



Углубленный курс по работе с ПО для ПЛК GX Works2

Настоящий курс посвящен функциям, предназначенным для улучшения среды разработки предприятия с недостатками в области производительности, качества, управления проектами и мер безопасности. Курс предназначен для программистов последовательных программ, которые уже используют в работе программное обеспечение MELSOFT GX Works2.

Введение **Цель курса**

Настоящий курс описывает возможности и функции, предназначенные для улучшения качества процесса разработки проектов автоматизации, являющиеся особенно полезными для решения задач увеличения производительности работы, задач управления проектами и обеспечения безопасности. Курс предназначен для программистов, которые уже используют в работе программное обеспечение MELSOFT GX Works2.

Введение Структура курса

Данный курс имеет следующее содержание.
Рекомендуем вам начать с Главы 1.

Глава 1. Оптимизация процесса разработки с помощью инструментов ПО GX Works2

Вы познакомитесь со сложностями, возникающими в процессе разработки проектов и способами упрощения и оптимизации процесса разработки.

Глава 2. Программирование

Вы изучите функции, применяемые в программировании.

Глава 3. Отладка

Вы изучите функции, используемые для отладки.

Глава 4. Управление проектами и меры безопасности

Вы изучите функции для управления проектами и обеспечения безопасности на этапе разработки и обслуживания системы после ее ввода в эксплуатацию.

Заключительный тест

Проходной балл: 60% или выше

Введение Как использовать этот инструмент электронного обучения

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. Окно (например, «Содержание») будет закрыто, а обучение — завершено.

Введение **Меры предосторожности при использовании**

Меры безопасности

Если при обучении используются реальные продукты, внимательно прочтите меры безопасности в соответствующих инструкциях к ним.

Меры предосторожности относительно данного курса

- Отображаемые экраны с версией используемого программного обеспечения могут отличаться от тех, которые используются в данном курсе.

Глава 1 Оптимизация процесса разработки с помощью инструментов ПО GX Works2

Этапы обучения, предусмотренные в главе 1

Настоящий курс предназначен для программистов ПЛК. Вы научитесь использовать функции программного обеспечения **MELSOFT GX Works2** для решения проблем при проектировании, связанных с **производительностью, качеством, управлением проектами и безопасностью**.

В главе 1 Вы познакомитесь со сложностями, возникающими в процессе разработки проектов и способами упрощения и оптимизации процесса разработки.

- 1.1. Сложности, возникающие в процессе разработки проектов
 - 1.1.1. Повышение производительности
 - 1.1.2. Повышение качества
 - 1.1.3. Управление проектами
 - 1.1.4. Меры безопасности
- 1.2. Процедура обучения

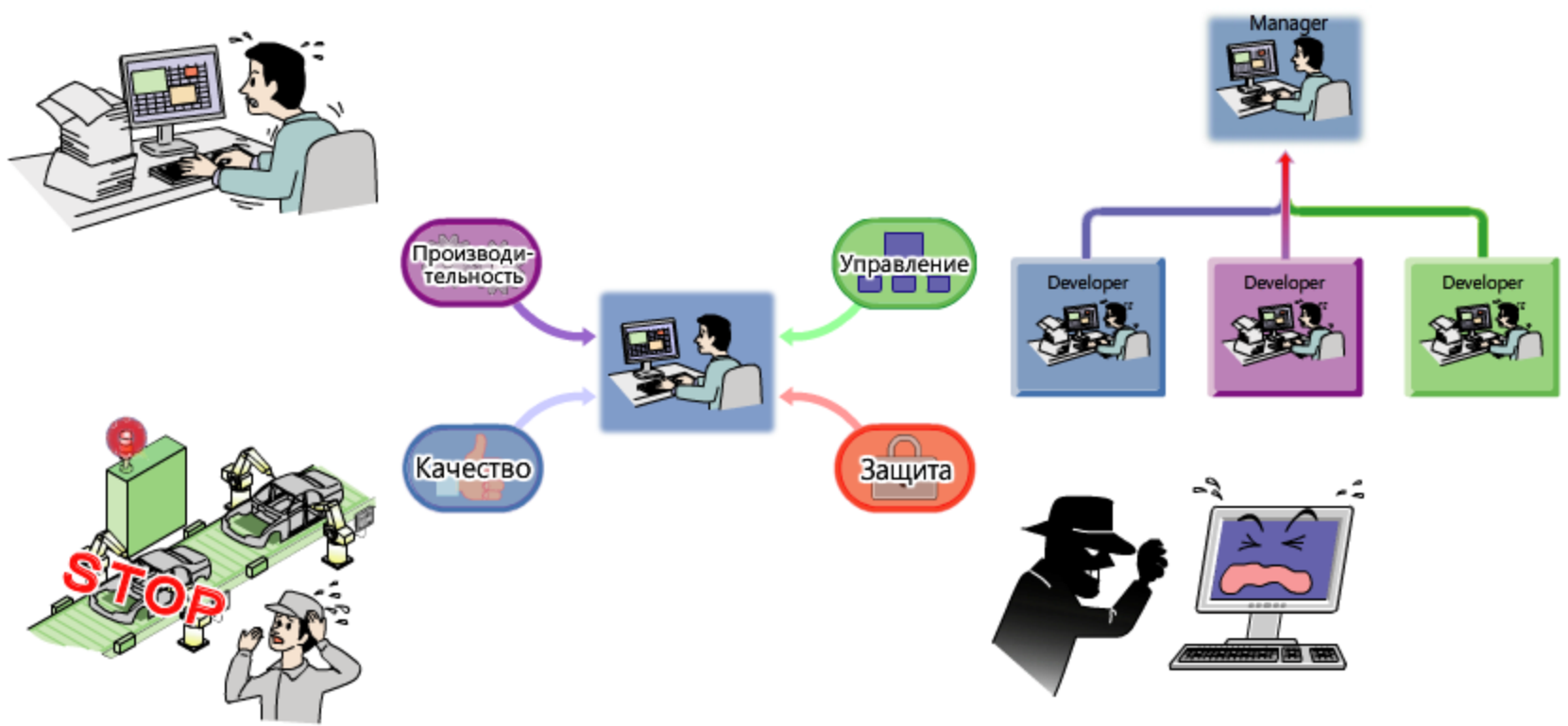


1.1 Сложности, возникающие в процессе разработки проектов

Требования по оптимизации процесса разработки проектов с целью снижения издержек тесно пересекаются с вопросами качества создаваемого продукта.

Одновременно с этим стоит уделять внимание вопросам управления проектами, что позволит обеспечить совместную командную работу по проекту и избежать проблем по восстановлению проекта в случае непредвиденных сложностей

Также стоит не забывать про меры обеспечения информационной безопасности.



1.1.1 Повышение производительности

Процесс создания проектов комплексной масштабной автоматизации зачастую оказывается очень затратным.

Стоимость разработки растет одновременно с увеличением времени на разработку проекта

Таким образом важным становится требование к оптимизации процесса разработки с целью снижения издержек.

Для решения вопросов оптимизации разработки в ПО GX Works2 предусмотрены следующие функции:

- Создание меток
- Использование функциональных блоков;
- Использование начальных значений операндов и памяти операндов*;
- Встроенный структурированный текст;
- Импорт шаблонов комментариев;

* В ПЛК серии MELSEC-F нельзя настраивать начальные значения операндов.



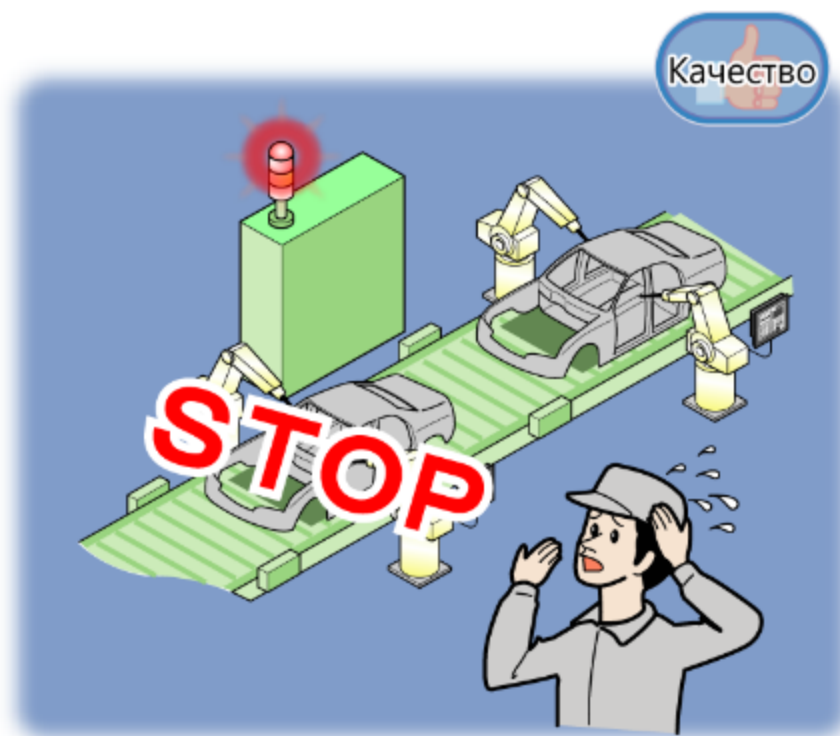
1.1.2 Повышение качества

Ошибка в программе ПЛК может привести к остановке всей системы, неполадкам или аварии, которые могут повлечь за собой остановку производства и потерю прибыли.

Таким образом предъявляются требования к разработке качественных и надежных программ, не содержащих ошибок.

Для решения этой проблемы в ПО GX Works2 предусмотрены следующие функции:

- Создание меток
- Использование функциональных блоков;
- Использование начальных значений операндов и памяти операндов*;
- Встроенный структурированный текст;
- Импорт шаблонов комментариев;
- Отслеживание;
- Перекрестные ссылки;
- Трассировка*;
- Проверка операндов с условием выполнения*;
- Функция пошаговой отработки программы*;
- Настройка системы ввода/вывода.



* В ПЛК серии MELSEC-F не могут использоваться функции настройки начальных значений операндов, трассировки, проверки операндов с условием выполнения и пошаговой отработки.

1.1.3 Управление проектами

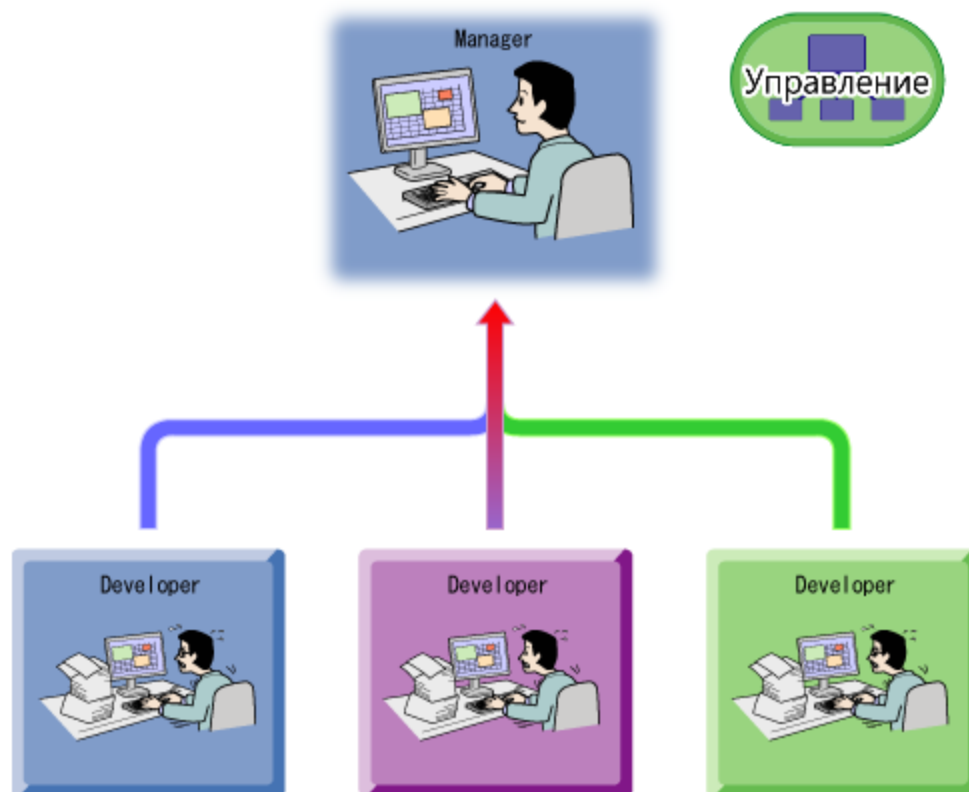
Комплексные и сложные проекты ПЛК часто разрабатываются командой программистов.

С точки зрения управления проектами и обеспечения конфиденциальности данных необходимо иметь возможность ограничивать доступ к данным проекта и использованию функций.

Кроме того, для обеспечения быстрого восстановления в случае потери программы вследствие сбоя программируемого контроллера нужно реализовать управление версиями программ и организовать их периодическое резервное копирование.

Для решения этих проблем в ПО GX Works2 предусмотрены следующие функции:

- Защита.
- Журнал изменений;
- Верификация проектов ПЛК и ПК.



1.1.4 Меры безопасности

Проекты ПЛК могут содержать данные, которые разработчик хочет оставить конфиденциальными (интеллектуальная собственность).

Утечка таких данных может оказать разрушительное воздействие на бизнес.

Несанкционированное изменение программы может привести к большим производственным проблемам, таким как остановка системы.

Для предотвращения таких проблем необходимо принять надлежащие меры безопасности.

Для решения этих проблем в ПО GX Works2 предусмотрена следующая функция защиты данных

- Защита.



1.2 Процедура обучения

В этом курсе вы познакомитесь со следующими процедурами по разработке реальной системы в ПО GX Works2.

(1) Программирование Глава 2



(2) Отладка Глава 3



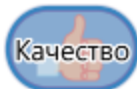
(3) Управление проектами и меры безопасности Глава 4

Обозначение пиктограмм

Пиктограммы, которые встречаются в главах 2, 3 и 4, соответствуют функциям, используемым для улучшения тех или иных аспектов.



Функция, используемая для повышения производительности программы



Функция, используемая для повышения качества программы



Функция, используемая для управления проектом



Функция, используемая для обеспечения безопасности и защиты данных

Глава 2 Программирование

Этапы обучения, предусмотренные в главе 2

В главе 2 вы изучите функции, применяемые в программировании.

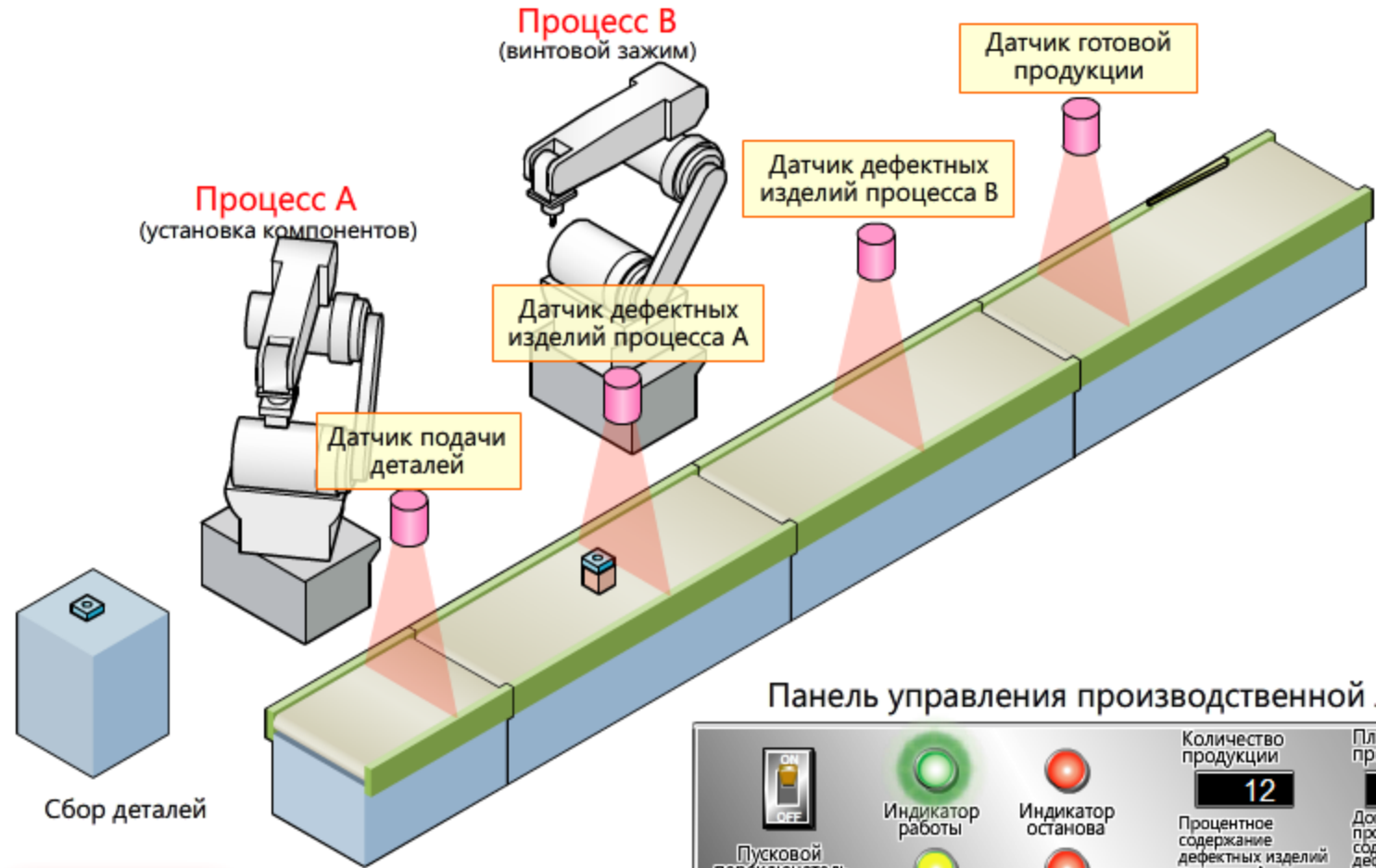
В ПО GX Works2 предусмотрено множество различных функций для обеспечения эффективного программирования и повышения качества программ.

- 2.1. Замена названия операнда на название, связанное с приложением
 - 2.1.1. Типы меток
 - 2.1.2. Типы назначения и сохраненных значений меток
 - 2.1.3. Регистрация меток
 - 2.1.4. Автоматическое присваивание меток операндам
- 2.2. Преобразование неоднократно используемых блоков программ на языке релейной логики в функциональные блоки
 - 2.2.1. Создание и размещение функциональных блоков
 - 2.2.2. Использование библиотеки функциональных блоков
- 2.3. Изменение начальных значений операндов без исправления программы
- 2.4. Упрощение программ на языке релейной логики
- 2.5. Создание программ, простых для понимания и чтения
 - 2.5.1. Запись комментариев для каждого блока программы на языке релейной логики
 - 2.5.2. Запись комментариев для каждого выхода (катушки, инструкции)
- 2.6. Упрощение для чтения программ со специальными реле/регистрами

Глава 2 Система, используемая в настоящем курсе для обучения



В настоящем курсе используется следующая гипотетическая система, с помощью которой вы научитесь использовать функции GX Works2:



Панель управления производственной линии

			Количество продукции	Плановый объем продукции
Пусковой переключатель	Индикатор работы	Индикатор останова	12	20
			Процентное содержание дефектных изделий процесса А	Допустимое процентное содержание дефектных изделий процесса А
Сброс счетчика количества продукции	Достигнуто количество продукции	Индикатор ошибки процентного содержания дефектных изделий процесса А	7	10
			Процентное содержание дефектных изделий процесса В	Допустимое процентное содержание дефектных изделий процесса В
		Индикатор ошибки процентного содержания дефектных изделий процесса В	7	5

Серийное производство

Ошибка процентного содержания дефектных изделий процесса А

Ошибка процентного содержания дефектных изделий процесса В

При нажатии кнопки «К началу» происходит возврат к изначальному выбору операции.

К началу

2.1

Замена названия операнда на название, связанное с приложением

Производи-
тельность

Качество

Используемому в программе **операнду** присваивается название, которое состоит из буквы с цифрой, например «M0» или «D5».

Таким образом, название операнда не дает представления о его применении.

В больших программах используется много различных типов операндов. Это означает, что во время программирования необходимо постоянно проверять документацию по проектированию системы для определения назначения операндов. Это снижает эффективность работы и отрицательно влияет на качество программы из-за ошибок при выборе операндов.

Для решения этих проблем можно использовать **«метки»**.

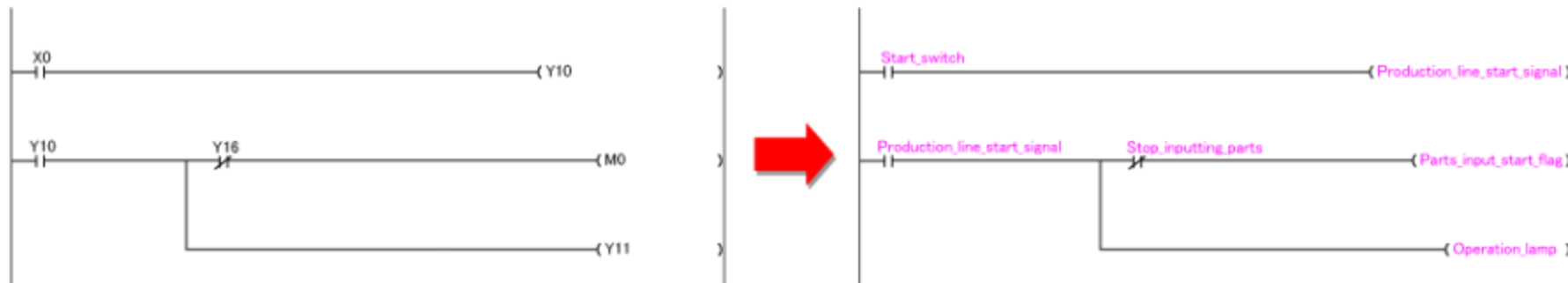
Вместо названия операнда можно использовать имя (метку), обозначающую фактическое применение операнда, например **«Сигнал запуска производственной линии»**.

В этом типе названия помимо буквенно-цифровых могут использоваться японские символы (хирагана, катакана и кандзи).

<Операнд M0 определяет сигнал запуска производственной линии (в состоянии ВКЛ.) с помощью команды SET>

Операнд	—[SET M0]—
Метка	—[SET Сигнал запуска производственной линии]—

Этот тип меток позволяет создавать удобные для чтения программы, повышать эффективность разработки программ и предотвращать появление ошибок ввода операндов.



2.1.1

Типы меток



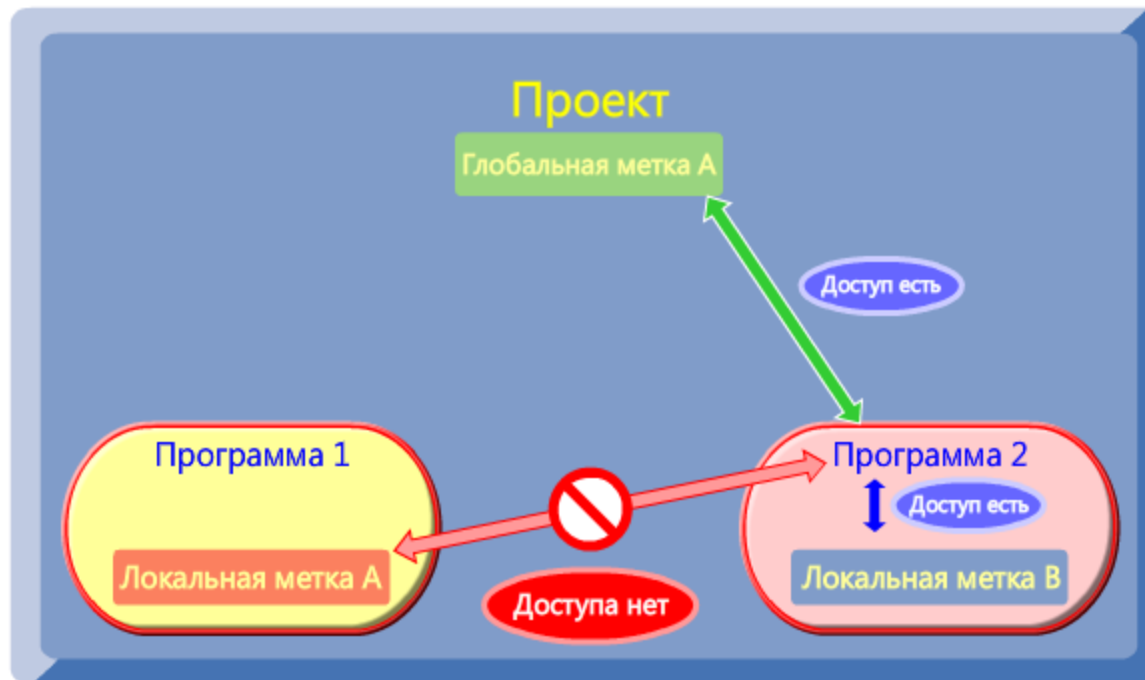
Существует два типа меток: **«глобальные метки»** и **«локальные метки»**.

<Глобальная метка>

Глобальные метки используются для всего проекта. Любая программа в этом проекте может получить к ним доступ.

<Локальная метка>

Локальные метки используются в отдельной программе и доступны только в той программе, в которой зарегистрированы.



«Программа 1» и «Программа 2» представляют собой кнопки. Нажмите на любую программу, чтобы проверить, есть ли у нее доступ к двум типам меток.

2.1.2

Типы назначения и сохраненных значений меток

Производительность

Качество

При регистрации метки нужно указать ее назначение и тип значения, которое может быть записано, используя параметры «Класс» и «Тип данных».

<Класс>

Класс определяет диапазон использования и назначение метки.

Классы, которые можно выбрать, различаются в зависимости от типа метки.

Класс	Предполагаемое использование	Область установки метки		
		Глобальная метка	Локальная метка программы	Локальная метка функционального блока
VAR_GLOBAL	Общая метка, которая может использоваться программами и функциональными блоками в проекте	<input type="radio"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VAR_GLOBAL_CONSTANT	Общая метка с константой, которая может использоваться программами и функциональными блоками в проекте	<input type="radio"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
VAR	Метка, которая может использоваться программами и функциональными блоками, для которых она предназначена	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
VAR_CONSTANT	Метка с константой, которая может использоваться программами и функциональными блоками, для которых она предназначена	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
VAR_RETAIN	Метка (защелка), которая может использоваться программами и функциональными блоками, для которых она предназначена	<input checked="" type="checkbox"/>	<input type="radio"/>	<input type="radio"/>
VAR_INPUT	Метка, используемая для входа функционального блока, для которого она предназначена * Это значение невозможно изменить в компоненте программы.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>
VAR_OUTPUT	Метка, используемая для выхода функционального блока, для которого она предназначена	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>
VAR_IN_OUT	Метка, используемая для входа и выхода функционального блока, для которого она предназначена * Это значение можно изменить в компоненте программы.	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>

2.1.2

Типы назначения и сохраненных значений меток



<Тип данных>

Под типом данных понимается тип значения, записанного в метке.

Тип данных, присваиваемый метке, определяет тип и диапазон значений, которые могут храниться в метке и соответствующем операнде.

Ниже перечислены типы данных, которые могут использоваться в программах на языке релейной логики.

Тип данных	Описание	Длина в битах	Диапазон значений
Bit (Битовый)	Значение «ВКЛ.» или «ВЫКЛ.». Соответствует операнду «M»	1 бит	1: ВКЛ., 0: ВЫКЛ.
Word (signed) (Словный (со знаком))	Целое число без дробей. Соответствует операнду «D»	16 бит	от -32768 до 32767
Double-word (signed) (Двойной словный (со знаком))		32 бит	от -2147483648 до 2147483647
FLOAT (single precision) (ПЛАВАЮЩАЯ ЗАПЯТАЯ (с одинарной точностью))	Действительные числа, включая дроби. Соответствует операнду «D»	16 бит	от -2^{128} до -2^{-126} , 0, от 2^{-126} до 2^{128}
FLOAT (double precision) (ПЛАВАЮЩАЯ ЗАПЯТАЯ (с двойной точностью))		32 бит	от -2^{1024} до -2^{-1022} , 0, от 2^{-1022} до 2^{1024}
String (Строка)	Символьная строка. Соответствует операнду «D»	Переменная	До 255 символов
Timer (Таймер)	Включается при достижении заданного времени. Соответствует операнду таймера «T»	-	-
Retentive timer (Фиксирующий таймер)	Включается при достижении заданного времени. Соответствует операнду фиксирующего таймера «ST»	-	-
Counter (Счетчик)	Включается при достижении заданного счета. Соответствует операнду счетчика «C»	-	-
Pointer (Указатель)	Начальная позиция подпрограммы. Соответствует операнду указателя «P»	-	-

2.1.3

Регистрация меток



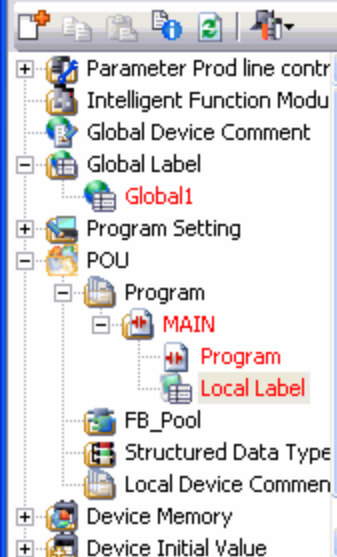
MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [Local Label Setting MAIN [PRG]]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project



Project

User Library

Connection Destination

	Class	Label Name	Data Type	Constant
1	VAR	Production_qty_atained	Bit	...
2	VAR	Parts_input_start_flag	Bit	...
3	VAR	Start_reject_pct_calc	Bit	...
4				...
5				...
6				...
7				...
8				...
9				...
10				...
11				...
12				...
13				...
14				...
15				...
16				...
17				...
18				...
19				...
20				...
21				...
22				...
23				...
24				...
25				...
26				...

Завершение ввода настроек для регистрации меток.

Для продолжения нажмите кнопку .

English

Simple

Q03UDE

Host Station

Line N...



2.1.4

Автоматическое присваивание меток операндам

Производи-
тельность

Качество

Во время преобразования программы происходит автоматическое присваивание соответствующих операндов меткам в соответствии с их классами и типами данных.

При использовании меток не обязательно знать, какие операнды им присвоены.

Используйте функцию **«Device/Label Automatic-Assign Setting»** (Настройка автоматического присваивания операндов/меток), чтобы изменить диапазон присвоенных метке операндов.

* На этом экране отображается окно настройки автоматического присваивания операндов для ПЛК серии MELSEC-Q и MELSEC-L. При использовании ПЛК серии MELSEC-F это окно может иметь другой вид.

<Запуск функции «Device/Label Automatic-Assign Setting» (Настройка автоматического присваивания операндов/меток)>

В меню GX Works2 выберите [Tool] (Инструмент) — [Device/Label Automatic-Assign Setting] (Настройки автоматического присваивания операндов/меток).

Device/Label Automatic-Assign Setting

Set a device range to automatically assign to labels.

Labels will be assigned from its way down the displayed device list when multiple devices are selected.

	Device	Digit	Assign Selection	Assignment Range		Total Points	PLC Parameter Device Setting Range	
				Start	End			
Word Device	D	10	<input checked="" type="checkbox"/>	6144	12287	6144	0 -- 12287	
	W	16	<input type="checkbox"/>				0 -- 1FFF	
	R	10	<input type="checkbox"/>					
VAR_RETAIN Range Latch(1)	D Latch	10	<input type="checkbox"/>			0		
	W Latch	16	<input type="checkbox"/>					
	ZR Latch	10	<input type="checkbox"/>					
Bit Device	M	10	<input checked="" type="checkbox"/>	4096	8191	4096	0 -- 8191	
	B	16	<input type="checkbox"/>				0 -- 1FFF	
	VAR_RETAIN Range Latch(1)	L Latch	10	<input type="checkbox"/>			0	
B Latch		16	<input type="checkbox"/>					
Pointer	P	10	<input checked="" type="checkbox"/>	2048	4095	2048	2048 -- 4095	
Timer	T	10	<input checked="" type="checkbox"/>	64	2047	1984	0 -- 2047	
	VAR_RETAIN Range Latch(1)	T Latch	10	<input type="checkbox"/>			0	
Retentive Timer	ST	10	<input type="checkbox"/>			0		
	VAR_RETAIN Range Latch(1)	ST Latch	10	<input type="checkbox"/>			0	
Counter	C	10	<input checked="" type="checkbox"/>	512	1023	512	0 -- 1023	
	VAR_RETAIN Range Latch(1)	C Latch	10	<input type="checkbox"/>			0	

Latch(1) : Able to clear the value by using a latch clear.
Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program.

(Caution)
1. Label-nonassigned devices, of the automatically assigned ones while compiling, will be allotted the device that displayed at the lowest of the selected ones. Ex:) Device will be assigned to ZR when D and ZR are selected.
2. Changing the assignment target device may also change the processing speed since the arithmetic processing speed for R and ZR is difference from other devices.

OK Cancel

Robot control - [[PRG]Write MAIN 194 Step]

Tool Window Help

- IC Memory Card
- Check Parameter...
- Options...
- Key Customize...
- Device/Label Automatic-Assign Setting...**
- Block Password...
- Confirm Memory Size...
- Set TEL Data/Connect via Modem
- LCPU Logging Configuration Tool...
- Ethernet Adapter Module Configuration Tool...
- Built-in I/O Module Tool
- Check Intelligent Function Module Parameter
- Intelligent Function Module Tool
- Language Selection...

2.2 Преобразование неоднократно используемых блоков программ на языке релейной логики в функциональные блоки

В больших программах на языке релейной логики можно использовать некоторые блоки неоднократно.

Некоторые из этих же блоков также можно использовать в других программах.

Нельзя повысить эффективность работы, если приходится вводить один и тот же блок программы на языке релейной логики каждый раз, когда это необходимо.

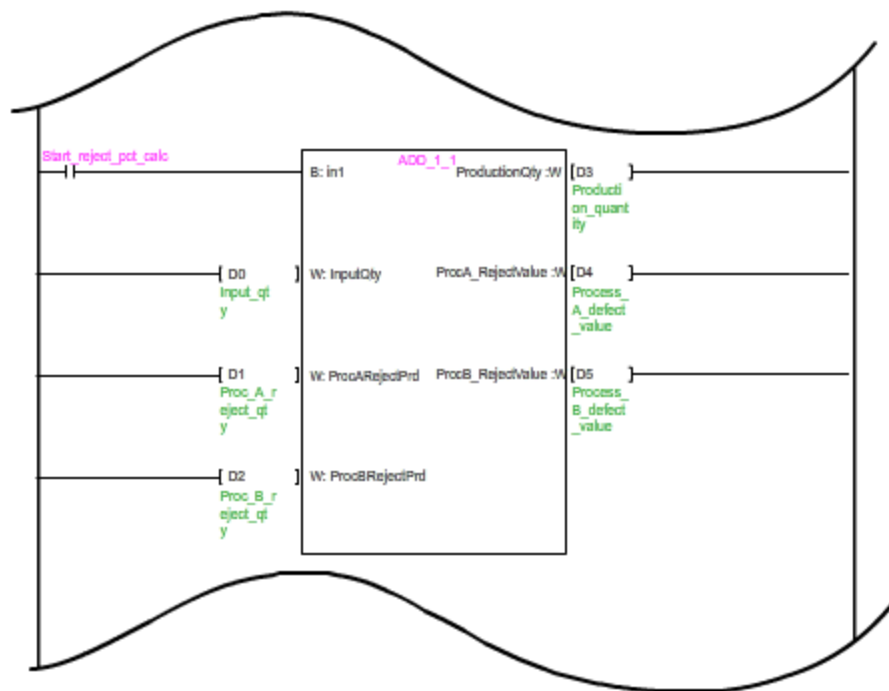
Кроме того, в случае обнаружения ошибки в общем блоке программы на языке релейной логики необходимо найти и исправить каждый такой блок. Это приводит к значительным тратам времени.

Для решения этих проблем используйте функцию **«Функциональный блок (ФБ)»**.

Неоднократно используемый блок программы на языке релейной логики можно **преобразовать в функциональный блок**, который можно использовать во всех программах. Это значительно повышает эффективность работы.

В случае обнаружения ошибки в общем блоке требуется внести изменения только в функциональный блок.

Это не только позволяет экономить время, но и исключает ситуации, когда программист не вносит изменения в один из блоков.



Нажмите эту кнопку, чтобы повторить анимацию.



2.2.1

Создание и размещение функциональных блоков

Производи-
тельность

Качество

Поскольку функциональные блоки могут использоваться в различных программах, в них нельзя использовать реальные операнды (например, X, Y и D). Эти операнды нужно заменить на **метки**. Как это сделать, вы уже узнали в главе 2.1.

Процедура преобразования блоков программ на языке релейной логики в функциональные блоки показана на анимационной схеме.

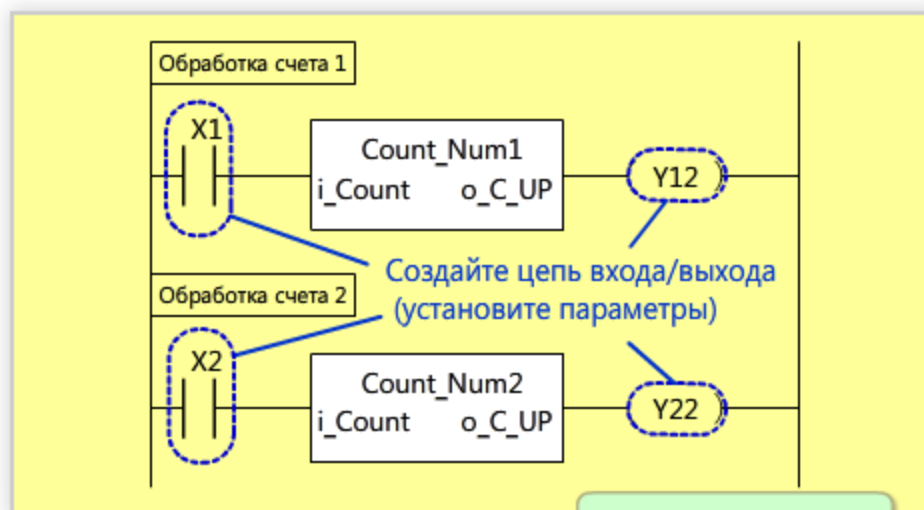
1. Подготовьте программу для создания функционального блока.

2. Чтобы преобразовать программу в функциональный блок, разделите ее на вход и выход и замените внутренние операнды на внутренние метки.

3. Перетащите созданный функциональный блок в программу.

4. Назначьте параметры для меток входа и выхода перенесенной в программу функционального блока.

5. Создайте входную цепь, передающую параметр в метку входа, и выходную цепь, получающую параметр из метки выхода, перед функциональным блоком и после него.



Нажмите эту кнопку, чтобы вернуться к началу.



2.2.1

Создание и размещение функциональных блоков

Производи-
тельность

Качество

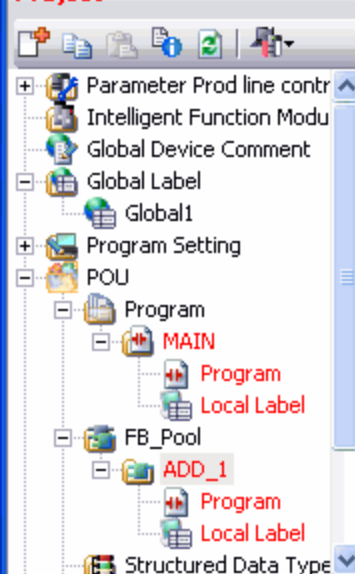
MELSOFT Series GX Works2 C:\Sequential Programs\e-learning\Robot control - [[PRG]Write MAIN (188)Step *]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Write MAIN (188)Ste... Global Label Setting Global1 Local Label Setting MAIN [PRG] Function/FB Lat

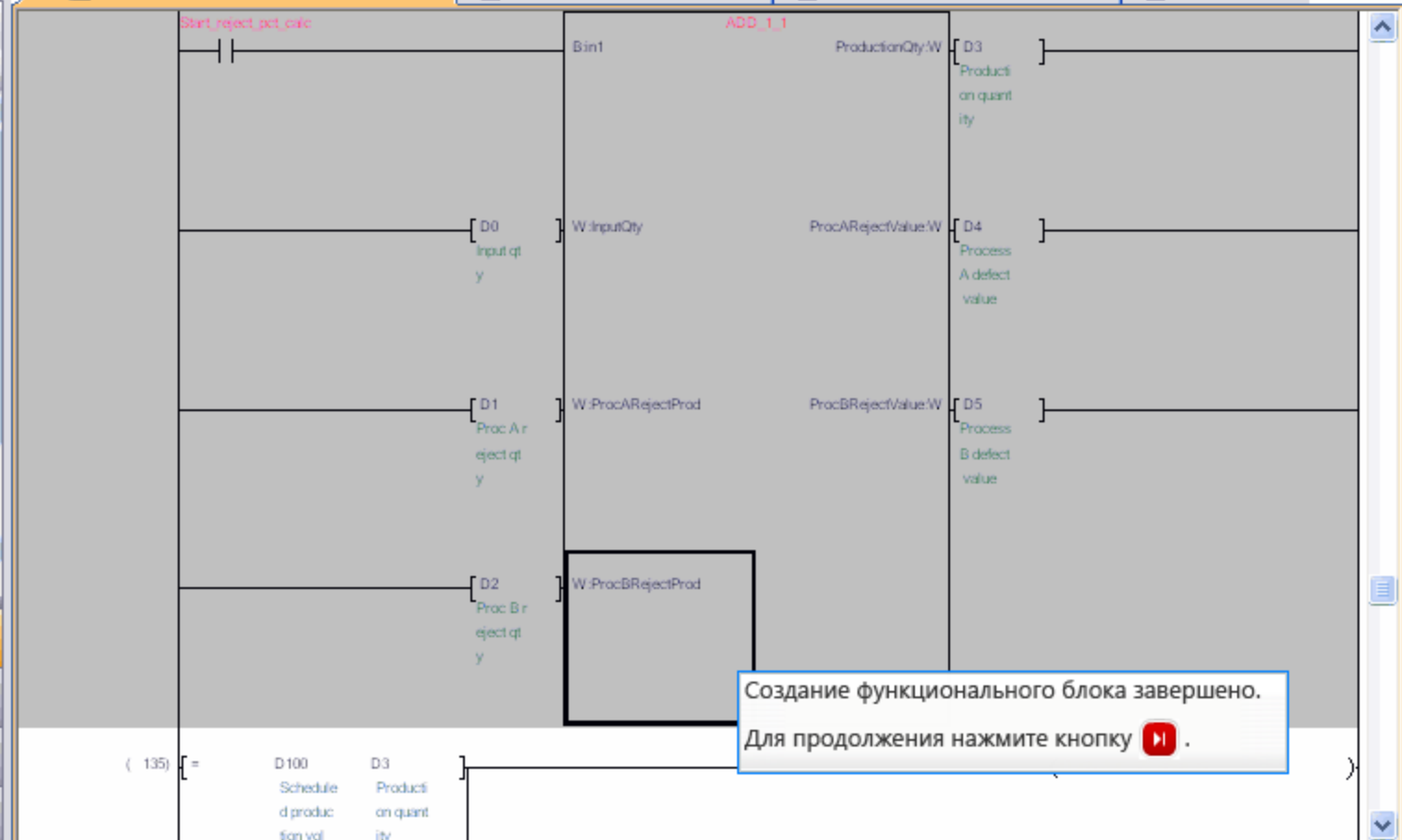
Project



Project

User Library

Connection Destination



Создание функционального блока завершено.

Для продолжения нажмите кнопку .

English

Simple

Q03UDE

Host Station

(13 ML)



2.2.2

Использование библиотеки функциональных блоков

Производи-
тельность

Качество

Вы уже научились создавать функциональные блоки.

Инженеры компании Mitsubishi Electric создали программы управления для различных модулей, таких как ЦП, аналоговые модули ввода/вывода, сети и модули позиционирования, в функциональные блоки. Все они доступны бесплатно в **«Библиотеке ФБ»**.

Использование библиотеки ФБ устраняет необходимость в разработке программ управления модулями, что обычно приходилось делать пользователям. Вы можете использовать даже неизвестные вам модули.

Библиотеку ФБ можно загрузить на странице **«Загрузка библиотеки MELSOFT» веб-сайта MITSUBISHI ELECTRIC FA**.

<Раньше>



<С использованием библиотеки ФБ>



* Для ПЛК серии MELSEC-F библиотека ФБ отсутствует.

На странице загрузки перечислены поддерживаемые модели. Перед загрузкой библиотеки убедитесь, что ваша модель поддерживается.

2.3 Изменение начальных значений операндов без исправления программы

Производи-
тельность

Качество

Начальное значение или константа операнда обычно задается с помощью команды MOV перед обработкой основной программы.

В этом случае программу нужно непосредственно исправлять каждый раз, когда ее работа изменяется в зависимости от применения системы.

При этом пользователь не только тратит много времени, но и может ошибиться при внесении исправлений или не суметь внести нужные изменения в программу.

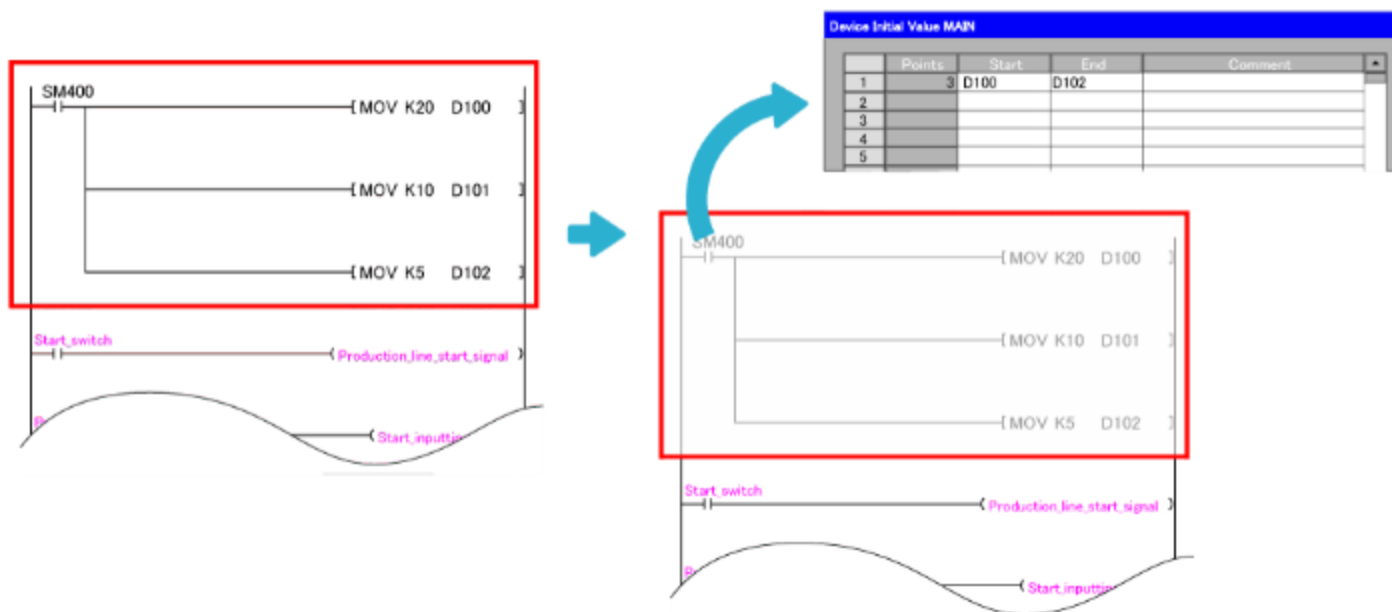
Для решения этих проблем используйте функцию **«Начальное значение операнда»**.

С помощью этой функции GX Works2 обеспечивает надлежащее управление начальными значениями операндов и устраняет необходимость внесения в программу исправлений, что позволяет существенно повысить эффективность создания программ.

Кроме того, эта функция устраняет необходимость в создании программы для присвоения операндам начальных значений, что позволяет уменьшить объем программы (использование памяти) и вероятность сбоев.

При использовании функции «Начальное значение операнда» нужно указать диапазон операндов для присвоения начальных значений.

Фактические начальные значения хранятся в **памяти операндов** и присваиваются заданным диапазонам операндов. Благодаря тому, что области памяти операндов подготовлены в соответствии с применениями системы, начальные значения операндов можно изменять, просто изменив присваиваемую область памяти операндов.



2.3 Изменение начальных значений операндов без исправления программы



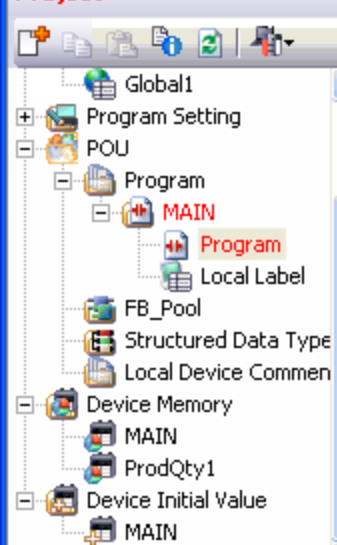
MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN (166)Step *]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Write MAIN (166)Ste... Global Label Setting Global1 Local Label Setting MAIN [PRG]

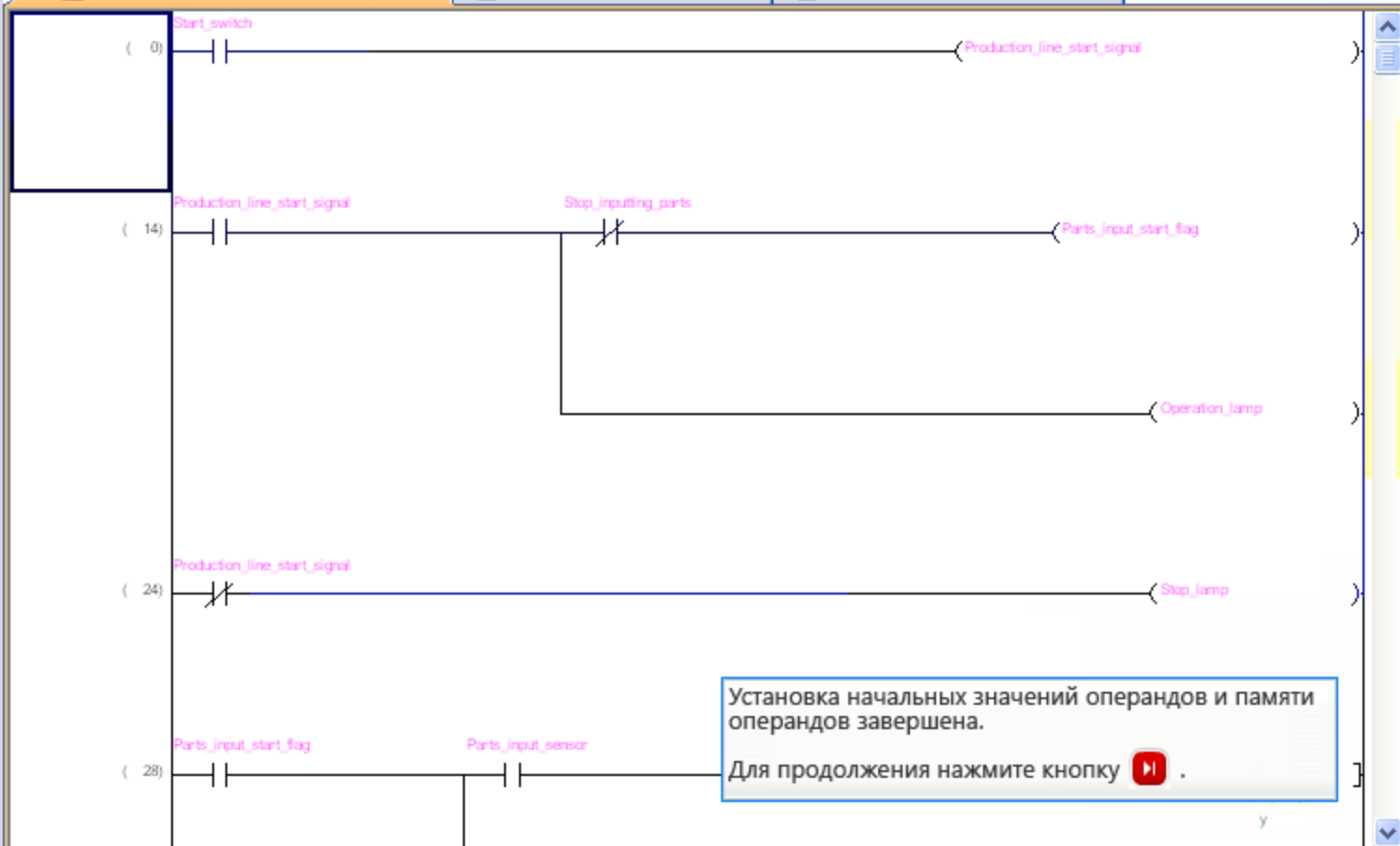
Project



Project

User Library

Connection Destination



English

Simple

Q03UDE

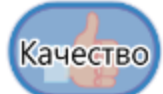
Host Station

(0/ N/5



2.4

Упрощение программ на языке релейной логики



В больших программах на языке релейной логики связь между операндами, командами и линиями может быть весьма сложной, из-за чего бывает тяжело понять, какие действия выполняет программа. В частности, численные расчеты, например формулы, выполненные на одной строке, необходимо программировать, используя сочетания команд.

Для решения этих проблем используйте функцию **«Встроенный структурированный текст»**.

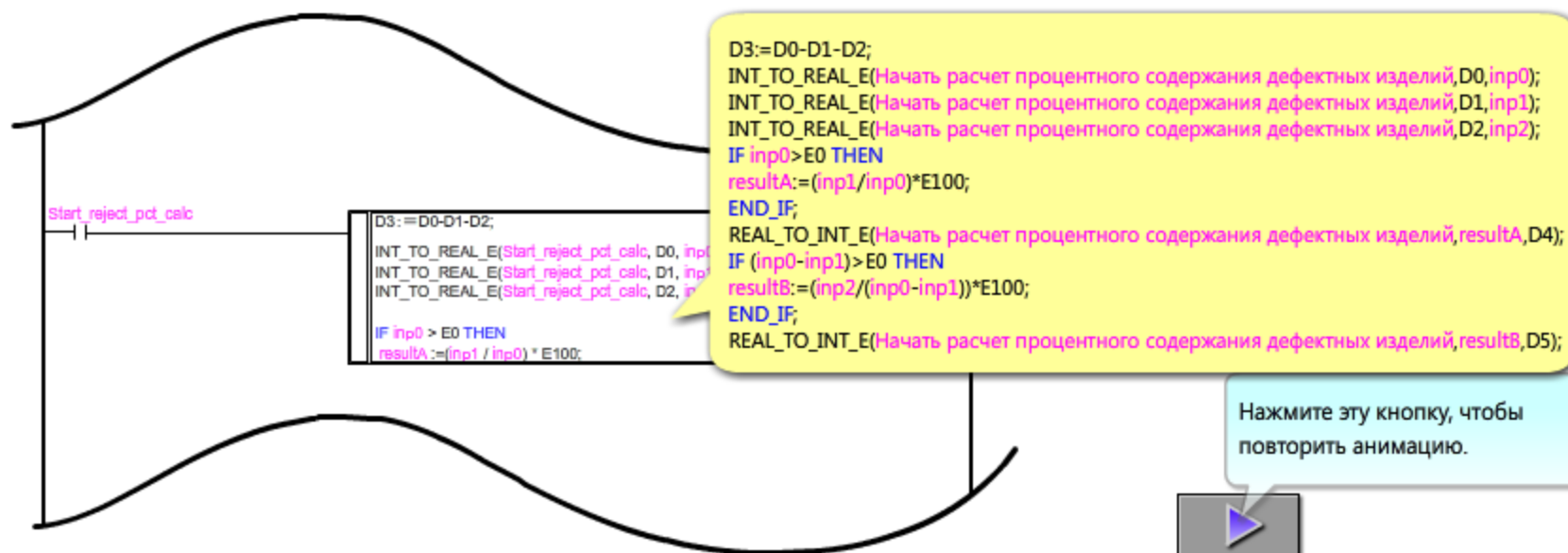
Программу на языке релейной логики можно заменить на программу, частично написанную на **языке структурированного текста (ST)**.

ST — это язык программирования, в котором используются последовательности управления, похожие на язык C для программирования ПО для компьютеров.

Численные расчеты могут быть написаны с использованием формул, поэтому даже программисты, не знакомые с языком C, могут использовать встроенный структурированный текст.

Ниже показан пример, в котором часть программы на языке релейной логики для системы заменена встроенным структурированным текстом.

Как видите, сложная программа стала понятной.



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 114 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Global1
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Type
 - Local Device Commen
- Device Memory
- Device Initial Value

Project

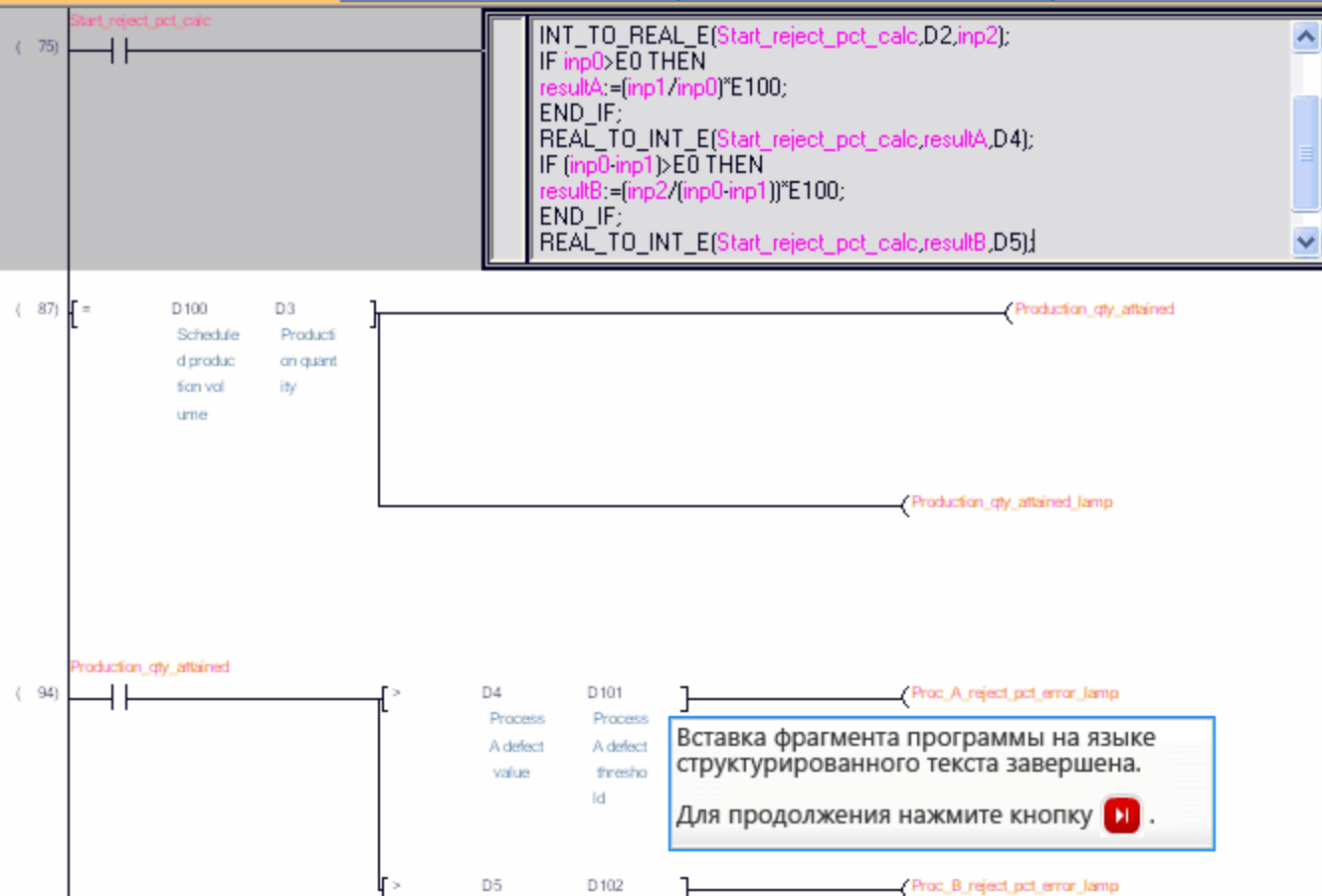
User Library

Connection Destination

[PRG]Write MAIN 114 Step

Global Label Setting Global1

Local Label Setting MAIN [PRG]



Вставка фрагмента программы на языке структурированного текста завершена.
Для продолжения нажмите кнопку .

English

Simple

Q03UDE

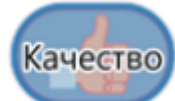
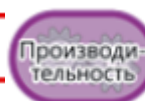
Host Station

(83 N...



2.5

Создание программ, простых для понимания и чтения



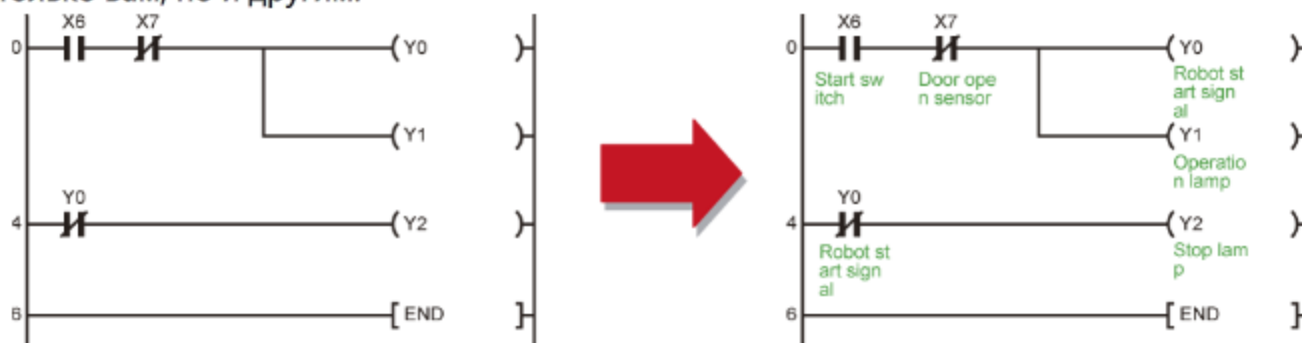
Понять детали управления в большой программе, просто взглянув на нее, бывает сложно. При этом могут возникнуть следующие проблемы.

- Вы делаете ошибки в программе (например, ввод неверных команд или операндов).
- Вы не можете найти причины ошибок программирования.
- Специалист, работающий над программой после вас, не может понять принципы управления.

Для решения этих проблем используйте функцию **«Комментарий»**.

Для облегчения понимания принципов управления к программе можно прикреплять заметки, содержащие информацию об управлении и названия операндов.

Эти комментарии необходимо вводить при любой возможности, что позволит создавать программы, понятные не только вам, но и другим.



В GX Works2 можно вводить следующие комментарии.

Тип комментария	Диапазон комментария
Device comment (Комментарий к операнду)	Присваивается операнду. Такой комментарий описывает применение каждого операнда и тип подключенного устройства ввода/вывода.
Statement (Пояснение)	Присваивается блоку программы на языке релейной логики. Такой комментарий позволяет сделать поток обработки данных доступным для понимания.
Note (Примечание)	Присваивается катушкам/инструкциям в программе. Такой комментарий облегчает понимание назначения выходной катушки или инструкции.

2.5.1

Запись комментариев для каждого блока программы на языке релейной логики

Производи-
тельность

Качество

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 145 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

[PRG]Write MAIN 145 Step

Scheduled prod qty setting

(0) SM400 Always ON N

[MOV K20 D100 Schedule d produc tion vol ume]

[MOV K10 D101 Process A defect thresho ld]

[MOV K5 D102 Process B defect thresho ld]

Prod line start

(7) X0 Start sw itch


(Y10 Product on line start si gnal)

(9) Y10 Y16

(M0)

English Simple Q03UDE Host Station (7/ M...

Оператор строки вставлен в начало блока программы на языке релейной логики.

Настройка оператора строки завершена.
Для продолжения нажмите кнопку .



2.5.2

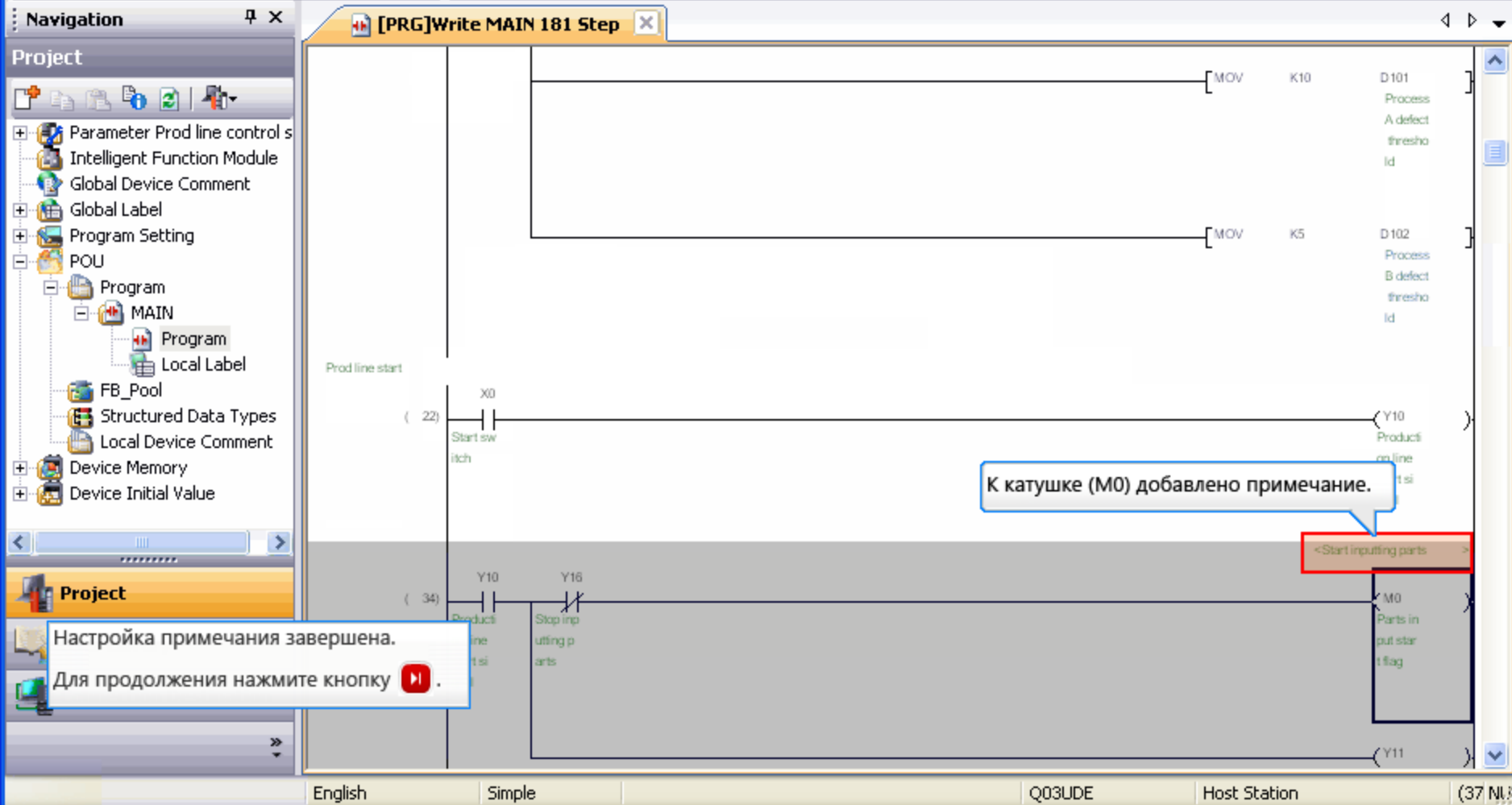
Запись комментариев для каждого выхода (катушки, инструкции)

Производительность

Качество

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 181 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



2.6 Упрощение для чтения программ со специальными реле/регистрами

Если в программе используются специальные реле, специальные регистры и (или) операнды специальных функциональных модулей, то все применения и функции этих устройств может быть сложно понять. Вам придется читать программу, держа в руках руководство.

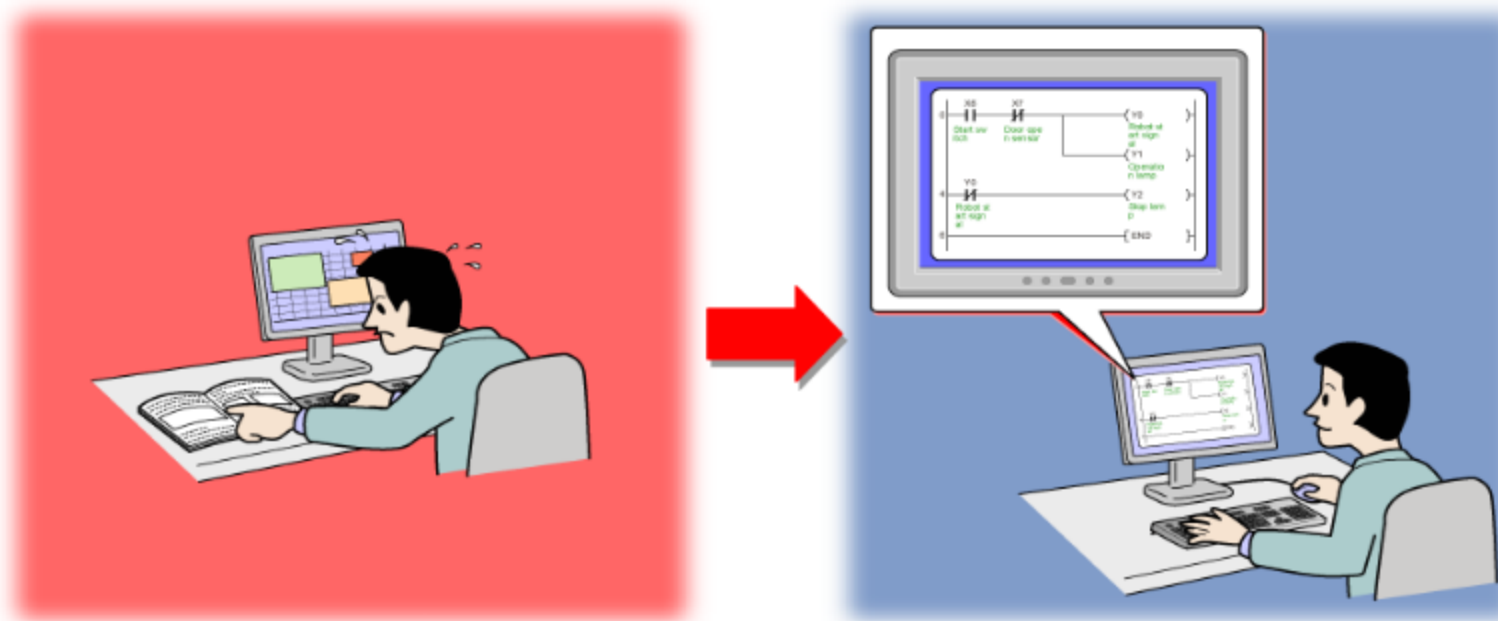
Хотя программу будет легче читать, если присвоить комментарий каждому операнду, в случае использования большого количества операндов на это уйдет много времени и усилий.

Для решения этих проблем используйте функцию **«Шаблонный комментарий»**.

В ПО GX Works2 предусмотрены шаблонные комментарии, описывающие применение и функции специальных реле, специальных регистров и операндов специальных функциональных модулей.

С помощью этих шаблонных комментариев присваивать комментарии операндам для облегчения понимания программы гораздо удобнее.

При необходимости шаблонные комментарии можно изменять.



2.6 Упрощение для чтения программ со специальными реле/регистрами

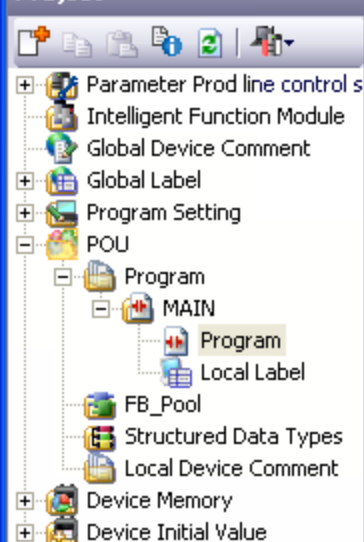
MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

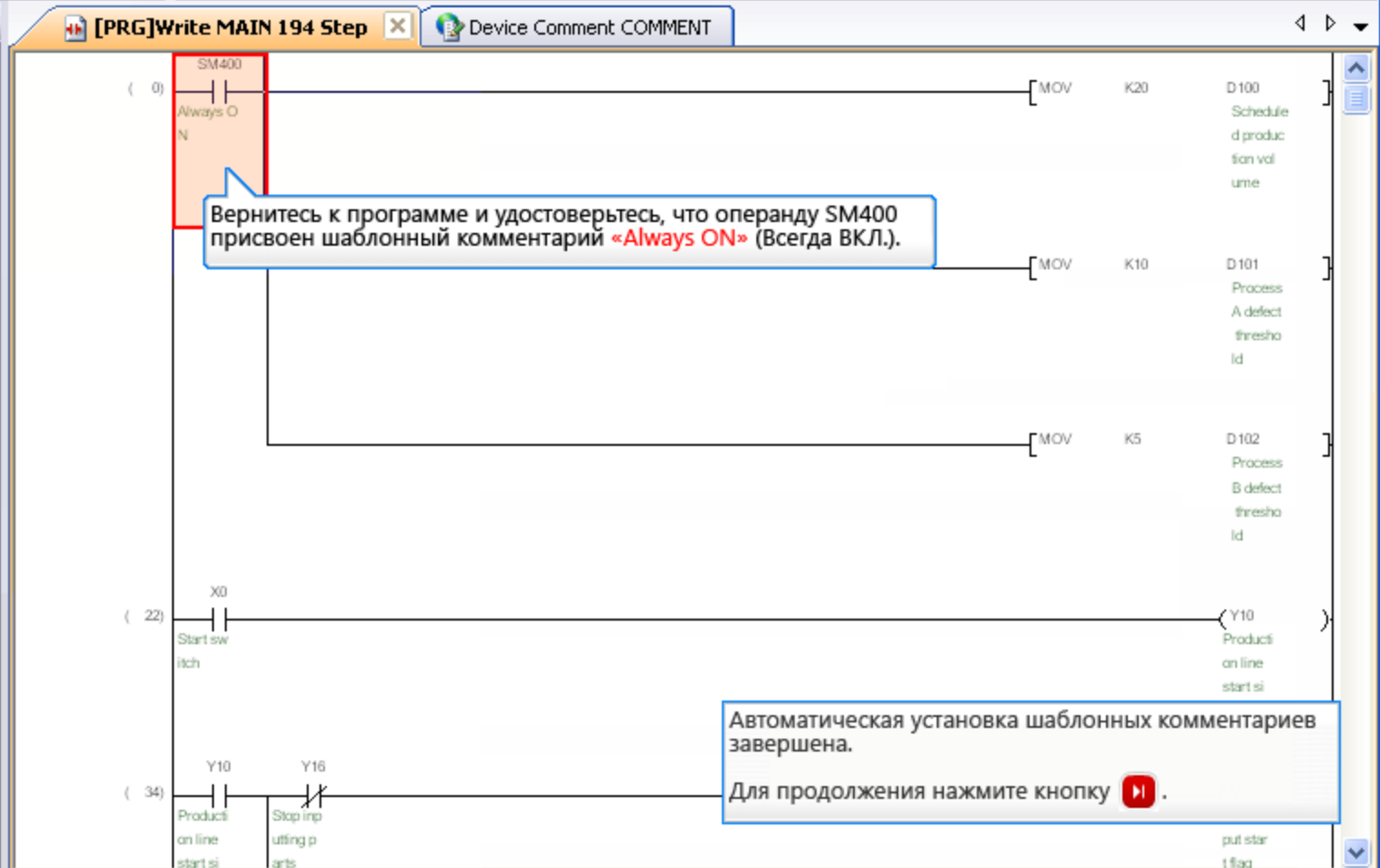
Project



Project

User Library

Connection Destination



English

Simple

Q03UDE

Host Station

(15 N)



Глава 3 Отладка

Этапы обучения, предусмотренные в главе 3

В главе 3 вы изучите функции, используемые для отладки.

В ПО GX Works2 предусмотрено множество различных инструментов для мониторинга, отладки и исправления ошибок. Создавайте качественные программы без ошибок, используя эти инструменты мониторинга и отладки.

- 3.1. Отслеживание только целевых операндов и меток
- 3.2. Проверка состояния использования операндов и меток
- 3.3. Сбор информации об изменении значений операндов с течением времени
- 3.4. Изменение значений операндов без исправления программы
- 3.5. Пошаговая отладка программы
 - 3.5.1. Использование функции отработки шага
- 3.6. Имитация работы внешнего устройства
 - 3.6.1. Ввод значений операндов для настройки системы ввода/вывода
 - 3.6.2. Использование формата временной диаграммы для настройки системы ввода/вывода

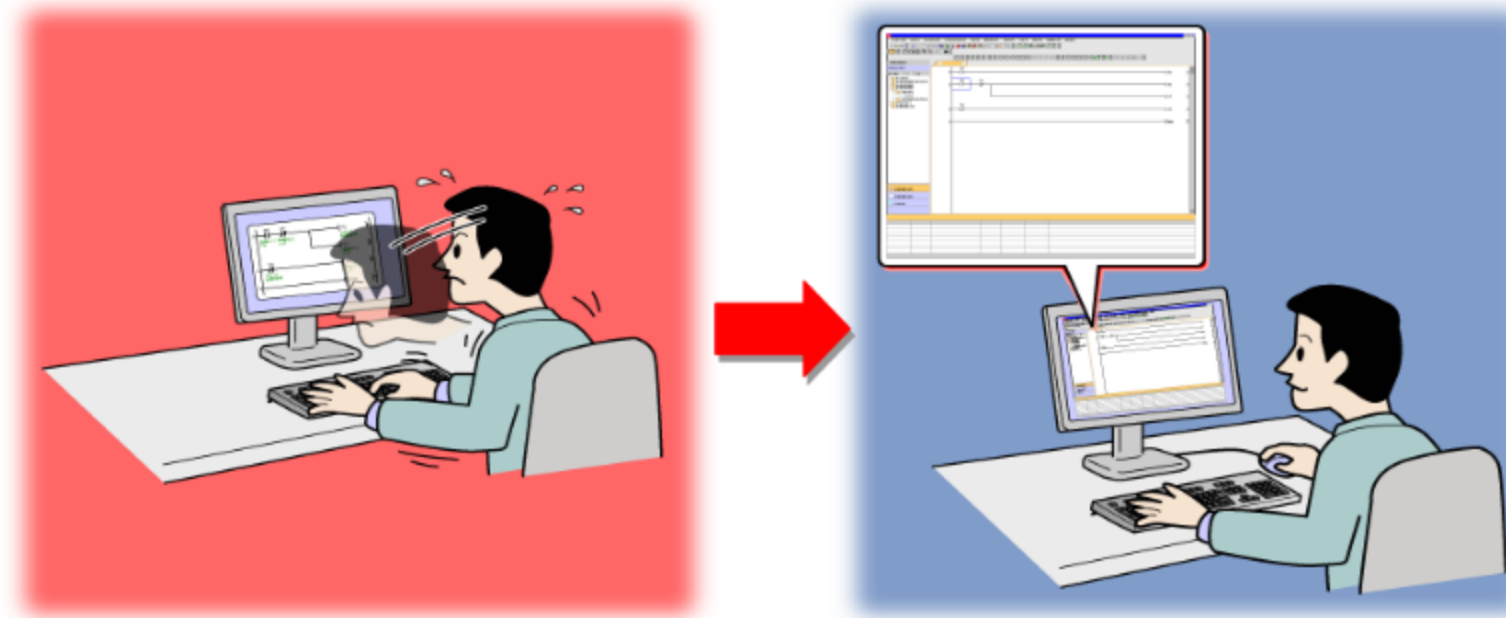
3.1

Отслеживание только целевых операндов и меток

Качество

В программе используется множество команд и операндов. Кроме того на экране компьютера может одновременно отображаться только ее часть. Таким образом, только мониторинг не может контролировать работу всей программы.

Для решения этих вопросов используйте функцию **«Наблюдение»**. Эта функция может использоваться для мониторинга только операндов и меток, которые вы указали заранее. Для обеспечения мониторинга всех диапазонов можно создать два и более окон наблюдения.



3.1

Отслеживание только целевых операндов и меток


 Качество

MELSOFT Series GX Works2 C:\Sequential Programs\e-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Monitor Executing M...

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
- Program
 - MAIN
 - Program

Prod qty calc

(74) M1

Start re
ject pct
calc

D1 D2 D6

GX Simulator 2

Tool Options

Switch

RESET RUN

LED

MODE

RUN

ERR.


USER

Происходит отслеживание предустановленных значений операндов, зарегистрированных в Watch 1.

Watch 1

Device/Label	Current Value	Data Type	Class	Device	Comment
X0	0	Bit		X0	Start switch
X1	0	Bit		X1	Parts input sensor
X2	0	Bit		X2	
X3	0	Bit		X3	

Регистрация и отслеживание с помощью функции Watch завершены.

Для продолжения нажмите кнопку .

Watch 1

Watch 2

English

Simple

Q03UDE

Simulation

(84 N/3



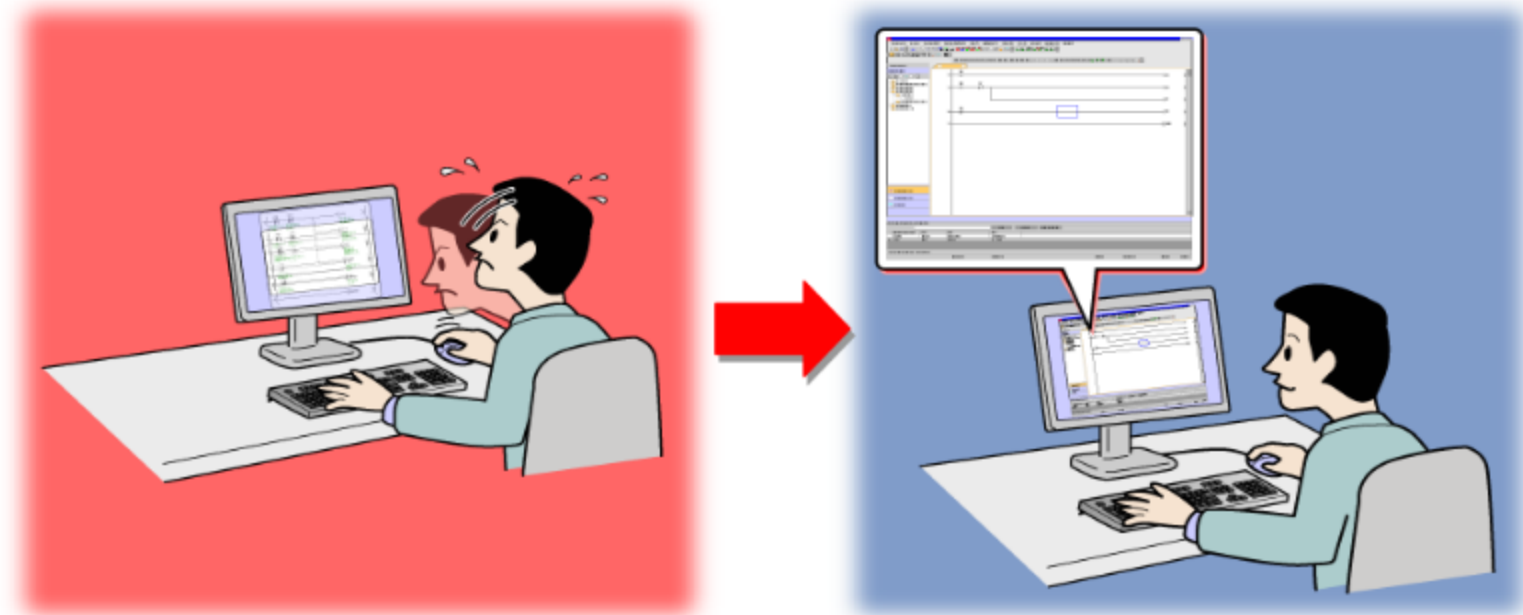
3.2 Проверка состояния использования операндов и меток

Качество

В разных местах программы используются одинаковые операнды и метки. Вы можете проверить состояние использования этих операндов и меток, сравнивая их в разных расположениях.

Используйте для этого функцию **«Перекрестные ссылки»**.

Эта функция выводит перечень расположений операндов и меток, соответствующих условиям поиска, позволяя вам сравнить их и убедиться, что они используются правильно.



3.2 Проверка состояния использования операндов и меток


 Качество

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN

Project

Cross Reference

Cross Reference Information

Condition Setting

Device/Label D1


Find

Print...

Print Preview...

Device/Label	Device	Instruction	Ladder Symbol	Position	Data Name
Filtering Condition	Filtering Con...	Filtering Condit...		Filtering Condition	
D1	D1	+P	-[]-	Step No.63	
D1	D1	+	-[]-	Step No.84	
D1	D1	FLT	-[]-	Step No.92	

Использование функции Cross Reference (Перекрестные ссылки) завершено.

 Для продолжения нажмите кнопку .

S: device/cross reference information of label "D1"

Analyze and display current program after pressing Find.

English

Simple


Q03UDE

Host Station

(84 M...

Курсор программы переместился в позицию шага № 74.

Использование функции Cross Reference (Перекрестные ссылки) завершено.

 Для продолжения нажмите кнопку .

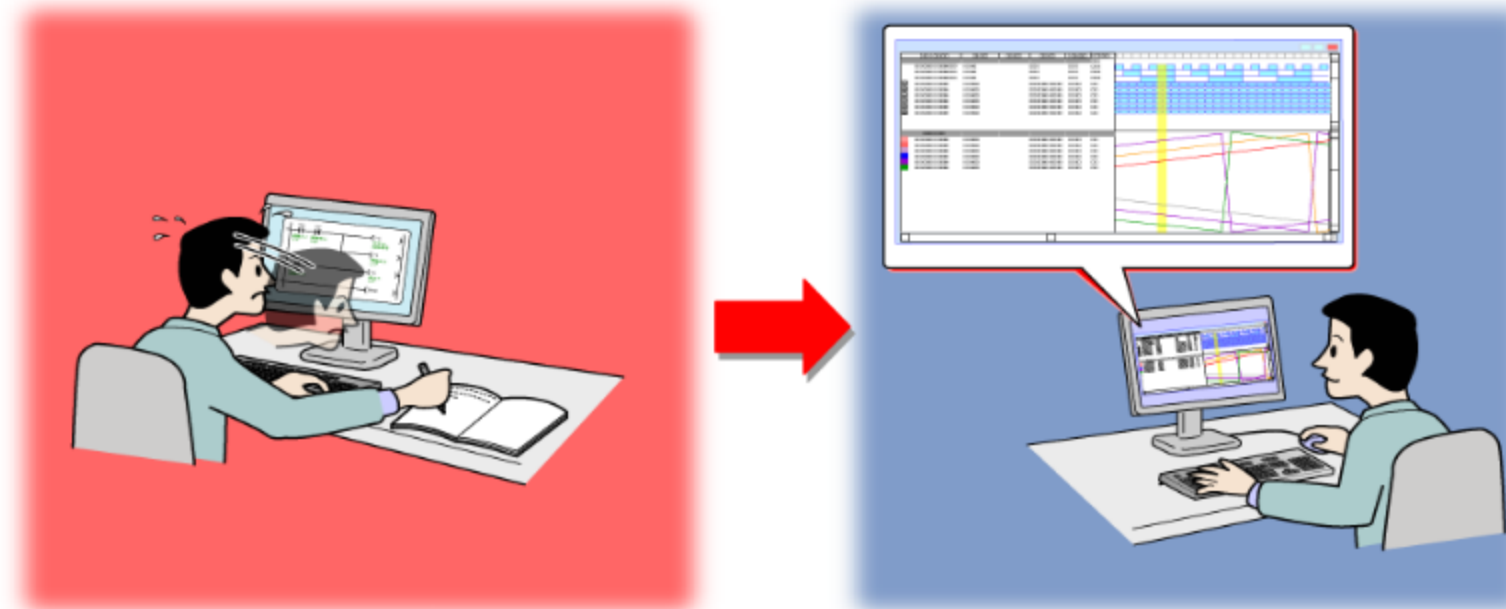
3.3

Сбор информации об изменении значений операндов с течением времени

Качество

Вы можете захотеть убедиться, что изменения в операндах и метках находятся в пределах допустимого диапазона, или выполнить устранение неисправностей в случае сбоя, просмотрев изменения в программе.

В этом случае можно использовать функцию **«Трассировка»**. (Только для ПЛК серии MELSEC-Q и MELSEC-L)
Эту функцию можно использовать для мониторинга и записи изменений в значениях операндов и меток с течением времени. Эти изменения можно сохранить в виде файла.



3.4

Изменение значений операндов без исправления программы

Качество

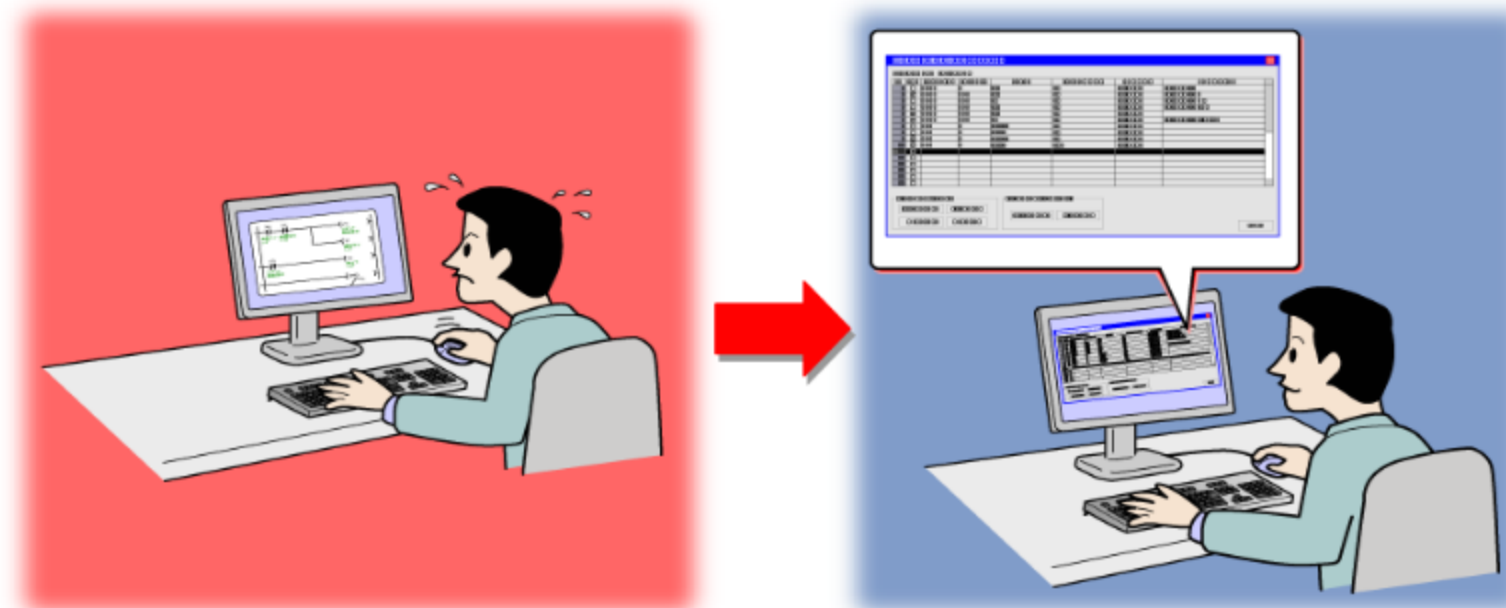
При выполнении отладки можно принудительно изменить значения операндов, чтобы проверить различия в работе программы.

Однако это означает, что вам придется изменять программу каждый раз, когда изменяется значение операнда, что отнимает много времени и усилий.

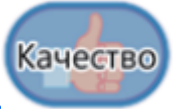
Кроме того, если вы забыли восстановить исходные параметры в измененной программе, это может стать дополнительной причиной сбоев.

В таком случае используйте функцию **«Проверка операндов с условием выполнения»**. (Только для ПЛК серии MELSEC-Q и MELSEC-L)

Эта функция изменяет значения операндов при выполнении заданного заранее шага без изменения программы.



3.4 Изменение значений операндов без исправления программы



MELSOFT Series GX Works2 C:\Sequential Programs\e-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step

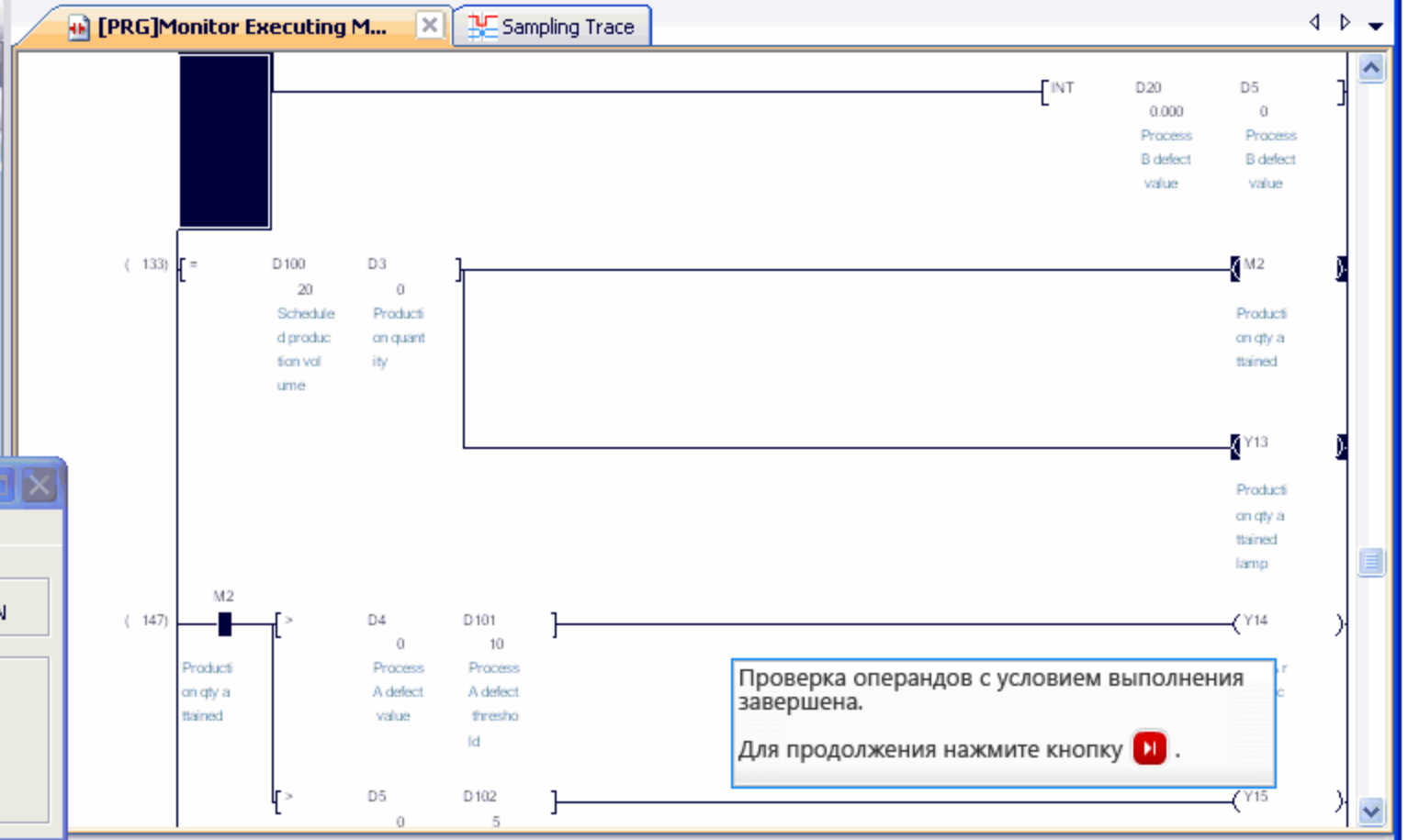
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Toolbar with various icons for file operations, editing, and simulation control.

Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool



GX Simulator 2

Tool Options

Switch

RESET STOP RUN

LED

MODE

RUN

ERR.

USER

Проверка операндов с условием выполнения завершена.

Для продолжения нажмите кнопку .

Simulation control icons: stop, play, back, forward, volume

3.5 Пошаговая отладка программы

Качество

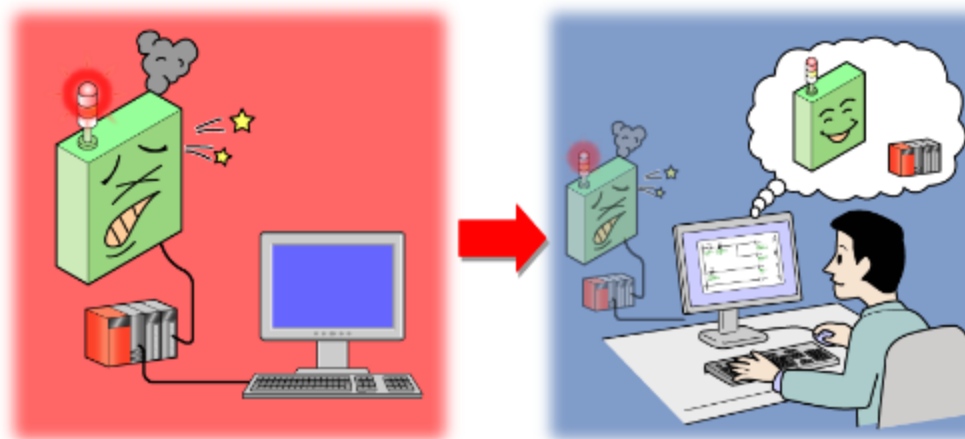
Во время выполнения отладки у вас может появиться необходимость в подтверждении выполнения команд в каждом шаге или проверке наличия изменений в значении операнда.

Однако пошаговая отладка может быть затруднена из-за высокой скорости выполнения программы (время цикла).

Для решения этой проблемы используйте функцию **«Отработка шага»**. (Только для ПЛК серии MELSEC-Q и MELSEC-L) Эта функция позволяет выполнять программу пошагово, благодаря чему можно осуществлять пошаговую отладку. Используйте эту функцию пошагового выполнения вместе с **функцией симуляции**. (Функцию пошагового выполнения нельзя использовать для отладки на настоящем ПЛК.)

Функция пошагового выполнения может использовать следующие функции.

Функция	Описание
Прерывание выполнения	Эта функция выполняет программу до тех пор, пока не будут выполнены условия прерывания. Выполнение программы прекращается, когда удовлетворяются условия прерывания. Используйте точки прерывания и операнд прерывания, чтобы настроить условия прерывания.
Отработка шага	Эта функция выполняет программу шаг за шагом.
Частичное исполнение	Эта функция выполняет программу только с указанного места.



3.5.1

Использование функции отработки шага

Качество

Укажите **точку прерывания** и **операнд прерывания** как место начала отладки и условие запуска, соответственно. Кроме того, можно указать **пропускаемый диапазон**, в пределах которого выполнять программу не нужно.
(Только для ПЛК серии MELSEC-Q и MELSEC-L)

Когда после запуска **прерывания выполнения** удовлетворяются условия прерывания, выполнение программы приостанавливается.

Впоследствии при пошаговом выполнении программы с использованием **функции отработки шага** проверяйте изменения в значениях операндов, чтобы обнаружить ошибку.

<Точка прерывания>

Установите точку прерывания в месте, где требуется прервать выполнение программы.

Укажите ее в единицах шагов.

В проекте можно установить до 64 точек прерывания.

<Операнд прерывания>

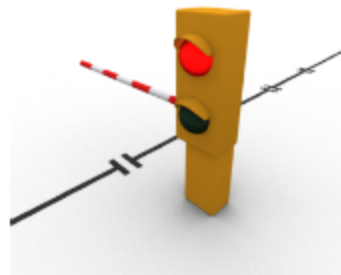
Установите операнд прерывания, на основе которого будет приостанавливаться выполнение программы, когда значение операнда или метки будет удовлетворять заданному условию.

Можно задать до 16 битовых или словных операндов.

<Пропускаемый диапазон>

Установите диапазон, в пределах которого программа не должна выполняться во время отработки шага, в единицах блоков программы на языке релейной логики.

В проекте можно задать до 16 диапазонов.



<Приложение пропускаемого диапазона>

Точку сбоя можно сузить, используя функцию **пропускаемого диапазона**.

Запустите прерывание выполнения с заданным пропускаемым диапазоном и без него.

Если сбой происходит только тогда, когда не используется функция пропускаемого диапазона, это указывает на то, что диапазон включает в себя ошибку.

3.5.1

Использование функции отработки шага



MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Read MAIN (Read Only) 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

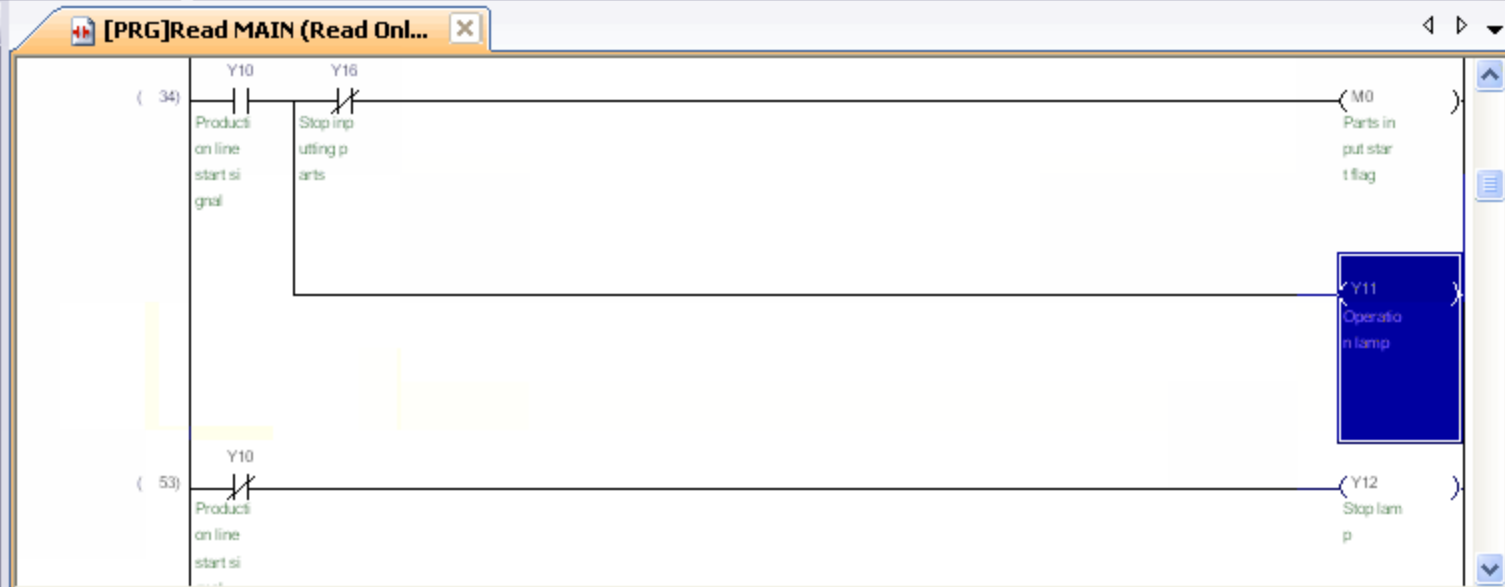


Navigation

Project

- Parameter Prod line contr
- Intelligent Function Modu
- Global Device Comment
- Global Label
- Program Setting
- POU
- Program
 - MAIN

Project



Break Device

New Condition Cancel Cancel All Combination
 Judge each break device (OR condition) Judge all break devices (AND condition)

Enable/Disable	Comparative Source (Device/Label)	Condition	Comparative Target (Value/Device/Label)	Comparative Type

Отладка завершена. Функция отработки шага завершена.
 Для продолжения нажмите кнопку .

Skip Range Break Point Break Device

English

Simple

Q03UDE

Host Station

(52 N)



3.6

Симуляция работы внешнего устройства

Качество

Отладка в сочетании с работой внешнего устройства невозможна в среде разработки, в которой нельзя использовать внешнее устройство, например при работе функции симуляции.

Чтобы решить эту проблему, обычно добавляется программа отладки, которая имитирует работу внешнего устройства.

Однако, помимо того, что создание программы имитации отнимает много времени и усилий, программу необходимо редактировать при изменении операции.

Для решения этой проблемы используйте функцию **«Настройка системы ввода/вывода»**.

Эта функция может имитировать работу внешнего устройства без использования программы отладки.

В окне настройки можно без труда установить или изменить параметры работы внешнего устройства. Таким образом, создавать или изменять обычную программу не обязательно.

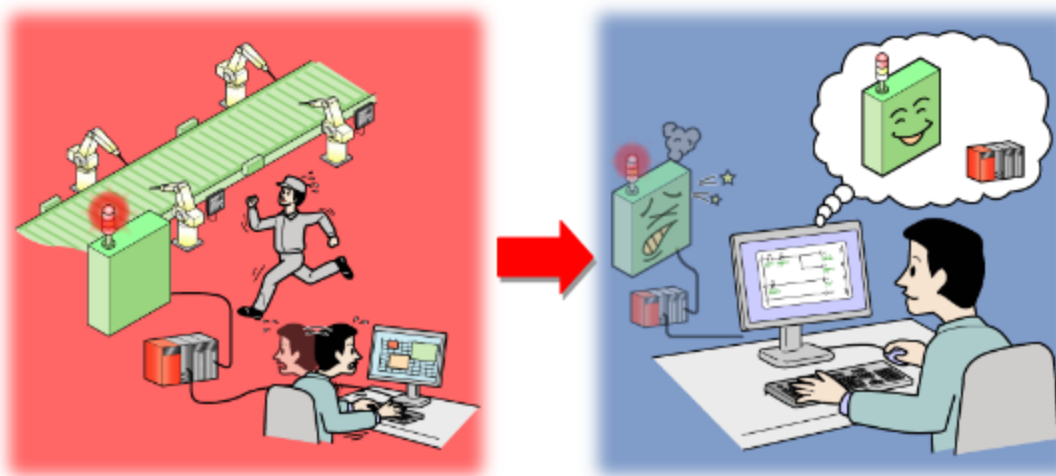
Существует два способа настройки внешнего устройства:

<Установка значений операндов>

Заданное значение операнда можно изменить во время, установленное на таймере, после удовлетворения заданных условий.

<Установка в формате временной диаграммы>

После удовлетворения заданных условий можно выполнить изменение операндов, указанных в формате временной диаграммы.



3.6.1

Ввод значений операндов для настройки системы ввода/вывода

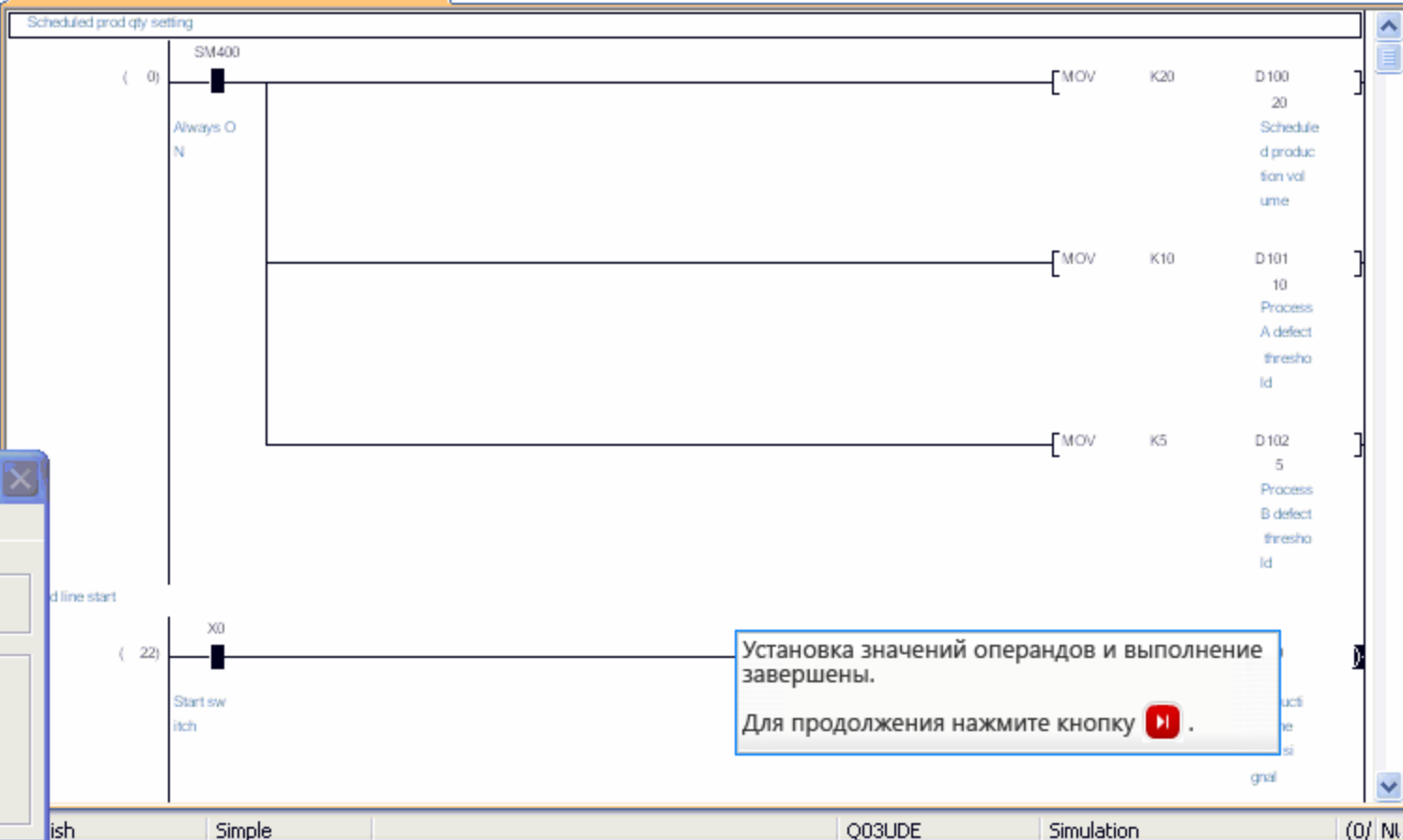
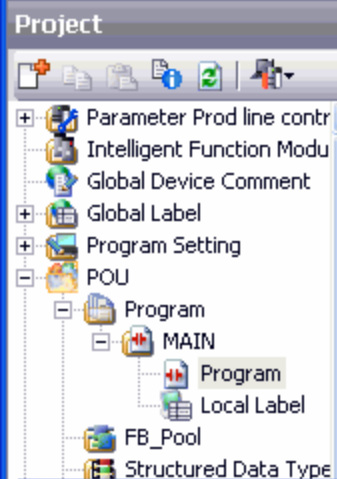

 Качество

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Monitor Executing M...



ish

Simple

Q03UDE

Simulation

(0/ N)



3.6.2

Использование формата временной диаграммы для настройки системы ввода/вывода

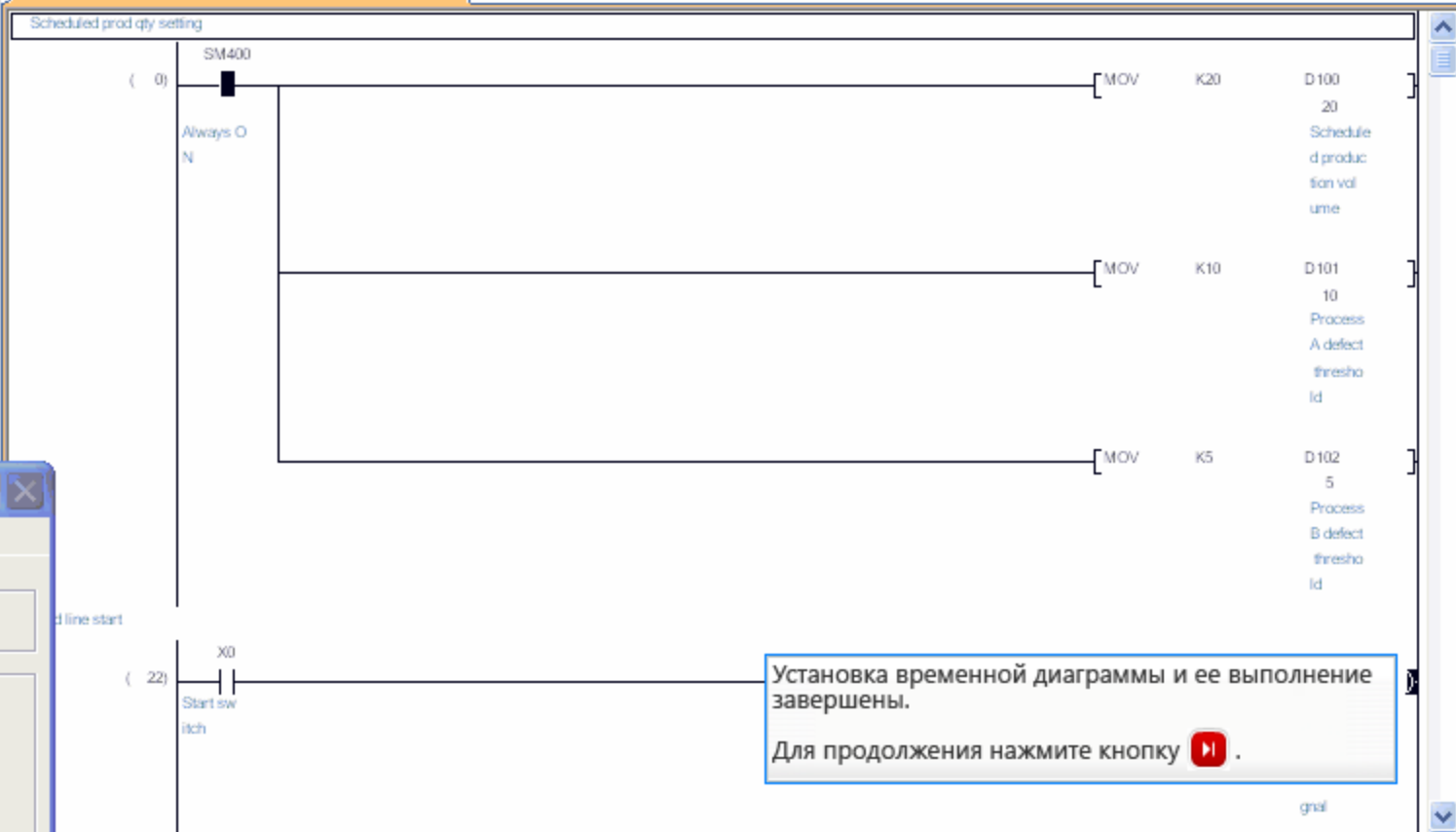
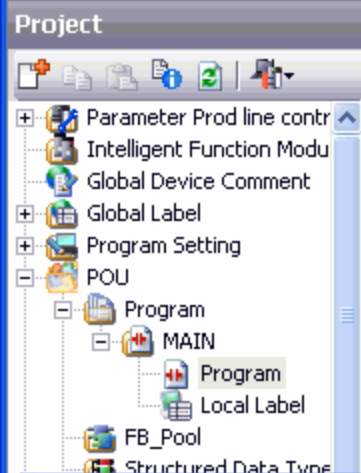

 Качество

MELSOFT Series GX Works2 C:\Sequential Programs\e-learning\Robot control - [[PRG]Monitor Executing MAIN (Read Only) 194 Step]


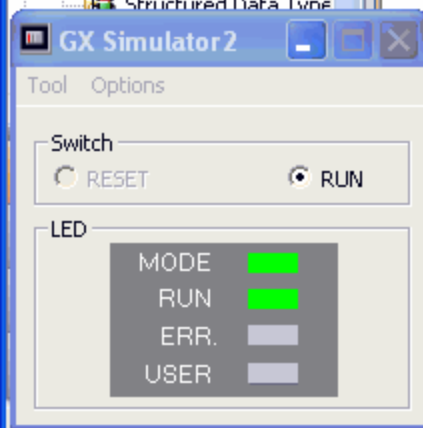
Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Monitor Executing M...



Установка временной диаграммы и ее выполнение завершены.

 Для продолжения нажмите кнопку  .


Simulation (0/ N)



Глава 4 Управление проектами и меры безопасности

Этапы обучения, предусмотренные в главе 4

В главе 4 вы изучите функции, используемые для управления проектами и осуществления мер безопасности.

- 4.1. Предотвращение утечек информации и несанкционированных изменений программ
 - 4.1.1. Ограничение доступа к данным для отдельных пользователей
- 4.2. Резервирование проектов и управление версиями
- 4.3. Сравнение программ, записанных в программируемом контроллере и персональном компьютере

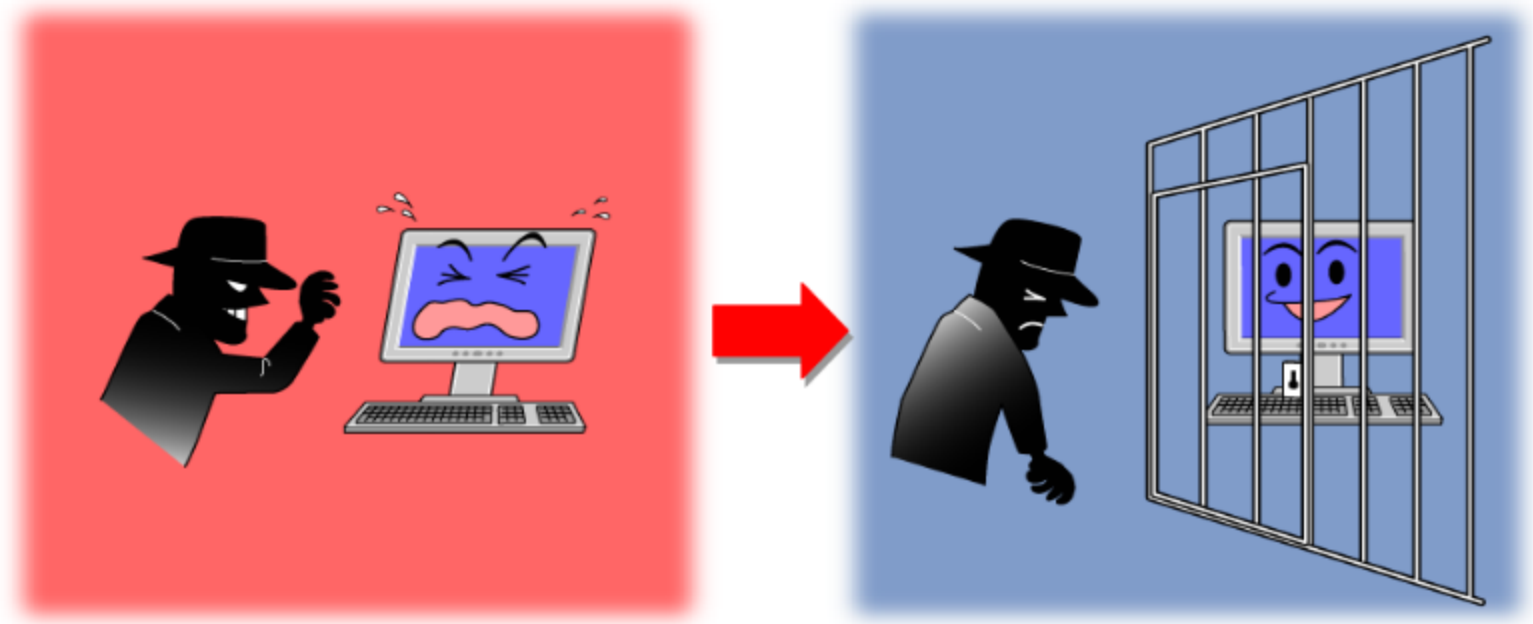
4.1

Предотвращение утечек информации и несанкционированных изменений программ



Программы ПЛК могут содержать важные данные и ноу-хау. Утечка такой информации может оказать разрушительное воздействие на бизнес. Несанкционированное изменение программы может привести к таким производственным проблемам как остановка системы.

Для решения этих проблем используйте функцию **«Защита»**. Эта функция защищает проект паролем, ограничивая пользователей, которые могут иметь к нему доступ. Кроме того, она может ограничить диапазон данных или функций, которыми может управлять тот или иной пользователь или к которым он может получить доступ. Таким образом, эта функция предотвращает просмотр и редактирование программ несанкционированными пользователями.



4.1.1

Ограничение доступа к данным для отдельных пользователей



Большие проекты ПЛК часто совместно разрабатывают несколько программистов.

В случае командной разработки проекта необходимо определить доступные данные и доступные функции в соответствии с работами и уровнем навыков каждого программиста, а также уровнем конфиденциальности данных, с которыми работает тот или иной программист.

Такое управление доступом можно осуществлять, настроив **уровни доступа**.

<Уровень доступа>

Уровень доступа для работы с данными, включенными в проект, можно задавать для каждого пользователя в индивидуальном порядке.

Можно установить следующие пять уровней доступа.

Уровень доступа		Права доступа	
Высокий Низкий	Administrators (Администраторы)	Уровень администратора	Разрешается использовать все функции.
	Developers (Level3) (Разработчики (Уровень 3))	Уровень разработчика	Ограничены параметры безопасности, доступ к данным и некоторые операции.
	Developers (Level2) (Разработчики (Уровень 2))		
	Developers (Level1) (Разработчики (Уровень 1))		
	Users (Пользователи)	Уровень оператора	Доступен только просмотр данных проекта. Чтение с ЦП ПЛК запрещено.

4.1.1

Ограничение доступа к данным для отдельных пользователей



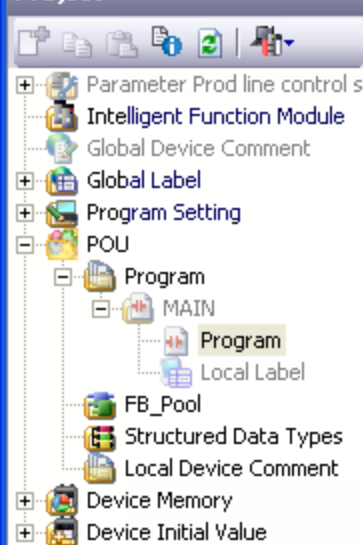
MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control - [[PRG]Read MAIN (Read Only) 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

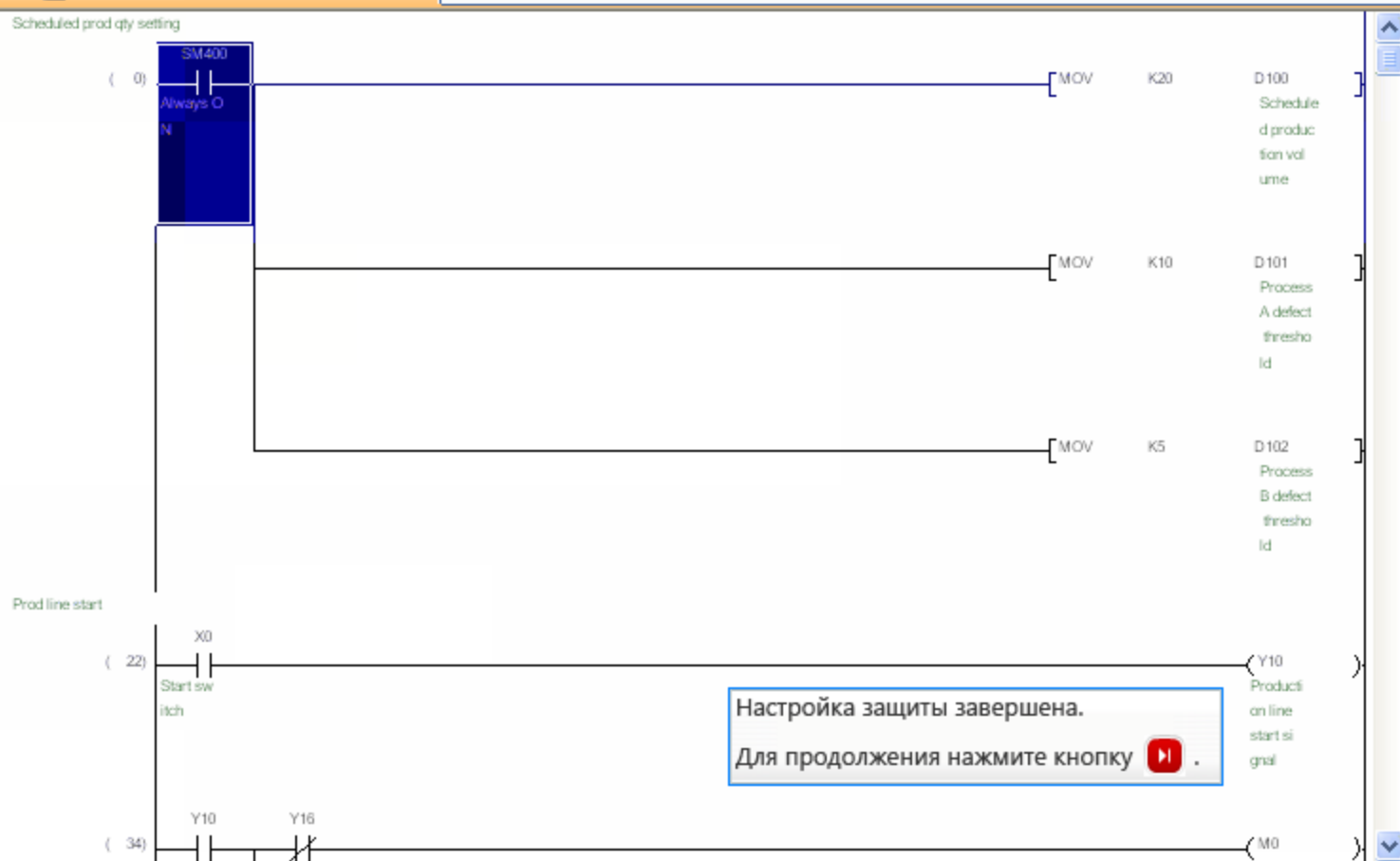


Project

User Library

Connection Destination

[PRG]Read MAIN (Read Onl...



Настройка защиты завершена.

Для продолжения нажмите кнопку



English

Simple

User

Q03UDE

Host Station

(15 M...



4.2

Резервирование проектов и управление версиями



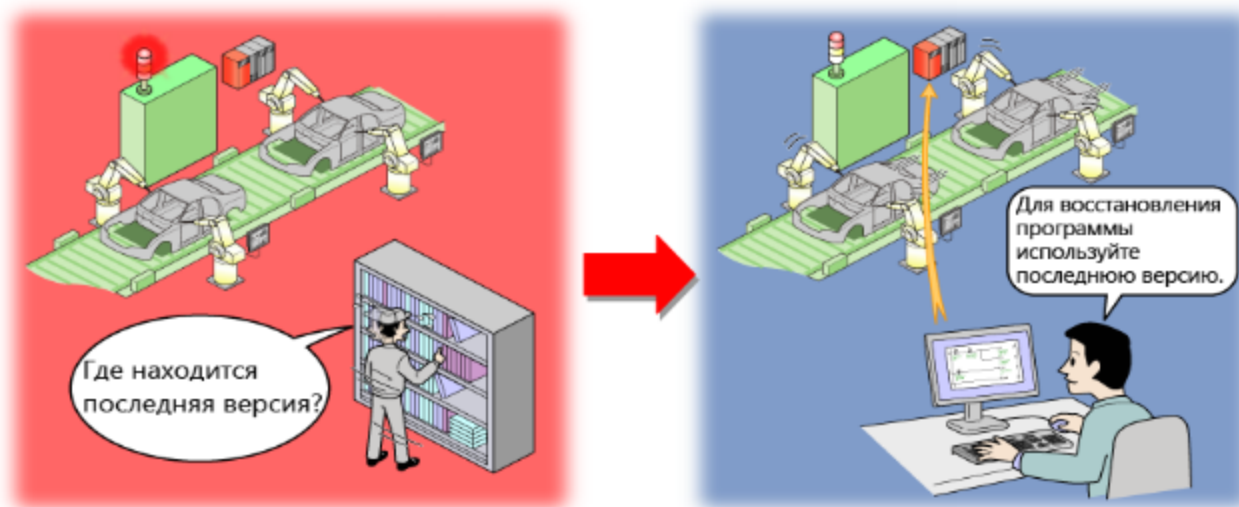
Сбой ЦП программируемого контроллера может привести к потере важной программы. Если резервная копия программы недоступна, надеяться на быстрое восстановление не приходится. Даже при наличии резервной копии программы трудно определить, совпадает ли ее версия с потерянной, если не используется функция управления версиями. Чтобы быть готовым к непредвиденным обстоятельствам, необходимо периодически выполнять резервное копирование и обеспечить управление версиями.

Для решения этих проблем используйте функцию **«Журнал изменений»**.

Эта функция может выполнять до 100 записей журнала изменений проекта (включающих номер изменения, дату/время, пользователя, название, комментарий).

При этом также выполняется резервное копирование данных проекта.

Управление версиями с помощью функции журнала изменений позволяет восстанавливать потерянные программы, сравнивать версии программ и таким образом быстро восстанавливать данные в непредвиденных ситуациях.



4.2

Резервирование проектов и управление версиями

MELSOFT Series GX Works2 C:\Sequential Programs\learning\Robot control (Revision No.3 : Revision B) - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation [PRG]Write MAIN 194 Step

Project

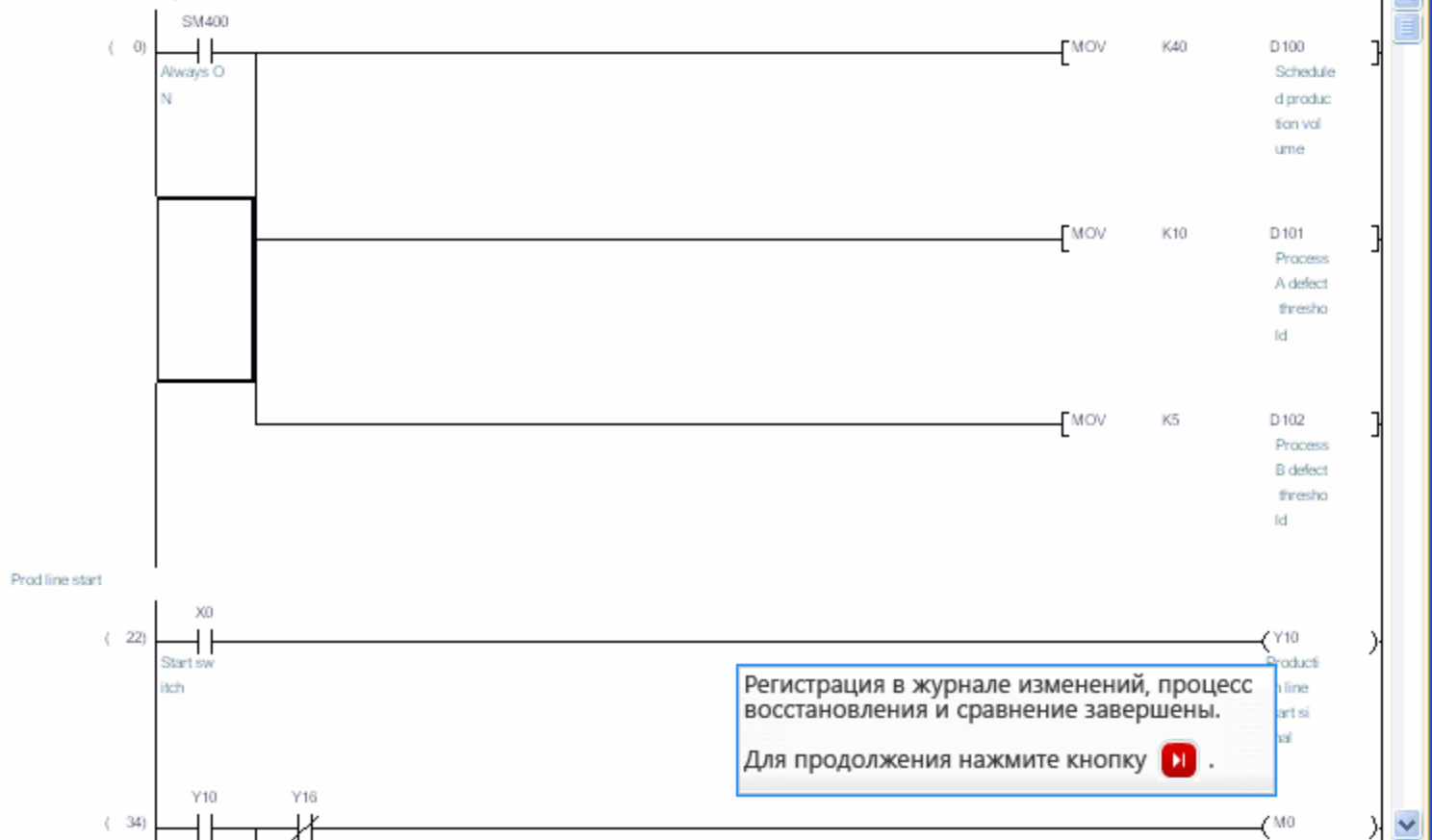
- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
- FB_Pool
- Structured Data Types
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

Scheduled prod qty setting



Регистрация в журнале изменений, процесс
восстановления и сравнение завершены.
Для продолжения нажмите кнопку .

English

Simple

Q03UDE

Host Station

(18 M)



4.3

Сравнение программ, записанных в программируемом контроллере и персональном компьютере



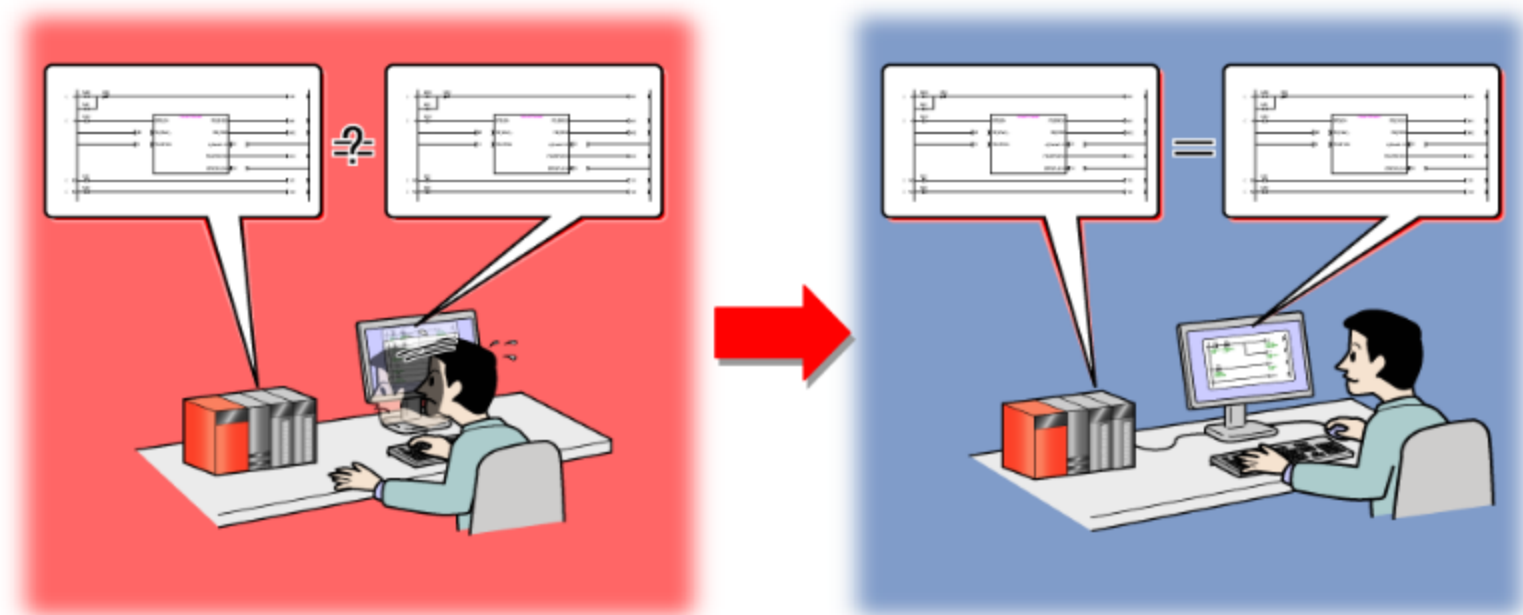
Обычно программы сохраняются на ПК в среде разработки, а также записываются в ПЛК.

Иногда, после внесения изменений в программы, например, только в ПЛК, одни и те же программы на ПК и на ПЛК начинают отличаться.

Выполнение только визуальной проверки того, одинаковы ли они, может привести к ошибкам.

Для решения этой проблемы используйте функцию **«Сравнить с ПЛК»**.

С помощью этой функции можно проверять, соответствует ли программа, открытая в GX Works2, программе, записанной в ПЛК.



4.3

Сравнение программ, записанных в программируемом контроллере и персональном компьютере



MELSOFT Series GX Works2 C:\Sequential Programs\le-learning\Robot control - [[PRG]Write MAIN 194 Step]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help



Navigation

Project

- Parameter Prod line control s
- Intelligent Function Module
- Global Device Comment
- Global Label
- Program Setting
- POU
 - Program
 - MAIN
 - Program
 - Local Label
 - FB_Pool
 - Structured Data Types
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

[PRG]Write MAIN 194 Step

Verify Result [PLC Verify]

Scheduled prod qty setting



Сравнение программ в режиме онлайн завершено.

Для продолжения нажмите кнопку .

English

Simple

Q03UDE

Host Station

(16 M...



Тест **Заключительный тест**

Вы завершили все уроки углубленного курса **Работа с ПО для ПЛК GX Works2** и готовы к прохождению итогового теста. Если вам неясны какие-либо из рассмотренных тем, воспользуйтесь возможностью еще раз просмотреть информацию по этим темам прямо сейчас. **Данный заключительный тест содержит всего 8 вопроса (8 пунктов).** Вы можете проходить заключительный тест любое количество раз.

Порядок подсчета баллов за тест

После выбора ответа обязательно нажмите кнопку **Ответить**. Если вы продолжите, не нажав кнопку «Ответить», ваш ответ будет потерян. (Будет считаться, что вы не ответили на вопрос.)

Результаты теста

Количество правильных ответов, количество вопросов, процент правильных ответов и результат (успешно ли пройден тест) будут отображаться на странице результатов.

Правильные ответы: **4**

Всего вопросов: **4**

Процент: **100%**

Для успешного прохождения теста вы должны ответить правильно на **60%** вопросов.

Продолжить

Просмотреть

- Нажмите кнопку **Продолжить**, чтобы завершить тест.
- Для просмотра теста нажмите кнопку **Просмотреть**. (Правильные ответы будут отмечены.)
- Нажмите кнопку **Повторить попытку**, чтобы пройти тест еще раз.

Тест**Заключительный тест 1**

Какая из следующих функций повышает эффективность программирования, позволяя использовать блоки программ на языке релейной логики как совместно используемые компоненты? (Выберите один вариант ответа.)

- встроенный структурированный текст;
- Метка
- функциональный блок;

Тест**Заключительный тест 2**

Какая из следующих функций позволяет создавать удобные для чтения программы, изменяя названия операндов на названия, связанные с их применением? (Выберите один вариант ответа.)

- Device comment (Комментарий к операнду)
- Метка
- Note (Примечание)

Тест**Заключительный тест 3**

Какая из следующих функций позволяет создавать удобные для чтения программы, предоставляя информацию по обработке для каждого блока программы на языке релейной логики? (Выберите один вариант ответа.)

- Device comment (Комментарий к операнду)
- Оператор строки
- Note (Примечание)

Тест

Заключительный тест 4

Выберите правильное описание функции «Сравнить с ПЛК». (Выберите один вариант ответа.)

- Эта функция сравнивает редактируемую программу с программой, записанной в журнале изменений.
- Эта функция сравнивает редактируемую программу с выбранной программой, сохраненной в ПК.
- Эта функция сравнивает редактируемую программу с программой, записанной в ЦП ПЛК.

Тест

Заключительный тест 5

Выберите правильное описание функции «Настройка системы ввода/вывода». (Выберите один вариант ответа.)

- Эта функция имитирует работу внешнего оборудования ввода/вывода на персональном компьютере во время отладки.
- Эта функция удаленно управляет работой внешнего оборудования ввода/вывода на персональном компьютере во время отладки.
- Эта функция имитирует работу ЦП ПЛК на персональном компьютере во время отладки.

Тест**Заключительный тест 6**

Выберите правильное описание функции «Журнал изменений». (Выберите один вариант ответа.)

- Пошагово записывает работу ПО GX Works2, чтобы ее можно было восстановить позднее.
- Ведет журнал и учет резервных копий проекта, позволяя позднее выполнять проверку и восстановление.

Ответить

Назад

Тест**Заключительный тест 7**

Какую из следующих функций можно использовать во время отладки для изменения значений операндов без внесения изменений в программу? (Выберите один вариант ответа.)

- Прерывание выполнения
- настройка системы ввода/вывода.
- Проверка операндов с условием выполнения

Тест

Заключительный тест 8

Допустим, что проект включает две программы, А и В, и вы используете функцию «метка». К каким из следующих типов меток будет иметь доступ программа В? (Выберите два варианта ответов.)

- Глобальная метка
- Локальная метка программы А
- Локальная метка программы В

Тест**Результат теста**

Вы завершили заключительный тест. Ваша область результатов является следующей.
Чтобы закончить заключительный тест, перейдите к следующей странице.

Правильные ответы: **8**

Всего вопросов: **8**

Процент: **100%**

Продолжить

Просмотреть

**П о з д р а в л я е м ! В ы п р о ш л и
т е с т .**

Вы завершили углубленный курс по работе с ПО для ПЛК GX Works2.

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](#)

[Заккрыть](#)