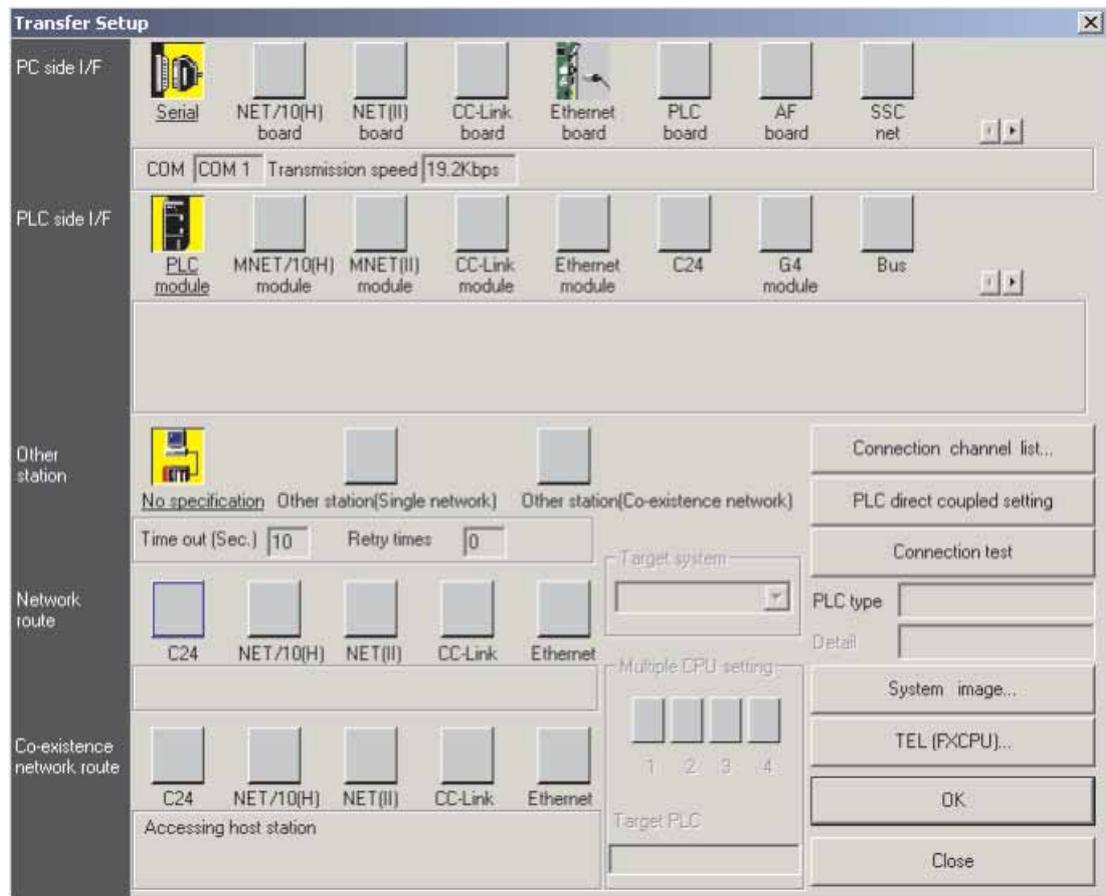
4.3.8 Конфигурация коммуникационного порта

Перед тем, как проект можно будет в первый раз загрузить в ЦП ПЛК, необходимо сконфигурировать настройки связи и загрузки.

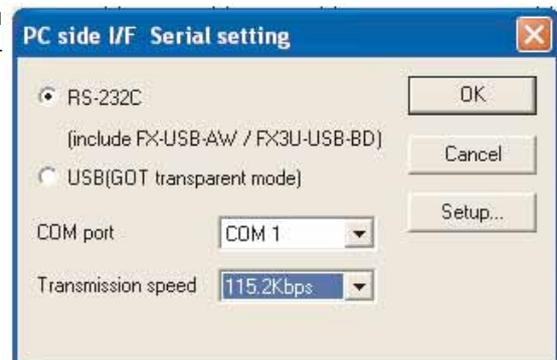
① Из меню **Online** выберите **Transfer Setup** и затем **Ports**:



Будет показано окно **Connection Setup**, представленное на следующей странице.

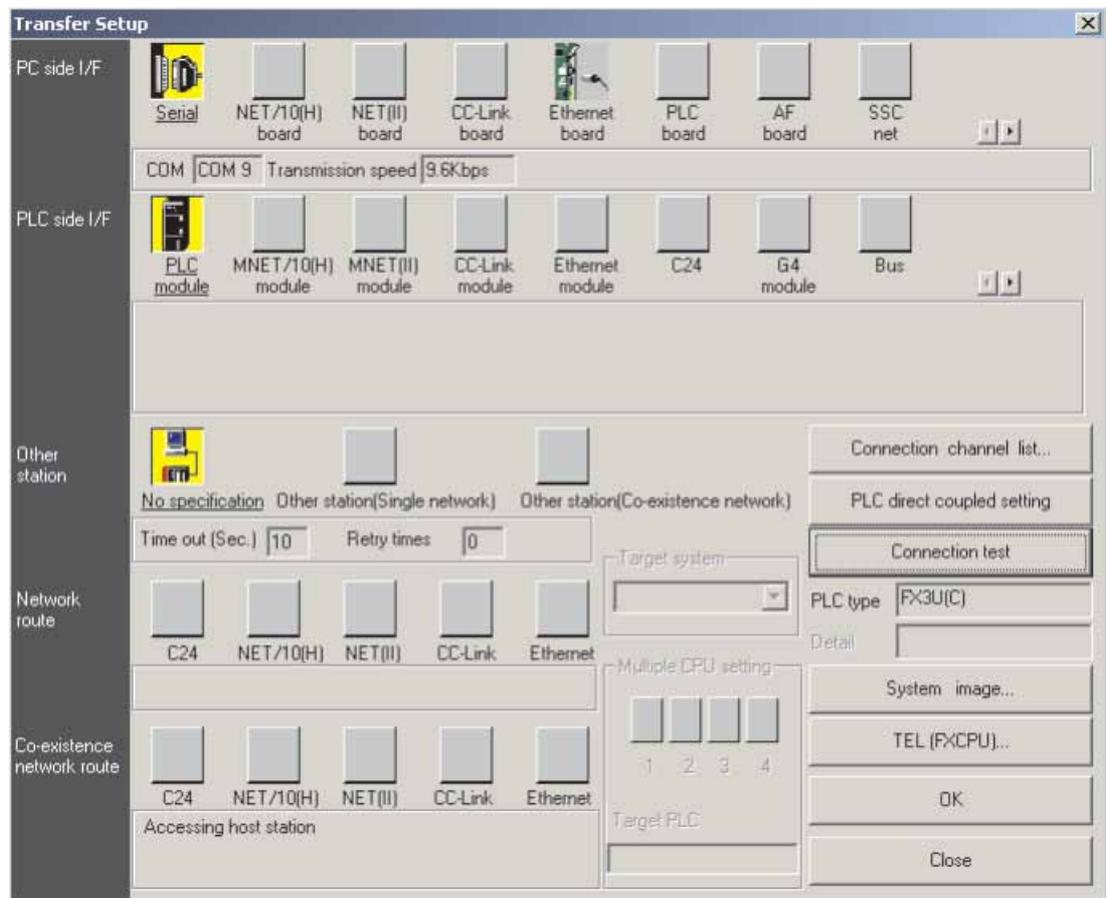


- ② Дважды щелкните мышкой на желтой кнопке **PC side I/F - Serial**, и откроется следующее диалоговое окно:

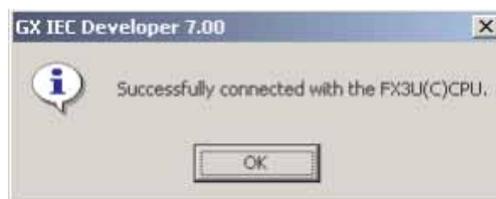


- ③ Выберите **RS232C**, как показано выше, и щелкните на **OK**.

- ④ Щелкните на кнопке **Connection Test**, чтобы проверить наличие связи между ПК и ПЛК:



Должно появиться следующее сообщение:

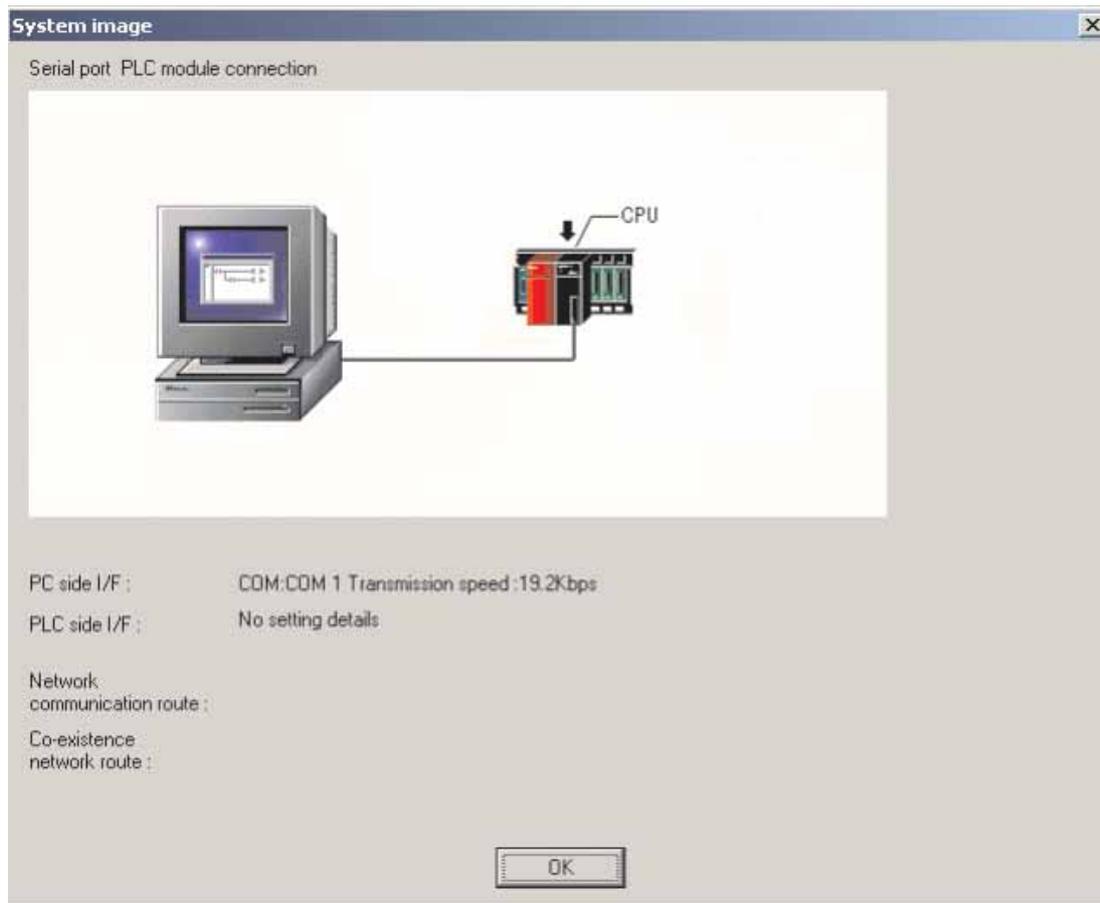


- ⑤ Щелкните на **OK**, чтобы закрыть это сообщение.

Если появится сообщение об ошибке, проверьте соединения и настройки с ПЛК.

Процедура установки соединения

- ① Чтобы получить наглядное представление о процедуре установки соединения, выберите кнопку **System Image**.



- ② Щелкните на кнопке **OK**, чтобы очистить дисплей.

ПРИМЕЧАНИЕ

При использовании стандартного последовательного порта RS232 для связи с ПЛК, если к выбранному COM (n) интерфейсу уже подключено другое устройство, например, последовательная мышь, выберите другой свободный последовательный порт.

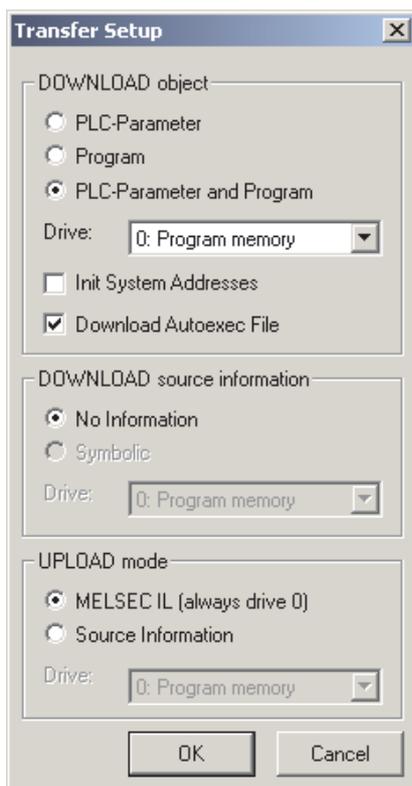
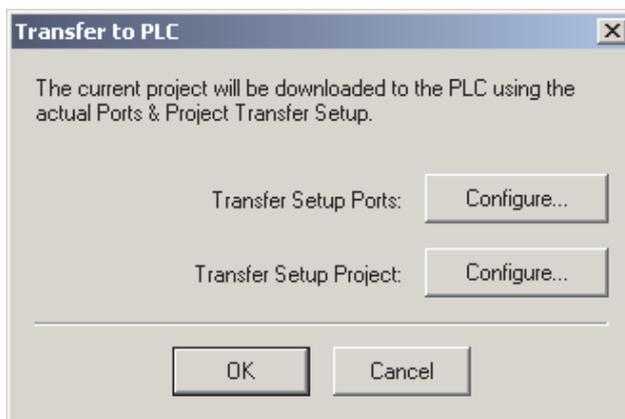
- ③ Выберите **OK**, чтобы закрыть окно **System image** и вернуться в окно **Connection setup**. Затем щелкните на кнопке **OK**, чтобы закрыть окно **Connection Setup**. Если вы выйдете из окна **Connection Setup**, используя кнопку **Close**, настройки не будут сохранены.

4.3.9 Загрузка проекта

- ① Завершив процедуры настройки, щелкните на значке "Download Project"  на панели инструментов.

Настройки передачи

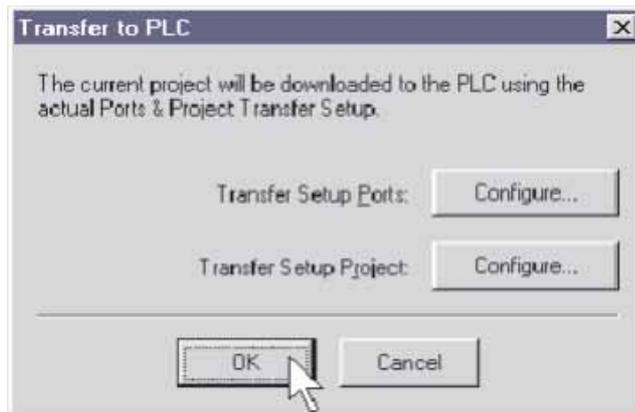
- ② Щелкните на кнопке **Configure**, чтобы настроить "Параметры передачи" для проекта.



- ③ Щелкните на **PLC-Parameter and Program**

- ④ Щелкните на **OK**, чтобы подтвердить выбор.

- ⑤ Для передачи проекта в ПЛК щелкните на кнопке **OK**, чтобы выполнить передачу.

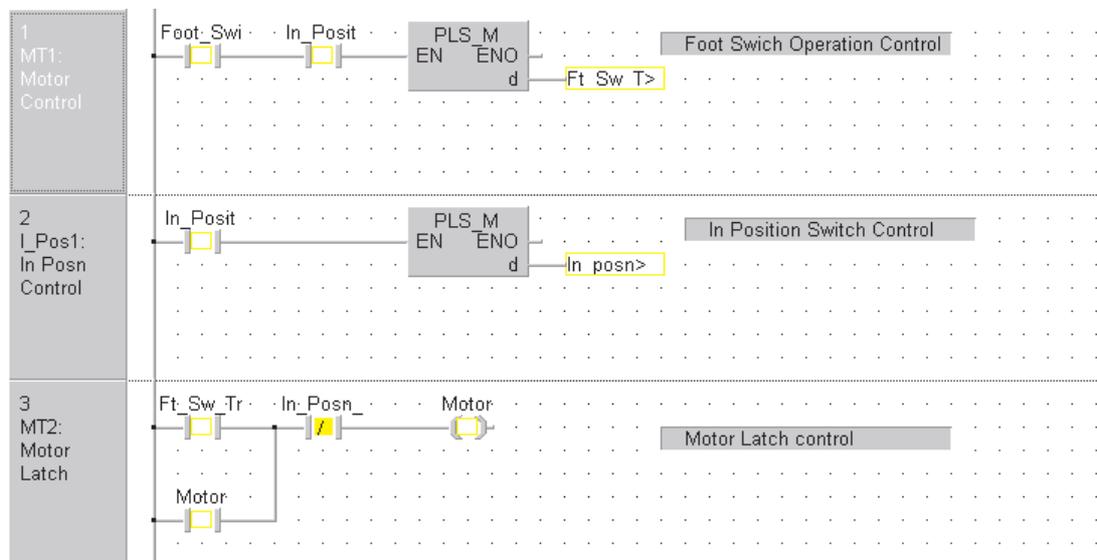


4.4 Мониторинг проекта

Проверьте, что ПЛК переключен в режим RUN и отсутствуют ошибки.

Выведите на дисплей тело релейной диаграммы MAIN.

Щелкните на значке режима мониторинга  на панели инструментов и наблюдайте экран релейной диаграммы:



ПРИМЕЧАНИЕ

В зависимости от установленных цветовых атрибутов переменные будут показаны с цветным фоном (по умолчанию: желтый). Значения любой аналоговой переменной будут показаны на соответствующих контролируемых звеньях диаграммы.

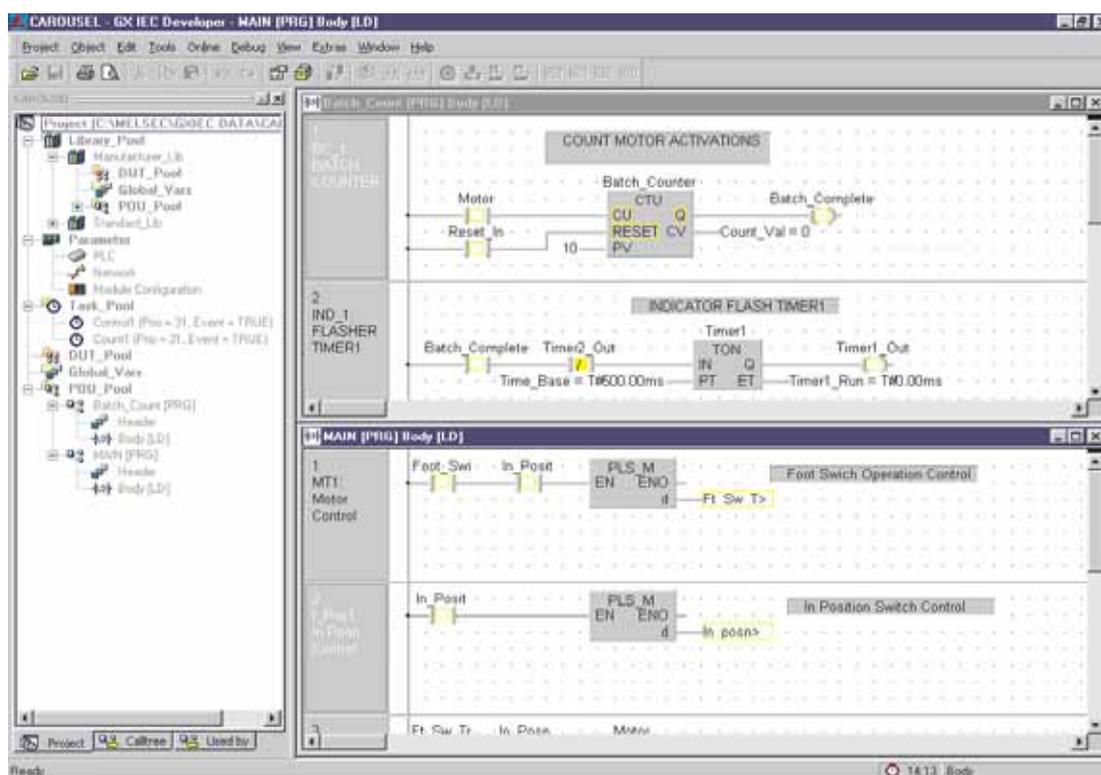
4.4.7 Раздельный / Многооконный мониторинг

Чтобы одновременно контролировать оба программных модуля проекта, откройте тела обоих программных модулей тела и выберите **Tile Horizontally** из меню **Window**.

ПРИМЕЧАНИЕ

Важно: Следует отметить, что при входе в режим мониторинга с помощью , будет контролироваться только передний экран. Это позволяет избежать ненужного коммуникационного трафика с другими экранами, которые были открыты, но не нужны на переднем плане (т.е. открыты, но находятся на заднем плане).

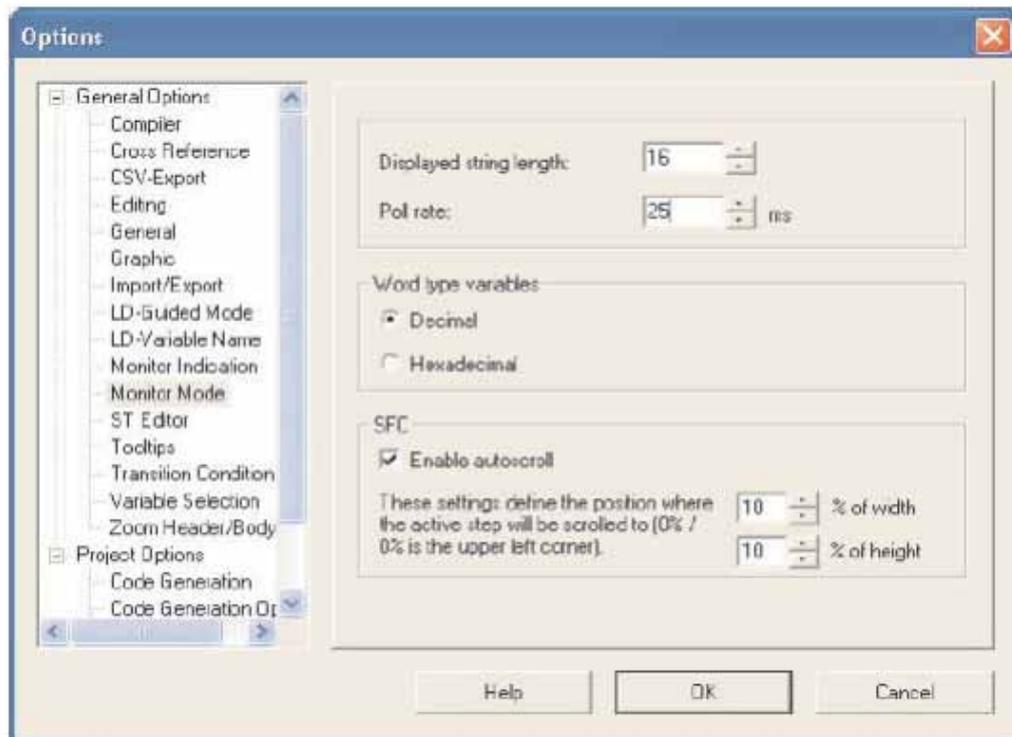
Чтобы начать мониторинг содержания от дополнительных окон, щелкните внутри данного окна и выберите **Start Monitoring** из меню **Online**:



ПРИМЕЧАНИЕ

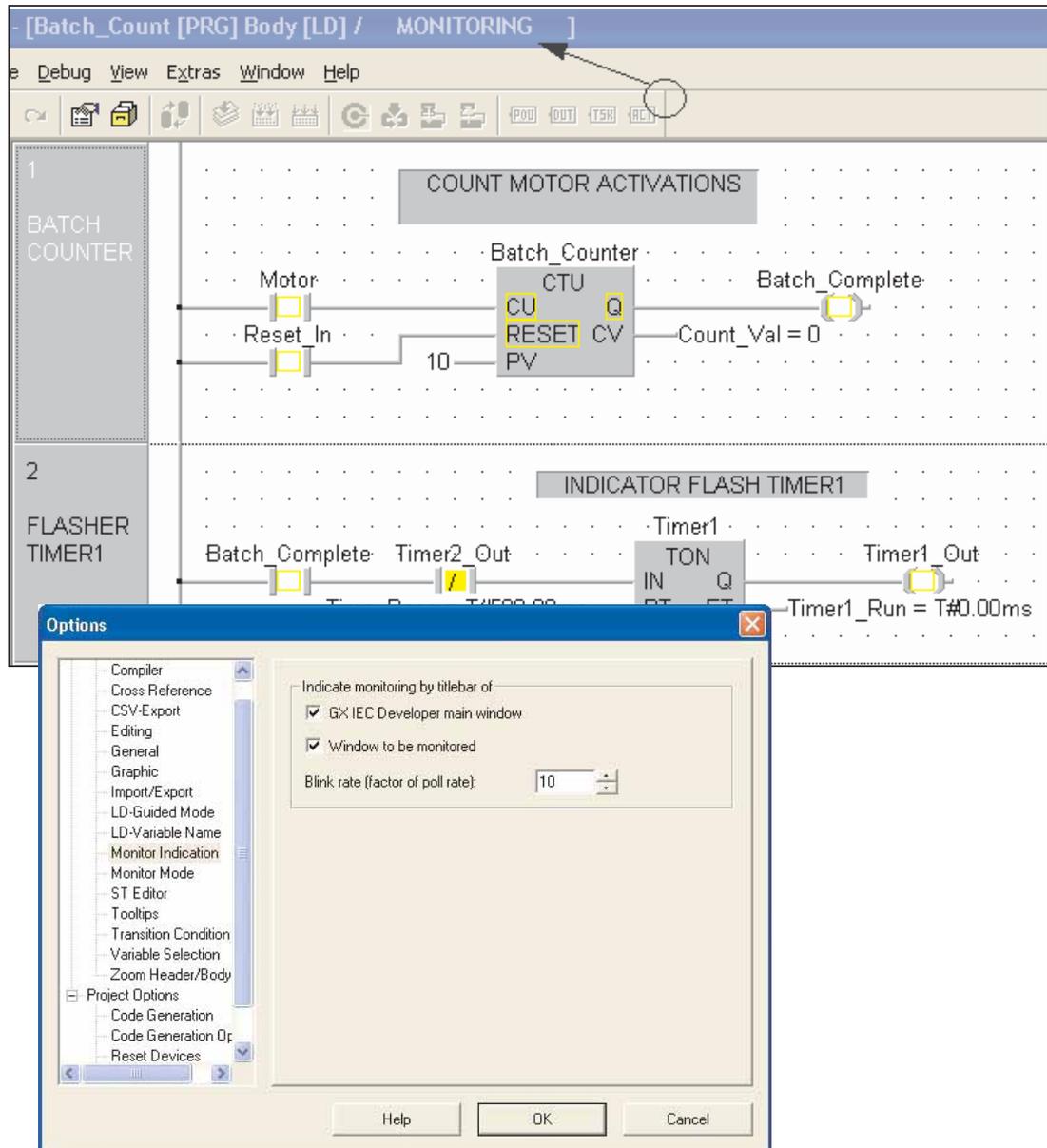
Вследствие квитиования последовательной связи будьте готовы выждать несколько секунд, пока контрольная информация будет зарегистрирована между GX IEC Developer и ПЛК.

Скорость коммуникационного опроса из GX IEC Developer в ПЛК можно увеличить, настроив следующие параметры из меню **Extras** → **Options** и выбрав **Monitor Mode**; измените настройку скорости опроса (**poll rate**):



4.4.8 Настройка видимости монитора

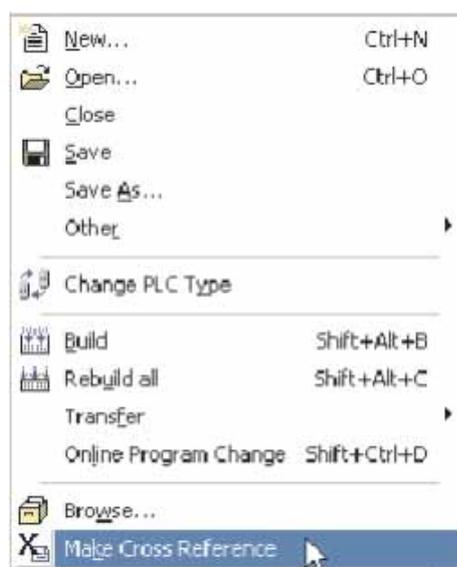
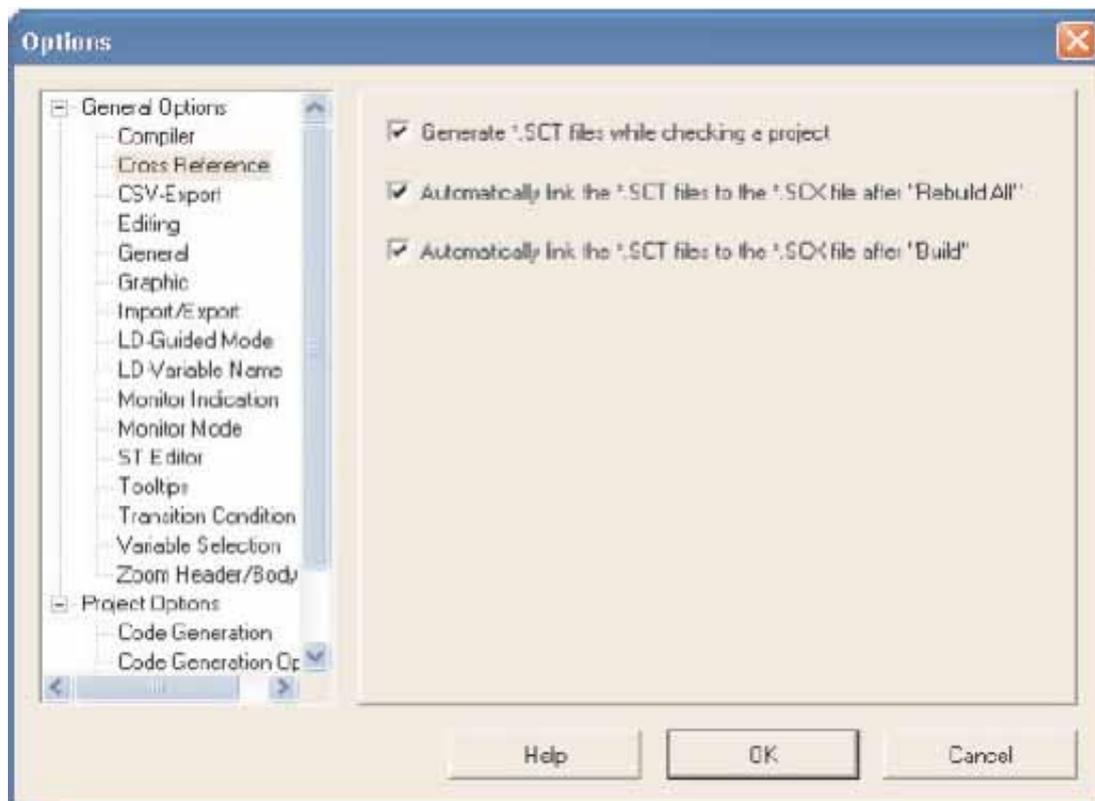
Чтобы регулировать видимость режима мониторинга, выберите "**Extras** → **Options** → **Monitor Indication**", где можно разрешить мигающее сообщение, которое будет показано в выбранном месте. Пользователь может установить наилучшую частоту мигания надписи "Monitoring":



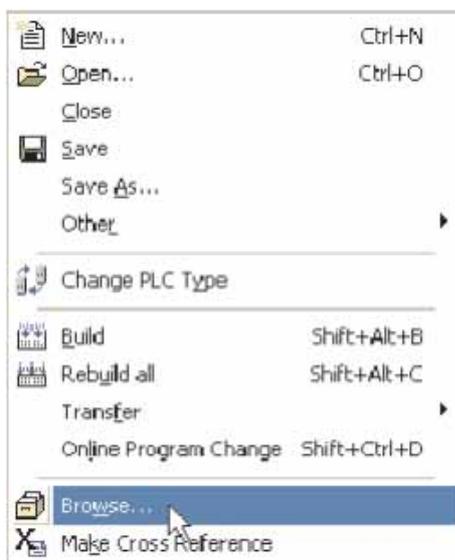
4.5 Список перекрестных ссылок

Чтобы генерировать Список перекрестных ссылок:

- ① Откройте меню **Extras / Options** и выберите **Cross Reference**.
- ② Выберите все показанные опции и перекомпилируйте проект.

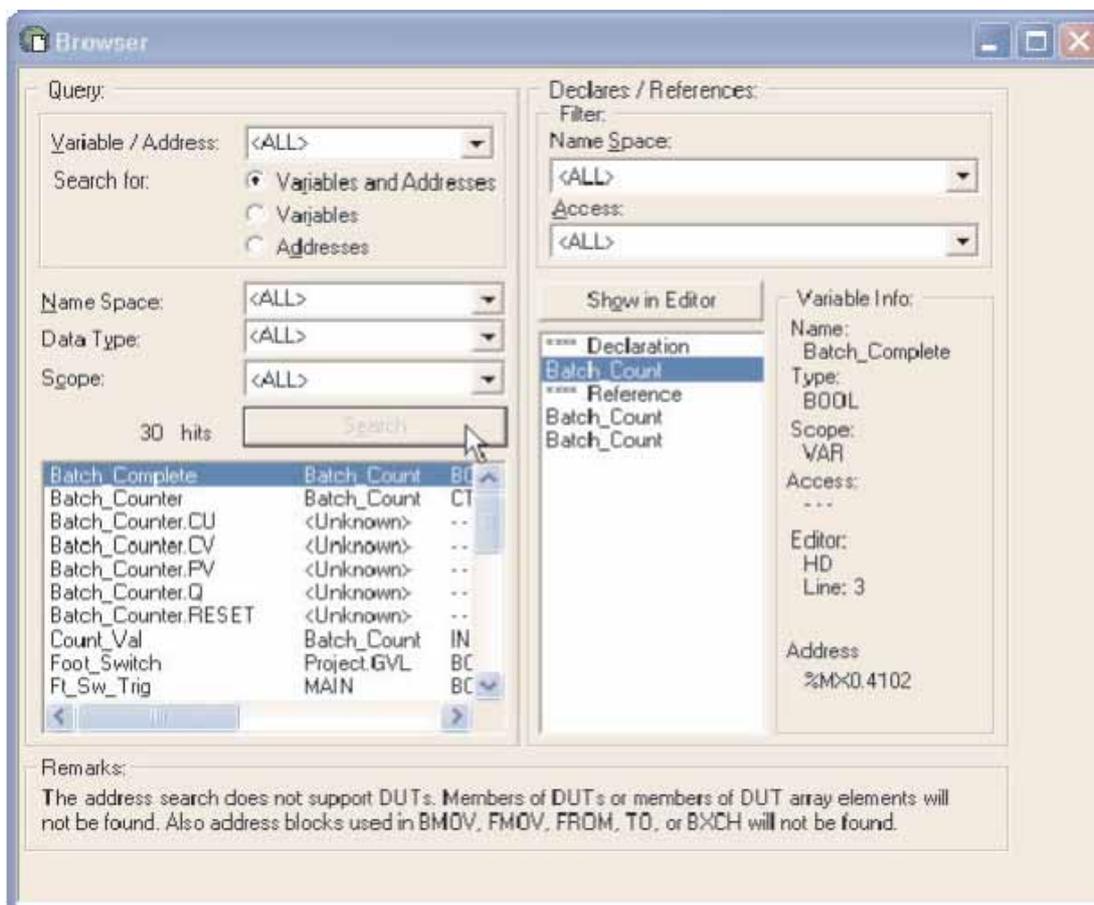


- ③ Затем выберите **Make Cross Reference** из меню **Project**, и генерируется список.



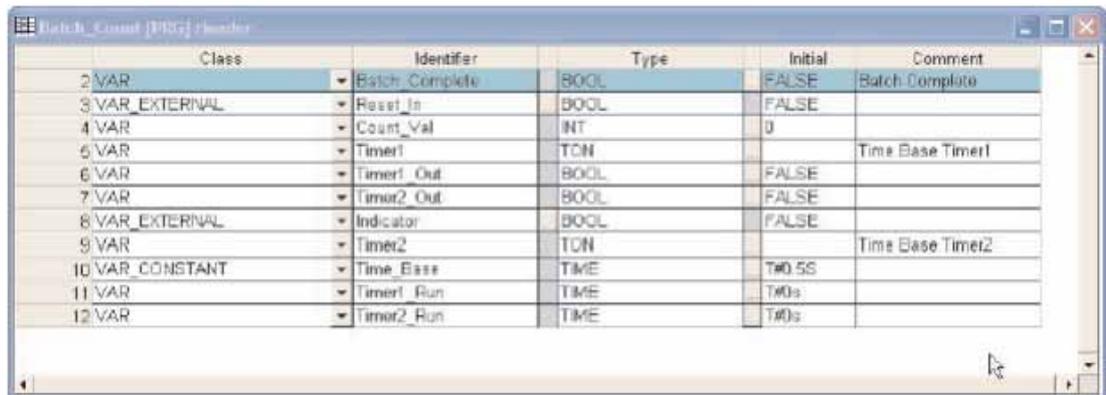
④ Откройте проводник из меню **Project**, или используя значок  на панели инструментов.

⑤ Щелкните на кнопке **Search**, и будет показан полный список.



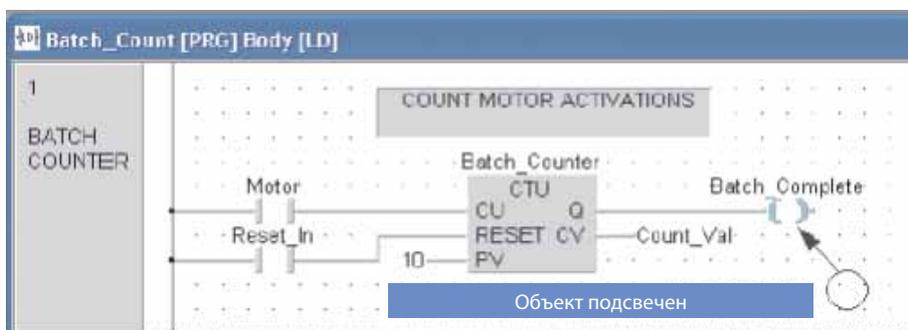
Для поиска конкретных переменных и т.д. можно использовать окна выбора запросов. Затем отдельные детали подсвеченного элемента показываются в правой стороне окна.

Кнопка **Show in Editor** открывает заголовок подсвеченного элемента из списка в правой стороне, например:



	Class	Identifier	Type	Initial	Comment
2	VAR	Batch_Complete	BOOL	FALSE	Batch Complete
3	VAR_EXTERNAL	Reset_In	BOOL	FALSE	
4	VAR	Count_Val	INT	0	
5	VAR	Timer1	TON		Time Base Timer1
6	VAR	Timer1_Out	BOOL	FALSE	
7	VAR	Timer2_Out	BOOL	FALSE	
8	VAR_EXTERNAL	Indicator	BOOL	FALSE	
9	VAR	Timer2	TON		Time Base Timer2
10	VAR_CONSTANT	Time_Base	TIME	T#0.5S	
11	VAR	Timer1_Run	TIME	T#0s	
12	VAR	Timer2_Run	TIME	T#0s	

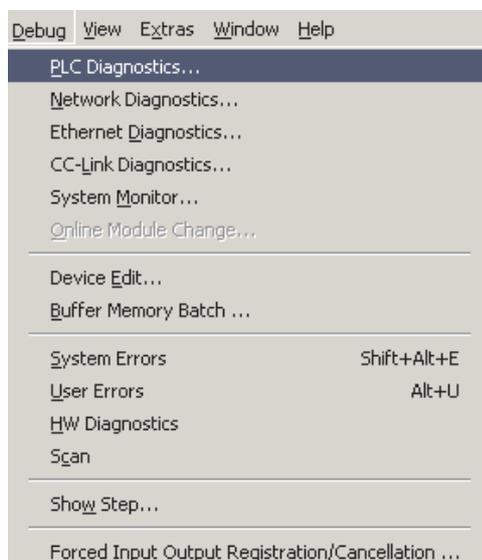
Или



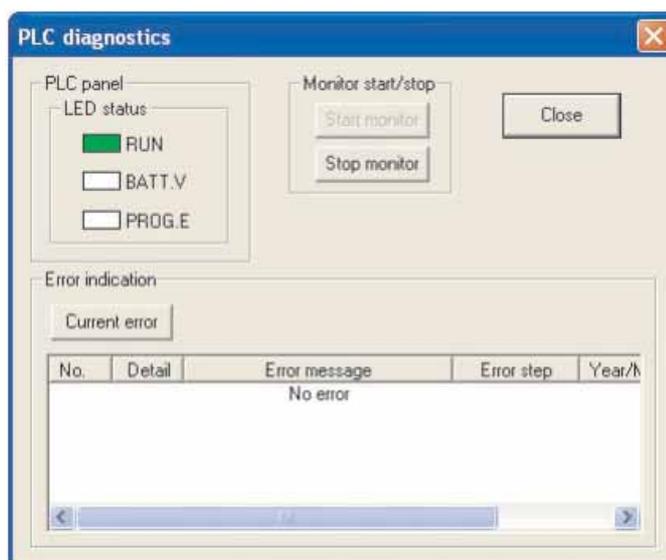
Список перекрестных ссылок можно распечатать, используя возможность печати в GX IE Developer.

4.6 Диагностика ПЛК

В GX IEC Developer имеются различные диагностические функции. Функции в меню **Debug** позволяют выполнять точную отладку и анализ ошибок в вашем приложении.



Щелкните на **PLC Diagnostics**, чтобы открыть показанное ниже окно.



Сообщение об ошибке открытым текстом

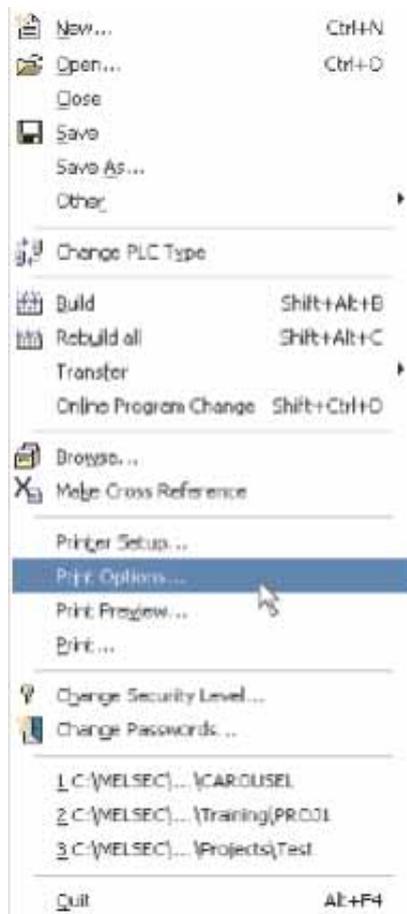
Оцениваются регистры данных об ошибках в ПЛК с выдачей открытого текста и соответствующих тексты контекстной справки.

Наиболее важные аппаратные ошибки типа "Перегорел предохранитель" показываются в окне и оцениваются.

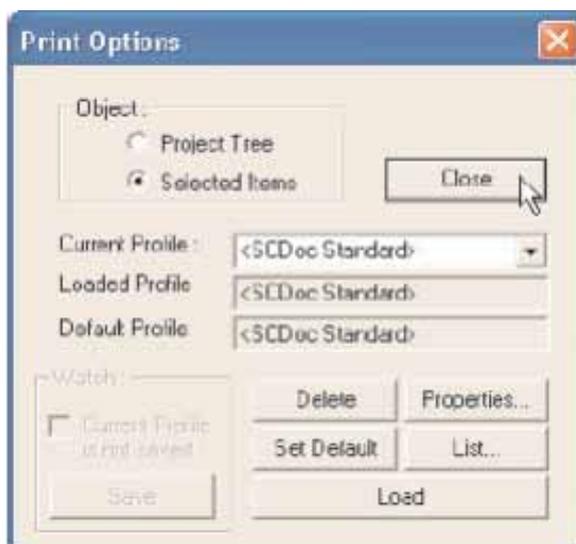
Можно определять пользовательские ошибки. Эти пользовательские ошибки хранятся с самостоятельно созданным текстовым файлом (USER_ERR.TXT) и позволяют быстро исправлять ошибки. Последние восемь пользовательских ошибок хранятся в FIFO регистре; они удаляются, только когда больше не возникают.

4.7 Документация проекта

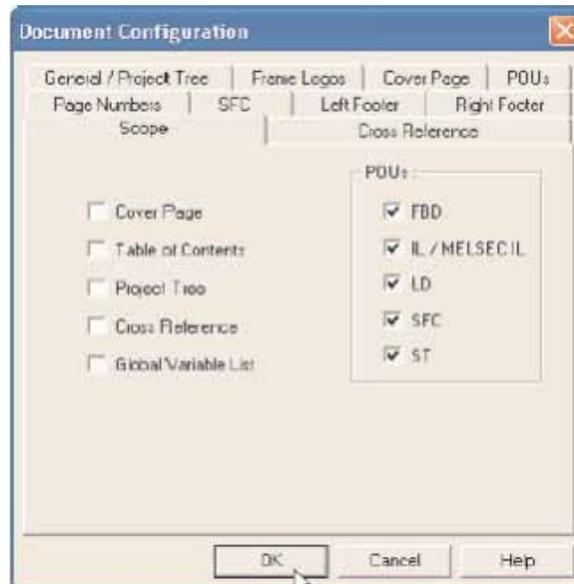
Документацию проекта можно составить, используя пункт **Print Option** из меню **Project**:



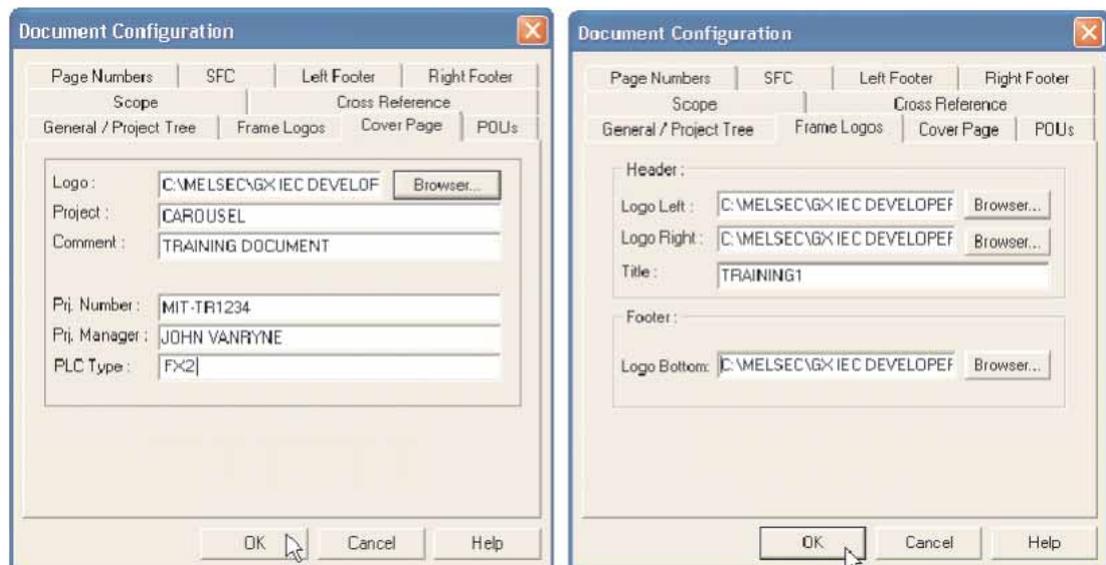
Появится диалоговое окно изменения конфигурации "**Change Configuration**". Здесь можно восстановить предыдущие профили проекта или работать с профилем по умолчанию. Выберите **Project Tree** для всех элементов, или **Selected Items** для конкретных подсвеченных пунктов, и откройте **Properties**:



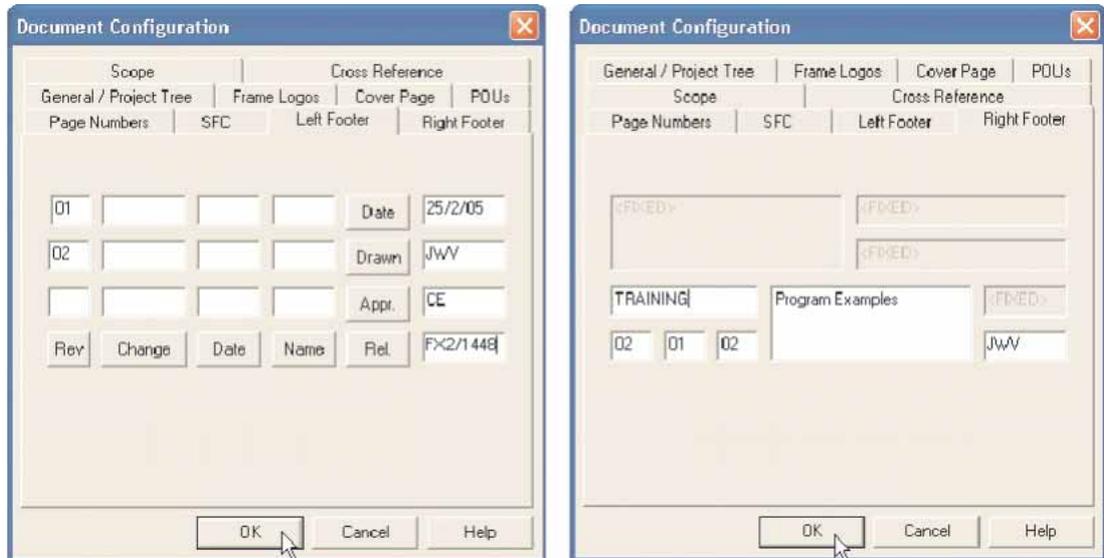
Ниже показано окно **Document Configuration**. Выберите вкладки, чтобы конфигурировать документ согласно требованиям. В этом примере будет напечатан только COUNTER_FB_CE, согласно выбранной опции **Selected Items**:



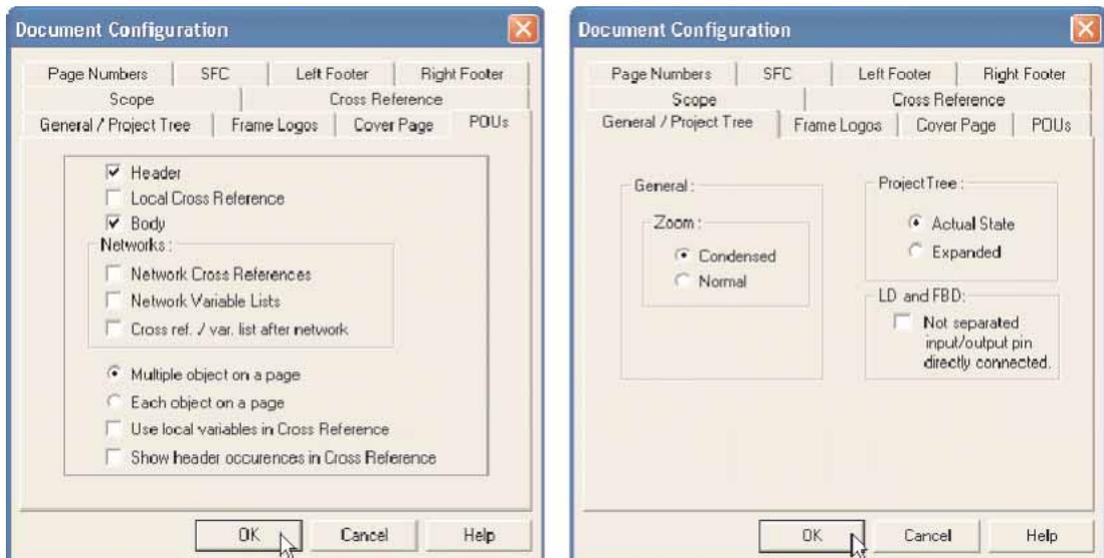
На вкладке **Cover Page** определенные пользователем логотипы и информация могут быть заданы для титульного листа и для кадра на вкладке **Frame Logos**:



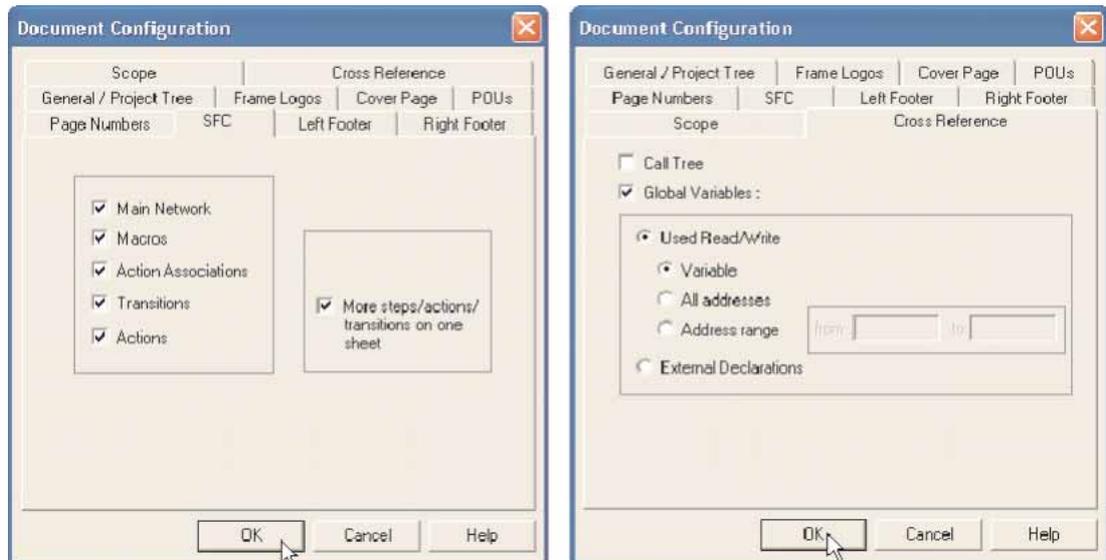
Подробную информация можно присвоить левым и правым нижним сноскам. При необходимости можно переименовать содержимое полей в диалоговом окне левого нижнего колонтитула **Left Footer**, щелкнув на кнопках имен:



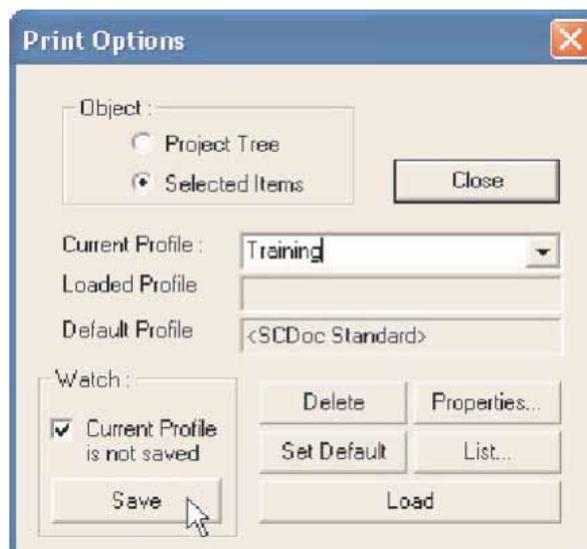
Спецификация вида программного модуля и общие спецификации проекта имеются на вкладках **POUs** и **General/Project Tree**.



Спецификация вида **SFC** и спецификации перекрестных ссылок имеются на вкладках **SFC** и **Cross Reference**:



Конфигурированный профиль можно сохранить, просто задав название в поле "Current Profile" и затем щелкнув на кнопке **Save**. Затем его можно повторно вызвать в любое время, используя окно выбора:



5 Пример программы

5.1 QUIZMASTER - ВЕДУЩИЙ ТЕЛЕВИКТОРИНЫ

Рассмотренные предметы:

- Синхронизация
- Счет
- Логические операции: Фиксация - Блокировки - Использование внутреннего операнда М.
- Функциональные инструкции: Функция сброса - Функция импульса

Описание

Всесторонний автоматический контроллер игровой викторины; определяет и фиксирует первого игрока, чтобы включить соответствующую "Кнопку реакции на ответ". Будет активизирована лампа реакции только одного соперника; все последующие ответы других соперников выключаются.

Задача

- Создать релейную диаграмму ПЛК, которая гарантирует, что включается индикаторная лампа только одного из соперников.
- Когда ведущий нажимает кнопку Start, соперники имеют 10-секундное окно, чтобы предложить ответ с помощью своих кнопок ответа.
- В ходе периода ожидания ответа прошедшее время (0 -10 с) показывается на аналоговом индикаторе учебного стенда.
- Ведущий может в любое время сбросить систему, используя отдельную кнопку.

Список ввода/вывода

– Входы

X0	-	Кнопка ответа игрока 1
X1	-	Кнопка ответа игрока 2
X2	-	Кнопка ответа игрока 3
X3	-	Кнопка ответа игрока 4
X4	-	Запуск отсчета ведущим
X5	-	Сброс игры

– Выходы

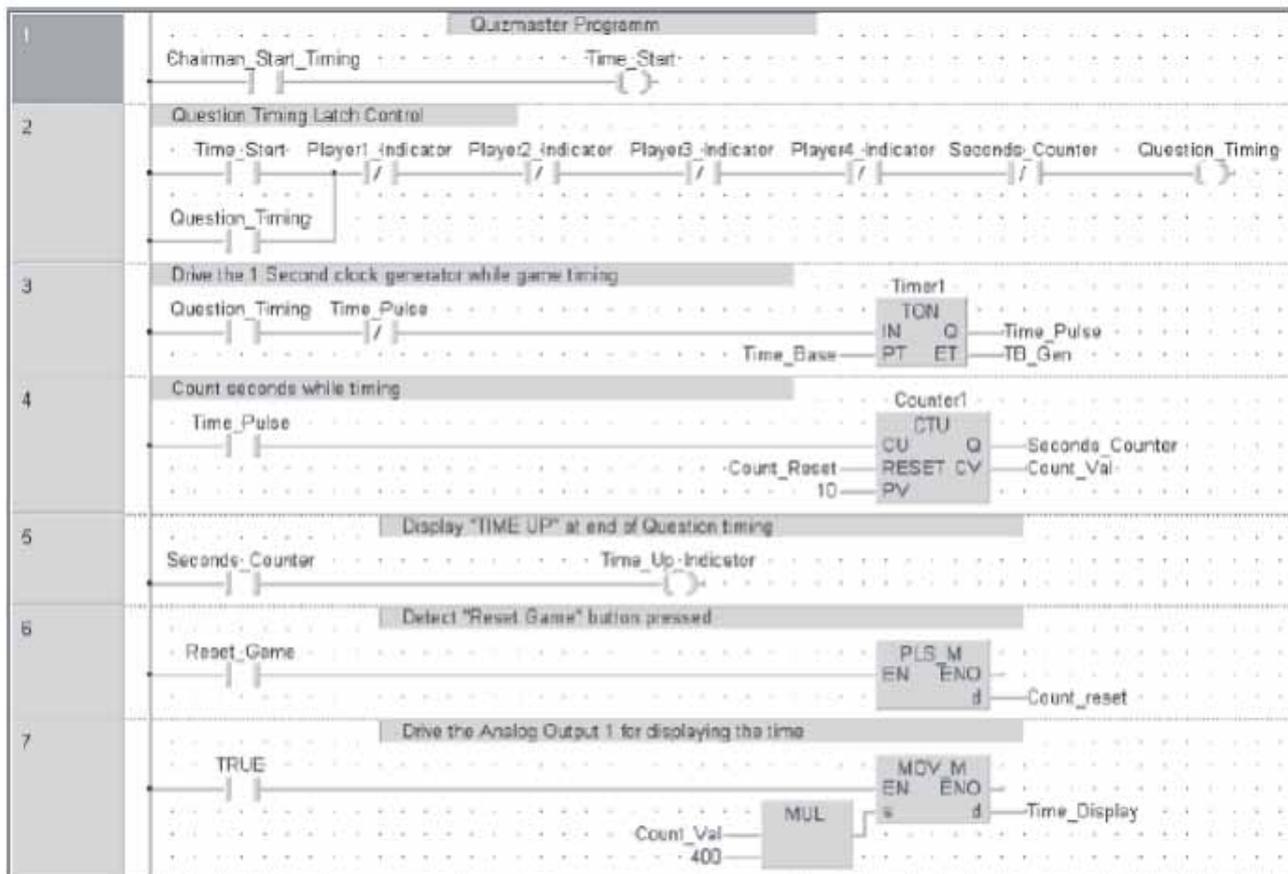
Y0	-	Лампа ответа игрока 1
Y1	-	Лампа ответа игрока 2
Y2	-	Лампа ответа игрока 3
Y3	-	Лампа ответа игрока 4
Y4	-	Индикация "Время истекло"
Y5	-	Отсчет времени вопроса

5.1.1 Метод

- ① Создайте новый проект и назовите его "Quizmaster".
- ② Введите следующие данные в список глобальных переменных **Global Variables List**:

	Class	Identifier	MT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	Player1_Response	X00	%X0	BOOL	FALSE
1	VAR_GLOBAL	Player2_Response	X01	%X1	BOOL	FALSE
2	VAR_GLOBAL	Player3_Response	X02	%X2	BOOL	FALSE
3	VAR_GLOBAL	Player4_Response	X03	%X3	BOOL	FALSE
4	VAR_GLOBAL	Chairman_Start_Timing	X04	%X4	BOOL	FALSE
5	VAR_GLOBAL	Reset_Game	X05	%X5	BOOL	FALSE
6	VAR_GLOBAL	Player1_Indicator	Y00	%Y00	BOOL	FALSE
7	VAR_GLOBAL	Player2_Indicator	Y01	%Y01	BOOL	FALSE
8	VAR_GLOBAL	Player3_Indicator	Y02	%Y02	BOOL	FALSE
9	VAR_GLOBAL	Player4_Indicator	Y03	%Y03	BOOL </td <td>FALSE</td>	FALSE
10	VAR_GLOBAL	Question_Timing	Y04	%Y04	BOOL	FALSE
11	VAR_GLOBAL	Time_Display	U4G1	%MW14.4.1	INT	0
12	VAR_GLOBAL	Time_Up_Indicator			BOOL	FALSE

- ③ Создайте новый программный модуль класса **PRG** (Program type) на языке **Ladder Diagram** и назовите его "Game_Control".
- ④ Введите следующий код в программный модуль.

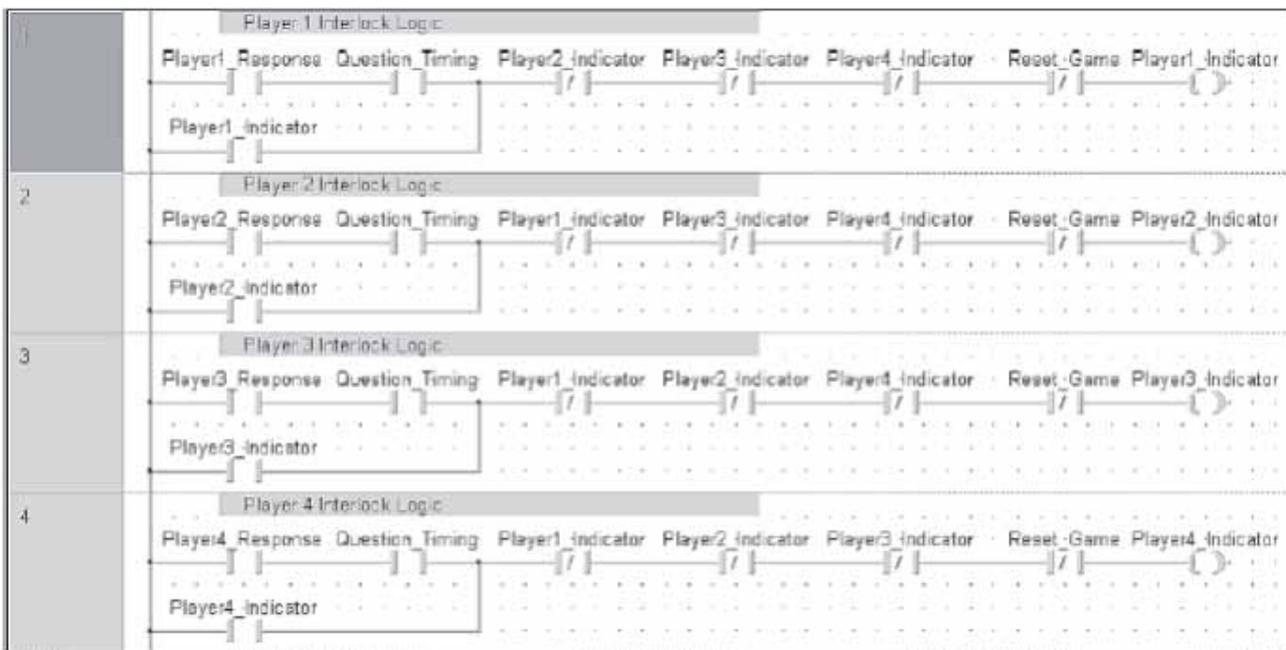


Окончательный заголовок программного модуля "Game_Control" должен иметь следующий вид:

	Class	Identifier	Type	Initial	Comment
0	VAR	Time_Start	BOOL	FALSE	
1	VAR	Time_Pulse	BOOL	FALSE	
2	VAR	TB_Gen	TIME	T#0s	
3	VAR_CONSTANT	Time_Base	TIME	T#1s	
4	VAR	Count_Reset	BOOL	FALSE	
5	VAR	Seconds_Counter	BOOL	FALSE	
6	VAR	Count_Val	INT	0	
7	VAR	Timer1	TON		
8	VAR	Counter1	CTU		
9	VAR	Config_Analog	BOOL	FALSE	

⑤ Создайте новый программный модуль класса **PRG** и типа **Ladder** и назовите его "Player_Logic"

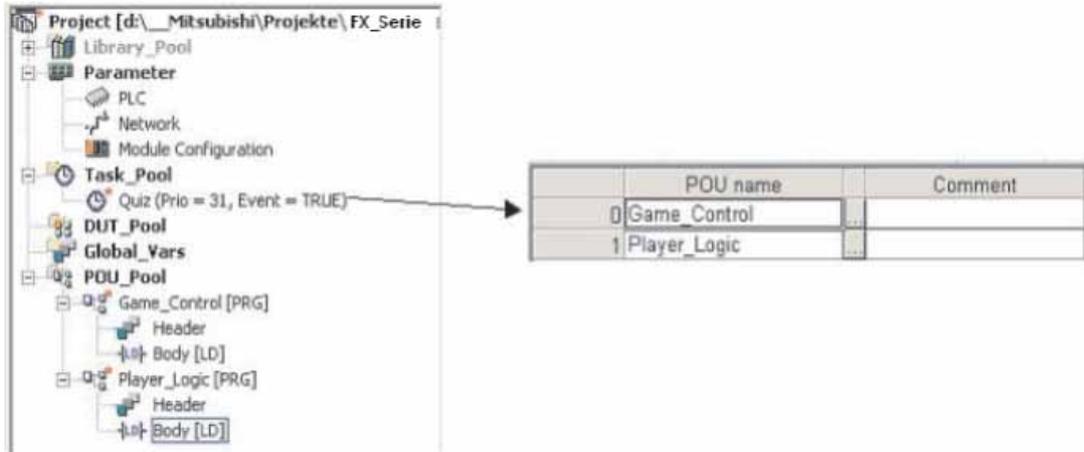
⑥ Введите следующий код релейной диаграммы в новый программный модуль:



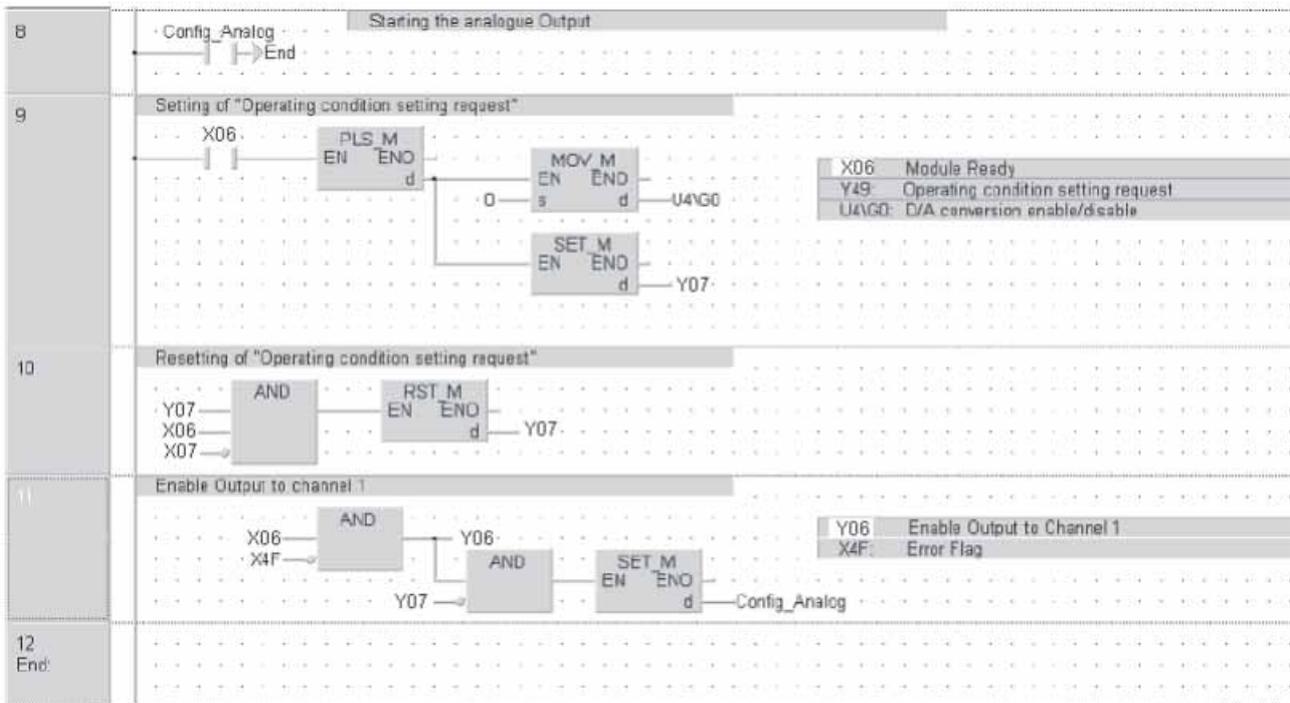
Окончательный заголовок программного модуля "Player_Logic" должен иметь следующий вид:

	Class	Identifier	Type	Initial	Comment
0	VAR			...	

- ⑦ Создайте новую задачу в задачном пуле "QUIZ". Свяжите ПМ "Player_Logic" и "Game_Control" соответственно с новой задачей, как показано ниже:



- ⑧ Добавьте следующие звенья в программный модуль "Game_Control", чтобы включить аналоговый выход в канале 1, который подключен к индикатору.



5.1.2 Quizmaster - Принцип работы

- ① Введите, проверьте и сохраните проект "Quizmaster", включая аннотацию.
- ② Загрузите проект в ПЛК серии FX.
- ③ Убедитесь, что проект работает правильно, контролируя работу при активизации входов.
- ④ На мгновение включите вход X4, чтобы начать хронометраж реакции на ответ соперника.
- ⑤ Ждите начальной реакции соперников из X0, X1, X2 или X3 и блокируйте индикатор соответствующего соперника. Блокируйте дальнейшую работу всех входов.
- ⑥ Ожидая ответа, включите таймер ответа на период 10 с и выведите текущее время на дисплей.
- ⑦ В конце временного периода блокируйте дальнейшую работу входов ответов всех соперников, остановите индикацию времени и включите индикатор истечения времени "Time Up".
- ⑧ Ждите, пока ведущий активизирует вход сброса "Reset" X5, чтобы очистить все флаги состояния и выходы и начать новый раунд.
- ⑨ Возвратитесь к шагу 1 или закончите игру.

5.1.3 Описание программы Quizmaster

Программный модуль "Game_Control"

- Звено 1
Когда ведущий нажимает кнопку начала отсчета Start Timing , локальная переменная "Time_Start" пульсирует благодаря инструкции PLS_M.
- Звено 2
Переменная Question_Timing фиксирована, обеспечивая, что никакие индикаторы игроков не включены, и счетчик секунд не считает.
- Звено 3
Контакт Question_Timing позволяет включить игровой таймер с 1-секундной шкалой времени. 1-секундные импульсы генерируются на выходе "Time_Pulse".
- Звено 4
Импульсы от флага Time_Pulse подсчитываются с использованием счетчика CTU "Count UP", который считает в течение 10-секундного периода.
- Звено 5
Когда работает флаг Seconds_Counter, активизируется Time_Up_Indicator и включает лампу.
- Звено 6
Когда активизируется вход "Reset_Game", генерируется импульс, чтобы сбросить счетчик секунд в звене 7 ниже.

- Звено 7

Вход TRUE "всегда включен", поэтому значение Count_Val, умноженное на смещение 400 бит/В, постоянно посылается для индикации времени "Time_Display" на аналоговый выходной модуль.

Программный модуль "Player_Logic"

- Звенья 1- 4

Эти процедуры управляют блокировкой игров. Например, если игрок 1 первым нажал на свою кнопку ответа, то его лампа включается и блокирует все последующие ответы от других игроков.

Процедура управляющей логики каждого игрока блокирует последующие ответы других игроков.

Игроки могут предлагать ответы, только когда флаг "Question_Timing" активен.

6 Функции и функциональные блоки

Ниже приведена таблица, иллюстрирующая отличия между "Функциями" и "Функциональными блоками":

Элемент	Функциональный блок	Функция
Хранение внутренней переменной	Хранение	Нет хранения
Создание копии функционального блока	Требуется	Не требуется
Выходы	Нет выхода	Один выход
Повторное выполнение с теми же входными значениями	Не всегда выдает то же значение выхода	Всегда выдает то же значение выхода

- Функции являются частью системы команд.
- Функции включены в стандартные библиотеки и библиотеки производителей. Например TIMER_M является функцией, как и MOV_M, PLUS_M и т.д. из системы команд Mitsubishi в библиотеке производителя.
- Пользовательские функции можно легко создавать из проверенных частей программы.
 Это означает, что функции могут создаваться, например, для системных вычислений/вычислений процесса; они могут храниться в библиотеках и многократно использоваться с различными объявлениями переменных. Они могут служить таким же образом, как, например, команда MOV, но с преимуществом учета специфики конкретного пользователя.

6.1 Функции

Большинство управляющих программ включает некоторые виды математических вычислений, например, для преобразования аналоговых сигналов, индикации технических единиц и т.д. Часто они многократно используются в структуре программы.

Используя пользовательские функции, можно существенно уменьшить время разработки программы.

6.1.1 Пример: Создание функции

Цель:

Создать функцию для перехода от градусов Фаренгейта к градусам Цельсия.

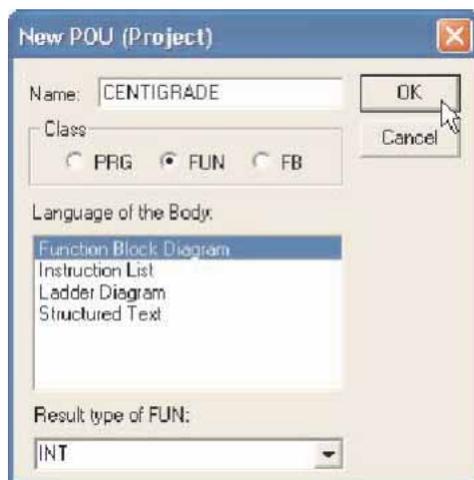
Используется формула:

$$\text{Centigrade} = \frac{(\text{Fahrenheit} - 32) \times 5}{9}$$

Функция будет называться "CENTIGRADE", и входная переменная будет называться "Fahrenheit".

Процедура

- 1 Выберите новый программный модуль и назовите его "CENTIGRADE".

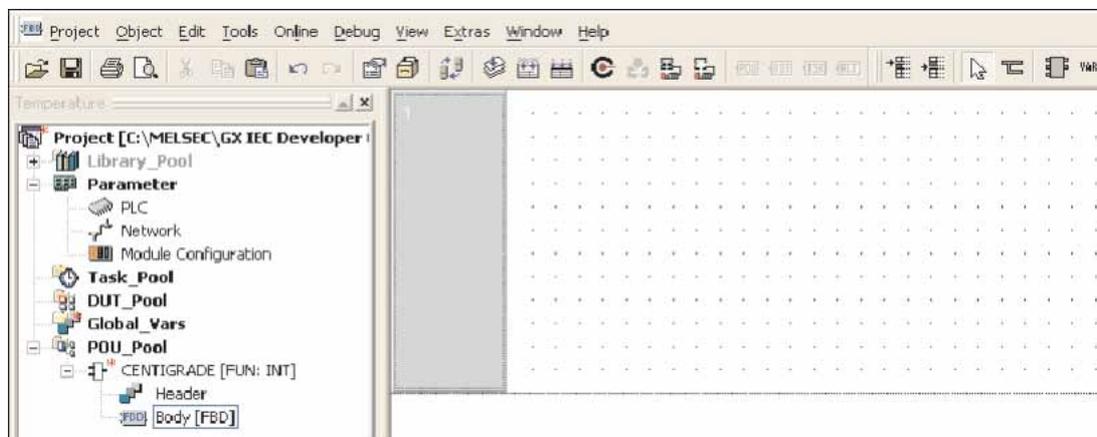


На этот раз щелкните на опции **FUN**, а не PRG. Выберите в качестве редактора **Function Block Diagram**. Тип результата функции **Result Type of FUN** следует оставить как **INT** (целочисленный тип).

Теперь "CENTIGRADE" появился на дереве программных модулей:

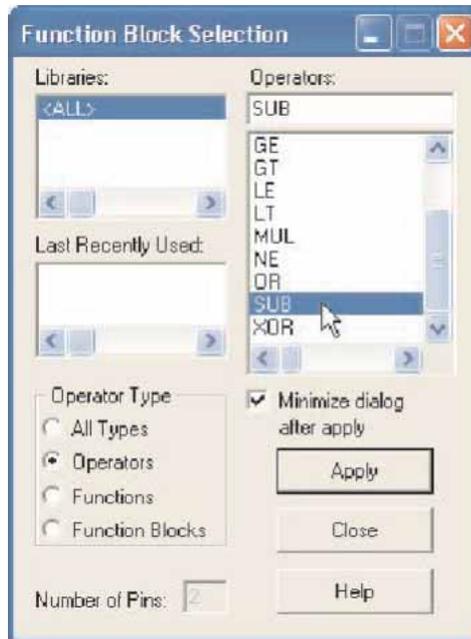


- 2 Дважды щелкните на значке тела FBD, чтобы открыть звено тела:

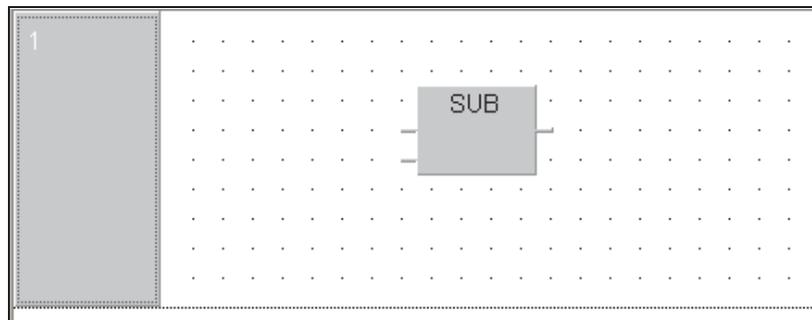


Выбор функции:

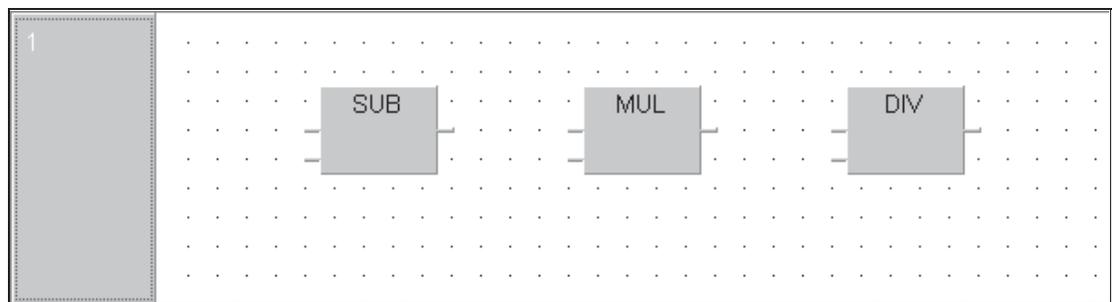
- 1 Выберите значок функционального блока  из панели инструментов и **SUB** из списка операторов:



- 2 Используя **Apply** или дважды щелкнув на выбранном объекте, поместите его на экран:



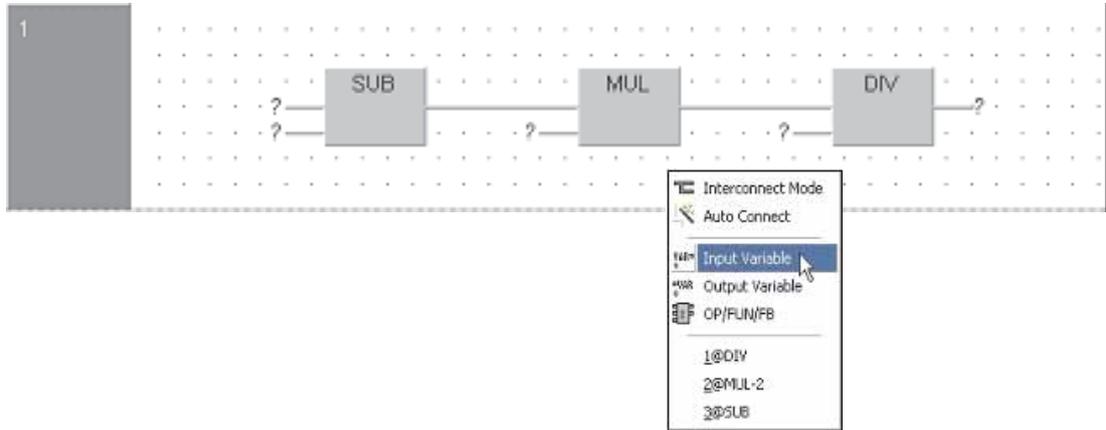
- 3 Повторите описанный выше процесс, чтобы экран принял следующий вид:



Объявление переменных

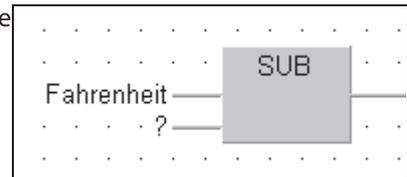
Можно использовать различные методы объявления переменных. Следующая процедура иллюстрирует, как объявлять переменные из тела FBD:

- 1 Поместите входные и выходные переменные, щелкнув правой кнопкой мыши в рабочей области. Из следующего всплывающего меню выберите и поместите теги входных и выходных переменных на FBD, как показано ниже:

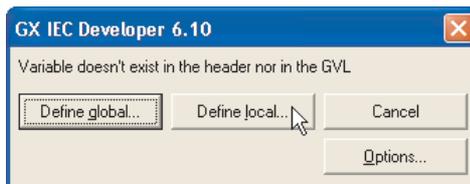


Альтернативно, щелкните на кнопке панели

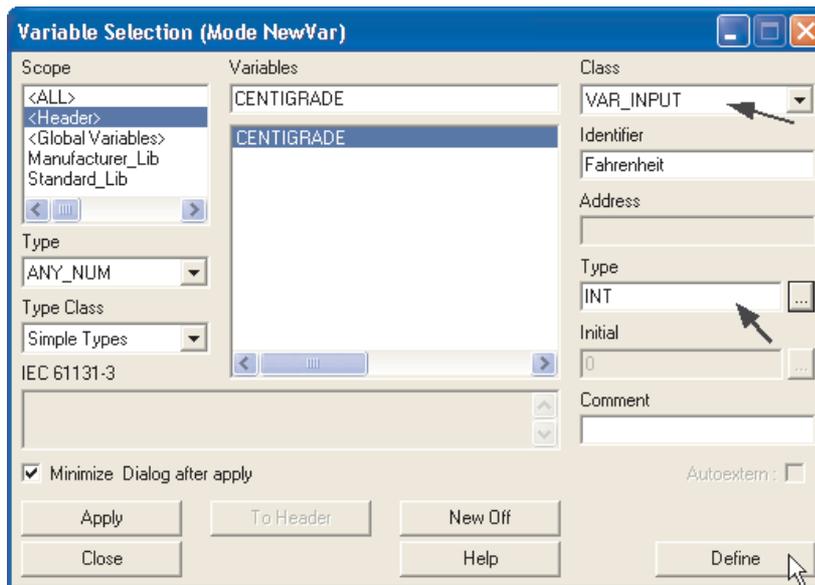
- 2 Объявите переменную "Fahrenheit", просто напечатав ее в область переменной:



Поскольку это имя переменной еще не было определено в заголовке (LVL), будет представлено диалоговое окно запроса, чтобы выбрать глобальную или локальную переменную, щелкните на **Define Local**.



- 3 Заполните свойства переменной следующим образом: **Class:** VAR_INPUT, **Type:** INT, как показано ниже:



ПРИМЕЧАНИЯ

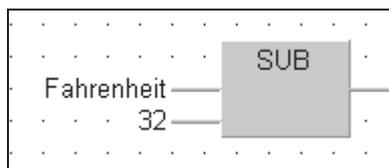
Требуется класс VAR_INPUT, поскольку эта переменная позволяет вводить значения в функцию, когда она подключена как часть программы. Это показывается точкой соединения со входом слева на символе функции.

Обратите внимание, что переменная "CENTIGRADE" также автоматически внесена в список. Причина заключается в том, что "имя выходной переменной" должно быть таким же, как "имя функции".

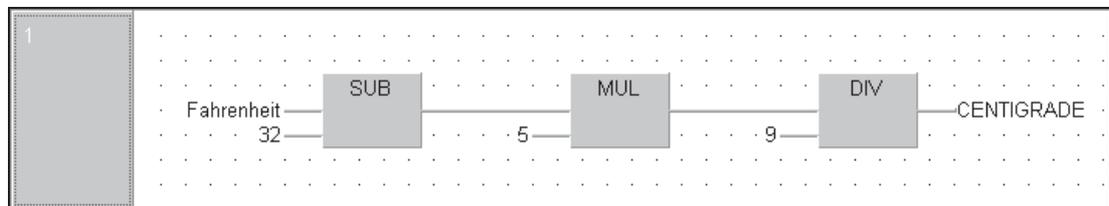
- ④ Щелкните на **Define**, и переменная будет записана в заголовок функции "CENTIGRADE". Вы можете проверить это, открыв заголовок.

Объявление констант

- ① Объявите константу "32", просто напечатав число в окне переменной:



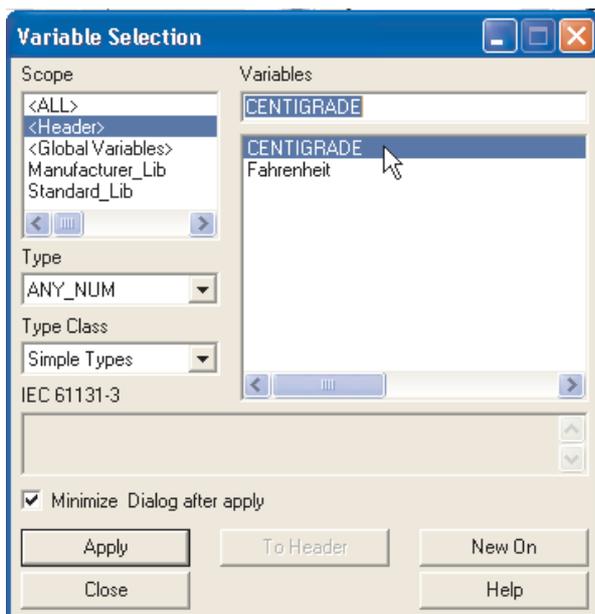
- ② Полная диаграмма функции "CENTIGRADE" имеет следующий вид:



Полезный совет: При вводе переменной "CENTIGRADE" не требуется печатать ее имя - просто щелкните правой кнопкой на окне переменной (или нажмите F2).

В окне **Variable Selection** дважды щелкните на "CENTIGRADE" или щелкните, чтобы выбрать, и нажмите **Apply**.





"CENTIGRADE" автоматически помещается в список переменных заголовка, так как является именем функции и поэтому должна также быть задана как выходной аргумент.

При желании проверьте правильность заголовка функции "CENTIGRADE"; он должен иметь следующий вид:

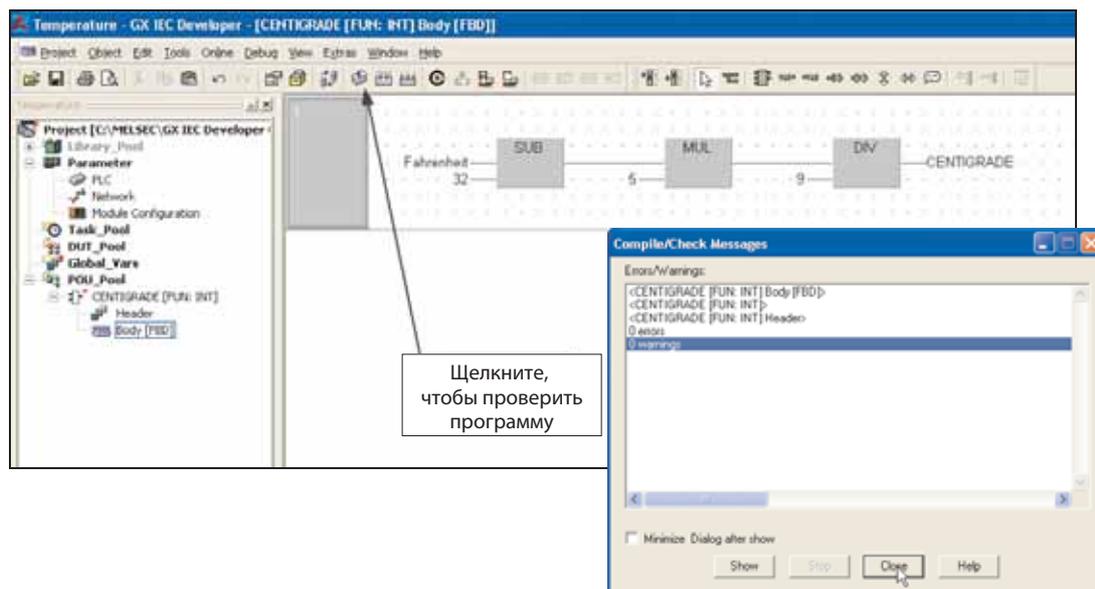
Class	Identifier	Type	Initial	Comment
VAR_INPUT	Fahrenheit	INT	0	

ПРИМЕЧАНИЕ

Альтернативно, переменную "Fahrenheit" можно ввести непосредственно в заголовок (как показано выше) и выбрать (клавишей F2 или щелкнув правой кнопкой на окне переменной) в точке ввода в теле.

Проверка целостности звена

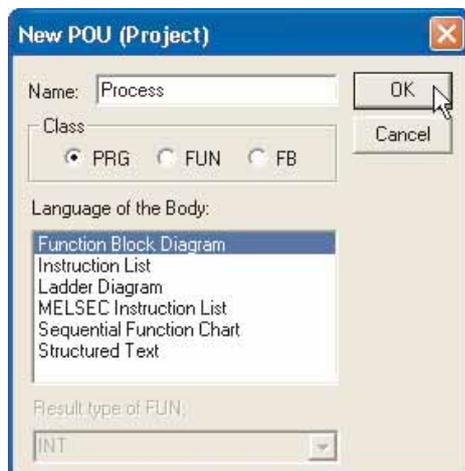
- ① Проверьте звено; не должно быть никаких ошибок и никаких предупреждений!



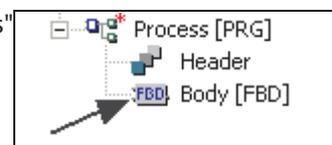
- ② Закройте все рабочие окна и любые диалоговые окна, которые могут быть открытыми.

Создание программного модуля новой программы

- 1 Создайте новый программный модуль, называемый "Process", класса **PRG** на языке функциональных блок-схем **Function Block Diagram (FBD)**:

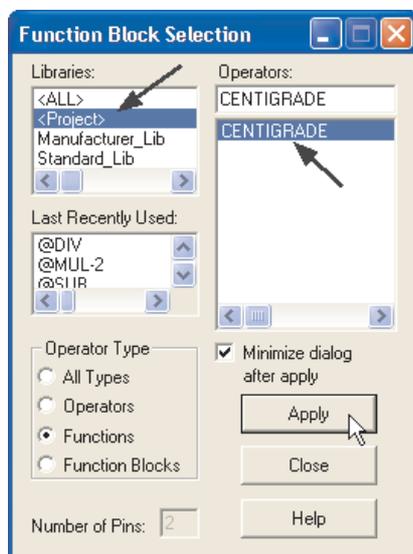


- 2 Откройте (двойным щелчком) тело релейного ПМ "Process" в пуле программных модулей проекта.



Размещение функции пользователя

- 1 Снова щелкните на значке функционального блока , но в этот раз выберите **Functions** и выберите **Project Library**. Отметим, что только что созданная функция "Centigrade" теперь отфильтрована в списке операторов:

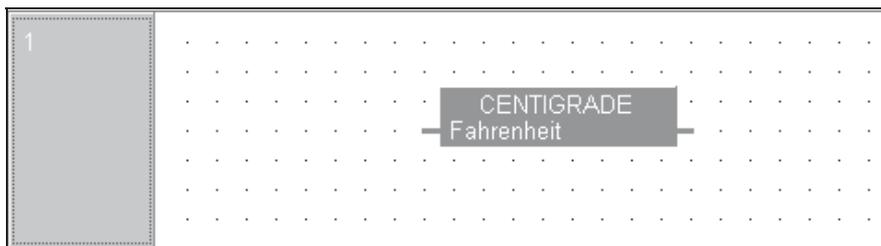


- 2 Выберите CENTIGRADE и щелкните на **Apply**.

ПРИМЕЧАНИЕ

При необходимости можно минимизировать окно выбора функционального блока **Function Block selection** после применения **Apply**, выбрав показанную выше опцию **Minimize dialog after apply**.

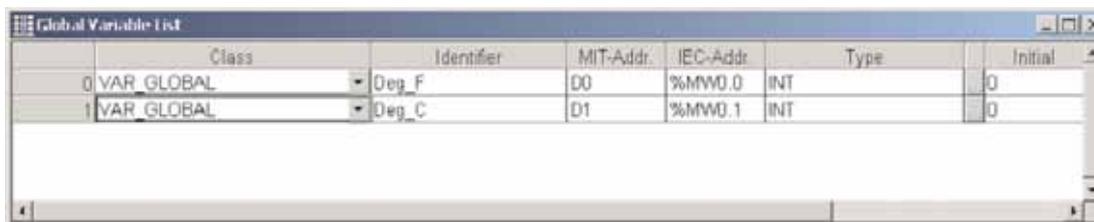
Экран примет следующий вид:



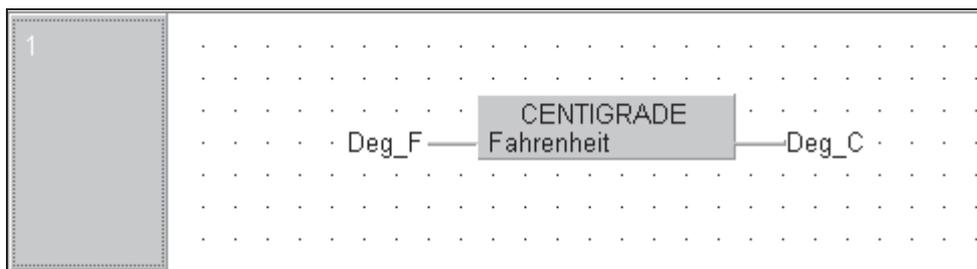
Назначение глобальных переменных

Когда функция помещена на новое звено, присвойте ей переменные.

- 1 Присвойте имена переменных в списке глобальных переменных, как показано ниже:



Тело программного модуля "Process" должно иметь следующий вид:



- 2 Создайте новую задачу в задачном пуле Task Pool с именем "Main".

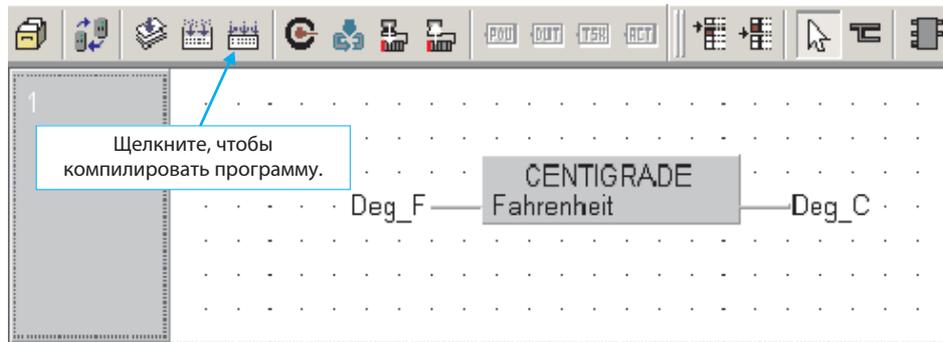


- 3 Свяжите программный модуль "Process" с задачей "Main":

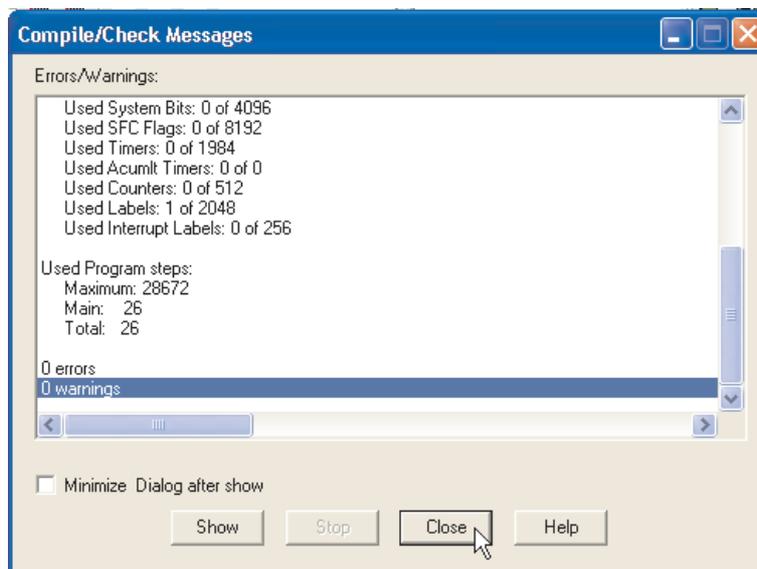
	POU name	Comment
0	Process	...

Компилирование программы

Компилируйте проект, используя операцию **Rebuild All** из панели инструментов:



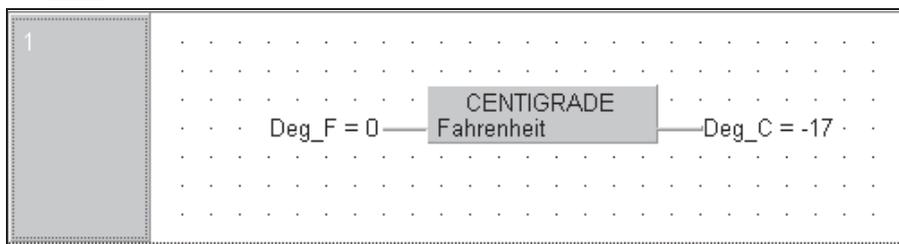
После компиляции должно быть показано следующее окно:



Если имеются ошибки, щелкните на деталях ошибки и устраните проблему(ы).

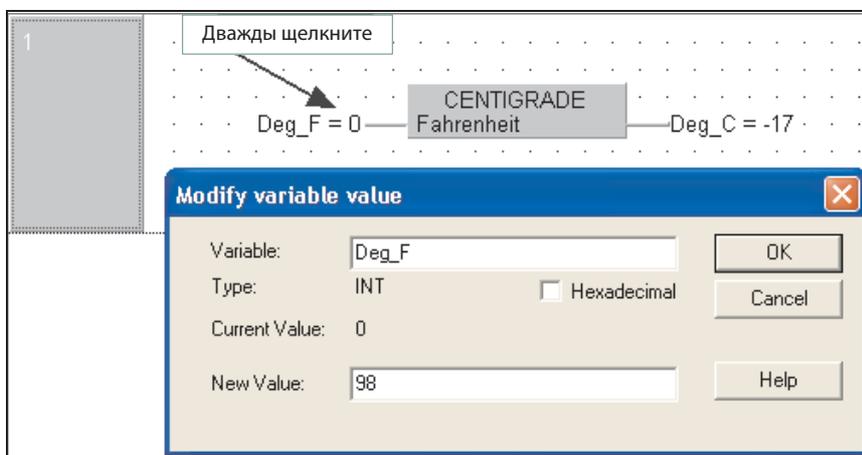
Мониторинг программы

- ① Передайте проект в ПЛК и проконтролируйте это звено, используя кнопку мониторинга  на панели инструментов:



- ② Используя возможность принудительной установки экранной переменной, введите числа в переменную "Deg_F" следующим образом:

Дважды щелкните на входной переменной и введите значение в диалоговое окно **Modify variable value**, как показано ниже:



Для ссылки, 100 градусов F = 37 градусов C (фактически 37.7 градусов C)

6.1.2 Обработка чисел формата REAL (с плавающей запятой)

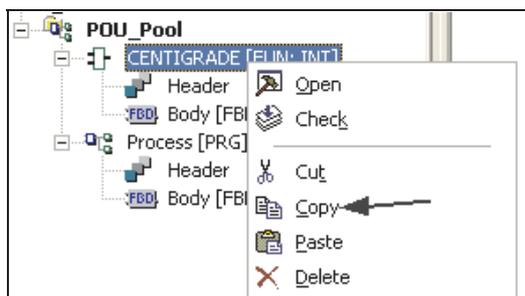
Существующая функция "CENTIGRADE" сейчас может обрабатывать только 16-битовые целочисленные значения (от +32767 до -32768); этот формат используется по умолчанию при создании функций. В следующем примере будет использована функция "CENTIGRADE", модифицированная для обработки значений формата "REAL" с плавающей запятой*.

* Допустимо только на процессорах, поддерживающих эту возможность.

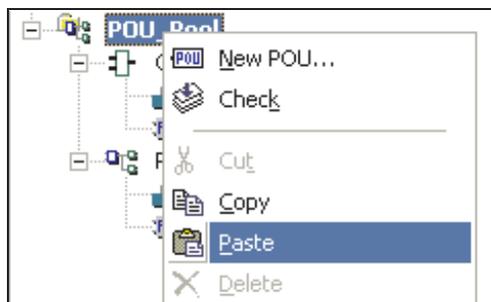
Дублирование функции

Сделайте дубликат функции "CENTIGRADE" и переименуйте его в "CENTIGRADE1" следующим образом:

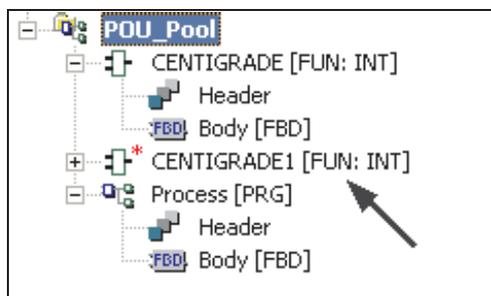
- Щелкните правой кнопкой на значке "CENTIGRADE" в пуле программных модулей проекта и выберите **Copy**.



- Щелкните правой кнопкой на значке пула программных модулей проекта и выберите **Paste**.

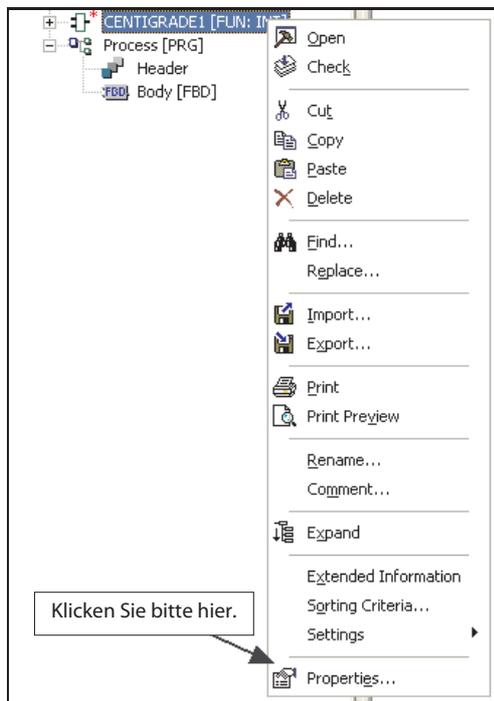


Система автоматически вставит дубликат "CENTIGRADE" и переименует его в "CENTIGRADE1":

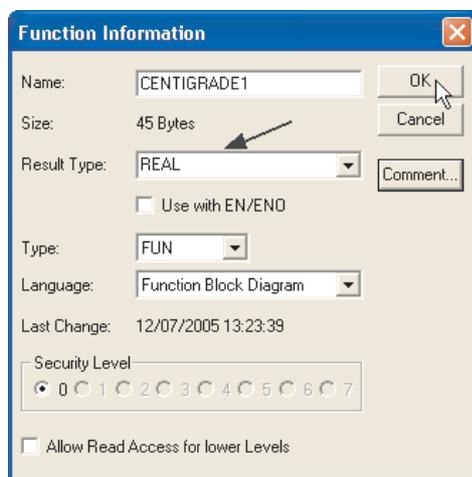


Изменение типа результата функции

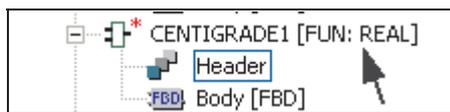
- Щелкните правой кнопкой на только что созданной функции "CENTIGRADE1" и щелкните на **Properties**.



- В открывшемся окне **Function Information** установите для типа результата "REAL".



Теперь в навигационном окне проекта тип должен быть показан как "REAL":

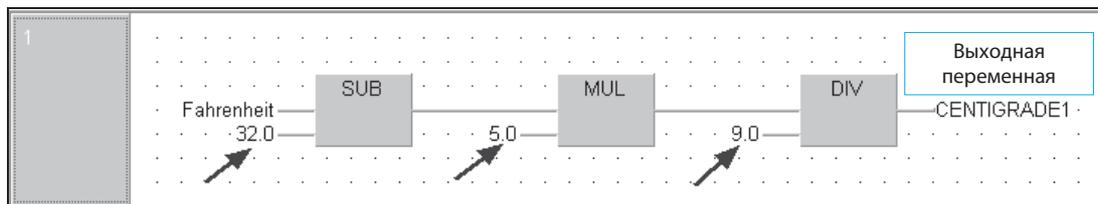


- Измените заголовок "CENTIGRADE1", чтобы переменная Fahrenheit имела тип "REAL":

	Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	Fahrenheit	REAL	0.0	

Изменение констант к типу "REAL"

- Откройте тело "CENTIGRADE1" и измените константы к типу "Floating Point" (например, 32.0), а также имя выходной переменной, чтобы экран принял следующий вид:



ПРИМЕЧАНИЕ: Не забудьте изменить CENTIGRADE на CENTIGRADE1.

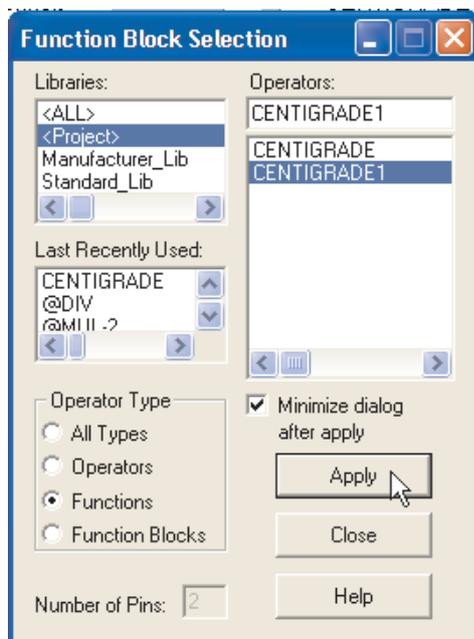
- Закройте редакторы и сохраните все изменения.

Помещение функции "CENTIGRADE" формата "REAL" в рабочий программный модуль "Process"

- В редакторе GVL создайте две новые переменные следующим образом:

	Class	Identifier	MIT-Addr	IEC-Addr	Type	Initial
0	VAR_GLOBAL	Deg_F	D0	%MWD.0	INT	0
1	VAR_GLOBAL	Deg_C	D1	%MWD.1	INT	0
2	VAR_GLOBAL	Deg_F Real	D2	%MDO.2	REAL	0.0
3	VAR_GLOBAL	Deg_C Real	D4	%MDO.4	REAL	0.0

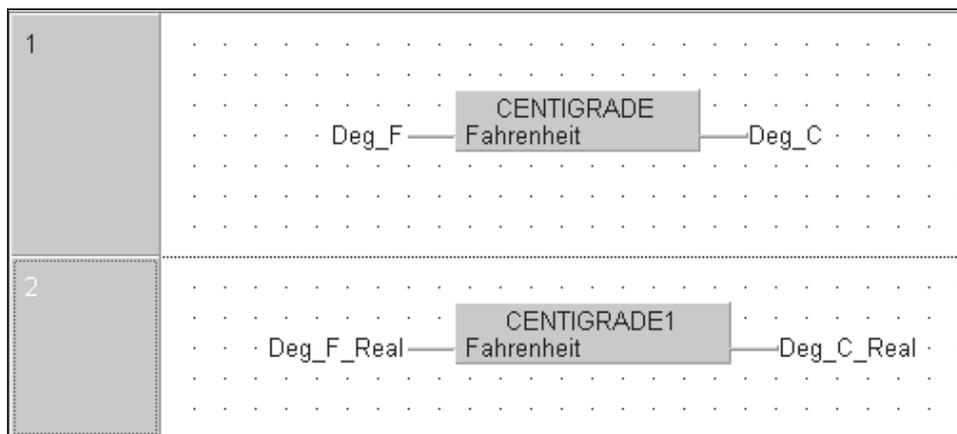
- Откройте тело программного модуля "Process" и поместите в него функцию "CENTIGRADE", как показано ниже:



ПРИМЕЧАНИЕ

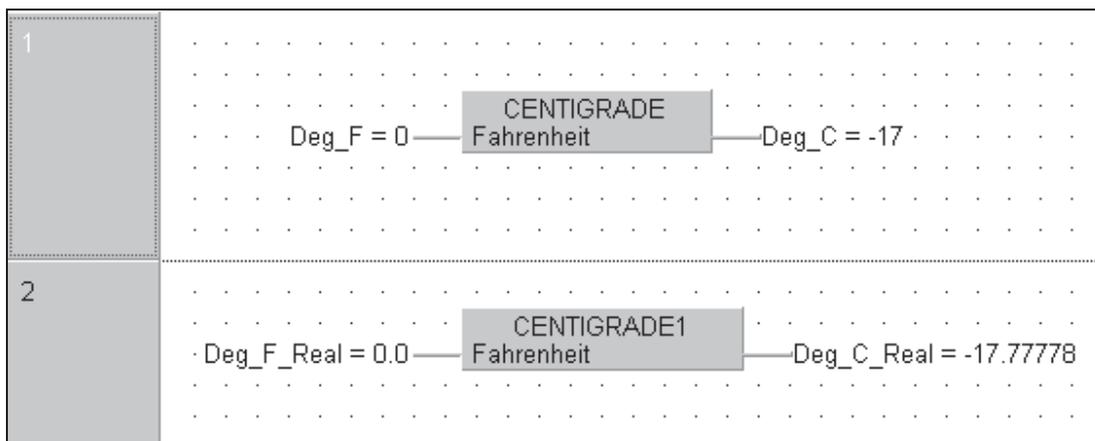
Числа формата REAL используют 2 последовательных регистра (32 бита) и хранятся в специальном портативном IEE-формате, см. назначение в приведенном выше примере GVL.

③ Полный программный модуль "Process" должен иметь следующий вид:



④ Сохраните проект, закройте все открытые диалоговые окна и компилируйте проект .

⑤ Передайте проект в ПЛК и проконтролируйте это звено, используя кнопку мониторинга  на панели инструментов:



Изменяйте значение входной переменной "Deg_F_Real" и наблюдайте выходной результат на дисплее. Обратите внимание на 7-разрядную точность формата с плавающей запятой.

6.2 Создание функционального блока

Цель:

Создать функциональный блок, действующий как пусковой переключатель со звезды на треугольник. Объявите следующие переменные:

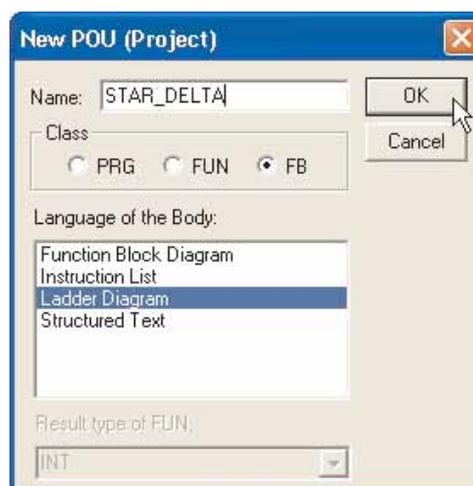
- Кнопка Старт: **START**
- Кнопка СТОП: **STOP**
- Контакт перегрузки: **OVERLOAD**
- Время переключения: **TIMEBASE**
- Временной регистр: **TIME_COIL**
- Выход контактора ЗВЕЗДА: **STAR_COIL**
- Выход контактора ТРЕУГОЛЬНИК: **DELTA_COIL**

Назовите функциональный блок **STAR_DELTA**.

Процедура:

① Запустите новый "Пустой" проект в GX-IEC Developer, называемый "Motor Control", без ПМ.

② Создайте новый программный модуль  под названием "STAR_DELTA" класса "Function Block" (FB) с типом языка тела **Ladder Diagram**.



Теперь "STAR_DELTA" появился на дереве программных модулей.

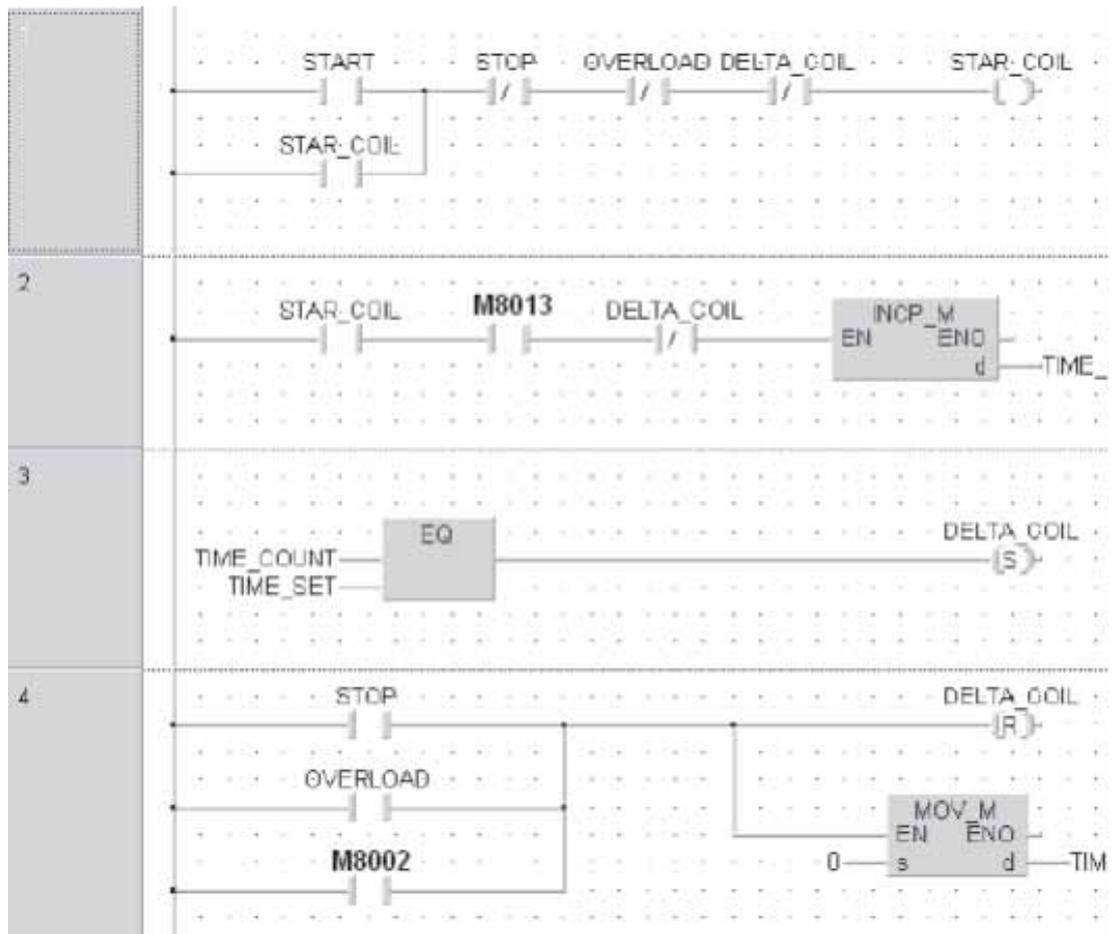
- ③ Щелкните один раз, чтобы открыть ветви заголовка и тела.
- ④ Дважды щелкните, чтобы открыть заголовок.

Объявление локальных переменных

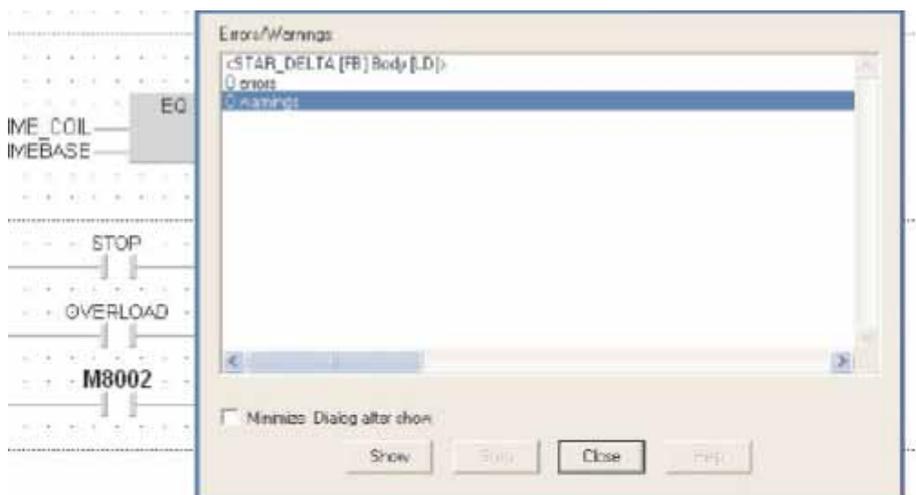
① Объявите переменные, как показано ниже:

	Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	START	BOOL	... FALSE	
1	VAR_INPUT	STOP	BOOL	... FALSE	
2	VAR_INPUT	OVERLOAD	BOOL	... FALSE	
3	VAR_INPUT	TIME_SET	INT	... 0	
4	VAR_OUTPUT	DELTA_COIL	BOOL	... FALSE	
5	VAR_OUTPUT	STAR_COIL	BOOL	... FALSE	
6	VAR_OUTPUT	TIME_COUNT	INT	... 0	

- ② Проверьте, сохраните и затем закройте окно заголовка.
- ③ Откройте тело и сформируйте звенья релейной диаграммы, как показано ниже:

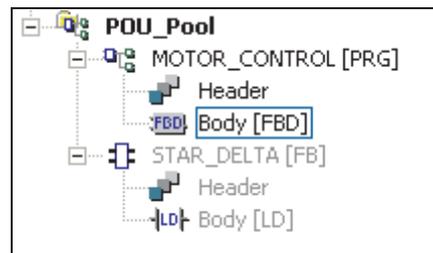


- ④ Проверьте тело, не должно быть ошибок и предупреждений!



Создание программного модуля новой программы "Motor Control"

- 1 Закройте все рабочие окна и любые диалоговые окна, которые могут быть открытыми.
- 2 Создайте новый программный модуль "MOTOR_CONTROL", Class "PRG" и "FBD" (Function Block Diagram) в качестве языка тела.



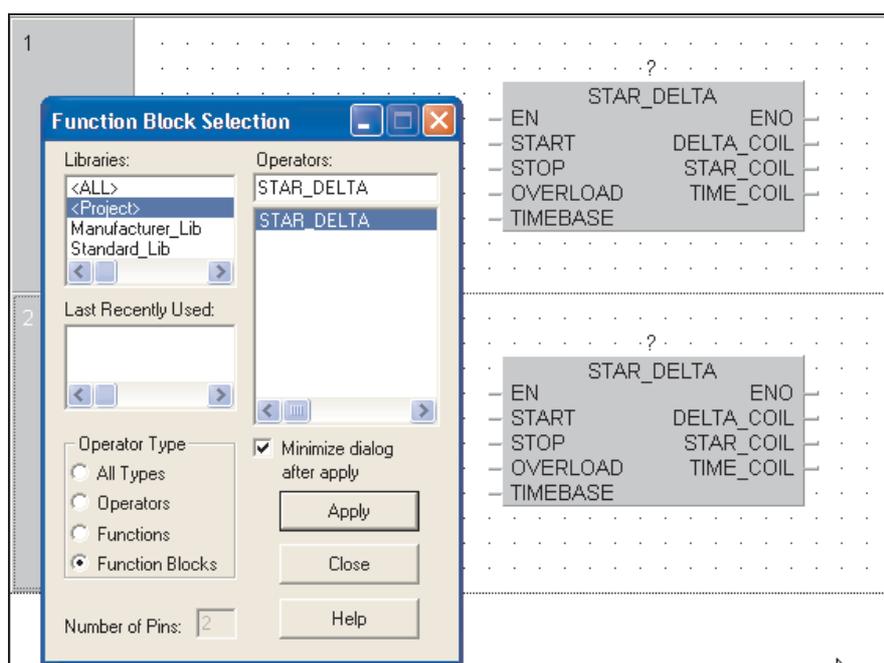
Создание нового списка глобальных переменных

Откройте GVL и введите следующие детали ввода-вывода:

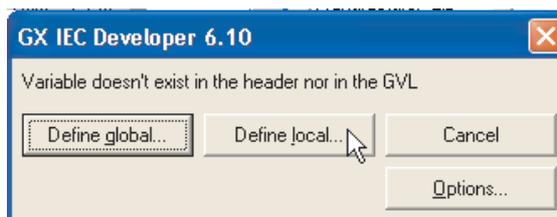
	Class	Identifier	MIT-Addr	IEC-Addr	Type	Initial
0	VAR_GLOBAL	START1	X0	%I0	BOOL	FALSE
1	VAR_GLOBAL	STOP1	X1	%I1	BOOL	FALSE
2	VAR_GLOBAL	OVERLOAD1	X2	%I2	BOOL	FALSE
3	VAR_GLOBAL	STAR_COIL1	Y00	%Q0	BOOL	FALSE
4	VAR_GLOBAL	DELTA_COIL1	Y01	%Q1	BOOL	FALSE
5	VAR_GLOBAL	TIME_COIL1	Q0	%MW0.0	INT	0
6	VAR_GLOBAL	START2	X3	%I3	BOOL	FALSE
7	VAR_GLOBAL	STOP2	X4	%I4	BOOL	FALSE
8	VAR_GLOBAL	OVERLOAD2	X5	%I5	BOOL	FALSE
9	VAR_GLOBAL	STAR_COIL2	Y02	%Q2	BOOL	FALSE
10	VAR_GLOBAL	DELTA_COIL2	Y03	%Q3	BOOL	FALSE
11	VAR_GLOBAL	TIME_COIL2	Q1	%MW0.1	INT	0

Назначение имен экземпляров

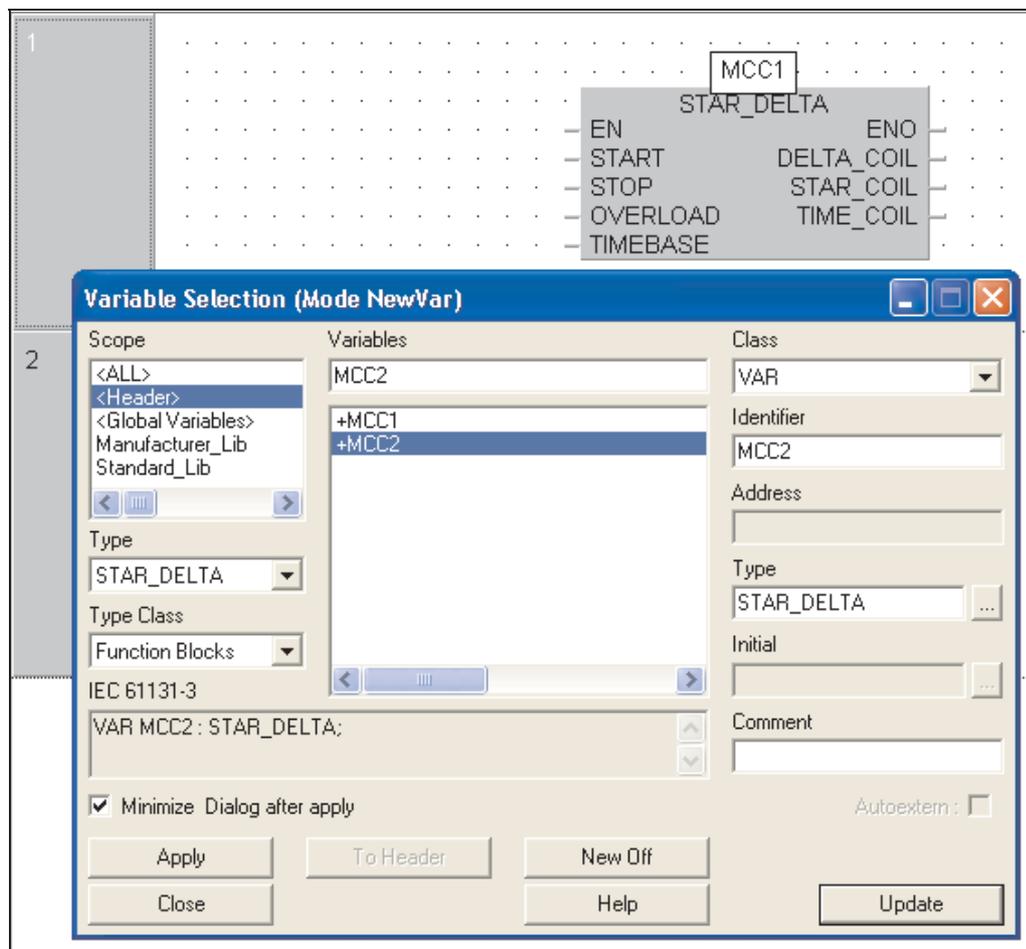
- 1 Откройте тело "MOTOR_CONTROL" и введите следующее, чтобы создать два звена. Поместите экземпляр функционального блока "STAR_DELTA" в каждое звено, как показано на следующем рисунке:



- ② Присвойте "имена экземпляров" обоим экземплярам функционального блока "STAR_DELTA", напечатав "MCC1" и "MCC2" в имена экземпляров над каждым экземпляром функционального блока. При запросе системы щелкните на **Define Local**.



Создайте элементы для имен экземпляров в заголовке для "MCC1" и "MCC2" следующим образом:

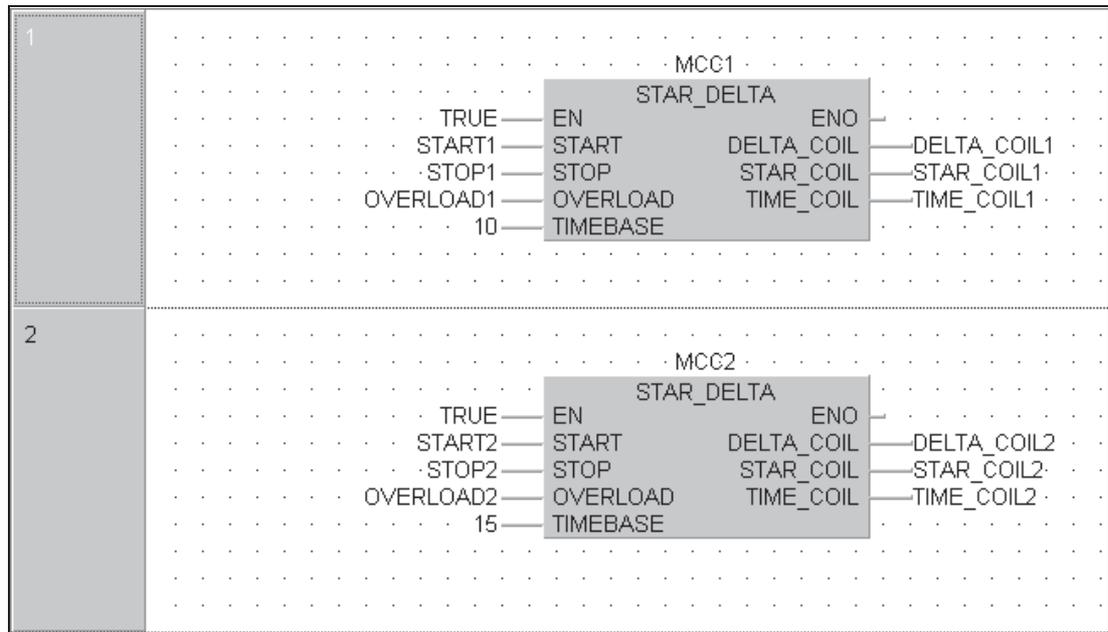


Экземпляр - это копия функционального блока для данного программного модуля. В этом примере просто напечатайте **MCC1** и **MCC2**. Заметим, что после введения экземпляры перечисляются в окне выбора переменных как "+MCC1" и "+MCC2" с типом: "STAR_DELTA".

Экземпляры должны объявляться в заголовке ПМ. Как можно видеть на рисунках выше, имена экземпляров добавляются таким же образом, как добавление любой другой новой переменной из тела программного модуля.

Назначение переменных функциональному блоку

Теперь закончите программный модуль, назначив переменные вашим функциональным блокам, как показано ниже:



ПРИМЕЧАНИЯ

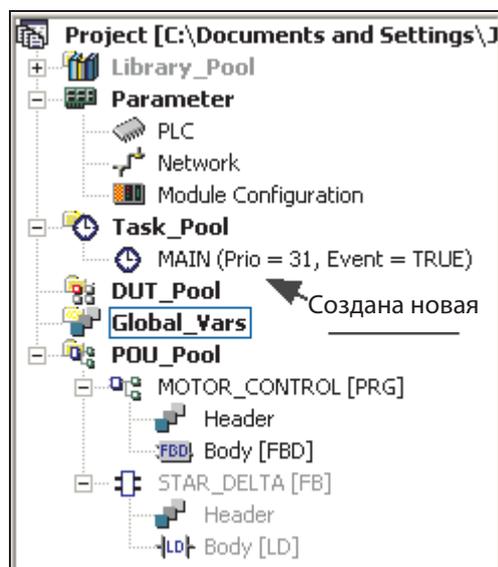
Могут использоваться адреса или объявление символических переменных Mitsubishi. Однако если используются прямые адреса MELSEC Mitsubishi, то программа больше не будет соответствовать соглашениям IEC.

Определив переменную "TRUE", как показано выше, вы автоматически назначите "нормально замкнутый" контакт (SM400 Q-серии), что изящнее и соответствует соглашениям IEC.

Функциональный блок "STAR_DELTA" может многократно использоваться в проекте и должен применяться с различными именами экземпляров.

Создание новой задачи:

- ① Создайте новую задачу "MAIN" в задачном пуле:

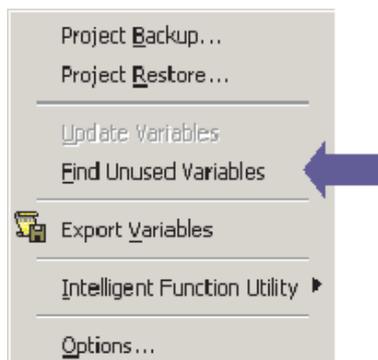


- ② Дважды щелкните на задаче и свяжите программный модуль "MOTOR_CONTROL" с задачей "MAIN":

POU name	Comment
MOTOR_CONTROL	...

- ③ Сохраните программу, закройте все окна и диалоговые окна.

Найти неиспользованные переменные



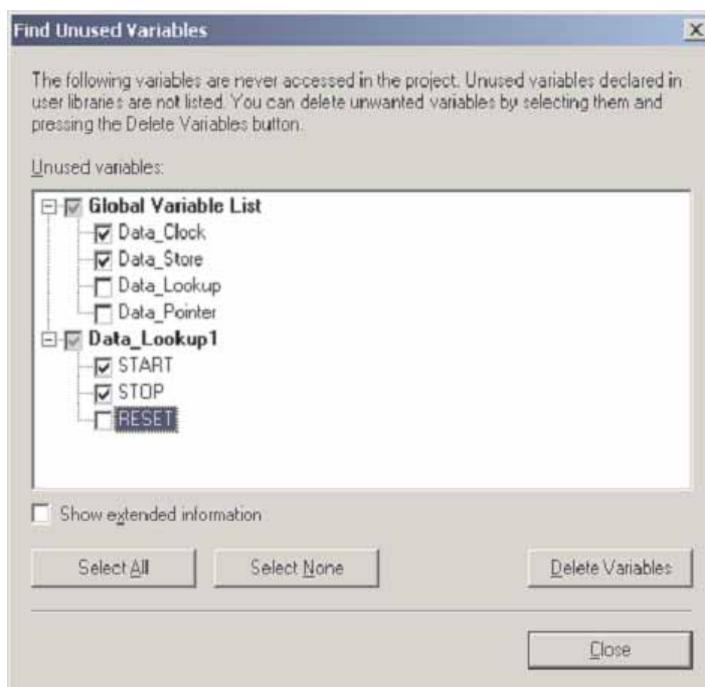
Используя функцию **Extras** → **Find unused Variables** вы можете найти и удалить все неиспользованные глобальные и локальные переменные, которые объявлены, но не используются в проекте.

Неиспользуемые глобальные и локальные переменные будут обнаружены во всем проекте, за исключением пользовательских библиотек.

ПРИМЕЧАНИЕ

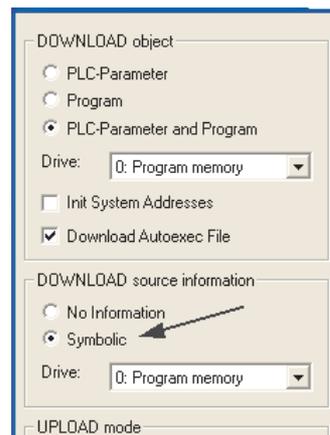
Обнаружение неиспользованных переменных можно выполнить, если проект был создан и с тех пор не изменялся. В противном случае будет показано предупреждающее сообщение.

Каждая неиспользованная переменная указывается под контейнером с ее объявлением: списком глобальных переменных для глобальных переменных, или соответствующим программным модулем для локальных переменных. Перечислены только контейнеры, содержащие неиспользованные переменные. Например, если глобальная переменная отсутствует, список глобальных переменных не приводится. Контейнеры представлены полужирным текстом и находятся над содержащимися в них компонентами.

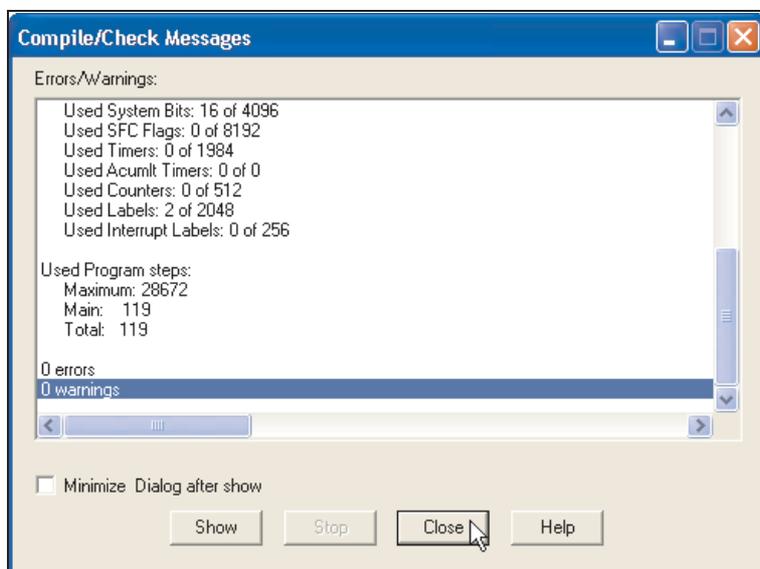


ПРИМЕЧАНИЕ

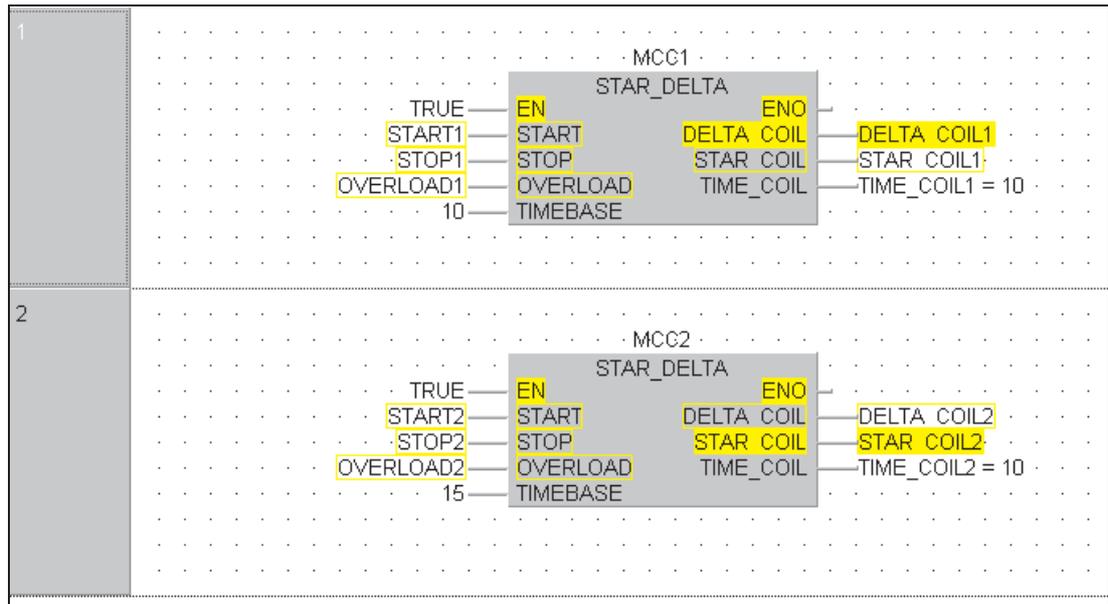
Это может значительно сократить размер исходного текста программы, что особенно важно, если для загрузки была выбрана опция передать весь символический **Symbolic** (исходный) код в ПЛК:



Компилируйте программу стандартным образом, используя кнопку "Rebuild All"  на панели инструментов:



Откройте программный модуль "MOTOR_CONTROL" и контролируйте  правильную работу программы.

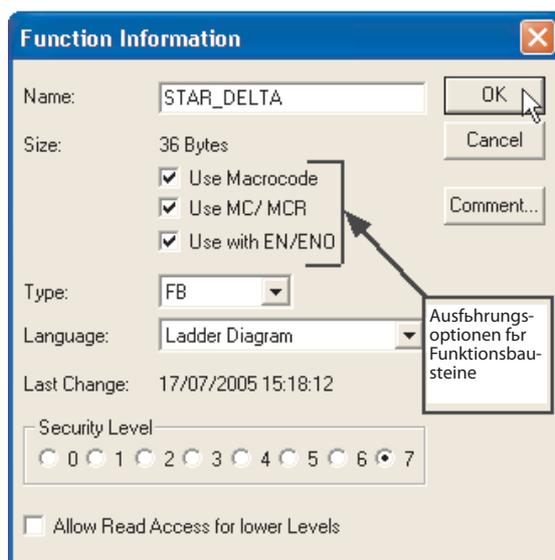


6.3 Режимы выполнения функциональных блоков

Функциональные блоки могут выполняться различными способами:

- Выполнение макрокоманды
- Выполнение MC - MCR
- Использование с EN/ENO

Режим выполнения выбирается в диалоговом окне **Function Information**:



Как установить режим выполнения:

- ① Выберите функциональный блок в окне "Project Navigator".
- ② Откройте диалоговое окно **Function Information**, щелкнув правой кнопкой мыши и выбрав **Properties**.
- ③ Активизируйте флажок. Опцию MC-MCR можно активировать, только когда уже активированы две другие опции.

Это не создает каких-либо изменений в создании и программировании экземпляров в различных языках программирования.

6.3.1 Выполнение макрокоманды

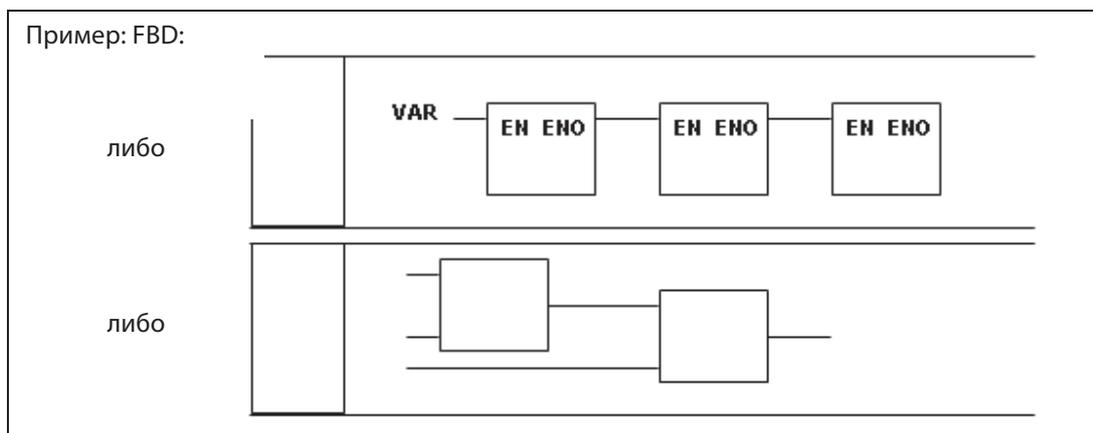
- Стандартное выполнение: Функциональный блок вызывается через системную метку.
- Выполнение макрокоманды: Функциональный блок расширяется внутренним образом.

Преимущества функциональных блоков с макрокомандой

С макрокомандой	Без макрокоманды (стандартное выполнение)
Для выполнения экземпляра функционального блока не требуются внутренние системные метки. Результат: число используемых функциональных блоков ограничено только размером памяти ПЛК, поскольку функциональные блоки не зависят от системных меток.	Каждый экземпляр использует внутренние системные метки (указатели). Результат: Поскольку число доступных системных меток ограничено (FX: 128, A: 256, Q: 1024) невозможно использовать больше, чем теоретическое ограниченное количество функциональных блоков. Практически их количество даже меньше, так как системные метки необходимы и для других внутренних процессов.
Ориентированное на пользователя выполнение функционального блока.	Структура функционального блока соответствует стандарту IEC 61131-3.
Отсутствуют ограничения на обработку таймеров и катушек в функциональном блоке.	Ограничения на обработку таймеров и катушек в функциональном блоке (подпрограмме).

6.3.2 Enable / Enable Output (EN/ENO)

- Вход EN делает функцию (или функциональный блок, см. ниже) условной (включает/выключает)
- ENO отражает состояние линии EN.
- В звене должны использоваться только инструкции, включающие или не включающие EN, не комбинируйте оба типа.
- Цепь EN/ENO должна включать все свои предварительные условия вначале:

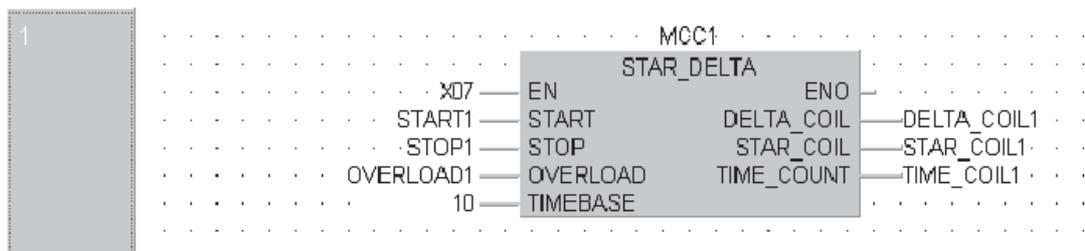


Определения функций

- Все операнды с суффиксом "_E" имеют EN / ENO линии, в противном случае нет.
- Все операнды с суффиксом "_M" являются инструкциями производителей; в данном случае из соответствующей системы команд Mitsubishi.
- Необходимо следить за тем, особенно при использовании редактора FBD, чтобы не нарушать правила программирования Mitsubishi. При создании цепей, как в предыдущем примере, соблазнительно соединить множество инструкций вместе, чтобы получить, например, необходимое вычисление. Однако, если выбранная инструкция Mitsubishi должна находиться в конечной позиции звена, почему она должна внезапно превратиться в последовательный элемент просто потому, что вы используете FBD?
- Выберите правильную инструкцию для работы; возможно, это будет команда из набора IEC.
- Также помните, что 16-битовое умножение Mitsubishi создает 32-битовый результат. Если используются переменные, то "тип" результата должен это отражать, т.е. операнды могут быть типа INT, а результат типа DINT.

Упражнение (Управление импульсами)

Отредактируйте функциональный блок STAR_DELTA, чтобы он имел возможность ввода/вывода EN/ENO. Разрешающий вход EN управляется внешним контактом MELSEC X17:

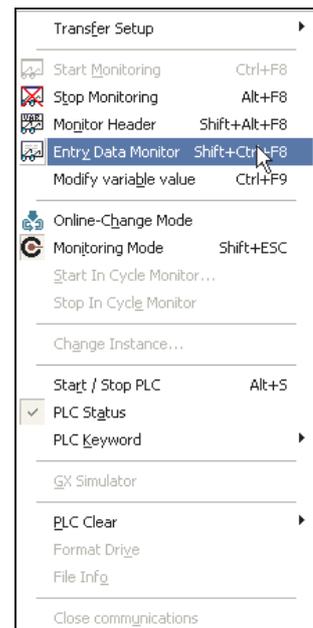


7 Расширенные функции мониторинга

Приведенные ниже диаграммы используются только для иллюстрации, используйте в следующих процедурах проект STAR_DELTA и его соответствующие операнды.

7.1 Контроль входных данных (EDM)

- ① Находясь в режиме **Monitor**, выберите **Entry Data Monitor** из меню **Online**:



Появится следующая таблица:

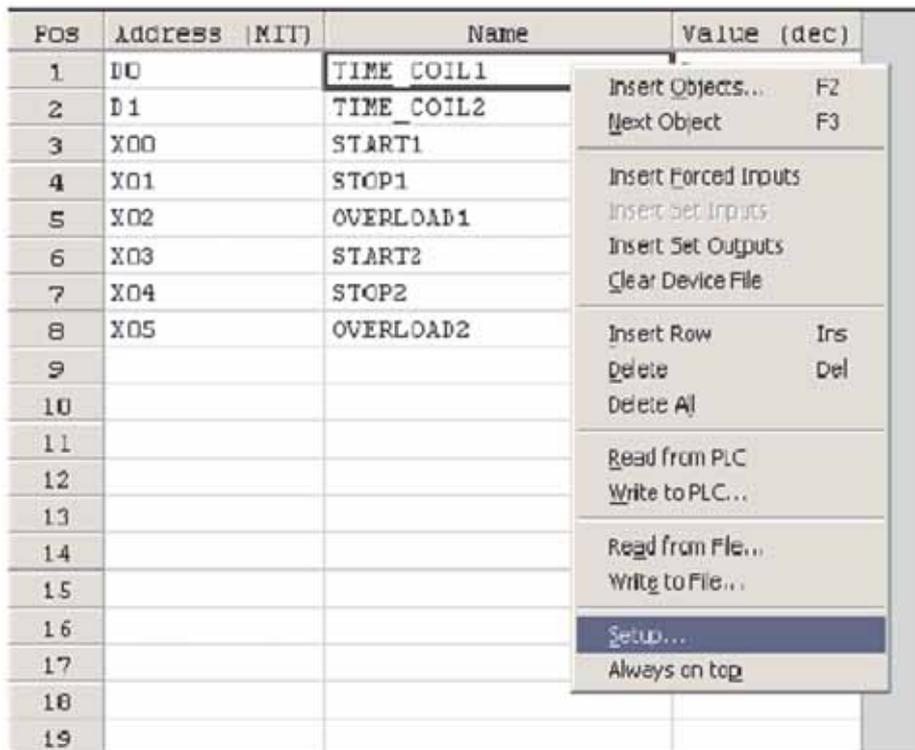
Pos	Address (MIT)	Name	Value (dec)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

- ② Щелкните на левом столбце адреса Mitsubishi **Address (MIT)** и напечатайте необходимый операнд; будет автоматически показан любой идентификатор вместе с текущим значением. Ширину столбцов можно изменять. В строке заголовков таблицы поместите курсор на левую границу столбца, который вы хотите изменить. Затем нажмите левую кнопку мыши и передвиньте границу влево или вправо. Отпустите левую кнопку мыши в желательной позиции.

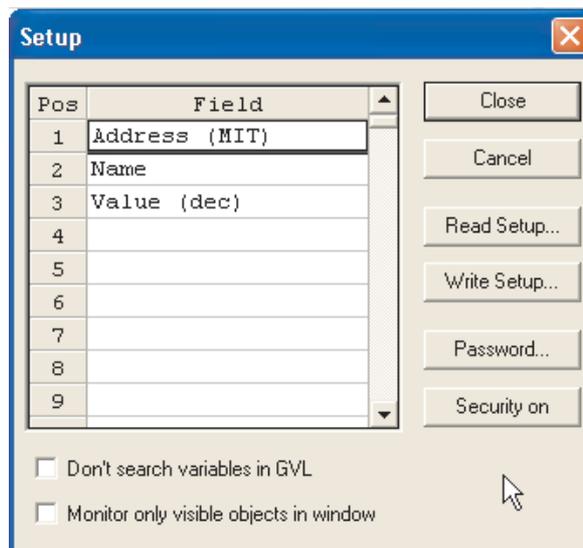
Pos	Address (MIT)	Name	Value (dec)
1	D0	TIME_COIL1	0
2	D1	TIME_COIL2	0
3	X00	START1	0
4	X01	STOP1	0
5	X02	OVERLOAD1	0
6	X03	START2	0
7	X04	STOP2	0
8	X05	OVERLOAD2	0

7.1.1 Настройка EDM

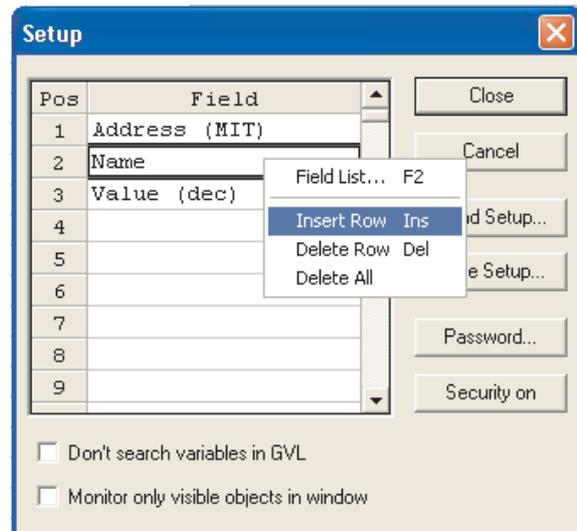
① Щелкнув правой кнопкой мыши, откройте следующее окно. Выберите **Setup**.



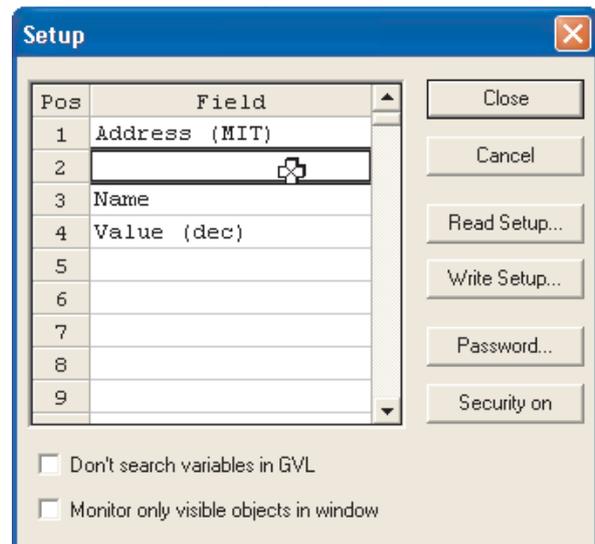
Окно **Setup** дает пользователю возможность конфигурировать EDM; щелкнув правой кнопкой мыши, вы откроете окно конфигуратора. Эта процедура добавляет в таблицу EDM столбцы для адреса IEC и монитора шестнадцатеричных значений.



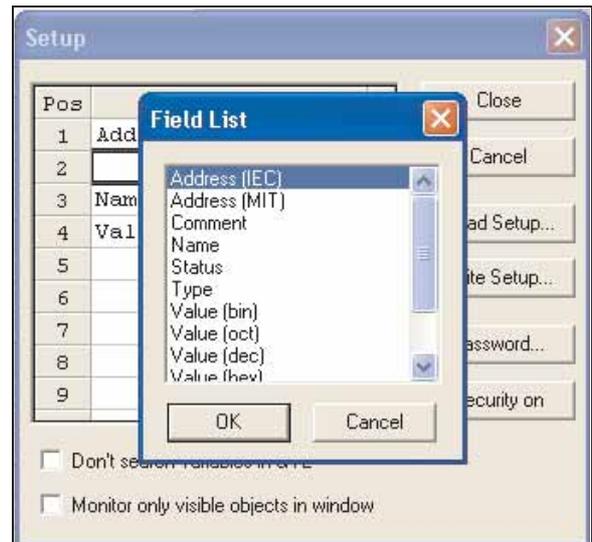
- ② Выделите или щелкните правой кнопкой на поле **Name** и выберите **Insert Row**, как показано.



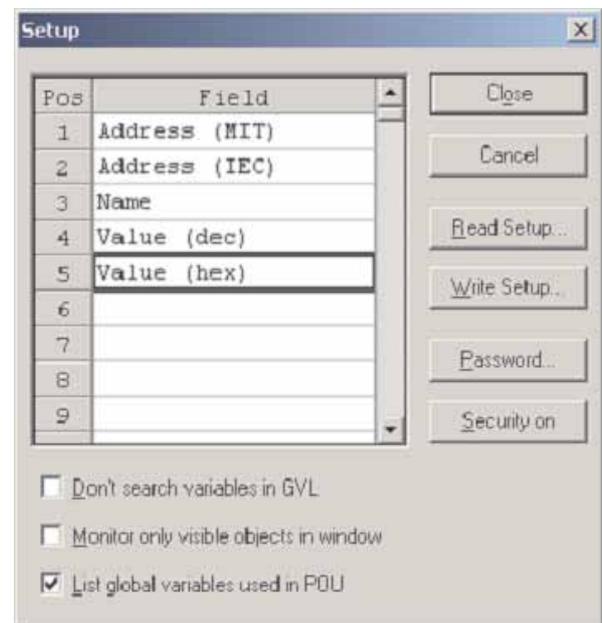
Появляется второе окно с опциями для этой строки; выберите Value (hex), Value (bin). Повторите для Address (IEC) и Type.



- ③ Дважды щелкните на пустом поле или нажмите F2 и выберите из списка **Address (IEC)**, как показано.



- ④ Щелкните на кнопке ОК, и элемент будет добавлен в структуру EDM. Добавьте **Value (hex)** в поле Pos 5 в таблице.



- ⑤ Щелкните, чтобы закрыть окно конфигурации, и наблюдайте измененную структуру EDM:

Pos	Address (MIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1	D0	↳N00.0	TIME_COIL1	0	0
2	D1	↳N00.1	TIME_COIL2	0	0
3	X00	↳IX0	START1	0	0
4	X01	↳IX1	STOP1	0	0
5	X02	↳IX2	OVERLOAD1	0	0
6	X03	↳IX3	START2	0	0
7	X04	↳IX4	STOP2	0	0
8	X05	↳IX5	OVERLOAD2	0	0

Таким образом таблицу EDM можно использовать, чтобы показывать различные данные в одной таблице.

Попробуйте отрегулировать ширину столбцов и изменить размер окна из меню **View**, чтобы показать полную картину. Размер экрана существенно зависит от разрешения дисплея установленного на используемом компьютере.

Отсюда можно ввести значения для любого показанного объекта; например, значение D100 можно изменить, напечатав число в соответствующем поле.

7.1.2 Ограничения монитора

ПРИМЕЧАНИЕ

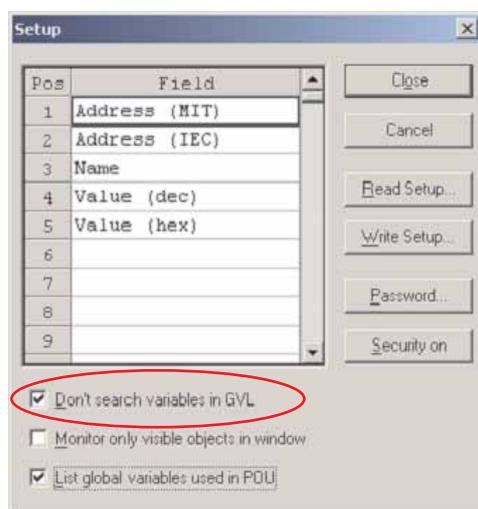
Помните, что поведение монитора зависит от кода, выполняющегося в ПЛК; если код ПЛК записывает константу в данный адрес, введенное значение будет перезаписано программой. Эта ситуация будет здесь преобладающей, так как значения D0 и D1 непрерывно записываются кодом ПЛК.

В этом примере программа циклически изменяет содержимое регистров D0 и D1.

Параметры конфигурации

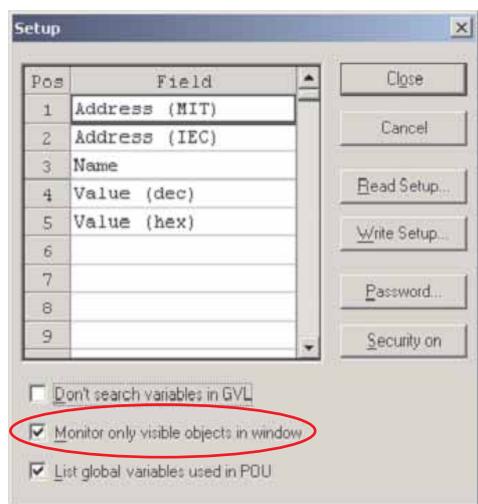
- **Don't Search Variables in GVL - Не искать переменные в списке глобальных переменных**

После того, как вы введете адрес Мицубиси, например, M0 в **Entry Data Monitor** система автоматически ищет идентификатор для этого устройства в списке глобальных переменных. В больших проектах на это может потребоваться значительное время. Вы можете заблокировать этот поиск, используя опцию **Don't Search Variables in GVL**.



- **Monitor only Visible Objects in Window - Контролировать только видимые объекты в окне**

По умолчанию контролируются все элементы в "Entry Data Monitor", даже если в настоящее время они не показаны. Вы можете ускорить время отклика, используя опцию **Monitor Only Visible Objects**.



7.1.3 Переключение логических переменных

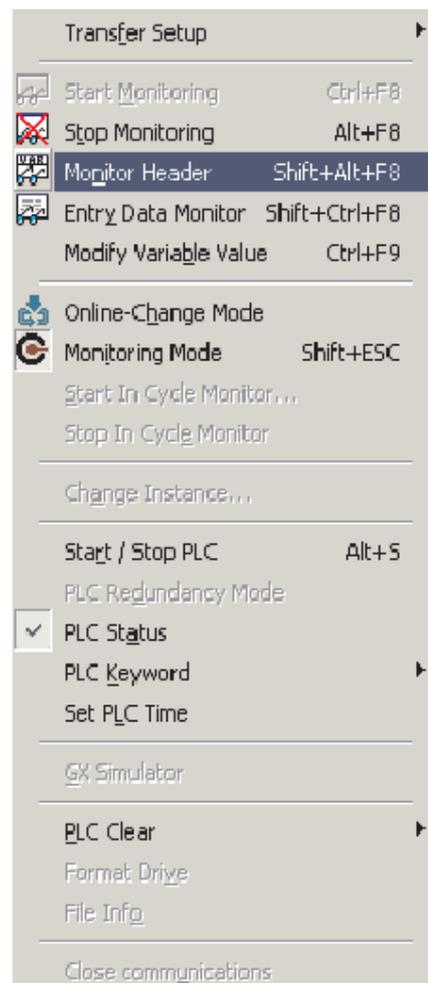
Если физический вход ПЛК неактивен, то можно переключать (включать/выключать) отображение входов в CPU, дважды щелкнув на поле значений для этих булевых адресов, как показано ниже:

Pos	Address (KIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1	D0	%M0.0	TIME_COIL1	10	A
2	D1	%M0.1	TIME_COIL2	0	0
3	X00	%IX0	START1	1	1
4	X01	%IX1	STOP1	0	0
5	X02	%IX2	OVERLOAD1	0	0
6	X03	%IX3	START2	0	0
7	X04	%IX4	STOP2	0	0
8	X05	%IX5	OVERLOAD2	1	1
9					
10					
11					
12					
13					
14					

Дважды щелкните, чтобы переключить состояние операндов

7.2 Мониторинг заголовков

В режиме мониторинга **Monitor Mode**, когда подсвечено тело программного модуля, имеется другая возможность - функция мониторинга заголовка **Monitor Header** в меню **Online**. Она также доступна из панели инструментов **Online**.



Теперь показаны и контролируются все элементы идентификаторов заголовка подсвеченного программного модуля:

Pos	Address (MIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1			-MOTOR_CONTROL		
2	X00	XIX0	START1	1	1
3	X01	XIX1	STOP1	0	0
4	X02	XIX2	OVERLOAD1	0	0
5	Y01	YOX1	DELTA_COIL1	1	1
6	Y00	YOX0	STAR_COIL1	0	0
7	D0	DNV0.0	TIME_COIL1	10	A
8	X03	XIX3	START2	0	0
9	X04	XIX4	STOP2	0	0
10	X05	XIX5	OVERLOAD2	1	1
11	Y03	YOX3	DELTA_COIL2	0	0
12	D1	DNV0.1	TIME_COIL2	0	0
13	Y02	YOX2	STAR_COIL2	0	0
14			+NCC1		
15			+NCC2		
16					

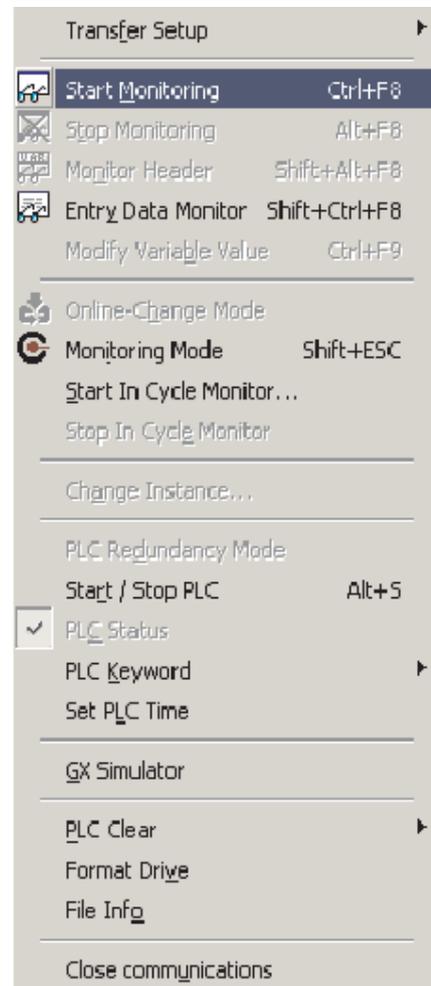
Учтите, что при мониторинге логические переменные в EDM показаны подсвеченными.

7.3 Основные возможности в режиме мониторинга

Можно контролировать несколько окон одновременно, сначала открыв их по отдельности и используя опцию "Tile Windows" в меню "Window". Важно понимать, что сначала при входе в

режим Монитор  будет контролироваться только целевое окно на первом плане.

Дальнейшие окна можно контролировать, сначала перенеся их на первый план и щелкая индивидуально на пункте **Start Monitoring** (Ctrl+F8) из меню **Online**:



ПРИМЕЧАНИЕ

Этот метод инициализации мониторинга предотвращает одновременный контроль всех открытых окон, даже если они открыты, но находятся не на первом плане. В противном случае возможен значительный рост трафика связи между ПЛК и компьютером. В конечном счете это может привести к очень медленным временам отклика мониторинга на экранах GX IEC Developer, особенно для ПЛК серии FX.

Одновременный мониторинг заголовка и тела

Ниже показан пример одновременного мониторинга программного модуля и его заголовка:

The screenshot displays two windows from a monitoring software. The top window, titled 'MOTOR_CONTROL (MOTOR_CONTROL)', shows a table of variables:

Pos	Address (BIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1			-MOTOR_CONTROL		
2	X00	X130	START1	1	1
3	X01	X131	STOP1	0	0
4	X02	X132	OVERLOAD1	0	0
5	Y01	Y021	DELTA_COIL1	1	1
6	Y00	Y020	STAR_COIL1	0	0
7	D0	W00.0	TIME_COIL1	10	A
8	X03	X133	START2	0	0
9	X04	X134	STOP2	0	0
10	X05	X135	OVERLOAD2	1	1
11	Y03	Y023	DELTA_COIL2	0	0
12	D1	W00.1	TIME_COIL2	0	0
13	Y02	Y022	STAR_COIL2	0	0
14			+MCC1		
15			+MCC2		

The bottom window, titled 'MOTOR_CONTROL (PRG) 0-17 (780)', shows two ladder logic diagrams for MCC1 and MCC2. Each diagram features a 'STAR_DELTA' function block with the following connections:

- MCC1:**
 - EN: X07
 - START: START1
 - STOP: STOP1
 - OVERLOAD: OVERLOAD1
 - TIMEBASE: 10
 - DELTA COIL: DELTA_COIL1
 - STAR COIL: STAR_COIL1
 - TIME_COUNT: TIME_COIL1 = 10
- MCC2:**
 - EN: TRUE
 - START: START2
 - STOP: STOP2
 - OVERLOAD: OVERLOAD2
 - TIMEBASE: 15
 - DELTA COIL: DELTA_COIL2
 - STAR COIL: STAR_COIL2
 - TIME_COUNT: TIME_COIL2 = 0

7.4 Мониторинг с использованием "бланка перевода" Mitsubishi

Также можно производить контроль, используя обозначение Mitsubishi Kn (официально - "Бланк перевода") для логических объектов. Например, K1X0 контролирует X0 - X3, как показано в следующем примере:

Pos	Address (MIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1	D0	%X0.0	TIME_COIL1	10	A
2	D1	%X0.1	TIME_COIL2	0	0
3	X00 0	%IX0	START1	1	1
4	X01	%IX1	STOP1	0	0
5	X02	%IX2	OVERLOAD1	0	0
6	X03	%IX3	START2	0	0
7	X04	%IX4	STOP2	0	0
8	X05	%IX5	OVERLOAD2	1	1
9					
10	K1X5	%IX19.1.5	K1X5	1	1
11					
12					

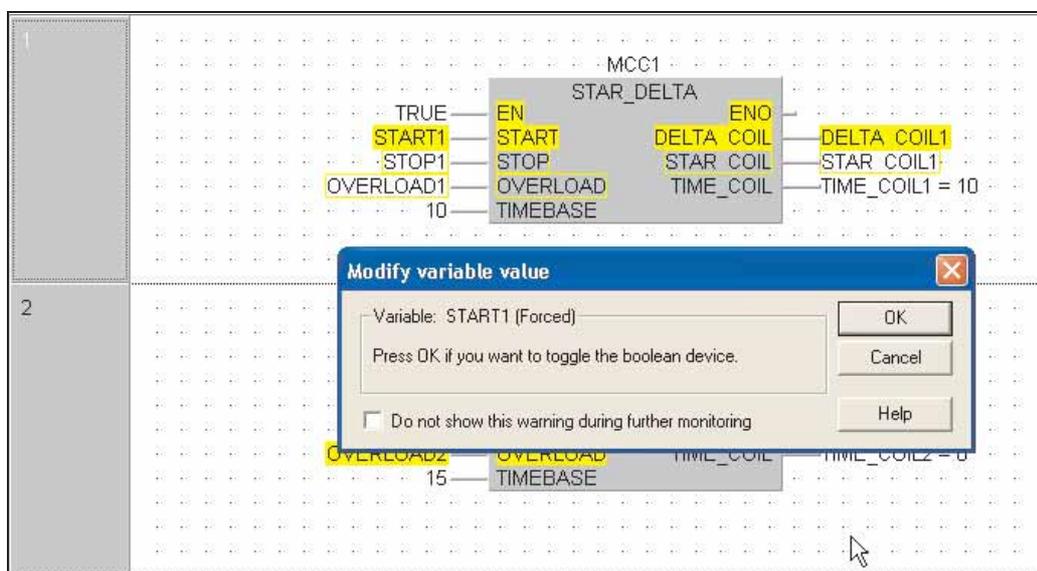
7.5 Модификация значений переменных из тела программного модуля

В режиме Monitor можно изменить значение переменной из тела программного модуля. Вы можете переключать логическую переменную или записывать значение в переменные типа Integer/Real, и т.д. Для этого дважды щелкните на метке переменной, чтобы вызвать функцию. Появится это диалоговое окно. Щелкните на **OK**, чтобы установить переменную, снова щелкните на **OK**, чтобы сбросить переменную. Если код ПЛК записывает в данную переменную, то он перезапишет это значение.

Диалоговое окно можно заблокировать, так что операция производится просто мышью.

Для переменных формата "Integer/Real" используйте ту же процедуру, т.е. дважды щелкните на имени переменной, находясь в режиме "Monitor". Новое значение можно ввести как десятичное или как шестнадцатеричное значение.

Как и ранее, если код ПЛК записывает в данную переменную, то он перезапишет это значение.



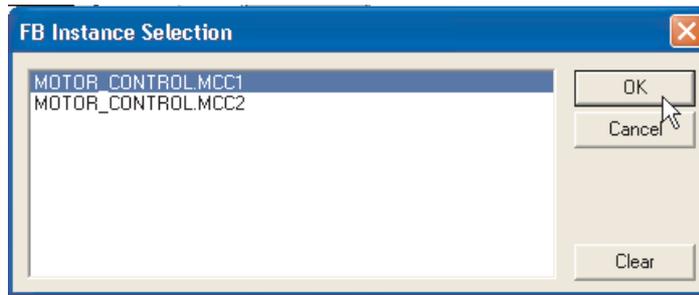
ПРИМЕЧАНИЕ

Обе операции также работают с прямыми адресами MELSEC (дальнейшие иллюстрации см. в предыдущем разделе: "Функции").

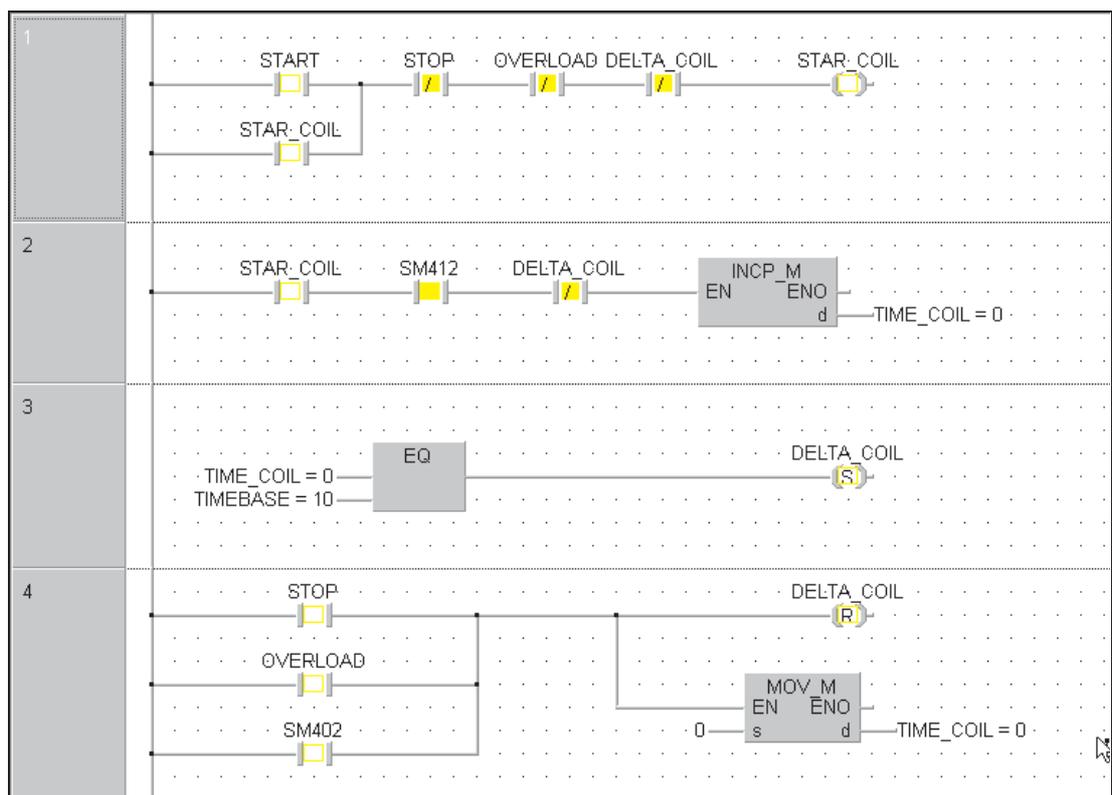
7.6 Мониторинг "экземпляров" функциональных блоков

Отдельные "экземпляры" функциональных блоков можно контролировать независимо.

- Чтобы контролировать экземпляр программного модуля функционального блока STAR_DELTA в текущем проекте, откройте тело программного модуля и щелкните на кнопке режима Monitor . Появится следующее диалоговое окно выбора:



- Выберите экземпляр функционального блока MOTOR_CONTROL.MCC1 и наблюдайте контролируемую страницу:



Этот способ позволяет автономно контролировать каждый экземпляр любого функционального блока.

8 Принудительная установка цифровых входов и выходов

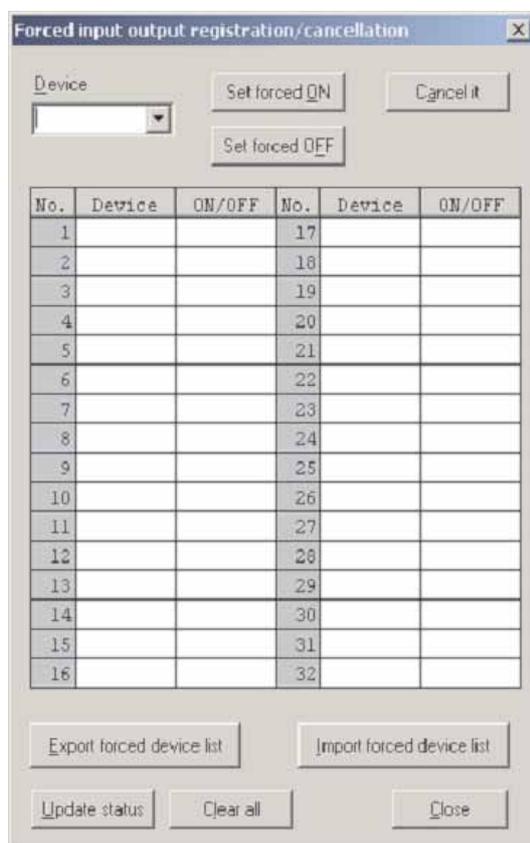
Эта функция GX IEC Developer позволяет принудительно устанавливать регистры физических аппаратных входов и выходные регистры независимо от цикла программы.

Хотя в реальных ситуациях эту функцию следует использовать очень осторожно, она весьма полезна, позволяя перезаписывать состояния всех операндов физических входов и выходов.

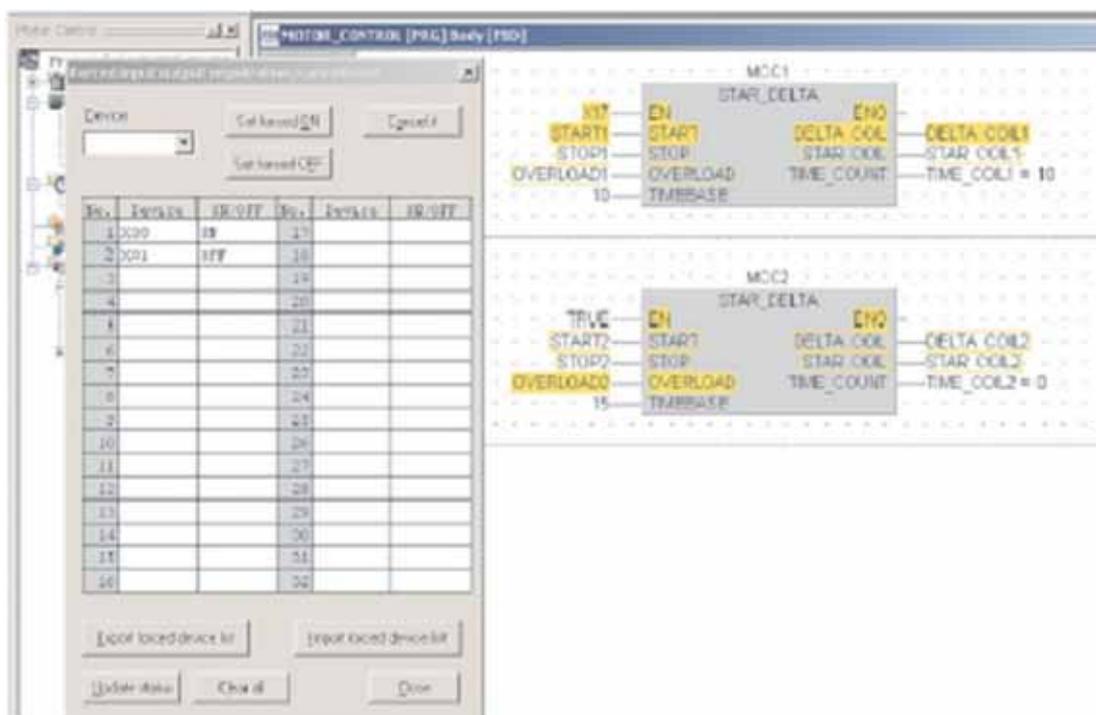
- ① Чтобы активизировать эту функцию и выбрать **Forced input output registration/cancellation**, выберите ее из меню **Debug** следующим образом:



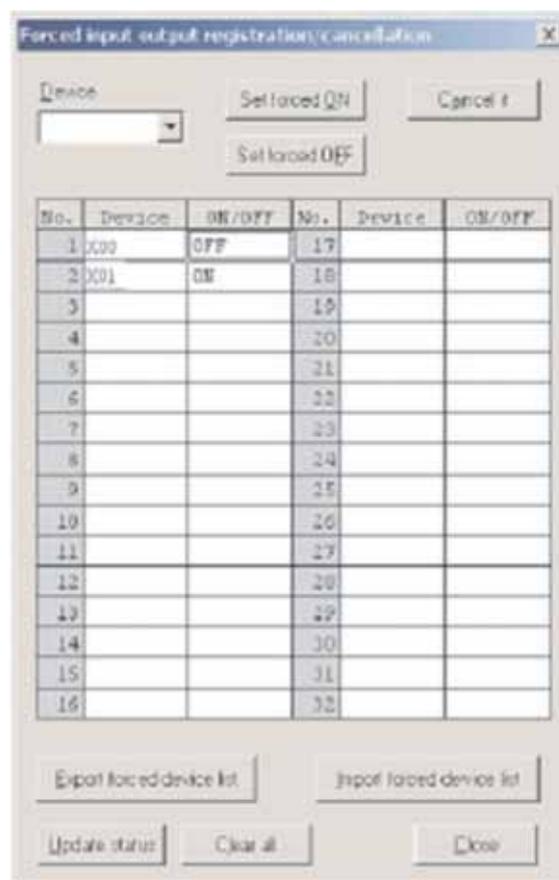
Появится следующее окно:



- ② Введите X10 и X11 в диалоговое окно **Device** и щелкните на кнопке **Set Forced ON** для обеих переменных:

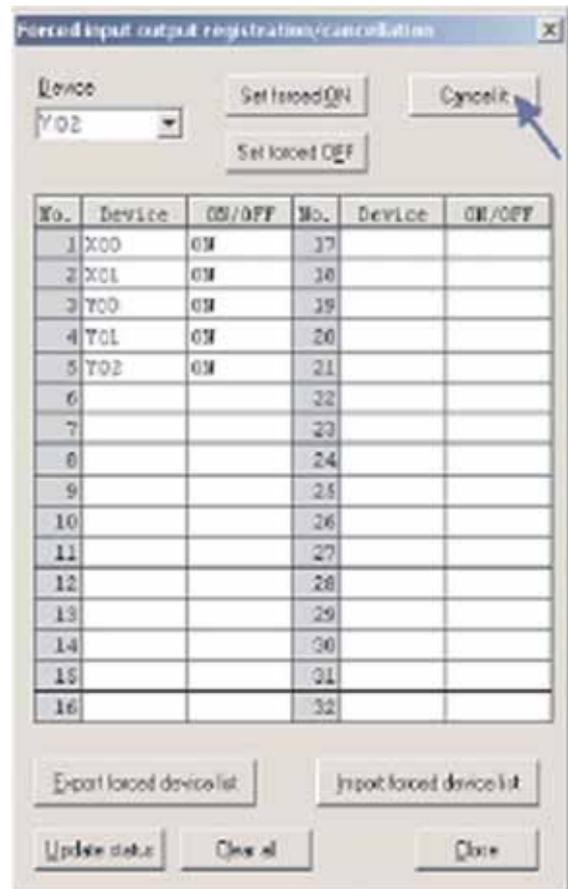


- ③ Чтобы переключить состояние X10 или X11, дважды щелкните левой кнопкой мыши на поле состояния **ON/OFF**.

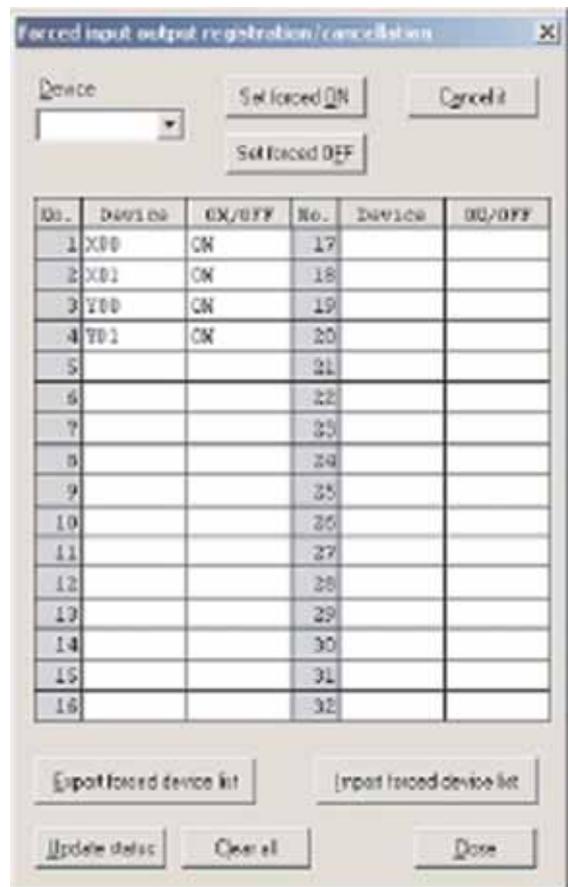


- ④ Воспользуйтесь этим методом принудительной установки для Y0, Y1 и Y2, отмечая воздействие на операнды.

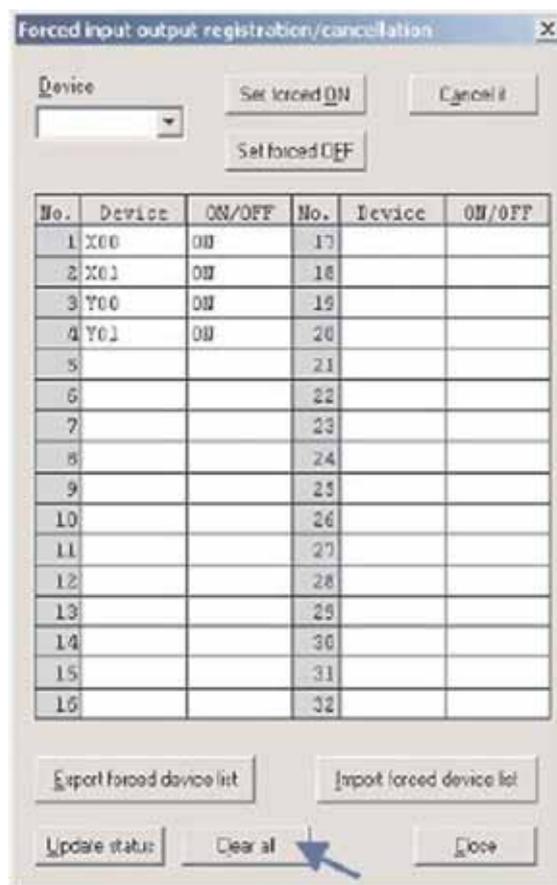
- ⑤ Чтобы сбросить принудительную установку для отдельного операнда, введите операнд, затем щелкните на кнопке **Cancel it**.



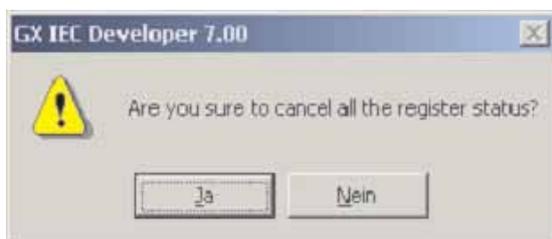
- ⑥ Дисплей примет следующий вид:



- ⑦ Чтобы сбросить все принудительные установки, зарегистрированные в ЦП, щелкните на кнопке **Clear All**.



- ⑧ Подтвердите запрос на отмену в следующем окне:



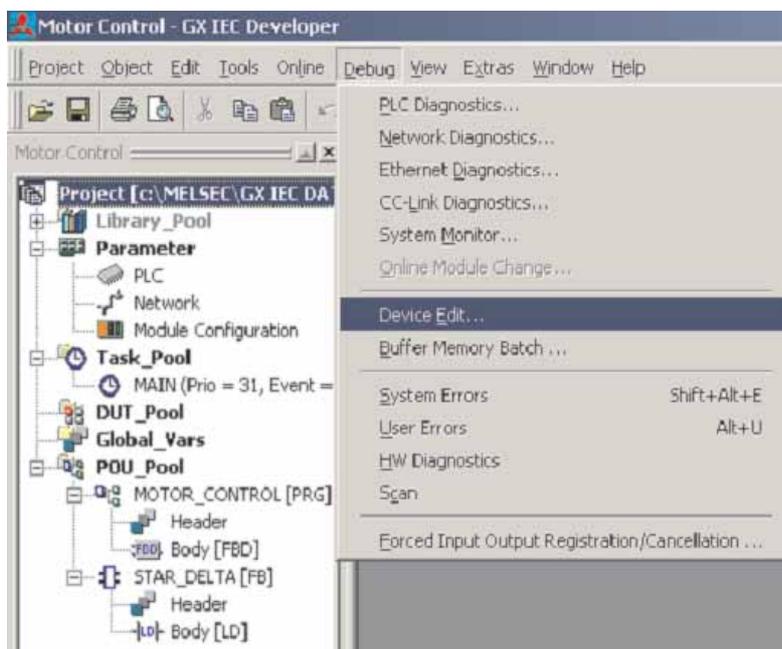
ПРИМЕЧАНИЕ

Отдельные принудительные установки можно удалить из таблицы активных принудительных установок, щелкнув на кнопке Cancel it для соответствующего элемента.

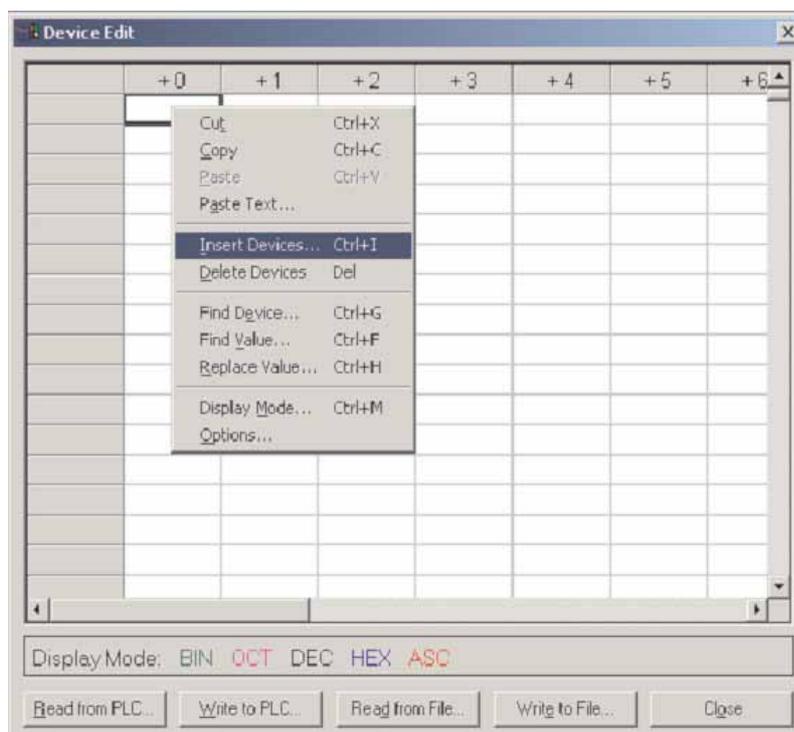
9 Функция Device Edit

Функция **Device Edit** сходна с набором **D,W,R** в MELSEC MEDOC и функцией **Device Memory** в программе GX IEC Developer.

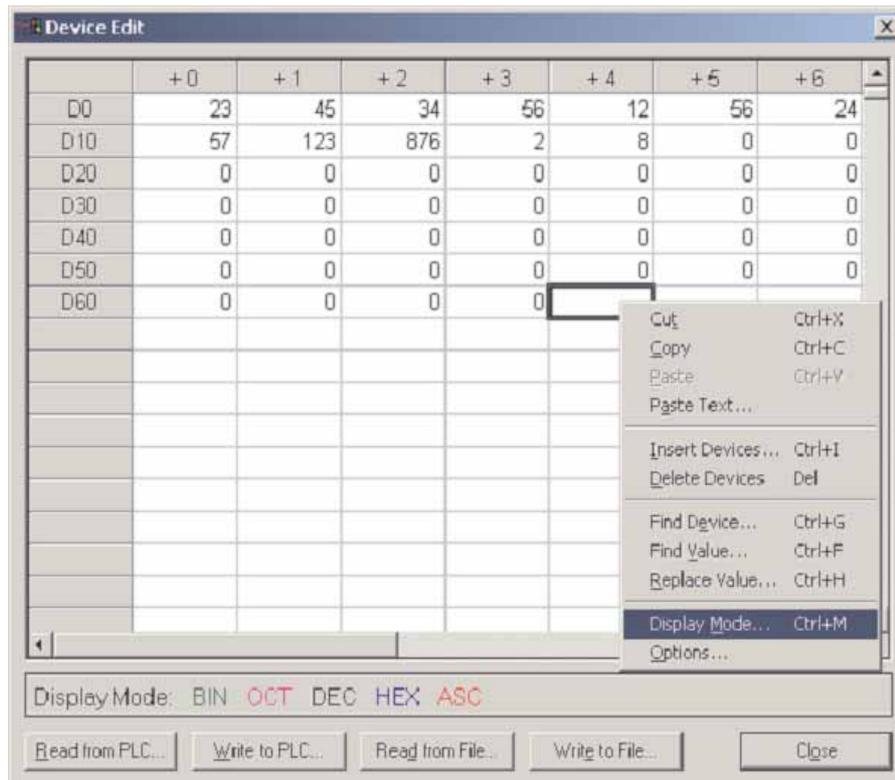
- ① Выберите **Device Edit** из меню **Debug**.



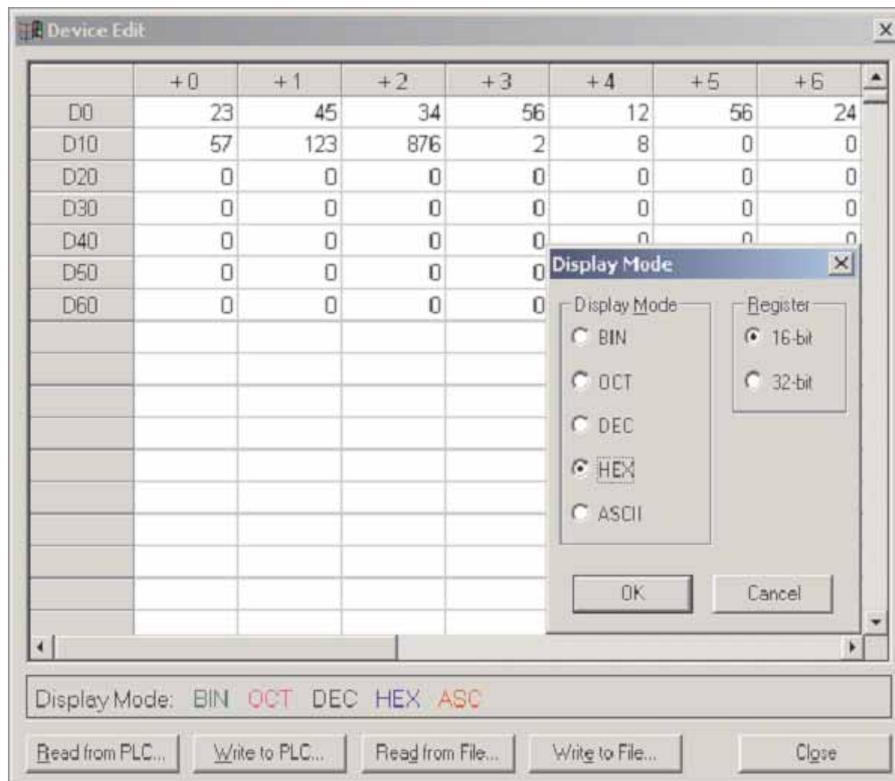
- ② Выделите ячейку в верхнем левом углу. Щелкните правой кнопкой мыши и затем выберите **Insert Devices**:



- ④ Выделите строку, щелкнув на левой ячейке, например, "D0". Выберите **Display Mode:**



Это окно позволяет изменить формат представления на дисплее - попробуйте **HEX**.



Вы должны заметить, что теперь в выбранной строке значения показаны в 16-ричном формате, а другие значения остаются неизменными. Фактически, в отдельных ячейках можно задать различные форматы отображения, поэтому эта возможность обеспечивает очень высокую гибкость.

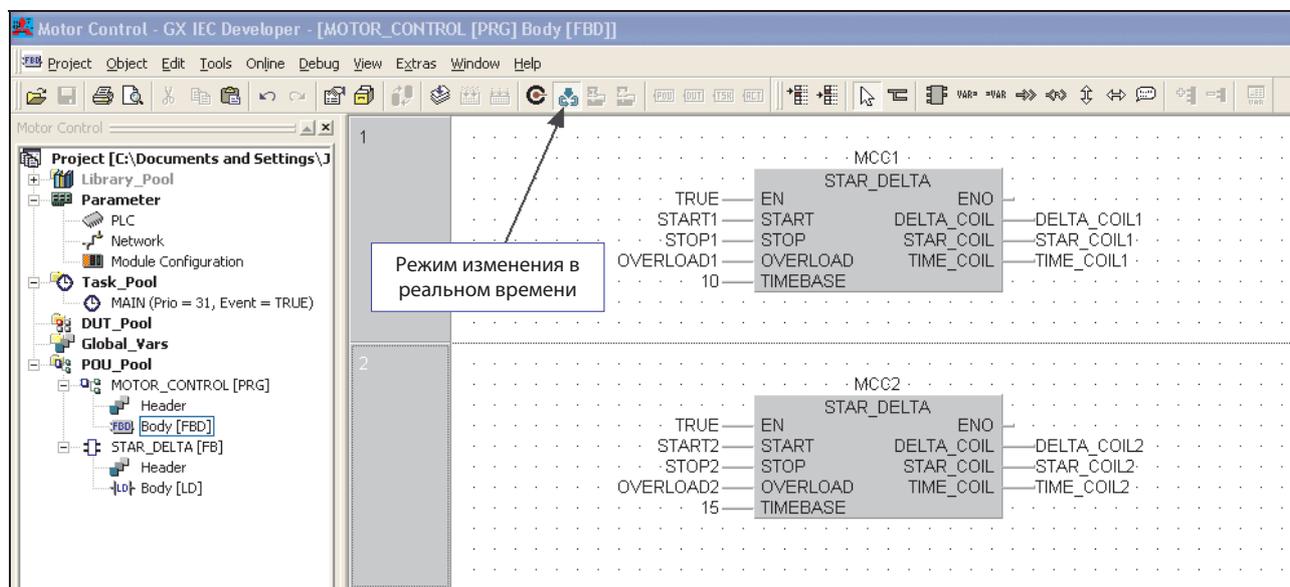
10 Режим онлайн

Имеются два метода, позволяющие вызвать редактирование в режиме онлайн - через меню **Online** или значок на панели инструментов. Используйте **Save as** в меню Project, чтобы создать копию текущего проекта. Переименуйте копию в "Motor_Control_Mod". Следующие операции будут применяться к этой модифицированной программе.

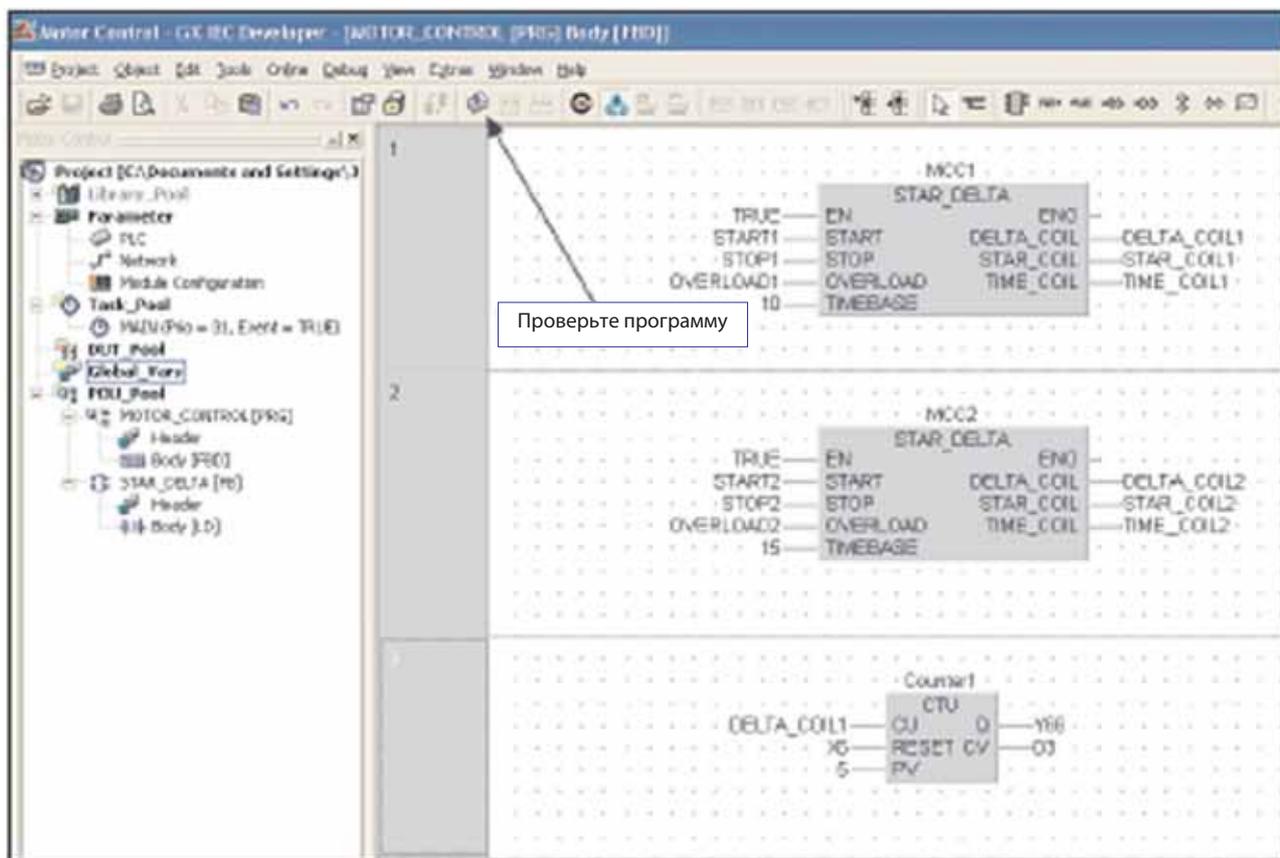
Компилируйте проект и загрузите его в ПЛК.

10.1 Режим изменения в реальном времени

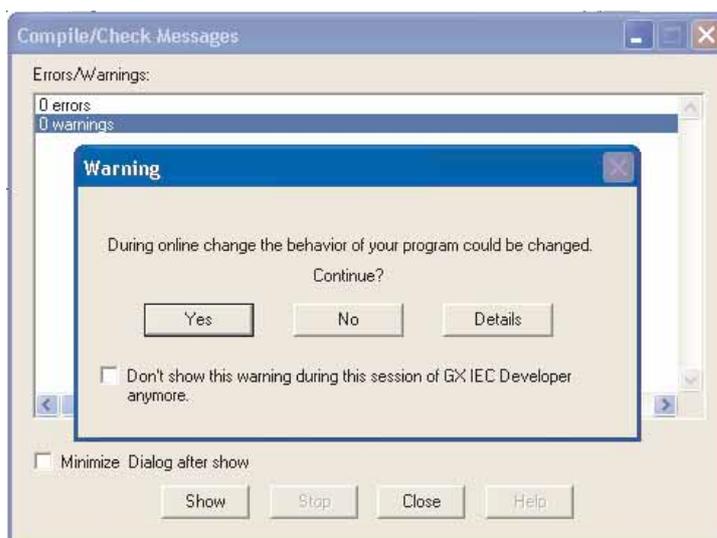
① Откройте тело программного модуля "MOTOR_CONTROL" и выберите **Online change mode**:



- ② Добавьте дополнительное звено, как показано ниже:



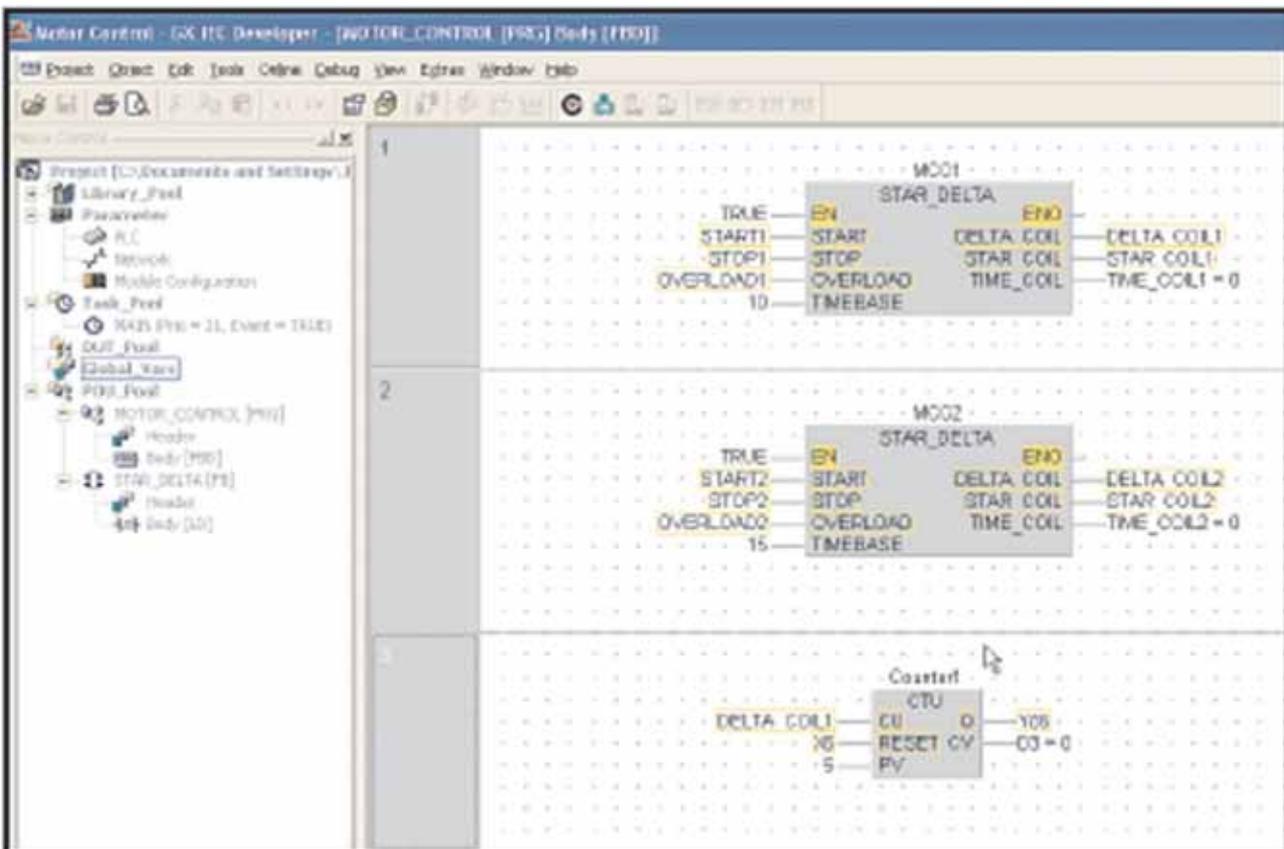
- ③ Затем щелкните мышью вне этого звена или на кнопке проверки, и изменения компилируются и посылаются в ПЛК автоматически после запроса о выполнении или прерывании действия:



ПРИМЕЧАНИЕ

Редактирование в режиме онлайн разрешено, только если код резидентного проекта идентичен коду в ПЛК.

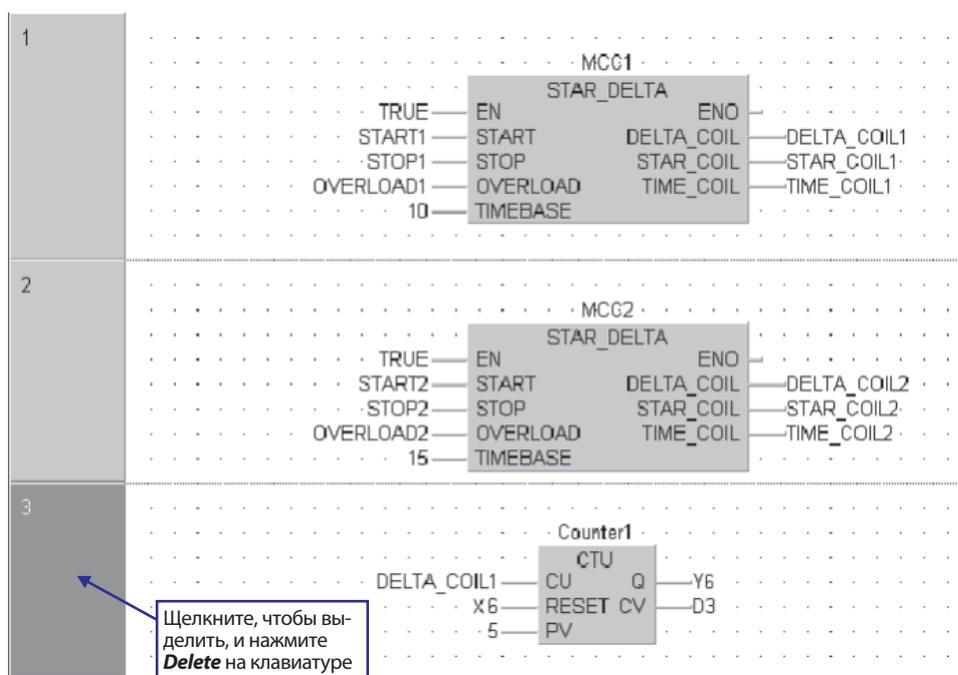
④ Войдите в режим Monitor и наблюдайте работу модифицированного блока:



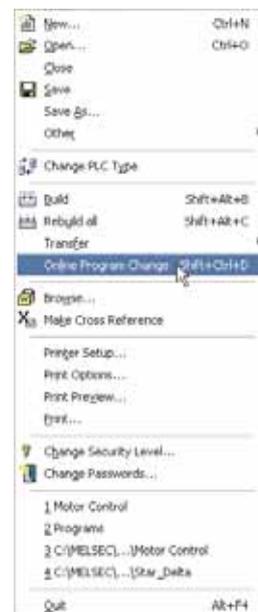
10.2 Изменение программы в режиме онлайн

Если необходимо добавить или удалить целые звенья, должна использоваться операция **Online Program Change**. Этот метод предпочтителен для внесения изменений в программу в режиме онлайн. Приведем пример: Если недавно добавленное звено счетчика необходимо удалить из программы, выполните следующую процедуру (Перед продолжением помните, что программы в ПЛК и GX IEC Developer должны быть идентичными).

- 1 Выделите звено 3 на теле программного модуля "MOTOR_CONTROL" и нажмите **Delete** на клавиатуре.

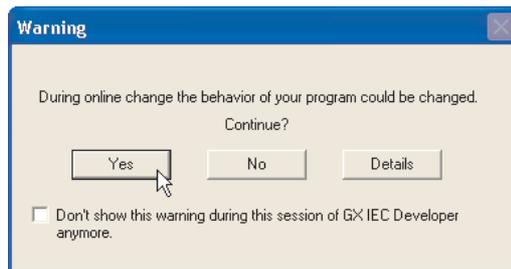


- 2 Вызовите функцию **Online Program Change** из меню **Project**. GX IEC Developer автоматически компилирует и запишет изменение в режиме реального времени.

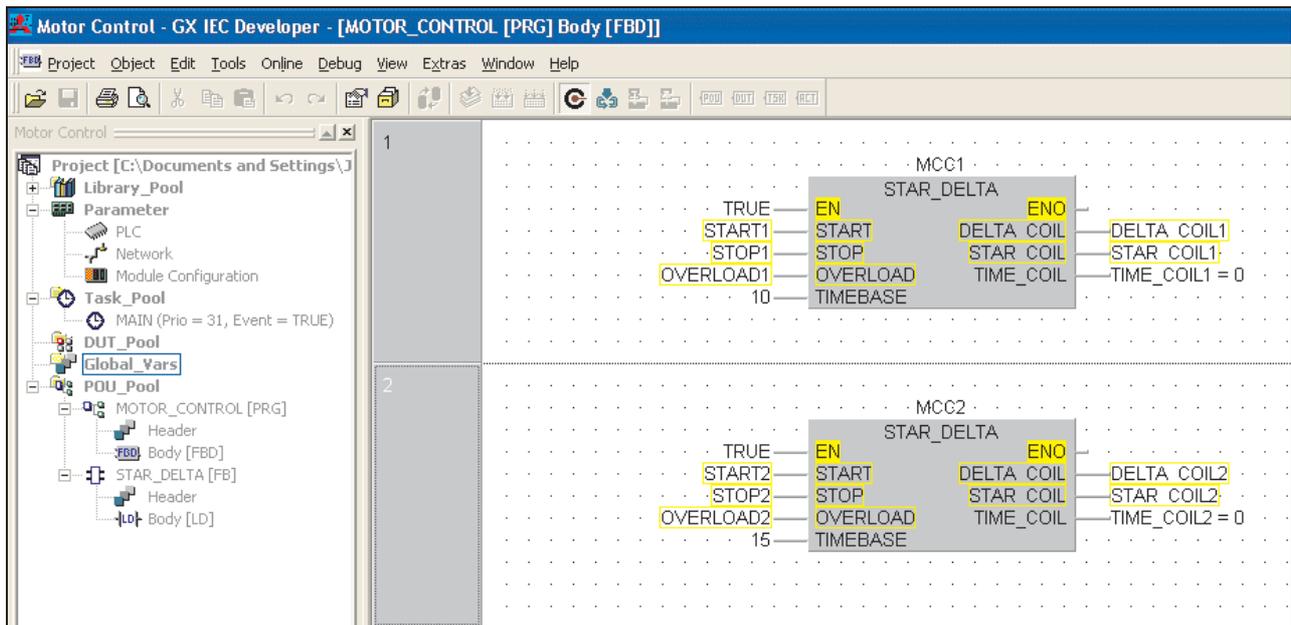


В этой точке система выдаст запрос о продолжении или прерывании процесса.

- ③ Щелкните на **Yes** и ждите, пока закончится процесс синхронизации загрузки:



- ④ Подтвердите правильность работы, войдя в режим Monitor в активном программном модуле.



11 Типы блоков данных (DUT)

Следующий пример иллюстрирует работу DUT (Data Unit Types).

Чтобы проиллюстрировать процедуры создания и использования DUT, будет использован предыдущий пример "управление двигателем".

Можно создавать пользовательские типы блоков данных (DUT). Это может быть полезным для программ, содержащих общие части, например, управление числом идентичных пускателей электродвигателей "Звезда - треугольник". Поэтому может быть создан тип блока данных, названный "SD", состоящий из структур различных элементов, например, INT, BOOL и т.д.

Заканчивая список глобальных переменных, можно использовать идентификаторы типа SD. Это означает, что предварительно заданную группу "SD" можно использовать с элементами, определенными для управления каждым двигателем, что сокращает время разработки и позволяет повторно использовать DUT вместе с функциональными блоками.

Если элемент START существует в типе "SD", то он может повторно использоваться в каждом экземпляре блока управления двигателем "Звезда - треугольник", когда объявлен в GVL; STAR_DELTA1.START, STAR_DELTA2.START и т.д.

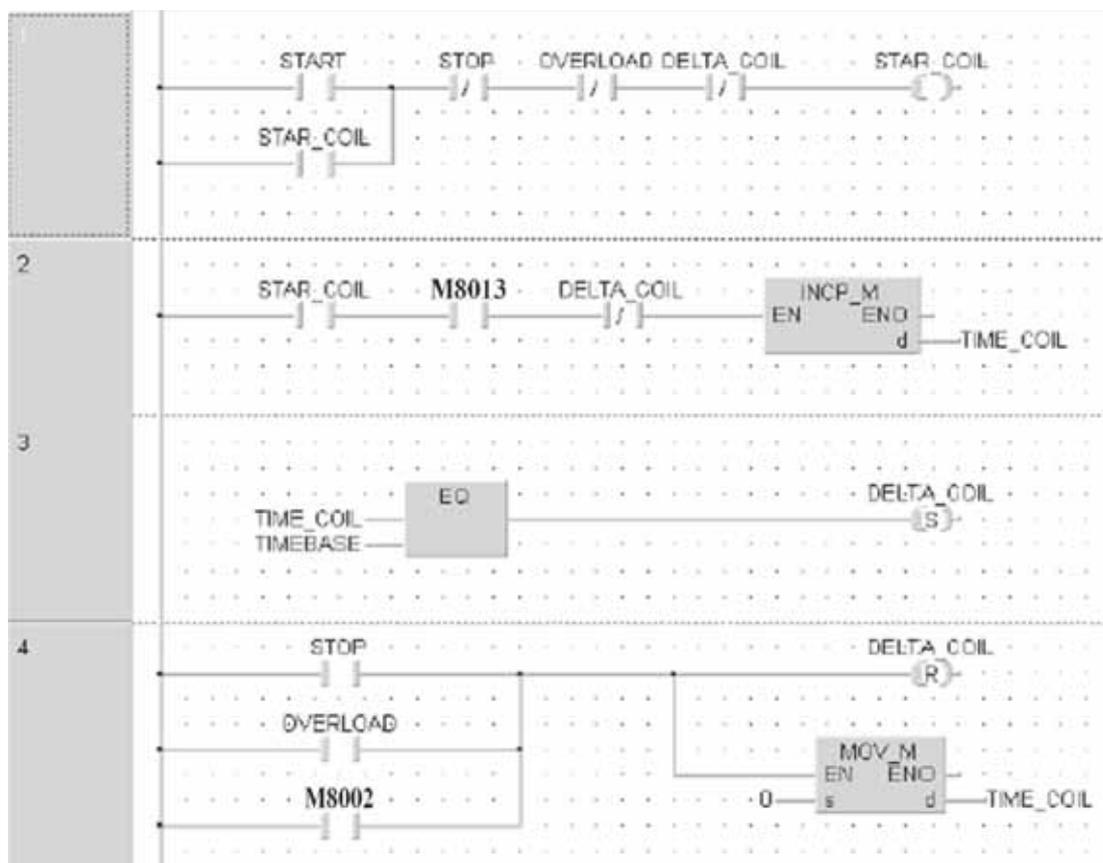
Это означает, что для одного объявления можно использовать многие производные. Один пример конкретного использования этой процедуры - интерфейс к группам тегов в системах SCADA. Она позволяет поддерживать быстрые и эффективные коммуникационные циклы благодаря использованию более коротких и последовательных сеансов передачи данных, вместо многократных запросов на фрагментированные данные к и от ПЛК.

11.1 Пример использования DUT

Следующий пример иллюстрирует использование DUT.

- ① Создайте новый проект "Motor Control DUT":
- ② Создайте программный модуль новой программы MOTOR_CONTROL
- ③ Создайте в задачном пуле новую задачу с именем MAIN и свяжите с ней программу MOTOR_CONTROL.
- ④ Создайте новый функциональный блок "STAR_DELTA" и повторно введите следующий код программы. Альтернативно, скопируйте-вставьте исходный функциональный блок, "тело и заголовок", из проекта "Motor Control" следующим образом:

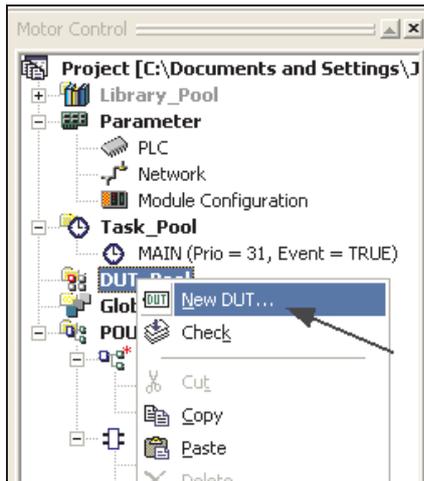
Тело: STAR_DELTA



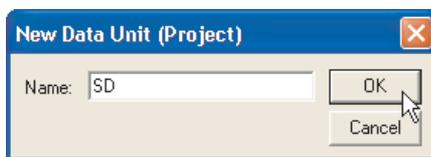
Заголовок: STAR_DELTA

	Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	START	BOOL	... FALSE	
1	VAR_INPUT	STOP	BOOL	... FALSE	
2	VAR_INPUT	OVERLOAD	BOOL	... FALSE	
3	VAR_INPUT	TIMEBASE	INT	... 0	
4	VAR_OUTPUT	DELTA_COIL	BOOL	... FALSE	
5	VAR_OUTPUT	STAR_COIL	BOOL	... FALSE	
6	VAR_OUTPUT	TIME_COIL	INT	... 0	

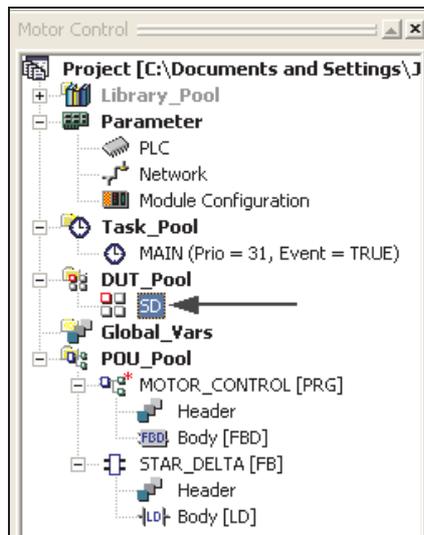
Заголовок содержит определения (маску) типов данных, которые будут использоваться при создании DUT "SD".



⑤ Создайте новый DUT, щелкнув правой кнопкой мыши на значке **DUT Pool** в навигационном окне программы или используя значок DUT  на панели инструментов.



⑥ По запросу введите имя нового DUT как SD.



Теперь новый DUT будет показан под **DUT Pool** в проекте.

⑦ Откройте DUT, щелкнув на значке, и будет показано следующее:

	Identifier	Type	Initial	Comment
0			...	

⑧ Введите следующие данные в DUT "SD".

	Identifier	Type	Initial	Comment
0	DELTA	BOOL	... FALSE	
1	O_L	BOOL	... FALSE	
2	STAR	BOOL	... FALSE	
3	START	BOOL	... FALSE	
4	STOP	BOOL	... FALSE	
5	TB	INT	... 0	
6	TV	INT	... 0	

- ⑨ Закройте DUT и сохраните программу.
- ⑩ Откройте GVL и создайте 2 новых элемента STAR_DELTA1 и STAR_DELTA2.
- ⑪ Щелкните на "многоточии" , чтобы определить **Type** как "Data Unit Types" **SD** для обоих элементов:

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
- 0	VAR_GLOBAL	STAR_DELTA1			SD	...
- 1	VAR_GLOBAL	STAR_DELTA2			SD	...

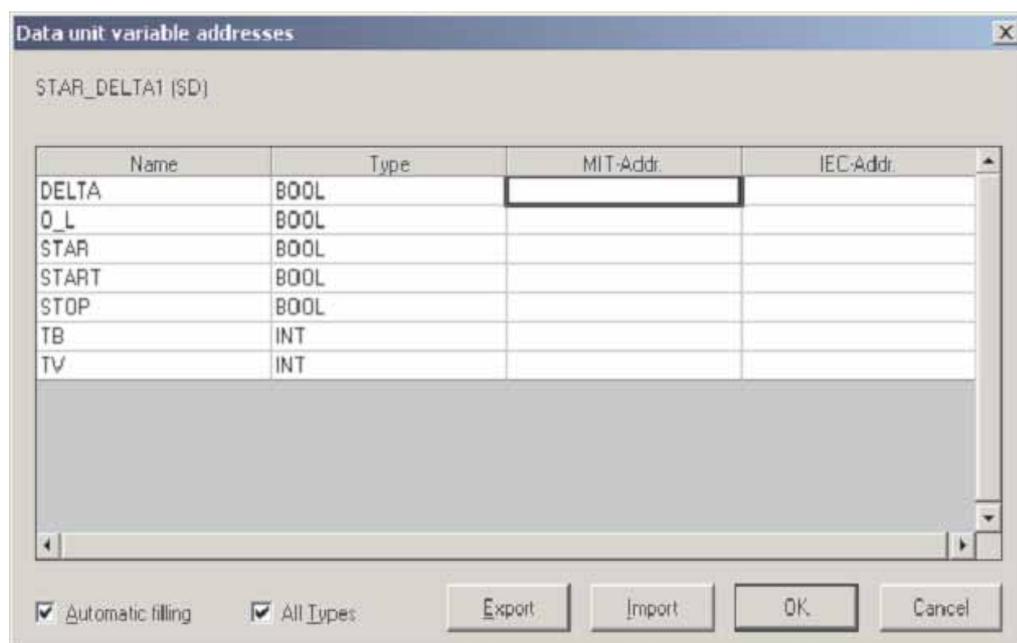


- ⑫ Затем щелкните на ячейке **MIT-Addr.** для STAR_DELTA1, чтобы ввести данные переменных для выбранного элемента DUT:

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
- 0	VAR_GLOBAL	STAR_DELTA1			SD	...
- 1	VAR_GLOBAL	STAR_DELTA2			SD	...

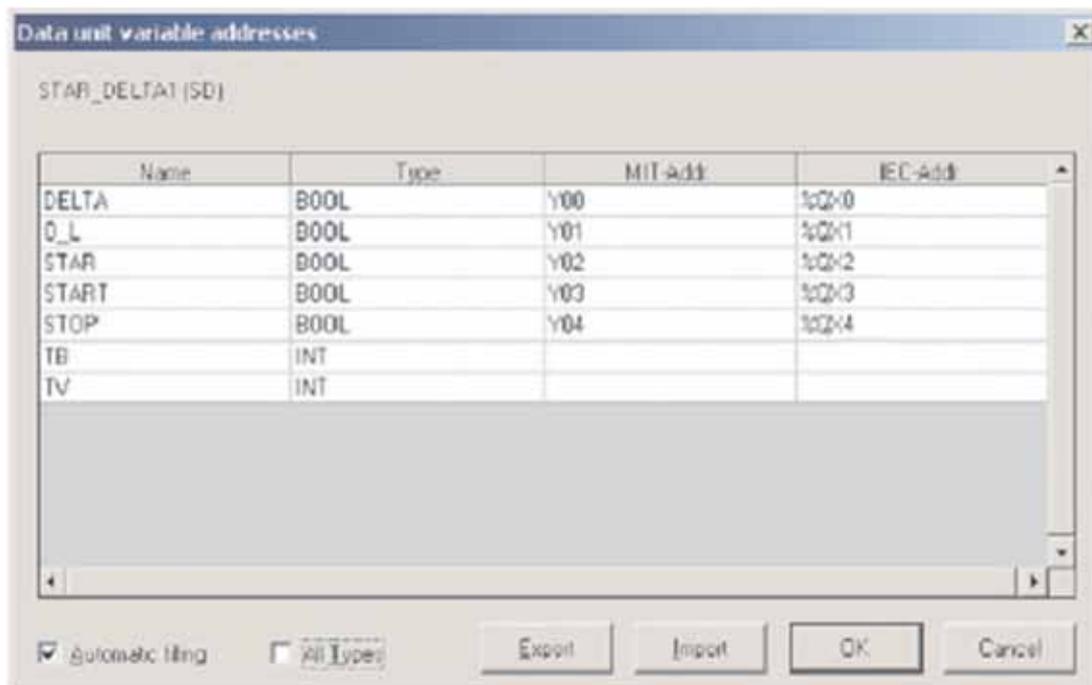
Щелкните, чтобы выбрать

В результате получим окно:



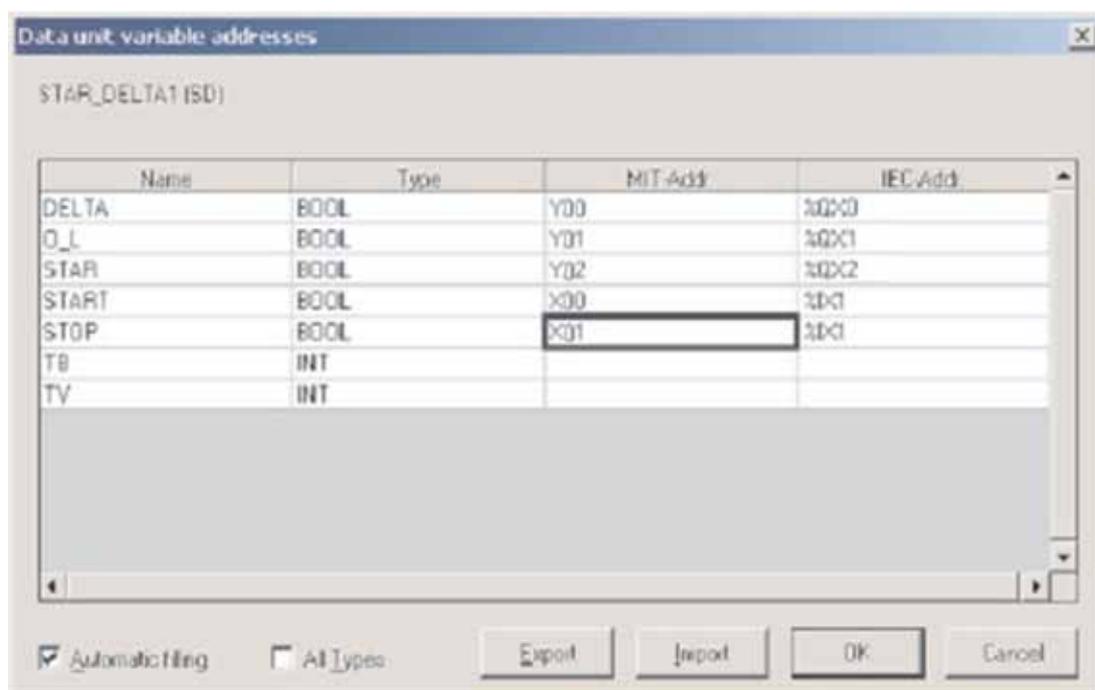
11.2 Автоматическое заполнение, переменные

- ① При использовании смешанных типов переменных не разрешается снимать выделение с опции **All types** в этой операции.
- ② Введите Y00 в позицию **MIT-Addr.** для переменной: "DELTA":

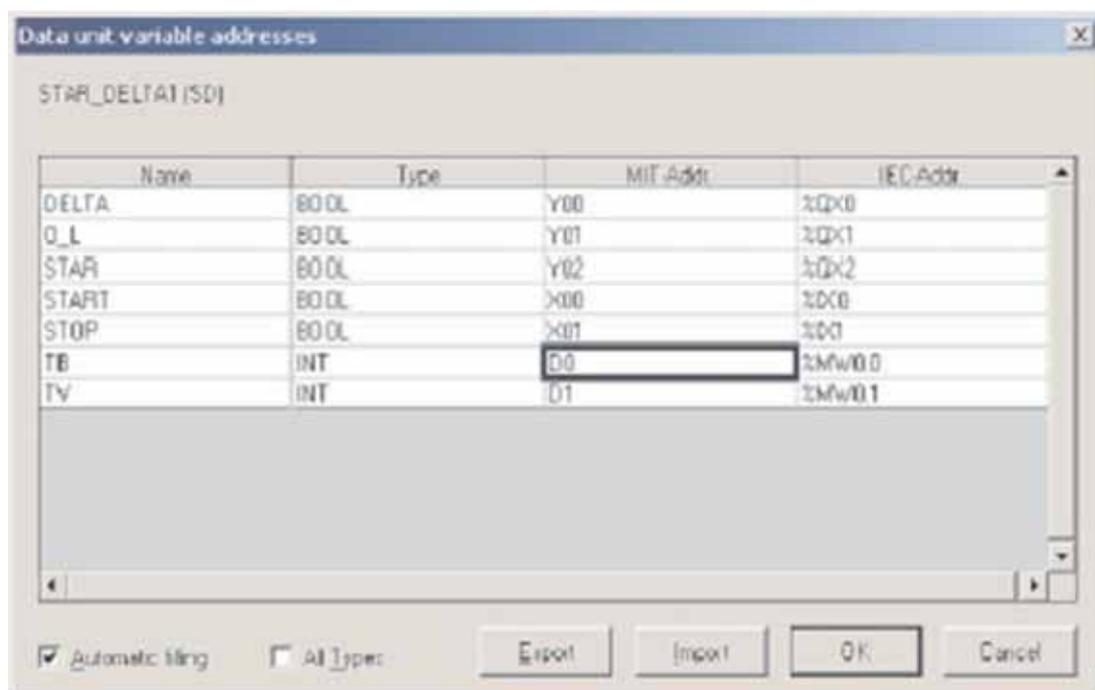


Система попытается последовательно "автозаполнять" переменные типа BOOL. Хотя этот подход рекомендуется во многих ситуациях, в данном случае он лишь частично успешен.

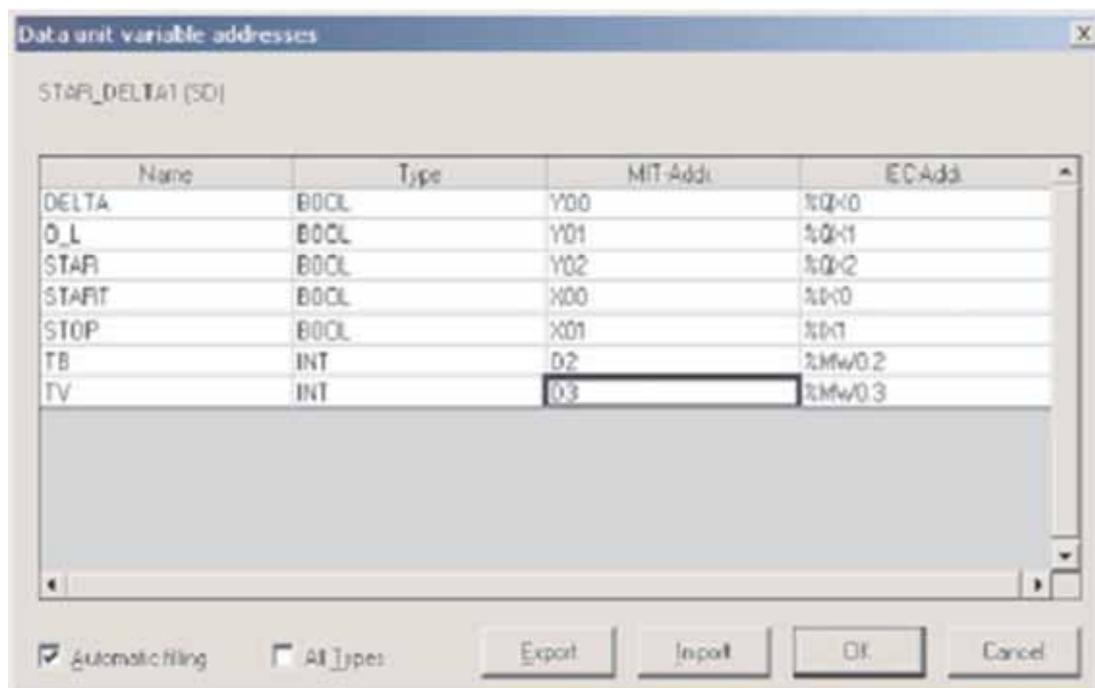
- ③ Поэтому перезапишите для переменных "START и STOP" X00 и X01 следующим образом:



- ④ Наконец, введите две остающиеся целочисленные переменные "TB" и "TV", используя адреса MELSEC D0 и D1 и опцию **Auto Fill**:



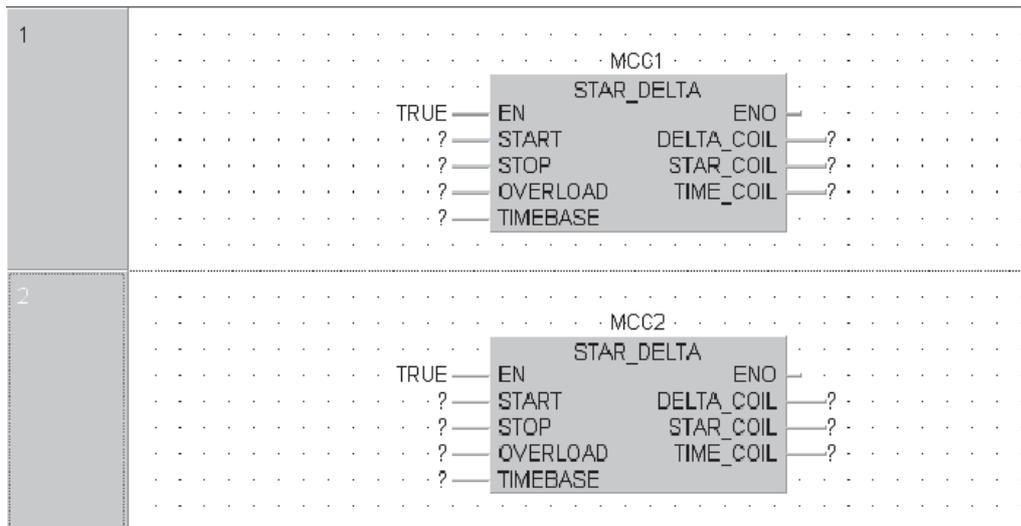
- ⑤ Щелкните на **OK** чтобы сохранить текущую конфигурацию.
- ⑥ Повторите эту серию операций для "STAR_DELTA2" введя следующий последовательный адрес заголовка для каждой переменной **type**:



- ⑦ Проверьте GVL, он должен иметь следующий вид:

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
+0	VAR_GLOBAL	STAR_DELTA1	DELTA:	DELTA:	SD	...
+1	VAR_GLOBAL	STAR_DELTA2	DELTA:	DELTA:	SD	...

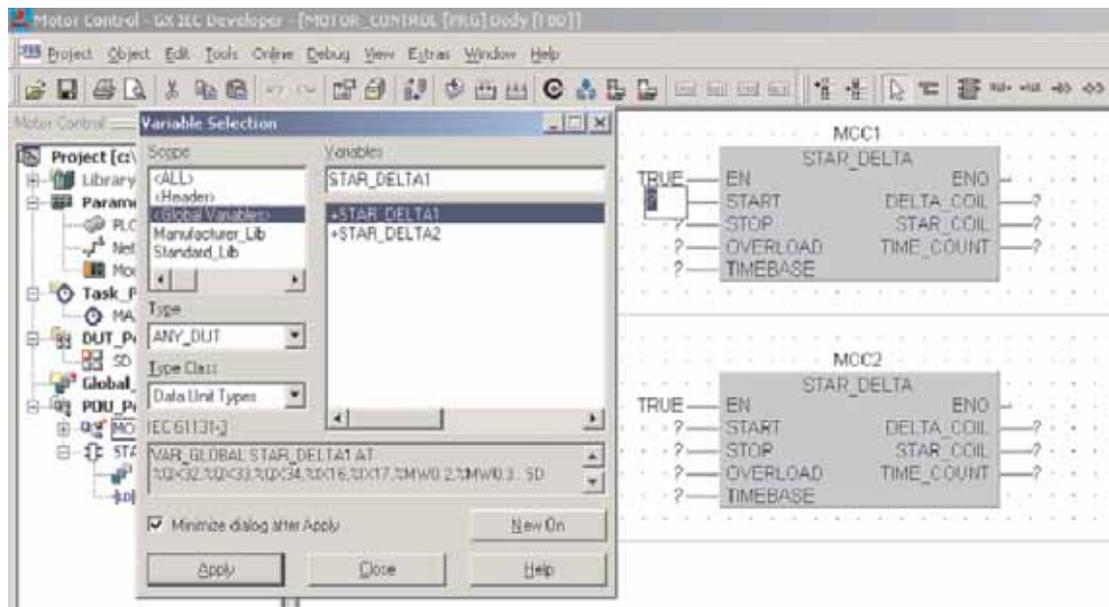
Откройте программный модуль программы MOTOR_CONTROL и поместите 2 экземпляра пользовательского функционального блока STAR_DELTA, как показано:



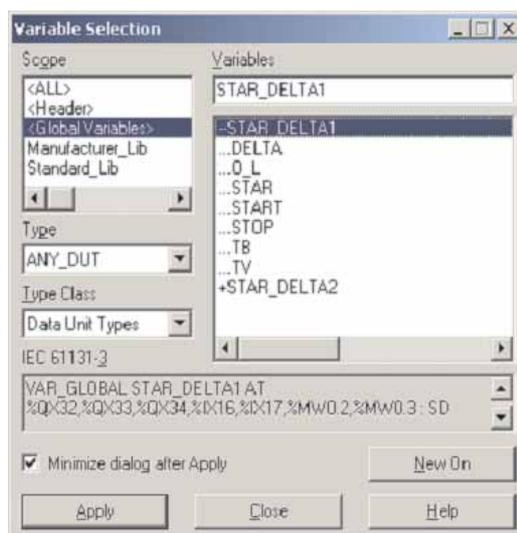
11.3 Назначение переменных DUT функциональным блоком

Чтобы назначить переменные функциональным блоком...

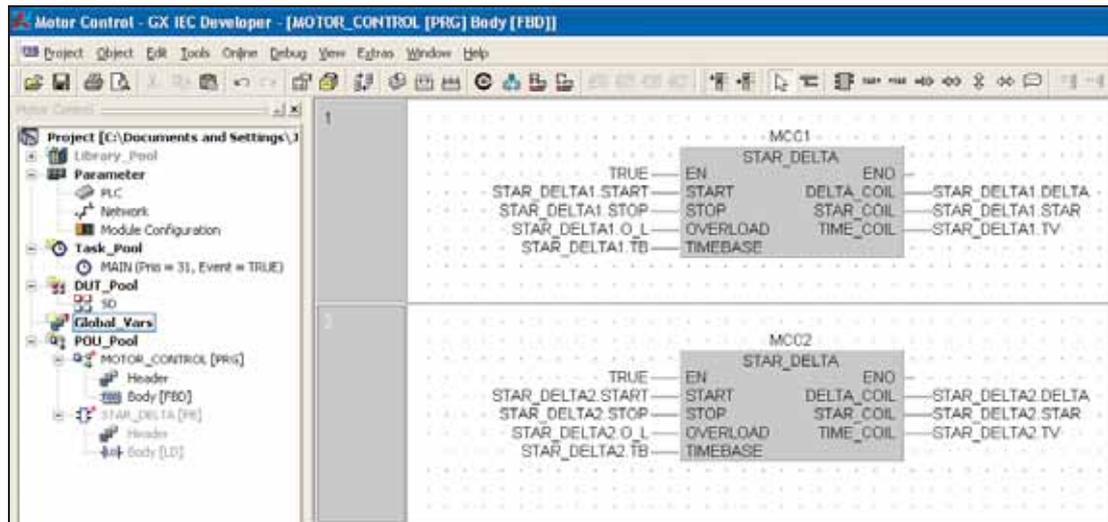
- 1 ...щелкните правой кнопкой на переменной (или F2). Появляется следующее окно выбора переменных:



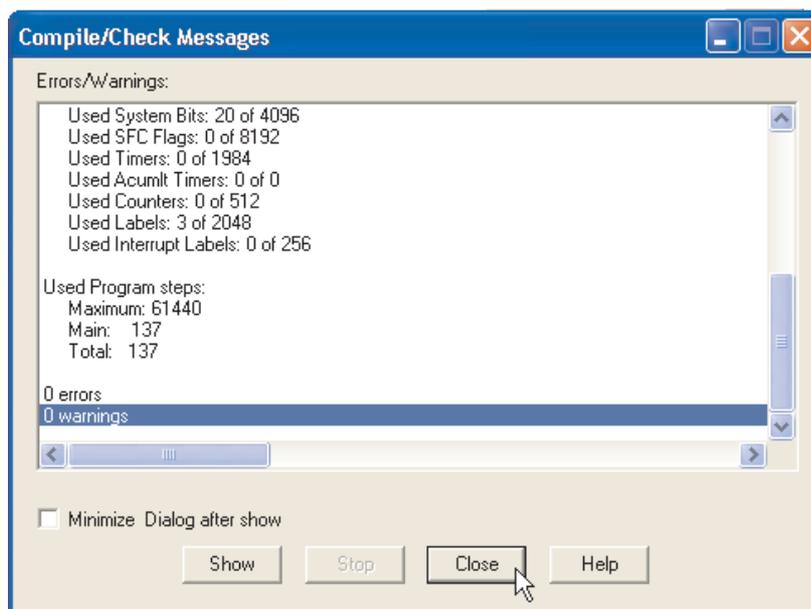
- 2 Установите для **Scope** значение **Header**, для **Type Class** значение **Data Unit Types** и для **Type** значение ANY_DUT.
- 3 Дважды щелкните на +STAR_DELTA1, появляется следующий расширенный список переменных DUT:



- ④ Выберите и присвойте переменные двум функциональным блокам STAR_DELTA на программном модуле программы MOTOR_CONTROL, как показано:

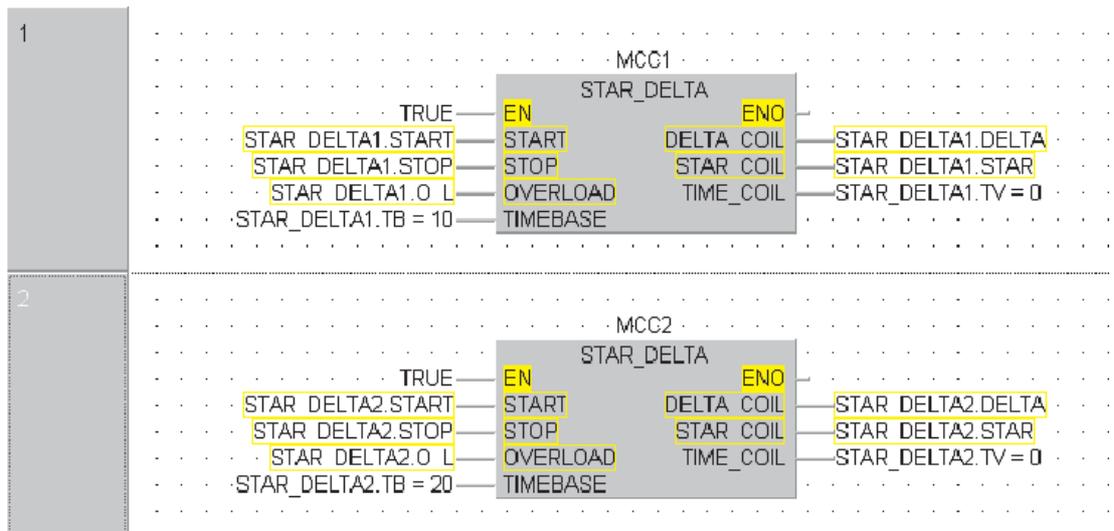


Сохраните проект и выполните **Rebuild All**, чтобы компилировать код:



Загрузите и проверьте проект. Перед тем как функциональные блоки смогут работать, необходимо записать значения во входы TIMEBASE: STAR_DELTA1.TB и STAR_DELTA2.TB. Это выполняется с использованием метода модификации переменных в режиме онлайн, описанного в предыдущем разделе.

Моделируйте работу обоих функциональных блоков, как показано на следующей странице, чтобы подтвердить, что все работает так, как ожидалось:



12 Массивы

12.1 Обзор

Массив – это область или матрица переменных конкретного типа.

Например, **ARRAY [0..2] OF INT** – это одномерный массив трех целочисленных элементов (0,1,2). Если начальный адрес массива - D0, то массив состоит из D0, D1 и D2.

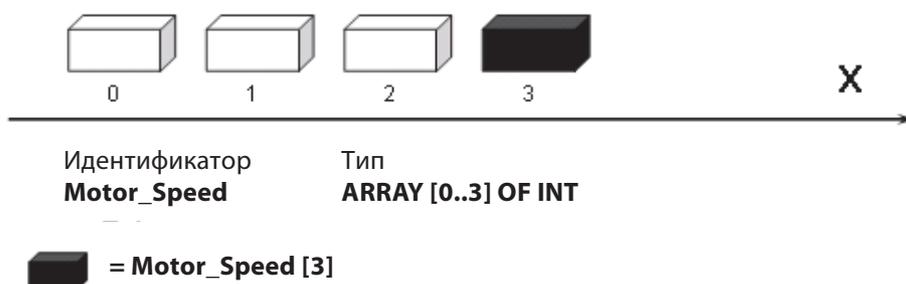
В программе элементы программы могут использовать: Motor_Volts[1] и Motor_Volts[2] как объявления; в данном примере это означает, что адресуются D1 и D2 .

Массивы могут иметь до трех размерностей, например: ARRAY [0..2, 0..4] содержит три элемента в первой размерности и пять во второй.

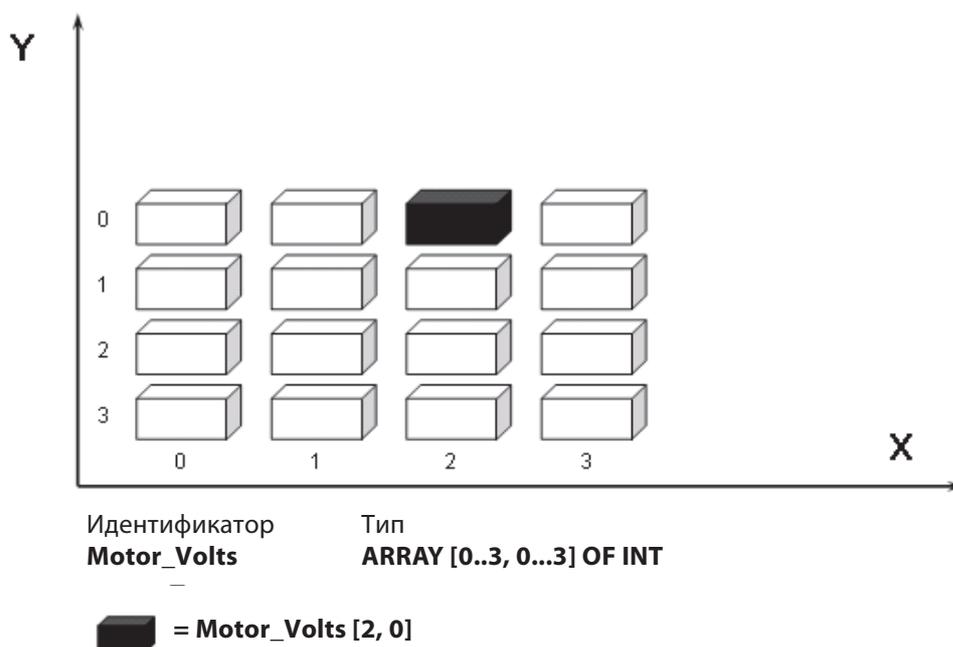
Массивы могут обеспечить удобный способ "индексации" имен тегов, т.е. одно объявление в таблице локальных или глобальных переменных может иметь доступ ко многим элементам.

Следующие диаграммы иллюстрируют графическое представление трех типов массивов.

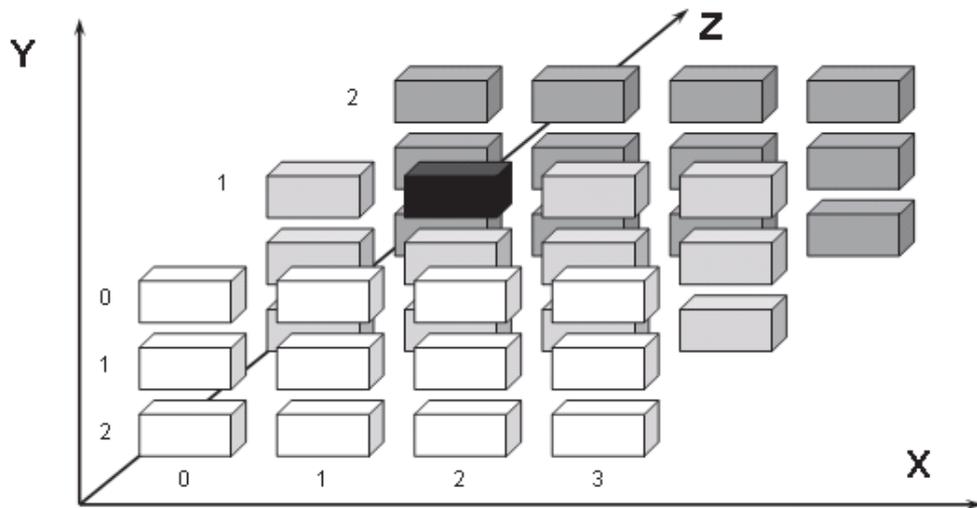
Одномерный массив



Двумерный массив



Трёхмерный массив



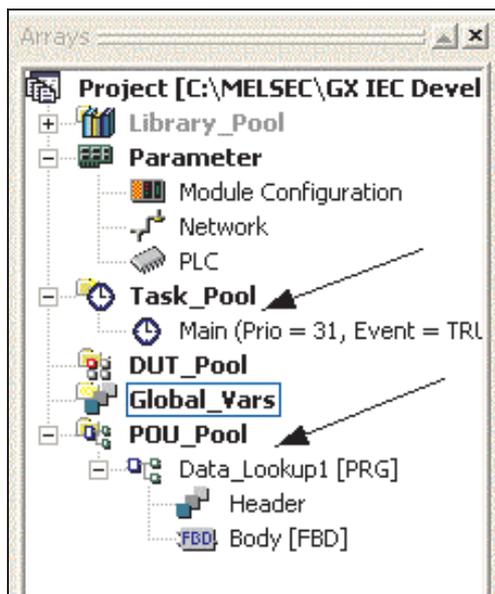
Идентификатор Тип
Motor_Current ; **ARRAY [0..3, 0..2, 0..2] OF INT**

 = **Motor_Volts [1, 0, 1]**

12.2 Пример массива: Одномерный массив

Следующий пример используется для иллюстрации одномерного массива. Длина массива равна 10 слов; используются глобальные адреса MELSEC D100-D109. В данном примере используются только стандартные IEC операторы, функции и функциональные блоки.

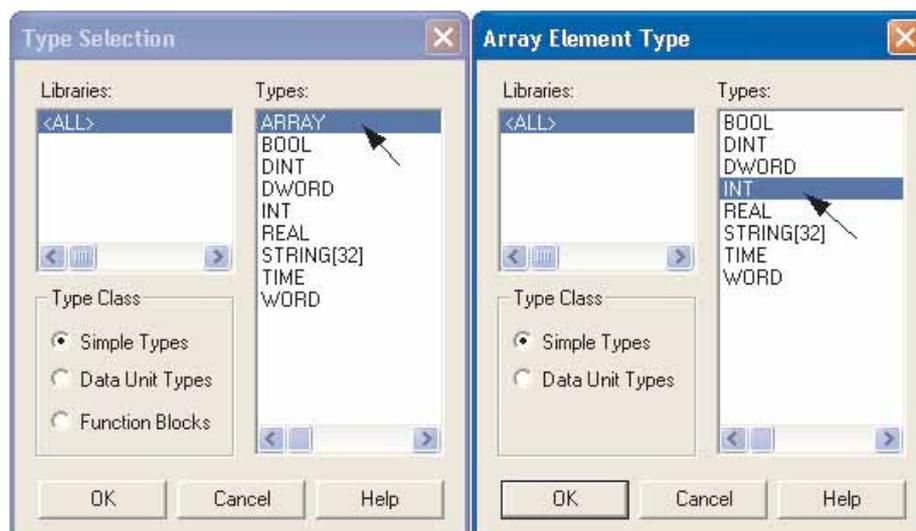
- ① Создайте новый проект и определите один новый программный модуль класса "Program", используя тело на языке **FBD** под названием "Data_Lookup1"
- ② Создайте в задачном пуле новую задачу с именем "Main" и свяжите с ней программный модуль программы "Data_Lookup1":



- ③ Откройте список глобальных переменных и создайте следующие элементы:

	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	Data_Clock	X0	%IX0	BOOL	FALSE
1	VAR_GLOBAL	Data_Store	D100	%MWD.100	ARRAY [0..9] OF INT	[10(0)]
2	VAR_GLOBAL	Data_Lookup	D10	%MWD.10	INT	0
3	VAR_GLOBAL	Data_Pointer	D11	%MWD.11	INT	0

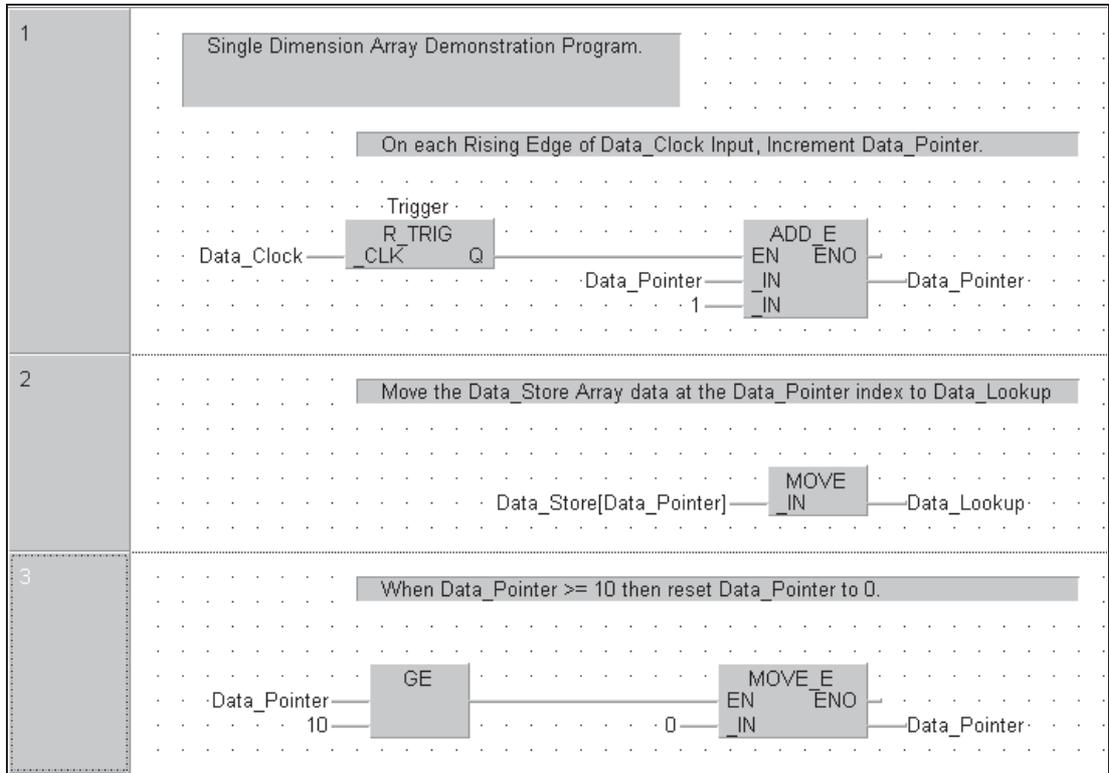
Тип переменной "Array" вводится следующим образом:



Учтите, что когда элемент массива появляется в первый раз, ему будет присвоено значение по умолчанию ARRAY [0..3] OF INT. Для этого примера необходимо изменить его размер на [0..9] INT, как показано ниже:

```
1 VAR_GLOBAL          Data_Store  |D100  |%M%WD.100|ARRAY [0..9] OF INT  |(10(0))
```

④ Откройте программный модуль программы "Data_Lookup1" и введите следующую функциональную блок-схему:



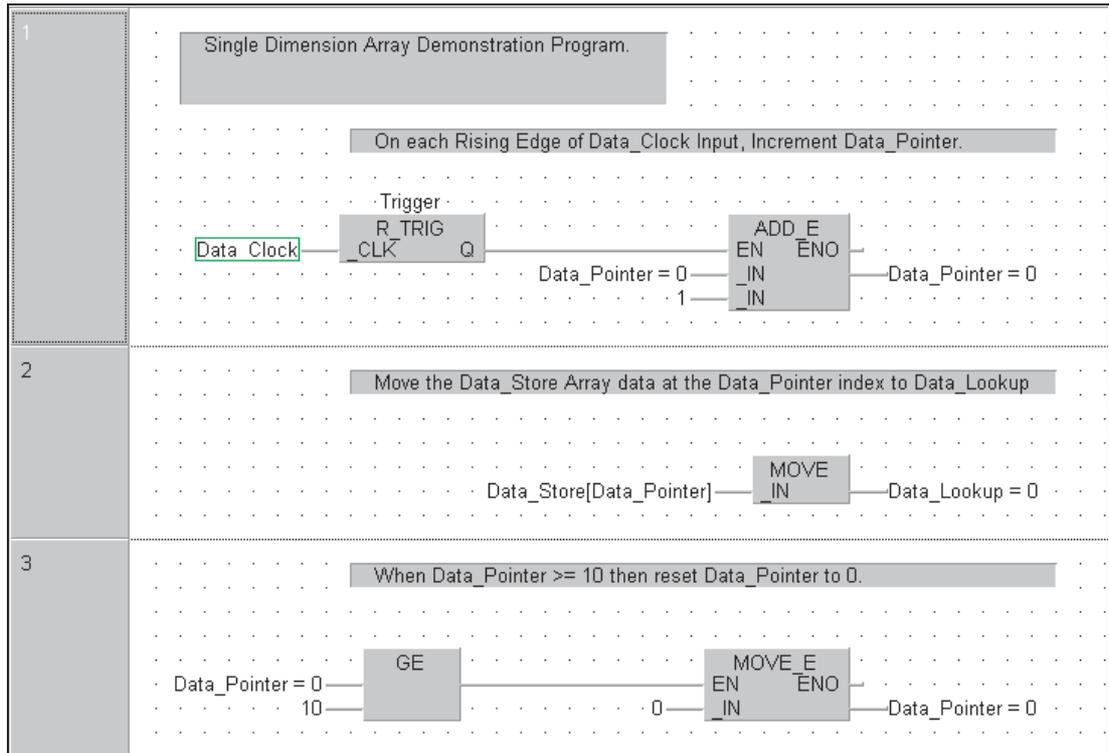
ПРИМЕЧАНИЕ

| Определите функциональный блок "R_TRIG" с именем экземпляра "Trigger".

⑤ Проверьте, что заголовок имеет следующий вид:

	Class	Identifier	Type	Initial	Comment
VAR	Trigger	R_TRIG			

- ⑥ Сохраните программу и используйте **Rebuild All**, чтобы компилировать программу.
- ⑦ Загрузите программу в ПЛК.
- ⑧ Проверьте тело программного модуля (см. следующую страницу)



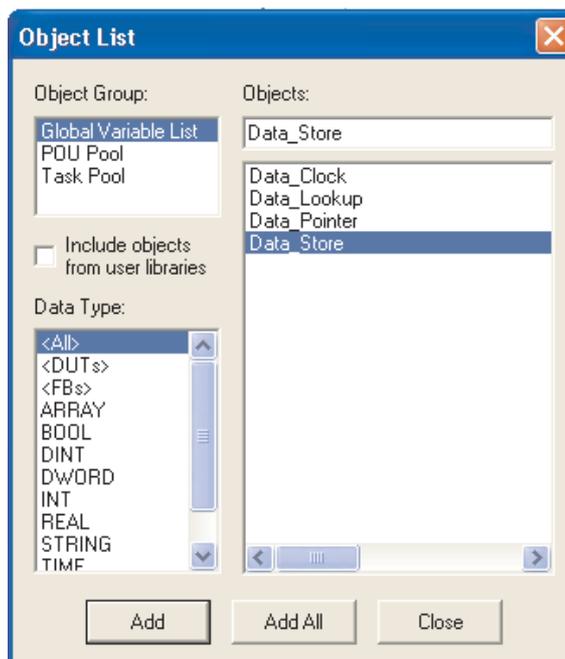
Перед тем как программа сможет работать надлежащим образом, необходимо ввести данные в физические адреса MELSEC, занятые переменными массива. Это можно сделать двумя способами:

- Используйте функцию **Device Edit** из меню **Debug**, как описано выше, применяя **Insert Devices** в диапазоне от D100 до D109, и введите любые 10 случайных целочисленных значений в диапазоне от -32768 до +32767 и запишите их в ПЛК.
- Откройте функцию **Entry Data Monitor** из меню **Online**.
 - Щелкните правой кнопкой на заголовках столбцов "Address" или "Name" и выберите Insert Objects из списка меню, как показано:

Pos	Address (MIT)	Name	Value (dec)
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			

Insert Objects...	F2
Next Object	F3
Insert Forced Inputs	
Insert Set Inputs	
Insert Set Outputs	
Clear Device File	
Insert Row	Ins
Delete	Del
Delete All	
Read from PLC	
Write to PLC...	
Read from File...	
Write to File...	
Setup...	
Always on top	

- Из результирующего окна выберите имя переменной "Data_Store" и щелкните на **Add**:



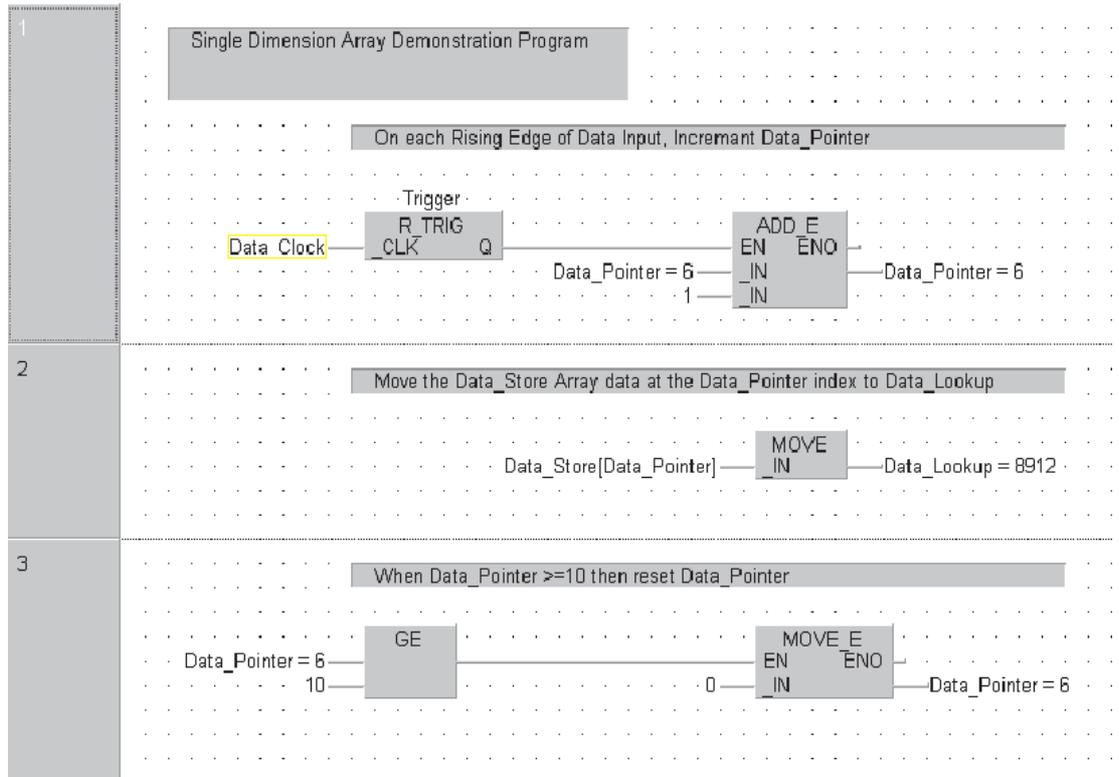
- Поскольку имя переменной "Data_Store" является массивом, система представляет элемент с префиксом "+". Щелчок на имени переменной расширяет детали массива в таблицу, как показано:

Pos	Address (MIT)	Name	Value (dec)
1		-Data Store	
2	D100	[0]	0
3	D101	[1]	0
4	D102	[2]	0
5	D103	[3]	0
6	D104	[4]	0
7	D105	[5]	0
8	D106	[6]	0
9	D107	[7]	0
10	D108	[8]	0
11	D109	[9]	0

- Щелчок на префиксе "-" убирает детали массива.
- При мониторинге значений переменных введите любые 10 случайных целочисленных значений в диапазоне от -32768 до +32767, как показано ниже:

Pos	Address (MIT)	Name	Value (dec)
1		-Data_Store	
2	D100	[0]	1234
3	D101	[1]	4321
4	D102	[2]	7654
5	D103	[3]	4236
6	D104	[4]	17
7	D105	[5]	32766
8	D106	[6]	8912
9	D107	[7]	43
10	D108	[8]	186
11	D109	[9]	9999

- ⑨ Вернитесь к мониторингу тела программного модуля "Data_Lookup1" и наблюдайте работу программы, замечая, как изменяется значение выходной переменной "Data_Lookup" по мере увеличения указателя данных:



Программа сбрасывает указатель к нулю на 10 элементе, тем самым повторяя цикл таблицы с инкрементом (Индекс 0-9).

13 Работа с библиотеками

13.1 Пользовательские библиотеки

Все созданные до сих пор функции и функциональные блоки оставались в текущем проекте и были доступны только для этого проекта.

Пользовательские библиотеки позволяют создавать библиотеки, содержащие разработанные пользователем программные модули, функции, функциональные блоки и т.д. Эти библиотеки имеются глобально, т.е. к ним могут обращаться другие проекты.

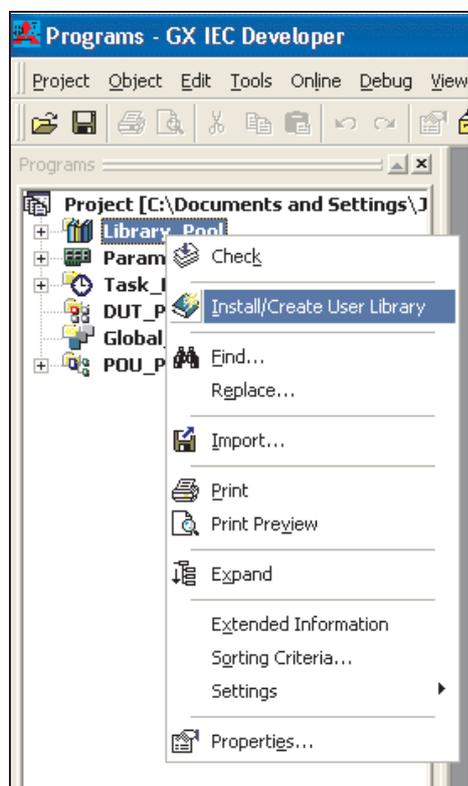
Поэтому инженеры, работающие с отдельными проектами, могут иметь доступ к общим библиотекам частей стандартных схем.

Как уже отмечалось, при вызове функций программы, **Стандартная библиотека** содержит функции IEC. **Библиотека изготовителя** содержит функции Mitsubishi functions (они обозначены как *_M), где M означает изготовитель (manufacturer), а не Mitsubishi!

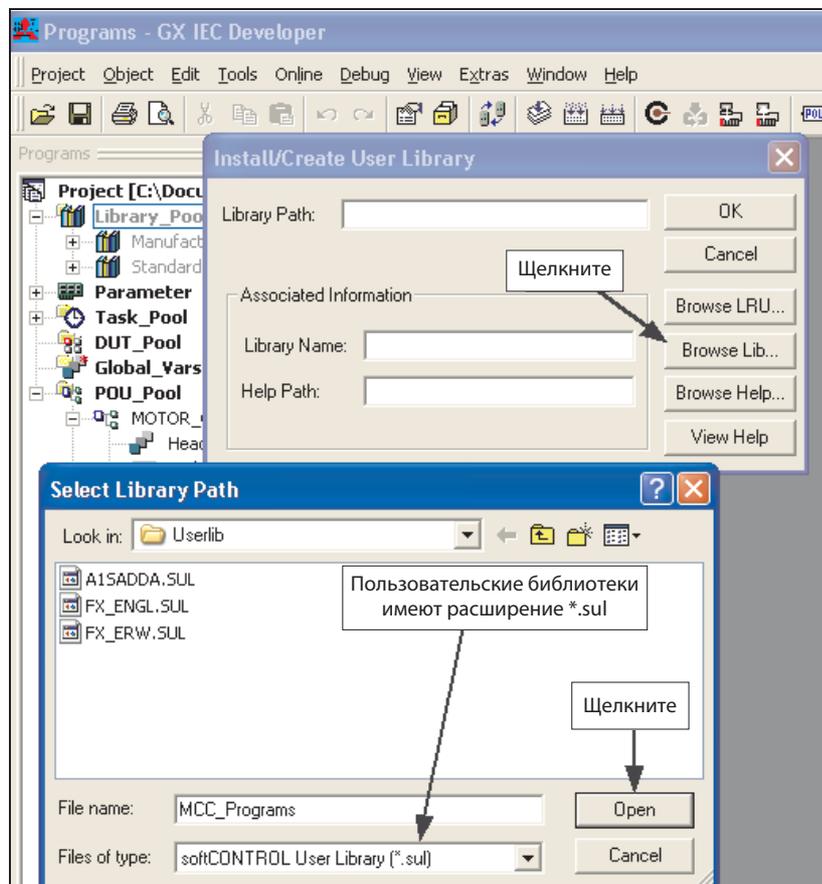
Любые пользовательские библиотеки также появятся в этом списке.

13.1.1 Пример - Создание новой библиотеки

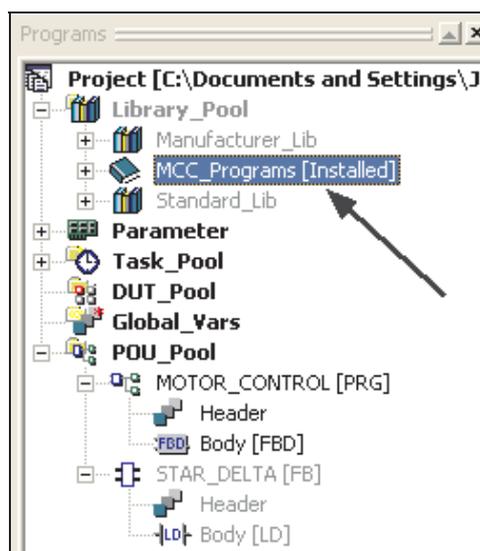
- 1 Присвойте функциональный блок STAR_DELTA новой библиотеке.
- 2 Щелкните правой кнопкой на пуле библиотек в окне **Project Navigator** и из показанного меню выберите **User Library** и **Install/Create Library**.



- ③ Щелкните на **Browse Lib** и введите имя файла "MCC_Programs" в показанном ниже окне. При необходимости путь к каталогу можно изменить. В данном случае предположим, что используется заданный по умолчанию путь. Это "C:\MELSEC\GX IEC DEVELOPER 7.00\Userlib".



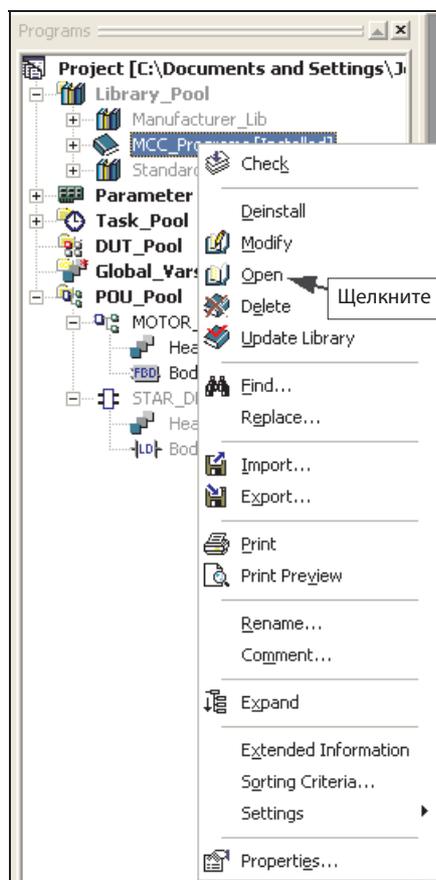
- ④ По завершению щелкните на **Open**:



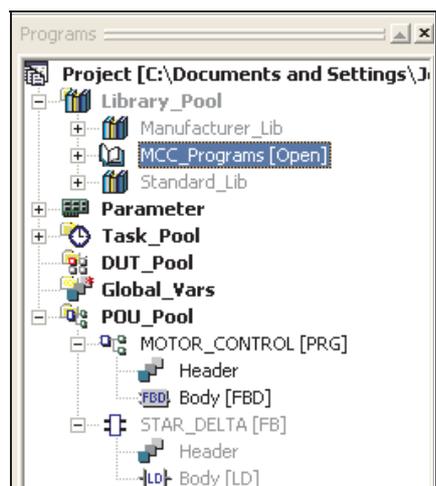
Обратите внимание, что теперь в пуле библиотек проекта представлена новая библиотека "MCC_Programs".

13.1.2 Открытие библиотеки

- 1 Откройте библиотеку, щелкнув правой кнопкой мыши на значке "MCC_Programs" и щелкните на **Open** из меню:



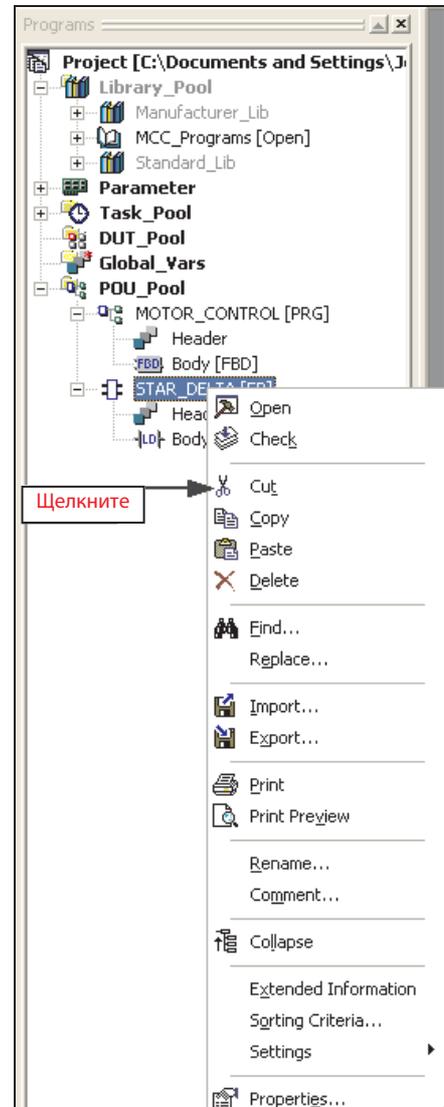
Библиотека теперь открыта; к ней можно обращаться и редактировать:



13.1.3 Перемещение программного модуля "Функциональный блок" в открытую библиотеку

Сейчас функциональный блок STAR_DELTA будет перемещен в библиотеку "MCC_Programs".

- Щелкните правой кнопкой на значке STAR_DELTA в навигационном окне проекта и щелкните на **Cut**:

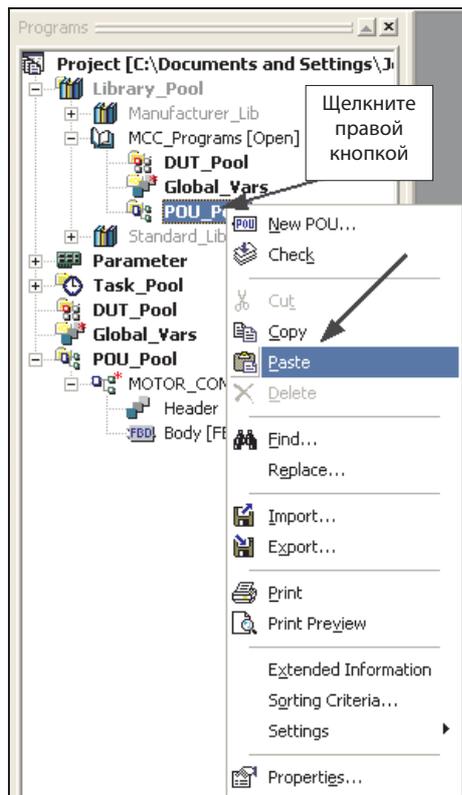


Появится следующее диалоговое окно:

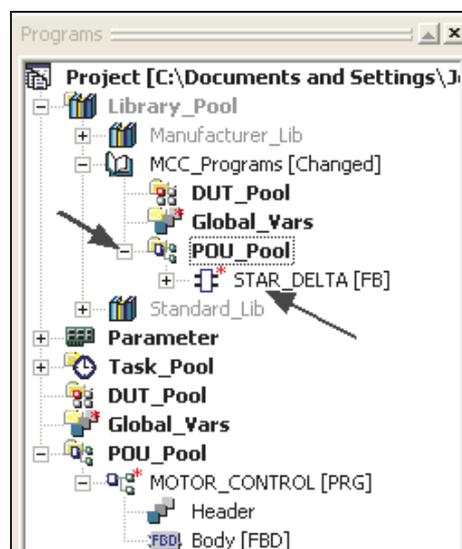


- Выберите **Yes**

- ③ Щелкните правой кнопкой на значке пользовательской библиотеки и выберите **Paste** из меню:



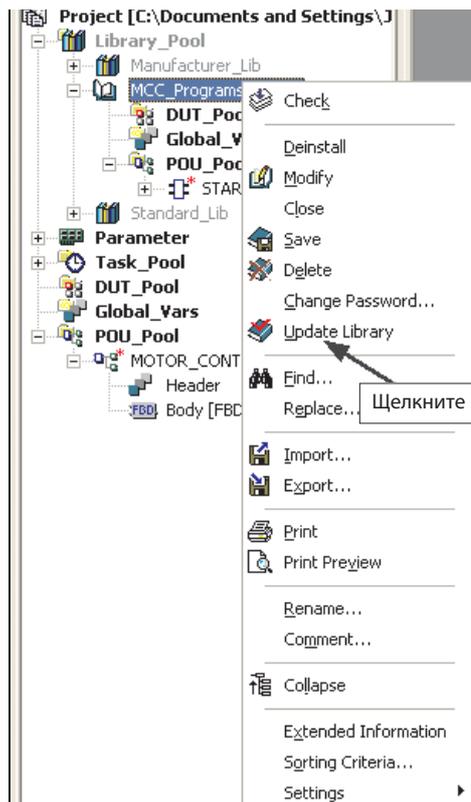
- ④ Щелкните на "+" на новом элементе в библиотечном пуле программных модулей, чтобы расширить функциональный блок "STAR_DELTA":



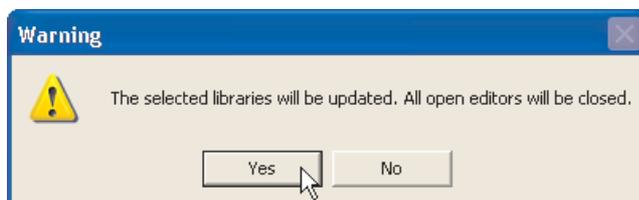
Сейчас программный модуль функционального блока "STAR_DELTA" представлен в библиотеке "MCC_Programs", а не в пуле программных модулей проекта.

Любой программный модуль, функция, функциональный блок, PRG или DUT можно подключить к библиотеке подобным образом.

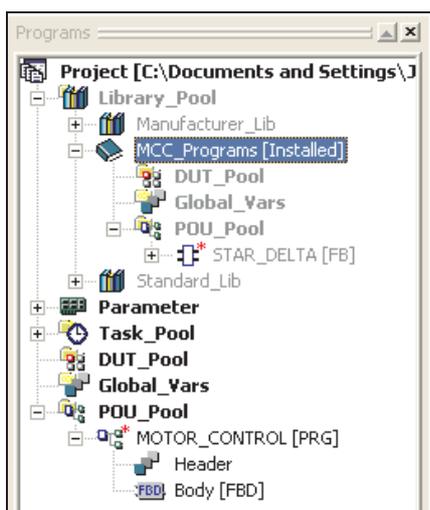
- ⑤ Завершив редактирование библиотеки, щелкните на **Update Library**. Это обновит и закроет библиотеку.



Будет показано следующее сообщение:



- ⑥ Щелкните на **Yes**, и библиотека будет обновлена, сохранена и закрыта.

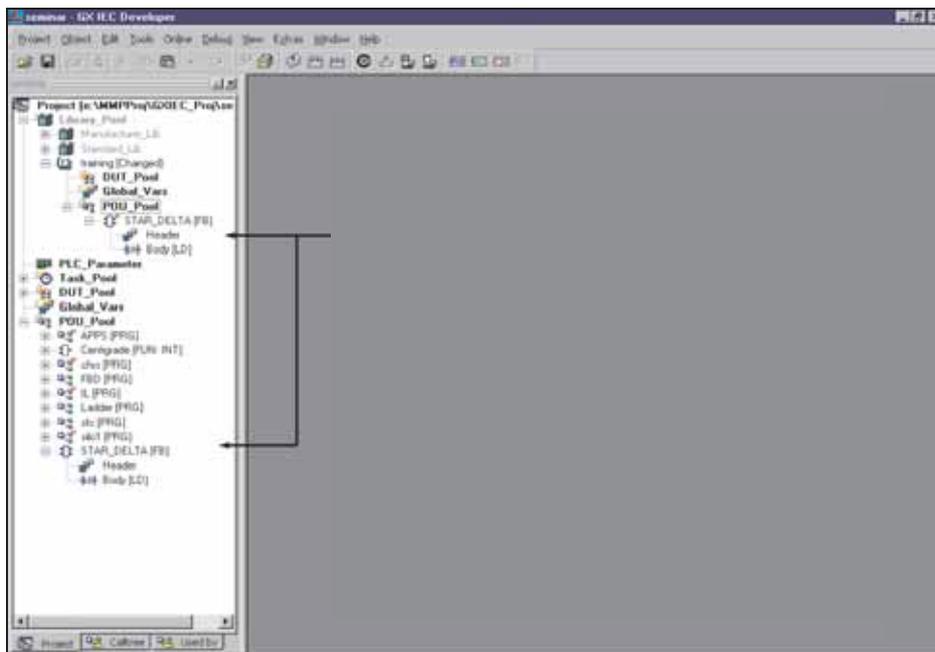


Теперь библиотека сохранена в каталоге по умолчанию "C:\MELSEC\GX IEC Developer 7.00\Userlib", как установлено при создании библиотеки.

13.2 Специальное замечание о библиотеках

ПРИМЕЧАНИЕ: Если библиотека создана в подкаталоге проекта, например E:\MMProj\GXIEC_Proj\Seminar.su1, то элементы библиотеки не могут существовать также в пуле программных модулей проекта: компилятор выдаст ошибку "Doubled in List" (Дважды в списке), поэтому элементы необходимо удалить из пула программных модулей проекта.

Этого НЕ произойдет, если, как обычно, библиотека была генерирована извне проекта, например, из корневого каталога.



13.3 Импорт библиотек в проекты

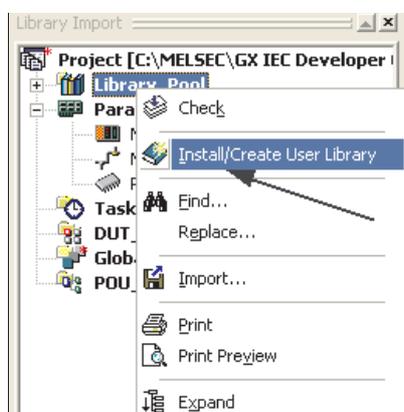
Когда созданы "Пользовательские библиотеки", можно повторно использовать подпрограммы, импортируя их в другие приложения. Компания Mitsubishi Electric выпустила много библиотек с часто используемыми подпрограммами. Например, интерфейсы к "интеллектуальным модулям" типа функциональных блоков А/Ц и Ц/А содержат весь код, чтобы облегчить работу с интерфейсом для этих и многих других модулей. Эти функциональные блоки свободно выложены на многих вебсайтах Mitsubishi, а некоторые представлены на дистрибутивном диске GX IEC Developer.

Следующие два примера описывают методы, которые используются для импорта библиотек в рабочие приложения.

13.3.1 Импорт функционального блока из библиотеки Мицубиси

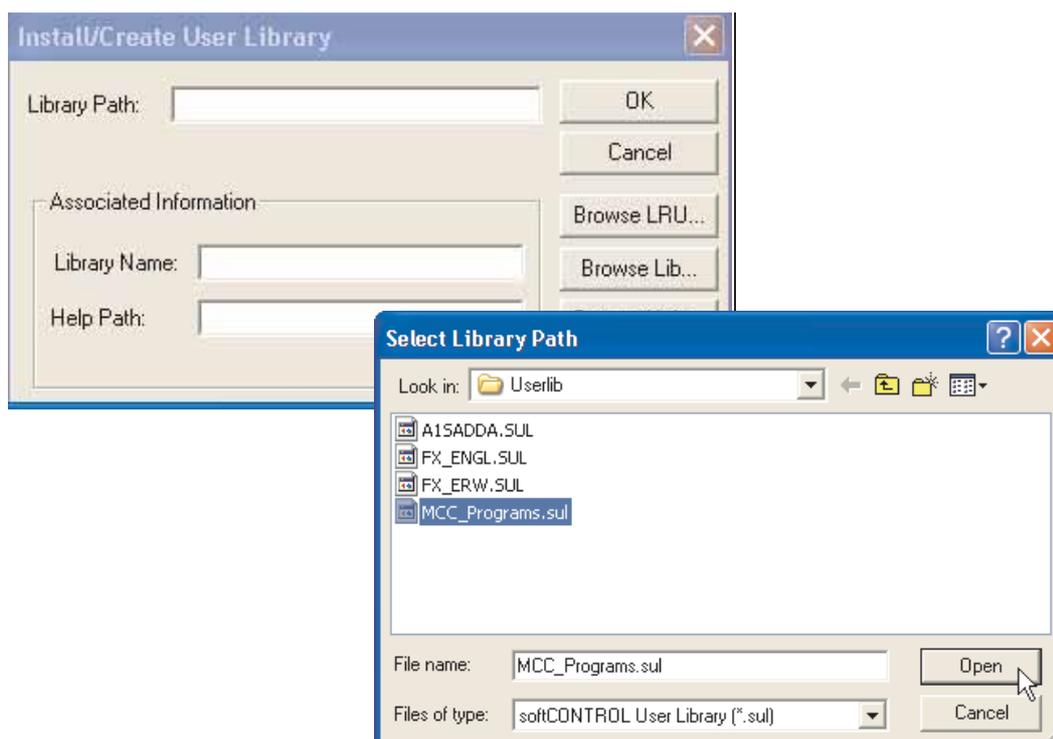
Предварительно сохраненная библиотека "MCC_Programs" будет импортирована в текущий проект, и содержащийся в ней функциональный блок будет использован повторно.

- 1 Создайте новый пустой проект без программных модулей, называемый "**Library Import**".

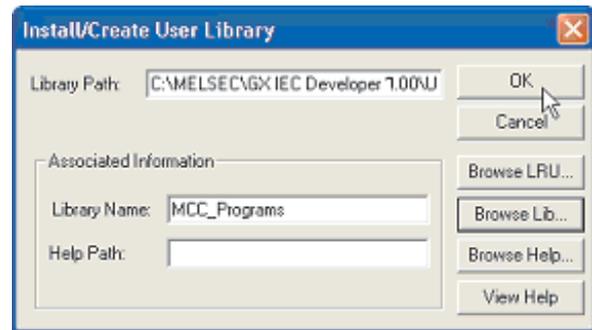


- 2 Щелкните правой кнопкой мыши на пуле библиотек, а затем в меню на **Install/Create User Library**.

- 3 Введите следующие детали в окно запроса:



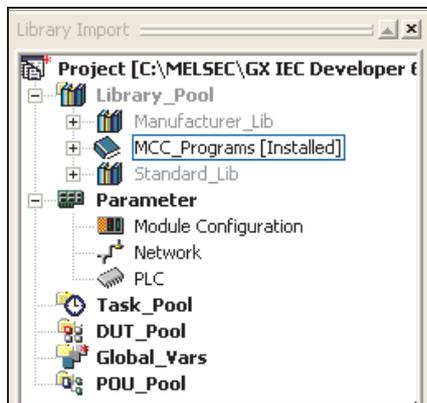
- ④ Затем щелкните на **OK**, чтобы принять элементы.



ПРИМЕЧАНИЕ

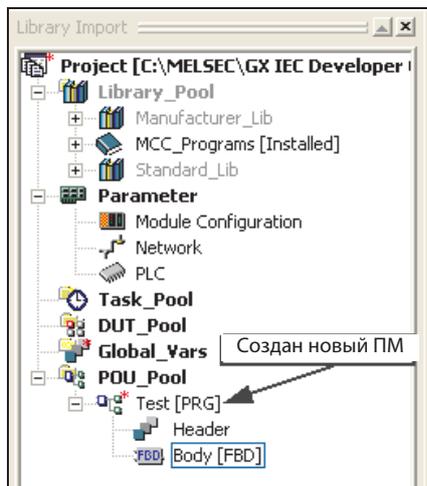
Опция Help Path используется для пользовательских справочных файлов, которые могут быть созданы, чтобы описать работу подпрограмм, содержащихся в библиотеке. Эти файлы можно создавать в MS-Word, например, в формате HTML, и вручную сохранить с расширением *.CHM. Эти файлы можно связать с библиотекой, щелкнув на **Browse Help** таким же образом, как при показанном выше выборе Library Name.

Теперь новая импортированная библиотека установлена в приложение и может использоваться в проекте, как показано ниже:

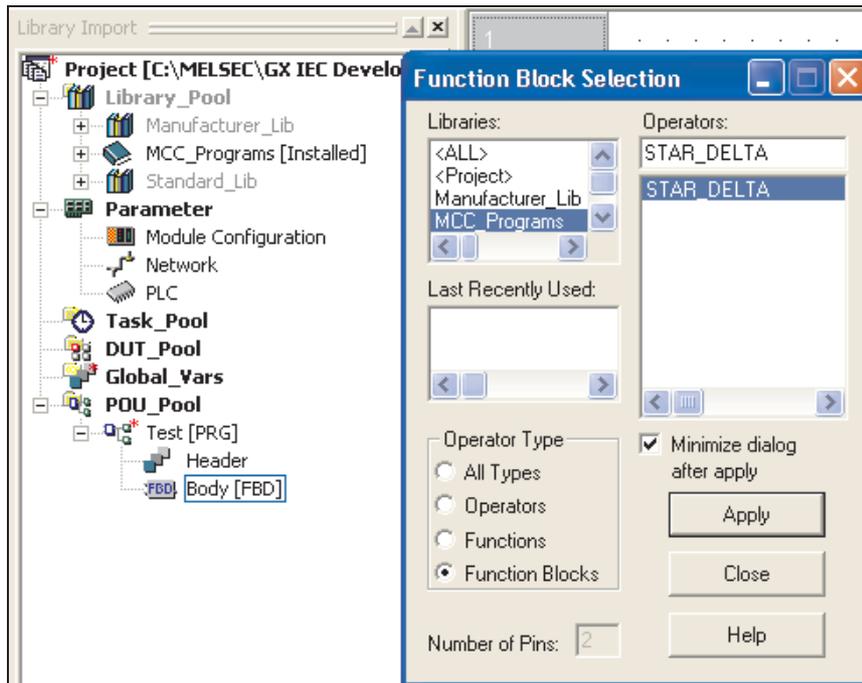


Компоненты, сохраненные в библиотеках, могут легко вызываться и выбираться в проекте, как показано на следующих иллюстрациях:

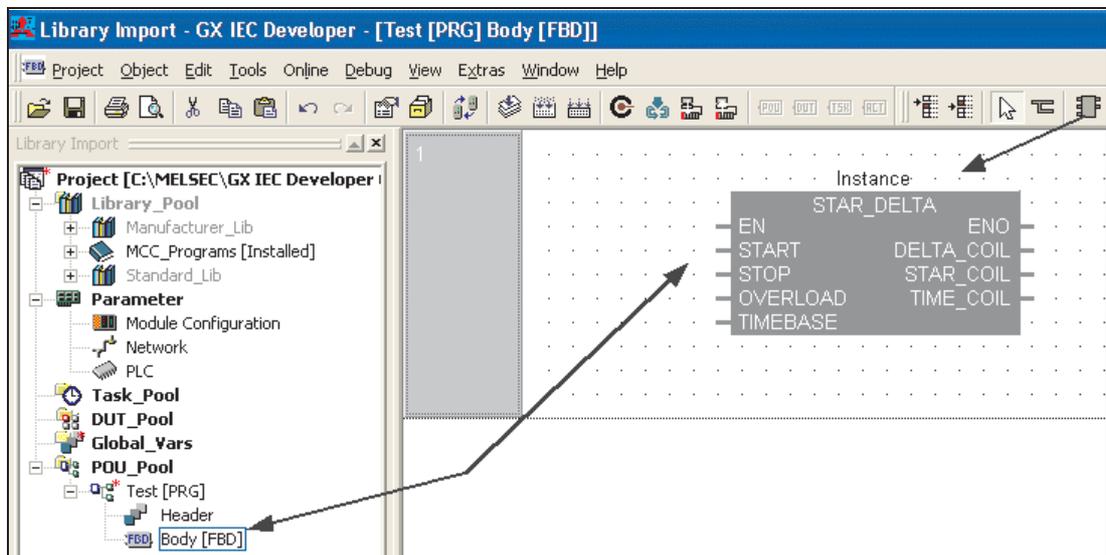
- ① Создайте новый программный модуль, тип **FBD** с названием "Test":



- ② Откройте новый программный модуль и выберите функциональный блок, как показано ниже:



Как можно видеть, в области появляется новая библиотека; ее можно выбрать, как показано ниже:



13.3.2 Пример: Импорт функционального блока из библиотеки Mitsubishi

Следующие иллюстрации демонстрируют процедуры, необходимые, чтобы импортировать функциональный блок Mitsubishi для аналогового ввода с использованием модуля Q64AD серии Q.

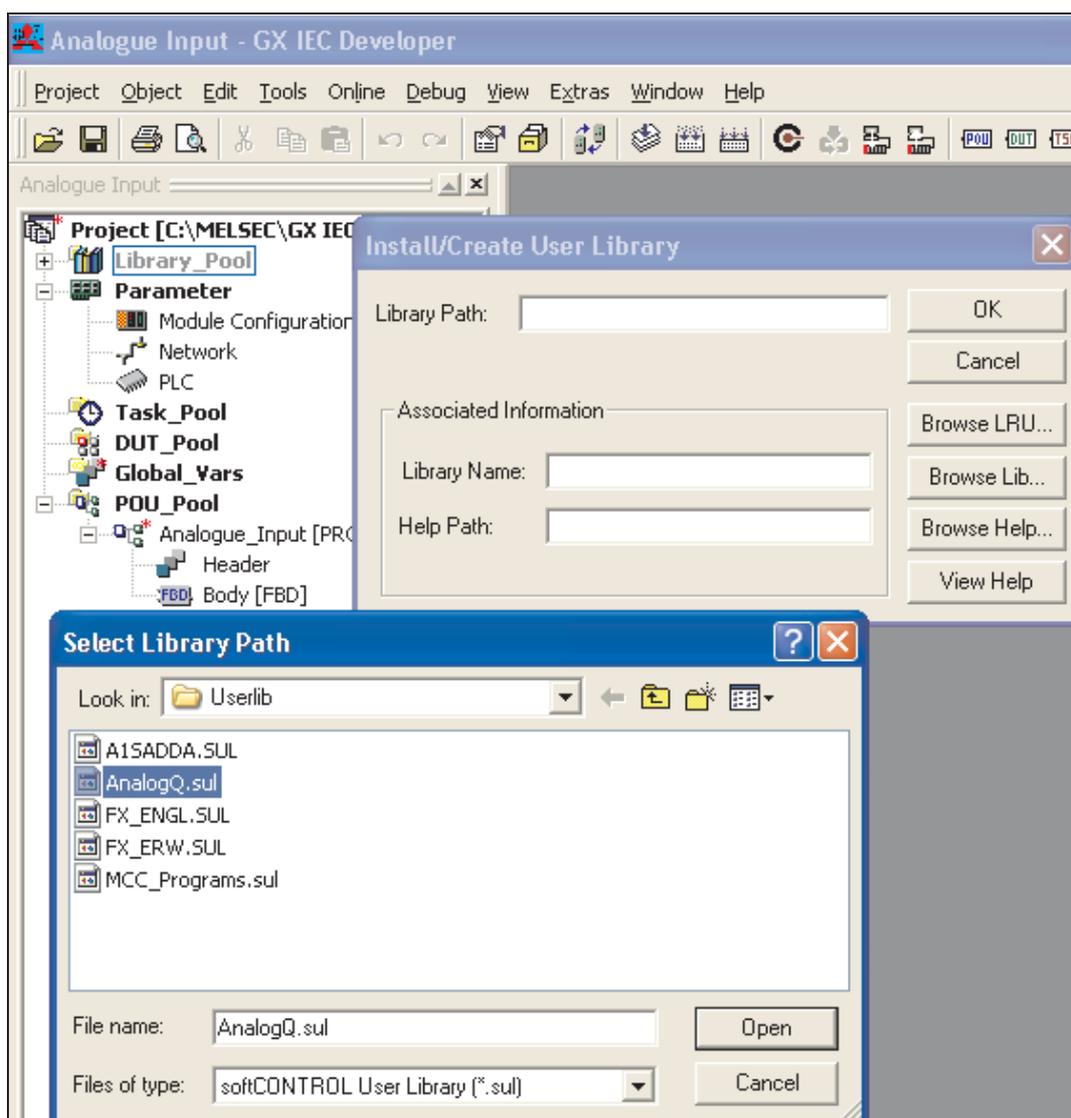
ПРИМЕЧАНИЕ

| Этот пример работает только для ПЛК серии MELSEC System Q.

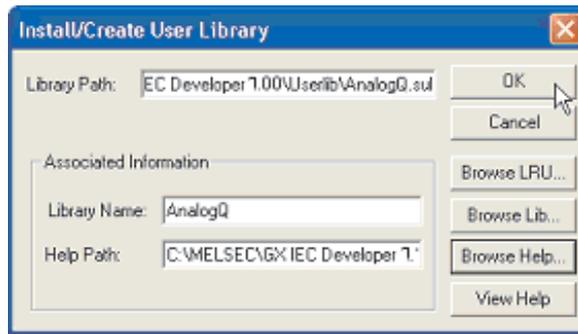
Для правильной работы следующего примера необходимо установить в проект библиотеку Analogue Library для серии Q Mitsubishi.

Библиотеку аналоговых функциональных блоков "AnalogQ" следует найти на вебсайте Mitsubishi или можно установить непосредственно с диска GX IEC Developer из раздела выбора функционального блока в инсталляционной программе. Теперь можно получить доступ к библиотеке в каталоге "Userlib".

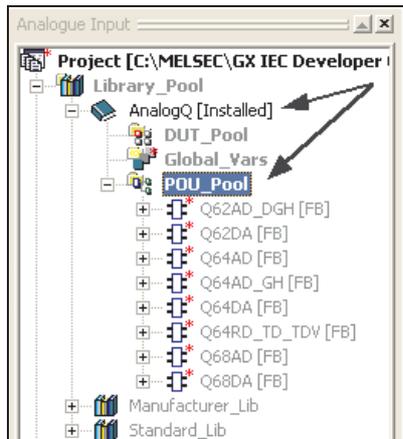
- ① Создайте новый пустой проект без программных модулей, называемый "Analogue_Demo".
- ② Создайте новый программный модуль (Type: **FBD**, Class: **PRG**) и назовите его "Analogue_Input"
- ③ Щелкните правой кнопкой на значке библиотечного пула **Library_Pool** и выберите **Browse Lib**. Выберите файл библиотеки "AnalogQ.sul" и щелкните на **Open**.



- ④ Щелкните на кнопке OK в окне запроса **Install/Create User Library**:

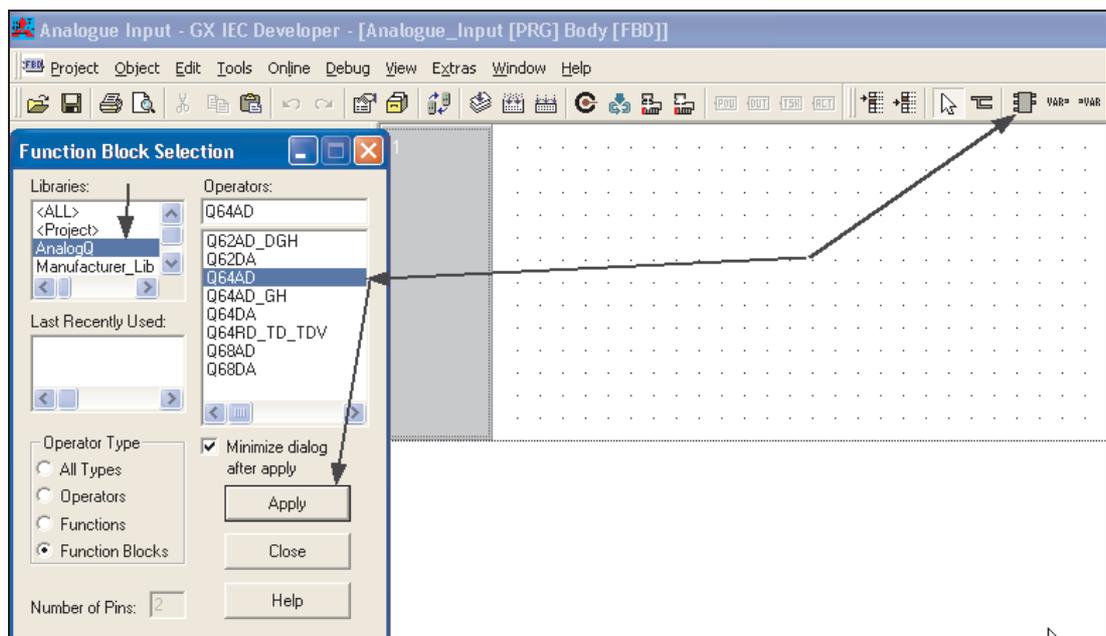


Видим, что в навигационном окне проекта появилась новая библиотека "AnalogQ".

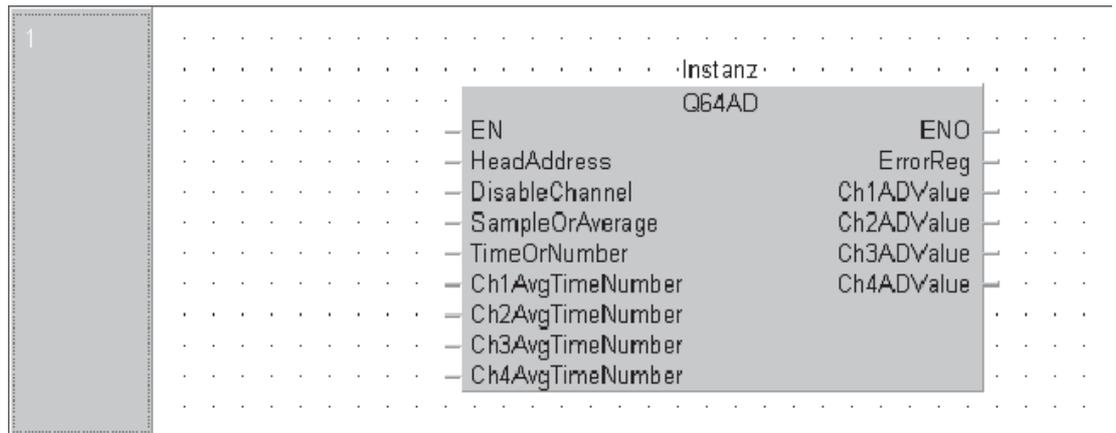


- ⑤ Создайте новую задачу в задачном пуле: "MAIN" и свяжите с ней программный модуль "Analogue_Input".

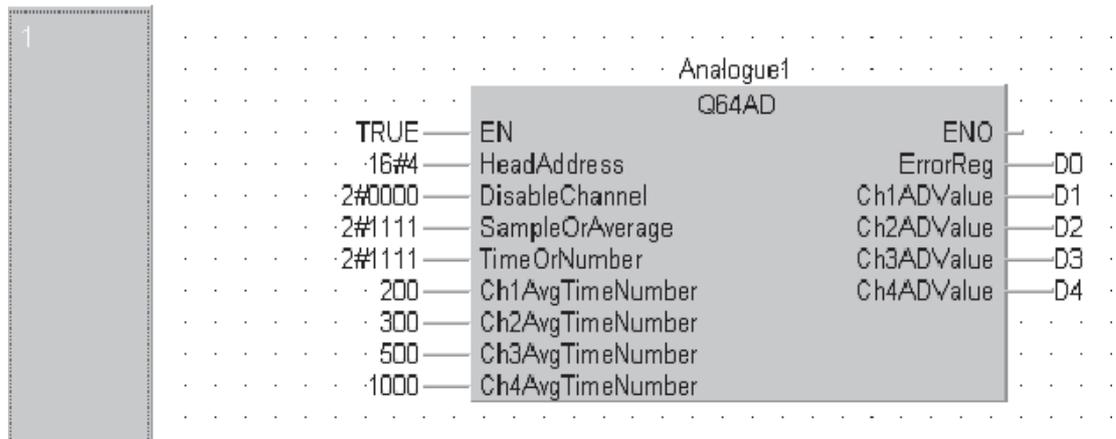
- ⑥ Поместите функциональный блок Q64AD в программный модуль, как показано ниже:



Функциональный блок будет иметь следующий вид:



⑦ Определите все переменные, как показано ниже:



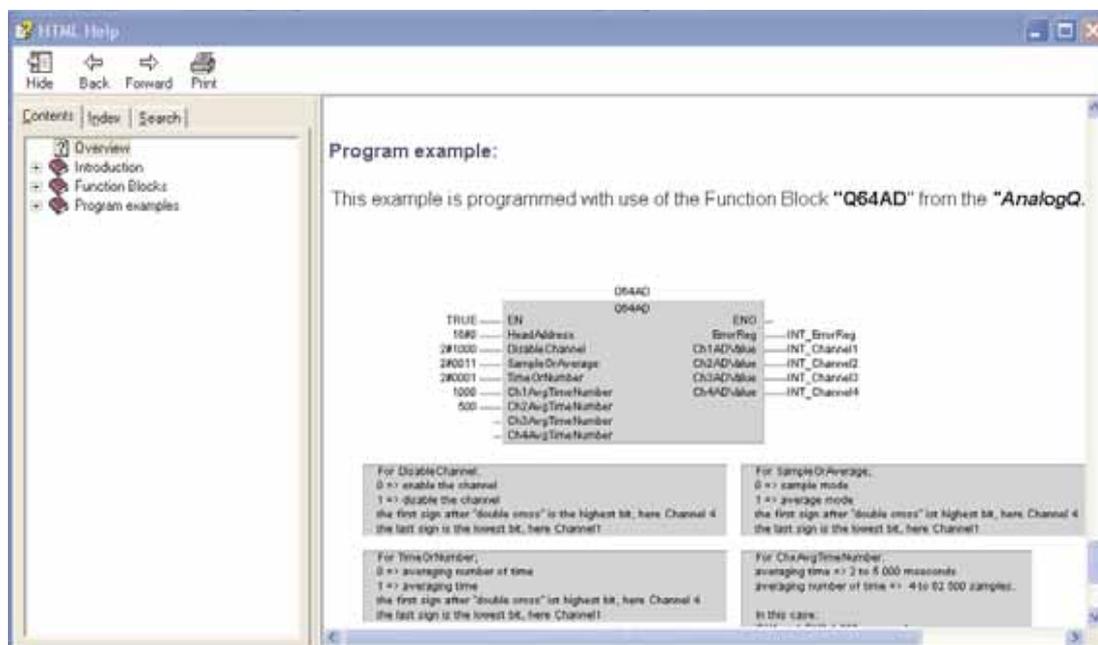
⑧ Компилируйте и загрузите программу в ПЛК.

⑨ Перейдите в режим Монитор и проверьте правильность работы. Наблюдайте поведение аналоговых выходов в зависимости от "настроек выборки".

13.3.3 Контекстная справка библиотечного функционального блока:

Если был импортирован соответствующий файл справки библиотеки для полного объяснения с примерами всех функциональных блоков библиотеки Analogue Q, щелчком выделите функциональный блок и нажмите клавишу "F1".

Будет показан следующий HTML справочный экран:



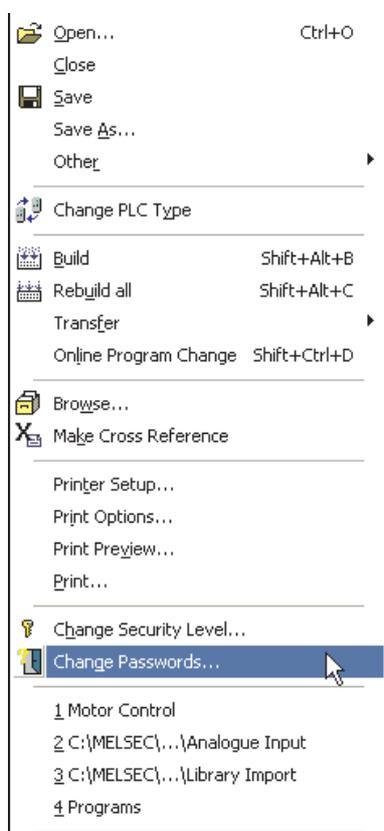
Справочные файлы описывают каждый аспект конфигурации аналоговых аппаратных модулей серии Q, что облегчает использование библиотечных функциональных блоков.

14 Безопасность

14.1 Пароль

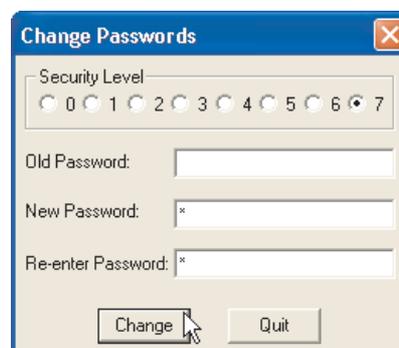
Вы можете защитить паролем всю программу или ее части. Вы можете защитить части программы от редактирования, а также защитить цепи от просмотра другими пользователями. Это особенно уместно для пользовательских функциональных блоков. Кроме того, имеется также пароль (ключевое слово) для ПЛК.

14.1.1 Настройка пароля

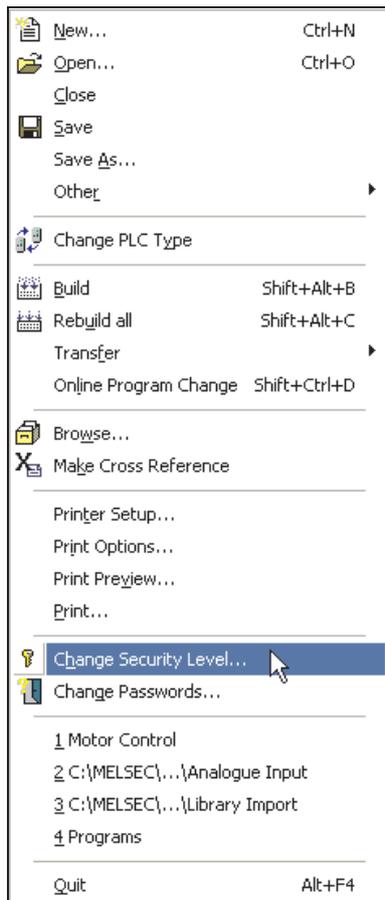


Можно ввести пароли и изменить уровни безопасности, используя эти окна, через меню **Project**.

Чтобы иллюстрировать работу паролей, выберите **Security Level** и введите новый пароль для этого уровня (для простоты нажмите 7). Повторно введите пароль и щелкните на **Change**.

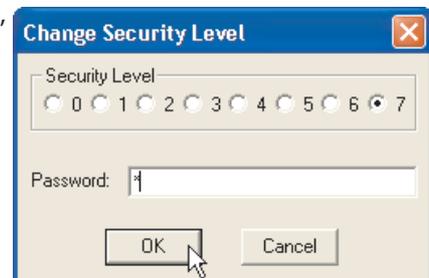


14.1.2 Изменение уровня безопасности



① Выберите **Change Security Level** из меню **Project**:

② Введите пароль для "Level 7" и, если он будет принят, пользователь войдет в систему на этом уровне.



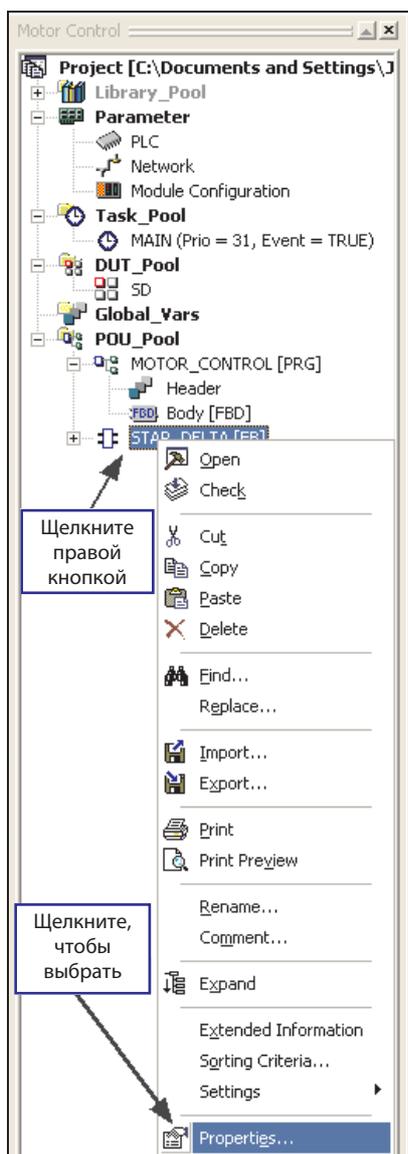
Войдя в систему, можно изменить атрибуты безопасности для многих компонентов. Одной из наиболее распространенных опций безопасности является изменение доступа к ПМ, например, пользовательским функциям и функциональным блокам.

14.1.3 Модификация парольного доступа к программному модулю

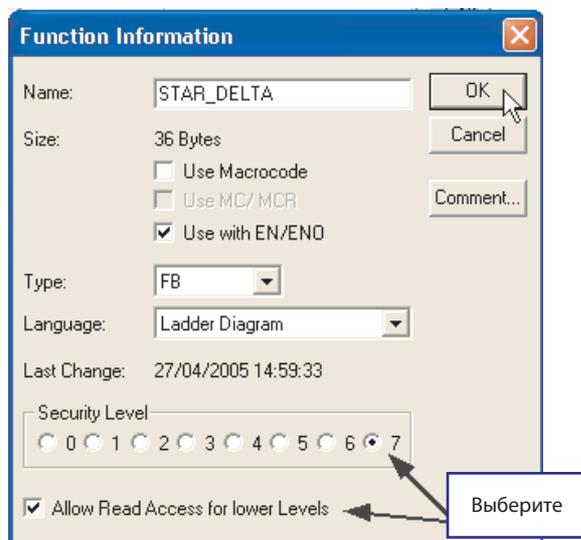
Чтобы защитить содержимое или управлять доступом к пользовательским программным модулям, можно следующим образом настроить атрибуты безопасности, зарегистрировавшись на текущем уровне безопасности:

Настройка уровня безопасности

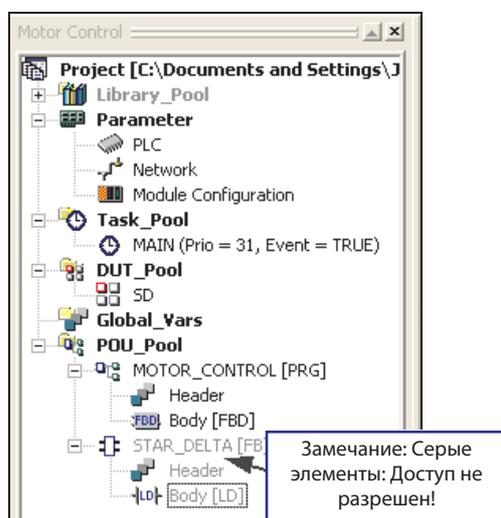
- 1 Откройте проект "Motor Control" и откройте заголовок функционального блока "STAR_DELTA":



- ② Выберите уровень безопасности "7" и щелкните на **Allow Read Access for lower Levels**. Это даст пользователям низшего уровня только "Доступ для чтения" к заголовку и телу функционального блока:



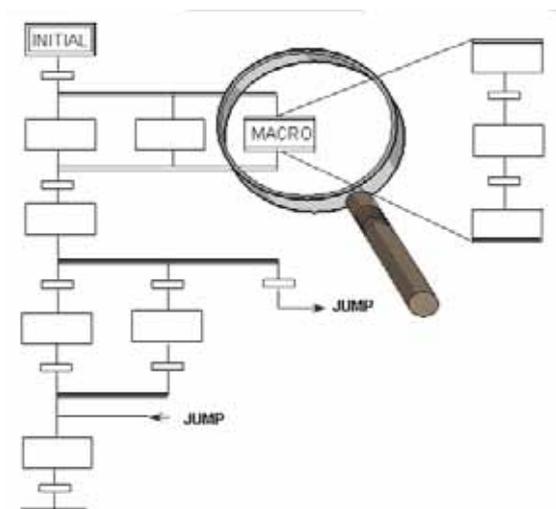
- ③ Измените уровень безопасности на "0" и получите доступ к заголовку и телу функционального блока "STAR_DELTA". Будет разрешен только доступ для чтения с целью мониторинга, но любое изменение кода невозможно.
- ④ Снова зарегистрируйтесь на уровне 7 и измените атрибуты безопасности функционального блока "STAR_DELTA" так, чтобы доступ для чтения был не разрешен для пользователей более низких уровней.
- ⑤ Измените уровень безопасности на "0" и попробуйте получить доступ к телу функционального блока "STAR_DELTA". Заголовок и тело программного модуля будут показаны серым цветом, таким образом доступ к программному модулю полностью заблокирован:



В показанном выше окне "Project Navigation" можно индивидуально установить атрибуты доступа для любого отдельного объекта или всего каталога, что обеспечивает высокую гибкость настроек безопасности программы.

15 Последовательная функциональная схема - SFC

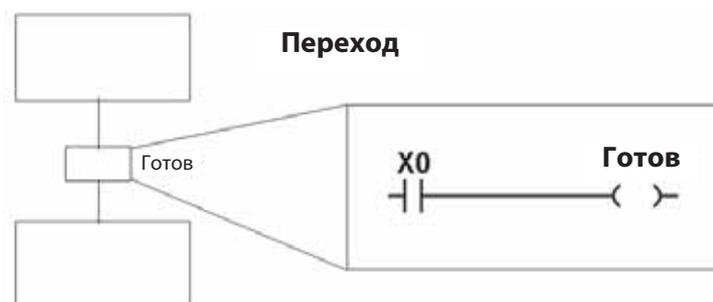
15.1 Что такое SFC?



- Редактор "Последовательных функциональных схем" является управляемым редактором.
- Представление графической блок-схемы.
- На основе французского Grafset (IEC 848).
- SFC является структурным языком, который разделяет процесс на шаги и переходы.
- Этапы "скрывают" действия (не программные модули) и / или непосредственно переключаемые битовые операнды.
- Переходы всегда содержат один указатель/звено, которое активизирует инструкцию выполнения последовательности (имя перехода). (Также возможно использовать дискретный адрес вместо имени.)
- Действия могут создаваться в любом редакторе, кроме SFC.
- Переходы могут создаваться в любом редакторе, кроме SFC.
- Код SFC находится в микрокомпьютерной области ПЛК, поэтому пространство памяти выделяется в параметрах ПЛК PLC Parameters (только для серии A).

15.2 Элементы SFC

15.2.1 Переходы SFC



- Переход представляет указатель, который начинает последовательность.
- Они могут создаваться в любом IEC-редакторе.
- Но не в редакторе SFC.
- Также возможно использовать бит непосредственно, вместо имени READY.

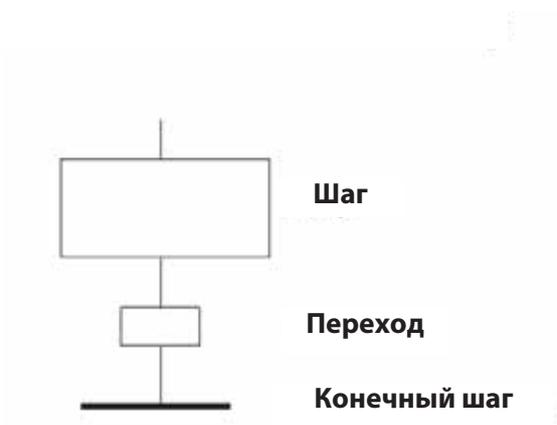
15.2.2 Начальный шаг

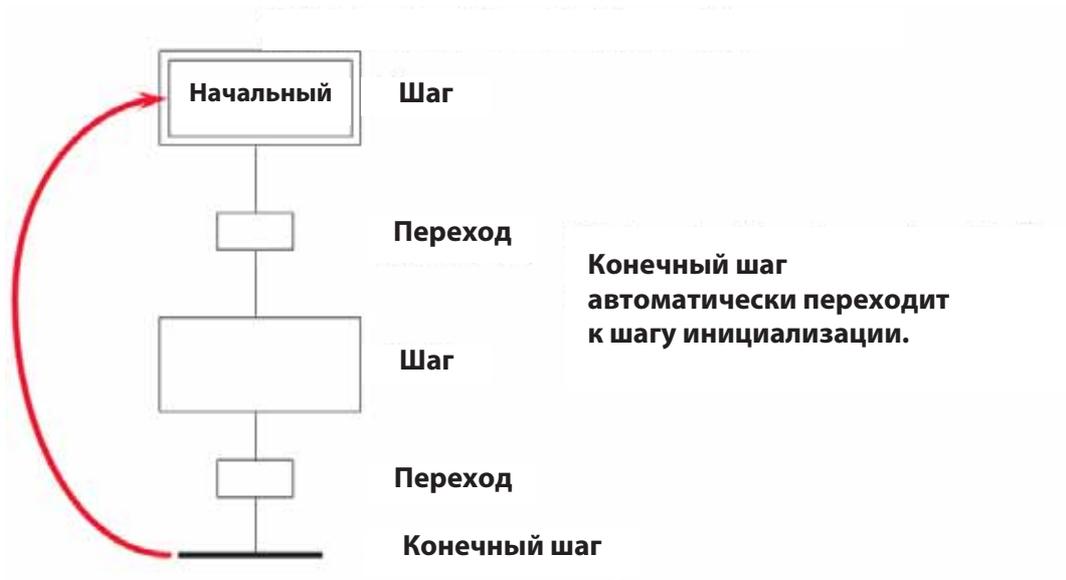
Программы SFC начинаются с функции начального шага, которая указывает начало последовательности:



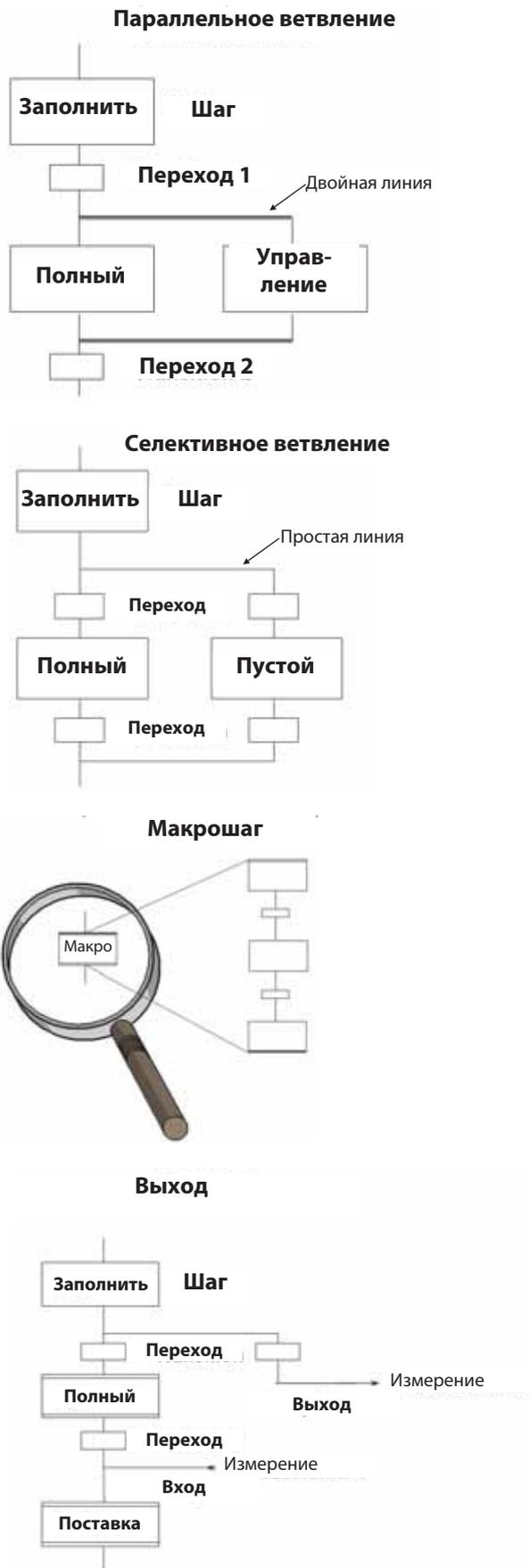
15.2.3 Конечный шаг

Все последовательности заканчиваются конечным шагом:



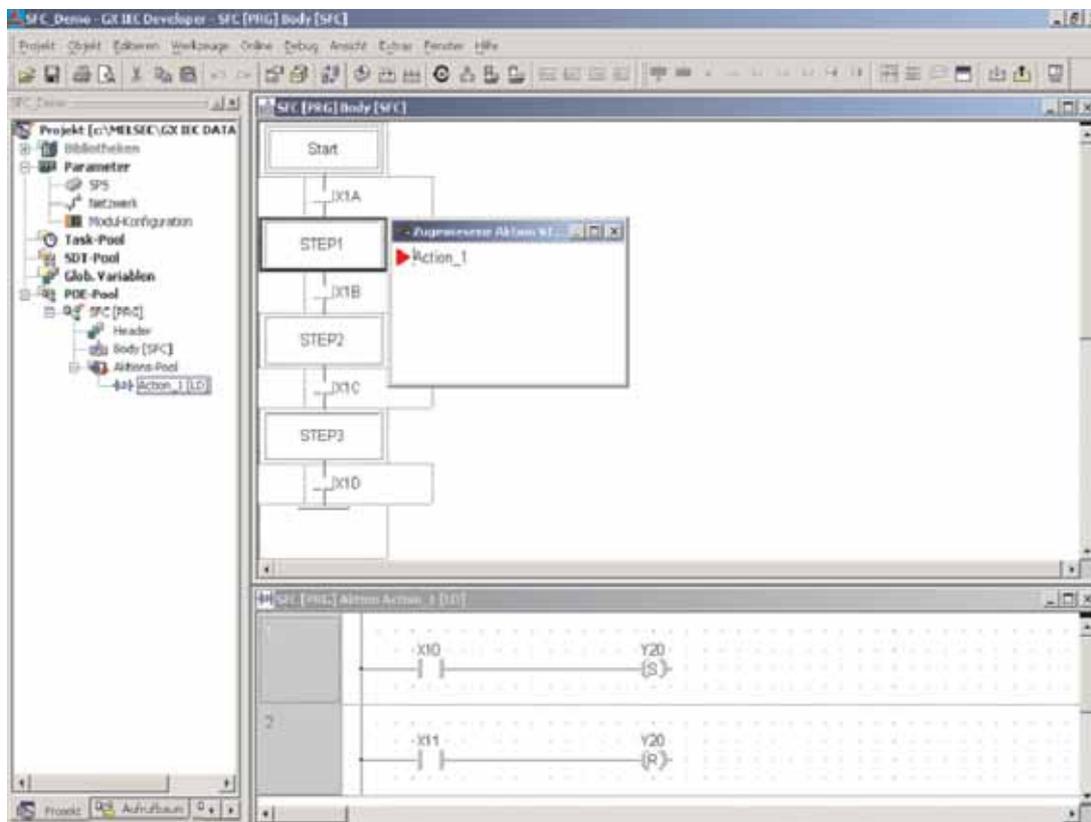


15.3 Примеры конфигурации SFC

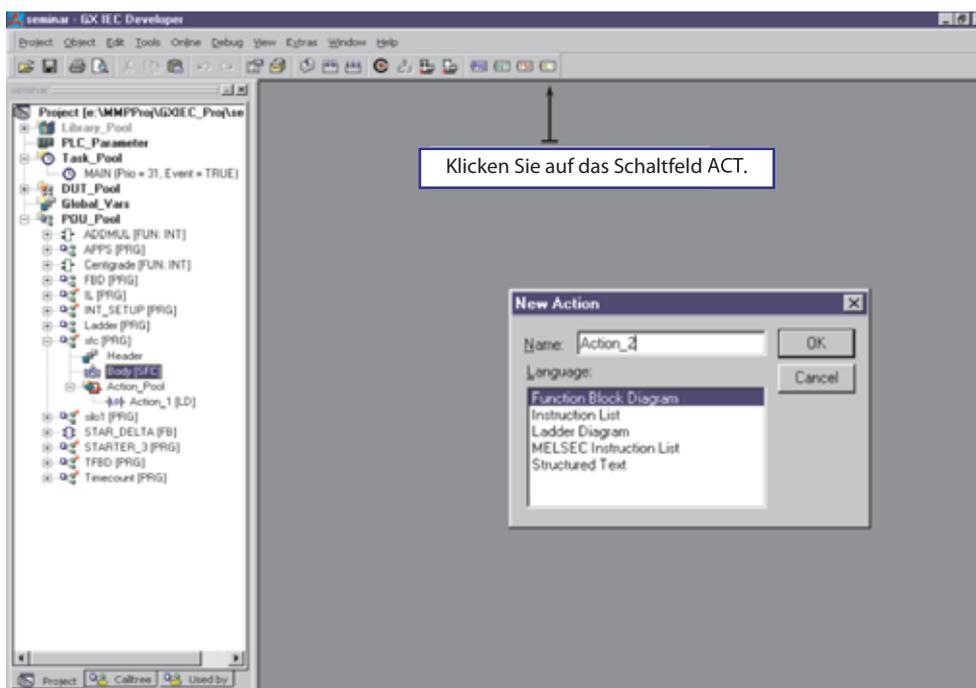


15.4 Действия SFC

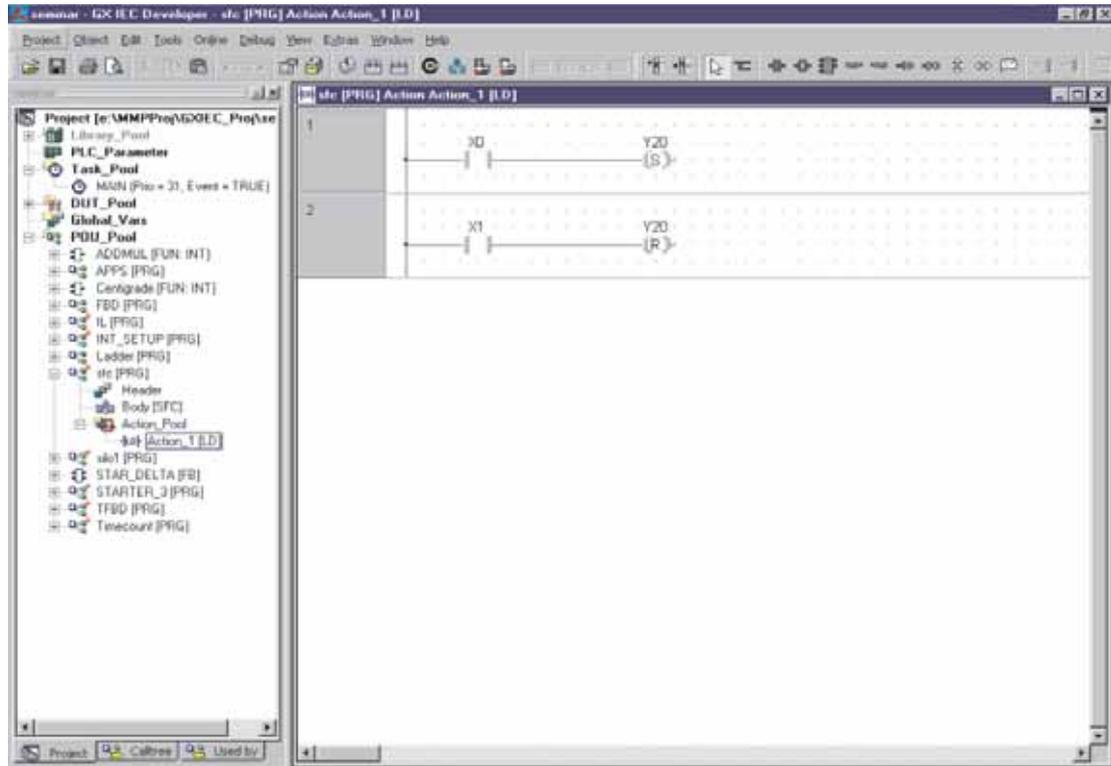
Каждый шаг имеет связанные действия. Действие является просто программой, как для программного модуля. Каждое действие имеет соответствующую логику, написанную на одном из языков IEC LD, IL, FBD или ST:



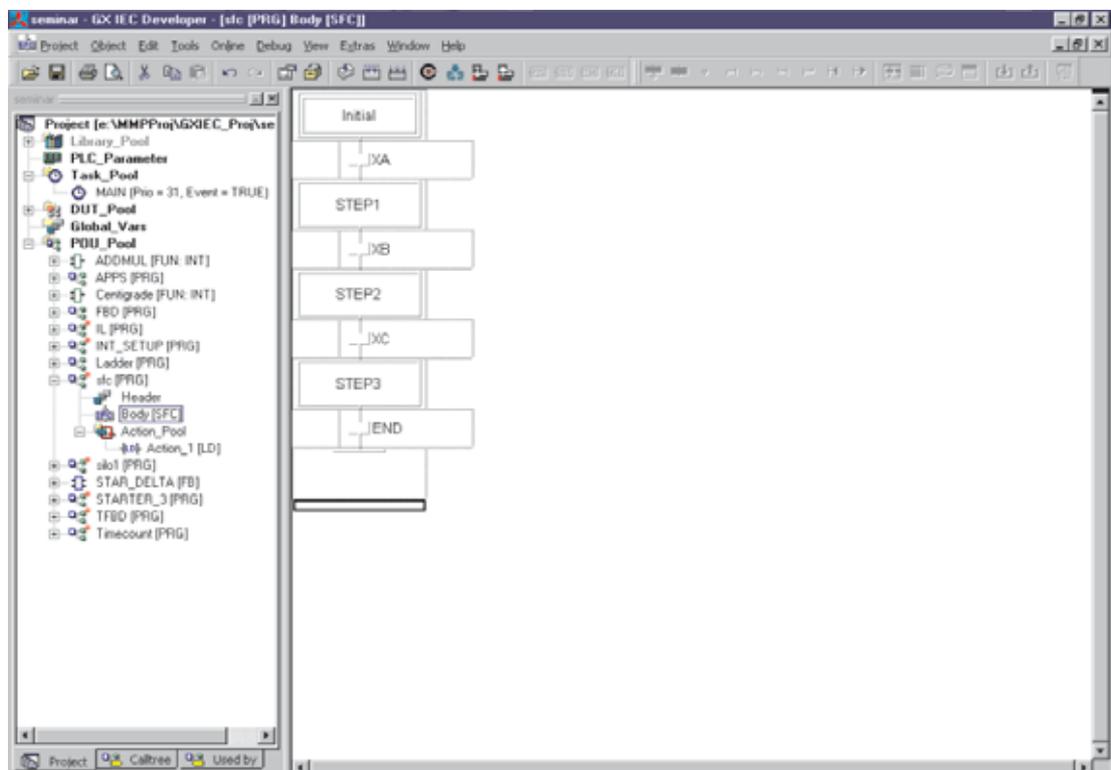
Чтобы создать новое действие, щелкните на кнопке **ACT** на панели инструментов. Выберите необходимый редактор, как для программного модуля:



Действия могут быть программами в их собственном смысле. Action_1 может быть полной релейной подпрограммой блокировки, включающей много звеньев.

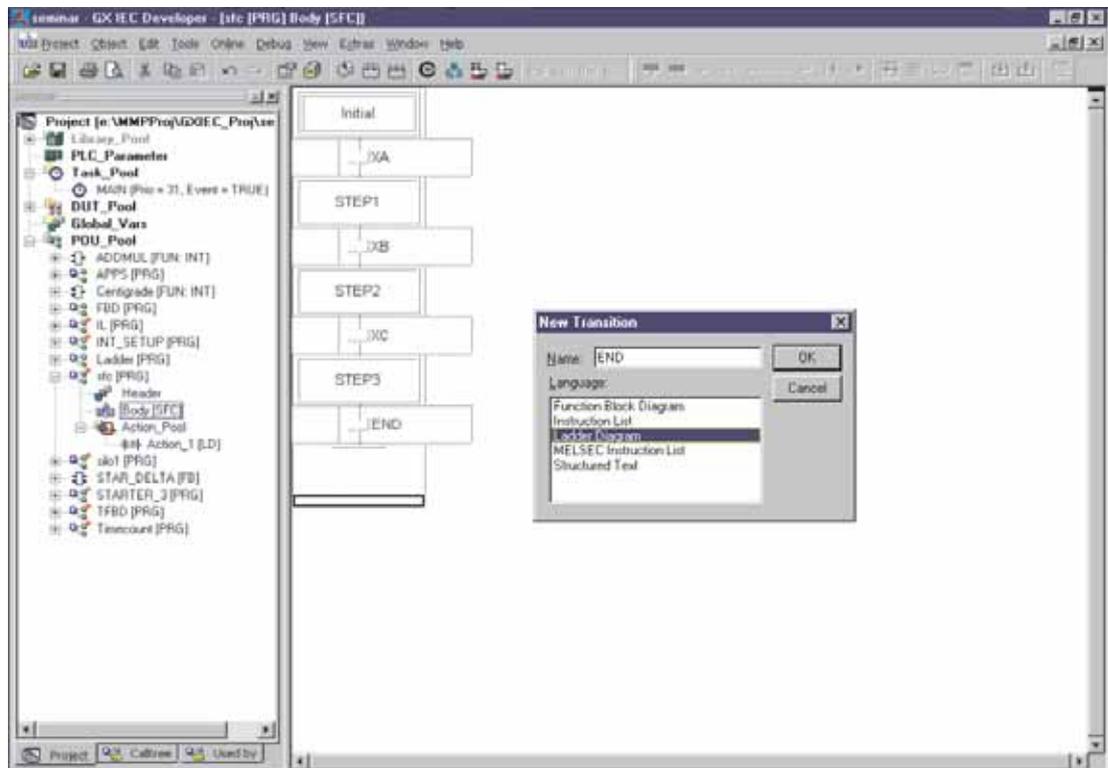


Каждый переход может быть простым операндом, например, адресом Mitsubishi XA, или именем идентификатора, или более сложным объектом, например, программой для одного звена, написанной на одном из языков IEC, IL, LD или FBD:

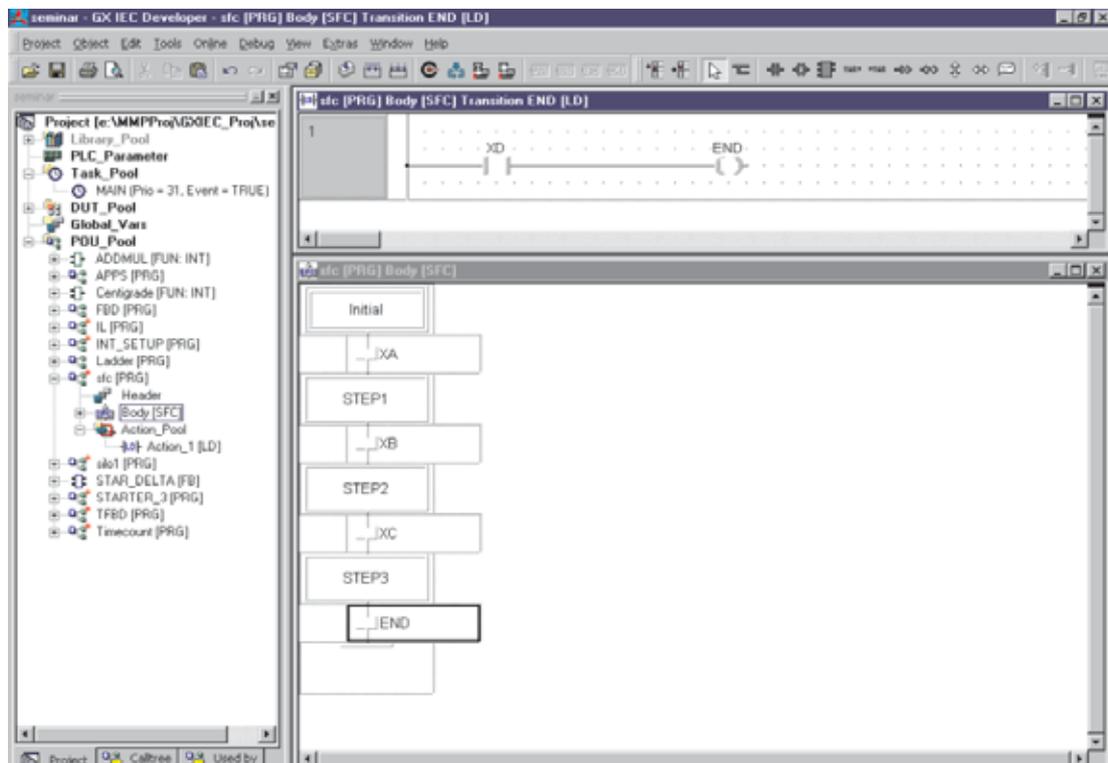


15.5 Сложные переходы

Для программирования сложного перехода введите имя перехода и нажмите клавишу **ENTER**. Выберите необходимый редактор, как для действия:

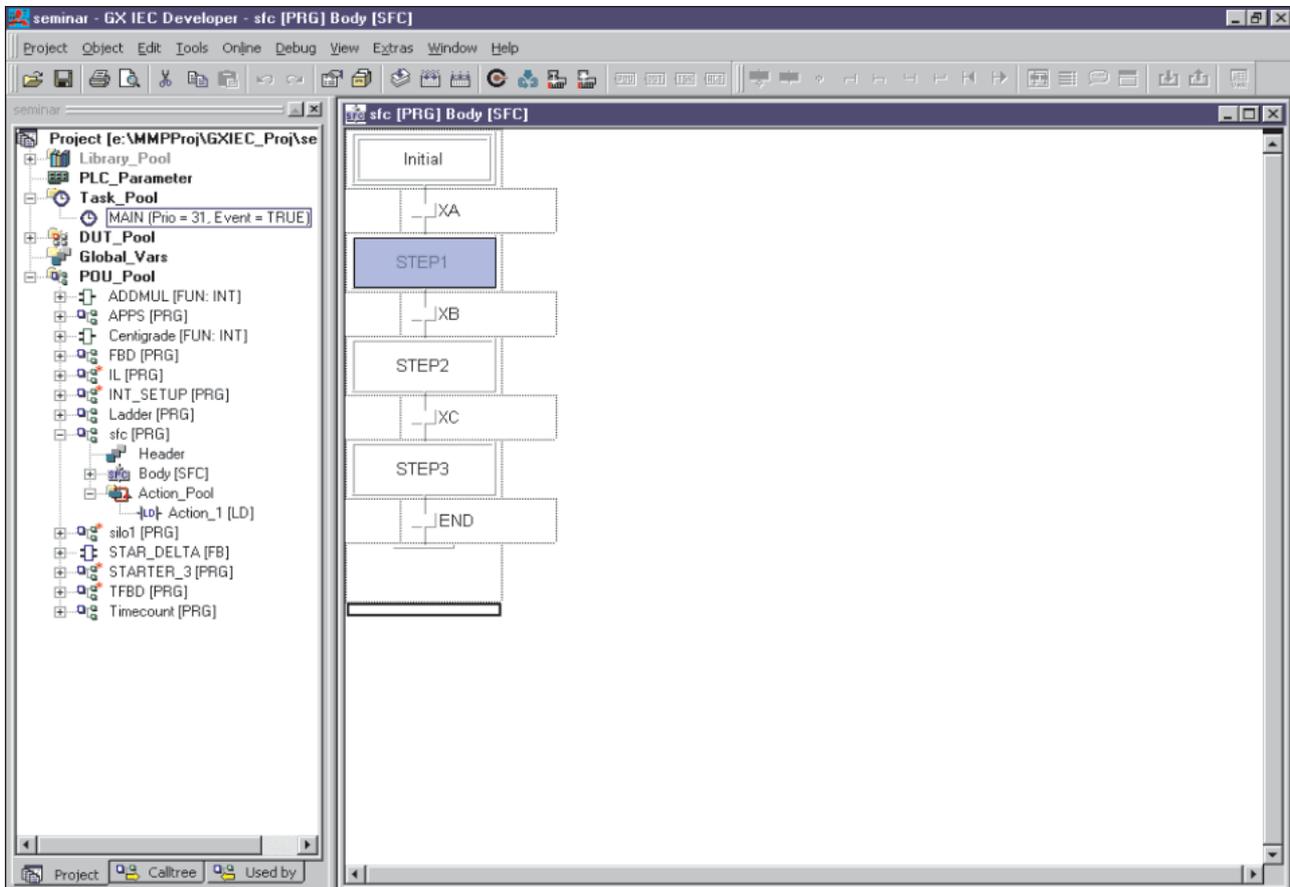


Переход может быть сложным выражением, но он состоит только из одного звена:



15.6 Экран программы SFC в режиме монитора

Одна из основных возможностей SFC заключается в том, что в режиме Монитора текущий шаг подсвечивается. Это означает, что при поиске ошибок инженеры могут точно видеть ход обработки последовательности и соответственно исследовать ситуацию:



16 Список инструкций IEC

- Редактор "списка инструкций" является свободным текстовым редактором.
- Адреса строк не требуются.
- Могут вызываться функции и функциональные блоки.
- В дополнение к фрагментам IEC могут быть включены фрагменты MELSEC.
- Можно включать комментарии, заключив их в (* *).
- Используя функциональность Windows, можно написать программу, например, в WinWord, и затем скопировать через буфер обмена в GX IEC Developer.

16.1 Пример списка инструкций IEC (IL)

```
LD      X4          (* Опрос X4 *)
ANDN    M5          (* ANDN M5 *)
ST      Y20        (* Назначение OUT к Y20 *)

LD      ПРОВЕРЯТЬ (* Загрузить TEST в сумматор *)
BCD_TO_INT      (* Преобразовать сумматор *)
ST      RESULT    (* Записать сумматор приводить в RESULT *)
```

16.1.1 Некоторые полезные подсказки

Чтобы выполнить "+ D0 D1 D2" в IEC IL, напишите:

```
LD      D0
ADD     D1
ST      D2
```

Чтобы выполнить "+ D0 D1 D2" и затем "+ D2 K50 D3", напишите:

```
LD      D0
ADD     D1,D2,50
ST      D3
```

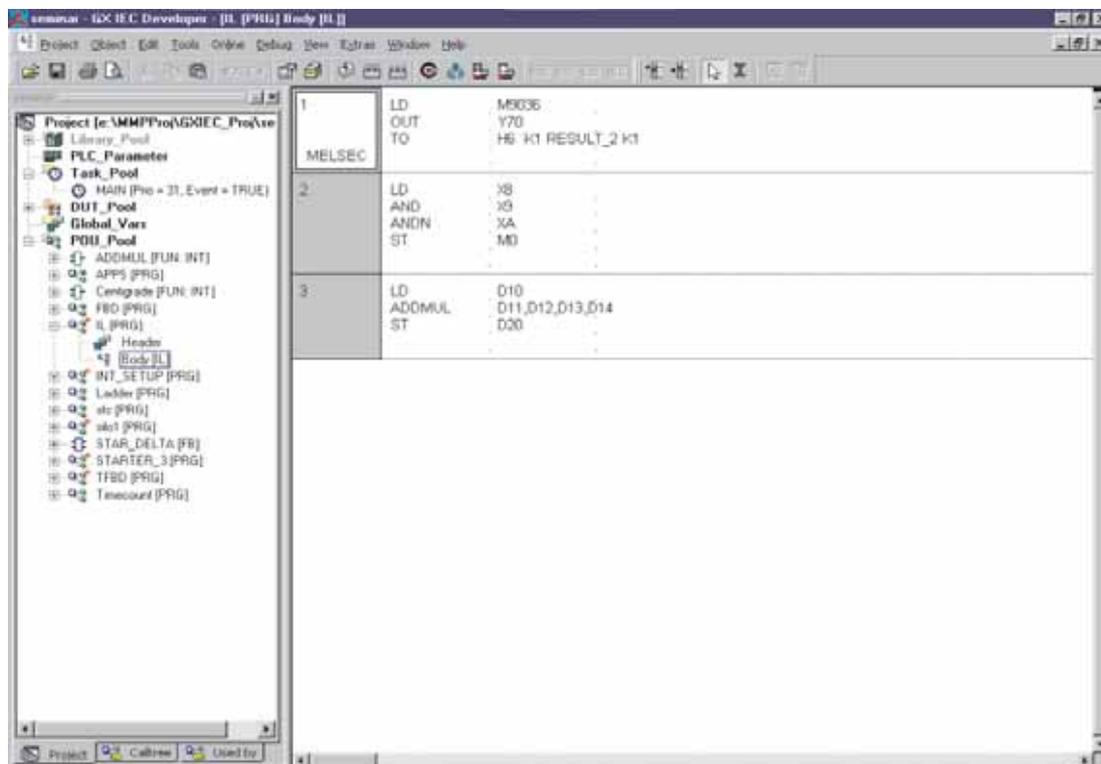
Использование функции "_E" позволяет еще больше все упростить. Чтобы выполнить "+ D0 D1 D2" и затем "+ D2 K50 D3" из условного входа X0, напишите:

```
LD      X0
ADD_E   D0,D1,D2,50,D3
```

Причина заключается в том, что функция ADD_E имеет возможность Enable Output (ENO).

16.2 Смешивание IEC IL и MELSEC IL в программных модулях

Фрагменты IEC IL и MELSEC IL могут включаться в один программный модуль. Для этого подсвечите текущее звено, выберите из меню **Edit New Network** затем **Melsec Before** из списка **Options**:



17 Структурированный текст IEC

ST - текстовый редактор высокого уровня, имеющий PASCAL - подобный вид, но являющийся специализированным языком для приложений управления производственными процессами.

Используя ST, можно создавать программные модули, функции и функциональные блоки.

Пример структурированного текста IEC:

IFTHEN ELSE conditions
CASE ...ELSE ... END_CASE structures
REPEAT
RETURN
Expression Evaluation
Variable Declaration etc

Используя эти операторы, можно создать сложные математические выражения в нескольких строках текста.

17.1 Операторы языка структурированного текста

Оператор	Описание	Приоритет
(...)	Выражение в скобках	Наивысший
Функция (...)	Список параметров функции, оценка функции	
**	Возведение в степень	
-	Отрицание	
NOT	Булево дополнение	
*	Умножение	
/	Деление	
MOD	Операция по модулю	
+	Сложение	
-	Вычитание	
<, >, <=, >=	Операторы сравнения	
=	Равенство	
<>	Неравенство	
AND, &	Булево И	
XOR	Булево исключающее ИЛИ	
OR	Булево ИЛИ	Наинизший

17.2 Пример программы языка структурированного текста

Мы создадим новый функциональный блок, преобразующий "Градусы Цельсия в градусы Фаренгейта", аналогичный использованному в предыдущем примере, чтобы проиллюстрировать использование редактора языка "структурированного текста".

Используется следующая формула:

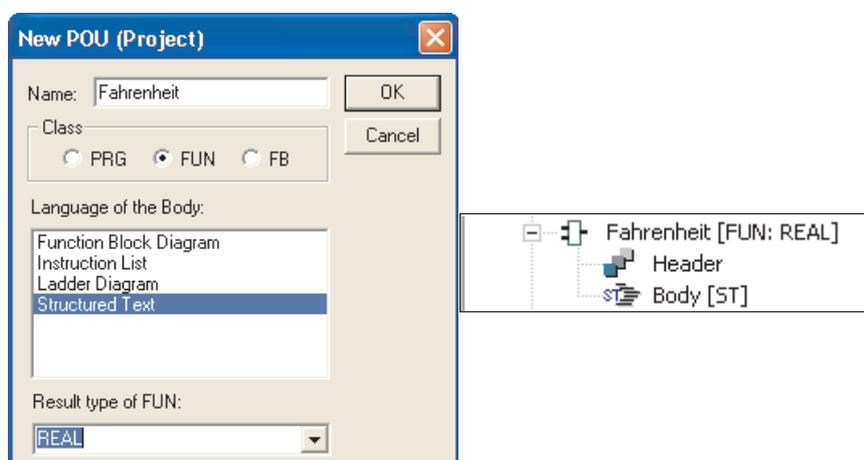
$$\text{Fahrenheit} = \frac{\text{Celsius} \times 9}{5} + 32$$

Входная переменная и переменная результата будут в формате с плавающей запятой (REAL).

ПРИМЕЧАНИЕ

Для ПЛК серии FX вычисления с плавающей запятой возможны только с главными блоками серии FX2N, FX2NC и FX3U.

- ① Создайте новый проект, называемый "Structured_Text".
- ② Создайте новый программный модуль "Fahrenheit", Class: **FUN**, Тип результата: **REAL**, язык "ST" (Structured Text):



- ③ Создайте элемент в заголовке (LVL) функции "Fahrenheit":

	Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	Centigrade	REAL	0.0	

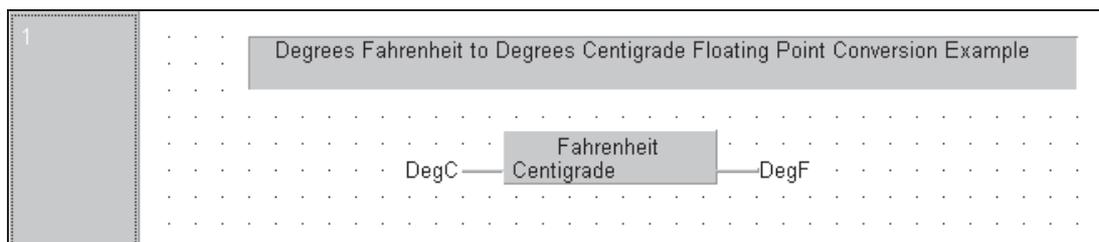
- ④ Откройте тело функции "Fahrenheit" и введите следующую простую программу на языке ST:

Fahrenheit := (Centigrade*9.0/5.0+32.0);

- ⑤ Создайте новый программный модуль "Temp_Conv", Class: **PRG**, Language: **Function Block Diagram**.



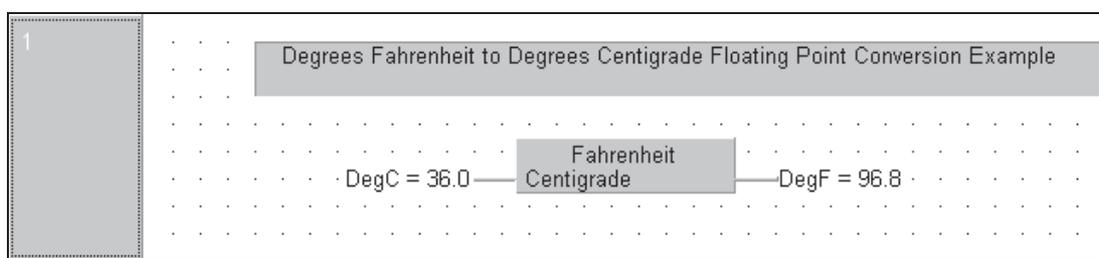
- ⑥ Откройте тело программного модуля программы "Temp_Conv" и введите следующую иллюстрационную программу:



- ⑦ Отредактируйте LVL (Заголовок) программного модуля "Temp_Conv", чтобы включить две локальные переменные, как показано ниже:

	Class	Identifier	Type	Initial	Comment
0	VAR	DegC	REAL	0.0	
1	VAR	DegF	REAL	0.0	

- ⑧ Закройте все открытые редакторы, компилируйте проект, используя "Rebuild All". Сохраните и загрузите в ПЛК.
- ⑨ Контролируйте выполнение тела программы "Temp_Conv" и наблюдайте значения на экране.
- ⑩ Принудительно установите новые значения входной переменной "DegC" уравнения, дважды щелкнув на символическом имени переменной.



ПРИМЕЧАНИЕ

В этом примере используются локальные переменные для непосредственного ввода значений через интерфейс программирования / мониторинга GX-IEC Developer; обычно значения вводятся через глобальные переменные.

18 Связь по PROFIBUS/DP

Открытая сеть PROFIBUS/DP поддерживает чрезвычайно быстрый обмен данными с очень широким кругом подчиненных устройств, включая устройства удаленного цифрового ввода/вывода, устройства удаленного аналогового ввода/вывода, преобразователи частоты и диапазон других устройств прочих производителей. Конечно, подчиненные устройства PROFIBUS/DP от Mitsubishi Electric можно подключать к главным устройствам других производителей.

Установка удаленных устройств цифрового или аналогового ввода/вывода помогает уменьшать затраты на прокладку кабелей.

Структура

Максимальный охват шинного сегмента составляет 1200 м (при макс. 93,75 кбит/с). Допускается использование до 3-х ретрансляторов. Таким образом, максимальное расстояние между 2-я станциями составляет 4800 м.

Типы кабелей

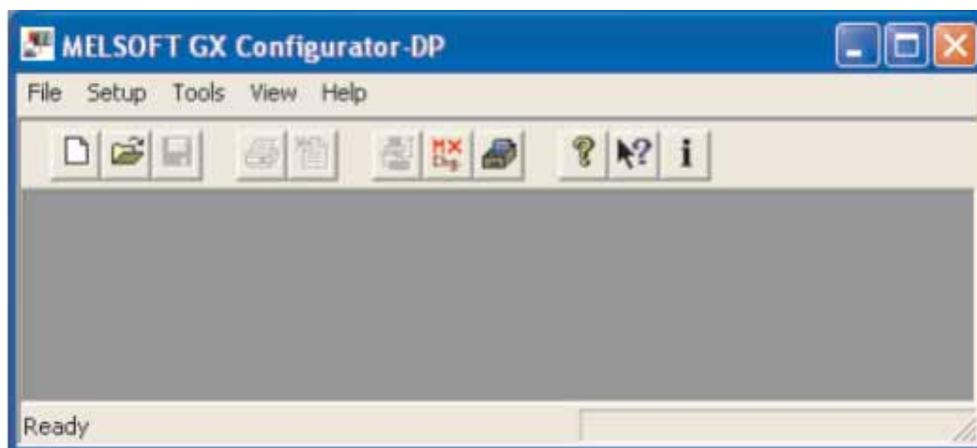
Чтобы сократить стоимость, в PROFIBUS/DP используется технология RS-485 с экранированным 2-проводным кабелем.

18.1 Конфигурирование сети PROFIBUS/DP

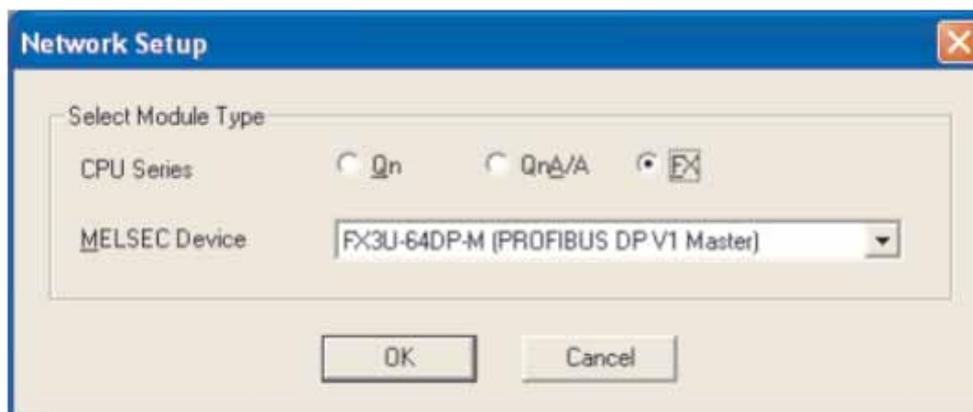
В комбинации с программным обеспечением GX Configurator DP модуль ведущего узла FX3U-64DP-M, а также модули ведущего узла серии A или MELSEC System Q предоставляют вам простую в использовании plug-and-play технологию. ПО для конфигурирования имеет интуитивно понятный характер, используя графическую модель для настройки сети. Пользователь просто выбирает подчиненный модуль, присваивает номера станций и указывает, где хранится информация на главной станции.

В этой главе показано конфигурирование ведущего модуля PROFIBUS/DP - FX3U-64DP-M, установленного в базовый блок FX3U. К ведущему модулю подключена подчиненная станция, состоящая из цифровых и аналоговых модулей серии MELSEC ST. За дополнительной информацией по сериям ST обращайтесь в Технический каталог по сетям, артикул 136730.

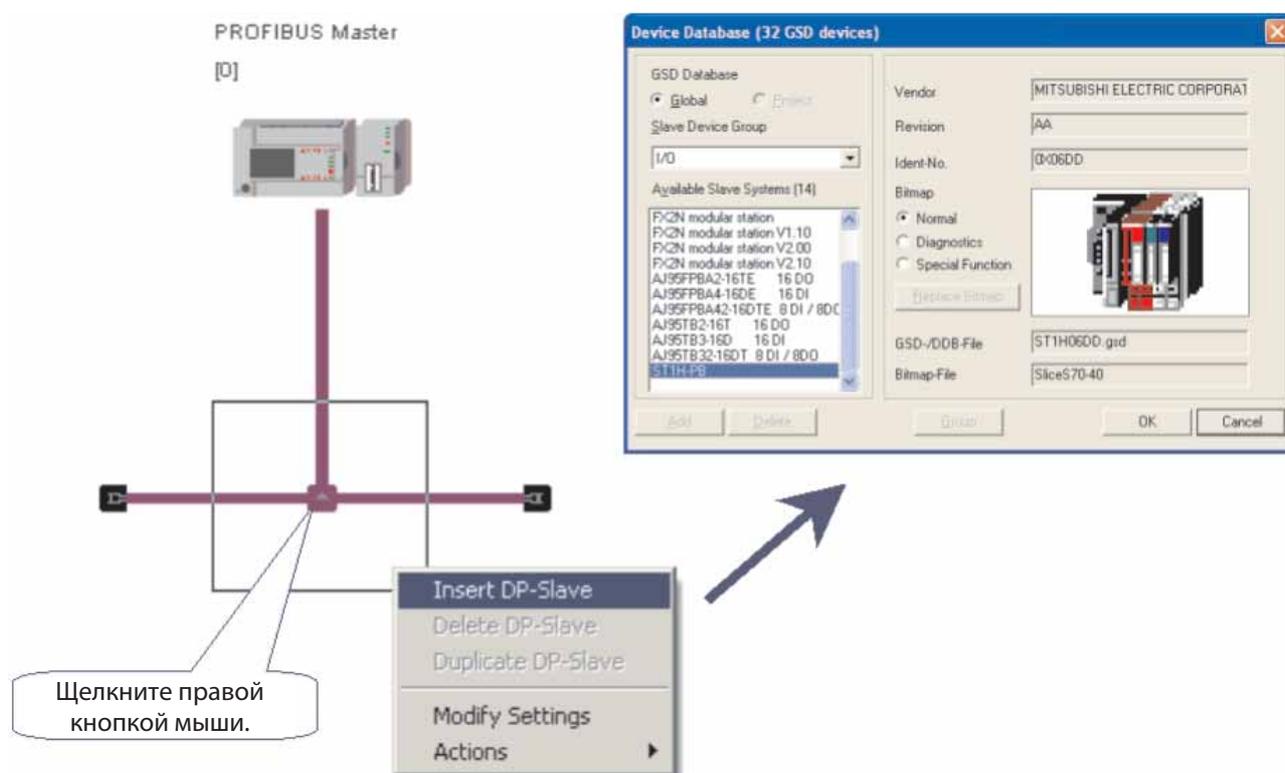
- ① Запустите GX configurator DP и откройте новый проект.



- ② В диалоговом окне Network Setup выберите **FX**. В качестве **MELSEC Device** автоматически вводится **FX3U-64DP-M**.



- ③ Вставьте подчиненное устройство DP в пустой проект.



- ④ Определите адрес заголовка ведущего модуля.

The diagram shows a PROFIBUS Master [0] connected to a Slave_Nr_001 via a bus. The Master Settings window is shown with the following configuration:

Field	Value	Range
Module	FX3U-64DP-M	
Vendor	MITSUBISHI ELECTRIC CORPORATION	
Name	PROFIBUS DP	
Baudrate	1.5 Mbaud	
FDL address	0	[0 - 125]
Head address on PLC	1	[0x0 - 0x7]
Error action flag	<input type="checkbox"/> Goto 'Clear' State	
Min. slave interval	30	[1 - 65535] * 100 μs
Polling timeout	50	[1 - 65535] * 1 ms
Data control time	100	[1 - 65535] * 10 ms
Watchdog	<input type="checkbox"/> Watchdog	
Slave Watchdog time	5	[1 - 65025] * 10 ms
Auton. Refresh	<input type="checkbox"/> Consistency	
Watchdog for time sync.	0	[0 - 65535] * 10 ms

A callout bubble points to the 'Head address on PLC' field with the text: "Введите в это поле адрес заголовка ведущего модуля PROFIBUS/DP. В данном примере это 2-ой специальный функциональный модуль. Поэтому он имеет адрес '1'."

- ⑤ Конфигурируйте подчиненную станцию. В данном примере это головная станция серии MELSEC ST (**ST1H-PB**).

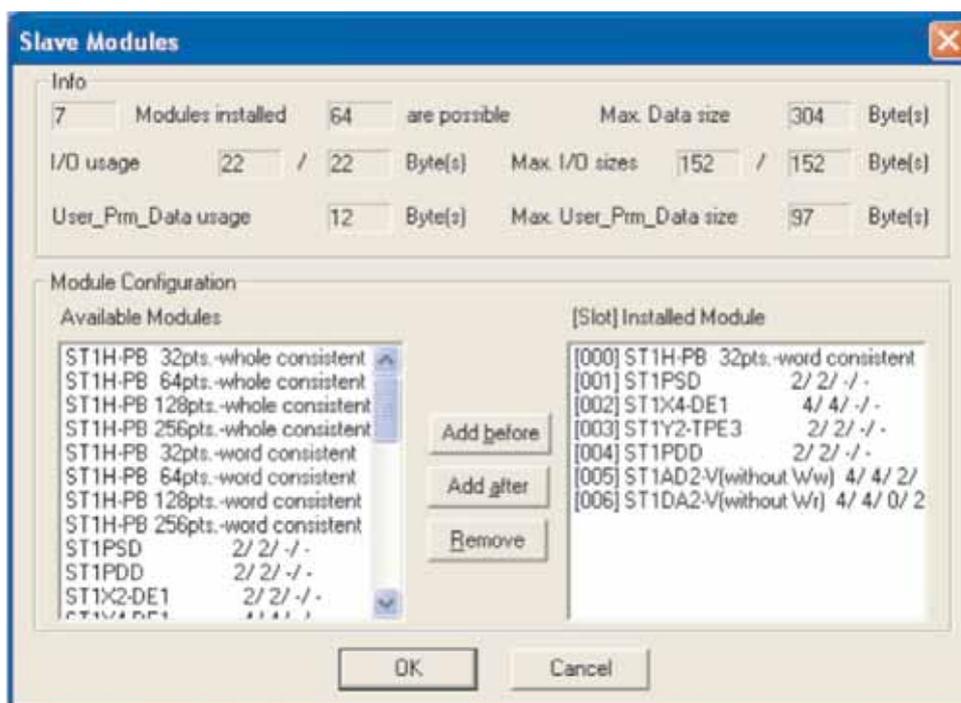
The Slave Parameter Settings window is shown with the following configuration:

Field	Value	Range
Model	ST1H-PB	
Revision	AA	
Vendor	MITSUBISHI ELECTRIC CORPORATION	
Slave Properties		
Name	Slave_Nr_001	
FDL Address	1	[0 - 125]
Watchdog	<input type="checkbox"/> Watchdog	
Slave Watchdog time	5	[1 - 65025] * 10 ms
min T_sdr	11	[1 - 255]
Group identification number	<input type="checkbox"/> Grp 1 <input type="checkbox"/> Grp 2 <input type="checkbox"/> Grp 3 <input type="checkbox"/> Grp 4 <input type="checkbox"/> Grp 5 <input type="checkbox"/> Grp 6 <input type="checkbox"/> Grp 7 <input type="checkbox"/> Grp 8	
Slave is active	<input checked="" type="checkbox"/> Slave is active	
Ignore AutoClear	<input type="checkbox"/> Ignore AutoClear	
Swap I/O Bytes in Master	<input type="checkbox"/> Swap I/O Bytes in Master	
DP V1/V2 Slave Parameters	<input type="checkbox"/> Sync (Output) <input type="checkbox"/> Freeze (Input) <input type="checkbox"/> Initialize slave when failing to respond	

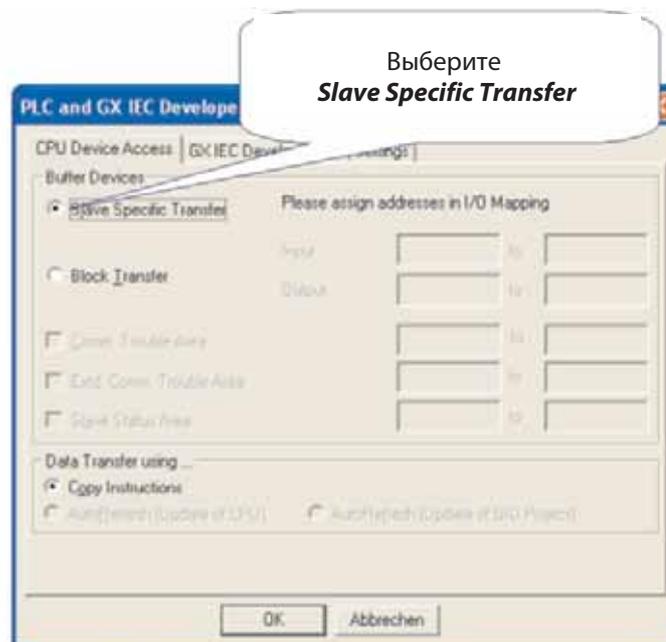
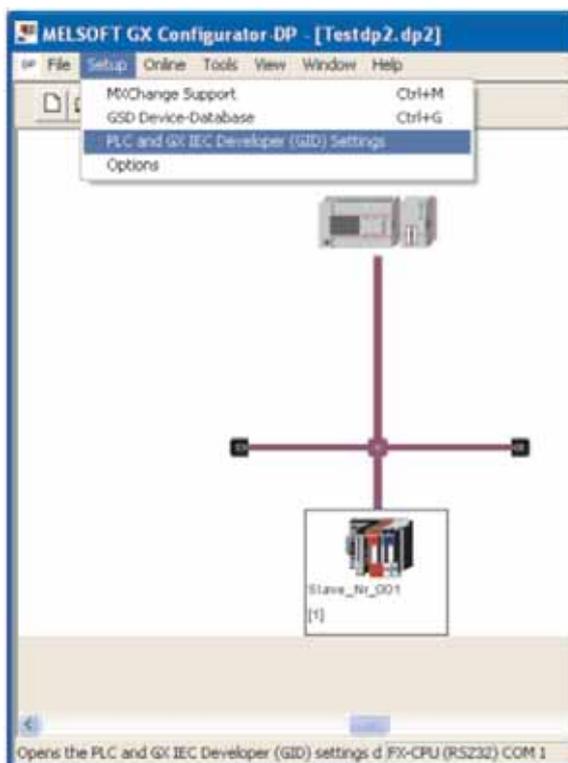
Two callout bubbles provide instructions:

- The first bubble points to the 'FDL Address' field: "Сначала выберите адрес PROFIBUS подчиненной станции."
- The second bubble points to the 'Select Modules' button: "Затем выберите установленные модули системы ST (см. следующую страницу)."

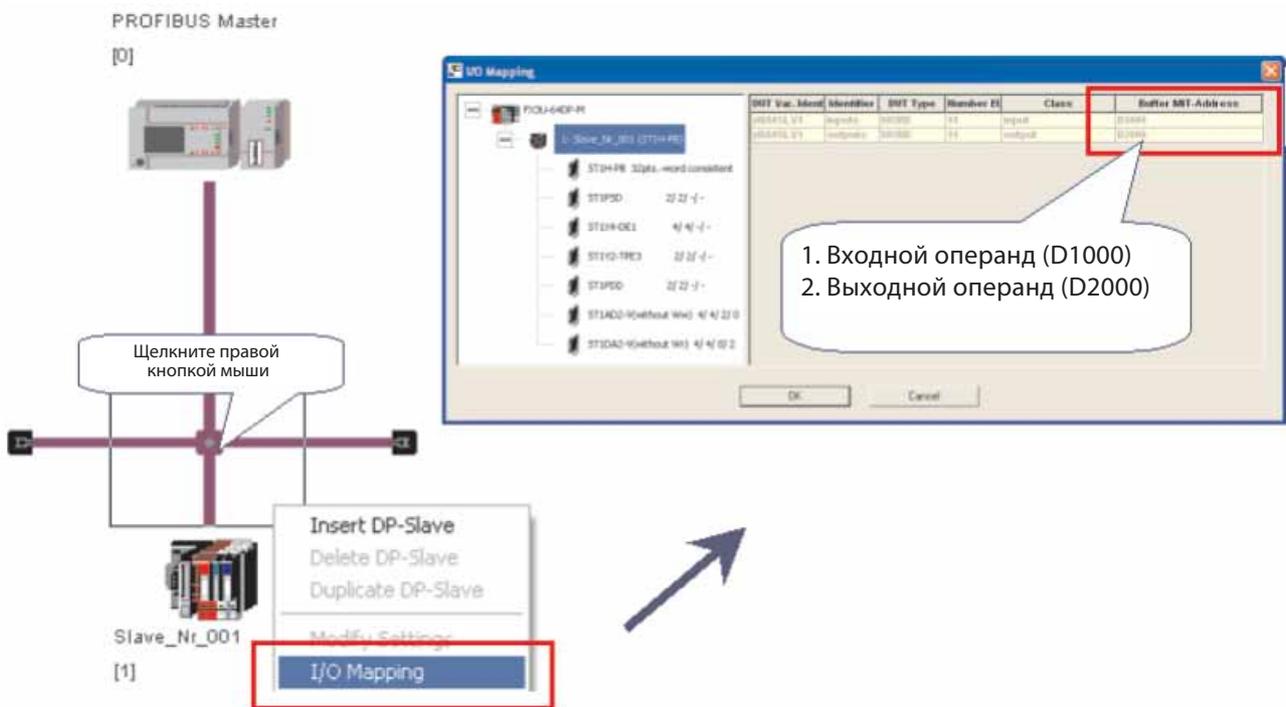
⑥ Выберите модули



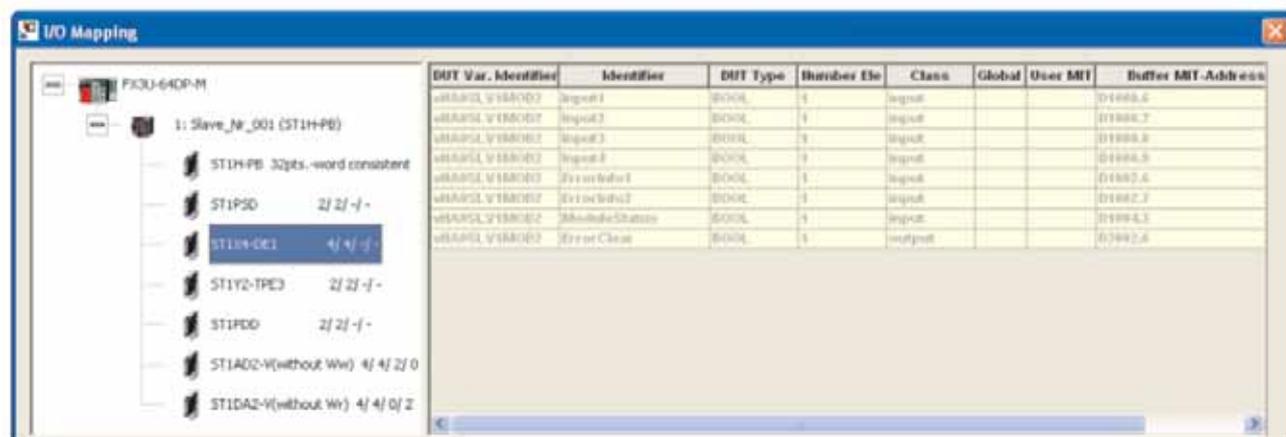
⑦ Выполните настройки ПЛК для устройств ввода и вывода.



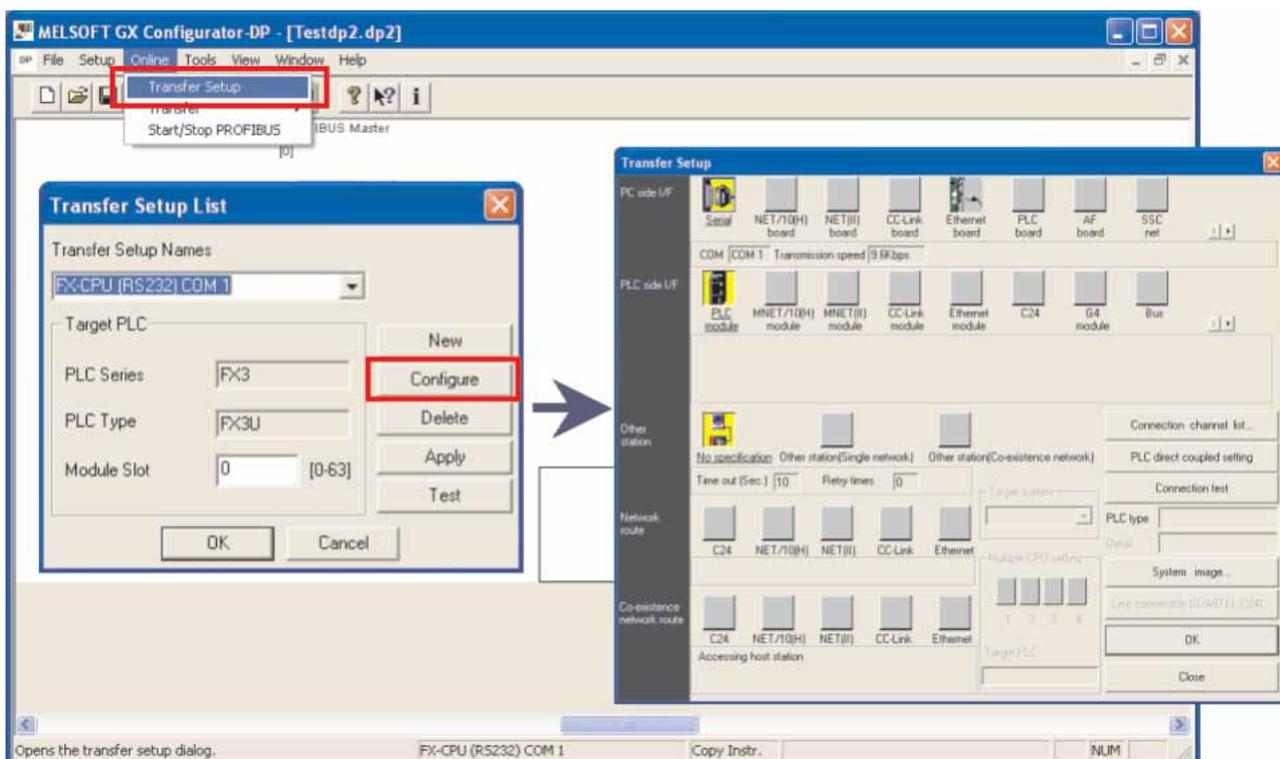
⑧ Передача данных для подчиненного устройства.



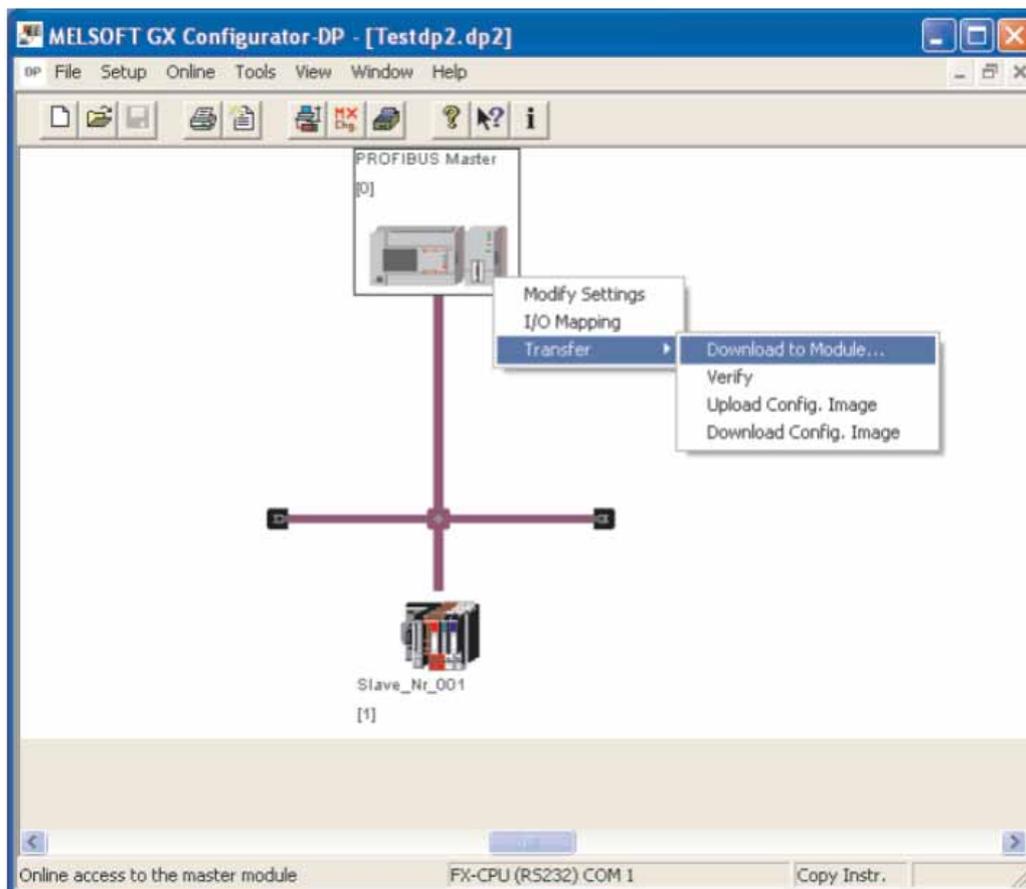
⑨ Распределение устройств ввода/вывода



⑩ Перед загрузкой, пожалуйста, выберите **Transfer Setup**.

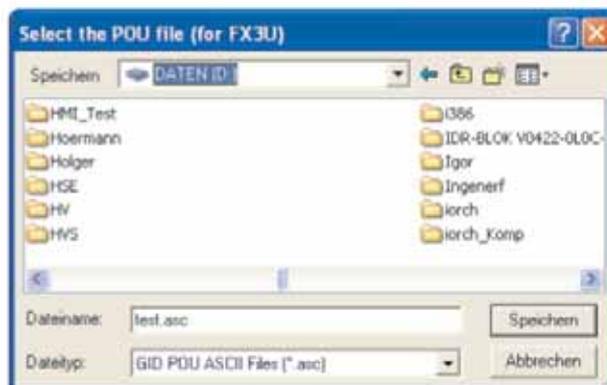
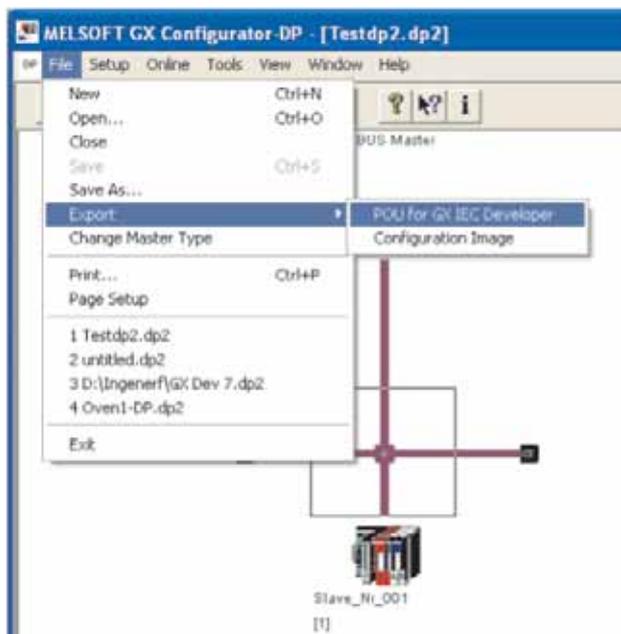


⑪ Перенос конфигурации в ведущий модуль PROFIBUS/DP.

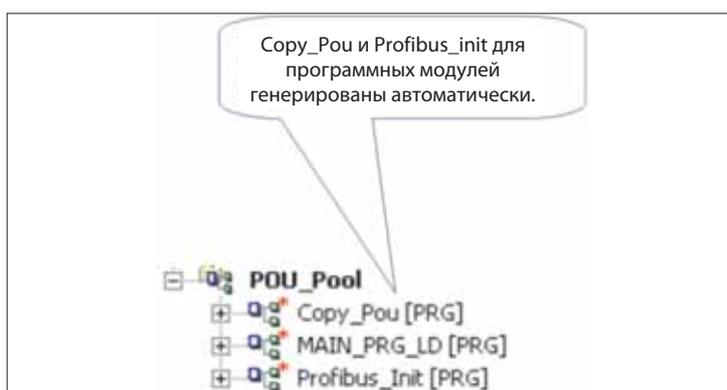
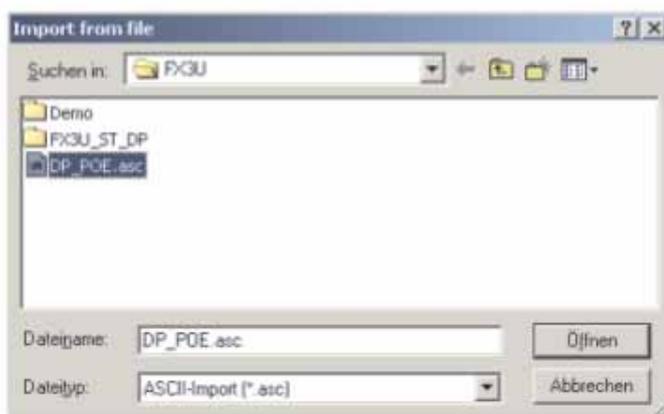
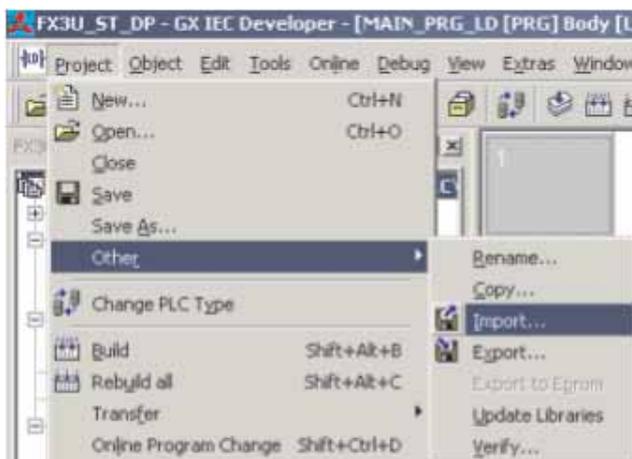


⑫ Программный модуль для GX IEC Developer

Созданный программный модуль может быть экспортирован в проект GX IEC Developer. Этот программный модуль инициализирует ведущий модуль PROFIBUS/DP в программе ПЛК.



- ⑬ Импорт программного модуля в проект GX IEC Developer.
(Новый проект с правильным ЦП уже был создан и сохранен.)



1	<pre> (* Exchange PLC data with Profibus DP *) (* Module Type FX3U-64DP-M: Mode 3 *) LD M8000 FROM_M K1,K5,K1,TEMP_WORD (* read profibus start ready flag *) WORD_TO_BOOL_E TEMP_WORD,DATA_EXCHANGE_FLAG LD DATA_EXCHANGE_FLAG FROM_M K1,K84,K4,DP_ARRAY_INPUT_CONSISTENCY_WORD[0] (* read input consistency flag *) FROM_M K1,K92,K4,DP_ARRAY_OUTPUT_CONSISTENCY_WORD[0] (* read output consistency flag *) WORD_TO_INT_E DP_ARRAY_OUTPUT_CONSISTENCY_WORD[0], DP_ARRAY_TEMP_INT[0] WORD_TO_INT_E DP_ARRAY_OUTPUT_CONSISTENCY_WORD[1], DP_ARRAY_TEMP_INT[1] WORD_TO_INT_E DP_ARRAY_OUTPUT_CONSISTENCY_WORD[2], DP_ARRAY_TEMP_INT[2] WORD_TO_INT_E DP_ARRAY_OUTPUT_CONSISTENCY_WORD[3], DP_ARRAY_TEMP_INT[3] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[0],K16,DP_ARRAY_OUTPUT_CONSISTENCY[0] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[1],K16,DP_ARRAY_OUTPUT_CONSISTENCY[16] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[2],K16,DP_ARRAY_OUTPUT_CONSISTENCY[32] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[3],K16,DP_ARRAY_OUTPUT_CONSISTENCY[48] WORD_TO_INT_E DP_ARRAY_INPUT_CONSISTENCY_WORD[0], DP_ARRAY_TEMP_INT[0] WORD_TO_INT_E DP_ARRAY_INPUT_CONSISTENCY_WORD[1], DP_ARRAY_TEMP_INT[1] WORD_TO_INT_E DP_ARRAY_INPUT_CONSISTENCY_WORD[2], DP_ARRAY_TEMP_INT[2] WORD_TO_INT_E DP_ARRAY_INPUT_CONSISTENCY_WORD[3], DP_ARRAY_TEMP_INT[3] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[0],K16,DP_ARRAY_INPUT_CONSISTENCY[0] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[1],K16,DP_ARRAY_INPUT_CONSISTENCY[16] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[2],K16,DP_ARRAY_INPUT_CONSISTENCY[32] INT_TO_BITARR_E DP_ARRAY_TEMP_INT[3],K16,DP_ARRAY_INPUT_CONSISTENCY[48] (* write output data after profibus start is possible *) (* Output data *) </pre>
2	<pre> LD DATA_EXCHANGE_FLAG AND DP_ARRAY_OUTPUT_CONSISTENCY[0] (* check if no data consistency *) </pre>

- ⑭ Скомпилируйте проект GX IEC Developer и передайте его в FX3U. После перезагрузки ПЛК начнется связь по PROFIBUS.

19 Связь по Ethernet

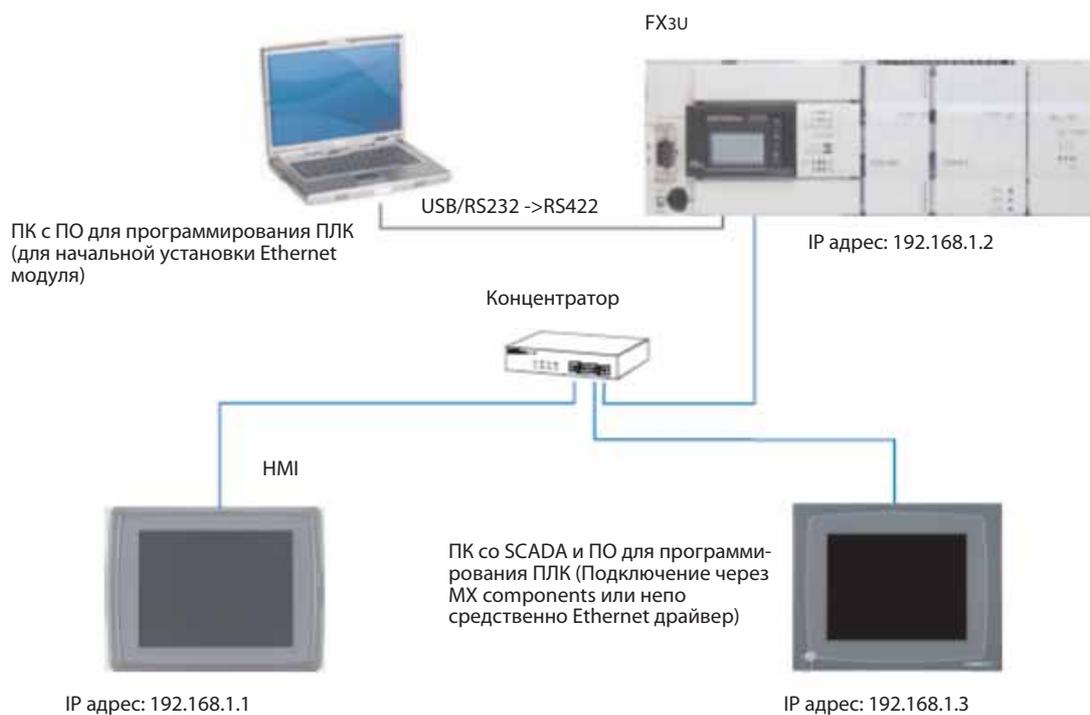
19.1 Конфигурирование параметров Ethernet модуля FX3U

В этом разделе представлены подробные инструкции по настройке Ethernet модуля FX3U-ENET (далее модуль) путем настройки параметров для GX Developer 8.00 или выше.

В качестве примера в этом разделе будет показано, как настроить модуль для организации TCP/IP связи между FX3U, SCADA ПК и HMI E1071. Также показано, как можно сконфигурировать программное обеспечение для программирования связи с FX3U через Ethernet после того, как выполнены настройки.

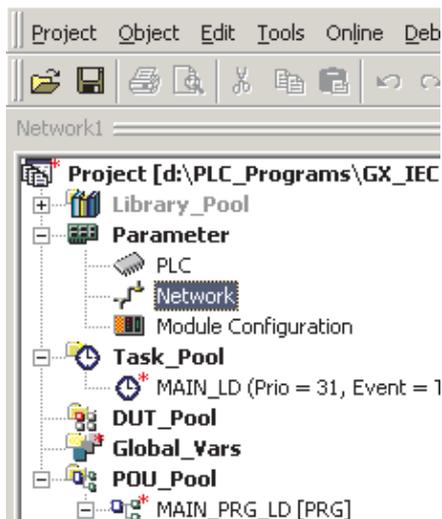
Ниже на диаграмме показана структура сети Ethernet. Предложенные IP-адреса показаны рядом с Ethernet узлами.

Учтите, что больше внимания уделено настройке ПЛК, нежели ПК или HMI, поскольку пользователю может потребоваться больше конкретных настроек, чем обсуждается в этом разделе.

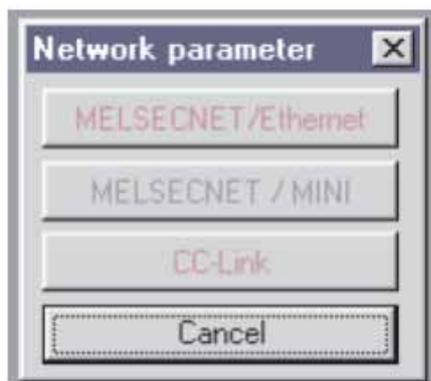


19.1.1 Конфигурирование ПЛК (с использованием ПК для начальной настройки)

- Используя программное обеспечение для программирования, вызовите окно выбора параметров **Network Parameter**, дважды щелкнув на опции, отмеченной стрелкой.



- Когда откроется окно, выберите **MELSECNET/Ethernet**, как показано ниже.



После этого откроется диалоговое окно для конфигурации модуля Ethernet, как показано ниже.

- В окне типов сети **Network type** щелкните на стрелке вниз, чтобы показать имеющиеся опции:

Module 1	
Network type	None ▼
Starting I/O No.	
Network No.	
Total stations	
Group No.	
Station No.	
Mode	▼

- ④ Ethernet является последней опцией в списке. Выберите ее, как показано ниже:

Module 1	
Network type	Ethernet
Starting I/O No.	MNET/H mode (Normal station) MNET/I/O mode (Control station)
Network No.	MNET/I/O mode (Normal station)
Total stations	MNET/H Stand by station MNET/H(Remote master)
Group No.	Ethernet
Station No.	
Mode	

Теперь в диалоговом окне показаны варианты настроек для модуля. Кнопки в нижней половине таблицы, отмеченные красным, предназначены для установки обязательных частей модуля; пурпурным маркированы опции, которые устанавливаются при необходимости.

Module 1	
Network type	Ethernet
Starting I/O No.	
Network No.	
Total stations	
Group No.	0
Station No.	
Mode	On line
	Operational settings
	Initial settings
	Open settings
	Router relay parameter
	Station No.<->IP information
	FTP Parameters
	E-mail settings
	Interrupt settings

- ⑤ Щелкните на ячейках в верхней половине таблицы и введите требуемые значения. В приведенной ниже таблице показаны настройки для FX3U в описанном ранее примере системы.

Module 1	
Network type	Ethernet
Starting I/O No.	0000
Network No.	1
Total stations	
Group No.	0
Station No.	2
Mode	On line
Operational settings	
Initial settings	
Open settings	
Router relay parameter	
Station No. <-> IP information	
FTP Parameters	
E-mail settings	
Interrupt settings	

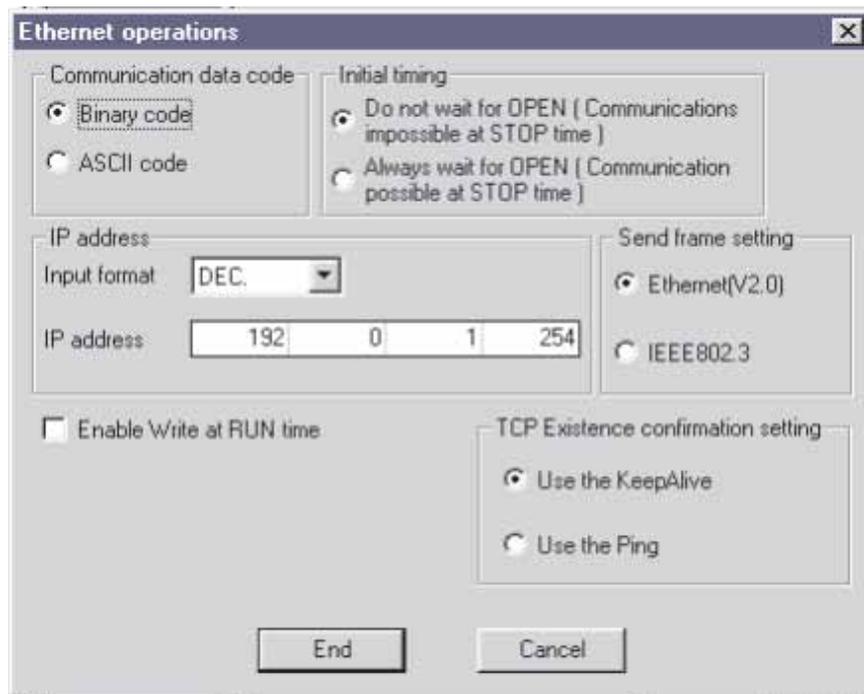
← См. замечание ниже

← См. замечание ниже

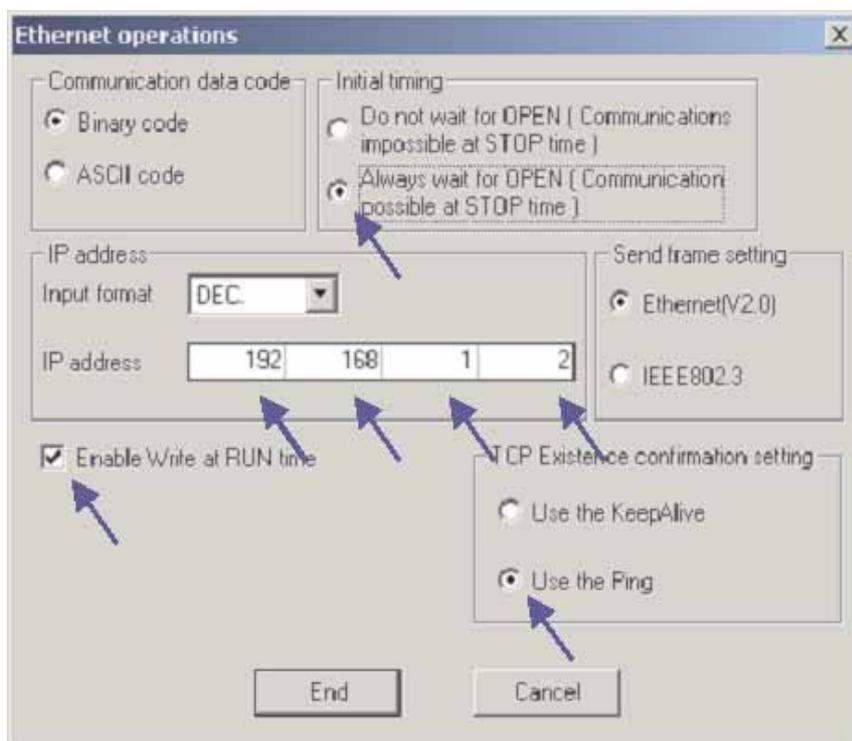
ПРИМЕЧАНИЕ

Настройки "номер сети" и "номер станции" используются для идентификации модуля, когда ПЛК FX3U использует Ethernet для одноранговой связи (не рассматривается в этом руководстве). Эти настройки также используются, когда ПО для программирования устанавливает связь с контроллером FX3U через сеть Ethernet. Этот вопрос рассматривается ниже в данном руководстве.

- ⑥ Далее щелкните на **Operational settings**, чтобы вызвать показанное ниже диалоговое окно. Имеющиеся настройки - это настройки по умолчанию, применяемые в программном обеспечении для программирования.



- ⑦ Ниже в диалоговом окне показаны настройки, необходимые для описанного ранее примера системы. Для ясности различия указаны стрелками.



- ⑧ Задав все настройки, щелкните **End**, чтобы вернуться в окно основных настроек сетевых параметров. Обратите внимание, что кнопка **Operational settings** стала синей, указывая на внесение изменений.

Module 1	
Network type	Ethernet
Starting I/O No.	0000
Network No.	1
Total stations	
Group No.	0
Station No.	2
Mode	On line
	Operational settings
	Initial settings
	Open settings
	Router relay parameter
	Station No. (↔) IP information
	FTP Parameters
	E-mail settings
	Interrupt settings

- ⑨ Далее щелкните на **Open settings** для вызова следующего диалогового окна. Здесь будут выполняться настройки для Scada и HMI.

ПРИМЕЧАНИЕ

Здесь не требуется ничего настраивать, если только Ethernet карта используется для контроля/редактирования программы с помощью программного обеспечения для программирования (как описано ниже).

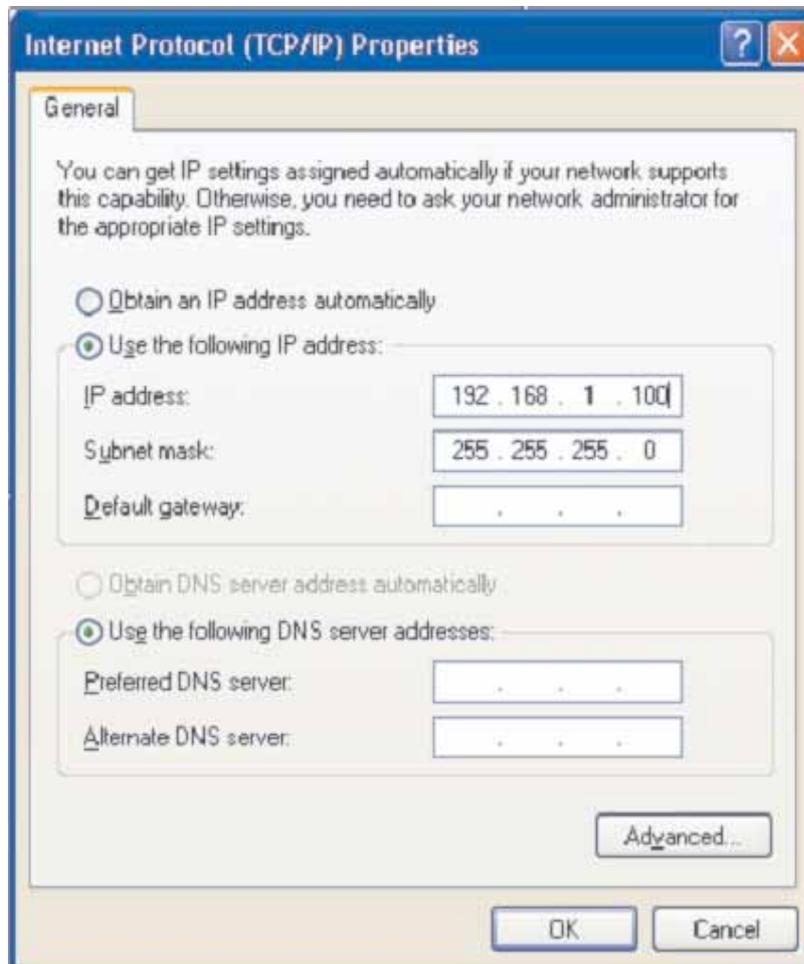
	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No.	Transmission target device IP address	Transmission target device Port No.
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

Ниже в диалоговом окне показаны настройки, необходимые для связи и со Scada и с HMI, для описанного ранее примера системы. Настройки выполняются путем выбора необходимых опций из открывающихся списков в каждом окне или, при необходимости, ввода значений.

	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No.	Transmission target device IP address	Transmission target device Port No.
1	TCP	Unpassive	Receive	Procedure exist	Disable	Confirm	0401	напр., HMI	
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

19.2 Конфигурирование ПК для Ethernet

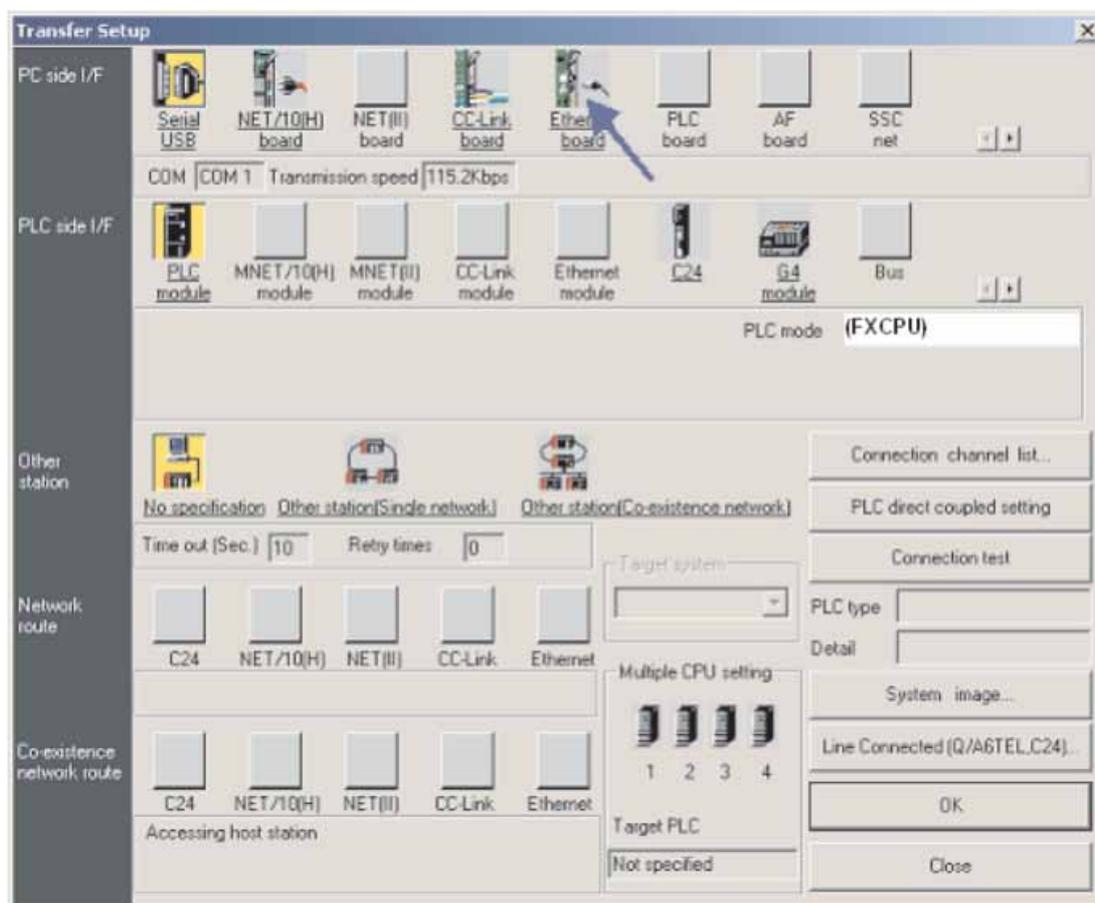
- ① Откройте Свойства сети в Windows и назначьте IP-адрес и маску подсети в диалоговом окне свойств **TCP/IP** для используемого сетевого адаптера Ethernet. Учтите, что после изменения IP-адреса может потребоваться перезагрузка ПК.



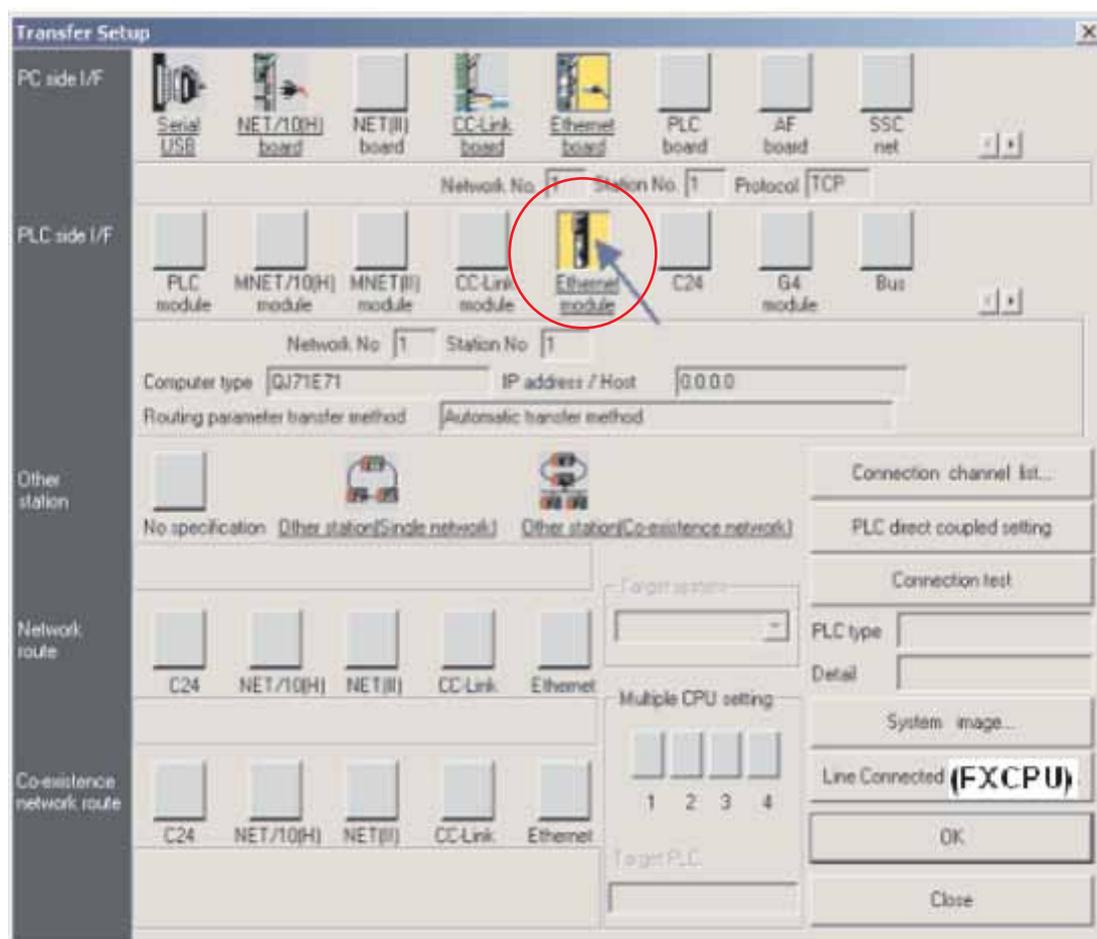
19.3 Конфигурирование GX IEC Developer для доступа контроллера в Ethernet

Для доступа GX IEC Developer к ПЛК через сеть Ethernet и модуль Ethernet вам понадобится конфигурировать следующие настройки:

- 1 Откройте диалоговое окно настроек, как показано ниже.



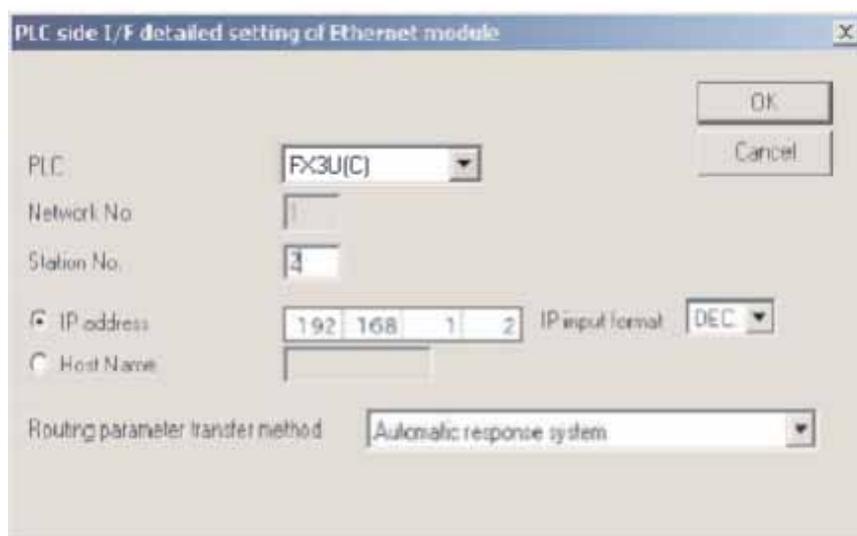
- 2 По умолчанию для соединения для **PC Side I/F** используется последовательная связь с модулем ЦП ПЛК. Измените **PC Side I/F** на **Ethernet board**, щелкнув на нем, как показано выше, и ответив **Yes** на запрос о потере имеющихся настроек (т.е. настроек последовательной связи с ЦП).
- 3 По умолчанию для **PC Side I/F** будет Номер сети = 1, Номер станции = 1 и Протокол = TCP, как показано выше. Если здесь этого НЕ ПОКАЗАНО, дважды щелкните на **Ethernet board** и сделайте такие настройки в соответствующих местах.



- ④ Далее дважды щелкните на **Ethernet module** в **PLC side I/F**, как показано выше. Откроется диалоговое окно, чтобы выбрать ПЛК для связи через Ethernet. Введите приведенные настройки, так как это были настройки, ранее введенные в ПЛК. (см. пункты ⑤ и ⑥ в разделе 19.1.1)

ПРИМЕЧАНИЕ

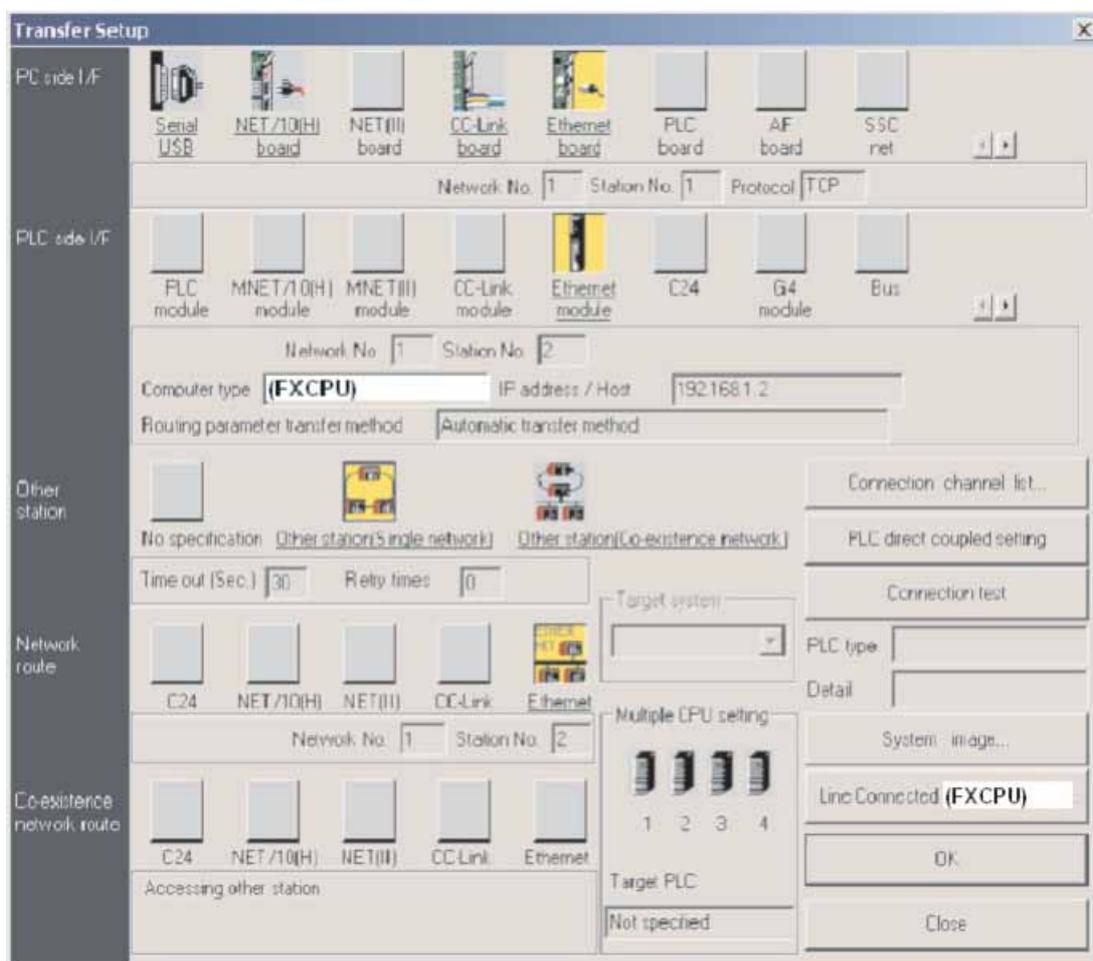
Не требуется указывать номер порта, поскольку в программном обеспечении для программирования используется протокол MELSOFT с назначением порта по умолчанию.



- ⑤ По завершению щелкните **OK**.

Далее один раз щелкните на Other station (Single network) , как показано ниже.

- ⑥ Это завершит установку, создав диалоговое окно, как показано ниже. Щелкните на **Connection test** для подтверждения, что установки заданы правильно. Затем по завершению щелкните на **OK**.

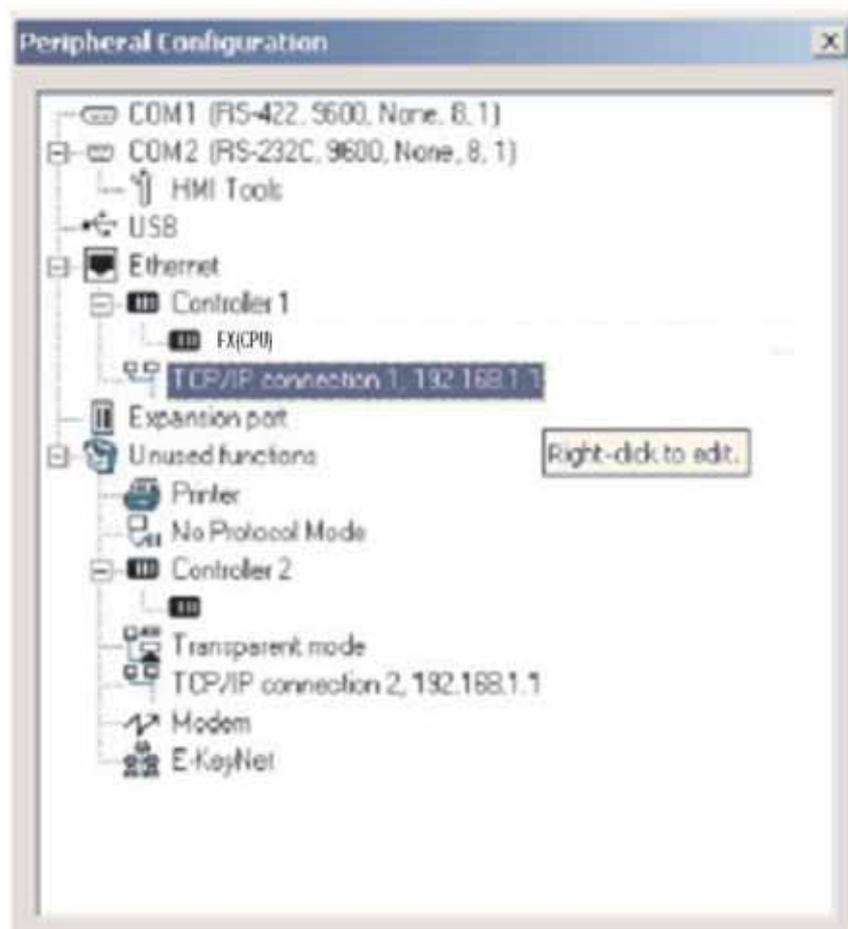


19.4 Установка интерфейса человек - машина (HMI)

- ① Проект в E-Designer для иллюстративной системы должен иметь следующие настройки.



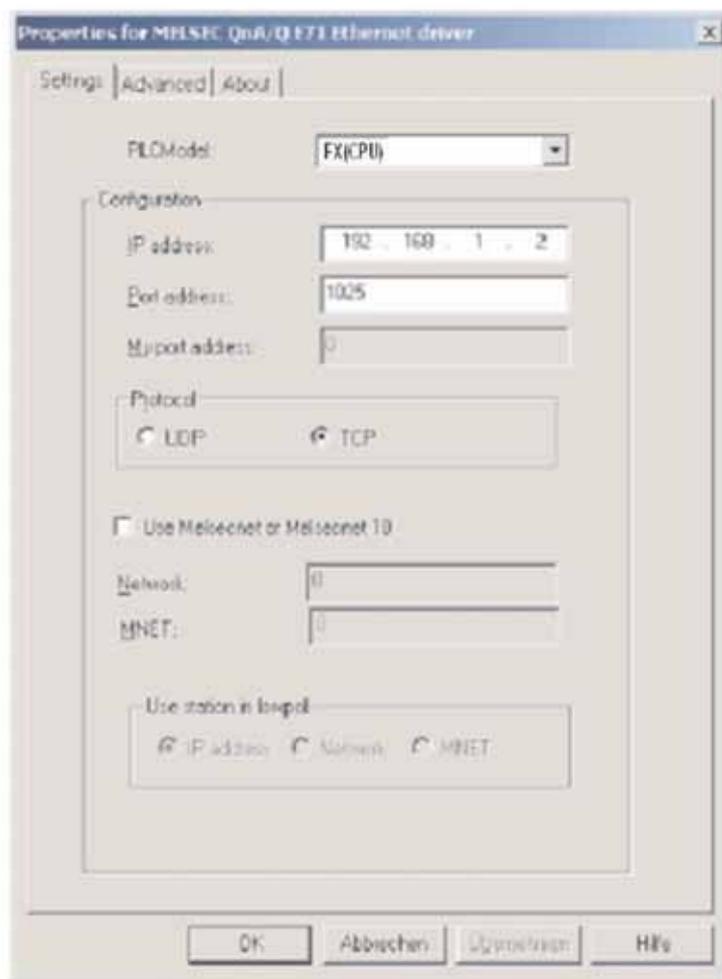
- ② Далее откройте опции **Peripherals** в системном меню и сконфигурируйте TCP/IP соединение для HMI, как показано:





- ③ Затем сделайте следующие настройки для "Controller 1" (т.е. целевой ПЛК), согласно настройкам, сделанным ранее в ПЛК.





Как для MQE настроек ранее, учтите, что номер E71 порта 1025, десятичное число 1025 равно шестнадцатеричному 401 (для установки номера порта локальной станции ПЛК см. часть 10 в разделе 19.1.1).

- ④ Щелкните на **OK**, выйдите из настроек периферийного оборудования и загрузите эти настройки с проектом.

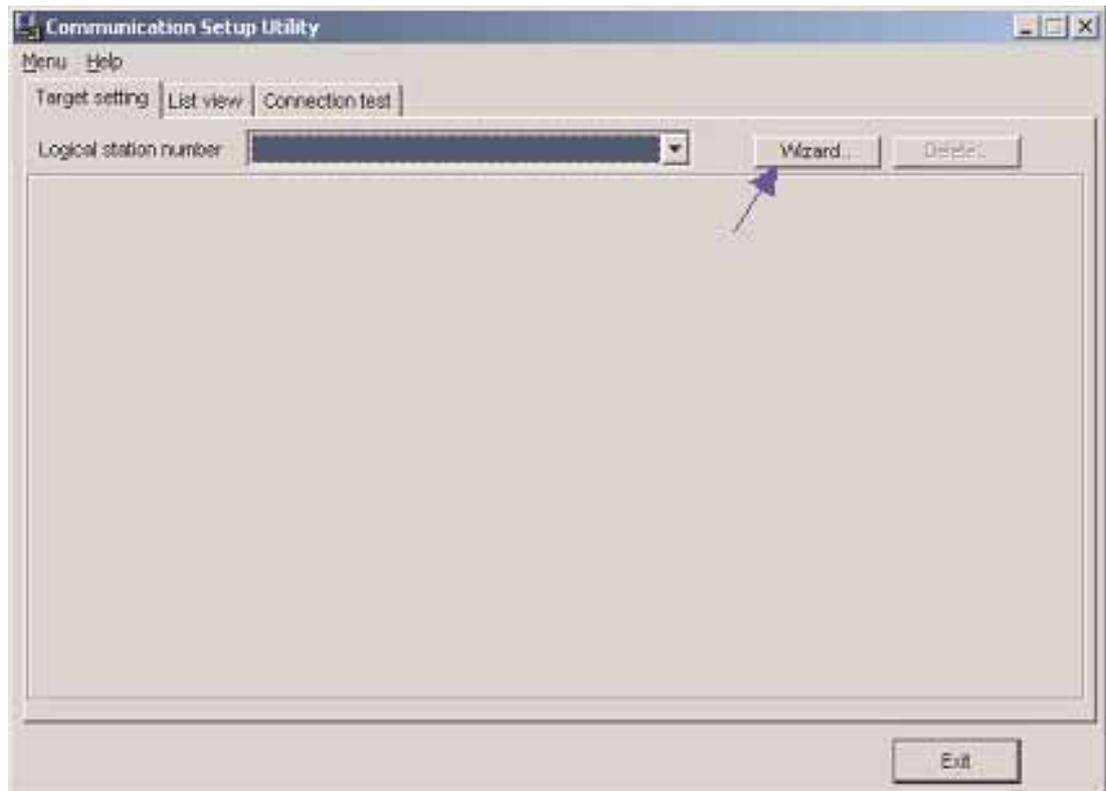
19.5 Связь через MX Component

MX Component - это инструмент, предназначенный для организации связи между ПК и ПЛК, не требующий от пользователя никаких знаний о протоколах связи и модулях.

Он поддерживает соединение через последовательный порт ЦП, соединения через последовательные каналы ПК (RS-232C, RS-422), сети Ethernet, CC-Link и MELSEC.

Ниже на рисунке показан простой способ создания связи между ПК и ПЛК через MX Component.

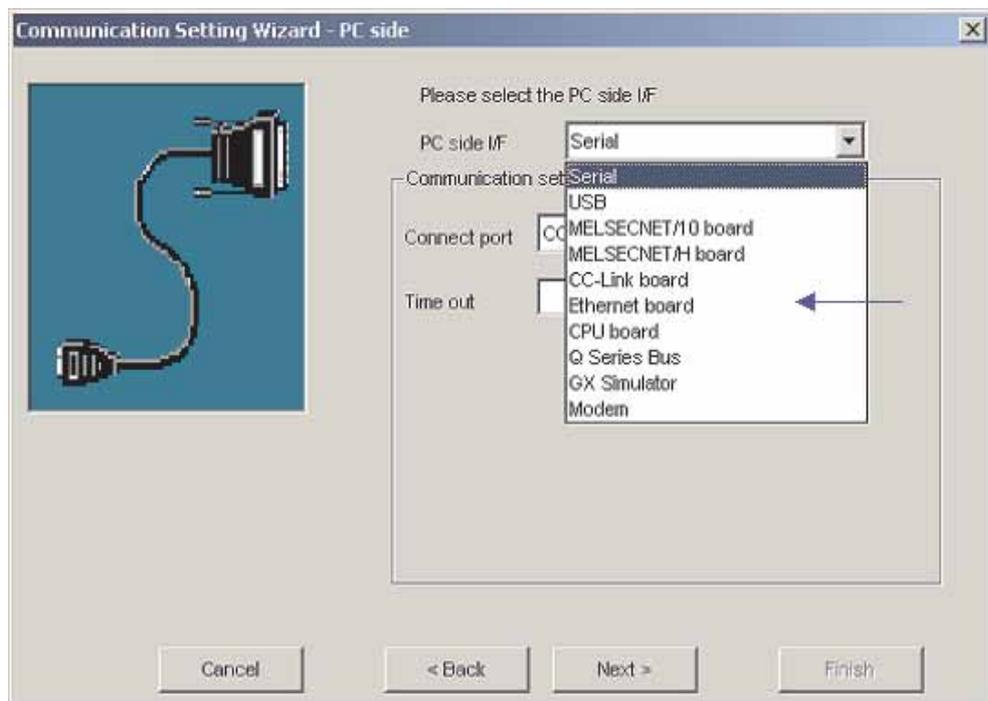
- ① Запустите **Communication Setting Utility** и выберите **Wizard**.



- ② Во-первых, необходимо определить **Logical station number**.



- ③ Затем сконфигурируйте **Communication Settings** на стороне ПК



- ④ Выберите UDP протокол и по умолчанию порт 5001

Communication Setting Wizard - PC side

Please select the PC side I/F

PC side I/F: Ethernet board

Communication setting

Connect module: (FXCPU)

Protocol: UDP

Network No: 1

Station No: 3

Port No: 5001

Time out: 60000 ms

Buttons: Cancel, < Back, Next >, Finish

- ⑤ Сконфигурируйте настройки связи на стороне ПЛК, необходимые для описанного ранее примера системы.

Communication Setting Wizard - PLC side

Please select the PLC side I/F

PLC side I/F: Ethernet module

Communication setting

Module type: (FXCPU)

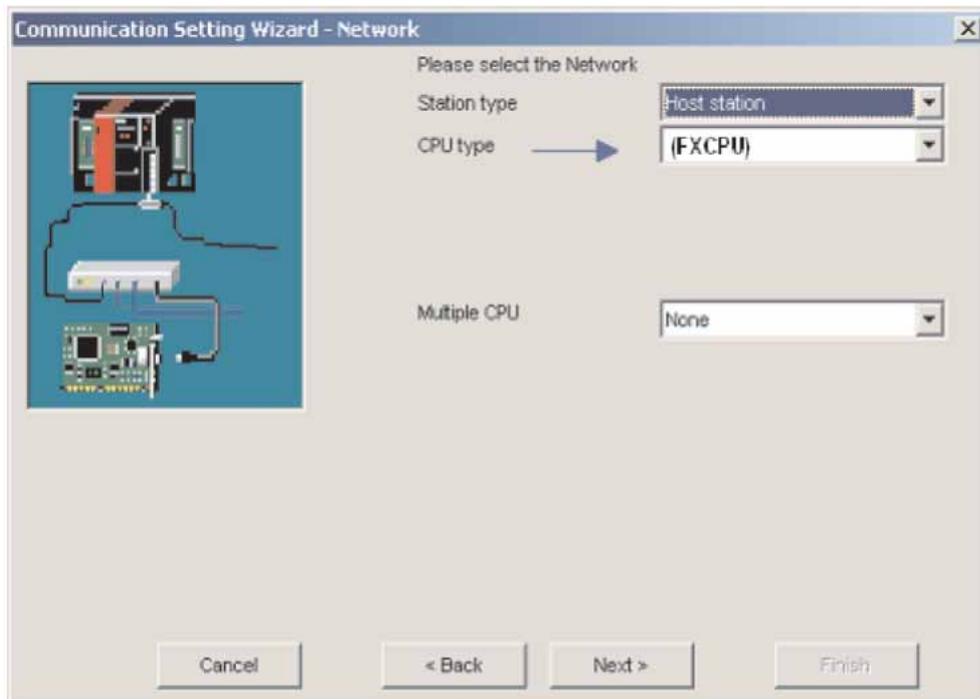
Host(IP Address): 192.168.1.2

Network No: 1

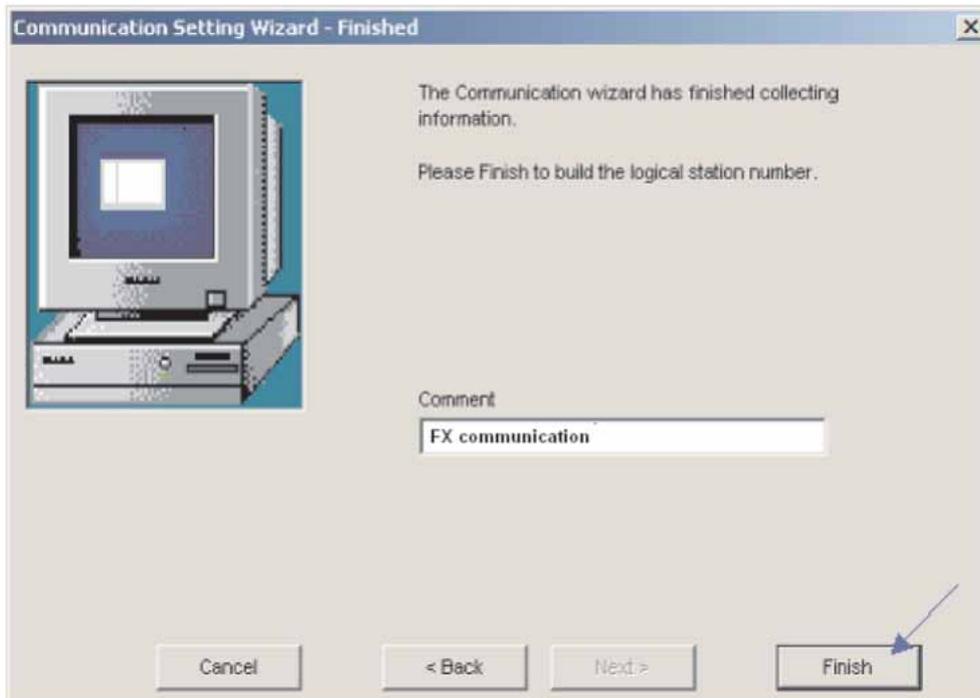
Station No: 2

Buttons: Cancel, < Back, Next >, Finish

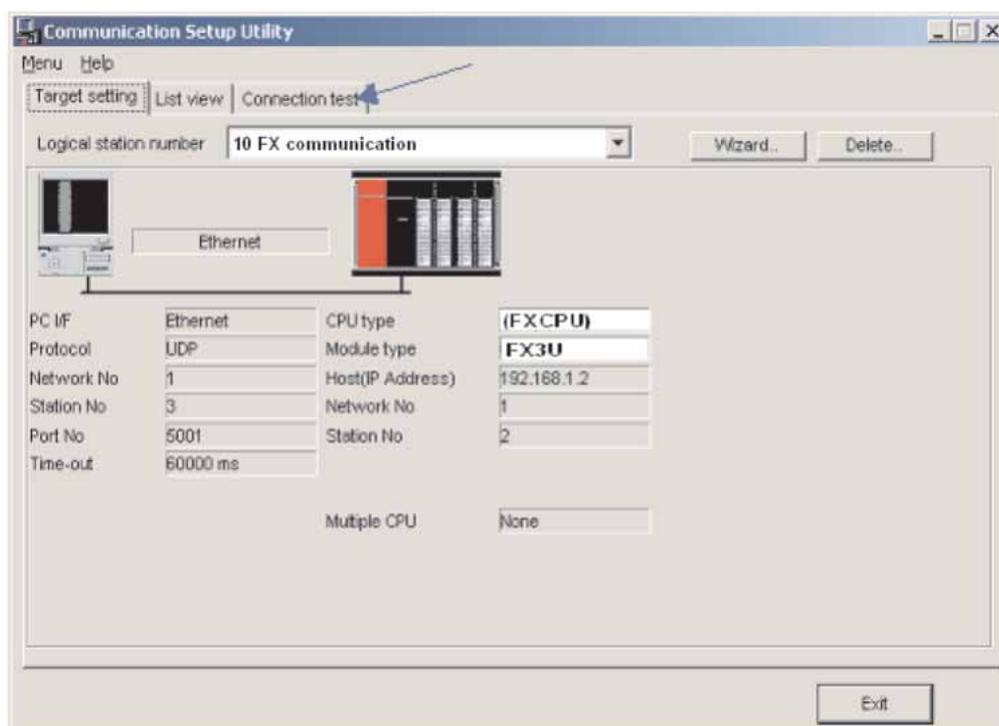
- ⑥ Выберите правильный тип ЦП.



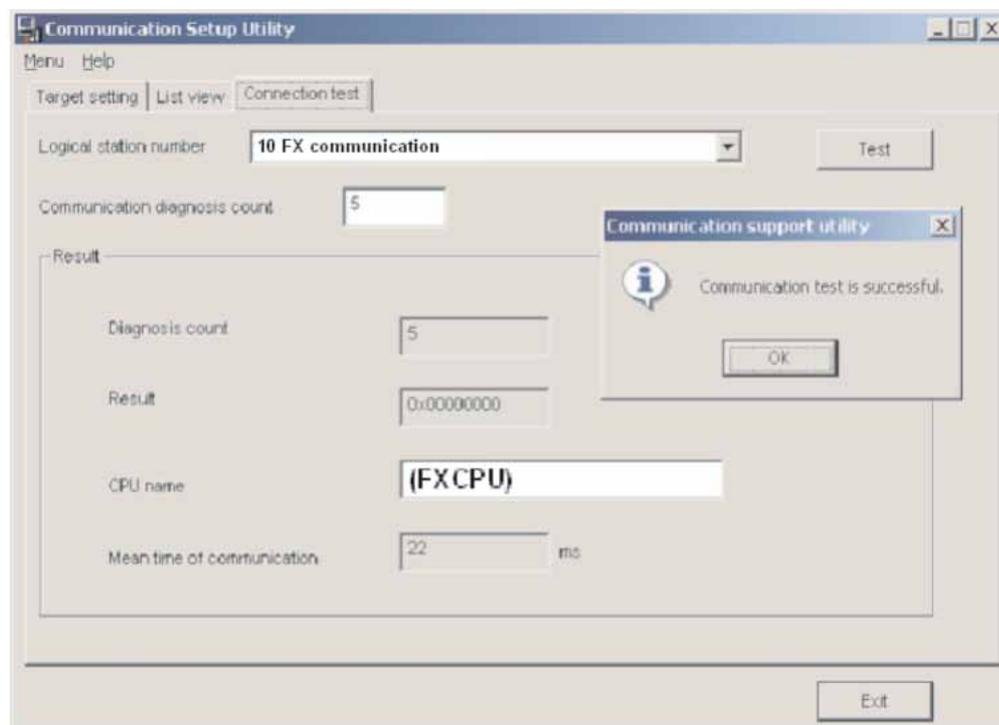
- ⑦ Для завершения конфигурации задайте имя и нажмите кнопку **Finish**.



Теперь определение связи закончено. Вы можете проверить соединение на вкладке **Connection test**.



Выберите **Logical station number**, для которого вы хотите выполнить тест. В **Diagnosis count** показывается, насколько успешно выполнено соединение. В **Result** показаны результаты тестирования. В случае ошибки указывается номер ошибки.



После конфигурирования каналов связи можно получить доступ ко всем контроллерным устройствам (считывание/запись) с помощью языков программирования Microsoft, таких как MS Visual Basic, MS C++ и т.д.

Описанные выше MX Component Mitsubishi - это эффективные, простые в использовании инструменты, облегчающие подключение контроллеров Мицубиси к миру ПК.

А Приложение

А.1 Специальные маркеры

Помимо маркеров, которые пользователь может произвольно включать и выключать в программе, существуют также специальные или диагностические маркеры. Эти маркеры занимают область, начиная с адреса M8000, и показывают определенные состояния системы или влияют на обработку программы. Специальные маркеры не могут использоваться, как другие внутренние маркеры в программном цикле. Однако некоторых из них можно установить или сбросить для управления ЦП. Здесь представлены некоторые, наиболее часто используемые операнды.

Специальные маркеры можно разделить на две группы:

- Специальные маркеры, сигнальные состояния которых можно только считывать программой (например, с помощью инструкции LD или LDI).
- Специальные маркеры, сигнальные состояния которых можно считывать и записывать программой (устанавливать или сбрасывать).

В следующих таблицах имеются колонки "Чтение" и "Запись". Если в одной из этих колонок показан символ "●", соответствующее действие возможно. Символ "-" означает, что соответствующее действие не допускается.

В ЦП серии FX имеются также специальные регистры для информации в формате слово. Они описываются в последующих разделах.

А.1.1 Диагностическая информация состояния ПЛК (M8000 - M8009)

Специальный маркер	Чтение	Запись	ЦП	Функция
M8000	●	—	FX1S FX1N FX2N FX2NC FX3U	Контроль RUN (нормально разомкнутый контакт)
M8001	●	—		Контроль RUN (нормально замкнутый контакт)
M8002	●	—		Импульс инициализации (нормально разомкнутый контакт)
M8003	●	—		Импульс инициализации (нормально замкнутый контакт)
M8004	●	—		Появление ошибки
M8005	●	—	FX2N FX2NC FX3U	Низкое напряжение батареи (Устанавливается, когда напряжение батареи ниже значения, установленного в D8006)
M8006	●	—		Самоблокировка ошибки батареи (M8006 устанавливается, когда обнаружено низкое напряжение батареи)
M8007	●	—		Кратковременный сбой питания
M8008	●	—		Обнаружен сбой питания
M8009	●	—		Отказ 24 В пост. тока (источник сервисного напряжения)

А.1.2 Таймеры и часы реального времени (M8011 - M8019)

Специальный маркер	Чтение	Запись	ЦП	Функция
M8010	—	—	—	Не используются
M8011	●	—	FX1S FX1N FX2N FX2NC FX3U	Тактовый импульс 10 мс ВКЛ и ВЫКЛ в цикле 10 мс (ВКЛ: 5 мс, ВЫКЛ: 5 мс)
M8012	●	—		Тактовый импульс 100 мс ВКЛ и ВЫКЛ в цикле 100 мс (ВКЛ: 50 мс, ВЫКЛ: 50 мс)
M8013	●	—		Тактовый импульс 1 с ВКЛ и ВЫКЛ в цикле 1 с (ВКЛ: 500 мс, ВЫКЛ: 500 мс)
M8014	●	—		Тактовый импульс 1 мин ВКЛ и ВЫКЛ в цикле 1 мин (ВКЛ: 30 с, ВЫКЛ: 30 с)
M8015	●	●		Остановка и настройка часов (для часов реального времени)
M8016	●	—		Останавливается вывод считанного времени (для часов реального времени) Содержимое D8013–D8019 замораживается, но часы все еще идут.
M8017	●	●		коррекция ±30 секунд (для часов реального времени)
M8018	●	—		Обнаружение установки часов реального времени (Всегда ВКЛ) Для FX2NC должна быть установлена карта памяти со встроенными часами реального времени (RTC).
M8019	●	—		Ошибка установки часов реального времени (RTC)

А.1.3 Режим работы ПЛК (M8030 - M8039)

Специальный маркер	Чтение	Запись	ЦП	Функция
M8030	●	—	FX2N FX2NC FX3U	Отключение светодиода индикации батареи Когда M8030 установлен, светодиод на ПЛК не горит, даже если обнаружено низкое напряжение батареи.
M8031	●	●	FX1S/ FX1N FX2N FX2NC FX3U	Очистка всей не фиксируемой памяти
M8032	●	●		Очистка всей фиксируемой памяти
M8033	●	●		Фиксация памяти в режиме STOP Когда ПЛК переключается из режима RUN в STOP, память изображений и память данных сохраняется.
M8034	●	●		Блокировка всех выходов Все внешние выходные контакты ПЛК переключаются в режим ВЫКЛ. Однако программа все еще выполняется.
M8035	●	●		Принудительный режим RUN
M8036	●	●		Принудительный сигнал RUN
M8037	●	●		Принудительный сигнал STOP
M8038	—	●		Флаг установки параметра связи (для установок сети N:N)
M8039	●	●		Режим постоянного времени цикла Когда M8039 установлен, ПЛК ожидает до времени цикла, заданного в D8039, и затем осуществляет циклическое выполнение.

А.1.4 Обнаружение ошибок (M8060 - M8069)

Специальный маркер	Чтение	Запись	ЦП	Функция
M8060	●	—	FX2N FX2NC FX3U	Ошибка конфигурации ввода-вывода
M8061	●	—	FX1S FX1N FX2N FX2NC FX3U	Аппаратная неисправность контроллера
M8062	●	—	FX2N FX2NC	Ошибка связи программирующего устройства/ПЛК
M8063 ^①	●	—	FX1S FX1N FX2N FX2NC FX3U	Ошибка последовательной передачи 1 [канал 1]
M8064	●	—		Ошибка параметров
M8065	●	—		Синтаксическая ошибка
M8066	●	—		Ошибка релейной диаграммы
M8067 ^②	●	—		Ошибка выполнения
M8068	—	●		Фиксация ошибки выполнения
M8069	—	●		FX2N FX2NC FX3U

① Действие меняется в зависимости от ПЛК: В FX1S, FX1N, FX2N, FX1NC или FX2NC сбрасывается при переключении ПЛК из режима STOP в RUN. Не сбрасываются в ПЛК FX3U.

② О сбое при последовательной связи по каналу 2 в контроллерах FX3U сигнализирует маркер M8438.

③ Сбрасывается, когда ПЛК переключается из режима STOP в RUN.

③ После установки M8069 выполняется проверка шины ввода-вывода. Если при этом распознается неисправность, в специальный регистр D8069 записывается код ошибки 6130 и устанавливается специальный маркер M8061.

А.1.5 Платы расширения (предназначенные для FX1S и FX1N)

Специальный маркер	Чтение	Запись	ЦП	Функция
M8112	●	●	FX1S FX1N	Плата расширения FX1N-4EX-BD: Вход ВХ0
				Плата расширения FX1N-2AD-BD: изменение режима входа канала 1
				Плата расширения FX1N-1DA-BD: изменение режима входа
M8113	●	●		Плата расширения FX1N-4EX-BD: Вход ВХ1
				Плата расширения FX1N-2AD-BD: изменение режима входа канала 2
				Плата расширения FX1N-4EX-BD: Вход ВХ2
M8114	●	●		Плата расширения FX1N-4EX-BD: Вход ВХ3
M8115	●	●		Плата расширения FX1N-2EYT-BD: Выход ВУ0
M8116	●	●		Плата расширения FX1N-2EYT-BD: Выход ВУ1
M8117	●	●		

А.1.6 Аналоговый специальный адаптер для FX3U (M8260 - M8299)

Специальный регистр	Чтение	Запись	ЦП	Функция
M8260 -M8269	●	●	FX3U	1-й* специальный адаптер
M8270 -M8279	●	●		2-ой* специальный адаптер
M8280 -M8289	●	●		3-ий* специальный адаптер
M8290 -M8299	●	●		4-ый* специальный адаптер

* Номер модуля аналогового специального адаптера отсчитывается со стороны главных блоков.

А.2 Специальные регистры

По аналогии со специальными маркерами (раздел А.1), начинающимися с адреса M8000, в контроллерах серии FX также имеются специальные или диагностические регистры, адреса которых начинаются с D8000. Часто имеется даже непосредственная взаимосвязь между специальным маркером и специальным регистром. Так, например, специальный маркер M8005 показывает, что напряжение батарейки контроллера слишком низкое, а специальный регистр D8005 содержит измеренное значение напряжения. Некоторая небольшая часть специальных регистров приведена в следующих таблицах.

Специальные регистры можно разделить на две группы:

- Специальные регистры, значения данных которых можно только считывать программой
- Специальные маркеры, значения данных которых можно считывать и записывать программой.

В следующих таблицах имеются колонки "Чтение" и "Запись". Если символ "●" показан в одной из этих колонок, соответствующее действие возможно. Символ "-" означает, что соответствующее действие не допускается.

А.2.1 Диагностическая информация состояния ПЛК (D8000 - D8009)

Специальный регистр	Чтение	Запись	ЦП	Функция
D8000	●	●		Настройка сторожевого таймера (шаг 1 мс). (Записывается из системного ПЗУ при включении) Значение, перезаписанное в программе, действительно после выполнения инструкции END или WDT. Значение должно быть больше, чем время цикла (записанное в D8012). Значение по умолчанию 200 мс.
D8001	●	—		Тип ПЛК и версия системы FX1S: 22V _W FX1N: 26V _W FX2N/FX2NC/FX3U: 24V _W (например, FX1N Версия 1.00 → 26100)
D8002	●	—	FX1S FX1N FX2N FX2NC FX3U	Объем памяти 0002 (r) 2 кШагов (только FX1S) 0004 (r) 4 кШагов (только FX2N/FX2NC) 0008 (r) 8 кШагов или более (не для FX1S) При 16 кШагов или более в D8002 записывается "K8" и в D8102 записывается "16" или "64".
D8003	●	—		Тип памяти: 00 _H → RAM (Кассета памяти) 01 _H → EPROM (Кассета памяти) 02 _H → EEPROM (Кассета памяти или флеш-память) 0A _H → EEPROM (Кассета памяти или флеш-память), защита от записи) 10 _H → Встроенная память в ПЛК
D8004	●	—		Номер ошибки (M) Если D8004 содержит, например, значение 8060, установлен специальный маркер M8060.
D8005	—	—		Напряжение батареи (Пример: "36" → 3.6 В)
D8006	—	—	FX2N FX2NC FX3U	Уровень обнаружения низкого напряжения батареи. Настройки по умолчанию: FX2N/FX2NC: 3.0 В ("30") FX3U: 2.7 В ("27")
D8007	—	—		Количество кратковременных отключений напряжения Рабочая частота M8007 сохраняется. Очищается при выключении питания.
D8008	—	—	FX2N FX2NC FX3U	Обнаружение сбоя питания. Настройки по умолчанию: FX2N/FX3U: 10 мс (БП переменного напряжения) FX2NC: 5 мс (электропитание постоянного напряжения)
D8009	—	—	FX2N FX2NC FX3U	Сбой устройства 24 В пост. тока Минимальный номер входного устройства блоков расширения и блоков питания расширения, в которых произошел сбой 24 В=.

А.2.2 Информация цикла сканирования и часы реального времени (D8010 - D8019)

Специальный регистр	Чтение	Запись	ЦП	Функция
D8010	●	—	FX1S FX1N FX2N FX2NC FX3U	Текущее время цикла программы (шаг 0.1 мс)
D8011	●	—		Минимальное значение времени цикла (шаг 0.1 мс)
D8012	●	—		Максимальное значение времени цикла (шаг 0.1 мс)
D8013	●	●	FX1S FX1N FX2N FX2NC FX3U	Часы реального времени: Секунды (0–59)
D8014	●	●		Часы реального времени: Минуты (0–59)
D8015	●	●		Часы реального времени: Часы (0–23)
D8016	●	●		Часы реального времени: Дата (День, 1–31)
D8017	●	●		Часы реального времени: Дата (Месяц, 1–12)
D8018	●	●		Часы реального времени: Дата (Год, 0–99)
D8019	●	●		Часы реального времени: День недели (0 (Воскресенье) - 6 (Суббота))

А.2.3 Режим работы ПЛК (D8030 - D8039)

Специальный регистр	Чтение	Запись	ЦП	Функция
D8030	●	—	FX1S FX1N	Значение от аналоговой величины VR1 (Целое число от 0 до 255)
D8031	●	—		Значение аналоговой величины VR2 (Целое число от 0 до 255)
D8032 – D8038	—	—	—	Не используются
D8039	—	●	FX1S FX1N FX2N FX2NC FX3U	Постоянная продолжительность цикла: По умолчанию: 0 мс (с шагом 1 мс) (Записывается из системного ПЗУ при включении) Можно перезаписывать в программе

А.2.4 Коды ошибок (D8060 – D8069)

Специальный регистр	Чтение	Запись	ЦП	Функция
D8060	●	—	FX2N FX2NC FX3U	Если модуль или блок, соответствующий запрограммированному номеру входа-выхода, в действительности не загружен, устанавливается M8060 и номер первого операнда ошибочного блока записывается в D8060 Значение четырехразрядного кода: 1-ая цифра: 0 = Выход, 1 = Вход, 2-4 цифра: Номер первого операнда ошибочного блока
D8061	●	—	FX1S FX1N FX2N FX2NC FX3U	Код ошибки для аппаратной неисправности контроллера
D8062	●	—	FX2N FX2NC FX3U	Код ошибки для ошибки связи ПЛК/программатор
D8063	●	—	FX1S FX1N FX2N FX2NC FX3U	Код ошибки для ошибки последовательной передачи данных 1 [канал 1]
D8064	●	—		Код ошибки для ошибки параметров
D8065	●	—		Код синтаксической ошибки программы
D8066	●	—		Код ошибки для ошибки релейной диаграммы
D8067	●	—		Код ошибки для ошибки выполнения
D8068*	—	●		Фиксация номера шага ошибки выполнения В случае 32 kШагов или более номер шага хранится в [D8313, D8312].
D8069*	●	—		Номер шага ошибки от M8065 до M8067 В случае 32 kШагов или более адрес шага хранится в [D8315, D8314].

* Сбрасывается, когда ПЛК переключается из режима STOP в RUN.

А.2.5 Платы расширения (предназначенные для FX1S и FX1N)

Специальный регистр	Чтение	Запись	ЦП	Функция
D8112	●	—	FX1S FX1N	Адаптер FX1N-2AD-BD: Значение цифрового входа, канал 1
D8113	●	—		Адаптер FX1N-2AD-BD: Значение цифрового входа, канал 2
D8114	●	●		Адаптер FX1N-1DA-BD: Значение цифрового выхода, канал 1

А.2.6 Аналоговый специальный адаптер для FX3U (D8260 - D8299)

Специальный регистр	Чтение	Запись	ЦП	Функция
D8260 – D8269	●	●	FX3U	1-й* специальный адаптер
D8270 – D8279	●	●		2-ой* специальный адаптер
D8280 – D8289	●	●		3-ий* специальный адаптер
D8290 – D8299	●	●		4-ый* специальный адаптер

* Номер модуля аналогового специального адаптера отсчитывается со стороны главных блоков.

А.3 Список кодов ошибок

При обнаружении ошибки в ПЛК код ошибки записывается в специальные регистры D8060 – D8067 и D8438. Для диагностических ошибок должны выполняться следующие действия.

Здесь представлены некоторые, наиболее часто встречаемые ошибки.

А.3.1 Коды ошибок 6101 - 6409

Ошибка	Специальный регистр	Код ошибки	Описание	Устранение
Аппаратная неисправность контроллера	D8061	0000	Безошибочно	—
		6101	Ошибка RAM	
		6102	Ошибка выполнения программы	
		6103	Ошибка шины ввода-вывода (M8069 = ВКЛ)	Проверьте правильность соединения кабелей расширения.
		6104	Сбой модуля расширения с блоком питания 24 В (M8069 = ВКЛ)	
		6105	Ошибка сторожевого таймера	Проверьте прикладную программу. Время цикла превышает значение, сохраненное в D8000.
		6106	Ошибка при создании таблицы ввода-вывода (ошибка CPU) При включении питания на главном блоке, в питаемых блоках расширения происходит сбой питания 24 В. (Ошибка возникает, если питание 24 В не подается в течение 10 секунд или более после включения основного питания.)	Проверьте электропитание для модулей расширения с собственным блоком питания.
		6107	Ошибка конфигурации системы	Проверьте количество подключенных специальных функциональных блоков/модулей. Для некоторых специальных функциональных блоков/модулей ограничено количество подключаемых устройств.
Ошибка связи между ПЛК и программирующим устройством (Только FX2N и FX2NC)	D8062	0000	Безошибочно	—
		6201	Ошибка контроля по четности, превышения скорости передачи или синхронизации кадров	Проверьте кабельное подключение между программатором и ПЛК. Эта ошибка может возникать, когда кабель отсоединен и повторно подсоединен в ходе мониторинга ПЛК.
		6202	Ошибка передачи символа	
		6203	Ошибка контрольной суммы при передаче данных	
		6204	Ошибка формата данных	
		6205	Ошибка инструкции	
		0000	Безошибочно	
Ошибка последовательной передачи	D8063	6301	Ошибка контроля по четности, превышения скорости передачи или синхронизации кадров	<ul style="list-style-type: none"> ● Связь с инвертором, линия связи с компьютером и программирование: Проверьте, что параметры связи правильно установлены, согласно их приложениям. ● Сеть N:N, parallel link, и т.п.: Проверьте программы согласно приложениям. ● Дистанционное техническое обслуживание: Убедитесь, что модем включен и проверьте настройки AT-команд. ● Соединение: Проверьте, что кабели связи корректно проложены.
		6302	Ошибка передачи символа	
		6303	Ошибка контрольной суммы при передаче данных	
		6304	Ошибка формата данных связи	
		6305	Ошибка инструкции	
		6306	Обнаружена задержка связи	
		6307	Ошибка инициализации модема	
		6308	Ошибка параметров сети N:N	
		6312	Неправильный символ Parallel Link	
		6313	Ошибка суммы Parallel Link	
		6314	Ошибка формата Parallel Link	
		6320	Ошибка связи с инвертором	

Ошибка	Специальный регистр	Код ошибки	Описание	Устранение
Ошибка параметров	D8064	0000	Безошибочно	— ПереклЮчите ПЛК в режим STOP, и корректно установите параметры.
		6401	Ошибка контрольной суммы программы	
		6402	Ошибка установки объема памяти	
		6403	Ошибка установки фиксируемой области операнда	
		6404	Ошибка установки области комментария	
		6405	Ошибка установки области файлового регистра	
		6406	Ошибка контрольной суммы настроек начальных значений специального блока (BFM), настроек инструкций позиционирования	
		6407	Ошибка настроек начальных значений специального блока (BFM), настроек инструкций позиционирования	
		6409	Другие ошибочные установки	

A.3.2 Коды ошибок 6501 - 6510

Ошибка	Специальный регистр	Код ошибки	Описание	Устранение
Синтаксическая ошибка	D8065	0000	Без ошибок	В ходе программирования проверяется каждая инструкция. Если обнаружена синтаксическая ошибка, исправьте инструкцию.
		6501	Неправильная комбинация инструкций, символов операндов и номеров операндов	
		6502	Отсутствует OUT T или OUT C перед заданием значения	
		6503	— Отсутствует OUT T или OUT C перед заданием значения — Несоответствующее количество операндов для прикладной инструкции	
		6504	— Один и тот же номер метки используется более одного раза. — Один и тот же вход прерываний или вход высокоскоростного счетчика используется более одного раза.	
		6505	Номер операнда вне допустимого диапазона.	
		6506	Неправильная инструкция	
		6507	Неправильный номер метки [P]	
		6508	Неправильный вход прерываний [I]	
		6509	Другая ошибка	
		6510	Ошибка количества вложений MC	

А.3.3 Коды ошибок 6610 - 6632

Ошибка	Специальный регистр	Код ошибки	Описание	Устранение
Ошибка схемы	D8066	0000	Безошибочно	—
		6610	LD, LDI непрерывно используются 9 или более раз.	Эта ошибка возникает, когда комбинация инструкций некорректна во всем блоке или при неправильной связи между парой инструкций. Измените инструкции в программном режиме, так чтобы их взаимосвязь стала корректной.
		6611	Инструкций ANB/ORB больше, чем LD/LDI	
		6612	Инструкций ANB/ORB меньше, чем LD/LDI	
		6613	MPS непрерывно используются 12 или более раз.	
		6614	Нет инструкции MPS	
		6615	Нет инструкции MPP	
		6616	Нет катушки между MPS, MRD и MPP, или неправильная комбинация	
		6617	Инструкции ниже не связаны с линией шины: STL, RET, MCR, P, I, DI, EI, FOR, NEXT, SRET, IRET, FEND или END	
		6618	STL, MC или MCR могут использоваться только в основной программе, но они используются где-то в другом месте (например, в программе обработки прерываний или подпрограмме).	
		6619	Неправильная инструкция используется в петле FOR - NEXT: STL, RET, MC, MCR, I (указатель прерывания) или IRET.	
		6620	Для инструкции FOR - NEXT превышен уровень вложения	
		6621	Количество инструкций FOR и NEXT не совпадает.	
		6622	Нет инструкции NEXT	
		6623	Нет инструкции MC	
		6624	Нет инструкции MCR	
		6625	Инструкция STL непрерывно используется 9 или более раз.	
		6626	Неправильная инструкция запрограммирована в петле STL-RET: MC, MCR, I (указатель прерывания), SRET или IRET.	
		6627	Нет инструкции RET	
		6628	Неправильная инструкция используется в основной программе: I (указатель прерывания), SRET или IRET	
6629	Нет P или I (указатель прерывания)			
6630	Нет инструкции SRET или IRET			
6631	SRET запрограммирована в неправильном месте			
6632	FEND запрограммирована в неправильном месте			

А.3.4 Коды ошибок 6701 - 6710

Ошибка	Специальный регистр	Код ошибки	Описание	Устранение
Ошибка выполнения	D8067	0000	Безошибочно	—
		6701	<ul style="list-style-type: none"> — Для инструкции SJ или CALL не указана точка перехода (указатель) — Метка не определена или вне P0 - P4095 из-за индексации — Метка P63 выполняется в инструкции CALL; не может использоваться в инструкции CALL, так как P63 используется для перехода к инструкции END. 	Эта ошибка происходит при выполнении операции. Пересмотрите программу, или проверьте содержимое операндов, используемых в прикладных инструкциях.*
		6702	Уровень вложения инструкции CALL 6 или выше	
		6703	Уровень вложения прерывания 3 или выше	
		6704	Уровень вложения инструкции FOR - NEXT 6 или выше.	
		6705	В прикладной инструкции неприменимый операнд.	
		6706	Диапазон операндов или значение данных для операнда прикладной инструкции вне допустимой области.	
		6707	Доступ к файловому регистру без установки параметра файлового регистра.	
		6708	Ошибка инструкции FROM/TO	Эта ошибка происходит при выполнении операции. Пересмотрите программу, или проверьте содержимое операндов, используемых в прикладных инструкциях. Проверьте, что в оборудовании имеется указанная буферная память. Проверьте, что кабели расширения правильно соединены.
		6709	Другая (например, недопустимое ответвление)	Эта ошибка происходит при выполнении операции. Пересмотрите программу, или проверьте содержимое операндов, используемых в прикладных инструкциях.*
		6710	Несоответствие между параметрами	Эта ошибка возникает, когда один и тот же операнд указан как источник и цель в команде сдвига, и т.п.

* Даже при корректном синтаксисе и схемном решении могут возникать ошибки выполнения. Например: "T200Z" само по себе не ошибка. Но если Z имело значение 400, возможно, что была неудачная попытка доступа к таймеру T600. Это могло вызвать ошибку выполнения, поскольку устройство T600 недоступно.

A.4 Количество занятых точек ввода-вывода и потребление тока

В следующих таблицах показано, сколько точек ввода-вывода занимают определенные модули в базовом блоке, а также тип блока питания и величины потребления тока, необходимые для выбранного продукта.

Потребляемый ток определяется по-разному в следующих случаях.

Напряжения 5 В пост. тока и внутреннее 24 В пост. тока подаются на продукты через кабель расширения, и необходимо рассчитывать потребляемый ток

Расчет потребляемого тока при внутреннем напряжении 24 В пост. тока выполняется следующим образом.

- Для базового блока с питанием от переменного напряжения вычитите потребляемый ток в линии внутр. напряжении 24 В пост. тока из нагрузки источника сервисного напряжения 24 В пост. тока.
- Для базового блока с питанием от постоянного напряжения, вычитите потребляемый ток в линии внутр. 24 В пост. тока из нагрузки блока электропитания для внутр. 24 В пост. тока.
- Некоторым специальным функциональным модулям требуется "внешнее напряжение 24 В пост. тока". Включите этот ток в расчет энергопотребления, когда ток подается от источника сервисного напряжения 24 В пост. тока. Когда ток поставляется от внешнего блока питания, он не включается в расчет энергопотребления.

A.4.1 Интерфейсные адаптерные платы и платы коммуникационных адаптеров

Тип	Кол-во занятых точек ввода-вывода	Потребляемый ток [мА]		
		5 В пост. тока	24 В пост. тока (внутр.)	24 В пост. тока (внешн.)
FX1N-232-BD	—	20	—	—
FX2N-232-BD	—			
FX3U-232-BD	—			
FX1N-422-BD	—	60*	—	—
FX2N-422-BD	—			
FX3U-422-BD	—			
FX1N-485-BD	—	60	—	—
FX2N-485-BD	—			
FX3U-485-BD	—			
FX3U-USB-BD	—	15	—	—
FX1N-CNV-BD	—	—	—	—
FX2N-CNV-BD				
FX3U-CNV-BD				

* Когда подключены средство разработки приложений или GOT, добавьте ток, потребляемый этим блоком (см. следующую страницу)

А.4.2 Средство разработки приложений, преобразователь интерфейса, дисплейный модуль и GOT

Тип	Кол-во занятых точек ввода-вывода	Потребляемый ток [мА]		
		5 В пост. тока	24 В пост. тока (внутр.)	24 В пост. тока (внешн.)
FX-20P(-E)	—	150	—	—
FX-232AWC-H	—	120	—	—
FX-USB-AW	—	15	—	—
FX3U-7DM	—	20	—	—
FX10DM-E	—	220	—	—
F920GOT-BBD5-K-E	—	220	—	—

А.4.3 Специальные адаптеры

Тип	Кол-во занятых точек ввода-вывода	Потребляемый ток [мА]			
		5 В пост. тока	24 В пост. тока (внутр.)	24 В пост. тока (внешн.)	При старте системы
FX3U-4HSX-ADP	—	30	30	0	30*
FX3U-2HSY-ADP	—	30	60	0	120*
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	—
FX2NC-232ADP	—	100	0	0	—
FX3U-232ADP	—	30	0	0	—
FX3U-485ADP	—	20	0	0	—

* При подключении к базовому блоку с источником питания постоянного тока необходимо учитывать потребляемый ток при старте системы.

А.4.4 Блоки расширения

Тип	Кол-во занятых точек ввода-вывода	Потребляемый ток [мА]		
		5 В пост. тока	24 В пост. тока (внутр.)	24 В пост. тока (внешн.)
FX2N-8ER-ES/UL	16	—	125	0
FX2N-8EX-ES/UL	8	—	50	0
FX2N-16EX-ES/UL	16	—	100	0
FX2N-8EYR-ES/UL	8	—	75	0
FX2N-8EYT-ESS/UL	8	—	75	0
FX2N-16EYR-ES/UL	16	—	150	0
FX2N-16EYT-ESS/UL	16	—	150	0

А.4.5 Специальные функциональные модули

Тип	Кол-во занятых точек ввода-вывода	Потребляемый ток [мА]			
		5 В пост. тока	24 В пост. тока (внутр.)	24 В пост. тока (внешн.)	При старте системы
FX3U-4AD	8	110	0	90	—
FX3U-4DA	8	120	0	160	—
FX3U-20SSC-H	8	100	0	220	—
FX2N-2AD	8	20	50 ^①	0	170
FX2N-2DA	8	30	85 ^①	0	190
FX2N-4AD	8	30	0	55	—
FX2N-4DA	8	30	0	200	—
FX2N-4AD-TC	8	30	0	50	—
FX2N-4AD-PT	8	30	0	50	—
FX2N-8AD	8	50	0	80	—
FX2N-5A	8	70	0	90	—
FX2N-2LC	8	70	0	55	—
FX2N-1HC	8	90	0	0	—
FX2N-1PG-E	8	55	0	40	—
FX2N-10PG	8	120	0	70 ^②	—
FX2N-232IF	8	40	0	80	—
FX2N-16CCCL-M	8 ^③	0	0	150	—
FX2N-32CCCL-M	8	130	0	50	—
FX2N-32ASI-M	8 ^④	150	0	70	—
FX0N-3A	8	30	90 ^①	0	165
FX2N-10GM	8	—	—	5	—
FX2N-20GM	8	—	—	10	—

- ① Когда аналоговые специальные функциональные модули (FX0N-3A, FX2N-2AD и FX2N-2DA) подключены к расширительному модулю ввода-вывода с собственным блоком питания (FX2N-32E□ или FX2N-48E□) необходимо учитывать следующие ограничения. (Когда модули подключены к главному блоку, ограничения не действуют.)
Полное потребление тока аналоговых специальных функциональных модулей (FX0N-3A, FX2N-2AD и FX2N-2DA) должно быть меньше следующих значений токов.
- При подключении к FX2N-32E□: 190 мА или ниже
- При подключении к FX2N-48E□: 300 мА или ниже.
- ② При напряжении внешнего блока питания постоянного тока 5 В ток составляет 100 мА.
- ③ FX2N-16CCCL-M нельзя использовать вместе с FX2N-32ASI-M. Следующее количество точек добавляется в зависимости от продуктов, подключенных к сети: (Количество станций удаленного ввода-вывода) x 32 точки.
- ④ FX2N-32ASI-M нельзя использовать вместе с FX2N-16CCCL-M. К полной системе можно добавить только один блок. Следующее количество точек добавляется в зависимости от продуктов, подключенных к сети: (Количество активных подчиненных) x 8 точек.

ПРИМЕЧАНИЕ

При подключении к базовому блоку с БП постоянного напряжения необходимо учитывать мощность, потребляемую при включении системы.

А.5 Глоссарий функциональных компонентов

В следующей таблице разъяснено назначение и принцип работы отдельных компонентов и узлов программируемого контроллера.

Компонент	Описание
Разъем для плат адаптеров расширения	К этому интерфейсу можно подсоединять опциональные расширительные адаптеры. Адаптеры имеются в различных исполнениях для всех серий FX (кроме FX2NC). Эти адаптеры позволяют оснастить контроллеры дополнительными функциями или коммуникационными интерфейсами. Адаптеры можно вставлять непосредственно в гнездо.
Разъем для программаторов	К этому разъему можно подключить ручной программатор FX-20P-E, а также внешний персональный компьютер или ноутбук со средой программирования (например, GX Developer).
EEPROM	Запоминающие устройства двустороннего действия, в которые записывается рабочая программа из среды программирования и из которых она считывается. Эти запоминающие устройства являются постоянными запоминающими устройствами, т. е. они сохраняют информацию даже при пропадании напряжения, а поэтому не нуждаются в буферном питании от батарейки.
Гнездо для кассет памяти	В этот разъем можно вставлять опциональные кассеты памяти. При установке этих кассет внутренняя память контроллера отключается, после чего обрабатывается только программа, находящаяся во вставленной кассете памяти.
Расширительная шина	Помимо дополнительных расширительных модулей ввода-вывода, к этой расширительной шине можно подключить модули, расширяющие функциональные возможности ПЛК. Обзор таких устройств содержится в разделе 6 этого руководства.
Аналоговые потенциометры	С помощью аналоговых потенциометров можно устанавливать заданные значения. Соответствующую настройку можно опрашивать с помощью программы и использовать для таймеров, вывода импульсов или подобных целей.
Источник сервисного напряжения	Источник сервисного напряжения (не имеется у FX2NC) поставляет регулируемое постоянное напряжение 24 В для питания входных сигналов и датчиков. Допустимая нагрузка этого источника напряжения зависит от типа контроллера (например, FX1S и FX1N: 400мА; от FX2N-16М□-□□ до FX2N-32М□-□□: 250 мА, от FX2N-48М□-□□ до FX2N-64М□-□□: 460 мА)
Цифровые входы	Через цифровые входы принимаются управляющие сигналы от подключенных к ним выключателей, кнопок или датчиков. Эти входы могут считывать состояния ВКЛ (напряжение подано) и ВыКЛ (напряжение снято).
Цифровые выходы	В зависимости от применения и типа выхода, к цифровым выходам можно подключать сервоэлементы и исполнительные устройства, например, контакторы.
Светодиоды для индикации состояния входов	Светодиоды состояния входов показывают, к какому входу приложен сигнал, т.е. определенное напряжение. Если соответствующий светодиод горит, это означает, что напряжение имеется, т. е. ко входу приложен управляющий сигнал и вход включен.
Светодиоды для индикации состояния выходов	Состояния выходов (т. е. включен выход или выключен) показываются с помощью светодиодов. При этом выходы контроллера могут коммутировать различные напряжения, в зависимости от их типа.
Светодиоды для индикации рабочего состояния	Светодиоды RUN, POWER и ERROR показывают текущее рабочее состояние программируемого контроллера. POWER указывает наличие напряжения питания, RUN светится при выполнении программы, находящейся в памяти контроллера, а ERROR - при наличии сбоя.
Батарейка буферного питания	При пропадании напряжения батарейка обеспечивает буферное питание внутренних запоминающих устройств типа RAM в контроллере MELSEC (только FX2N, FX2NC и FX3U). Батарейка служит для буферизации областей фиксации таймеров, счетчиков и маркеров. Кроме того, она питает встроенные часы, когда питание программируемого контроллера выключено.
Выключатель RUN/STOP	Контроллеры MELSEC имеют два рабочих режима: С помощью выключателя RUN/STOP можно переключать контроллер между этими двумя режимами. В режиме RUN контроллер выполняет программу, хранящуюся в памяти. В режиме STOP выполнение программы не происходит. В этом режиме контроллер можно программировать.

Указатель

A

AS-интерфейс	2 - 27
сетевой модуль	2 - 27
Auto connect	4 - 20

C

CANopen	
сетевой модуль	2 - 26
CC-Link	
сетевые модули	2 - 25

D

Device Edit (функция в меню Debug)	9 - 1
DeviceNet	
сетевые модули	2 - 26
DUT	3 - 17
См. Типы блоков данных	

E

EEPROM	A - 14
ENO-выход	6 - 22
Entry Data Monitor	
настройка	7 - 2
выбор	7 - 1
EN-Вход	6 - 22
ETHERNET	
конфигурация	19 - 1
сетевые модули	2 - 22

F

FX0N-32NT-DP	2 - 23
FX1N-8AV-BD	2 - 30
FX1N-CNV-BD	2 - 29
FX2N-10PG	2 - 21
FX2N-16CCL-M	2 - 25
FX2N-1HC	2 - 20
FX2N-1PG-E	2 - 21
FX2N-232IF	2 - 28
FX2N-32ASI-M	2 - 27
FX2N-32CAN	2 - 26
FX2N-32CCL	2 - 25
FX2N-32DP-IF	2 - 24
FX2N-64DNET	2 - 26
FX2N-8AV-BD	2 - 30

FX2NC-ENET-ADP	2 - 22
FX2N-CNV-BD	2 - 29
FX3U-20SSC-H	2 - 21
FX3U-2HSY-ADP	2 - 20
FX3U-4HSX-ADP	2 - 20
FX3U-64DP-M	2 - 23
FX3U-CNV-BD	2 - 29
FX3U-ENET	2 - 22

G

GVL	
См. Список глобальных переменных	
GX Configurator DP	18 - 1

I

IEC61131-3	3 - 1
Interconnect Mode	4 - 20

L

LVL	
См. Список глобальных переменных	

M

MELSEC	2 - 6
Monitor headers (функция в режиме Monitor)	7 - 6

P

PLCopen	3 - 1
PROFIBUS/DP	
конфигурация	18 - 1
сетевой модуль	2 - 23

S

SCADA	2 - 2
SFC	
начальный шаг	15 - 2
обзор	3 - 14
конечный шаг	15 - 2
переходы	15 - 2
Start Monitoring (функция в Меню Online)	7 - 7

A

Адаптерные платы	2 - 17
Аналоговые модули	2 - 18

Б

Базовый блок	
FX1N	2 - 9
FX1S	2 - 8
FX2N	2 - 9
FX2NC	2 - 10
FX3U	2 - 10
Обзор	2 - 6
источник питания	2 - 11
клемма S/S	2 - 12

Б

Батарейка буферного питания	A - 14
-----------------------------	--------

В

Входы	
назначение	2 - 37
подключение	2 - 12
Выбор из заголовка ПМ	4 - 16
Выключатель RUN/STOP	A - 14
Выполнение макрокоманды	6 - 21
Выходы	
назначение	2 - 37
подключение	2 - 13

Г

Глобальные переменные	
назначение	4 - 9
определение	3 - 6
список	3 - 6
Глоссарий	A - 14

Д

Документация	
Комментарии к звену	4 - 33
заголовок звена	4 - 33
опции печати	4 - 51

З

Заголовок звена	4 - 33
Загрузка программ	4 - 41
Задача	
назначение программного модуля	4 - 30
атрибуты	4 - 31
создание нового	4 - 29
определение	3 - 3
пул	3 - 7

свойства	4 - 31
Задачный пул	
определение	3 - 7
Заземление	2 - 11
Значения с плавающей запятой	
См. Числа формата REAL	

И

Изменение программы в режиме онлайн (функция в меню Project)	10 - 3
Интерфейс	
адаптеры	2 - 28
модули	2 - 28
Источник сервисного напряжения	A - 14

К

Коды ошибок	A - 7
Комментарии к звену	4 - 33
Комментарий	
копирование	4 - 34
удаление	4 - 34
для звеньев программы	4 - 33
Коммуникационные адаптеры	2 - 29

Л

Локальные переменные	
определение новой	4 - 19
определение	3 - 6
список	3 - 6

М

Массивы	
обзор	3 - 16
программирование	12 - 1
Меню Debug	
Device Edit	9 - 1
принудительная установка входов/выходов	8 - 1
Меню Online	
Entry Data Monitor	7 - 1
Monitor Header	7 - 6
Start Monitoring	7 - 7
Меню Online	
настройки передачи	4 - 37
Меню Project	
изменение паролей	14 - 1
изменение уровня безопасности	14 - 2
изменение программы в режиме онлайн	10 - 3
Метки	3 - 10

Модули позиционирования	2 - 21
Модули счетчиков	2 - 20
Модуль регулирования температуры	2 - 18, 2 - 19

Н

Направляемый режим ввода релейной диаграммы	4 - 36
Номер станции (Ethernet-параметр)	19 - 4

О

Обработка отображения процесса	2 - 4
Образ системы	4 - 40
Операционные настройки (Ethernet)	19 - 4
Оптопары	2 - 8
Открытые настройки (Ethernet)	19 - 6
Отрицательная логика	
входы	2 - 12
выходы	2 - 14

П

ПАНЕЛЬ ОПЕРАТОРА	2 - 2
Перекрёстные ссылки	4 - 47
Переменные	
назначение инструкции	4 - 19
глобальные (определение)	3 - 6
см. также Глобальные переменные	
локальные (определение)	3 - 6
см. также Локальные переменные	
ПЛК	
сравнение с релейными системами	2 - 1
диагностика	4 - 50
история	2 - 1
ПМ	
назначение задаче	4 - 30
создание нового	4 - 8
определение	3 - 2
заголовок	4 - 13
программирование	4 - 14
Поиск неисправностей	
коды ошибок	A - 7
специальные регистры	A - 6
специальные маркеры	A - 3
Положительная логика	
входы	2 - 12
выходы	2 - 14
Последовательная функциональная схема	

обзор	3 - 14
Проверка соединения	4 - 39
Программа	
проверка	4 - 23
перекрестная ссылка	4 - 47
загрузка в ПЛК	4 - 37
мониторинг	4 - 44

Программируемый контроллер	
см. ПЛК	

Проект	
назначение входов-выходов	2 - 37
Пул программных модулей	
определение	3 - 4

Р

Расширение	
блоки	2 - 16
платы расширения	2 - 15
модули	2 - 15
Реальные переменные	
модификация в режиме Монитор	7 - 10
Реле	
сравнение с системами ПЛК	2 - 1
Релейная диаграмма	
ввод функционального блока	4 - 18
направляемый режим ввода релейной диаграммы	4 - 36
обзор	3 - 12
предосторожности	4 - 21
программирование	4 - 14

С

Свойства (задачи)	4 - 31
Серия FX	
потребляемый ток	A - 11
занятые точки ввода-вывода	A - 11
обзор	2 - 6
электропитание	2 - 11
Сетевой номер (Ethernet-параметр)	19 - 4
Сетевые модули	
AS интерфейс	2 - 27
CANopen	2 - 26
CC-Link	2 - 25
DeviceNet	2 - 26
ETHERNET	2 - 22
PROFIBUS	2 - 23

Сетевые параметры	19 - 2
Специальные маркеры	
описание	A - 1
диагностическая информация	A - 1
обнаружение ошибок	A - 3
режим работы ПЛК	A - 2
часы реального времени	A - 2
Специальные функциональные модули	
адрес	2 - 38
описание	2 - 17
Специальный адаптер	
подключение	2 - 32
описание	2 - 17
Специальный регистр	
описание	A - 4
диагностическая информация	A - 5
коды ошибок	A - 6
режим работы ПЛК	A - 5
часы реального времени	A - 5
Список глобальных переменных	
добавление элементов	4 - 26
назначение переменных	4 - 9
проверка	4 - 12
Список инструкций	3 - 11
Способ индикации	9 - 3
Структурированный текст	3 - 12
Счетчик	
адреса операндов	3 - 20
программирование	4 - 25

Т

Таймер	
адреса операндов	3 - 20
программирование	4 - 27

Тип результата	
для функции	6 - 11
Типы блоков данных	3 - 17, 11 - 1
Типы данных	3 - 15

У

Установка соединения	19 - 9
Установка соединения	4 - 37

Ф

Функциональная блок-схема	3 - 13
Функциональный блок	
назначение имен экземпляров	6 - 16
назначение переменных DUT	11 - 8
назначение переменных	6 - 18
сравнение с функцией	6 - 1
создание	6 - 14
ввод в релейную диаграмму	4 - 18
режимы выполнения	6 - 21
экземпляры	4 - 25
мониторинг экземпляров	7 - 11
Функция	
сравнение с функциональными блоками	6 - 1
создание	6 - 2
дублирование	6 - 10
тип результата	6 - 11

Э

Экземпляр	
для функциональных блоков	6 - 16

MITSUBISHI ELECTRIC EUROPE B.V. /// РОССИЯ /// Москва /// Космодамианская наб. 52, стр. 5
Тел.: +7 495 721-2070 /// Факс: +7 495 721-2071 /// automation@mer.mee.com /// www.mitsubishi-automation.ru