

# **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](http://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: 211687**

<b>Версия</b>	<b>Изменения / дополнения / исправления</b>
А 04/2009 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 специалистами, имеющими соответствующее 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:



#### **ОПАСНОСТЬ:**

*Предупреждения об опасности для здоровья и возможности травмирования персонала. Означает, что непринятие 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

<b>1</b>	<b>Обзор курса и требования</b>	
1.1	Модульный ПЛК для обучения.....	1-1
<b>2</b>	<b>Аппаратура контроллера</b>	
2.1	Программируемые контроллеры.....	2-1
2.1.1	История и развитие.....	2-1
2.1.2	Принципиальные требования к контроллеру.....	2-1
2.1.3	Сравнение между контроллером и схемной системой управления (системой управления на основе физического монтажа).....	2-1
2.1.4	Программирование на языке релейно-контактных схем.....	2-2
2.1.5	SCADA и интерфейс "человек-машина".....	2-2
2.2	Конструкция программируемого контроллера.....	2-3
2.2.1	Спецификации контроллера.....	2-3
2.3	MELSEC System Q.....	2-4
2.3.1	Конфигурация системы.....	2-4
2.3.2	Монтажная шина.....	2-6
2.3.3	Сопоставление адресов ввода-вывода на главной монтажной шине....	2-8
2.3.4	Присвоение адресов ввода-вывода на расширительных монтажных шинах..	2-9
2.4	Расширительный кабель.....	2-10
2.5	Сетевые блоки.....	2-10
2.5.1	Выбор подходящего сетевого блока.....	2-11
2.6	Модули центральных процессоров.....	2-12
2.6.1	Технические данные.....	2-13
2.7	Подключение внешних сигналов.....	2-20
2.7.1	Электропроводка входов и выходов.....	2-20
2.8	Цифровые входные и выходные модули.....	2-21
2.8.1	Цифровые входные модули.....	2-22
2.8.2	Цифровые выходные модули.....	2-30
2.9	Специальные модули.....	2-38
2.9.1	Аналоговые входные модули.....	2-38
2.9.2	Аналоговые выходные модули.....	2-38
2.9.3	Модули регулирования температуры с ПИД-алгоритмом.....	2-39
2.9.4	Модули высокоскоростных счетчиков.....	2-39
2.9.5	Модули позиционирования.....	2-40
2.9.6	Интерфейсные модули для последовательной коммуникации.....	2-40
2.9.7	Интерфейсные модули, программируемые на языке BASIC.....	2-41

2.9.8	Модули Ethernet .....	2-41
2.9.9	Модули MELSECNET .....	2-42
2.9.10	Главный/локальный модуль для CC-Link .....	2-42
2.9.11	Модуль PROFIBUS/DP .....	2-43
2.9.12	Главный модуль DeviceNet QJ71DN91 .....	2-43
2.9.13	Веб-серверный модуль .....	2-44
2.10	Основы контроллеров .....	2-45
2.10.1	Среда программирования .....	2-45
2.10.2	Обработка программы в контроллере .....	2-46
2.10.3	Операнды программируемого контроллера .....	2-48

### **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
4.2.1	Документирование программы .....	4-34
4.2.2	Проверка и построения кода проекта .....	4-36
4.2.3	Иллюстрация: Направляемый режим ввода релейной диаграммы .....	4-37
4.3	Процедуры загрузки проекта .....	4-38
4.3.1	Подключение с помощью периферийных устройств .....	4-38
4.3.2	Конфигурация коммуникационного порта .....	4-39

4.3.3	Форматирование памяти контроллера .....	4-42
4.3.4	Загрузка проекта.....	4-43
4.4	Мониторинг проекта .....	4-45
4.4.1	Раздельный / Многооконный мониторинг .....	4-46
4.4.2	Настройка видимости монитора.....	4-48
4.5	Список перекрестных ссылок .....	4-49
4.6	Диагностика ПЛК .....	4-52
4.7	Документация проекта .....	4-53

## **5 Пример программы**

5.1	QUIZMASTER - ВЕДУЩИЙ ТЕЛЕВИКТОРИНЫ .....	5-1
5.1.1	Метод .....	5-2
5.1.2	Quizmaster - Принцип работы .....	5-6
5.1.3	Описание программы Quizmaster .....	5-6

## **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 .....	7-2
7.1.2	Изменение состояния битовых операндов в EDM .....	7-6
7.2	Мониторинг заголовков.....	7-7
7.3	Основные возможности в режиме мониторинга .....	7-8
7.4	Указание групп битов .....	7-10
7.5	Модификация значений переменных из тела программного модуля .....	7-11
7.6	Мониторинг "экземпляров" функциональных блоков .....	7-12

<b>8</b>	<b>Принудительная установка цифровых входов и выходов</b>	
<b>9</b>	<b>Device Edit</b>	
<b>10</b>	<b>Режим онлайн</b>	
10.1	Режим изменения в реальном времени .....	10-1
10.2	Изменение программы в режиме онлайн .....	10-4
<b>11</b>	<b>Типы блоков данных (DUT)</b>	
11.1	Пример использования DUT .....	11-2
11.2	Автоматическое заполнение, переменные .....	11-5
11.3	Назначение переменных DUT функциональным блокам .....	11-8
<b>12</b>	<b>Массивы</b>	
12.1	Обзор .....	12-1
12.2	Пример массива: Одномерный массив .....	12-3
<b>13</b>	<b>Работа с библиотеками</b>	
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 Mitsubishi .....	13-8
13.3.2	Пример: Импорт функционального блока из библиотеки Mitsubishi .....	13-11
13.3.3	Контекстная справка библиотечного функционального блока: .....	13-14
<b>14</b>	<b>Безопасность</b>	
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

<b>15</b>	<b>Последовательная функциональная схема - SFC</b>	
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
<b>16</b>	<b>Список инструкций IEC</b>	
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
<b>17</b>	<b>Структурированный текст IEC</b>	
17.1	Операторы языка структурированного текста .....	17-1
17.2	Пример программы языка структурированного текста .....	17-2
<b>18</b>	<b>Связь по Ethernet</b>	
18.1	Конфигурирование параметров Ethernet модуля .....	18-1
18.1.1	Конфигурирование ПЛК (с использованием ПК для начальной настройки) .....	18-2
18.2	Конфигурирование ПК для Ethernet .....	18-8
18.3	Конфигурирование GX IEC Developer для доступа контроллера в Ethernet .....	18-9
18.4	Установка интерфейса человек - машина (HMI) .....	18-13
18.5	Связь через MX Component .....	18-16
<b>A</b>	<b>Приложение A</b>	
A.1	Определение времени обработки (SM) .....	A-1
A.2	Соответствия между специальными маркерами и маркерами диагностики .....	A-7
A.3	Обзор регистров диагностики (SD) .....	A-13
A.3.1	Информация цикла программы .....	A-31



# 1 Обзор курса и требования

Этот курс специально разработан как введение в ПЛК MITSUBISHI ELECTRIC MELSEC System Q с использованием программного пакета GX Developer версии 8.

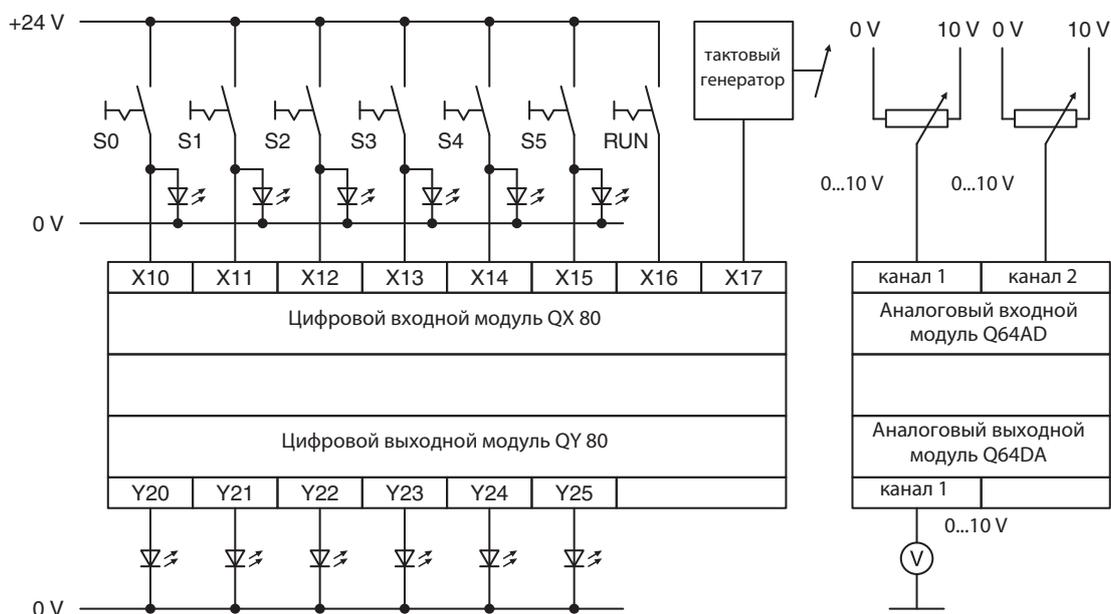
Содержание курса подобрано таким образом, чтобы ознакомить с функциональными возможностями продуктов MITSUBISHI ELECTRIC MELSEC System Q, а также с системой программирования GX Developer. В первом разделе рассматривается аппаратная конфигурация и работа ПЛК, а последующая часть курса охватывает использование 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 ELECTRIC MELSEC System Q. Большинство задач в данном руководстве основано на использовании установок, предложенных в этих тренировочных системах. Используемые в курсе примеры представлены для следующей конфигурации:

- 6 выключателей для ввода цифровых сигналов: X10-X15
- Регулируемый тактовый вход (1...100 Гц и 0.1...10 кГц): X17
- 6 светодиодов для индикации состояния цифровых выходов: Y20-Y25
- 4 аналоговых входных канала: модуль Q64AD с головным адресом 30H
- 4 аналоговых выходных канала: Q64DA с головным адресом 40H



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



## 2 Аппаратура контроллера

### 2.1 Программируемые контроллеры

#### 2.1.1 История и развитие

Первая система управления с программированием от запоминающего устройства была разработана в 1968-м году фирмой Bedford Associates, основанной Ричардом Морли (Richard Morley). От обозначения этой первой программируемой системы управления - Modular Digital Controller - происходит название фирмы MODICON.

Программируемые контроллеры разрабатывались для замены сложных контакторных систем управления, в которых изменить процесс управления можно лишь путем многозатратных работ по изменению кабельной проводки или замене компонентов. В отличие от таких систем управления, в программируемом контроллере обычно достаточно лишь изменить программу, хранящуюся в памяти контроллера.

Разработка микропроцессоров (начиная приблизительно с 1970-го года) и все большее возрастание скоростей обработки позволили применять программируемые контроллеры в сложных системах и придать им дополнительные функции. В наше время контроллер стал стандартным ядром автоматизации. При этом он часто соединен с главным уровнем управления (SCADA, Supervisory Control And Data Acquisition), панелями управления (интерфейс "человек-машина") или экспертными системами. Помимо универсальных функций управления, требования к контроллерам лежат также в области обработки данных и техники управления производственными процессами.

#### 2.1.2 Принципиальные требования к контроллеру

- Контроллер должен легко программироваться. Должна иметься простая возможность изменения программ по месту.
- Удобство для технического обслуживания и ремонта - лучше всего на основе модульной конструкции
- Контроллер должен быть механически и электрически устойчив к суровым условиям эксплуатации в промышленной окружающей среде
- Контроллер должен быть меньше, чем сравнимые контакторные или обычные системы управления
- Контроллер должен быть дешевле, чем сравнимые контакторные или обычные системы управления

#### 2.1.3 Сравнение между контроллером и схемной системой управления (системой управления на основе физического монтажа)

Свойство	Контроллер	Система управления на основе физического монтажа с контакторами
Стоимость каждой функции	Низкая	Низкая - если в системе управления используются более 10 контакторов
Размеры	Очень компактная	Громоздкая
Скорость обработки	Высокая	Низкая
Устойчивость к электромагнитным помехам	Хорошая	Превосходная
Конструкция	Простое программирование	Обременительный электромонтаж
Комплексные функции	Возможны	Не возможны
Изменение рабочего процесса	Очень просто	Очень сложно (изменение проводки)
Удобство для техобслуживания	Превосходно (контроллер редко выходит из строя)	Плохо - контакторы требуют постоянного техобслуживания

## 2.1.4 Программирование на языке релейно-контактных схем

Контроллер должен быть доступен для понимания техникам и заводским электрикам. По этой причине разработано программирование на языке релейно-контактных схем (LD). Элементы этого языка программирования напоминают символы электросхем, которые используются в релейных системах управления и знакомы каждому сотруднику, имеющему электротехническое образование.

В ранних программах контроллера еще не имелось возможностей для документирования программ (или имелись лишь очень ограниченные возможности). В связи с тем, что можно было указывать лишь адреса или простые комментарии, большие программы было трудно понимать. После разработки таких прогрессивных инструментов программирования, как **GX Developer** Mitsubishi возможности документирования стремительно улучшились.

Долгое время не имелось единого стандарта программирования контроллеров. Такой стандарт появился в 1998-м году в виде стандарта **IEC 61131-3**. Среда программирования **GX-IEC Developer** Mitsubishi Электрик дает возможность структурированного программирования по стандарту IEC61131-3.

## 2.1.5 SCADA и интерфейс "человек-машина"

В первых контроллерах оператор делал вводы точно так, как это происходит в обычных системах управления – т. е. с помощью клавишей и выключателей. Для индикации применялись сигнальные лампы.

Появление персональных компьютеров в восьмидесятых годах прошлого столетия позволило разработать устройства ввода-вывода на базе компьютера. Если компьютер оснащен специальным программным обеспечением, то такую систему называют также SCADA (Supervisory Control And Data Acquisition) – система для управления и сбора данных.

Специальные панели управления обозначаются как MMI (Man-Machine Interface - интерфейс "человек-машина"), так как они образуют интерфейс между управляемым процессом и оператором. На сегодня SCADA и MMI занимают основное место в технике управления и характеризуются большим удобством для пользователей благодаря применению программируемых контроллеров.

Mitsubishi предлагает большой выбор интерфейсов "человек-машина" и решений в области SCADA для любых задач.



## 2.2 Конструкция программируемого контроллера

В отличие от системы управления, функция которой определяется лишь схемой и электропроводкой, функция программируемого контроллера определяется программой. Хотя контроллер также нуждается в соединениях и внешних схемах, содержимое памяти для хранения программы можно в любое время изменить, приспособив программу к различным задачам управления.

В случае контроллеров данные вводятся, обрабатываются и результаты обработки снова выводятся. Этот процесс подразделяется на следующие уровни:

- уровень ввода,
- уровень обработки
- уровень вывода.

### Уровень ввода

Уровень ввода служит для передачи управляющих сигналов, вырабатываемых выключателями, кнопками или датчиками, на уровень обработки.

Сигналы этих компонентов возникают в процессе управления и подаются на входы в виде логических состояний. Уровень ввода передает сигналы – в подготовленной форме – на уровень обработки.

### Уровень обработки (центральный процессор)

На уровне обработки сигналы, собранные и подготовленные на уровне ввода, обрабатываются хранящейся в памяти программой в соответствии с определенными логическими взаимосвязями. Память для хранения программ, относящаяся к уровню обработки, свободно программируется. Имеется возможность в любое время повлиять на ход обработки, изменив или заменив программу.

### Уровень вывода

Результаты, полученные при обработке входных сигналов в программе, на выходном уровне влияют на коммутирующие устройства, подключенные к выходам (например, контакторы, сигнальные лампы, электромагнитные клапаны и т. п.).

### 2.2.1 Спецификации контроллера

Ниже выборочно перечислены некоторые соображения, которые необходимо учитывать при выборе конфигурации программируемого контроллера.

#### Внешняя аппаратура, входы и выходы

- Требования к входам и выходам:
- Сигнальное напряжение: постоянное 24 В или переменное 110/240 В?
- В случае постоянного напряжения 24 В пост. тока – по какой логике работают подключаемые ко входам датчики – по положительной или отрицательной?
- Тип выходов: транзисторные (положительная или отрицательная логика), симисторные, релейные или беспотенциальные контакты?

#### Напряжение питания

- Постоянное 24 В или переменное 110/240 В?

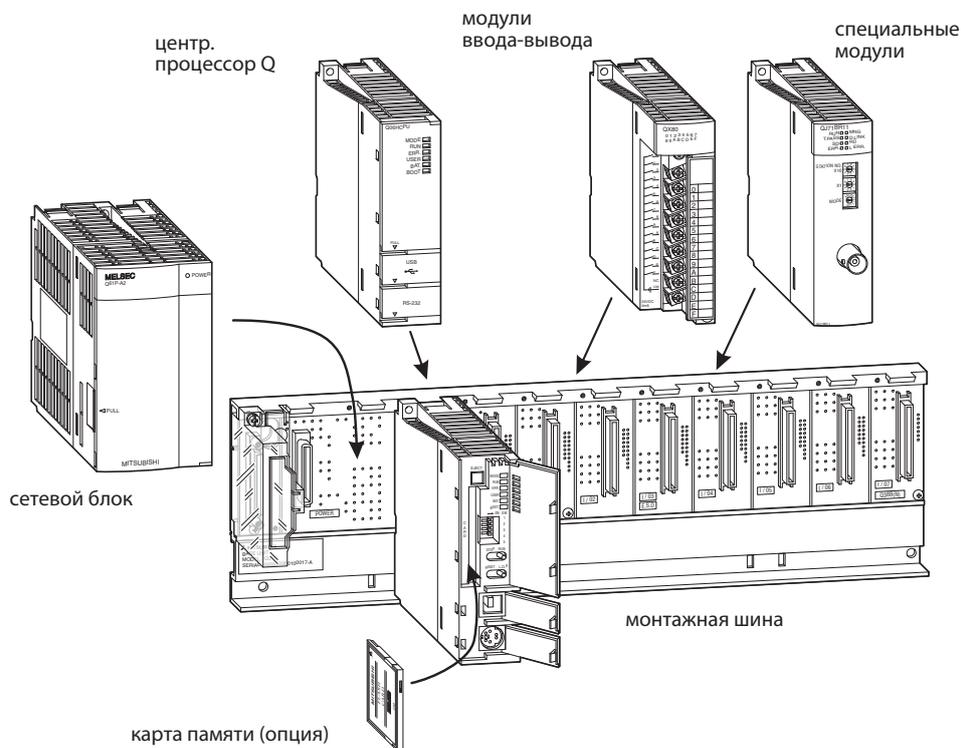
#### Специальные модули

- Количество специальных модулей (например, аналоговых, сетевой коммуникации, интерфейсных) в системе
- Нужно ли внешнее электропитание для специальных модулей?

## 2.3 MELSEC System Q

В следующем разделе дан обзор конструкции контроллера "MELSEC System Q".

### 2.3.1 Конфигурация системы



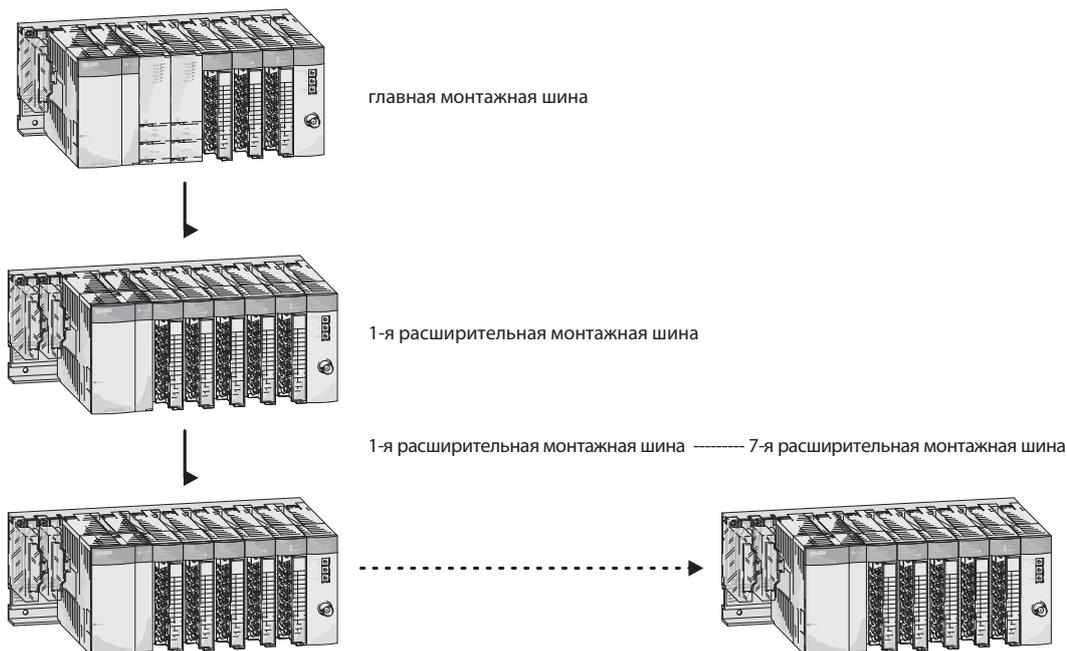
Центральный процессор и модули монтируются на главной монтажной шине. Отдельные модули сообщаются друг с другом через заднюю стенку монтажной шины. Электропитание всей системы осуществляет сетевой блок, также установленный на монтажной шине.

Главную монтажную шину можно заказать в различных исполнениях, с количеством слотов для модулей ввода-вывода или специальных модулей от 3 до 12. Подсоединив расширительные монтажные шины с дополнительными разъемами, систему можно расширить.

Свободные слоты на монтажной шине можно защитить от загрязнений или механических повреждений, вставив в них пустые модули. Кроме того, с помощью пустого модуля можно зарезервировать адреса ввода-вывода для более позднего расширения системы.

При построении сложных установок или машин с модульной конструкцией децентрализованный ввод и вывод (с помощью станций удаленного ввода-вывода) дает преимущества местного расположения аппаратуры. При этом соединения между входами или выходами и датчиками или коммутационными компонентами имеют малую длину. Для связи между удаленной станцией ввода-вывода и системой с центральным процессором контроллера требуется лишь один кабель сетевой коммуникации.

### Главная и расширительная монтажная шина



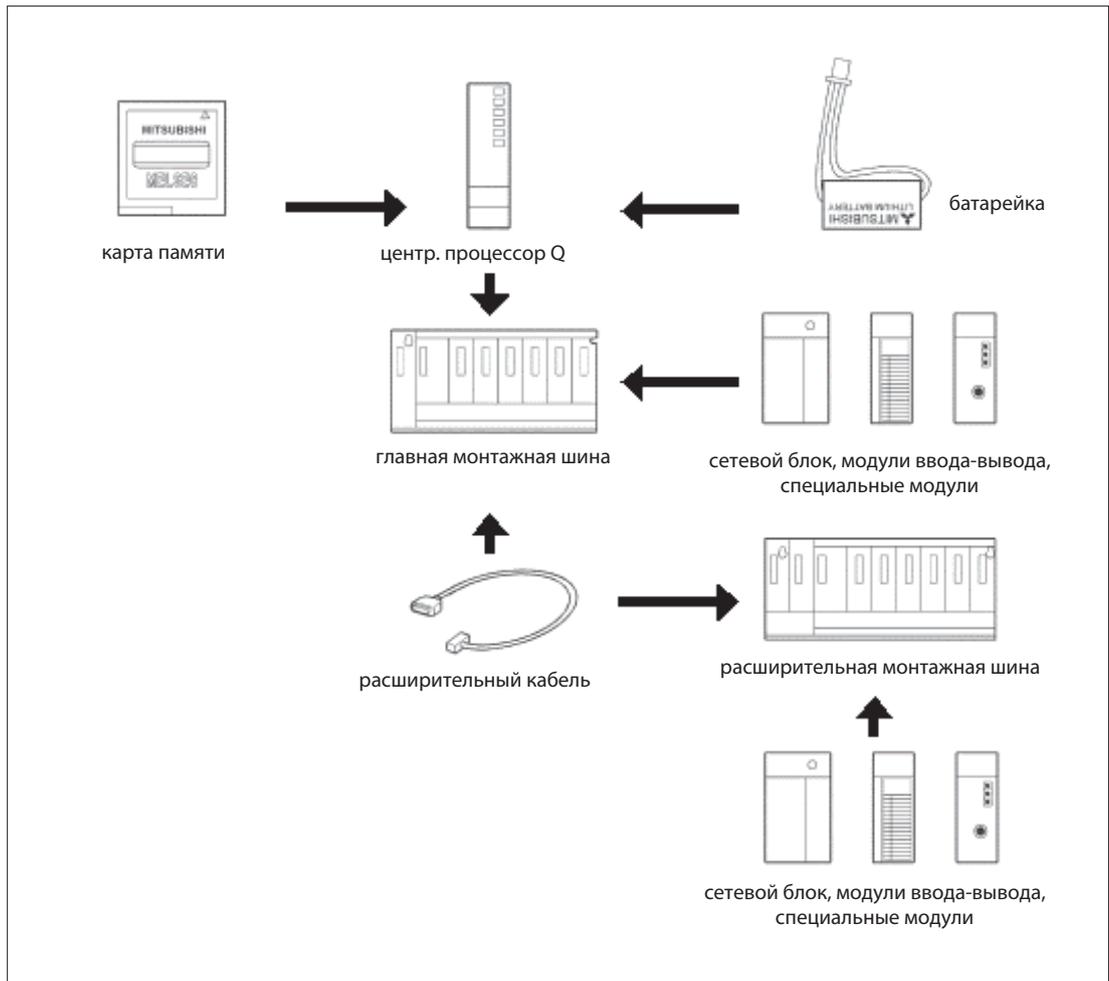
Главная и расширительная монтажная шина легко соединяются друг с другом кабелем. Этот расширительный кабель служит и для питания расширительной шины, если она не имеет собственного сетевого блока.

К главной шине "MELSEC System Q" можно подключить до семи расширительных шин с количеством модулей до 64. Общая длина расширительных кабелей не должна превышать 13.2 м.

При выборе сетевого блока необходимо учитывать ток, потребляемый входными и выходными модулями, специальными модулями и периферийными приборами. Если необходимо, следует применять расширительную шину с собственным сетевым блоком.

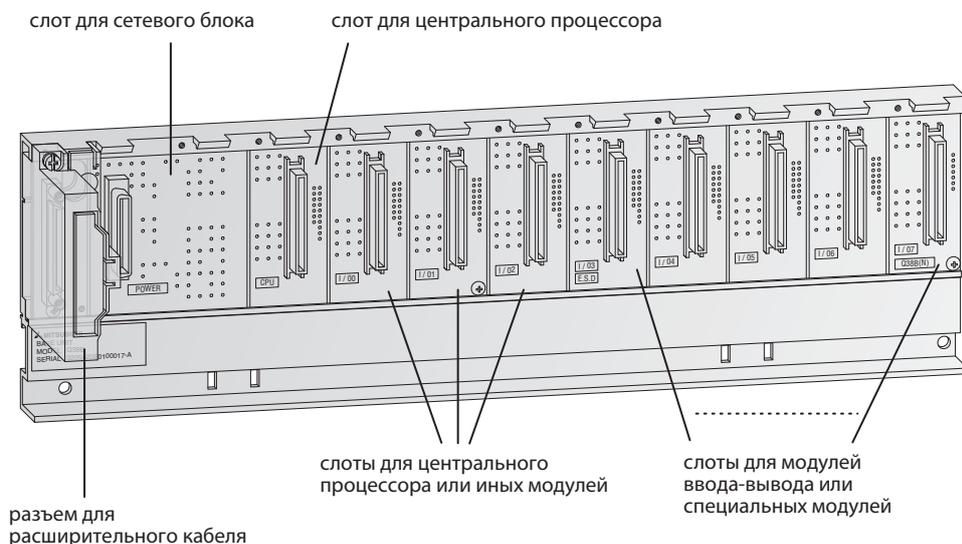
#### Количество подключаемых расширительных шин

- К главной монтажной шине с процессором Q00 или Q01 можно подключить до 4 расширительных шин с количеством модулей ввода-вывода до 24.
- Контроллер системы "System Q" с центральным процессором Q02, Q02H, Q06H, Q12H или Q25H может содержать до 7 расширительных шин и 64 модулей ввода-вывода.



### 2.3.2 Монтажная шина

Главная монтажная шина несет сетевой блок, один или несколько модулей центральных процессоров, а также модули ввода-вывода или специальные модули. На расширительной шине можно установить модули ввода-вывода и специальные модули. Аппаратура крепится либо непосредственно (например, в распределительном шкафу), либо с помощью переходников на стандартной DIN-рейке.



В следующих таблицах перечислены все предлагаемые монтажные шины.

Свойство	Расширительная монтажная шина				
	Q33B	Q35B	Q38B	Q38RB	Q312B
Количество слотов для сетевых блоков	1	1	1	2*	1
Количество слотов для модулей ввода-вывода или специальных модулей	3	5	8	8	12

\* На этой главной шине можно использовать избыточные сетевые блоки.

Свойство	Расширительная монтажная шина						
	Q52B	Q55B	Q63B	Q65B	Q68B	Q68RB	Q612B
Количество слотов для сетевых блоков	—	—	1	1	1	2*	1
Количество слотов для модулей ввода-вывода или специальных модулей	2	5	3	5	8	8	12

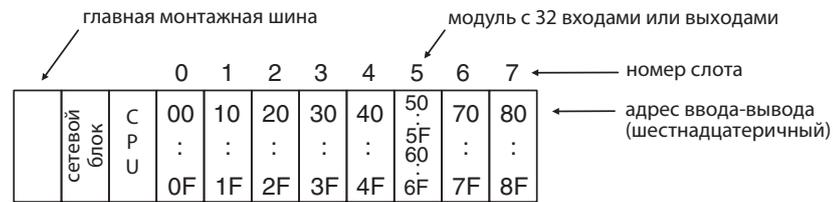
\* На этой расширительной шине можно использовать избыточные сетевые блоки.

### 2.3.3 Сопоставление адресов ввода-вывода на главной монтажной шине

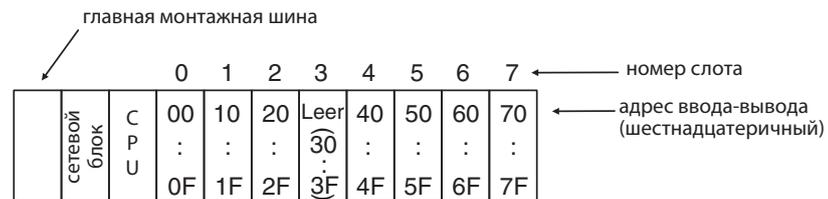
Входам и выходам контроллера необходимо присвоить адреса, чтобы к ним могла обращаться программа. Адреса входов и выходов модулей ввода-вывода, установленных на главной монтажной шине, и головные адреса специальных модулей присваиваются слотам автоматически. Однако их может присваивать и пользователь.



При присвоении адресов ввода-вывода система предполагает, что на всех слотах установлены модули с 16 входами или выходами. Поэтому от слота к слоту адреса ввода-вывода повышаются на 16 (от 0 до F в шестнадцатеричном счислении). Если, однако, в слот вставлен модуль с 32 входами или выходами (на следующей иллюстрации это слот 5), это учитывается и адреса последующих слотов соответственно смещаются.



16 адресов ввода-вывода присваиваются и пустому слоту. На рисунке ниже показана конфигурация, при которой в слот 3 не вставлено никакого модуля ввода-вывода.

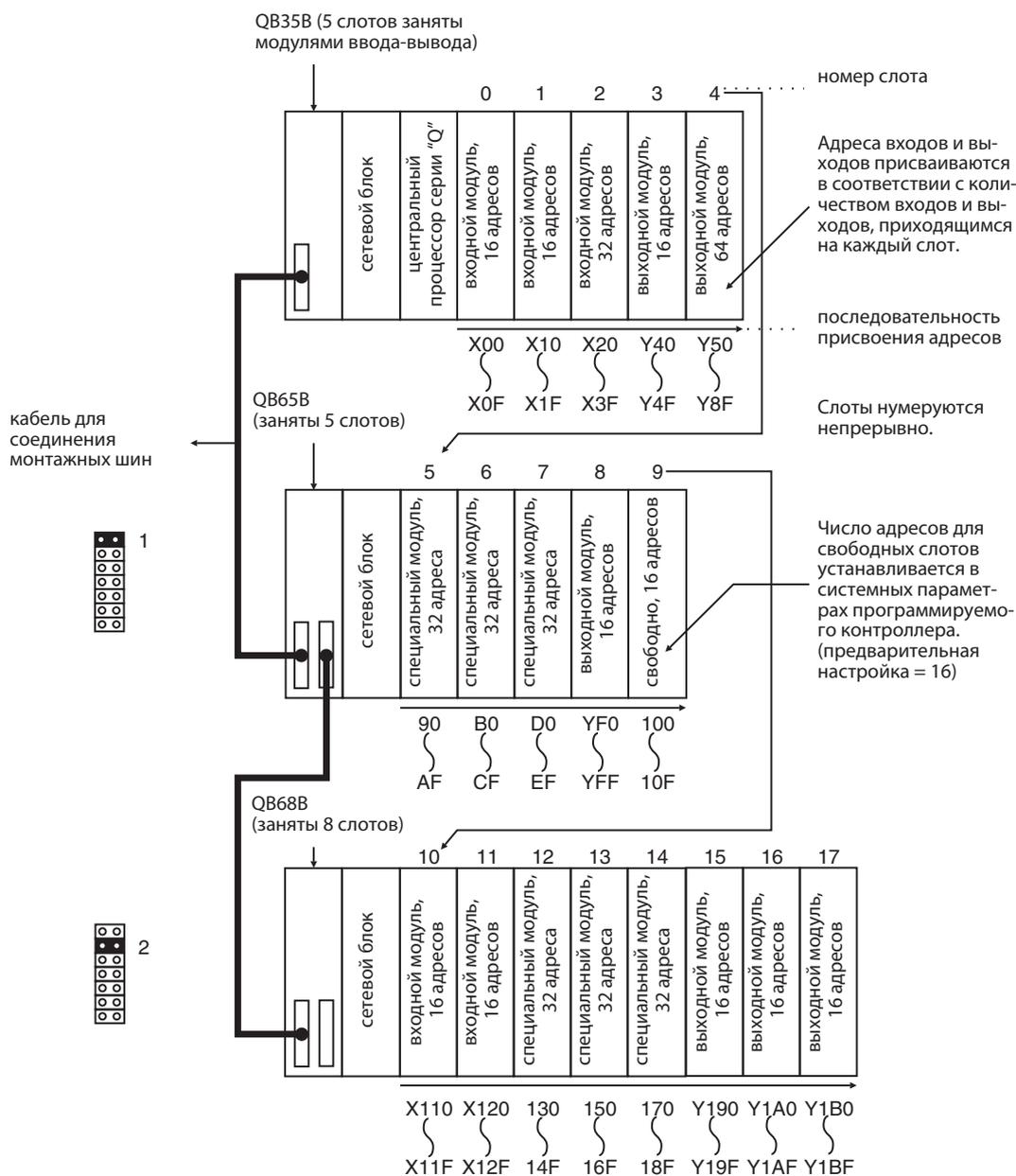


### 2.3.4 Присвоение адресов ввода-вывода на расширительных монтажных шинах

Если помимо слотов на главной монтажной шине нужны дополнительные слоты, можно подсоединить расширительную монтажную шину. Адреса ввода-вывода присваиваются по следующему принципу:

- Адреса ввода-вывода слотов расширительной шины присваиваются в возрастающей последовательности с шестнадцатеричной нумерацией.
- Адресация главной шины продолжается с первого слота первого расширительной шины после главной шины.

Принцип адресации наглядно пояснен на следующей иллюстрации:



## 2.4 Расширительный кабель

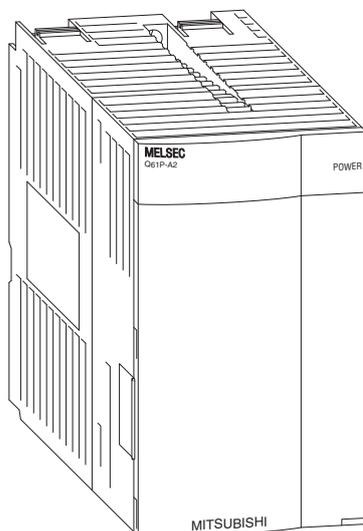
Расширительные кабели служат для соединения главной и расширительной монтажной шины.

Расширительный кабель	QC05B	QC06B	QC12B	QC30B	QC50B	QC100B
Длина	0.45 м	0.50 м	1.2 м	3.0 м	5.0 м	10.0 м

Общая длина всех соединительных кабелей не должна превышать 13.2 м.

Для подключения расширительной шины без собственного сетевого блока (Q52B, Q55B) рекомендуется использовать кабель QC05B.

## 2.5 Сетевые блоки



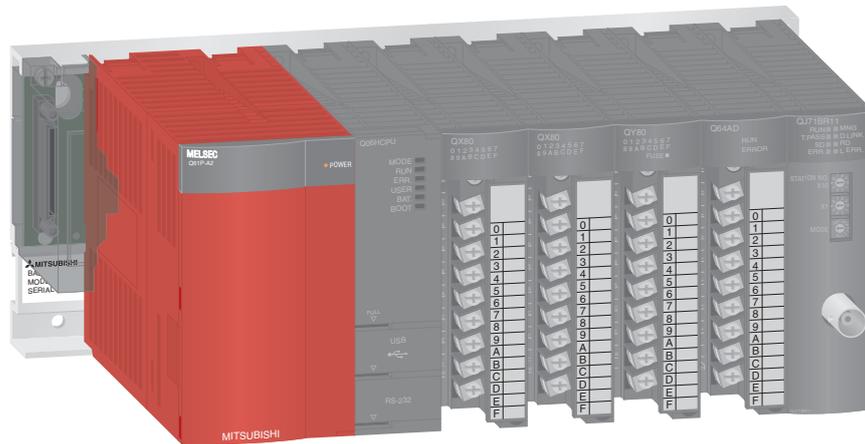
Для питания контроллера "System Q" используется постоянное напряжение 5 вольт. Имеются сетевые блоки с входными напряжениями 24 В пост. т. и 100...240 В пер. т..

Свойство	Q63P	Q61P-A1	Q61P-A2	Q62P	Q64P
Входное напряжение	24 В пост.	100 - 120 В пер. т.	200 - 220 В пер. т.	100 - 240 В пер. т.	100 - 120 В пер. т. 200 - 240 В пер. т.
Потребляемая мощность	45 Вт	105 ВА	105 ВА	105 ВА	105 ВА
Выходное напряжение	5 В пост. т.	5 В пост. т.	5 В пост. т.	5 В пост. т., 3 А 24 В пост. т., 0.6 А	5 В пост. т.
Выходной ток	6 А	6 А	6 А		8.5 А

### 2.5.1 Выбор подходящего сетевого блока

Ток, потребляемый модулями на монтажной шине, не должен превышать номинальный ток, который способен поставлять сетевой блок. Если ток превышает, необходимо уменьшить количество модулей на шине.

#### Пример расчета потребляемого тока:

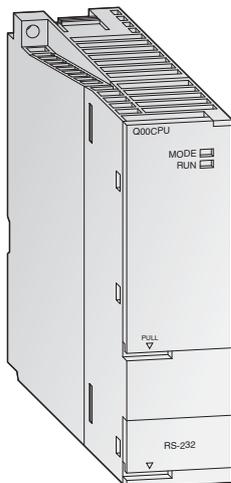


Модуль	Тип модуля	Потребляемый ток
Q06HCPU	Модуль центрального процессора	0.64 A
QX80	Цифровой входной модуль	0.16 A
QX80	Цифровой входной модуль	0.16 A
QY80	Цифровой выходной модуль	0.008 A
Q64AD	Аналоговый входной модуль	0.63 A
QJ71BR11	Модуль MELSECNET/H	0.75 A
Суммарное потребление тока		2.42 A

Сумма потребляемых токов равна 2.42 А, т. е. меньше номинального тока 6 А, который может вырабатывать сетевой блок. Поэтому при эксплуатации контроллера никакие проблемы не возникнут.

## 2.6 Модули центральных процессоров

### Базовые центральные процессоры контроллера



Модули центральных процессоров MELSEC System Q могут быть однопроцессорными и многопроцессорными, что позволяет достичь большей универсальности применения. По мере совершенствования и расширения установки можно наращивать и возможности контроллера - путем простой замены центрального процессора (за исключением процессора Q00J).

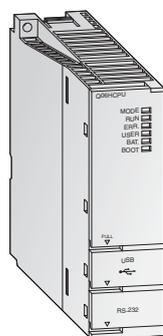
В то время как процессоры Q00 и Q01 представляют собой классические модули центральных процессоров, процессор Q00J образует неделимый блок из центрального процессора, сетевого блока и монтажной шины и является недорогой, "вступительной" моделью модульного контроллера.

Стандартные центральные процессоры специально разработаны для прикладных задач, у которых основными требованиями являются простота реализации и компактная конструкция системы.

#### Особенности:

- Каждый центральный процессор оснащен интерфейсом RS232C для простого программирования и контроля контроллера с компьютера или панели управления.
- Имеется встроенная флэш-ROM для использования памяти без дополнительного слота для карты памяти
- Обработка входов и выходов в режиме отображения процесса

### Мощные центральные процессоры контроллера



У мощных центральных процессоров главное внимание уделено высокой скорости обработки и расширяемости. Они оснащены многообразными функциями и оптимизированной средой программирования и отладки, чтобы обеспечить гибкое реагирование на любые системы.

Два процессора для управления процессами Q12PH и Q25PH оснащены расширенными функциями регулирования с 2 степенями свободы, каскадированным ПИД-регулированием и функцией автонастройки. Кроме того, в них имеются 52 дополнительных командных функции для управления процессами. Количество контуров ПИД-регулирования не ограничено.

#### Особенности:

- Каждый многопроцессорный H-CPU оснащен интерфейсом USB для простого и быстрого программирования и контроля контроллера с компьютера.
- Обработка входов и выходов в режиме отображения
- Арифметика с плавающей запятой в соответствии с IEEE 754
- Непосредственное обращение и обработка контуров ПИД-регулирования
- Математические функции, например, тригонометрические, экспоненциальные и логарифмические
- Замена модуля в режиме RUN (при использовании центральных процессоров для управления процессами)
- Возможен многопроцессорный режим с использованием до 4 модулей центральных процессоров.

### 2.6.1 Технические данные

Свойство		Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Тип управления		циклическая обработка сохраненной программы						
Управление вводом-выводом		обновление отображения процесса						
Язык программирования		языки по стандарту МЭК (IEC): релейно-контактная схема (LD), список инструкций (IL), язык функциональных блоков (FBD), структурированный текст (ST), язык последовательных функциональных схем (SFC)						
Скорость обработки	LD	160 нс	100 нс	79 нс	34 нс			
	MOV	560 нс	350 нс	237 нс	102 нс			
	смешанных команд на каждую мкс	2.0	2.7	4.4	10.3			
	сложение чисел с плавающей запятой	27 мкс*		1.8 мкс	0.78 мкс			
Количество команд (кроме команд для интеллектуальных специальных модулей)		249		363				
Вычислительные команды для чисел с плавающей запятой		возможны *		возможны				
Команды для обработки строковых величин		возможна только \$MOV		возможны				
Команды для ПИД-регулирования		возможны *		возможны				

\* Только у процессоров Q00/Q01, начиная с функциональной версии "B" (у этих процессоров первые 5 разрядов серийного номера равны "04122" или выше)

Свойство		Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Постоянное время цикла (запуск программы через неизменные интервалы)		от 1 до 2000 мс (параметрируется с шагом 1 мс)		от 0.5 до 2000 мс (параметрируется с шагом 0.5 мс)				
Память для хранения программ (количество шагов)		8 k	14 k	28 k	60 k	124 k	252 k	
Емкость памяти	встроенная память для хранения программ (дисковод 0)	94 кбайт		112 кбайт	240 кбайт	496 кбайт	1 Мб	
	Карта памяти RAM (дисковод 1)	—		в зависимости от установленной карты памяти (макс. 1 Мб)				
	Карта памяти RAM (дисковод 2)	—		в зависимости от установленной карты памяти (макс. 4 Мб в случае флэш-ROM, макс. 32 Мб в случае карт памяти ATA)				
	встроенная RAM (дисковод 3)	128 кбайт*	64 кбайт		256 кбайт			
	встроенная ROM (дисковод 4)	94 кбайт	112 кбайт	240 кбайт	496 кбайт	1 Мб		
	Общая область памяти для многопроцессорного режима	1 кбайт**		8 кбайт				

\* 64 кбайт при функциональной версии А

\*\* Только у процессоров Q00/Q01, начиная с функциональной версии "B" (у этих процессоров первые 5 разрядов серийного номера равны "04122" или выше)

**Количество операндов**

Операнд (символ)	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Маркеры (M)	8192		8192				
Фиксируемые маркеры (L)	2048		8192				
Маркеры связи (B)	2048		8192				
Таймеры (T)	512		2048				
Фиксируемые таймеры (ST)	0		0				
Счетчики (C)	512		1024				
Регистры данных (D)	11136		12288				
Регистры связи (W)	2048		8192				
Маркеры ошибок (F)	1024		2048				
Маркеры фронта (V)	1024		2048				

В предшествующей таблице перечислены предварительно настроенные операнды. Количество операндов можно изменять в параметрах.

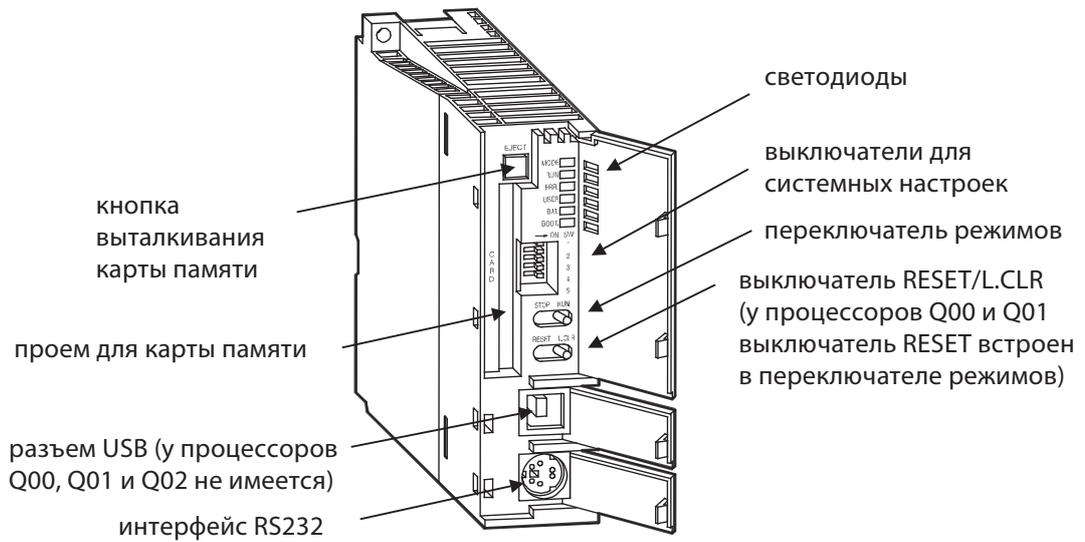
Операнд (символ)	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Регистры файлов (R)	32768		32768*	65536*		131072*	
Специальные маркеры связи (SB)	1024		2048				
Специальные регистры связи (SW)	1024		2048				
Шаговые маркеры (S)	2048 (S0...127 / блок)		8192				
Индексные регистры (Z)	10		16				
Указатели (P)	300		4096				
Указатели прерывания (I)	128		256				
Специальные маркеры (SM)	1024		2048				
Специальные регистры (SD)	1024		2048				
Функциональные входы	16		16				
Функциональные выходы	16		16				
Функциональные регистры	5		5				

\* В случае применения встроенной памяти. В центральных процессорах типов Q02, Q02H, Q06H, Q12H и Q25H количество регистров файлов можно увеличить до 1.042.432 адресов, применив карты памяти.

**Элементы управления, интерфейсы и потребляемый ток модулей центральных процессоров**

Свойство	Q00CPU	Q01CPU	Q02CPU	Q02HCPU	Q06HCPU	Q12HCPU	Q25HCPU
Функции переключателя режимов	RUN, STOP, RESET		RUN, STOP, RESET, L.CLR (стирание области фиксации)				
Интерфейсы	RS232		RS232	RS232, USB			
Слоты для карты памяти	—		1 слот				
Светодиоды для индикации рабочего состояния	RUN, ERR.		MODE, RUN, ERR., USER, BAT., BOOT, POWER				
Потребляемый ток при 5 В пост. т.	0.25 A	0.27 A	0.60 A	0.64 A			

**Элементы управления модулей центральных процессоров**



**Светодиоды**

- Светодиоды "MODE" и "RUN"

<p>Q06HCPU</p> <p>MODE <input checked="" type="checkbox"/></p> <p>RUN <input checked="" type="checkbox"/></p> <p>ERR. <input type="checkbox"/></p> <p>USER <input type="checkbox"/></p> <p>BAT. <input type="checkbox"/></p> <p>BOOT <input type="checkbox"/></p>	Зеленый: режим Q
	<p>Горит: процессор находится в режиме RUN</p> <p>Не горит: процессор находится в режиме STOP или возникла ошибка, прервавшая обработку программы</p> <p>Мигает: после изменения программы или параметра переключатель режимов процессора переключен на RUN, однако процессор еще не находится в режиме RUN</p>

Возврат центрального процессора в состояние "RUN" (например, после изменения программы или параметра, выполненных в режиме "STOP") осуществляется следующим образом:

1. Установить выключатель "RESET/L.CLR" в положение "RESET".
2. Установить выключатель "RUN/STOP" в положение "RUN".

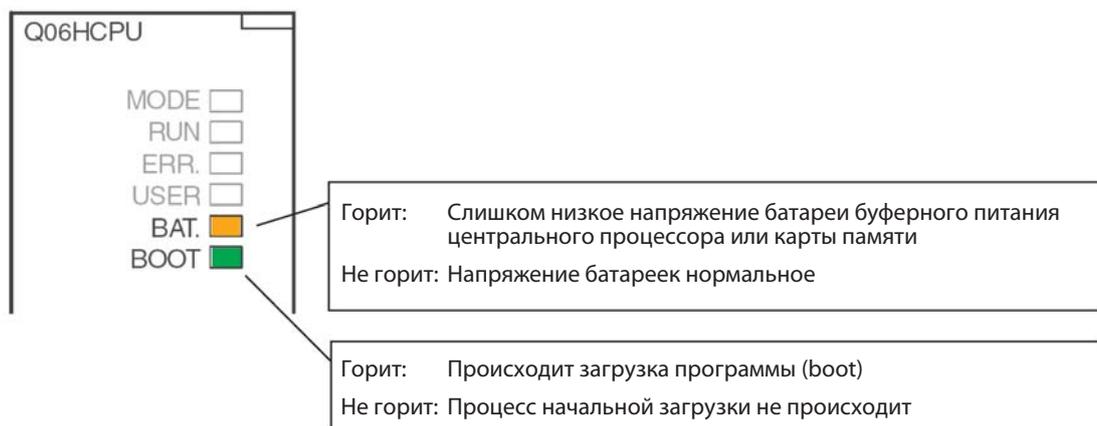
Или, если сброс выполнять не требуется:

1. Перевести выключатель "RUN/STOP" из положения "STOP" в положение "RUN"
2. Снова установить выключатель "RUN/STOP" в положение "STOP".
3. Переключить выключатель "RUN/STOP" в положение "RUN".

● Светодиоды "ERR." и "USER"



● Светодиоды "BAT" и "BOOT"



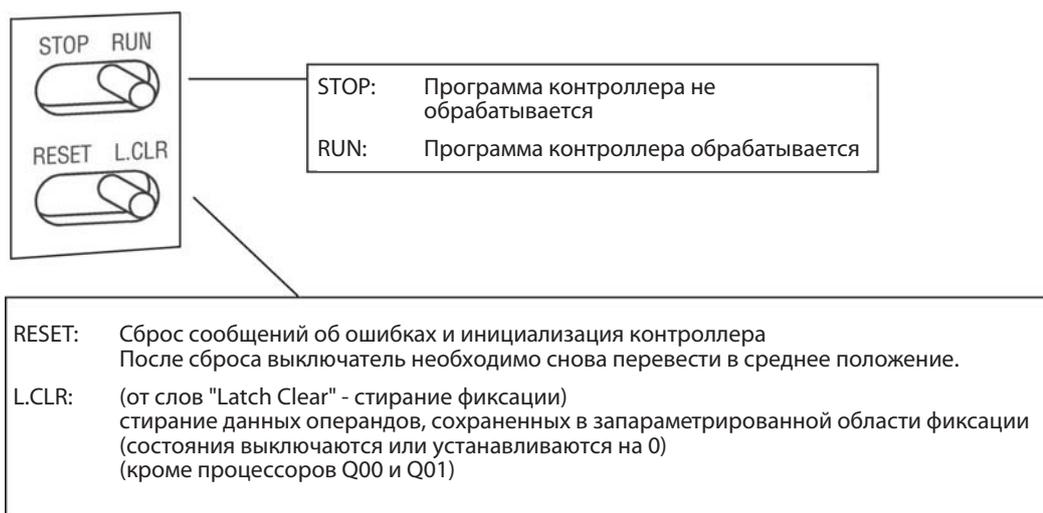
### Системные выключатели



Во встроенной RAM (дисковод 3) параметры сохранять невозможно.

При отправке модуля центрального процессора с завода-изготовителя все выключатели находятся в выключенном положении.

### Выключатели RUN/STOP , RESET/L.CLR



### Конфигурация памяти



### Что и где можно сохранять?

Процессоры Q00 и Q01

Данные	Встроенная память		
	Память для хранения программ (дисковод 0)	RAM (дисковод 3)	ROM (дисковод 4)
Программа	●	○	●
Параметры	●	○	●
Параметры для специальных модулей	●	○	●
Комментарии к операндам	●	○	●

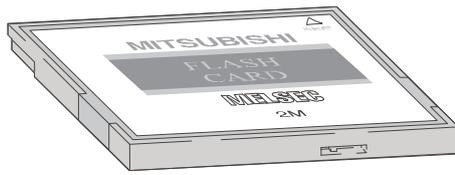
- = сохранение возможно
- = сохранение не возможно

Q02, Q02H, Q06H, Q12H и Q25H:

Данные	Встроенная память			Карты памяти		
	Память для хранения программ (дисковод 0)	RAM (дисковод 3)	ROM (дисковод 4)	RAM (дисковод 1)	Флэш-ROM (дисковод 2)	ATA ROM (дисковод 2)
Программа	●	○	●	●	●	●
Параметры	●	○	●	●	●	●
Параметры для специальных модулей	●	○	●	●	●	●
Комментарии к операндам	●	○	●	●	●	●
Значения инициализации	●	○	●	●	●	●
Регистры файлов	○	●	○	●	●	○
Локальные операнды	○	●	○	●	○	○
Данные трассировки	○	○	○	●	○	○
История ошибок	○	○	○	●	○	○
Данные, введенные с помощью команды FWRITE	○	○	○	○	○	●

- = сохранение возможно
- = сохранение не возможно

**Карты памяти**

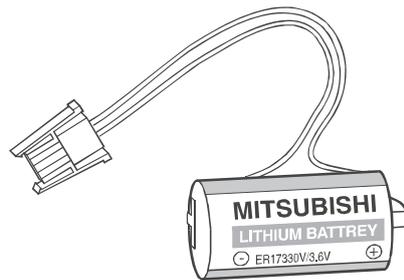


Сохраненные данные можно защитить от случайного стирания с помощью защиты от записи. В карте памяти SRAM имеется встроенная батарея, которая сохраняет данные при исчезновении напряжения питания.

**Предлагаемые карты памяти**

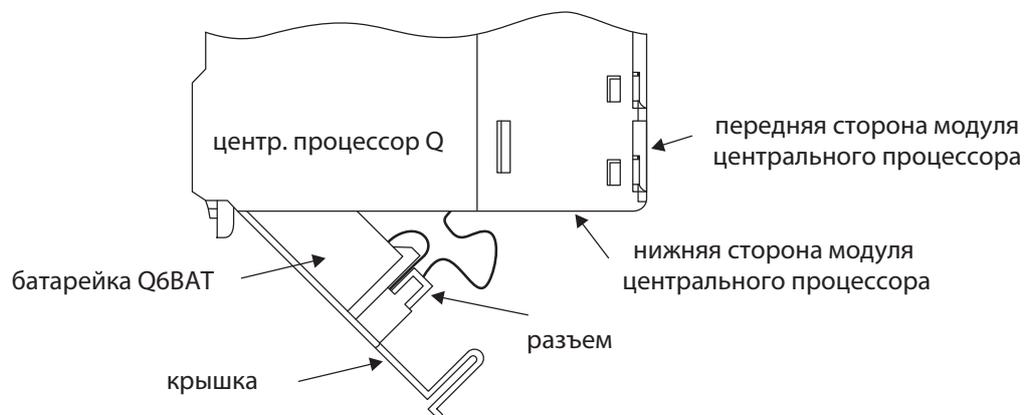
Обозначение	Тип памяти	Емкость памяти [байт]	Емкость памяти [файлов]	Количество процессов записи
Q2MEM-1MBS	SRAM	1011 k	256	Без ограничений
Q2MEM-2MBS		2034 k	288	
Q2MEM-2MBF	Флэш-ROM	2035 k	288	100 000
Q2MEM-4MBF		4079 k		
Q2MEM-8MBA	ATA ROM	7940 k	512	1 000 000
Q2MEM-16MBA		15932 k		
Q2MEM-32MBA		31854 k		

**Вставление батареи буферного питания в модуль центрального процессора**



Батарея расположена с нижней стороны модуля центрального процессора. При исчезновении напряжения она может на протяжении нескольких тысяч часов (в зависимости от типа центрального процессора) поддерживать содержимое памяти для хранения программ, встроенной RAM и часов центрального процессора.

При отправке модуля центрального процессора с завода-изготовителя батарея вложена в модуль, однако для защиты от коротких замыканий и предотвращения ее разрядки разъем между батареей и центральным процессором отсоединен. Перед вводом центрального процессора в эксплуатацию батарею необходимо подключить.



Батарею следует заменять каждые 10 лет.

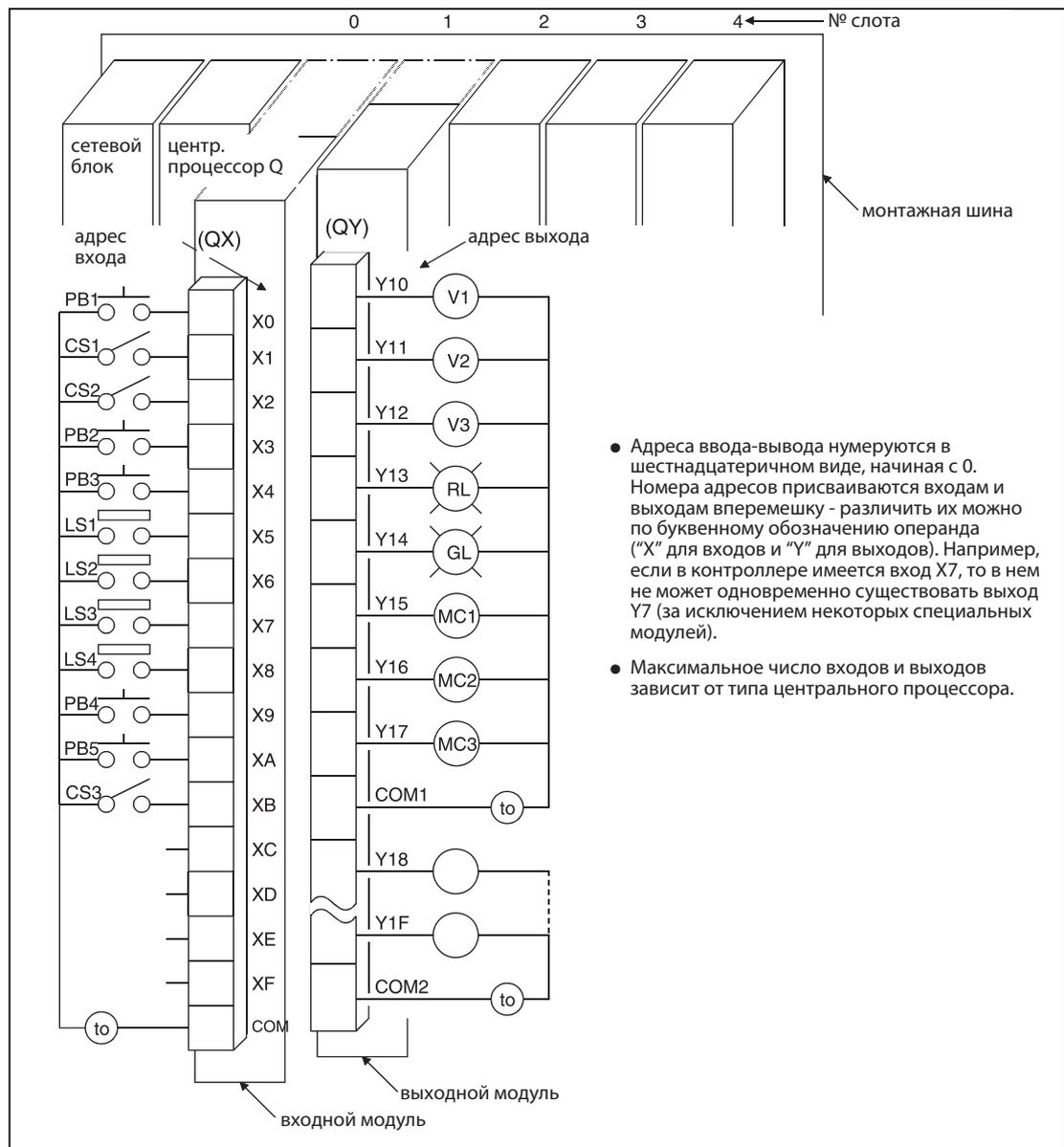
## 2.7 Подключение внешних сигналов

### 2.7.1 Электропроводка входов и выходов

Сигналы, подаваемые внешними устройствами на входы контроллера, для обработке в программе преобразуются в адреса входов. Адрес входа контроллера определяется слотом монтажной шины, в который вставлен входной модуль (см. раздел 2.3.3), и входом модуля, к которому подключен сигнал.

Адреса выходов, управляемых программой, также определяются слотом и клеммой модуля. Чтобы можно было переключить внешнее устройство, оно должно быть соединено с соответствующим выходом контроллера.

Входы и выходы адресуются в шестнадцатеричном виде (0, 1, 2 ...9, A, B, C, D, E, F). Поэтому образуются группы по 16 входов или выходов.



## 2.8 Цифровые входные и выходные модули

Входные и выходные модули соединяют центральный процессор контроллера с управляемым процессом. Цифровые входные модули преобразуют сигналы внешних устройств в информацию типа "включено/выключено" для центрального процессора. Цифровые выходные модули могут включать и выключать внешние коммутирующие устройства.

**Входные сигналы** могут вырабатываться различными датчиками и устройствами:

- кнопочными выключателями
- поворотными ручками с несколькими положениями
- выключателями, запираемыми ключом
- концевыми выключателями
- реле уровня
- датчиками для контроля расхода
- фоторелейными барьерами и фотореле
- бесконтактными выключателями (индуктивными или емкостными). Как правило, бесконтактные выключатели оснащены транзисторным выходом, работающим либо по положительной, либо по отрицательной логике.

**Выходные сигналы** управляют, например, следующими устройствами:

- контакторами
- сигнальными лампами
- электромагнитными клапанами
- входами внешних приборов, например, преобразователя частоты

### Обзор цифровых модулей ввода-вывода

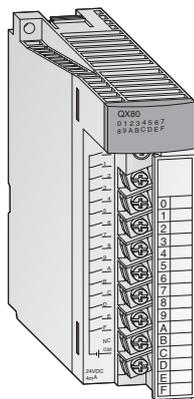
Тип модуля		Количество входов и выходов			
		8	16	32	64
Входные модули	120 В пер. т.	○	●	○	○
	240 В пер. т.	●	○	○	○
	24 В пост. т.	○	●	●	●
	24 В пост. т. (быстрые входы)	●	○	○	○
	5 / 12 В пост. т.	○	●	●	●
Выходные модули	Релейные	●	●	○	○
	Реле с отдельными контактами	●	○	○	○
	Симисторные	○	●	○	○
	Транзисторные (отрицательная логика)	●	●	●	●
	Транзисторные (положительная логика)	○	●	●	○
Комбинированные модули ввода-вывода		●	○	●	○

● = модуль имеется

○ = модуля не имеется

## 2.8.1 Цифровые входные модули

Имеются цифровые входные модули для различных входных напряжений:



Входное напряжение	Количество входов			
	8	16	32	64
5 ... 12 В пост. т.		QX70	QX71	QX72
24 В пост. т.		QX80	QX81	QX82
24 В пост. т. (модуль прерываний)		QI60		
100 ... 120 В пер. т.		QX10		
100 ... 240 В пер. т.	QX28			

У входных модулей с 8 или 16 входами внешние сигналы подключаются через съемные колодки с винтовыми клеммами. Модули с 32 или 64 входами подключаются с помощью разъемов.

### Основные сведения о цифровых входных модулях

Все входы изолированы с помощью оптронов. Благодаря этому на чувствительную электронику контроллера не влияют электромагнитные помехи, порождаемые внешними устройствами.

Другой часто встречающейся проблемой является дребезг контактов механических выключателей. Чтобы эти помехи не влияли на контроллер, входные сигналы фильтруются. Изменившееся состояние сигнала регистрируется только в том случае, если он приложено ко входу в течение определенного времени. Таким образом, контроллер не интерпретирует в качестве входных сигналов кратковременные сигналы помех.

#### ПРИМЕЧАНИЕ

**Серия "А":** время фильтра для стандартных входных модулей предварительно установлено на 10 мс.

**System Q:** Для стандартных входных модулей время фильтра предварительно установлено на 10 мс. Однако в параметрах эту предварительную настройку можно изменить для каждого установленного модуля в диапазоне от 1 до 70 мс. Учитывайте при этом технические данные модулей.

Настроенное время фильтра влияет на время реакции контроллера, поэтому его следует учитывать при программировании. При малом времени фильтра уменьшается время реакции контроллера, однако одновременно возрастает его чувствительность к внешним помехам. В этом случае входные сигналы следует подавать по экранированным проводам, и эти сигнальные провода следует прокладывать отдельно от проводки, являющейся потенциальным источником помех. Если требуется очень малое время реакции, следует применять специальные модули (например, модуль прерываний QI60).

Чтобы контроллер распознавал включенный вход, через этот вход должен течь (на вход или из него) минимальный ток. Этот ток зависит от типа входного модуля и в большинстве случаев равен 3 мА. Если вход предположительно включен, но этот ток не достигнут, центральный процессор продолжает считать этот вход выключенным. Входной ток ограничивается внутренним сопротивлением входного модуля. Если из-за слишком высокого входного напряжения течет слишком большой входной ток, входной модуль повреждается. Допускаются входные токи до 7 мА.

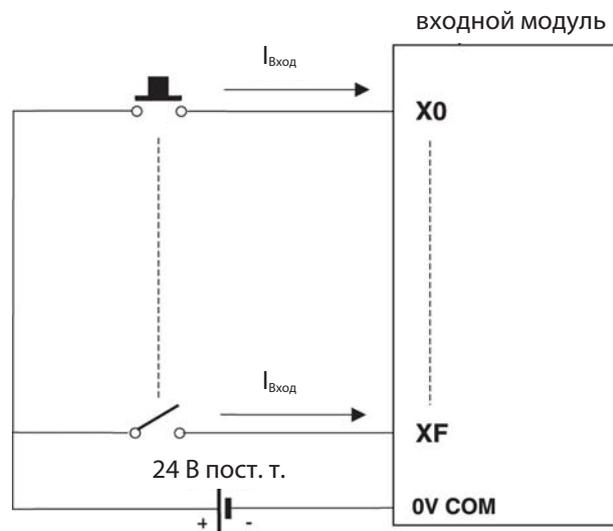
Центральный процессор контроллера определяет состояние входов в начале циклической обработки программы и сохраняет эту информацию в памяти. В программе обрабатываются лишь сохраненные состояния. В очередной раз состояния входов обновляются лишь перед следующим прогоном программы.

### Положительная и отрицательная логика входов

В контроллерах "MELSEC System Q" имеются входные модули постоянного напряжения для датчиков, работающих по принципу положительной или отрицательной логики. К некоторым модулям QX71 можно по выбору подключать датчики положительной или отрицательной логики. В англоязычной литературе положительная логика датчиков обозначается словом "Source" ("источник", т. е. источник тока), а отрицательная логика - словом "Sink" (сток). Эти обозначения относятся к направлению, в котором ток течет при включенном входе. На немецком языке положительная логика обозначается словом "plusschaltend" ("подключающий к плюсу"), а отрицательная логика - словом "minusschaltend" ("подключающий к минусу").

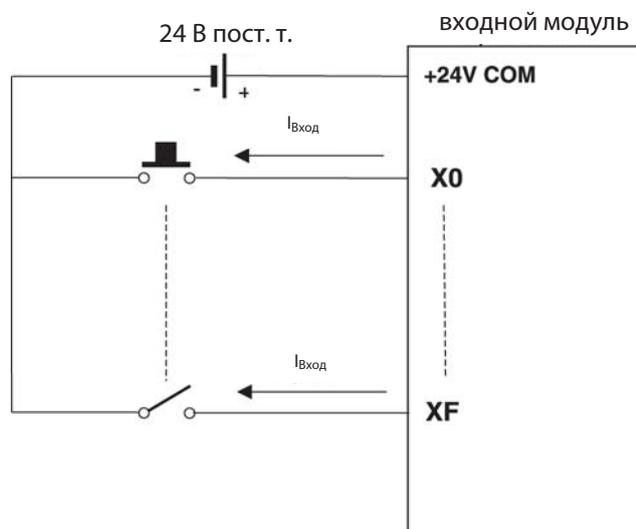
### Подключение датчика с положительной логикой ("Source")

Датчик с положительной логикой соединяет плюсовой полюс источника напряжения с входом контроллера. Минусовой полюс источника напряжения образует общий потенциал для всех входов группы. При включенном датчике ток течет во входной модуль, отсюда английское обозначение "Source" - т. е. датчик работает в качестве источника тока.



### Подключение датчика с отрицательной логикой ("Sink")

Датчик с отрицательной логикой соединяет минусовой полюс источника напряжения с входом контроллера. Общим потенциалом для всех входов одной группы является плюсовой полюс источника напряжения. При включенном датчике ток течет из входного модуля, т. е. датчик выступает в качестве стока. Отсюда английское обозначение "Sink".



### **Бесконтактные выключатели и оптические датчики**

**Бесконтактные выключатели** или выключатели приближения - это выключатели, переключающиеся без механического воздействия. Они посылают сигнал на контроллер, если объект приблизился к выключателю на малое расстояние. От объекта не требуется прикосновения к выключателю. Такие выключатели находят широкое применение в автоматизации установок. Бесконтактные выключатели могут работать по индуктивному или емкостному принципу.

Кроме того, в промышленных системах управления широко распространены оптические датчики в виде фоторелейных барьеров или фотореле. (В фоторелейных барьерах имеется зеркало, возвращающее луч света. В случае фотореле излученный свет отражается самим объектом.)

Бесконтактные выключатели и фоторелейные барьеры или фотореле оснащены внутренней электроникой, для которой в большинстве случаев необходимо постоянное напряжение питания 24 В. Выходы этих электронных выключателей, как правило, представляют собой транзисторные выходы, работающие по принципу положительной или отрицательной логики:

- транзисторный выход типа pnp: положительная логика (source)
- транзисторный выход типа npn: отрицательная логика (sink)

**Пример входного модуля для датчиков с положительной логикой**

Свойство		Технические данные
Обозначение модуля		QX80
Входы		16
Развязка		с помощью оптрона
Номинальное входное напряжение		24 В пост. т. (+20/-15 %, пульсации до 5 %)
Входной ток		ок. 4 мА
Одновременно включаемые входы		100 % (все входы могут быть включены одновременно)
Пик тока включения		Макс. 200 мА в течение 1 мс (при 132 В пер. т.)
Напряжение и ток для включения		≥ 19 В пост. т. / ≥ 3 мА
Напряжение и ток для выключения		≤ 11 В пост. т. / ≤ 1.7 мА
Входное сопротивление		Ок. 5.6 кОм
Время реагирования	ВЫКЛ. → ВКЛ.	1, 5, 10, 20, 70 мс (параметрируется, предварительная настройка: 10 мс)*
	ВКЛ. → ВЫКЛ.	1, 5, 10, 20, 70 мс (параметрируется, предварительная настройка: 10 мс)*
Диэлектрическая прочность		Действующее значение 560 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопротивление изоляции		≥ 10 МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		Испытана с помощью имитатора помех (пиковое напряжение помех: 500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25...60 Гц) высокочастотное аperiodическое напряжение помех (IEC61000-4-4): 1 кВ
Группы входов		1 группа с 16 входами, общий потенциал: клемма 18
Индикация состояния входов		по одному светодиоду на вход
Подсоединение электропроводки		клеммная колодка с 18 винтовыми клеммами (М3 х 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм
Внутреннее потребление тока (5 В пост. т.)		50 мА (если все входы включены)
Вес		0.16 кг

\* Время реагирования при переключении из состояния "ВЫКЛ." в состояние "ВКЛ.", и из состояния "ВКЛ." в состояние "ВЫКЛ." отдельно регулировать невозможно.

Внешний вид модуля	Схема	Клемма	Сигнал
		1	X00
		2	X01
		3	X02
		4	X03
		5	X04
		6	X05
		7	X06
		8	X07
		9	X08
		10	X09
		11	X0A
		12	X0B
		13	X0C
		14	X0D
		15	X0E
		16	X0F
		17	не используется
		18	COM

**Функция входного модуля с датчиками положительной логики**

При включении подключенного к входному модулю датчика (например, кнопочного выключателя с функцией замыкающего контакта) вход контроллера включается. При этом происходят следующие процессы, относящиеся к схеме на предыдущей странице:

- При нажатой кнопке плюсовой полюс внешнего 24-вольтового источника напряжения соединяется с клеммой 1 входного модуля.
- Клемма 1 через резистор и светодиод оптрона соединена с минусовым полюсом внешнего источника напряжения (соединение 18). Таким образом, ток течет через светодиод оптрона.
- Ток вызывает свечение светодиода, в результате чего фототранзистор оптрона переходит в проводящее состояние.
- Благодаря оптрону внешнее входное напряжение отделено от напряжения питания контроллера. Поэтому помехи, которые в промышленном окружении часто накладываются на внешние постоянные напряжения, не проникают в напряжение питания контроллера. Кроме того, благодаря оптрону вход становится нечувствительным к помехам.
- Если фототранзистор оптрона открыт, во входную логику модуля передается сигнал. В этом примере электроника регистрирует включенное состояние входа X0. В этом случае загорается светодиод с передней стороны входного модуля, что указывает на данное состояние сигнала.

**Пример входного модуля для датчиков с отрицательной логикой**

Свойство		Технические данные
Обозначение модуля		QX40
Входы		16
Развязка		С помощью оптрона
Номинальное входное напряжение		24 В пост. т. (+20/-15 %, пульсации до 5 %)
Входной ток		Ок. 4 мА
Одновременно включаемые входы		100 % (все входы могут быть включены одновременно)
Пик тока включения		Макс. 200 мА в течение 1 мс (при 132 В пер. т.)
Напряжение и ток для включения		$\geq 19$ В пост. т. / $\geq 3$ мА
Напряжение и ток для выключения		$\leq 11$ В пост. т. / $\leq 1.7$ мА
Входное сопротивление		Ок. 5.6 кОм
Время реагирования	ВЫКЛ. → ВКЛ.	1, 5, 10, 20, 70 мс (параметрируется, предварительная настройка: 10 мс)*
	ВКЛ. → ВЫКЛ.	1, 5, 10, 20, 70 мс (параметрируется, предварительная настройка: 10 мс)*
Диэлектрическая прочность		Действующее значение 560 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопротивление изоляции		$\geq 10$ МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		Испытана с помощью имитатора помех (пиковое напряжение помех: 500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25...60 Гц) Высокочастотное аperiодическое напряжение помех (IEC61000-4-4): 1 кВ
Группы входов		1 группа с 16 входами, общий потенциал: клемма 17
Индикация состояния входов		По одному светодиоду на вход
Подсоединение электропроводки		Клеммная колодка с 18 винтовыми клеммами (М3 x 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм

\* Время реагирования при переключении из состояния "ВЫКЛ." в состояние "ВКЛ.", и из состояния "ВКЛ." в состояние "ВЫКЛ." отдельно регулировать невозможно.

Внешний вид модуля	Схема	Клемма	Сигнал
		1	X00
		2	X01
		3	X02
		4	X03
		5	X04
		6	X05
		7	X06
		8	X07
		9	X08
		10	X09
		11	X0A
		12	X0B
		13	X0C
		14	X0D
		15	X0E
		16	X0F
		17	COM
		18	не используется

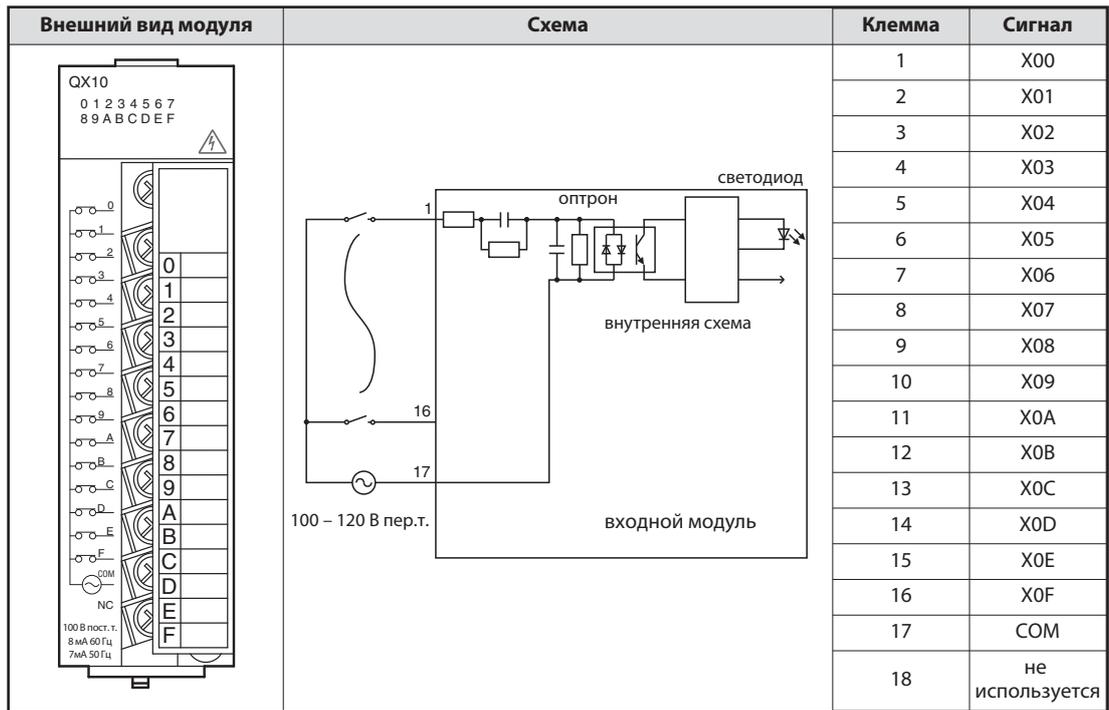
### Функция входного модуля с датчиками отрицательной логики

При включении выключателя, подключенного к клемме 1 на схеме предыдущей страницы, ток течет следующим образом:

- От плюсового полюса внешнего 24-вольтового источника напряжения к клемме общего потенциала (клемма 17).
- Через светодиод оптрона и добавочный резистор к клемме 1 входного модуля (являющейся входом X0).
- Ток через светодиод оптрона вызывает его свечение. В результате этого включается фототранзистор оптрона.
- Если фототранзистор оптрона открыт, во входную логику модуля передается сигнал. В этом примере электроника регистрирует включенное состояние входа X0. В этом случае загорается светодиод с передней стороны входного модуля, что указывает на данное состояние сигнала.
- Из клеммы X0 ток через включенный выключатель течет к минусовому полюсу внешнего источника напряжения.

### Пример входного модуля для переменных напряжений

Свойство		Технические данные
Обозначение модуля		QX10
Входы		16
Развязка		С помощью оптрона
Номинальное входное напряжение		100 ... 120 В пер. т. (+10/-15 %) 50/60 Гц ( $\pm 3$ Гц) (искажения до 5 %)
Входной ток		Ок. 8 мА при 100 В пер. т., 60 Гц; ок. 7 мА при 100 В пер. т., 50 Гц
Одновременно включаемые входы		См. диаграмму
Пик тока включения		Макс. 200 мА в течение 1 мс (при 132 В пер. т.)
Напряжение и ток для включения		$\geq 80$ В пер. т. / $\geq 5$ мА (50 Гц, 60 Гц)
Напряжение и ток для выключения		$\leq 30$ В пер. т. / $\leq 1$ мА (50 Гц, 60 Гц)
Входное сопротивление		Ок. 15 кОм при 60 Гц, ок. 18 кОм при 50 Гц
Время реагирования	ВЫКЛ. $\rightarrow$ ВКЛ.	$\leq 15$ мс (100 В пер. т., 50 Гц, 60 Гц)
	ВКЛ. $\rightarrow$ ВЫКЛ.	$\leq 20$ мс (100 В пер. т., 50 Гц, 60 Гц)
Диэлектрическая прочность		Действующее значение 1780 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопротивление изоляции		$\geq 10$ МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		Испытана с помощью имитатора помех (пиковое напряжение помех: 1500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25...60 Гц) Высокочастотное аperiодическое напряжение помех (IEC61000-4-4): 1 кВ
Группы входов		1 группа с 16 входами, общий потенциал: клемма 17
Индикация состояния входов		По одному светодиоду на вход
Подсоединение электропроводки		Клеммная колодка с 18 винтовыми клеммами (M3 x 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм
Внутреннее потребление тока (5 В пост. т.)		50 мА
Вес		0.17 кг

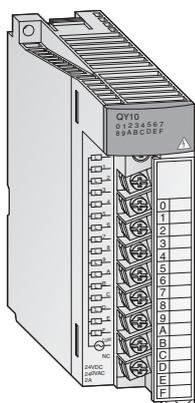


Максимальное количество одновременно включаемых входов модуля QX10 зависит от температуры окружающего воздуха.

У входных модулей для переменных напряжений для переключения входов должно использоваться то же самое напряжение (переменное 100...120 В), которое используется для питания контроллера. Так предотвращается подключение ко входам неправильного напряжения.

## 2.8.2 Цифровые выходные модули

Благодаря многообразию переключающих элементов, выходные модули способны решать любые задачи управления:



Тип выхода	Номинальное напряжение	Количество выходов		
		8	16	32
Релейные	24 В пост. т. / 240 В пер. т.	QY18A	QY10	
Симисторные	100 - 240 В пер. т.		QY22	
Транзисторные	5/12 В пост. т.		QY70	QY71
	12/24 В пост. т.		QY80	QY81P
	5 ... 24 В пост. т.	QY68A		

Модули с 8 или 16 выходами имеют съемные клеммные колодки с винтовыми клеммами для выходных сигналов. Модули с 32 выходами подключаются с помощью разъема.

### Типы выходов

Цифровые выходные модули MELSEC System Q могут иметь выходы четырех типов.

- релейные
- симисторные
- транзисторные (положительная логика)
- транзисторные (отрицательная логика)

Тип	Преимущества	Недостатки
Релейные	<ul style="list-style-type: none"> <li>● Модуль может коммутировать различные напряжения</li> <li>● Беспотенциальные контакты</li> <li>● Возможна коммутация больших токов</li> </ul>	<ul style="list-style-type: none"> <li>● Медленные (макс. 1 Гц)</li> <li>● Ограниченный срок службы (электромеханический)</li> <li>● Опасность обгорания переключающих контактов</li> <li>● Громкие (переключение слышно)</li> </ul>
Симисторные	<ul style="list-style-type: none"> <li>● Надежные</li> <li>● Высокая скорость переключения</li> <li>● Удовлетворяют высоким требованиям</li> </ul>	<ul style="list-style-type: none"> <li>● Коммутируют только переменное напряжение</li> <li>● Макс. коммутируемый ток 0.6 А на каждый выход</li> <li>● Время переключения 10 мс при напряжении 50 Гц</li> </ul>
Транзисторные	<ul style="list-style-type: none"> <li>● очень надежные</li> <li>● очень высокая скорость переключения</li> <li>● особенно хорошо пригодны для высоких требований</li> </ul>	<ul style="list-style-type: none"> <li>● коммутируют только низкие постоянные напряжения</li> <li>● макс. коммутируемый ток 0.1 А на каждый выход</li> </ul>

### Релейные выходные модули

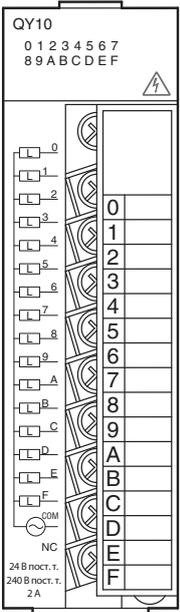
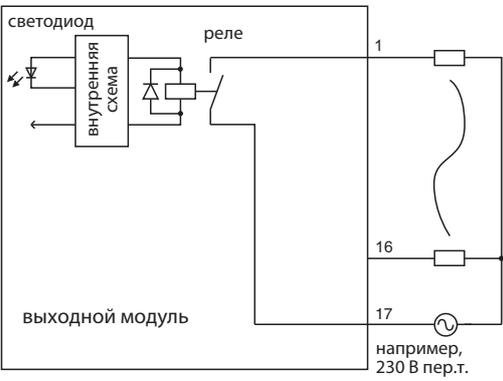
Каждый выход релейного выходного модуля содержит реле, переключающий контакт которого коммутирует подключенное силовое напряжение. Так обеспечивается разделение внутреннего напряжения контроллера и внешней нагрузки.

Имеются релейные выходные модули с общим потенциалом и модули с независимыми, беспотенциальными контактами реле.

Как и у других выходных модулей, выход управляется программой контроллера. В конце программы происходит обновление выходов контроллера. Это означает, что в этот момент все логические состояния выходов, образованные программой, передаются на физические выходы. Включенный выход отображается светящимся светодиодом. Так можно контролировать состояние выхода непосредственно на контроллере. Время реакции релейного выходного модуля приблизительно равно 10 мс.

### Пример релейного выходного модуля

Свойство		Технические данные
Обозначение модуля		QY10
Выходы		16
Развязка		С помощью реле
Номинальное выходное напряжение / ток		24 В пост. 2 А (омическая нагрузка) на каждый выход 240 В пер. 2 А ( $\cos\varphi = 1$ ) на каждый выход; макс. 8 А на группу
Минимальная коммутируемая мощность		5 В пост. т., 1 мА
Макс. коммутируемое напряжение		125 В пост. т. / 264 В пер. т.
Время реагирования	ВЫКЛ. → ВКЛ.	≤ 10 мс
	ВКЛ. → ВЫКЛ.	≤ 12 мс
Срок службы контактов	Механический	≥ 20 млн. переключений
	Электрический	≥ 100000 переключений при номинальном выходном напряжении / токе
		≥ 100000 переключений при 200 В пер. т., 1.5 А; 240 В пер. 1 А ( $\cos\varphi = 0.7$ )
		≥ 300000 переключений при 200 В пер. т., 0.4 А; 240 В пер. 0.3 А ( $\cos\varphi = 0.7$ )
		≥ 100000 переключений при 200 В пер. т., 1 А; 240 В пер. 0.5 А ( $\cos\varphi = 0.35$ ) ≥ 300000 переключений при 200 В пер. т., 0.3 А; 240 В пер. 0.15 А ( $\cos\varphi = 0.35$ )
≥ 100000 переключений при 24 В пост. 1 А; 100 В пост. 0.1 А ( $L/R = 0.7$ мс) ≥ 300000 переключений при 24 В пост. 0.3 А; 100 В пост. 0.03 А ( $L/R = 0.7$ мс)		
Макс. частота переключений		3600 переключений/час
Сетевой фильтр		—
Предохранитель		—
Диэлектрическая прочность		Действующее значение 2830 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопrotивление изоляции		≥ 10 МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		Испытана с помощью имитатора помех (пиковое напряжение помех: 1500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25...60 Гц) Высокочастотное апериодическое напряжение помех (IEC61000-4-4): 1 кВ
Группы выходов		1 группа с 16 выходами, общий потенциал: клемма 17
Индикация состояния выходов		По одному светодиоду на выход
Подсоединение электропроводки		Клеммная колодка с 18 винтовыми клеммами (М3 х 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм
Внутреннее потребление тока (5 В пост. т.)		430 мА
Вес		0.22 кг

Внешний вид модуля	Схема	Клемма	Сигнал
		1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	COM
			не используется

### Симисторные выходные модули

Цифровые симисторные выходные модули коммутируют переменные напряжения от 100 до 240 В. Коммутируемое напряжение отделено от напряжения питания контроллера с помощью оптрона. Время реакции симисторных выходных модулей меньше, чем у релейных. Для включения нужна лишь 1 мс, а для выключения - 10 мс.

Симистор может коммутировать токи до 0.6 А. Установку с симисторными выходными модулями необходимо рассчитать так, чтобы не превышался этот максимальный коммутируемый ток.

Даже при выключенном выходе через симистор течет ток утечки макс. 3 мА. Из-за этого небольшого ток сигнальные лампы могут продолжать гореть даже при выключенном выходе. Кроме того, этот ток может вызывать притягивание контактов маломощных реле.



**ОПАСНОСТЬ:**

*В связи с током утечки имеется опасность удара током даже при выключенном симисторном выходе. Перед работами на электроустановке обязательно выключайте всё электропитание.*

**Пример симисторного выходного модуля**

Свойство		Технические данные
Обозначение модуля		QY22
Выходы		16
Развязка		с помощью оптрона
Номинальное выходное напряжение / ток		100 ... 240 В пер. (+20/-15%), 0.6 А на каждый выход, 4.8 А на модуль
Минимальная коммутируемая мощность		24 В пер. т., 100 мА; 100 В пер. т., 25 мА, 240 В пер. т., 25 мА
Макс. пик тока включения		20 А
Ток утечки при выключенном выходе		≤ 3 мА при 120 В пер. т., 60 Гц ≤ 1.5 мА при 240 В пер. т., 60 Гц
Макс. падение напряжения при включенном выходе		1,5 В
Время реагирования	Выкл. → Вкл.	0.5 x длительность периода + макс. 1 мс
	Вкл. → Выкл.	0.5 x длительность периода + макс. 1 мс
Сетевой фильтр		RC-элемент
Предохранитель		—
Диэлектрическая прочность		действующее значение 2830 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопротивление изоляции		≥ 10 МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		испытана с помощью имитатора помех (пиковое напряжение помех: 1500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25...60 Гц)
		высокочастотное апериодическое напряжение помех (IEC61000-4-4): 1 кВ
Группы выходов		1 группа с 16 выходами, общий потенциал: клемма 17
Индикация состояния выходов		по одному светодиоду на выход
Подсоединение электропроводки		клеммная колодка с 18 винтовыми клеммами (М3 x 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм

Внешний вид модуля	Схема	Клемма	Сигнал
		1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	COM
		18	не используется

### Транзисторные выходные модули

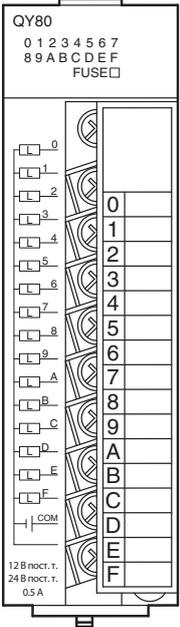
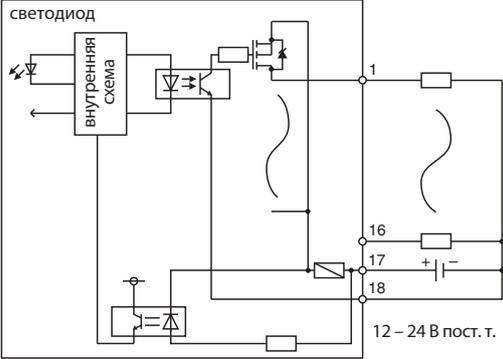
У транзисторных выходных модулей коммутируемое напряжение также отделено от напряжения питания контроллера с помощью оптрона.

Транзисторный выходной модуль переключает выход всего за 1 мс. Технические данные, например, коммутируемые токи указаны в руководствах по эксплуатации модулей или в руководстве по монтажу входных и выходных модулей (артикул 141758).

В системе контроллеров MELSEC System Q предлагаются выходные модули с положительной и отрицательной логикой.

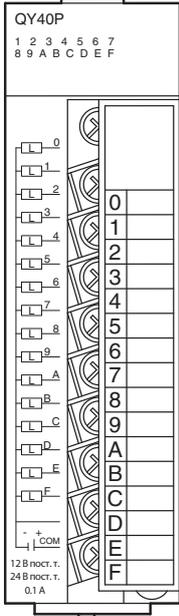
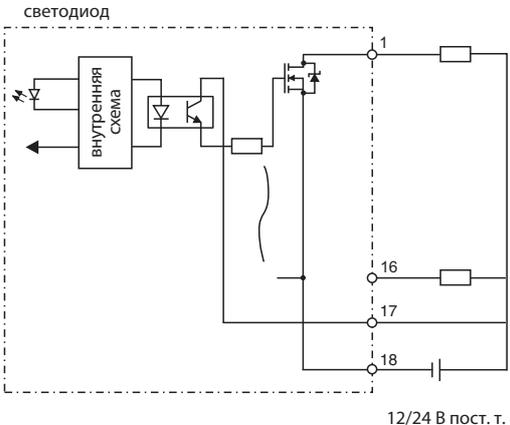
#### Пример выходного модуля с положительной логикой

Свойство		Технические данные
Обозначение модуля		QY80
Выходы		16
Развязка		С помощью оптрона
Номинальное выходное напряжение		От 12 до 24 В пост. т. (+20/-15 %)
Диапазон выходного напряжения		От 10.2 до 28.8 В пост. т.
Макс. условие переключения		0.5 А на каждый выход, 4 А на группу
Макс. пик тока включения		4 А в течение 10 мс
Ток утечки при выключенном выходе		≤0.1 мА
Падение напряжения при включенном выходе		Типичное значение 0.2 В пост. при 0.5 А, макс. 0.3 В при 0.5 А
Время реагирования	ВЫКЛ. → ВКЛ.	≤1 мс
	ВКЛ. → ВЫКЛ.	≤1 мс (при номинальных условиях переключения и омической нагрузке)
Сетевой фильтр		Стабилитрон
Предохранитель		6.7 А; незаменяемый
Индикация неисправного предохранителя		загорание светодиода и подача сигнала на центральный процессор
Питание модуля	Напряжение	12 ... 24 В пост. т. (+20/-15 %, пульсации 5 %)
	Ток	20 мА (при 24 В пост. т. и если все выходы включены)
Диэлектрическая прочность		Действующее значение 560 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопrotивление изоляции		≥10 МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		Испытана с помощью имитатора помех (пиковое напряжение помех: 500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25...60 Гц) Высокочастотное апериодическое напряжение помех (IEC61000-4-4): 1 кВ
Группы выходов		1 группа с 16 выходами, общий потенциал: клемма 17
Индикация состояния выходов		По одному светодиоду на выход
Подсоединение электропроводки		Клеммная колодка с 18 винтовыми клеммами (М3 х 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм

Внешний вид модуля	Схема	Клемма	Сигнал
 <p>QY80 0 1 2 3 4 5 6 7 8 9 A B C D E F FUSE □</p> <p>0 1 2 3 4 5 6 7 8 9 A B C D E F COM</p> <p>12 В пост. т. 24 В пост. т. 0.5 А</p>	 <p>светодиод</p> <p>внутренняя схема</p> <p>1</p> <p>16</p> <p>17</p> <p>18</p> <p>12 - 24 В пост. т.</p>	1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	COM
		18	0 B

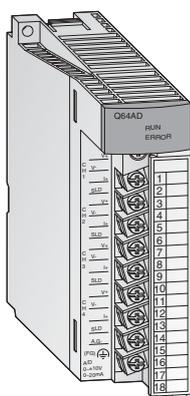
**Пример выходного модуля с отрицательной логикой**

Свойство		Технические данные
Обозначение модуля		QY40P
Выходы		16
Развязка		С помощью оптрона
Номинальное выходное напряжение		12 ... 24 В пост. т. (+20/-15 %)
Диапазон выходного напряжения		10.2 ... 28.8 В пост. т.
Макс. условие переключения		0.1 А на каждый выход, 1.6 А на группу
Макс. пик тока включения		0.7 А в течение 10 мс
Ток утечки при выключенном выходе		≤0.1 мА
Падение напряжения при включенном выходе		Типичное значение 0.1 В пост. при 0.1 А, макс. 0.2 В при 0.1 А
Время реагирования	ВЫКЛ. → ВКЛ.	≤1 мс
	ВКЛ. → ВЫКЛ.	≤1 мс (при номинальных условиях переключения и омической нагрузке)
Сетевой фильтр		Стабилитрон
Предохранитель		—
Индикация неисправного предохранителя		Загорание светодиода и подача сигнала на центральный процессор
Питание модуля	Напряжение	12 ... 24 В пост. т. (+20/-15 %, пульсации 5 %)
	Ток	10 мА (при 24 В пост. т., если включены все выходы)
Диэлектрическая прочность		Действующее значение 560 В пер. т. на протяжении 3 циклов (при применении на высоте 2000 м)
Сопротивление изоляции		≥10 МОм (при измерении с помощью прибора для проверки изоляции)
Помехоустойчивость		Испытана с помощью имитатора помех (пиковое напряжение помех: 500 В, длительность включения напряжения помех: 1 мкс, частота напряжения помех: 25 ... 60 Гц)
		Высокочастотное аперiodическое напряжение помех (IEC61000-4-4): 1 кВ
Группы выходов		1 группа с 16 выходами, общий потенциал: клемма 18
Индикация состояния выходов		По одному светодиоду на выход
Подсоединение электропроводки		Клеммная колодка с 18 винтовыми клеммами (М3 х 6)
Рекомендуемое поперечное сечение проводника		0.3 ... 0.75 мм <sup>2</sup> , макс. диаметр жил: 2.8 мм
Внутреннее потребление тока (5 В пост. т.)		65 мА
Вес		0.16 кг

Внешний вид модуля	Схема	Клемма	Сигнал
 <p>QY40P 1 2 3 4 5 6 7 8 9 A B C D E F</p> <p>0 1 2 3 4 5 6 7 8 9 A B C D E F</p> <p>COM 12 В пост. т. 24 В пост. т. 0,1 А</p>	 <p>светодиод</p> <p>внутренняя схема</p> <p>12/24 В пост. т.</p>	1	Y00
		2	Y01
		3	Y02
		4	Y03
		5	Y04
		6	Y05
		7	Y06
		8	Y07
		9	Y08
		10	Y09
		11	Y0A
		12	Y0B
		13	Y0C
		14	Y0D
		15	Y0E
		16	Y0F
		17	12/24 В пост. т.
		18	COM

## 2.9 Специальные модули

### 2.9.1 Аналоговые входные модули



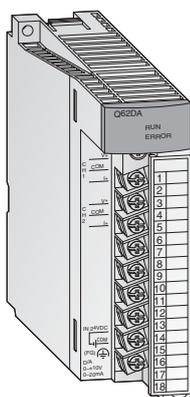
Для преобразования аналоговых сигналов процесса в цифровые значения для их дальнейшей обработки в центральном процессоре применяются аналоговые входные модули.

Модули "System Q" сочетают в себе высокую разрешающую способность до 0.333 мВ или 1.33 мА с чрезвычайно малым временем преобразования - всего 80 мкс на каждый вход.

У всех модулей входные сигналы подключаются с помощью съемной клеммной колодки с винтовыми клеммами.

Вид входа	Номинальный входной диапазон	Регулируемый входной диапазон	Количество входов	
			4	8
Напряжение	-10 ... +10 В	1 ... 5 В 0 ... 5 В 0 ... 10 В -10 ... +10 В		Q68ADV
Ток	0 ... 20 мА	0 ... 20 мА 4 ... 20 мА		Q68ADI
Напряжение или ток (для каждого входа можно выбрать отдельно)	-10 ... +10 В 0 ... 20 мА	Как у Q68ADV и Q68ADI	Q64AD	

### 2.9.2 Аналоговые выходные модули



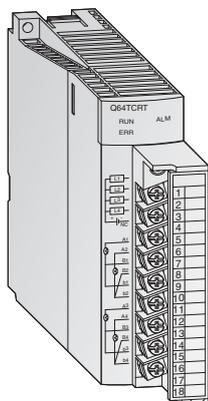
Аналоговые выходные модули преобразуют цифровые значения в аналоговый токовый или потенциальный сигнал. При чрезвычайно малом времени преобразования (всего 80 мкс на каждый выход) достигается разрешающая способность до 0.333 мВ или 0.83 мкА. Выходы устойчивы к короткому замыканию и изолированы от контроллера с помощью оптрона.

У всех модулей для подключения используется съемная клеммная колодка с винтовыми клеммами.

Вид выхода	Номинальный выходной диапазон	Регулируемый выходной диапазон	Количество выходов		
			2	4	8
Напряжение или ток (для каждого выхода можно выбрать отдельно)	-10 ... +10 В 0 ... 20 мА	1 ... 5 В -10 ... +10 В 0 ... 20 мА 4...20 мА	Q62DA	Q64DA	
Напряжение	-10 ... +10 В	-10 ... +10 В			Q68DAV
Ток	0 ... 20 мА	0 ... 20 мА 4...20 мА			Q68DAI

### 2.9.3 Модули регулирования температуры с ПИД-алгоритмом

Модули регулирования температуры позволяют регулировать температуру, не загружая задачами регулирования центральный процессор контроллера.

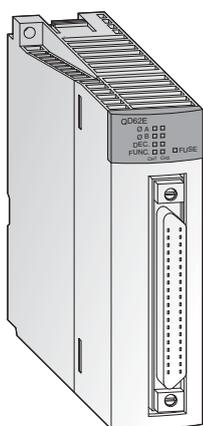


#### Особенности:

- 4 канала для определения температуры и 4 ПИД-регулирующих контура на каждый модуль
- Измерение температуры с помощью термометров сопротивления Pt100 (Q64TCRT и Q64TCRTBW) или термопар (Q64TCCT и Q64TCCTBW)
- Встроенное распознавание обрыва нагревательной проводки у модулей Q64TCRTBW и Q64TCCTBW
- Оптимизация регулирования путем автонастройки
- Транзисторный выход для управления сервоэлементом

### 2.9.4 Модули высокоскоростных счетчиков

Модули счетчиков QD62E, QD62 и QD62D регистрируют импульсы, частота которых для обычных входных модулей слишком высока.



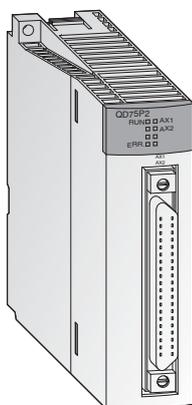
#### Особенности:

- Макс. частота счета до 500 кГц
- Вход для инкрементального датчика вращения с автоматическим определением прямого и обратного направления
- Задание состояния счетчика и выбор функции с помощью цифровых входов
- 32-битный диапазон счета с арифметическим знаком (от -2 147 483 648 до +2 147 483 647)
- Возможно применение в качестве счетчика возрастающего или убывающего счета либо в качестве кольцевого счетчика
- Все модули имеют два входа счетчика
- На каждый канал счета имеется 2 цифровых выхода, коммутируемых в зависимости от состояния счета

Все модули подключаются с помощью 40-полюсного разъема.

## 2.9.5 Модули позиционирования

В сочетании с шаговыми двигателями или сервоусилителями можно использовать модули позиционирования QD75P1, QD75P2 и QD75P4 для регулирования положения или скорости.

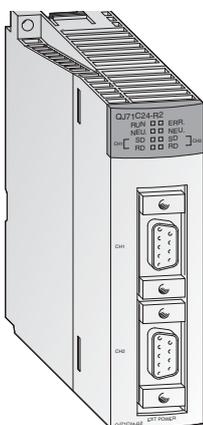


### Особенности:

- Число управляемых осей - до четырех с линейной интерполяцией (QD75P4) или до двух с круговой интерполяцией (QD75P2 и QD75P4)
- Сохранение до 600 позиций во флэш-ROM
- Единицами позиционирования могут быть импульсы, мкм, дюймы или угловые градусы
- Параметрирование и задание позиций происходит с помощью программы контроллера или с помощью среды программирования GX Configurator QP.

## 2.9.6 Интерфейсные модули для последовательной коммуникации

Модули QJ71C24 и QJ71C24-R2 служат для коммуникации с периферийными устройствами. При этом используются стандартизованные последовательные интерфейсы.

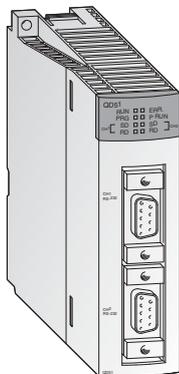


### Особенности:

- Два интерфейса RS232C (у QJ71C24-R2) или один интерфейс RS422/485 и один интерфейс RS232C (у QJ71C24)
- Скорости передачи до 115200 бод
- Возможность доступа к данным контроллера с вышестоящего персонального компьютера, на котором установлено визуализирующее или мониторинговое программное обеспечение
- Возможно подключение принтера
- Встроенная память для хранения данных обеспечения качества, производственных данных или сигнализации, передаваемых по мере необходимости
- Возможность свободного определения протокола для обмена данными
- Возможно программирование контроллера через интерфейсные модули.

## 2.9.7 Интерфейсные модули, программируемые на языке BASIC

Модули QD51S-R24 и QD51 независимо от центрального процессора контроллера обрабатывают собственную программу, написанную на языке AD51H-Basic. Благодаря этому они могут обмениваться данными с периферийными устройствами, не загружая этими задачами центральный процессор контроллера.

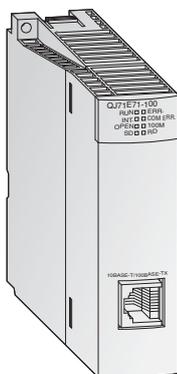


### Особенности:

- Два интерфейса RS232C (у QD51) или один RS422/485 и один RS232C (у QD51S-R24)
- Скорости передачи до 38400 бод
- Возможность обращения к операндам центрального процессора контроллера и буферной памяти специальных модулей.
- С помощью интерфейсных модулей можно дистанционно изменять режим центрального процессора контроллера (переключать его между состояниями RUN и STOP)

## 2.9.8 Модули Ethernet

С помощью модулей QJ71E71 и QD71E71-B2 контроллер MELSEC System Q можно через сеть Ethernet соединить с другой аппаратурой, например, персональным компьютером. Помимо обмена данными по протоколу TCP/IP или UDP/IP, через Ethernet можно считывать и изменять данные контроллера, а также контролировать работу и состояние центрального процессора.

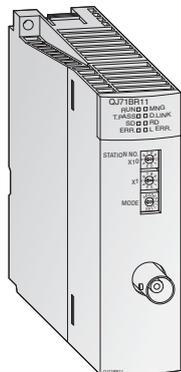


### Особенности:

- Интерфейс 10BASE5, 10BASE2 или 10BASE-T
- Скорость передачи 10 или 100 Мбит/с
- Возможна функция FTP-сервера
- Обмен данными через фиксированный буфер приема-передачи
- Могут быть одновременно установлены до 16 логических соединений
- С компьютера, на котором установлено программное обеспечение GX Developer или GX IEC Developer, можно через Ethernet изменять программу контроллера.

## 2.9.9 Модули MELSECNET

Модули QJ71BR11 и QJ71LP21 позволяют подключить контроллеры MELSEC System Q к сети MELSECNET/10 или MELSECNET/H и, тем самым, обеспечить коммуникацию с контроллерами серий Q, QnA и QnAS.

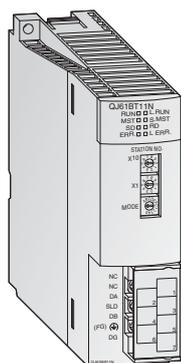


### Особенности:

- Можно использовать две различные топологии сети: коаксиальную шину (QJ71BR11) или оптическое двойное кольцо (QJ71LP21)
- Высокая скорость передачи: 10 Мбит/с в случае коаксиальной шины и, по выбору, 10 или 20 Мбит/с в случае оптического двойного кольца
- Возможен обмен данными с контроллером, компьютером и удаленными станциями ввода-вывода
- Возможен обмен данными с любыми станциями - независимо от количества сетей, имеющих между станциями
- Игнорирование дефектной станции в случае коаксиальной шины и функция петлевого контроля в случае оптического двойного кольца, если в станции имеется неполадка
- При выходе из строя контрольной станции ее задачи автоматически перенимает другая станция

## 2.9.10 Главный/локальный модуль для CC-Link

Модуль QJ61BT11 можно использовать в качестве главной или локальной станции в системе CC-Link. Он служит для управления удаленными (децентрализованными) входами и выходами и их контроля.



### Особенности:

- Параметрирование всех имеющихся в сети модулей происходит непосредственно из главного модуля (Master).
- Автоматическая коммуникация между удаленной аппаратурой и главным модулем. Время опроса для 2048 входов-выходов составляет всего лишь 3.3 мс.
- Скорости передачи до 10 Мбит/с
- Возможность с помощью одного главного модуля расширить систему до 2048 децентрализованных входов-выходов
- Возможность создания избыточной системы с дополнительным резервным главным устройством. После выхода из строя одной из главных станций коммуникация продолжается.
- Автоматический запуск CC-Link без параметрирования
- Возможность запуска программ прерывания в зависимости от условий в сети.

### 2.9.11 Модуль PROFIBUS/DP

Главный модуль Profibus/DP QJ71PB92D и подчиненный модуль Profibus/DP QJ71PB93D позволяют обмениваться данными между контроллерами MELSEC System Q и другими устройствами, подключенными к сети Profibus/DP.

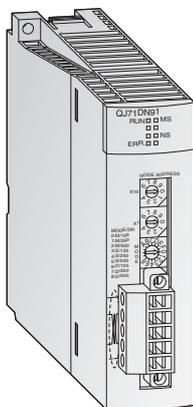


#### Особенности:

- Главная станция может обмениваться данными с 60 подчиненными станциями.
- Возможность обработки 244 входных и 244 выходных байтов на каждое подчиненное устройство.
- Поддержка глобальных служб (например, SYNC и FREEZE), а также функций диагностики для определенных подчиненных устройств.
- Обмен данными может происходить автоматически и, дополнительно, с помощью блочных команд.

### 2.9.12 Главный модуль DeviceNet QJ71DN91

Модуль QJ71DN91 соединяет контроллер MELSEC System Q с сетью DeviceNet. DeviceNet представляет собой недорогое решение для сетевой коммуникации с оконечными устройствами "нижнего уровня".

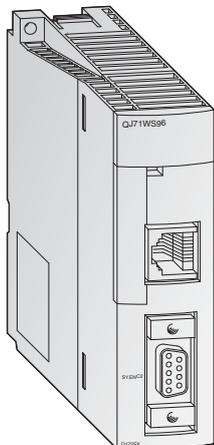


#### Особенности:

- Пользователь может свободно выбирать место расположения главной станции и подчиненных станций
- Скорости передачи 125, 250 или 500 кбит/с
- Длина проводки может достигать 500 м
- Методы коммуникации:
  - опрос
  - стробирование битов
  - изменение состояния
  - циклическая

### 2.9.13 Веб-серверный модуль

Веб-серверный модуль QJ71WS96 дает возможность дистанционного контроля контроллера MELSEC System Q.



#### Особенности:

- Доступ к контроллеру через интернет
- Простейшее параметрирование
- Для настройки и дистанционного контроля пользователю нужен только браузер
- Интерфейс RS232 для подключения модема
- Для коммуникации можно использовать различные сетевые подключения: ADSL, модем, LAN и т. п.
- Передача и прием данных по электронной почте или по FTP
- Самостоятельно оформленные сайты и возможность встраивания Java-апплетов
- Стандартная связь через Ethernet для обмена данными с другими контроллерами или персональными компьютерами
- Определение и сохранение событий и состояний центрального процессора

## 2.10 Основы контроллеров

### 2.10.1 Среда программирования

Чтобы контроллер можно было программировать с обычного компьютера, на компьютере должна быть установлена особая среда программирования. Она должна отвечать следующим требованиям:

- Для программирования должны использоваться хорошо понятные и узнаваемые символы или сокращения, как это имеет место при программировании на языке релейно-контактных схем (LD) или списка инструкций (IL).
- Возможность проверки введенных команд (синтаксиса) и функциональности программы перед передачей программы в контроллер.
- Возможность хранения программ контроллера на жестком диске компьютера или ином носителе данных.
- Возможность загрузки уже имеющихся программ с жесткого диска компьютера или иного носителя данных.
- Возможность снабжения программы подробными комментариями.
- Возможность распечатки программы.
- Возможность передачи программы в контроллер через последовательный интерфейс, а также передачи программы из контроллера в компьютер.
- Возможность наблюдения за выполнением программы и состояниями операндов в реальном масштабе времени.
- Возможность изменения программы во время выполнения программы контроллером.
- Возможность изменения настроек и параметров для работы контроллера.
- Возможность сохранения состояний операндов контроллера и их повторной загрузки.
- Возможность имитации работы программы контроллера без подключения самого контроллера.

В этом списке перечислены лишь некоторые из требований, предъявляемых к среде программирования.

### 2.10.2 Обработка программы в контроллере

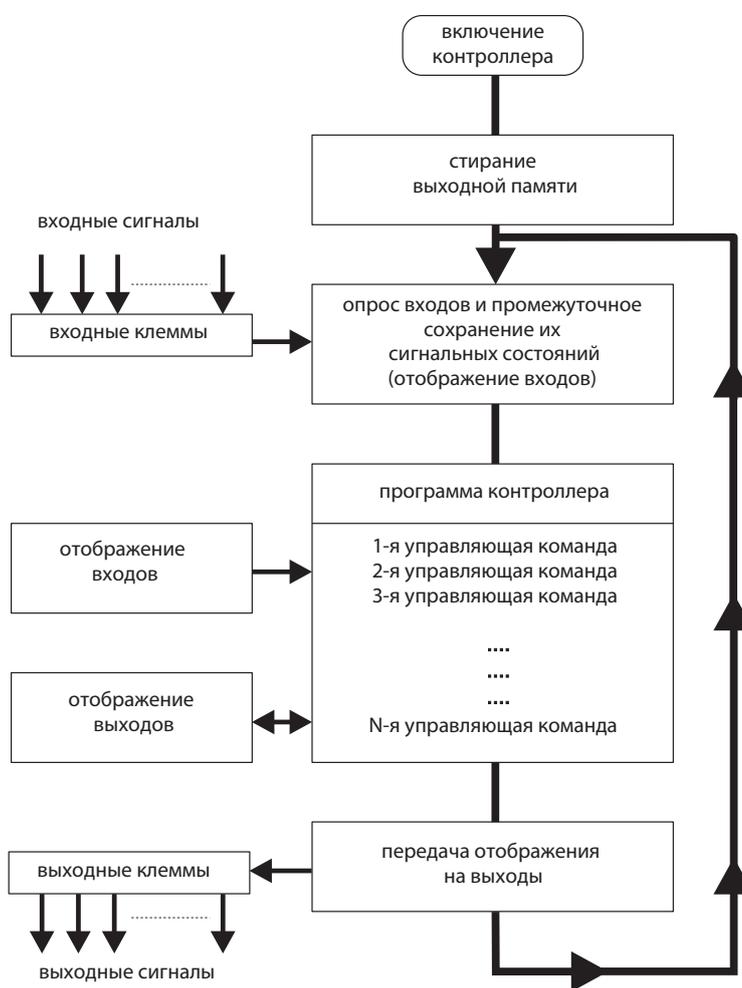
Контроллер работает по заданной программе, которая, как правило, создается вне контроллера, а затем передается в контроллер и сохраняется в памяти для хранения программ. При программировании важно знать, как контроллер обрабатывает программу.

Программа состоит из очереди отдельных команд, определяющих функции контроллера. Контроллер поочередно обрабатывает управляющие команды в запрограммированной последовательности.

Прогон программы постоянно повторяется, т. е. происходит циклически. Время, необходимое для одного прогона программы, обозначается как "время цикла программы".

#### Метод отображения

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



#### Отображение входов

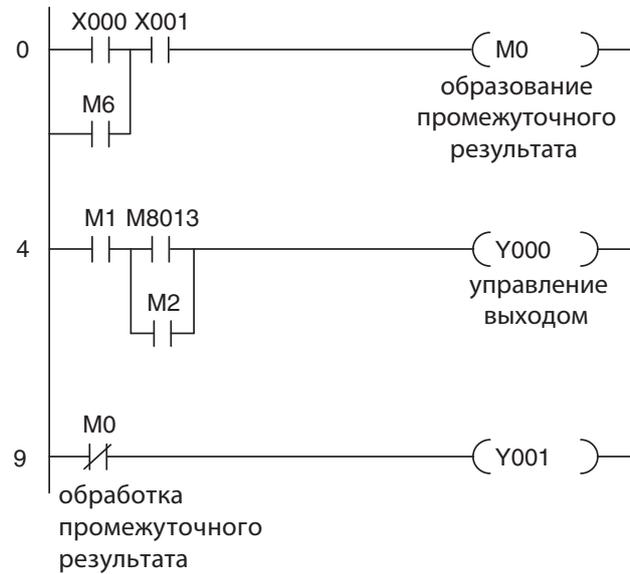
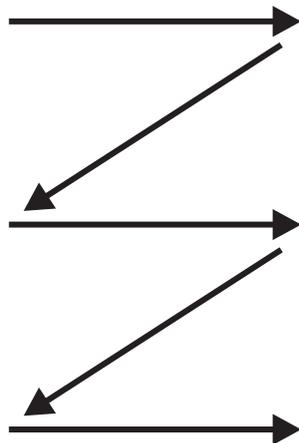
В начале программного цикла сигнальные состояния входов опрашиваются и сохраняются в промежуточной памяти: создается так называемое "отображение входов".

### Прогон программы

Во время последующего прогона программы контроллер обращается к отображенным состояниям входов. Поэтому изменения сигналов на входах распознаются лишь в следующем программном цикле.

Программа обрабатывается сверху вниз, в последовательности ввода команд. Промежуточные результаты могут использоваться в том же программном цикле.

обработка программы



### Отображение выходов

Результаты логических операций, относящиеся к выходам, записываются в выходную буферную память (отображение выходов). Лишь в конце прогона программы промежуточные результаты передаются на выходы. Отображенные выходы в выходной буферной памяти сохраняются до их очередной перезаписи. После присвоения значений выходам программный цикл начинается заново.

### Обработка сигналов в контроллере по сравнению со схемной системой управления (т. е. системой управления на основе физического монтажа)

В схемной системе управления программа определяется типом функциональных звеньев и их связями (электропроводкой). Все процессы управления выполняются одновременно (параллельно). Любое изменение состояний входных сигналов сразу вызывает изменение состояния выходных сигналов.

В программируемом контроллере изменение состояний входных сигналов, произошедшее во время прогона программы, может быть учтено лишь в следующем программном цикле. Этот недостаток компенсируется малым временем программного цикла. Время цикла программы зависит от количества и типа управляющих команд.

### 2.10.3 Операнды программируемого контроллера

В управляющих командах контроллера используются операнды. При этом контроллер опрашивает их сигнальные состояния или значения, а также влияет на эти состояния и значения посредством программы. Операнд состоит из

- буквенного обозначения операнда и
- адреса операнда.

пример указания операнда (например, вход 0):



Примеры обозначения операндов:

Буквенное обозначение операнда	Тип	Значение
X	Вход	Входная клемма контроллера (например, выключатель)
Y	Выход	Выходная клемма контроллера (например, контактор или лампа)
M	Маркер	Промежуточная память в контроллере, способная принимать одно из двух состояний ("Вкл." или "Выкл.")
T	Таймер	"Реле времени" для реализации функций, зависящих от времени
C	Счетчик	Счетчик
D	Регистр данных	для хранения данных в контроллере (например, результатов измерений или вычислений)

## 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 IEC1131-3 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

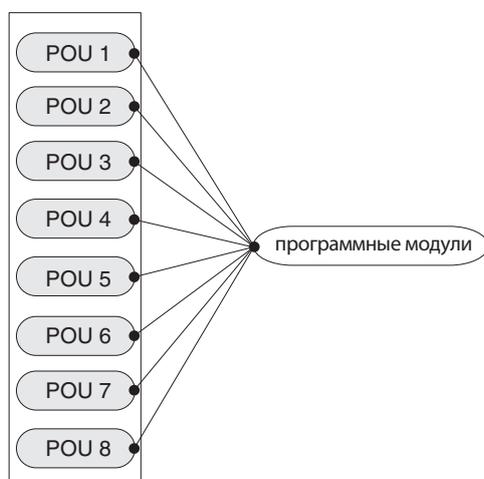
### 3.2.1 Определение терминов в IEC61131-3

#### 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.



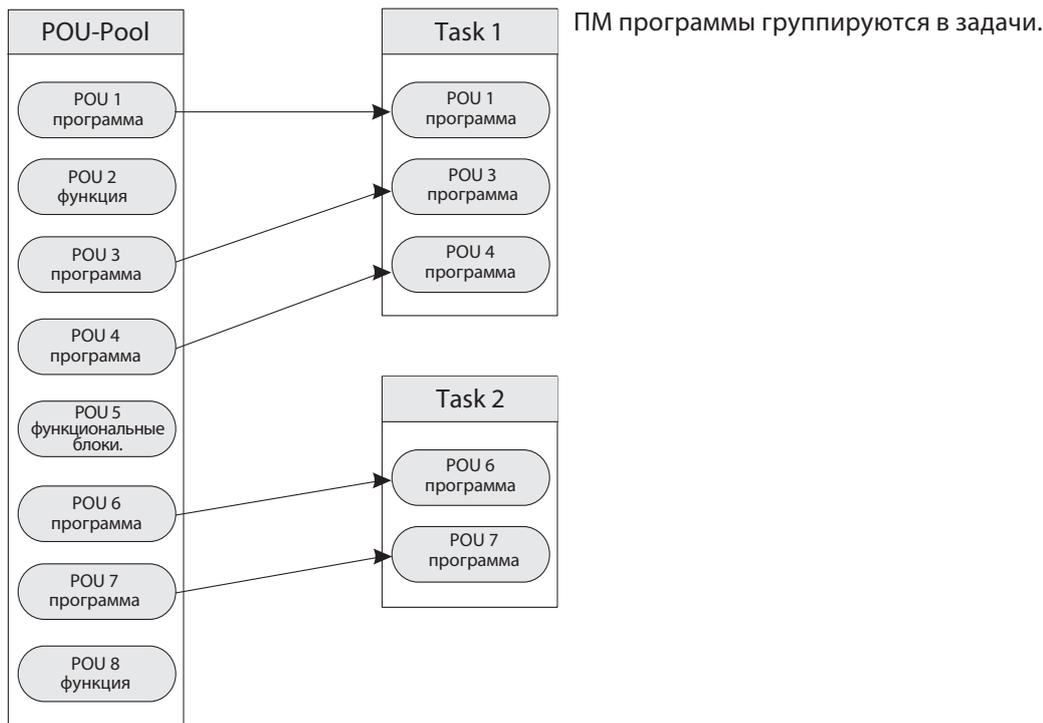
Программные модули (ПМ) используются для решения всех задач программирования.

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

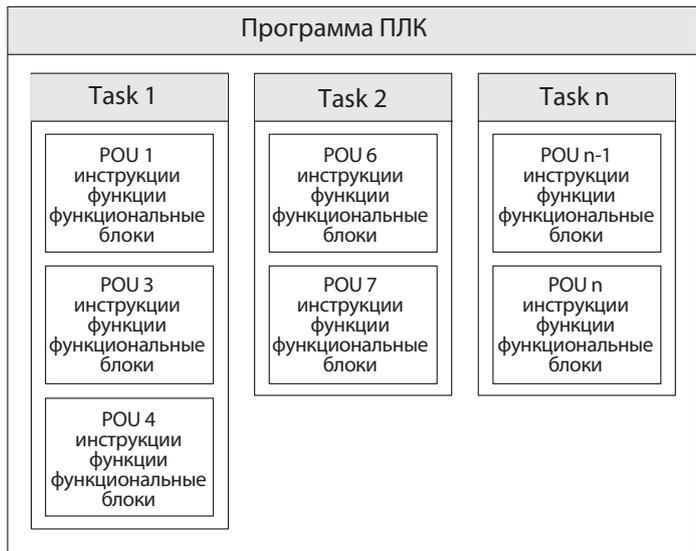
- Программы
- Функции
- Функциональные блоки

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.

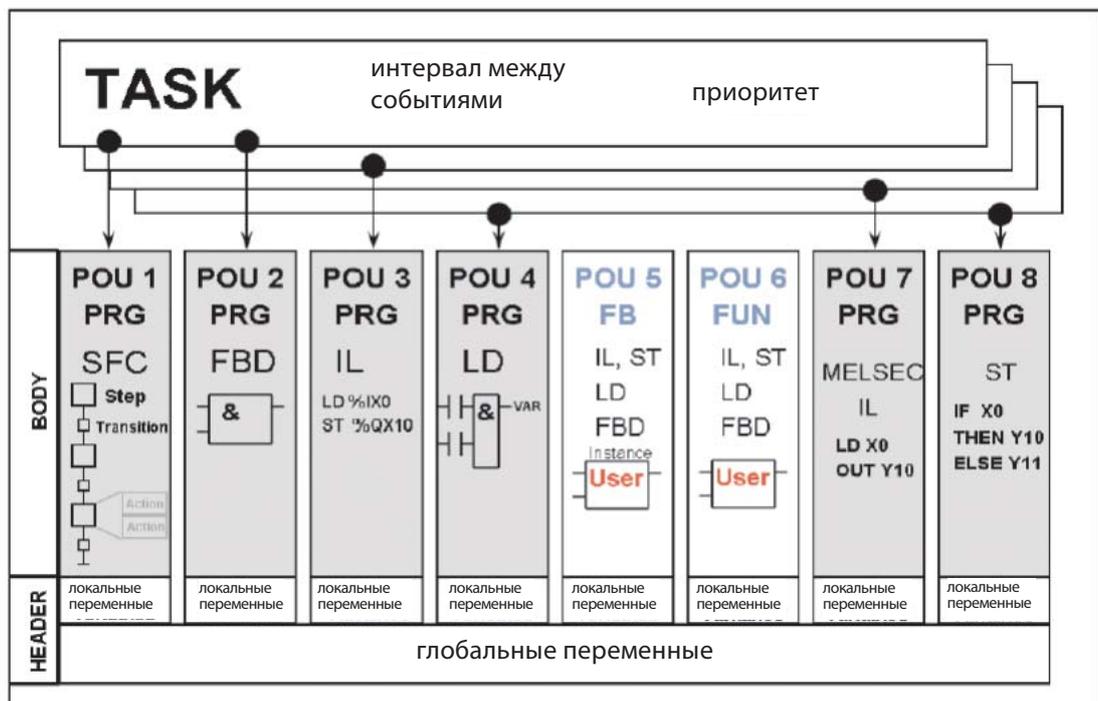


Большинство программ ПЛК состоит из областей кода, которые выполняют специфические задачи. Они могут формировать часть одной большой программы или быть написанными в виде подпрограмм с инструкциями управления программой, позволяющими выбирать текущую подпрограмму, например, CALL, CJ и т.д.



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.

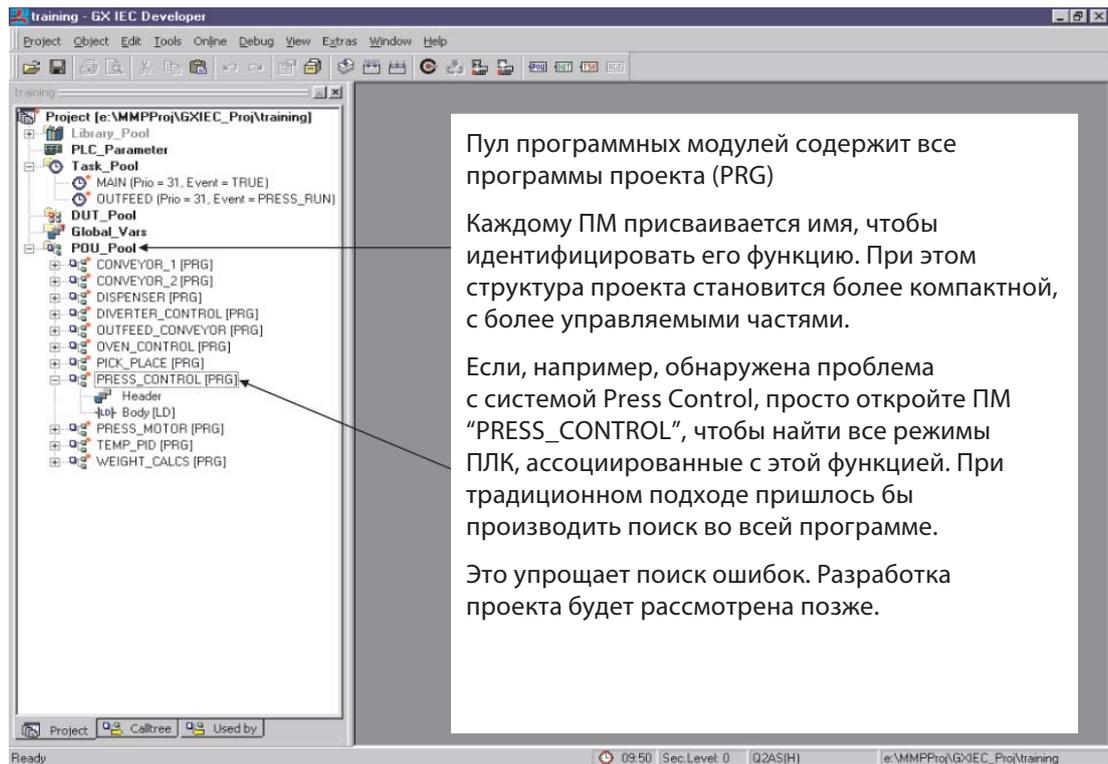


### Пул программных модулей

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.



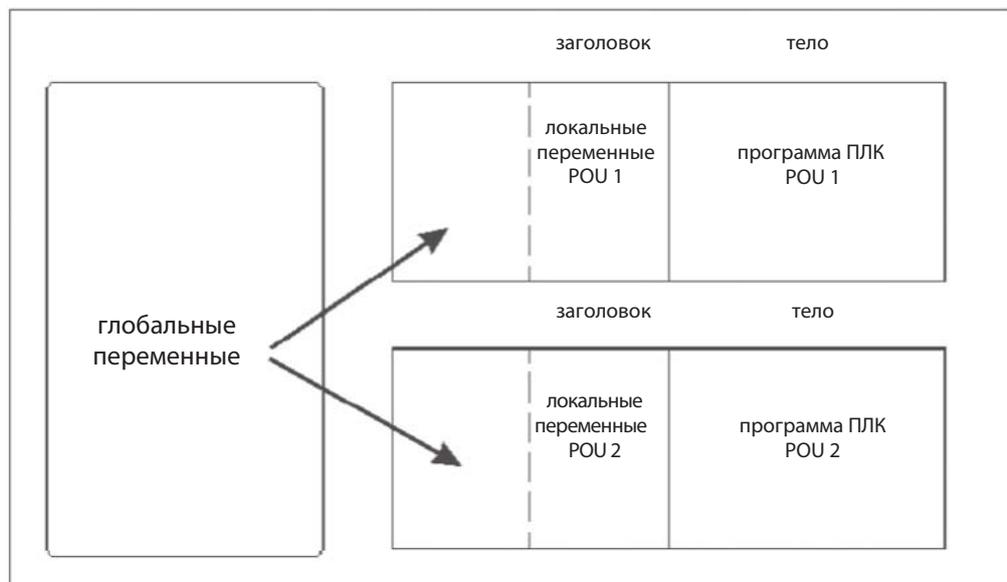
**Состав ПМ**

Каждый программный модуль (ПМ) состоит из:

- Заголовка
- 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.

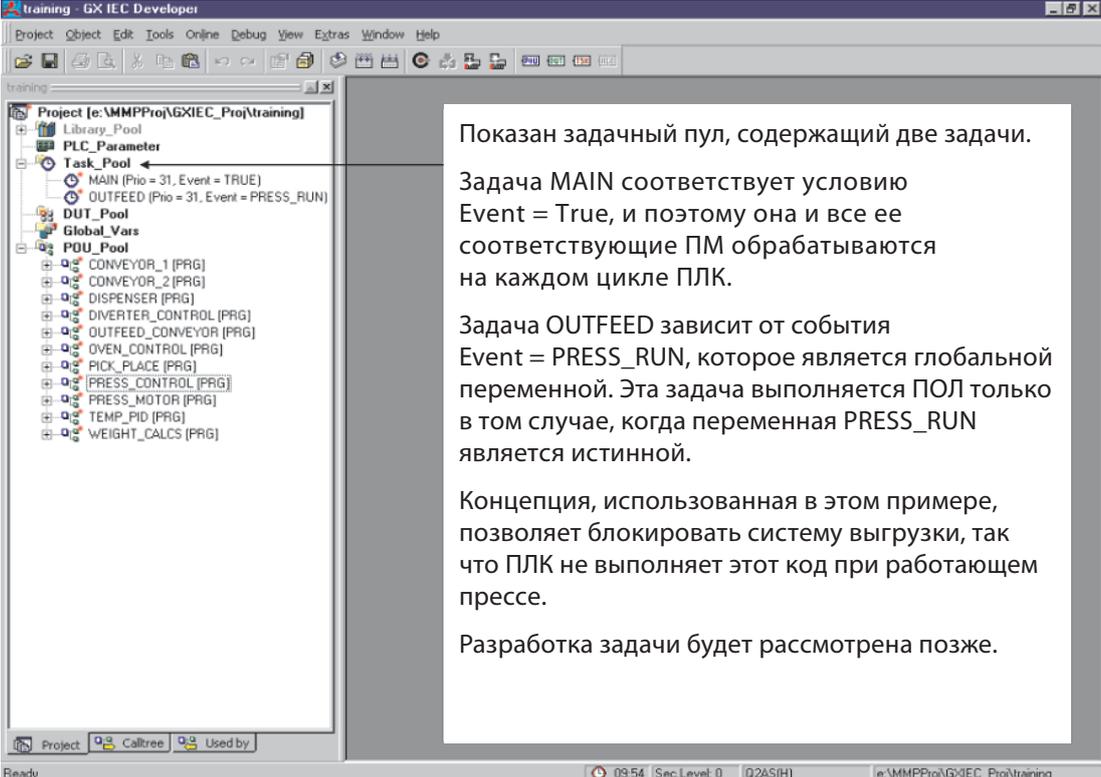
Когда GX IEC Developer компилирует проект, он автоматически вставляет в программу команды ветвления программы в соответствии с управляемыми событиями задачами.

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.



Project [e:\MMPProj\GXIEC\_Proj\training]

- Library\_Pool
- PLC\_Parameter
- Task\_Pool
  - MAIN (Prio = 31, Event = TRUE)
  - OUTFEED (Prio = 31, Event = PRESS\_RUN)
- DUT\_Pool
- Global\_Vars
- POU\_Pool
  - CONVEYOR\_1 (PRG)
  - CONVEYOR\_2 (PRG)
  - DISPENSER (PRG)
  - DIVERTER\_CONTROL (PRG)
  - OUTFEED\_CONVEYOR (PRG)
  - OVEN\_CONTROL (PRG)
  - PICK\_PLACE (PRG)
  - PRESS\_CONTROL (PRG)
  - PRESS\_MOTOR (PRG)
  - TEMP\_PID (PRG)
  - WEIGHT\_CALC (PRG)

Project Calltree Used by

Ready 09:54 Sec Level: 0 Q2AS(H) e:\MMPProj\GXIEC\_Proj\training

Показан задачный пул, содержащий две задачи.

Задача MAIN соответствует условию Event = True, и поэтому она и все ее соответствующие ПМ обрабатываются на каждом цикле ПЛК.

Задача OUTFEED зависит от события Event = PRESS\_RUN, которое является глобальной переменной. Эта задача выполняется ПОЛ только в том случае, когда переменная PRESS\_RUN является истинной.

Концепция, использованная в этом примере, позволяет блокировать систему выгрузки, так что ПЛК не выполняет этот код при работающем прессе.

Разработка задачи будет рассмотрена позже.

**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.

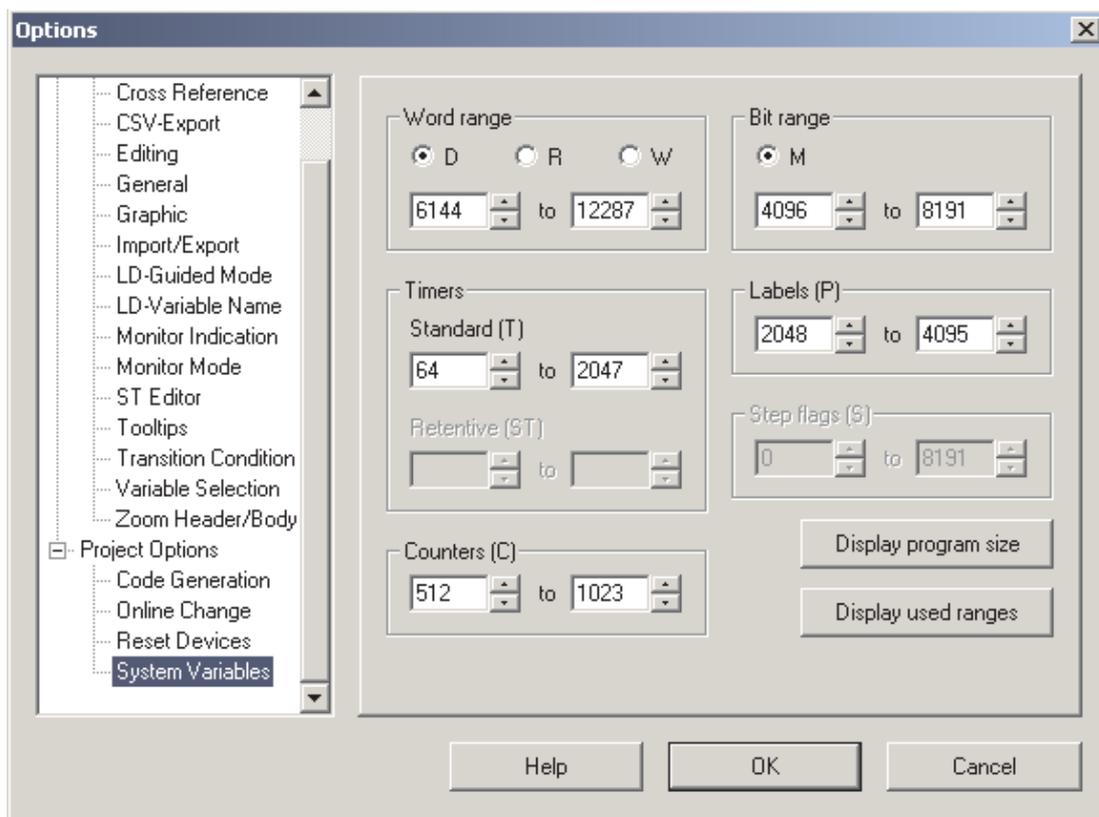
При этом программный инженер должен думать только о содержании программы, а не о том, правильны ли команды перехода и согласуются ли они с правилами.

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. (У иных контроллеров кроме MELSEC System Q возможны не все описываемые здесь настройки)

- **Word range** - Область регистров типа слово
  - D: D-операнды используются как системные переменные типа слово.
  - R: R-операнды используются как системные переменные типа слово.
  - W: W-операнды используются как системные переменные типа слово.
  - 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

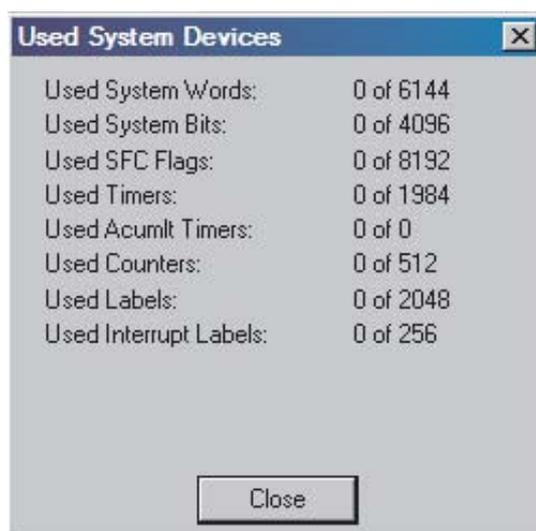
Диапазон: В зависимости от типа ПЛК, как определено в соответствующем TYP-файле.

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

Сводка используемого размера программы, показанная в отдельном диалоговом окне. Если программа не скомпилирована, 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.

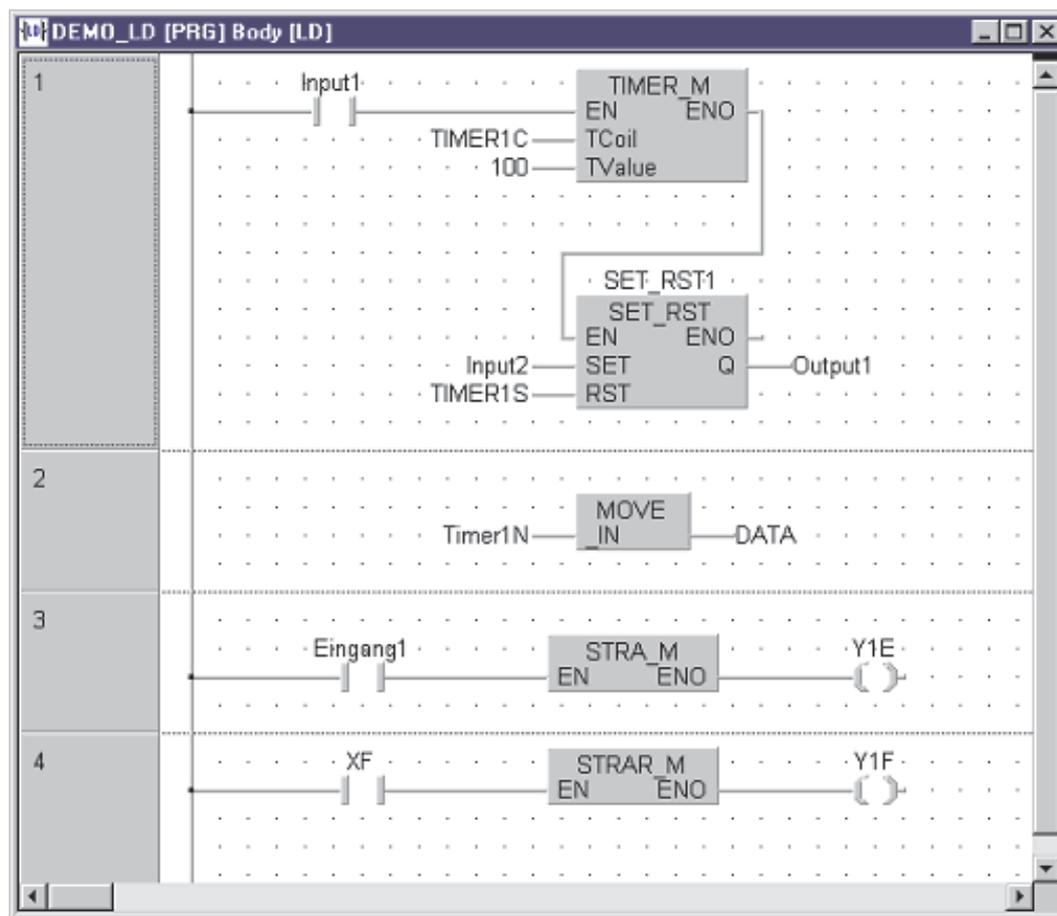
### 3.3.2 Графические редакторы

#### 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:

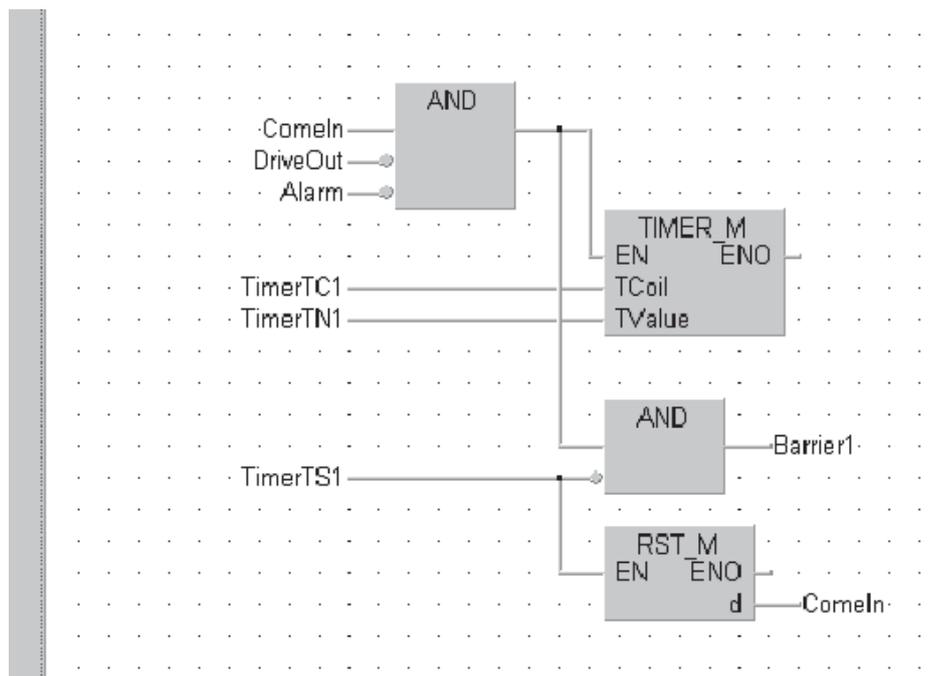


#### Функциональная блок-схема

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

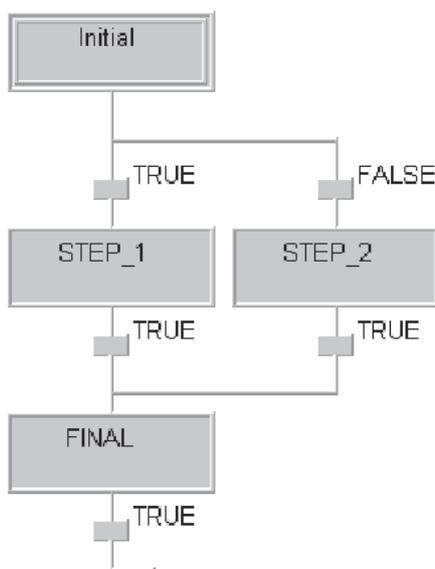


**Последовательная функциональная схема**

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
<b>BOOL</b>	Булев	Битовый операнд	0 (Ложный), 1 (Истинный)	1 бит	X, Y, M, B
<b>INT</b>	Целочисленный	Регистр	-32768 ... +32767	16 бит	D, W, R
<b>DINT</b>	Двойной целочисленный		-2 147 483 648 ... 2 147 483 647	32 бита	
<b>WORD</b>	Строка битов	K4M0	0 - 65 535	16 бит	X, Y, M, B
<b>DWORD</b>		K8M0	0 - 4 294 967 295	32 бита	
<b>REAL</b>	Значение с плавающей запятой	7 разрядов		32 бита	Все контроллеры MELSEC System Q *
<b>STRING</b>	Строка символов	20 символов (по умолчанию)		32 бита	
<b>TIME</b>	Значение времени	-T#24d0h31m23s64800ms ... T#24d20h31m23s64700 ms		32 бита	

\* Ранние версии Q00JCP не поддерживают эти типы данных.

#### 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

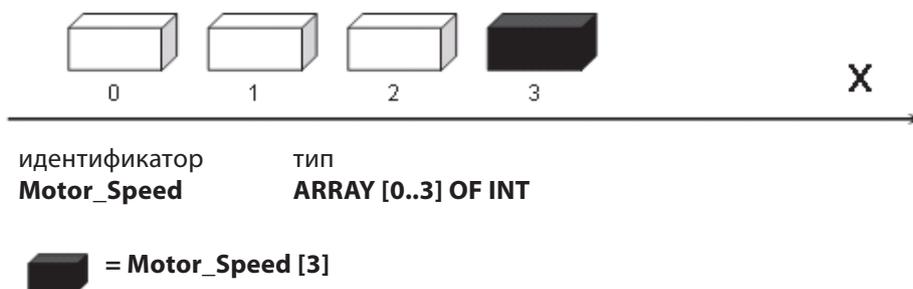
В программе элементы программы могут использовать: Motor\_Volts[1] и Motor\_Volts[2], как объявления. В данном примере это означает, что адресуются D1 и D2.

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

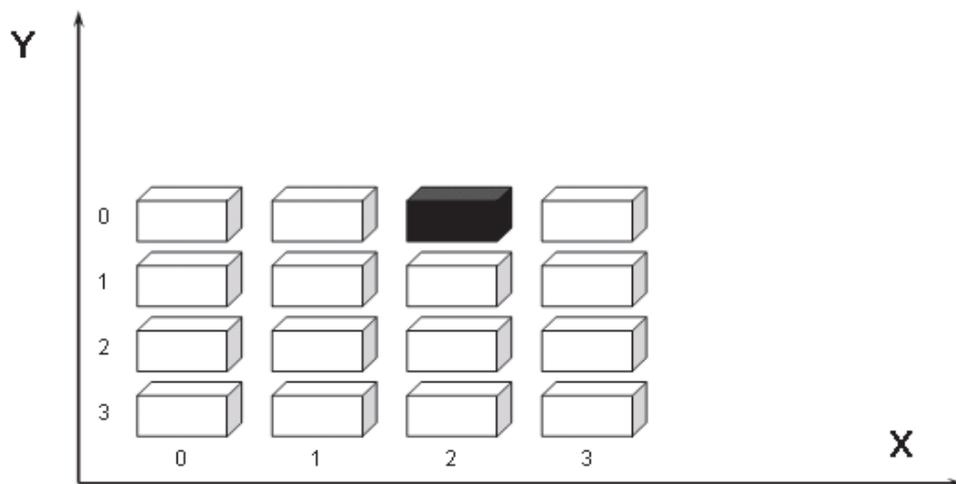
Массивы могут обеспечить удобный способ "индексации" имен тегов, т.е. одно объявление в таблице локальных или глобальных переменных может иметь доступ ко многим элементам.

Следующие диаграммы иллюстрируют графическое представление трех типов массивов.

##### Одномерный массив



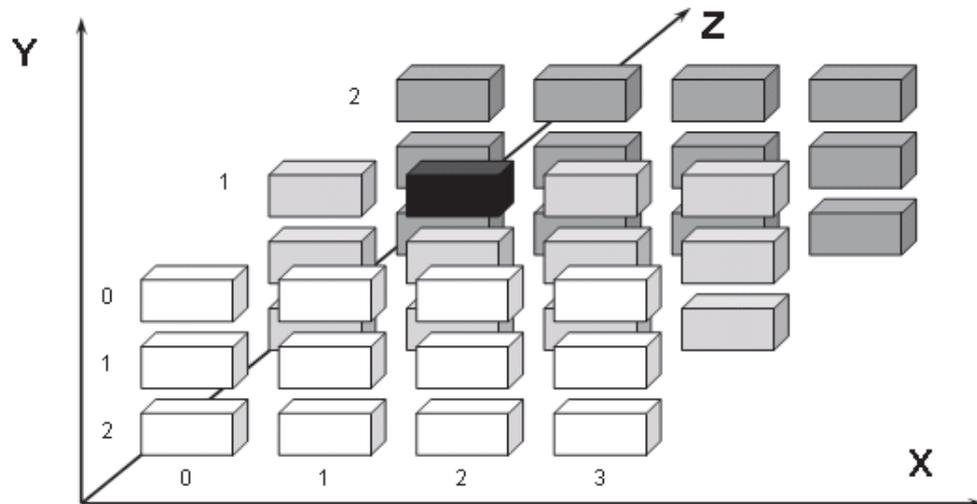
**Двумерный массив**



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]

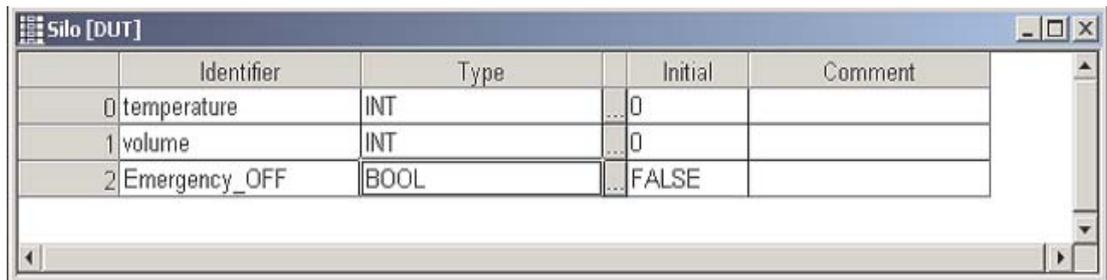
### Типы блоков данных(DUT)

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

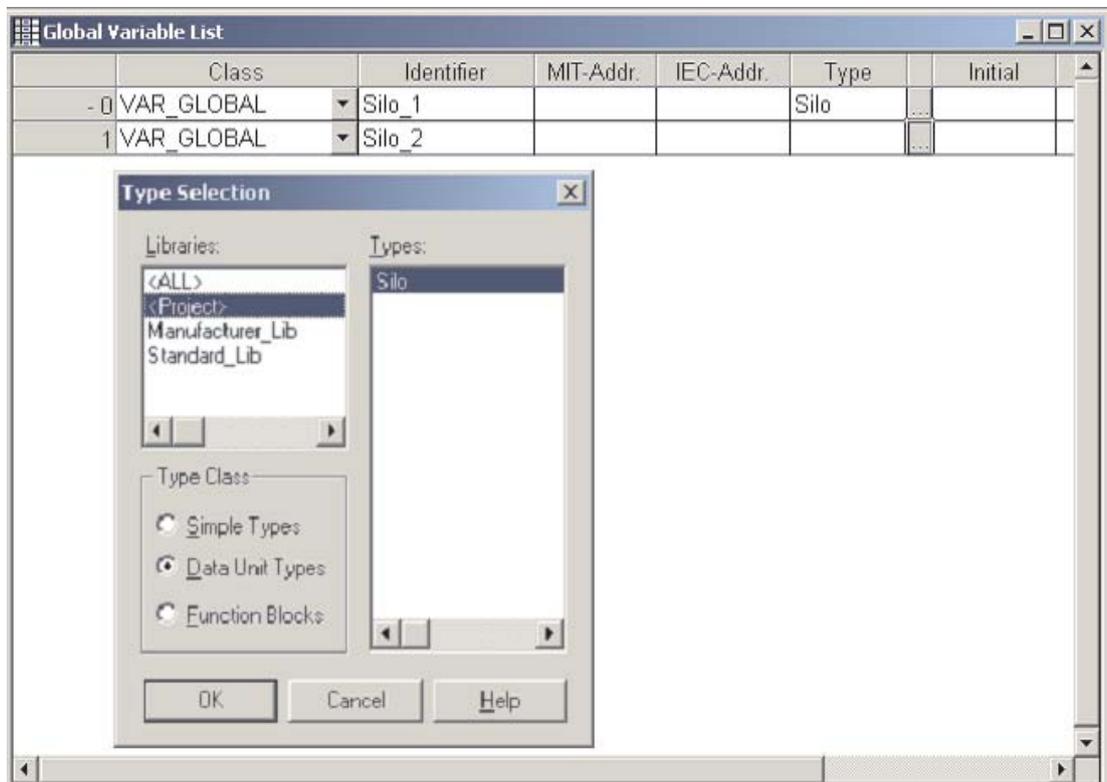
Следующий пример показывает создание типа данных, названного Silo. Набор переменных Silo содержит две переменные INT и одну переменную типа BOOL.



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

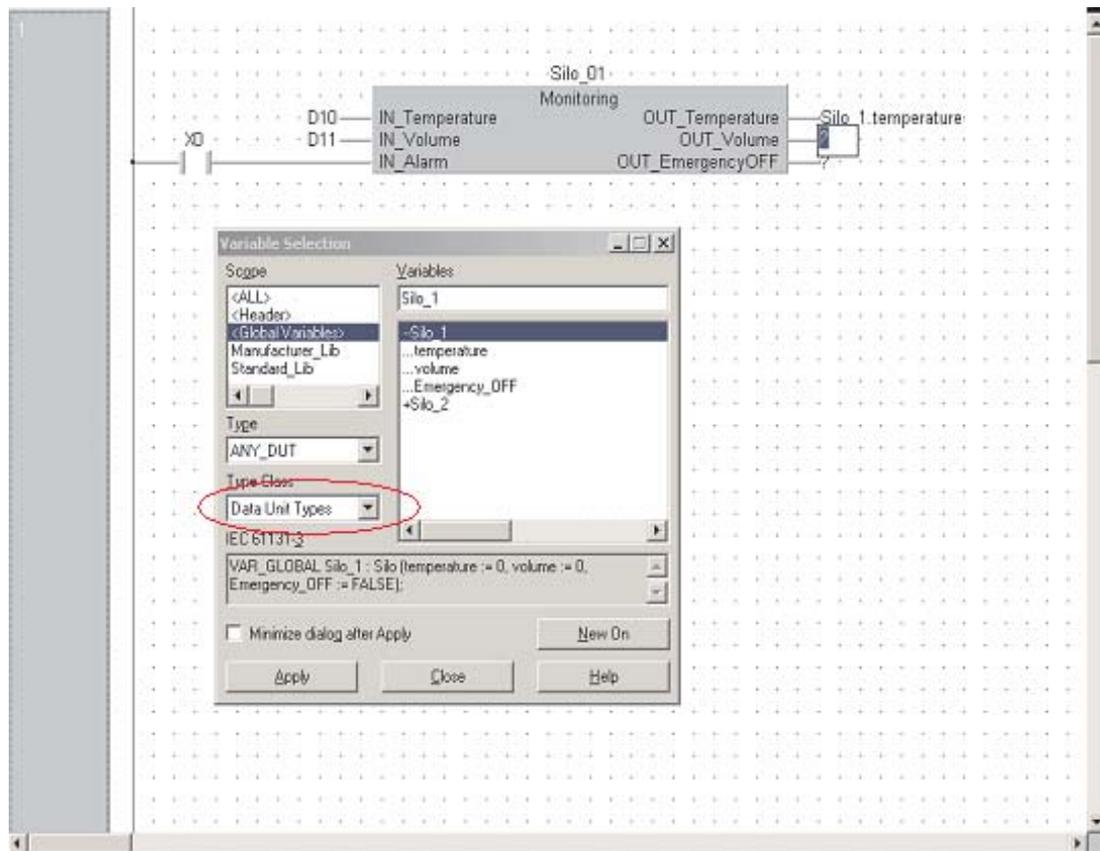
### Как объявлять DUT

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

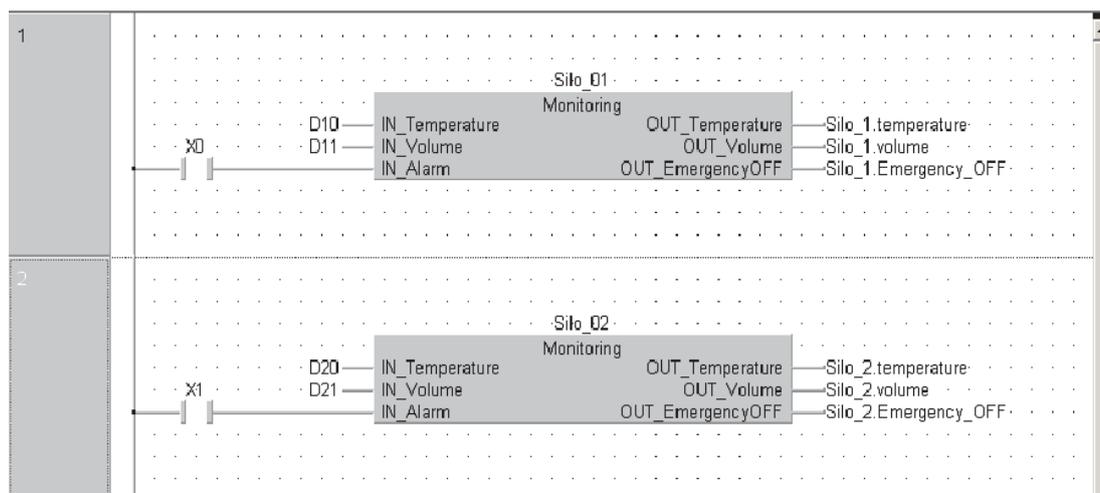


Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0 VAR_GLOBAL	Silo_1			Silo	
1 VAR_GLOBAL	Silo_2				

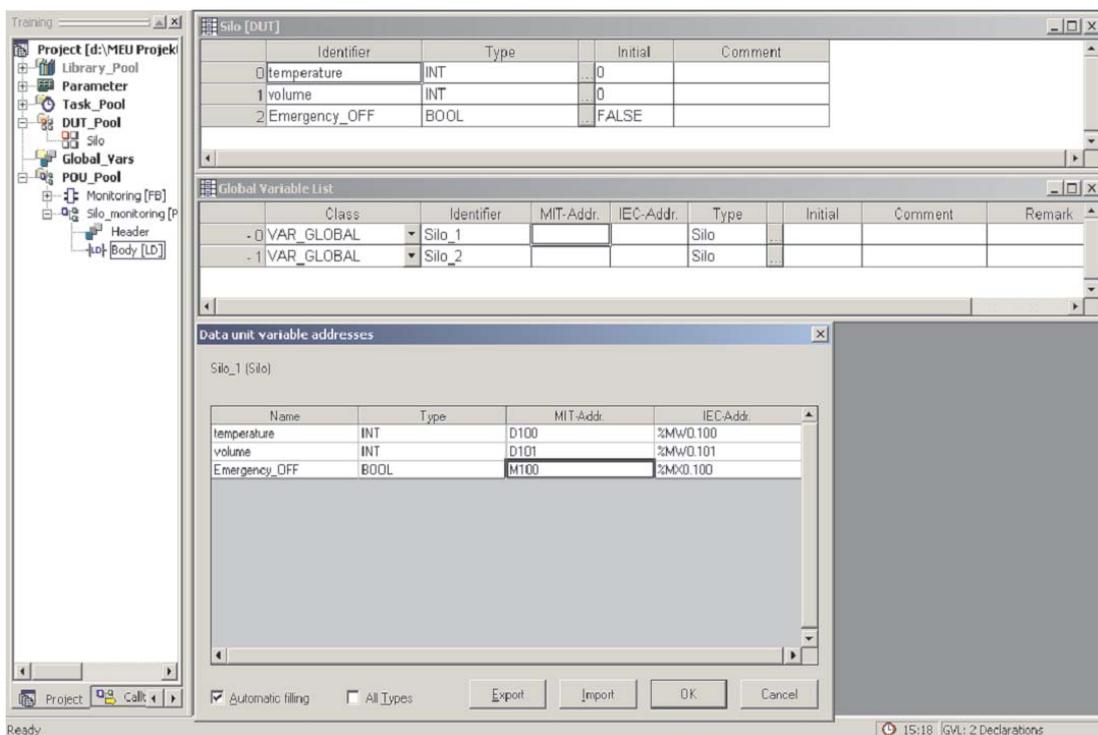
Переменные хранятся в списке глобальных переменных. Структура обеих переменных, Silo\_1 и Silo\_2, идентична, поэтому для ссылки на отдельную переменную каждого DUT вам нужен лишь префикс в их именах с именем соответствующей глобальной переменной.



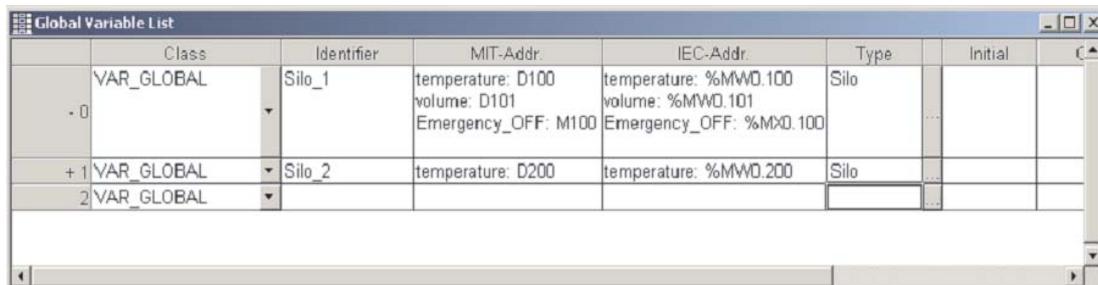
В этом примере функциональный блок типа "Monitoring" был запрограммирован для того, чтобы присвоить значение регистра и Булева входа для элементов DUT. Затем для двух бункеров были созданы два отдельных экземпляра (Silo\_01 и Silo\_02) этих функциональных блоков.



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.



Еще один пример применения структурированного типа данных имеется в разделе 11.

### 3.4.3 Таймеры и счетчики MELSEC

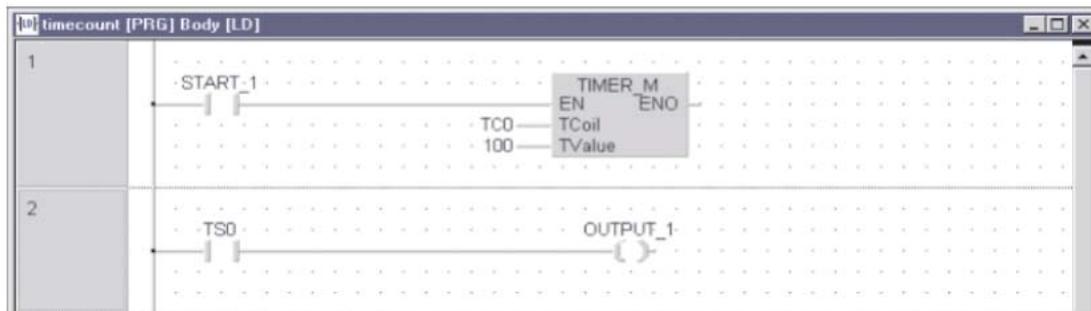
При программировании стандартных Таймеров/Счетчиков должны соблюдаться правила IEC:

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

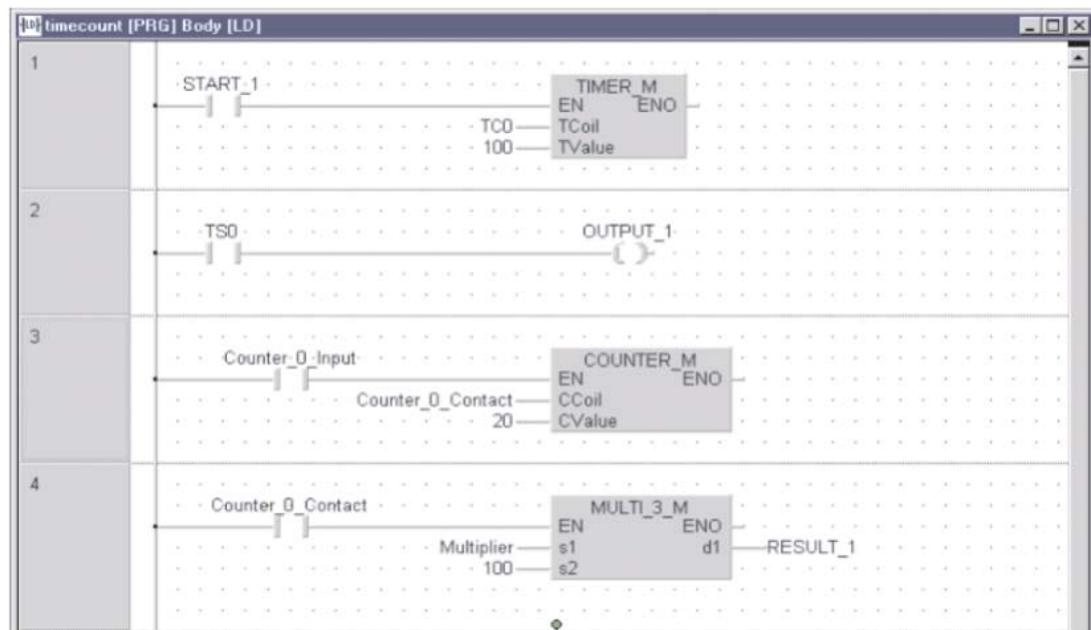
Программирование контакта таймера/счетчика: **TSn / CSn**

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

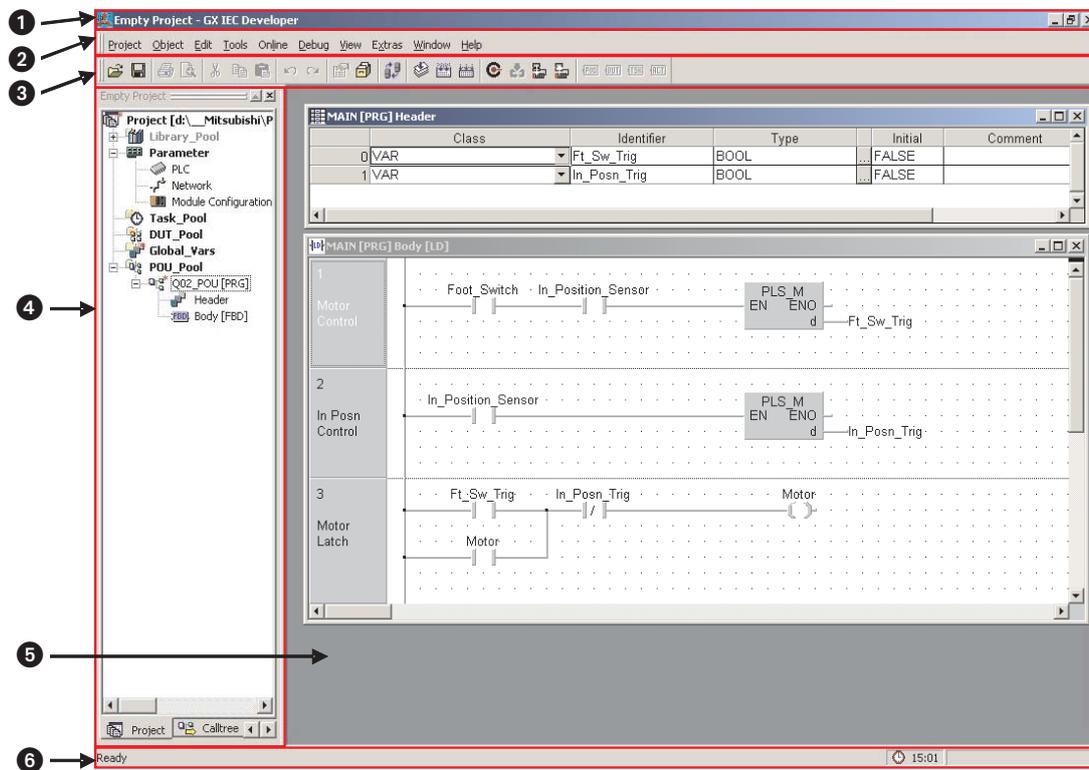
В следующем разделе мы разработаем наш первый проект, сначала используя редактор лестничных диаграмм.

### 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.

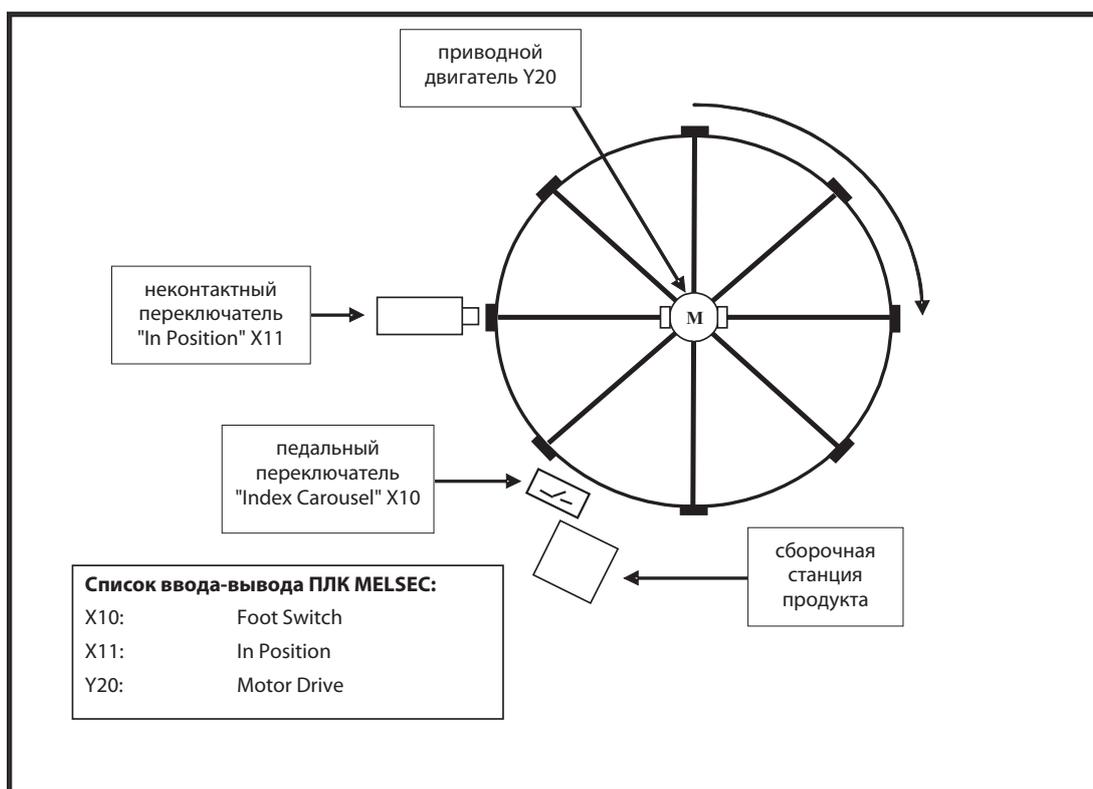
## 4.2 Прикладная программа

### 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 педальный 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:

- Педальный 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" X11, 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".

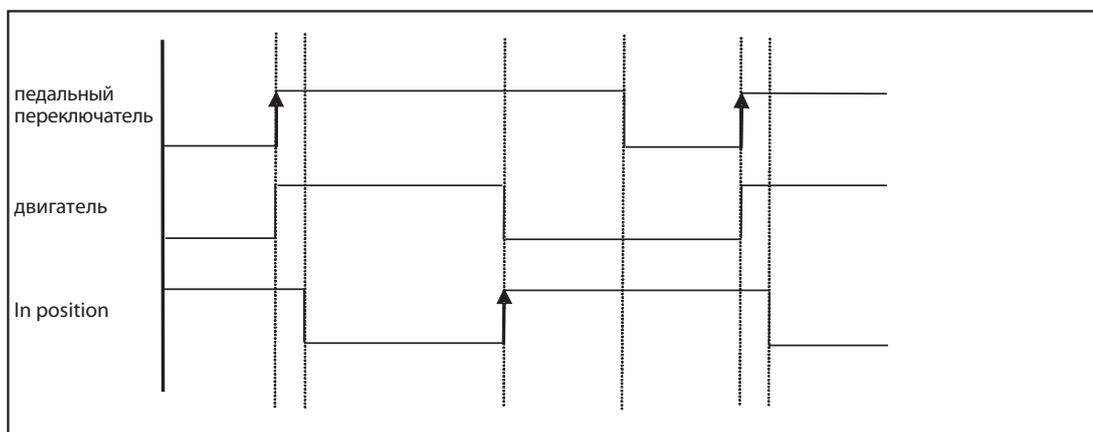
В этом приложении наиболее целесообразно использовать команду MELSEC "PLS" (Установка операнда при возрастающем фронте). Она используется здесь вместо инструкции IEC R\_TRIG (триггер, тактируемый нарастающим фронтом), которая также может применяться.

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.

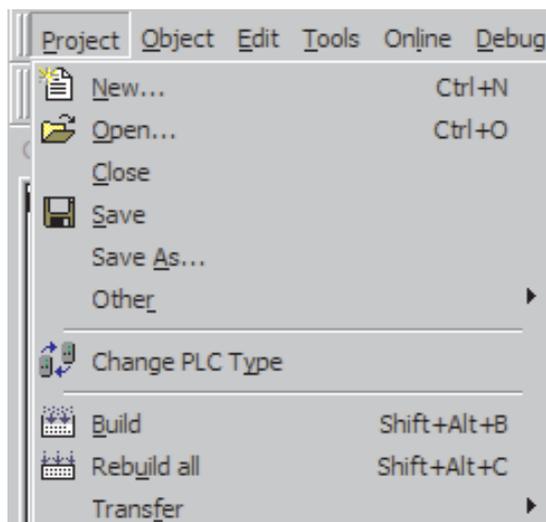
Двигатель может начать вращение, только когда педальный выключатель отпускается и нажимается повторно. Следовательно, двигатель включается снова по нарастающему фронту педального выключателя.

#### Временная диаграмма логики управления каруселью:

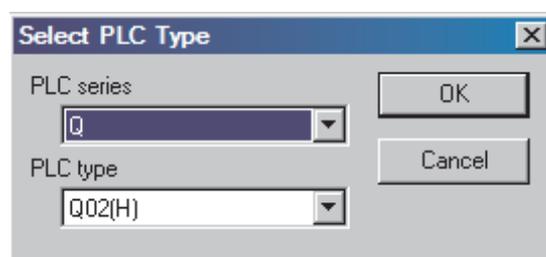


## 4.2.2 Создание нового проекта

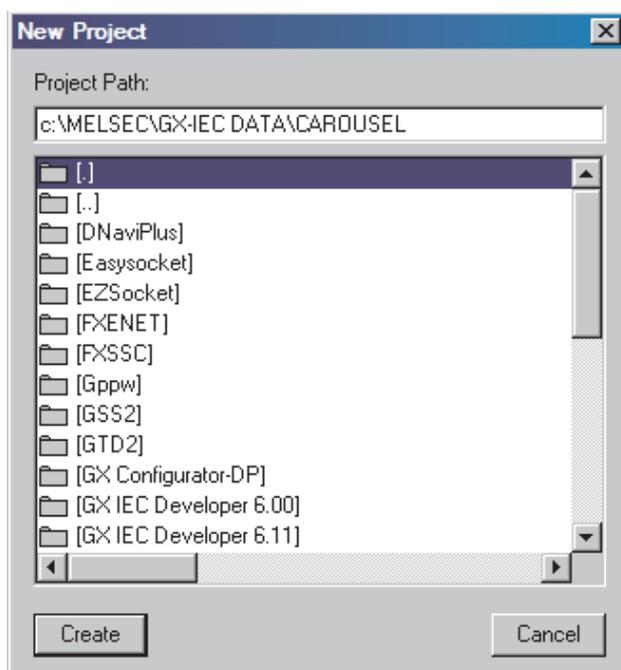
- ① В меню **Project** выберите **New**.



- ② Выберите соответствующий тип ПЛК из списка:

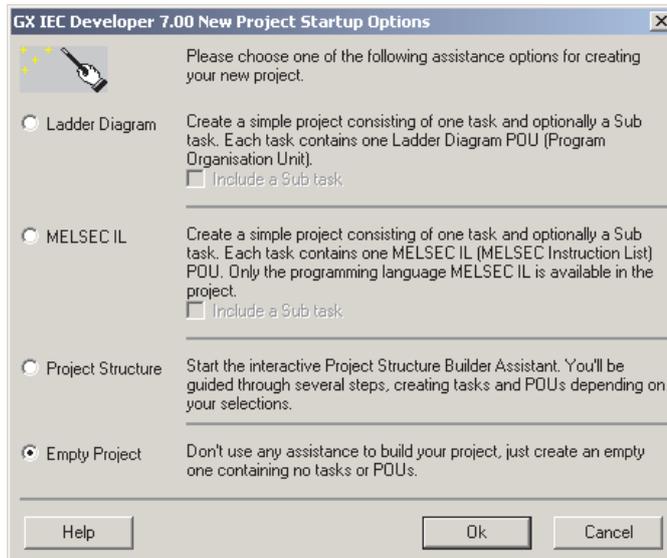


- ③ Задайте название проекта в **New Project Path**. В этом случае используйте "\GX-IEC DATA\CAROUSEL" и щелкните на **Create**, как на следующей иллюстрации:



## Мастер

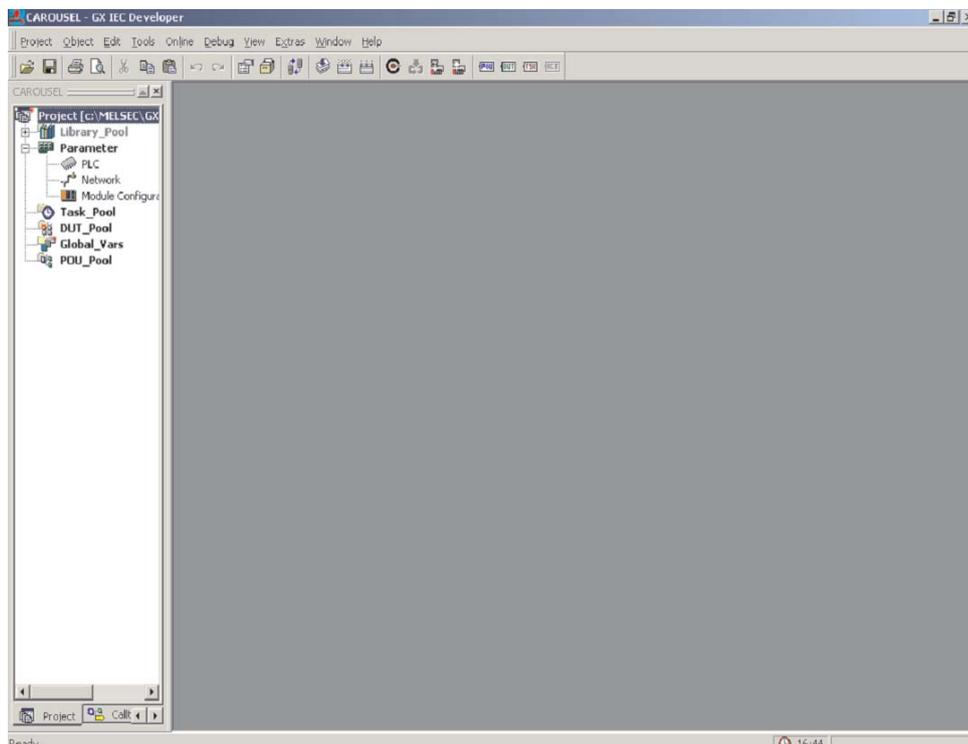
Будет показан мастер запуска проекта **Project Startup Wizard**:



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.

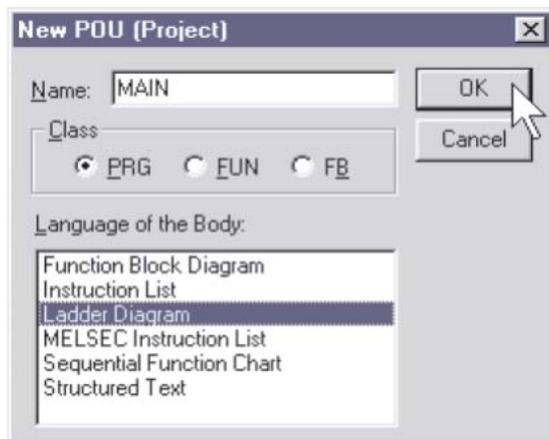
Показан экран проекта дисплея, представленный ниже:



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

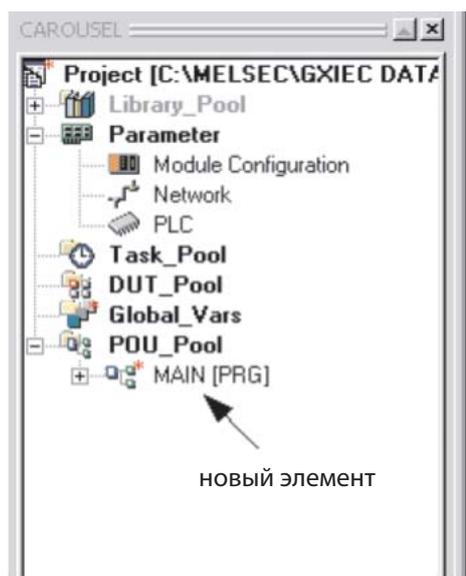
### 4.2.3 Создание нового программного модуля "POU"

- 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:

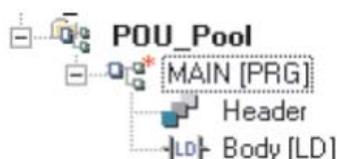


Имя POU будет "MAIN", и его необходимо задать как релейную диаграмму **Ladder Diagram** типа **PRG** (программа).

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



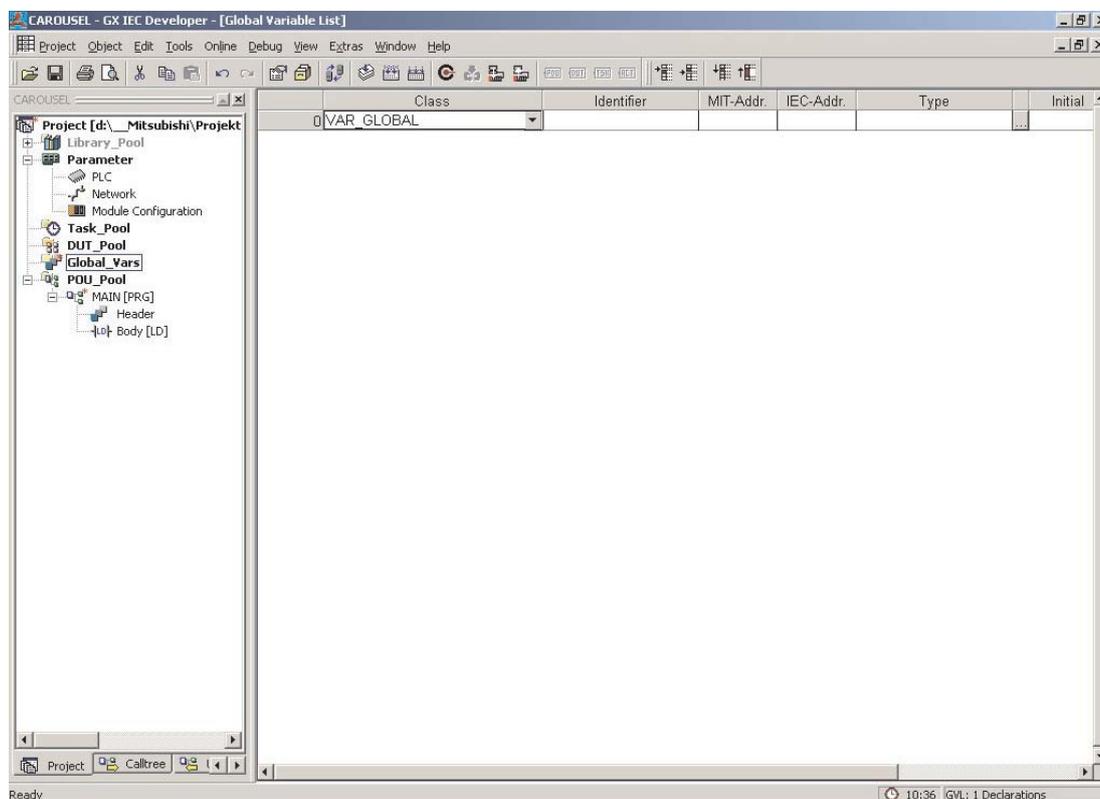
- Дважды щелкните на значке программы **MAIN** или щелкните на символе в POU Pool, чтобы показать структуру каталога и элементы заголовка **Header** и тела **Body** :



## 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 регистры 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 POU.

### 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

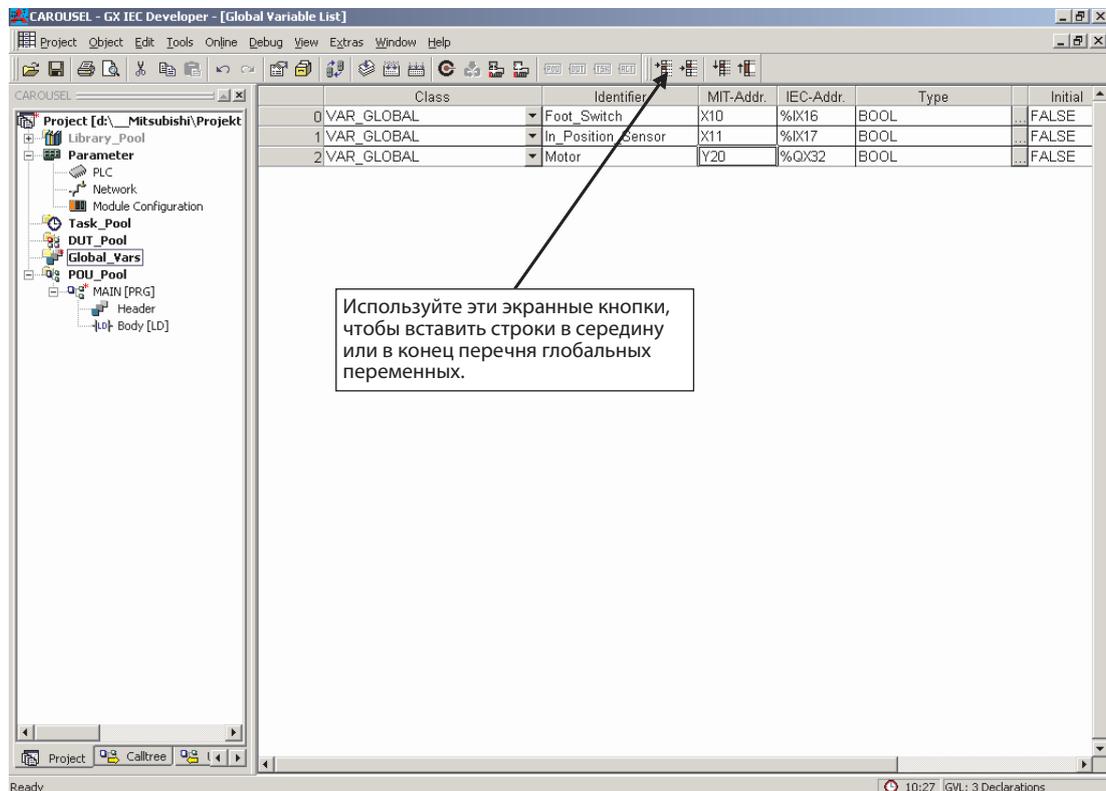
Начальные значения автоматически устанавливаются системой и не могут изменяться пользователем.

- **Comment** - Комментарий

Для каждой переменной можно добавить комментарий до 64 символов

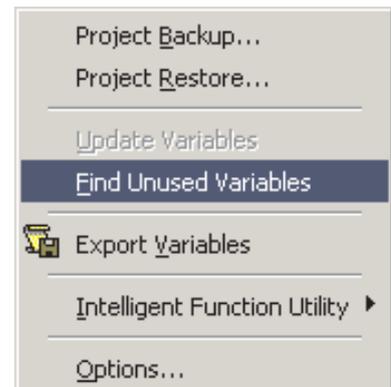
Если в программе не предполагается использовать символические идентификаторы, а только адреса Mitsubishi, то нет никакой необходимости заполнять Список глобальных переменных (GVL). Однако программа потеряет совместимость с IEC61131-3.

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.



### 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

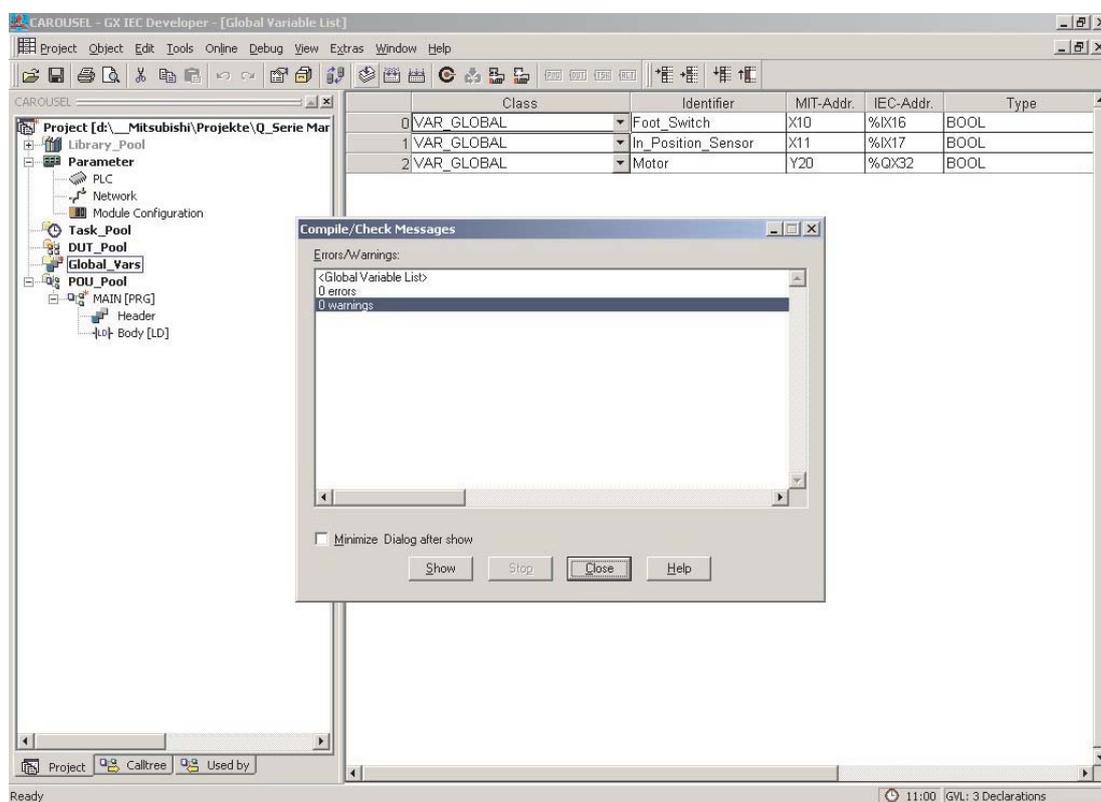
Обнаружение неиспользованных переменных можно выполнить, если проект был создан и с тех пор не изменялся. В противном случае будет показано предупреждающее сообщение.

**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.

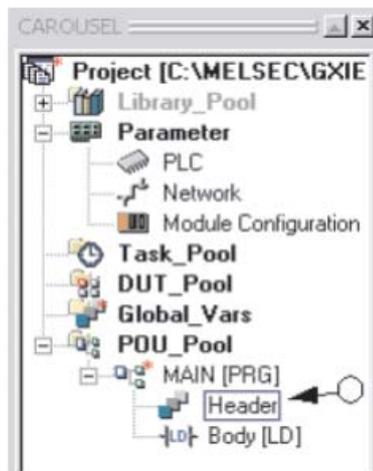
Для всех типов ЦП FX2N, FX3U, Q & AnA(S) или выше полностью поддерживаются значения IEC типа REAL (с плавающей запятой).

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

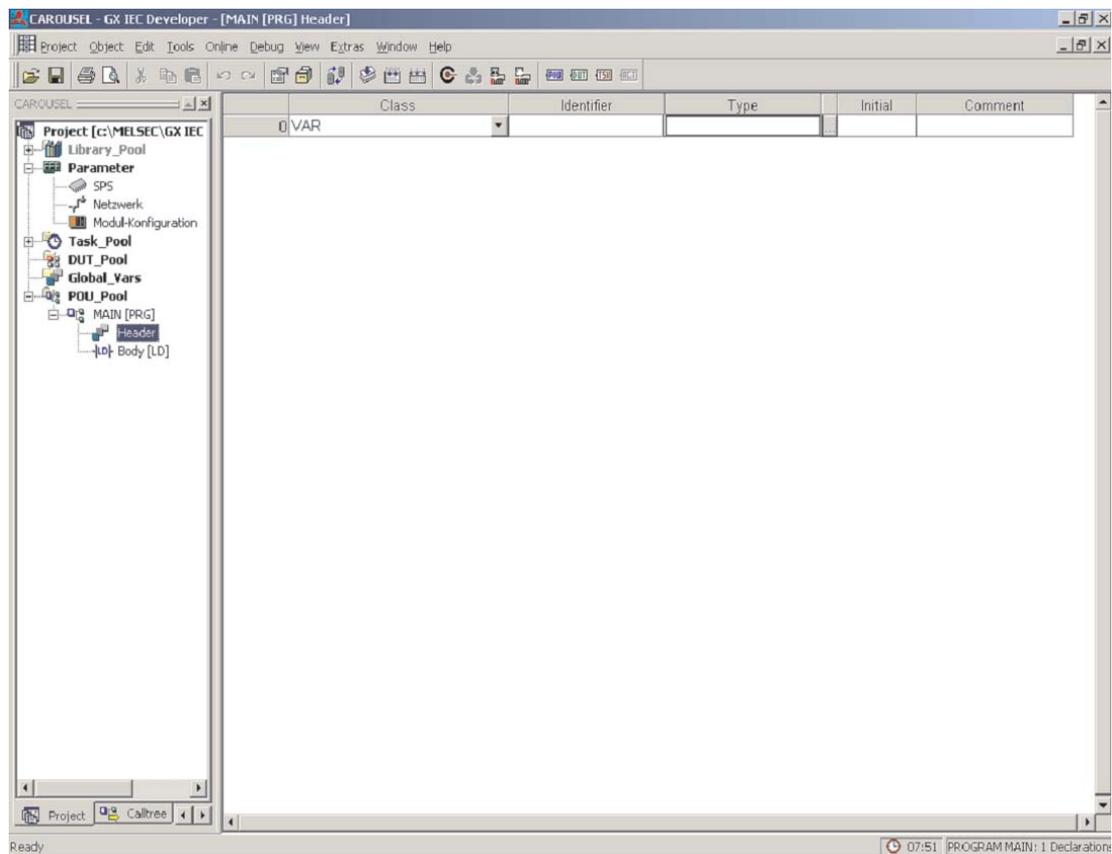


### Открытие заголовка POU

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



Появится следующий экран:

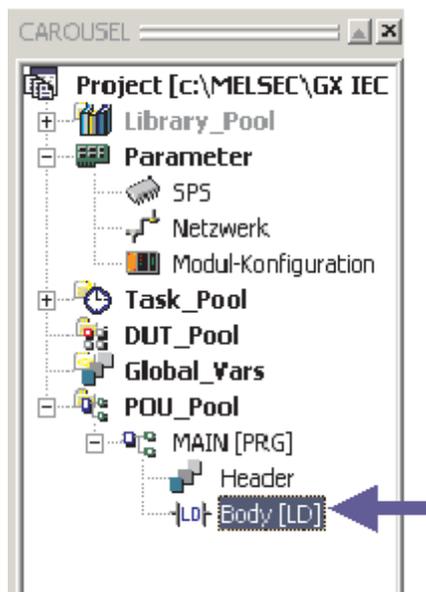


Закройте этот экран заголовка программного модуля.

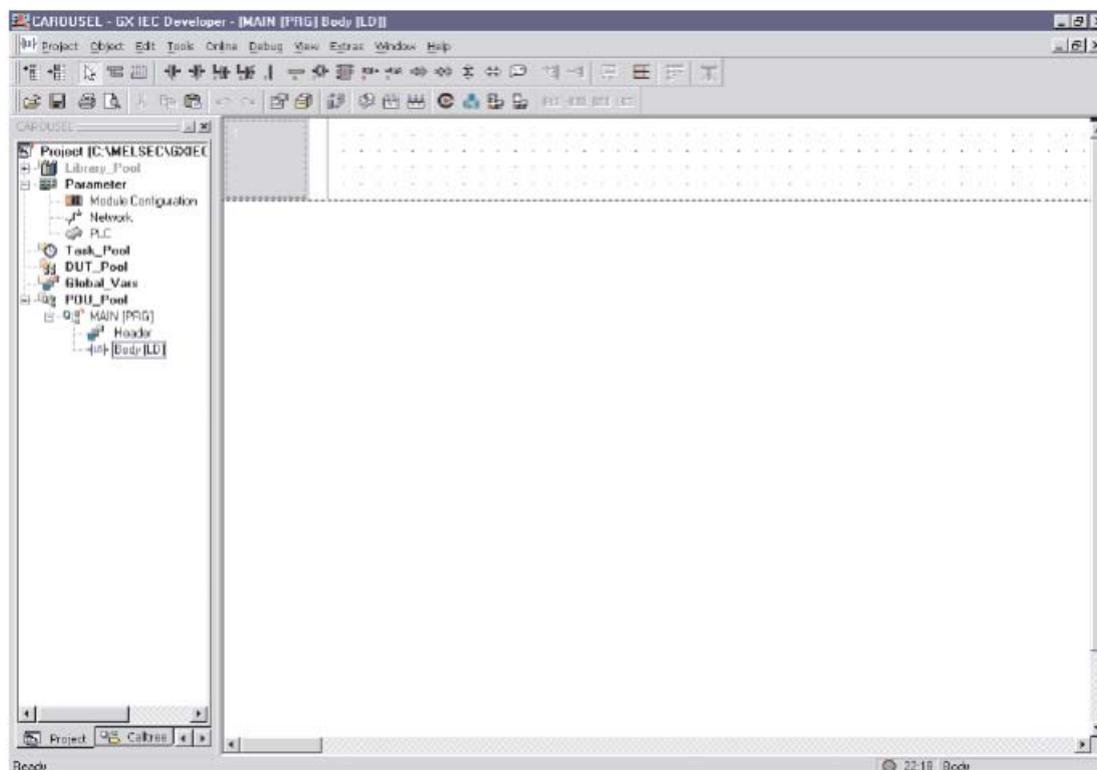
## 4.2.5 Программирование тела программного модуля

Фактическая программа ПЛК хранится в теле программного модуля.

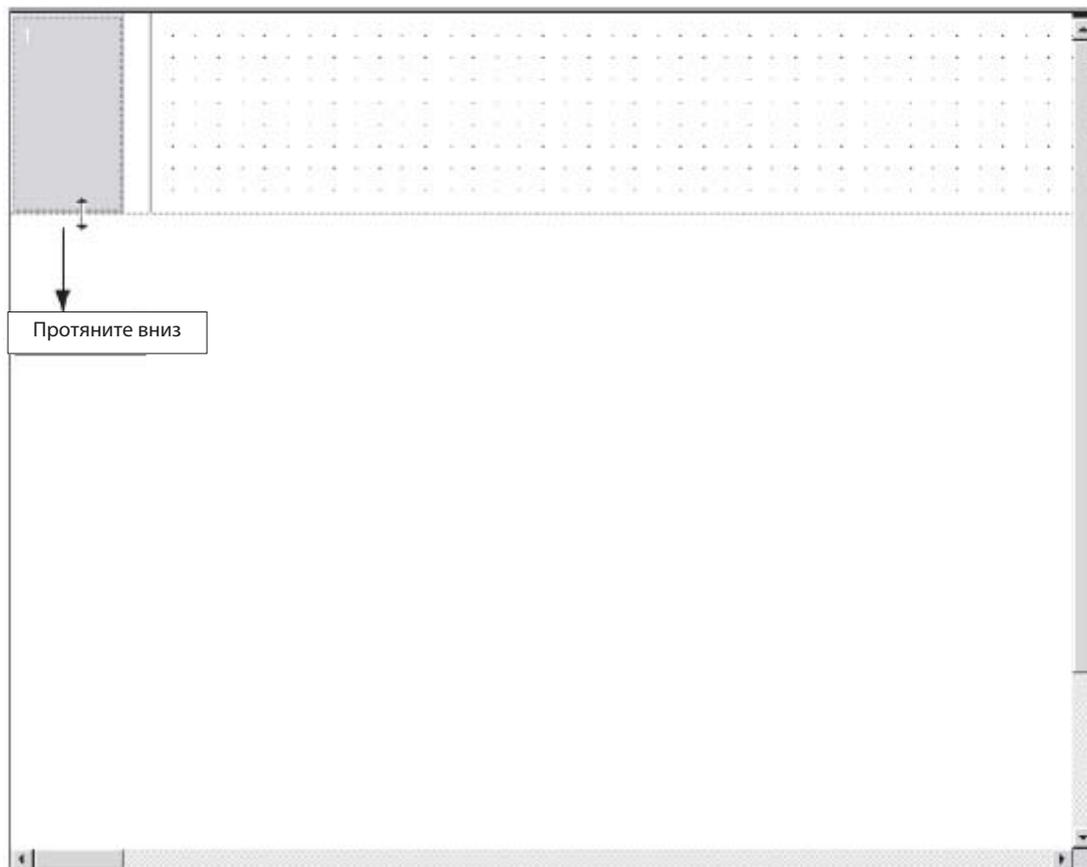
- ① Чтобы открыть Редактор релейных диаграмм, дважды щелкните на пункте **Body** в пуле программных модулей POU\_Pool в навигационном окне проекта:



Появится следующее окно:

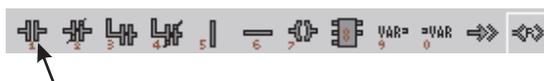


- ② Поместив указатель на границу окна, щелкните и перетащите ее вниз, чтобы увеличить вертикальный размер цепи:



### Использование панели символов релейной диаграммы

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

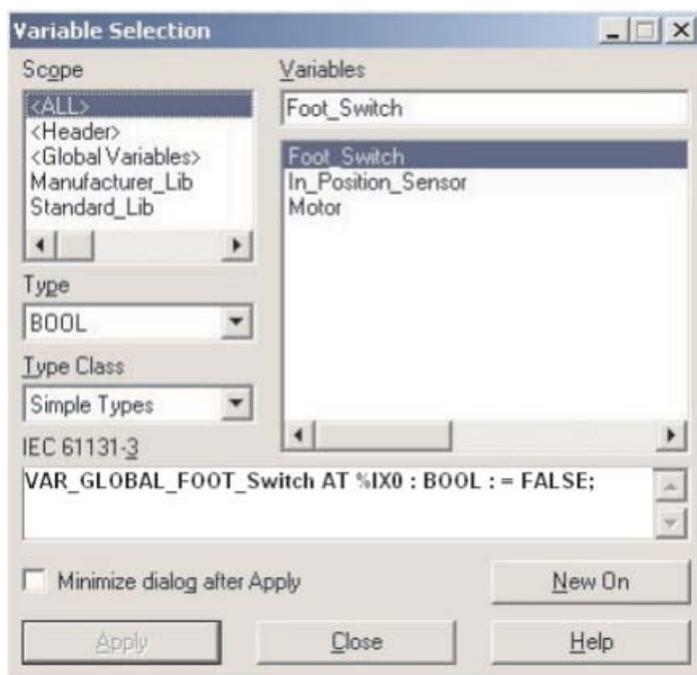


- ④ Перемещая указатель мыши по рабочей области, щелкните, чтобы задать позицию помещения в окне:



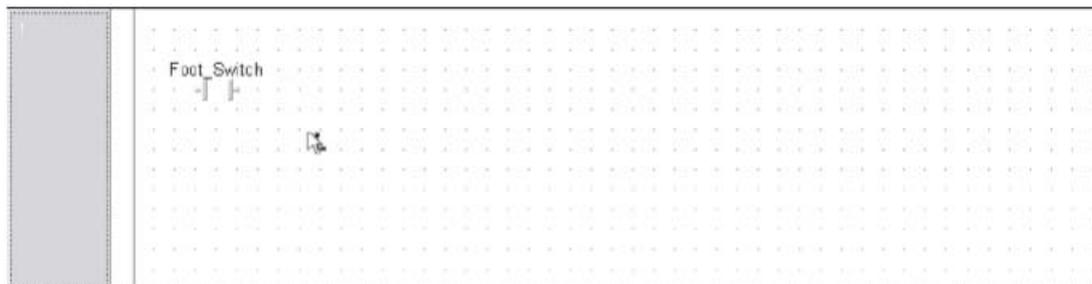
### Выбор переменных из заголовка программного модуля

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



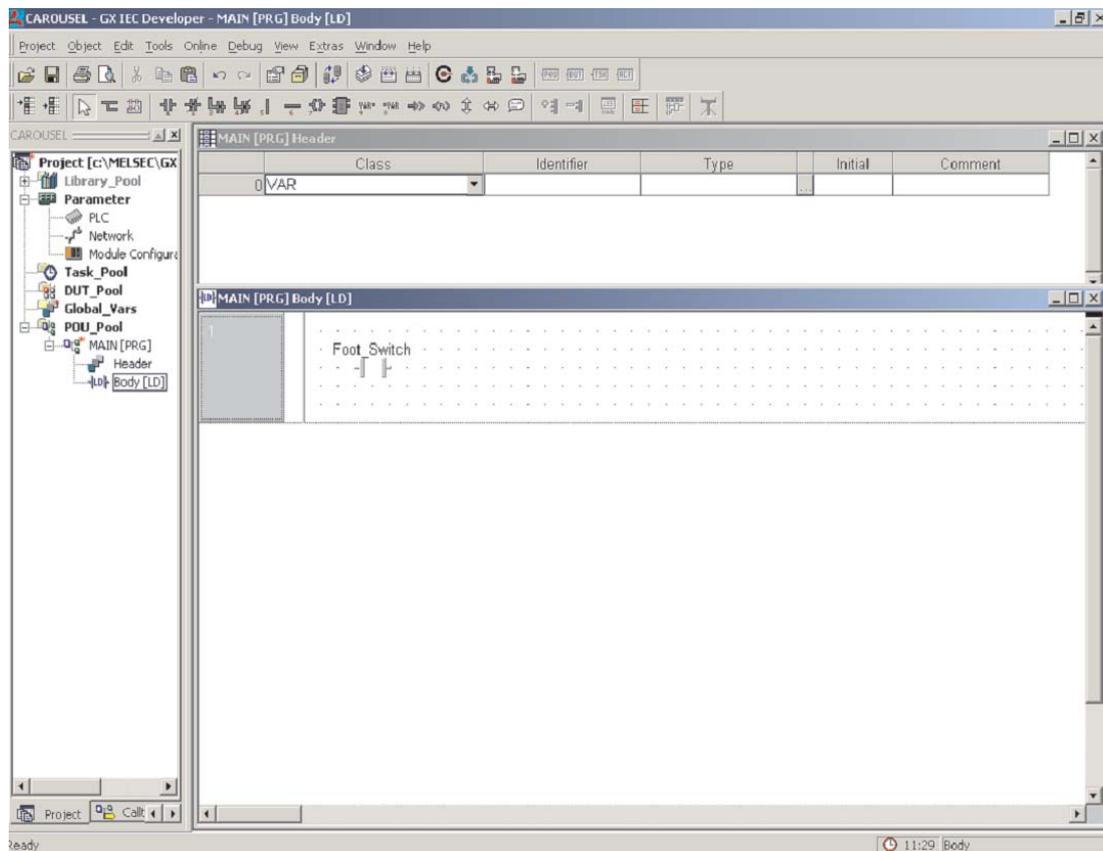
Учтите, что в области **Scope** диалогового окна должен быть выбран текущий заголовок "Header".

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



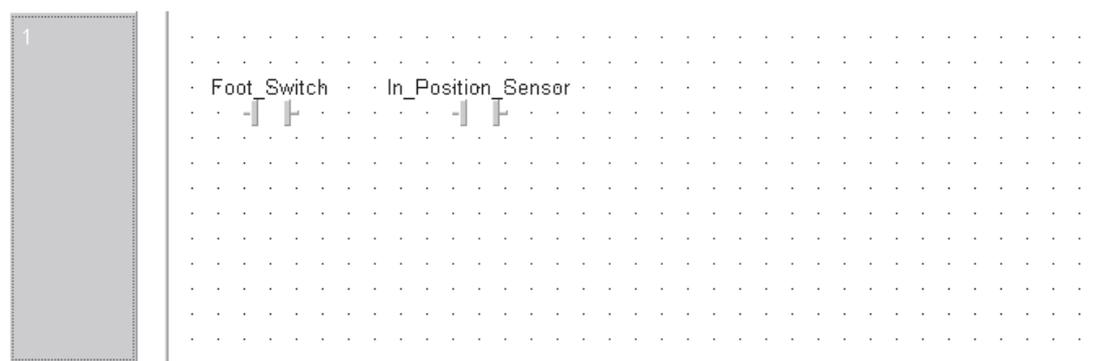
**Альтернативный метод задания переменной: Редактирование на разделенном экране**

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



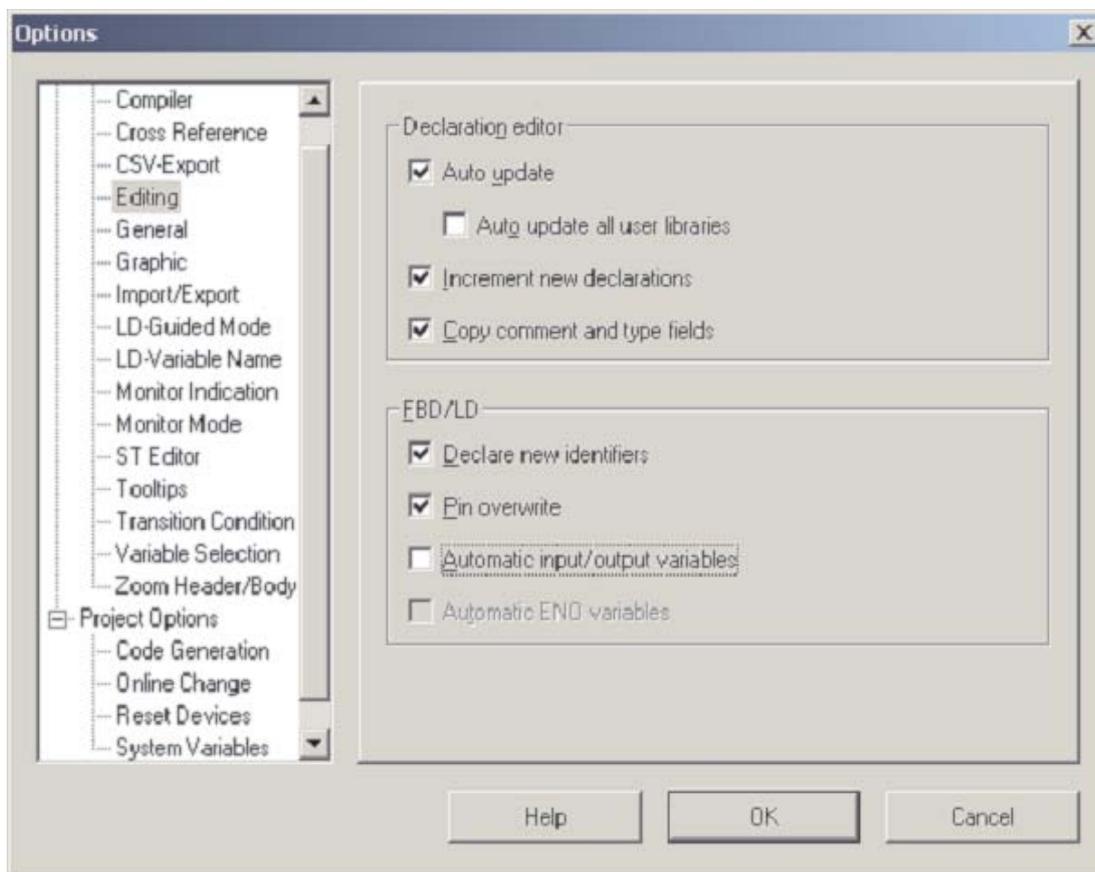
**Продолжайте редактировать проект "Carousel"**

Введите нормально разомкнутый контакт датчика "In\_Position\_Sensor" в положении, показанном на текущем экране, так же, как показано ниже:



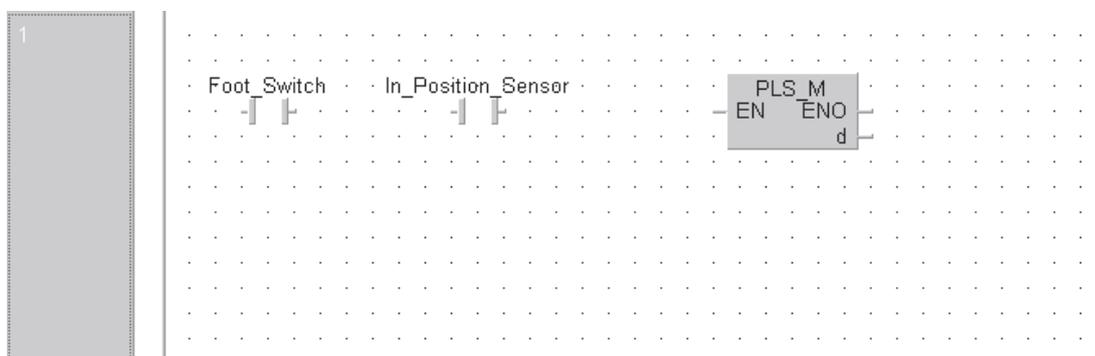
### Ввод команды функционального блока в релейную диаграмму

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:



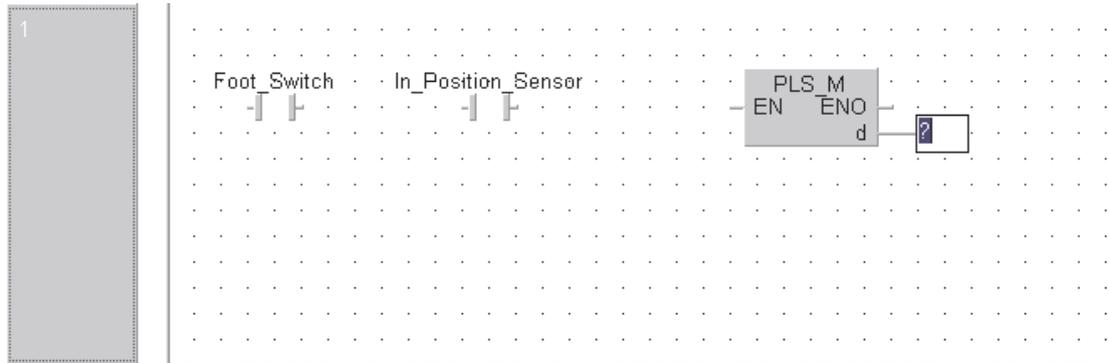
Команда функционального блока MELSEC, "PLS\_M", будет добавлена в программу как выходная функция.

- 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.

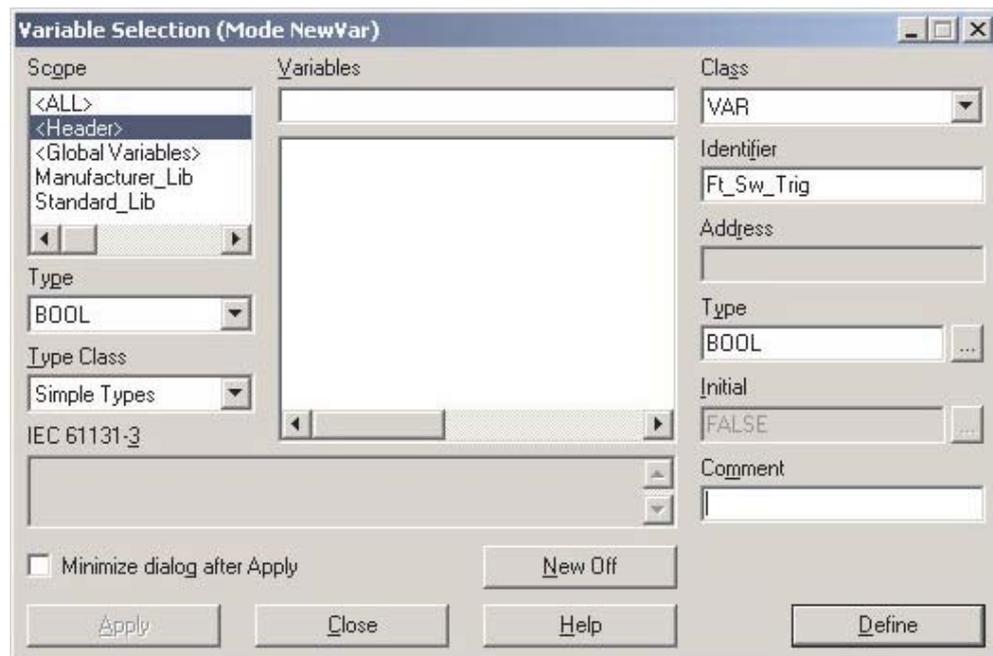


- ③ Введите имя переменной Ft\_Sw\_Trig в пустое окно "?".

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:

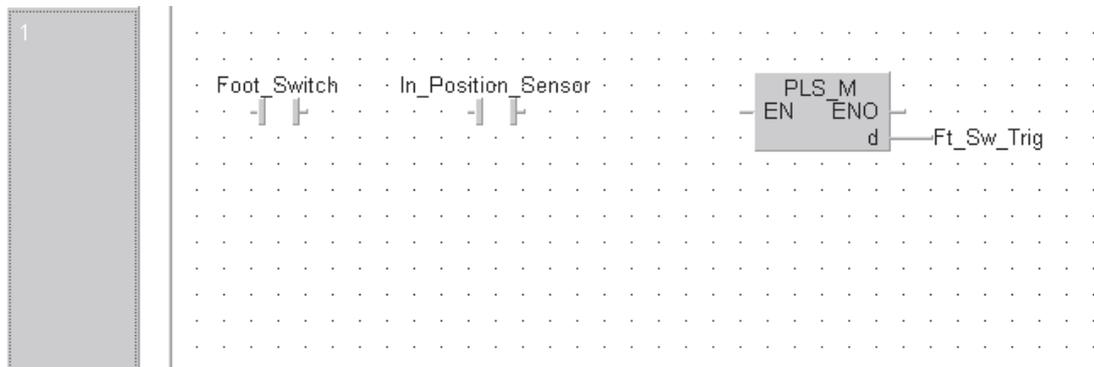


- ⑤ Щелкните на **Define**, чтобы ввести новую переменную в LVL (Локальный заголовок).

**ПРИМЕЧАНИЕ**

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

Дисплей должен иметь следующий вид:

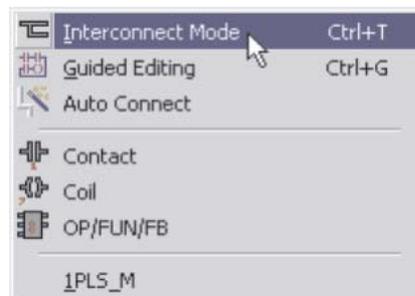


В заключение необходимо составить схему релейной логики, соединив элементы следующим образом.

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

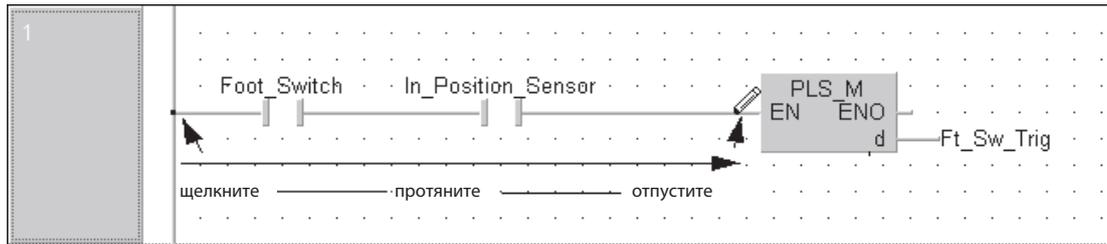


- ⑦ Аналогичным образом, щелкните, чтобы выбрать **Interconnect Mode**.



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:

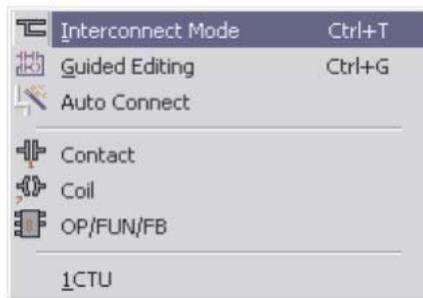


Теперь схема закончена.

### Изменение режима курсора

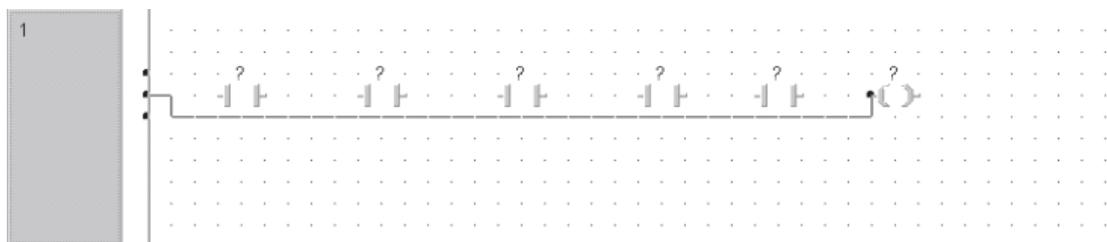
Перед тем, как продолжать работу с нашим примером, необходимо понять работу курсора и различные режимы редактирования, которые имеются в программном пакете.

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.

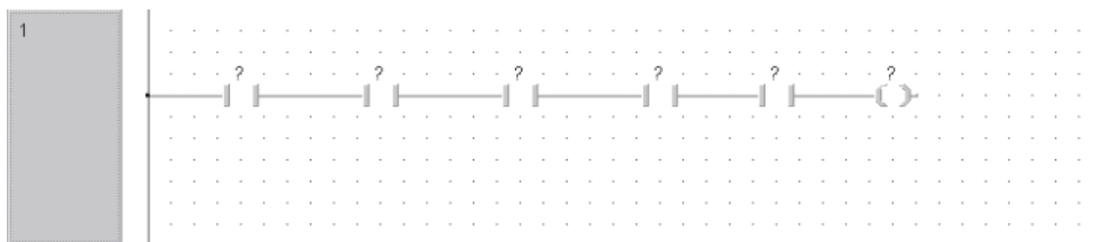


### Предосторожности при использовании редактора релейных диаграмм

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:



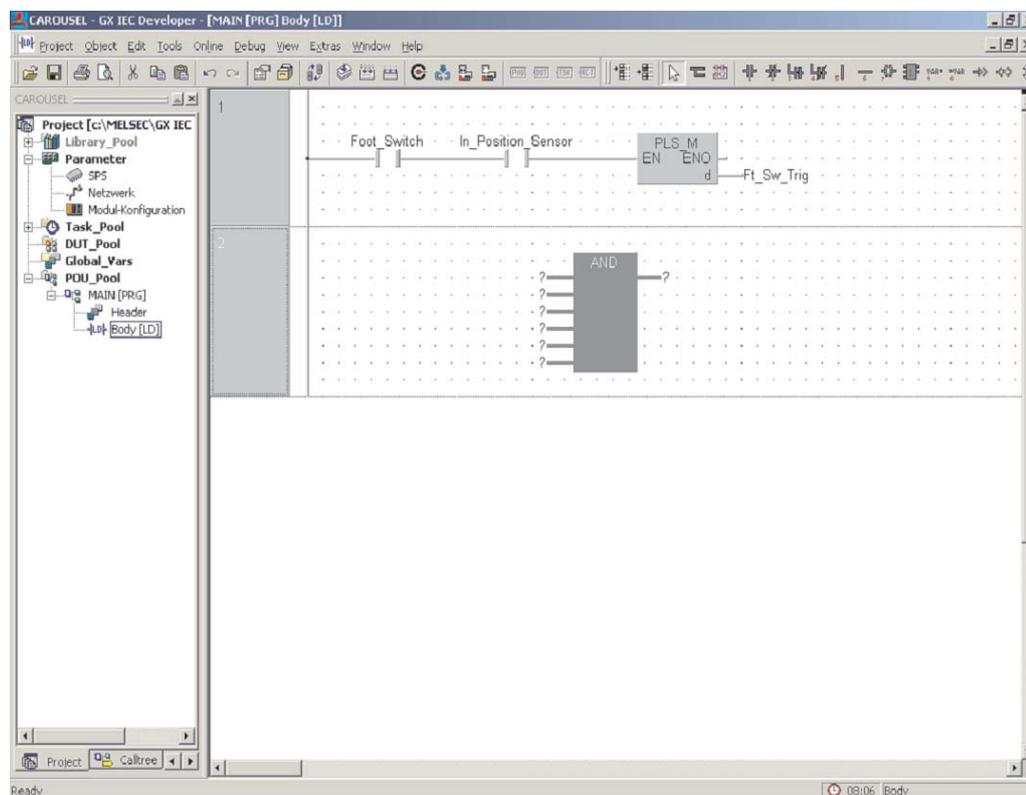
Ручка позволяет затем пройти через все контакты, от сборной шины до катушки. В редакторе релейных диаграмм мы предлагаем активизировать опцию **Auto connect** при помещении элементы на тело программного модуля или соединении параллельных элементов. Однако, эту опцию следует отключить при подключении 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. 

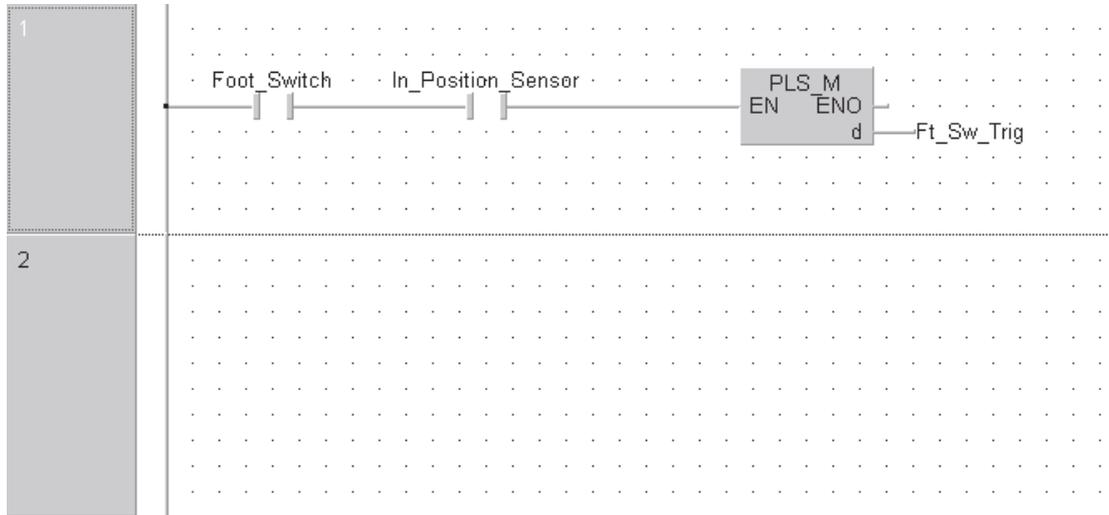
При использовании функций с несколькими входами, например, MUL, число входов можно увеличить/уменьшить, используя специальную панель инструментов с соответствующими значками. Это также можно сделать, поместив курсор на нижний край функции, удерживая левую кнопку мыши, и затем перетащив курсор, как показано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



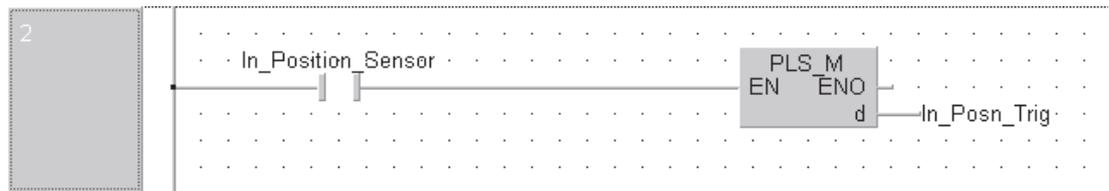
### Создание новой цепи программы

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

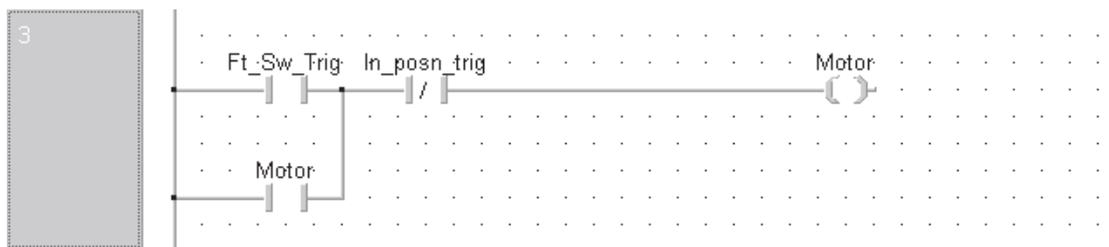
Появляется пустая схема:



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

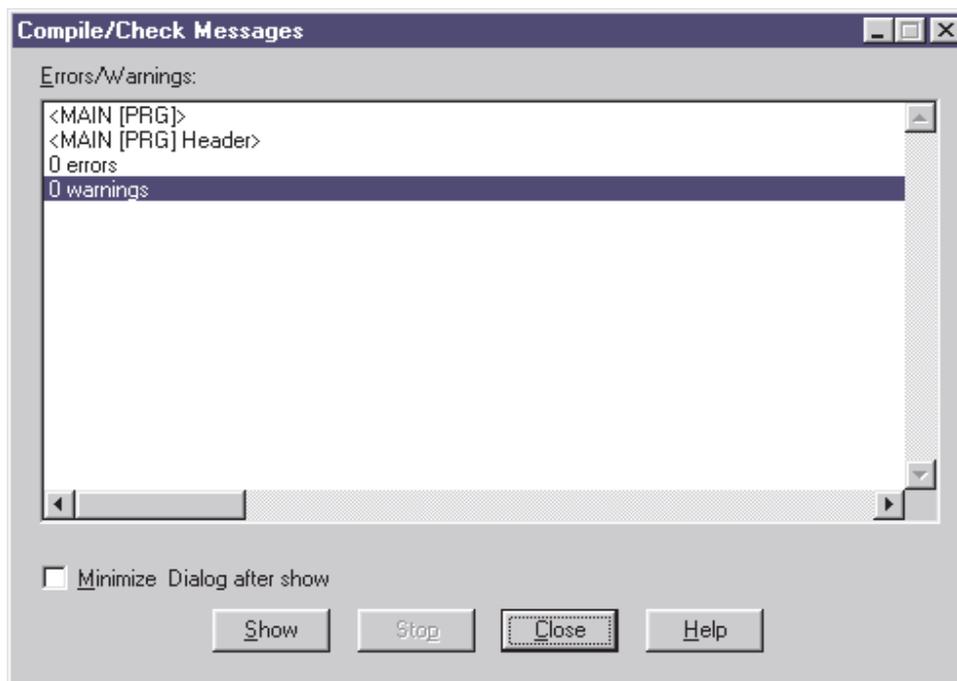


- ③ Наконец, введите следующую цепь, как показано:



### Проверка введенной программы

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



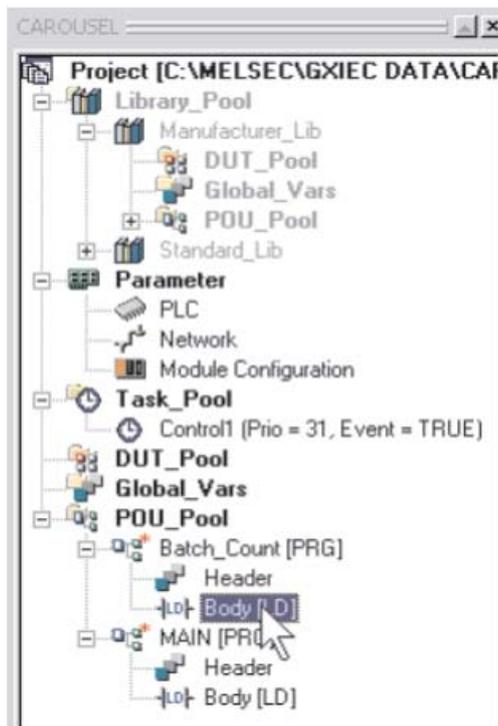
### Программирование счетчиков и таймеров

На этом функции для перемещения поворотного стола, описанные в начале раздела, запрограммированы. На примере дополнительных функций продемонстрируем применение счетчиков и таймеров:

- Счет изготовленного количества  
Этот счетчик регистрирует каждое включение приводного двигателя. Так как при этом каждый раз к рабочему месту транспортируется новая обрабатываемая деталь, значение счета соответствует количеству изготовленных изделий.
- Индикация каждых 10 изготовленных изделий  
После подачи на рабочее место 10 деталей лампа мигает с секундным тактом до тех пор, пока счетчик не будет сброшен нажатием на кнопку.

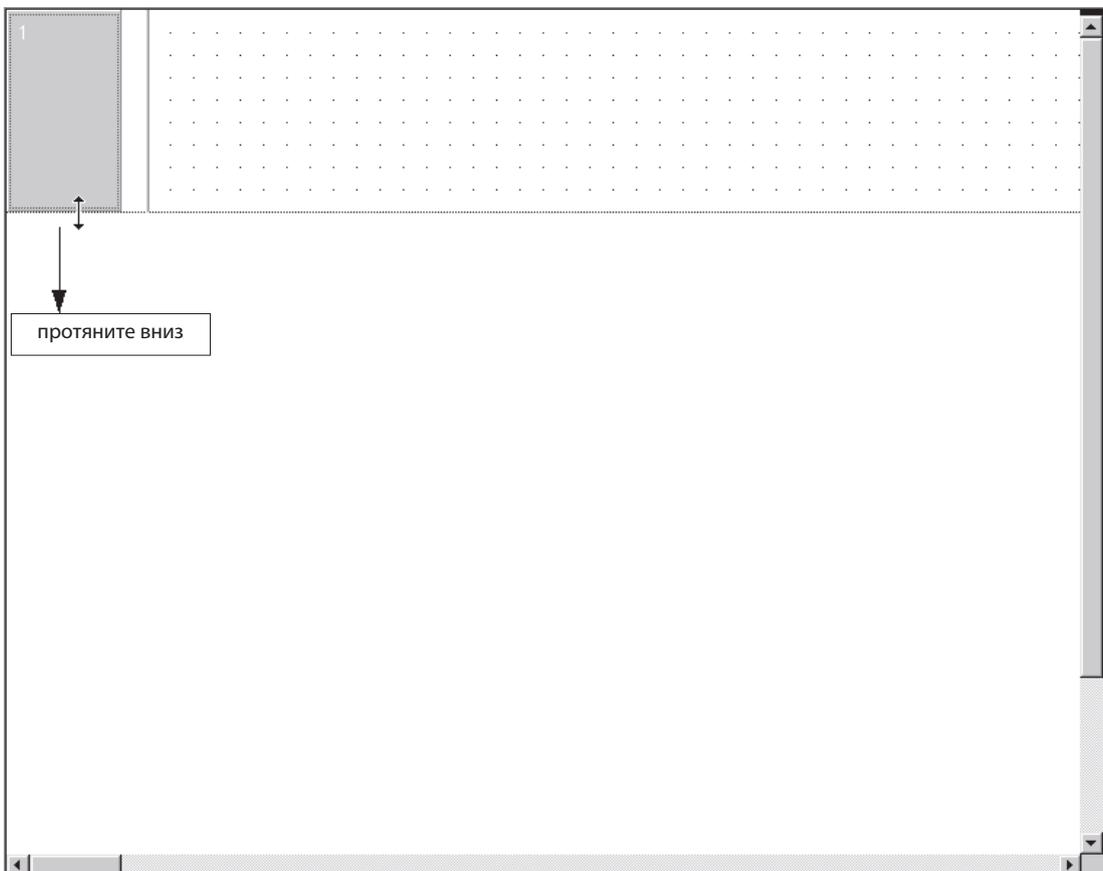
Эти функции счета и индикации программируются в новом POU:

① Создайте новый программный модуль, щелкнув на кнопке .



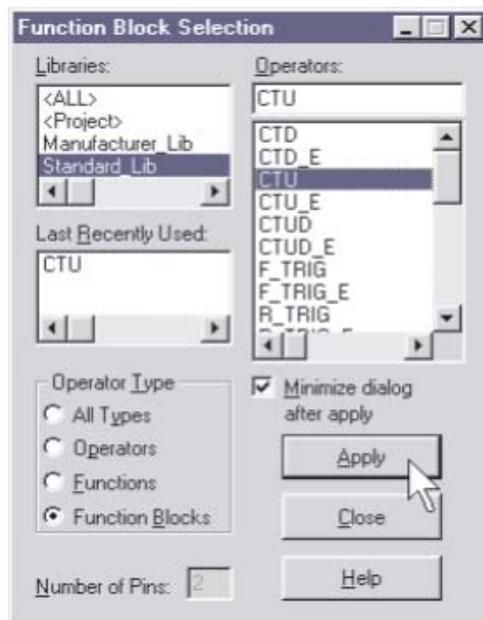
② Выберите тело нового программного модуля, открыв только что созданный элемент в навигационном окне проекта.

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:

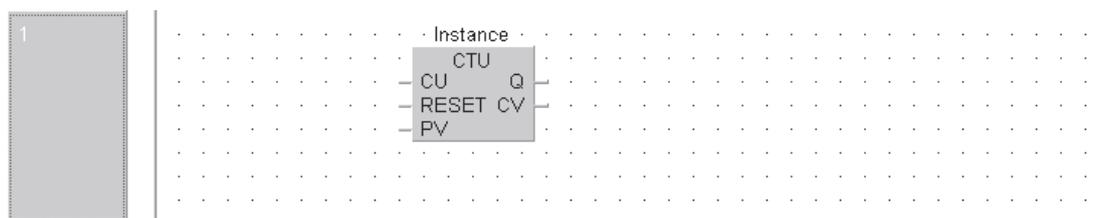


## Счетная функция

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



Поместите функциональный блок IEC на пустую схему релейной логики:

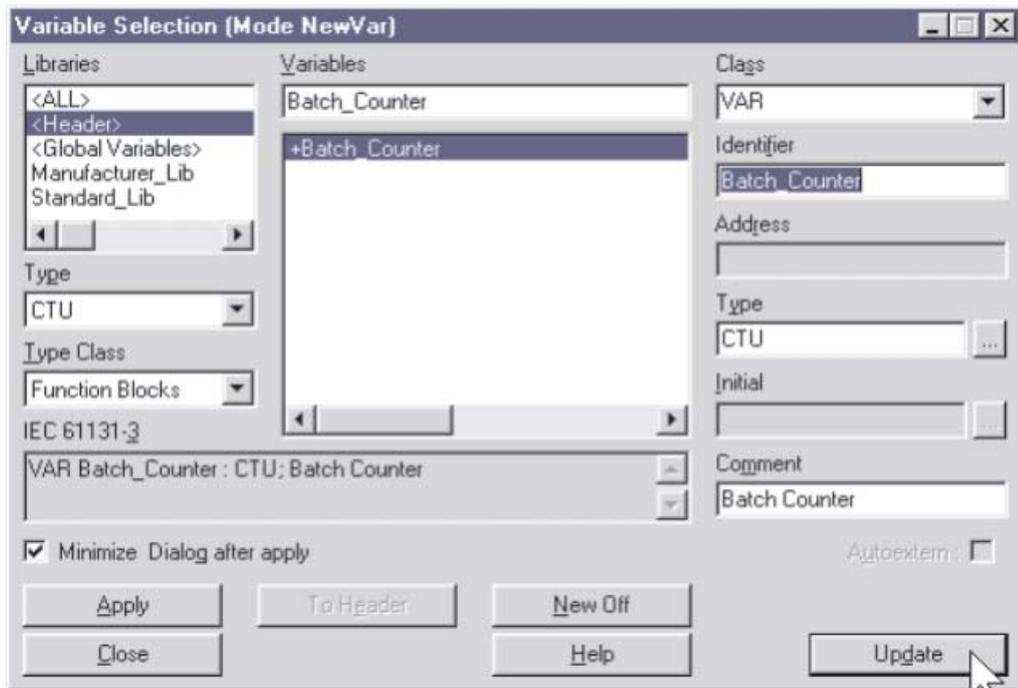


## Экземпляры функциональных блоков

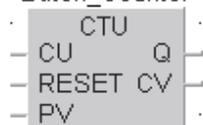
Функциональные блоки можно вызывать только как "**Экземпляры**." Процесс создания копии функционального блока выполняется в заголовке программного модуля, в котором должен использоваться экземпляр. В этом заголовке функциональный блок будет объявлен как переменная и результирующему экземпляру присваивается 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

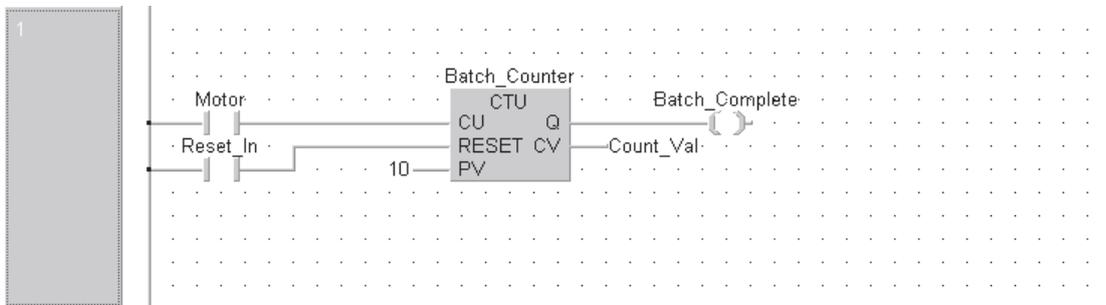
- ① Чтобы создать новое имя для этого экземпляра функционального блока CTU в данном программном модуле, щелкните на имени переменной **Instance** над функциональным блоком CTU. Нажмите F2, чтобы открыть диалоговое окно выбора переменной **Variable selection**. Заполните результирующее окно, как показано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



· Batch\_Counter · ② Щелкните на **Apply**, затем на **Update**, и имя переменной изменится, как показано слева.



③ 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 X12 или IEC %IX18. Для этого необходимо новый следующий элемент в GVL:

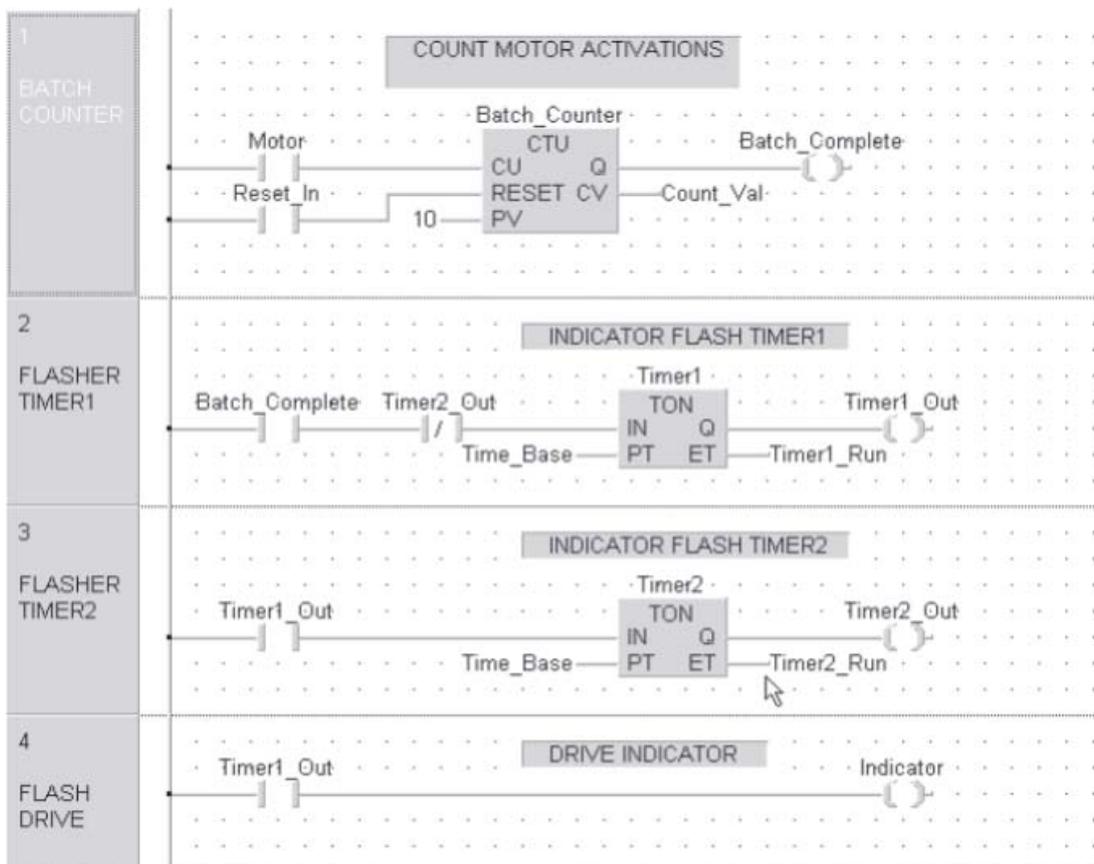
	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	Foot_Switch	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	In_Position_Sensor	X11	%IX17	BOOL	FALSE
2	VAR_GLOBAL	Reset_In	X12	%IX18	BOOL	FALSE
3	VAR_GLOBAL	Motor	Y20	%QX32	BOOL	FALSE

	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 скомпилировать 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	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	In_Position_Sensor	X11	%IX17	BOOL	FALSE
2	VAR_GLOBAL	Reset_In	X12	%IX18	BOOL	FALSE
3	VAR_GLOBAL	Motor	Y20	%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 скомпилировать 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	

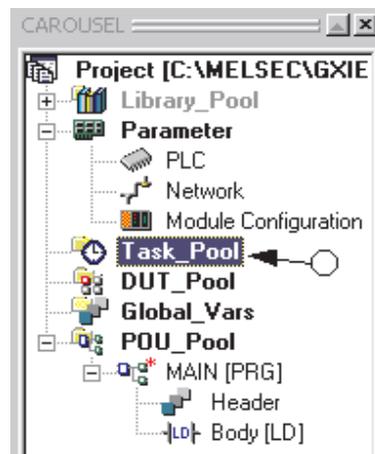
### Заголовок для программного модуля "MAIN":

	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**.

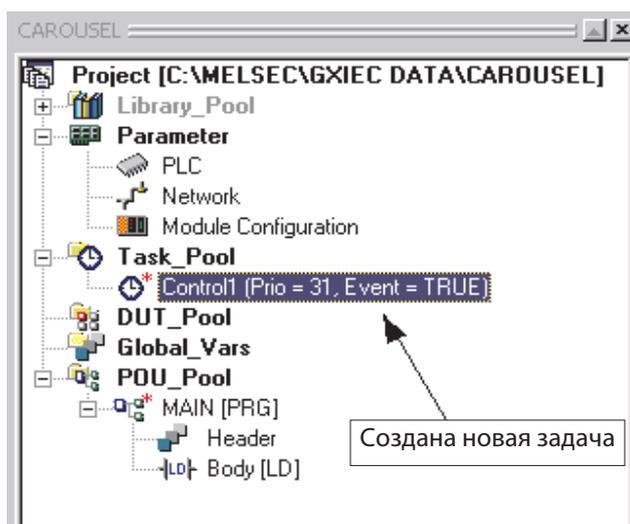
- Щелкните один раз, чтобы выделить значок **TASK\_Pool** в навигационной области проекта.



- 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**.
- Введите имя новой задачи ("Control1") в окне запроса.



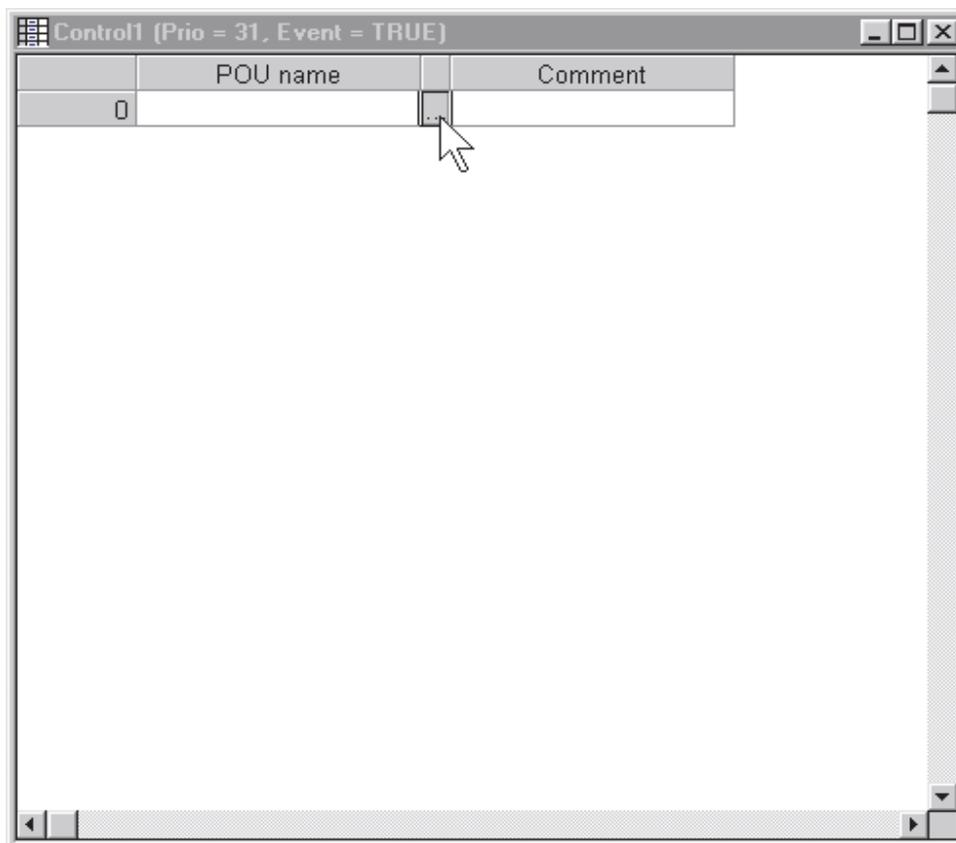
- Щелкните на кнопке **OK** и в окне Project Navigation теперь показана только что созданная задача "Control1":



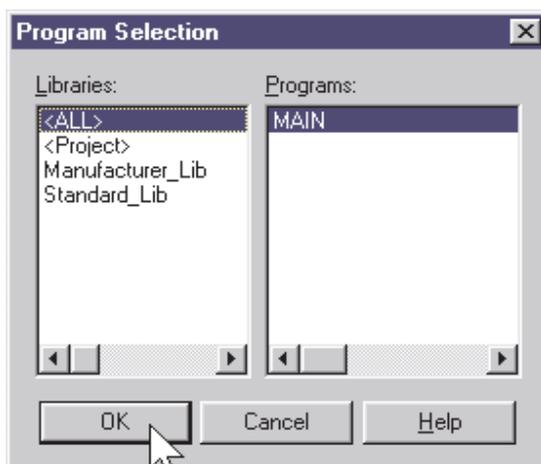
### Назначение программного модуля задаче

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":



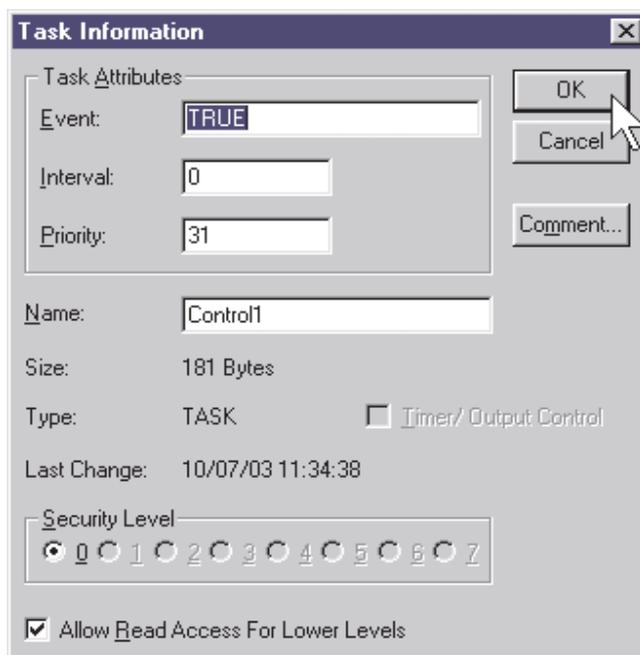
- 2 Щелкните на среднем многоточии "просмотр вариантов", как показано выше. Будет показано следующее диалоговое окно:



- 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 .

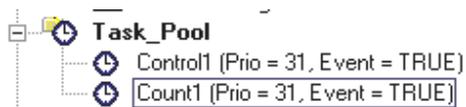
### Создание новой задачи для программного модуля "Batch-Count"

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.

## 4.2.1 Документирование программы

### Заголовок звена

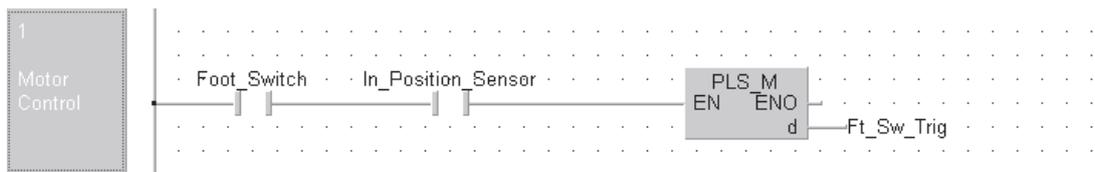
Использование заголовка звена является опциональным; он позволяет идентифицировать звено программы описательным названием длиной до 22 символов. Это может помочь в работе с проектами с большим количеством звеньев.

- 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:



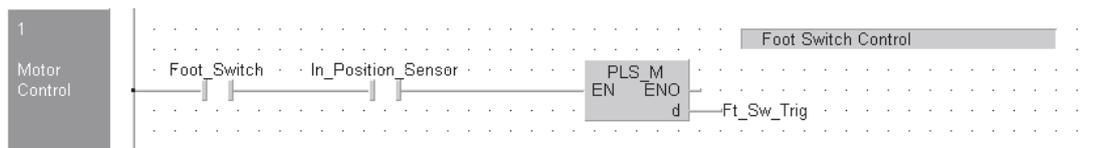
Учтите, что для названия может потребоваться предварительное форматирование (дополнительные пробелы), в зависимости от установленного разрешения экрана, чтобы текст правильно читался с автоматическим переходом слов на следующую строку и вписывался в доступное горизонтальное пространство (макс. 22 символа).

### Комментарии к звену

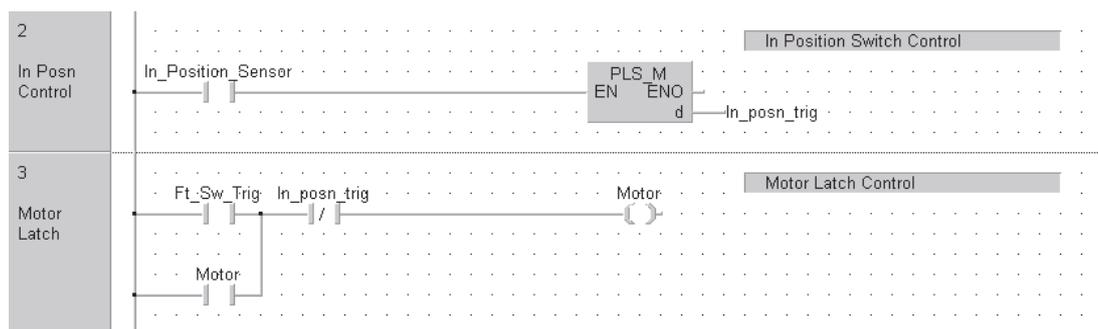
Комментарии позволяют добавлять описатели с фактически ручным текстом в любом месте в звене релейной диаграммы. Они жизненно важны, чтобы обеспечить описание работы программы.

- 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

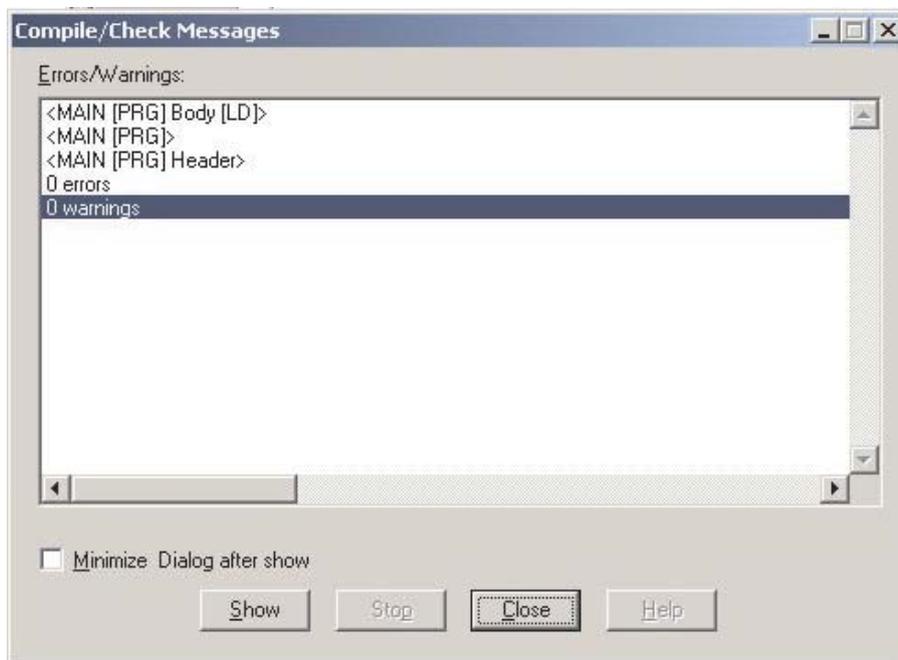
Щелкните один раз на комментарии, чтобы выделить его, и нажмите кнопку.

### Вырезание / копирование комментария

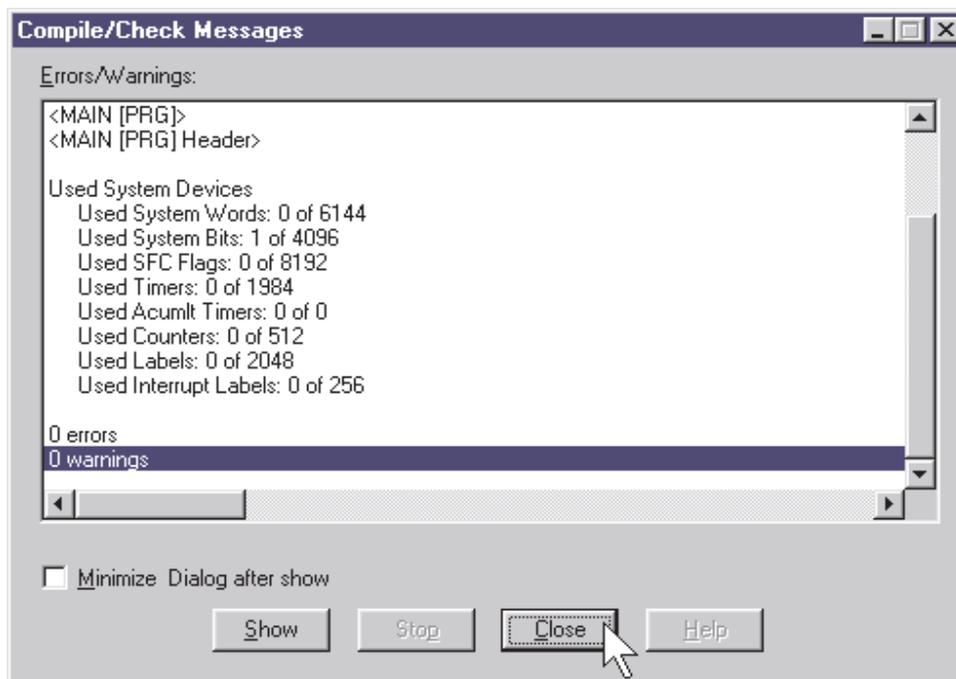
Для дублирования комментариев щелкните на левом конце исходного комментария, чтобы выделить его. Используйте процедуру вырезания/копирования - вклеивания Windows и щелкните мышью еще раз, чтобы установить конечную позицию комментария в другом звене.

## 4.2.2 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:

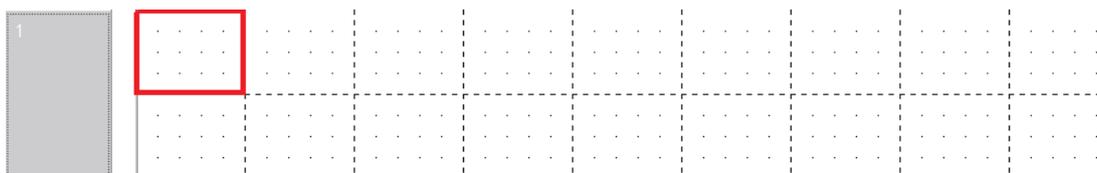


- ③ Щелкните на **Close**, чтобы выйти из этого экрана.

### 4.2.3 Иллюстрация: Направляемый режим ввода релейной диаграммы

В дополнение к ручным методам ввода релейных диаграмм, в GX IEC Developer версии 6 имеется продвинутое средство мониторинга направляемого метода ввода релейной диаграммы, которое 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.

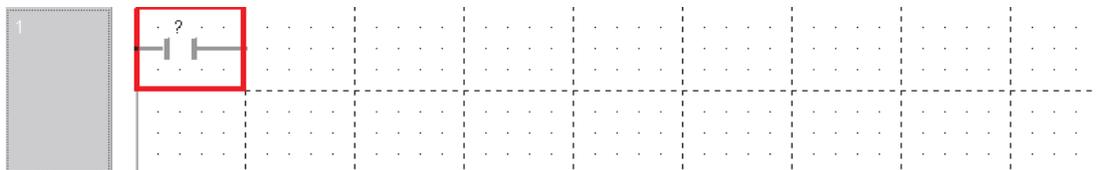
- ① Войдите в режим **Guided Entry Monitor**, нажав кнопку  на панели инструментов. В области редактирования появится следующая матрица:



- 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

### 4.3.1 Подключение с помощью периферийных устройств

Для передачи программы контроллер должен быть соединен с программатором и должно быть включено питание контроллера.

Для соединения компьютера со средой программирования GX IEC Developer и контроллером Mitsubishi имеются несколько возможностей:

- Интерфейс программатора серии MELSEC FX, A или QnA

Для подключения к интерфейсу программатора используется кабель SC 09. В этот кабель встроен конвертор интерфейсов RS232/RS422, согласующий сигналы компьютера с сигналами контроллера.

- Интерфейс программатора MELSEC System Q

Для подключения компьютера к интерфейсу программатора контроллеров MELSEC System Q используется специальный кабель RS232.

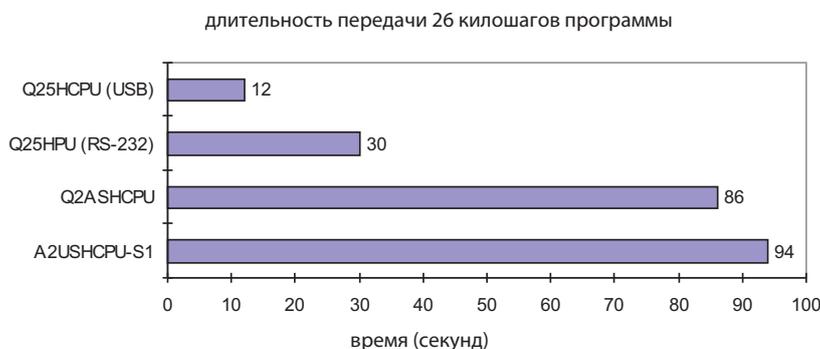
- Интерфейс USB MELSEC System Q

Для соединения компьютера с центральным процессором используется стандартный кабель USB. Рекомендуется использовать интерфейс USB в связи со свойственной ему высокой скоростью передачи.

Соедините компьютер с контроллером учебной стойки, как это показано на рисунке:



На следующей диаграмме сравнивается время передачи программы для самого быстрого процессора серии "A" с временем передачи для процессоров серий QnA и "MELSEC System Q". Обратите внимание на малое время передачи у "System Q" по сравнению с серией "A".



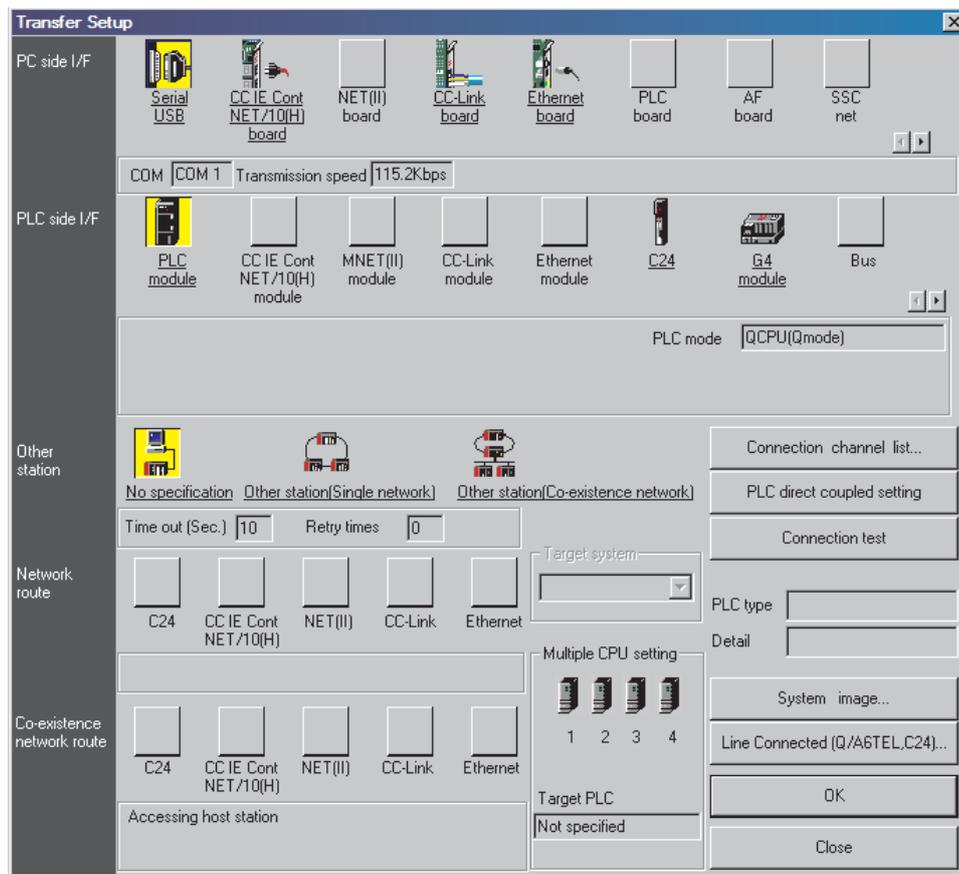
### 4.3.2 Конфигурация коммуникационного порта

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.

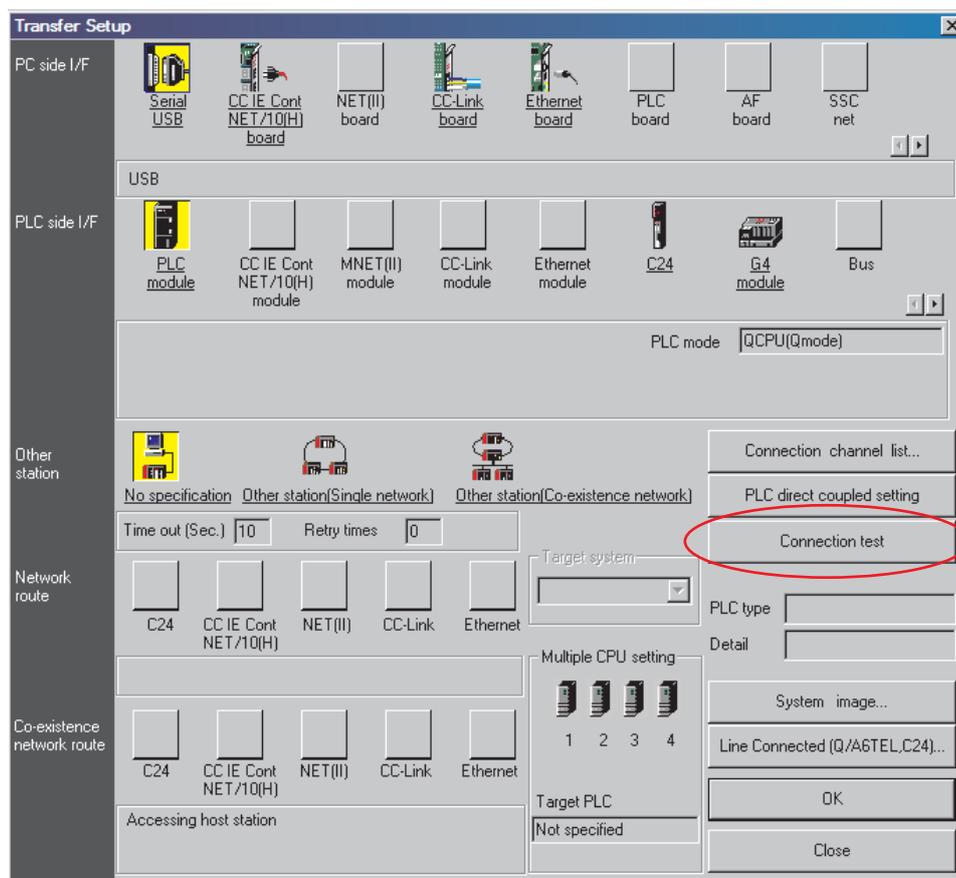


- ② Выберите интерфейс компьютера двойным щелчком по **"Serial/USB"** в строке **"PC side I/F"** (интерфейс со стороны компьютера). В результате появляется показанное справа окно диалога.



- ③ Выберите **USB**, как это показано вверху, и щелкните по **OK**.

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



Должно появиться следующее сообщение:

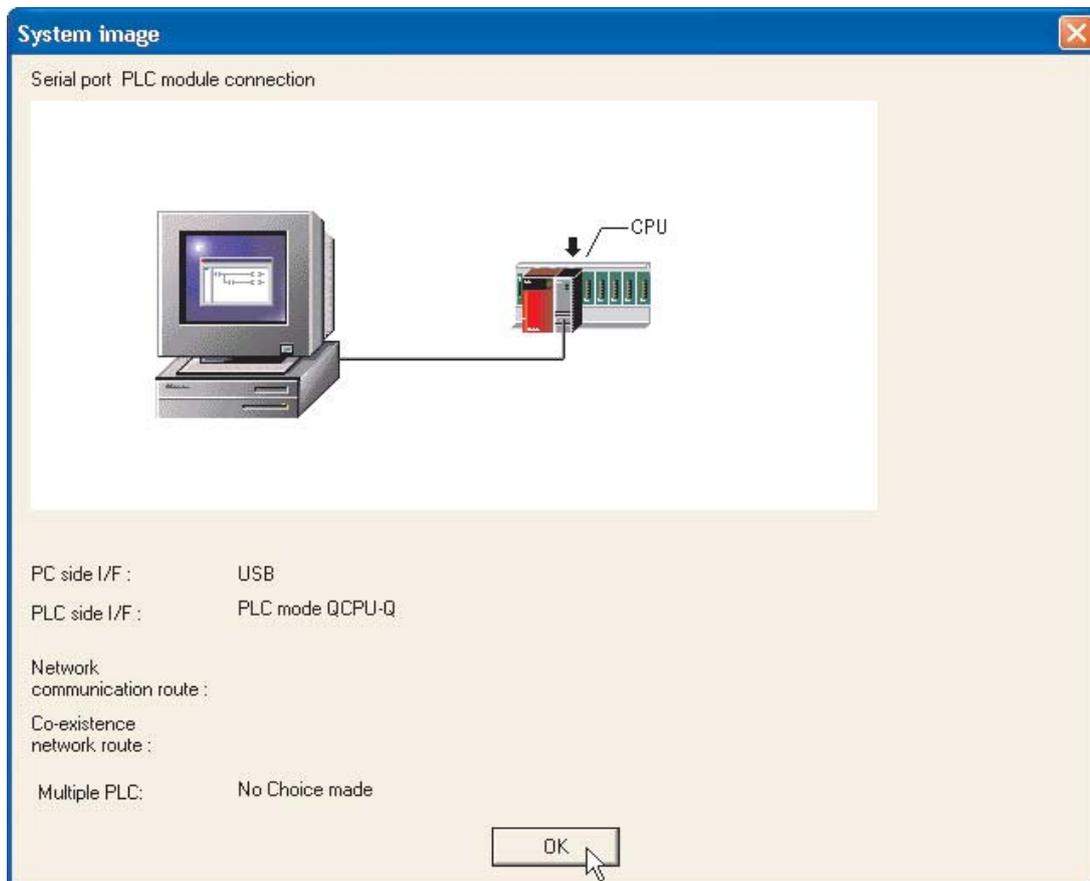


- ⑤ Щелкните на **OK**, чтобы закрыть это сообщение.

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**.



Из этого обзорного изображения можно видеть, что для соединения компьютера с контроллером используется интерфейс USB.

- 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.

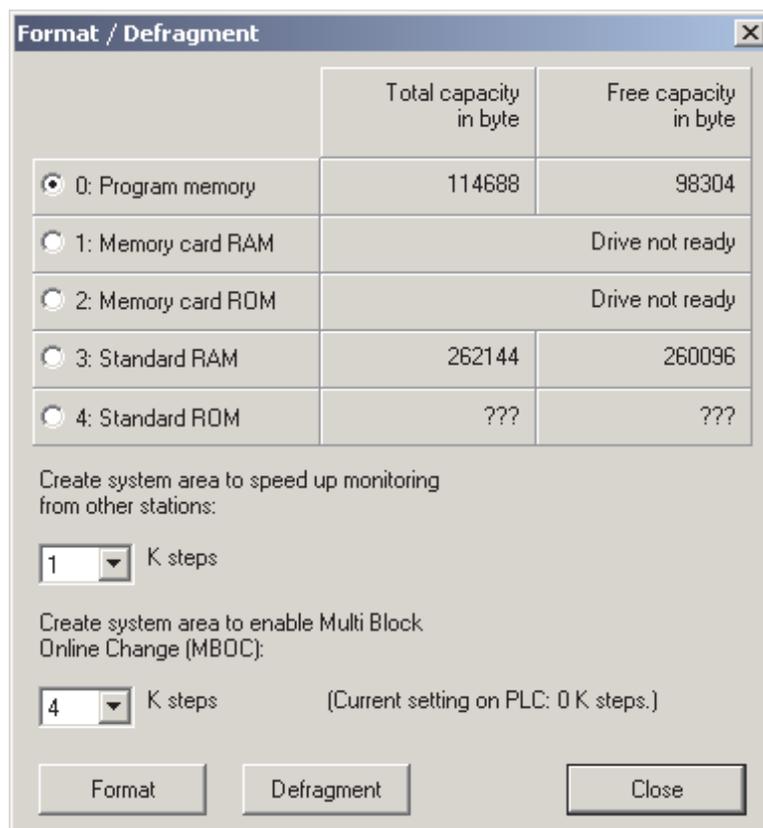
- ③ Выберите **OK**, чтобы закрыть окно **System image** и вернуться в окно **Connection setup**. Затем щелкните на кнопке **OK**, чтобы закрыть окно **Connection Setup**. Если вы выйдете из окна **Connection Setup**, используя кнопку **Close**, настройки не будут сохранены.

### 4.3.3 Форматирование памяти контроллера

Прежде чем впервые передавать параметры или программу в контроллер MELSEC серии "System Q" или QnA, необходимо отформатировать память центрального процессора. Перед первым применением карты памяти ее также необходимо отформатировать.

Память центрального процессора MELSEC System Q подразделена на различные дисководы (см. раздел 2.6.1).

Выберите в меню "**Online**" строку "**Format Drive**". Появляется диалоговое окно "**Format / Defragment**":



- 0 ... 4: дисковод, который требуется сформатировать или дефрагментировать
- Создание системной области для ускорения мониторинга со стороны других систем (**Create system area to speed up monitoring from other stations**)

Если контроллер MELSEC серии "System Q" или QnA соединен с GX IEC Developer и осуществляется наблюдение, например, за состояниями операндов другого контроллера MELSEC серии "System Q" или QnA, то для этого в обоих контроллерах должна быть создана системная область размером по меньшей мере 1 килошаг. Если в контроллерах эта системная область не создана, мониторинг удаленного контроллера не возможен.

- Создание системной области для многоблочного изменения в режиме он-лайн (MBOC) (**Create system are to enable Multi Block Online Change (MBOC)**)

В центральных процессорах, поддерживающих функцию MBOC, можно изменить 1024 шага в режиме он-лайн. Эти 1024 шага не обязательно должны находиться во взаимосвязанном блоке. Можно сделать и несколько изменений - различными блоками меньшего размера. Максимальное количество обрабатываемых блоков равно 64. Количество 1024 шагов превышать нельзя (это относится только к процессорам MELSEC System Q, за исключением Q00(J) и Q01)

- **Format**

Эта экранная кнопка запускает процесс форматирования.

- **Defragment**

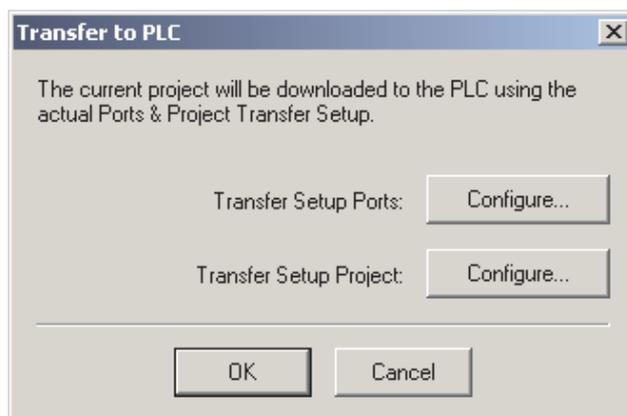
В связи со структурой памяти центральных процессоров MELSEC серий "System Q" и QnA, после частой передачи данных в центральный процессор данные в этих дисководах могут оказаться фрагментированными - аналогично тому, как это происходит с жестким диском компьютера. Для восстановления первоначальной скорости обращения к данным память можно дефрагментировать, нажав экранную кнопку "Defragment".

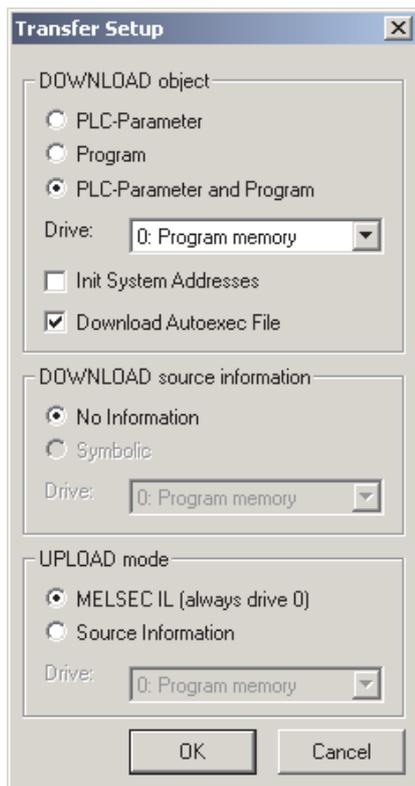
#### 4.3.4 Загрузка проекта

- 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.

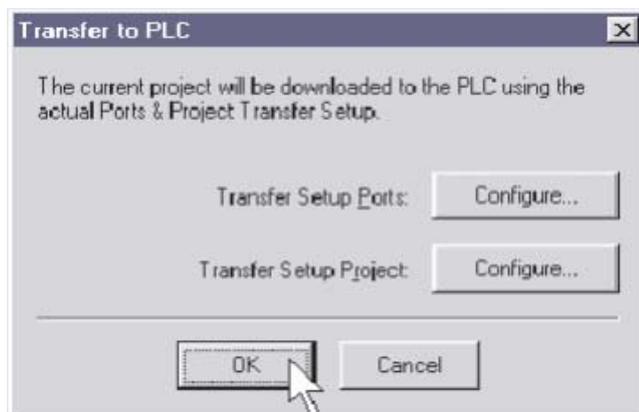




③ Щелкните на **PLC-Parameter and Program**

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

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



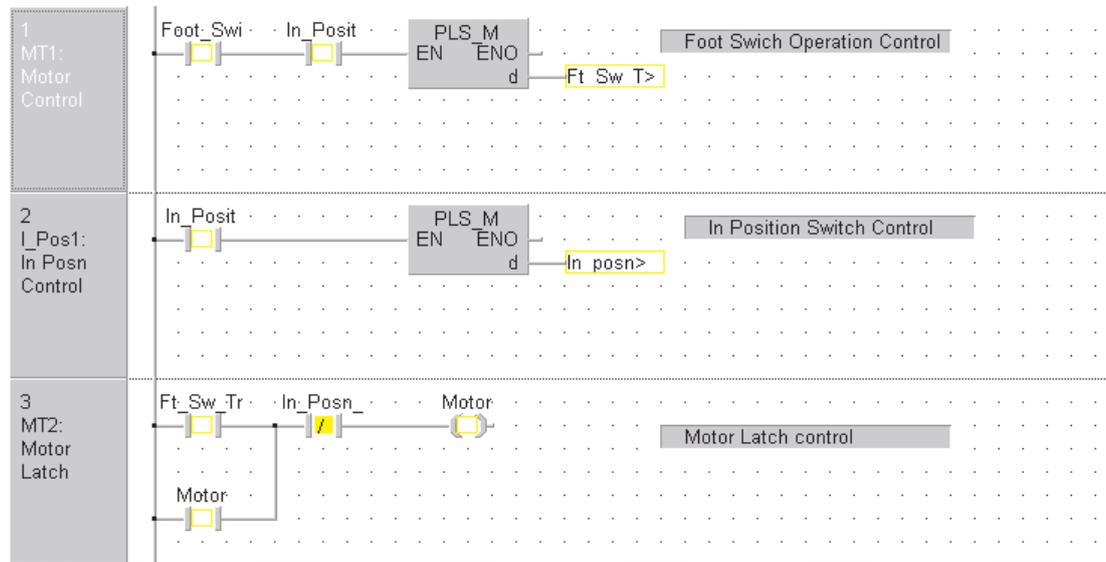
## 4.4 Мониторинг проекта

Тестирование и оптимизация программы, а также поиск ошибок облегчаются, если имеется возможность наблюдения за выполнением программы в контроллере. GX IEC Developer предлагает разнообразные возможности для отображения состояния программы и операндов.

В режиме мониторинга в программе дополнительно отображаются состояния операндов. Контроллер должен быть включен (RUN) и соединен с программатором, и в нем не должно иметься неполадок.

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.

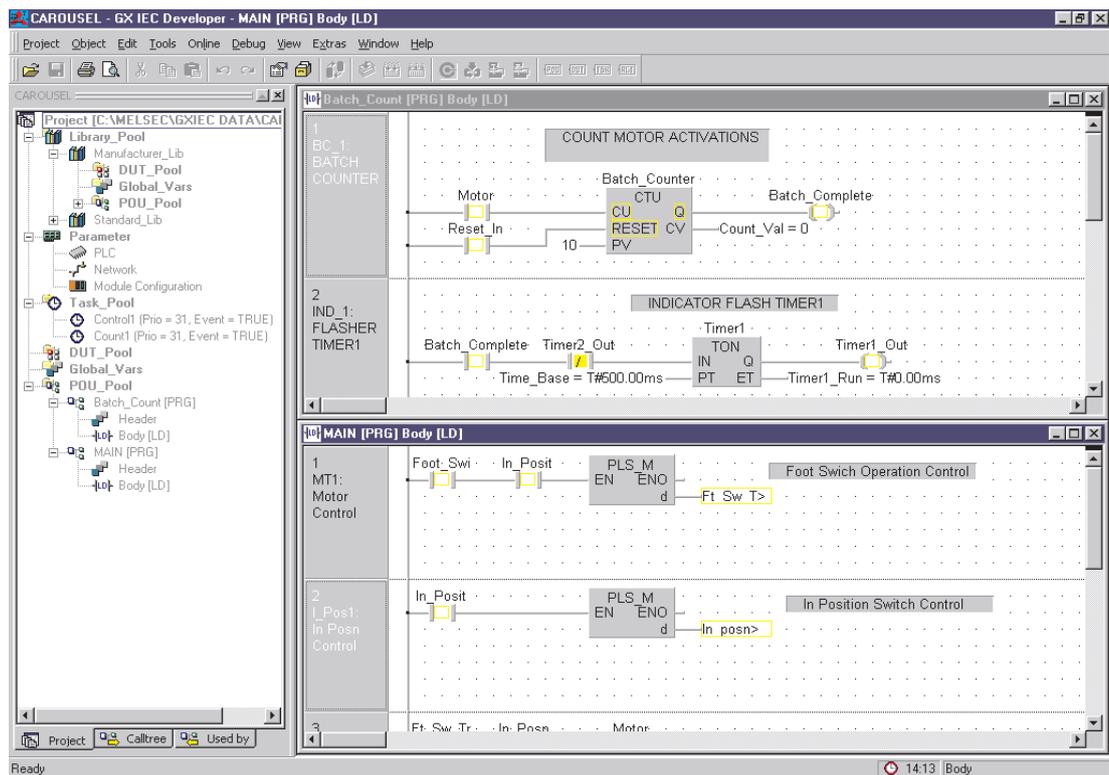
#### 4.4.1 Раздельный / Многооконный мониторинг

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).

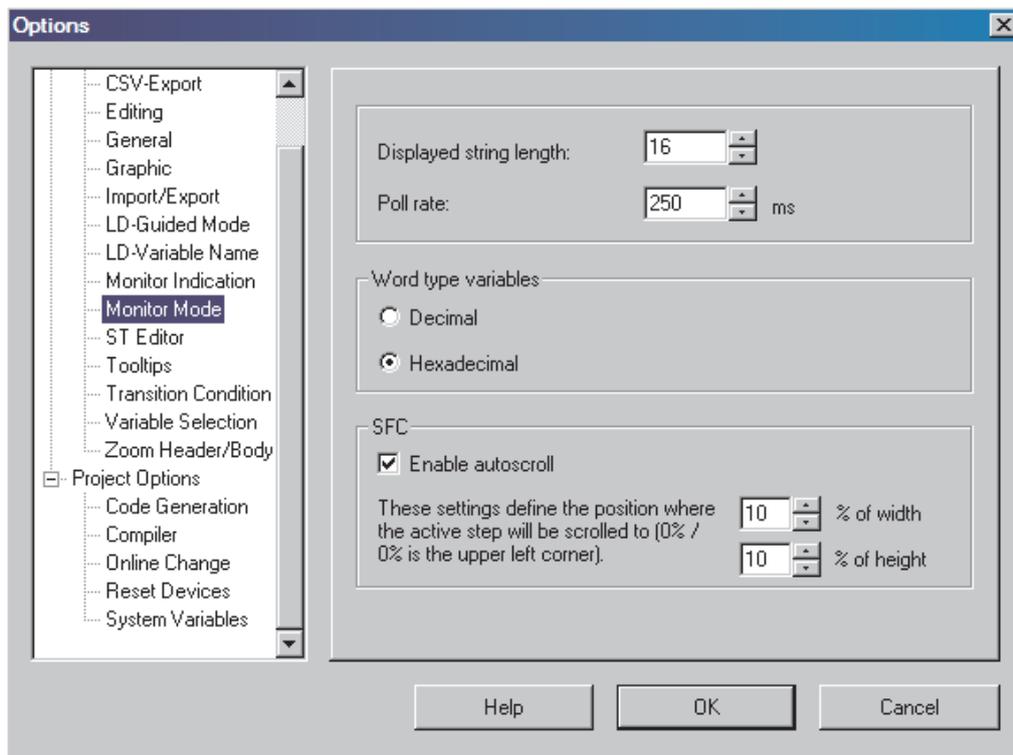
Чтобы начать мониторинг содержания от дополнительных окон, щелкните внутри данного окна и выберите **Start Monitoring** из меню **Online**:



##### ПРИМЕЧАНИЕ

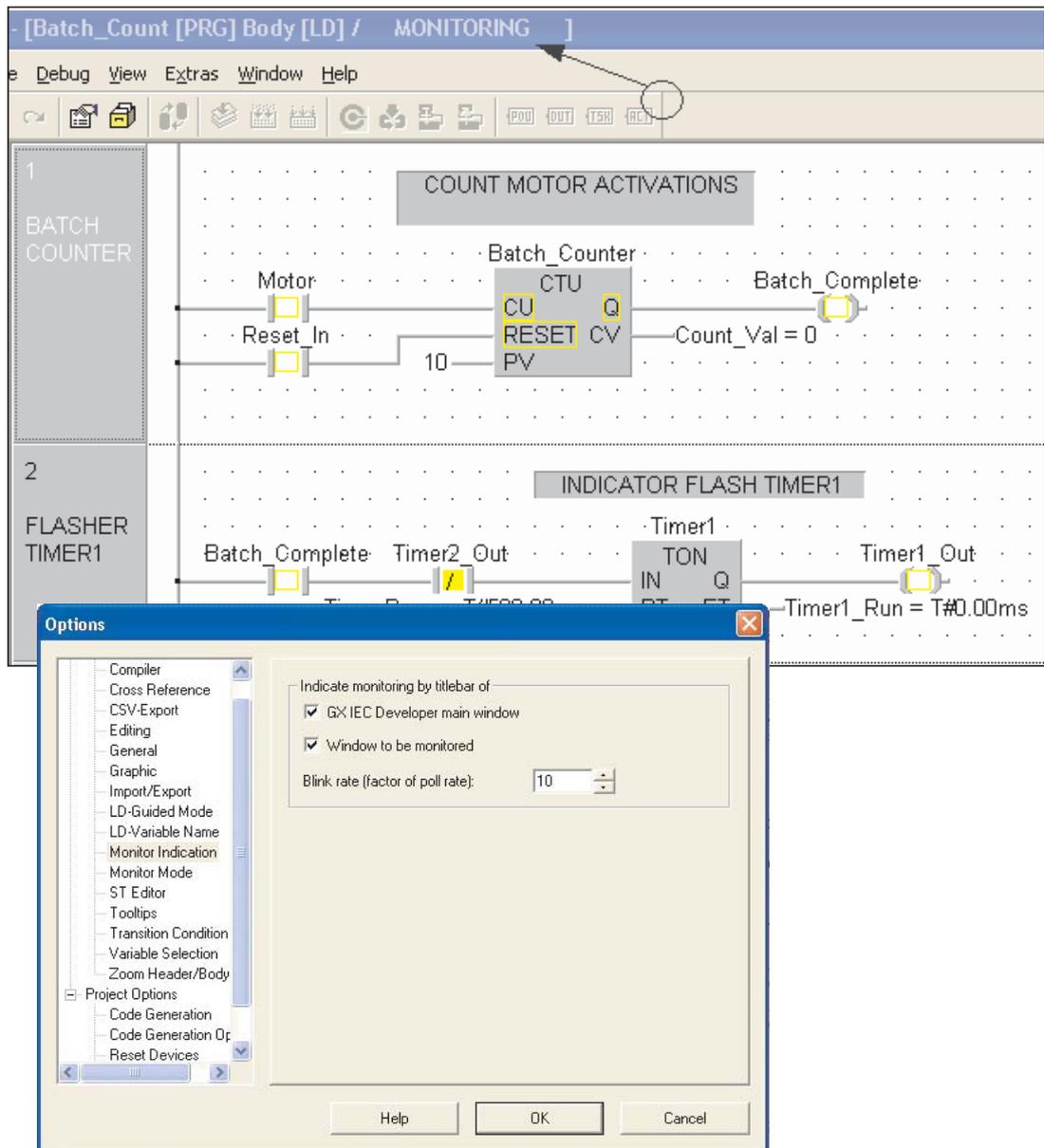
Вследствие кэширования последовательной связи будьте готовы выждать несколько секунд, пока контрольная информация будет зарегистрирована между GX IEC Developer и ПЛК.

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**):



### 4.4.2 Настройка видимости монитора

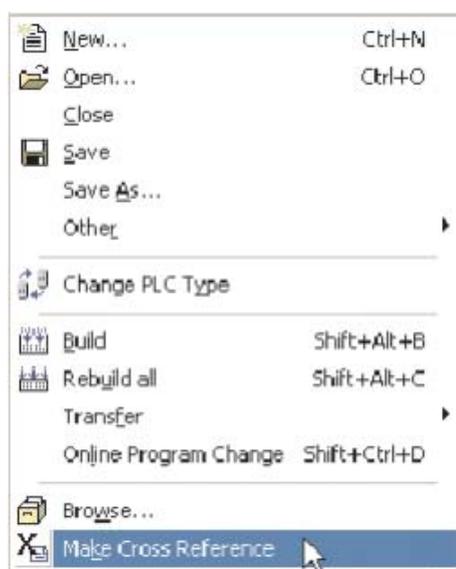
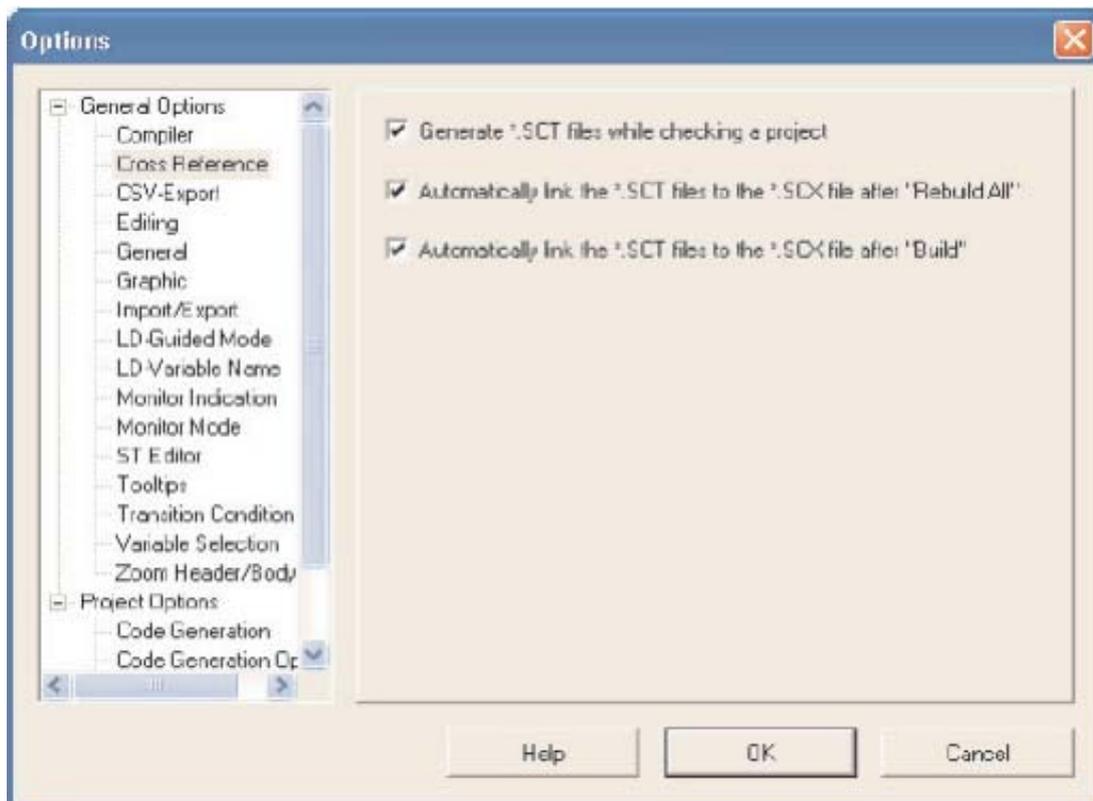
Чтобы регулировать видимость режима мониторинга, выберите "**Extras** → **Options** → **Monitor Indication**", где можно разрешить мигающее сообщение, которое будет показано в выбранном месте. Пользователь может установить наилучшую частоту мигания надписи "Monitoring":



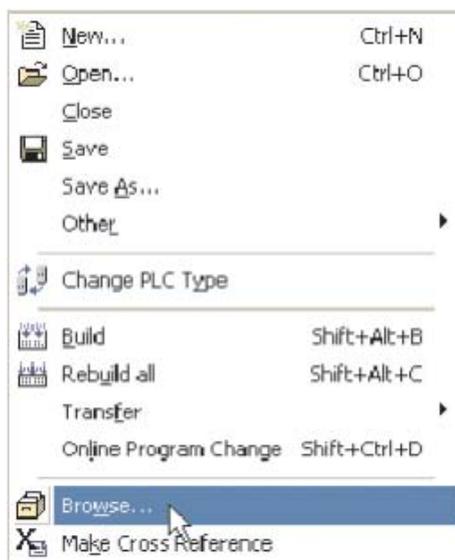
## 4.5 Список перекрестных ссылок

Чтобы генерировать Список перекрестных ссылок:

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

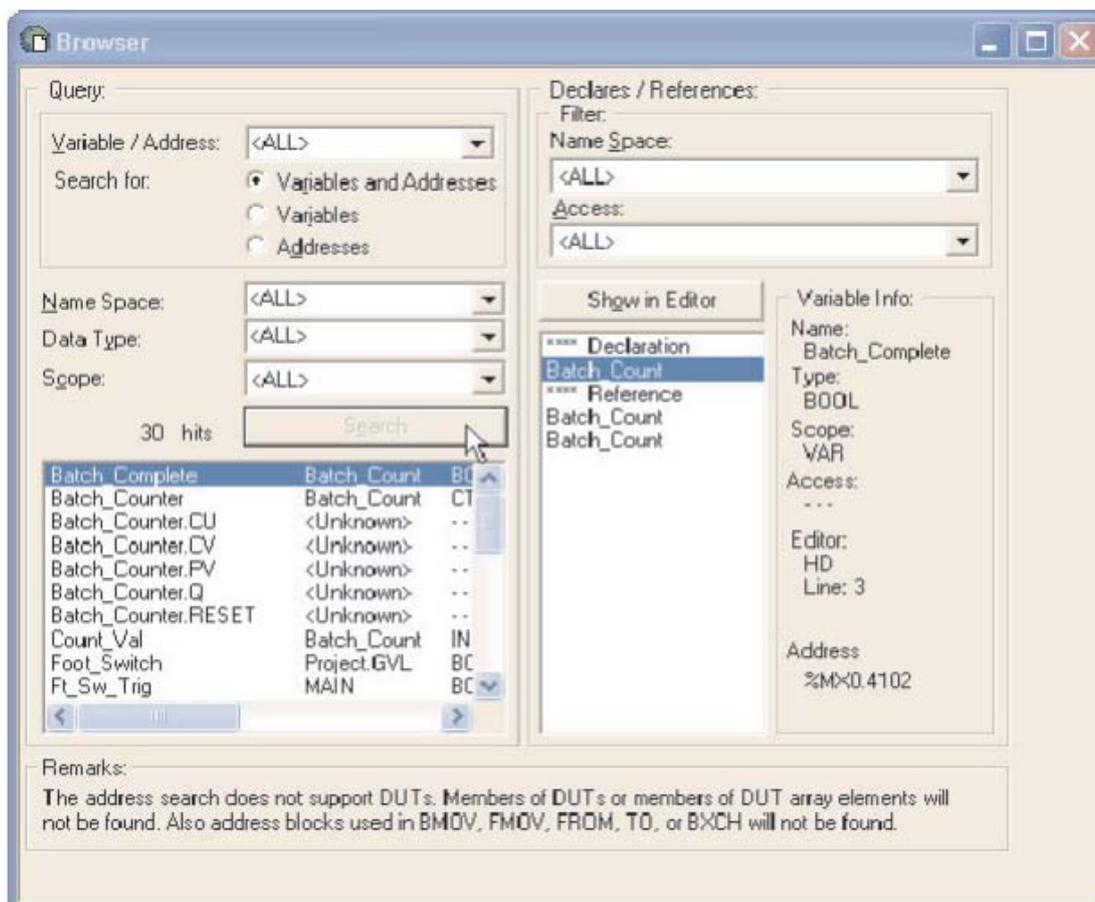


- ③ Затем выберите **Make Cross Reference** из меню **Project**, и генерируется список.



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

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

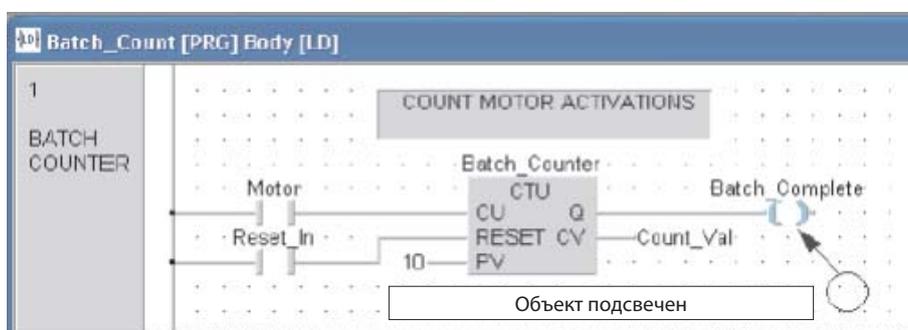


Для поиска конкретных переменных и т.д. можно использовать окна выбора запросов. Затем отдельные детали подсвеченного элемента показываются в правой стороне окна.

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



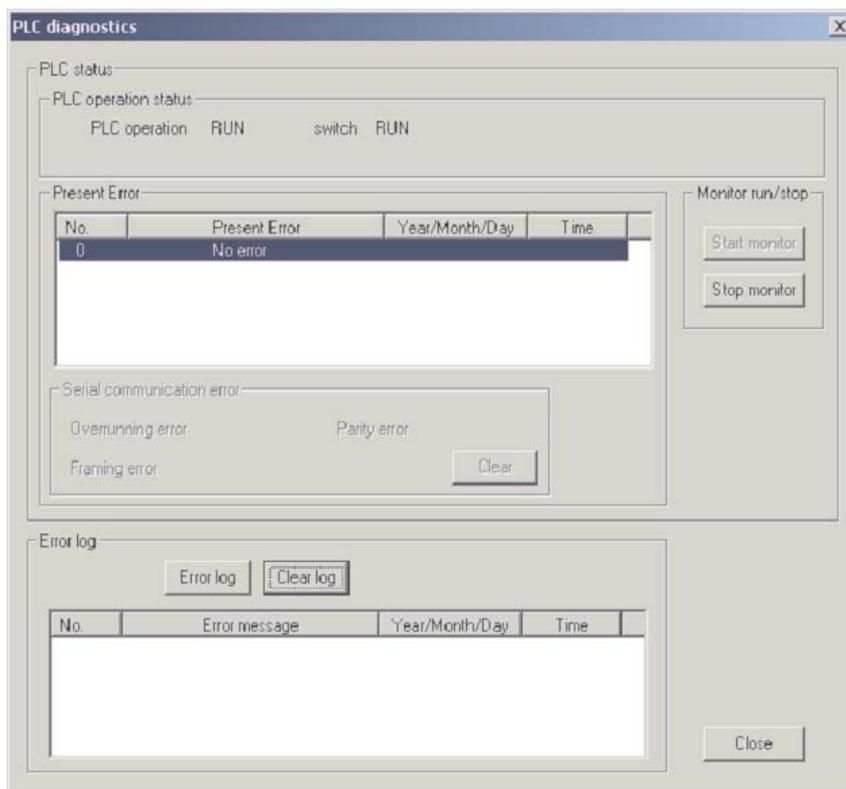
Список перекрестных ссылок можно распечатать, используя возможность печати в GX IEC Developer.

## 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.

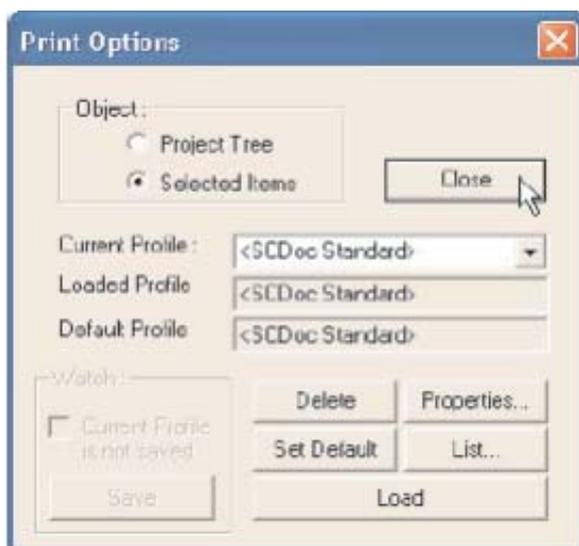
Можно определять пользовательские ошибки. Эти пользовательские ошибки хранятся с самостоятельно созданным текстовым файлом (USER\_ERR.TXT) и позволяют быстро исправлять ошибки. Последние восемь пользовательских ошибок хранятся в FIFO регистре; они удаляются, только когда больше не возникают.

## 4.7 Документация проекта

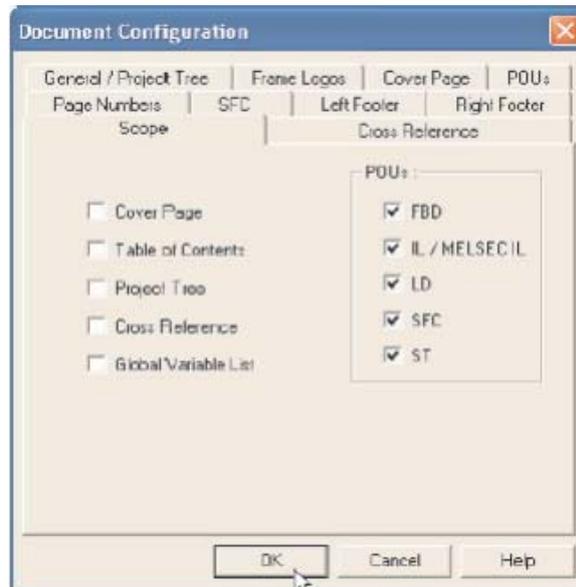
Документацию проекта можно составить, используя пункт **Print Option** из меню **Project**:



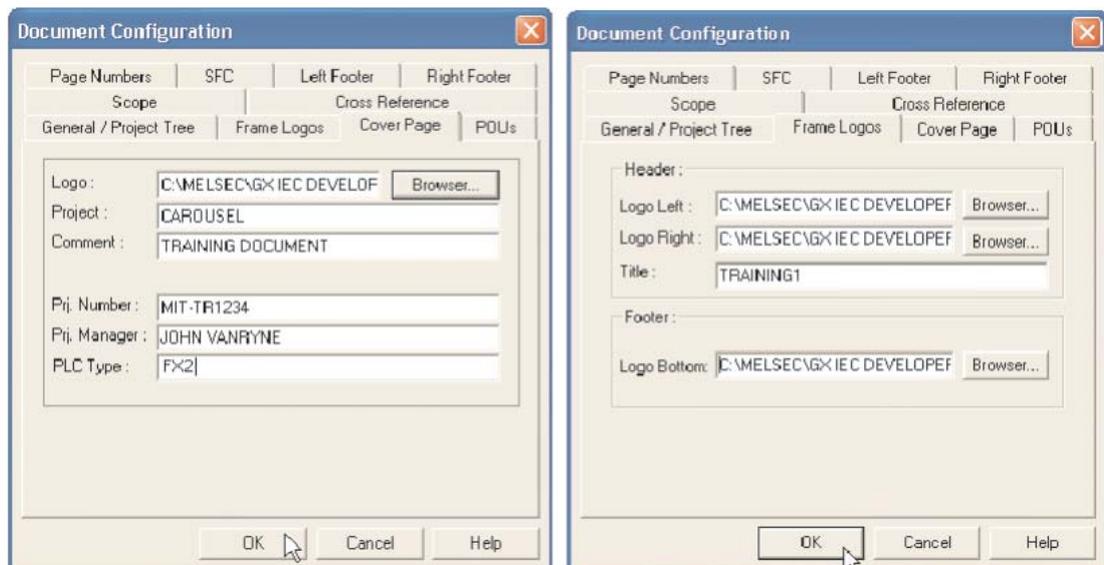
Появится диалоговое окно изменения конфигурации "**Change Configuration**". Здесь можно восстановить предыдущие профили проекта или работать с профилем по умолчанию. Выберите **Project Tree** для всех элементов, или **Selected Items** для конкретных подсвеченных пунктов, и откройте **Properties**:



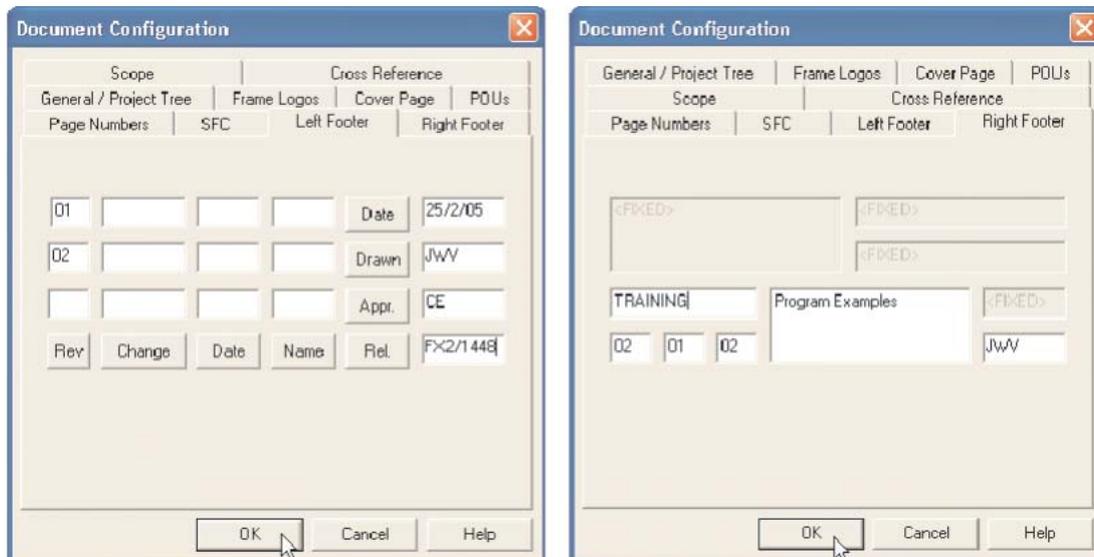
Ниже показано окно **Document Configuration**. Выберите вкладки, чтобы конфигурировать документ согласно требованиям. В этом примере будет напечатан только COUNTER\_FB\_CE, согласно выбранной опции **Selected Items**:



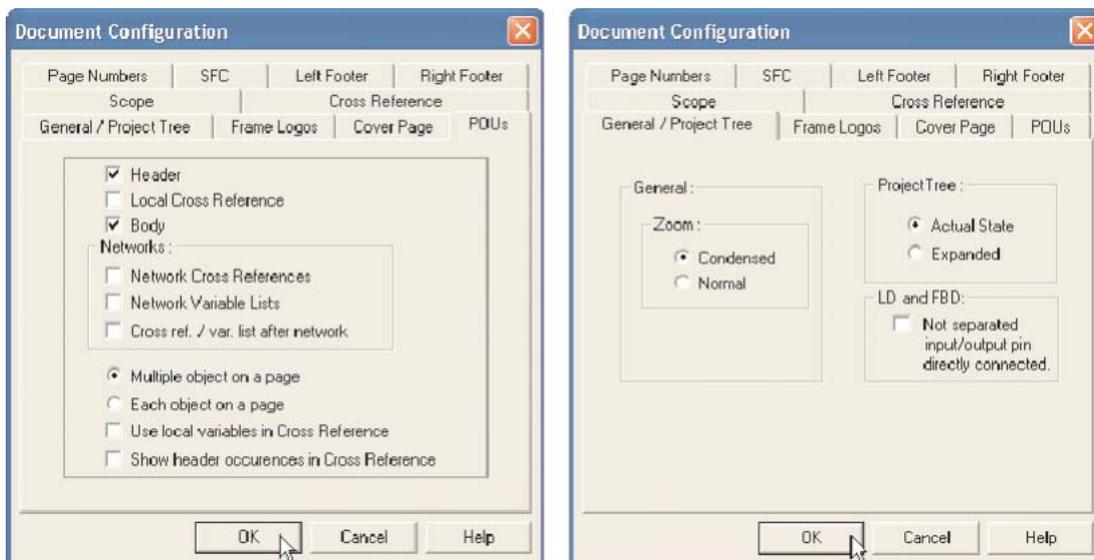
На вкладке **Cover Page** определенные пользователем логотипы и информация могут быть заданы для титульного листа и для кадра на вкладке **Frame Logos**:



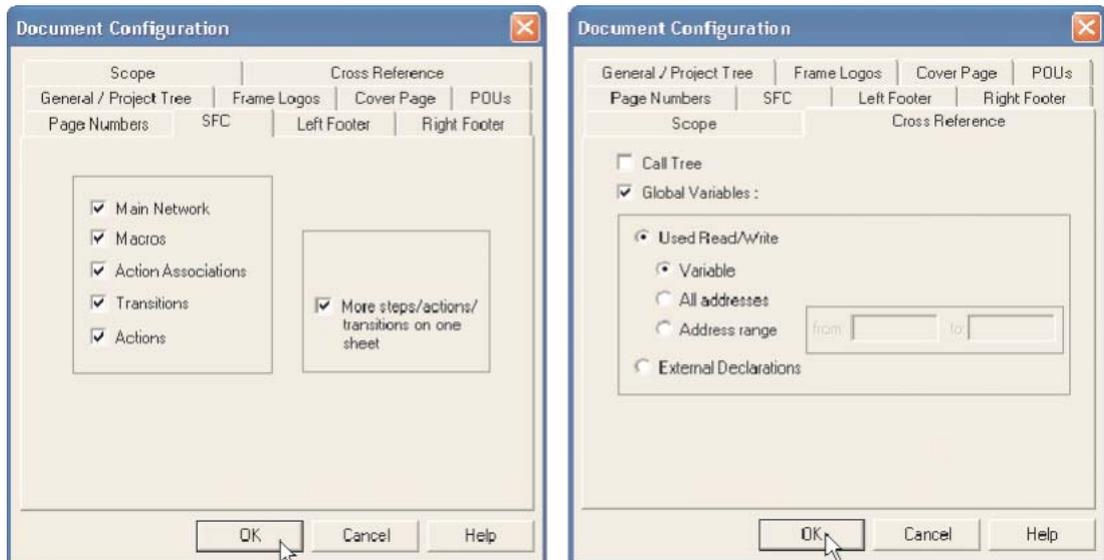
Подробную информация можно присвоить левым и правым нижним сноскам. При необходимости можно переименовать содержимое полей в диалоговом окне левого нижнего колонтитула **Left Footer**, щелкнув на кнопках имен:



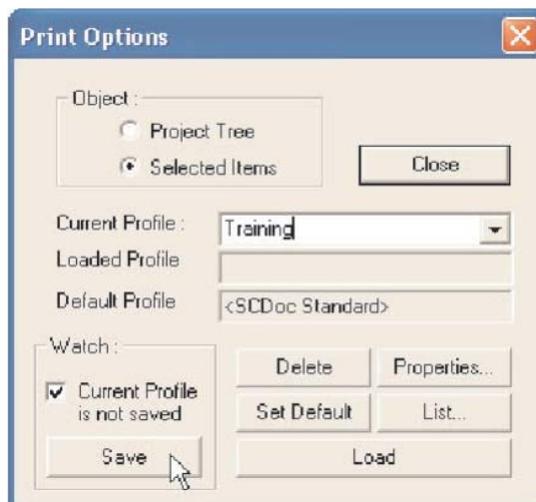
Спецификация вида программного модуля и общие спецификации проекта имеются на вкладках **POUs** и **General/Project Tree**.



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 M.
- 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

X10	-	Кнопка ответа игрока 1
X11	-	Кнопка ответа игрока 2
X12	-	Кнопка ответа игрока 3
X13	-	Кнопка ответа игрока 4
X14	-	Запуск отсчета ведущим
X15	-	Сброс игры

##### – Выходы

Y20	-	Лампа ответа игрока 1
Y21	-	Лампа ответа игрока 2
Y22	-	Лампа ответа игрока 3
Y23	-	Лампа ответа игрока 4
Y24	-	Индикация "Время истекло"

#### Специальный модуль

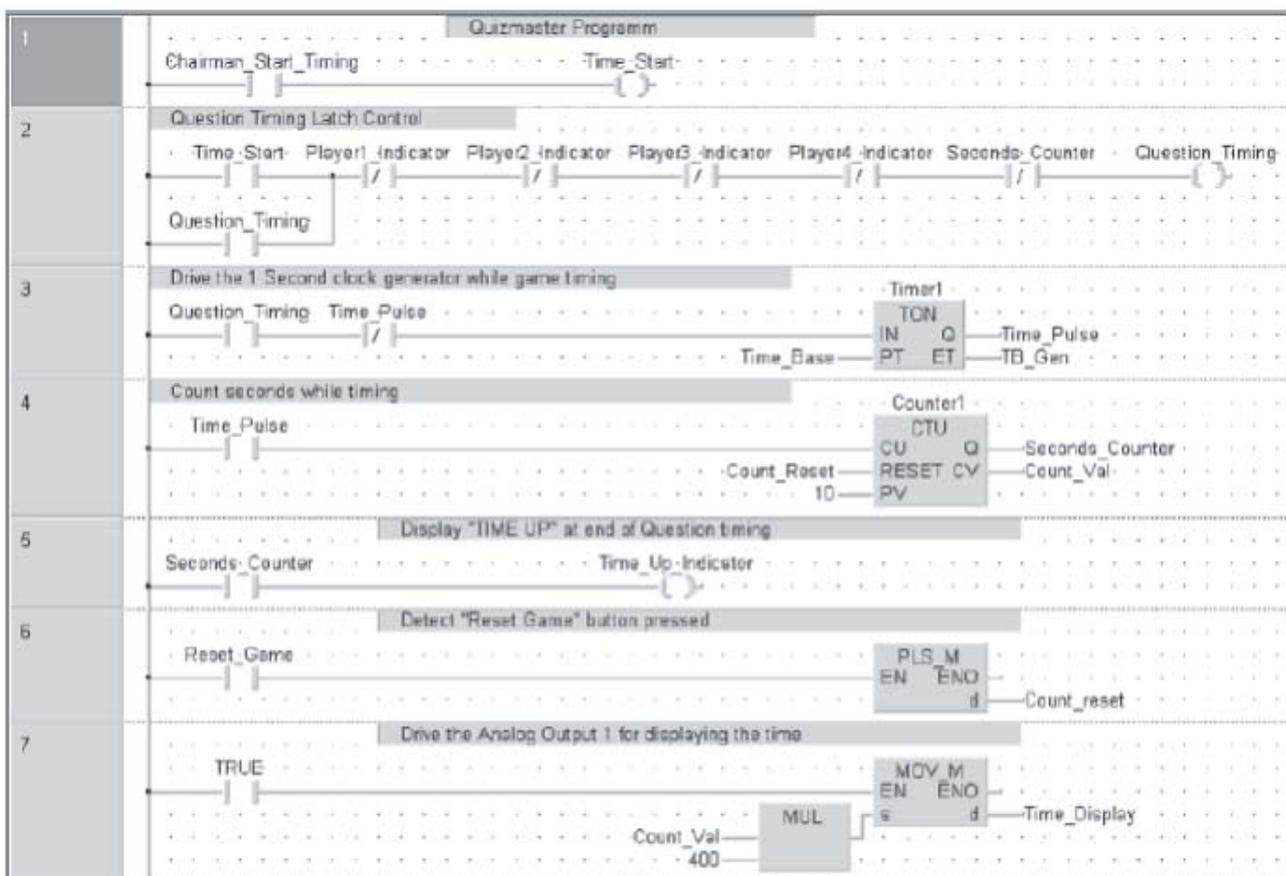
U4\G1	-	Адрес для аналогового выхода 1 в буферной памяти аналогового выходного модуля
-------	---	---

### 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	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	Player1_Response	X10	%IX16	BOOL	FALSE
1	VAR_GLOBAL	Player2_Response	X11	%IX17	BOOL	FALSE
2	VAR_GLOBAL	Player3_Response	X12	%IX18	BOOL	FALSE
3	VAR_GLOBAL	Player4_Response	X13	%IX19	BOOL	FALSE
4	VAR_GLOBAL	Chairman_Start_Timing	X14	%IX20	BOOL	FALSE
5	VAR_GLOBAL	Reset_Game	X15	%IX21	BOOL	FALSE
6	VAR_GLOBAL	Player1_Indicator	Y20	%QX32	BOOL	FALSE
7	VAR_GLOBAL	Player2_Indicator	Y21	%QX33	BOOL	FALSE
8	VAR_GLOBAL	Player3_Indicator	Y22	%QX34	BOOL	FALSE
9	VAR_GLOBAL	Player4_Indicator	Y23	%QX35	BOOL	FALSE
10	VAR_GLOBAL	Question_Timing	Y24	%QX36	BOOL	FALSE
11	VAR_GLOBAL	Time_Display	U4\G1	%MW14.4.1	INT	0
12	VAR_GLOBAL	Time_Up_Indicator			BOOL	FALSE

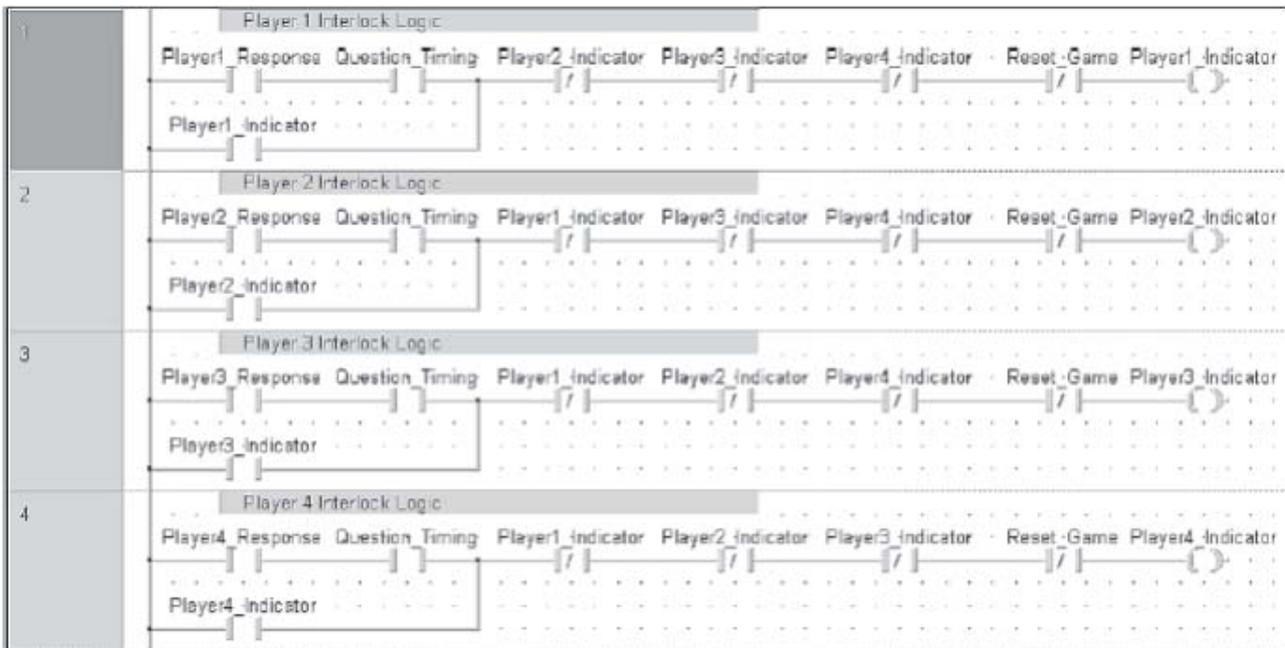
- ③ Создайте новый программный модуль класса **PRG** (Program type) на языке **Ladder Diagram** и назовите его "Game\_Control".
- ④ Введите следующий код в программный модуль.



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	

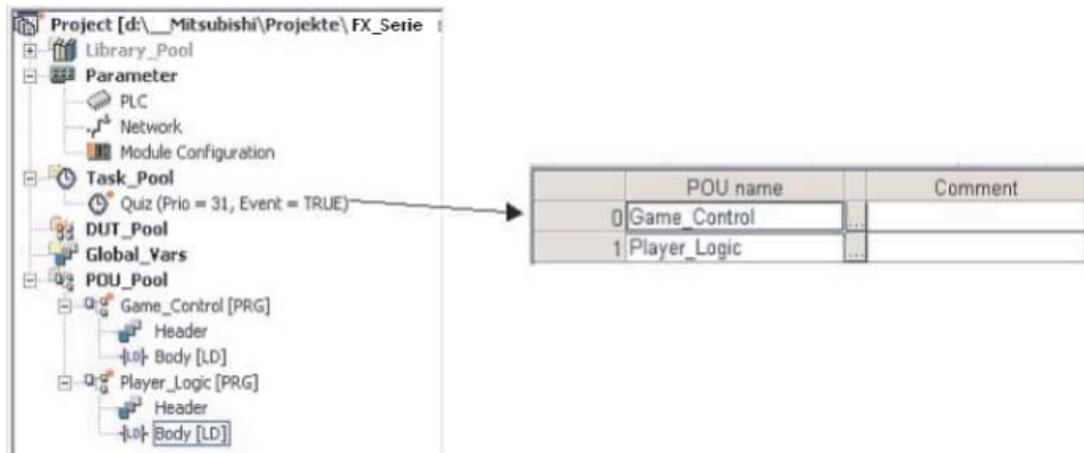
- ⑤ Создайте новый программный модуль класса **PRG** и типа **Ladder** и назовите его "Player\_Logic"
- ⑥ Введите следующий код релейной диаграммы в новый программный модуль:



Окончательный заголовок программного модуля "Player\_Logic" должен иметь следующий вид:

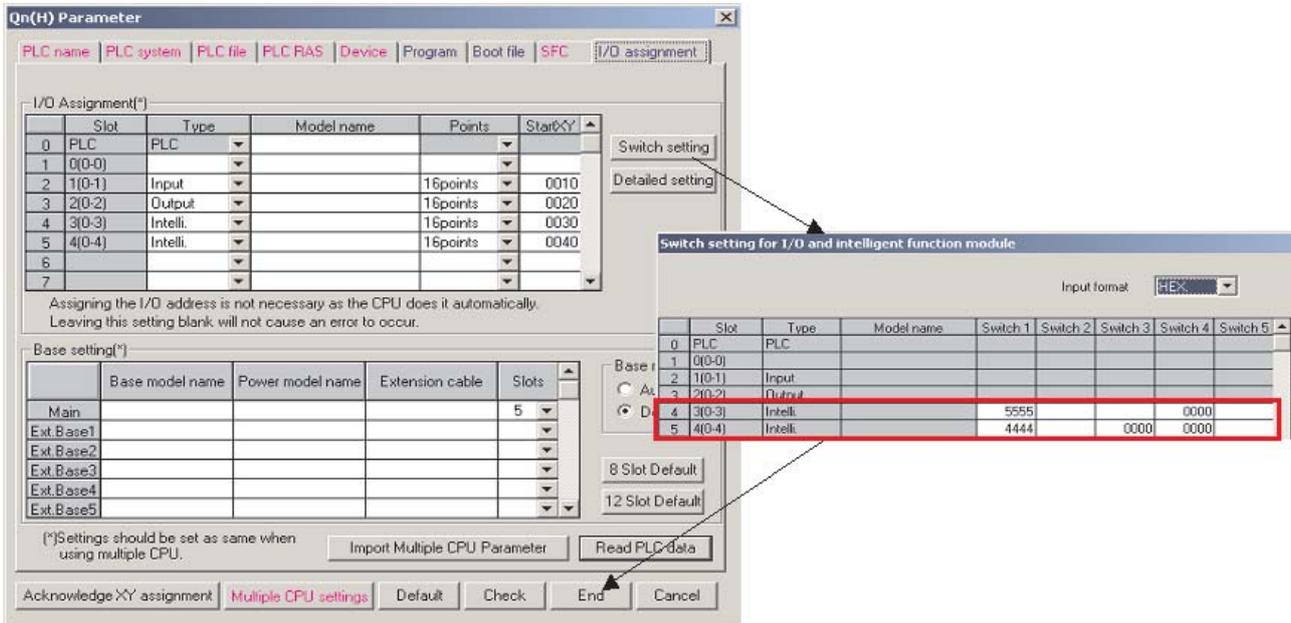
	Class	Identifier	Type	Initial	Comment
0	VAR				

- ⑦ Создайте новую задачу в задачном пуле "QUIZ". Свяжите ПМ "Player\_Logic" и "Game\_Control" соответственно с новой задачей, как показано ниже:

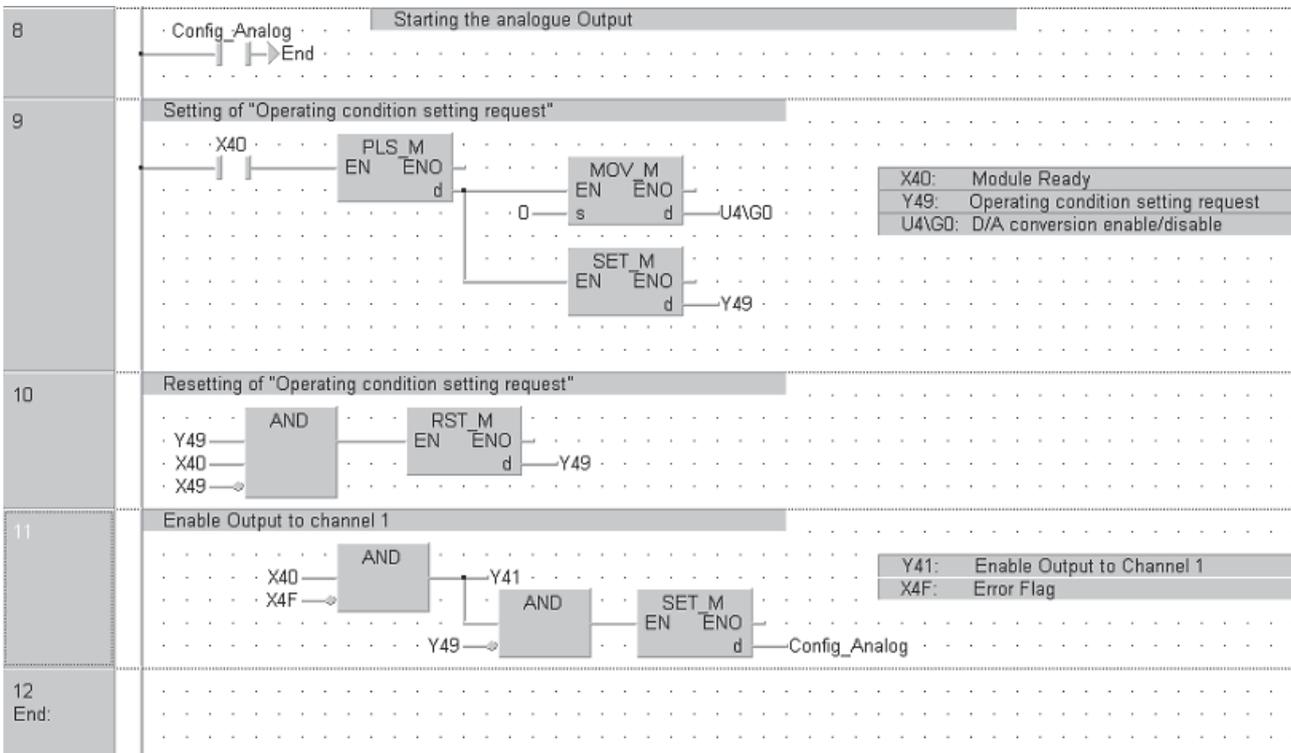


**Инициализация аналогового выходного модуля**

- 8) Затем необходимо сделать настройки для аналогового выходного модуля Q64DA. Для этого щелкните в окне навигатора двойным щелчком по "Parameter", а затем по подкаталогу "PLC". Затем щелкните по закладке "I/O assignment" и "Switch setting" и введите показанные внизу значения.



- 9) 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 MELSEC System Q.
- ③ Убедитесь, что проект работает правильно, контролируя работу при активизации входов.
- ④ На мгновение включите вход X14, чтобы начать хронометраж реакции на ответ соперника.
- ⑤ Ждите начальной реакции соперников из X10, X11, X12 или X13 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" X15, чтобы очистить все флаги состояния и выходы и начать новый раунд.

### 5.1.3 Описание программы Quizmaster

#### Программный модуль "Game\_Control"

- Звено 1  
Когда ведущий нажимает кнопку начала отсчета Start Timing , локальная переменная "Time\_Start" мигает благодаря инструкции PLS\_M.
- Звено 2  
Переменная Question\_Timing зафиксирована для того, чтобы ни один из индикаторов игроков не был включен и счетчик секунд не был активирован.
- 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

## 6.1 Функции

- Функции являются частью системы команд.
- Функции - это подпрограммы, многократно вызываемые в программе в виде команды.
- 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.

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.

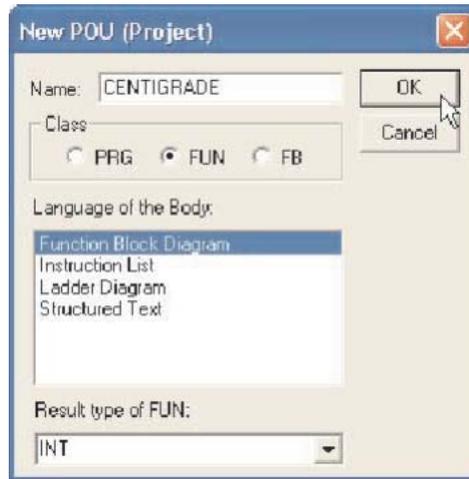
Используется формула:

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

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

**Процедура**

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

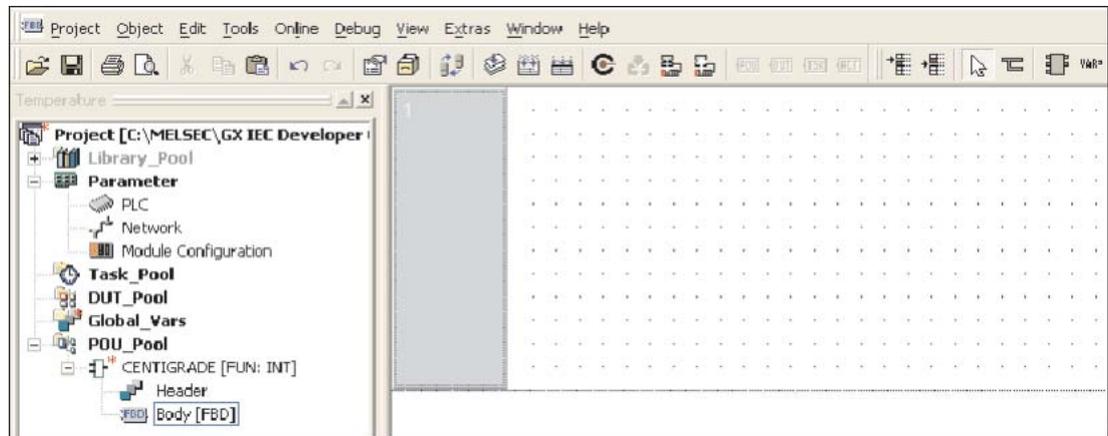


На этот раз щелкните на опции "FUN", а не "PRG". Выберите в качестве редактора **Function Block Diagram**. Тип результата функции "Result Type of FUN " следует оставить как **INT** (целочисленный тип).

Теперь "CENTIGRADE" появился на дереве программных модулей:

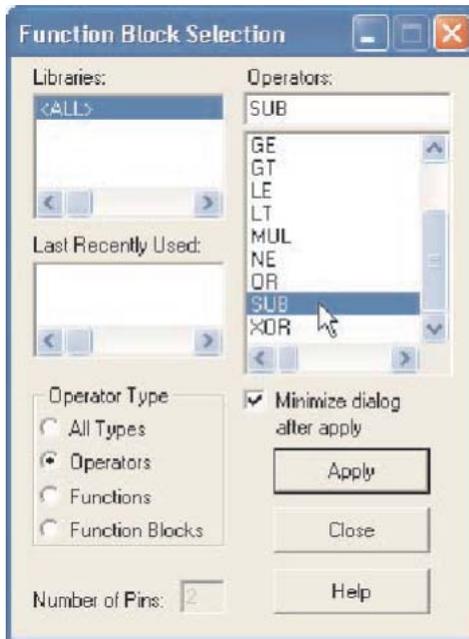


- 2 Дважды щелкните на значке тела FBD, чтобы открыть звено тела:

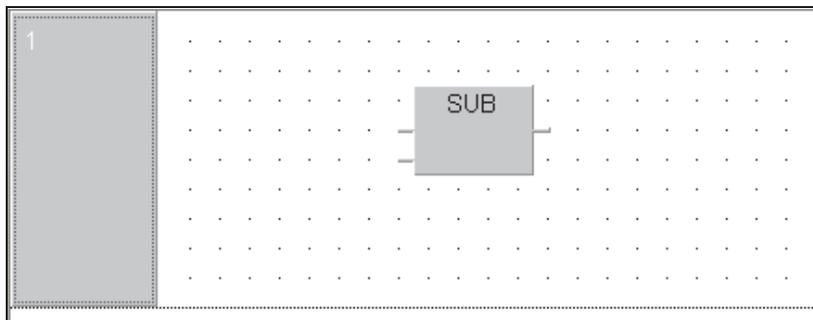


**Выбор функции:**

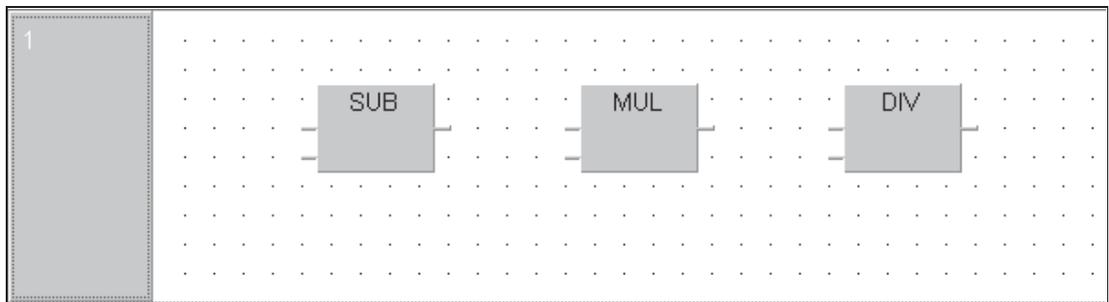
- 1 Выберите значок функционального блока  из панели инструментов и **SUB** из списка операторов:



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



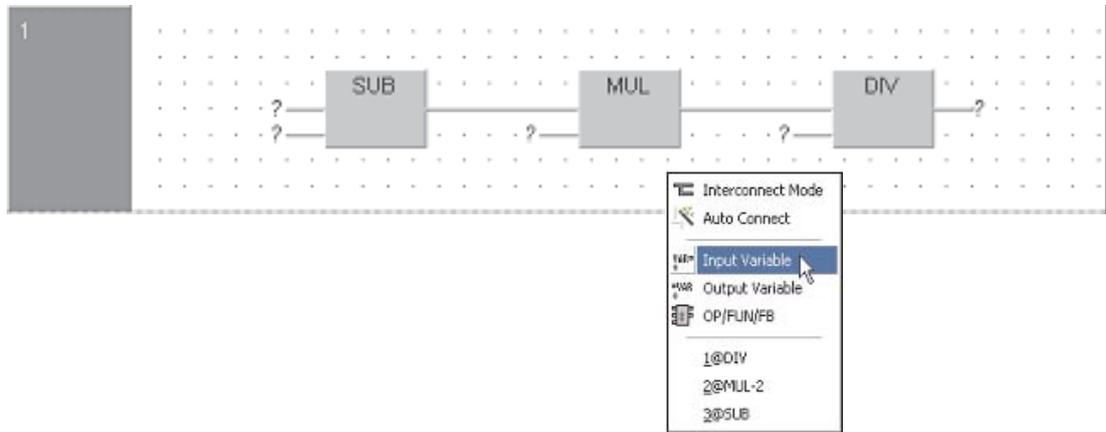
- 3 Повторите описанный выше процесс, чтобы экран принял следующий вид:



### Объявление переменных

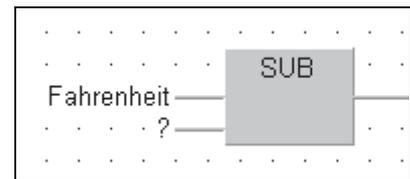
Можно использовать различные методы объявления переменных. Следующая процедура иллюстрирует, как объявлять переменные из тела FBD:

- 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:

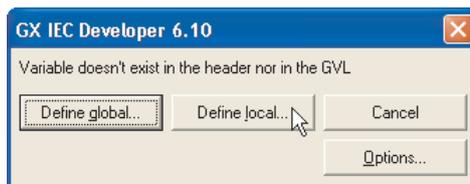


Альтернативно, щелкните на кнопке панели

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

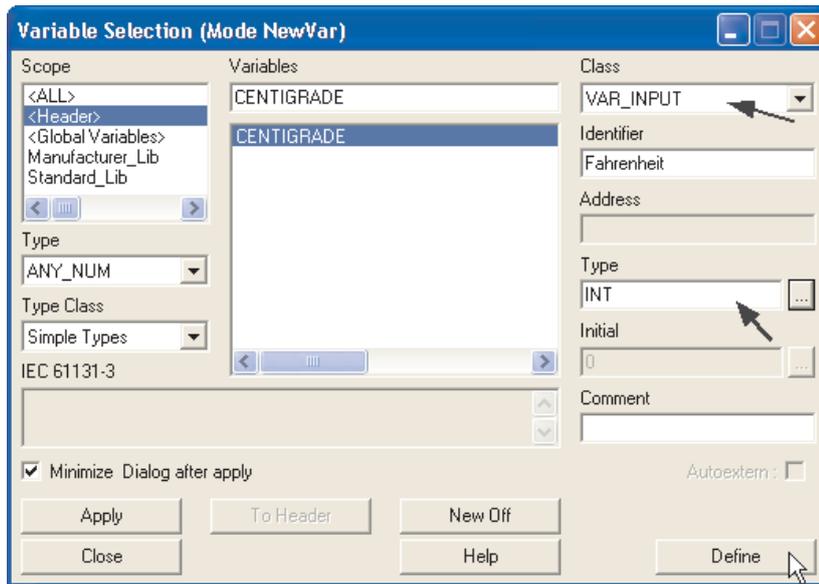


Так как название этой переменной еще не было введено в заголовке (перечне локальных переменных), появляется сообщение. Теперь вы можете определить эту переменную как глобальную или локальную.



- 3 Щелкните по **Define Local**.

- ④ После этого открывается диалоговое окно для объявления переменной. Введите в поле **Class** класс VAR\_INPUT", а в поле тип **Type** - тип "INT".



**ПРИМЕЧАНИЯ**

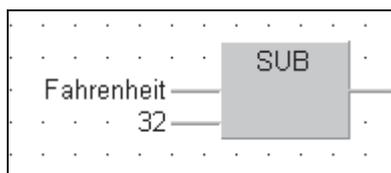
Требуется класс VAR\_INPUT, поскольку эта переменная позволяет вводить значения в функцию, когда она подключена как часть программы. Это показывается точкой соединения со входом слева на символе функции.

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".

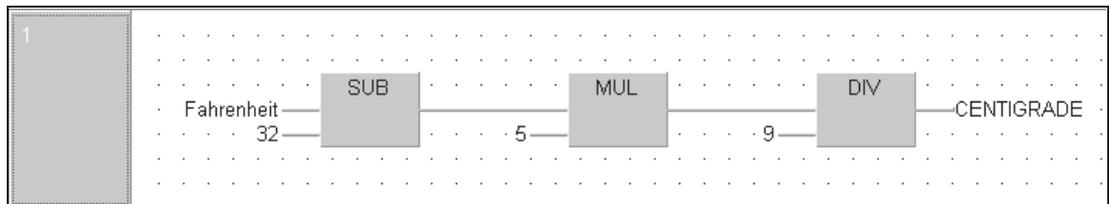
- ⑤ Щелкните на **Define**, и переменная будет записана в заголовок функции "CENTIGRADE". Вы можете проверить это, открыв заголовок.

**Объявление констант**

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

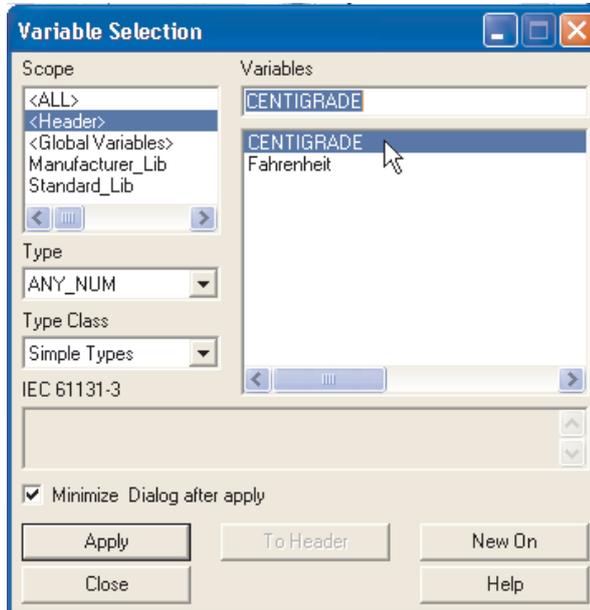


- ② Полная диаграмма функции "CENTIGRADE" имеет следующий вид:



**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:

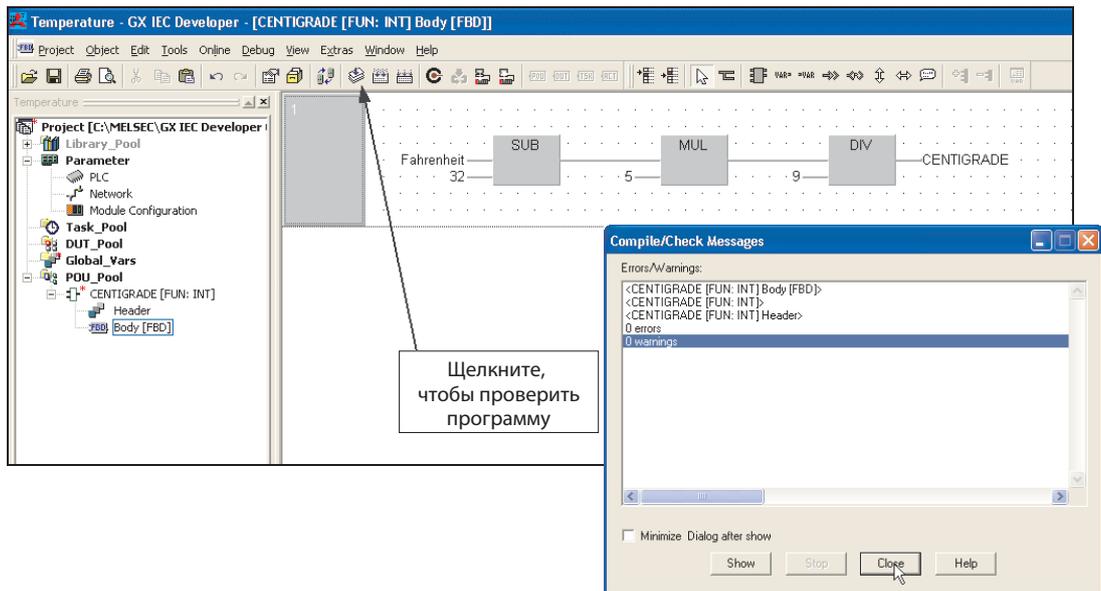
Class	Identifier	Type	Initial	Comment
VAR_INPUT	Fahrenheit	INT	0	

**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

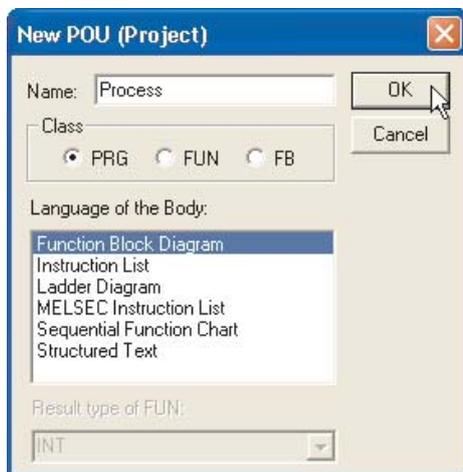
- 1 Проверьте звено; не должно быть никаких ошибок и никаких предупреждений!



- 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)**:

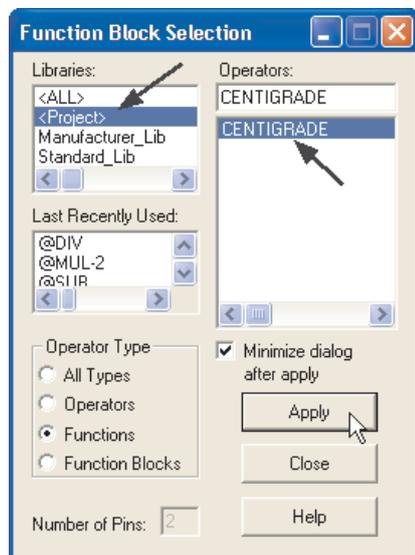


- 2 Откройте (двойным щелчком) тело релейного ПМ "**Process**" в пуле программных модулей проекта.



### Размещение функции пользователя

- 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:

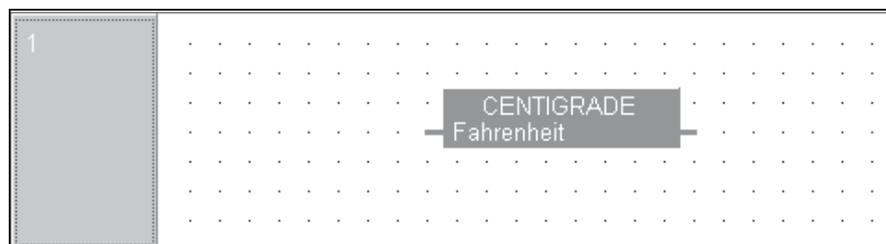


- Выберите CENTIGRADE и щелкните на **Apply**.

**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:



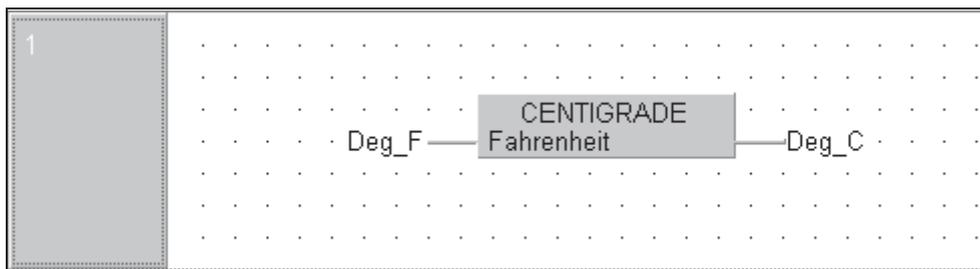
### Назначение глобальных переменных

Когда функция помещена на новое звено, присвойте ей переменные.

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

Global Variable List						
	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

Тело программного модуля "Process" должно иметь следующий вид:



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

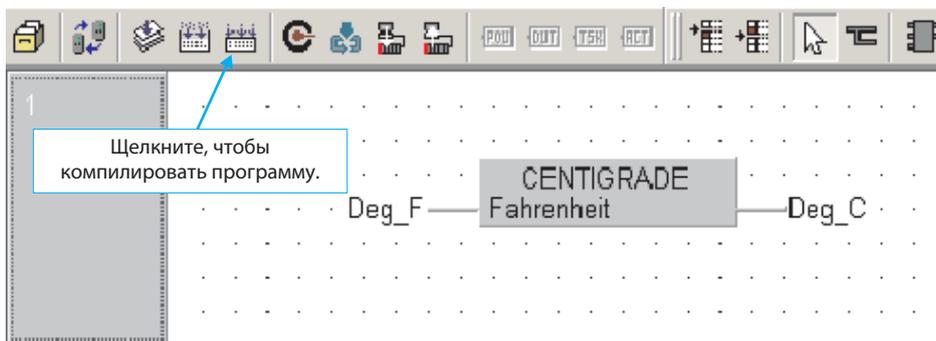


- ③ Свяжите программный модуль "Process" с задачей "Main":

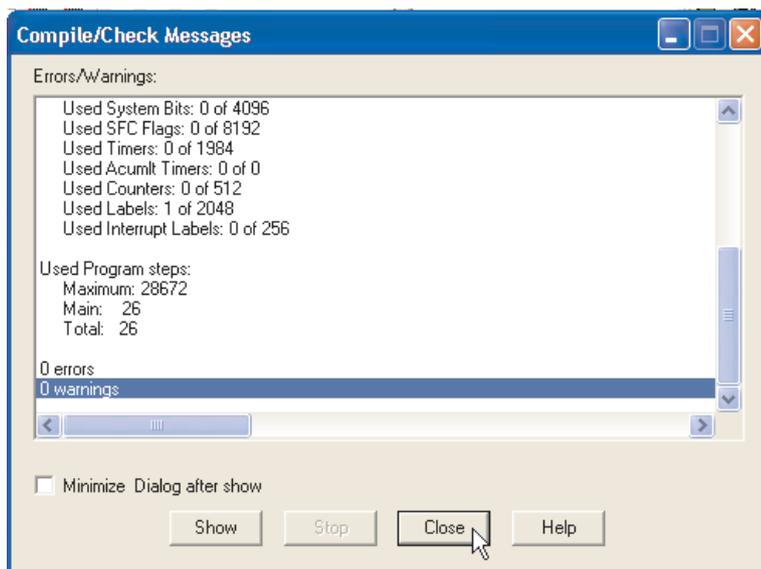
	POU name	Comment
□	Process	

### Компилирование программы

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



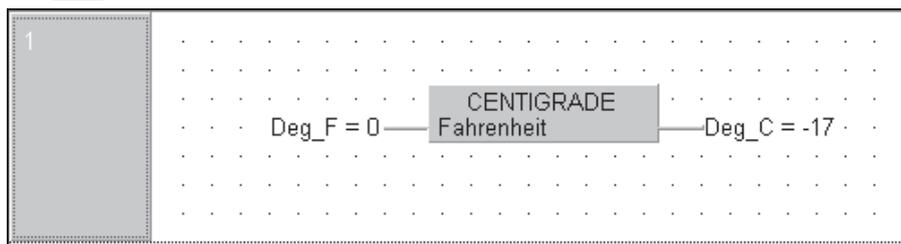
После компиляции должно быть показано следующее окно:



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

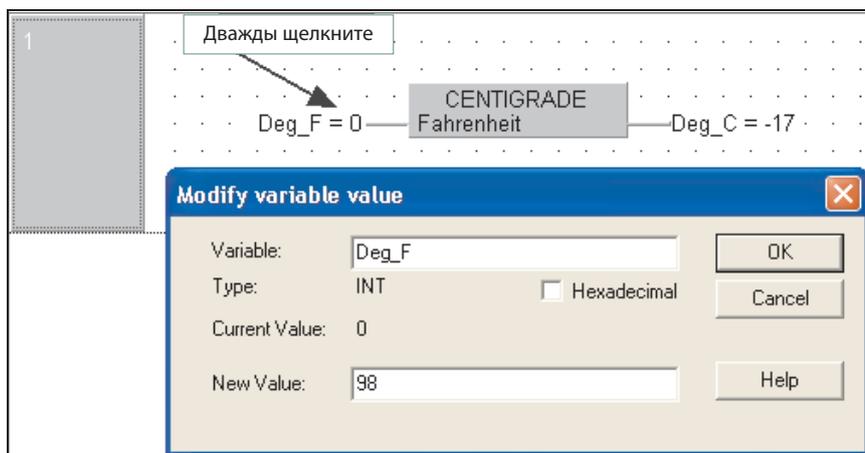
### Мониторинг программы

- 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)

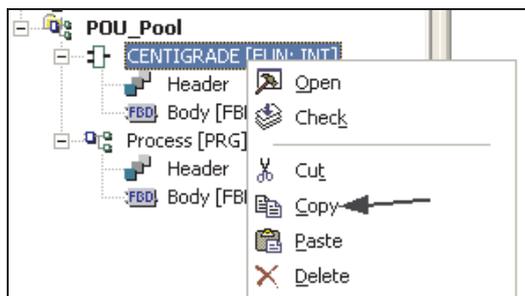
Существующая функция "CENTIGRADE" сейчас может обрабатывать только 16-битовые целочисленные значения (от +32767 до -32768); этот формат используется по умолчанию при создании функций. В следующем примере будет использована функция "CENTIGRADE", модифицированная для обработки значений формата "REAL" с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

\* Допустимо только на процессорах, поддерживающих эту возможность.

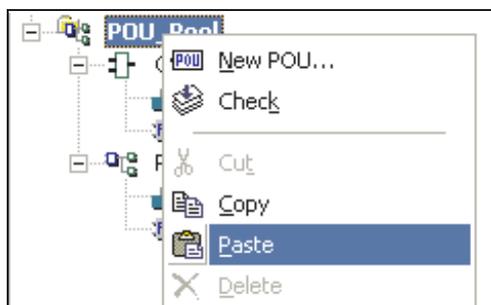
#### Дублирование функции

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

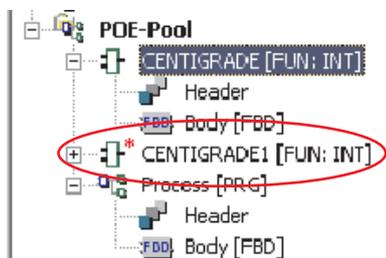
- Щелкните правой кнопкой на значке "CENTIGRADE" в пуле программных модулей проекта и выберите **Copy**.



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

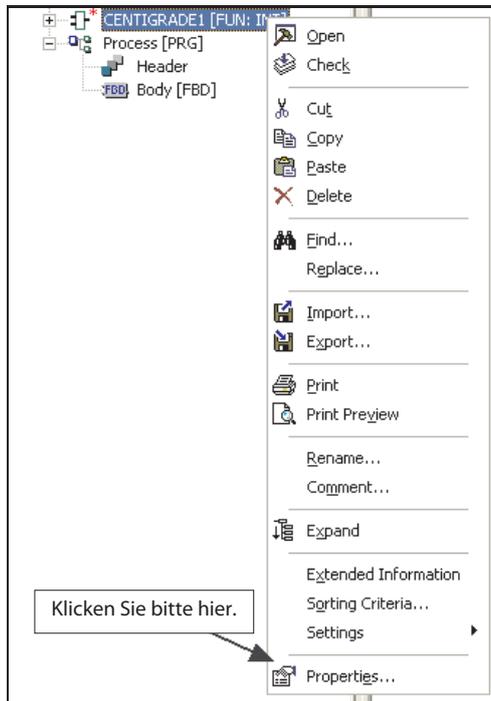


Система автоматически вставит дубликат "CENTIGRADE" и переименует его в "CENTIGRADE1":

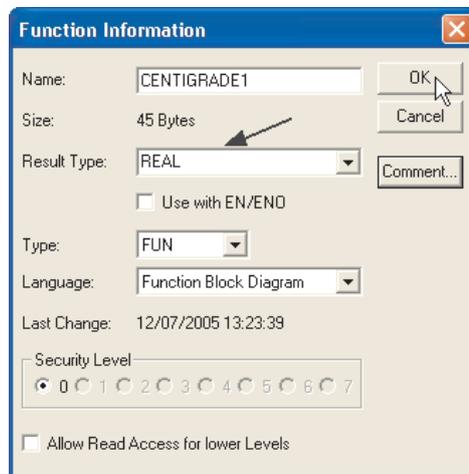


### Изменение типа результата функции

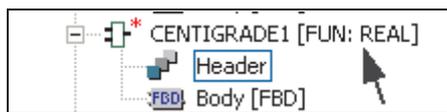
- Щелкните правой кнопкой на только что созданной функции "CENTIGRADE1" и щелкните на **Properties**.



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



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



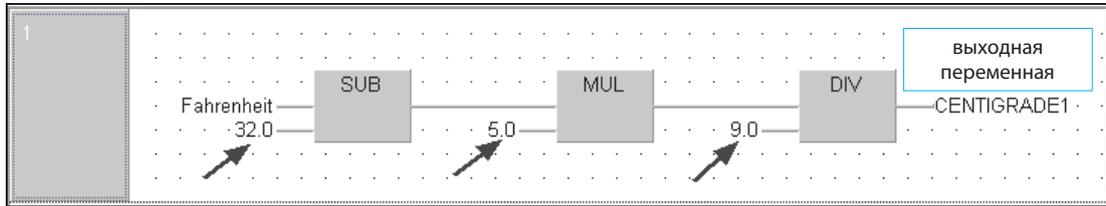
- Измените заголовок "CENTIGRADE1", чтобы переменная Fahrenheit имела тип "REAL":

	Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	Fahrenheit	REAL	0.0	

**Изменение констант к типу "REAL"**

- ① 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.



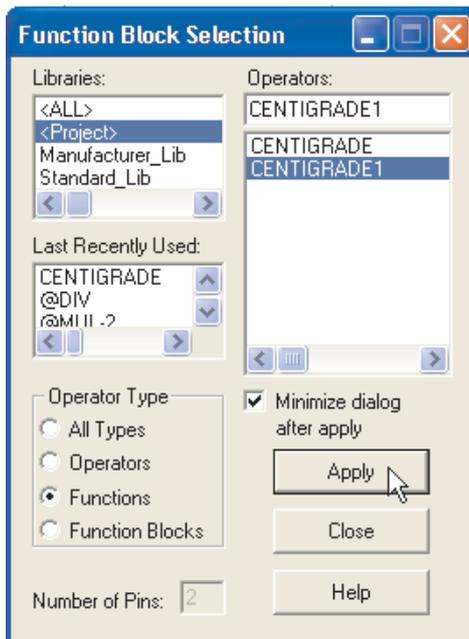
- ② Закройте редакторы и сохраните все изменения.

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

- ① 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

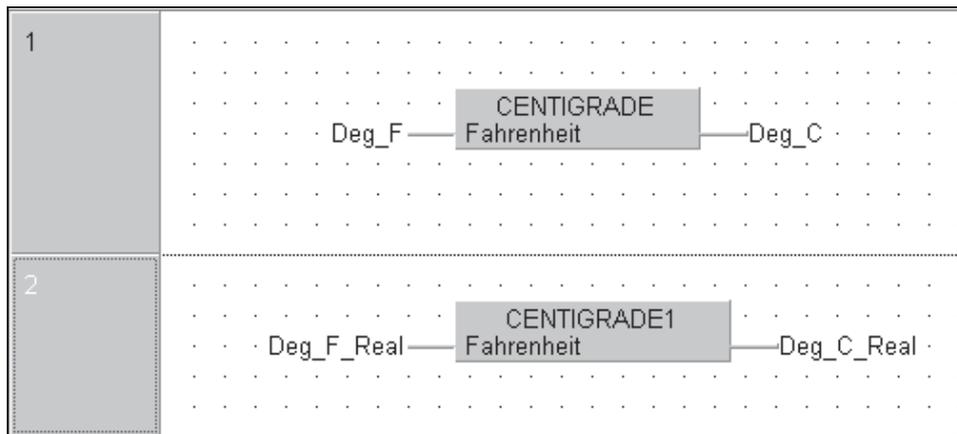
- ② Откройте тело программного модуля "Process" и поместите в него функцию "CENTIGRADE", как показано ниже:



**ПРИМЕЧАНИЕ**

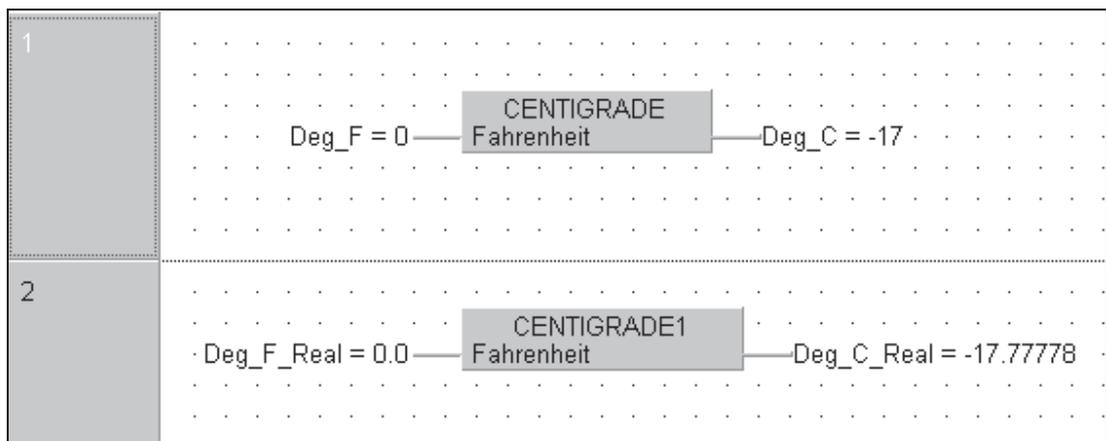
Числа формата REAL используют 2 последовательных регистра (32 бита) и хранятся в специальном портативном IEE-формате, см. назначение в приведенном выше примере GVL.

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:

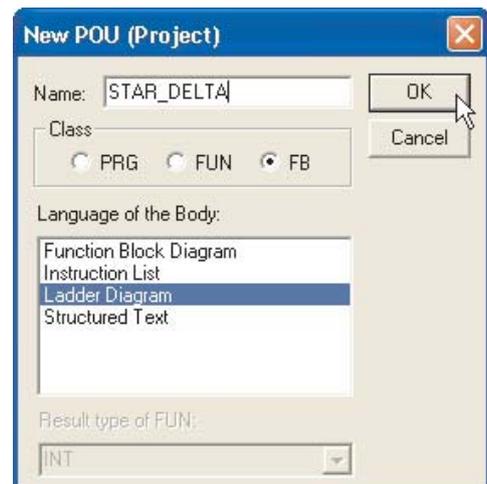
- Кнопка Старт: **START**
- Кнопка СТОП: **STOP**
- Контакт перегрузки: **OVERLOAD**
- Время переключения: **TIMEBASE**
- Временной регистр: **TIME\_COIL**
- Выход контактора ЗВЕЗДА: **STAR\_COIL**
- Выход контактора ТРЕУГОЛЬНИК: **DELTA\_COIL**

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.

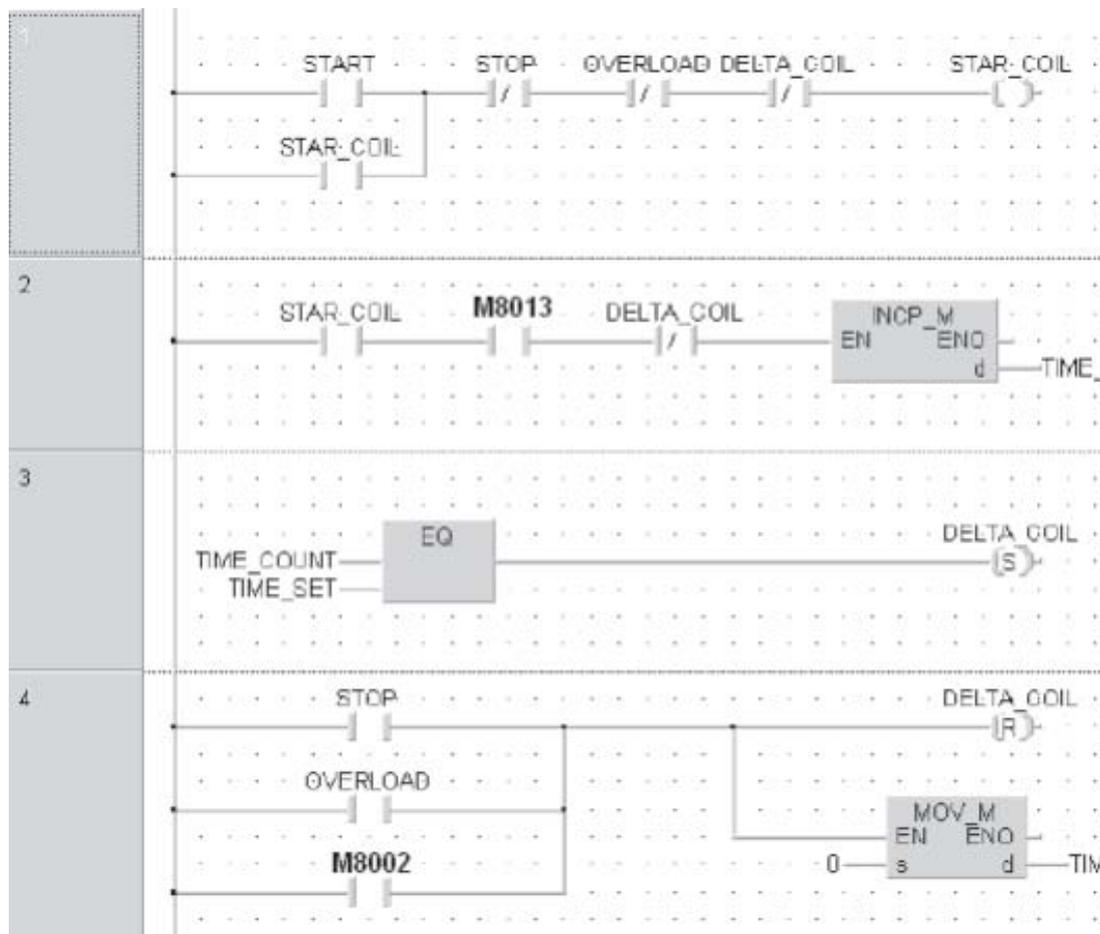
- ③ Щелкните один раз, чтобы открыть ветви заголовка и тела.
- ④ Дважды щелкните, чтобы открыть заголовок.

### Объявление локальных переменных

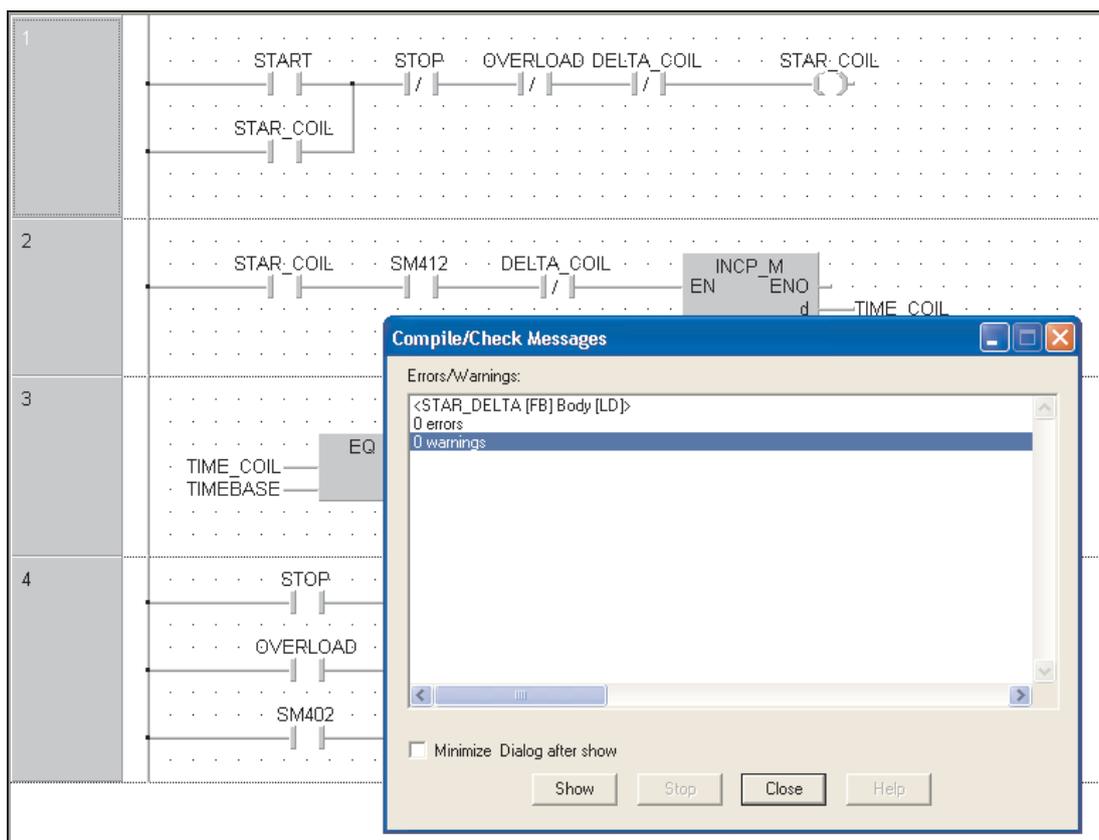
① Объявите переменные, как показано ниже:

	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:

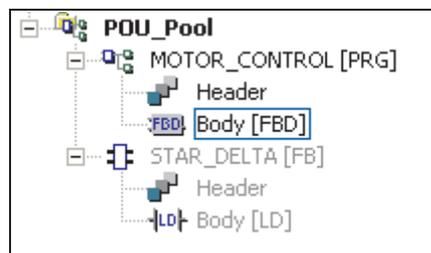


- ④ Проверьте тело, не должно быть ошибок и предупреждений!



### 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.



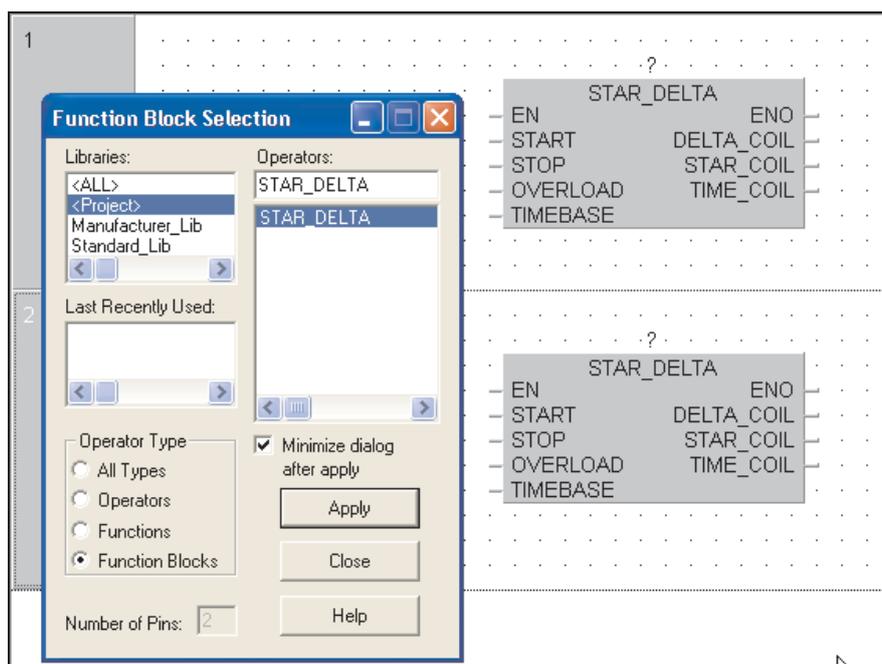
### Создание нового списка глобальных переменных

Откройте GVL и введите следующие детали ввода-вывода:

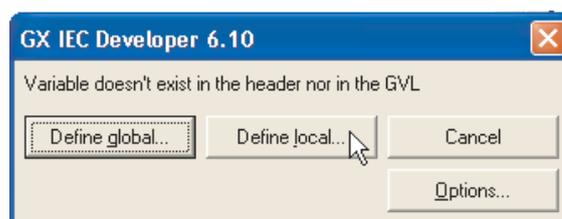
	Class	Identifier	MIT-Addr.	IEC-Addr.	Type	Initial
0	VAR_GLOBAL	START1	X0	%IX0	BOOL	FALSE
1	VAR_GLOBAL	STOP1	X1	%IX1	BOOL	FALSE
2	VAR_GLOBAL	OVERLOAD1	X2	%IX2	BOOL	FALSE
3	VAR_GLOBAL	STAR_COIL1	Y10	%QX16	BOOL	FALSE
4	VAR_GLOBAL	DELTA_COIL1	Y11	%QX17	BOOL	FALSE
5	VAR_GLOBAL	TIME_COIL1	D0	%MW0.0	INT	0
6	VAR_GLOBAL	START2	X3	%IX3	BOOL	FALSE
7	VAR_GLOBAL	STOP2	X4	%IX4	BOOL	FALSE
8	VAR_GLOBAL	OVERLOAD2	X5	%IX5	BOOL	FALSE
9	VAR_GLOBAL	STAR_COIL2	Y12	%QX18	BOOL	FALSE
10	VAR_GLOBAL	DELTA_COIL2	Y13	%QX19	BOOL	FALSE
11	VAR_GLOBAL	TIME_COIL2	D1	%MW0.1	INT	0

### Назначение имен экземпляров

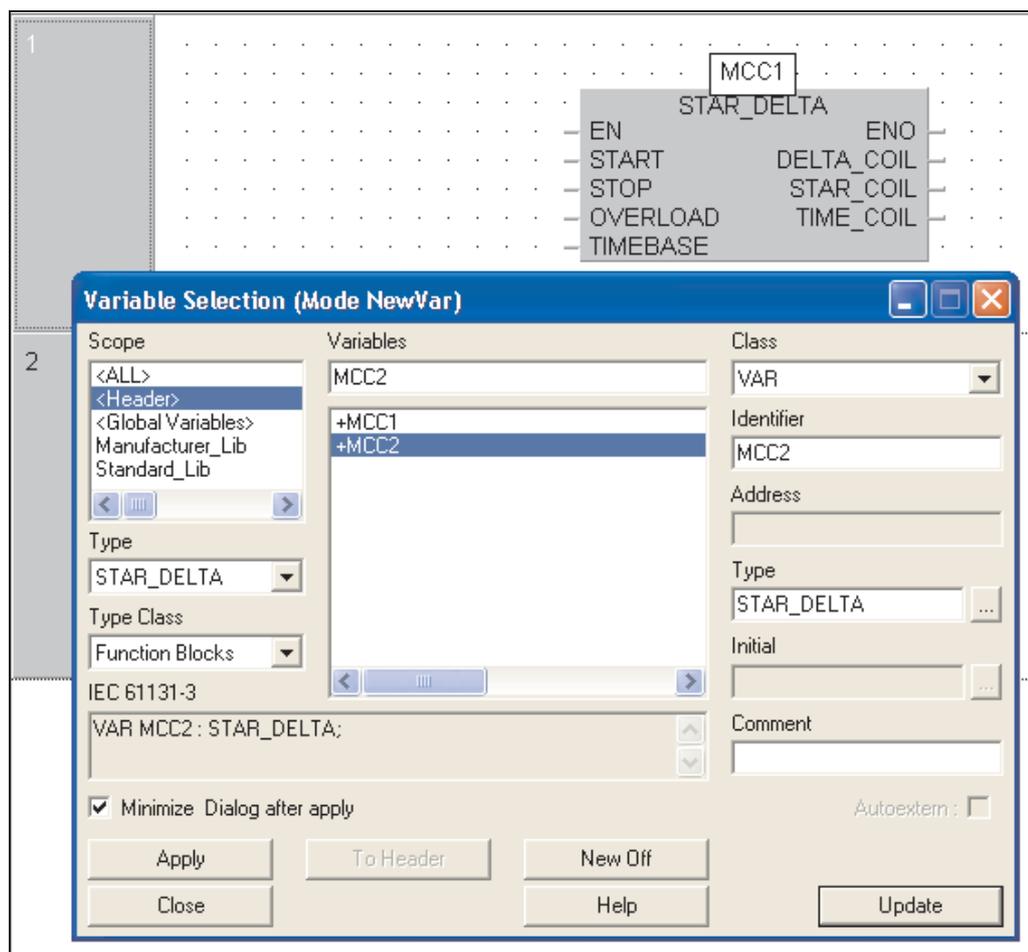
- Откройте тело "MOTOR\_CONTROL" и введите следующее, чтобы создать два звена. Поместите экземпляр функционального блока "STAR\_DELTA" в каждое звено, как показано на следующем рисунке:



- 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**.



- ③ Создайте элементы для имен экземпляров в заголовке для "MCC1" и "MCC2" следующим образом:

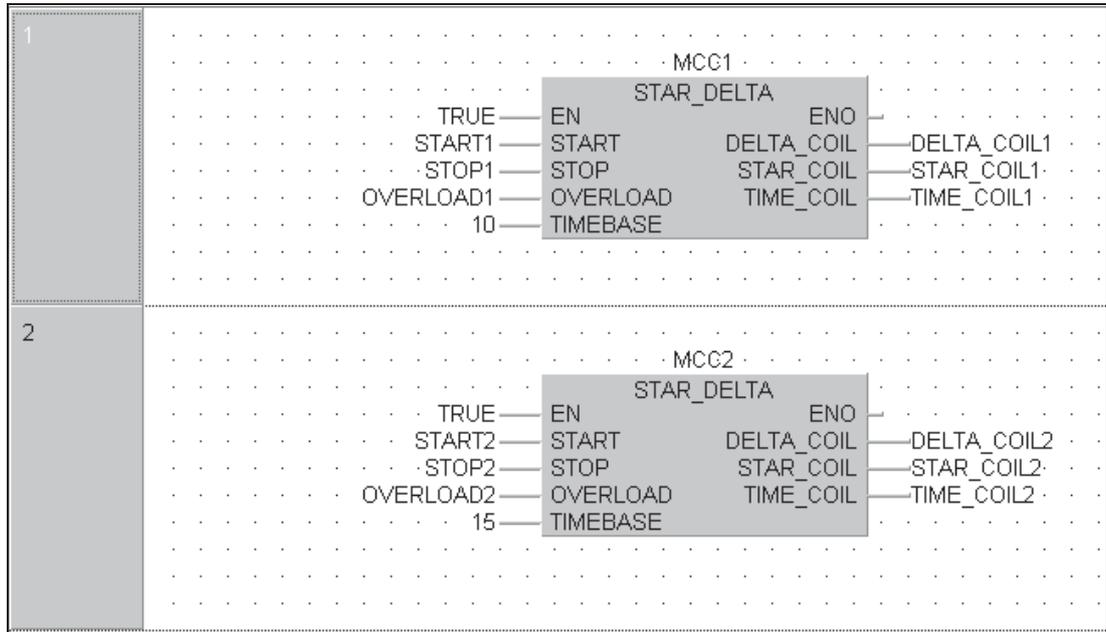


Экземпляр - это копия функционального блока для данного программного модуля. В этом примере просто напечатайте "**MCC1**" и "**MCC2**". Заметим, что после введения экземпляры перечисляются в окне выбора переменных как "+MCC1" и "+MCC2" с типом: "STAR\_DELTA".

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

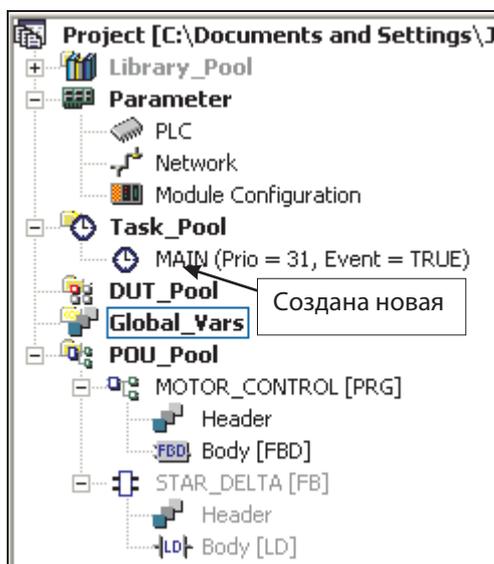
Могут использоваться адреса или объявление символических переменных Mitsubishi. Однако если используются прямые адреса MELSEC Mitsubishi, то программа больше не будет соответствовать соглашениям IEC.

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:

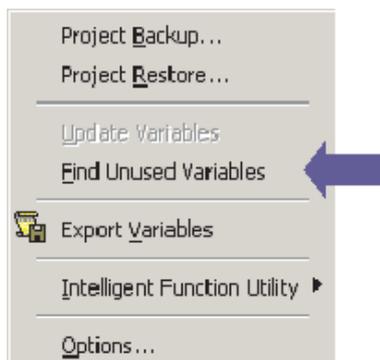


- ② Дважды щелкните на задаче и свяжите программный модуль "MOTOR\_CONTROL" с задачей "MAIN":

POU name	Comment
MOTOR_CONTROL	...

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

### Найти неиспользованные переменные



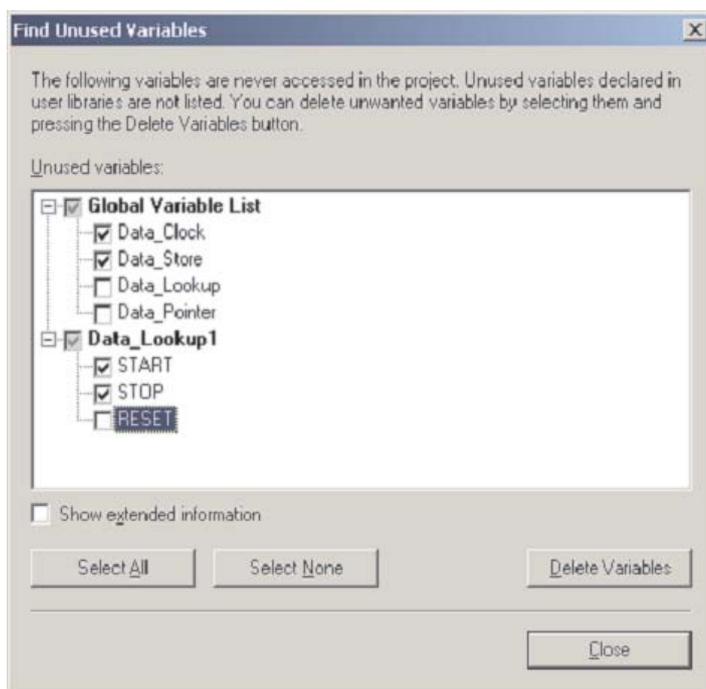
Используя функцию **Extras** → **Find unused Variables** вы можете найти и удалить все неиспользованные глобальные и локальные переменные, которые объявлены, но не используются в проекте.

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

**ПРИМЕЧАНИЕ**

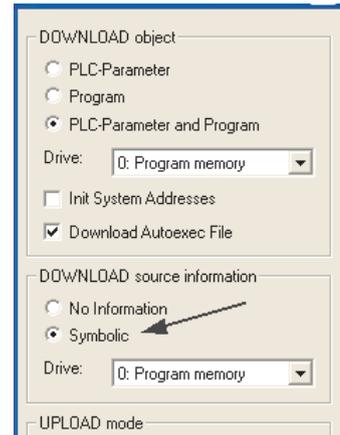
Обнаружение неиспользованных переменных можно выполнить, если проект был создан и с тех пор не изменялся. В противном случае будет показано предупреждающее сообщение.

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.

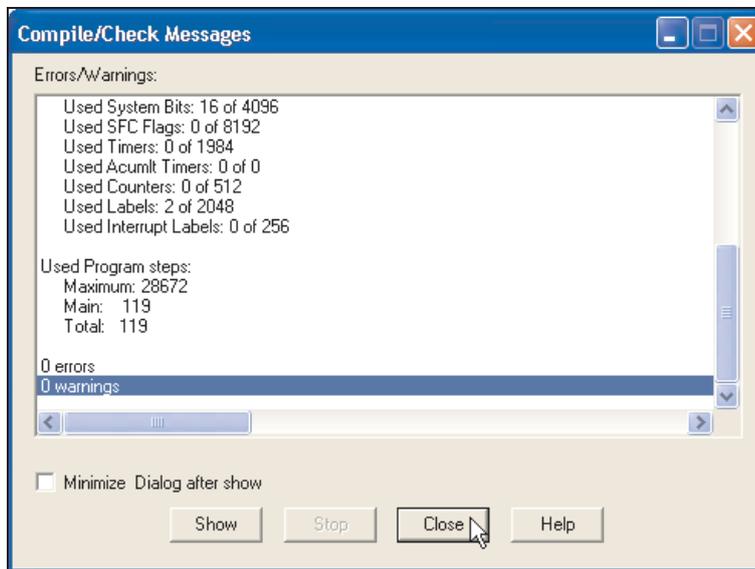


**ПРИМЕЧАНИЕ**

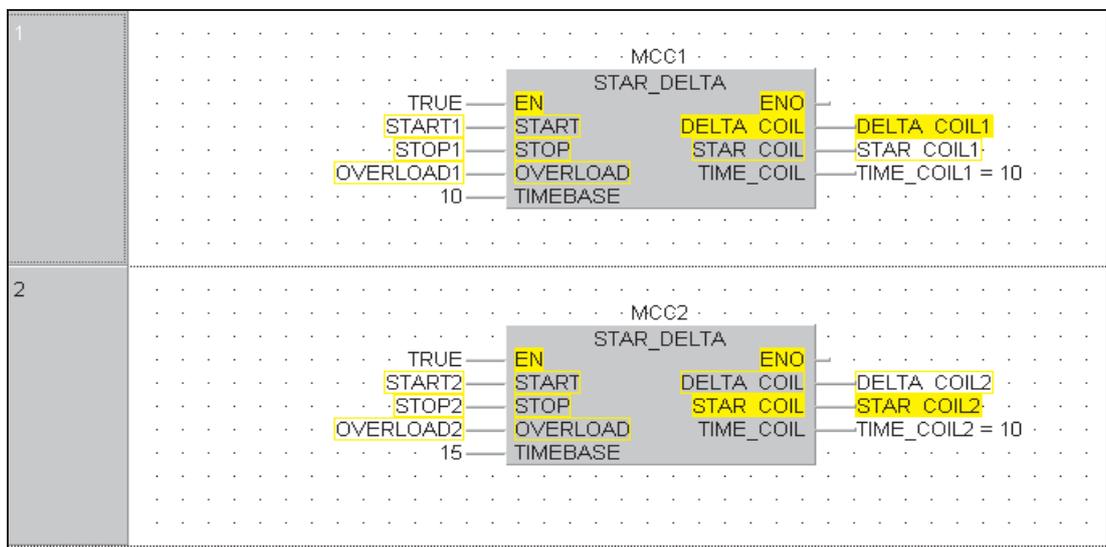
Это может значительно сократить размер исходного текста программы, что особенно важно, если для загрузки была выбрана опция передать весь символический **Symbolic** (исходный) код в ПЛК:



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



Откройте программный модуль "MOTOR\_CONTROL" и контролируйте  правильную работу программы.

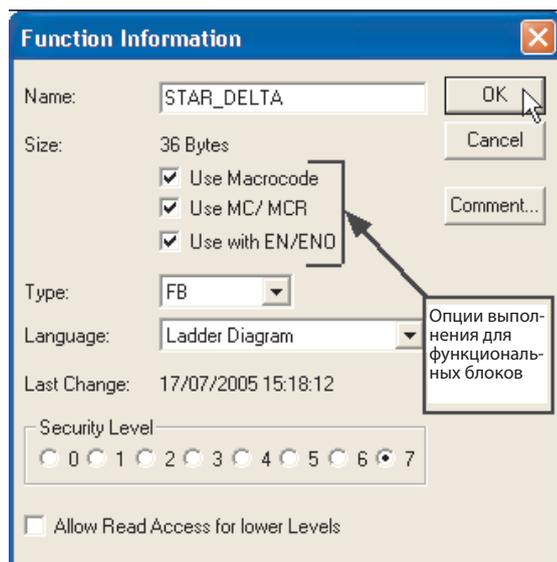


## 6.3 Режимы выполнения функциональных блоков

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

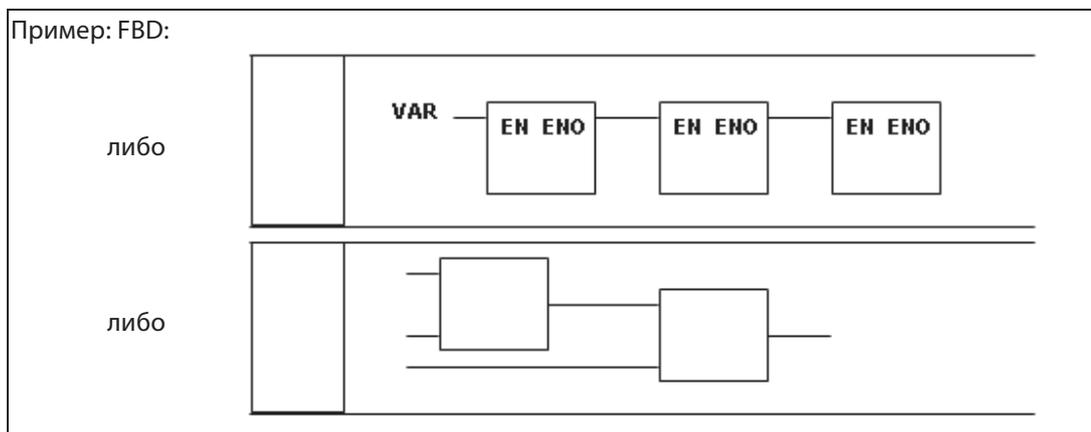
- Стандартное выполнение: Функциональный блок вызывается через системную метку.
- Выполнение макрокоманды: Функциональный блок расширяется внутренним образом.

#### 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:



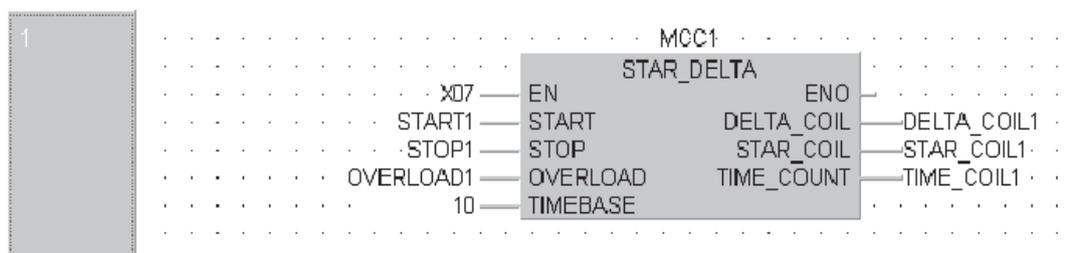
#### Определения функций

- 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:





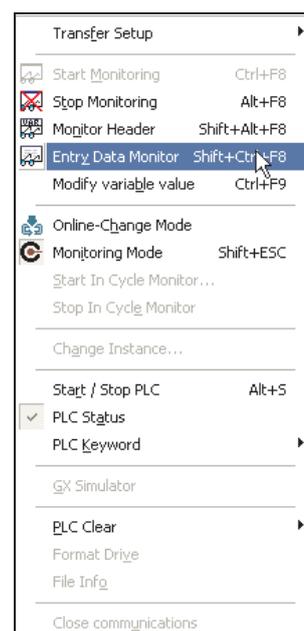
## 7 Расширенные функции мониторинга

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

### 7.1 Контроль входных данных (EDM)

Если требуется одновременно контролировать различные данные из различных частей программы, можно применять Entry Data Monitor (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	X10	START1	0
4	X11	STOP1	0
5	X12	OVERLOAD1	0
6	X13	START2	0
7	X14	STOP2	0
8	X15	OVERLOAD2	0

### 7.1.1 Настройка EDM

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

Pos	Address (MIT)	Name	Value (dec)
1	D0	TIME_COIL1	0
2	D1	TIME_COIL2	0
3	X10	START1	0
4	X11	STOP1	0
5	X12	OVERLOAD1	0
6	X13	START2	0
7	X14	STOP2	0
8	X15	OVERLOAD2	0
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

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...

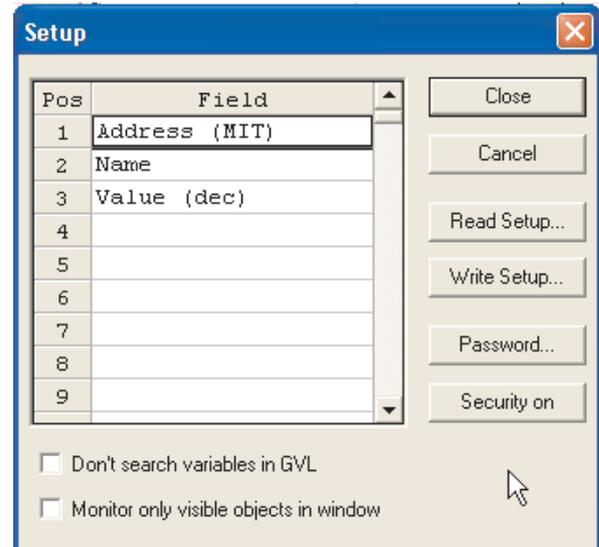
Write to File...

---

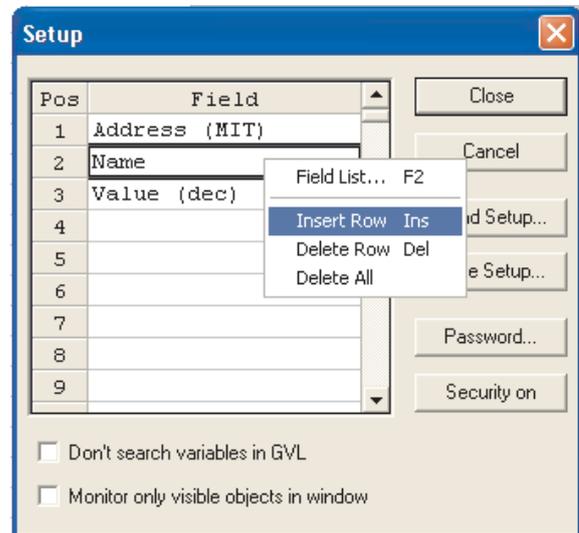
**Setup...**

Always on top

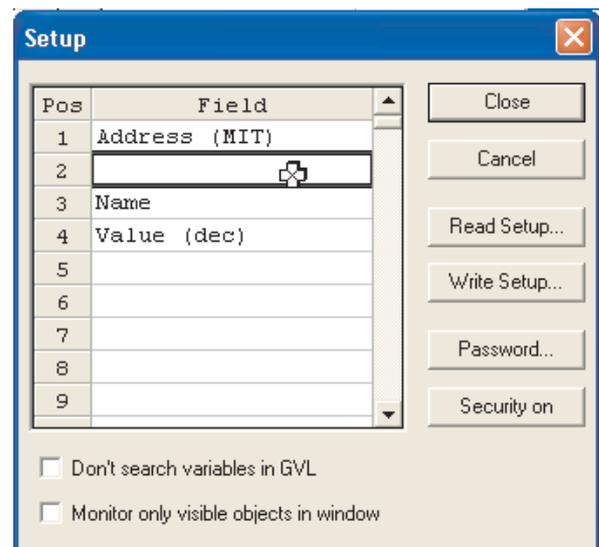
Окно **Setup** дает пользователю возможность конфигурировать EDM; щелкнув правой кнопкой мыши, вы откроете окно конфигуратора. Эта процедура добавляет в таблицу EDM столбцы для адреса IEC и монитора шестнадцатеричных значений.



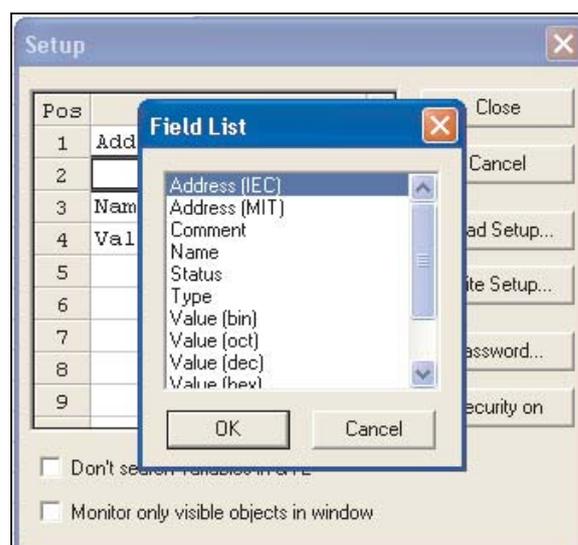
- ② Выделите или щелкните правой кнопкой на поле **Name** и выберите **Insert Row**, как показано.



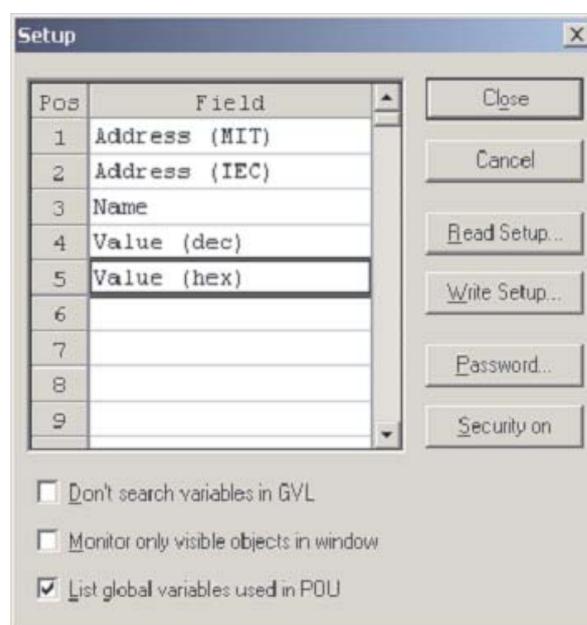
Появляется второе окно с опциями для этой строки; выберите Value (hex), Value (bin). Повторите для Address (IEC) и Type.



- 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	%MWO.0	TIME_COIL1	0	0
2	D1	%MWO.1	TIME_COIL2	0	0
3	X10	%IX16	START1	0	0
4	X11	%IX17	STOP1	0	0
5	X12	%IX18	OVERLOAD1	0	0
6	X13	%IX19	START2	0	0
7	X14	%IX20	STOP2	0	0
8	X15	%IX21	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.

В EDM можно также вводить значения для отображаемых объектов. Например, можно изменить значение регистра данных, пометив соответствующее поле и введя значение.

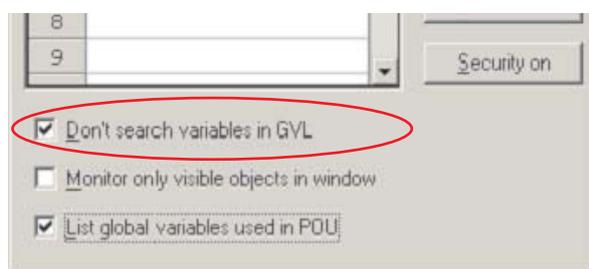
**ПРИМЕЧАНИЕ**

Состояния и значения операндов контроллера определяются программой контроллера. Entry Data Monitor перезаписывает операнды лишь на короткое время. После этого они снова принимают состояние или значение, присвоенное им программой.

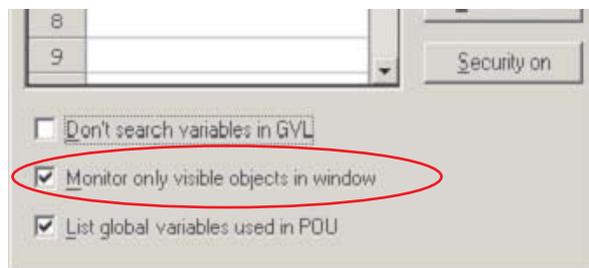
В этом примере программа циклически изменяет содержимое регистров Do и D1.

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

После того, как вы введете адрес Mitsubishi, 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**.



## 7.1.2 Изменение состояния битовых операндов в EDM

При тестировании программы на состояния битовых операндов (данные типа BOOL) можно влиять и непосредственно, с помощью Entry Data Monitor. Например, если для инициации определенного процесса нужен входной сигнал от выключателя в установке, этот вход можно установить в EDM, а затем наблюдать за дальнейшим ходом программы.

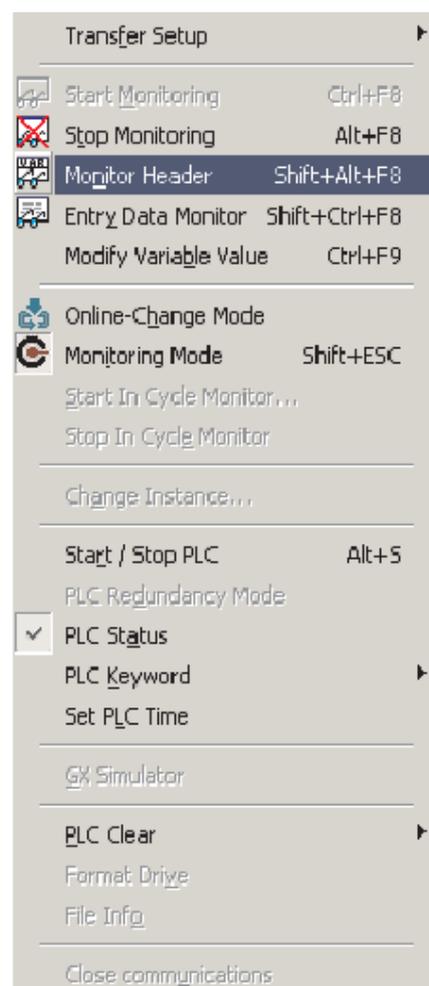
При двойном щелчке по полю со значением булевой переменной ее состояние попеременно изменяется (например, 0 → 1 → 0 → 1 и т. п.).

Pos	Address (MIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1	D0	%MWO.0	TIME_COIL1	10	A
2	D1	%MWO.1	TIME_COIL2	0	0
3	X10	%IX16	START1	1	1
4	X11	%IX17	STOP1	0	0
5	X12	%IX18	OVERLOAD1	0	0
6	X13	%IX19	START2	0	0
7	X14	%IX20	STOP2	0	0
8	X15	%IX21	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	X10	%IX16	START1	1	1
3	X11	%IX17	STOP1	0	0
4	X12	%IX18	OVERLOAD1	0	0
5	Y21	%QX33	DELTA_COIL1	1	1
6	Y20	%QX32	STAR_COIL1	0	0
7	DO	%MWO.0	TIME_COIL1	10	A
8	X13	%IX19	START2	0	0
9	X14	%IX20	STOP2	0	0
10	X15	%IX21	OVERLOAD2	1	1
11	Y23	%QX35	DELTA_COIL2	0	0
12	D1	%MWO.1	TIME_COIL2	0	0
13	Y22	%QX34	STAR_COIL2	0	0
14			+MCC1		
15			+MCC2		
16					

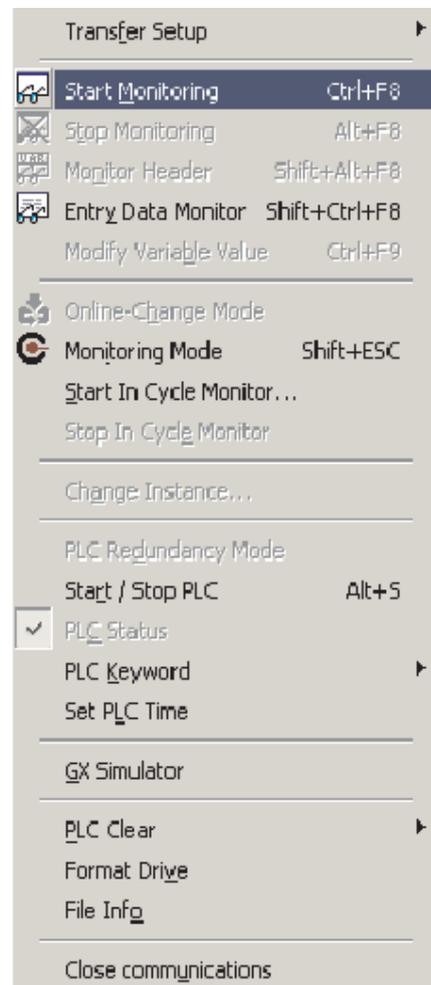
Учтите, что при мониторинге логические переменные в EDM показаны подсвеченными.

## 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 the SIMATIC Manager software. The top window, titled 'MOTOR\_CONTROL (MOTOR\_CONTROL)', shows a table of monitored variables. The bottom window, titled 'MOTOR\_CONTROL [PRG] Body [FBD]', shows the corresponding ladder logic for two motor control modules, MCC1 and MCC2.

Pos	Address (MIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1			-MOTOR_CONTROL		
2	X10	%IX16	START1	1	1
3	X11	%IX17	STOP1	0	0
4	X12	%IX18	OVERLOAD1	0	0
5	Y21	%QX33	DELTA_COIL1	1	1
6	Y20	%QX32	STAR_COIL1	0	0
7	D0	%MW0.0	TIME_COIL1	10	A
8	X13	%IX19	START2	0	0
9	X14	%IX20	STOP2	0	0
10	X15	%IX21	OVERLOAD2	1	1
11	Y23	%QX35	DELTA_COIL2	0	0
12	D1	%MW0.1	TIME_COIL2	0	0
13	Y22	%QX34	STAR_COIL2	0	0
14			+MCC1		
15			+MCC2		

The ladder logic in the bottom window shows two networks, MCC1 and MCC2. Each network contains a 'STAR\_DELTA' function block. The inputs and outputs are as follows:

- MCC1:**
  - EN: X17
  - START: START1
  - STOP: STOP1
  - OVERLOAD: OVERLOAD1
  - TIMEBASE: 10
  - ENO: (empty)
  - DELTA COIL: DELTA\_COIL1
  - STAR COIL: STAR\_COIL1
  - TIME\_COUNT: TIME\_COIL1 = 10
- MCC2:**
  - EN: TRUE
  - START: START2
  - STOP: STOP2
  - OVERLOAD: OVERLOAD2
  - TIMEBASE: 15
  - ENO: (empty)
  - DELTA COIL: DELTA\_COIL2
  - STAR COIL: STAR\_COIL2
  - TIME\_COUNT: TIME\_COIL2 = 0

## 7.4 Указание групп битов

Чтобы с помощью одной команды обратиться к нескольким следующим друг за другом битовым операндам, адрес первого битового операнда указывается вместе с коэффициентом "K", обозначающим количество операндов. Этот коэффициент "K" означает количество групп по 4 операнда: K1 = 4 операнда, K2 = 8 операндов, K3 = 12 операндов и т. п.

Например, указание "K2M0" определяет восемь маркеров с M0 по M7. Возможны коэффициенты от K1 (4 операнда) до K8 (32 операнда).

Также можно производить контроль, используя обозначение Mitsubishi Kn (официально - "Бланк перевода") для логических объектов. Например, K1X10 контролирует X10-X13, как показано в следующем примере:

Pos	Address (MIT)	Address (IEC)	Name	Value (dec)	Value (hex)
1	D0	%MWO.0	TIME_COIL1	10	A
2	D1	%MWO.1	TIME_COIL2	0	0
3	X10	%IX16	START1	1	1
4	X11	%IX17	STOP1	0	0
5	X12	%IX18	OVERLOAD1	0	0
6	X13	%IX19	START2	0	0
7	X14	%IX20	STOP2	0	0
8	X15	%IX21	OVERLOAD2	1	1
9					
10	K1X10	%IW19.1.16	K1X10	1	1
11					
12					

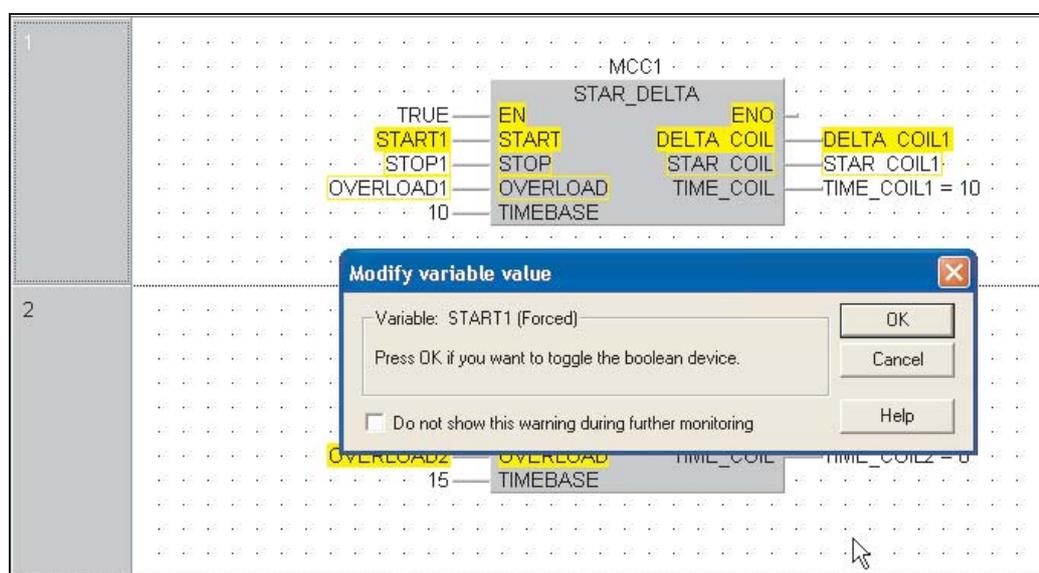
## 7.5 Модификация значений переменных из тела программного модуля

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.



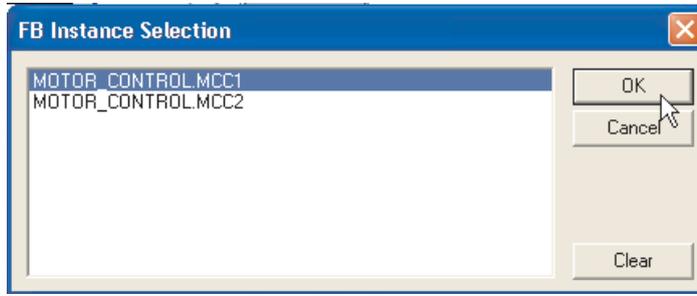
### ПРИМЕЧАНИЕ

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

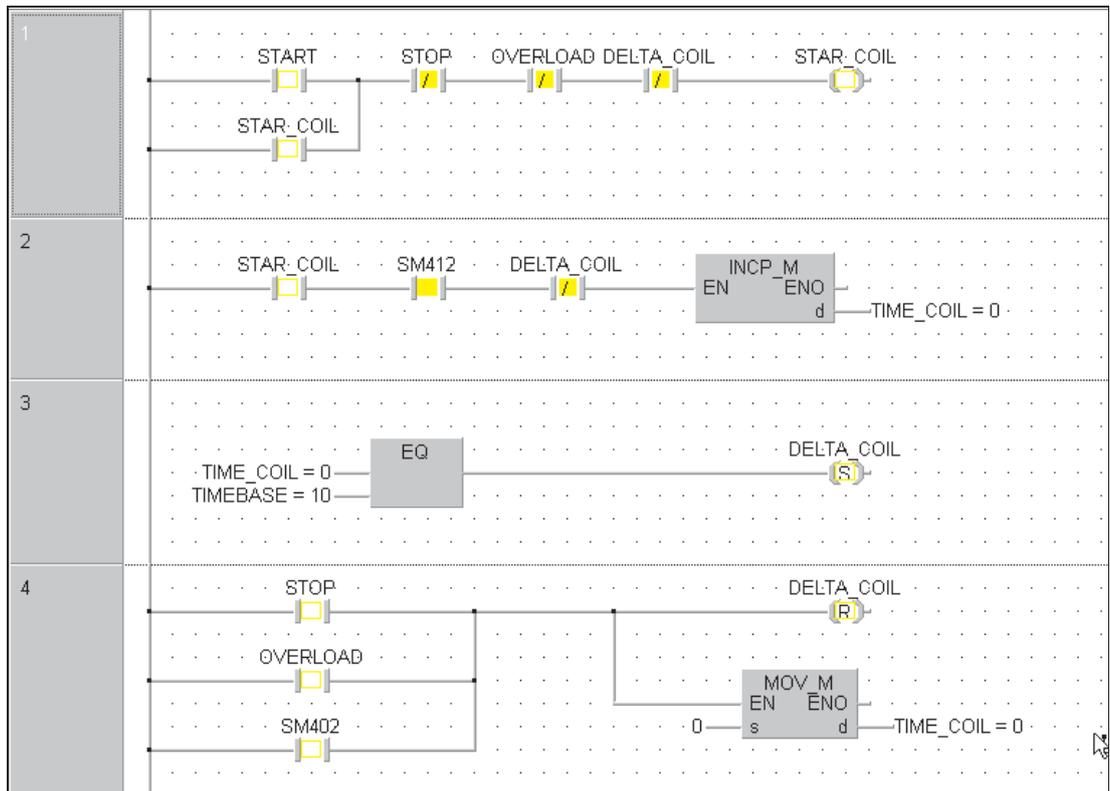
## 7.6 Мониторинг "экземпляров" функциональных блоков

Отдельные "экземпляры" функциональных блоков можно контролировать независимо.

- ① 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:



- ② Выберите экземпляр функционального блока MOTOR\_CONTROL.MCC1 и наблюдайте контролируемую страницу:



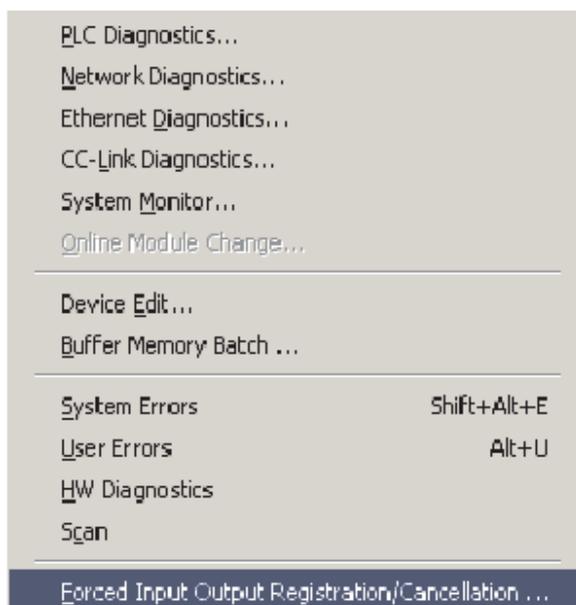
Этот способ позволяет автономно контролировать каждый экземпляр любого функционального блока.

## 8 Принудительная установка цифровых входов и выходов

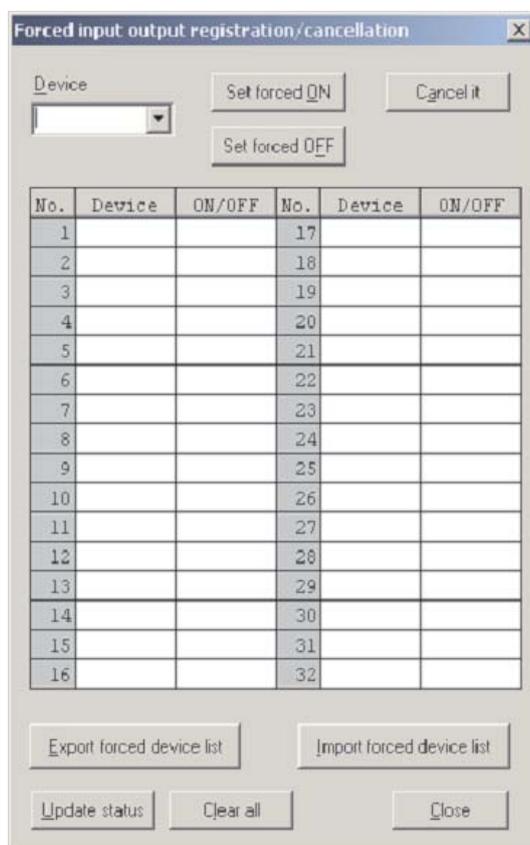
Эта функция GX IEC Developer позволяет принудительно устанавливать регистры физических аппаратных входов и выходные регистры независимо от цикла программы.

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

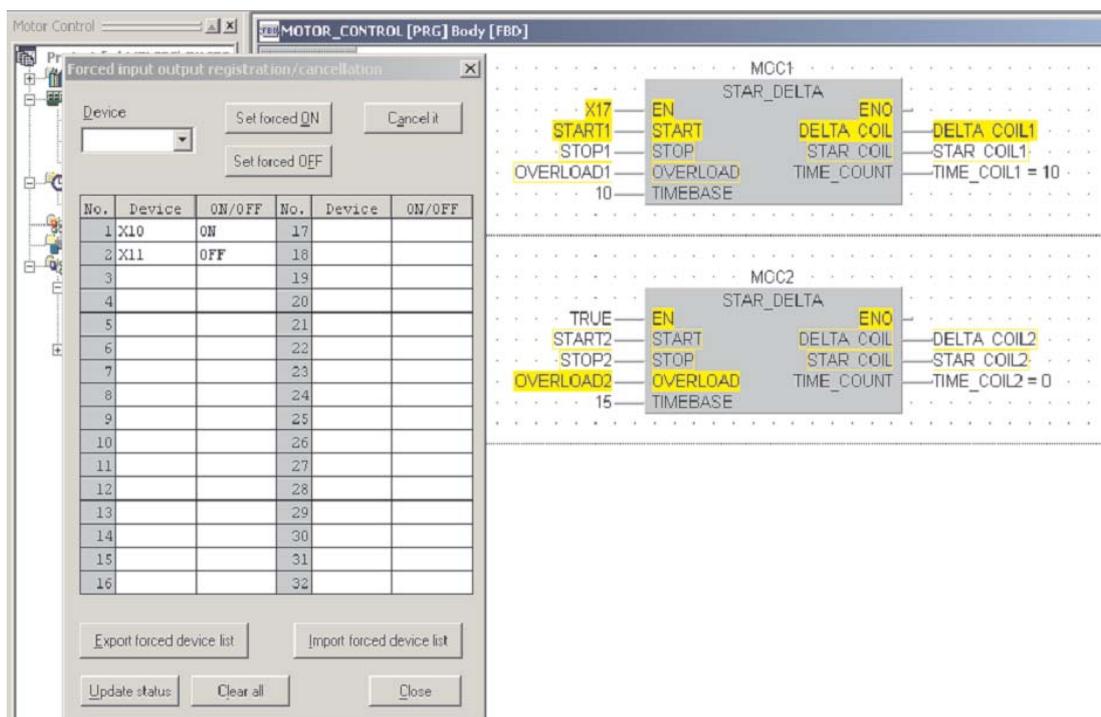
- ① Чтобы активизировать эту функцию и выбрать **Forced input output registration/cancellation**, выберите ее из меню **Debug** следующим образом:



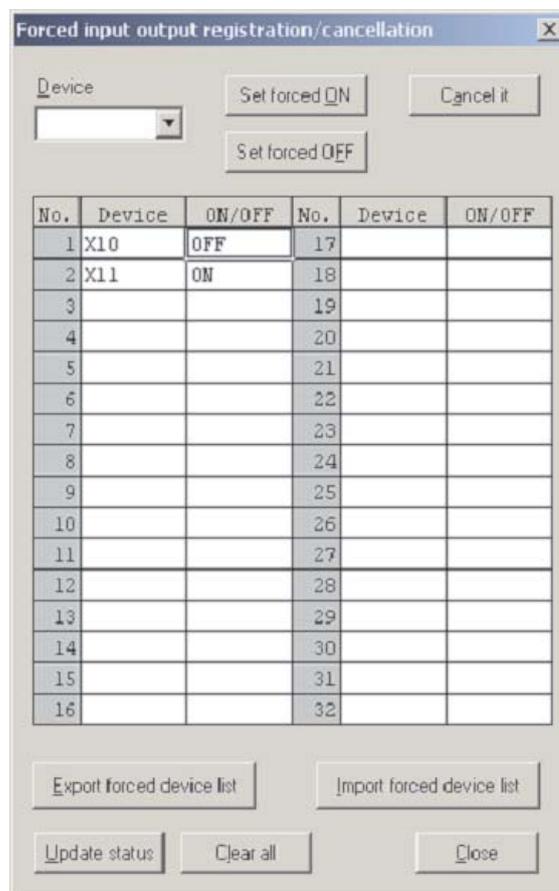
- Появится следующее окно:



- ② Введите X10 и X15 в диалоговое окно **Device** и щелкните на кнопке **Set Forced ON** для обеих переменных:

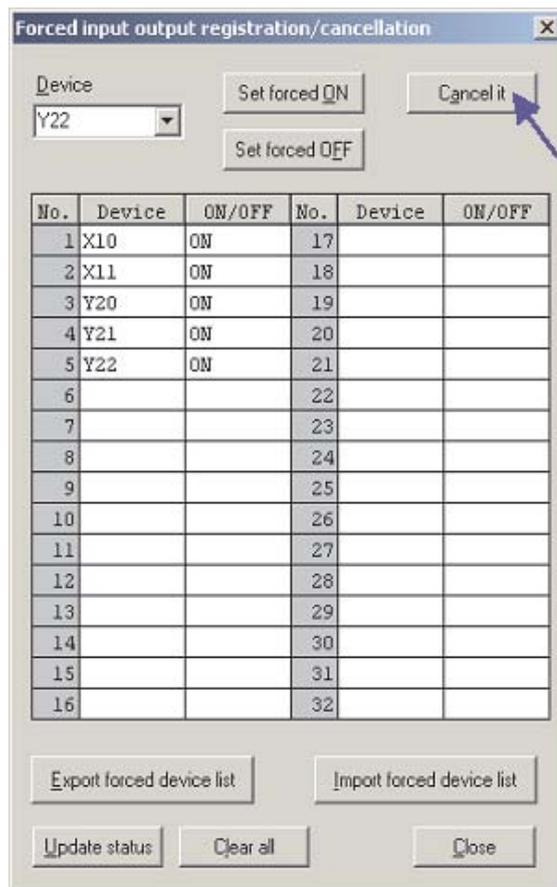


- ③ Чтобы переключить состояние X10 или X15, дважды щелкните левой кнопкой мыши на поле состояния **ON/OFF**.

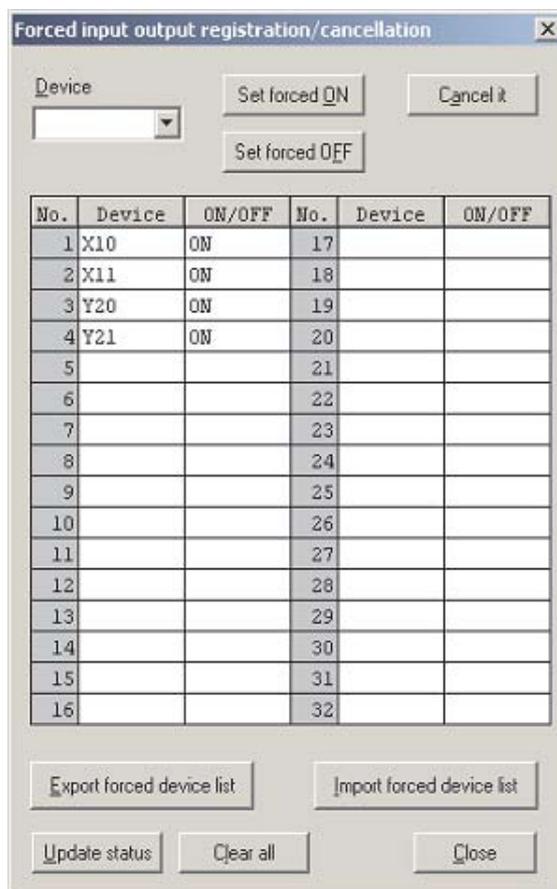


- ④ Воспользуйтесь этим методом принудительной установки для Y20, Y21 и Y22, отмечая воздействие на операнды.

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



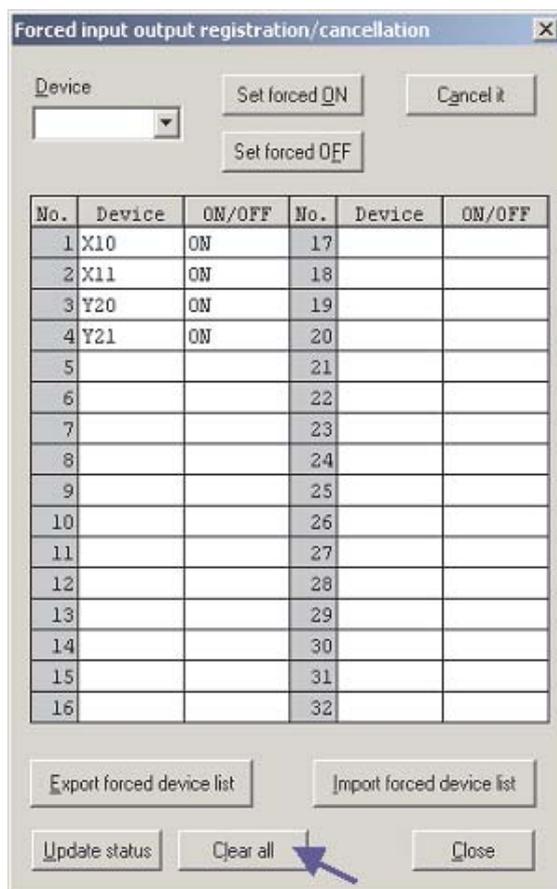
- ⑥ Дисплей примет следующий вид:



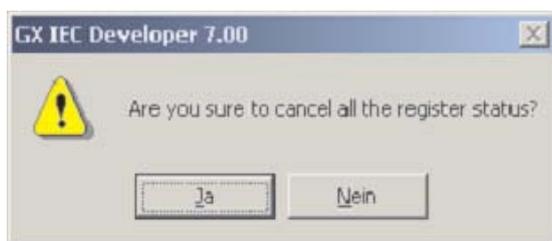
**ПРИМЕЧАНИЕ**

Если какому-либо входу или выходу центрального процессора присвоено принудительное состояние, светодиод "MODE" центрального процессора мигает с частотой 2 Гц.

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



- ⑧ Подтвердите запрос на отмену в следующем окне:



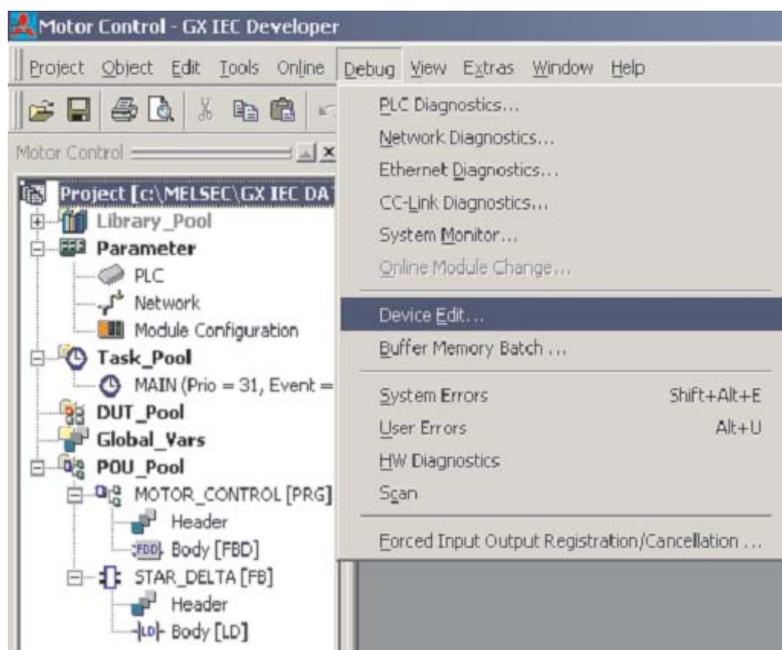
**ПРИМЕЧАНИЕ**

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

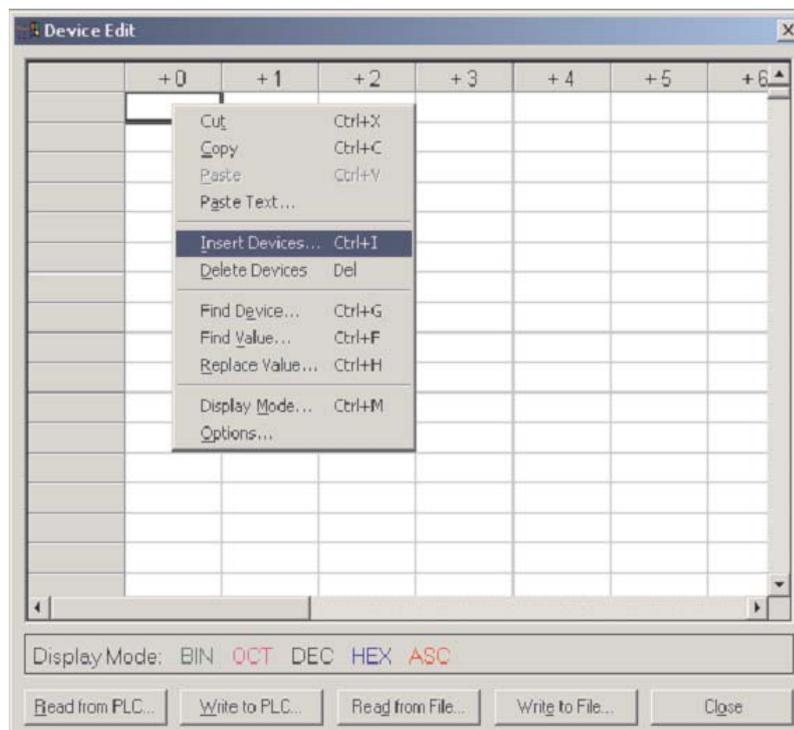
## 9 Device Edit

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

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

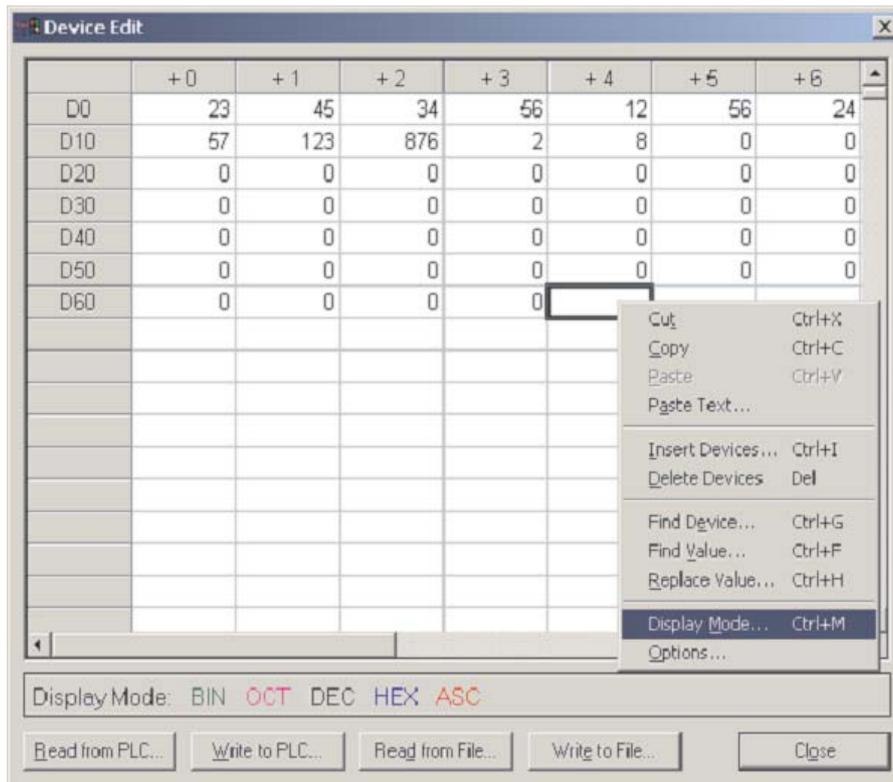


- 2 Выделите ячейку в верхнем левом углу. Щелкните правой кнопкой мыши и затем выберите **Insert Devices**:

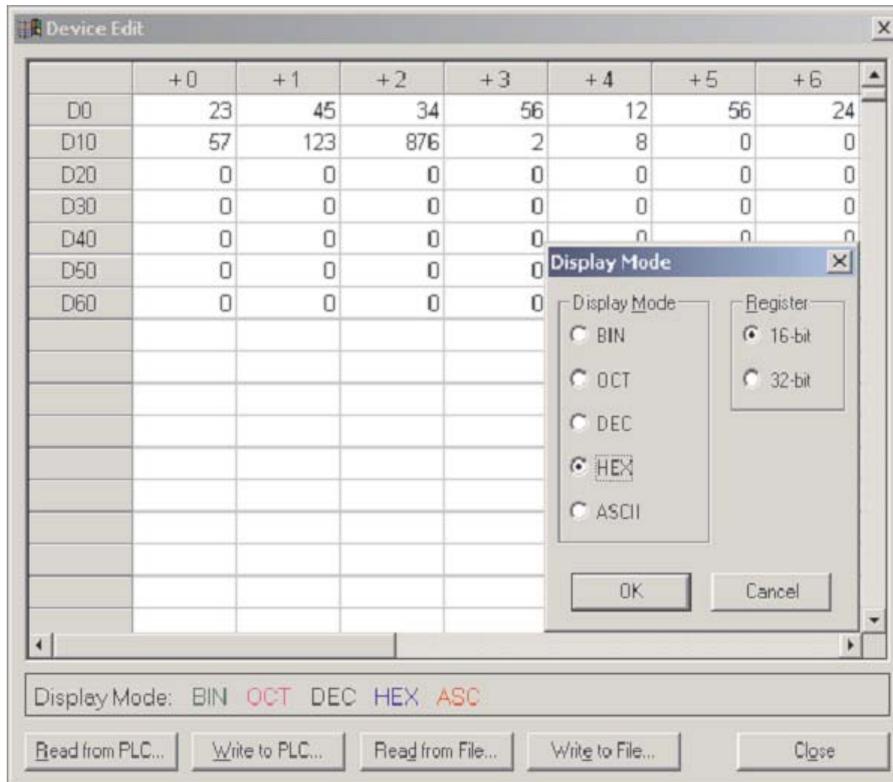




- ④ Выделите строку, щелкнув на левой ячейке, например, "D0". Выберите **Display Mode:**



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

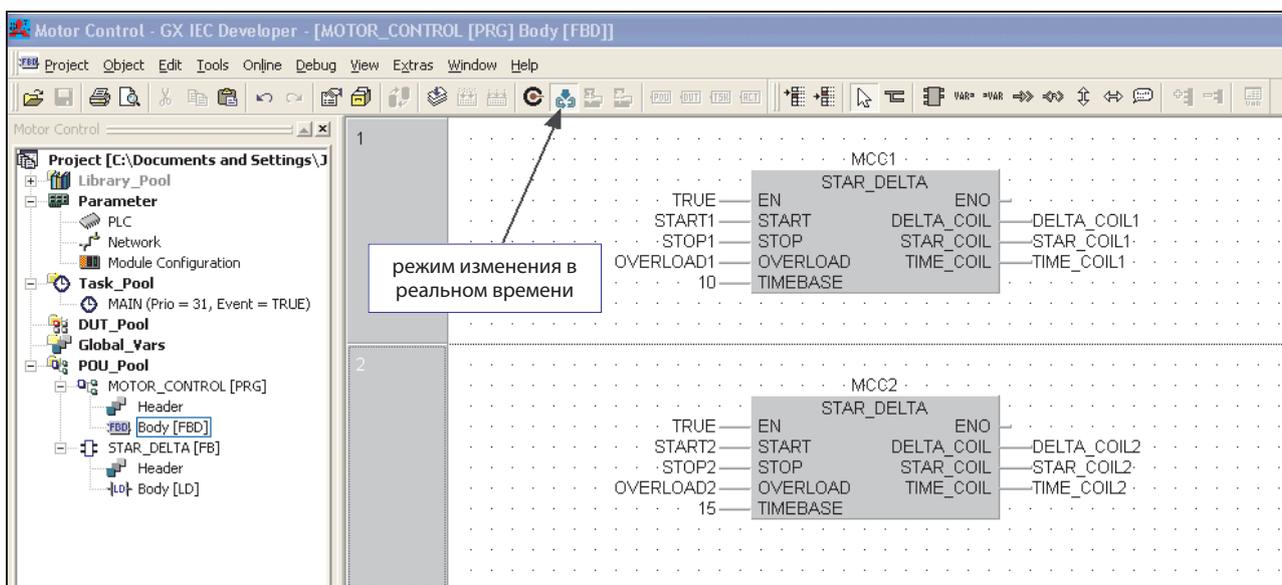
Режим мониторинга, описанный в разделе 4.4 и главе 7, удобен для наблюдения за состояниями операндов и выполнением программы. Если во время контроля требуется изменить программу в контроллере, GX IEC Developer предлагает две возможности, описываемые в следующем разделе.

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.

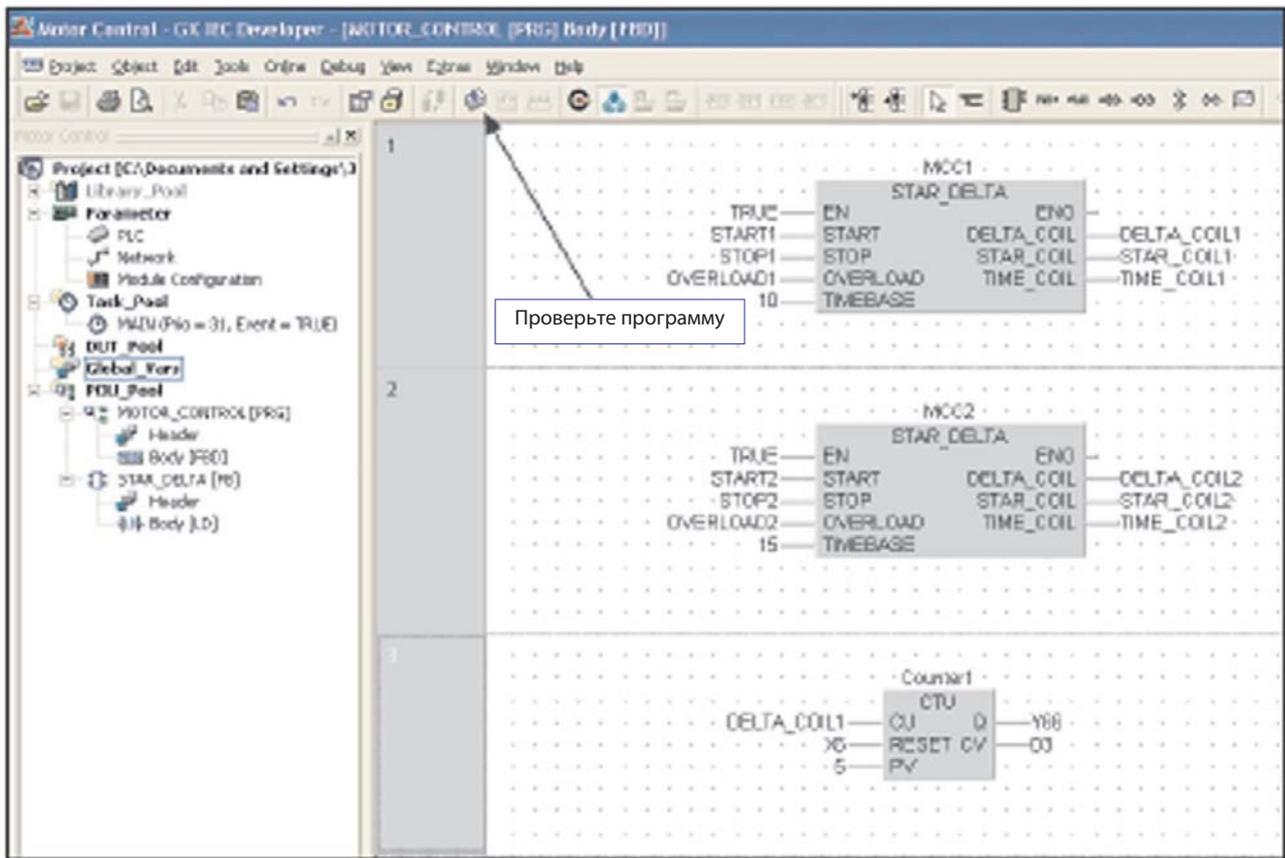
Компилируйте проект и загрузите его в ПЛК.

### 10.1 Режим изменения в реальном времени

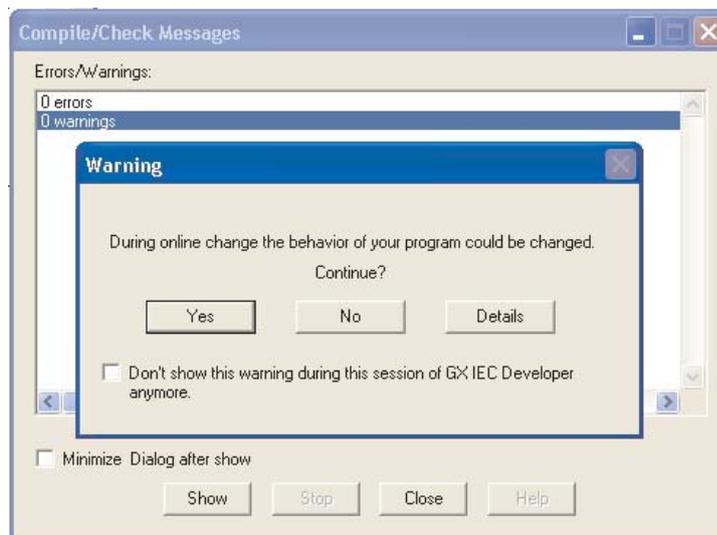
- 1 Откройте тело программного модуля "MOTOR\_CONTROL" и выберите **Online change mode**:



- ② Добавьте дополнительное звено, как показано ниже:



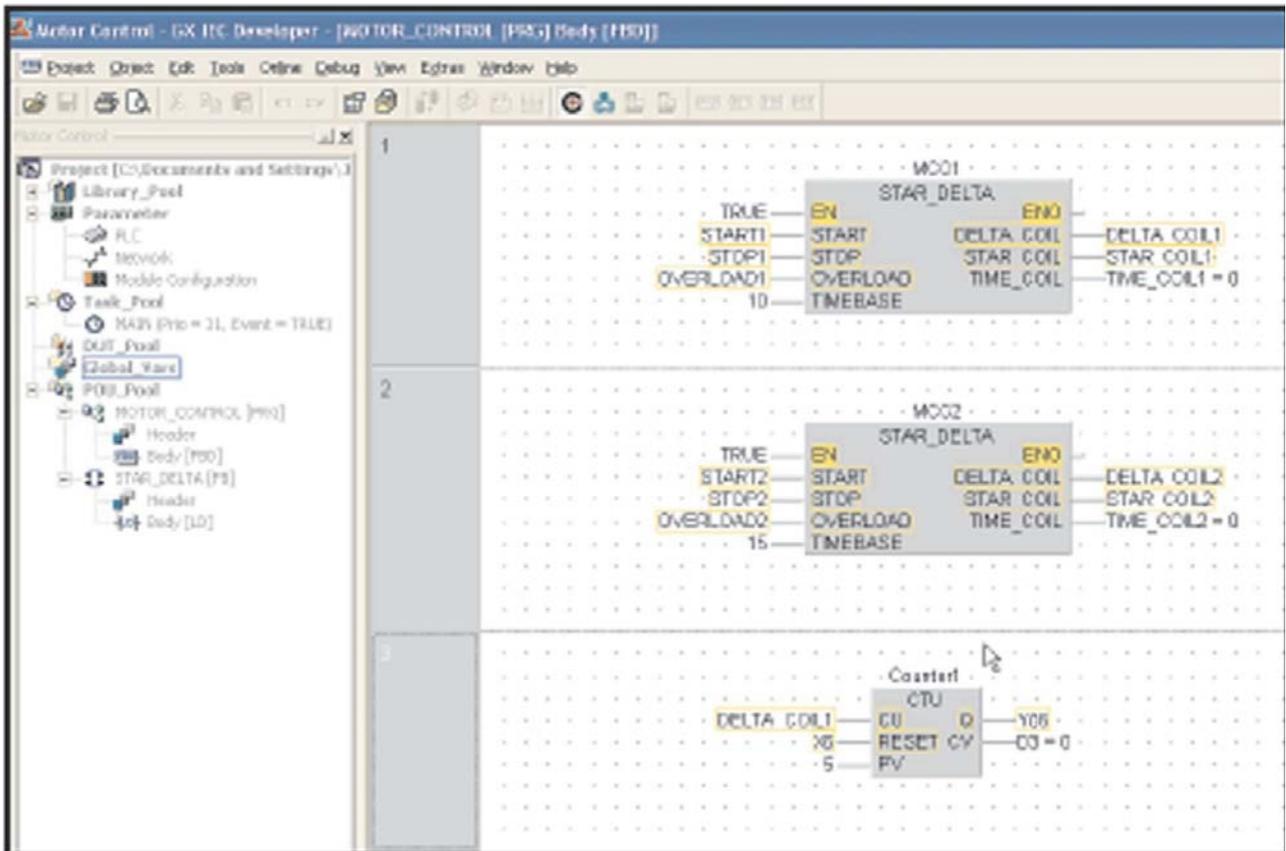
- ③ Затем щелкните мышью вне этого звена или на кнопке проверки, и изменения компилируются и посылаются в ПЛК автоматически после запроса о выполнении или прерывании действия:



**ПРИМЕЧАНИЕ**

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

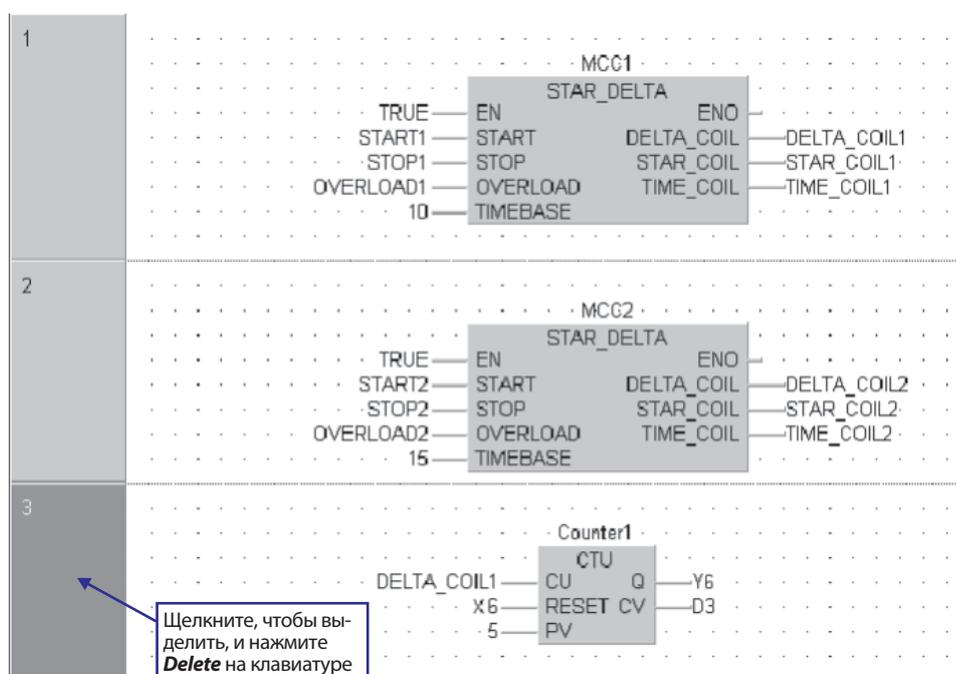
- ④ Войдите в режим Monitor и наблюдайте работу модифицированного блока:



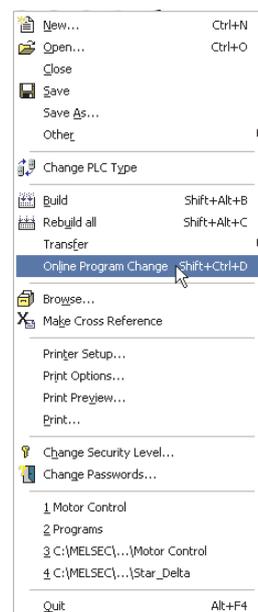
## 10.2 Изменение программы в режиме онлайн

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.

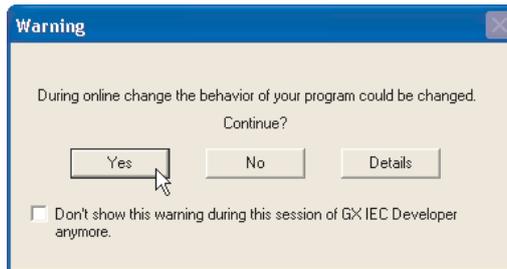


- 2 Вызовите функцию **Online Program Change** из меню **Project**. GX IEC Developer автоматически компилирует и запишет изменение в режиме реального времени.

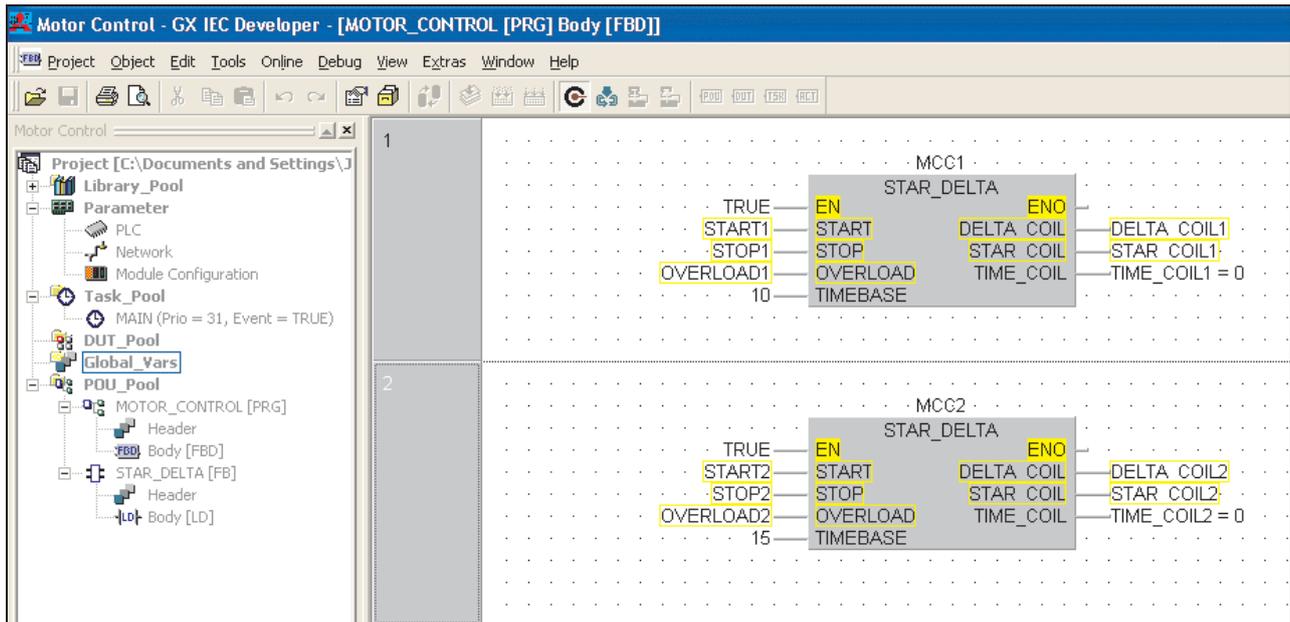


В этой точке система выдаст запрос о продолжении или прерывании процесса.

- 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.

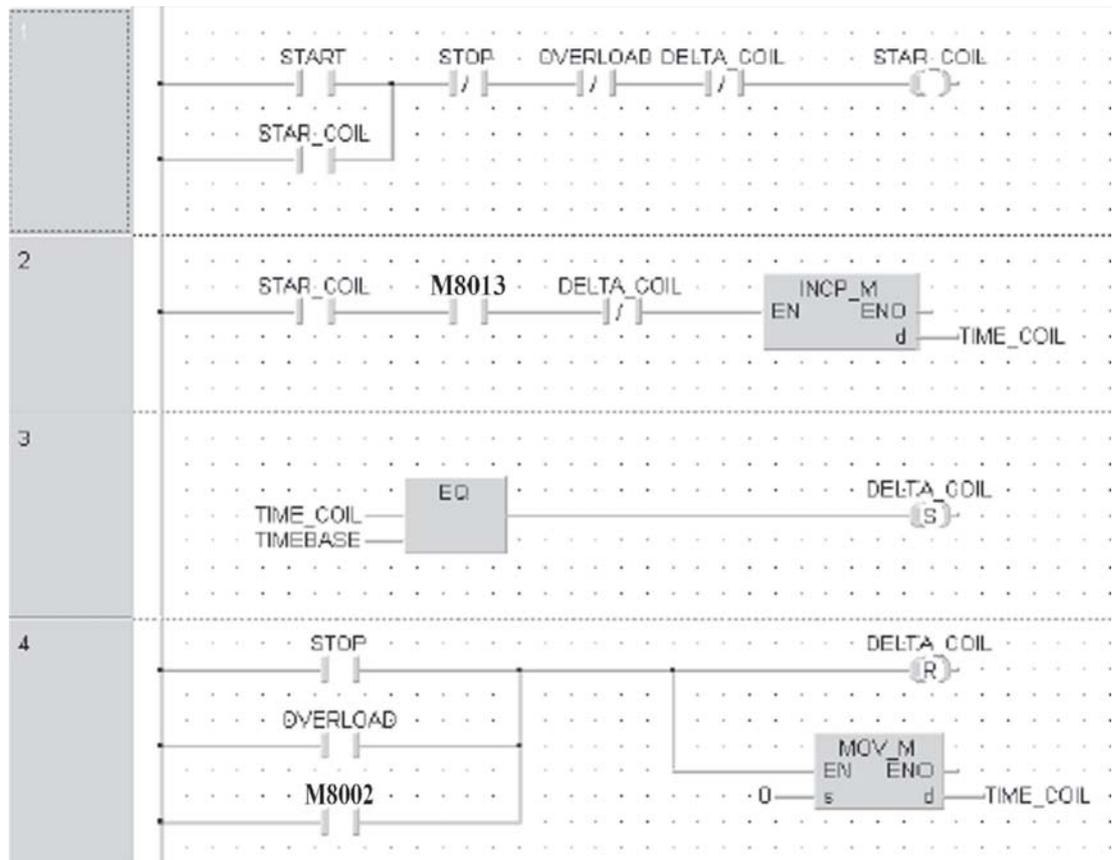
Если элемент START существует в типе "SD", то он может повторно использоваться в каждом экземпляре блока управления двигателем "Звезда - треугольник", когда объявлен в GVL; STAR\_DELTA1.START, STAR\_DELTA2.START и т.д.

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

- 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:

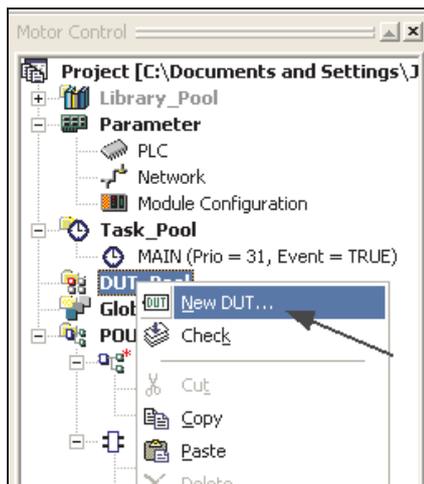
### Тело: STAR\_DELTA



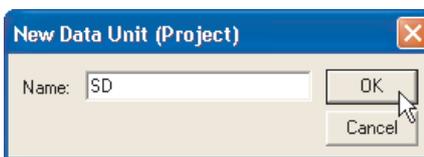
### Заголовок: STAR\_DELTA

	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:

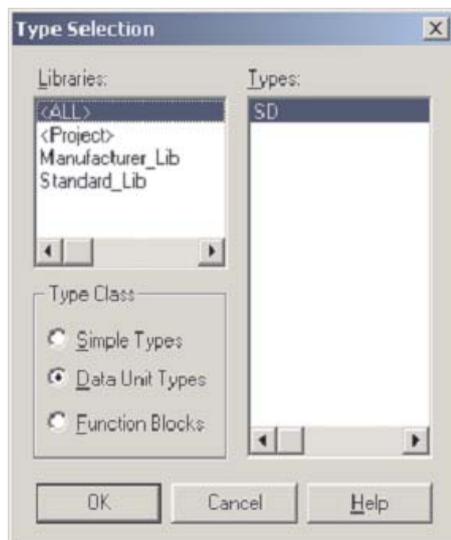
	Identifier	Type	Initial	Comment
0			...	

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	...

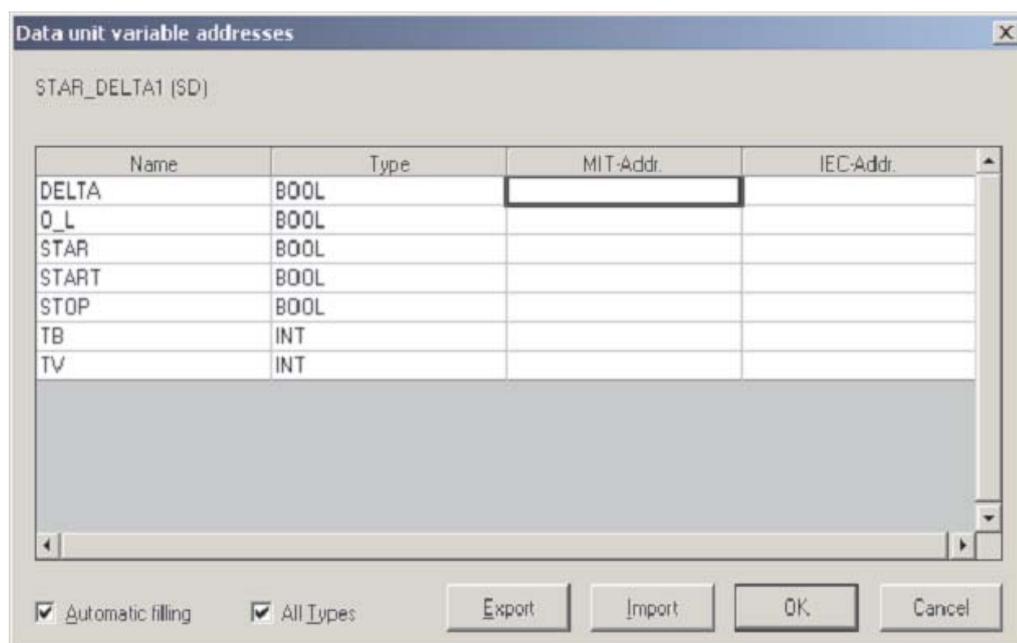


- ⑫ Затем щелкните на ячейке **MIT-Addr.** для STAR\_DELTA1, чтобы ввести данные переменных для выбранного элемента DUT:

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

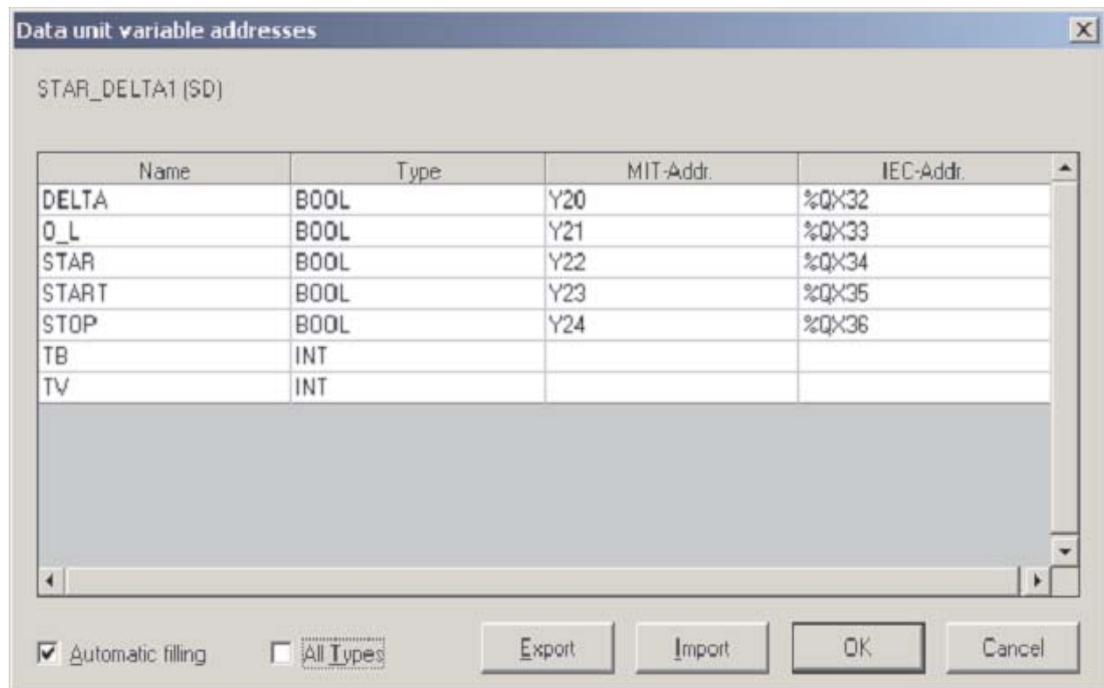
Щелкните, чтобы выбрать

В результате получим окно:



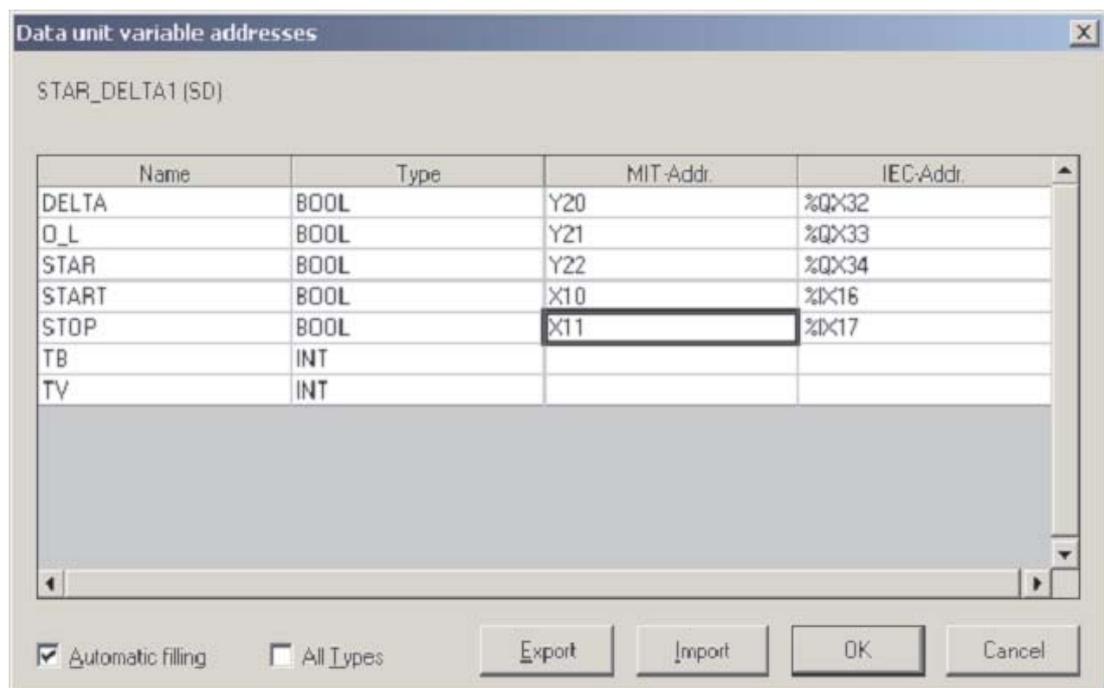
## 11.2 Автоматическое заполнение, переменные

- ① При использовании смешанных типов переменных не разрешается снимать выделение с опции **All types** в этой операции.
- ② Введите Y20 в позицию **MIT-Addr.** для переменной: "DELTA":

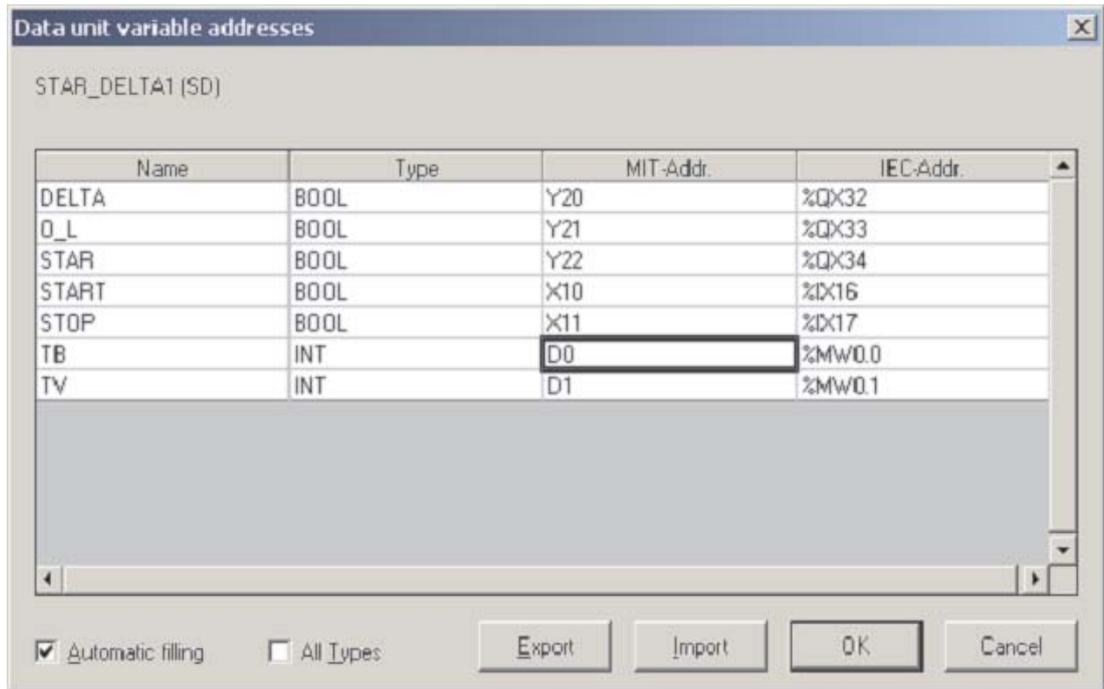


Система попытается последовательно "автозаполнять" переменные типа BOOL. Хотя этот подход рекомендуется во многих ситуациях, в данном случае он лишь частично успешен.

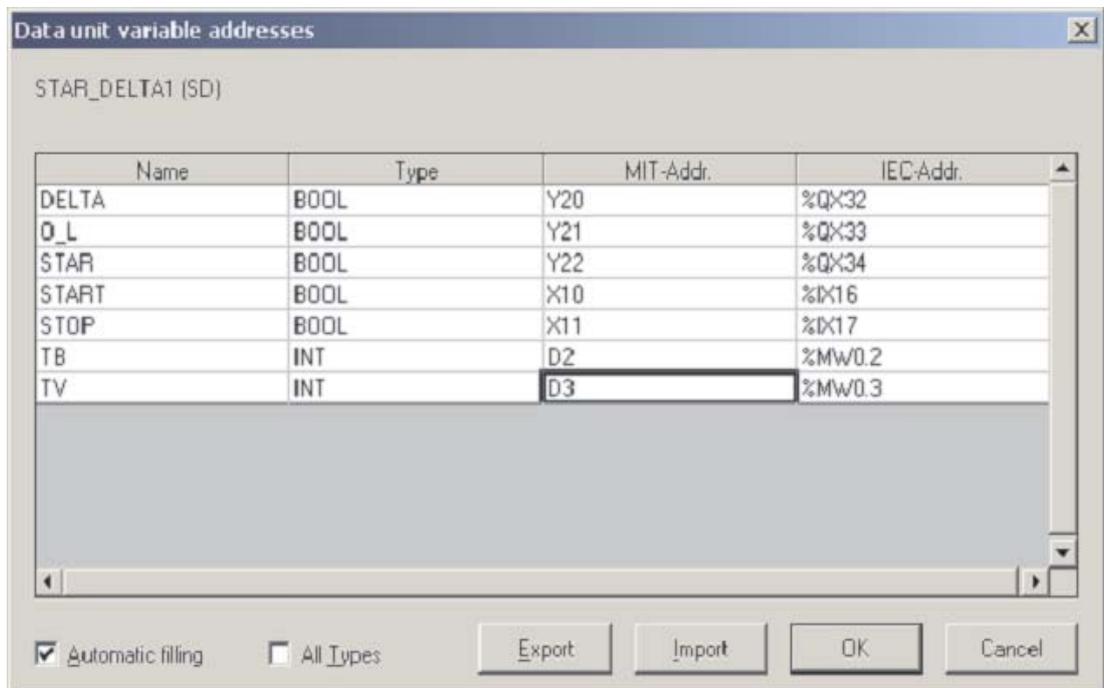
- ③ Поэтому перезапишите для переменных "START и STOP" X10 и X11 следующим образом:



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



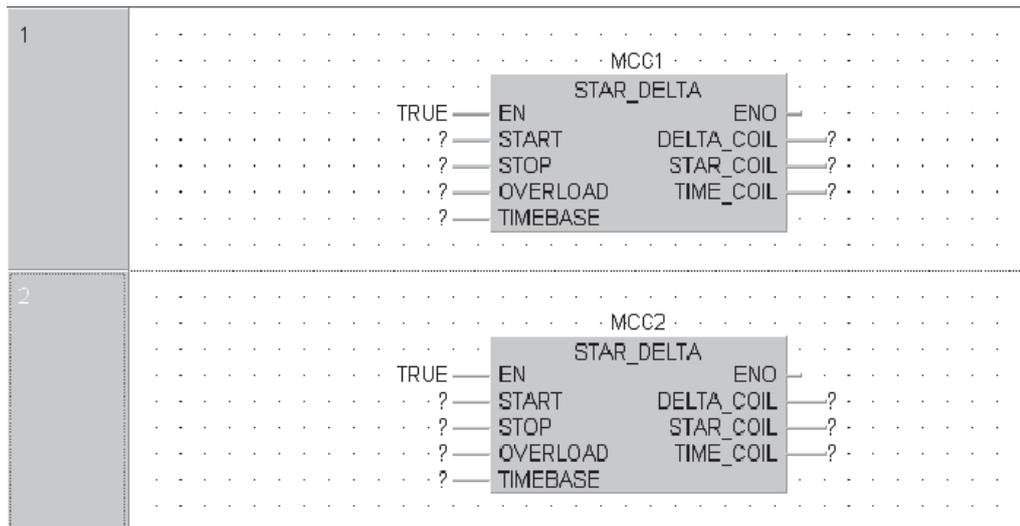
- ⑤ Щелкните на **OK** чтобы сохранить текущую конфигурацию.
- ⑥ Повторите эту серию операций для "STAR\_DELTA2" введя следующий последовательный адрес заголовка для каждой переменной **type**:



- 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	...

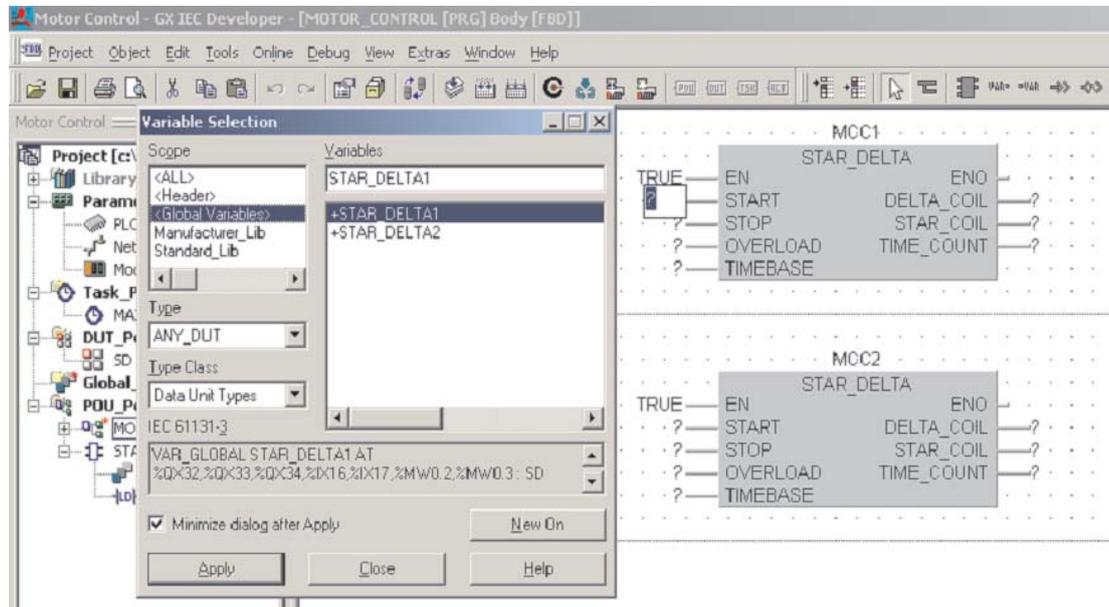
Откройте программный модуль программы MOTOR\_CONTROL и поместите 2 экземпляра пользовательского функционального блока STAR\_DELTA, как показано:



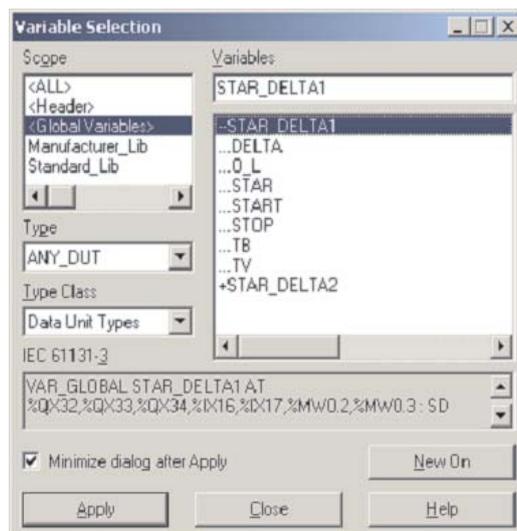
## 11.3 Назначение переменных DUT функциональным блоком

Чтобы назначить переменные функциональным блокам...

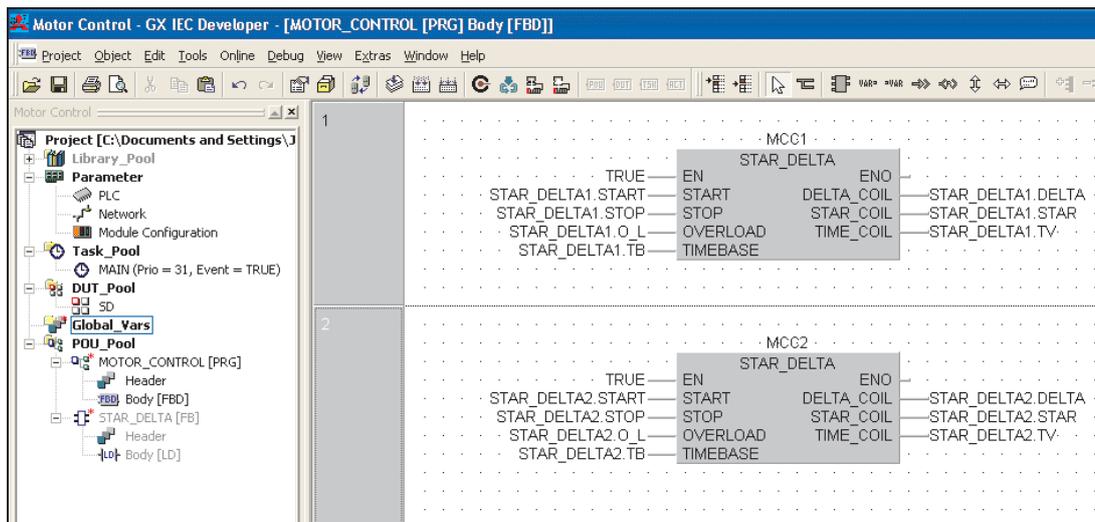
- 1 ...щелкните правой кнопкой на переменной (или F2). Появляется следующее окно выбора переменных:



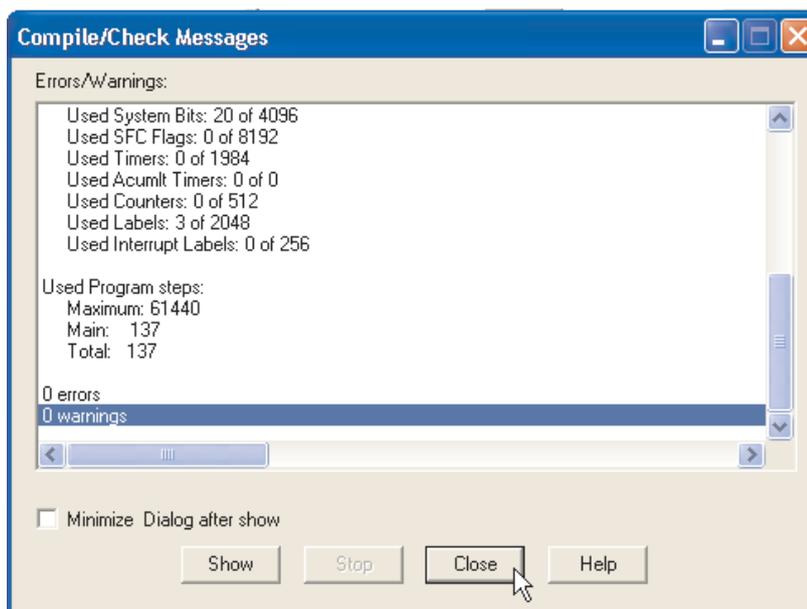
- 2 Установите для **Scope** значение **Header**, для **Type Class** значение **Data Unit Types** и для **Type** значение ANY\_DUT.
- 3 Дважды щелкните на +STAR\_DELTA1, появляется следующий расширенный список переменных DUT:



- ④ Выберите и присвойте переменные двум функциональным блокам STAR\_DELTA на программном модуле программы MOTOR\_CONTROL, как показано:

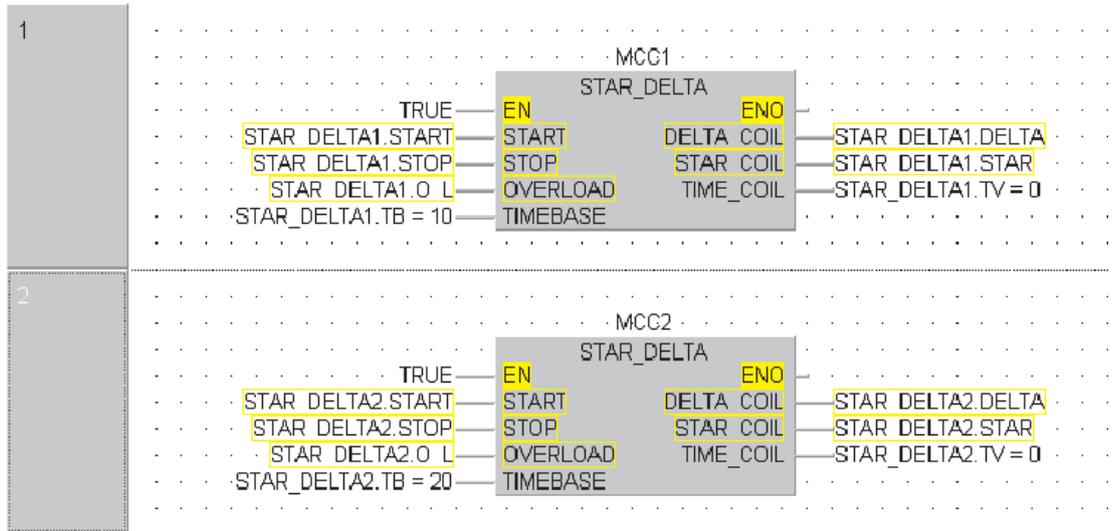


Сохраните проект и выполните **Rebuild All**, чтобы компилировать код:



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.

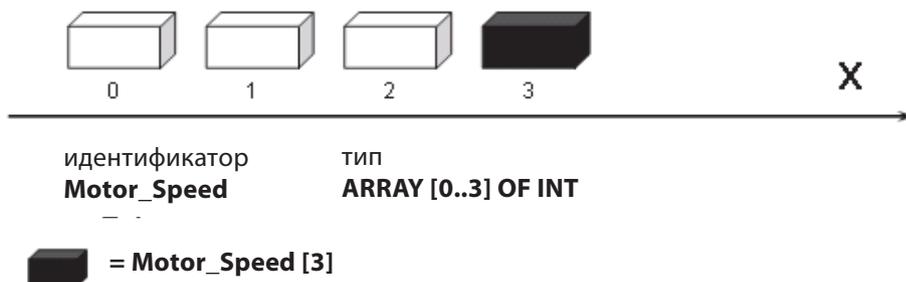
В программе элементы программы могут использовать: Motor\_Volts[1] и Motor\_Volts[2] как объявления; в данном примере это означает, что адресуются D1 и D2 .

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

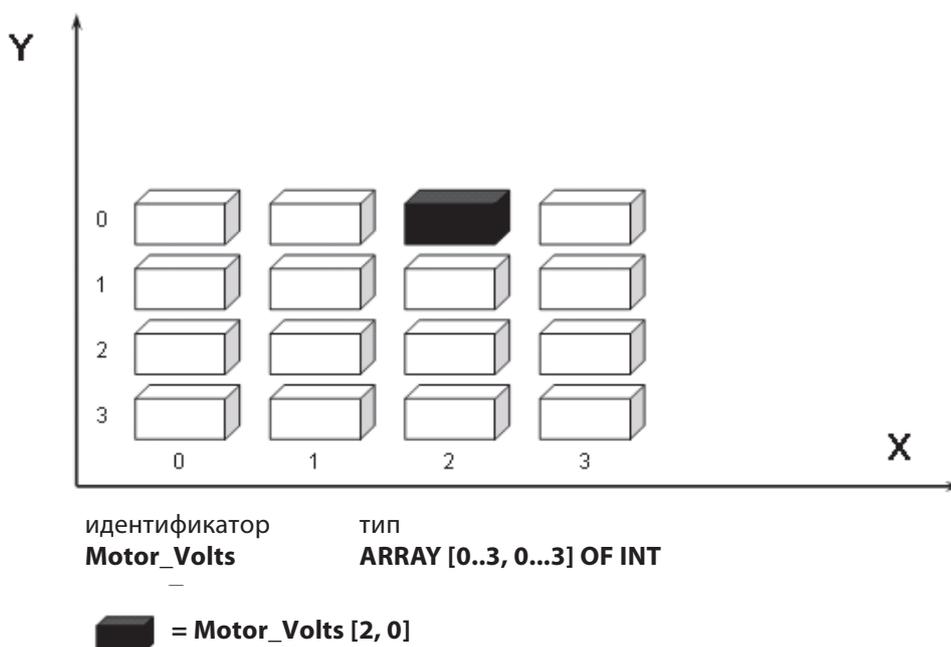
Массивы могут обеспечить удобный способ "индексации" имен тегов, т.е. одно объявление в таблице локальных или глобальных переменных может иметь доступ ко многим элементам.

Следующие диаграммы иллюстрируют графическое представление трех типов массивов.

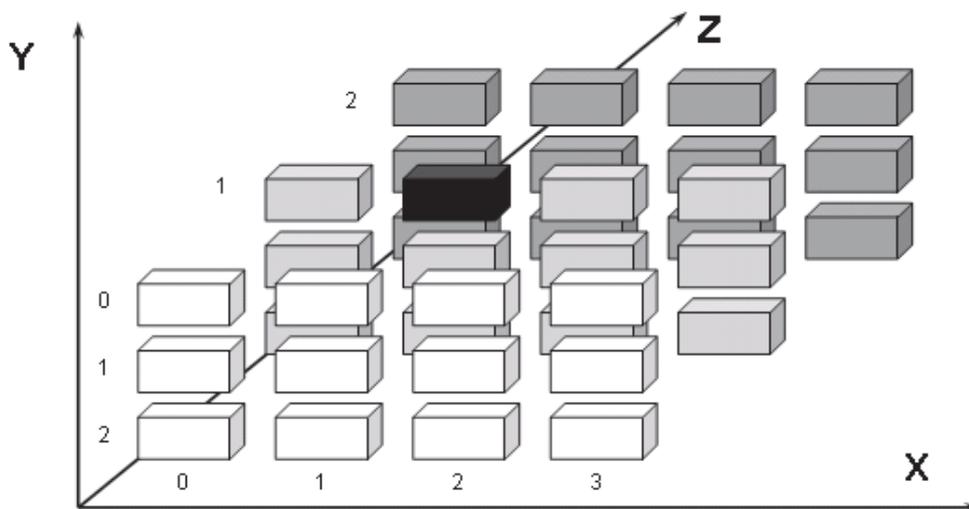
### Одномерный массив



### Двумерный массив



Трёхмерный массив



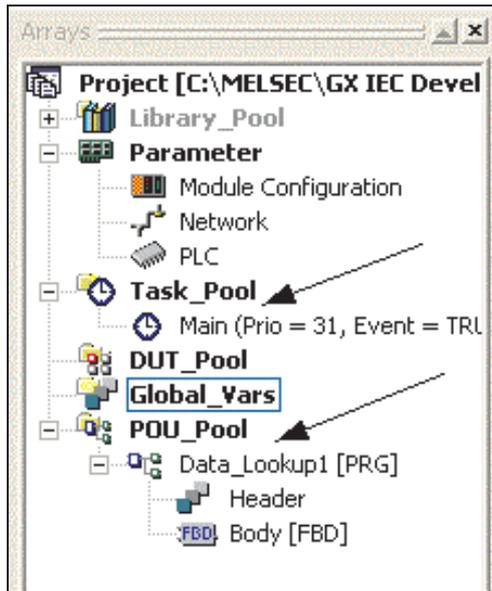
идентификатор            тип  
**Motor\_Current**        **ARRAY [0..3, 0..2, 0..2] OF INT**

 = **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.

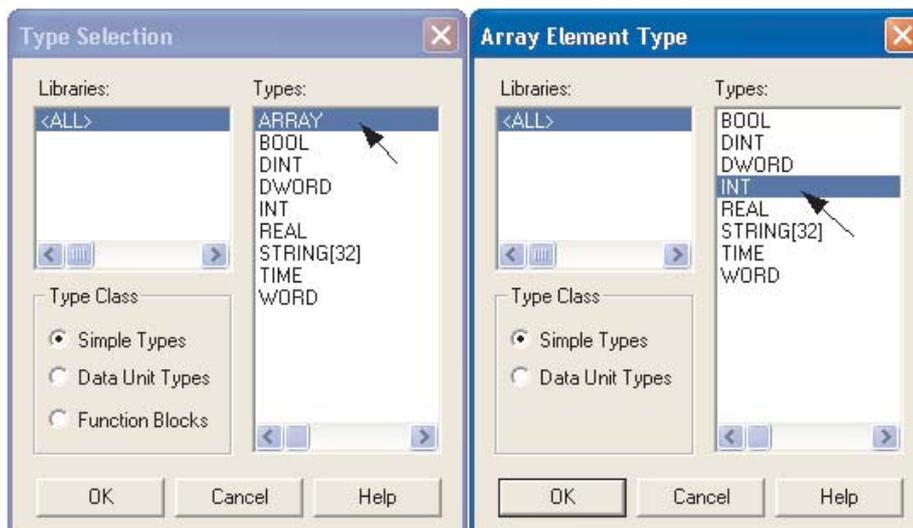
- ① Создайте новый проект и определите один новый программный модуль класса "Program", используя тело на языке **FBD** под названием "Data\_Lookup1"
- ② Создайте в задачном пуле новую задачу с именем "Main" и свяжите с ней программный модуль программы "Data\_Lookup1":



- 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

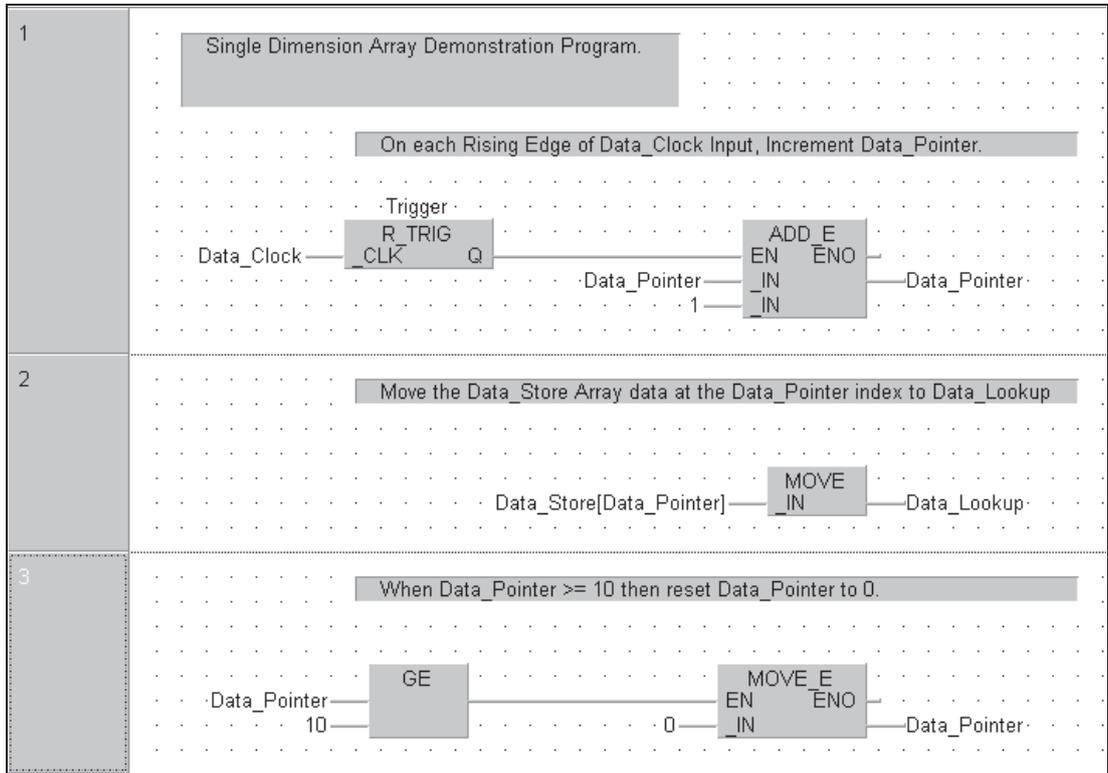
Тип переменной "Array" вводится следующим образом:



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:



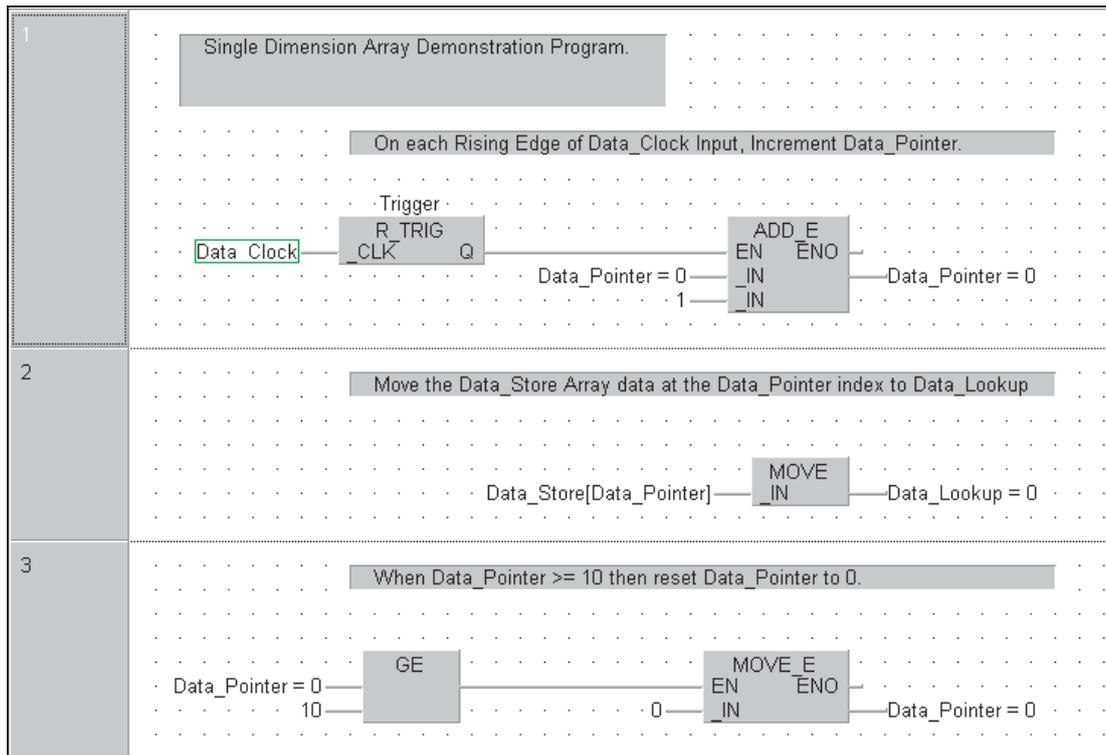
**ПРИМЕЧАНИЕ**

| 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...

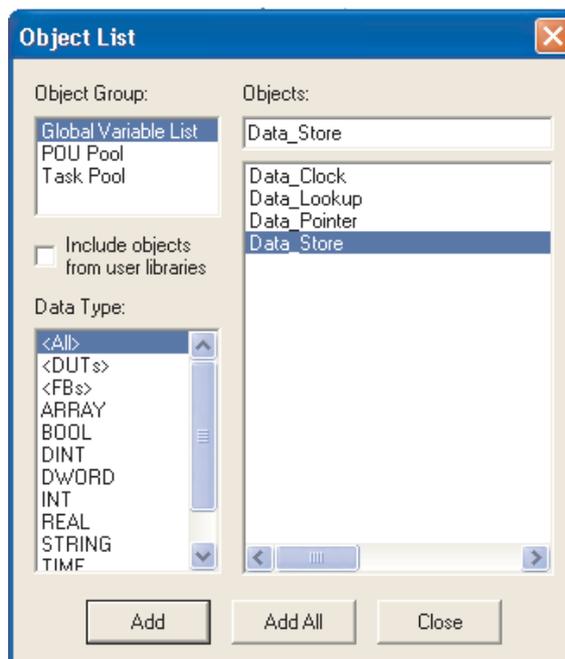
Read from File...

Write to File...

Setup...

Always on top

- Из результирующего окна выберите имя переменной "Data\_Store" и щелкните на **Add**:



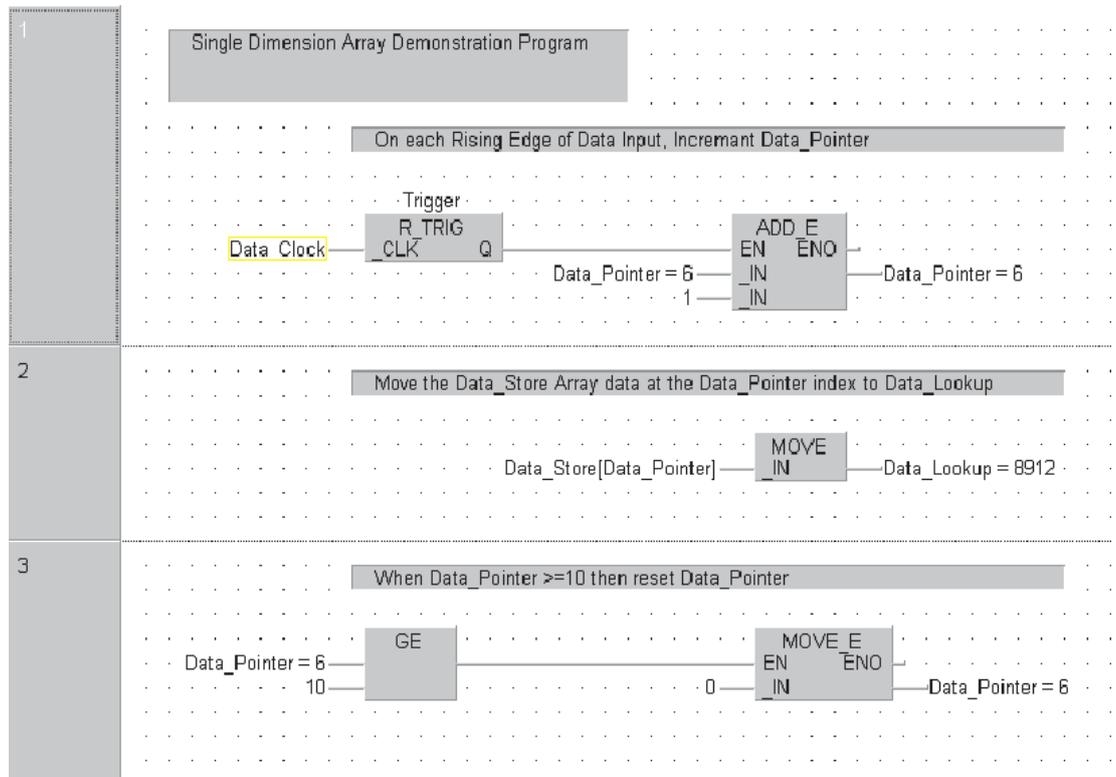
- Поскольку имя переменной "Data\_Store" является массивом, система представляет элемент с префиксом "+". Щелчок на имени переменной расширяет детали массива в таблицу, как показано:

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.

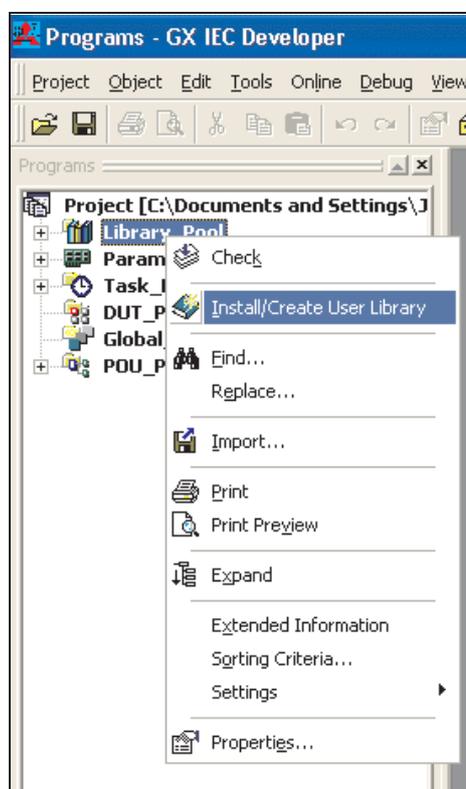
Как уже отмечалось, при вызове функций программы, **Стандартная библиотека** содержит функции IEC. **Библиотека изготовителя** содержит функции Mitsubishi functions (они обозначены как \*\_M), где M означает изготовитель (manufacturer), а не Mitsubishi!

Любые пользовательские библиотеки также появятся в этом списке.

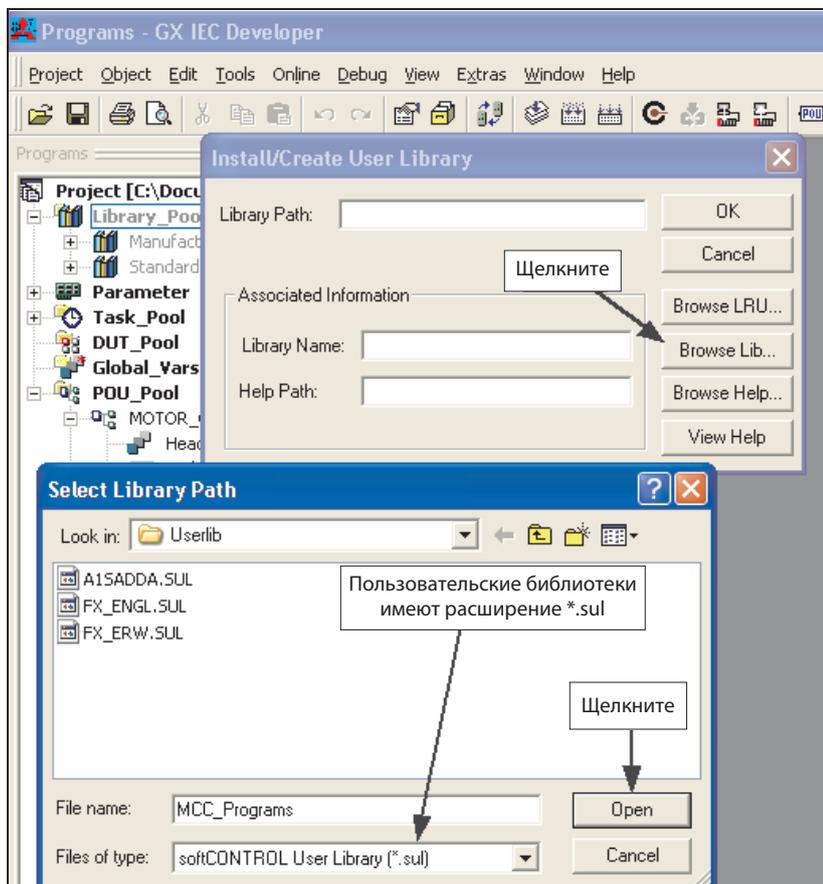
### 13.1.1 Пример - Создание новой библиотеки

Присвойте функциональный блок STAR\_DELTA новой библиотеке.

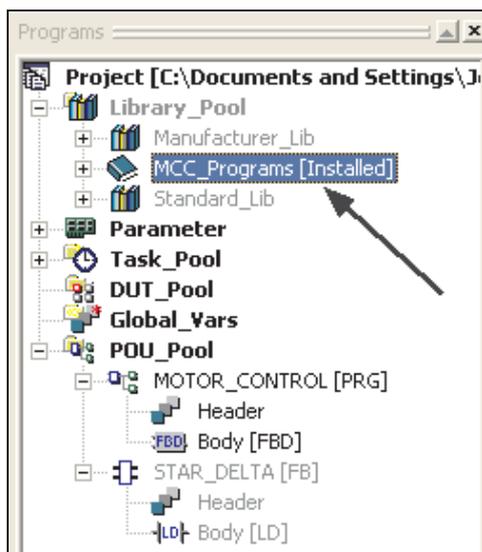
- Щелкните правой кнопкой на пуле библиотек в окне **Project Navigator** и из показанного меню выберите **User Library** и **Install/Create Library**.



- ② 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".



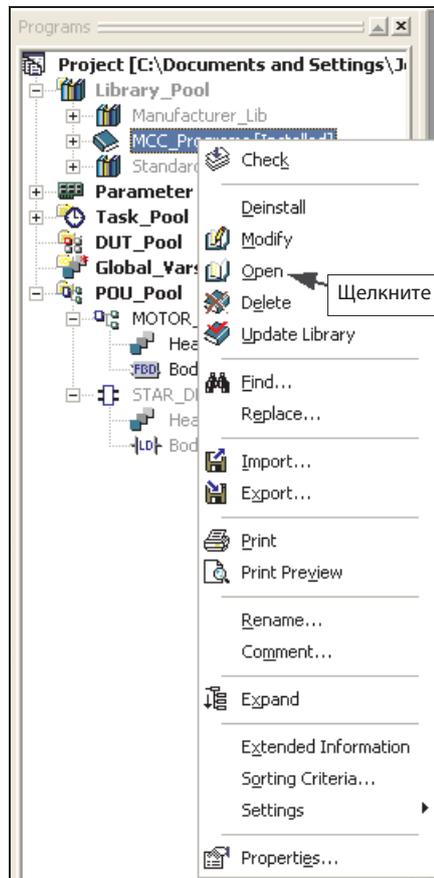
- ③ По завершению щелкните на **Open**:



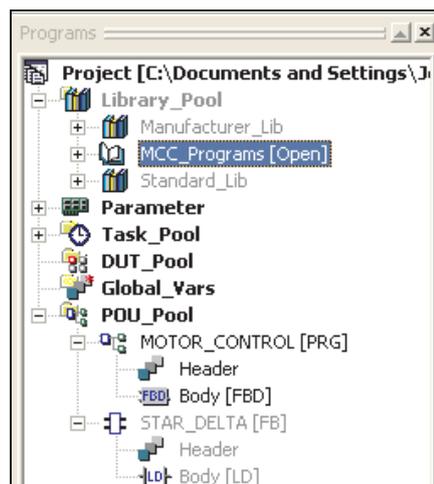
Обратите внимание, что теперь в пуле библиотек проекта представлена новая библиотека "MCC\_Programs".

### 13.1.2 Открытие библиотеки

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



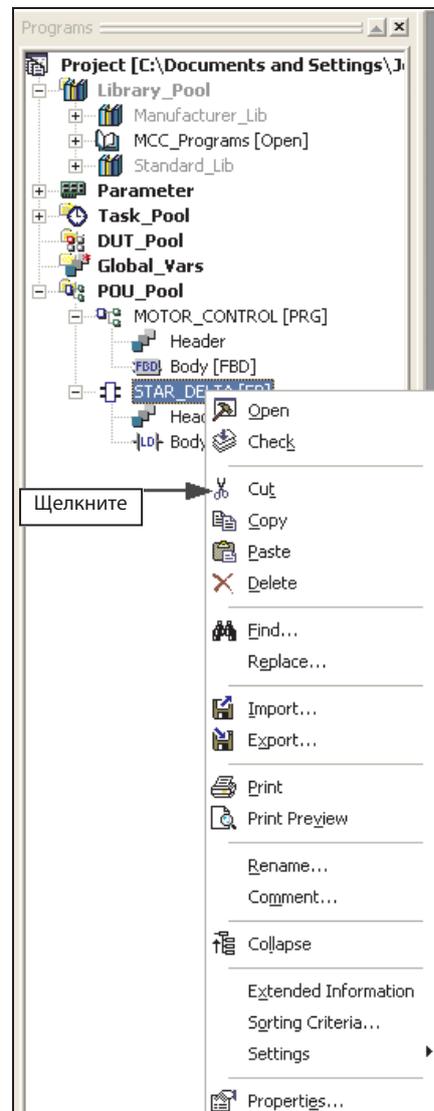
Библиотека теперь открыта; к ней можно обращаться и редактировать:



### 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**:

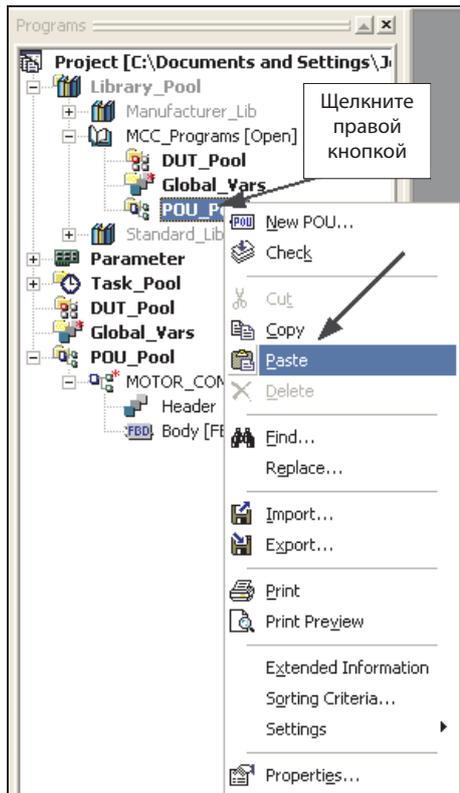


Появится следующее диалоговое окно:

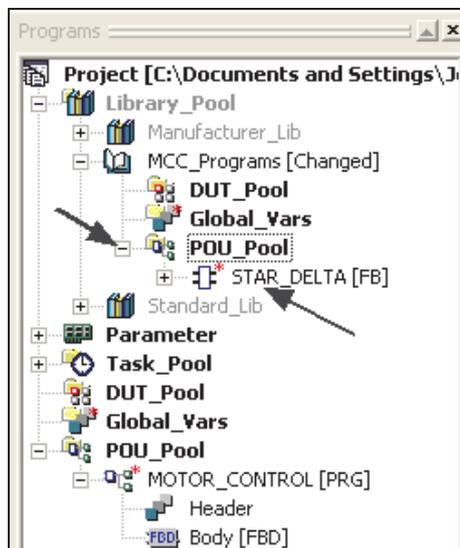


- Выберите **Yes**

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



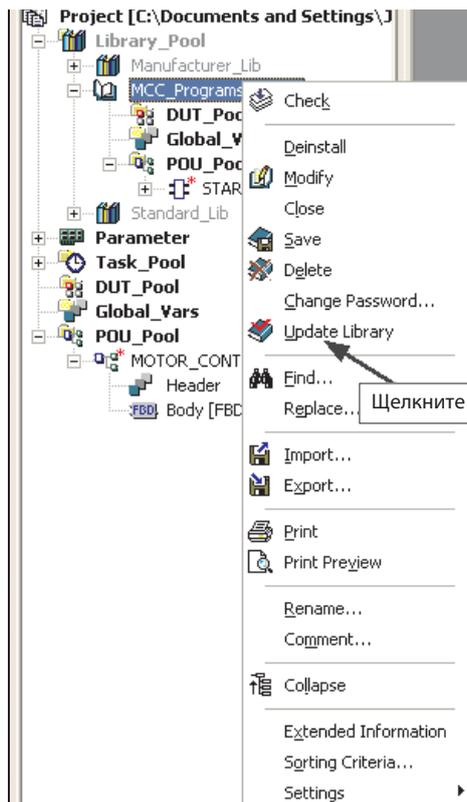
- ④ Щелкните на "+" на новом элементе в библиотечном пуле программных модулей, чтобы расширить функциональный блок "STAR\_DELTA":



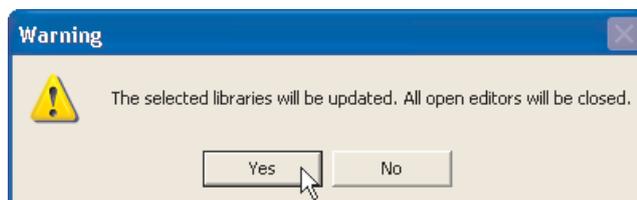
Сейчас программный модуль функционального блока "STAR\_DELTA" представлен в библиотеке "MCC\_Programs", а не в пуле программных модулей проекта.

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

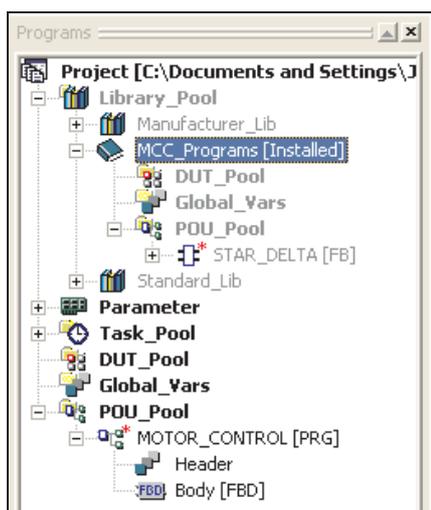
- ⑤ Завершив редактирование библиотеки, щелкните на **Update Library**. Это обновит и закроет библиотеку.



Будет показано следующее сообщение:



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

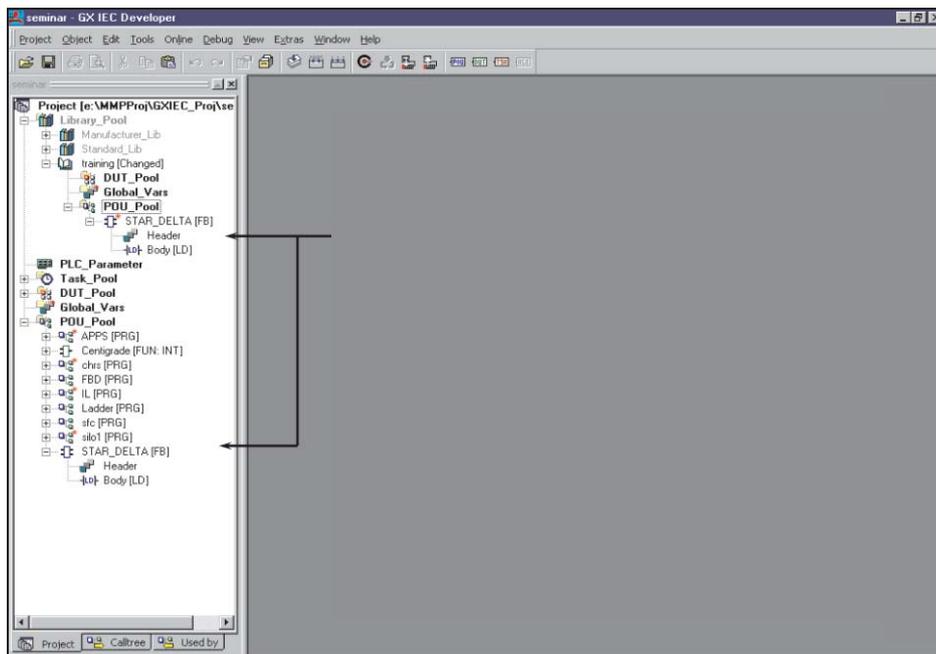


Теперь библиотека сохранена в каталоге по умолчанию "C:\MELSEC\GX IEC Developer 7.00\Userlib", как установлено при создании библиотеки.

## 13.2 Специальное замечание о библиотеках

ПРИМЕЧАНИЕ: Если библиотека создана в подкаталоге проекта, например E:\MMProj\GXIEC\_Proj\Seminar.sul, 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

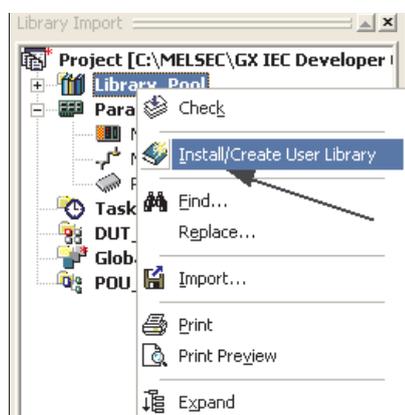
Когда созданы "Пользовательские библиотеки", можно повторно использовать подпрограммы, импортируя их в другие приложения. Компания Mitsubishi Electric выпустила много библиотек с часто используемыми подпрограммами. Например, интерфейсы к "интеллектуальным модулям" типа функциональных блоков А/Ц и Ц/А 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 Mitsubishi

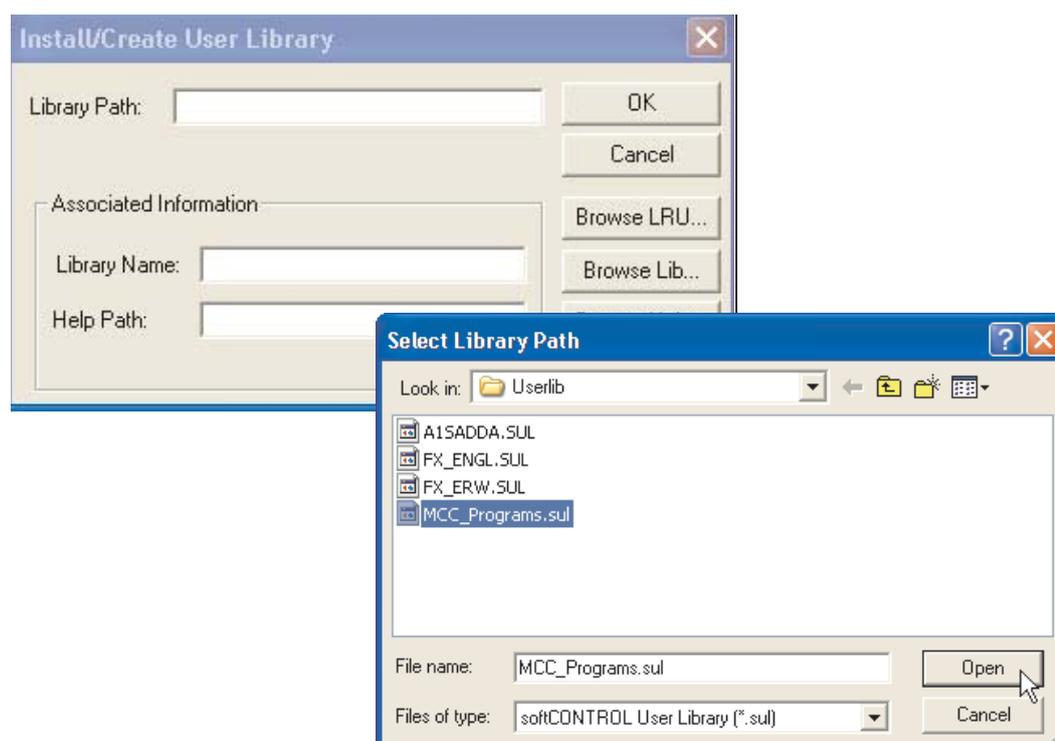
Предварительно сохраненная библиотека "**MCC\_Programs**" будет импортирована в текущий проект, и содержащийся в ней функциональный блок будет использован повторно.

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

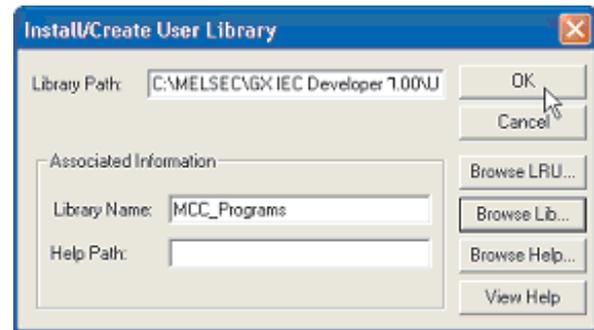


- 2 Щелкните правой кнопкой мыши на пуле библиотек, а затем в меню на **Install/Create User Library**.

- 3 Введите следующие детали в окно запроса:



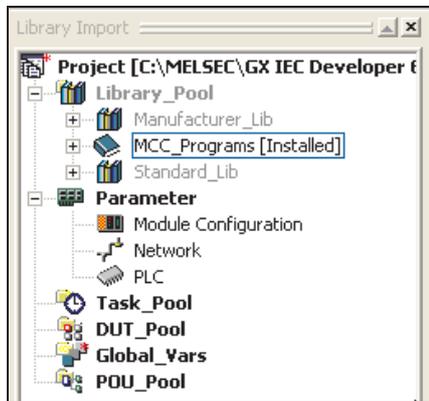
- ④ Затем щелкните на **OK**, чтобы принять элементы.



**ПРИМЕЧАНИЕ**

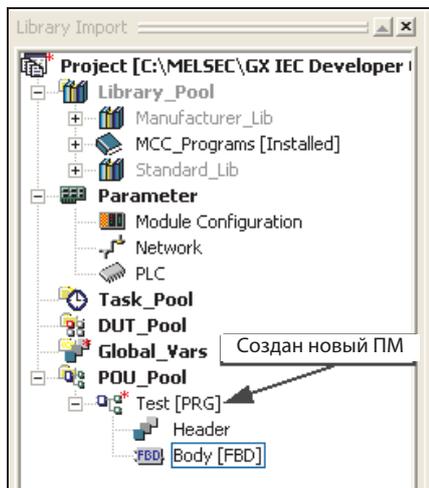
Опция Help Path используется для пользовательских справочных файлов, которые могут быть созданы, чтобы описать работу подпрограмм, содержащихся в библиотеке. Эти файлы можно создавать в MS-Word, например, в формате HTML, и вручную сохранить с расширением \*.СНМ. Эти файлы можно связать с библиотекой, 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:

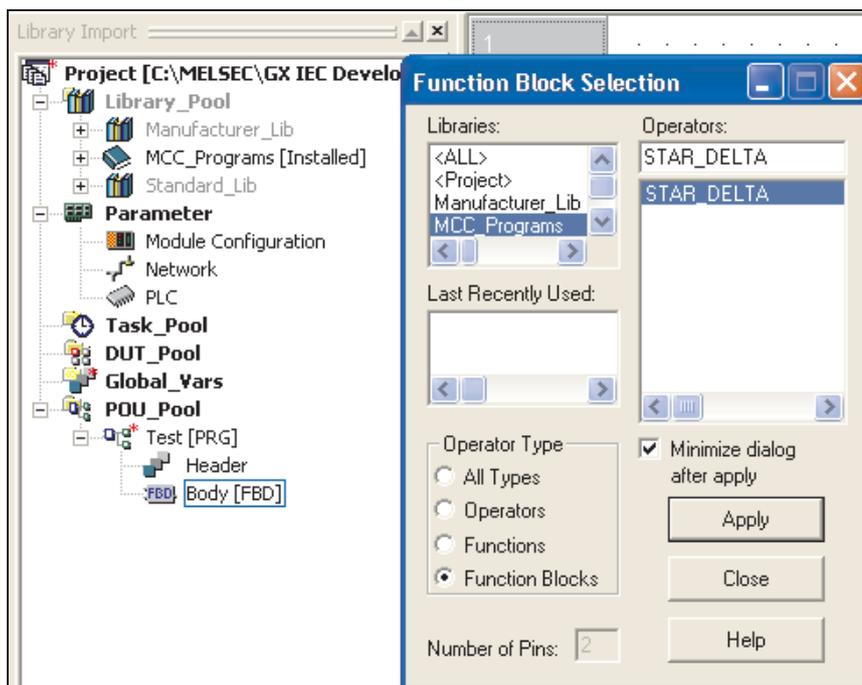


Компоненты, сохраненные в библиотеках, могут легко вызываться и выбираться в проекте, как показано на следующих иллюстрациях:

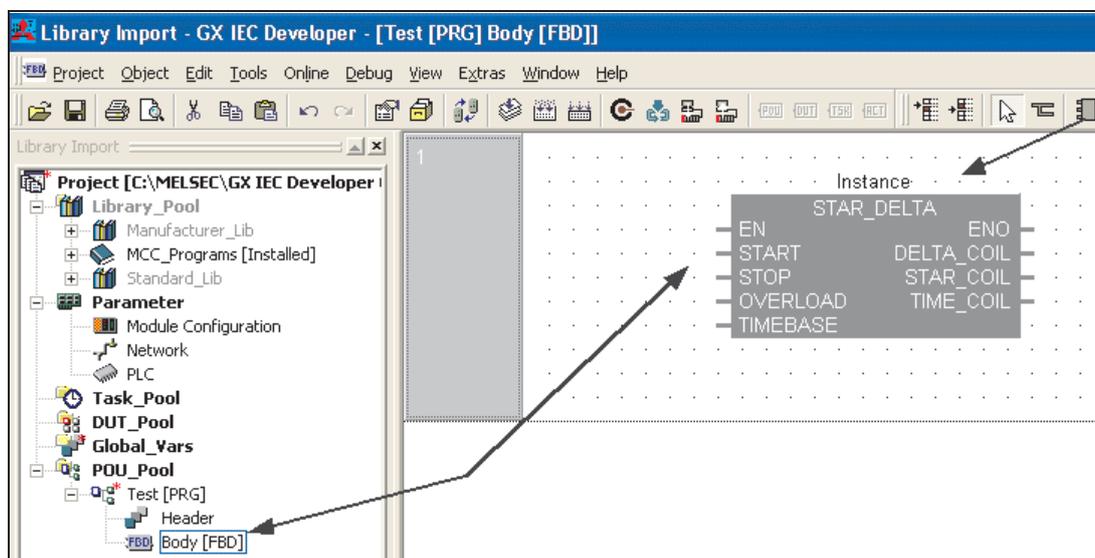
- ① Создайте новый программный модуль, тип **FBD** с названием **"Test"**:



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



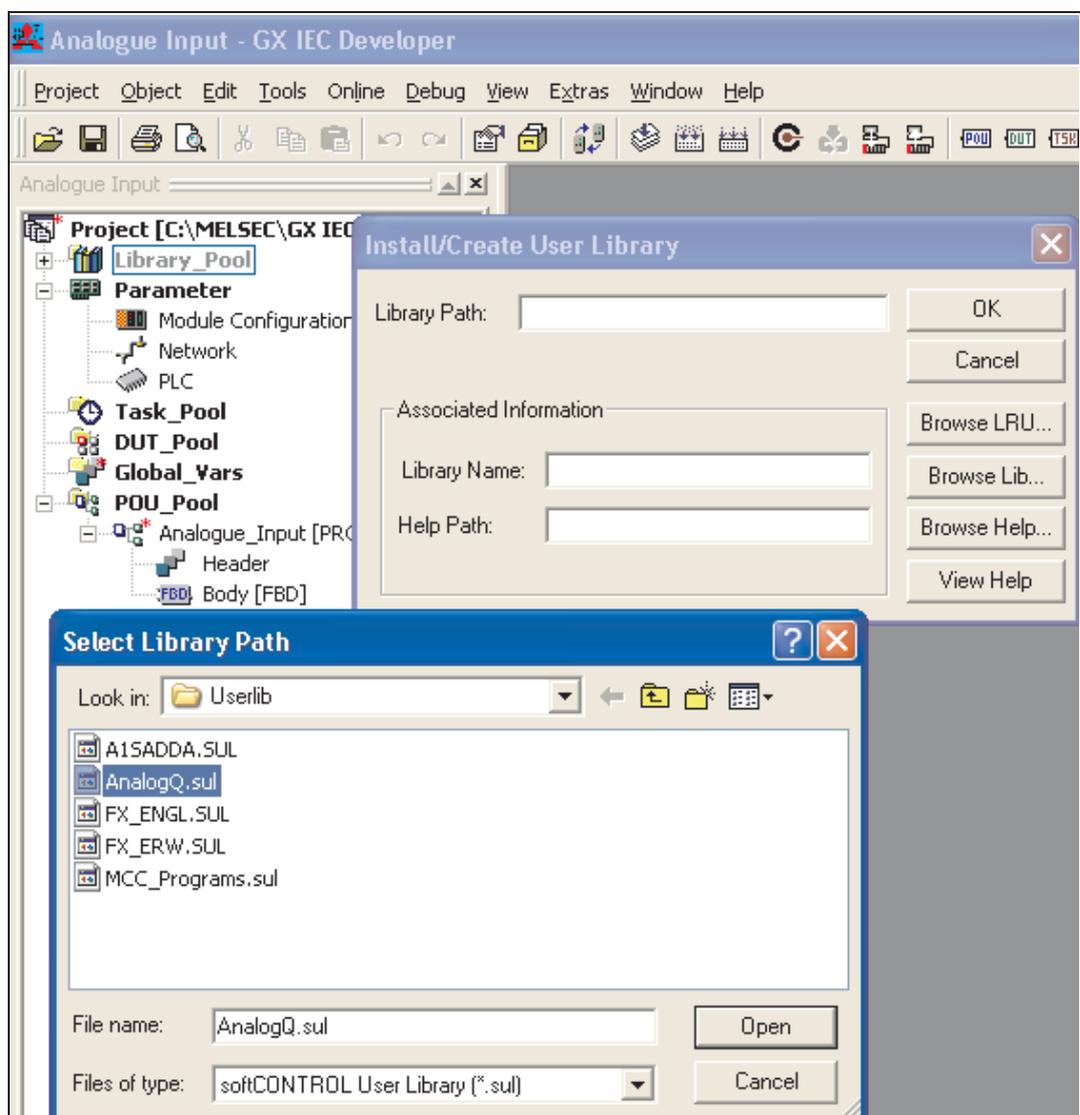
Как можно видеть, в области появляется новая библиотека; ее можно выбрать, как показано ниже:



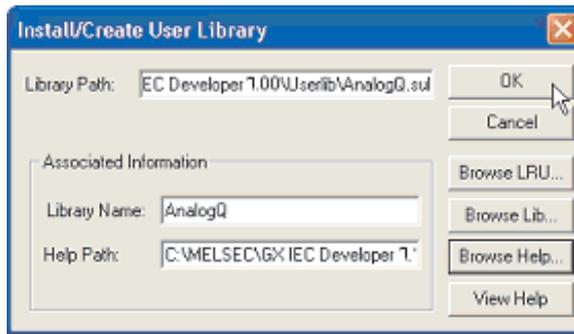
### 13.3.2 Пример: Импорт функционального блока из библиотеки Mitsubishi

В этом примере функциональный блок для чтения аналоговых значений аналогового входного модуля Q64AD импортируется в текущий проект. Этот функциональный блок предоставляет Mitsubishi, для чего в проект должна быть загружена библиотека "AnalogQ". Эту библиотеку можно скачать с сайта Mitsubishi ([www.mitsubishi-automation.ru](http://www.mitsubishi-automation.ru)) или установить с помощью установочной утилиты с компактдиска со средой программирования GX IEC Developer. После этого библиотека используется как обычная пользовательская библиотека.

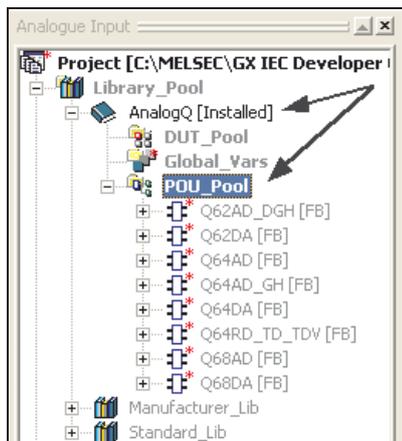
- ① Создайте новый пустой проект без программных модулей, называемый "Analogue\_Demo".
- ② Создайте новый программный модуль (Type: **FBD**, Class: **PRG**) и назовите его "Analogue\_Input"
- ③ Щелкните правой кнопкой на значке библиотечного пула **Library\_Pool** и выберите **Browse Lib**. Выберите файл библиотеки "AnalogQ.sul" и щелкните на **Open**.



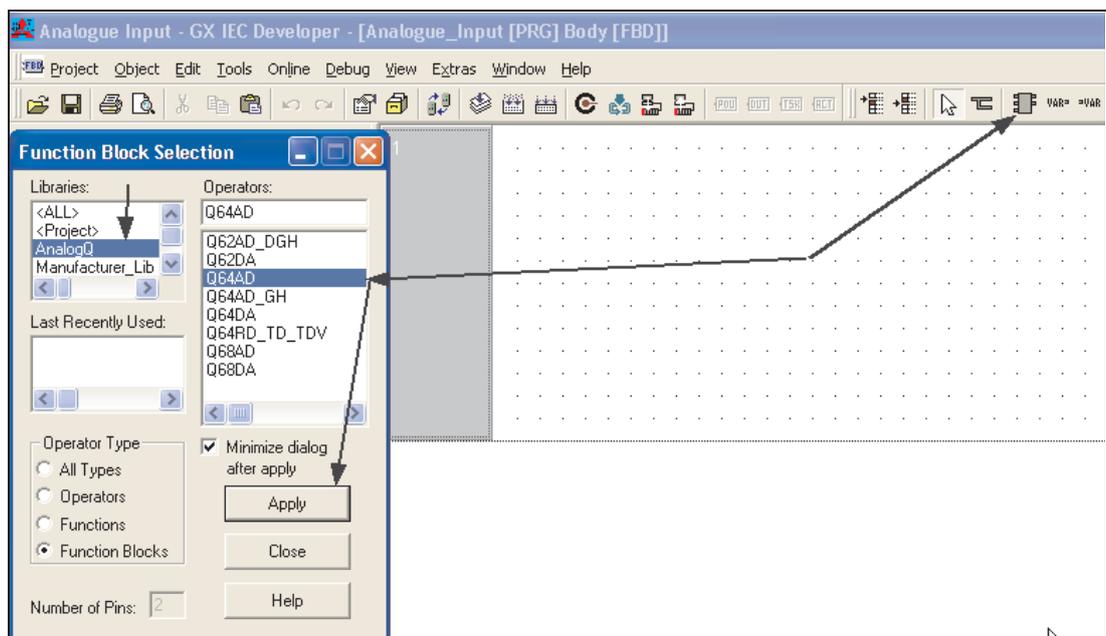
- ④ Щелкните на кнопке OK в окне запроса **Install/Create User Library**:



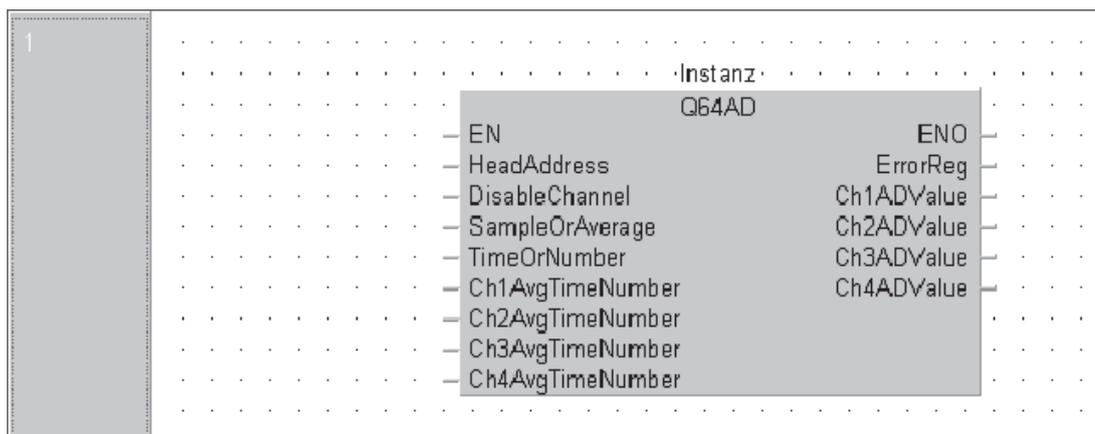
Видим, что в навигационном окне проекта появилась новая библиотека "AnalogQ".



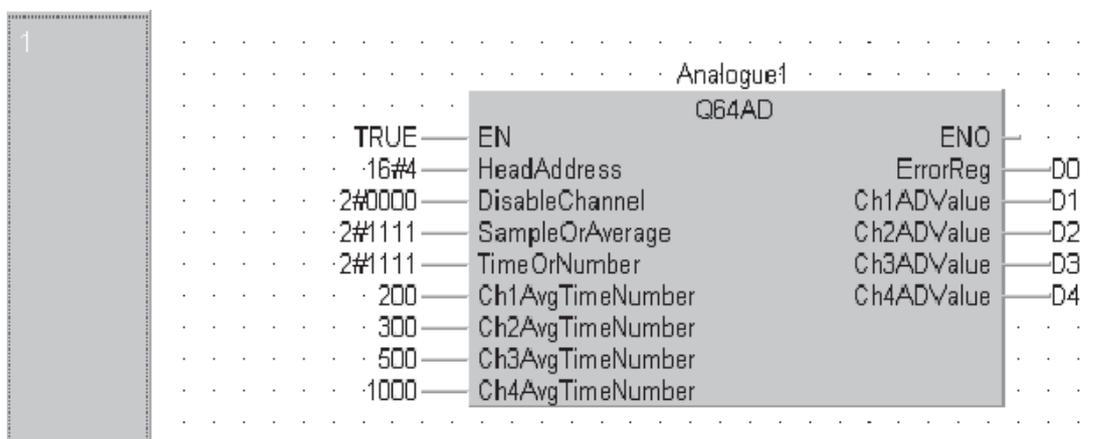
- ⑤ Создайте новую задачу в задачном пуле: "MAIN" и свяжите с ней программный модуль "Analogue\_Input".
- ⑥ Поместите функциональный блок Q64AD в программный модуль, как показано ниже:



Функциональный блок будет иметь следующий вид:



⑦ Определите все переменные, как показано ниже:



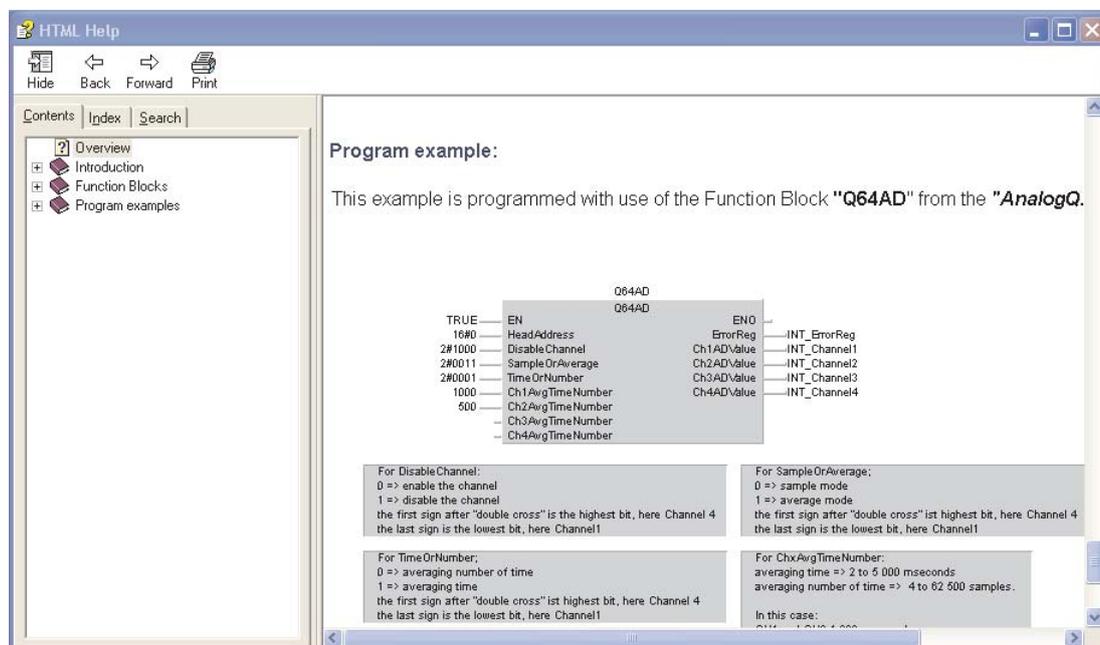
⑧ Компилируйте и загрузите программу в ПЛК.

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

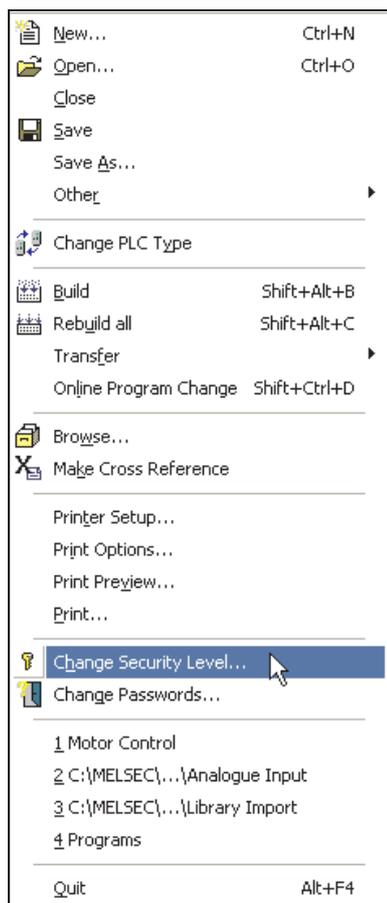


Можно ввести пароли и изменить уровни безопасности, используя эти окна, через меню **Project**.

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**.

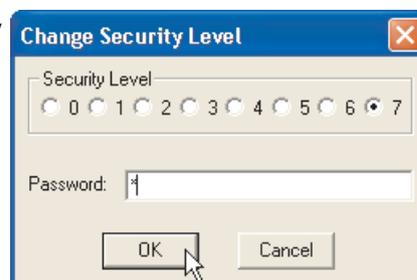


### 14.1.2 Изменение уровня безопасности



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.



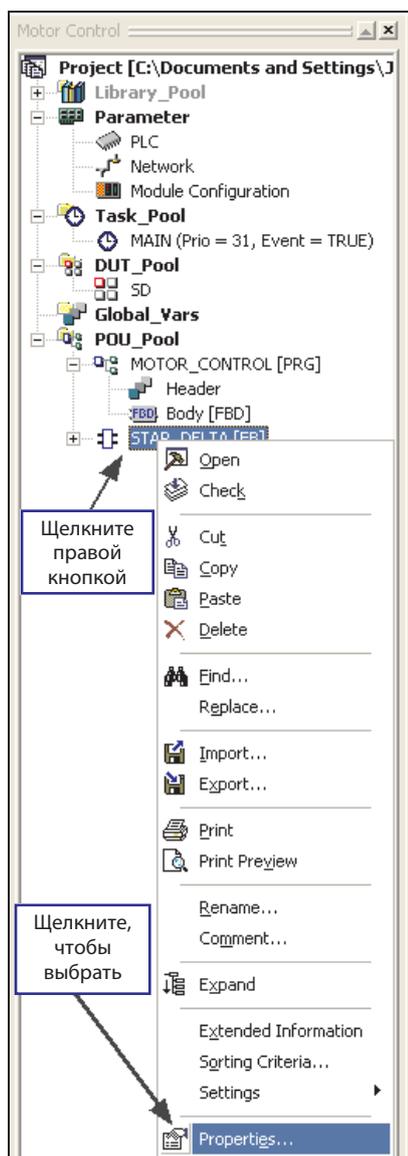
Войдя в систему, можно изменить атрибуты безопасности для многих компонентов. Одной из наиболее распространенных опций безопасности является изменение доступа к ПМ, например, пользовательским функциям и функциональным блокам.

### 14.1.3 Модификация парольного доступа к программному модулю

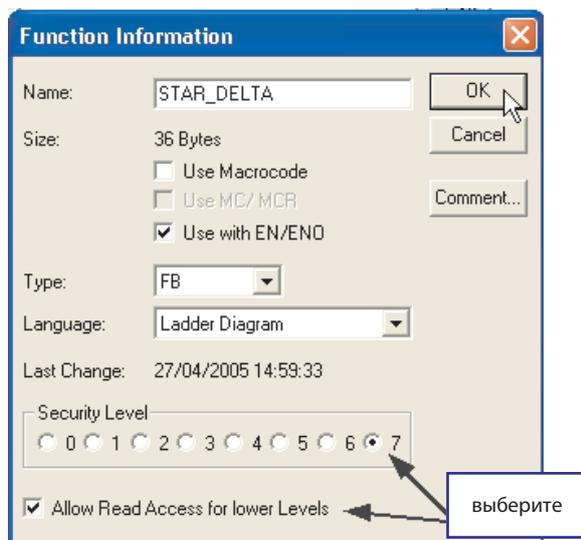
Чтобы защитить содержимое или управлять доступом к пользовательским программным модулям, можно следующим образом настроить атрибуты безопасности, зарегистрировавшись на текущем уровне безопасности:

#### Настройка уровня безопасности

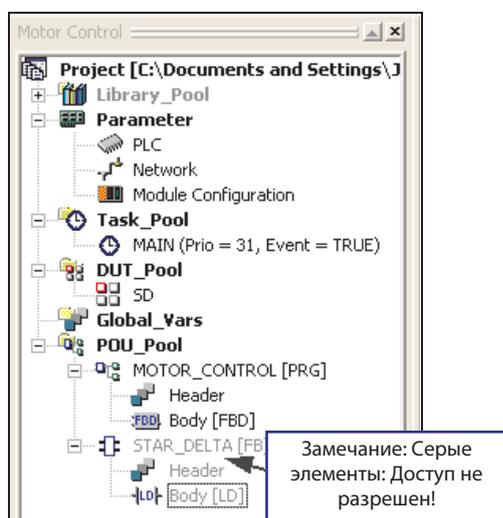
- 1 Откройте проект "Motor Control" и откройте заголовок функционального блока "STAR\_DELTA":



- 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:



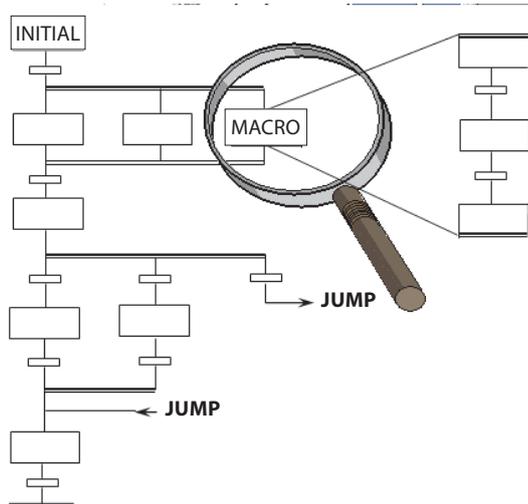
- ③ Измените уровень безопасности на "0" и получите доступ к заголовку и телу функционального блока "STAR\_DELTA". Будет разрешен только доступ для чтения с целью мониторинга, но любое изменение кода невозможно.
- ④ Снова зарегистрируйтесь на уровне 7 и измените атрибуты безопасности функционального 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

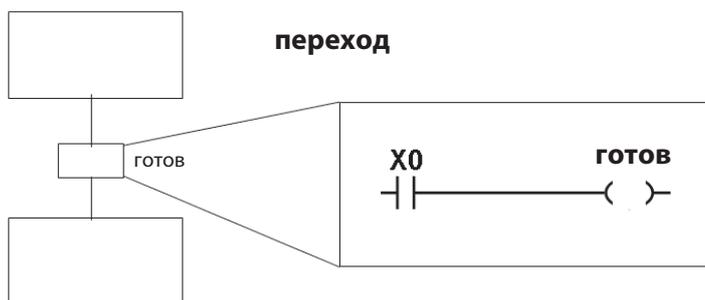
## 15.1 Что такое SFC?



- Редактор "Последовательных функциональных схем" является управляемым редактором.
- Представление графической блок-схемы.
- На основе французского Grafcet (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

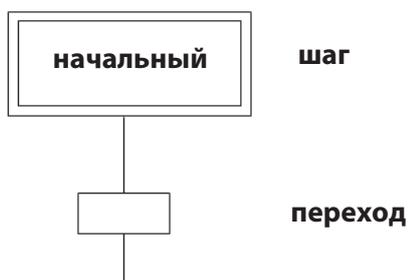
### 15.2.1 Переходы SFC



- Каждому переходу присваивается условие перехода (условие дальнейшего переключения). Если условие перехода выполнено, активируется очередной шаг.
- 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

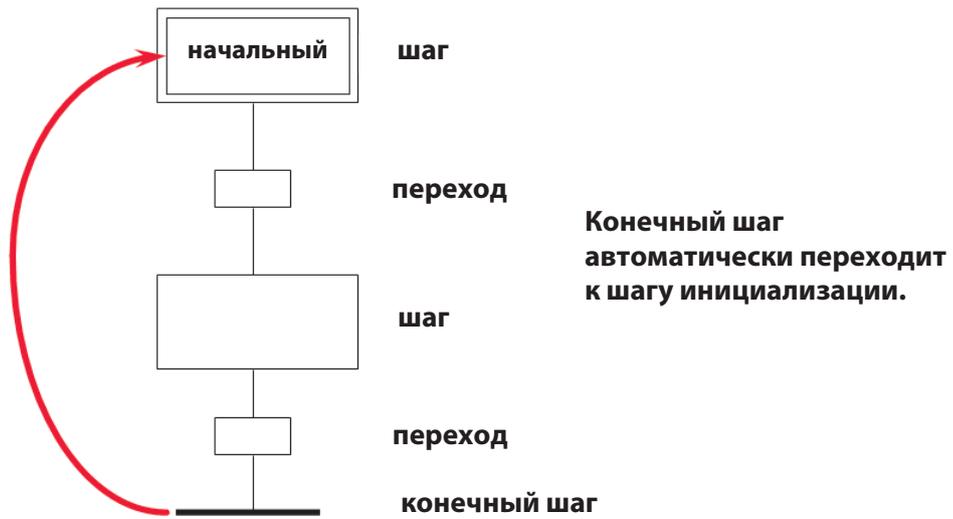
Программы SFC начинаются с функции начального шага, которая указывает начало последовательности:



### 15.2.3 Конечный шаг

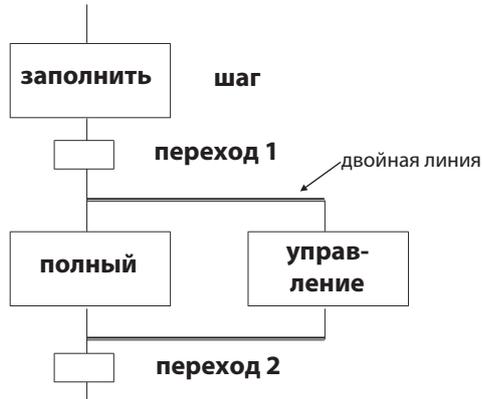
Все последовательности заканчиваются конечным шагом:



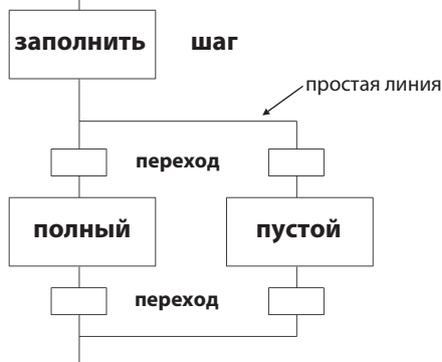


### 15.3 Примеры конфигурации SFC

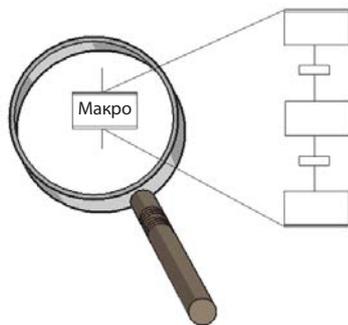
Параллельное ветвление



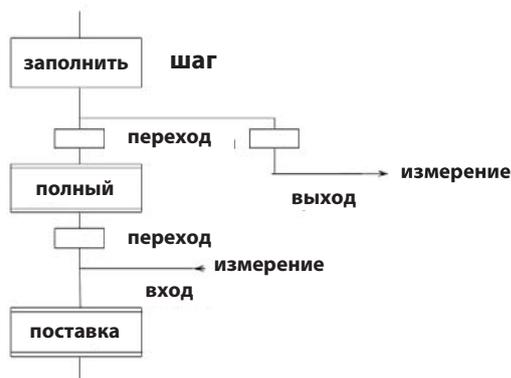
Селективное ветвление



Макрошаг

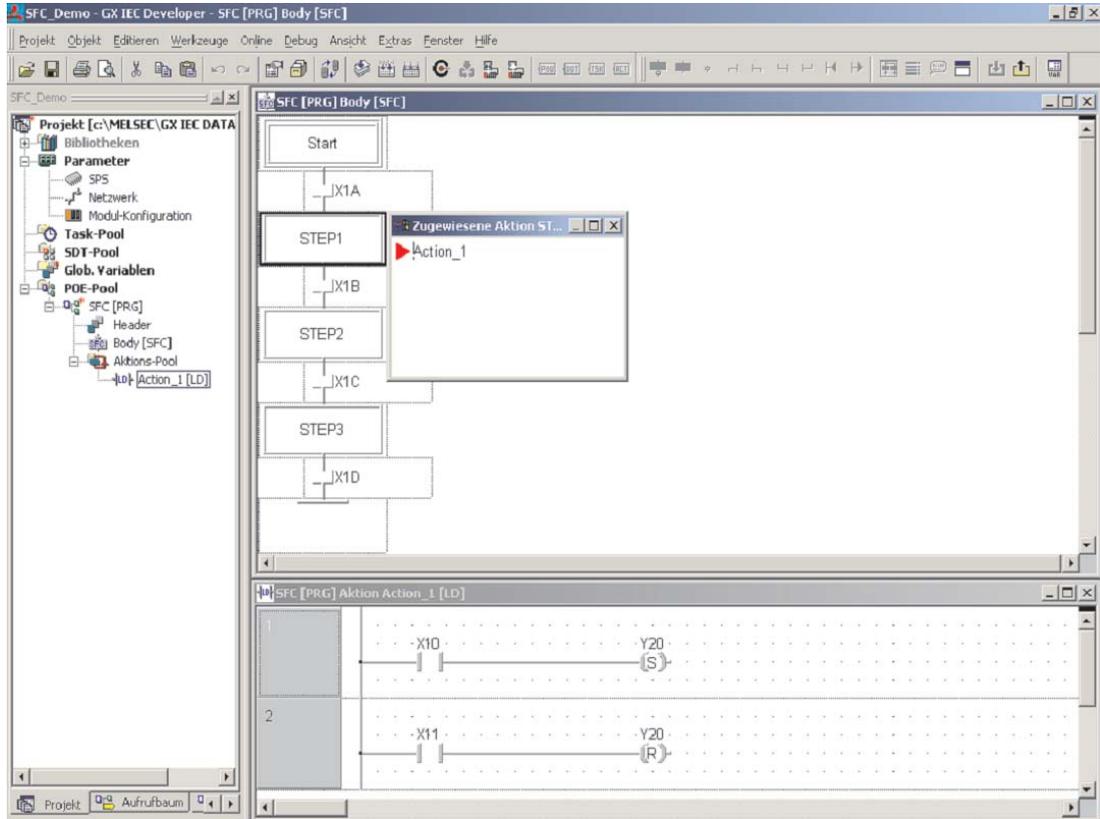


Выход

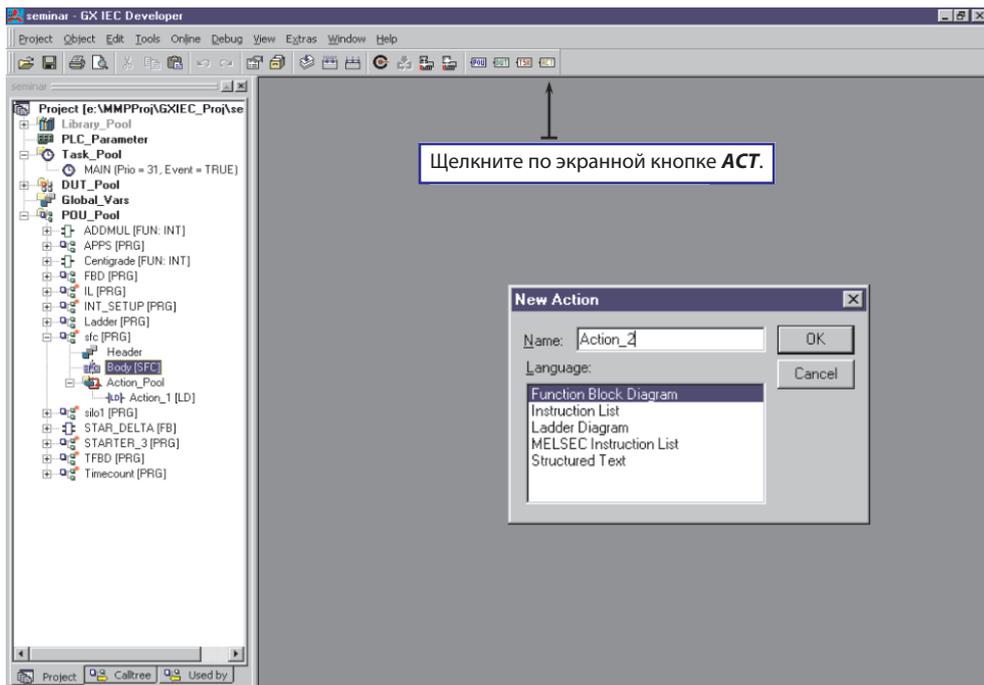


## 15.4 Действия SFC

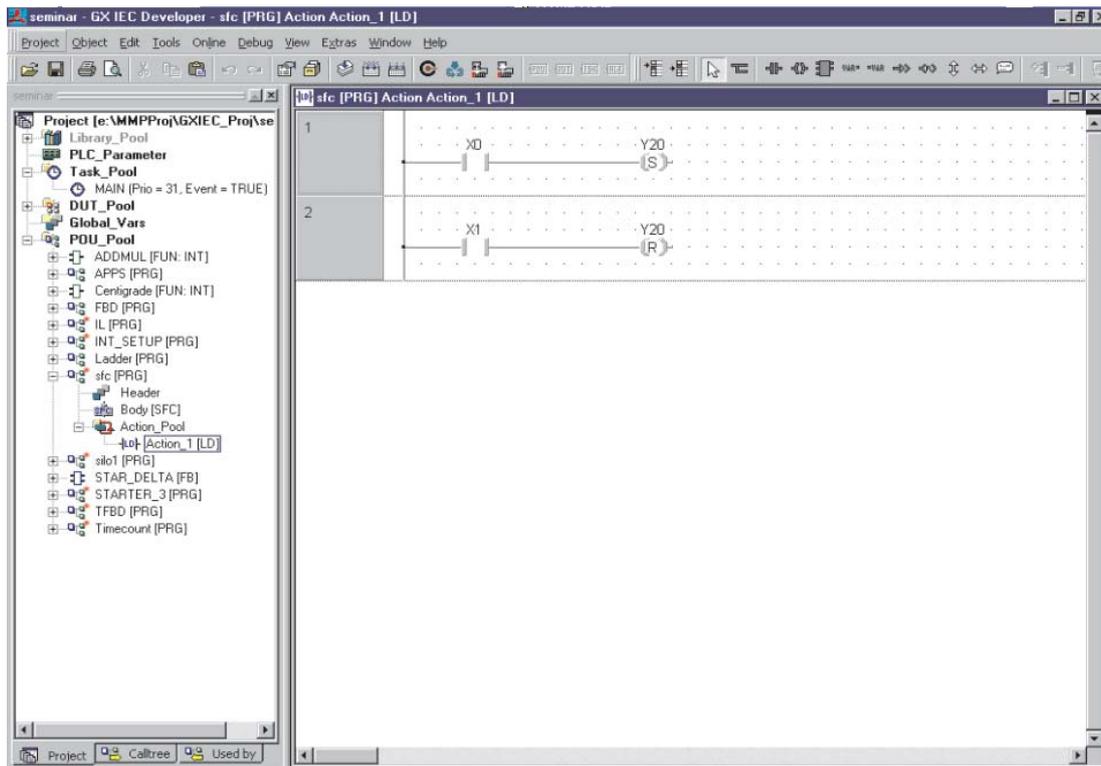
Каждый шаг имеет связанные действия. Действие является просто программой, как для программного модуля. Каждое действие имеет соответствующую логику, написанную на одном из языков IEC LD, IL, FBD или ST:



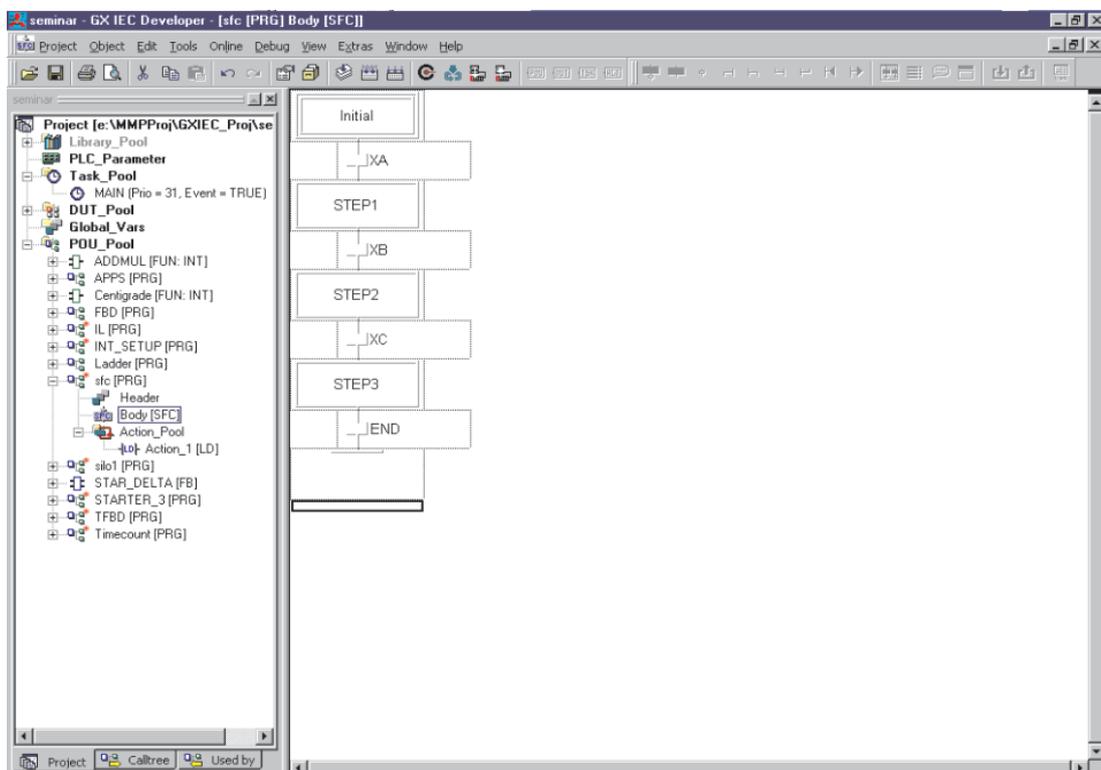
Чтобы создать новое действие, щелкните на кнопке **ACT** на панели инструментов. Выберите необходимый редактор, как для программного модуля:



Действия могут быть программами в их собственном смысле. Action\_1 может быть полной релейной подпрограммой блокировки, включающей много звеньев.

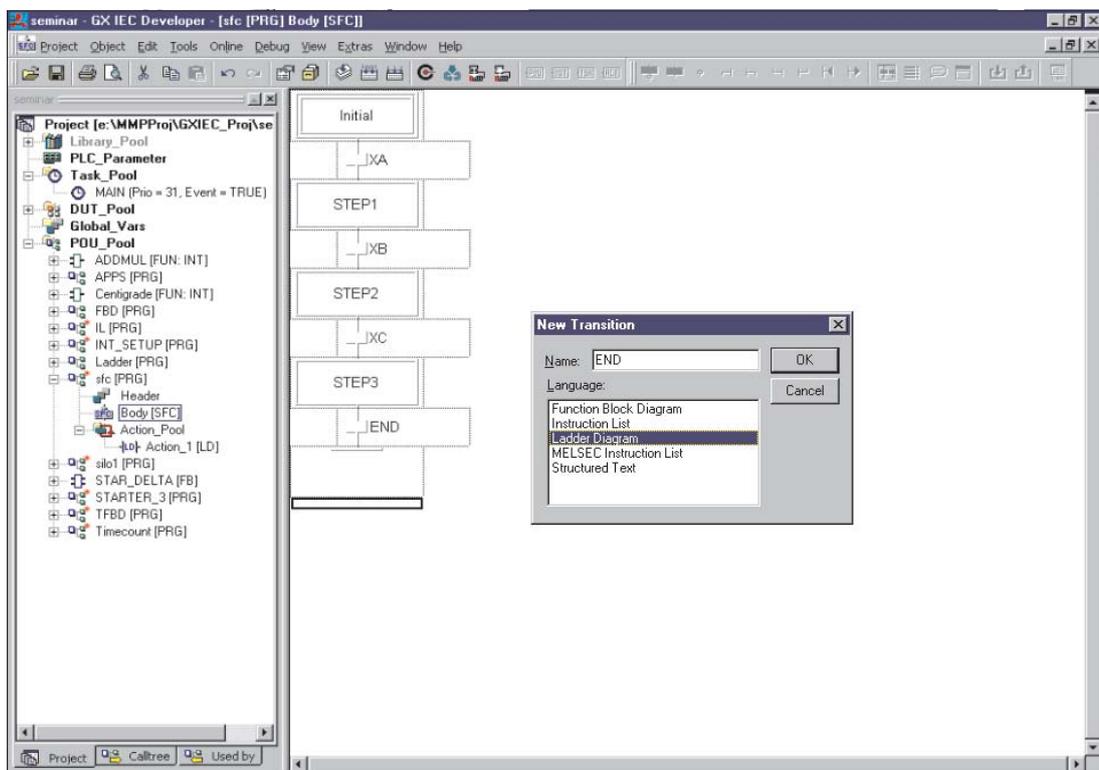


Каждый переход может быть простым операндом, например, адресом Mitsubishi XA, или именем идентификатора, или более сложным объектом, например, программой для одного звена, написанной на одном из языков IEC, IL, LD или FBD:

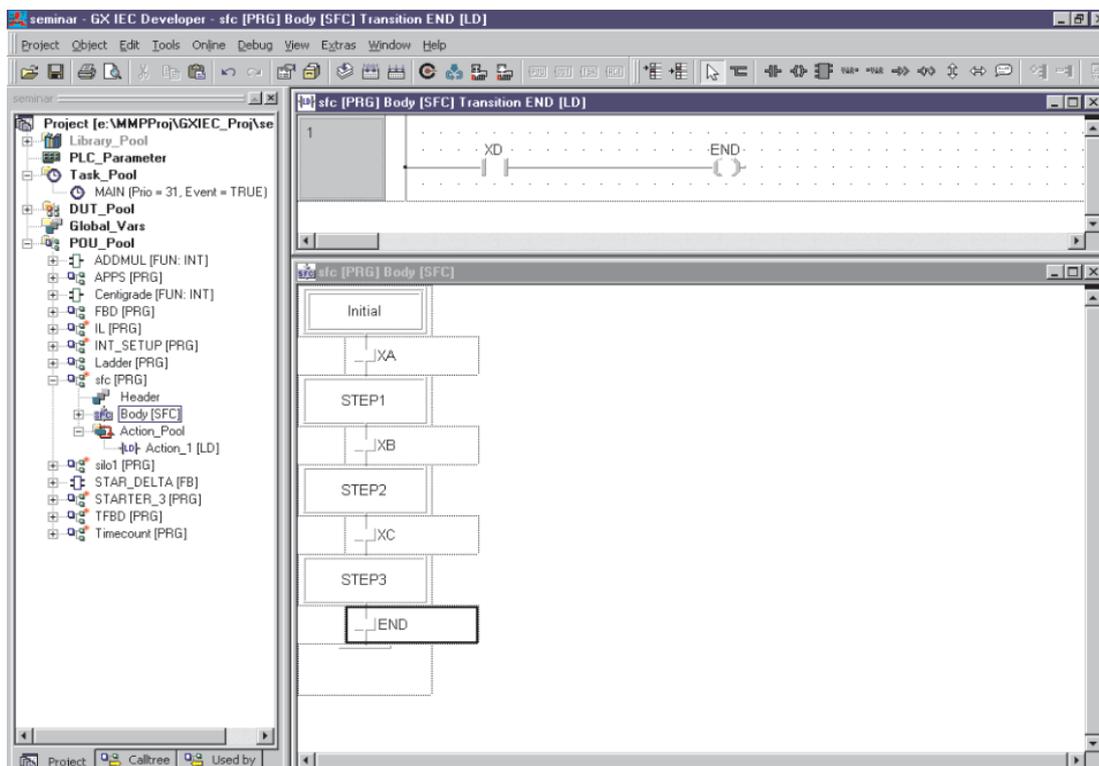


## 15.5 Сложные переходы

Для программирования сложного перехода введите имя перехода и нажмите клавишу ENTER. Выберите необходимый редактор, как для действия:

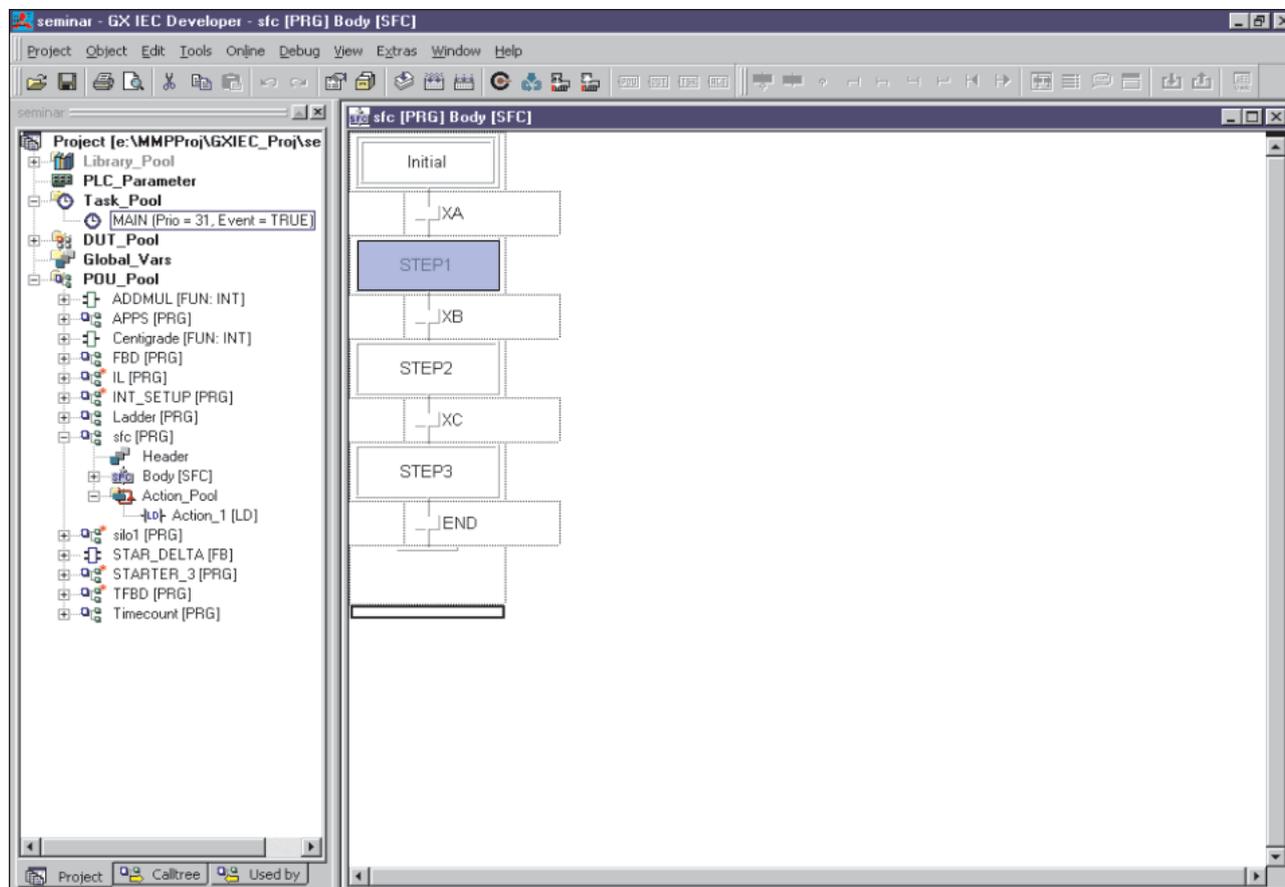


Переход может быть сложным выражением, но он состоит только из одного звена:



## 15.6 Экран программы SFC в режиме монитора

Одна из популярный 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:

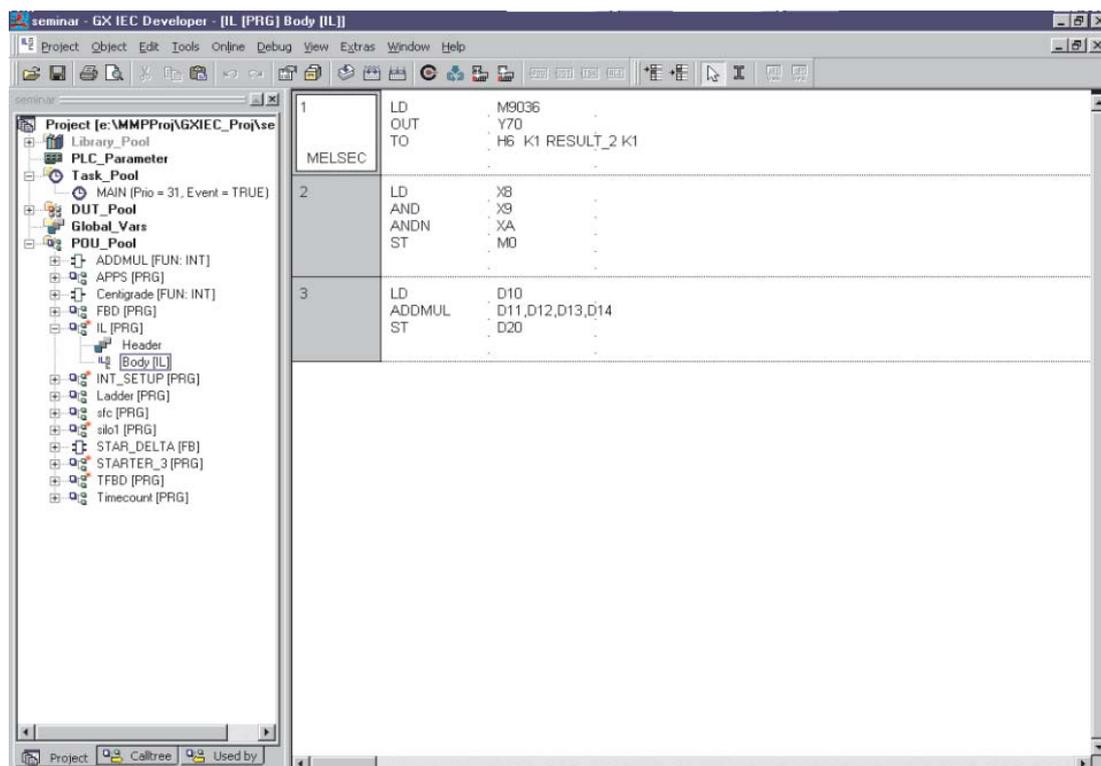
```
LD      X0
ADD_E   D0,D1,D2,50,D3
```

Причина заключается в том, что функция ADD\_E имеет возможность Enable Output (ENO).

## 16.2 Смешивание IEC IL и MELSEC IL в программных модулях

В POU могут содержаться схемы (звенья) на языке IL по IEC и схемы (звенья) на языке IL для MELSEC (MELSEC-IL).

Чтобы в POU, запрограммированный на языке IL по IEC, вставить схему на языке IL для MELSEC, щелкните слева по серой области схемы, чтобы пометить ее. Затем выберите в меню "Edit" пункт меню "New Network". В субменю щелкните по "Melsec Before". В результате этого схема на языке IL для MELSEC вставляется перед помеченной схемой.



# 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:

**IF .....THEN ..... 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

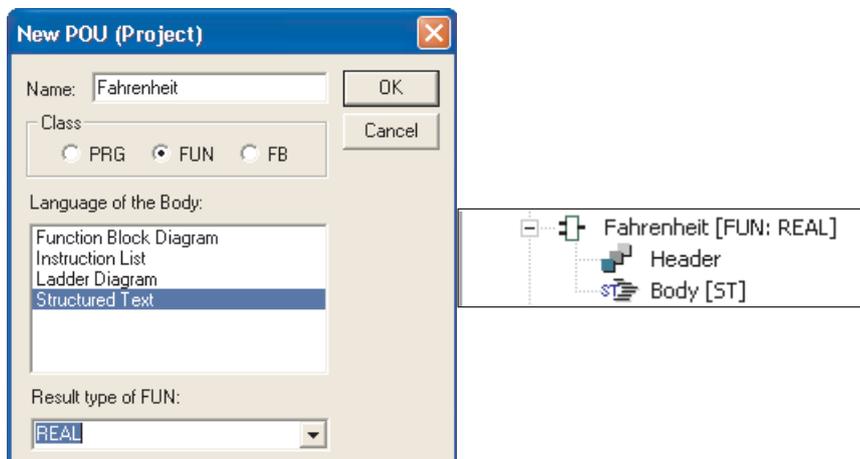
Мы создадим новый функциональный блок, преобразующий "Градусы Цельсия в градусы Фаренгейта", аналогичный использованному в предыдущем примере, чтобы проиллюстрировать использование редактора языка "структурированного текста".

Используется следующая формула:

$$\text{Fahrenheit} = \frac{\text{Celsius} \times 9}{5} + 32$$

Входная переменная и переменная результата будут в формате с плавающей запятой (REAL).

- 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	

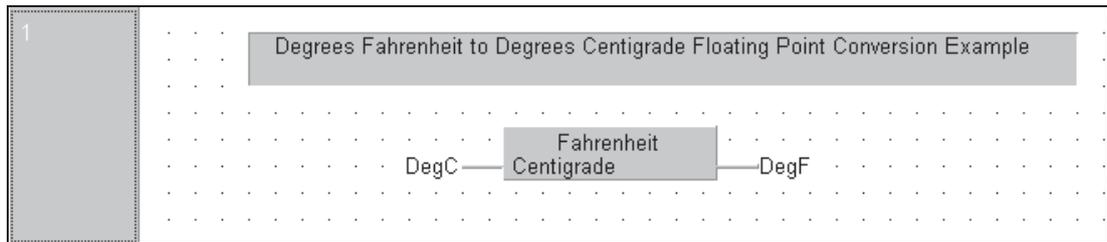
- ④ Откройте тело функции "Fahrenheit" и введите следующую простую программу на языке ST:

**Fahrenheit := (Centigrade\*9.0/5.0+32.0);**

- ⑤ Создайте новый программный модуль "Temp\_Conv", Class: **PRG**, Language: **Function Block Diagram**.



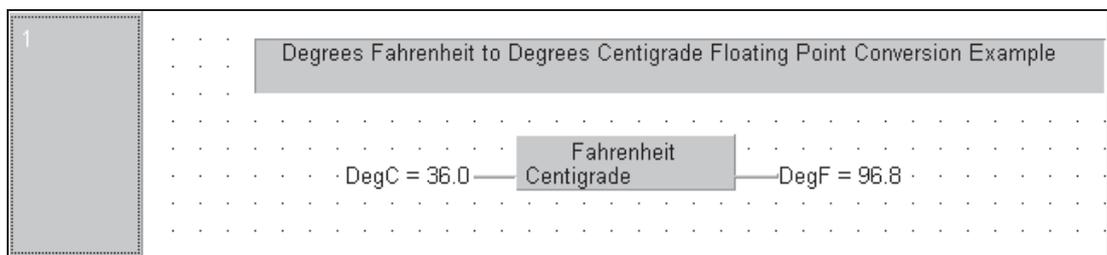
- ⑥ Откройте тело программного модуля программы "Temp\_Conv" и введите следующую иллюстрационную программу:



- 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 Ethernet

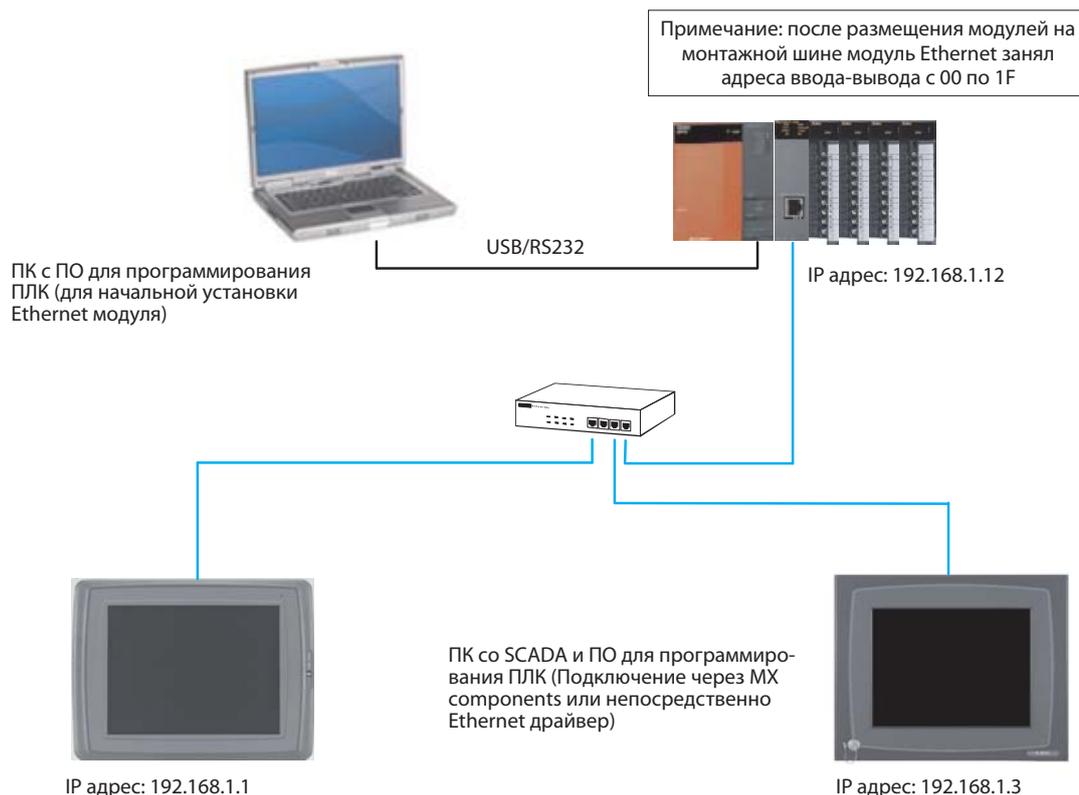
## 18.1 Конфигурирование параметров Ethernet модуля

В этом разделе поэтапно описывается конфигурирование модуля Ethernet QJ71E71 путем настройки параметров в GX IEC Developer (начиная с версии 7.00).

В качестве примера используется модуль Ethernet для коммуникации по протоколу TCP/IP между процессором Q02H и компьютером визуализации процесса с графической панелью управления типа E1071. Если в компьютере для визуализации процесса установлена среда программирования, то его можно использовать и для программирования контроллера. Поэтому в этом разделе разъясняется также, как с помощью среды GX IEC Developer через Ethernet обратиться к центральному процессору контроллера.

На рисунке ниже изображена конфигурация сети для этого примера. Указаны также используемые IP-адреса.

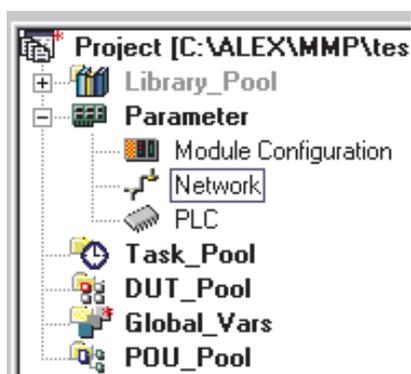
Пожалуйста, будьте снисходительны к тому, что настройки контроллера описаны подробнее, чем настройки компьютера или панели управления. У последних часто встречаются очень специфические настройки, которые не могут быть рассмотрены в рамках этого руководства.



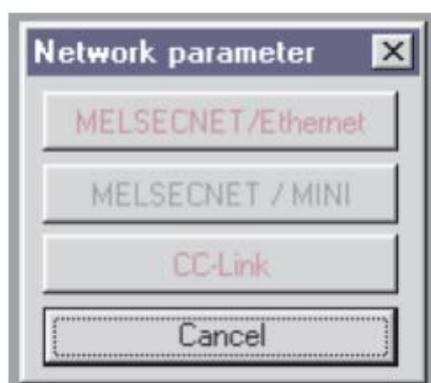
### 18.1.1 Конфигурирование ПЛК (с использованием ПК для начальной настройки)

При такой конфигурации предполагается, что среда программирования используется на основе стандартных настроек.

- ① Используя программное обеспечение для программирования, вызовите окно выбора параметров **Network Parameter**, дважды щелкнув на опции, отмеченной стрелкой.



- ② Когда откроется окно, выберите **MELSECNET/Ethernet**, как показано ниже.



После этого откроется диалоговое окно для конфигурации модуля Ethernet, как показано ниже.

- ③ 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)
Network No.	MNET/I O mode (Control station)
Total stations	MNET/I O mode (Normal station)
Group No.	MNET/H Stand by station
Station No.	MNET/H(Remote master)
Mode	Ethernet

- ⑤ 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

- ⑥ Щелкните в полях ввода в верхней половине столбца и введите необходимые данные. На рисунке ниже изображены настройки для примера конфигурации со страницы 18-1.

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

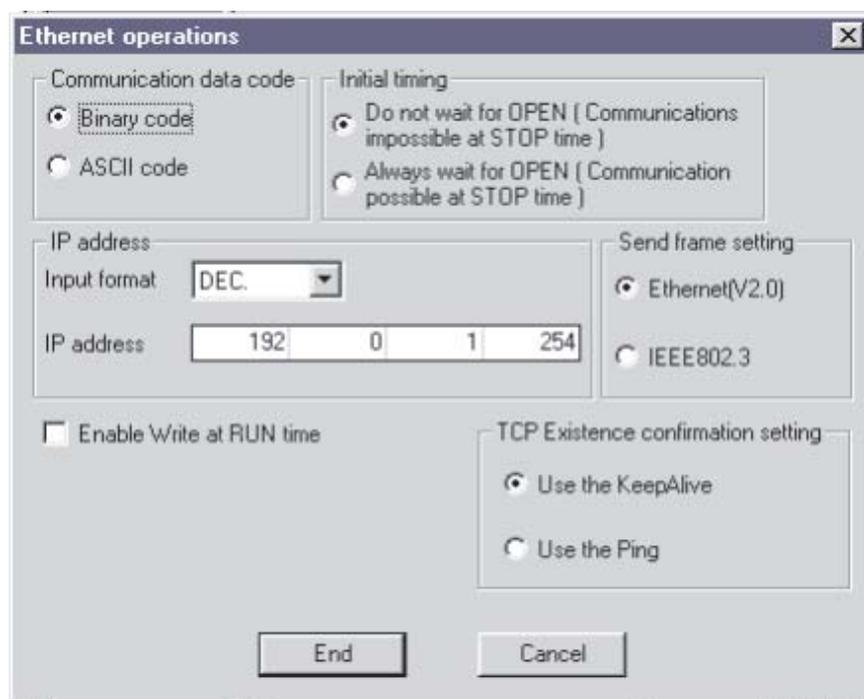
← См. замечание ниже

← См. замечание ниже

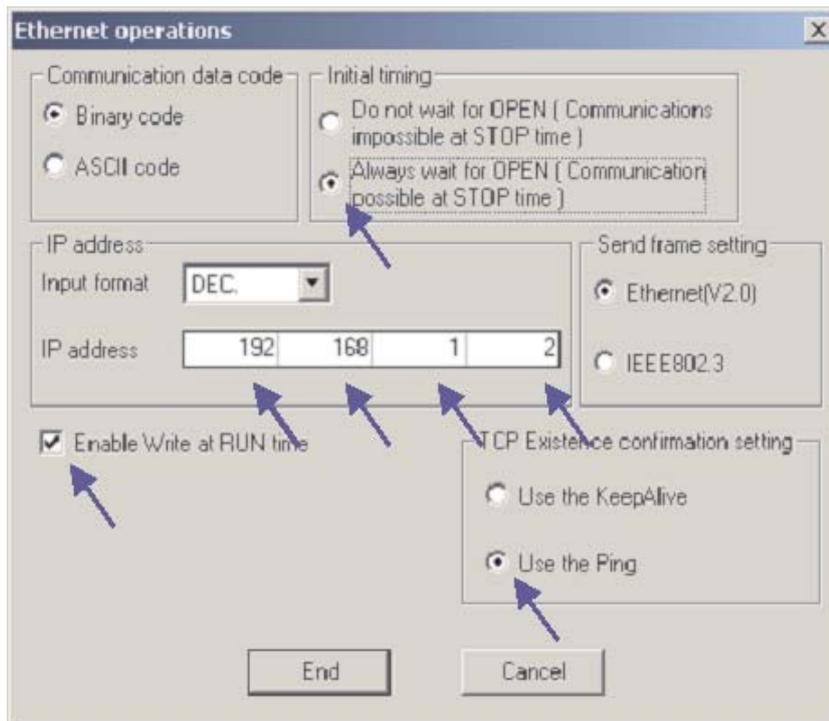
**ПРИМЕЧАНИЕ**

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.

- ⑦ Далее щелкните на **Operational settings**, чтобы вызвать показанное ниже диалоговое окно. Имеющиеся настройки – это настройки по умолчанию, применяемые в программном обеспечении для программирования.



- ⑧ Ниже в диалоговом окне показаны настройки, необходимые для описанного ранее примера системы. Для ясности различия указаны стрелками.



- ⑨ 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

- ⑩ 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.

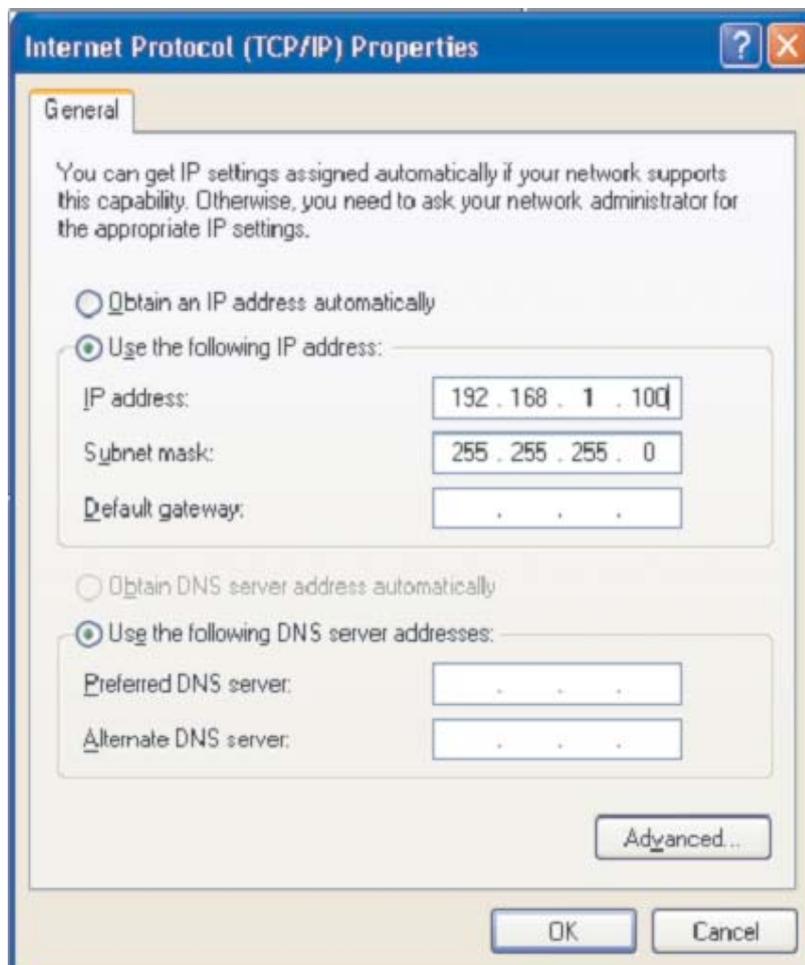
	Protocol	Open system	Fixed buffer	Fixed buffer communication procedure	Pairing open	Existence confirmation	Host station Port No.	Transmission target device IP address	Transmission target device Port No.
1	TCP	Unpassive	Receive	Procedure exist	Disable	Confirm	0401		
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

напр., HMI



## 18.2 Конфигурирование ПК для Ethernet

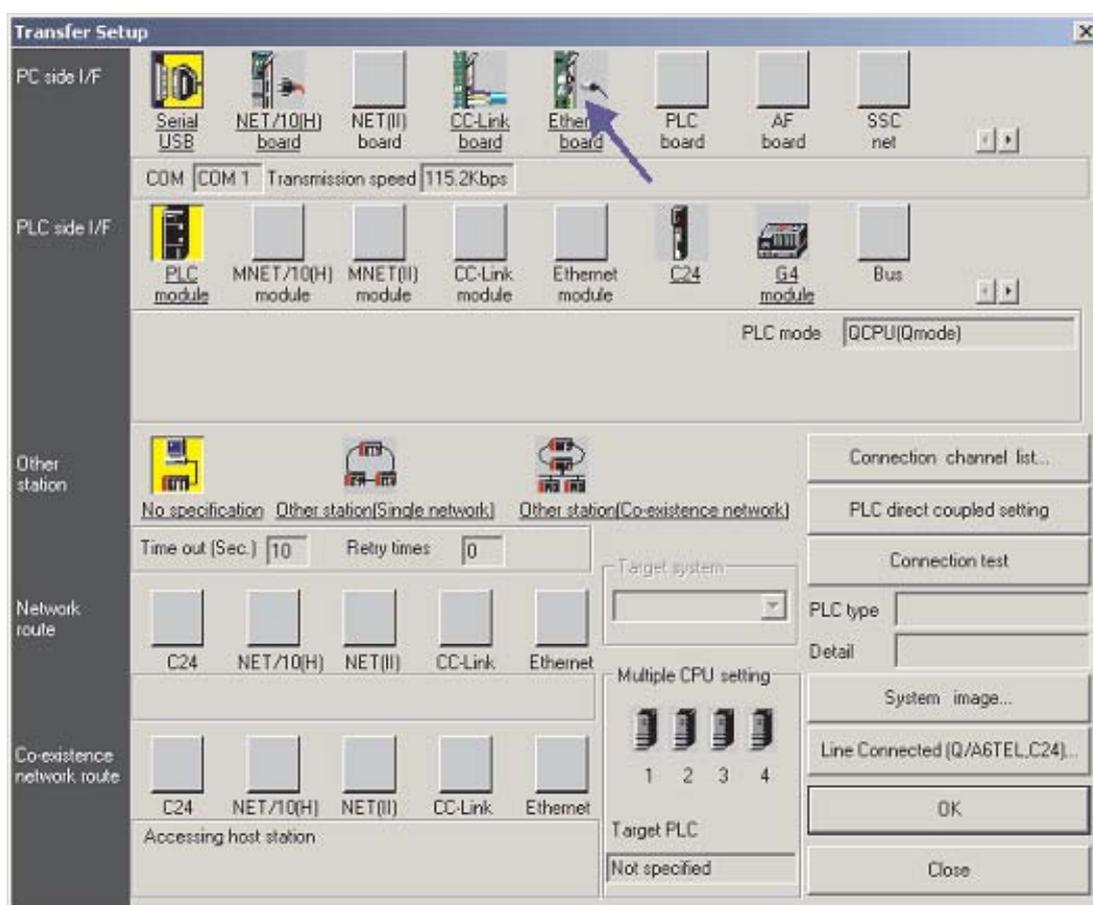
- ① Откройте Свойства сети в Windows® и назначьте IP-адрес и маску подсети в диалоговом окне свойств **TCP/IP** для используемого сетевого адаптера Ethernet. Учтите, что после изменения IP-адреса может потребоваться перезагрузка ПК.



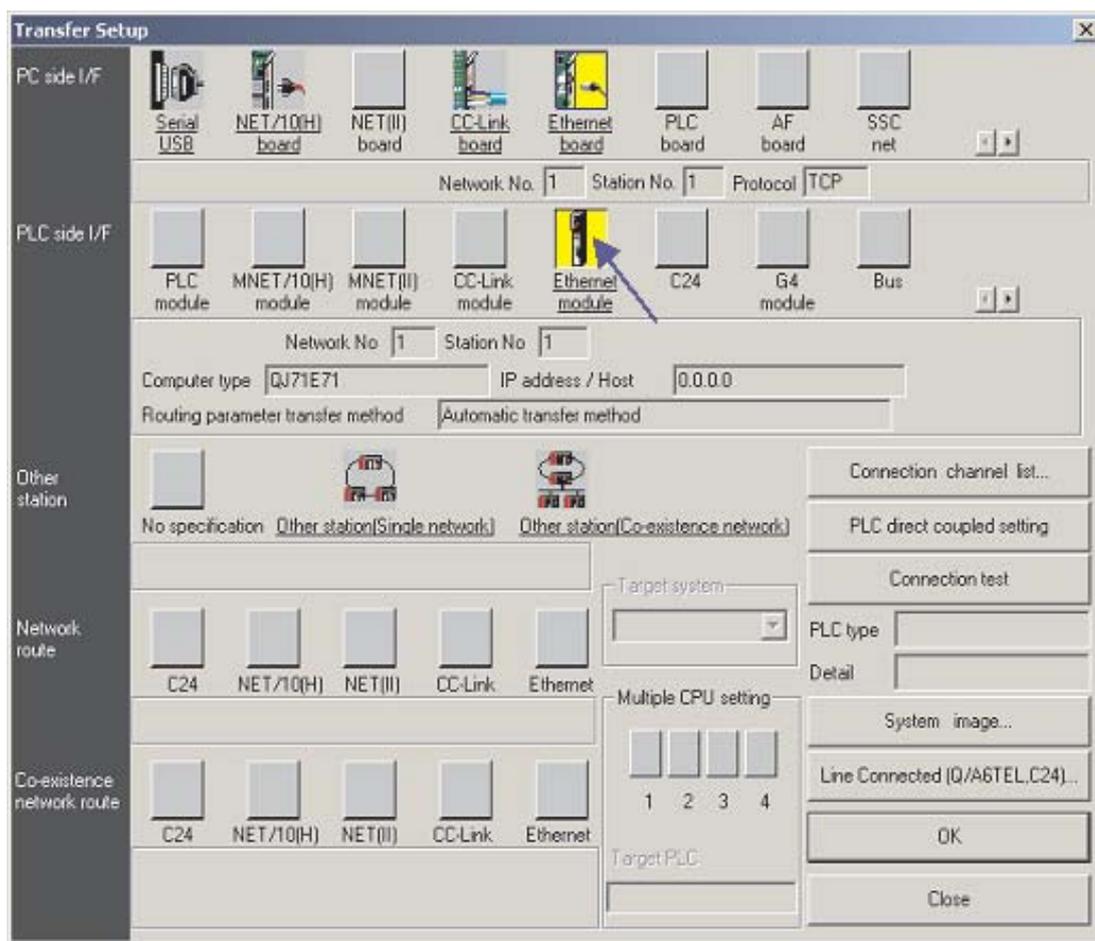
## 18.3 Конфигурирование GX IEC Developer для доступа контроллера в Ethernet

Для доступа GX IEC Developer к ПЛК через сеть Ethernet и модуль Ethernet вам понадобится конфигурировать следующие настройки:

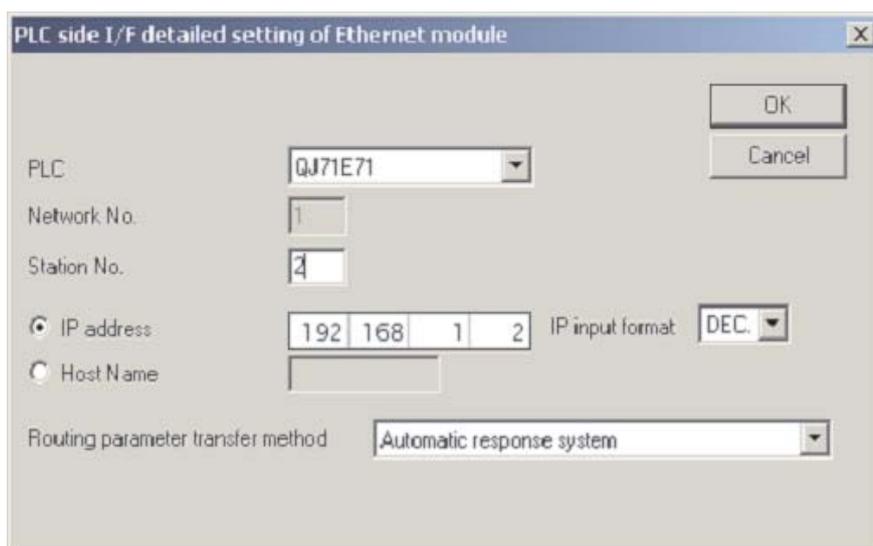
- 1 Откройте диалоговое окно настроек, как показано ниже.



- 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.



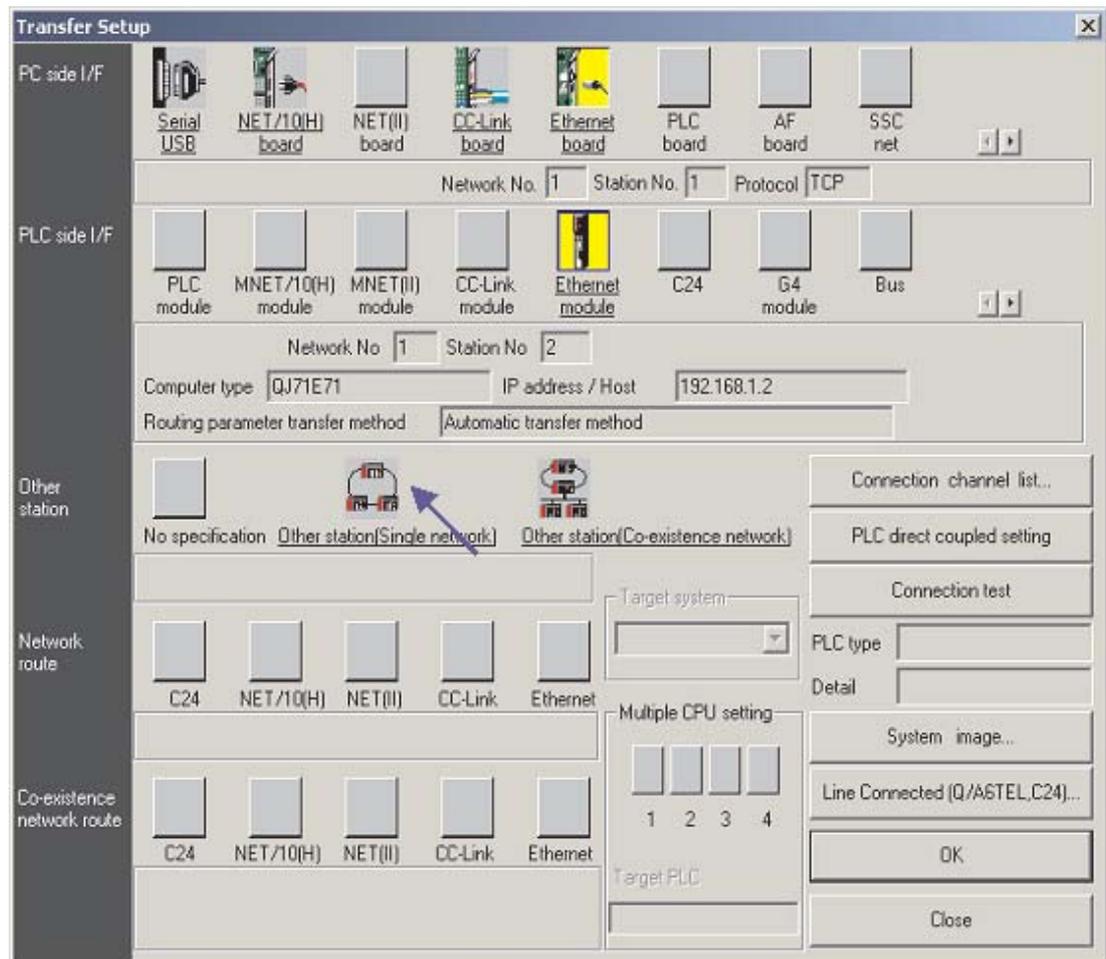
- ④ Далее дважды щелкните на **Ethernet module** в **PLC side I/F**, как показано выше. Откроется диалоговое окно, чтобы выбрать ПЛК для связи через Ethernet. Введите приведенные настройки, так как это были настройки, ранее введенные в ПЛК. (см. пункты ⑤ и ⑥ в разделе 19.1.1, 18.1.1)



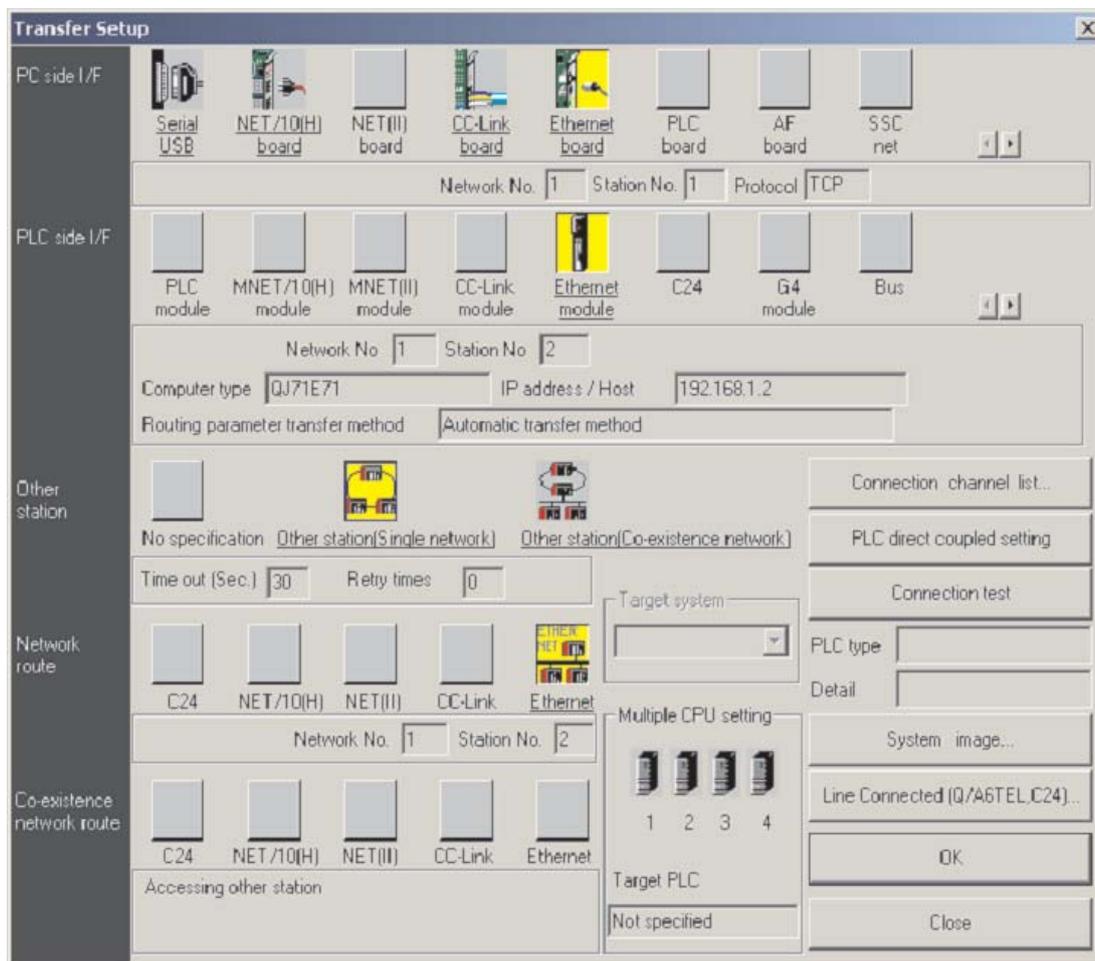
**ПРИМЕЧАНИЕ**

Не требуется указывать номер порта, поскольку в программном обеспечении для программирования используется протокол MELSOFT с назначением порта по умолчанию.

- ⑤ После настройки закройте это диалоговое окно, щелкнув по **OK**.
- ⑥ Затем щелкните по **Other station**.

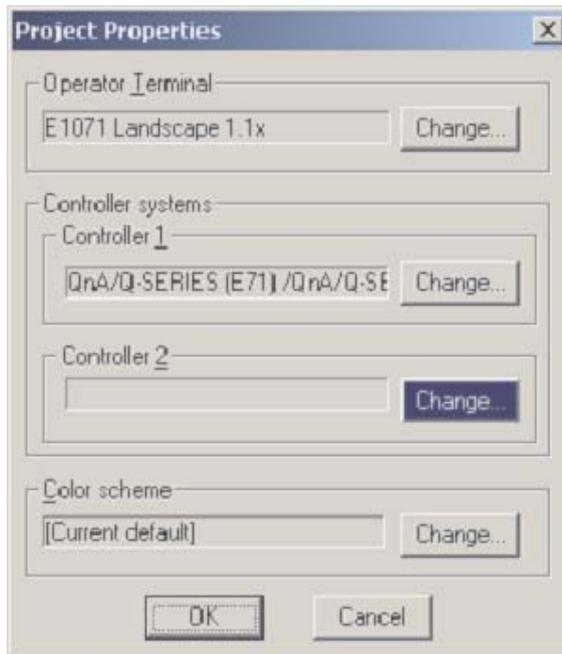


- ⑦ В результате этого настройка завершается. Теперь диалоговое окно настроек передачи должно выглядеть так, как это показано ниже. Для проверки настроек и коммуникации щелкните по "Connection test". После успешного завершения проверки щелкните по **OK**.

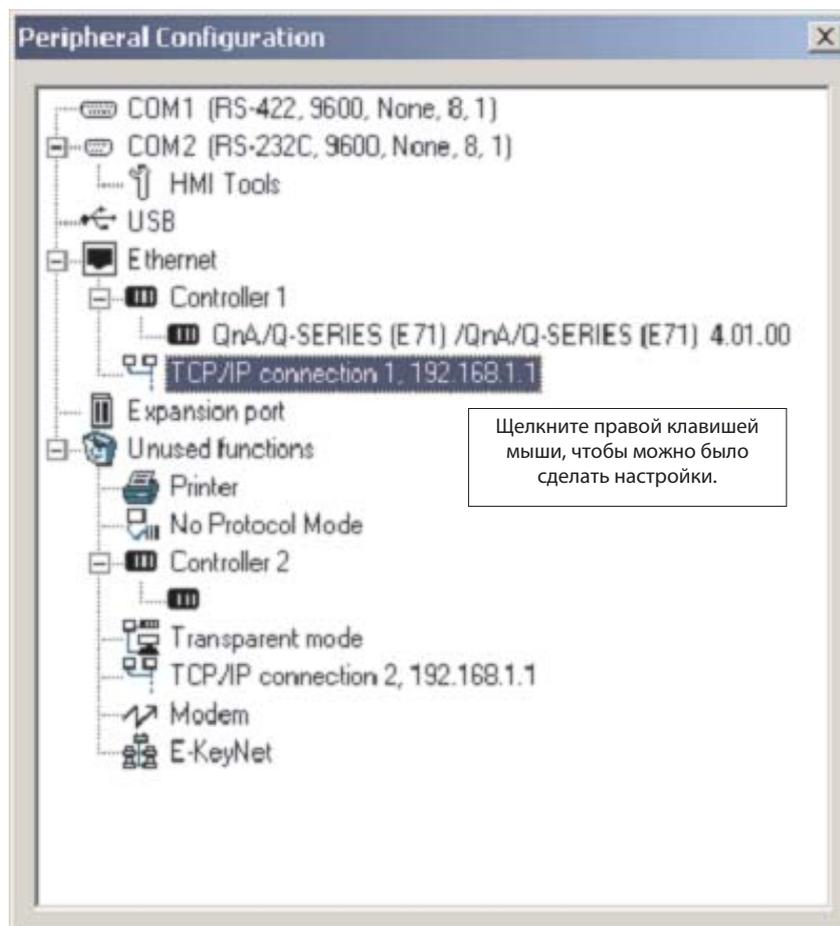


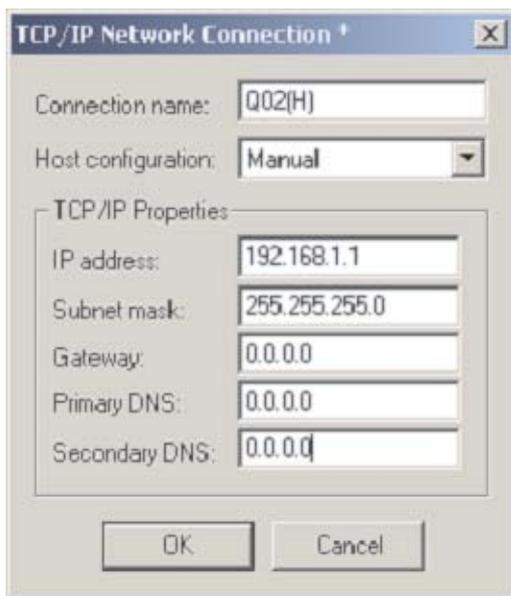
## 18.4 Установка интерфейса человек - машина (HMI)

- ① Проект в E-Designer для иллюстративной системы должен иметь следующие настройки.

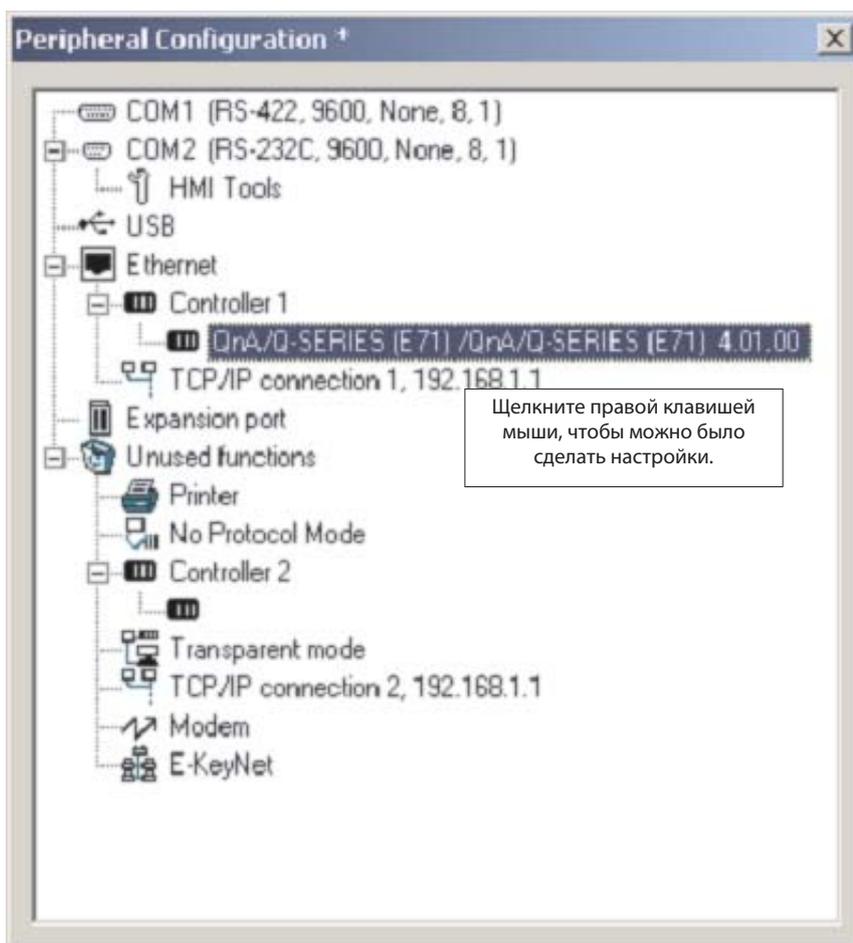


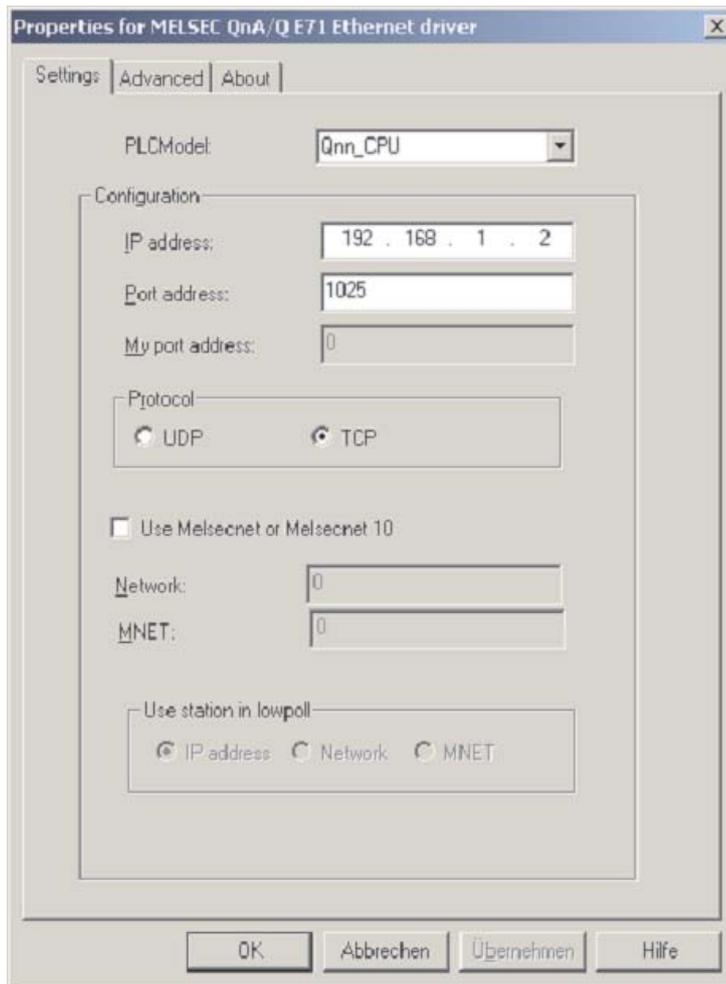
- ② Далее откройте опции **Peripherals** в системном меню и сконфигурируйте TCP/IP соединение для HMI, как показано:





- 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, 18.1.1).

- ④ Щелкните на **OK**, выйдите из настроек периферийного оборудования и загрузите эти настройки с проектом.

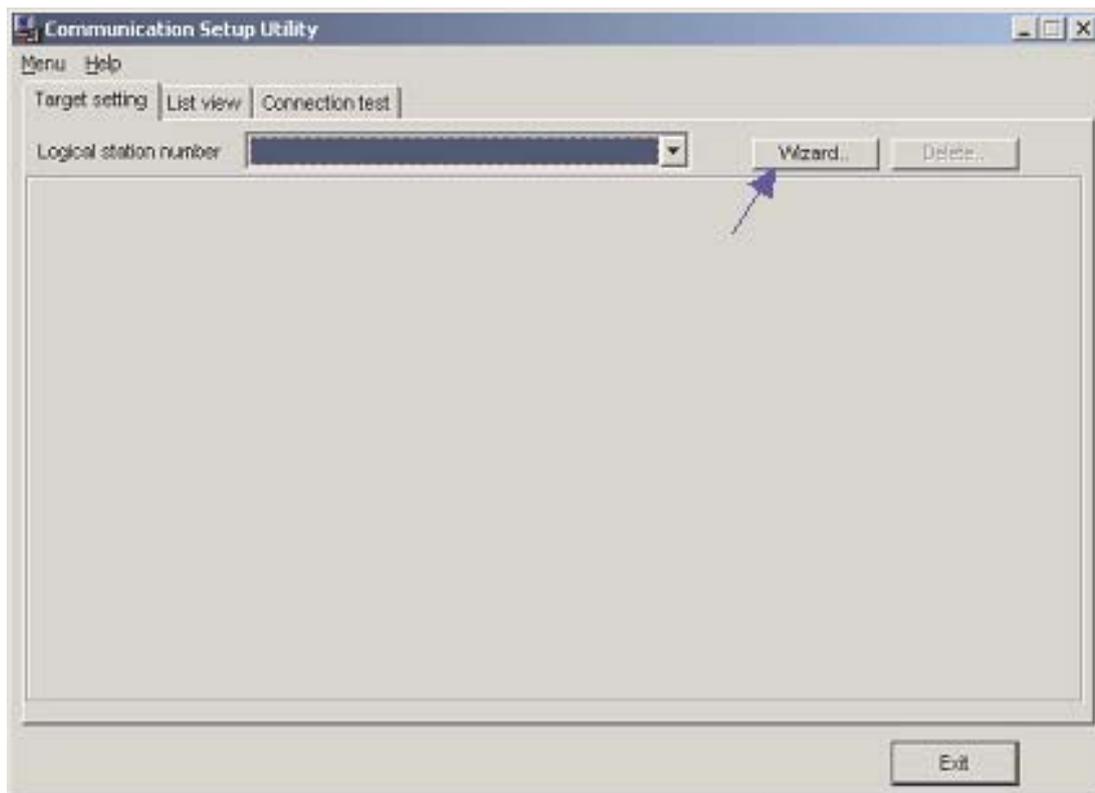
## 18.5 Связь через MX Component

MX Component - это инструмент, предназначенный для организации связи между ПК и ПЛК, не требующий от пользователя никаких знаний о протоколах связи и модулях.

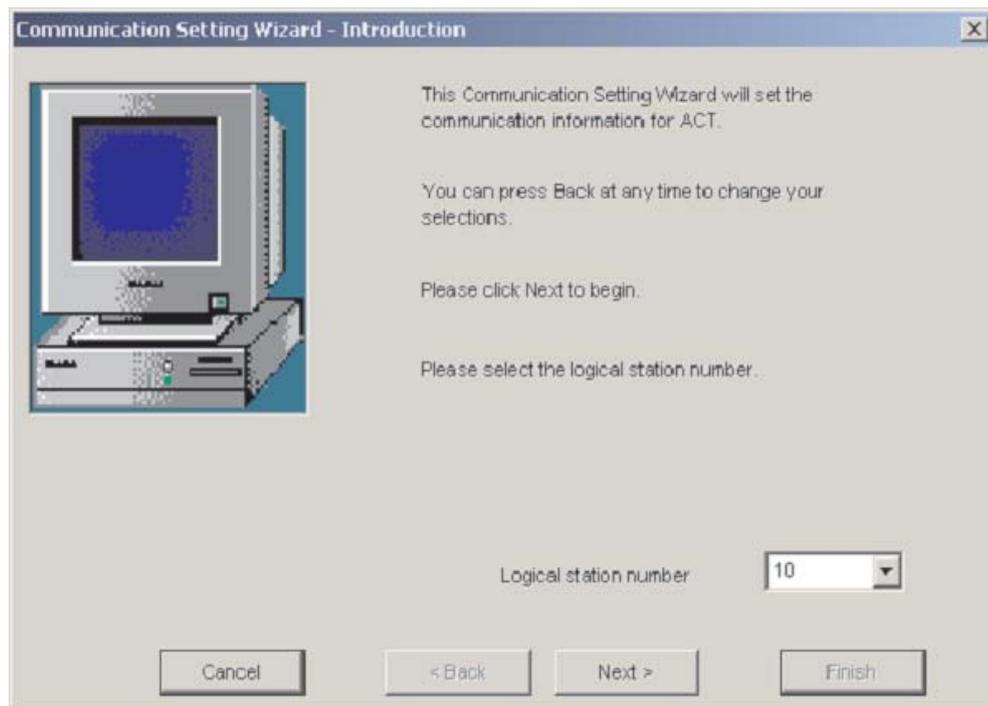
Он поддерживает соединение через последовательный порт ЦП, соединения через последовательные каналы ПК (RS232C, RS422), сети Ethernet, CC-Link и MELSEC.

Ниже на рисунке показан простой способ создания связи между ПК и ПЛК через MX Component.

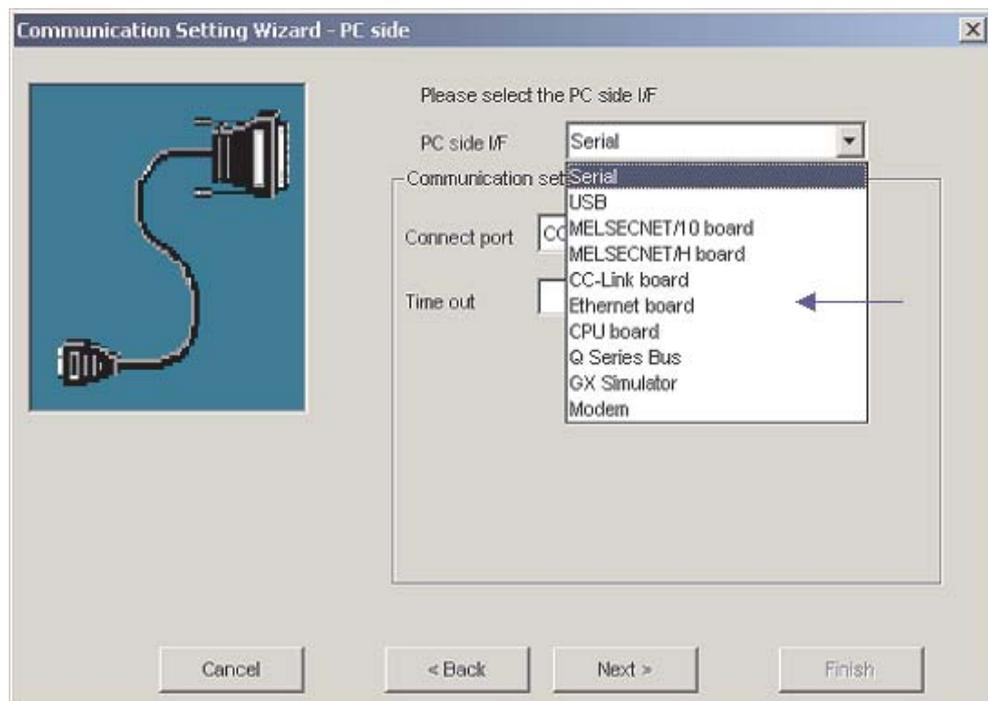
- ① Запустите **Communication Setting Utility** и выберите **Wizard**.



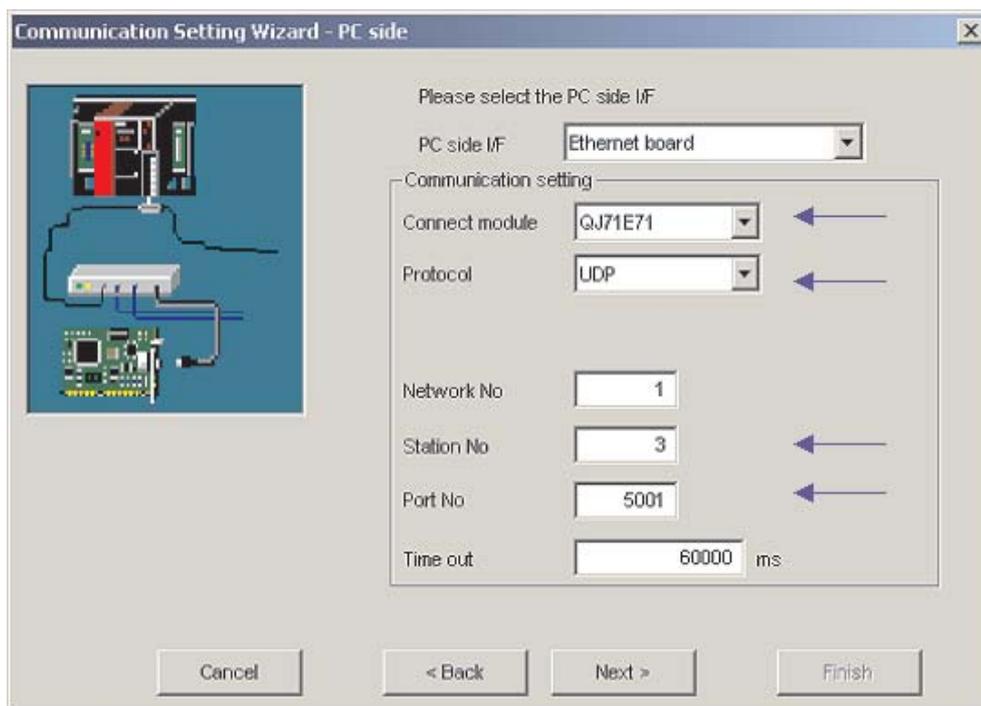
- ② Во-первых, необходимо определить **Logical station number**.



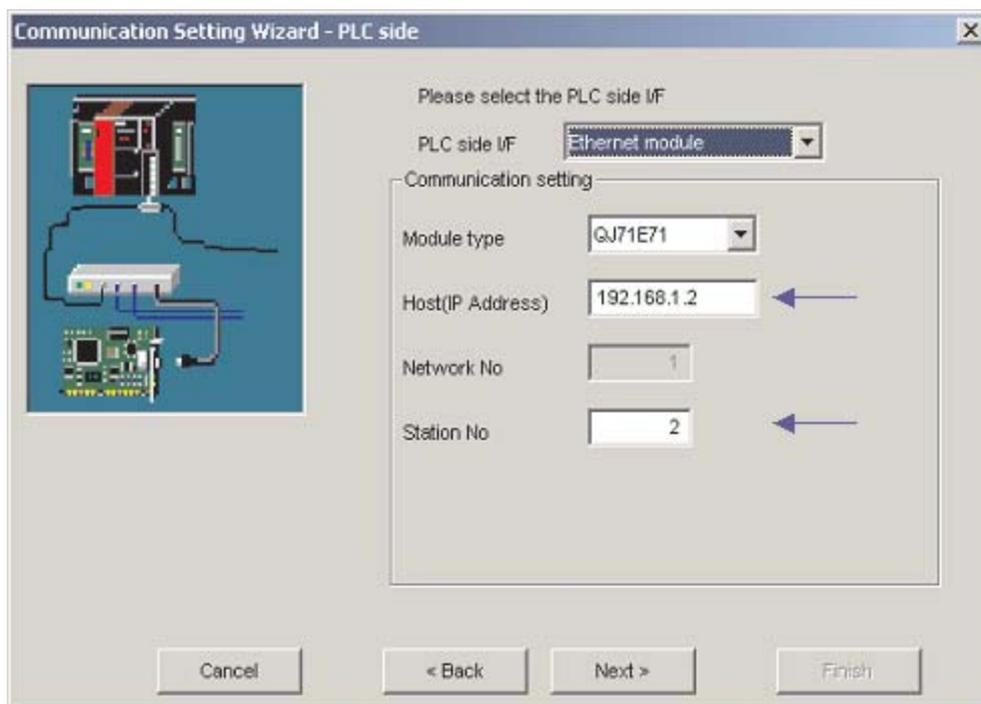
- ③ Затем сконфигурируйте **Communication Settings** на стороне ПК



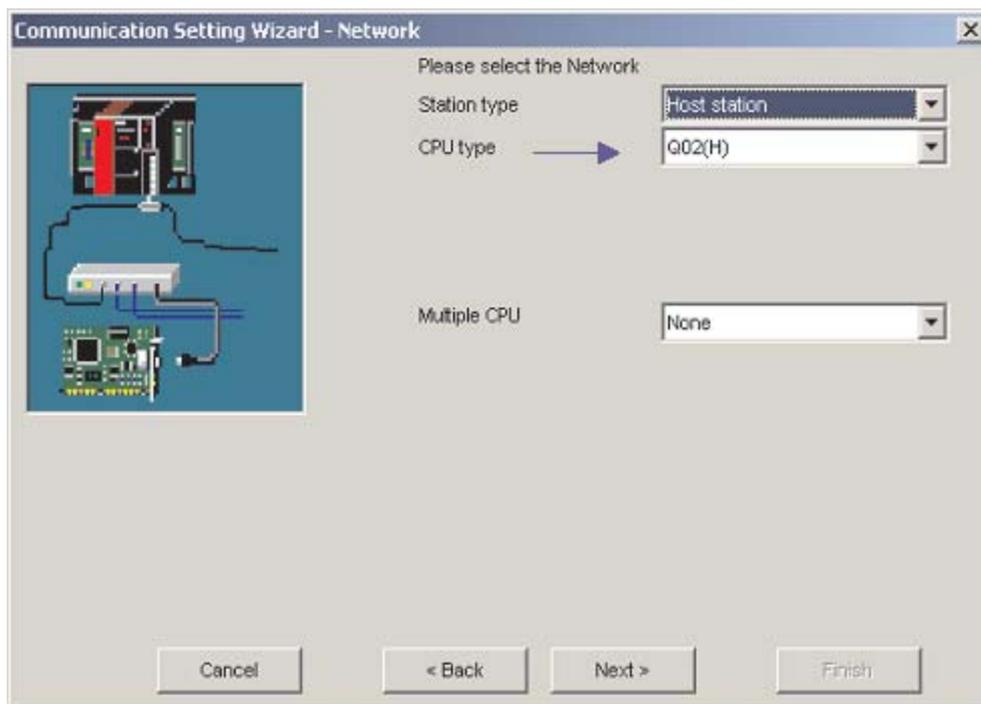
- ④ Выберите UDP протокол и по умолчанию порт 5001



- ⑤ Сконфигурируйте настройки связи на стороне ПЛК, необходимые для описанного ранее примера системы.



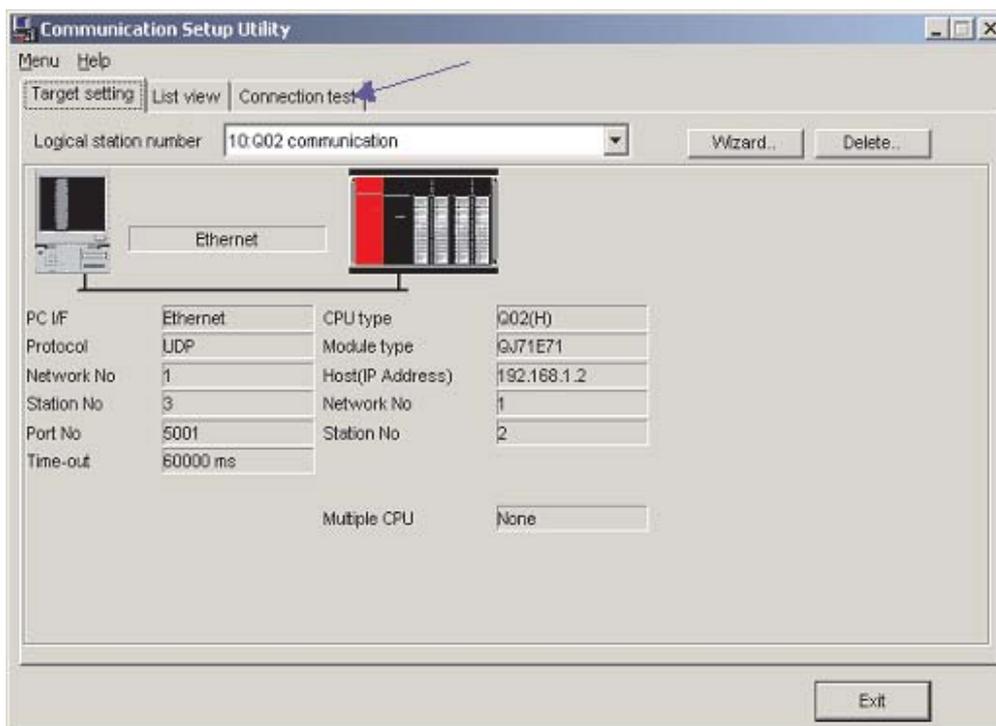
- ⑥ Выберите правильный тип ЦП.



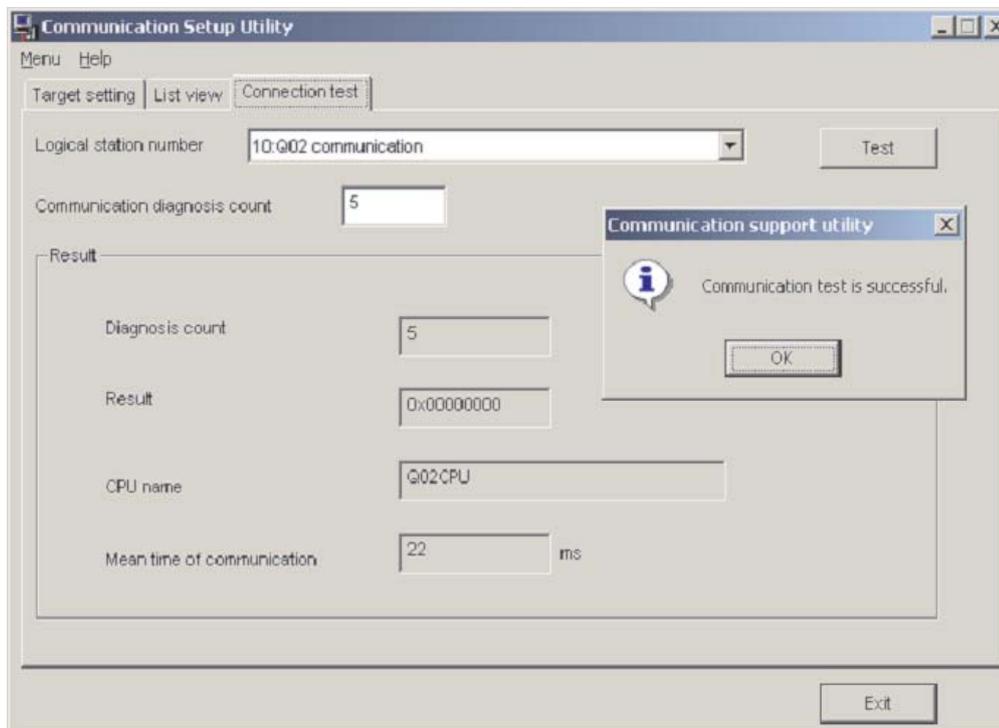
- ⑦ Для завершения конфигурации задайте имя и нажмите кнопку **Finish**.



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 пути коммуникации можно из среды программирования Microsoft (например, Visual Basic или C++) получить доступ для записи и чтения ко всем операндам контроллера.

# А Приложение А

## А.1 Определение времени обработки (SM)

Маркеры диагностики (SM) являются внутренними маркерами, применение которых в контроллере жестко установлено. Поэтому их нельзя использовать в программах аналогично внутренним маркерам. Однако их можно включать и выключать для управления центральным процессором.

### ПРИМЕЧАНИЕ

Маркеры диагностики SM1200 ... SM1255 используются в процессорах QnA. В процессорах серии "Q" эти маркеры не используются.

Маркеры диагностики, начиная с SM 1500, зарезервированы для процессора Q4AR.

В этой таблице разъясняются заголовки столбцов, используемые в таблицах на следующих страницах:

Заголовок таблицы	Значение
Адрес	Показывает адрес маркера диагностики.
Название	Показывает название маркера диагностики.
Значение	Краткое разъяснение значения маркера диагностики.
Описание	Подробная информация о значении маркера диагностики.
Устанавливает (если установлен)	<p>Маркер диагностики может устанавливаться системой или пользователем.</p> <p><b>&lt;Устанавливает&gt;</b>  <b>С:</b> устанавливает система  <b>П:</b> устанавливает пользователь (в основной программе или в тестовом режиме периферийного устройства)  <b>С/П:</b> устанавливает система и пользователь            Показано только в том случае, если установка была выполнена системой.</p> <p><b>&lt;Если установлен&gt;</b>  <b>Обработка команды END:</b> устанавливается при каждой обработке команды END  <b>Инициализация:</b> устанавливается только во время инициализации (при включении блока питания или при переключении центрального процессора из режима "STOP" в режим "RUN")  <b>Изменение состояния:</b> устанавливается только после изменения состояния  <b>Ошибка:</b> устанавливается только после возникновения ошибки  <b>Выполнение команды:</b> устанавливается при выполнении команды  <b>Запрос:</b> устанавливается только в том случае, если имеется запрос пользователя (сделанный с помощью маркера SM или т. п.)</p>
A-CPU M9[ ][ ][ ]	[ ][ ][ ], соответствующий процессору "А". (изменения и иное написание, если оно также изменяется) Если в процессоре "Q" он добавлен впервые, то в таблице он обозначен словом "новый".
Действ. для:	<p>Указывает, для какого центрального процессора предназначен этот специальный маркер.</p> <p>●: действителен для центральных процессоров всех типов  <b>Q:</b> действителен только для всех модулей центральных процессоров серий "System Q"  <b>QnA:</b> действителен для центральных процессоров серий QnA и Q2AS  <b>Тип ЦП:</b> действителен только для данного центрального процессора (например, Q4AR)  <b>Rem:</b> действителен для удаленных модулей ввода-вывода MELSECNET/H</p>

**Информация для диагностики ошибок**

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM0	Ошибка, выявленная при диагностике	выкл.: ошибок нет вкл.: ошибка	Устанавливается в состояние "включено", если результат диагностики показывает ошибку (включая внешнюю диагностику). После устранения ошибки маркер остается установленным.	С (ошибка)	новый	● Rem
SM1	Ошибка, выявленная путем самодиагностики	выкл.: при самодиагностике ошибок не выявлено вкл.: ошибка	Устанавливается в состояние "включено", если результат самодиагностики показывает ошибку. После устранения ошибки маркер остается установленным.	С (ошибка)	M9008	
SM5	Общая информация об ошибке	выкл.: общей информации об ошибке нет вкл.: общая информация об ошибке	При установленном SM0 и наличии общей информации об ошибке устанавливается в состояние "включено"	С (ошибка)	новый	
SM16	Специальная информация об ошибке	выкл.: специальной информации об ошибке нет вкл.: специальная информация об ошибке	При установленном SM0 и наличии специальной информации об ошибке устанавливается в состояние "включено"	С (ошибка)	новый	
SM50	Сброс ошибки	выкл. → вкл.: стирание ошибки	Ошибка сбрасывается. Дополнительная информация имеется в разделе 5.3.6.	П	новый	●
SM51	Низкое напряжение батареи (фиксируемый маркер)	выкл.: нормальное напряжение вкл.: напряжение упало	Напряжение батареи буферного питания центрального процессора или карты памяти снизилось ниже минимального предела. После замены батареи маркер остается установленным. Состояние маркера совпадает с состоянием светодиода "BAT. ALARM".	С (ошибка)	M9007	
SM52	Низкое напряжение батареи	выкл.: нормальное напряжение вкл.: напряжение упало	Напряжение батареи буферного питания снизилось ниже минимального предела. После замены батареи маркер сбрасывается.	С (ошибка)	M9006	
SM53	Падение напряжения питания	выкл.: нормальное напряжение вкл.: напряжение упало	Входное напряжение сетевого блока переменного напряжения исчезло на время менее 20 мс. Сброс происходит при выключении и повторном включении напряжения питания.	С (ошибка)	M9005	●
			Входное напряжение блока питания с входом постоянного напряжения исчезло на время менее 10 мс. Сброс происходит при выключении и повторном включении напряжения питания.			Q
			Входное напряжение блока питания с входом постоянного напряжения исчезло на время менее 1 мс. Сброс происходит при выключении и повторном включении напряжения питания.			QnA
SM54	Ошибка в MELSECNET/MINI	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается при возникновении ошибки связи в установленном модуле AJ71PT32 (S3). Маркер остается установленным и после исчезновения ошибки.	С (ошибка)	M9004	QnA
SM56	Ошибка обработки	выкл.: нормальное состояние вкл.: ошибка обработки	Этот маркер устанавливается при возникновении ошибки обработки. Маркер остается установленным и после исчезновения ошибки.	С (ошибка)	M9011	●
SM60	Неисправен предохранитель	выкл.: нормальное состояние вкл.: модуль с неисправным предохранителем	Этот маркер устанавливается при обнаружении неисправности предохранителя в одном из выходных модулей. Маркер остается установленным и после возврата в нормальное состояние.	С (ошибка)	M9000	● Rem
SM61	Ошибка, выявленная при сверке модулей ввода-вывода	выкл.: нормальное состояние вкл.: при сверке выявлена ошибка	Текущее состояние модулей ввода-вывода отличается от зарегистрированной информации после включения напряжения питания. Сверка модулей ввода-вывода выполняется и для удаленной станции.	С (ошибка)	M9002	
SM62	Индикатор маркера ошибки	выкл.: не распознан вкл.: распознан	Устанавливается, если установлен только один маркер ошибки F.	С (выполнение команды)	M9009	●

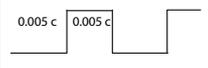
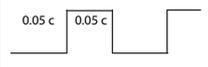
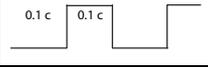
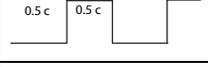
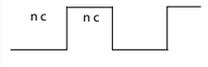
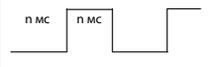
Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM80	Ошибка, выявленная с помощью команды СНК	выкл.: нормальное состояние вкл.: ошибка	Устанавливается при обнаружении ошибки с помощью команды СНК. Маркер остается установленным и после исчезновения ошибки.	С (выполнение команды)	новый	QnA, Q, кроме Q00J, Q00 и Q01
SM90	Запуск WDT (контрольного таймера) для контроля переходов (действует только при наличии программы на языке SFC)	выкл.: не запущен (WDT сброшен) вкл.: запуск (WDT запускается)	соответствует SD90	П	M9108	
SM91			соответствует SD91		M9109	
SM92			соответствует SD92		M9110	
SM93			соответствует SD93		M9111	
SM94			соответствует SD94		M9112	
SM95			соответствует SD95		M9113	
SM96			соответствует SD96		M9114	
SM97			соответствует SD97		новый	
SM98			соответствует SD98		новый	
SM99			соответствует SD99		новый	

### Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM202	Команда выключения светодиода	выкл. → вкл.: выключение светодиода	Светодиоды, сопоставленные битам SD202, гаснут при изменении состояния битов с "выкл." на "вкл."	П	новый	● кроме Q00J, Q00 и Q01
SM203	Маркер состояния "STOP"	Состояние "STOP"	Устанавливается при останове центрального процессора.	С (изменение состояния)	M9042	●
SM204	Маркер состояния "PAUSE"	Состояние "PAUSE"	Устанавливается, если центральный процессор находится в режиме "Пауза".	С (изменение состояния)	M9041	
SM205	Маркер режима "STEP-RUN"	Режим "STEP-RUN"	Устанавливается, если центральный процессор находится в режиме "STEP-RUN".	С (изменение состояния)	M9054	● кроме Q00J, Q00 и Q01
SM206	Условие выполнения для состояния "PAUSE"	выкл.: состояние не возможно вкл.: состояние возможно	Центральный процессор переходит в состояние "PAUSE", если установлены дистанционный контакт "PAUSE" и маркер.	П	M9040	●
	Состояние тестирования операнда	выкл.: тестирование операнда еще не выполнено вкл.: тестирование операнда выполнено	Этот маркер указывает состояние тестирования операнда, которое можно выполнить с помощью среды программирования.	С (запрос)	новый	Q00J, Q00 и Q01 Rem
SM210	Запрос на установку данных часов	выкл.: не обрабатывается вкл.: запрос	При установленном маркере данные часов после выполнения команды END сохраняются в регистрах SD210...SD213 и передаются в часы.	П	M9025	●
SM211	Ошибка данных часов	выкл.: нормальное состояние вкл.: ошибка	Маркер установлен, если в данных часов, сохраненных в регистрах с SD210 по SD213, имеется ошибка. Если ошибок нет, маркер не установлен.	С (запрос)	M9026	
SM212	Индикация данных часов	выкл.: не обрабатывается вкл.: индикация	Данные часов из регистров с SD210 по SD213 считываются и выводятся на светодиодный индикатор центрального процессора с индикацией месяца, дня, часа, минуты и секунды. (Это возможно только в центральных процессорах Q3A и Q4A.)	П	M9027	Q3A, Q4A, Q4AR
SM213	Запрос на считывание данных часов	выкл.: не обрабатывается вкл.: запрос	При установленном маркере данные часов считываются в регистры SD210...SD213 в виде двоично-десятичных значений.	П	M9028	● Rem

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM240	Маркер сброса центрального процессора 1	выкл.: сброса нет вкл.: на модуле центр. процессора 1 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 1 или при отсоединении центрального процессора от монтажной шины. Другие центральные процессоры многопроцессорной системы также сбрасываются.	С (изменение состояния)	новый	Q02, Q02H, Q06H, Q12H, Q25H, начиная с версии "B"
SM241	Маркер сброса центрального процессора 2	выкл.: сброса нет вкл.: на модуле центр. процессора 2 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 2 или при отсоединении центрального процессора от монтажной шины. В других процессорах многопроцессорной системы выводится сообщение об ошибке MULTI CPU DOWN (код ошибки 7000).	С (изменение состояния)	новый	
SM242	Маркер сброса центрального процессора 3	выкл.: без сброса вкл.: на модуле центр. процессора 3 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 3 или при отсоединении центрального процессора от монтажной шины. В других процессорах многопроцессорной системы выводится сообщение об ошибке MULTI CPU DOWN (код ошибки 7000).	С (изменение состояния)	новый	
SM243	Маркер сброса центрального процессора 4	выкл.: сброса нет вкл.: на модуле центр. процессора 4 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 4 или при отсоединении центрального процессора от монтажной шины. В других процессорах многопроцессорной системы выводится сообщение об ошибке MULTI CPU DOWN (код ошибки 7000).	С (изменение состояния)	новый	
SM244	Маркер ошибки центрального процессора 1	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 1, останавливающая центр. процессор	Установленный маркер показывает, что возникла ошибка, остановившая центральный процессор. Если ошибок нет либо имеется ошибка, не вызывающая перехода процессора в состояние "STOP", этот маркер сбрасывается.	С (изменение состояния)	новый	
SM245	Маркер ошибки центрального процессора 2	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 2, останавливающая центр. процессор		С (изменение состояния)	новый	
SM246	Маркер ошибки центрального процессора 3	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 3, останавливающая центр. процессор		С (изменение состояния)	новый	
SM247	Маркер ошибки центрального процессора 4	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 4, останавливающая центр. процессор		С (изменение состояния)	новый	

**Системные такты и счетчики**

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM400	Всегда включен	вкл.  выкл. 	Этот специальный маркер установлен (включен) всегда.	С (обработка команды END)	M9036	●
SM401	Всегда выключен	вкл.  выкл. 	Этот специальный маркер сброшен (выключен) всегда.	С (обработка команды END)	M9037	
SM402	Включен только на один программный цикл после RUN	вкл.  выкл. 	После установки режима RUN программа на один программный цикл устанавливается в состояние "включено". Этот способ могут использовать только программы, выполняемые один раз за цикл.	С (обработка команды END)	M9038	
SM403	Выключен только на один программный цикл после RUN	вкл.  выкл. 	После установки режима RUN программа на один программный цикл устанавливается в состояние "выключено". Этот способ могут использовать только программы, выполняемые один раз за цикл.	С (обработка команды END)	M9039	
SM404	Включен только на один программный цикл после RUN	вкл.  выкл. 	После установки режима RUN программа на один программный цикл устанавливается в состояние "включено". Этот контакт могут использовать только программы, которые могут выполняться в замедленном режиме.	С (обработка команды END)	новый	
SM405	Выключен только на один программный цикл после RUN	вкл.  выкл. 	После установки режима RUN программа на один программный цикл устанавливается в состояние "выключено". Этот контакт могут использовать только программы, которые могут выполняться в замедленном режиме.	С (обработка команды END)	новый	
SM409	Такт 0.01 с		Повторяющееся изменение между включенным и выключенным состоянием с 10-миллисекундным интервалом. После отключения блока питания или сброса центрального процессора маркер автоматически переводится из выключенного состояния во включенное.	С (изменение состояния)	новый	Q, кроме Q001, Q00 и Q01
SM410	Такт 0.1 с		Повторяющееся изменение между включенным и выключенным состоянием с указанным интервалом. Этот процесс продолжается и при состоянии STOP процессора. После отключения сетевого блока или сброса центрального процессора маркер автоматически переключается с выключенного состояния на включенное.	С (изменение состояния)	M9030	●
SM411	Такт 0.2 с				M9031	
SM412	Такт 1 с				M9032	
SM413	Такт 2 с				M9033	
SM414	Такт 2 x n с		Изменение между включенным и выключенным состоянием с интервалом в секундах, указанным в SD414.	С (изменение состояния)	M9034 изменился формат	
SM415	Такт 2 x n мс		Изменяется между включенным и выключенным состоянием с интервалом в миллисекундах, указанным в SD415.	С (изменение состояния)	новый	Q, кроме Q001, Q00 и Q01

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM420	Такт № 0		<p>Маркер непрерывно переключается между включенным и выключенным состоянием с заданным интервалом.</p> <p>После отключения сетевого блока или сброса центрального процессора маркер автоматически переключается с выключенного состояния на включенное.</p> <p>Длительность включенного и выключенного состояния устанавливается с помощью команды DUTY.</p>	С (обработка команды END)	M9020	●
SM421	Такт № 1				M9021	
SM422	Такт № 2				M9022	
SM423	Такт № 3				M9023	
SM424	Такт № 4				M9024	
SM430	Такт № 5		<p>Маркеры SM420...SM424 предназначены для программ с замедленным выполнением.</p>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SM431	Такт № 6					
SM432	Такт № 7					
SM433	Такт № 8					
SM434	Такт № 9					

## A.2 Соответствия между специальными маркерами и маркерами диагностики

При переходе с контроллеров MELSEC серии "A" на контроллеры MELSEC серии "Q" или "System Q" специальные маркеры M9000...M9255 (MELSEC серии "A") соответствуют маркерам диагностики SM1000...SM1255 (MELSEC серии "Q").

Все эти маркеры диагностики устанавливает система – их изменение с помощью пользовательской программы не возможно. Пользователи, желающие устанавливать или сбрасывать эти маркеры, должны изменить свои программы таким образом, чтобы использовались только настоящие маркеры диагностики серии QnA. Исключением являются специальные маркеры M9084 и M9200...M9255. Если перед переходом на контроллеры MELSEC серии "Q"/"System Q" была возможна установка и сброс этих маркеров, то после перехода можно устанавливать и сбрасывать и соответствующие маркеры диагностики SM1084 и SM1200...SM1255.

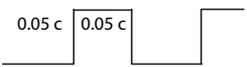
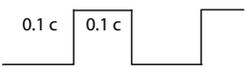
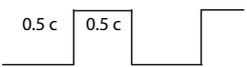
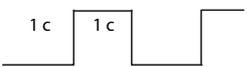
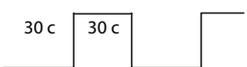
Подробную информацию о специальных маркерах серии "A" можно найти в руководствах по центральному процессору и сетям "MELSECNET" и "MELSECNET/B".

### ПРИМЕЧАНИЕ

При использовании конвертированных специальных маркеров время обработки в центральном процессоре Q может увеличиться. Если конвертированные специальные маркеры не используются, то в среде программирования (в параметрах контроллера, закладка "Система контроллера") следует отменить выбор опции "Контроллер A: использование специальных маркеров и регистров SM/SD 1000".

Если в таблице указан эквивалентный маркер диагностики для центральных процессоров "System Q" или QnA, то следует изменить программу и использовать этот маркер. Если эквивалентный маркер диагностики для "System Q"/QnA не указан, можно использовать конвертированный маркер

Специальный маркер процессора серии "A"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9000	SM1000	—	Неисправен предохранитель	выкл.: нормальное состояние вкл.: ошибка	Q и QnA
M9002	SM1002	—	Ошибка сверки модуля ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9004	SM1004	—	Неисправность в мастер-модуле MELSECNET MINI	выкл.: нормальное состояние вкл.: ошибка	QnA
M9005	SM1005	—	Падение сетевого напряжения	выкл.: нормальное состояние вкл.: падение напряжения	Q и QnA
M9006	SM1006	—	Низкое напряжение батареи	выкл.: нормальное состояние вкл.: падение напряжения	
M9007	SM1007	—	Низкое напряжение батареи (фиксируемый маркер)	выкл.: нормальное состояние вкл.: падение напряжения	
M9008	SM1008	SM1	Распознавание ошибки после самодиагностики	выкл.: нормальное состояние вкл.: ошибка	Q и QnA
M9009	SM1009	SM62	Маркер наличия маркера ошибки	выкл.: не обнаружен вкл.: обнаружен	
M9011	SM1011	SM56	Распознавание ошибки в процессе выполнения программы	выкл.: нормальное состояние вкл.: ошибка	
M9012	SM1012	SM700	Флаг переноса (маркер переноса)	выкл.: перенос выключен вкл.: перенос включен	
M9016	SM1016	не действует в процессорах Q и QnA	Маркер стирания сохраненных данных операндов	выкл.: не выполняется процесс стирания	
M9017	SM1017	в процессорах "System Q" и QnA не задействован	Маркер стирания сохраненных данных операндов	выкл.: не выполняется стирание	

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9020	SM1020	—	Такт № 0		Q и QnA
M9021	SM1021	—	Такт № 1		
M9022	SM1022	—	Такт № 2		
M9023	SM1023	—	Такт № 3		
M9024	SM1024	—	Такт № 4		
M9025	SM1025	—	Запрос на установку данных часов	выкл.: не обрабатывается вкл.: запрос	Q и QnA
M9026	SM1026	—	Ошибка данных часов	выкл.: нормальное состояние вкл.: ошибка	
M9027	SM1027	—	Индикация данных часов	выкл.: не обрабатывается вкл.: запрос	
M9028	SM1028	—	Запрос на считывание данных часов	выкл.: нормальное состояние вкл.: ошибка	
M9029	SM1029	в процессорах "System Q" и QnA не задействован	Пакетная обработка данных коммуникационного запроса	выкл.: пакетная обработка не выполняется вкл.: пакетная обработка выполняется	Q и QnA
M9030	SM1030	—	Датчик тактовых импульсов 0.1 секунды		
M9031	SM1031	—	Датчик тактовых импульсов 0.2 секунды		
M9032	SM1032	—	Датчик тактовых импульсов 1 секунда		
M9033	SM1033	—	Датчик тактовых импульсов 2 секунды		
M9034	SM1034	—	Датчик тактовых импульсов 1 минута		
M9036	SM1036	—	Постоянно включен	вкл. ————— выкл. —————	
M9037	SM1037	—	Постоянно выключен	вкл. ————— выкл. —————	
M9038	SM1038	—	Включен на 1 цикл только после RUN	вкл. ———— выкл. ———— ← 1 цикл →	
M9039	SM1039	—	Выключен только на 1 цикл после RUN	вкл. ———— выкл. ———— ← 1 цикл →	
M9040	SM1040	SM206	Условие паузы	выкл.: режим PAUSE не возможен вкл.: режим PAUSE возможен	Q и QnA
M9041	SM1041	SM204	Маркер состояния PAUSE	выкл.: PAUSE не имеет места вкл.: во время состояния PAUSE	
M9042	SM1042	SM203	Маркер состояния STOP	выкл.: STOP не имеет места вкл.: во время состояния STOP	
M9043	SM1043	SM805	Выборочная трассировка окончена	выкл.: во время выборочной трассировки вкл.: по окончании выборочной трассировки	

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9044	SM1044	SM803	Выборочная трассировка	0 → 1 аналогично выполнению команды STRA 1 → 0 аналогично выполнению команды STRAR	Q и QnA
M9045	SM1045	в процессорах "System Q" и QnA не задействован	Сброс контрольного таймера	выкл.: сброса нет вкл.: контрольный таймер сбрасывается	
M9046	SM1046	SM802	Выборочная трассировка	выкл.: контроль не активен вкл.: контроль активен	
M9047	SM1047	SM801	Подготовка выборочной трассировки	выкл.: останов выборочной трассировки вкл.: запуск выборочной трассировки	
M9049	SM1049	SM701	Количество выводимых знаков	выкл.: вывод до кода NUL вкл.: вывод 16 знаков	
M9051	SM1051	в процессорах "System Q" и QnA не задействован	Подавление команды CHG	выкл.: выполнение возможно вкл.: выполнение не возможно	
M9052	SM1052	в процессорах "System Q" и QnA не задействован	Переключение команды SEG	выкл.: 7-сегментный индикатор вкл.: частичное обновление ввода-вывода	
M9054	SM1054	SM205	Маркер режима "STEP RUN"	выкл.: иной режим вкл.: STEP RUN	
M9055	SM1055	SM808	Маркер фиксации состояния	выкл.: не окончена вкл.: окончена	
M9056	SM1056	в процессорах "System Q" и QnA не задействован	Запрос P, I для основной программы	выкл.: без запроса вкл.: запрос P, I	Q и QnA
M9057	SM1057		Запрос P, I для подпрограммы	выкл.: без запроса вкл.: запрос P, I	
M9058	SM1058		Основная программа, установка P, I завершена	Кратковременно включается после завершения установки P, I	
M9059	SM1059		Подпрограмма, установка P, I завершена	Кратковременно включается после завершения установки P, I	
M9060	SM1060		Запрос P, I для подпрограммы 2	выкл.: без запроса вкл.: запрос P, I	
M9061	SM1061		Запрос P, I для подпрограммы 3	выкл.: без запроса вкл.: запрос P, I	
M9065	SM1065	SM711	Маркер пошаговой передачи	выкл.: иная обработка вкл.: пошаговая передача	QnA
M9066	SM1066	SM712	Переключение обработки передачи	выкл.: пакетная передача вкл.: пошаговая передача	
M9070	SM1070	в процессорах "System Q" и QnA не задействован	A8UPU / A8PUJ Требуемое время поиска	выкл.: время считывания не сокращено вкл.: время считывания сокращено	Q и QnA
M9081	SM1081	SM714	Коммуникационный запрос на удаленный специальный модуль	выкл.: запрос возможен вкл.: запрос не возможен	QnA
M9084	SM1084	в процессорах "System Q" и QnA не задействован	Контроль на ошибки	выкл.: контроль на ошибки выполняется вкл.: без контроля на ошибки	Q и QnA
M9091	SM1091		Маркер ошибки команды	выкл.: нормальное состояние вкл.: ошибка	
M9094	SM1094	SM251	Маркер изменения модулей ввода-вывода	выкл.: изменение имеется вкл.: изменения нет	QnA

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9100	SM1100	SM320	Наличие/отсутствие программы на языке SFC	выкл.: программы на языке SFC не используются вкл.: программы на языке SFC используются	Q и QnA
M9101	SM1101	SM321	Запуск/останов программы на языке SFC	выкл.: останов программы на языке SFC вкл.: запуск программ на языке SFC	
M9102	SM1102	SM322	Вид запуска программы на языке SFC	выкл.: первоначальный запуск: вкл.: возобновление	
M9103	SM1103	SM323	Наличие/отсутствие непрерывных переходов	выкл.: переход не действует вкл.: переход действует	
M9104	SM1104	SM324	Флаг индикации непрерывного перехода	выкл.: при выполненном переходе вкл.: переход не происходит	
M9108	SM1108	SM90	Запуск контрольного таймера пошагового перехода (эквивалентно D9108)	выкл.: контрольный таймер сброшен вкл.: запуск сброса контрольного таймера	Q и QnA
M9109	SM1109	SM91	Запуск контрольного таймера пошагового перехода (эквивалентно D9109)		
M9110	SM1110	SM92	Запуск контрольного таймера пошагового перехода (эквивалентно D9110)		
M9111	SM1111	SM93	Запуск контрольного таймера пошагового перехода (эквивалентно D9111)		
M9112	SM1112	SM94	Запуск контрольного таймера пошагового перехода (эквивалентно D9112)		
M9113	SM1113	SM95	Запуск контрольного таймера пошагового перехода (эквивалентно D9113)		
M9114	SM1114	SM96	Запуск контрольного таймера пошагового перехода (эквивалентно D9114)		
M9180	SM1180	SM825	Флаг завершения семплирования активного шага	выкл.: семплирование запускается вкл.: семплирование завершено	Q и QnA
M9181	SM1181	SM822	Флаг выполнения семплирования активного шага	выкл.: семплирование не выполняется вкл.: семплирование выполняется в данный момент	
M9182	SM1182	SM821	Деблокировка семплирования активного шага	выкл.: семплирование не возможно или приостановлено вкл.: семплирование возможно	
M9196	SM1196	SM325	Вывод рабочего шага после останова блока	выкл.: выходы выкл. вкл.: выходы вкл.	Q и QnA
M9197 M9198	SM1197 SM1198	в процессорах "System Q" и QnA не задействован	Переключение между индикацией неисправности предохранителя и индикацией ошибки сверки модуля ввода-вывода	Индикация изменяется в зависимости от сочетания состояния маркеров M9197 и M9198	
M9199	SM1199		Онлайн-регистрация данных фиксации состояния выборочной трассировки	выкл.: регистрация данных не происходит вкл.: регистрация данных происходит	

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9200	SM1200	—	Прием команды LRDP	выкл.: не принята вкл.: принята	QnA
M9201	SM1201	—	Обработка команды LRDP	выкл.: не завершена вкл.: завершена	
M9202	SM1202	—	Прием команды LWTP	выкл.: не принята вкл.: принята	
M9203	SM1203	—	Обработка команды LWTP	выкл.: не завершена вкл.: завершена	
M9204	SM1204	—	Обработка команды LRDP	выкл.: не завершена вкл.: завершена	
M9205	SM1205	—	Обработка команды LWTP	выкл.: не завершена вкл.: завершена	
M9206	SM1206	—	Ошибка в параметрах связи хост-станции	выкл.: нормальное состояние вкл.: ошибка	
M9207	SM1207	—	Соответствие параметров связи нескольких мастер-станций	выкл.: нормальное состояние вкл.: соответствия нет	
M9208	SM1208	—	Диапазон передачи В и W для мастер-станции на нижнем уровне	выкл.: на 2-й и 3-й ярус вкл.: только на 2-й ярус	
M9209	SM1209	—	Проверка параметров связи (только для мастер-станций на нижнем уровне)	выкл.: проверка вкл.: без проверки	
M9210	SM1210	—	Ошибка карты связи в локальной станции	выкл.: нормальное состояние вкл.: ошибка	
M9211	SM1211	—	Ошибка карты связи в мастер-станции	выкл.: нормальное состояние вкл.: ошибка	
M9224	SM1224	—	Состояние связи	выкл.: онлайн вкл.: офлайн	QnA
M9225	SM1225	—	Ошибка в прямой петле	выкл.: нормальное состояние вкл.: ошибка	
M9226	SM1226	—	Ошибка в обратной петле	выкл.: нормальное состояние вкл.: ошибка	
M9227	SM1227	—	Состояние тестирования петли	выкл.: тестирование не происходит вкл.: тестирование прямой или обратной петли	
M9232	SM1232	—	Рабочее состояние локальной станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	
M9233	SM1233	—	Распознавание ошибки для локальной станции	выкл.: нормальное состояние вкл.: ошибка	
M9235	SM1235	—	Ошибка параметра в локальной или удаленной станции ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9236	SM1236	—	Состояние инициализации локальной или удаленной станции ввода-вывода	выкл.: передача не происходит вкл.: передача данных	
M9237	SM1237	—	Ошибка в локальной или удаленной станции ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9238	SM1238	—	Ошибка в петле локальной или удаленной станции ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9240	SM1240	—	Состояние связи	выкл.: онлайн вкл.: офлайн	
M9241	SM1241	—	Ошибка в прямой петле	выкл.: нормальное состояние вкл.: ошибка	
M9242	SM1242	—	Ошибка в обратной петле	выкл.: нормальное состояние вкл.: ошибка	
M9243	SM1243	—	Передача по обратной петле	выкл.: не выполняется вкл.: выполняется	

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9246	SM1246	—	Состояние приема данных	выкл.: данные приняты вкл.: данные не приняты	QnA
M9247	SM1247	—	Состояние приема данных	выкл.: данные приняты вкл.: данные не приняты	
M9250	SM1250	—	Состояние приема параметров	выкл.: параметры приняты вкл.: параметры не приняты	
M9251	SM1251	—	Обрыв передачи	выкл.: нормальное состояние вкл.: обрыв	
M9252	SM1252	—	Состояние тестирования петли	выкл.: тестирование не происходит вкл.: тестирование прямой или обратной петли	
M9253	SM1253	—	Рабочее состояние мастер-станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	
M9254	SM1254	—	Рабочее состояние другой локальной станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	
M9255	SM1255	—	Распознавание ошибки для других локальных станций	выкл.: нормальное состояние вкл.: ошибка	

## А.3 Обзор регистров диагностики

Регистры диагностики SD – это внутренние регистры определенного назначения в программируемом контроллере. Поэтому эти регистры нельзя использовать в основных программах аналогично обычным регистрам. Однако для управления центральным процессором запись данных в эти регистры возможна.

Данные в регистрах диагностики сохраняются в двоичном формате, если только не потребовался иной формат.

В этом разделе описаны только те регистры диагностики, которые чаще всего применяются.

### ПРИМЕЧАНИЕ

Специальные регистры с SD1200 по SD1255 используются в процессорах серии QnA. В процессорах MELSEC серии "System Q" эти регистры не используются.

Специальные регистры, начиная с SD1500, зарезервированы для процессора Q4AR.

В этой таблице разъясняются заголовки столбцов, используемые в таблицах на следующих страницах

Заголовок таблицы	Значение
Адрес	адрес регистра диагностики
Название	название регистра диагностики
Значение	краткое разъяснение значения регистра
Описание	подробная информация о значении регистра
Устанавливает (если установлен)	<p>Регистры может устанавливать пользователь или система.</p> <p><b>&lt;Устанавливает&gt;</b>  <b>С:</b> устанавливает система  <b>П:</b> устанавливает пользователь (с помощью основной программы или вручную в тестовом режиме периферийного устройства)  <b>С/П:</b> устанавливает система и пользователь            Отображается только в случае, если регистр устанавливается системой.</p> <p><b>&lt;Если установлен&gt;</b>  <b>Обработка команды END:</b> устанавливается при каждой обработке команды END  <b>Инициализация:</b> устанавливается только во время инициализации (при включении сетевого блока или переключении центр. процессора из режима STOP в режим RUN)  <b>Изменение состояния:</b> устанавливается только после изменения состояния  <b>Ошибка:</b> устанавливается только после возникновения ошибки  <b>Выполнение команды:</b> устанавливается при выполнении команды  <b>Запрос:</b> устанавливается только по запросу пользователя (с помощью маркера SM и т. п.)</p>
Соответствующий регистр центрального процессора серии "А" D9 [ ] [ ] [ ]	<p>Специальный регистр D9 [ ] [ ] [ ], соответствующий процессору серии "А". (Изменение, если таковое имеется, и примечание к нему.)</p> <p>Если в этом столбце указано "новый", то это означает, что этот регистр впервые появился в центральных процессорах серий Q и QnA.</p>
Действ. для:	<p>Указывает, для какого центрального процессора предназначен этот спец. регистр.</p> <p>●: действителен для центральных процессоров всех типов  <b>Q:</b> действителен только для центральных процессоров MELSEC "System Q"  <b>QnA:</b> действителен для центральных процессоров MELSEC серий QnA и Q2AS  <b>Тип ЦП:</b> действителен только для этого центрального процессора (например, Q4AR)  <b>Rem:</b> действителен для удаленных модулей ввода-вывода MELSENET/H</p>

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистрпроц. серии "А" D9 [][][]	Дейст. для:						
SD0	Ошибка, выявленная при диагностике	Код ошибки, выявленной путем диагностики	<ul style="list-style-type: none"> <li>Код ошибки, выявленной с помощью функции диагностики, сохраняется в двоичном формате.</li> <li>Содержимое этого регистра относится к последнему событию возникновения ошибки.</li> </ul>	С (ошибка)	D9008 изменился формат							
SD1	Время суток, в котором при диагностике была выявлена ошибка	Время суток, в котором при диагностике была выявлена ошибка	<ul style="list-style-type: none"> <li>Год (последние два разряда) и месяц, в которых было обновлено содержимого регистра SD0. Данные сохраняются в двузначной двоично-десятичной кодировке. Пример: <b>октябрь 1995 = 9510</b></li> </ul> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">b15</td> <td style="width: 33%; text-align: center;">b8 b7</td> <td style="width: 33%; text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">год (от 0 до 99)</td> <td style="text-align: center;">месяц (от 1 до 12)</td> <td></td> </tr> </table>	b15	b8 b7	b0	год (от 0 до 99)	месяц (от 1 до 12)		С (ошибка)	новый	
b15			b8 b7	b0								
год (от 0 до 99)			месяц (от 1 до 12)									
SD2	<ul style="list-style-type: none"> <li>День и час обновления данных в SD0. Данные сохраняются в двузначной двоично-десятичной кодировке. Пример: <b>25-е число, 22 часа = 2522</b></li> </ul> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">b15</td> <td style="width: 33%; text-align: center;">b8 b7</td> <td style="width: 33%; text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">день (от 1 до 31)</td> <td style="text-align: center;">час (от 0 до 23)</td> <td></td> </tr> </table>	b15	b8 b7	b0	день (от 1 до 31)	час (от 0 до 23)						
b15	b8 b7	b0										
день (от 1 до 31)	час (от 0 до 23)											
SD3	<ul style="list-style-type: none"> <li>Минута и секунда обновления данных в SD0. Данные сохраняются в двузначной двоично-десятичной кодировке. Пример: <b>35 мин 48 с = 3548</b></li> </ul> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">b15</td> <td style="width: 33%; text-align: center;">b8 b7</td> <td style="width: 33%; text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">минута (от 0 до 59)</td> <td style="text-align: center;">секунда (от 0 до 59)</td> <td></td> </tr> </table>	b15	b8 b7	b0	минута (от 0 до 59)	секунда (от 0 до 59)						
b15	b8 b7	b0										
минута (от 0 до 59)	секунда (от 0 до 59)											
SD4	Категории информации об ошибках	Коды категорий информации об ошибках	<p>С помощью кодов категорий можно определить, какого типа информация сохранена в области общей информации об ошибке (SD5 - SD15) и области специфической информации об ошибке (SD16 - SD26).</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">b15</td> <td style="width: 50%; text-align: center;">b8 b7</td> <td style="width: 50%; text-align: center;">b0</td> </tr> <tr> <td style="text-align: center;">специфическая информация об ошибках</td> <td style="text-align: center;">общая информация об ошибках</td> <td></td> </tr> </table> <ul style="list-style-type: none"> <li>Коды категорий общей информации об ошибках сохраняются следующим образом:                             <ol style="list-style-type: none"> <li>ошибок нет</li> <li>номер станции / модуля / центр. проц. / монт. шины</li> <li>название файла / дисковод</li> <li>время (установленное значение)</li> <li>локализация ошибки программы</li> <li>основание переключения (только в случае проц. Q4AR)</li> </ol> </li> <li>Коды категорий специфической информации об ошибках сохраняются следующим образом:                             <ol style="list-style-type: none"> <li>ошибок нет</li> <li>(открыто)</li> <li>название файла / дисковод</li> <li>время (фактическое измеренное значение)</li> <li>локализация ошибки программы</li> <li>номер параметра</li> <li>номер маркера ошибки</li> <li>номер ошибки команды СНК</li> </ol> </li> </ul>	b15	b8 b7	b0	специфическая информация об ошибках	общая информация об ошибках		С (ошибка)	новый	<ul style="list-style-type: none"> <li>Rem</li> </ul>
b15	b8 b7	b0										
специфическая информация об ошибках	общая информация об ошибках											

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [1][1][1]	Дейст. для:																																																																																																									
SD5	Общая информация об ошибках		<ul style="list-style-type: none"> <li>Здесь сохраняется общая информация, относящаяся к кодам ошибок (SD0).</li> <li>Сохраняется информация следующих 5 видов:                             <ol style="list-style-type: none"> <li>номер станции / модуля                                     <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td>номер станции/модуля</td></tr> <tr><td>SD6</td><td>номер входа или выхода</td></tr> <tr><td>SD7</td><td rowspan="10">свободно</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr><td>SD10</td></tr> <tr><td>SD11</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> </li> <li>Название файла / дисковода                                     <p style="margin-left: 20px;">Пример: название файла = ABCDEFGH.IJK</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td>дисковод</td></tr> <tr><td>SD6</td><td rowspan="3">название файла (ASCII-код: 8 знаков)</td></tr> <tr><td>SD7</td></tr> <tr><td>SD8</td></tr> <tr> <td>SD9</td> <td>расширение    2E<sub>n</sub>(.)</td> </tr> <tr><td>SD10</td><td rowspan="5">свободно</td></tr> <tr><td>SD11</td><td>(ASCII-код: 3 знака)</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> <div style="margin-left: 20px;">                     b15                    b0  <table border="1" style="display: inline-table;"> <tr><td>B</td><td>A</td></tr> <tr><td>D</td><td>C</td></tr> <tr><td>F</td><td>E</td></tr> <tr><td>H</td><td>G</td></tr> <tr><td>I</td><td>.</td></tr> <tr><td>K</td><td>J</td></tr> </table> </div> </li> <li>Время (установленное значение)                                     <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td>время: с шагом в 1 мкс (0 – 999 мкс)</td></tr> <tr><td>SD6</td><td>время: с шагом в 1 мс (0 – 65535 мс)</td></tr> <tr><td>SD7</td><td rowspan="10">свободно</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr><td>SD10</td></tr> <tr><td>SD11</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> </li> <li>Локализация ошибки программы                                     <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td rowspan="4">название файла (ASCII-код: 8 знаков)</td></tr> <tr><td>SD6</td></tr> <tr><td>SD7</td></tr> <tr><td>SD8</td></tr> <tr> <td>SD9</td> <td>расширение    2E<sub>n</sub>(.)</td> </tr> <tr><td>SD10</td><td>(ASCII-код: 3 знака)</td></tr> <tr><td>SD11</td><td>конфигурация битов *</td></tr> <tr><td>SD12</td><td>№ блока</td></tr> <tr><td>SD13</td><td>№ шага / перехода</td></tr> <tr><td>SD14</td><td>№ шага программы (L)</td></tr> <tr><td>SD15</td><td>№ шага программы (H)</td></tr> </tbody> </table> <p style="margin-left: 20px;">* Конфигурация битов:</p> <table border="1" style="margin-left: 20px;"> <tr> <td>15</td><td>14</td><td>—</td><td>—</td><td>—</td><td>—</td><td>3</td><td>2</td><td>1</td><td>0</td> <td style="font-size: small;">← (№ бита)</td> </tr> <tr> <td>0</td><td>0</td><td>—</td><td>—</td><td>—</td><td>—</td><td>0</td><td>*</td><td>*</td><td>*</td> <td></td> </tr> </table> <p style="margin-left: 20px; font-size: x-small;">                     не используется      блок CFS имеется (1) / не имеется (0)                                                       шаг CFS имеется (1) / не имеется (0)                                                       переход CFS имеется (1) / не имеется (0)                 </p> </li> </ol> </li> </ul>	Номер	Значение	SD5	номер станции/модуля	SD6	номер входа или выхода	SD7	свободно	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Номер	Значение	SD5	дисковод	SD6	название файла (ASCII-код: 8 знаков)	SD7	SD8	SD9	расширение    2E <sub>n</sub> (.)	SD10	свободно	SD11	(ASCII-код: 3 знака)	SD12	SD13	SD14	SD15	B	A	D	C	F	E	H	G	I	.	K	J	Номер	Значение	SD5	время: с шагом в 1 мкс (0 – 999 мкс)	SD6	время: с шагом в 1 мс (0 – 65535 мс)	SD7	свободно	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Номер	Значение	SD5	название файла (ASCII-код: 8 знаков)	SD6	SD7	SD8	SD9	расширение    2E <sub>n</sub> (.)	SD10	(ASCII-код: 3 знака)	SD11	конфигурация битов *	SD12	№ блока	SD13	№ шага / перехода	SD14	№ шага программы (L)	SD15	№ шага программы (H)	15	14	—	—	—	—	3	2	1	0	← (№ бита)	0	0	—	—	—	—	0	*	*	*		С (ошибка)	новый	●
Номер				Значение																																																																																																											
SD5				номер станции/модуля																																																																																																											
SD6				номер входа или выхода																																																																																																											
SD7				свободно																																																																																																											
SD8																																																																																																															
SD9																																																																																																															
SD10																																																																																																															
SD11																																																																																																															
SD12																																																																																																															
SD13																																																																																																															
SD14																																																																																																															
SD15																																																																																																															
Номер					Значение																																																																																																										
SD5				дисковод																																																																																																											
SD6	название файла (ASCII-код: 8 знаков)																																																																																																														
SD7																																																																																																															
SD8																																																																																																															
SD9	расширение    2E <sub>n</sub> (.)																																																																																																														
SD10	свободно																																																																																																														
SD11		(ASCII-код: 3 знака)																																																																																																													
SD12																																																																																																															
SD13																																																																																																															
SD14																																																																																																															
SD15																																																																																																															
B	A																																																																																																														
D	C																																																																																																														
F	E																																																																																																														
H	G																																																																																																														
I	.																																																																																																														
K	J																																																																																																														
Номер	Значение																																																																																																														
SD5	время: с шагом в 1 мкс (0 – 999 мкс)																																																																																																														
SD6	время: с шагом в 1 мс (0 – 65535 мс)																																																																																																														
SD7	свободно																																																																																																														
SD8																																																																																																															
SD9																																																																																																															
SD10																																																																																																															
SD11																																																																																																															
SD12																																																																																																															
SD13																																																																																																															
SD14																																																																																																															
SD15																																																																																																															
Номер		Значение																																																																																																													
SD5	название файла (ASCII-код: 8 знаков)																																																																																																														
SD6																																																																																																															
SD7																																																																																																															
SD8																																																																																																															
SD9	расширение    2E <sub>n</sub> (.)																																																																																																														
SD10	(ASCII-код: 3 знака)																																																																																																														
SD11	конфигурация битов *																																																																																																														
SD12	№ блока																																																																																																														
SD13	№ шага / перехода																																																																																																														
SD14	№ шага программы (L)																																																																																																														
SD15	№ шага программы (H)																																																																																																														
15	14	—	—	—	—	3	2	1	0	← (№ бита)																																																																																																					
0	0	—	—	—	—	0	*	*	*																																																																																																						
SD6																																																																																																															
SD7																																																																																																															
SD8																																																																																																															
SD9																																																																																																															
SD10																																																																																																															
SD11																																																																																																															
SD12																																																																																																															
SD13																																																																																																															
SD14																																																																																																															
SD15																																																																																																															

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [][][]	Дейст. для:
Значение расширений названия файлов:						
SD10 (SD9)		SD11 (SD10)		Расширение	Тип файла	
Старший байт	Младший байт	Старший байт				
51H	50H	41H		QPA	параметры	
51H	50H	47H		QPG	программы	
51H	43H	44H		QCD	комментарии к операндам	
51H	44H	49H		QDI	начальные значения операндов	
51H	44H	52H		QDR	регистры файлов	
51H	44H	53H		QDS	данные имитации	
51H	44H	4CH		QDL	локальные операнды	
51H	54H	53H		QTS	данные выб. трассировки (только QnA)	
51H	54H	4CH		QTL	данные фиксации состояния (только QnA)	
51H	54H	50H		QTP	данные трассировки программы (QnA)	
51H	54H	52H		QTR	файл трассировки для прог. на языке SFC	
51H	46H	44H		QFD	данные ошибок	

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [][][]	Дейст. для:																																																																									
SD16	Общая информация об ошибках		<ul style="list-style-type: none"> <li>Здесь сохраняется общая информация, относящаяся с кодам ошибок (SD0).</li> <li>Здесь сохраняются следующие 6 видов данных:</li> </ul> <p>(1) название файла/дисковода      Пример: <b>название файла = ABCDEFGH.IJK</b></p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>дисковод</td> </tr> <tr> <td>SD17</td> <td rowspan="4">название файла (ASCII-код: 8 знаков)</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> <td>расширение      2Ен(,)</td> </tr> <tr> <td>SD22</td> <td>(ASCII-код: 3 знака)</td> </tr> <tr> <td>SD23</td> <td rowspan="4">свободно</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table> <p>(2) Время (установленное значение)</p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>время: с шагом 1 мкс (0 – 999 мкс)</td> </tr> <tr> <td>SD17</td> <td>время: с шагом 1 мс (0 – 65535 мс)</td> </tr> <tr> <td>SD18</td> <td rowspan="10">свободно</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table> <p>(3) Локализация ошибки программы</p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td rowspan="4">название файла (ASCII-код: 8 знаков)</td> </tr> <tr> <td>SD17</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> <td>расширение      2Ен(,)</td> </tr> <tr> <td>SD21</td> <td>(ASCII-код: 3 знака)</td> </tr> <tr> <td>SD22</td> <td>конфигурация битов *</td> </tr> <tr> <td>SD23</td> <td>№ блока</td> </tr> <tr> <td>SD24</td> <td>№ шага/перехода</td> </tr> <tr> <td>SD25</td> <td>№ шага программы (L)</td> </tr> <tr> <td>SD26</td> <td>№ шага программы (H)</td> </tr> </tbody> </table> <p>* Конфигурация битов:</p> <table border="1"> <tr> <td>15</td><td>14</td><td>—</td><td>—</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>0</td><td>0</td><td>—</td><td>—</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td> </tr> </table> <p>← (№ бита)          не используется      блок CFS имеется (1) / не имеется (0)          шаг CFS имеется (1) / не имеется (0)          переход CFS имеется (1) / не имеется (0)</p>	Номер	Значение	SD16	дисковод	SD17	название файла (ASCII-код: 8 знаков)	SD18	SD19	SD20	SD21	расширение      2Ен(,)	SD22	(ASCII-код: 3 знака)	SD23	свободно	SD24	SD25	SD26	Номер	Значение	SD16	время: с шагом 1 мкс (0 – 999 мкс)	SD17	время: с шагом 1 мс (0 – 65535 мс)	SD18	свободно	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	Номер	Значение	SD16	название файла (ASCII-код: 8 знаков)	SD17	SD18	SD19	SD20	расширение      2Ен(,)	SD21	(ASCII-код: 3 знака)	SD22	конфигурация битов *	SD23	№ блока	SD24	№ шага/перехода	SD25	№ шага программы (L)	SD26	№ шага программы (H)	15	14	—	—	4	3	2	1	0	0	0	—	—	0	0	1	1	1	С (ошибка)	новый	●
Номер				Значение																																																																											
SD16				дисковод																																																																											
SD17				название файла (ASCII-код: 8 знаков)																																																																											
SD18																																																																															
SD19																																																																															
SD20																																																																															
SD21				расширение      2Ен(,)																																																																											
SD22				(ASCII-код: 3 знака)																																																																											
SD23				свободно																																																																											
SD24																																																																															
SD25																																																																															
SD26																																																																															
Номер	Значение																																																																														
SD16	время: с шагом 1 мкс (0 – 999 мкс)																																																																														
SD17	время: с шагом 1 мс (0 – 65535 мс)																																																																														
SD18	свободно																																																																														
SD19																																																																															
SD20																																																																															
SD21																																																																															
SD22																																																																															
SD23																																																																															
SD24																																																																															
SD25																																																																															
SD26																																																																															
Номер		Значение																																																																													
SD16	название файла (ASCII-код: 8 знаков)																																																																														
SD17																																																																															
SD18																																																																															
SD19																																																																															
SD20	расширение      2Ен(,)																																																																														
SD21	(ASCII-код: 3 знака)																																																																														
SD22	конфигурация битов *																																																																														
SD23	№ блока																																																																														
SD24	№ шага/перехода																																																																														
SD25	№ шага программы (L)																																																																														
SD26	№ шага программы (H)																																																																														
15	14	—	—	4	3	2	1	0																																																																							
0	0	—	—	0	0	1	1	1																																																																							
SD26																																																																															

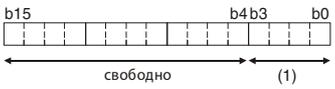
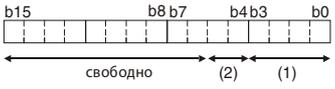
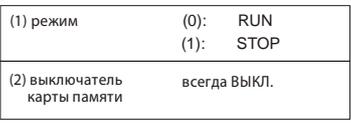
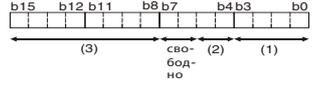
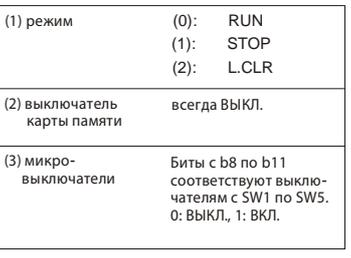
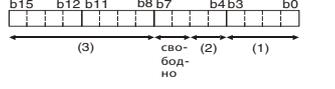
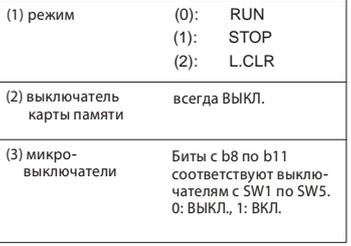
Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистрпроц. серии "А" D9 [][][]	Дейст. для:																																					
SD16	Специфическая информация об ошибках		(4) № параметра (5). № маркера ош. № ошибки команды CHK	С (ошибка)	новый	●																																					
SD17																																											
SD18																																											
SD19																																											
SD20																																											
SD21																																											
SD22																																											
SD23																																											
SD24																																											
SD25																																											
SD26																																											
SD26								(6) Ошибка параметрирования специальных модулей (только в случае процессоров MELSEC "System Q")																																			
								<table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>№ параметра</td> <td>SD16</td> <td>№ параметра</td> </tr> <tr> <td>SD17</td> <td rowspan="10">не используется</td> <td>SD17</td> <td rowspan="10">не используется</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>№ параметра</td> </tr> <tr> <td>SD17</td> <td>код ошибки для спец. модуля</td> </tr> <tr> <td>SD18</td> <td rowspan="9">не использ.</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table>	Номер	Значение	Номер	Значение	SD16	№ параметра	SD16	№ параметра	SD17	не используется	SD17	не используется	SD18	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	Номер	Значение	SD16	№ параметра	SD17	код ошибки для спец. модуля	SD18	не использ.	SD19	SD20	SD21	SD22	SD23	SD24
Номер	Значение	Номер	Значение																																								
SD16	№ параметра	SD16	№ параметра																																								
SD17	не используется	SD17	не используется																																								
SD18																																											
SD19																																											
SD20																																											
SD21																																											
SD22																																											
SD23																																											
SD24																																											
SD25																																											
SD26																																											
Номер	Значение																																										
SD16	№ параметра																																										
SD17	код ошибки для спец. модуля																																										
SD18	не использ.																																										
SD19																																											
SD20																																											
SD21																																											
SD22																																											
SD23																																											
SD24																																											
SD25																																											
SD26																																											
SD50	Сброс ошибки	Номер сброшенной ошибки	Сохраняет номер сброшенной ошибки	П	новый																																						
SD51	Слишком низкое напряжение батареи (фиксируемый маркер)	Набор битов, показывающий, где произошло падение напряжения батареи	<ul style="list-style-type: none"> <li>● Соответствующие биты устанавливаются, если упало напряжение батареи.</li> <li>● Этот бит остается установленным, даже если напряжение батареи снова достигло нормального уровня.</li> </ul> <p>В процессорах Q00J, Q00 и Q01 устанавливается только бит 0.</p>	С (ошибка)	новый																																						
SD52	Низкое напряжение батареи	Набор битов, показывающий, где снизилось напряжение батареи	<ul style="list-style-type: none"> <li>● Действует аналогично вышеописанному регистру SD51 (см. выше)</li> <li>● Этот бит сбрасывается после того, как напряжение батареи достигло нормального значения.</li> </ul>	С (ошибка)	новый																																						
SD53	Падение напряжения питания	Количество падений напряжения	<ul style="list-style-type: none"> <li>● При каждом падении напряжения во время работы более чем на 20 % от номинального напряжения значение этого регистра повышается на „1“. Значение сохраняется в двоичном виде.</li> </ul>	С (ошибка)	D9005	● Rem																																					

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистрпроц. серии "А" D9 [][][][]	Дейст. для:
SD54	Ошибка связи по MINI	Состояние обнаружения ошибки	<p>(1) Устанавливается бит соответствующей станции, если устанавливается один из головных адресов установленного модуля MINI (-S3) <math>X(n+0)/X(n+20), X(n+6)/(0n+26), X(n+7)/(n+27)</math> или <math>X(n+8)/Xn+28</math>.</p> <p>(2) Устанавливается соответствующий бит, если коммуникация между установленным модулем MINI (-S3) и центральным процессором не возможна.</p>	С (ошибка)	D9004 изменился формат	QnA
SD60	Номер неисправного предохранителя	Номер модуля, в котором неисправен предохранитель	Сохраненное здесь значение является самым нижним адресом станции модуля, в котором неисправен предохранитель, после деления этого адреса на 16.	С (ошибка)	D9000	● Rem
SD61	Ошибка сверки модуля ввода-вывода	Номер модуля, в котором имеется ошибка сверки	Самый низкий адрес модуля, в котором распознана первая ошибка сверки.	С (ошибка)	D9002	
SD62	№ маркера ошибки		Здесь сохраняется номер ошибки, обнаруженной первой.	С (выполнение команды)	D9009	●
SD63	Количество маркеров ошибок		Количество маркеров ошибок.	С (выполнение команды)	D9124	



Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [][][]	Дейст. для:	
SD90	Настройка контрольного таймера для контроля шагов и переходов (возможно только при наличии программы на языке SFC)	Номер маркера для ошибки настройки таймера и превышения времени	соответствует SM90	<ul style="list-style-type: none"> <li>● Номер маркера ошибки, устанавливаемого при указании неправильного времени для контрольного таймера или превышения времени контрольного таймера.</li> </ul> <div style="text-align: center;"> </div> <ul style="list-style-type: none"> <li>● Таймер запускается, если установлены маркеры диагностики SM90...SM99 и шаг активен. Если на протяжении настроенного времени условие дальнейшего переключения соответствующего шага не выполняется, устанавливается маркер ошибки (F).</li> </ul>	П	D9108	● кроме Q00, Q00 и Q01
SD91			соответствует SM91			D9109	
SD92			соответствует SM92			D9110	
SD93			соответствует SM93			D9111	
SD94			соответствует SM94			D9112	
SD95			соответствует SM95			D9113	
SD96			соответствует SM96			D9114	
SD97			соответствует SM97			новый	
SD98			соответствует SM98			новый	
SD99			соответствует SM99			новый	
SD100	Скорость передачи	Память для настроенной скорости передачи последовательного интерфейса	K96: 9600 бит/с, K192: 19,2 кбит/с, K384: 38,4 кбит/с, K576: 57,6 кбит/с, K1152: 115,2 кбит/с	С (при включении напряжения питания или после сброса)	новый	Q00, Q00 и Q01	
SD101	Настройки коммуникации	Память для настроек последовательной коммуникации	бит 4 = выкл.: без контрольной суммы бит 4 = вкл.: с контрольной суммой бит 5 = выкл.: изменение программы онлайн не допускается бит 5 = вкл.: изменение программы онлайн разрешено Прочие биты не имеют значения.		новый		
SD102	Время ожидания	Память для времени ожидания при последовательной коммуникации	0: без времени ожидания от 1 до F <sub>H</sub> : время ожидания в единицах по 10 мс Предварительная настройка: 0		новый		
SD105	Скорость передачи для CH1 (RS232)	Память для настроенной скорости передачи.	K3: 300 бит/с, K6: 600 бит/с, K24: 2400 бит/с, K48: 4800 бит/с, K96: 9600 бит/с, K192: 19,2 кбит/с, K384: 38,4 кбит/с, K576: 57,6 кбит/с, K1152: 115,2 кбит/с	С	новый	Q кроме Q00, Q00 и Q01	
SD110	Результат передачи	Код ошибки при передаче данных	Если при передаче данных путем последовательной коммуникации возникла ошибка, то здесь сохраняется код ошибки.	С (ошибка)	новый	Q00, Q00 и Q01	
SD111	Результат приема	Код ошибки при приеме данных	Если при приеме данных путем последовательной коммуникации возникла ошибка, то здесь сохраняется код ошибки.	С (ошибка)	новый		
SD120	Номер ошибки при исчезновении внешнего напряжения питания	Номер модуля, в котором исчезло внешнее электропитание	Сохраняется самый низкий адрес модуля "System Q", в котором исчезло напряжение питания. (в стадии подготовки)	С (ошибка)	новый	Q кроме Q00, Q00 и Q01	

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																																																				
SD130	Модули с неисправным предохранителем	Набор битов (16 бит) показывает модули с неисправным предохранителем 0: Неисправных предохранителей нет 1: Имеется неисправный предохранитель	<ul style="list-style-type: none"> <li>● Количество выходных модулей с неисправным предохранителем сохраняется в виде набора из 16 битов. (Если номер модуля установлен в параметрах, то сохраняется этот номер.)</li> <li>● Неисправные предохранители распознаются и в модулях вывода удаленных станций.</li> <li>● После замены неисправного предохранителя соответствующий бит не сбрасывается автоматически. Его необходимо стереть путем сброса сообщения об ошибке.</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: 0 auto;"> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>SD130</td> <td>0</td><td>0</td><td>0</td><td>1 (Y1C0)</td><td>0</td><td>0</td><td>0</td><td>1 (Y1B0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>SD131</td> <td>1 (Y1F0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y1A)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>   <table border="1" style="margin: 0 auto;"> <tr> <td>SD137</td> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y1F20)</td><td>0</td><td>0</td> </tr> </table> <p style="text-align: center;">↑ В модуле с адресом Y1F80 неисправен предохранитель.</p> </div>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD130	0	0	0	1 (Y1C0)	0	0	0	1 (Y1B0)	0	0	0	0	0	0	0	0	SD131	1 (Y1F0)	0	0	0	0	0	1 (Y1A)	0	0	0	0	0	0	0	0	0	SD137	0	0	0	0	1	0	0	0	0	0	0	0	0	1 (Y1F20)	0	0	С (ошибка)	новый	Q00, Q00 и Q01
				b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																							
SD130				0	0	0	1 (Y1C0)	0	0	0	1 (Y1B0)	0	0	0	0	0	0	0	0																																																							
SD131				1 (Y1F0)	0	0	0	0	0	1 (Y1A)	0	0	0	0	0	0	0	0	0																																																							
SD137				0	0	0	0	1	0	0	0	0	0	0	0	0	1 (Y1F20)	0	0																																																							
SD131																																																																										
SD132																																																																										
SD133																																																																										
SD134																																																																										
SD135																																																																										
SD136																																																																										
SD137																																																																										
SD150	Модули ввода-вывода с ошибкой сверки	Набор битов (16 бит), показывает модули с ошибкой сверки 0: ошибок сверки модулей ввода-вывода нет 1: имеется ошибка сверки модуля ввода-вывода	<ul style="list-style-type: none"> <li>● Если текущее состояние модуля ввода-вывода отличается от заданного состояния после включения напряжения питания, то информация этого модуля ввода-вывода сохраняется в регистре. (Если в параметрах установлен номер модуля, то сохраняется этот номер.)</li> <li>● Распознается также информация модуля ввода-вывода.</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: 0 auto;"> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>SD1350</td> <td>0</td><td>0</td><td>0</td><td>1 (Y1C0)</td><td>0</td><td>0</td><td>0</td><td>1 (Y1B0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>SD1351</td> <td>1 (Y1F0)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y1A)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>   <table border="1" style="margin: 0 auto;"> <tr> <td>SD1381</td> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1 (Y1F20)</td><td>0</td><td>0</td> </tr> </table> <p style="text-align: center;">↑ В модуле с адресом Y1F80 нет внешнего электропитания.</p> </div>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD1350	0	0	0	1 (Y1C0)	0	0	0	1 (Y1B0)	0	0	0	0	0	0	0	0	SD1351	1 (Y1F0)	0	0	0	0	1 (Y1A)	0	0	0	0	0	0	0	0	0	0	SD1381	0	0	0	0	1	0	0	0	0	0	0	0	0	1 (Y1F20)	0	0	С (ошибка)	новый	
				b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																							
SD1350				0	0	0	1 (Y1C0)	0	0	0	1 (Y1B0)	0	0	0	0	0	0	0	0																																																							
SD1351				1 (Y1F0)	0	0	0	0	1 (Y1A)	0	0	0	0	0	0	0	0	0	0																																																							
SD1381				0	0	0	0	1	0	0	0	0	0	0	0	0	1 (Y1F20)	0	0																																																							
SD151																																																																										
SD152																																																																										
SD153																																																																										
SD154																																																																										
SD155																																																																										
SD156																																																																										
SD157																																																																										

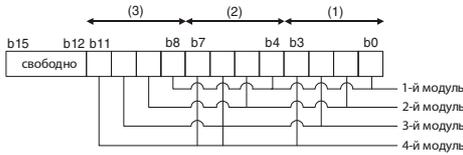
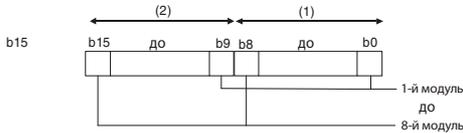
Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9[ ][ ][ ]	Дейст. для:
SD200	Коммутационное состояние	Состояние переключателя режимов на модуле центрального процессора	<ul style="list-style-type: none"> <li>Состояние переключателя режимов сохраняется в следующем формате:</li> </ul>  	С (постоянно)	новый	Rem
			<ul style="list-style-type: none"> <li>Состояние переключателя режимов сохраняется в следующем формате:</li> </ul>  		новый	Q00J, Q00 и Q01
			<ul style="list-style-type: none"> <li>Состояние переключателя режимов сохраняется в следующем формате:</li> </ul>  	С (обработка команды END)	новый	Q кроме Q00J, Q00 и Q01
			<ul style="list-style-type: none"> <li>Состояние выключателя с ключом на модуле центрального процессора сохраняется в следующем формате:</li> </ul>  	С (обработка команды END)	новый	QpA

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [][][]	Дейст. для:
SD201	Состояние светодиода	Состояние светодиодного индикатора на модуле центрального процессора	<ul style="list-style-type: none"> <li>Нижеприведенная информация относится к на светодиодным индикаторам центрального процессора.</li> </ul> <p>(1) : RUN (5) : BOOT                      (2) : ERROR (6) : свободно                      (3) : USER (7) : свободно                      (4) : BAT.ALARM (8) : режим</p> <p>Режим сохраняется в виде следующего набора битов: 0: не горит, 1: зеленый, 2: оранжевый</p> <p>В процессорах Q00J, Q00 и Q01 имеются только области 1 и 2.</p>	C (изменение состояния)	новый	Q
			<ul style="list-style-type: none"> <li>Нижеприведенная информация относится к светодиодным индикаторам центрального процессора и сохраняется в виде следующего набора битов:</li> <li>выкл. при 0; вкл. при 1; мигание при 2</li> </ul> <p>(1) : RUN (5) : BOOT                      (2) : ERROR (6) : Card A (карта памяти "А")                      (3) : USER (7) : Card B (карта памяти "В")                      (4) : BAT.ALARM (8) : свободно</p>	C (изменение состояния)	новый	QnA
SD202	Выключенные светодиоды	Набор битов выключенных светодиодов	<ul style="list-style-type: none"> <li>Сохраняет набор битов выключенных светодиодов (возможно только в отношении светодиодов "USER" и "BOOT")</li> <li>выключен при 0, включен при 1</li> </ul>	П	новый	QnA
SD203	Состояние обработки центрального процессора		<ul style="list-style-type: none"> <li>Состояние обработки сохраняется в следующем формате:</li> </ul> <p>(1) состояние обработки децентрализованных модулей ввода-вывода всегда 2: STOP</p>	C (постоянно)	новый	Rem
			<ul style="list-style-type: none"> <li>Состояние обработки центрального процессора сохраняется в следующем формате.</li> </ul> <p>(1) : Сост. обраб. 0: RUN                      центр. проц. 1: STEP-RUN (кроме Q00J, Q00 и Q01)                      2: STOP                      3: PAUSE</p> <p>(2) : Режим STOP/PAUSE вызван                      0: переключателем режимов                      1: дистанционным контактом                      2: периферийным устройством, соединением с компьютером или другим удаленным источником                      3: внутренними прогр. командами                      4: ошибкой</p> <p>Примечание: сохраняется только ошибка, возникшая первой.</p>	C (обработка команды END)	D9015 (изменился формат)	●

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9[ ][ ][ ]	Дейст. для:																				
SD206	Вид тестирования операндов	Отображение выполненного теста операндов	<ul style="list-style-type: none"> <li>При тестировании операндов с помощью программатора этот регистр содержит информацию о том, какие операнды были протестированы.</li> <li>0: тестирование операндов не активировано</li> <li>1: тестирование входов (X)</li> <li>2: тестирование выходов (Y)</li> <li>3: тестирование входов и выходов (X/Y)</li> </ul>	С (запрос)	новый	Rem																				
SD207	Приоритет индикации светодиода "ERR"	Приоритет от 1 до 4	<ul style="list-style-type: none"> <li>При возникновении ошибки она отображается миганием светодиодного индикатора в соответствии с номером ошибки, сохраненным в регистрах.</li> <li>Для приоритетов индикации возможны следующие настройки:</li> </ul> <table border="1" style="margin-left: 20px;"> <tr> <td>B15</td> <td>B12 B11</td> <td>B8 B7</td> <td>B4 B3</td> <td>B0</td> </tr> <tr> <td>SD207</td> <td>приор. 4</td> <td>приор. 3</td> <td>приор. 2</td> <td>приор. 1</td> </tr> <tr> <td>SD208</td> <td>приор. 8</td> <td>приор. 7</td> <td>приоритет 6</td> <td>приор. 5</td> </tr> <tr> <td>SD209</td> <td colspan="2"></td> <td>приор. 10</td> <td>приор. 9</td> </tr> </table> <p style="margin-left: 20px;">Заводская настройка: (4321н) (8765 н) (00A9 н)</p> <ul style="list-style-type: none"> <li>Если настройка равна „0“, индикация не происходит. Но даже при настройке "0" светодиод отображает информацию об ошибке, вызвавшую останов центрального процессора (отображаются также настройки параметров).</li> </ul>	B15	B12 B11	B8 B7	B4 B3	B0	SD207	приор. 4	приор. 3	приор. 2	приор. 1	SD208	приор. 8	приор. 7	приоритет 6	приор. 5	SD209			приор. 10	приор. 9	П	D9038	<ul style="list-style-type: none"> <li>кроме Q00J, Q00 и Q01</li> </ul>
B15		B12 B11		B8 B7	B4 B3	B0																				
SD207		приор. 4		приор. 3	приор. 2	приор. 1																				
SD208	приор. 8	приор. 7	приоритет 6	приор. 5																						
SD209			приор. 10	приор. 9																						
SD208	Приоритет от 5 до 8	D9039 (изменился формат)																								
SD209	Приоритет от 9 до 10	новый																								
SD210	Данные времени	Данные времени (год, месяц)	<ul style="list-style-type: none"> <li>Год (последние 2 разряда) и месяц сохраняются в двоично-десятичной кодировке в регистре SD210:</li> </ul> <table border="1" style="margin-left: 20px;"> <tr> <td>b15</td> <td>b12 b11</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td colspan="2">год</td> <td colspan="3">месяц</td> </tr> </table> <p style="margin-left: 20px;">Пример: июль 1993г. = 9307</p>	b15	b12 b11	b8 b7	b4 b3	b0	год		месяц			С/П (запрос)	D9025	<ul style="list-style-type: none"> <li>Rem</li> </ul>										
b15		b12 b11	b8 b7	b4 b3	b0																					
год		месяц																								
SD211	Данные времени (день, час)	<ul style="list-style-type: none"> <li>День и час сохраняются в двоично-десятичной кодировке в регистре SD211:</li> </ul> <table border="1" style="margin-left: 20px;"> <tr> <td>b15</td> <td>b12 b11</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td colspan="2">день</td> <td colspan="3">час</td> </tr> </table> <p style="margin-left: 20px;">Пример: 31-е число, 10 часов = 3110</p>	b15	b12 b11	b8 b7	b4 b3	b0	день		час			D9026													
b15	b12 b11	b8 b7	b4 b3	b0																						
день		час																								
SD212	Данные времени (минута, секунда)	<ul style="list-style-type: none"> <li>Минуты и секунды сохраняются в двоично-десятичной кодировке в регистре SD212:</li> </ul> <table border="1" style="margin-left: 20px;"> <tr> <td>b15</td> <td>b12 b11</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td colspan="2">минута</td> <td colspan="3">секунда</td> </tr> </table> <p style="margin-left: 20px;">Пример: 35 минут, 48 секунд = 3548</p>	b15	b12 b11	b8 b7	b4 b3	b0	минута		секунда			D9027													
b15	b12 b11	b8 b7	b4 b3	b0																						
минута		секунда																								

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ]	Дейст. для:		
SD213	Данные времени	Данные времени (день недели)	<ul style="list-style-type: none"> <li>День недели сохраняется в двоично-десятичной кодировке в регистре SD213.</li> </ul> <p>старшие разряды числа года (от 0 до 99)</p>	С/П (запрос)	D9028	Q Rem		
			<ul style="list-style-type: none"> <li>День недели сохраняется в двоично-десятичной кодировке в регистре SD213.</li> </ul> <p>всегда "0"</p>	С/П (запрос)		QnA		
SD220	Данные светодиодного дисплея	Данные индикации на дисплее	<ul style="list-style-type: none"> <li>Данные в формате ASCII (16 знаков) светодиодного дисплея сохраняются в нижеуказанных регистрах.</li> </ul>	С (изменение состояния)	новый	●		
SD221			SD220				15-й знак справа	16-й знак справа
SD222			SD221				13-й знак справа	14-й знак справа
SD223			SD222				11-й знак справа	12-й знак справа
SD224			SD223				9-й знак справа	10-й знак справа
SD226			SD224				7-й знак справа	8-й знак справа
SD227			SD225				5-й знак справа	6-й знак справа
			SD226				3-й знак справа	4-й знак справа
SD227	SD227	1-й знак справа	2-й знак справа					
SD240	Режим монтажной шины	0: автоматический режим 1: детализированный режим	Этот регистр служит для сохранения режима монтажной шины.	С (инициализация)	новый	Q Rem		
SD241	Количество расширительных монтажных шин	0: только главная монтажная шина 1 до 7: количество расширительных монтажных шин	В этом регистре сохраняется количество установленных расширительных монтажных шин.	С (инициализация)	новый			

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																	
SD242	Различение монтажных шин "А" и "Q"	0: Установлена шина типа QA [ ] [ ] B (режим А) 1: Установлена шина типа Q [ ] [ ] B (режим Q)	<p>Если расширительная монтажная шина не подсоединена, биты с 1-го по 4-й имеют состояние "0".</p>	С (инициализация)	новый	Q00J, Q00 и Q01																	
			<p>Если расширительная монтажная шина не подсоединена, биты с 1-го по 7-й имеют состояние "0"</p>			Q02, Q06H, Q12H, Q25H; Rem																	
SD243	Количество слотов на монтажных шинах	Количество слотов на монтажных шинах В случае процессоров Q00J, Q00 и Q01 разряды с 5-го по 7-й расширительной шины заняты нулями.		С (инициализация)	новый	Q																	
SD244			<table border="1"> <tr> <td>bF</td> <td>bC</td> <td>bB</td> <td>b8</td> <td>b7</td> <td>b4</td> <td>b3</td> <td>b0</td> </tr> <tr> <td>3-я РМШ</td> <td>2-я РМШ</td> <td>1-я РМШ</td> <td colspan="4">ГМШ</td> </tr> <tr> <td>7-я РМШ</td> <td>6-я РМШ</td> <td>5-я РМШ</td> <td colspan="4">4-я РМШ</td> </tr> </table> <p>Количество слотов указывается для главной монтажной шины (ГМШ) и расширительных монтажных шин (РМШ) в соответствующих областях.</p>				bF	bC	bB	b8	b7	b4	b3	b0	3-я РМШ	2-я РМШ	1-я РМШ	ГМШ				7-я РМШ	6-я РМШ
bF	bC	bB	b8	b7	b4	b3	b0																
3-я РМШ	2-я РМШ	1-я РМШ	ГМШ																				
7-я РМШ	6-я РМШ	5-я РМШ	4-я РМШ																				
SD250	Загружено максимальное число входов и выходов	Загружено максимальное количество входов и выходов	Если SM250 установлен, к двум старшим разрядам последнего загруженного адреса модуля ввода-вывода прибавляется 1 и результат сохраняется в виде двоичного значения.	С (обработка команды END)	новый	●																	
SD251	Адрес заменяемого модуля ввода-вывода	Начальный адрес модуля ввода-вывода	Регистр D9094 сохраняет в виде двоичного значения два старших разряда начального адреса модуля ввода-вывода, который во время режима онлайн извлекается из монтажной шины или вставляется в нее.	П	D9094	Q2A(S1), Q3A, Q4A, Q4AR																	
SD253	Скорость передачи для RS422	0: 9600 бит/с 1: 19,2 кбит/с 2: 38,4 кбит/с	Этот регистр сохраняет скорость передачи для интерфейса RS422.	С (при изменении)	новый	QpA																	

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9[ ][ ][ ]	Дейст. для:	
SD254	Информация MELSECNET/10	Количество установленных модулей	Показывает количество модулей, установленных в сети MELSECNET/10.	С (инициализация)	новый	●	
SD255		Информация первого модуля	адрес ввода-вывода				Адрес ввода-вывода первого модуля, установленного в сети MELSECNET/10.
SD256			номер сети				Сетевой адрес первого модуля, установленного в сети MELSECNET/10.
SD257			номер группы				Номер группы первого модуля, установленного в сети MELSECNET/10.
SD258			номер станции				Номер станции первого модуля, установленного в сети MELSECNET/10.
SD259			информация о резервировании				Если имеются резервные станции, сохраняется номер модуля резервной станции (от 1 до 4).
SD260 – SD264		Информация второго модуля	Конфигурация идентична первому модулю.				
SD265 – SD269		Информация третьего модуля					
SD270 – SD274		Информация четвертого модуля					
SD280	Ошибка CC-Link	Состояние при обнаружении ошибки	 <p>(1) При включении Xn0 установленной станции CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(2) Если выключается Xn1 или XnF установленной станции CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(3) Биты в этой области устанавливаются, если центральный процессор не может коммуницировать с установленной станцией CC-Link.</p>	С (при ошибке)	новый	Q Rem	
			 <p>(1) Если включается Xn0 установленной станции CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(2) Если выключается Xn1 или XnF установленной сети CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(3) Биты в этой области устанавливаются, если центральный процессор не может коммуницировать с установленной станцией CC-Link.</p>	С (при ошибке)	новый	QnA	

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9[1][1]	Дейст. для:	
SD290	Присвоение операндов (идентично содержимому параметров)	Число адресов операнда X	Настроенное в данный момент число адресов операндов X	С (инициализация)	новый	● Rem	
SD291		Число адресов операндов Y	Настроенное в данный момент число адресов операндов Y				
SD292		Число адресов операндов M	Настроенное в данный момент число адресов операндов M				
SD293		Число адресов операндов L	Настроенное в данный момент число адресов операндов L				
SD294		Число адресов операндов B	Настроенное в данный момент число адресов операндов B		новый	● Rem	
SD295		Число адресов операндов F	Настроенное в данный момент число адресов операндов F		новый	●	
SD296		Число адресов операндов SB	Настроенное в данный момент число адресов операндов SB		новый	● Rem	
SD297		Число адресов операндов V	Настроенное в данный момент число адресов операндов V		новый	●	
SD298		Число адресов операндов S	Настроенное в данный момент число адресов операндов S				
SD299		Число адресов операндов T	Настроенное в данный момент число адресов операндов T				
SD300		Число адресов операндов ST	Настроенное в данный момент число адресов операндов ST				
SD301		Число адресов операндов C	Настроенное в данный момент число адресов операндов C		(инициализация)	новый	● Rem
SD302		Число адресов операндов D	Настроенное в данный момент число адресов операндов D				
SD303		Число адресов операндов W	Настроенное в данный момент число адресов операндов W				
SD304	Число адресов операндов SW	Настроенное в данный момент число адресов операндов SW					
SD315	Время, зарезервированное для коммуникации	Время, зарезервированное для коммуникации.	Введенное здесь время (диапазон от 1 до 100 мс) выделено для коммуникации с программатором. Чем большее значение здесь введено, тем короче время, имеющееся для реакции при коммуникации с другими приборами (например, через последовательное соединение). Если это значение находится вне допустимого диапазона, оно игнорируется. Время цикла удлиняется на настроенное время.	Обработка команды END	новый	Q	

## Системные часы и счетчики

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD412	1-секундный счетчик	Считает с секундным шагом	<ul style="list-style-type: none"> <li>● При начале режима "RUN" центрального процессора счетчик начинает считать с секундным тактом.</li> <li>● Счетчик считает вверх от 0 до 32767, затем перескакивает на -32768 и считает до 0.</li> </ul>	С (изменение состояния)	D9022	●
SD414	Такт в 2п секунд	Единицы по 2п секунд	<ul style="list-style-type: none"> <li>● Сохраняет настройку для п, на основе которой рассчитывается такт в 2п секунд (предварительная настройка = 30).</li> <li>● Возможны значения между 1 и 32767.</li> </ul>	П	новый	
SD415	Такт в 2п миллисекунд	Единицы по 2п миллисекунд	<ul style="list-style-type: none"> <li>● Сохраняет настройку для п, на основе которой вычисляется такт в 2п миллисекунд (предварительная настройка = 30).</li> <li>● Возможны значения между 1 и 32767.</li> </ul>	П	новый	Q02, Q02H, Q06H, Q12PH, Q12PN, Q25H, Q25PH
SD420	Счетчик программных циклов	Считает количество программных циклов	<ul style="list-style-type: none"> <li>● После начала режима "RUN" центрального процессора счетчик при каждом программном цикле повышается на 1.</li> <li>● Счетчик считает вверх от 0 до 32767, затем перескакивает на -32768 и считает до 0.</li> </ul>	С (обработка команды END)	новый	●
SD430	Счетчик программных циклов замедленной обработки	Считает количество программных циклов, выполненных на пониженной скорости	<ul style="list-style-type: none"> <li>● После включения центрального процессора в режиме RUN счетчик при каждом программном цикле прирастает на 1.</li> <li>● Счетчик считает вверх от 0 до 32767, затем перескакивает на -32768 и считает до 0.</li> <li>● Этот счетчик можно использовать только для программ, выполняемых в замедленном режиме („Low Speed Execution“).</li> </ul>	С (обработка команды END)	новый	● кроме Q00J, Q00 и Q01

### А.3.1 Информация цикла программы

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [1][1]	Дейст. для:
SD500	Номер выполняемой программы	Тип выполнения программы, выполняемой в данный момент	<ul style="list-style-type: none"> <li>Номер выполняемой в данный момент программы сохраняется в виде двоичного значения.</li> </ul>	С (изменение состояния)	новый	● кроме Q00, Q00 и Q01
SD510	Номер программы "Low Speed Execution"	Название файла программы	<ul style="list-style-type: none"> <li>Номер программы, выполняемой в данный момент в замедленном режиме (Low Speed Execution), сохранен в виде двоичного значения.</li> <li>Это возможно только при установленном маркере SM510.</li> </ul>	С (обработка команды END)	новый	
SD520	Текущее время цикла	Время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет текущее время цикла программы (с шагом в 1 мс) в диапазоне от 0 до 65535.</li> <li>Сохраняет текущее время цикла программы (с шагом в 1 мкс) в диапазоне от 00000 до 900. Пример: время цикла программы 23,6 мс сохраняется следующим образом: D520 = 23 D521 = 600</li> </ul>	С (обработка команды END)	D9017 (изменился формат)	●
SD521		Время цикла (единица 1 мкс)			новый	
SD522	Время цикла инициализации	Время цикла инициализации (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время первого программного цикла (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD523		Время цикла инициализации (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время первого программного цикла (с шагом в 1 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD524	Минимальное время цикла	Минимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет минимальное время цикла программы (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	D9018 (изменился формат)	●
SD525		Минимальное время цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет минимальное время цикла программы (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>		новый	
SD526	Максимальное время цикла	Максимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет максимальное время цикла программы (с шагом в 1 мс), за исключением первого цикла.</li> <li>Диапазон от 0 до 65535</li> </ul>	С (обработка команды END)	D9019 (изменился формат)	●
SD527		Максимальное время цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет максимальное время цикла программы (с шагом в 100 мкс), за исключением первого цикла.</li> <li>Диапазон от 000 до 900</li> </ul>		новый	
SD528	Время цикла для программ, выполняемых в режиме "Low Speed Execution"	Текущее время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет текущее время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD529		Текущее время цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет текущее время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD532	Минимальное время цикла для программ, выполняемых в режиме "Low Speed Execution"	Минимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет минимальное время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD533		Минимальное время цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет минимальное время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD534	Максимальное время цикла для программ, выполняемых в режиме "Low Speed Execution"	Максимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет максимальное время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс, за исключением 1-го цикла).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD535		Максимальное время цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет максимальное время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс), за исключением 1-го цикла.</li> <li>Диапазон от 000 до 900.</li> </ul>			

## Информация цикла программы

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD540	Время обработки команды END	Время обработки команды END (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время от конца последнего программного цикла до начала следующего цикла (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	
SD541		Время обработки команды END (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время от конца последнего программного цикла до начала следующего цикла (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD542	Время ожидания при постоянном времени цикла	Время ожидания при постоянном времени цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время ожидания при установленном постоянном времени цикла (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (первая END)	новый	●
SD543		Время ожидания при постоянном времени цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время ожидания при установленном постоянном времени цикла (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD544	Суммарное время выполнения программ в режиме "Low Speed Execution"	Суммарное время выполнения программ, выполняемых в замедленном режиме (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет суммарное время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD545		Суммарное время выполнения программ, выполняемых в режиме "Low Speed Execution" (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет суммарное время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD546	Время выполнения программ, выполняемых в режиме "Low Speed Execution"	Время выполнения программ, выполняемых в режиме "Low Speed Execution" (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс), на протяжении одного цикла.</li> <li>Диапазон от 0 до 65535.</li> <li>Сохранение происходит каждый цикл.</li> </ul>	С (обработка команды END)	новый	
SD547		Время выполнения для программ, выполняемых в режиме "Low Speed Execution" (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс), на протяжении одного цикла.</li> <li>Диапазон от 000 до 900.</li> <li>Сохранение происходит каждый цикл.</li> </ul>			
SD548	Время выполнения программ, выполняемых в режиме "Scan Execution"	Время выполнения для программ, выполняемых в режиме "Scan Execution" (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в режиме "Scan Execution" (с шагом в 1 мс), на протяжении одного цикла.</li> <li>Диапазон от 0 до 65535.</li> <li>Сохранение происходит каждый цикл.</li> </ul>	С (обработка команды END)	новый	●
SD549		Время выполнения для программ, выполняемых в режиме "Scan Execution" (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в режиме "Scan Execution" (с шагом в 100 мкс), на протяжении одного цикла.</li> <li>Диапазон от 000 до 900.</li> <li>Сохранение происходит каждый цикл.</li> </ul>			
SD550	Измерение интервала сервисного обслуживания для модулей	№ станции/модуля	<ul style="list-style-type: none"> <li>Устанавливает адрес ввода-вывода модуля, для которого измеряется интервал сервисного обслуживания.</li> </ul>	П	новый	
SD551	Интервал сервисного обслуживания	Интервал сервисного обслуживания модуля (единица 1 мс)	<ul style="list-style-type: none"> <li>Если маркер SM551 установлен, то сохраняется интервал, по истечении которого должно быть выполнено обслуживание модуля, указанного в SD550 (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	С (запрос)	новый	● кроме Q00, Q00 и Q01
SD552		Интервал сервисного обслуживания модуля (единица 100 мкс)	<ul style="list-style-type: none"> <li>Если SM551 установлен, то сохраняется интервал, по истечении которого должно быть выполнено обслуживание модуля, указанного в SD550 (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			

**Карты памяти**

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																
SD600	Тип карты памяти "А"		<p>Показывает тип установленной карты памяти "А".</p>	С (при инициализации и извлечении карты памяти)	новый	серия "Q", кроме Q00, Q00 и Q01																
			<p>Показывает тип установленной карты памяти "А".</p>	С (при инициализации и извлечении карты памяти)	новый	QnA																
SD602	Емкость дисковода 1 (RAM)		Емкость дисковода 1 сохраняется с шагом в 1 кб.	С (при инициализации и извлечении карты памяти)	новый	● кроме Q00, Q00 и Q01																
SD603	Емкость дисковода 2 (ROM)		Емкость дисковода 2 сохраняется с шагом в 1 кб.	С (при инициализации и извлечении карты памяти)	новый																	
SD604	Условия использования карты памяти "А"		<ul style="list-style-type: none"> <li>Условия использования карты памяти "А" сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов:</li> </ul> <table border="1"> <tr> <td>b0 : процесс первонач. загр. (QBT)</td> <td>b8 : </td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок ЦП (QFD)</td> </tr> <tr> <td>b2 : комментарии к опер. (QCD)</td> <td>bA : трассир. языка (QTS)</td> </tr> <tr> <td>b3 : нач. значение операнда (QDI)</td> <td>bB : лок. переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>bC : </td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>bD : </td> </tr> <tr> <td>b6 : </td> <td>bE : </td> </tr> <tr> <td>b7 : </td> <td>bF : </td> </tr> </table>	b0 : процесс первонач. загр. (QBT)	b8 :	b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)	b2 : комментарии к опер. (QCD)	bA : трассир. языка (QTS)	b3 : нач. значение операнда (QDI)	bB : лок. переменная (QDL)	b4 : регистр файлов (QDR)	bC :	b5 : трассировка (QTS)	bD :	b6 :	bE :	b7 :	bF :	С (изменение состояния)	новый	серия "Q", кроме Q00, Q00 и Q01
			b0 : процесс первонач. загр. (QBT)	b8 :																		
b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)																					
b2 : комментарии к опер. (QCD)	bA : трассир. языка (QTS)																					
b3 : нач. значение операнда (QDI)	bB : лок. переменная (QDL)																					
b4 : регистр файлов (QDR)	bC :																					
b5 : трассировка (QTS)	bD :																					
b6 :	bE :																					
b7 :	bF :																					
			<ul style="list-style-type: none"> <li>Условия использования карты памяти "А" сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов:</li> </ul> <table border="1"> <tr> <td>b0 : процесс первонач. загр. (QBT)</td> <td>b8 : данные имитации</td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок ЦП (QFD)</td> </tr> <tr> <td>b2 : комментарии к опер. (QCD)</td> <td>b10 : трассир. языка SFC (QTS)</td> </tr> <tr> <td>b3 : нач. значение операнда (QDI)</td> <td>b11 : лок. переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>b12 : </td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>b13 : </td> </tr> <tr> <td>b6 : фиксация состояния (QTL)</td> <td>b14 : </td> </tr> <tr> <td>b7 : трассировка программы (QTP)</td> <td>b15 : </td> </tr> </table>	b0 : процесс первонач. загр. (QBT)	b8 : данные имитации	b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)	b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)	b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)	b4 : регистр файлов (QDR)	b12 :	b5 : трассировка (QTS)	b13 :	b6 : фиксация состояния (QTL)	b14 :	b7 : трассировка программы (QTP)	b15 :	С (изменение состояния)	новый	QnA
b0 : процесс первонач. загр. (QBT)	b8 : данные имитации																					
b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)																					
b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)																					
b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)																					
b4 : регистр файлов (QDR)	b12 :																					
b5 : трассировка (QTS)	b13 :																					
b6 : фиксация состояния (QTL)	b14 :																					
b7 : трассировка программы (QTP)	b15 :																					

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																
SD620	Тип карты памяти "В"		<ul style="list-style-type: none"> <li>Показывает тип установленной карты памяти "В".</li> </ul> <p>В связи с тем, что в процессор встроена флэш-РОМ, значение для диск. 4 неизменно установлено на "3".</p>	С (инициализация)	новый	Q																
			<ul style="list-style-type: none"> <li>Показывает тип установленной карты памяти "В".</li> </ul>	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR																
SD622	Емкость диск. 3 (RAM)		<ul style="list-style-type: none"> <li>Свободное место на диск. 3 регистрируется с шагом в 1 кб. В случае процессора серии "Q" с RAM емкостью 61 кб это значение неизменно установлено на "61".</li> </ul>	С (инициализация)	новый	Q																
			<ul style="list-style-type: none"> <li>Свободное место на диск. 3 регистрируется с шагом в 1 кб.</li> </ul>	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR																
SD623	Емкость диск. 4 (ROM)		<ul style="list-style-type: none"> <li>Емкость диск. 4 сохраняется с шагом в 1 кб.</li> </ul>	С (инициализация)	новый	серия "Q", Q2A(S1), Q3A, Q4A, Q4AR																
SD624	Условия использования диск. 3		<ul style="list-style-type: none"> <li>Условия использования диск. 3 отображаются с помощью бита 4: бит 4 = выкл.: не используется; бит 4 = вкл.: используется для хранения регистров файлов</li> </ul>	С (изменение состояния)	новый	Q00J, Q00 и Q01																
	Условия использования диск. 3 и 4		<ul style="list-style-type: none"> <li>Условия использования диск. 3 и 4 сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов разъяснено ниже.</li> </ul> <table border="1"> <tr> <td>b0 : процесс первонач. загр. (QBT)</td> <td>b8 :</td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок ЦП (QFD)</td> </tr> <tr> <td>b2 : комментарии к опер. (QCD)</td> <td>bA : трассир. языка (QTS)</td> </tr> <tr> <td>b3 : нач. значение операнда (QDI)</td> <td>bB : лок. переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>bC :</td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>bD :</td> </tr> <tr> <td>b6 :</td> <td>bE :</td> </tr> <tr> <td>b7 :</td> <td>bF :</td> </tr> </table>	b0 : процесс первонач. загр. (QBT)	b8 :	b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)	b2 : комментарии к опер. (QCD)	bA : трассир. языка (QTS)	b3 : нач. значение операнда (QDI)	bB : лок. переменная (QDL)	b4 : регистр файлов (QDR)	bC :	b5 : трассировка (QTS)	bD :	b6 :	bE :	b7 :	bF :	С (изменение состояния)	новый	серия "Q", кроме Q00J, Q00 и Q01
	b0 : процесс первонач. загр. (QBT)	b8 :																				
b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)																					
b2 : комментарии к опер. (QCD)	bA : трассир. языка (QTS)																					
b3 : нач. значение операнда (QDI)	bB : лок. переменная (QDL)																					
b4 : регистр файлов (QDR)	bC :																					
b5 : трассировка (QTS)	bD :																					
b6 :	bE :																					
b7 :	bF :																					
Условия использования карты памяти "В"		<ul style="list-style-type: none"> <li>Условия использования карты памяти "В" сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов разъяснено ниже.</li> </ul> <table border="1"> <tr> <td>b0 : процесс первонач. загр. (QBT)</td> <td>b8 : данные имитации</td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок ЦП (QFD)</td> </tr> <tr> <td>b2 : комментарии к опер. (QCD)</td> <td>b10 : трассир. языка SFC (QTS)</td> </tr> <tr> <td>b3 : нач. значение операнда (QDI)</td> <td>b11 : лок. переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>b12 :</td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>b13 :</td> </tr> <tr> <td>b6 : фиксация состояния (QTL)</td> <td>b14 :</td> </tr> <tr> <td>b7 : трассировка программы (QTP)</td> <td>b15 :</td> </tr> </table>	b0 : процесс первонач. загр. (QBT)	b8 : данные имитации	b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)	b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)	b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)	b4 : регистр файлов (QDR)	b12 :	b5 : трассировка (QTS)	b13 :	b6 : фиксация состояния (QTL)	b14 :	b7 : трассировка программы (QTP)	b15 :	С (изменение состояния)	новый	Q2A(S1), Q3A, Q4A, Q4AR	
b0 : процесс первонач. загр. (QBT)	b8 : данные имитации																					
b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)																					
b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)																					
b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)																					
b4 : регистр файлов (QDR)	b12 :																					
b5 : трассировка (QTS)	b13 :																					
b6 : фиксация состояния (QTL)	b14 :																					
b7 : трассировка программы (QTP)	b15 :																					

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																						
SD640	Дисковод регистра файлов	Номер дисковода	Сохраняет номер дисковода, используемого для регистра файлов.	С (изменение состояния)	новый																							
SD641	Регистр файлов Название файла		Сохраняет регистр файлов и название файла (с расширением) в ASCII-кодировке, указанные с помощью параметра или команды QCDSET.  <table border="1"> <thead> <tr> <th>b15</th> <th>b8 b7</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>SD641</td> <td>2-й знак</td> <td>1-й знак</td> </tr> <tr> <td>SD642</td> <td>4-й знак</td> <td>3-й знак</td> </tr> <tr> <td>SD643</td> <td>6-й знак</td> <td>5-й знак</td> </tr> <tr> <td>SD644</td> <td>8-й знак</td> <td>7-й знак</td> </tr> <tr> <td>SD645</td> <td>1-й знак расширения</td> <td>2EH (.)</td> </tr> <tr> <td>SD646</td> <td>3-й знак расширения</td> <td>2-й знак расширения</td> </tr> </tbody> </table>	b15	b8 b7	b0	SD641	2-й знак	1-й знак	SD642	4-й знак	3-й знак	SD643	6-й знак	5-й знак	SD644	8-й знак	7-й знак	SD645	1-й знак расширения	2EH (.)	SD646	3-й знак расширения	2-й знак расширения	С (изменение состояния)	новый	●	
b15				b8 b7	b0																							
SD641				2-й знак	1-й знак																							
SD642				4-й знак	3-й знак																							
SD643				6-й знак	5-й знак																							
SD644				8-й знак	7-й знак																							
SD645	1-й знак расширения	2EH (.)																										
SD646	3-й знак расширения	2-й знак расширения																										
SD642																												
SD643																												
SD644																												
SD645																												
SD646																												
SD647	Объем регистра файлов		Объем выбранного в данный момент регистра файлов в килословах.	С (изменение состояния)	новый																							
SD648	Номер блока регистра файлов		Сохраняет выбранный в данный момент номер блока регистра файлов.	С (изменение состояния)	D9035																							
SD650	Дисковод комментариев		Сохраняет номер дисковода комментариев, указанный с помощью параметра или команды QCDSET.	С (изменение состояния)	новый																							
SD651	Название файла комментариев		Сохраняет название файла (с расширением) в кодировке ASCII, указанное с помощью параметра или команды QCDSET.  <table border="1"> <thead> <tr> <th>b15</th> <th>b8 b7</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>SD651</td> <td>2-й знак</td> <td>1-й знак</td> </tr> <tr> <td>SD652</td> <td>4-й знак</td> <td>3-й знак</td> </tr> <tr> <td>SD653</td> <td>6-й знак</td> <td>5-й знак</td> </tr> <tr> <td>SD654</td> <td>8-й знак</td> <td>7-й знак</td> </tr> <tr> <td>SD655</td> <td>1-й знак расширения</td> <td>2EH (.)</td> </tr> <tr> <td>SD656</td> <td>3-й знак расширения</td> <td>2-й знак расширения</td> </tr> </tbody> </table>	b15	b8 b7	b0	SD651	2-й знак	1-й знак	SD652	4-й знак	3-й знак	SD653	6-й знак	5-й знак	SD654	8-й знак	7-й знак	SD655	1-й знак расширения	2EH (.)	SD656	3-й знак расширения	2-й знак расширения	С (изменение состояния)	новый	●	
b15				b8 b7	b0																							
SD651				2-й знак	1-й знак																							
SD652				4-й знак	3-й знак																							
SD653				6-й знак	5-й знак																							
SD654				8-й знак	7-й знак																							
SD655	1-й знак расширения	2EH (.)																										
SD656	3-й знак расширения	2-й знак расширения																										
SD652																												
SD653																												
SD654																												
SD655																												
SD656																												
SD660	Файл, указанный для процесса начальной загрузки	Номер дисковода, в котором находится указанный файл для начальной загрузки	Сохраняет номер дисковода, в котором находится файл (*.QBT), указанный для процесса загрузки.	С (инициализация)	новый	● кроме Q00, Q00 и Q01																						
SD661		Название файла, указанного для процесса начальной загрузки		Сохраняет название файла, указанного для процесса начальной загрузки (*.QBT).  <table border="1"> <thead> <tr> <th>b15</th> <th>b8 b7</th> <th>b0</th> </tr> </thead> <tbody> <tr> <td>SD661</td> <td>2-й знак</td> <td>1-й знак</td> </tr> <tr> <td>SD662</td> <td>4-й знак</td> <td>3-й знак</td> </tr> <tr> <td>SD663</td> <td>6-й знак</td> <td>5-й знак</td> </tr> <tr> <td>SD664</td> <td>8-й знак</td> <td>7-й знак</td> </tr> <tr> <td>SD665</td> <td>1-й знак расширения</td> <td>2EH (.)</td> </tr> <tr> <td>SD666</td> <td>3-й знак расширения</td> <td>2-й знак расширения</td> </tr> </tbody> </table>	b15		b8 b7	b0	SD661	2-й знак	1-й знак	SD662	4-й знак	3-й знак	SD663	6-й знак	5-й знак	SD664	8-й знак	7-й знак	SD665	1-й знак расширения	2EH (.)	SD666	3-й знак расширения	2-й знак расширения	С (инициализация)	новый
b15					b8 b7		b0																					
SD661					2-й знак		1-й знак																					
SD662					4-й знак		3-й знак																					
SD663					6-й знак		5-й знак																					
SD664	8-й знак	7-й знак																										
SD665	1-й знак расширения	2EH (.)																										
SD666	3-й знак расширения	2-й знак расширения																										
SD662																												
SD663																												
SD664																												
SD665																												
SD666																												

## Регистры, относящиеся к командам

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:												
SD705	Битовая схема		Во время обработки блока устанавливается маркер SM705. Это позволяет использовать сохраненную в регистре SD705 битовую схему (в случае применения двойных слов она сохранена в регистрах SD705 и SD706) для всех обрабатываемых данных блока.	П	новый	● кроме Q00, Q00 и Q01												
SD706																		
SD714	Количество свободных запросов коммуникации в области регистрации	0 ... 32	Количество свободных блоков в области коммуникационного запроса для удаленных специальных модулей, соединенных с AJ71PT32-S.	С (во время выполнения)	M9081	QnA												
SD715	Битовая схема команды IMASK	Битовая схема	При применении команды IMASK используется следующая битовая схема.  <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">.....</td> <td style="text-align: center;">b0</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">SD715</td> <td style="border: 1px solid black; padding: 2px;"> 15; 14; 13; 12; 11; 10; 9; 8; 7; 6; 5; 4; 3; 2; 1; 0</td> <td></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">SD716</td> <td style="border: 1px solid black; padding: 2px;"> 31; 30; 29; 28; 27; 26; 25; 24; 23; 22; 21; 20; 19; 18; 17; 16</td> <td></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">SD717</td> <td style="border: 1px solid black; padding: 2px;"> 47; 46; 45; 44; 43; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33; 32</td> <td></td> </tr> </table>	b15	.....	b0	SD715	15; 14; 13; 12; 11; 10; 9; 8; 7; 6; 5; 4; 3; 2; 1; 0		SD716	31; 30; 29; 28; 27; 26; 25; 24; 23; 22; 21; 20; 19; 18; 17; 16		SD717	47; 46; 45; 44; 43; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33; 32		С (во время выполнения)	новый	●
b15				.....	b0													
SD715				15; 14; 13; 12; 11; 10; 9; 8; 7; 6; 5; 4; 3; 2; 1; 0														
SD716	31; 30; 29; 28; 27; 26; 25; 24; 23; 22; 21; 20; 19; 18; 17; 16																	
SD717	47; 46; 45; 44; 43; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33; 32																	
SD716																		
SD717																		
SD718	Сумматор		Эти регистры эмулируют сумматоры контроллеров MELSEC серии "А".	С/П	новый													
SD719																		
SD720	Присвоение номера программы для команды PLOAD		Этот регистр сохраняет номер программы, который должен быть присвоен программе, загружаемой с помощью команды PLOAD. Возможны номера программ от 1 до 124.	П	новый	Q												
SD730	Количество свободных запросов коммуникации CC-Link в области регистрации	0 ... 32	Сохраняет количество свободных блоков в области коммуникационного запроса CC-Link для удаленных специальных модулей, соединенных с A(1S)J61QBT61.	С (во время выполнения)	новый	QnA												
SD736	Ввод с помощью PKEY		Этот регистр диагностики временно сохраняет данные, введенные с клавиатуры с помощью команды PKEY.	С (во время выполнения)	новый	● кроме Q00, Q00 и Q01												

# Указатель

## A

Auto Connect . . . . . 4 - 20

## C

Cross Reference (функция в меню Extras) . . . . . 4 - 49

## D

Debug (меню)

    диагностика контроллера . . . . . 4 - 52

Device Edit (функция в меню "Debug") . . . . . 9 - 1

DUT

    см. структурированный тип данных

## E

Entry Data Monitor . . . . . 7 - 1

Ethernet

    модуль . . . . . 2 - 41

    настройки . . . . . 18 - 3

Extras (меню)

    опции (индикация мониторинга) . . . . . 4 - 48

    опции (перекрестные ссылки) . . . . . 4 - 49

    опции (режим мониторинга) . . . . . 4 - 47

## G

GVL

    см. "Перечень глобальных переменных"

## I

IEC61131-3 . . . . . 3 - 1

## M

Monitor headers (функция в режиме мониторинга) . . . 7 - 7

## N

Network No. (параметр Ethernet) . . . . . 18 - 4

## O

Online Program Change (функция в меню "Project") . 10 - 4

Online (меню)

    запуск контроля . . . . . 7 - 8

    запуск мониторинга . . . . . 4 - 46

    мониторинг заголовков . . . . . 7 - 7

    настройки передачи . . . . . 4 - 39

    форматирование памяти . . . . . 4 - 42

Open settings (Ethernet) . . . . . 18 - 6

Operational sttings (Ethernet) . . . . . 18 - 4

## P

PLC Diagnostics (функция в меню "Debug") . . . 4 - 52

PLCopen . . . . . 3 - 1

POU

    заголовок . . . . . 4 - 13

    определение . . . . . 3 - 2

    присвоение задачи . . . . . 4 - 31

    тело . . . . . 3 - 5

POU-Pool . . . . . 3 - 4

Project (меню)

    Change Security Level . . . . . 14 - 2

    перечень перекрестных ссылок . . . . . 4 - 49

    присвоение пароля . . . . . 14 - 1

## Q

Q64TCRT . . . . . 2 - 39

Q64TCRTBW . . . . . 2 - 39

Q64TCTT . . . . . 2 - 39

Q64TCTTBW . . . . . 2 - 39

QD51 . . . . . 2 - 41

QD62 . . . . . 2 - 39

QD75 . . . . . 2 - 40

QJ71BR11 . . . . . 2 - 42

QJ71C24 . . . . . 2 - 40

QJ71DN91 . . . . . 2 - 43

QJ71E71 . . . . . 2 - 41

QJ71LP21 . . . . . 2 - 42

QJ71PB92D . . . . . 2 - 43

QJ71PB93D . . . . . 2 - 43

QJ71WS96 . . . . . 2 - 44

## R

Result type

    функция . . . . . 6 - 12

## S

SCADA . . . . . 2 - 2

SDT . . . . . 3 - 17

Sink

    выход . . . . . 2 - 36

    вход . . . . . 2 - 23

Source	
выход	2 - 34
вход	2 - 23
Start Monitoring (функция в меню "Online")	7 - 8
System Image (в диалоге "Transfer Setup")	4 - 41

## А

Аналоговые выходные модули	2 - 38
Аналоговые входные модули	2 - 38

## Б

Бесконтактные выключатели	2 - 24
---------------------------	--------

## В

Веб-серверный модуль	2 - 44
Выход ENO	6 - 24
Выходные модули	
обзор	2 - 30
релейные	2 - 31
симисторные	2 - 32
транзисторные (отрицательная логика)	2 - 36
транзисторные (положительная логика)	2 - 34
транзисторные	2 - 34
Вход EN	6 - 24
Входные модули	
Sink	2 - 27
Source	2 - 25
для датчиков с отрицательной логикой	2 - 27
для датчиков с положительной логикой	2 - 25
для переменных напряжений	2 - 28

## Г

Глобальные переменные	
декларирование	4 - 9
определение	3 - 6
перечень	3 - 6
проверка	4 - 12

## Д

Документация	
Print options (опции печати)	4 - 53
комментарии схемы (звена)	4 - 34
метка схемы (звена)	4 - 34

## З

Задача	
определение	3 - 3
признаки	4 - 32
присвоение POU	4 - 31
пул	3 - 7
свойства	4 - 32
создание	4 - 30
Задачный пул	
определение	3 - 7

## И

Интерфейс "человек-машина"	2 - 2
----------------------------	-------

## К

Карты памяти	2 - 19
Комментарий	
для схем (звеньев) программы	4 - 34
копирование	4 - 35
стирание	4 - 35
Комментарий звена (схемы)	4 - 34
Компиляция	4 - 36
Контакты	
сравнение с программируемым контроллером	2 - 1

## Л

Локальные переменные	
определение	3 - 6
переопределение	4 - 19
перечень	3 - 6

## М

Маркеры диагностики	
диагностика ошибок	A - 2
системная информация	A - 3
системные такты	A - 5
таблица сравнения	A - 7
Массив	
обзор	3 - 15
программирование	12 - 1
Метка звена (схемы)	4 - 34
Метод отображения	2 - 46

Модули CC-Link	2 - 42
Модули MELSECNET	2 - 42
Модули PROFIBUS	2 - 43
Модули высокоскоростных счетчиков	2 - 39
Модули позиционирования	2 - 40
Модули центральных процессоров	
батарея	2 - 19
выключатель RUN/STOP	2 - 17
карты памяти	2 - 19
количество операндов	2 - 14
светодиоды	2 - 15
системные выключатели	2 - 17
технические данные	2 - 13
Модуль DeviceNet	2 - 43
Монтажная шина	2 - 6

## Н

Направляющий редактор	4 - 37
Настройки передачи	
для передачи программы	4 - 39

## О

Операнды	
обозначения	2 - 48
Отработка макроккода	6 - 24

## П

Параметры сетевой коммуникации	18 - 2
Параметры сети	18 - 2
Передача программы	4 - 39
Переменные	
выбор из заголовка POU	4 - 16
глобальные (определение)	3 - 6
см. также "Глобальные переменные"	
локальные (определение)	3 - 6
см. также "Локальные переменные"	
присвоение команде	4 - 19
Перечень глобальных переменных	
информация о переменных	4 - 10
расширение	4 - 27
Перечень перекрестных ссылок	4 - 49
Порты (настройки передачи)	4 - 39
Проверка связи	4 - 40

Программа	
наблюдение за выполнением	4 - 45
передача в контроллер	4 - 43
перечень перекрестных ссылок	4 - 49
проверка	4 - 24
Программируемый контроллер	
конфигурация системы	2 - 4
сравнение с контакторным управлением	2 - 1
функции диагностики	4 - 52

## Р

Регистры диагностики	
встроенные часы центрального процессора	A - 25
диагностика ошибок	A - 14
информация цикла программы	A - 31
карты памяти	A - 33
относящиеся к команде	A - 36
регистры файлов	A - 35
системные такты	A - 30
Режим дисплея	9 - 3
Режим мониторинга	4 - 45
Режим связи	4 - 20
Реле	
выходные модули	2 - 31
история развития	2 - 1
Релейно-контактная схема	
вызов функционального блока	4 - 18
направляющий редактор	4 - 37
обзор	3 - 13
программирование	4 - 14

## С

Свойства	
задачи	4 - 32
Сетевые блоки	
критерии выбора	2 - 11
технические данные	2 - 10
Системная метка	3 - 10
Системные переменные	3 - 9
Список инструкций	3 - 11
Структурированный текст	
обзор	3 - 12
программирование	17 - 1

Структурированный тип данных  
 обзор ······ 3 - 17  
 пример применения ······ 11 - 1

Счетчики  
 адреса операндов ······ 3 - 20  
 программирование ······ 4 - 26

**Т**

Таймеры  
 адреса операндов ······ 3 - 20  
 программирование ······ 4 - 28  
 Типы данных ······ 3 - 15

**У**

Управляющая команда ······ 3 - 11

**Ф**

Фоторелейные барьеры ······ 2 - 24  
 Функциональный блок  
 виды выполнения ······ 6 - 23  
 вызов в программе ······ 4 - 26  
 мониторинг экземпляров ······ 7 - 12  
 присвоение имени экземпляра ······ 6 - 18  
 присвоение переменной ······ 6 - 20  
 программирование ······ 6 - 15  
 сравнение с функцией ······ 6 - 1  
 экземпляры ······ 4 - 26

Функция  
 Result type (тип результата) ······ 6 - 12  
 копирование ······ 6 - 11  
 программирование ······ 6 - 2  
 создание ······ 6 - 2  
 сравнение с функциональным блоком ······ 6 - 1  
 тип результата ······ 6 - 12

**Ч**

Число с плавающей запятой  
 как тип результата функции ······ 6 - 12  
 обзор ······ 3 - 15

**Э**

Экземпляр ······ 6 - 18

**Я**

Язык SFC  
 завершающий шаг ······ 15 - 2  
 инициализирующий шаг ······ 15 - 2  
 обзор ······ 3 - 14  
 переход ······ 15 - 2  
 Язык функциональных блоков ······ 3 - 13



**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](mailto:automation@mer.mee.com) /// [www.mitsubishi-automation.ru](http://www.mitsubishi-automation.ru)**