

Программное обеспечение SoMachine

Руководство
по началу работы

2010

Компания Schneider Electric не несёт ответственность за ошибки, которые могут содержаться в настоящем документе. Если у Вас есть предложения по улучшению или корректировке данной публикации либо Вы обнаружили в ней ошибки, просьба уведомить нас об этом.

Ни одна часть настоящего документа не может быть воспроизведена в какой бы то ни было форме или каким бы то ни было способом, электронным или механическим, включая фотокопирование, без прямого письменного разрешения компании Schneider Electric.

В процессе установки и эксплуатации данных изделий должны соблюдаться все местные, региональные и национальные нормы и правила. В целях безопасности и для гарантии соответствия задокументированным данным, ремонт компонентов должен выполняться только изготовителем.

В случае, если данные устройства эксплуатируются в составе оборудования, которое должно отвечать техническим требованиям по безопасности, необходимо руководствоваться соответствующими инструкциями.

В данных изделиях должно использоваться только программное обеспечение, разработанное или одобренное компанией Schneider Electric; несоблюдение этого требования может привести к телесным повреждениям, материальному ущербу или неправильной работе оборудования.

Невыполнение вышеперечисленных требований может привести к травмированию людей или порче оборудования.

© 2009 Schneider Electric. Все права защищены.

Содержание



Меры безопасности	5
О данном Руководстве	7
Глава 1 Введение	9
Общие сведения об аппаратных средствах	9
Глава 2 Как создать проект	13
Графический интерфейс пользователя	14
Создание проекта управления на основе контроллера M238	16
Создание программного модуля (POU) перед написанием логики	19
Создание переменных	25
Назначение переменных входам/выходам контроллера	30
Компиляция проекта	32
Глава 3 Конфигурирование устройств CANopen	33
Конфигурирование интерфейса CANopen	34
Добавление устройства CANopen	39
Глава 4 Конфигурирование интерфейса сканера входов/выходов Modbus	47
Конфигурирование последовательного линейного интерфейса для сканера входов/выходов Modbus	48
Добавление устройства Modbus SL	53
Глава 5 Связь с терминалом ЧМИ	61
Публикация переменных связи	62
Конфигурирование устройства ЧМИ в проекте	65
Конфигурирование связи с терминалами ЧМИ	67
Подписка переменных для терминалов ЧМИ	68
Глава 6 Подключение и загрузка	73
Подключение к контроллеру	74
Загрузка в контроллер	79
Загрузка приложений управления и ЧМИ посредством однократного подключения	80
Пуск и остановка контроллера	81
Изменения, вносимые в контроллер в оперативном режиме	82
Имитация контроллера	84

Глава 7	Отладка приложения	85
	Изменение значений переменных	86
	Пошаговое выполнение проекта с точками прерывания	87
Глава 8	Пример программирования контроллера	91
	Пример программирования на языке структурированного текста	91
Глоссарий	97
Предметный указатель	105

Меры безопасности



Важная информация

Уведомление

Внимательно прочитайте данные правила и ознакомьтесь с оборудованием прежде, чем осуществлять его установку, эксплуатацию или техническое обслуживание. Приведённые ниже специальные сообщения, которые могут находиться в документации или на аппарате, имеют целью предупредить вас о потенциальных опасностях или привлечь ваше внимание к информации, призванной разъяснить или упростить ту или иную процедуру.



Этот символ в комбинации с предупреждающей табличкой ОПАСНО или ОСТОРОЖНО сигнализирует об опасности поражения электрическим током, из-за которой несоблюдение правил техники безопасности может привести к телесным повреждениям.



Этот символ, обозначающий опасность, предупреждает Вас о риске получения телесных повреждений. Во избежание травм или летального исхода неукоснительно соблюдайте правила безопасности, указанные рядом с этим символом.

ОПАСНО

ОПАСНО означает реально опасную ситуацию, которая, если её не предотвратить, **приведёт** к серьёзным телесным повреждениям или летальному исходу.

ОСТОРОЖНО!

ОСТОРОЖНО означает потенциально опасную ситуацию, которая, если её не предотвратить, **может привести** к летальному исходу или серьёзным телесным повреждениям.



ВНИМАНИЕ!

ВНИМАНИЕ означает потенциально опасную ситуацию, которая, если её не предотвратить, **может привести** к телесным повреждениям малой или средней тяжести.

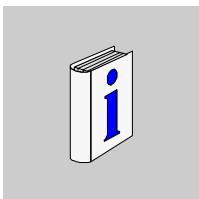
ВНИМАНИЕ!

ВНИМАНИЕ без символа опасности означает потенциально опасную ситуацию, которая, если её не предотвратить, **может привести** к материальному ущербу.

ВАЖНОЕ ЗАМЕЧАНИЕ

Установка, эксплуатация и техническое обслуживание электротехнического оборудования должны выполняться квалифицированными специалистами. Компания Schneider Electric не несёт никакой ответственности за возможные последствия, могущие иметь место в результате использования данной документации неквалифицированным персоналом.

О данном Руководстве



Общие сведения

Предназначение документа

Настоящее Руководство содержит описание программного обеспечения SoMachine, в том числе подробные указания по созданию проектов с использованием контроллеров и человеко-машинных интерфейсов (ЧМИ).

Область применения

Данная документация применима при использовании программного обеспечения SoMachine под операционной системой Microsoft Windows XP Professional или 32-битной Windows Vista.

Документы по теме

Наименование документа	№ по каталогу
Руководство по программированию SoMachine	EIO00000000067
Справочное руководство по аппаратной части контроллера M238	EIO0000000016

Вы можете загрузить указанные публикации и другую техническую информацию с нашего Web-сайта по адресу: www.schneider-electric.com.

Комментарии пользователя

Вы можете выслать свои комментарии по электронной почте по адресу:
techpub@schneider-electric.com

Введение

1

Общие сведения об аппаратных средствах

Введение

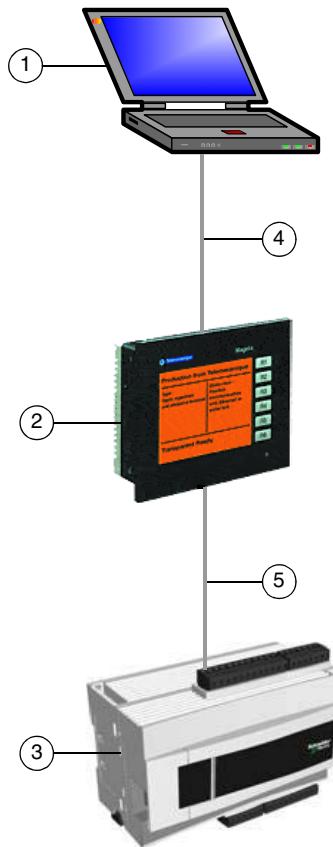
Ниже даны общие сведения об архитектуре аппаратных средств, описываемой в настоящем Руководстве.

Архитектура

В настоящем Руководстве описывается архитектура, обеспечивающая соединение компьютера, в котором выполняется программа SoMachine, одновременно с панелью человеко-машинного интерфейса (ЧМИ) и контроллером, при этом для ввода в действие обоих устройств (загрузка или изменение данных и т.д.) требуется только один кабель. Элементы архитектуры:

- ПК, в котором выполняется SoMachine, подключаемый через USB-соединение типа А
- к терминалу ЧМИ XBT GT Magelis, подсоединяемому по последовательному каналу
- к контроллеру M238.

Графическое представление

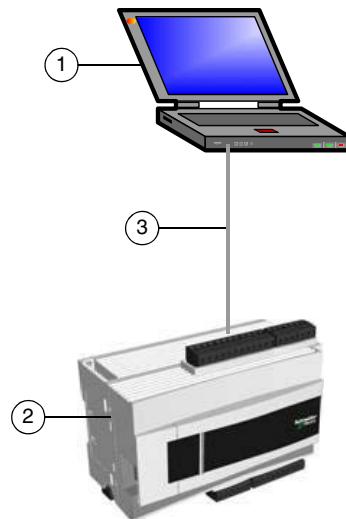


1. ПК, в котором выполняется SoMachine
2. Терминал ЧМИ XBT GT Magelis
3. Контроллер TM238LFDC24DT
4. Кабель XBTZG935 для USB-соединения типа А – А между ПК с SoMachine и терминалом ЧМИ
5. Кабель XBTZ9008 для последовательного соединения (Sub-D9 – RJ45) между терминалом ЧМИ и контроллером M238

Другие варианты архитектуры

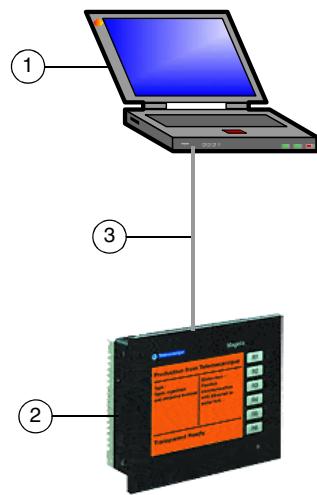
Также возможно раздельное подсоединение ПК с программой SoMachine к терминалу ЧМИ и к контроллеру, как показано на следующих иллюстрациях.

ПК с SoMachine, подсоединеный к контроллеру M238



1. ПК, в котором выполняется SoMachine
2. Контроллер M238
3. USB-кабель для USB-соединения типа А – мини В между ПК с SoMachine и контроллером M238

ПК с SoMachine, подсоединеный к терминалу ЧМИ XBT GT



1. ПК, в котором выполняется SoMachine
2. Терминал ЧМИ XBT GT Magelis
3. Кабель BTZG935 для USB-соединения типа А – А между ПК с SoMachine и терминалом XBT GT

Как создать проект

2

Введение

В этом разделе дано краткое описание пользовательского интерфейса ПО SoMachine и указаны необходимые шаги для создания примерного проекта управления для контроллера M238 и определения переменных.

Содержание данной главы

Данная глава содержит следующие темы:

Тема	Страница
Графический интерфейс пользователя	14
Создание проекта управления на основе контроллера M238	16
Создание программного модуля (POU) перед написанием логики	19
Создание переменных	25
Назначение переменных входам/выходам контроллера	30
Компиляция проекта	32

Графический интерфейс пользователя

Введение

Ниже дано краткое описание графического интерфейса пользователя ПО SoMachine.

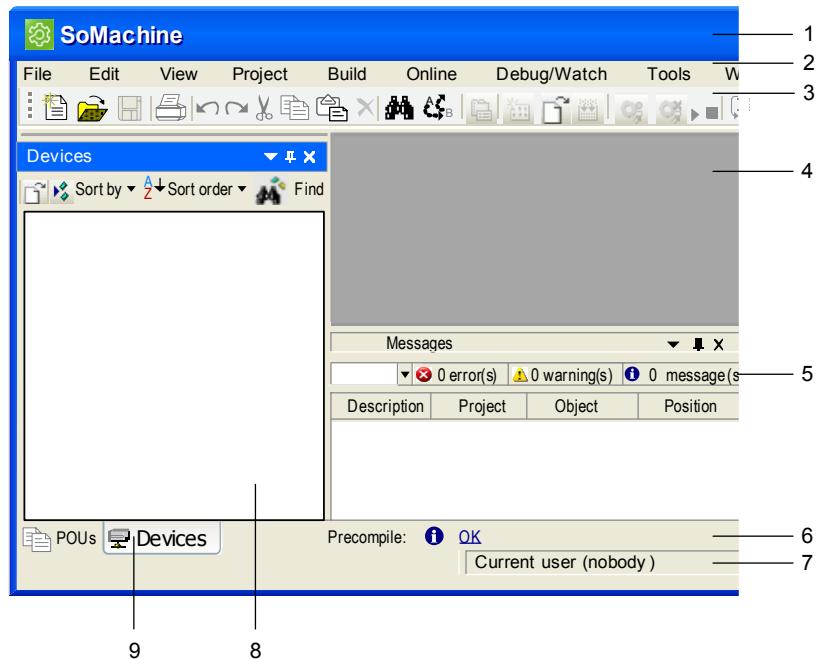
Запуск программы

Чтобы запустить программу, сделайте следующее:

Шаг	Действие
1	<p>Установите ПО SoMachine путём выполнения файла <i>setup.exe</i> и следуйте указаниям на экране.</p> <p>Результат: на вашем ПК устанавливаются следующие программы:</p> <ul style="list-style-type: none">● SoMachine● Vjeo-Designer● Microsoft .NET Framework 2.0 (если не была установлена ранее)
2	<p>После установки запустите ПО одним из следующих способов:</p> <p>Двойным щелчком по значку рабочего стола:</p>  <p>Или используйте меню Programs (Программы): Start → Programs → Schneider Electric → SoMachine (Пуск → Программы → Schneider Electric → SoMachine).</p> <p>Результат: запустится программа SoMachine.</p>

Графический интерфейс

Начальный экран программы имеет следующий вид:



1. Линейка заголовка
2. Линейка меню
3. Панель инструментов
4. Многовкладочное окно редактора
5. Okno сообщений
6. Информационная строка
7. Страна состояния

8+9. Okno Devices / POU (Устройства / Программный модуль) (в данном примере выбрана вкладка **Devices**)

Интерфейс пользователя содержит несколько плавающих окон, которые можно открывать с помощью меню **View** (Вид). Вы можете располагать различные окна внутри окна SoMachine в соответствии с вашими предпочтениями.

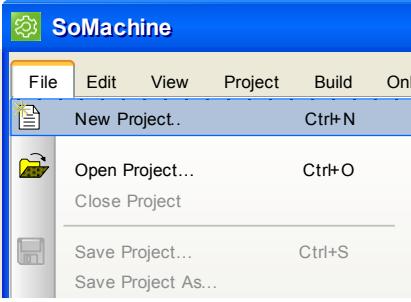
Создание проекта управления на основе контроллера M238

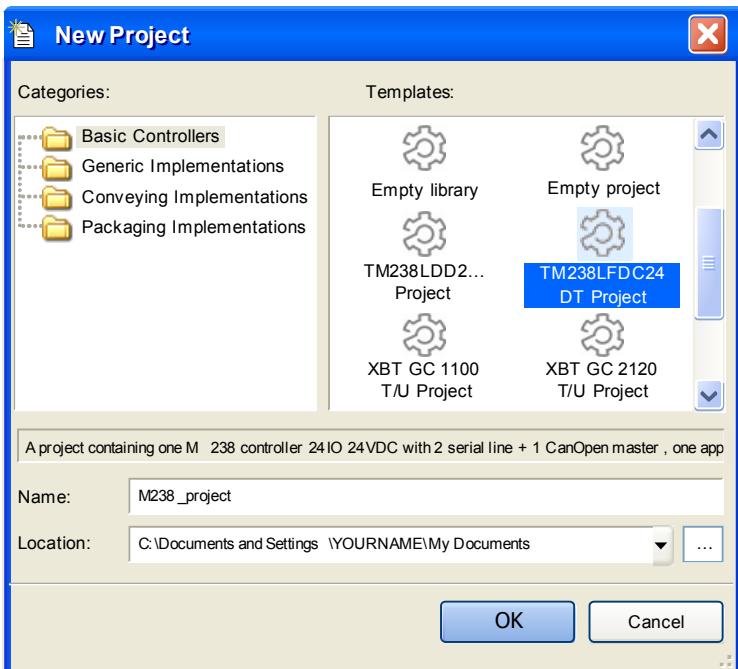
Введение

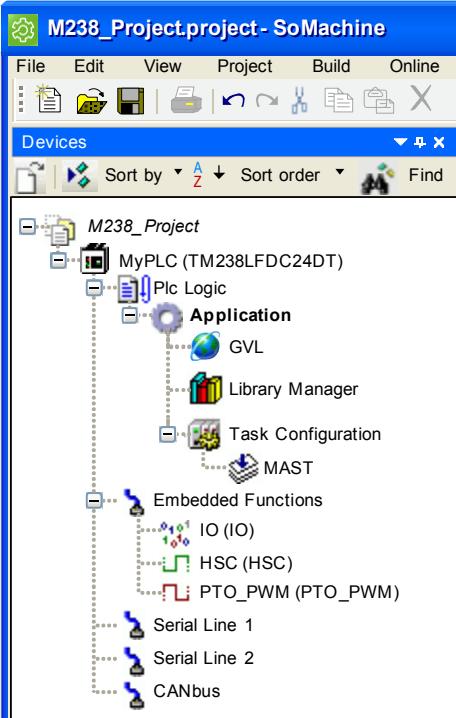
Ниже описывается, как создать пример проекта для контроллера M238 Schneider Electric.

Создание проекта

Чтобы создать проект, сделайте следующее:

Шаг	Действие
1	<p>В меню File (Файл) выберите команду New Project... (Новый Проект):</p>  <p>Результат: появится диалоговое окно New Project:</p>

Шаг	Действие
2	<p>В диалоговом окне New Project выберите шаблон и укажите наименование проекта:</p>  <p>В списке Categories (Категории) выберите Basic Controllers (Базовые контроллеры). В окне Templates (Шаблоны) выберите пиктограмму TM238LFDC24DT (контроллер M238 с ведущим устройством CANopen). В текстовом поле Name (Имя) введите наименование вашего проекта. В текстовом поле Location (Положение) укажите путь доступа к месту, где будут сохраняться файлы проекта, или щёлкните на кнопке просмотра для поиска подходящей папки.</p>
3	Щёлкните OK для подтверждения и выхода из диалогового окна.

Шаг	Действие
4	<p>В левом окне перейдите с вкладки POU (Программный модуль) на вкладку Devices (Устройства).</p> <p>Результат: новый проект включает в себя различные вложенные папки в окне Devices, как показано ниже.</p>  <p>Приложение M238 автоматически определяется как активное, соответственно строка Application (Приложение) выделена полужирным шрифтом.</p>

Активное Приложение

Примечание: если ваше устройство содержит несколько приложений, убедитесь, что активно используемое приложение. По умолчанию в активном приложении выполняется несколько команд (например, команда **Build** (Скомпилировать)).

В окне **Devices** активное приложение выделено полужирным шрифтом.

Чтобы активировать приложение, щёлкните правой кнопкой на окне **Devices** и выберите в контекстном меню **Set Active Application** (Определить Активное Приложение).

Создание программного модуля (POU) перед написанием логики

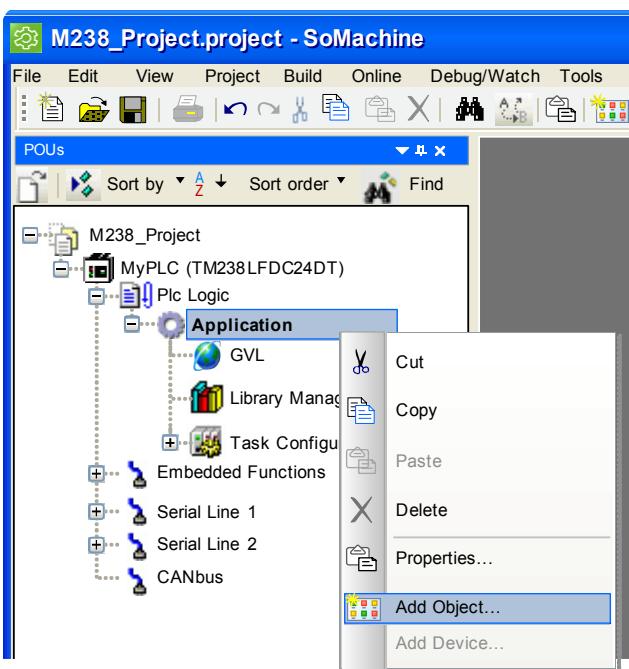
Введение

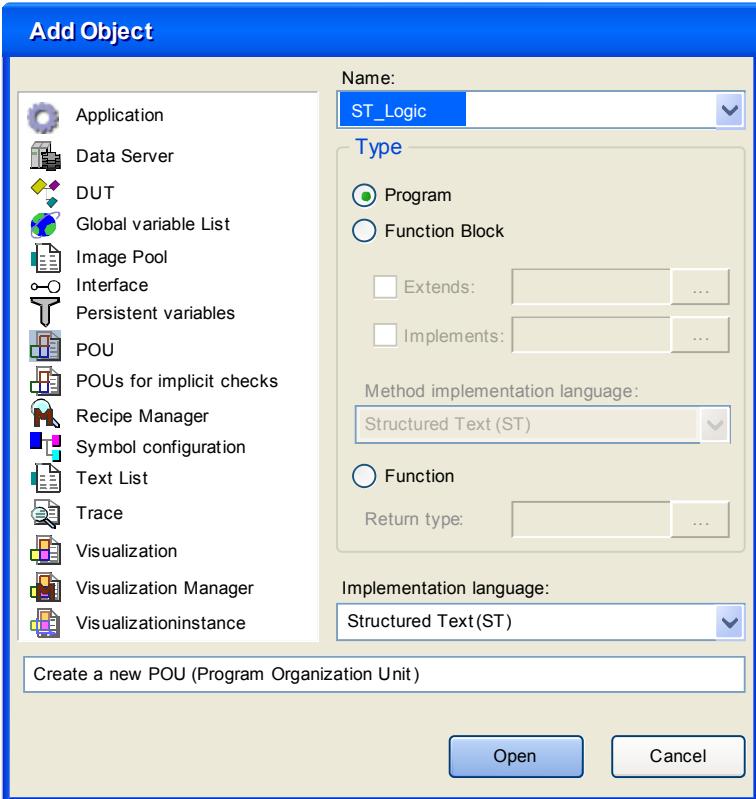
Перед написанием своей программы вам необходимо сначала определить программный модуль (POU = Program Organization Unit), в который вы введёте свой код программирования.

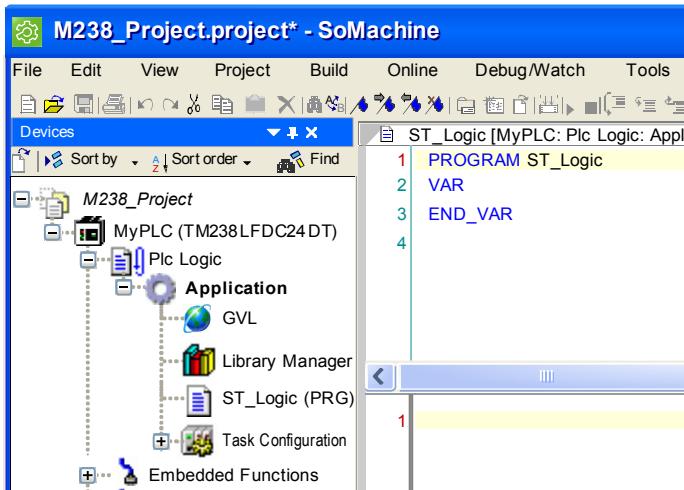
После этого вы сможете написать свою программу в программном редакторе.

Создание программного модуля

Чтобы создать программный модуль, сделайте следующее:

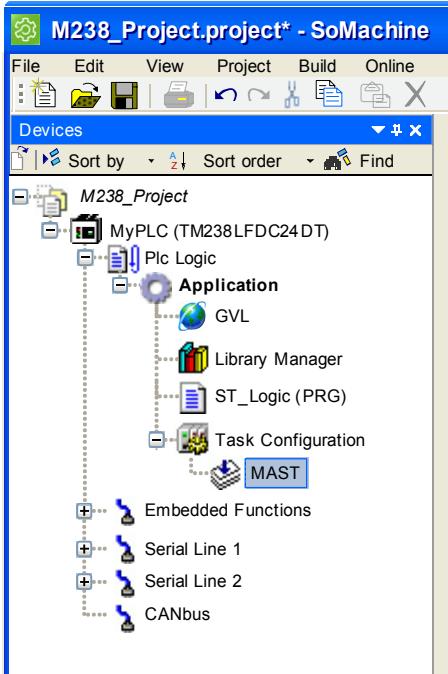
Шаг	Действие
1	<p>Щёлкните правой кнопкой на узле Application (Приложение) вашего проекта и выберите команду Add object... (Добавить объект):</p>  <p>Результат: появится диалоговое окно Add object....</p>

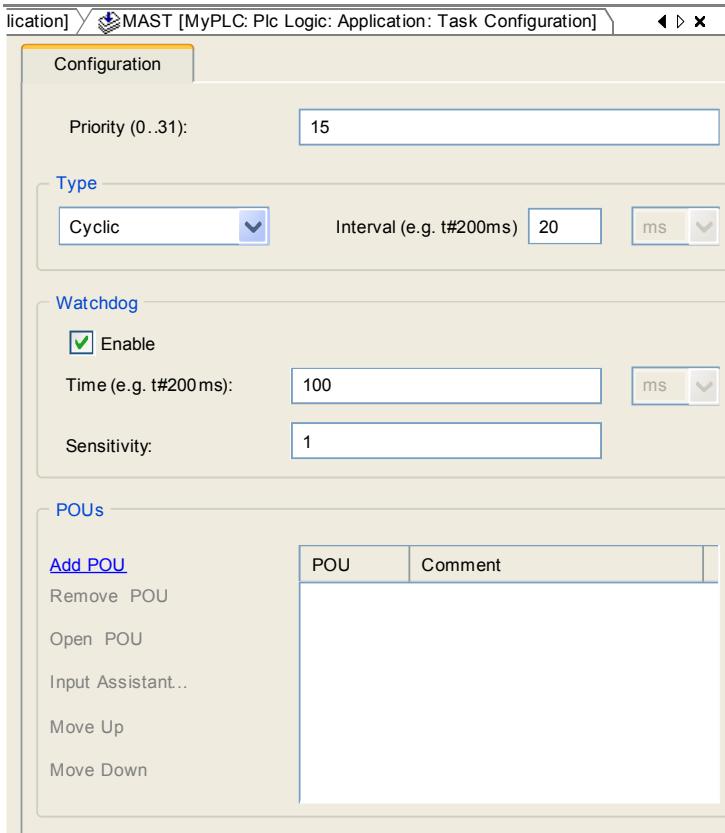
Шаг	Действие
2	<p>В диалоговом окне Add Object выберите пункт POU (Программный модуль) из расположенного слева списка и определите тип и язык программирования данного объекта:</p>  <p>Ведите имя программного модуля в поле Name (Имя). В зоне Type (Тип) выберите опцию Program (Программа). Из списка Implementation language (Язык реализации) выберите язык программирования. В приведённом примере использован язык Structured Text (ST) (Язык структурированного текста).</p>

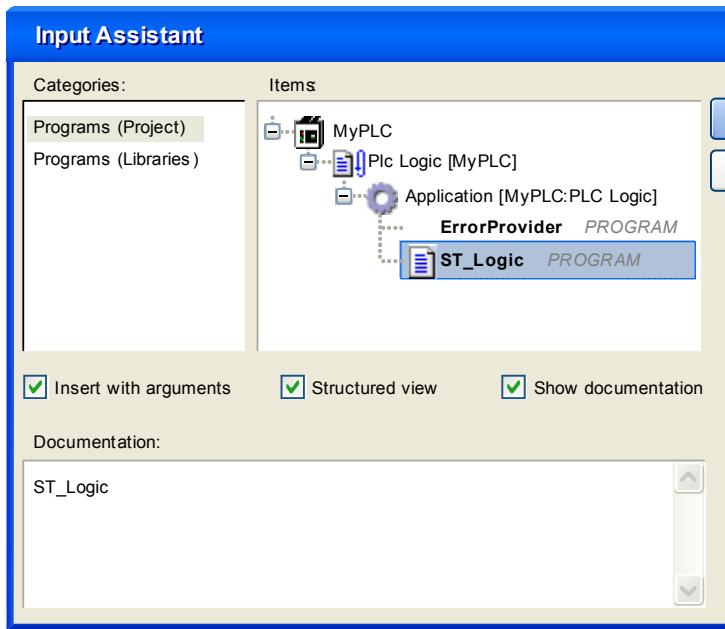
Шаг	Действие
3	<p>Щёлкните на Open (Открыть), чтобы закрыть диалоговое окно и открыть окно редактора программного модуля.</p> <p>Результат: с правой стороны появится редактор программного модуля (POU).</p>  <p>Редактор логики состоит из двух окон:</p> <ul style="list-style-type: none"> в верхнее окно служит для объявления данных (переменных и экземпляров функционального блока); в нижнем окне вводится код программы.
4	<p>Создайте свою логику в программном модуле.</p> <p>Подробный пример создания логики приведён в разделе «Пример программирования контроллера», стр. 91.</p> <p>Перед тем, как начать определение переменных в программном редакторе, добавьте программный модуль к задаче, как описано ниже.</p>

Добавление программного модуля к задаче

Выполнение программного модуля (POU) управляется посредством задачи. По умолчанию создаётся задача MAST, но вы можете создать другие задачи в **Task Configuration** (Конфигурирование задачи). Чтобы добавить программный модуль к задаче MAST, сделайте следующее:

Шаг	Действие
1	<p>Откройте узел Task Configuration в дереве проекта и дважды щёлкните на предварительно сконфигурированной задаче MAST:</p>  <p>Результат: в окне справа появится вкладка MAST.</p>

Шаг	Действие
2	<p>На вкладке задачи MAST указано, что задача MAST по умолчанию выполняется циклически с интервалом 20 мс:</p>  <p>В разделе POUs (Программные модули) вкладки задачи MAST выберите ссылку Add POU (Добавить программный модуль).</p>

Шаг	Действие
3	<p>Результат: появится диалоговое окно Input Assistant («Помощник» по вводу)</p>  <p>Щёлкните на пункте Programs (Project) (Программы (Проект)) раздела Categories (Категории) и откройте папки в зоне Items (Элементы), щёлкая на знаках «+».</p> <p>Выберите свой элемент программного модуля (в данном примере ST_Logic), являющийся подузлом Application, и щёлкните OK.</p> <p>Результат: диалоговое окно Input Assistant закроется и вы вернётесь в диалоговое окно Task Configuration. Вкладка задачи MAST теперь содержит новый программный модуль в своём разделе POU.</p>

Создание переменных

Введение

ПО SoMachine позволяет определять переменные с различными областями действия:

- локальные переменные, объявляемые в программном модуле и доступные только в этом программном модуле;
- глобальные переменные, объявляемые через GVL (Global Variable List = список глобальных переменных) и доступные во всех программных модулях приложения.

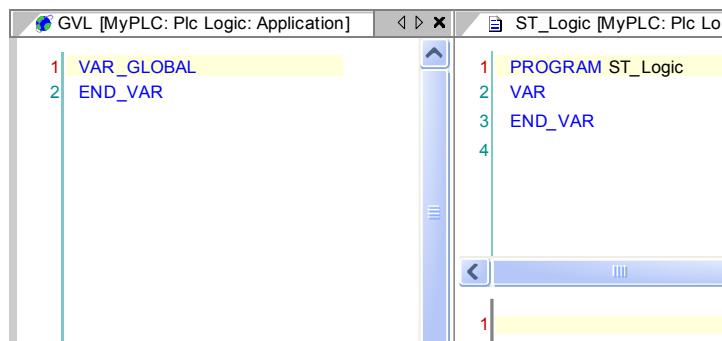
Функционирование редакторов переменных, относящихся к программному модулю (локальные переменные) и к списку глобальных переменных (глобальные переменные), идентично.

Ниже описывается создание переменных только с использованием редактора программного модуля.

Чтобы открыть редакторы, сделайте следующее:

- редактор списка глобальных переменных: щёлкните на узле **GVL** (Список глобальных переменных) в окне **Devices** (Устройства);
- редактор логики: щёлкните на узле **POU** (Программный модуль) в окне **Devices**.

Графический интерфейс пользователя с открытыми программным модулем и списком глобальных переменных (отображение посредством вертикальных вкладок):

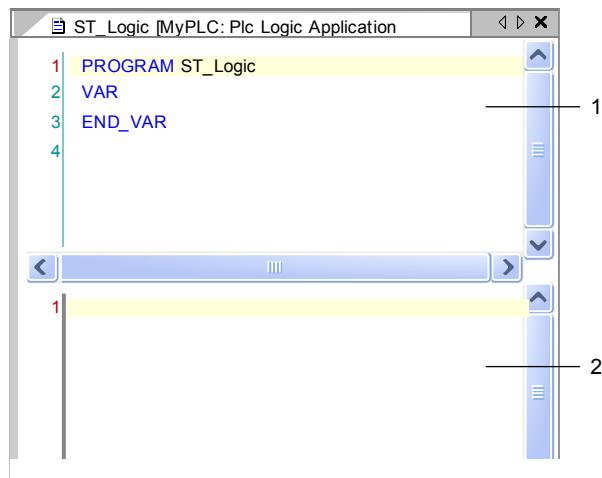


Два способа определения переменных

Вы можете создавать новые переменные двумя способами:

- набирая объявления переменных в списке глобальных переменных или в верхней части редактора логики – разделе описаний;
- вводя код с новыми переменными в нижней части редактора логики – разделе реализации. При объявлении переменных используется модуль оперативной помощи («мастер»).

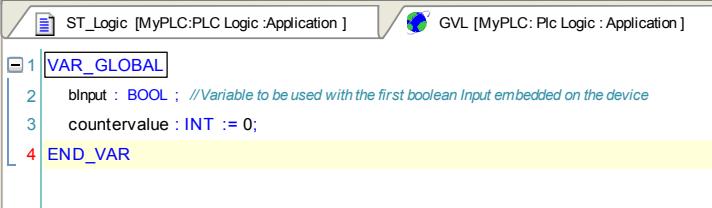
Редактор логики для определения переменных, состоящий из двух разделов:



1. Раздел описаний
2. Раздел реализации

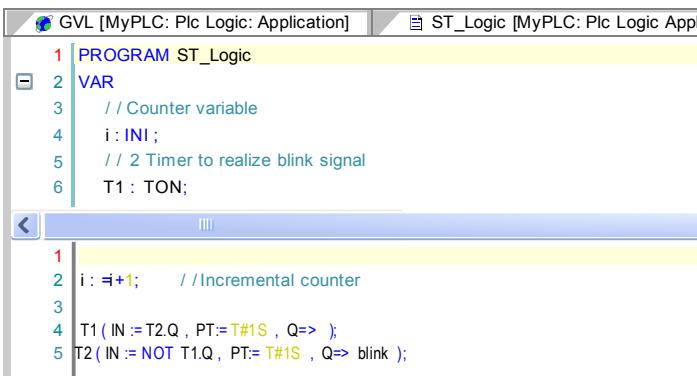
Ввод глобальных переменных и их объявления

Для непосредственного определения глобальных переменных в списке глобальных переменных, сделайте следующее:

Шаг	Действие
1	<p>Вставьте новую строку после VAR_Global / VAR (GVL) и введите переменные по вашему выбору с соответствующим объявлением. Соблюдайте следующий синтаксис:</p> <pre><identifiant>{AT<адреса>} :<Type>{:=initialization};</pre> <p>В данном примере мы объявляем глобальную переменную bInput1 логическим (булевым) значением, а countervalue – целым значением с нулевым исходным значением:</p>  <p>The screenshot shows the GVL editor interface. It has two tabs: "ST_Logic [MyPLC: PLC Logic : Application]" and "GVL [MyPLC: Plc Logic : Application]". The GVL tab is active. It displays the following code: 1 VAR_GLOBAL 2 binput : BOOL ; //Variable to be used with the first boolean Input embedded on the device 3 countervalue : INT := 0; 4 END_VAR</p> <p>Результат: переменная bInput1 теперь объявлена логическим значением.</p>

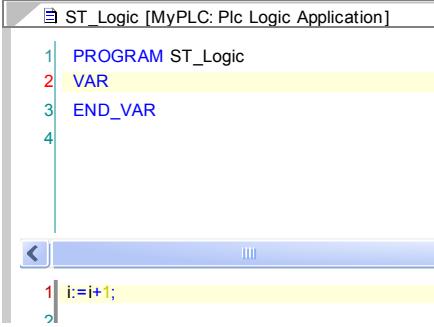
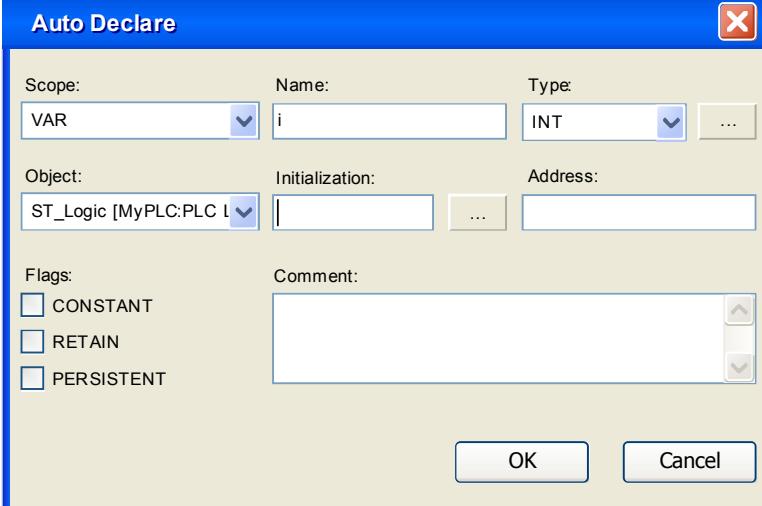
Ввод локальных переменных и их объявления

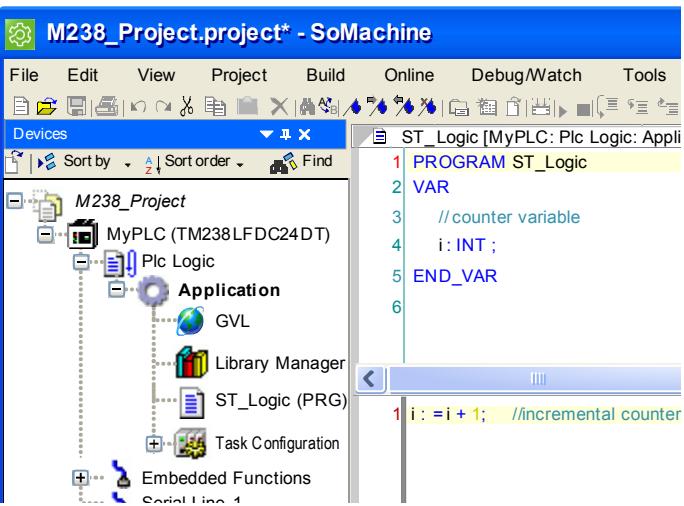
Для непосредственного определения локальных переменных в редакторе логики, сделайте следующее:

Шаг	Действие
1	<p>Вставьте новую строку после VAR/END_VAR (редактор логики) и введите переменные по вашему выбору с соответствующим объявлением. Соблюдайте следующий синтаксис:</p> <pre><identifiant>{AT<адрес>} :<Type>{:=initialization};</pre> <p>В данном примере мы объявляем локальную переменную i целым значением:</p>  <pre> GVL [MyPLC: Plc Logic: Application] PROGRAM ST_Logic VAR i : INT; // 2 Timer to realize blink signal T1 : TON; i : =+1; // Incremental counter T1(IN :=T2.Q , PT:=T#1S , Q=>); T2(IN :=NOT T1.Q , PT=T#1S , Q=> blink); </pre> <p>Результат: переменная <code>i</code> теперь объявлена логическим значением.</p>

Определение переменных в разделе реализации редактора логики

Чтобы определить локальные переменные в нижней части программного редактора (раздел реализации), сделайте следующее:

Шаг	Действие
1	<p>В разделе реализации программного редактора щёлкните на первой строке и введите код программы.</p> <p>В нашем примере мы вводим следующий код программы с целой переменной <i>i</i> :</p> 
2	<p>Результат: откроется диалоговое окно Auto Declare (Автообъявление) с уже определёнными параметрами Scope (Область действия), Name (Имя) и Type (Тип):</p> 

Шаг	Действие
3	<p>В списке Type (Тип) выберите тип, который вы хотите присвоить этой переменной (INT, WORD, BOOL, и т.д.).</p> <p>С помощью поля Object (Объект) определите программный модуль, для которого действительна переменная.</p> <p>С помощью поля Initialization (Инициализация) вы можете сконфигурировать исходное значение переменной.</p> <p>В поле Address (Адрес) вы можете ввести физический адрес переменной.</p> <p>При желании вы также можете ввести текст в текстовом поле Comment (Комментарий); этот текст будет вставлен в качестве описания в раздел описаний.</p> <p>Щёлкните OK, чтобы закрыть диалоговое окно.</p> <p>Результат: объявление переменной <code>i</code> будет вставлено в раздел описаний программного редактора.</p>  <pre> M238_Project.project* - SoMachine File Edit View Project Build Online Debug/Watch Tools Devices ▾ X M238_Project MyPLC (TM238LFDC24DT) Plc Logic Application GVL Library Manager ST_Logic (PRG) <-- Selected Task Configuration Embedded Functions Serial Line 1 ST_Logic [MyPLC: Plc Logic: Application] 1 PROGRAM ST_Logic 2 VAR 3 //counter variable 4 i: INT ; 5 END_VAR 6 1 i := i + 1; //incremental counter </pre>

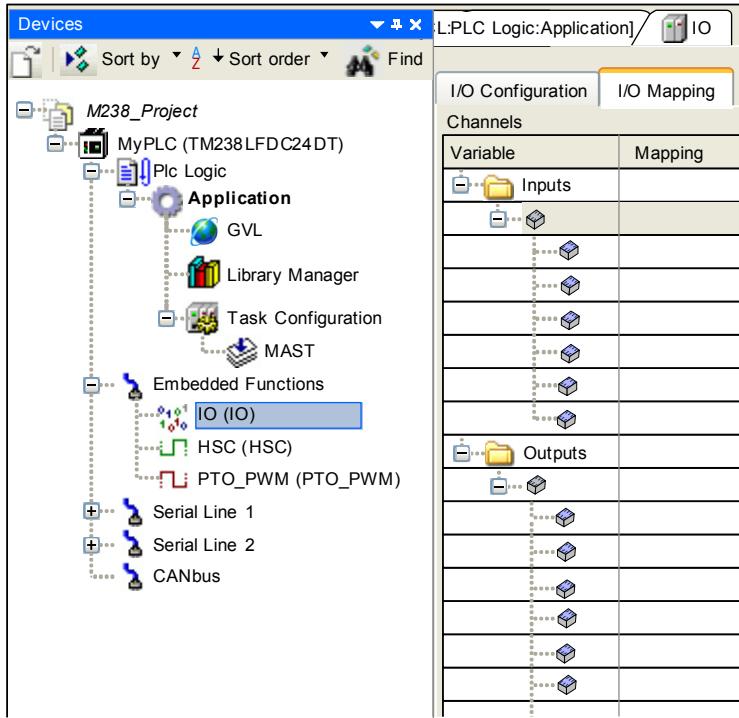
Назначение переменных входам/выходам контроллера

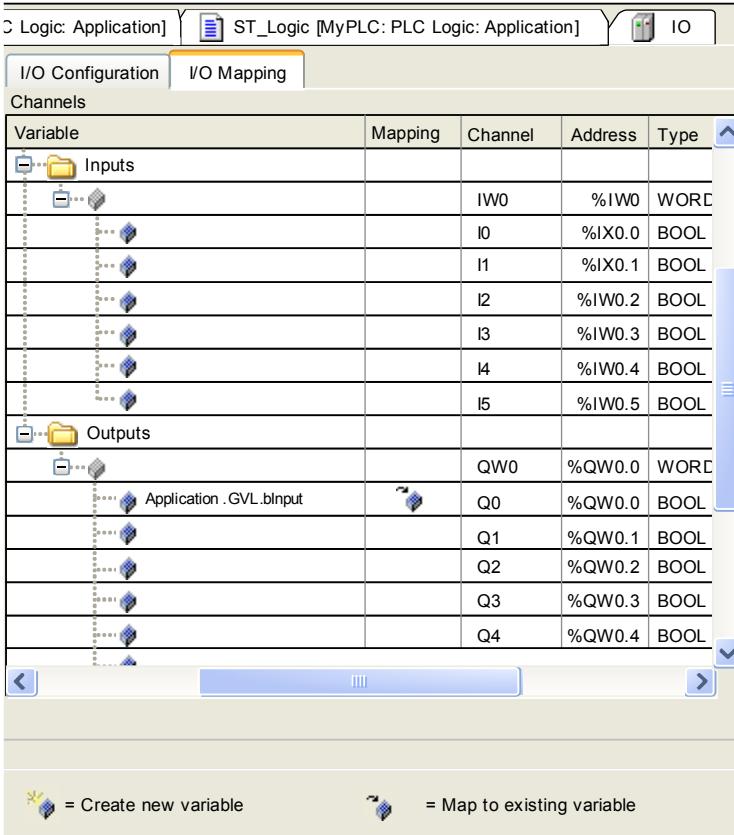
Введение

Ниже описана процедура назначения вновь созданных переменных входам/выходам автомата.

Назначение входов/выходов

Чтобы назначить переменные различным каналам ввода/вывода контроллера, сделайте следующее:

Шаг	Действие
1	<p>Дважды щёлкните на пункте IO(IO) (Ввод/вывод) в разделе Embedded Functions (Вложенные функции) окна Devices (Устройства).</p> 

Шаг	Действие																																																																																
2	<p>Окно, посвящённое оборудованию, имеет несколько вкладок. Выберите локальную вкладку I/O Mapping (Назначение входов/выходов) для встроенных входов/выходов контроллера.</p>  <table border="1"> <thead> <tr> <th>Variable</th> <th>Mapping</th> <th>Channel</th> <th>Address</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Inputs</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>IW0</td> <td></td> <td></td> <td>%IW0</td> <td>WORD</td> </tr> <tr> <td>I0</td> <td></td> <td></td> <td>%IX0.0</td> <td>BOOL</td> </tr> <tr> <td>I1</td> <td></td> <td></td> <td>%IX0.1</td> <td>BOOL</td> </tr> <tr> <td>I2</td> <td></td> <td></td> <td>%IW0.2</td> <td>BOOL</td> </tr> <tr> <td>I3</td> <td></td> <td></td> <td>%IW0.3</td> <td>BOOL</td> </tr> <tr> <td>I4</td> <td></td> <td></td> <td>%IW0.4</td> <td>BOOL</td> </tr> <tr> <td>I5</td> <td></td> <td></td> <td>%IW0.5</td> <td>BOOL</td> </tr> <tr> <td>Outputs</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>QW0</td> <td></td> <td></td> <td>%QW0.0</td> <td>WORD</td> </tr> <tr> <td>Application.GVL.blinput</td> <td>Map to existing variable</td> <td>Q0</td> <td>%QW0.0</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>Q1</td> <td>%QW0.1</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>Q2</td> <td>%QW0.2</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>Q3</td> <td>%QW0.3</td> <td>BOOL</td> </tr> <tr> <td></td> <td></td> <td>Q4</td> <td>%QW0.4</td> <td>BOOL</td> </tr> </tbody> </table> <p>= Create new variable = Map to existing variable</p>	Variable	Mapping	Channel	Address	Type	Inputs					IW0			%IW0	WORD	I0			%IX0.0	BOOL	I1			%IX0.1	BOOL	I2			%IW0.2	BOOL	I3			%IW0.3	BOOL	I4			%IW0.4	BOOL	I5			%IW0.5	BOOL	Outputs					QW0			%QW0.0	WORD	Application.GVL.blinput	Map to existing variable	Q0	%QW0.0	BOOL			Q1	%QW0.1	BOOL			Q2	%QW0.2	BOOL			Q3	%QW0.3	BOOL			Q4	%QW0.4	BOOL
Variable	Mapping	Channel	Address	Type																																																																													
Inputs																																																																																	
IW0			%IW0	WORD																																																																													
I0			%IX0.0	BOOL																																																																													
I1			%IX0.1	BOOL																																																																													
I2			%IW0.2	BOOL																																																																													
I3			%IW0.3	BOOL																																																																													
I4			%IW0.4	BOOL																																																																													
I5			%IW0.5	BOOL																																																																													
Outputs																																																																																	
QW0			%QW0.0	WORD																																																																													
Application.GVL.blinput	Map to existing variable	Q0	%QW0.0	BOOL																																																																													
		Q1	%QW0.1	BOOL																																																																													
		Q2	%QW0.2	BOOL																																																																													
		Q3	%QW0.3	BOOL																																																																													
		Q4	%QW0.4	BOOL																																																																													
3	<p>Для выбора переменных щёлкайте на полях Variable (Переменная) и назначайте новые или существующие переменные каналам:</p> <p>Для ввода нового имени дважды щёлкните на ячейке и введите имя переменной.</p> <p>Для выбора существующей переменной щёлкните на ячейке Variable и выберите кнопку</p> <p>Результат: откроется Input Assistant («Помощник» по вводу). Выберите из списка существующие Global Variables (Глобальные переменные) или Local Variables (Локальные переменные).</p>																																																																																

Подсказка: пиктограмма в графе **Mapping** (Назначение) показывает для каждой переменной, нужно ли её создавать заново или она может быть присвоена существующей переменной. Для переключения между двумя состояниями дважды щёлкните на пиктограмме графы **Mapping**.

Компиляция проекта

Проверка проекта

Чтобы проверить, правильно ли написана программа, выберите команду **Build** (Скомпилировать) в меню **Online** (Оперативный режим).

Задачи команды **Build**

Команда **Build** выполняет следующие задачи:

- проверка конфигурации аппаратных средств;
- проверка приложения;
- создание автономной генерации кода, в дальнейшем загружаемого в контроллер;
- отображение результата в окне **Messages** (Сообщения).

Сообщения, отображаемые в процессе выполнения команды **Build**

Все **ошибки и предупреждения**, обнаруженные в процессе выполнения команды **Build**, появляются в категории **Build** окна **Messages**.

Пример окна **Messages** после выполнения команды **Build**:

The screenshot shows the 'Messages' window with the following details:

Messages			
Build		1 error(s)	0 warning(s)
Description	Project	Object	Pos
-----Build started : Application : MyPLC.Application -----			
typify code ...			
✖ Types of channel and mapped variable do not match	M238_Project_1	IO [MyPLC: Em...	
Compile complete -- 1 errors, 0 warnings			

Для отображения более подробной информации об обнаруженных ошибках дважды щёлкните на соответствующем пункте списка – редактор откроется и отобразит ошибку в программе.

Конфигурирование устройств CANopen

3

Введение

В этом разделе описывается, как сконфигурировать устройства CANopen.

Содержание данной главы

Данная глава содержит следующие темы:

Тема	Страница
Конфигурирование интерфейса CANopen	34
Добавление устройства CANopen	39

Конфигурирование интерфейса CANopen

Введение

Для осуществления связи по протоколу CANopen вы должны сначала определить скорость передачи данных и сконфигурировать ведущее устройство CANopen. Ниже описывается, как определить скорость передачи данных в диалоговом окне конфигурирования CANopen и сконфигурировать менеджера CANopen, служащего ведущим устройством сети CANopen.

Прежде чем установить соединение CANopen, необходимо сначала физически подключить устройства CANopen к контроллеру. Для получения подробной информации об аппаратной части этого соединения обращайтесь к справочному руководству по аппаратной части данного контроллера.

ОСТОРОЖНО!

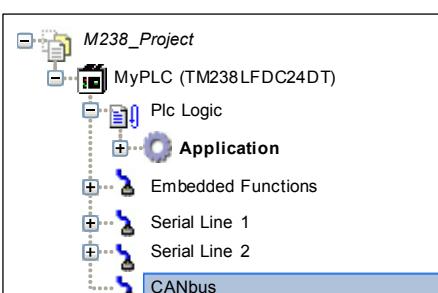
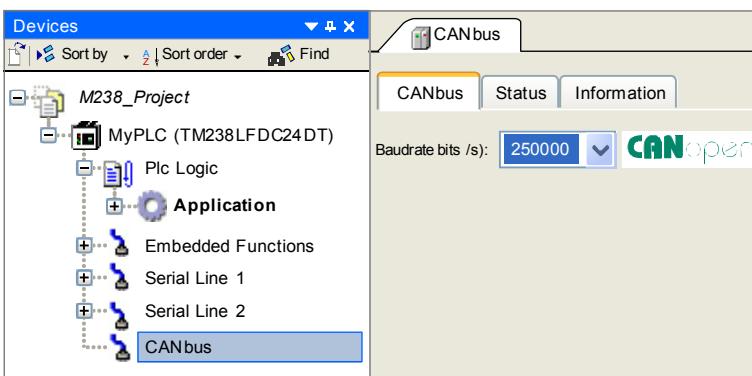
ПОТЕРЯ КОНТРОЛЯ

Не нажмайте кнопку **Update IO while in Stop** (Обновить входы/выходы во время останова) вкладки **PLC Settings** (Настройки ПЛК), если вы не хотите, чтобы некоторые устройства сети CANopen продолжали функционировать в то время как контроллер находится в режиме останова (STOP).

Невыполнение данных требований может привести к тяжелой травме вплоть до смертельного исхода или повреждению оборудования.

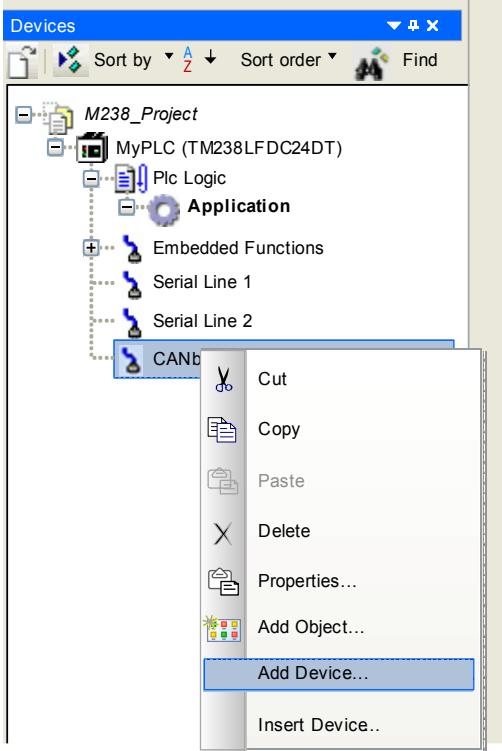
Настройка скорости передачи данных

Чтобы настроить скорость передачи данных в сети CANopen, сделайте следующее:

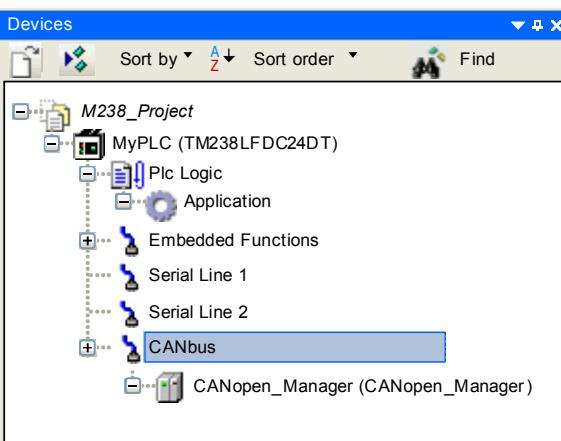
Шаг	Действие
1	<p>В окне Devices (Устройства) дважды щёлкните на пункте CANbus (Шина CAN), представленном в виде подэлемента контроллера M238, если последний оснащён интерфейсом CANopen.</p>  <p>Результат: справа появится диалоговое окно с вкладками для конфигурирования сети CANopen.</p>
2	<p>В первой вкладке, CANbus, настройте скорость передачи данных:</p>  <p>В списке Baudrate (Скорость) выберите скорость передачи данных соединения CANopen (по умолчанию 250 000 бит/с). Остальные вкладки содержат дополнительную информацию о соединении CANopen. За дополнительной информацией обращайтесь к интерактивной справке по системе CoDeSys.</p>

Создание и конфигурирование менеджера CANopen

Чтобы создать и сконфигурировать менеджер CANopen, служащий ведущим устройством CANopen и необходимый для коммуникаций CANopen, сделайте следующее:

Шаг	Действие
1	<p>В окне Devices (Устройства) щёлкните правой кнопкой на узле CANbus (Шина CAN), представленном в виде подузла контроллера M238, и выберите в контекстном меню команду Add Device... (Добавить устройство...):</p>  <p>Результат: откроется диалоговое окно Add Device.</p>

Шаг	Действие
2	<p>В диалоговом окне Add Device (Добавить устройство) выберите CANopenManager (Менеджер CANopen). Введите имя в поле Name (Имя) щёлкните на кнопке Add Device:</p>

Шаг	Действие
3	<p>Снова откроется диалоговое окно Add Device (Добавить устройство). Добавьте другое устройство или щёлкните на кнопке Close (Закрыть), чтобы закрыть диалоговое окно.</p> <p>Результат: узел CANbus (Шина CAN) окна Devices (Устройства) теперь содержит подузел CANopen Manager (Менеджер CANopen).</p> 
4	<p>Для конфигурирования устройства CANopen дважды щёлкните на соответствующем пункте в окне Devices.</p> <p>Результат: справа появится диалоговое окно для конфигурирования устройств CANopen, которое называется CANopen_Manager.</p>

Добавление устройства CANopen

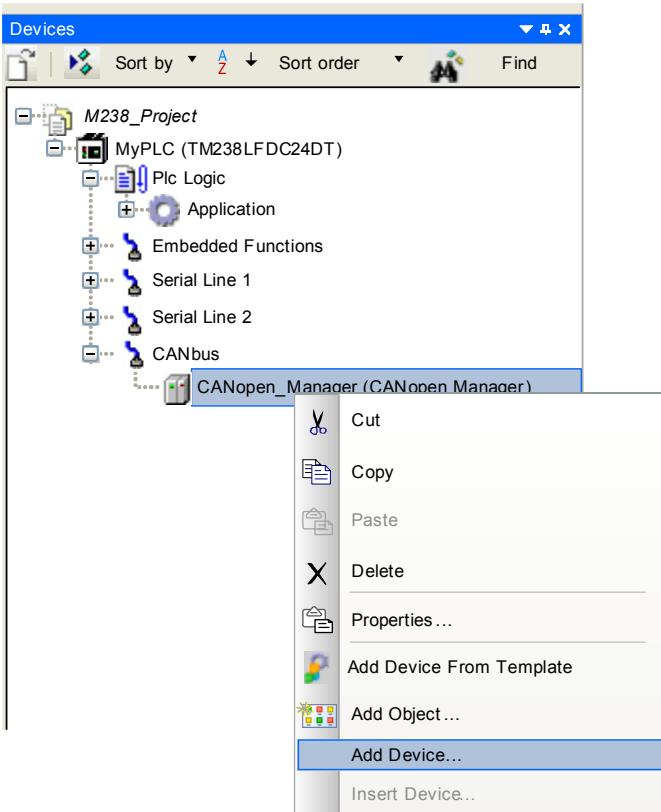
Введение

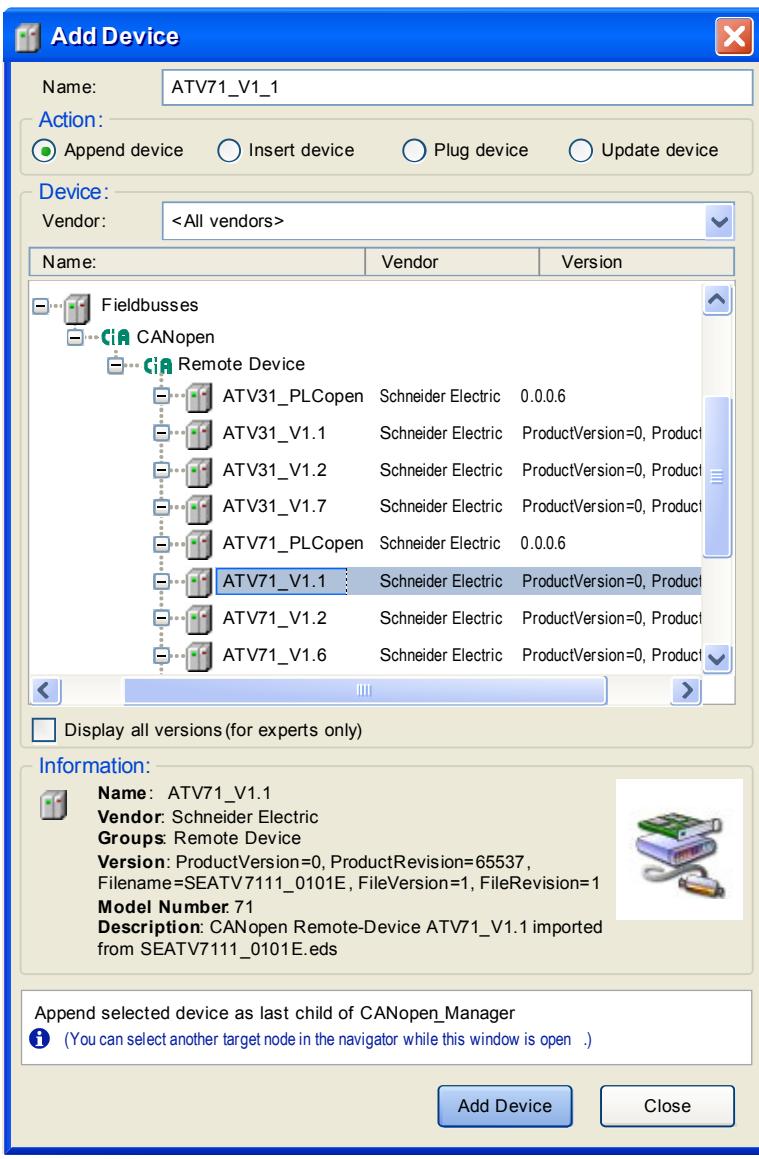
Вы можете добавлять в проект ведомые устройства CANopen двумя способами:

- выбрав устройство и сконфигурировав его от начала до конца;
- с помощью шаблона устройства, уже содержащего конфигурацию полевой шины CANopen, а также код (факультативно) и визуализацию (факультативно).

Добавление ведомого устройства CANopen

Чтобы добавить ведомое устройство CANopen, сделайте следующее:

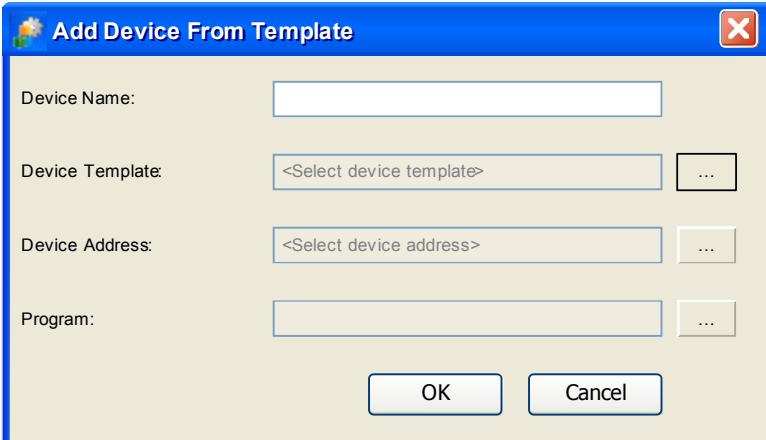
Шаг	Действие
1	<p>В окне Devices (Устройства) щёлкните правой кнопкой на CANopen Manager (Менеджер CANopen), теперь представленном в виде подузла узла CANbus (Шина CAN), и выберите в контекстном меню команду Add Device (Добавить устройство).</p>  <p>Результат: справа появится диалоговое окно Add Device для коммуникаций CANopen.</p>

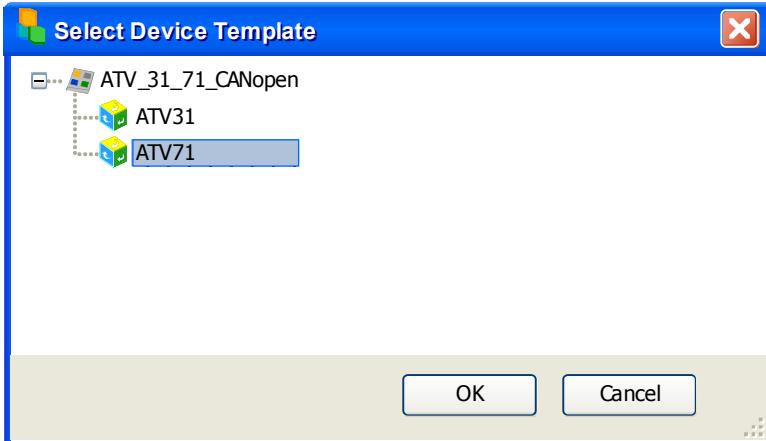
Шаг	Действие																											
2	<p>В диалоговом окне Add Device (Добавить устройство) выберите соответствующее устройство из списка и щёлкните на Add Device.</p>  <p>Name: ATV71_V1_1</p> <p>Action:</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Append device <input type="radio"/> Insert device <input type="radio"/> Plug device <input type="radio"/> Update device <p>Device:</p> <p>Vendor: <All vendors></p> <table border="1"> <thead> <tr> <th>Name</th> <th>Vendor</th> <th>Version</th> </tr> </thead> <tbody> <tr> <td>ATV31_PLCopen</td> <td>Schneider Electric</td> <td>0.0.0.6</td> </tr> <tr> <td>ATV31_V1.1</td> <td>Schneider Electric</td> <td>ProductVersion=0, Product</td> </tr> <tr> <td>ATV31_V1.2</td> <td>Schneider Electric</td> <td>ProductVersion=0, Product</td> </tr> <tr> <td>ATV31_V1.7</td> <td>Schneider Electric</td> <td>ProductVersion=0, Product</td> </tr> <tr> <td>ATV71_PLCopen</td> <td>Schneider Electric</td> <td>0.0.0.6</td> </tr> <tr> <td>ATV71_V1.1</td> <td>Schneider Electric</td> <td>ProductVersion=0, Product</td> </tr> <tr> <td>ATV71_V1.2</td> <td>Schneider Electric</td> <td>ProductVersion=0, Product</td> </tr> <tr> <td>ATV71_V1.6</td> <td>Schneider Electric</td> <td>ProductVersion=0, Product</td> </tr> </tbody> </table> <p><input type="checkbox"/> Display all versions (for experts only)</p> <p>Information:</p> <p>Name: ATV71_V1.1 Vendor: Schneider Electric Groups: Remote Device Version: ProductVersion=0, ProductRevision=65537, Filename=SEATV7111_0101E, FileVersion=1, FileRevision=1 Model Number: 71 Description: CANopen Remote-Device ATV71_V1.1 imported from SEATV7111_0101E.eds</p> <p>Append selected device as last child of CANopen_Manager <i>(You can select another target node in the navigator while this window is open ...)</i></p> <p>Add Device Close</p>	Name	Vendor	Version	ATV31_PLCopen	Schneider Electric	0.0.0.6	ATV31_V1.1	Schneider Electric	ProductVersion=0, Product	ATV31_V1.2	Schneider Electric	ProductVersion=0, Product	ATV31_V1.7	Schneider Electric	ProductVersion=0, Product	ATV71_PLCopen	Schneider Electric	0.0.0.6	ATV71_V1.1	Schneider Electric	ProductVersion=0, Product	ATV71_V1.2	Schneider Electric	ProductVersion=0, Product	ATV71_V1.6	Schneider Electric	ProductVersion=0, Product
Name	Vendor	Version																										
ATV31_PLCopen	Schneider Electric	0.0.0.6																										
ATV31_V1.1	Schneider Electric	ProductVersion=0, Product																										
ATV31_V1.2	Schneider Electric	ProductVersion=0, Product																										
ATV31_V1.7	Schneider Electric	ProductVersion=0, Product																										
ATV71_PLCopen	Schneider Electric	0.0.0.6																										
ATV71_V1.1	Schneider Electric	ProductVersion=0, Product																										
ATV71_V1.2	Schneider Electric	ProductVersion=0, Product																										
ATV71_V1.6	Schneider Electric	ProductVersion=0, Product																										

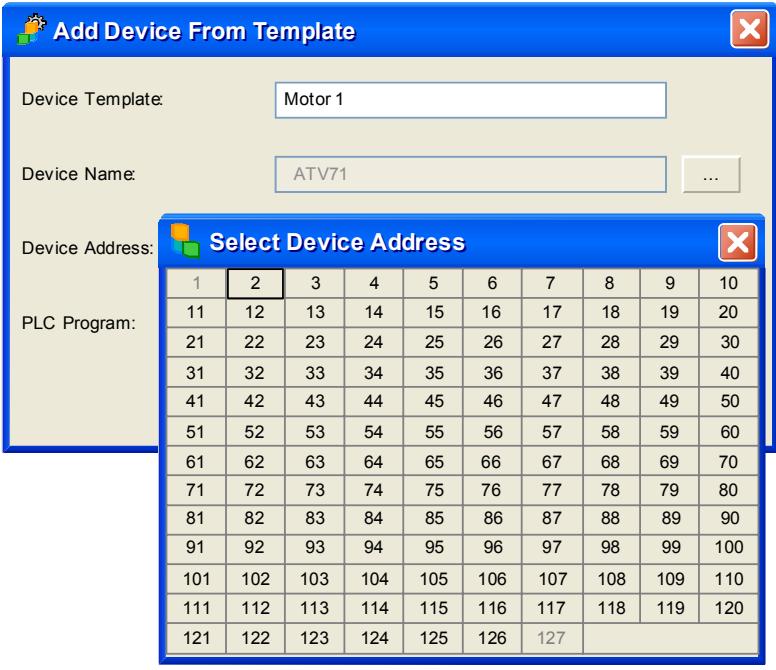
Шаг	Действие
3	<p>Закройте диалоговое окно Add Device (Добавить устройство).</p> <p>Результат: справа появится диалоговое окно с вкладками для конфигурирования выбранного устройства.</p> <p>Сконфигурируйте удалённое устройство CANopen как описано в интерактивной справке по системе CoDeSys.</p>

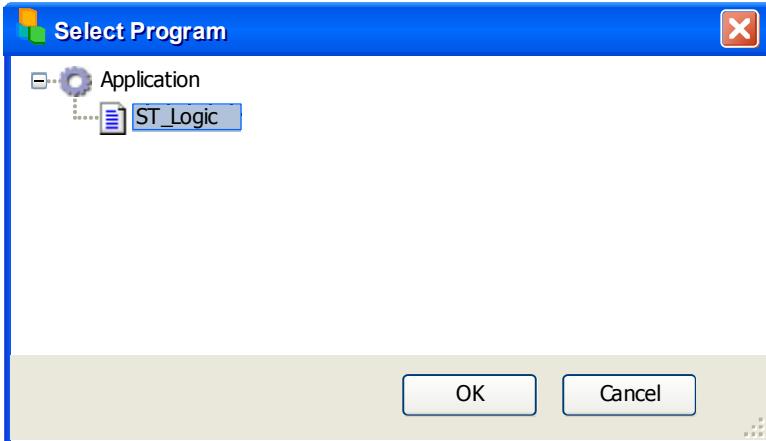
Добавление ведомого устройства CANopen с помощью шаблона устройства

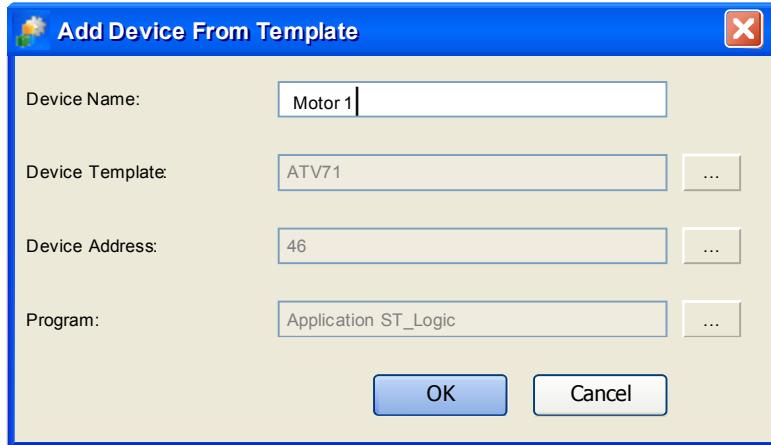
Чтобы добавить удалённое устройство CANopen с помощью шаблона устройства, сделайте следующее:

Шаг	Действие
1	<p>В окне Devices (Устройства) щёлкните правой кнопкой на CANopen Manager (Менеджер CANopen) и выберите в контекстном меню команду Add Device From Template (Добавить устройство из шаблона).</p> <p>Результат: появится диалоговое окно Add Device From Template.</p>  <p>В этом диалоговом окне определите ведомое устройство CANopen, его адрес и программный модуль, в котором будет написана программа контроллера.</p>
2	<p>В текстовом поле Device Name (Имя устройства) диалогового окна Add Device From Template введите имя устройства CANopen (например, <code>Motor1</code>).</p>

Шаг	Действие
3	<p>Чтобы определить шаблон используемого ведомого устройства CANopen, щёлкните на кнопке ... справа от поля Device Template (Шаблон устройства).</p> <p>Результат: появится диалоговое окно Select Device Template (Выбрать шаблон устройства).</p>  <p>В списке шаблонов устройства выберите шаблон, соответствующий используемому устройству, и щёлкните OK.</p>

Шаг	Действие
4	<p>Чтобы определить адрес используемого ведомого устройства CANopen, щёлкните на кнопке ... справа от поля Device Address (Адрес устройства).</p> <p>Результат: появится диалоговое окно Select Device Address (Выбрать адрес устройства).</p>  <p>Щёлкните на соответствующем поле между 1 и 127 для определения адреса ведомого устройства.</p>

Шаг	Действие
5	<p>Чтобы определить программный модуль, в котором будет написана программа контроллера, щёлкните на кнопке ... справа от поля PLC Program (Программа ПЛК).</p> <p>Результат: появится диалоговое окно Select PLC Program (Выбрать программу ПЛК).</p>  <p>Выберите в списке программный модуль, в котором вы хотите написать программу контроллера, и щёлкните OK.</p>

Шаг	Действие
6	<p>Откроется диалоговое окно Add Device From Template (Добавить устройство из шаблона). Теперь оно содержит всю необходимую информацию.</p>  <p>Щёлкните OK, чтобы закрыть диалоговое окно.</p> <p>Результат: устройство Altivar сконфигурировано в соответствии с шаблоном и полностью готово к работе.</p> <ul style="list-style-type: none"> Ведомое устройство CANopen добавлено в качестве подпункта пункта CANopen_Manager (Менеджер CANopen) в окне Devices (Устройства). Библиотека Toolbox (Инструментальные средства) добавлена к Library manager (Менеджер библиотек) проекта. Функциональный блок ATV_31_71_CANopen (расположенный в библиотеке Toolbox) добавлен к выбранному программному модулю.

Конфигурирование интерфейса сканера входов/выходов Modbus

4

Введение

В этом разделе описывается, как сконфигурировать интерфейс сканера входов/выходов Modbus.

Содержание данной главы

Данная глава содержит следующие темы:

Тема	Страница
Конфигурирование последовательного линейного интерфейса для сканера входов/выходов Modbus	48
Добавление устройства Modbus SL	53

Конфигурирование последовательного линейного интерфейса для сканера входов/выходов Modbus

Введение

Чтобы осуществлять связь по протоколу Modbus SL через сканер входов/выходов Modbus, необходимо сначала создать и сконфигурировать коммуникационный менеджер **Modbus IOScanner** (Сканер входов/выходов Modbus). Эта процедура описана в последующих параграфах.

Пример применения

Ниже приводится пример установления соединения сканера входов/выходов Modbus по последовательному каналу 1 (Serial Line 1) с ведомым устройством Altivar 31 с использованием сканера входов/выходов для автоматического обмена данными. Режим чтения/записи используется для чтения регистров ATV31 с 8601 DriveCom Control Word au 8603 DriveCom Status Word и записи в регистр ATV31 8601 DriveCom Control Word. Устройство реализовано с помощью описания типового ведомого устройства Modbus Slave в ПО SoMachine.

В нашем примере применения это ведомое устройство Altivar 31 названо Motor 2.

В последующих разделах описана процедура создания этого соединения.

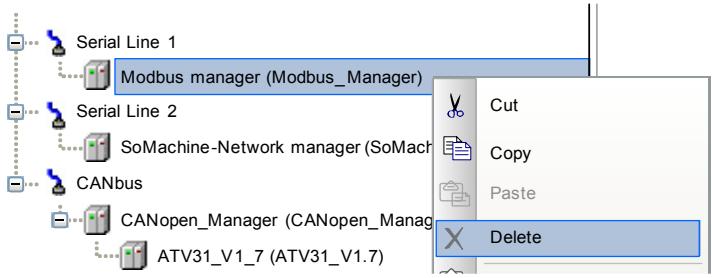
Удаление менеджера Modbus

По умолчанию, устройство TM238LFDC24DT предварительно сконфигурировано следующим образом:

- Serial Line 1 (Последовательный канал 1): **Modbus manager** (менеджер Modbus);
- Serial Line 2 (Последовательный канал 2): **SoMachine - Network manager** (менеджер сети).

Чтобы использовать сканер входов/выходов Modbus, удалите предварительно сконфигурированный менеджер Modbus следующим образом:

Шаг	Действие
1	Выберите элемент Modbus Manager (Менеджер Modbus), расположенный ниже узла Serial Line 1 (Последовательный канал 1) в окне Devices (Устройства).
2	Щёлкните правой кнопкой на элементе Modbus Manager и выберите в контекстном меню Delete (Удалить).

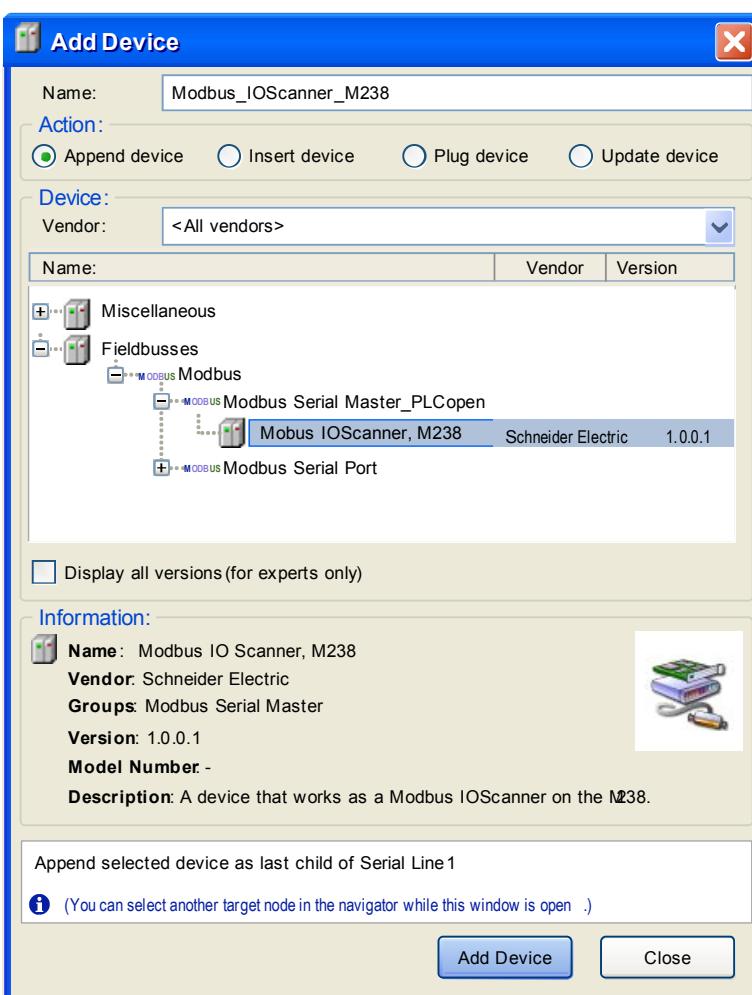


Примечание: используйте параметры по умолчанию последовательного канала, как они определены в окне свойств узла последовательного канала.

Создание сканера входов/выходов Modbus

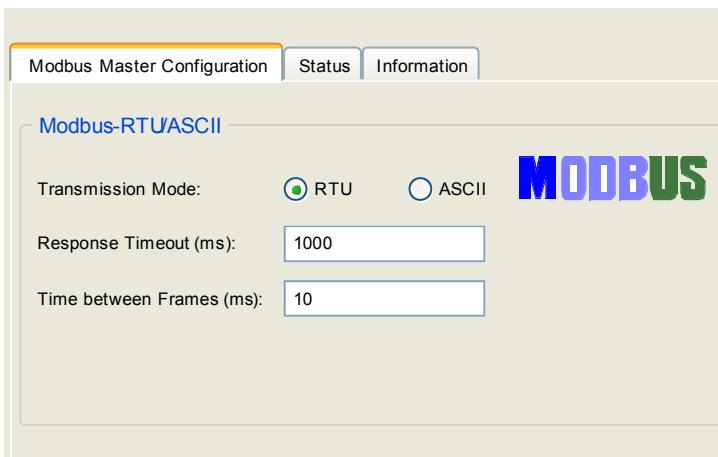
Чтобы создать сканер входов/выходов Modbus, сделайте следующее:

Шаг	Действие
1	<p>В окне Devices (Устройства) щёлкните правой кнопкой на пункте Serial Line1 (Последовательный канал 1), представленном в виде подэлемента контроллера M238, и выберите в контекстном меню команду Add Device (Добавить устройство).</p> A screenshot of the SIMATIC Manager software interface. The main window shows a project structure with 'M238_Project' expanded, revealing 'MyPLC (TM238LFDC24 DT)', 'Pic Logic', and 'Application'. Under 'Application', 'Embedded Functions', 'Serial Line 1', 'Serial Line 2', and 'CANbus' are listed. A context menu is open over the 'Serial Line 1' node, with the 'Add Device...' option highlighted. The menu also includes other options like Cut, Copy, Paste, Delete, Properties..., Add Object..., and Insert Device... <p>Результат: справа появится диалоговое окно Add Device для имеющихся коммуникационных менеджеров.</p>

Шаг	Действие
2	<p>В диалоговом окне Add Device (Добавить устройство) выберите Modbus IOScanner (Сканер входов/выходов Modbus.) Для подтверждения выбора щёлкните на кнопке Add Device:</p>  <p>Результат: пункт Serial Line 1 (Последовательный канал 1) окна Devices (Устройства) теперь содержит пункт коммуникационного менеджера Modbus IOScanner.</p>

Конфигурирование ведущего устройства Modbus SL

Чтобы сконфигурировать ведущее устройство Modbus SL, сделайте следующее:

Шаг	Действие
1	<p>Дважды щёлкните на пункте Modbus IOScanner (Сканер входов/выходов Modbus) в окне Devices (Устройства), чтобы открыть диалоговое окно для конфигурирования коммуникаций Modbus SL. Это диалоговое окно включает в себя три вкладки.</p> <p>Прежде всего, настройте параметры передачи во вкладке Modbus Master Configuration (Конфигурация ведущего устройства Modbus).</p>  <p>В поле Transmission Mode (Способ передачи) выберите подходящий способ передачи для коммуникаций Modbus. В поле Response Timeout (ms) (Тайм-аут ответа) определите выдержку времени с целью предотвращения слишком долгого ожидания ответов системой. Настройте параметр Time between Frames (ms) (Интервал между фреймами) – временной интервал, который должен быть между фреймами Modbus для разделения конца предыдущего фрейма и начала следующего фрейма.</p> <p>Примечание: параметры последовательного канала определяются в окне свойств узла последовательного канала.</p>
2	Для получения информации о текущем состоянии выберите вкладку Status (Статус).
3	Для получения общей информации о подсоединенном к последовательному порту устройству щёлкните на вкладке Information (Информация).

Добавление устройства Modbus SL

Введение

Вы можете добавлять в проект устройства Modbus SL для сканера входов/выходов двумя способами:

- выбрав устройство и сконфигурировав его от начала до конца;
- с помощью шаблона устройства, уже содержащего конфигурацию полевой шины сканера входов/выходов Modbus, а также код (факультативно) и визуализацию (факультативно).

Конфигурирование устройства ввода/вывода Modbus

Чтобы сконфигурировать Altivar 31 как ведомое устройство Modbus SL, сделайте следующее:

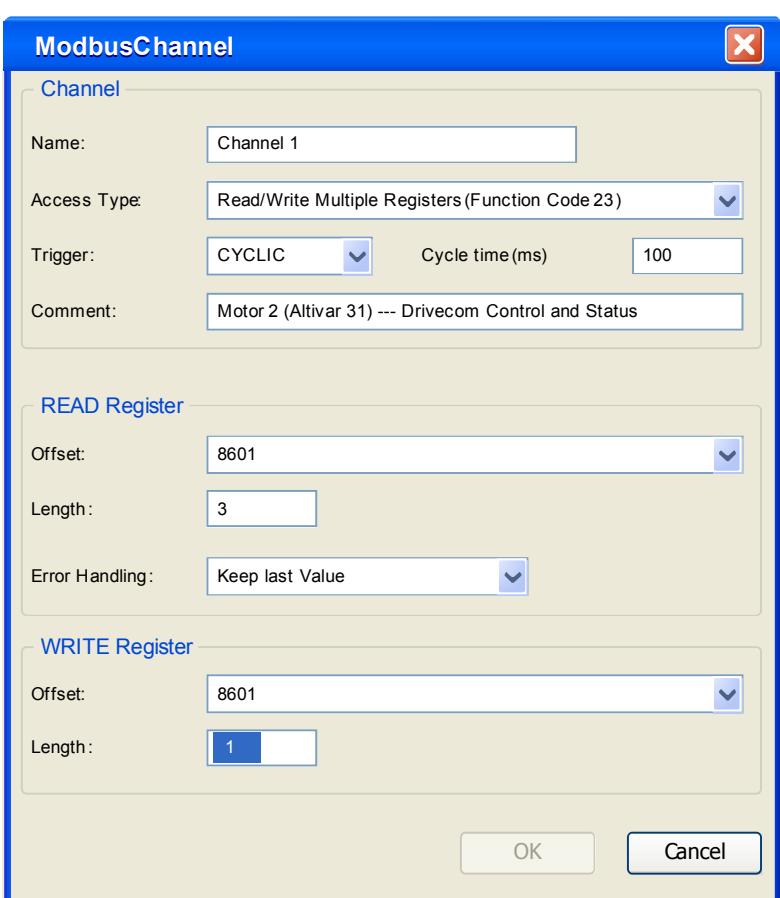
Шаг	Действие
1	<p>Щёлкните правой кнопкой на подпункте Modbus IOScanner (Сканер входов/выходов Modbus) в окне Devices (Устройства) и выберите команду Add Device (Добавить устройство) для определения устройства ввода/вывода.</p> <p>В диалоговом окне Add Device выберите устройство в списке и щёлкните на кнопке Add Device. В данном примере мы используем обобщённый тип, именуемый Modbus Slave (Ведомое устройство Modbus).</p>

Шаг	Действие
2	Закройте диалоговое окно Add Device (Добавить устройство). Результат: пункт Modbus IOScanner в окне Devices (Устройства) теперь содержит в качестве подпункта ваше устройство ввода/вывода Modbus.
3	В окне Devices дважды щёлкните на подузле устройства узла Modbus. Результат: появится диалоговое окно Modbus Slave Configuration (Конфигурация ведомого устройства Modbus).
4	Во вкладке Modbus Slave Configuration выполните следующие настройки:  <p>Ведите Slave Address (Адрес ведомого устройства) для Altivar 31 (ведомое устройство Modbus). Определите Response Timeout (Тайм-аут ответа) для предотвращения слишком долгого ожидания ответов устройством.</p>

Конфигурирование каналов ведомого устройства Modbus

Чтобы сконфигурировать каналы Altivar 31 (ведомое устройство Modbus SL), сделайте следующее:

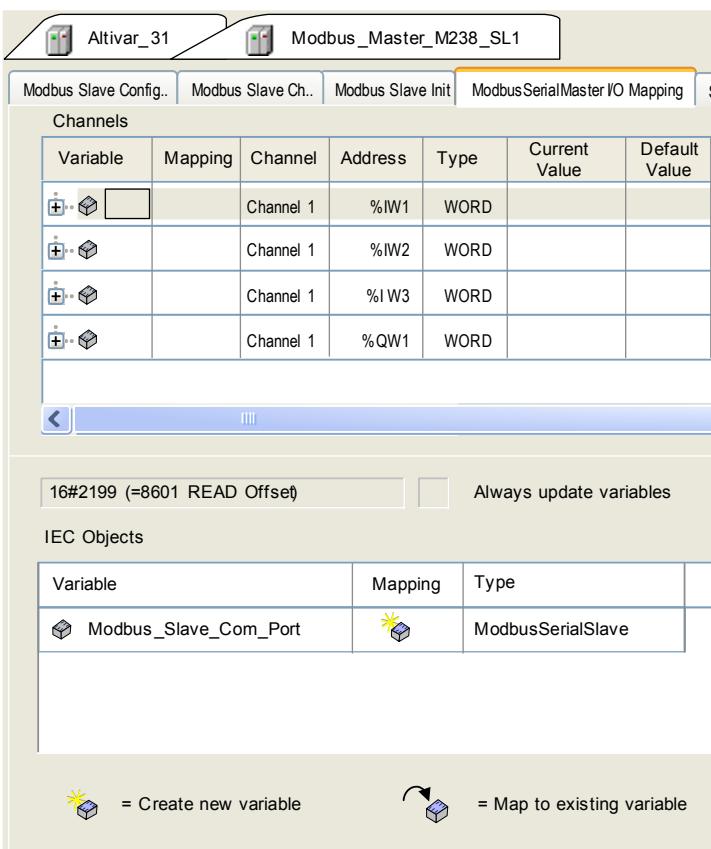
Шаг	Действие
1	<p>Откройте вкладку Modbus Slave Channel (Канал ведомого устройства Modbus) и щёлкните на кнопке Add Channel (Добавить канал).</p> <p>Результат: появится диалоговое окно ModbusChannel (Канал Modbus).</p>

Шаг	Действие
2	<p>В диалоговом окне ModbusChannel (Канал Modbus) выполните следующие настройки:</p>  <p>Общие настройки канала в разделе Channel (Канал) диалогового окна: В поле Name (Имя) введите имя настраиваемого канала. Определите Access Type (Тип доступа), выбрав соответствующий пункт в списке. В нашем примере мы хотим связать с этим каналом запрос Modbus на чтение/запись. Соответственно, мы выбираем пункт Read/Write Multiple Registers (Чтение/запись нескольких регистров) (Function Code 23 (Функциональный код 23)). Для периодических чтения и записи на этом канале настройте параметр Trigger (Запуск) на CYCLIC (Циклический) и определите Cycle Time (Длительность цикла). Введите комментарий в поле Comment (Комментарий) для лучшего определения канала, указав, например, ведомое устройство Modbus (Altivar 31) и тип адреса.</p>

Шаг	Действие																
3	<p>Определение регистров READ (Чтение): Введите Offset (Смещение) первого регистра, считываемого на ведомом устройстве. В нашем примере это регистр 8601 DriveCom Control Word.</p> <p>Введите Length (Длину) таблицы считываемых регистров (в нашем примере регистры 8601, 8602 и 8603). Определите процедуру, реализуемую в случае ошибки связи, путём настройки параметра Error Handling (Обработка ошибок) на Keep last value (Сохранить последнее значение) (ведущее устройство будет сохранять последнее полученное от ведомого устройства значение вплоть до восстановления связи) или на Set to ZERO (Установить на нуль) (ведущее устройство будет обнулять значение вплоть до восстановления связи).</p>																
4	<p>Определение регистров WRITE (Запись): Введите Offset первого регистра, записываемого на ведомом устройстве. В нашем примере это регистр 8601 DriveCom Control Word, доступный для чтения и записи. Введите Length таблицы записываемых регистров (в нашем примере только регистр 8601).</p>																
5	<p>Щёлкните OK, чтобы добавить новый канал к списку Modbus Slave Channel (Канал ведомого устройства Modbus).</p> <p>Результат: диалоговое окно Modbus Slave Channel теперь содержит новый канал:</p>  <table border="1"> <thead> <tr> <th>Name</th> <th>Access Type</th> <th>Trigger</th> <th>READ Offset</th> <th>Length</th> <th>Error Handling</th> <th>Write Offset</th> <th>L</th> </tr> </thead> <tbody> <tr> <td>Ch...</td> <td>ReadWrite...</td> <td>CYCLIC,...</td> <td>16#2199</td> <td>3</td> <td>Keep last Value</td> <td>16#2199</td> <td>1</td> </tr> </tbody> </table>	Name	Access Type	Trigger	READ Offset	Length	Error Handling	Write Offset	L	Ch...	ReadWrite...	CYCLIC,...	16#2199	3	Keep last Value	16#2199	1
Name	Access Type	Trigger	READ Offset	Length	Error Handling	Write Offset	L										
Ch...	ReadWrite...	CYCLIC,...	16#2199	3	Keep last Value	16#2199	1										

Назначение входов/выходов

Чтобы назначить переменные каналам ввода/вывода, сделайте следующее:

Шаг	Действие						
1	<p>Откройте вкладку ModbusSerialMaster I/O Mapping (Назначение входов/выходов ведущего устройства последовательного канала Modbus) устройства Altivar 31:</p>  <p>16#2199 (=8601 READ Offset) <input type="checkbox"/> Always update variables</p> <p>IEC Objects</p> <table border="1"> <thead> <tr> <th>Variable</th> <th>Mapping</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Modbus_Slave_Com_Port</td> <td></td> <td>ModbusSerialSlave</td> </tr> </tbody> </table> <p> = Create new variable = Map to existing variable</p> <p>Примечание: запрос на чтение/запись сначала записывает новые значения в регистры ведомого устройства, затем читает считываемые регистры ведомого устройства. В нашем примере регистр 8601 сначала записывается, затем считывается (это позволяет проверить, получено ли новое значение). По этой причине не используйте одну и ту же переменную для чтения и записи регистра 8601.</p>	Variable	Mapping	Type	Modbus_Slave_Com_Port		ModbusSerialSlave
Variable	Mapping	Type					
Modbus_Slave_Com_Port		ModbusSerialSlave					

Шаг	Действие
2	<p>Назначьте переменные данным каналов.</p> <p>Чтобы назначить новую переменную, введите её в поле Variable (Переменная) и щёлкните на кнопке Create new variable (Создать новую переменную) в нижней части диалогового окна.</p> <p>Чтобы назначить существующую переменную, выберите существующую переменную в списке, который можно открыть с помощью кнопки</p> <p>Адреса Modbus устройства Altivar 31 содержатся в графе Description (Описание).</p> <p>Вы можете назначить переменные командному слову ATV 31 (8601 – чтение и запись) и слову состояния ATV 31 (8603 – чтение) или различным битам слов.</p>

Конфигурирование устройства Modbus SL с помощью шаблона устройства

Чтобы сконфигурировать устройство Modbus SL с помощью шаблона устройства, выполните действия, описанные в отношении устройств CANopen (см. стр. 42).

Связь с терминалом ЧМИ

5

Введение

В этой главе описаны действия, необходимые для установления соединения с терминалом человека-машинного интерфейса (ЧМИ). Терминал ЧМИ подключается к контроллеру M238 по последовательному каналу (последовательный канал 2 контроллера M238).

Содержание данной главы

Данная глава содержит следующие темы:

Тема	Страница
Публикация переменных связи	62
Конфигурирование устройства ЧМИ в проекте	65
Конфигурирование связи с терминалами ЧМИ	67
Подписка переменных для терминалов ЧМИ	68

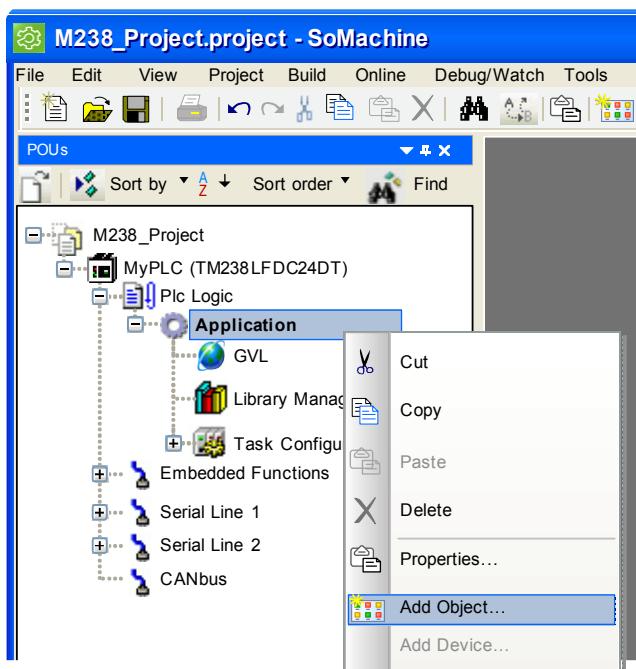
Публикация переменных связи

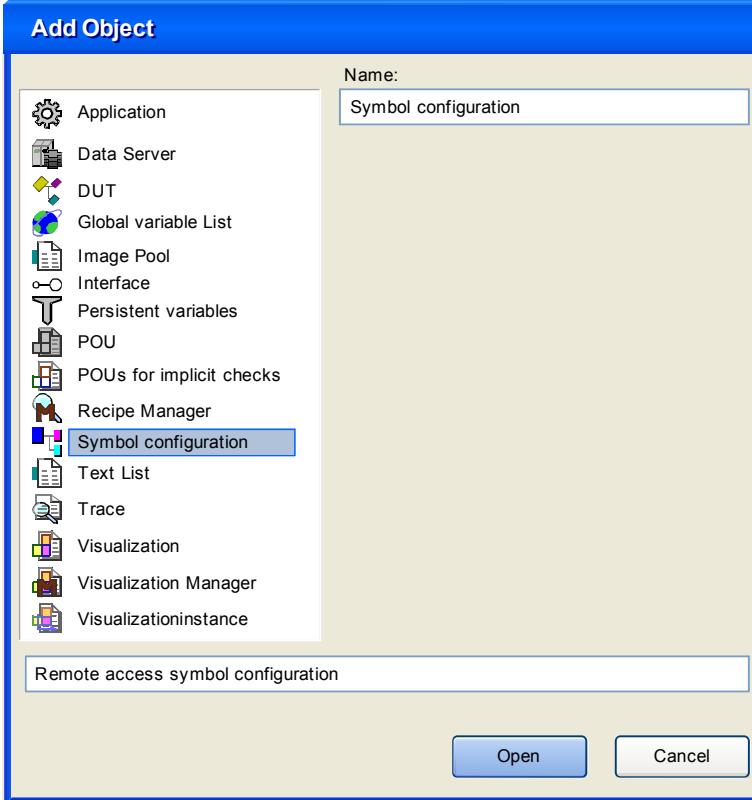
Введение

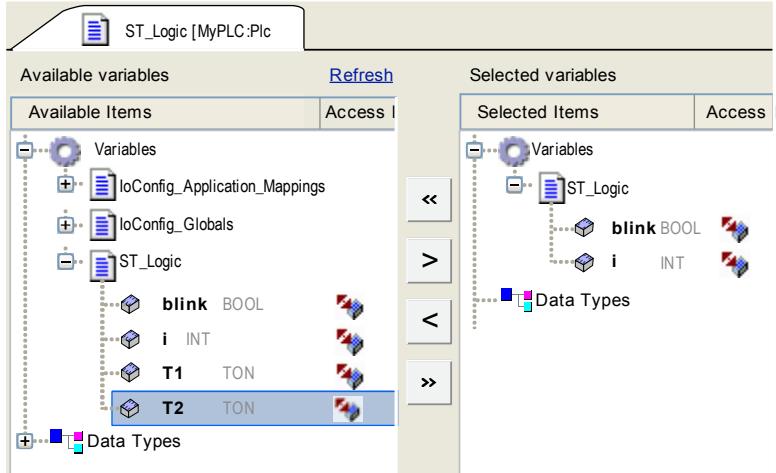
Чтобы выдать переменные на терминалы ЧМИ, необходимо прежде всего опубликовать все переменные, которые должны быть для них доступны, преобразовав их в символы. Ниже описана процедура преобразования переменных в символы.

Процедура

Чтобы преобразовать элементарные переменные в символы для терминалом ЧМИ, сделайте следующее:

Шаг	Действие
1	<p>Щёлкните правой кнопкой на узле Application (Приложение) своего проекта и выберите команду Add Object... (Добавить объект...):</p>  <p>Результат: появится диалоговое окно Add Object.</p>

Шаг	Действие
2	<p>В диалоговом окне Add Object (Добавить объект) выберите пункт Symbol Configuration (Конфигурация символа) (в списке слева) и щёлкните на Open (Открыть):</p>  <p>Результат: справа появится редактор Symbol Configuration, элемент Symbol Configuration будет добавлен в окне Devices (Устройства) под узлом Application (Приложение).</p>

Шаг	Действие
3	<p>В редакторе Symbol Configuration (Конфигурация символа) выберите элементарные переменные, которые должны быть доступны для терминала ЧМИ:</p>  <p>The screenshot shows the 'Symbol Configuration' dialog with two main panes: 'Available variables' and 'Selected variables'. In the 'Available variables' pane, under 'Available Items', there are several items: 'Variables' (expanded), 'IoConfig_Application_Mappings', 'IoConfig_Globals', 'ST_Logic' (expanded), 'blink' (BOOL), 'i' (INT), 'T1' (TON), and 'T2' (TON). The 'T2' item is highlighted with a blue selection bar. Below these are 'Data Types'. In the 'Selected variables' pane, under 'Selected Items', the 'ST_Logic' item is expanded, showing 'blink' (BOOL) and 'i' (INT). Between the two panes are four buttons: '<<', '>', '<', and '>>'. A 'Refresh' button is located at the top of the 'Available variables' pane.</p> <p>В левой части окна редактора показаны переменные, имеющиеся в настоящий момент в приложении. Для обновления окна щёлкните на кнопке Refresh (Обновить). Чтобы видеть все переменные, откройте подэлементы, щёлкнув на знаках «+». Чтобы выдать переменные на терминалы ЧМИ, выберите переменные индивидуально или целыми узлами в списке слева, затем щёлкните на кнопке со стрелкой в центре для их копирования в правую часть, откуда они выдаются на терминалы ЧМИ.</p> <p>Вы также можете назначить права доступа для чтения/записи индивидуально каждой переменной в графе Access Rights (Права доступа).</p> <p>Примечание: в версии 1.0 ПО SoMachine учитываются только переменные элементарных данных.</p>
4	Для подтверждения настроек выйдите из редактора, щёлкнув на другом окне SoMachine.

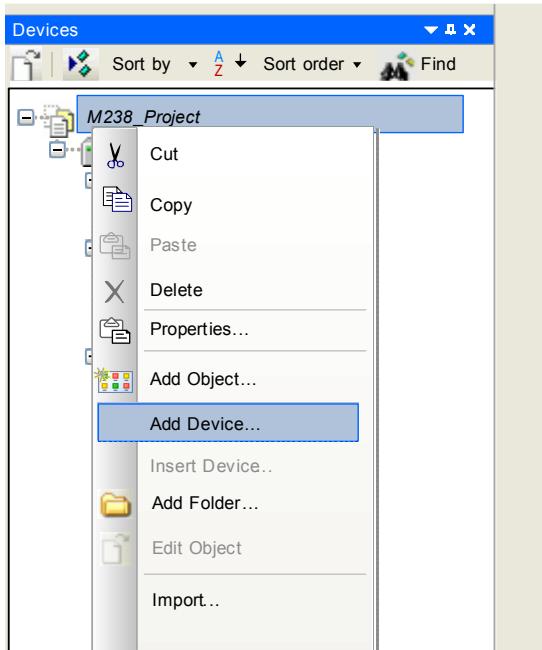
Конфигурирование устройства ЧМИ в проекте

Введение

Ниже описана процедура добавления устройства ЧМИ в проект.

Процедура

Чтобы добавить устройство ЧМИ в свой проект, сделайте следующее:

Шаг	Действие
1	Щёлкните правой кнопкой на пункте проекта в окне Devices (Устройства).
2	Выберите в контекстном меню команду Add Device... (Добавить устройство...).  <p>Результат: появится диалоговое окно Add Device.</p>

Шаг	Действие
3	<p>В диалоговом окне Add Device (Добавить устройство) выберите устройство, которое хотите добавить:</p>

Чтобы установить соединение с терминалом ЧМИ Magelis, выберите в списке ЧМИ **Vijeo-Designer** и щёлкните на кнопке **Add Device**.

Результат: новое устройство будет добавлено в проект в окне **Devices**, будет запущено конфигурирование ЧМИ на основе программы Vijeo-Designer. В программе Vijeo-Designer автоматически создаётся проект под тем же именем, что и проект SoMachine.

Конфигурирование связи с терминалами ЧМИ

Введение

Редактор конфигурации ЧМИ на основе программы Vijeo-Designer автоматически становится доступен с момента, когда устройство Magelis сконфигурировано в проекте. Для доступа к редактору конфигурации дважды щёлкните на узле приложения ЧМИ в окне **Devices** (Устройства) ПО SoMachine.

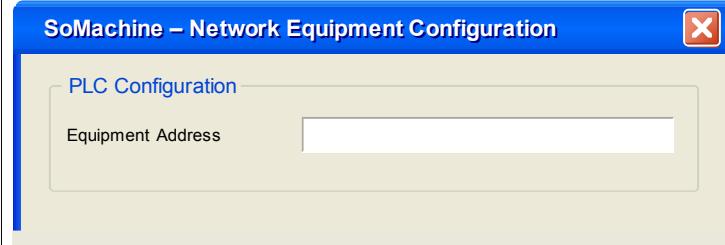
Подписка переменных для терминалов ЧМИ

Введение

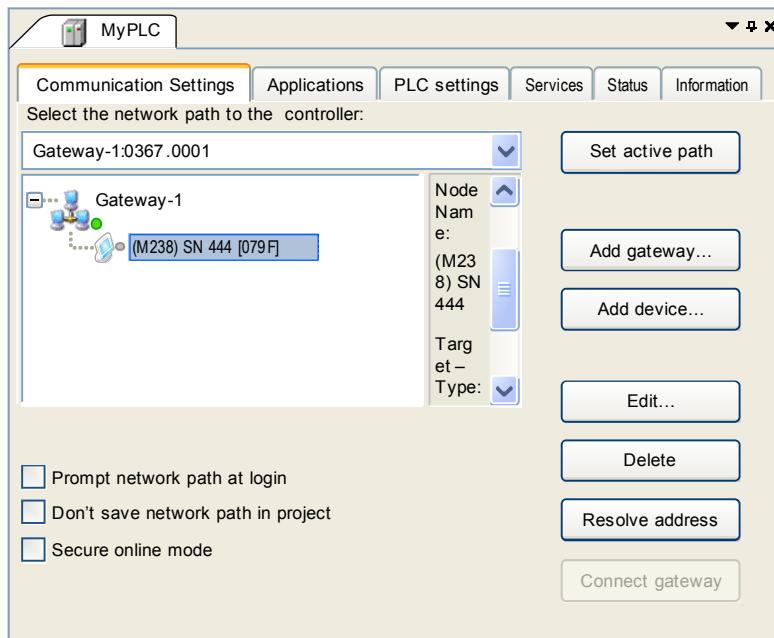
Для того, чтобы каждое устройство ЧМИ Magelis могло использовать некоторые из опубликованных для него элементарных переменных контроллера (подробнее см. в разделе Публикация переменных связи для ЧМИ, стр. 62), необходимо осуществить подписку на требуемые переменные. Ниже описана процедура подписки переменных для терминалов ЧМИ в программном обеспечении для программирования Vjeo-Designer.

Предварительные условия для устройств, не относящихся к устройствам XBT GC

Ниже описан процесс конфигурирования подузла **Equipment** (Устройство), который необходимо выполнить в первую очередь для всех устройств, не являющихся XBT GC:

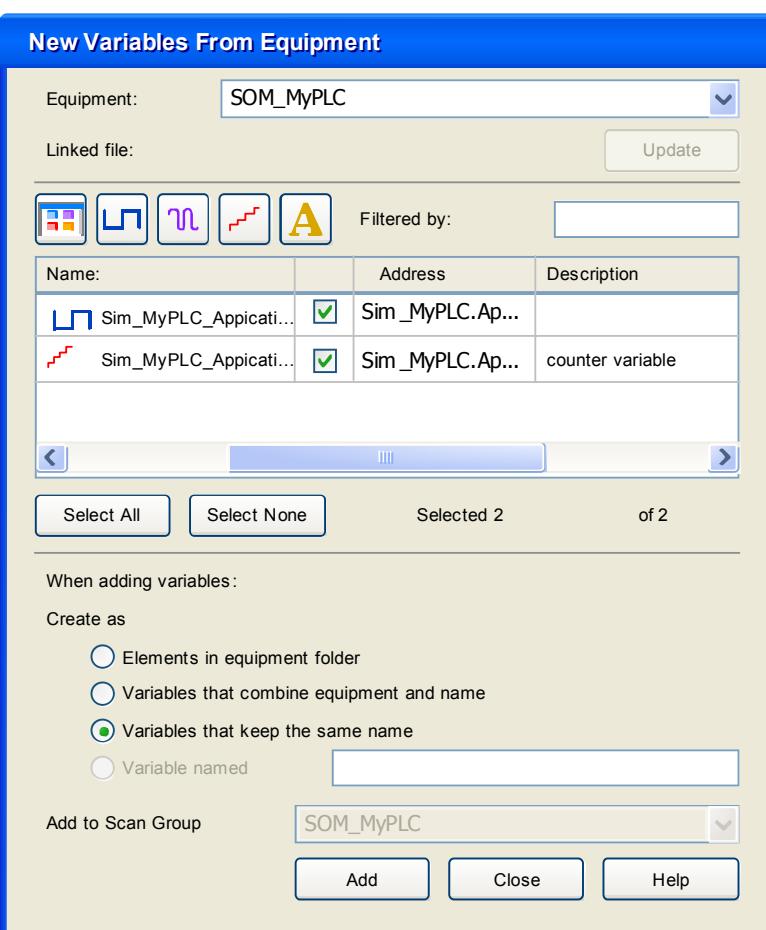
Шаг	Действие
1	<p>В программе Vjeo-Designer дважды щёлкните на элементе IO Manager (Менеджер ввода/вывода) в окне Navigator (Навигатор) (слева) и выберите в контекстном меню команду Configure (Сконфигурировать).</p>  <p>Результат: появится диалоговое окно Network Equipment Configuration (Конфигурация сетевого устройства).</p>
2	<p>В текстовом поле Equipment Address (Адрес устройства) диалогового окна Network Equipment Configuration введите адрес вашего контроллера, например: (M238) SN 444.</p> 

Примечание: вы можете непосредственно скопировать имя контроллера из диалогового окна **Communication Settings** (Настройки связи) с помощью кнопки **Edit** (Редактирование).



Процедура

Чтобы выполнить подпись переменных (символов) для терминала ЧМИ, сделайте следующее:

Шаг	Действие									
1	<p>В программе Vigeo-Designer щёлкните правой кнопкой на элементе Variables (Переменные) в расположеннном слева окне Navigator (Навигатор) и выберите в контекстном меню команду New Variables from Equipment... (Новые переменные от устройства...), чтобы открыть диалоговое окно New Variables From Equipment.</p>  <p>The dialog box displays the following information:</p> <ul style="list-style-type: none"> Equipment: SOM_MyPLC Linked file: (dropdown menu) Filtered by: (text input field) Table: <table border="1"> <thead> <tr> <th>Name:</th> <th>Address</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Sim_MyPLC_Applicati...</td> <td><input checked="" type="checkbox"/></td> <td>Sim_MyPLC.Ap...</td> </tr> <tr> <td>Sim_MyPLC_Applicati...</td> <td><input checked="" type="checkbox"/></td> <td>Sim_MyPLC.Ap... counter variable</td> </tr> </tbody> </table> Buttons: Select All, Select None, Selected 2 of 2 When adding variables: <ul style="list-style-type: none"> Create as: <input type="radio"/> Elements in equipment folder <input type="radio"/> Variables that combine equipment and name <input checked="" type="radio"/> Variables that keep the same name <input type="radio"/> Variable named: (text input field) Add to Scan Group: SOM_MyPLC Buttons: Add, Close, Help 	Name:	Address	Description	Sim_MyPLC_Applicati...	<input checked="" type="checkbox"/>	Sim_MyPLC.Ap...	Sim_MyPLC_Applicati...	<input checked="" type="checkbox"/>	Sim_MyPLC.Ap... counter variable
Name:	Address	Description								
Sim_MyPLC_Applicati...	<input checked="" type="checkbox"/>	Sim_MyPLC.Ap...								
Sim_MyPLC_Applicati...	<input checked="" type="checkbox"/>	Sim_MyPLC.Ap... counter variable								

Это диалоговое окно содержит все переменные контроллера, которые были преобразованы в символы в программе SoMachine и на которые теперь могут быть подписаны терминалы ЧМИ.

Элементы графы **Address** (Адрес) показывают местоположение каждой переменной в SoMachine.

Примечание: рекомендуется выбирать опцию **Variables that keep the same name** (Переменные, сохраняющие то же имя).

Шаг	Действие
2	Выберите в списке устройство публикации.
3	<p>Выберите в списке символы (переменные), на которые вы хотите подписать терминал.</p> <p>В поле Create as (Создать как) выберите соответствующую опцию, чтобы сохранить прежнее имя (как в поле управления) или изменить его в соответствии с предложенными вариантами.</p> <p>В списке Add to Scan Group (Добавить к группе сканирования) выберите соответствующий пункт.</p> <p>Щёлкните на Add (Добавить), чтобы закрыть диалоговое окно.</p> <p>Результат: выбранные вами символы являются теперь переменными, доступными для терминала ЧМИ.</p>

Примечание: вы можете повлиять на имена переменных в терминале ЧМИ путём выбора опции в разделе **When adding variables: Create as** (При добавлении переменных: Создать как) диалогового окна **New Variables from Equipment** (Новые переменные от устройства):

- **Elements in Equipment folder** (Элементы в папке устройства)
`<nomd'équipement>_<nomd'application>_<objet>_symbol`
 Located in the variable folder (Размещение в папке переменных)
`<nomdupilote>`
- **Variables that combine equipment and name** (Переменные, сочетающие устройство и имя)
`<nomdupilote>_<nomd'équipement>_<nomd'application>_<objet>_symbol`
- **Variables that keep the same name** (Переменные, сохраняющие то же имя)
`<nomd'application>_<objet>_symbol`

Примечание: имена переменных в терминале ЧМИ ограничены 32 знаками.

Подключение и загрузка

6

Введение

В этом разделе дано краткое описание процедуры установления программного соединения между ПК с запущенной программой SoMachine и контроллером с целью передачи программы в контроллер.

Перед тем как установить программное соединение, вы должны сначала физически подключить контроллер к компьютеру. Подробная информация об этом аппаратном подключении дана в справочном руководстве по аппаратной части контроллера.

Содержание данной главы

Данная глава содержит следующие темы:

Тема	Страница
Подключение к контроллеру	74
Загрузка в контроллер	79
Загрузка приложений управления и ЧМИ посредством однократного подключения	80
Пуск и остановка контроллера	81
Изменения, вносимые в контроллер в оперативном режиме	82
Имитация контроллера	84

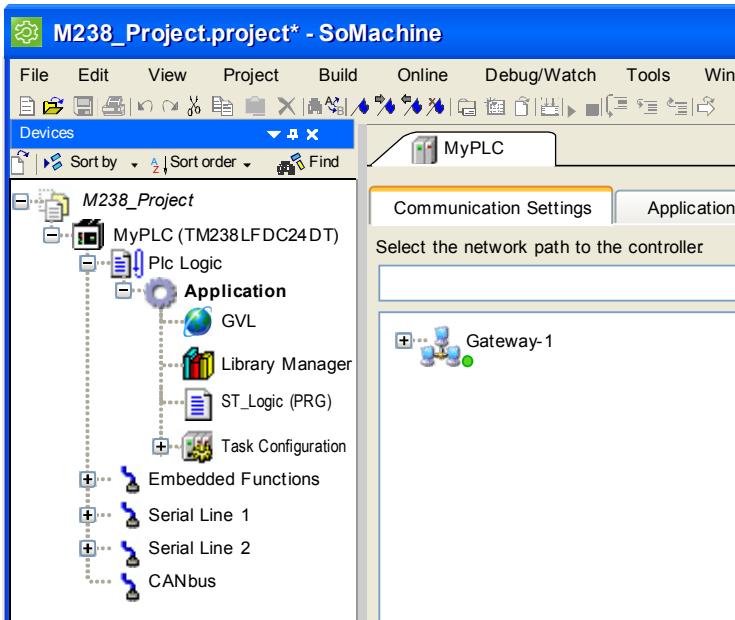
Подключение к контроллеру

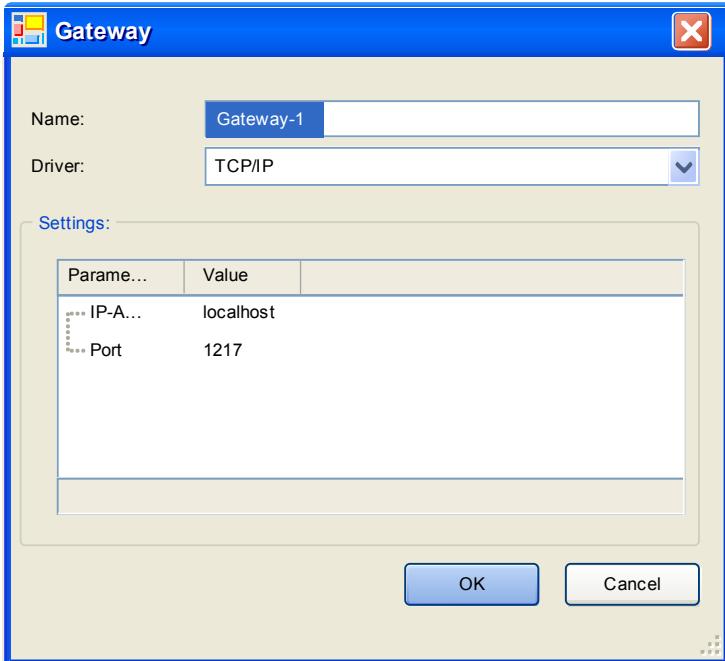
Введение

Ниже описана процедура подключения компьютера к контроллеру для загрузки программы в контроллер.

Конфигурирование соединения с контроллером

Чтобы сконфигурировать соединение с контроллером, сделайте следующее:

Шаг	Действие
1	<p>В окне Devices (Устройства) дважды щёлкните на узле контроллера.</p> <p>Результат: появится диалоговое окно с вкладками для конфигурирования контроллера. Первая вкладка служит для настройки параметров связи:</p> 

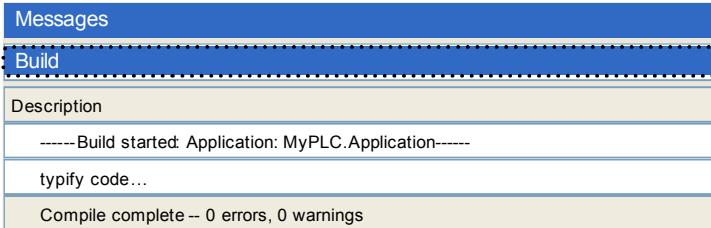
Шаг	Действие
2	<p>Для реализации подключения к контроллеру необходим шлюз. В случае отсутствия шлюза щёлкните на кнопке Add gateway... (Добавить шлюз...) в правой части диалогового окна.</p> <p>Результат: появится диалоговое окно Gateway (Шлюз):</p> 
3	<p>В диалоговом окне Gateway сделайте следующее:</p> <p>В текстовом поле Name (Имя) введите имя шлюза.</p> <p>В списке Driver (Драйвер) выберите пункт TCP/IP.</p> <p>В графе Value (Значение) для параметра IP-Address (IP-адрес) введите <code>localhost</code>.</p> <p>Не меняя параметр Port (Порт), щёлкните OK.</p> <p>Результат: диалоговое окно для конфигурирования контроллера теперь содержит шлюз. Зелёная метка перед шлюзом свидетельствует о его нормальному функционировании.</p> <p>Новый шлюз также присутствует в поле Select the network path to the controller (Выбрать сетевой путь к контроллеру).</p> <p>Примечание: по умолчанию шлюз предварительно сконфигурирован в программе SoMachine.</p>
4	<p>Для поиска контроллера, к которому вы хотите подключиться, щёлкните на кнопке Scan network (Сканировать сеть).</p> <p>Результат: будет отображён список устройств, подсоединённых к шлюзу. Зелёный кружок перед шлюзом показывает, что шлюз активен. Неактивные шлюзы обозначены серыми кружками.</p>

Шаг	Действие
5	<p>Чтобы выполнить подключение к контроллеру, выберите соответствующий пункт в списке и щёлкните на кнопке Set active path (Настроить активный путь).</p>
6	<p>Щёлкните на кнопке Edit (Редактирование), чтобы назначить устройствам символьное имя (эквивалентное наименование устройства).</p> <p>Результат: имя будет загружено в устройства и сохранено в них. Это имя затем отобразиться при каждом новом сканировании для облегчения идентификации устройств. Если вы не назначите специальное имя устройствам, их имя может меняться при каждом сканировании.</p>
7	<p>В остальных вкладках окна конфигурирования контроллера не требуется выполнять никаких дополнительных настроек. Настройки связи сохраняются в проекте. Адресация символов остаётся статичной для различных точек доступа к устройству.</p>

Компиляция приложения

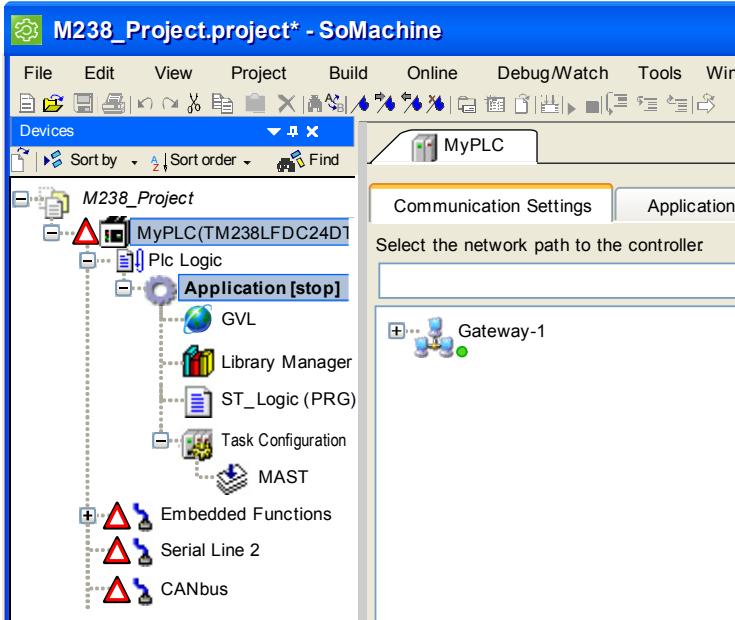
Перед подключением к контроллеру необходимо скомпилировать (см. стр. 32) приложение.

Процесс компиляции запускается автоматически при подключении к контроллеру или выполняется вручную следующим образом:

Шаг	Действие
1	<p>В меню Build (Скомпилировать) выберите команду Build 'Application[YourApplicationName; Plc Logic]' (Скомпилировать 'Приложение[ИмяВашегоПриложения; Логика Плк]').</p> <p>Результат: программное обеспечение проверит программу на синтаксические ошибки и отобразит возможные ошибки (errors), предупреждения (warnings) и сообщения в окне Messages (Сообщения).</p>  <p>The screenshot shows the 'Messages' window with the following content: Build started Application: MyPLC.Application typify code... Compile complete -- 0 errors, 0 warnings</p>

Подключение к контроллеру

Чтобы выполнить подключение к контроллеру, сделайте следующее:

Шаг	Действие
1	<p>В меню Online (Оперативный режим) выберите команду Login to Application[YourApplicationName; Plc Logic] (Вход в приложение[Имя Вашего Приложения; Логика Плк]).</p> <p>Результат: будет установлено соединение с контроллером и в него будет загружена программа.</p> <p>Активное соединение с контроллером в программе обозначено путём выделения зелёным цветом строк MyPLC... (Мой ПЛК...) и Application (Приложение).</p>  <p>Если нет подтверждённых настроек связи, окно конфигурирования контроллера с параметрами связи откроется автоматически.</p>

Загрузка в контроллер

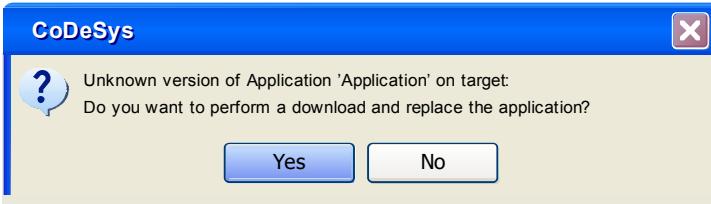
Введение

Ниже описана процедура загрузки программы в контроллер.

Процедура

После проверки программы с помощью команды **Build** (Скомпилировать) вы можете подключиться к контроллеру и загрузить программу:

Шаг	Действие
1	Чтобы выполнить подключение к контроллеру, выберите Online → Login to 'Application[YourApplicationName; Plc Logic]' (Оперативный режим ? 'Вход в приложение[ИмяВашегоПриложения; Логика Плк]').
2	Если ваш проект не совместим с приложением, хранящимся в контроллере (например, контроллер в исходном состоянии без приложения), отобразится следующее сообщение:



The screenshot shows a blue-themed dialog box from the CoDeSys interface. The title bar says 'CoDeSys'. The message area contains a question mark icon and the text: 'Unknown version of Application 'Application' on target: Do you want to perform a download and replace the application?'. At the bottom are two buttons: 'Yes' (blue) and 'No' (white).

Для загрузки приложения в контроллер щёлкните на **Yes** (Да).

Результат: теперь вы находитесь в оперативном режиме.

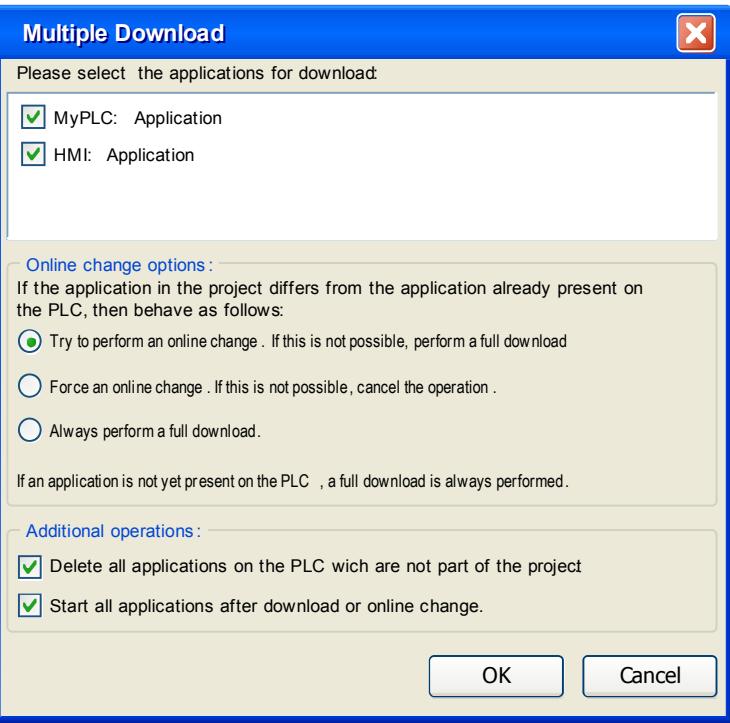
Загрузка приложений управления и ЧМИ посредством однократного подключения

Введение

Ниже описана процедура загрузки полного проекта M238 / Magelis, при условии наличия физического соединения между ПК, в котором выполняется программа SoMachine, терминалом Magelis и контроллером M238.

Процедура

Чтобы полностью загрузить проект, сделайте следующее:

Шаг	Действие
1	<p>В меню Online (Оперативный режим) выберите команду Download Multiple... (Загрузить несколько...).</p> <p>Результат: появится диалоговое окно Multiple Download (Мультизагрузка).</p> 
2	Сконфигурируйте процесс загрузки как описано в интерактивной справке по системе CoDeSys.

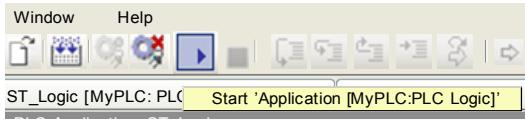
Пуск и остановка контроллера

Введение

Ниже описана процедура запуска/остановки приложения на контроллере.

Процедура

После загрузки приложения в контроллер вы должны явным образом запустить приложение, при этом контроллер остаётся подключённым к ПК:

Шаг	Действие
1	Выполните команду пуска, выбрав Online → Start (Оперативный режим ? Пуск) или щёлкнув на кнопке Start (Пуск) панели инструментов. 
2	Приложение запускается, в строке состояния SoMachine отображается RUN (Выполнение). 

Изменения, вносимые в контроллер в оперативном режиме

Введение

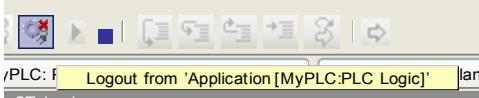
Ниже описана процедура внесения изменений в приложение в оперативном режиме. При этом считается, что вы подключены к контроллеру.

Изменения в оперативном режиме влияют на выполнение прикладной программы. Они могут немедленно оказать значительное воздействие на технологическое оборудование или технологический процесс.

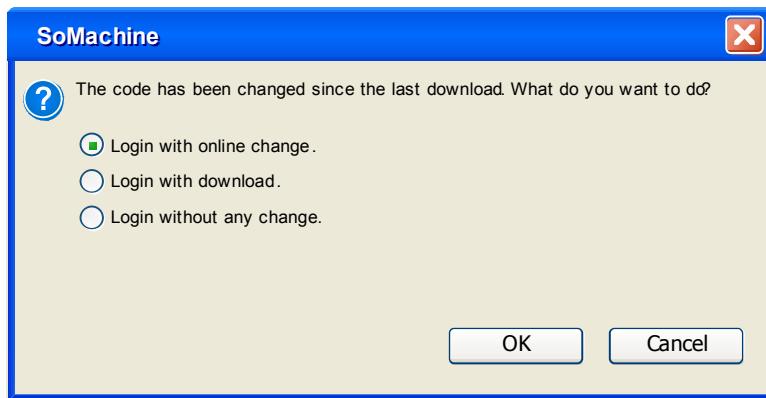


Процедура

Если вы подключены к контроллеру, внесите изменения в приложение контроллера в оперативном режиме следующим образом:

Шаг	Действие
1	Отключитесь от контроллера, выбрав Online → Logout (Оперативный режим ? Выход) или щёлкнув на кнопке Logout (Выход) панели инструментов. 
2	Внесите изменения в приложение контроллера, например, путём изменения логики в программном модуле ST_Logic . Примечание: изменения в оперативном режиме невозможны, если вы изменяете конфигурацию приложения (например, добавляя оборудование).

Шаг	Действие
3	Выполните подключение к контроллеру, выбрав Online → Login (Оперативный режим ? Вход) или щёлкнув на кнопке Login (Вход) панели инструментов.
4	Щёлкните OK в диалоговом окне, чтобы подключиться к контроллеру и ввести изменения в приложение контроллера в оперативном режиме.



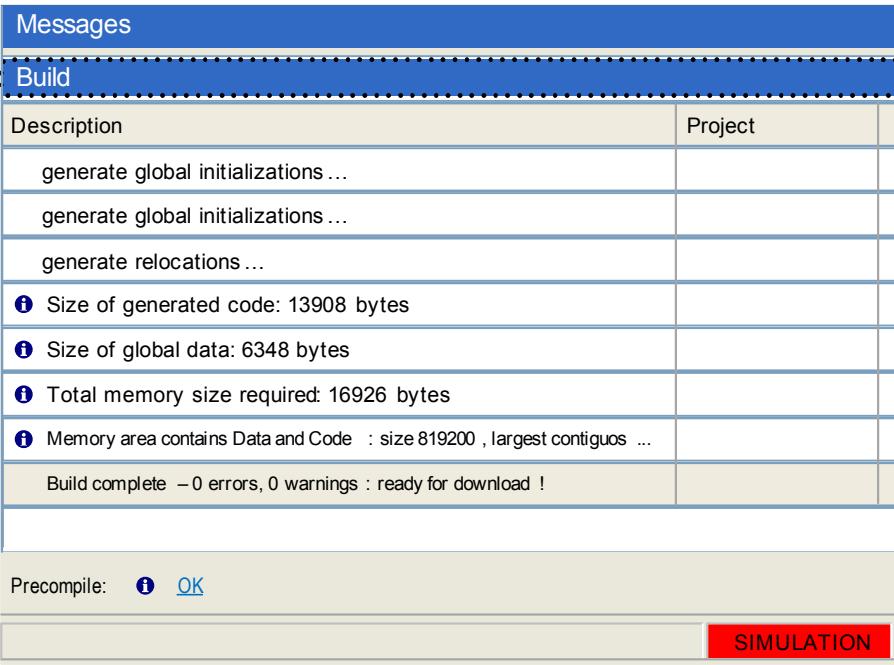
Имитация контроллера

Введение

Функция имитации контроллера позволяет протестировать программу в отсутствие физического устройства.

Процедура

Чтобы имитировать контроллер, сделайте следующее:

Шаг	Действие
1	<p>Для активного приложения выберите в меню Online (Оперативный режим) команду Simulate MyPLC (Имитировать мой ПЛК).</p> <p>Результат: будет установлено соединение с виртуальным контроллером, соответствующим сконфигурированному контроллеру.</p> <p>Режим имитации обозначается в программе посредством красного прямоугольника с надписью SIMULATION (Имитация), который отображается в информационной строке диалогового окна (как показано ниже); в окне Devices (Устройства) имя контроллера указано курсивом.</p>  <p>The screenshot shows the 'Messages' window with the following log entries:</p> <ul style="list-style-type: none"> generate global initializations ... generate global initializations ... generate relocations ... Size of generated code: 13908 bytes Size of global data: 6348 bytes Total memory size required: 16926 bytes Memory area contains Data and Code : size 819200 , largest contiguous ... Build complete – 0 errors, 0 warnings : ready for download ! <p>At the bottom, there is a 'Precompile' section with an 'OK' button and a large red 'SIMULATION' button.</p>
2	Вы можете работать с этим виртуальным контроллером так же, как с настоящим, то есть выполнять любые команды (например, Login (Войти), Download (Загрузить)).

Отладка приложения

7

Введение

В этом разделе описывается, как определить точки прерывания программы с целью её пошагового выполнения.

Содержание данной главы

Данная глава содержит следующие темы:

Тема	Страница
Изменение значений переменных	86
Пошаговое выполнение проекта с точками прерывания	87

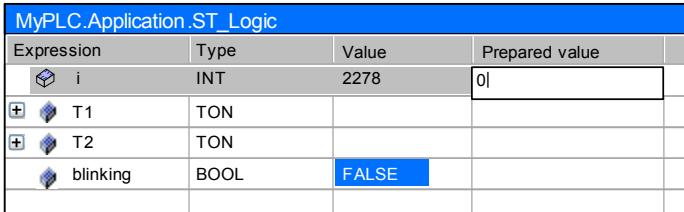
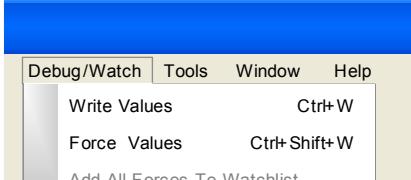
Изменение значений переменных

Введение

Ниже описана процедура изменения данных в программе в оперативном режиме. При этом считается, что вы подключены к контроллеру.

Процедура

Если вы подключены к контроллеру, вы можете выполнять однократное или циклическое изменение данных посредством программы SoMachine следующим образом:

Шаг	Действие																				
1	<p>Откройте программный модуль ST_Logic, дважды щёлкнув на соответствующем узле в окне Devices (Устройства).</p> <p>Результат: программный модуль ST_Logic отобразится в оперативном режиме.</p>																				
2	<p>Введите новые значения данных в поле объявления переменных (верхняя часть) редактора программного модуля, в графе Prepared value (Подготовленное значение).</p>  <table border="1"> <thead> <tr> <th>Expression</th> <th>Type</th> <th>Value</th> <th>Prepared value</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>INT</td> <td>2278</td> <td>0</td> </tr> <tr> <td>+ T1</td> <td>TON</td> <td></td> <td></td> </tr> <tr> <td>+ T2</td> <td>TON</td> <td></td> <td></td> </tr> <tr> <td>blinking</td> <td>BOOL</td> <td></td> <td>FALSE</td> </tr> </tbody> </table>	Expression	Type	Value	Prepared value	i	INT	2278	0	+ T1	TON			+ T2	TON			blinking	BOOL		FALSE
Expression	Type	Value	Prepared value																		
i	INT	2278	0																		
+ T1	TON																				
+ T2	TON																				
blinking	BOOL		FALSE																		
3	<p>Передайте изменения в контроллер посредством меню Debug/Watch → Write Values (Отладка/Контроль ? Запись значений).</p>  <p>Результат: изменённые данные переменных будут немедленно внесены в приложение контроллера.</p> <p>Примечание: если вы выбрали команду Force Values (Принудительно задать значения), Prepared value (Подготовленное значение) будет однократно записываться в начале каждого цикла автомата.</p>																				

Примечание: вы можете выполнить те же действия для списков переменных.

Пошаговое выполнение проекта с точками прерывания

Введение

Ниже описана процедура отладки программы с использованием точек прерывания. Точки прерывания служат для построчной отладки программы контроллера. При этом считается, что вы подключены к контроллеру.

Определение точек прерывания изменяет выполнение прикладной программы. Эти изменения могут немедленно оказать значительное воздействие на технологическое оборудование или технологический процесс.



Процедура

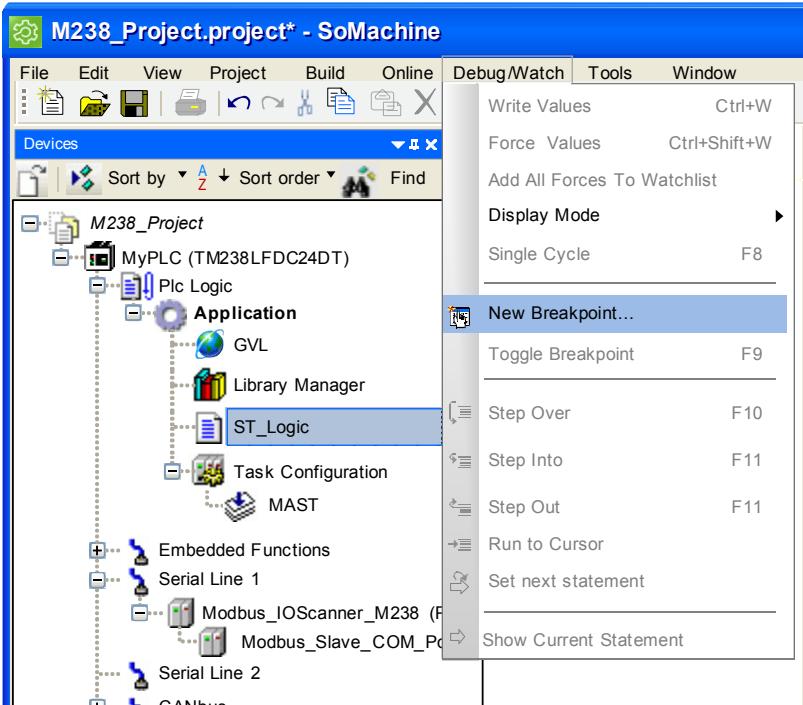
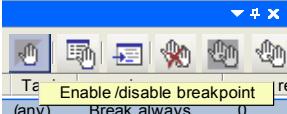
В нижеследующей процедуре изложен метод определения и удаления точек прерывания и пошагового выполнения программы при условии подключения к контроллеру:

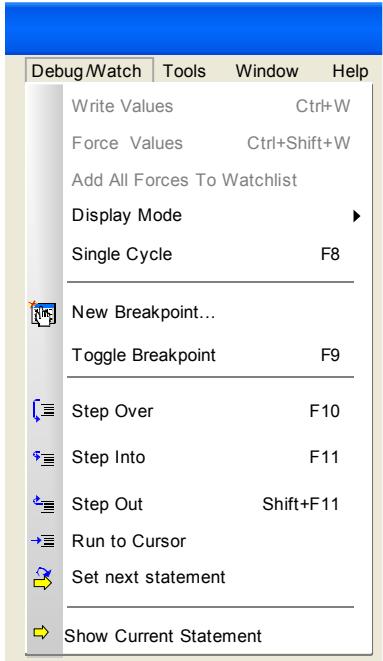
Шаг	Действие
1	Откройте редактор точек прерывания, выбрав View → Breakpoints (Просмотр → Точки прерывания).
2	Откройте программный модуль ST_Logic и выберите с помощью мыши строку для отладки (строка 2 в приведённом ниже примере).

```

1 i 9274 :=i 9274 +1;      //Incremental counter
2 T1(IN[FALSE]:= T2.Q[FALSE], PT[ T#1s ]:=T#1S, Q=> );
3 T2(IN[TRUE]:= NOT T1.Q[FALSE], PT[ T#1s ]:=T#1S, Q=> blinking[FALSE] );
4
5
6
7 RETURN

```

Шаг	Действие
3	<p>Чтобы создать точку прерывания, в меню Debug/Watch (Отладка/Контроль) выберите команду New Breakpoint (Новая точка прерывания).</p>  <p>Результат: новая точка прерывания (неактивная) будет определена и отображена в окне точек прерывания.</p>
4	<p>Выберите точку прерывания в окне точек прерывания и активируйте её (переключение):</p>  <p>Результат: программа будет остановлена в точке прерывания.</p>

Шаг	Действие
5	<p>Осуществите пошаговое выполнение программы с помощью команд меню Debug/Watch (Отладка/Контроль).</p>  <p>Результат: в соответствии с выбранной командой происходит построчное выполнение программы.</p>

За дополнительной информацией обращайтесь к интерактивной справке по системе CoDeSys.

Пример программирования контроллера

8

Пример программирования на языке структурированного текста

Введение

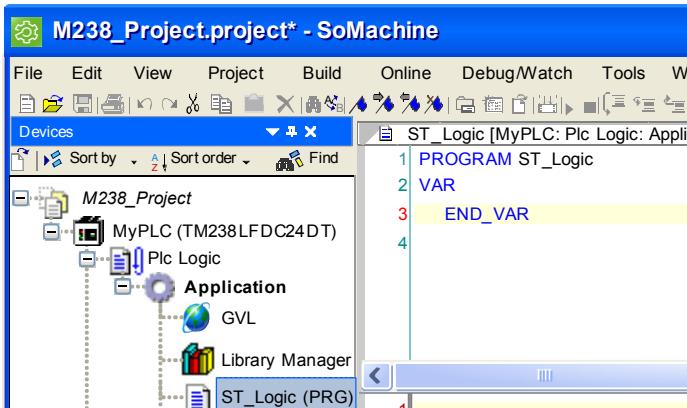
Ниже приведён пример программирования на языке структурированного текста.

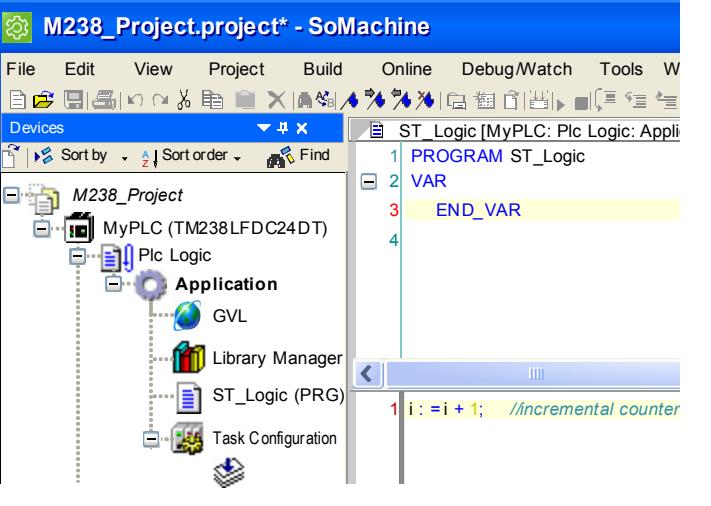
В этом примере программирования выполняются следующие задачи:

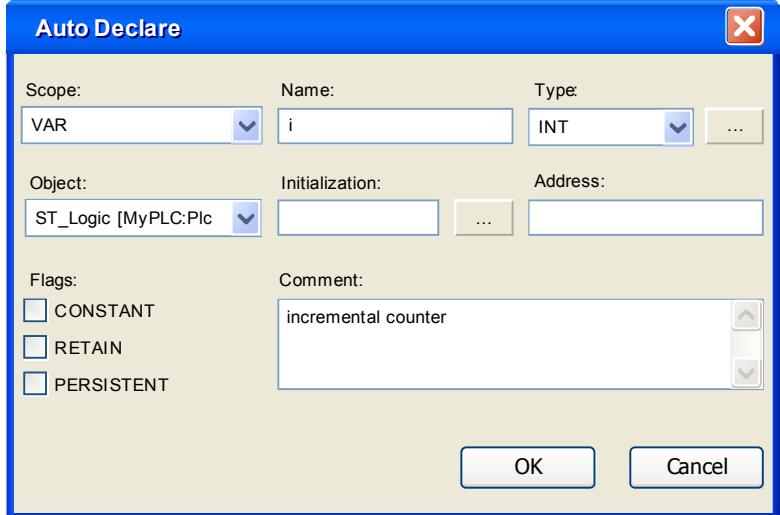
- создание счётчика добавлением смещения 1 к переменной;
- 1-секундный мигающий сигнал путём использования 2 таймеров TON.

Процедура

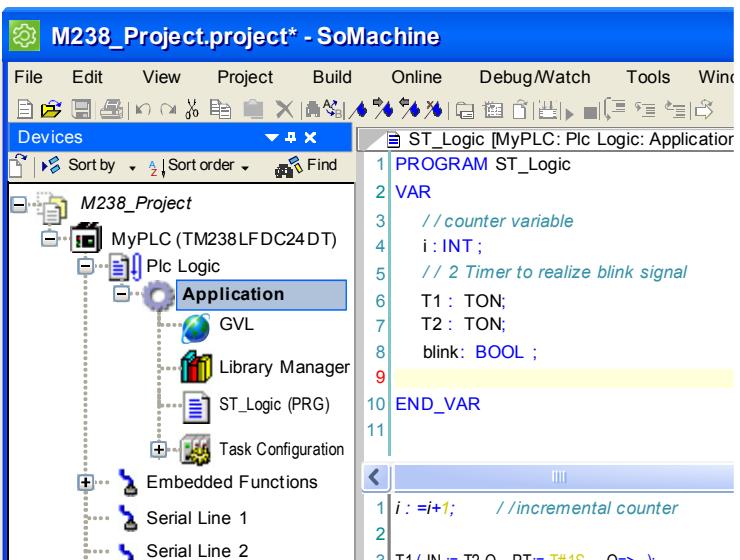
Чтобы создать логику на языке структурированного текста, сделайте следующее:

Шаг	Действие
1	<p>Чтобы создать свою собственную логику, дважды щёлкните на узле программного модуля (в нашем примере, ST_Logic (PRG)), созданного вами в соответствии с указаниями параграфа «Создание программного модуля перед написанием логики» (см. стр. 19).</p> <p>Результат: справа появится редактор, позволяющий создать логику для программного модуля.</p> 

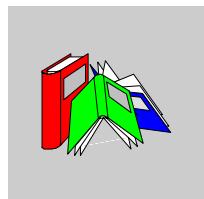
Шаг	Действие
2	<p>Чтобы создать счётчик добавлением смещения 1 к переменной, введите следующий код в нижней части редактора логики:</p> <pre>i := i + 1;</pre> <p>Чтобы добавить комментарий, отделите его от кода двумя наклонными чертами вправо, как показано ниже.</p>  <pre>1 PROGRAM ST_Logic 2 VAR 3 END_VAR 4 1 i := i + 1; //incremental counter</pre>

Шаг	Действие
3	<p>Нажмите клавишу ENTER (Ввод).</p> <p>Результат: так как переменная <i>i</i> ещё не была определена в проекте, диалоговое окно Auto Declare (Автообъявление) откроется с уже определёнными параметрами Scope (Область действия), Name (Имя) и Type (Тип):</p> 
4	<p>При необходимости адаптируйте параметры диалогового окна Auto Declare и щёлкните OK для создания переменной <i>i</i>.</p> <p>Результат: новая переменная <i>i</i> отобразится в верхней части редактора логики.</p> <p>Вы также можете ввести новую переменную непосредственно в редактор логики, как описано в параграфе «Создание переменных» (см. стр. 25).</p>

Шаг	Действие
5	<p>Чтобы создать сигнал, мигающий каждые две секунды, необходимы два функциональных блока TON, T1 и T2. Блок T1 использует выходной сигнал Q2 блока T2 в качестве входа. Блок T2 использует инвертированный выходной сигнал блока T1 в качестве входа. Объявите эти функциональные блоки как показано на нижеприведённом рисунке. Создайте следующий код в нижней части редактора логики:</p> <pre>T1(IN:= T2.Q, PT:=T#1S , Q=>); T2(IN:= NOT T1.Q, PT:=T#1S , Q=> blink);</pre>

Шаг	Действие
6	<p>Нажмите клавишу ENTER (Ввод) и объявит новые переменные в диалоговом окне Auto Declare (Автообъявление) или в верхней части редактора логики.</p> <p>Определите переменные как показано ниже:</p>  <pre> M238_Project.project* - SoMachine File Edit View Project Build Online Debug/Watch Tools Window Help Devices M238_Project MyPLC (TM238LFDC24DT) Plc Logic Application GVL Library Manager ST_Logic (PRG) Task Configuration Embedded Functions Serial Line 1 Serial Line 2 CANbus ST_Logic [MyPLC: Plc Logic: Application] 1 PROGRAM ST_Logic 2 VAR 3 // counter variable 4 i : INT; 5 // 2 Timer to realize blink signal 6 T1 : TON; 7 T2 : TON; 8 blink: BOOL ; 9 10 END_VAR 11 i := i+1; // incremental counter 2 3 T1(IN := T2.Q , PT:= T#1S , Q=>); 4 T2(IN := NOT T1.Q , PT:= T#1S , Q=> blink); 5 </pre>

Глоссарий



0-9

%

Префикс, который идентифицирует внутренние адреса памяти в контроллере, которые используются для хранения значений переменных программы, констант, входов/выходов и т.д.

A

application

Приложение (прикладная программа). Прикладная программа SoMachine состоит из программы, данных конфигурации, символов и документации.

ASCII

American Standard Code for Information Interchange (американский стандартный код для информационного обмена) – протокол связи для представления алфавитно-цифровых символов, особенно символов, рисунков и некоторых графических символов и управляющих символов.

C

CANopen

Протокол связи и спецификация профиля устройства для логического контроллера Modicon M238.

Continuous function chart language (CFC)

Язык непрерывных функциональных блоков. Язык CFC (расширение стандарта МЭК 61131-3) – графический язык программирования, созданный на основе языка функциональных блоковых схем (FBD). При этом данный язык использует не схемы, а свободное расположение графических элементов, что позволяет применять цепи обратной связи.

Controller

Контроллер. Промышленный программируемый контроллер, используемый для автоматизации промышленных процессов.

E

EEPROM

Electrically Erasable Programmable Read-Only Memory – электрически стираемое программируемое постоянное запоминающее устройство (ЭСППЗУ). Энергонезависимое запоминающее устройство, используемое для хранения данных, которые должны сохраняться при отключении питания.

EIA rack

Стойка EIA. Стандартизированная система стоек (стандарты EIA 310-D, МЭК 60297 и DIN 41494 SC48D) для монтажа различных электронных модулей, шириной 19 дюймов (48,26 см).

Expansion bus

Шина расширения. Служит для соединения модулей расширения входов/выходов с контроллером Modicon M238.

Expansion I/O modules

Модули расширения входов/выходов. Опциональные модули, позволяющие добавить входов/выходов контроллеру Modicon M238 (не все модели контроллеров допускают расширение).

F

Firmware

Аппаратно-реализованное ПО. Программа, «зашитая» в контроллер Modicon M238.

Function block (FB)

Функциональный блок. Программный модуль входов и переменных, организованных, чтобы вычислить значения для выходов, основанных на определённой функции типа таймера или счётчика.

Function block diagram**language (FBD)**

Язык функциональных блоковых схем. Функциональная блоковая схема описывает функцию между входными и выходными переменными. Функция описывается как совокупность элементарных блоков. Входные и выходные переменные соединяются с блоками посредством соединительных линий. Выход блока может быть также соединён с входом другого блока.

G

GRAFCET

GRAFCET используется, чтобы представить функционирование последовательной операции в структурированной и графической форме.

Это – аналитический метод, который делит любую последовательную систему управления на ряд шагов, с которыми связаны действия, переходы и условия.

GVL

Global Variable List – список глобальных переменных в ПО SoMachine; управляет глобальными переменными во всех программных модулях приложения.

H

HSC

High-Speed Counter – быстродействующий счётчик.

|

IEC 61131-3

Стандарт МЭК 61131-3.

Международная электротехническая комиссия (МЭК) – международная некоммерческая неправительственная организация по стандартизации, которая разрабатывает и публикует международные стандарты в области электрических, электронных и смежных технологий.

- МЭК 61131 – стандарт на программируемые логические контроллеры (ПЛК).
- МЭК 61131-3 касается языков программирования ПЛК; определяет два графических и два текстовых языка программирования:
 - многоступенчатые схемы (LD) – графический язык;
 - функциональные блоковые схемы (FBD) – графический язык;
 - структурированный текст (ST) – текстовой язык;
 - список инструкций (IL) – текстовой язык;
 - язык последовательных функциональных блоков (SFC) содержит элементы, позволяющие организовать программы для последовательной и параллельной обработки.

Instruction list language (IL)

Язык списка инструкций. Программа, написанная на языке списка инструкций, состоит из ряда инструкций (команд), последовательно выполняемых контроллером. Каждая инструкция включает в себя номер строки, код инструкции и операнд. Соответствует стандарту МЭК 61131-3.

IP 20

Ingress Protection – защита от проникновения. Система классификации степеней защиты оболочки электрооборудования. Степень защиты обозначается буквами IP и двумя цифрами. Первая цифра отражает два фактора: защиту людей и защиту оборудования. Вторая цифра показывает защиту от проникновения воды. Подробнее см. в стандарте NF EN 60.529.

L

Ladder diagram language (LD)

Язык многоступенчатых схем.

Программа, написанная на языке многоступенчатых схем, составлена из графического представления команд программы контроллера с символами для контактов, катушек, и блоков в наборы ступеней, выполняемых последовательно контроллером. Соответствует стандарту МЭК 61131-3.

Latching input

Запирающийся вход. Поступающий импульс фиксируется и записывается для более поздней проверки прикладной программой.

M**Master/slave**

Ведущий/ведомый – тип протокола связи, в котором одно устройство или процесс осуществляет одностороннее управление одним или несколькими устройствами.

Modbus

Протокол связи, обеспечивающий связь между многочисленными устройствами, подключёнными к одной сети.

N**NEMA**

National Electrical Manufacturers Association – Национальная ассоциация производителей электрооборудования. Издаёт стандарты на характеристики различных классов оболочек электрооборудования. Стандарты NEMA касаются коррозионной стойкости, защиты от дождя и погружения и т.д. Для стран-членов МЭК, степень защиты от проникновения (IP) оболочки определяется стандартом МЭК 60529.

Network

Сеть. Взаимосоединённые устройства, имеющие общий путь прохождения данных и общий протокол связи.

Node

Узел. Адресуемое устройство в коммуникационной сети.

P**PC**

Personal Computer – персональный компьютер (ПК).

POU

Program organization unit – программный модуль SoMachine, в котором записан программирования.

Protocol

Протокол. Соглашение или стандарт, который контролирует или делает возможным соединение, связь и передачу данных между двумя окончными точками компьютерной сети.

PTO

Pulse Train Output – выход для группы импульсов.

PWM

Pulse Width Modulation – широтно-импульсная модуляция (ШИМ)

R

Real-time clock (RTC)

Часы реального времени. Опционная функция, отчитывающая время, даже если контроллер не включен в течение ограниченного промежутка времени.

Reflex output

Рефлексный выход. В режиме счёта (HSC), текущее значение быстродействующего счётчика измеряется по отношению к пороговым значениям, чтобы определить состояние этих выделенных выходов.

RFID

Radio-Frequency Identification – радиочастотная идентификация. Метод автоматической идентификации, основанный на хранении и дистанционном сборе данных с использованием RFID-меток или транспондеров.

RPDO

Process Data Object Reception – приём объектов данных процесса

RUN

Выполнить. Команда для выполнения контроллером прикладной программы.

S

Scan

Сканирование. Контроллер сканирует программу и по существу исполняет три основных функции. Сначала, он читает входы и размещает эти значения в память. Затем, он выполняет прикладную программу по одной инструкции одновременно и сохраняет результаты в память. Наконец, он использует результаты, чтобы обновить выходы.

Sequential function chart language (SFC)

Язык последовательных функциональных блоков. Данный язык программирования позволяет разделять процессы на шаги (этапы). SFC включает в себя связанные с действиями шаги, связанные с логическими условиями переходы, а также направленные связи между шагами и переходами. Стандарт SFC определён в МЭК 848. Соответствует стандарту МЭК 61131-3.

SoMachine

SoMachine – комплексное программное средство разработки программ для контроллера, позволяющее конфигурировать и программировать контроллер Modicon M238 и устройства, соответствующие стандарту МЭК 61131-3.

STOP

Остановить. Команда для остановки выполнения прикладной программы контроллером.

Structured text language (ST)

Язык структурированного текста. Программа, написанная на языке структурированного текста, состоит из сложных операторов и вложенных команд, таких как: итерационные циклы, условные функции и исполнения. Соответствует стандарту МЭК 61131-3.

Symbol

Символ. Элементарная переменная контроллера, публикуемая для терминалов ЧМИ.

T

Task

Задача. Совокупность секций и подпрограмм, выполняемых циклически или периодически в случае задачи MAST, или периодически в случае задачи FAST.

Задача имеет уровень приоритета и привязана к входам и выходам ПЛК. Соответственно, эти входы/выходы обновляются.

TPDO

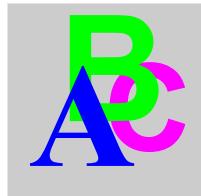
Process Data Object Transmission – передача объектов данных процесса

V

variable

Переменная. Элемент памяти, который может быть адресован и изменен программой.

Предметный указатель



3

Загрузка в контроллер, 79
Загрузка, 74
Задачи
 Добавление программного модуля, 22

И

Изменения, вносимые в оперативном режиме, 82
Имитация, 84
Интерфейс CANopen, 34

К

Команда Build, 32
Компиляция приложения, 77
Конфигурирование устройства ЧМИ, 65

М

M238, 16

Н

Назначение входов/выходов, 30
Назначение переменных, 30

О

Общие сведения об аппаратных средствах, 9
Определение переменных, 25
Остановка контроллера, 81

П

Переменные, 62
 Терминалы ЧМИ, 68
Подключение к контроллеру, 74
Пример программирования, 91
Программа Vigeo-Designer, 67
Программный модуль
 Создание программного модуля, 19
Пуск контроллера, 81

С

Связь с терминалом ЧМИ, 62
Сканера входов/выходов Modbus, 48
Структурированный текст, 91

Т

Терминалы ЧМИ
 Конфигурирование, 67
Точки прерывания, 87

Ш

Шаблоны устройств, 39

Schneider Electric в странах СНГ

- **Алматы**, Казахстан, 050050, ул. Табачнозаводская, 20, Швейцарский центр, тел.: (727) 244 15 05 (многоканальный), факс: (727) 244 1506, 244 15 07 • **Астана**, Казахстан, 010000, ул. Бейбитшилик, 18, бизнес-центр «Бейбитшилик 2002», офис 402, тел.: (3172) 91 06 69, факс: (3172) 91 06 70 • **Атырау**, Казахстан, 060002, ул. Абая, 2 А, бизнес-центр «Утас - С», офис 407, тел.: (3122) 32 31 91, 32 66 70, факс: (3122) 32 37 54 • **Волгоград**, Россия, 400089, ул. Профсоюзная, 15, офис 12, тел./факс: (8442) 93 08 41 • **Воронеж**, Россия, 394026, пр-т Труда, 65, офис 267, тел.: (4732) 39 06 00, факс: (4732) 39 06 01 • **Днепропетровск**, Украина, 49000, ул. Глинки, 17, этаж 4, тел.: (380567) 90 08 88, факс: (380567) 90 09 99 • **Донецк**, Украина, 83087, ул. Инженерная, 1 В, тел.: (38062) 385 48 45, 385 48 65, факс: (38062) 385 49 23 • **Екатеринбург**, Россия, 620014, ул. Радищева, 28, этаж 11, тел.: (343) 378 47 36, 378 47 37 • **Иркутск**, Россия, 664047, ул. 1-я Советская, 3 Б, офис 312, тел.: (3952) 29 00 07, факс: (3952) 29 20 43 • **Казань**, Россия, 420107, ул. Спартаковская, 6, этаж 7, тел./факс: (843) 526 55 84 / 85 / 86 / 87 • **Калининград**, Россия, 236040, Гвардейский пр., 15, тел.: (4012) 53 59 53, факс: (4012) 57 60 79 • **Киев**, Украина, 03057, ул. Смоленская, 31-33, кор. 29, тел.: (38044) 538 14 70, факс: (38044) 538 14 71 • **Краснодар**, Россия, 350063, ул. Кубанская набережная, 62 / ул. Комсомольская, 13 , офис 224, тел.: (861) 278 00 49, тел./факс: (861) 278 00 62, 278 00 63, 278 01 13 • **Красноярск**, Россия, 660021, ул. Горького, 3 А, офис 302, тел.: (3912) 56 80 95, факс: (3912) 56 80 96 • **Львов**, Украина, 79015, ул. Тургенева, 72, корп. 1, тел./факс: (38032) 298 85 85 • **Минск**, Беларусь, 220006, ул. Белорусская, 15, офис 9, тел./факс: (37517) 226 06 74, 227 60 34, 227 60 72 • **Москва**, Россия, 129281, ул. Енисейская, 37, стр. 1, тел.: (495) 797 40 00, факс: (495) 797 40 02 • **Мурманск**, Россия, 183038, ул. Воровского, 5/23, Конгресс-отель «Меридиан», офис 739, тел.: (8152) 28 86 90, факс: (8152) 28 87 30 • **Нижний Новгород**, Россия, 603000, пер. Холодный, 10 А, этаж 8, тел./факс: (831) 278 97 25 / 26 • **Николаев**, Украина, 54030, ул. Никольская, 25, бизнес-центр «Александровский», офис 5, тел.: (380512) 58 24 67, факс: (380512) 58 24 68 • **Новосибирск**, Россия, 630132, ул. Красноярская, 35, бизнес-центр «Гринвич», офис 1309, тел./факс: (383) 227 62 53, 227 62 54 • **Одесса**, Украина, 65079, ул. Куліково поле, 1, офис 213, тел.: (38048) 728 65 55, факс: (38048) 728 65 35 • **Пермь**, Россия, 614010, Комсомольский пр-т, 98, офис 11, тел./факс: (342) 290 26 11 / 13 / 15 • **Ростов-на-Дону**, Россия, 344002, ул. Социалистическая, 74, литера А, тел.: (863) 200 17 22 / 23, факс: (863) 200 17 24 • **Самара**, Россия, 443096, ул. Коммунистическая, 27, тел./факс: (846) 266 50 08, 266 41 41 / 11 • **Санкт-Петербург**, Россия, 198103, ул. Циолковского, 9, корп. 2 А, тел.: (812) 380 64 64, факс: (812) 320 64 63 • **Симферополь**, Украина, 95013, ул. Севастопольская, 43/2, офис 11, тел.: (380652) 44 38 26, факс: (380652) 54 81 14 • **Сочи**, Россия, 354008, ул. Виноградная, 20 А, офис 54, тел.: (8622) 96 06 01, факс: (8622) 96 06 02 • **Уфа**, Россия, 450098, пр-т Октября, 132/3, бизнес-центр КПД, этаж 9, блок-секция 3, тел.: (347) 279 98 29, факс: (347) 279 98 30 • **Хабаровск**, Россия, 680000, ул. Муравьева-Амурского, 23, этаж 4, тел.: (4212) 30 64 70, факс: (4212) 30 46 66 • **Харьков**, Украина, 61070, ул. Ак. Проскуры, 1, бизнес-центр «Telesens», офис 569, тел.: (38057) 719 07 79, факс: (38057) 719 07 49

Центр поддержки клиентов

Тел.: 8 (800) 200 64 46 (многоканальный)

Тел.: (495) 797 32 32, факс: (495) 797 40 04

ru.csc@ru.schneider-electric.com

www.schneider-electric.ru

МКР-???-???-10

11/2010