

ПЛК

Прикладное программирование (релейно-контактная схема/серия MELSEC iQ-R)

Этот курс предназначен для пользователей, понимающих основы программирования ПЛК серии MELSEC iQ-R и желающих освоить следующий уровень программирования.

L(CTS)00692RUS

Курс рассчитан на тех участников, которые завершили курс «Основы программирования» (релейно-контактная схема) или обладают соответствующими знаниями. В ходе этого курса будут даны знания по эффективному программированию и отладке программируемых контроллеров серии MELSEC iQ-R, использованию памяти ПЛК и переменных

Предварительным условием для изучения данного курса является прохождение перечисленных ниже курсов либо владение соответствующими знаниями.

- Основные сведения о ПЛК серии MELSEC iQ-R
- Основы программирования

Данный курс имеет следующее содержание.

Глава 1. Эффективное программирование

Методы и настройки для эффективного программирования

Глава 2. Расширенное программирование





Расширенное использование операндов ПЛК и переменных

Глава 3. Эффективная отладка

Функции программного обеспечения, используемые для эффективной отладки проектов ПЛК

Заключительный тест

Проходной балл: не менее 60%

Переход к следующей странице		Переход к следующей странице.
Возврат к предыдущей странице		Возврат к предыдущей странице.
Переход к требуемой странице		Появится экран «Содержание», на котором вы сможете перейти к требуемой странице.
Завершение обучения		Завершение обучения.

Меры безопасности

Если вы обучаетесь с использованием реальных изделий, внимательно изучите правила техники безопасности, приведенные в соответствующих руководствах.

Меры предосторожности относительно данного курса

Отображаемые экраны зависят от версии ПО и могут отличаться от представленных в данном курсе. В данном курсе используется следующая версия программного обеспечения:

- GX Works3 версии 1.044W

В этой главе описываются настройки программного обеспечения для эффективного программирования и основы программирования переменных.

- 1.1 Упрощение использования программ в различных системах
- 1.2 Настройка памяти в соответствии с использованием операндов
- 1.3 Использование имен переменных, связанных с приложениями
- 1.4 Улучшение читаемости программы

1.1 Упрощение использования программ в различных системах

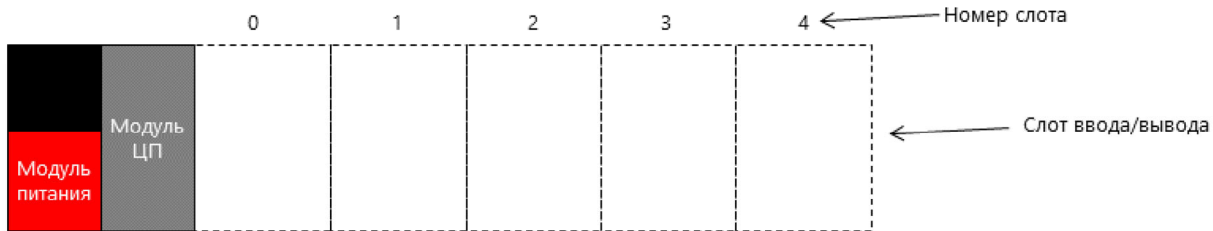
В этом разделе описано эффективное назначение номеров ввода/вывода для упрощения использования программ в различных системах.

1.1.1 Автоматическое назначение номера ввода/вывода

Модулям, установленным на базовом шасси, номера ввода/вывода назначаются последовательно, начиная с ближайшего к модулю центрального процессора (ЦП) слота.

Номера ввода/вывода назначаются в виде блоков с 16 точками (от 0 до F).

Доступное количество назначаемых (занимаемых) точек зависит от типа модуля (16, 32, 64 и т. д.).



В рассматриваемом далее примере установлено пять модулей с 16 точками.



1.1.1

Автоматическое назначение номера ввода/вывода

При одновременном использовании модулей с 16, 32 и 64 занятыми точками номера ввода/вывода назначаются следующим образом:

		0	1	2	3	4
Модуль питания	Модуль ЦП	16 точек	32 точки	64 точки	32 точки	16 точек
		от 00 до 0F	от 10 до 2F	от 30 до 6F	от 70 до 8F	от 90 до 9F

Если между установленными модулями имеется пустой слот, номера ввода/вывода также назначаются пустому слоту.

		0	1	2	3	4
Модуль питания	Модуль ЦП	16 точек	32 точки	64 точки	16 точек	16 точек
		от 00 до 0F	от 10 до 2F	от 30 до 6F	от 70 до 7F	от 80 до 8F

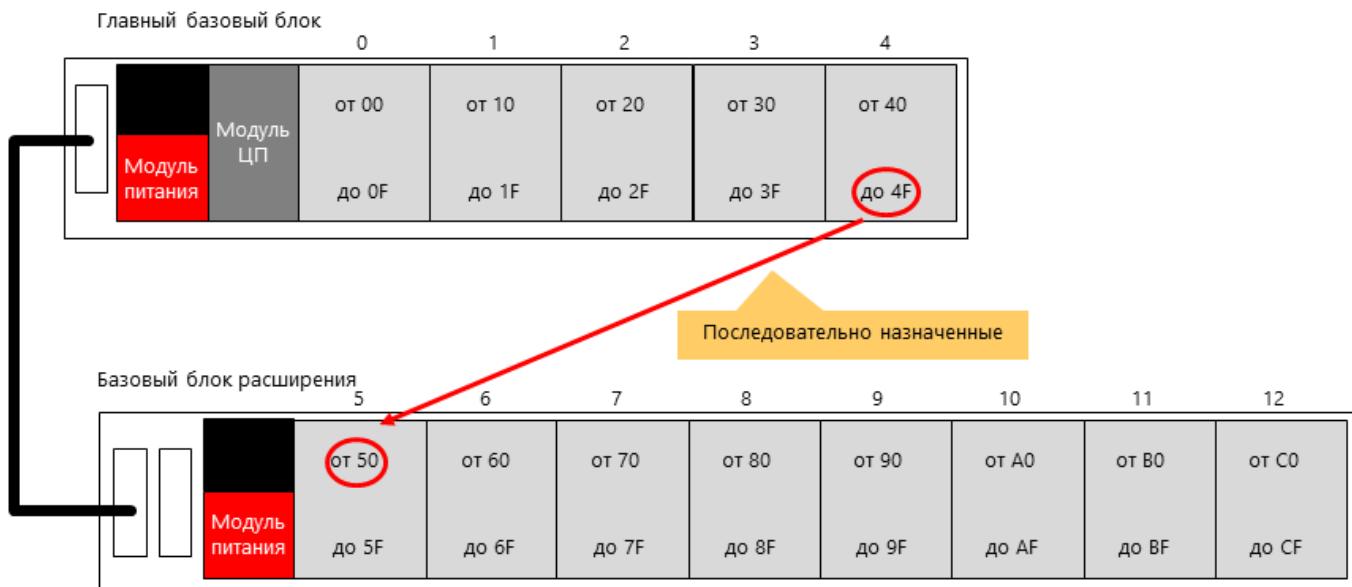
Пустой слот

По умолчанию пустому слоту назначается 16 точек. Количество назначенных точек можно изменить с помощью установки параметра в диапазоне от 0 до 1024 точек (в блоке с 16 точками).

1.1.1

Автоматическое назначение номера ввода/вывода

Номера ввода/вывода базового блока расширения назначаются автоматически после последнего номера ввода/вывода главного базового блока.



1.1.2

Фиксированное назначение номера ввода/вывода

В случае назначения номеров ввода/вывода вручную назначенные номера фиксируются и остаются неизменными даже при изменении конфигурации модуля. Это означает, что одна и та же управляющая программа может использоваться для управления одним и тем же процессом независимо от конфигурации модуля.

Автоматическое назначение

Перед добавлением модулей

Модуль питания	Модуль ЦП	Модуль входов 64 точки	Модуль выходов 64 точки	Специальный функциональный модуль 16 точек	
		от X00 до X3F	от Y40 до Y7F	от X/Y80 до X/Y8F	

После добавления модулей
(Добавлен модуль входов с 32 точками и модуль выходов с 16 точками.)

Добавленные модули

Модуль питания	Модуль ЦП	Модуль входов 64 точки	Модуль входов 32 точки	Модуль выходов 64 точки	Модуль выходов 16 точек	Специальный функциональный модуль 16 точек
		от X00 до X3F	от X40 до X5F	от Y60 до Y9F	от YA0 до YAF	от X/YB0 до X/YBF

После добавления модулей номера ввода/вывода переназначаются.

1.1.2

Фиксированное назначение номера ввода/вывода

Назначение фиксированных номеров ввода/вывода вручную

Перед добавлением модулей

	Модуль ЦП	Модуль входов 64 точки	Модуль выходов 64 точки	Специальный функциональный модуль 16 точек	
Модуль питания		от X00 до X3F	от Y40 до Y7F	от X/Y80 до X/Y8F	

После добавления модулей

(Добавлен модуль входов с 32 точками и модуль выходов с 16 точками.)

Добавленные модули

	Модуль ЦП	Модуль входов 64 точки	Модуль входов 32 точки	Модуль выходов 64 точки	Модуль выходов 16 точек	Специальный функциональный модуль 16 точек
Модуль питания		от X00 до X3F	от X90 до XAF	от Y40 до Y7F	от YB0 до YBF	от X/Y80 до X/Y8F

Одни и те же номера ввода/вывода назначаются независимо от добавления модулей.

Поскольку номера ввода/вывода существующих модулей остаются неизменными, необходимо добавить или изменить только связанные с добавленными модулями программы.

1.1.3

Автоматическое назначение номера ввода/вывода с помощью схемы настройки конфигурации модуля

Конфигурацию модуля можно настроить с помощью схемы настройки конфигурации модуля программного обеспечения MELSOFT GX Works3.

Выберите название модели модуля и перетащите его на слот, чтобы поместить туда модуль.

Номера ввода/вывода последовательно назначаются начиная с модуля, ближайшего к модулю ЦП, как показано на схеме.

Выберите размещенный модуль для просмотра начального номера ввода/вывода модуля.

Перетащите название модели модуля.

Module Model	Points	Type
RY40PT5B(S2S)	16 points	(Source type)
RY40PT5P	16 points	(Source type)
RY41NT2H	32 points	(Sink type/High-Speed)
RY41NT2P	32 points	(Sink type)
RY41PT1P	32 points	(Source type)
RY41PT2H	32 points	(Source type/High-Speed)
RY42NT2P	64 points	(Sink type)
RY42PT1P	64 points	(Source type)

Input the Configuration Detailed Information

Field	Value
Start XY	0040
Points	64 Points

Начальный номер ввода/вывода выбранного модуля

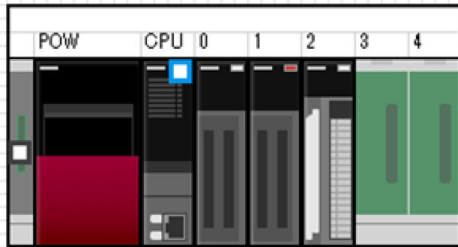
1.1.4

Назначение номера ввода/вывода с помощью схемы настройки конфигурации модуля

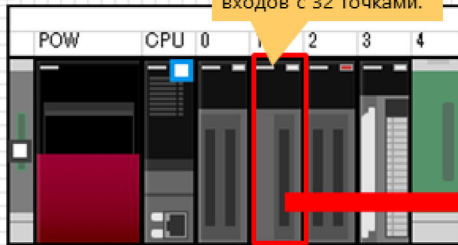
На следующем примере описывается процедура назначения номера вводов/выводов вручную с помощью схемы настройки конфигурации модуля в GX Works3.

При изменении конфигурации модуля путем добавления нового модуля в схему настройки конфигурации номер ввода/вывода добавленного модуля накладывается на существующие номера, если место расположения существующих модулей изменилось. Отредактируйте номера ввода/вывода таким образом, чтобы они не накладывались друг на друга, и определите конфигурацию модуля. Если определить конфигурацию модуля с наложенными друг на друга номерами ввода/вывода, появится сообщение об ошибке. На этом этапе можно выполнить автоматическое назначение с помощью окна отображаемого сообщения об ошибке.

Перед добавлением модулей



После добавления модулей

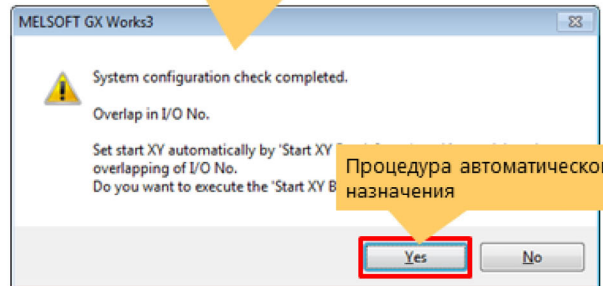


Добавлен модуль входов с 32 точками.

Номера от X00 до X1F накладываются друг на друга.

От	От	От	От
X00	X00	X40	X80
До	До	До	До
X3F	X1F	X7F	X8F

Если определить конфигурацию модуля с наложенными друг на друга номерами ввода/вывода, появится сообщение об ошибке.



Процедура автоматического назначения

Измените начальный номер ввода/вывода модуля входов с 32 точками с 0000 на 0090.

Input the Configuration Detailed Information

RX41C4	
Start XY	0090
Points	32 Points

Определение конфигурации модуля

1.2

Настройка памяти в соответствии с использованием операндов

1.2.1

Настройки области памяти операндов/переменных

Количество используемых для модулей ЦП операндов зависит от типа модуля ЦП. Начальное количество операндов назначается в зависимости от емкости памяти области операндов в модуле ЦП.

Емкость области операндов в R04CPU составляет 40 тысяч слов.

При уменьшении неиспользуемых областей увеличивается количество операндов по сравнению с начальным значением.

На следующем рисунке показано окно «Device/Label Memory Area Setting» (Настройка области памяти операндов/переменных).

Емкость области операндов увеличивается при уменьшении вместимости области переменных или области хранения файлов.

Кроме того, емкость всей области памяти операндов/переменных можно увеличить с помощью расширенной кассеты SRAM.

Item	Setting
Device/Label Memory Area Setting	
Extended SRAM Cassette Setting	Not Mounted
Device/Label Memory Area Capacity Setting	
Device Area	
Device Area Capacity	40 K Word
Label Area	
Label Area Capacity	30 K Word
Latch Label Area Capacity	2 K Word
File Storage Area Capacity	128 K Word
Device/Label Memory Configuration Confirmation	<Confirmation>
Device/Label Memory Area Detailed Setting	
Device Setting	<Detailed Setting>
Latch Type Setting of Latch Type Label	Latch (1)

Емкость области операндов

1.2.2

Настройки операндов

Количество операндов каждого типа можно изменить в окне «Device Setting» (Настройка операндов). Начальное значение некоторых операндов равно 0 точек. Назначьте количество точек при использовании таких операндов.

Item	Symbol	Points	Device	Latch (1)	Latch (2)
Input	X	12K			
Output	Y	12K	0 to 211		
Internal Relay	M	12K	0 to 1228		
Link Relay	B	8K	0 to 1FFF		
Link Special Relay	SB	2K	0 to 7FF		
Annunciator	F	2K	0 to 2047	No Setting	No Setting
Edge Relay	V	2K	0 to 2047		
Step Relay	S	0			
Timer	T	1K	0 to 1023		
Long Timer	LT	1K	0 to 1023		
Retentive Timer	ST	0			
		0			
		512	0 to 511		
		512	0 to 511		
		18K	0 to 18431		
		8K	0 to 1FFF		
		2K	0 to 7FF		
		8K	0 to 8191		
				No Setting	
Total Device			38.4K Word		0.0K Word
Total Word Device			34.5K Word		
Total Bit Device			62.0K Bit		

Общее количество операндов
Количество операндов автоматически пересчитывается в единицы слова.

Количество операндов
Задайте количество точек, используемых каждым операндом.

- Начальные значения предварительно определены
- Значения в белых ячейках могут изменяться
- Установите количество операндов в блоках с 16 точками.
- Количество точек 1K означает 1024 точки

Если общее количество операндов превышает емкость модуля ЦП, появляется сообщение, указывающее на необходимость изменения настроек.

Максимальное количество операндов = емкость модуля ЦП
Например, емкость модуля ЦП в R04CPU составляет 40 тысяч слов.

Окно «Device Setting» (Настройка операндов)

MELSOFT GX Works...
It will exceed the (standard) device area capacity. Please set it so that the total number of device points will not exceed the (standard) device area capacity.
OK

1.3

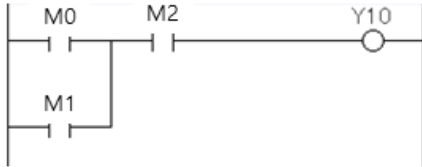
Использование имен переменных, связанных с приложениями

1.3.1

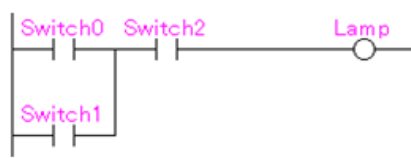
Преимущества использования переменных

Названия операндов, используемых в управляющих программах, должны состоять из букв и цифр, например «M0» и «D5». Если имя переменной связано с приложением, например «StartSwitch», цель обработки становится очевидной. Имена переменных можно устанавливать произвольно в соответствии с приложениями. Пользователям не нужно учитывать адреса операндов для областей, где используются операнды.

Программирование с использованием названий операндов



Программирование с использованием переменных



В зависимости от области применения переменные подразделяются на два указанных далее типа.

- **Глобальные переменные**

Глобальные переменные можно использовать для всех программ в проекте.

- **Локальные переменные**

Локальные переменные можно использовать только в программе, в которой они зарегистрированы.

При использовании переменных для фактических устройств (X, Y) названия операндов должны назначаться глобальным переменным с помощью GX Works3.

1.3.2

Тип данных переменных

Тип данных следует задавать для каждой переменной для определения диапазона обрабатываемых значений.

Типы данных включают бит и целое число, как показано ниже.

Тип данных		Диапазон данных
Тип бита		Включенное/выключенное состояние битовых операндов и истинное/ложное состояние результатов выполнения
Целочисленный тип	Слово (без знака)	от 0 до 65 535
	Слово (со знаком)	от -32 768 до 32 767
	Двойное слово (без знака)	от 0 до 4 294 967 295
	Двойное слово (со знаком)	от -2 147 483 648 до 2 147 483 647

При использовании целочисленного типа выберите тип слова или двойного слова в соответствии с диапазоном данных, а также выберите тип со знаком или без знака, если необходимо обрабатывать отрицательные значения. Задайте тип данных переменной при установке имени переменной с помощью GX Works3.

Label Name	Data Type
Switch0	Bit
Data0	Word [Unsigned]/Bit String [16-bit]
Data1	Double Word [Signed]

Задайте диапазон значений переменной.

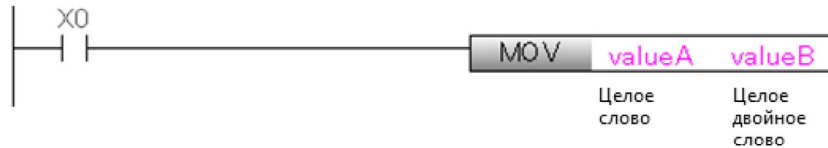
Окно установки переменной

1.3.3

Имена переменных, которые представляют типы данных

Использование на источнике передачи и в месте назначения разных типов данных может привести к ошибке преобразования или неожиданному результату.

Ниже приведен пример программы, иллюстрирующий такой случай.



Значения двойного слова нельзя перенести в переменную словесного типа. Тем не менее тип данных нельзя идентифицировать по имени переменной.

Таким образом, префиксы, представляющие тип данных, можно добавлять к именам переменных, что даст возможность визуально идентифицировать типы данных.

Такой тип именования переменных известен как венгерская нотация.

Тип данных		Диапазон данных	Префикс	Расширение префикса
Тип бита		Включенное/выключенное состояние битовых операндов и истинное/ложное состояние результатов выполнения	b	bit (бит)
Целочисленный тип	Слово (без знака)	от 0 до 65 535	u	unsigned word (слово без знака)
	Слово (со знаком)	от -32 768 до 32 767	w	signed word (слово со знаком)
	Двойное слово (без знака)	от 0 до 4 294 967 295	ud	unsigned double-word (двойное слово без знака)
	Двойное слово (со знаком)	от -2 147 483 648 до 2 147 483 647	d	signed double-word (двойное слово со знаком)

Пример программы в верхней части этой страницы можно записать с помощью венгерской нотации указанным далее образом.



С помощью венгерской нотации в процессе написания программы могут быть выявлены несоответствия между типами данных. В остальной части курса имена переменных в примерах написаны в венгерской нотации.

1.3.4

Использование подготовленных переменных

Когда конфигурация модуля установлена в схеме конфигурации модуля, переменные (переменные модулей), представляющие сигналы модуля или значения установки, которые соответствуют положениям установки модуля, регистрируются автоматически.

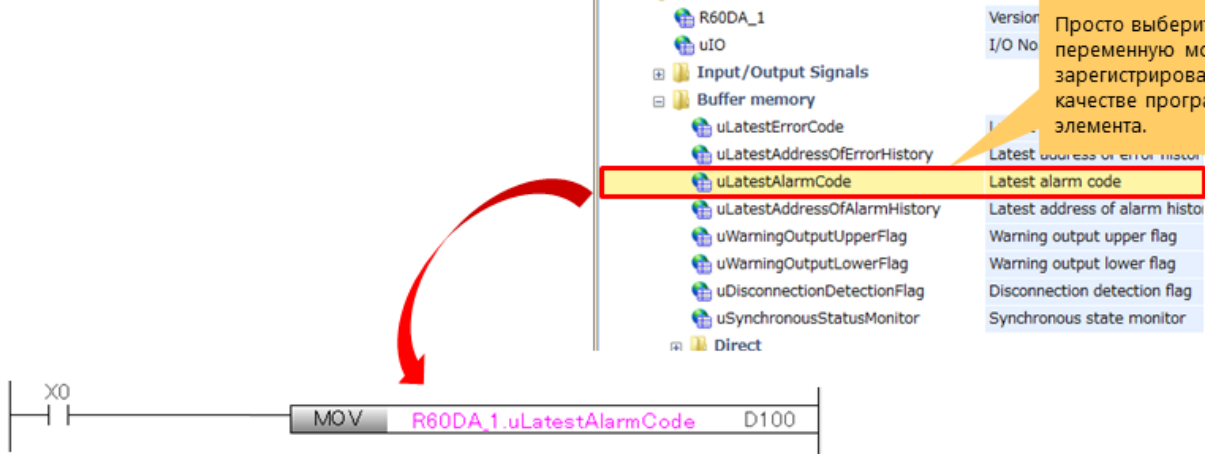
Для программирования с использованием названий операндов адреса операндов и адреса буферной памяти, соответствующие сигналам, следует проверять в руководствах, что требует времени. Время программирования можно сократить путем использования переменных модулей, поскольку пользователям нужно только выбрать переменные из перечня.

При использовании названий операндов



Проверьте и опишите положение установки и адрес буферной памяти.

При использовании переменных модулей



1.3.5

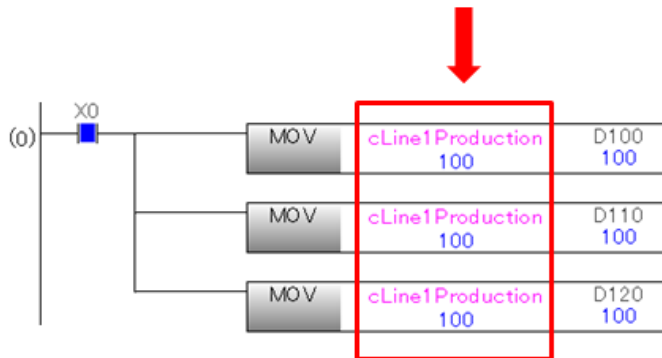
Назначение констант переменным

Переменным можно назначать константы.

При назначении констант переменным значения можно изменять без изменения программы.

Одну и ту же константу, используемую для нескольких переменных, можно изменять совместно.

Назначьте константу 100 метке «cLine1Production».



Назначьте константу 200.



Чтобы назначить константу переменной, измените класс, который определяет задание переменной, в окне для установки переменных.

Для локальных переменных выберите «VAR_CONSTANT».

Label Name	Data Type	Class	Initial Value	Constant
uData	Word [Unsigned]/Bit String [16-bit]	VAR_CONSTANT		100

Задайте область применения переменной.

1.4

Улучшение читаемости программы

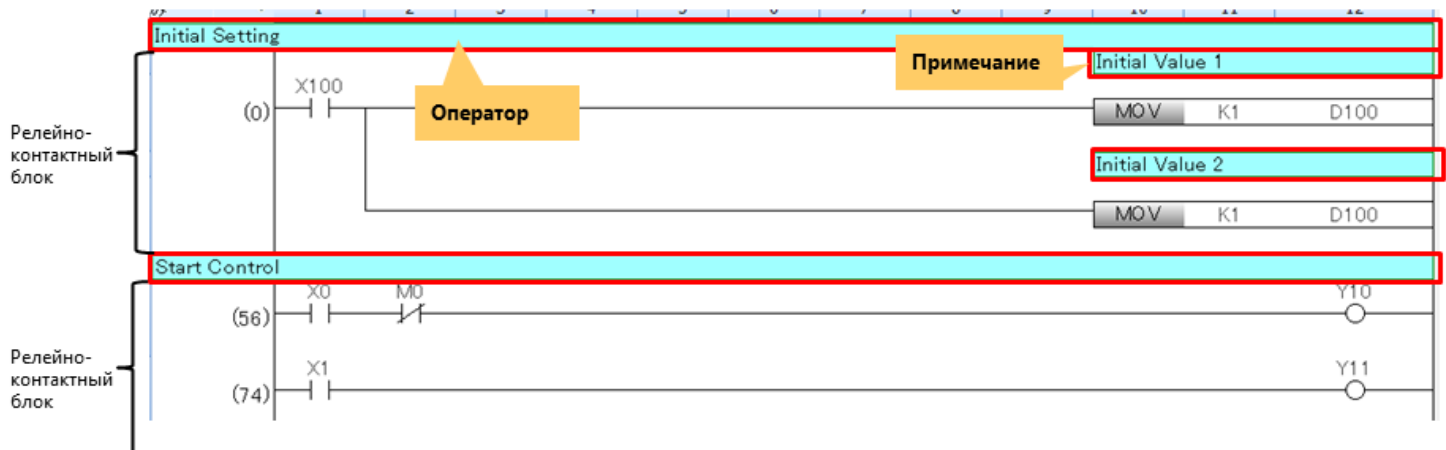
В программу можно добавлять комментарии, поясняющие программу и описывающие названия операндов. Комментарии помогают прояснить принципы работы программы.



Тип комментария можно выбирать в зависимости от элемента или диапазона программы.

В приведенном выше примере программы «**device/label comments**» (комментарии к операнду/переменной) были добавлены для прояснения применения операндов и переменных и типов операндов ввода/вывода.

Кроме того, можно использовать также комментарии типа «**statement**» (пояснение), который добавляется к релейно-контактному блоку, чтобы пояснить какую-либо часть программы, и «**notes**» (примечания), которые описывают выполняемые инструкции или выходные реле.



Из этой главы вы узнали о:

- назначении номера ввода/вывода;
- настройке области памяти;
- программировании с помощью переменных;
- комментариях в программах.

Важные моменты

Назначение номера ввода/вывода	<ul style="list-style-type: none">• Номера ввода/вывода автоматически назначаются слотам, начиная с ближайшего к модулю ЦП• Программы могут использоваться в различных системах, когда фиксированные номера ввода/вывода назначаются модулям вручную
Настройки устройства и области памяти	<ul style="list-style-type: none">• Количество операндов зависит от модуля ЦП• Количество операндов можно увеличить, уменьшив неиспользуемые области памяти• Количество операндов для каждого устройства можно изменить в соответствии с состоянием его использования
Программирование с помощью переменных	Цель обработки становится очевидной при использовании переменных
Комментарии	При добавлении комментариев программа становится легче для понимания.

В этой главе описывается расширенное использование устройств и программирование переменных.

- 2.1 Использование словного операнда в битах
- 2.2 Включение операнда только при изменении состояния контакта
- 2.3 Удержание таймером значения измеренного времени
- 2.4 Изменение единицы измерения таймера
- 2.5 Работа с множеством операндов (индексный регистр)
- 2.6 Работа с множественными значениями (массив)
- 2.7 Работа с множественными значениями (структура)
- 2.8 Удержание состояния операнда (фиксация)
- 2.9 Удержание состояния операнда (файловый регистр)
- 2.10 Использование с операндов с предустановленными функциями и операциями
- 2.11 Вычисления с действительными числами

2.1

Использование словного операнда в битах

Словные операнды, такие как регистр данных, обычно используются в количестве слов, но их также можно использовать и в битах.

Единицы битов используются для задания конкретного бита в регистре данных (D).

Пример. Регистр данных (D)

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

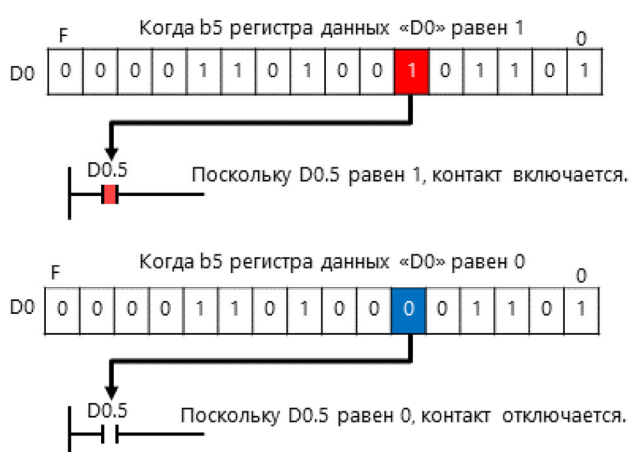
Формат задания бита

D□.□

Символ словного операнда (D, W или R)

Адрес операнда

Бит (от 0 до F)



Для использования переменных создайте описание в виде «uData.2» и «uData.5».

Можно задать сигнал, который включается только для одного скана программы на переднем или заднем фронте сигнала контакта.

Эта функция полезна для контроля переднего или заднего фронта входного сигнала.

Задание переднего фронта для контакта

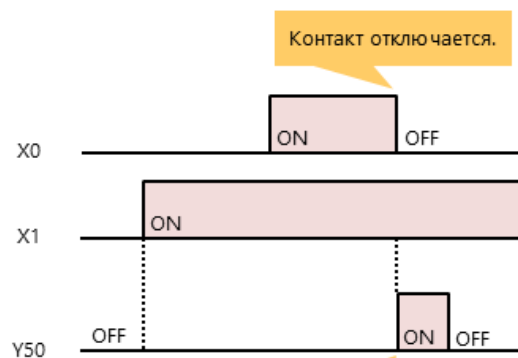
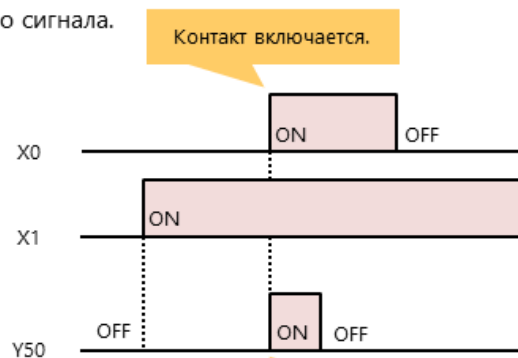


Сигнал включается только в пределах одного скана программы, в котором определено включенное состояние контакта «X0».

Задание заднего фронта для контакта



Сигнал включается только в пределах одного скана программы, в котором определено выключенное состояние контакта «X0».



2.3 Удержание таймером значения измеренного времени

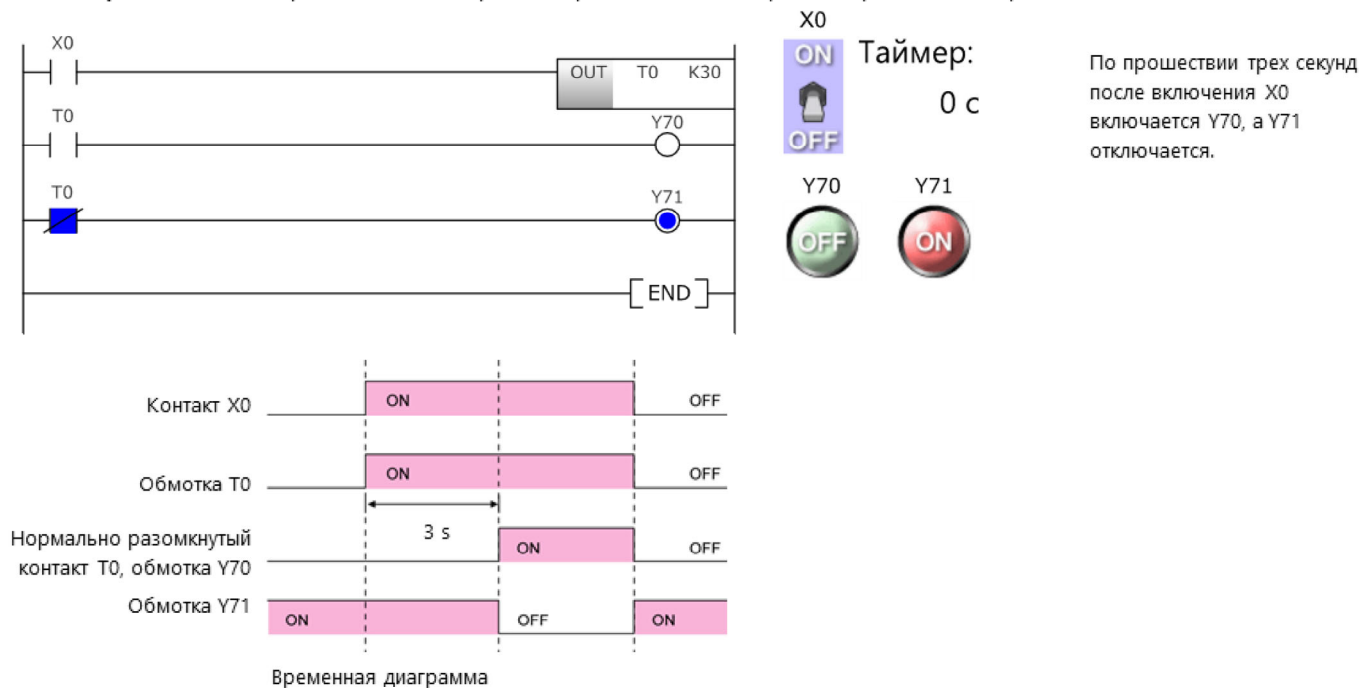
В этом разделе описывается фиксируемый таймер, который может удерживать значение измеренного времени.

2.3.1 Разница между таймером и фиксируемым таймером

Прежде чем описывать фиксируемый таймер, следует рассмотреть принцип работы таймера.

Таймер начинает измерение при включении обмотки. По истечении указанного времени контакт включается. При отключении обмотки значение измеренного времени сбрасывается на «0». Символ устройства таймера: «Т».

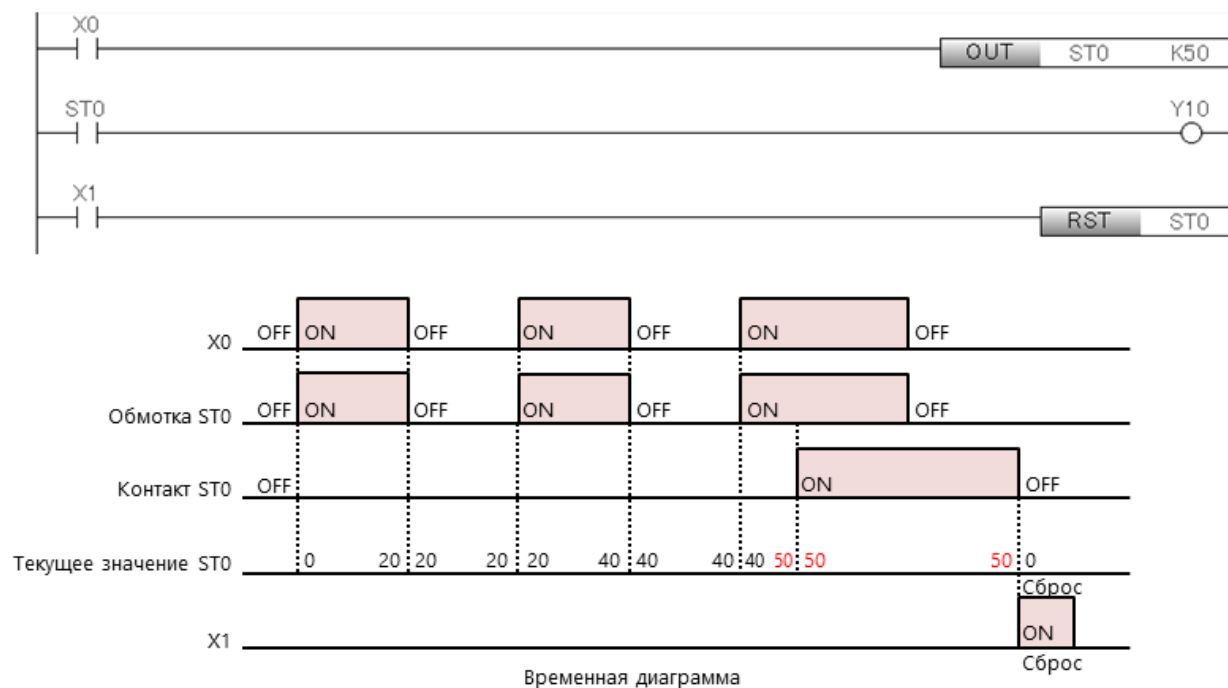
Воспользуйтесь входным переключателем на правой стороне, чтобы посмотреть, как работает таймер.



2.3.1

Разница между таймером и фиксируемым таймером

Фиксируемый таймер полезен для измерения общего времени работы. Фиксируемый таймер начинает измерение при включении обмотки. По истечении указанного времени контакт включается. При отключении обмотки значение измеренного времени не сбрасывается. При повторном включении обмотки измерение начинается заново с удерживаемого значения. Символ операнда фиксируемого таймера: «ST».



2.3.2

Пример программы фиксируемого таймера

Давайте рассмотрим принципы работы фиксируемого таймера, создав имитацию работающей машины с помощью входных переключателей (от X0 до X2).

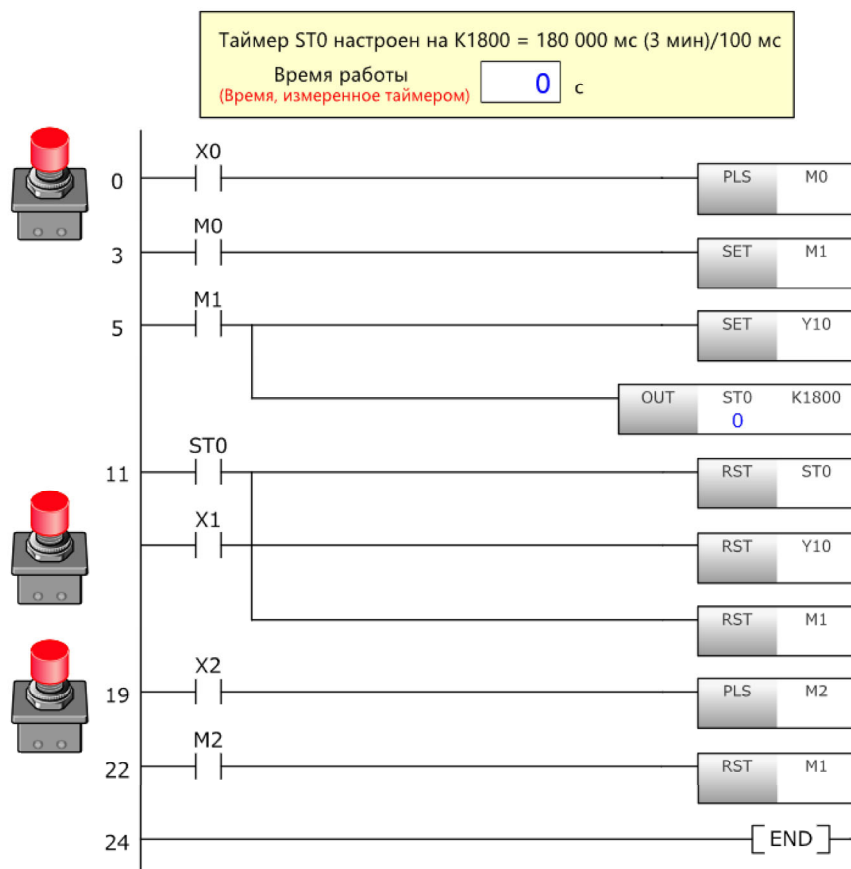
* Фиксируемый таймер (ST0) установлен с шагом 100 мс.

1. Когда включается X0, начинается работа.
2. Когда включается X2, работа приостанавливается и удерживается текущее значение.
3. При повторном включении X0 работа возобновляется.
4. Когда включается X1, работа прекращается и текущее значение сбрасывается.



От X0 до X2: входные переключатели

Y10: сигнал запуска



2.3.3

Настройка фиксируемого таймера

Количество точек, используемых фиксируемого таймером, по умолчанию равно «0».

Перед использованием фиксируемого таймера установите количество точек в «Device Setting» (Настройка операндов) параметра ЦП с помощью GX Works3.

В следующем примере для фиксируемого таймера установлено 64 точки (от ST0 до ST63).

Item	Symbol	Device		Local Device			Latch (1)	Latch (2)
		Points	Range	Start	End	Points		
Input	X	12K	0 to 2FFF					
Output	Y	12K	0 to 2FFF					
Internal Relay	M	12K	0 to 12287				No Setting	No Setting
Link Relay	B	16K	0 to 3FFF				No Setting	No Setting
Link Special Relay	SB	16K	0 to 3FFF					
Annunciator	F	2K	0 to 2047				No Setting	No Setting
Edge Relay	V	2K	0 to 2047				No Setting	No Setting
Step Relay	S	0						
Timer	T	1K	0 to 1023				No Setting	No Setting
Long Timer	LT	1K	0 to 1023				No Setting	No Setting
Retentive Timer	ST	64	0 to 63				No Setting	No Setting
Long Retentive Time	LST	0					No Setting	No Setting
Counter	C	512	0 to 511				No Setting	No Setting
Long Counter	LC	512	0 to 511				No Setting	No Setting
Data Register	D	18K	0 to 18431				No Setting	No Setting
Link Register	W	8K	0 to 1FFF				No Setting	No Setting
Link Special Register SW		2K	0 to 7FF					
Latch Relay	L	8K	0 to 8191					No Setting
Total Device			39.9K Word			0.0K Word		
Total Word Device			34.6K Word			0.0K Word		
Total Bit Device			84.2K Bit			0.0K Bit		

2.3.4

Использование переменной для задания таймера

Установите тип данных «Таймер», когда переменной используется таймер.

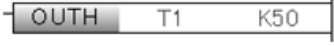

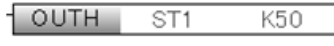


Label Name	Data Type
uTimer1	Timer
uTimer2	Retentive Timer

Для использования фиксируемого таймера необходимо настроить операнд, как описано в предыдущем разделе. Тем не менее, это необязательно для переменных.

Единица измерения и время зависят от типа таймера.

- Высокоскоростной таймер (единицы для кратковременного измерения)
- Низкоскоростной таймер (единицы для долговременного измерения)
- Таймер для долговременного измерения может выполнять длительные измерения

В каждом из вышеперечисленных таймеров имеется функция фиксируемого таймера.

Тип	Единица измерения	Пример программы	Работа	Удержание текущего значения
Низкоскоростной таймер	100 мс (по умолчанию)		Низкоскоростной таймер T0 выполняет измерение в течение 5 секунд.	16 бит
Высокоскоростной таймер	10,00 мс (по умолчанию)		Высокоскоростной таймер T1 выполняет измерение в течение 0,5 секунды.	
Низкоскоростной фиксируемый таймер	100 мс (по умолчанию)		Низкоскоростной фиксируемый таймер ST0 выполняет измерение в течение 5 секунд.	
Высокоскоростной фиксируемый таймер	10,00 мс (по умолчанию)		Высокоскоростной фиксируемый таймер ST1 выполняет измерение в пределах 0,5 секунды.	
Таймер для долговременного измерения	0,001 мс (по умолчанию)		Таймер для долговременного измерения LT0 выполняет измерение в течение 0,005 секунды.	32 бита
Фиксируемый таймер для долговременного измерения			Фиксируемый таймер для долговременного измерения LST0 выполняет измерение в течение 0,005 секунды.	

Начальные единицы измерения: 100 мс для низкоскоростного таймера, 10 мс для высокоскоростного таймера и 0,001 мс для таймера для долговременного измерения.

Порядок изменения единиц измерения см. на следующей странице.

2.4

Изменение единицы измерения в таймере

Единицы измерения таймера можно изменить в пункте «Timer Limit Setting» (Установка ограничения таймера) параметра ЦП.

Timer Limit Setting	
Low Speed Timer/Low Speed Retentive Timer	100 ms
High Speed Timer/High Speed Retentive Timer	10.00 ms
Long Timer/Long Retentive Timer	0.001 ms

2.5

Работа с множеством операндов (индексный регистр)

Индексный регистр (Z) используется совместно с другим устройством для косвенного задания (изменения) адреса операнда устройства объекта управления. Индексный регистр полезен для упрощения программ, поскольку с его помощью можно описать несколько устройств в пакете.

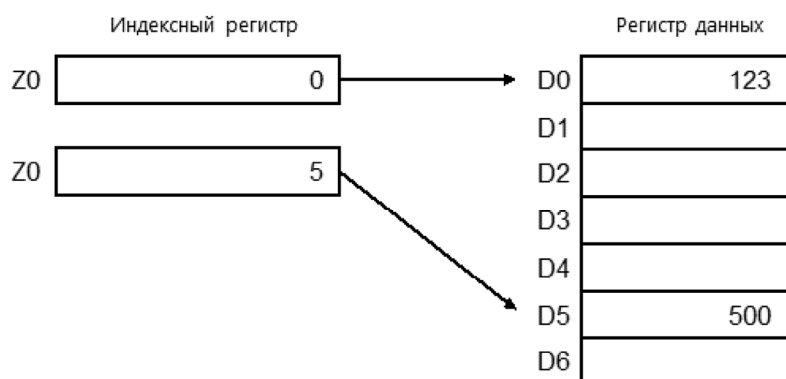
- Индексный регистр описывается после символа устройства и адреса операнда.
- Фактический адрес операнда устройства объекта управления = символ устройства (адрес операнда + индексный регистр)
- Количество операндов для индексного регистра по умолчанию составляет 20 точек (от $Z0$ до $Z19$)

2.5.1

Пример применения индексного регистра

Когда операнд описывается как «D0Z0», это означает $D(0+Z0)$.

Пример. Если $Z0$ равно 0, то адрес операнда равен D0.
Если $Z0$ равно 5, то адрес операнда равен D5.



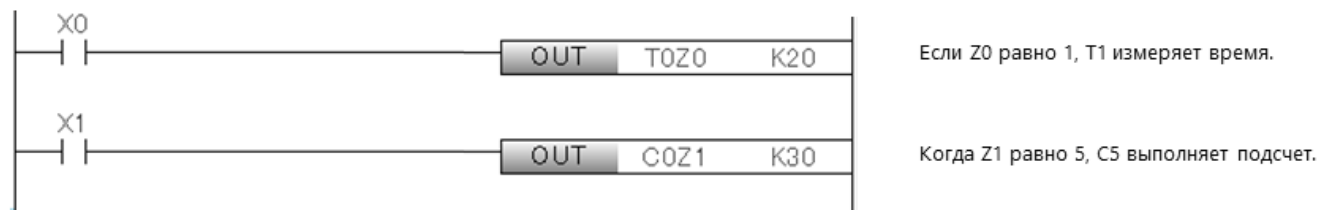
2.5.2

Операнды, которые могут быть изменены индексным регистром

Операнды, которые могут быть изменены индексным регистром включают следующие:

Битовый операнд	X, Y, M, L, S, B, F
Словный операнд	T, C, D, R, W
Константа	K, H
Указатель	P

Примечание. Для контактов и обмоток, используемых в таймерах и счетчиках, можно использовать только индексный регистр «Z0» или «Z1».



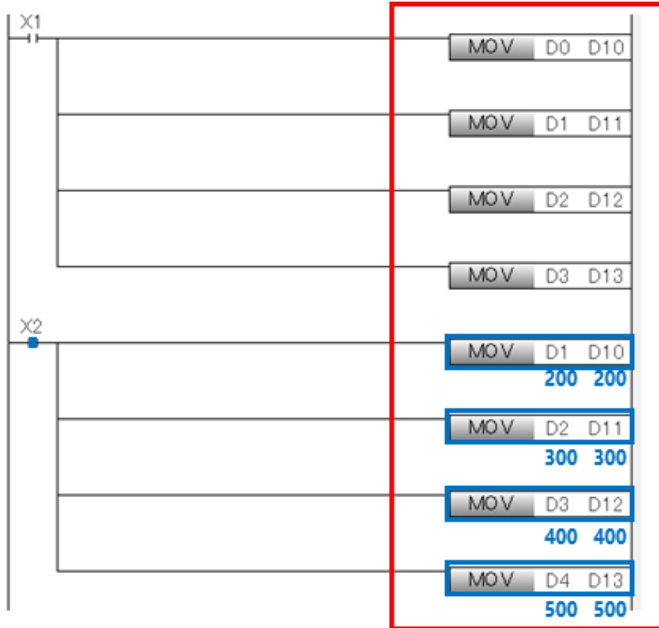
2.5.3

Упрощение программ с помощью индексного регистра

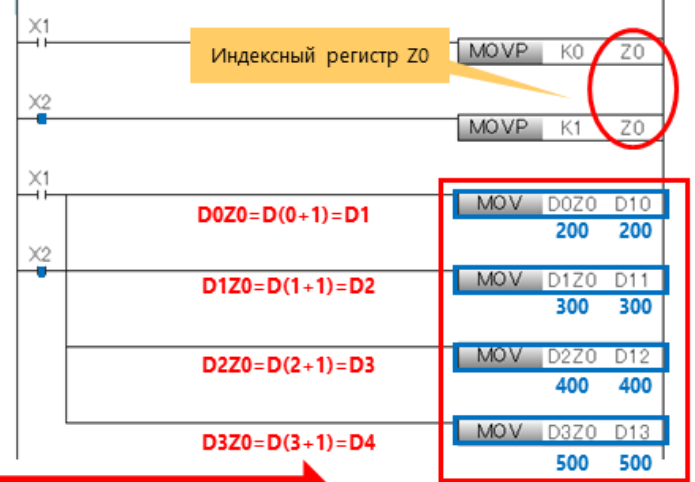
Показанные ниже программы передают значения в формате «от D0 до D4» до «от D10 до D13», когда включается X1 или X2. Программы (1) и (2) обеспечат одинаковый результат. В программе (1) данные передаются напрямую. В программе (2) данные передаются косвенно через регистр индексов.

Начальные сохраненные значения
 D0=100
 D1=200
 D2=300
 D3=400
 D4=500

(1) Если индексный регистр не используется



(2) Если индексный регистр используется



Программа упрощена.

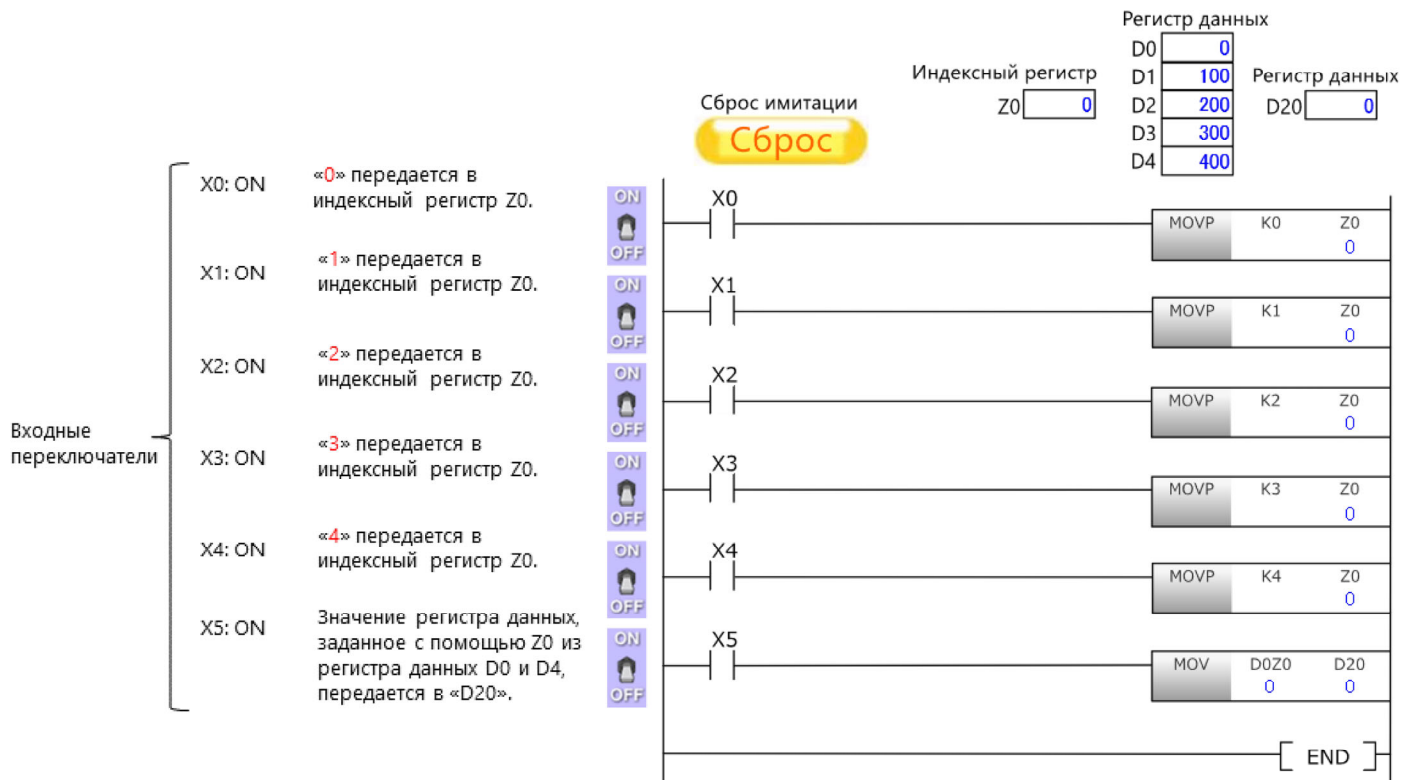
2.5.4

Пример программы индексного регистра




Создать имитацию работы индексного регистра Z0 можно путем включения входных переключателей с X0 по X5.

* С K0 по K400 уже сохранено в регистре данных с D0 по D4.

Включите входные переключатели с X0 по X5, чтобы проверить значения, сохраненные в каждой области операндов.



С помощью массива посредством имени одной переменной можно обработать несколько значений. На следующем примере данные об объемах производства на автомобильном заводе хранятся по месту назначению.

Назначение			
Объем производства	35 единиц	75 единиц	65 единиц

Данные об объеме производства по назначению назначаются переменной.

Если массив не используется, имена переменных следует создать для каждого пункта назначения.

Тем не менее, с помощью массива объем производства для нескольких мест назначения можно назначить и сохранить в одном имени переменной.

Если массив не используется

```
uProductionA
uProductionB
uProductionC
```



Если массив используется

```
uProduction
```

Индивидуальные переменные в массиве задаются с помощью номеров элементов. Номера элементов начинаются с [0].

```
uProduction [0]
```

Имя
переменной

Номер элемента



Назначение
(ряд)

Country A	[0]	35
Country B	[1]	75
Country C	[2]	65



В следующем примере программы планируемый объем производства для Country A переносится в другую переменную.

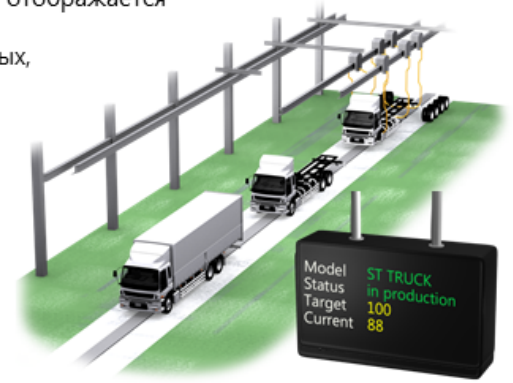
```
MOV    uProduction[0]    uShowProductionPlan
```

При использовании структуры одним именем переменной можно определить несколько связанных между собой переменных.

На следующем примере состояние производственной линии для автомобилей отображается на экране Andon (информационное табло).

В следующей таблице перечислены имена переменных, значения и типы данных, соответствующие отображаемым элементам.

Элемент	Имя переменной	Значение	Тип данных переменной
Модель	sModel	'ST TRUCK'	Строковый тип
Рабочее состояние	bStatus	'in production'	Тип бита
Плановый объем производства на сегодня	uPlanQty	'100' единиц	Целочисленный тип (слово, без знака)
Текущий объем производства	uActualQty	'88' единиц	Целочисленный тип (слово, без знака)



Если структура не используется для завода с несколькими производственными линиями, имя переменной следует изменить для каждой линии.

Ниже приведены примеры имен переменных с добавленными названиями производственных линий.

Первая производственная линия

```
s1stLineModel
b1stLineStatus
u1stLinePlanQty
u1stLineActualQty
```

Вторая производственная линия

```
s2ndLineModel
b2ndLineStatus
u2ndLinePlanQty
u2ndLineActualQty
```

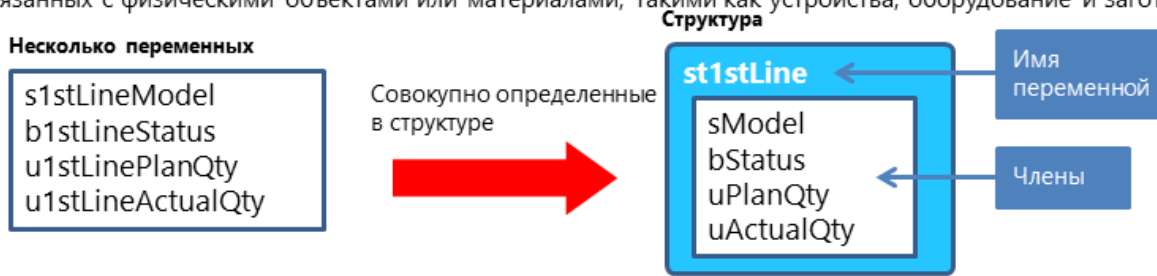


По мере увеличения количества производственных линий увеличивается количество подлежащих обработке переменных.

В результате этого программа становится длиннее и сложнее для чтения.

Структура позволяет одному имени переменной сопоставлять несколько переменных, связанных с одной производственной линией.

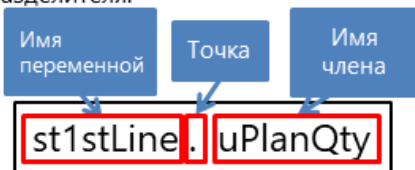
Таким образом, структуры используются для совокупной организации, хранения и обработки условий и технических данных, связанных с физическими объектами или материалами, такими как устройства, оборудование и заготовки.



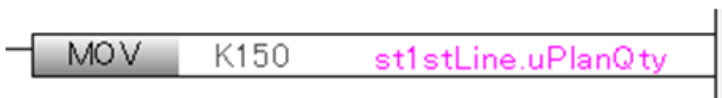
Структурные переменные имеют префикс «st», который представляет собой **structure** (структуру).

Отдельные определенные структурой переменные называются членами. Типы данных каждого члена могут быть разными.

Чтобы задать каждый член в структуре, добавьте имя члена после имени переменной структуры с точкой (.) в качестве разделителя.



В следующем примере программы константа назначается члену переменной структурного типа для первой производственной линии.



2.8

Удержание состояния операнда (фиксация)

В этом разделе описана функция фиксации, которая удерживает значения операндов при прекращении работы модуля ЦП. Например, даже при сбое электроснабжения, продолжительность которого превышает допустимое время кратковременного сбоя электроснабжения, программируемый контроллер может перезапустить последовательное управление, используя данные, сохраненные на момент останова операции.

Если функция фиксации не используется, значения операндов сбрасываются и возвращаются к исходным параметрам (выключено для битовых операндов и имеет значение «0» для словных операндов) в указанных далее случаях.

- Выключение питания
- Сброс переключателем RUN/STOP/RESET (ЗАПУСК/ОСТАНОВ/СБРОС)
- Сбой электроснабжения, превышающий по продолжительности допустимое время кратковременного сбоя электроснабжения

2.8.1

Настройка фиксации операндов

Установите диапазон фиксации в окне «Device Setting» (Настройка операндов) параметра ЦП. Далее приведен пример установки фиксации регистра данных, от D0 до D128.

No.	Device	Points (Decimal)	Start	End
1	D	129	0	128
2				

Операнд, подлежащий фиксации

Начальный номер фиксированных операндов

Конечный номер фиксированных операндов

2.8.2

Типы фиксации и методы сброса

Существует два типа фиксации (фиксация (1) и фиксация (2)) в соответствии с методами сброса.

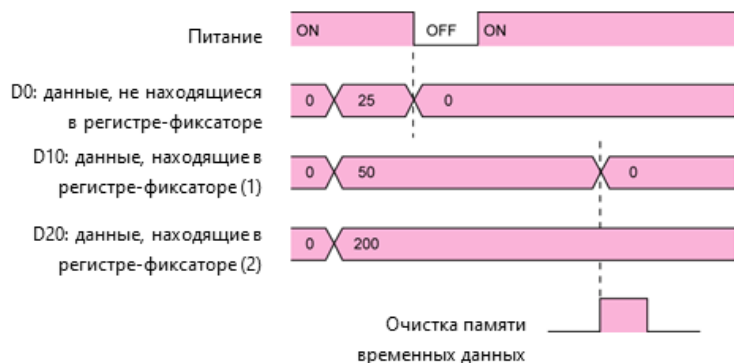
- Latch (1) (Фиксация (1))

Фиксированные данные можно сбросить с помощью обращения к памяти процессора* в GX Works3. Фиксацию 1 можно использовать, когда фиксированные данные необходимо сбросить непосредственно вручную на модуле ЦПУ.

- Latch (2) (Фиксация (2))

Фиксированные данные можно сбросить с помощью инструкции в программе. Фиксацию 2 можно использовать, когда фиксированные данные не сброшены непосредственно вручную на модуле ЦПУ.

Ниже приведена временная диаграмма очистки памяти временных данных.



No.	Device	Points (Decimal)	Start	End
1	D	129	0	128
2				

* Выполните функцию, выбрав команду [Online] (Онлайн) → [CPU Memory Operation] (Обращение к памяти ЦП) в меню GX Works3.

2.8.3

Установка фиксации на переменные

Для установки фиксации на переменные выберите имя класса, содержащее «RETAIN», в окне для установки переменных.

Для локальных переменных выберите «VAR_RETAIN».

Label Name	Data Type		Class
uData	Word [Unsigned]/Bit String [16-bit]	...	VAR_RETAIN
		...	

2.9

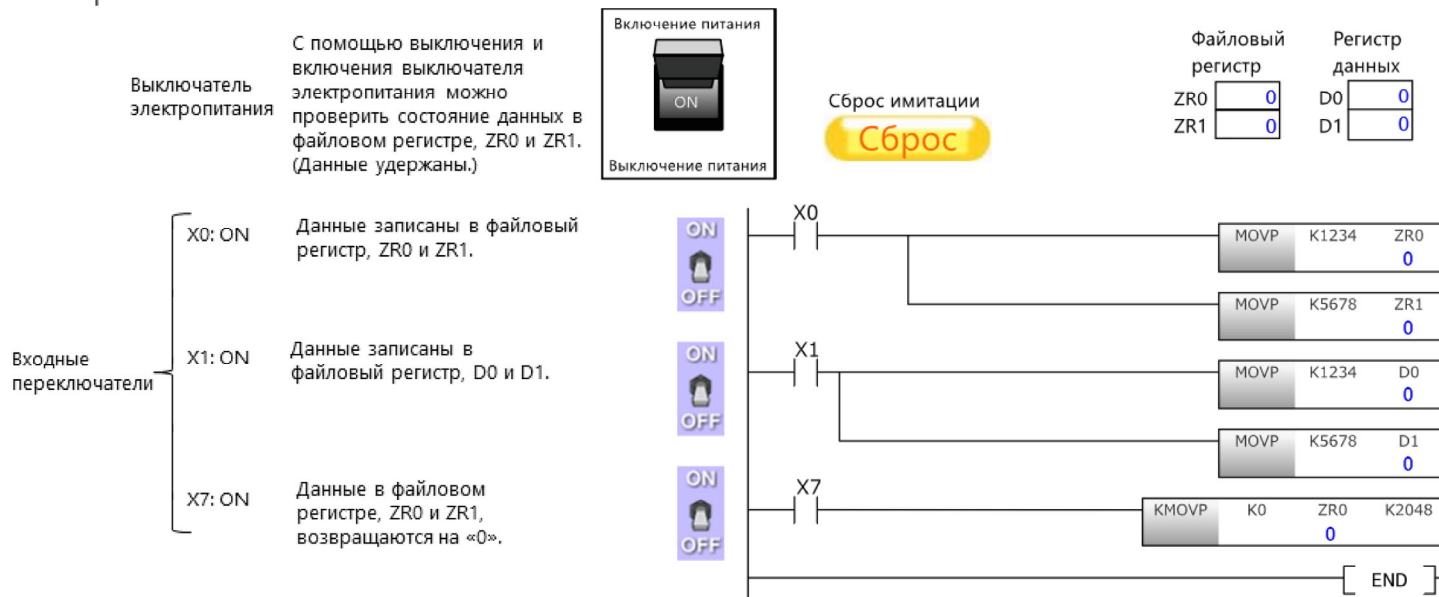
Удержание состояния операнда (файловый регистр)

- Файловый регистр представляет собой словный операнд, используемый для расширения регистра данных (D).
- По сравнению с регистром данных файловый регистр может обрабатывать больший объем данных.
- Файловый регистр хранится в памяти для хранения данных модуля ЦП или на карте памяти.
- Сохраненные в файловом регистре данные будут удерживаться даже при выключении электропитания системы или сбросе модуля ЦП.
Файловый регистр полезен для сохранения заранее заданных значений, таких как стандартные значения.
- Чтобы сбросить данные, запишите 0 для диапазона файлового регистра или очистьте память из GX Works3.
- Символ операнда: «ZR».

2.9.1

Работа программы на языке релейной логики

Создать имитацию работы файлового регистра можно путем включения входных переключателей и выключателя электропитания.



2.9.2

Настройка для файлового регистра

В этом разделе описывается установка, задающая локальный файловый регистр в качестве места хранения каждой программы.

Выберите «File Register Setting» (Настройка файлового регистра) в параметре ЦП, а затем выберите «Use File Register of Each Program» (Использовать файловый регистр каждой программы).

Item	Setting
File Register Setting	
Use Or Not Setting	Use File Register of Each Program
Capacity	
File Name	

Для записи данных в программируемый контроллер настройку файлового регистра необходимо записать в каждую программу.

The screenshot shows the 'Online Data Operation' software interface. The main window displays a table of parameters and their settings. The 'File Register' parameter is highlighted with a red box. A red arrow points from the 'File Register' row in the table to the 'File Register Detail Setting' dialog box. The dialog box shows the 'Specify Range' option selected, with the range set to 'ZR 0 - 32767'. A yellow callout bubble points to the dialog box with the text 'Установите диапазон регистров файлов на запись.' (Set the range of file registers for writing.)

Module Name/Data Name	Detail	Title
Memory Card Parameter		
Remote Password	<input type="checkbox"/>	
Global Label	<input type="checkbox"/>	
Global Label Setting	<input type="checkbox"/>	
Program	<input type="checkbox"/>	Detail
MAIN	<input type="checkbox"/>	
Device Memory	<input type="checkbox"/>	
MAIN	<input type="checkbox"/>	Detail
File Register	<input type="checkbox"/>	Detail
MAIN	<input type="checkbox"/>	Detail
Common Device Comment	<input type="checkbox"/>	2018/05/08 10:35:52 -
COMMENT	<input type="checkbox"/>	2018/05/08 10:35:52 Not Calculated

File Register Detail Setting

Whole Range

Specify Range ZR 0 - 32767

Default OK Cancel

2.10

Использование с операндов с предустановленными функциями и операциями

Специальное реле и специальный регистр, используемые в модуле ЦП, имеют заранее предустановленные функции и операции.

Специальное реле (SM) представляет собой внутренний маркер, используемый для битовой информации (вкл./выкл.), а специальный регистр (SD) является внутренним регистром, используемым для информации о слове.

2.10.1

Типы специальных реле и специальных регистров

Специальное реле и специальный регистр классифицируются по типам информации. Основные типы перечислены ниже. В пользовательских программах специальное реле и специальный регистр используются в качестве условий определения для параметров управления.

Они также используются для контроля операции, которая может выполняться на мониторе операндов в GX Works3.

Диагностическая информация
Результаты диагностики модуля ЦП
Диагностические ошибки и коды ошибок
Информация о системе
Информация о системе модуля ЦП
Рабочее состояние модуля ЦП, данные о часах и прочая информация
Системные часы
Сигналы синхронизации и значения счета, которые используются в качестве основных синхронизирующих элементов.
Различные сигналы синхронизации (всегда ON (ВКЛ.), ON/OFF (ВКЛ./ВЫКЛ.) через определенные промежутки и другие сигналы)

2.10.2

Пример программы специального реле и специального регистра

Ниже приведен пример программы для считывания данных о часах модуля ЦП с помощью специального реле и специального регистра.

Специальное реле (Always ON (всегда ВКЛ.))

SM400

SM213

X0

MOV

SD210

D100

MOV

SD211

D101

MOV

SD212

D102

MOV

SD213

D103

MOV

SD214

D104

MOV

SD215

D105

MOV

SD216

D106

Специальный регистр (от SD210 до SD216), где хранятся данные о часах

Специальное реле, отправляющее запрос на считывание данных о часах модуля ЦП

SM213 (запрос на считывание данных о часах) включается во время РАБОТЫ.

Данные о часах (год)

Данные о часах (месяц)

Данные о часах (день)

Данные о часах (час)

Данные о часах (минута)

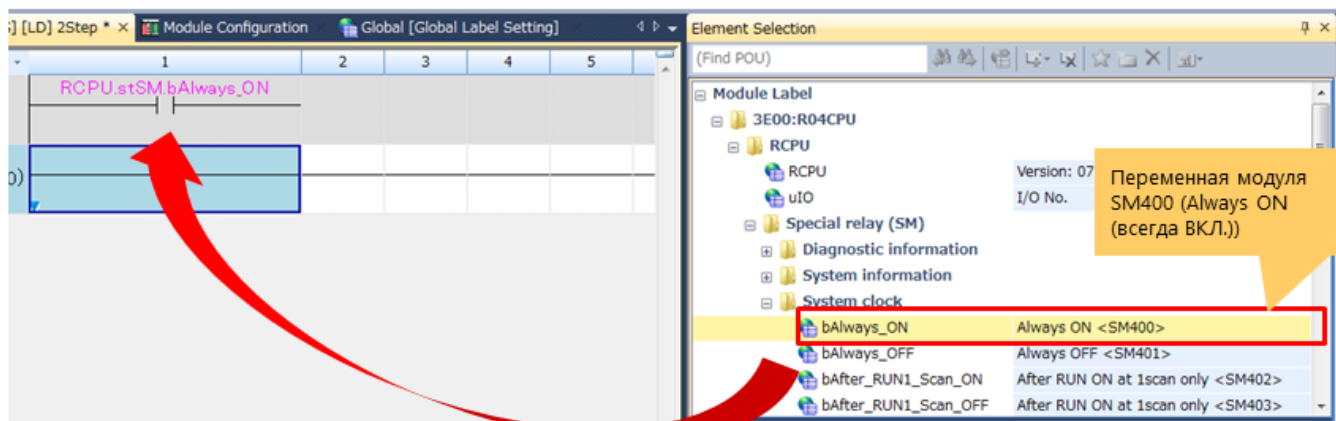
Данные о часах (секунда)

Данные о часах (день недели)

2.10.3

Использование переменных для специальных реле и специальных регистров

Специальное реле и специальный регистр подготавливаются в виде переменных модуля в модуле ЦП. Они могут быть использованы путем простого выбора и размещения соответствующих имен переменных без проверки адресов операнда в руководстве по эксплуатации.



2.11

Вычисления с действительными числами

2.11.1

Применение действительных чисел

- «Действительные числа» представляют собой числовые значения с десятичной запятой
- Целые числа обычно используются в управляющих программах. Тем не менее, действительные числа используются в программах для расширенного контроля работы, например для тригонометрической функции или операции возведения в степень, поскольку числовые значения с десятичной запятой должны обрабатываться в таких программах.
- Числовые данные действительных чисел, обрабатываемых в модуле ЦП, называются «данные с плавающей точкой».

Примечание.

- Одно действительное число **всегда использует два последовательных словных операнда** (32-битная область памяти), независимо от числа.*
- В управляющих программах подготовлены **специальные инструкции по эксплуатации** (например, сложение, вычитание, умножение, деление и специальные функции), которые управляют действительными числами. Также подготовлены инструкции по преобразованию целых чисел в действительные числа и наоборот.

* Если требуется более точный расчет с большим количеством значащих цифр, используйте четыре словных операнда.

- Действительные числа, которые используют два словных операнда, называются «действительные числа с одинарной точностью»
- Действительные числа, которые используют четыре словных операнда, называются «действительные числа с двойной точностью»

В этом курсе основное внимание уделяется действительным числам с одинарной точностью.

«E» используется для представления действительного числа.

Выражение константы с действительными числами

Чтобы записать константу, начните с «E».

Нормальное выражение	Запишите числовое значение как есть. (Пример) Запишите 10,2345 в виде «E10.2345».
Экспоненциальное выражение	Запишите числовое значение в виде «(числовое значение) $\times 10^n$ ». (Пример) Запишите 1234,0 в виде «E1.234+3».

Символ инструкции	Пример релейно-контактной схемы	
E+ (добавление действительных чисел одинарной точности)	 Операция с действительным числом « $D + S = D$ » выполнена.	 Операция с действительным числом « $S1 + S2 = D$ » выполнена.
E- (вычитание действительных чисел одинарной точности)	 Операция с действительным числом « $D - S = D$ » выполнена.	 Операция с действительным числом « $S1 - S2 = D$ » выполнена.

S (источник): данные перед операцией (константа, адрес операнда).

D (назначение): назначение данных после операции (адрес операнда).

P: инструкции, которые выполняются на переднем фронте сигнала (от выключения к включению).

S1 и S2: две единицы данных для работы.

Примечание.

Целые и действительные числа нельзя совместно использовать в одной операции.

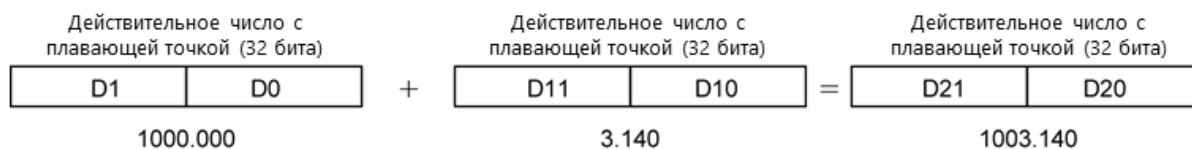
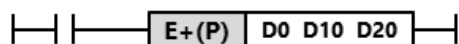
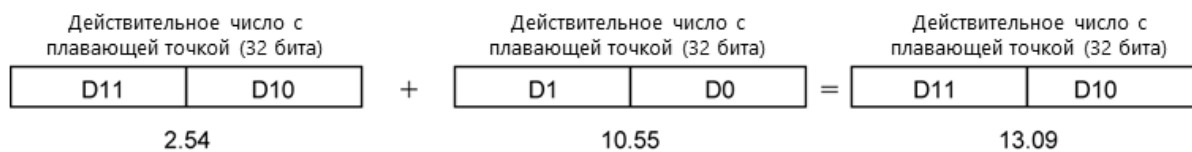
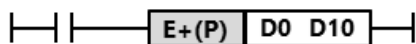
Для операций с действительными числами одинарной точности значения S, S1 и S2 в выражении операции должны быть действительными числами одинарной точности.

Действительные числа одинарной точности сохраняются в D.

2.11.3

Руководство по эксплуатации (сложение и вычитание)

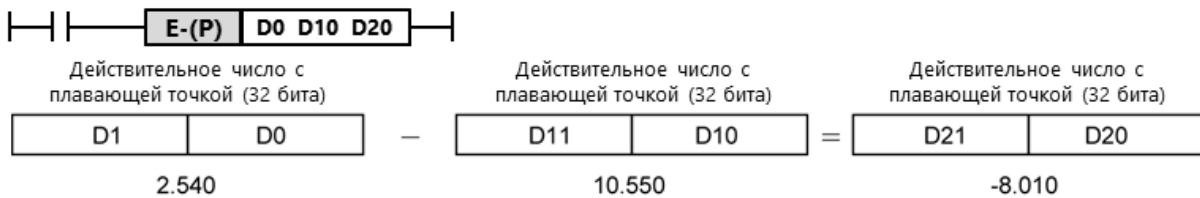
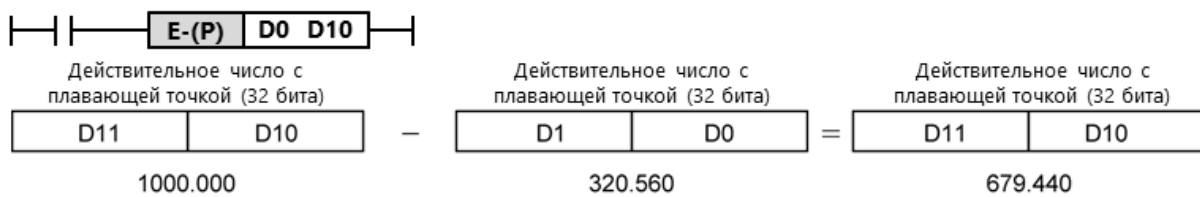
Пример применения инструкции по сложению



2.11.3

Руководство по эксплуатации (сложение и вычитание)

Пример применения инструкции по вычитанию



2.11.4

Инструкция по эксплуатации (умножение и деление)

Символ инструкции	Пример релейно-контактной схемы
E* (умножение действительных чисел одинарной точности)	 Операция с действительным числом « $S1 * S2 = D$ » выполнена.
E/ (деление действительных чисел одинарной точности)	 Операция с действительным числом « $S1 / S2 = D$ » выполнена.

S1, S2 (источник): две единицы данных для работы.

D (назначение): назначение данных после операции (адрес операнда).

P: инструкции, которые выполняются на переднем фронте сигнала (от выключения к включению).

Примечание.

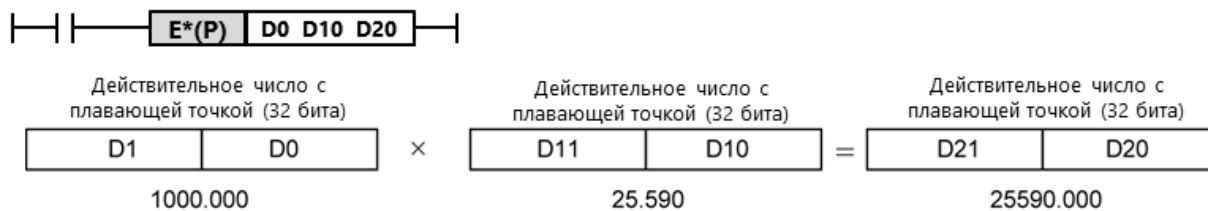
Для операций с действительными числами одинарной точности значения S1 и S2 в выражении операции должны быть действительными числами одинарной точности.

Действительные числа одинарной точности сохраняются в D.

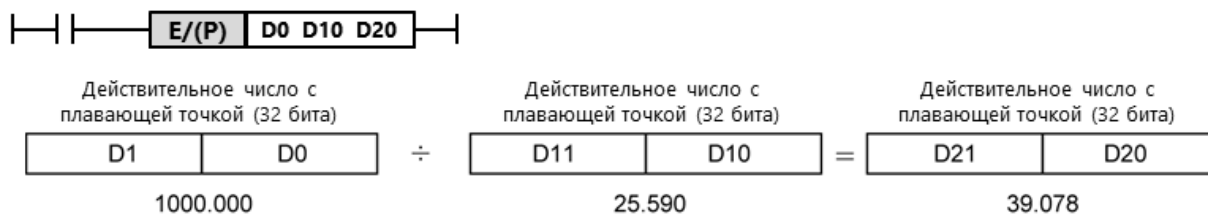
2.11.4





Инструкция по эксплуатации (умножение и деление)

Пример применения инструкции по умножению



Пример применения инструкции по делению



Символ инструкции	Пример релейно-контактной схемы	
INT2FLT (преобразование целого числа в действительное число одинарной точности)	Целое число (16 бит) преобразовывается в действительное число (32 бита).  S (16 бит) преобразовано и сохранено в D.	Целое число (32 бита) преобразовывается в действительное число (32 бита).  S (32 бита) преобразовано и сохранено в D.
FLT2INT (преобразование действительного числа одинарной точности в целое число)	Действительное число (32 бита) преобразовывается в целое число (16 бит).  S преобразовано и сохранено в D (16 бит).	Действительное число (32 бита) преобразовывается в целое число (32 бита).  S преобразовано и сохранено в D (32 бита).

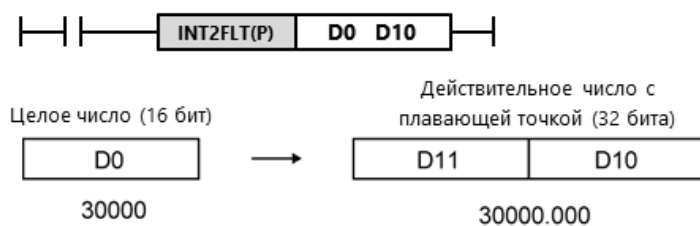
S (источник): данные перед операцией (константа, адрес операнда).

D (назначение): назначение данных после операции (адрес операнда).

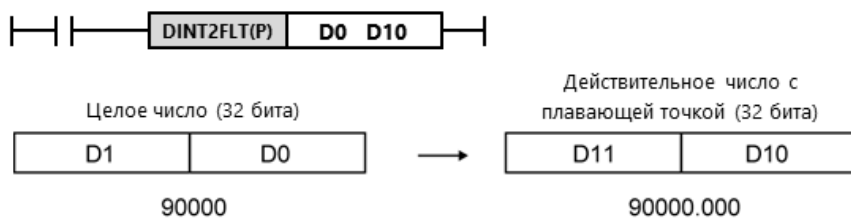
2.11.5

Инструкции по преобразованию целых чисел в действительные числа и наоборот

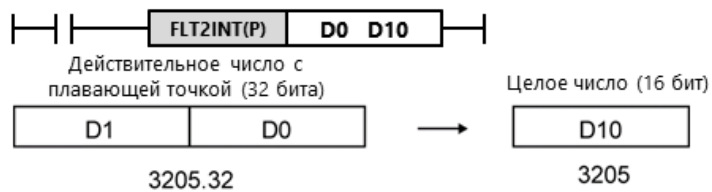
Пример применения инструкции по преобразованию целого числа (16 бит) в действительное число (32 бита)



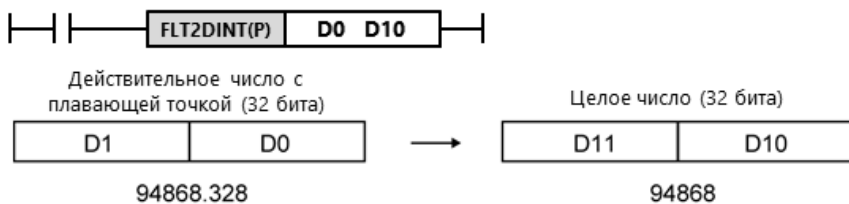
Пример применения инструкции по преобразованию целого числа (32 бита) в действительное число (32 бита)



Пример применения инструкции по преобразованию действительного числа (32 бита) в целое число (16 бит)



Пример применения инструкции по преобразованию действительного числа (32 бита) в целое число (32 бита)



2.11.6

Использование меток, представляющих действительные числа

Для использования меток для действительных чисел установите для типа данных значение «Single Precision» (Одинарная точность) или «Double Precision» (Двойная точность) в окне установки меток.

Label Name	Data Type
eData	FLOAT [Single Precision]
leData	FLOAT [Double Precision]

Из этой главы вы узнали о:

- задании бита в словном операнде;
- задании переднего или заднего фронта для контактов;
- фиксируемом таймере;
- характеристике единицы измерения таймера;
- индексном регистре;
- массиве;
- структуре;
- фиксации;
- файловом регистре;
- специальном реле и специальном регистре;
- расчетах с действительными числами.

Важные моменты

Фиксируемый таймер	Измеренное время удерживается даже при выключении обмотки, и измерение возобновляется при последующем включении обмотки.
Единица измерения таймера	Единицу измерения таймера можно изменить в параметре.
Индексный регистр	В одном пакете можно описать несколько устройств.
Массив	Несколько значений можно обрабатывать с помощью имени одной переменной.
Структура	Несколько связанных между собой переменных можно обработать с помощью имени одной переменной.
Фиксация	<ul style="list-style-type: none"> • Значения фиксированного операнда удерживаются, когда модуль ЦП прекращает работу • Значения фиксированного операнда сбрасываются путем обращения к памяти ЦП или использования программной инструкции
Файловый регистр	<ul style="list-style-type: none"> • По сравнению с регистром данных файловый регистр может обрабатывать больший объем данных • Значения операнда удерживаются, когда модуль ЦП прекращает работу • Значения операнда можно сбросить, обратившись к памяти ЦП или записав 0 в диапазон операндов
Специальное реле, специальный регистр	Внутреннее состояние модуля ЦП, например диагностическая информация и системная информация, уже сохранено в этих устройствах.
Действительное число	<ul style="list-style-type: none"> • Используется не менее двух словных операндов (32 бита) • Предоставляются специальные инструкции по эксплуатации • Целые и действительные числа нельзя совместно использовать в одной операции.

В этой главе описываются функции GX Works3 для эффективной отладки.

- 3.1 Временное изменение диапазона программы
- 3.2 Проверка работы при изменении значений операнда
- 3.3 Имитация работы программы

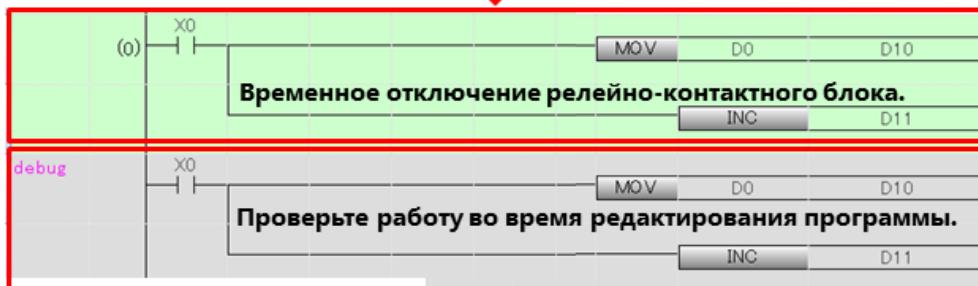
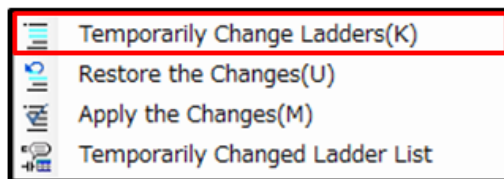
3.1

Временное изменение диапазона программы

При редактировании широкого диапазона программы для отладки очень сложно отменить все изменения. Временно отключив необходимый релейно-контактный блок, пользователи могут использовать копию программы для отладки без изменения оригинала.

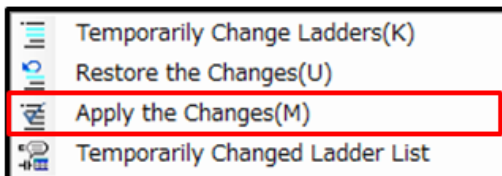
(Функция Temporarily change ladders (Временное изменение релейно-контактных блоков))

После изменения релейно-контактного блока и проверки работы с этой функцией изменения применяются при условии отсутствия проблем или если изменения восстанавливаются при возникновении проблемы.

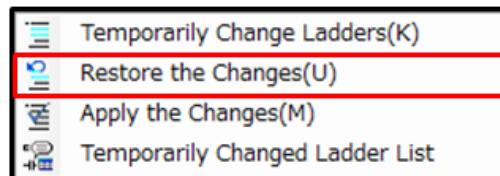


Скопируйте отключенный релейно-контактный блок.

Если проблемы не обнаружено



Если обнаружена проблема

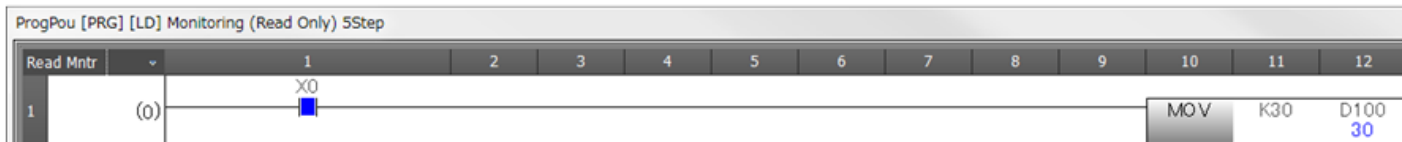


3.2

Проверка работы при изменении значений операнда

При запуске созданной программы состояние включения/выключения битовых операндов и значений, сохраненных в словных операндах, можно отобразить в редакторе программы. (Функция Monitor (Монитор))

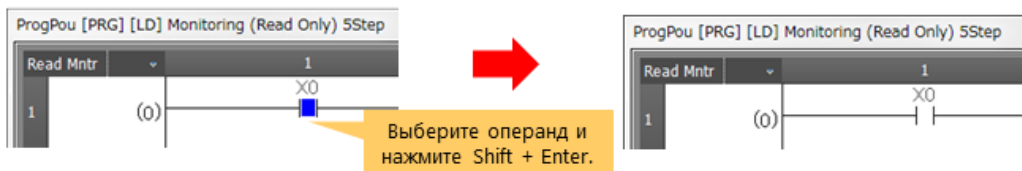
С помощью функции монитора пользователи могут легко проверить рабочее состояние программы.



Текущие значения устройства можно принудительно изменить во время мониторинга. (Изменение текущего значения)

С помощью функции изменения текущего значения изменения можно выполнить без редактирования всей программы или ее запуска на реальной системе.

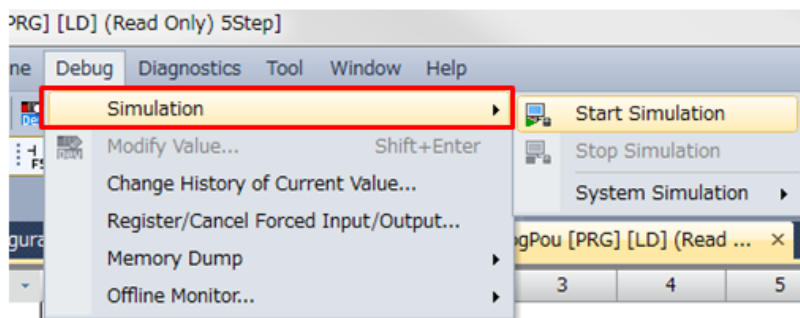
Состояние битовых операндов можно включить в редакторе программ.



С помощью окна «Watch» (Просмотр) можно зарегистрировать словные операнды, подлежащие мониторингу, а их текущие значения можно изменить.

Name	Current Value	Display Format	Data Type
X0	FALSE	BIN	Bit
D0	100	Decimal	Word [Signed]

Если запустить созданную программу на реальной системе, это может привести к неожиданной ошибке. Создать имитацию работы программы можно без использования реального программируемого контроллера. (Функция Simulation (Имитация).)



С помощью функции имитации можно проверить работу программы так, как если бы она выполнялась на реальном программируемом контроллере.

Из этой главы вы узнали о:

- временном изменении релейно-контактного блока;
- мониторинге программы и изменении текущих значений;
- имитация программы.

Важные моменты

Временное изменение релейно-контактного блока	Релейно-контактный блок можно временно отключить, а копию программы использовать для отладки без изменения исходной программы.
Монитор	<ul style="list-style-type: none">• Процесс работы программы можно визуализировать.• Работу программы можно проверить при принудительном изменении текущих значений операнда.
Имитация	Создать имитацию работы программы можно без использования реального программируемого контроллера.

На этом мы завершаем этот курс электронного обучения.

Далее приводится краткое описание этого курса.

- **Эффективное программирование**

- Назначение модулям фиксированных номеров ввода/вывода для удобства использования программ в различных системах.
- Настройка области памяти в соответствии с состоянием использования операндов.
- Использование переменных для упрощения программирования и лучшего понимания работы.
- Добавление комментариев для улучшения читаемости программы.

- **Расширенное программирование**

- Использование фиксируемого таймера для удержания измеренного времени.
- Использование индексного регистра, массивов или структур для совокупной работы со значениями.
- Использование функции фиксации и файлового регистра для удержания состояния операндов.
- Предоставление специального реле и специального регистра, в котором сохраняется внутреннее состояние модуля ЦП.
- Действительное число представляется в виде двух или четырех словных операндов. Целые и действительные числа нельзя совместно использовать в одной операции.

- **Эффективная отладка**

- С помощью GX Works3 пользователи могут выполнять указанные далее действия.
- Выполнять отладку программы без изменения исходной программы.
- Визуализировать процесс работы программы.
- Имитировать работу программы.

Чтобы перейти на следующий этап, пройдите следующие курсы по процедуре структурирования, с помощью которой программа разбивается на слои и компоненты, пригодные для повторного использования.

- Эффективное программирование
- Эффективное программирование (практика) (будет опубликовано позднее)

Какое из следующих предложений верно в отношении присвоения модулям номеров вводов/выводов?

Q1

- Номера вводов/выводов можно назначить каждому модулю вручную, что исключает необходимость изменять программу при изменении конфигурации модуля.
- Автоматически назначаемые номера вводов/выводов нельзя изменять.

Какое из следующих предложений верно в отношении установки количества операндов?

Q1

- Каждому операнду должна быть назначена хотя бы одна точка, даже если операнд не используется.
- Точки могут быть назначены в соответствии с используемым количеством точек.

Какие из следующих предложений верны в отношении переменных? (Несколько ответов)

Q1

Использование переменных упрощает программирование

Предусматриваются переменные, определяющие сигналы модуля и значения настроек

К элементам могут быть добавлены комментарии для улучшения читаемости программы

Поскольку переменным могут быть назначены константы, их значения можно изменять без изменения программы

Выполните следующий текст, описывающий фиксируемый таймер.

Фиксируемый таймер запускается при включении (Q1) (включение обмотки (Q2)).

Фиксируемый таймер сохраняет измеренное время даже при включении условия на входе (Q3) и продолжает измерение с сохраненного значения при повторном включении условия на входе (Q4).

Тайм-аут фиксируемого таймера происходит, когда значение измеренного времени достигает значения

Q1

-- Select --



Q2

-- Select --



Q3

-- Select --



Q4

-- Select --



Q5

-- Select --



Q6

-- Select --



Выполните программу управления на основании приведенного ниже алгоритма.

- Использовать фиксируемый таймер (ST0) для измерения времени включения входного сигнала X0 или X1
- Когда время включения X0 или X1 достигнет 30 секунд, включается обмотка Y70 и индикатор тайм-аута
- Когда X2 включается, отключается контакт фиксируемого таймера (ST0) и сбрасывается значение измеренного времени (текущее значение)

Q1

-- Select --



Q2

-- Select --



Q3

-- Select --



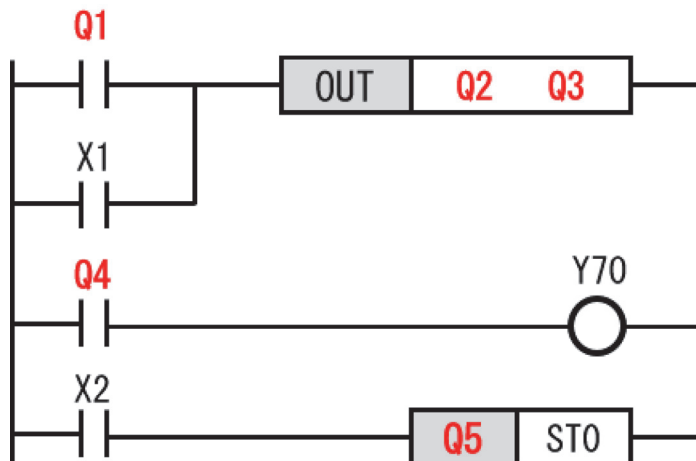
Q4

-- Select --



Q5

-- Select --



[+]

Выберите значение, которое сохраняется в регистре данных D20, когда X0 включается при каждом из следующих условий в указанной ниже управляющей программе.

Q1) Когда значение, сохраненное в Z2, равно «0»

Q2) Когда значение, сохраненное в Z2, равно «1»

Q3) Когда значение, сохраненное в Z2, равно «2»

Q1

-- Select --



Q2

-- Select --



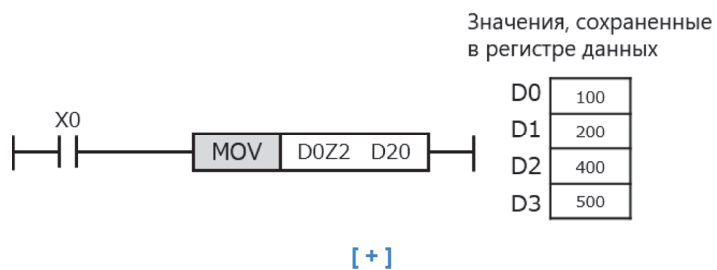
Q3

-- Select --



Q4

-- Select --



Какое из следующих предложений верно в отношении того, как задавать элемент массива?

Q1

- Добавить номер элемента к концу имени переменной
- Косвенно задать адрес операнда

Какое из следующих предложений не верно в отношении структур?

Q1

- Структуры используются для организации и сохранения условия и технических данных, относящихся к физическим объектам или материалам
- Благодаря использованию структур процедуру обработки большого количества данных можно описать кратко
- Определенные в структуре данные должны иметь одинаковый тип

Какое из следующих предложений верно в отношении функции фиксации?

Q1

- Операнды изначально имеют функцию для удержания значений
- Для сохранения значений операндов необходимо настроить соответствующие параметры

Дополните следующий текст, описывающий файловый регистр.

Файловый регистр представляет собой словный операнд, используемый для расширения регистра данных (D), а его символ операнда – (Q1).

Файловый регистр имеет вместимость (Q2) по сравнению с регистром данных, и содержит сохраненные данные (Q3), даже когда система выключена или выполнен сброс модуля центрального процессора.

Q1

-- Select --

Q2

-- Select --

Q3

-- Select --

Q4

-- Select --

Какое из следующих предложений верно в отношении специального реле и специального регистра?

Q1

- Внутреннее состояние модуля центрального процессора хранятся в специальных реле и специальных регистрах, и эти специальные операнды могут использоваться в управляющей программе
- Специальные функции можно свободно назначать специальному реле и специальному регистру

Дополните следующий текст, описывающий действительные числа (одинарная точность).

- Одно действительное число использует словные операнды в количестве **(Q1)** и сохраняется в **(Q2)** -битной области памяти.
- Данные числового значения действительных чисел называются **(Q3)** . Например, 2,035 описывается как **(Q4)** в программе управления.

Q1

-- Select --



Q2

-- Select --



Q3

-- Select --



Q4

-- Select --



Q5

-- Select --



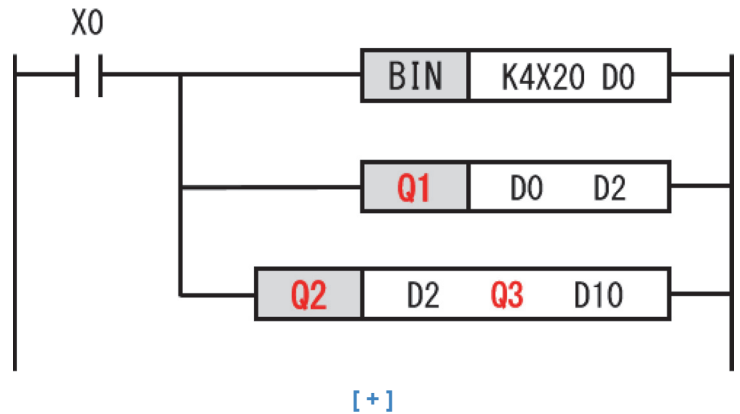
Дополните программу управления, в соответствии с приведенным ниже алгоритмом.

- Считать данные с X20 по X2F (двоично-десятичные данные), когда X0 включено, и сохранить их в D0.
- Преобразовать значение в D0 в действительное число и сохранить преобразованное значение в D2.
- Умножить значение в D2 на 3,14 и сохранить результат в D10.

Q1

Q2

Q3



Какое из следующих предложений верно в отношении отладки программы управления?

Q1

- Работу программы можно симитировать с помощью специального функционала программного обеспечения
- Для отладки программы она должна выполняться в реальной системе.

Вы завершили заключительный тест.
Ваша область результатов является следующей.

	1	2	3	4	5	6	7	8	9	10
Заключительный тест 1	✓									
Заключительный тест 2	✓									
Заключительный тест 3	✓									
Заключительный тест 4	✓	✓	✓	✓	✓	✓				
Заключительный тест 5	✓	✓	✓	✓	✓					
Заключительный тест 6	✓	✓	✓	✓						
Заключительный тест 7	✓									
Заключительный тест 8	✓									
Заключительный тест 9	✓									
Заключительный тест 10	✓	✓	✓	✓						
Заключительный тест 11	✓									
Заключительный тест 12	✓	✓	✓	✓	✓					
Заключительный тест 13	✓	✓	✓							
Заключительный тест 14	✓									

Всего вопросов: **35**

Правильные ответы: **35**

Процент: **100 %**

Сброс

Вы завершили курс **Прикладное программирование
(релейно-контактная схема/серия MELSEC iQ-R).**

Благодарим за прохождение этого курса.

Надеемся, что вам понравились уроки, а информация, полученная в рамках этого курса, окажется полезной в будущем.

Вы можете проходить данный курс любое количество раз.

[Просмотреть](#)

[Закреть](#)