Контроллеры Logix5000
Организация данных
Данные ввода/вывода

Руководство по программированию

Номера по каталогу 1756 ControlLogix, 1769 CompactLogix, 1789 SoftLogix, 1794 FlexLogix, PowerFlex 700S с DriveLogix

Allen-Bradley • Rockwell Software
Rockwell Automation

Важная информация для пользователей

Рабочие характеристики полупроводникового оборудования отличаются от параметров электромеханического оборудования. Публикация SGI-1.1 Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls (Основы безопасности при использовании, установке и обслуживании полупроводниковых устройств), которую можно получить в региональном офисе отдела продаж корпорации Rockwell Automation или в Интернете (http://literature.rockwellautomation.com), описывает некоторые важные различия между полупроводниковым оборудованием и электромеханическими устройствами. Из-за этих различий, а также ввиду широкого разнообразия в применении различных полупроводниковых устройств, персонал, ответственный за работу с указанным оборудованием, должен убедиться, что в каждом конкретном случае такое применение является целесообразным.
Корпорация Rockwell Automation, Inc. не берет на себя ответственность за прямой или косвенный ущерб, возникший при использовании этого оборудования.
Примеры и схемы в данном руководстве приведены исключительно в иллюстративном качестве. Поскольку с любым конкретным устройством связано множество переменных параметров и требований, корпорация Rockwell Automation, Inc. не может принять на себя каких-либо обязательств или ответственности за практическое применение приведенных здесь примеров и схем.
Корпорация Rockwell Automation, Inc. не принимает на себя никаких патентных обязательств в отношении использования информации, схем подключения, оборудования и программного обеспечения, приведенных в данном руководстве.
Полное или частичное воспроизведение содержимого данного документа без письменного разрешения Rockwell Automation, Inc. запрещено. В данном руководстве мы обращаем ваше внимание на вопросы техники безопасности с помощью следующих оозначений.

ПРЕДУПРЕЖДЕНИЕ	Обозначает информацию о действиях и обстоятельствах, которые могут привести к взрыву в опасных условиях, к травмам или смерти людей, повреждению собственности или экономическому ущербу.
ВАЖНО	Обозначает информацию, наиболее важную для успешной эксплуатации устройства и понимания особенностей его работы.
ВНИМАНИЕ	Обозначает информацию о действиях и обстоятельствах, которые могут привести к взрыву в опасных условиях, к травмам или смерти людей, повреждению собственности или экономическому ущербу. Данное обозначение помогает определить опасность, избежать опасности и оценить последствия.
ОПАСНОСТЬ ПОРАЖЕНИЯ ЭЛЕКТРИЧЕСКИМ ТОКОМ	Этот знак может находиться снаружи или внутри оборудования, например, привода или электродвигателя, с целью предупреждения людей о возможном присутствии опасного уровня напряжения.
ОПАСНОСТЬ ОЖОГА	Этот знак может находиться снаружи или внутри оборудования, например, привода или электродвигателя, с целью предупреждения людей об опасных температурных значениях.

Allen-Bradley, Rockwell Automation, TechConnect, CompactLogix, POINT I/O, FLEX I/O, PowerFlex 70, PowerFlex 40, RSLogix 5000, KwikLink, RSNetWorx, RSNetWorx для ControlNet, RSNetWorx для DeviceNet, ControlFlash, RSLinx, RSView, RSView Studio, Compact I/O, NetLinx, FactoryTalk Activation, RSView Machine Edition, RSViewME, RSLinx Classic и PanelView Plus являются торговыми знаками компании Rockwell Automation.

Торговые знаки, не принадлежащие компании Rockwell Automation, являются собственностью соответствующих компаний.

Предисловие
Назначение данного руководства 5
Как использовать данное руководство 5

Глава 1

Обмен данными с модулями ввода/вывода
Введение 7
Запрашиваемый межпакетный интервал 8
Формат обмена данными 9
Прямое или оптимизированное для рэка соединение. 9
Отношения собственности 10
Электронный ключ 12
Адресация данных ввода/вывода. 13
Буферизация ввода/вывода 14
Буферизация ввода/вывода 14

Глава 2

Организация тегов
Введение 17
Тип тега. 18
Тип данных. 19
Область видимости 21
Рекомендации по созданию тегов. 22
Создание тега 25
Создание массива. 26
Создание массива. 29
Создание пользовательского типа данных. 30
Рекомендации по пользовательским типам данных. 32
Создание пользовательского типа данных 32
Описание пользовательского типа данных 34
Включение и выключение сквозных (pass-through) и присоединяемых (append) описаний 35
Вставка сквозных описаний 35
Адресация данных в тегах. 36
Задание тегов-псевдонимов 37
Отображение информации о псевдонимах 39
Задание псевдонима 39
Задание косвенного адреса. 40
Выражения 42
Индекс массива за границей диапазона 43
Информация по тегам. 43
Информация о базовых тегах 44
Информация о тегах-псевдонимах 44
Информация о производимых тегах 44
Информация о потребляемых тегах. 44
Переключение языков. 44

Глава 3

Форсировка входов/выходов
Введение 47
Меры предосторожности 47
Разрешение форсировок 48
Запрещение или удаление форсировки. 48
Проверка состояния форсировок 49
Светодиод FORCE 50

Инструкция GSV 50
Когда использовать форсировку ввода/вывода 51
Форсировка входного значения. 51
Форсировка выходного значения 52
Добавление форсировки ввода/вывода 52
Удаление или запрещение форсировок. 53
Удаление отдельной форсировки. 53
Запрещение всех форсировок ввода/вывода 54
Удаление всех форсировок ввода/вывода. 54
Служба поддержки Rockwell Automation 33
Содействие при установке 33
Возврат продукции 33

Назначение данного руководства

В данном руководстве рассматриваются возможности доступа к данных ввода/вывода и теговым данным в контроллерах Logix5000. . Это руководство входит в пакет руководств по общей методике программирования и использования контроллеров Logix5000. Полный перечень руководств по общей методике приведен в руководстве по программированию «Общая методика для программируемых контроллеров Logix5000», публикация 1756-PM001.

Понятие контроллер Logix5000 относится к любому контроллеру на базе операционной системы Logix5000, включая следующие:

- Контроллеры CompactLogix.
- Контроллеры ControlLogix.
- Контроллеры DriveLogix.
- Контроллеры FlexLogix.
- Контроллеры SoftLogix5800.

Как использовать данное руководство

Форматирование некоторых фрагментов текста отличается от остального текста.

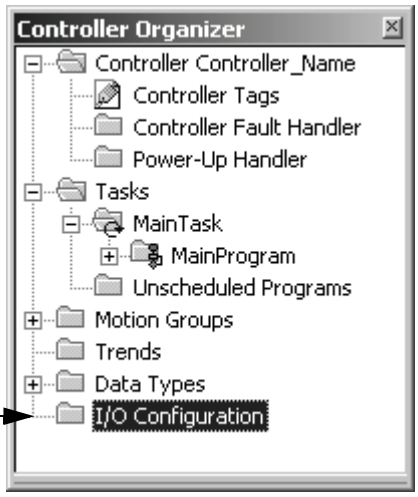
Форматирование текста	Обозначает	Например	Означает
курсив	название элемента, отображаемое на экране или в примере	Щелкните правой кнопкой мыши на User-Defined…	Щелкните правой кнопкой мыши по элементу с названием User-Defined.
шрифт `courier`	данные, которые вы должны подставить в соответствии с вашим приложением (переменная)	Щелкните правой кнопкой мыши на name_of_program…	Вы должны выбрать конкретную программу в вашем приложении. Обычно так обозначаются определяемые пользователем переменные.
в скобках	клавиша на клавиатуре	Нажмите [Enter].	Нажмите клавишу Enter.

Для заметок:

Обмен данными с модулями ввода/вывода

Введение

Чтобы обеспечить обмен данными с модулем ввода/вывода в вашей системе, добавьте соответствующий модуль в папку I/O Configuration (Конфигурация ввода/вывода) данного контроллера.



Добавляйте модули ввода/вывода сюда.

При добавлении модуля вы также задаете его конкретную конфигурацию. Хотя разные модули имеют разные конфигурируемые параметры, имеется и ряд общих опций, которые вам обычно требуется сконфигурировать:

- Запрашиваемый межпакетный интервал (RPI - Requested Packet Interval)
- Формат обмена данными (communication format)
- Электронный ключ (electronic keying)

Запрашиваемый межпакетный интервал

Для передачи данных ввода/вывода контроллеры Logix5000 используют соединения.

Термин	Определение
Соединение	Коммуникационная связь между двумя устройствами, такая как связь между контроллером и модулем ввода/вывода, терминалом PanelView и другим контроллером. Соединения – это ресурсы, для обеспечения более надежной связи между устройствами, чем обмен сообщениями без соединения. Каждый контроллер поддерживает ограниченное количество соединений. Косвенно, вы определяете количество соединений, используемых контроллерами при конфигурировании контроллера для связи с другими устройствами данной системы. Соединения используются следующими типами связи: · модулями ввода/вывода, · производящими и потребляющими тегами, · определенными типами инструкций Message (MSG) (Не для всех типов соединений)
Запрашиваемый межпакетный интервал (RPI)	RPI задает период обновления данных через соединение. Например, модуль входных данных будет отправлять данные в контроллер с тем RPI, который вы определите для этого модуля. · RPI задается в миллисекундах. Диапазон от 0.2 (200 микросекунд) до 750 миллисекунд. · Если сеть ControlNet соединяет заданные устройства, RPI резервирует участок в потоке в потоке данных, проходящих через сеть ControlNet. Расчет времени для этого участка может не совпадать с точным значением RPI, но система управления гарантирует, что данные будут передаваться с частотой, заданной RPI.

В контроллерах RSLogix5000 значения ввода/вывода обновляются с периодичностью, сконфигурированной вами посредством папки конфигурации ввода/вывода (I/O configuration) вашего проекта. Эти значения обновляются асинхронно по отношению к выполнению логики. Контроллер обновляет значение с заданной периодичностью, независимо от выполнения логики.

ВНИМАНИЕ

Позаботьтесь о том, чтобы в процессе выполнения задачи в памяти содержались адекватные значения. Вы можете скопировать данные или запомнить их в буфере в начале сканирования, чтобы обеспечить опорные значения для вашей логики.

- Программы задачи обращаются к данным ввода и вывода непосредственно в памяти, находящейся в области видимости контроллера.
- Логика любой задачи может изменять данные в области видимости контроллера.
- Значения в тегах и в тегах ввода/вывода являются асинхронными и могут изменяться в процессе выполнения задачи.
- Входное значение, к которому идет обращение в начале выполнения задачи, может быть другим при последующем обращении.
- Чтобы предотвратить изменение входного значения во время сканирования, скопируйте это значение в другой тег и используйте данные из него (запомните значения в буфере).

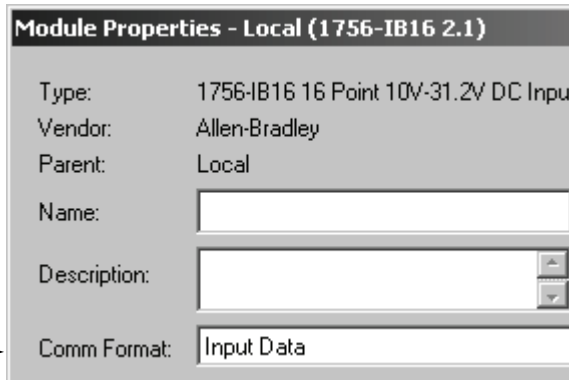
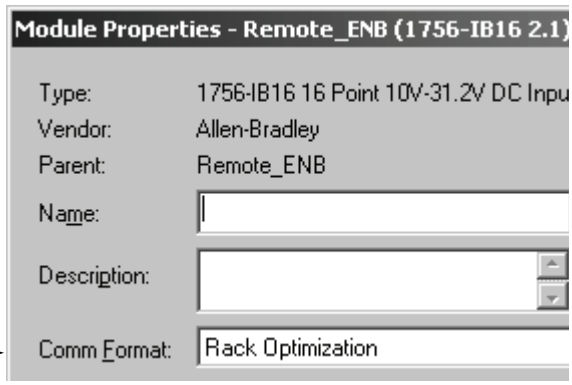
Формат обмена данными

Выбранный вами формат обмена данными определяет структуру данных для тегов, связанных с определенными модулем. Многие модули ввода/вывода поддерживают различные форматы. Каждый формат использует свою структуру данных. Выбранный вами формат обмена данными также определяет:

- Прямое или оптимизированное для рэка соединение
- Отношения собственности

Прямое или оптимизированное для рэка соединение

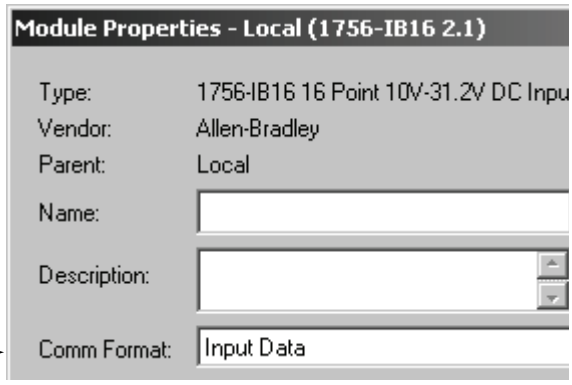
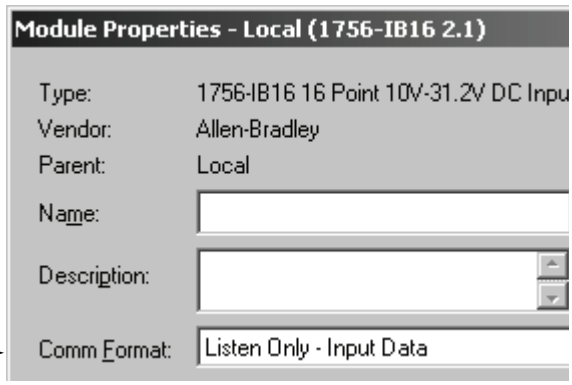
Контроллер Logix5000 использует соединения для передачи данных ввода/вывода. Эти соединения могут быть прямыми или оптимизированными для рэка.

Термин	Определение
Прямое соединение (direct connection)	Прямое соединение – это линия передачи данных в реальном времени между контроллером и модулем ввода/вывода. Контроллер поддерживает и контролирует такое соединение с модулем ввода/вывода. Всякое нарушение соединения, например, ошибка модуля или удаление модуля при включенном питании, устанавливает биты ошибки в области данных, относящейся к соответствующему модулю. Прямое соединение – это любое соединение, которое не использует формат Rack Optimization Come Format. → 
Соединение, оптимизированное для рэка (rack-optimized connection)	Для модулей дискретного ввода/вывода вы можете выбрать оптимизированный для рэка обмен данными. Оптимизированное для рэка соединение консолидирует использование соединений между контроллером и модулями дискретного вода/вывода в шасси . Вместо отдельных прямых соединений с каждым модулем ввода/ вывода имеется одно соединение для всего шасси. Соединение, оптимизированное для рэка → 

Отношения собственности

В системе Logix5000 модули осуществляют многоадресную передачу данных. Это означает, что несколько устройств могут одновременно получать одни и те же данные от одного устройства.

При выборе формата обмена данными вы должны задать, будет ли установлен собственник, или же контроллер по отношению к модулю будет лишь слушателем.

Контроллер-собственник	Контроллер, создающий основную конфигурацию и коммуникационное соединение с модулем. Контроллер-собственник записывает данные по конфигурации и может устанавливать соединение с модулем..
	Соединение собственника - это любое соединение, в формате обмена данными которого (поле Comm Format) не содержится Listen-Only (Только прослушивание). → 
Соединение только для прослушивания (Listen-Only connection)	Соединение с модулем ввода/вывода, при котором другой контроллер владеет данными по конфигурации/предоставляет такие данные модулю ввода/вывода. Контроллер, использующий соединение только для прослушивания, лишь контролирует соответствующий модуль. Он не записывает данные конфигурации, и может поддерживать соединение с модулем ввода/вывода лишь тогда, когда контроллер-собственник активно управляет данным модулем ввода/вывода.
	Соединение только для прослушивания → 

Для выбора типа отношений собственности для модуля используйте следующую таблицу:

Если вы используете модуль:	А другой контроллер:	И вы хотите:	То используйте этот тип соединения:
Модуль ввода	не является собственником модуля	→	Собственник (т.е. не только прослушивание)
	является собственником модуля	поддерживать связь с модулем, если он потеряет связь с другим контроллером	Собственник (т.е. не только прослушивание) Используйте такую же конфигурацию, как у другого контроллера-собственника.
		прекратить связь с модулем, если он потеряет связь с другим контроллером	только прослушивание
Модуль вывода	не является собственником модуля	→	собственник (т.е. не только прослушивание)
	является собственником модуля	→	только прослушивание

Управление модулями ввода заметно отличается от управления модулями вывода.

При управлении	Это отношение собственности	Означает
Модулями ввода	Собственник	Модуль ввода конфигурируется контроллером, который устанавливает соединение в качестве собственника. Конфигурирующим является контроллер первым установивший соединение собственника. Когда модуль ввода будет сконфигурирован (и контроллер станет его собственником), другие контроллеры смогут установить соединения собственника с этим модулем. Это позволяет дополнительным собственникам продолжать получать широковещательные данные при нарушении соединения первоначального контроллера-собственника с данным модулем. Все остальные дополнительные собственники должны иметь идентичные с первоначальным контроллером-собственником данные по конфигурации и формат обмена данными, в противном случае попытка установить соединение будет отвергнута.
	Только прослушивание	Когда модуль ввода будет сконфигурирован (и контроллер станет его собственником), другие контроллеры смогут установить соединение с этим модулем только для прослушивания. Эти контроллеры смогут получать широковещательные данные, притом, что собственником модуля является другой контроллер. Если будут нарушены соединения всех контроллеров-собственников с данным модулем ввода, все контроллеры с соединениями только для прослушивания прекратят получать широковещательные данные.
Модулями вывода	Собственник	Модуль вывода конфигурируется контроллером, который устанавливает соединение в качестве собственника. Для модуля вывода разрешается только соединение собственника. Если другой контроллер попытается установить соединение собственника, такая попытка будет отвергнута.
	Только прослушивание	Когда модуль вывода будет сконфигурирован (и контроллер станет его собственником), другие контроллеры смогут установить соединения с этим модулем только для прослушивания. Эти контроллеры смогут получать широковещательные данные, притом, что собственником модуля является другой контроллер. Если будет нарушено соединение контроллера-собственника с данным модулем ввода, все контроллеры с соединениями только для прослушивания прекратят получать широковещательные данные.

Электронный ключ

ВНИМАНИЕ

Будьте осторожны при отключении функции электронного ключа. При неправильном использовании эта опция может привести к травмам или смерти персонала, материальному ущербу или экономическим потерям.

При конфигурировании модуля вы задаете номер слота для данного модуля. Однако существует возможность преднамеренной или случайной установки другого модуля в этот слот.

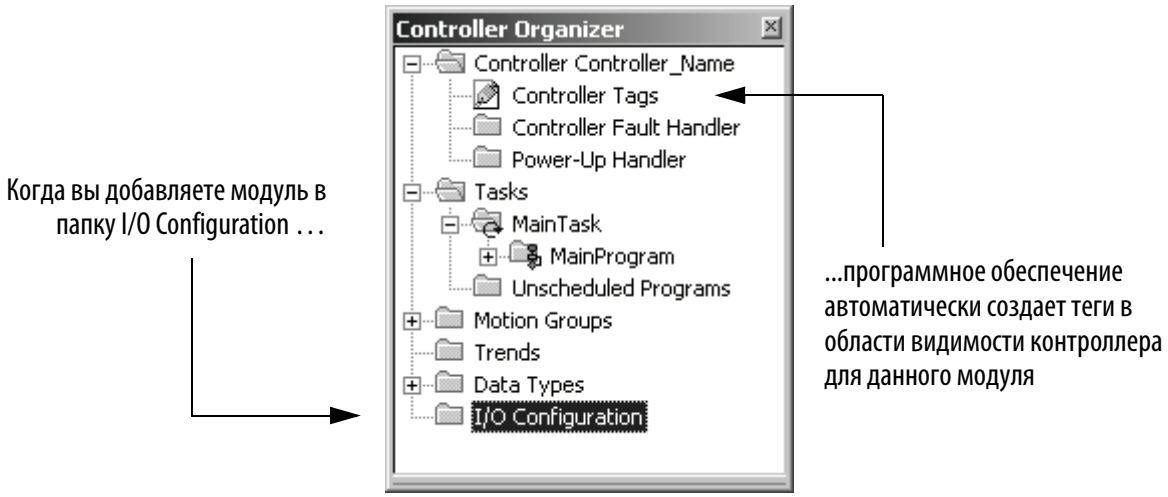
Электронный ключ позволяет вам защитить вашу систему от случайной установки в слот не того модуля. Задаваемая вами опция ключа определяет, насколько устанавливаемый в слот модуль должен соответствовать заданной для данного слота конфигурации.

Если:	То выберите:
Вся информация должна соответствовать по: · типу · каталожному номеру · производителю · основному и дополнительному номеру версии	Exact Match (Точное соответствие)
Соответствие всей информации, кроме дополнительного номера версии	Compatible Module (Совместимый модуль)
Соответствие не проверяется	Disable Keying (Отключить ключ)

Адресация данных ввода/вывода

Информация ввода/вывода представляется в виде набора тегов.

- Каждый тег использует определенную структуру данных. Эта структура зависит от характеристик конкретного модуля ввода/вывода.
- Имена тегов определяются местоположением соответствующего модуля ввода/вывода в системе.



Адрес ввода/вывода имеет следующий формат:

Location | *:Slot* | *:Type* | *.Member* | *.SubMember* | *.Bit*

= Опционально

Где	Это
Location	Местоположение в сети LOCAL = то же шасси, что и у контроллера *ADAPTER_NAME* = обозначает удаленный коммуникационный адаптер или модуль моста
Slot	Номер слота модуля ввода/вывода в соответствующем шасси
Type	Тип данных I = ввод O = вывод C = конфигурация S = статус
Member	Конкретные данные, поступающие от модуля ввода/вывода; зависят от того, данные какого типа могут храниться в данном модуле. · Для дискретного модуля, в члене Data (Данные) обычно хранятся двоичные значения ввода или вывода. · Для аналогового модуля, в члене Channel (CH#) (Канал) обычно хранятся данные для канала.
SubMember	Конкретные данные, относящиеся к члену (Member).
Bit	Конкретная точка в модуле дискретного ввода/вывода; зависит от размера модуля ввода/вывода (0-31 для 32-канального модуля)

Буферизация ввода/вывода

Буферизация – это способ, при котором логика не обращается непосредственно к тегам реальных устройств ввода/вывода и не производит действий с ними. Вместо этого логика использует копию данных ввода/вывода. Используйте буферизацию ввода/вывода в следующих случаях:

- Для предотвращения изменения входного или выходного значения во время выполнения программы. (Ввод/вывод обновляется асинхронно по отношению к выполнению логики.)
- Для копирования тега ввода или вывода в член структуры или элемент массива.

Буферизация ввода/вывода

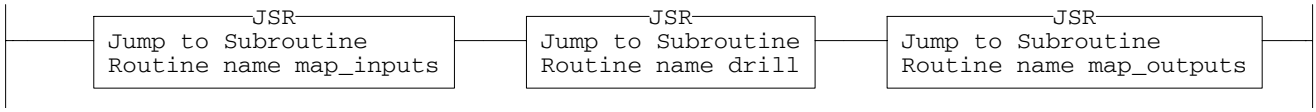
Для буферизации ввода/вывода выполните следующие действия:

1. В цепи перед логикой для функции (ий) скопируйте или переместите данные из требуемых тегов ввода в соответствующие им теги буфера.
2. В логике для функции (ий) обратитесь к тегам буфера.
3. В цепи после функции (ий) скопируйте данные из тегов буфера в соответствующие теги вывода.

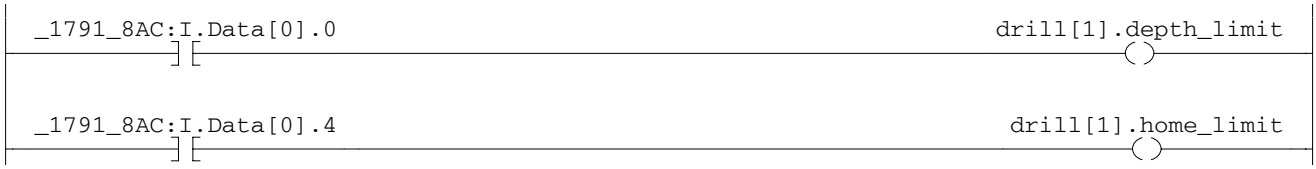
В следующем примере входные и выходные значения копируются в теги структуры для сверлильного станка.

ПРИМЕР Буферизация ввода/вывода

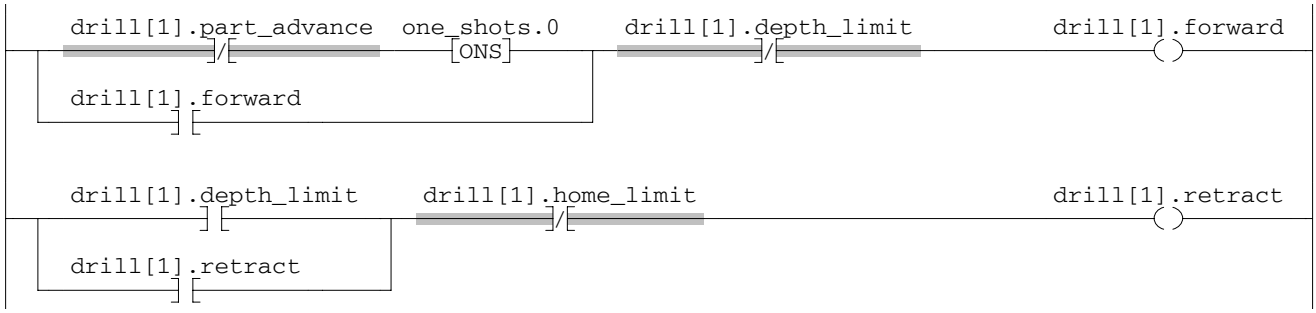
Главная процедура программы выполняет следующие подпрограммы в указанной последовательности.



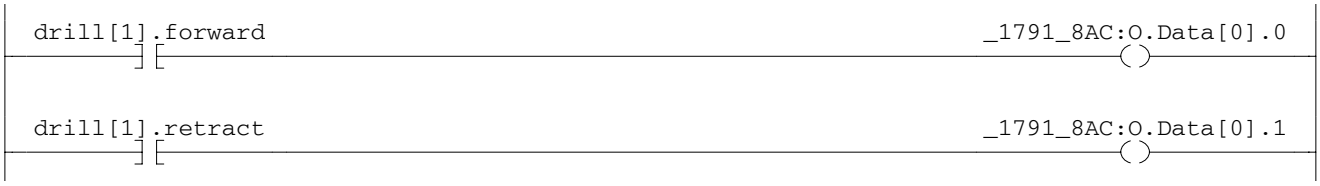
Процедура map_inputs копирует значения из устройств ввода в соответствующие теги, используемые в процедуре drill.



Процедура drill выполняет логику для сверлильного станка.



Процедура map_outputs копирует значения тегов вывода из процедуры drill в соответствующие устройства вывода.



42369

В следующем примере используется инструкция CPS для копирования массива данных, отображающих устройства ввода в сети DeviceNet.

ПРИМЕР Буферизация ввода/вывода

В Local:0:I.Data хранятся входные данные для сети DeviceNet, подключенной к модулю 17563 DNB в слоте 0. Чтобы синхронизировать входы с приложением, инструкция CPS копирует входные данные в input_buffer.

- Во время копирования данных при помощи инструкции CPS никакие обновления ввода/вывода не могут изменить данные.
- При выполнении приложения оно в качестве своих входов использует входные данные, находящиеся в input_buffer.

42578

Организация тегов

Введение

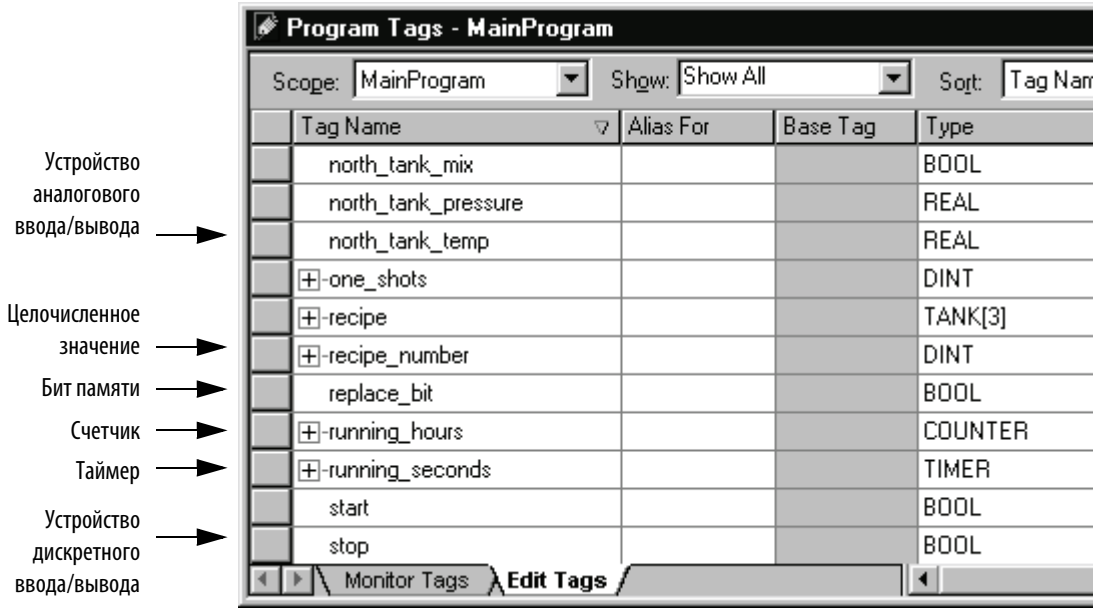
Для обращения к данным (переменным) в контроллере Logix5000 используется тег (буквенно-цифровое имя).

Термин	Определение
Тег	Текстовое имя области памяти контроллера, где хранятся данные. · Теги являются основным механизмом распределения памяти, обращения к данным из логики и контроля данных. · Минимальный размер памяти, выделяемой для тега – 4 байта. · Когда вы создаете тег для хранения данных, занимающих менее 4 байт, контроллер выделяет 4 байта, но данные заполняют лишь требуемый им объем.

Контроллер использует имя тега для внутренних целей, и ему не требуются перекрестные ссылки на физический адрес.

- В традиционных программируемых контроллерах каждый элемент данных идентифицируется физическим адресом.
  - Адреса имеют фиксированный числовой формат, определяемый типом данных, например, N7:8, P8:3.
  - Для облегчения интерпретации логики требуются символьные обозначения.
- В контроллерах Logix5000 нет фиксированного числового формата. Данные идентифицируются самим именем тега. Это позволяет вам:
  - организовывать ваши данные таким образом, чтобы они отображали ваше технологическое оборудование;
  - документировать (при помощи имен тегов) ваше приложение в процессе его разработки.

ПРИМЕР Теги



Тип тега

Тип тега определяет действие данного тега в вашем проекте.

Если вы хотите, чтобы тег:	Выберите этот тип тега:
хранил значение или значения для использования логикой внутри проекта	Base (Базовый)
представлял другой тег	Alias (Псевдоним)
отправлял данные в другой контроллер	Produced (Производимый)
получал данные от другого контроллера	Consumed (Потребляемый)

Если вы собираетесь использовать производимые или потребляемые теги, вы должны выполнить дополнительные действия при организации ваших тегов.

Тип данных

Термин	Определение
Тип данных	Тип данных определяет тип хранимых в теге данных, например, двоичное, целочисленное значение, значение с плавающей запятой, строковое и т.п.
Структура	Тип данных, представляющий собой комбинацию других типов данных. · Структура имеет свой формат, создающий уникальный тип данных, предназначенный для конкретных целей. · Каждый отдельный тип данных внутри структуры называется членом. · Члены, как и теги, имеют имя и тип данных. · Контроллер Logix5000 содержит набор стандартных структур (типов данных), предназначенных для использования конкретными инструкциями, такими как таймеры, счетчики, функциональный блоки и т.д. · Вы можете создать свои собственные структуры, которые называются задаваемым пользователем (пользовательским) типом данных.

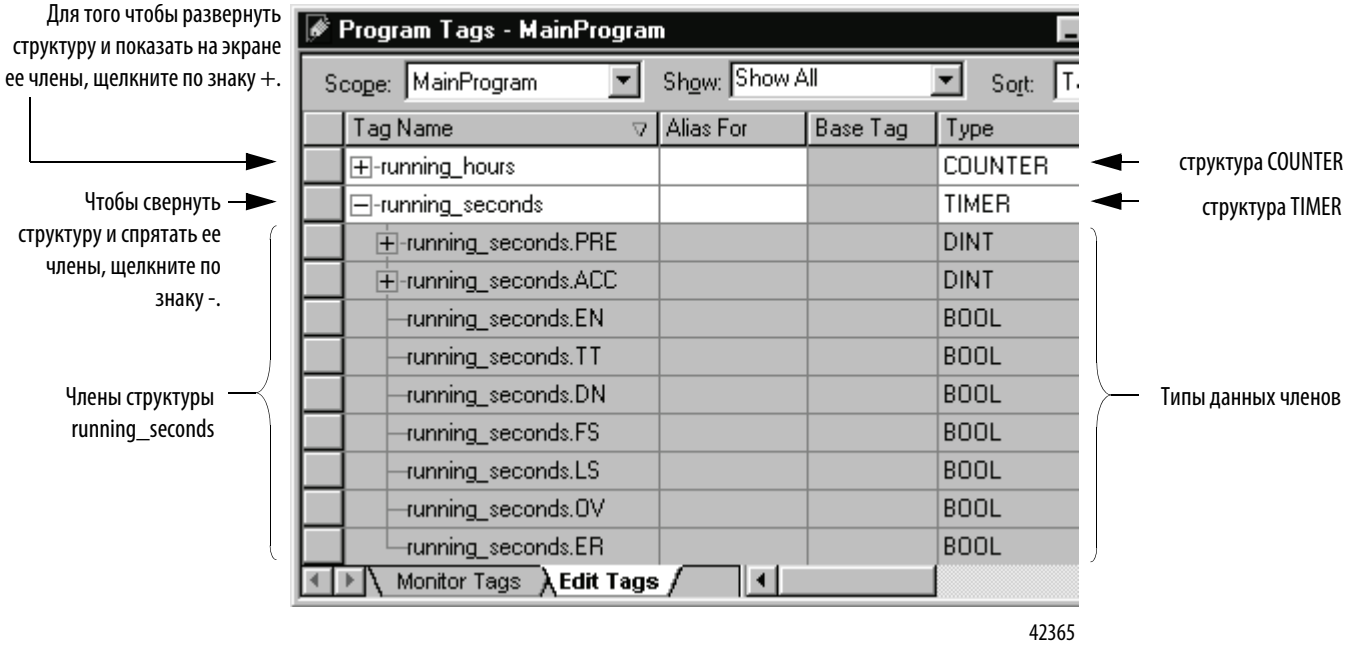
В следующей таблице указываются наиболее распространенные типы данных и их применение.

Для	Выберите
аналогового устройства в режиме с плавающей запятой	REAL
аналогового устройства в целочисленном режиме (для очень большой частоты дискретизации)	INT
символов ASCII	String
бита	BOOL
счетчика	COUNTER
точки дискретного ввода/вывода	BOOL
числа с плавающей запятой	REAL
целого числа	DINT
секвенсера	CONTROL
таймера	TIMER

Минимальный размер памяти, выделяемый для тега – 4 байта. Когда вы создаете тег для хранения данных, занимающих менее 4 байтов, контроллер выделяет 4 байта, но данные заполняют лишь требуемый им объем.

Тип данных	Биты			
	31 16	15 8	7 1	0
BOOL	не используется			0 или 1
SINT	не используется		от -128 до +127	
INT	не используется	от -32,768 до +32767		
DINT	от -2,147,483,648 до +2,147,483,647			
REAL	от $-3.40282347E^{38}$ до $-1.17549435E^{-38}$ (отрицательные значения) 0 от $1.17549435E^{-38}$ до $3.40282347E^{38}$ (положительные значения)			

Типы данных COUNTER и TIMER являются примерами широко используемых структур.

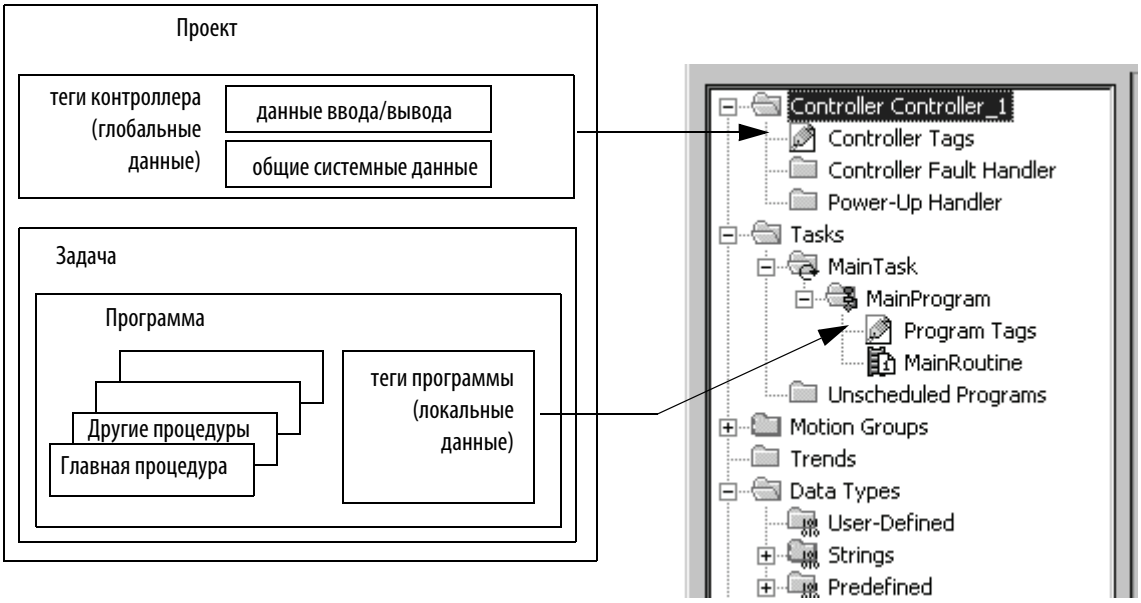


Чтобы скопировать данные в структуру, используйте инструкцию COP.

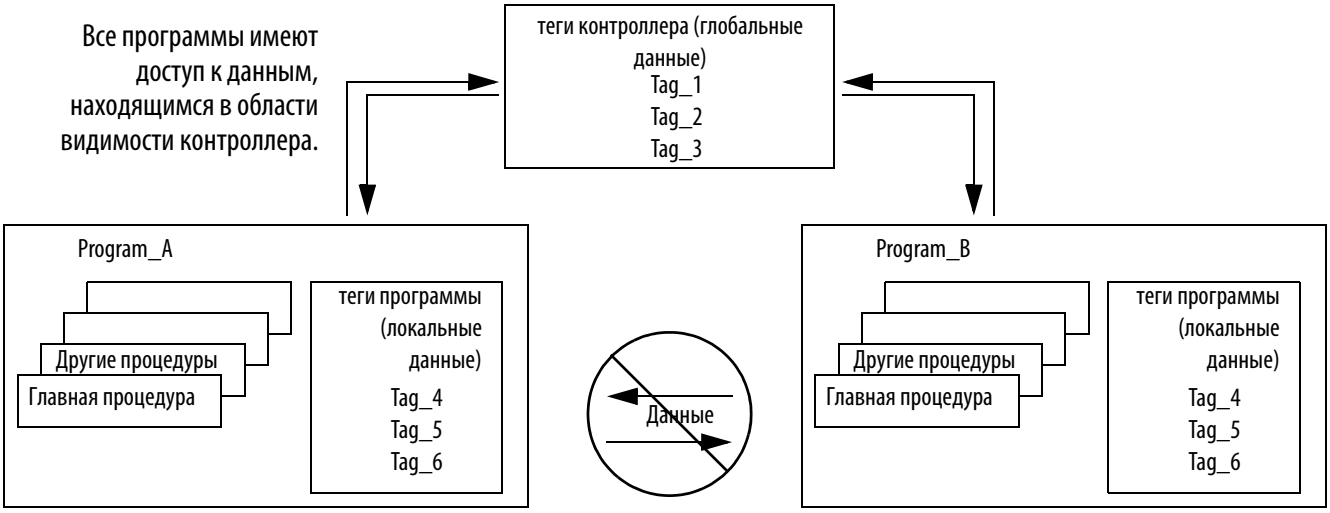
Обратитесь к публикации 1756-RM003 «Справочное руководство по общему набору инструкций контроллеров Logix5000».

Область видимости

При создании тега вы определяете его либо как тег контроллера (глобальные данные), либо как программный тег для конкретной программы (локальные данные).



Контроллер Logix5000 позволяет вам разделить ваше приложение на несколько программ, с отдельными данными каждая. При этом наличие тегов с идентичными именами в разных программах разрешено. Это облегчает многократное использование управляющей логики и тегов в нескольких программах.



Данные в области видимости программы изолированы от других программ:

- Процедуры не могут обращаться к данным, находящимся в области видимости другой программы.
- Вы можете многократно использовать в нескольких программах имя тега, находящегося в области видимости программы. Например, обе программы (Program_A и Program_B) могут иметь тег программы с именем Tag_4.

Избегайте использования одного и того же имени для тега контроллера и тега программы. Внутри программы вы можете обращаться к тегу контроллера, если тег с таким же именем существует в качестве программного тега для данной программы.

Определенные теги должны находиться в области видимости контроллера (теги контроллера).

Если вы хотите использовать тег:	То задайте эту область видимости:
в нескольких программах проекта	область видимости контроллера (теги контроллера)
в инструкции Message (MSG)	
для производства или потребления данных	
для обмена данными с терминалом PanelView	
для других целей, кроме вышеперечисленного	область видимости программы (теги программы)

Рекомендации по созданию тегов

При создании тегов для проекта Logix5000 придерживайтесь следующих рекомендаций:

Указание:	Подробное описание:
❑ 1. Создайте пользовательские типы данных.	Пользовательские типы данных (структуры) позволяют вам организовать данные типы таким образом, чтобы они соответствовали вашей установке или процессу. Пользовательский тип данных имеет следующие преимущества: · Один тег содержит все данные по определенному аспекту вашей системы. Это обеспечивает совместное размещение взаимосвязанных данных и простоту их поиска независимо от типа данных. · Каждая отдельная часть данных (член) получает описательное имя. Это автоматически создает начальный уровень документирования вашей логики. · Вы можете использовать этот тип данных для создания множества тегов при том же макете данных. Например, используйте пользовательский тип данных для хранения всех параметров резервуара, включая температуры, давления, позиции клапанов и предварительно установленные значения. Затем создайте тег для каждого из ваших резервуаров на основе этого типа данных.

Указание:	**Подробное описание:**
❑ 2. Используйте массив для быстрого создания группы аналогичных тегов.	Массив создает множество экземпляров данных определенного типа под общим именем тега. · Массивы позволяют вам организовать блок тегов, использующих один и тот же тип данных и выполняющих аналогичную функцию. · Организуйте данные в одно-, двух- или трехмерные массивы в зависимости от того, что отображают эти данные. Например, используйте двухмерный массив для организации данных по хранилищу сырья с несколькими резервуарами. Каждый элемент массива соответствует одному резервуару. Местоположение элемента в массиве соответствует географическому расположению резервуара. **Важно**: Минимизируйте использование массивов BOOL. Многие инструкции массивов не работают для массивов BOOL. Это затрудняет инициализацию и очистку массива данных типа BOOL. · Используйте массив BOOL в основном для объектов битового уровня экрана PanelView. · Также можно использовать отдельные биты тега DINT или массив данных типа DINT.
❑ 3. Воспользуйтесь тегами в области видимости программы.	Если вы хотите, чтобы у вас было множество тегов с одним и тем же именем, задайте каждый тег как программный тег (тег в области видимости программы) для другой программы. Это позволяет вам многократно использовать в разных программах и логику, и имена тегов. Избегайте использования одного и того же имени для тегов контроллера и программных тегов. Внутри программы вы не можете обращаться к тегу контроллера, если существует программный тег с таким же именем для этой программы. Определенные теги должны иметь область видимости контроллера (тег контроллера).

Если вы хотите использовать тег:	**Задайте эту область видимости:**
в нескольких программах проекта	область видимости контроллера (controller scope) (теги контроллера)
в инструкции Message (MSG)	
для производства или потребления данных	
для обмена данными с терминалом PanelView	
для других целей, кроме вышеперечисленного	область видимости программы (program scope) (теги программы)

Указание:	**Подробное описание:**
❑ 4. Для целочисленных значений используйте тип данных DINT.	Для увеличения эффективности вашей логики минимизируйте использование типов данных SINT и INT. Везде, где это возможно, для целочисленных значений используйте тип данных DINT. · Контроллер Logix5000 выполняет сравнение или манипуляции со значениями как с 32-разрядными величинами (тип данных DINT или REAL). · Перед использованием значения контроллер преобразует значение SINT или INT в DINT или REAL. · Если приемником данных является тег SINT или INT, то контроллер преобразует значение обратно в SINT или INT. · Преобразование в или из SINT или INT происходит автоматически, не требуя никакого дополнительного программирования. Однако оно требует дополнительного времени выполнения и дополнительной памяти.
❑ 5. Ограничьте имя тега 40 символами.	Правила для имени тега: · допускаются только буквы (A-Z или a-z), цифры (0-9) и знаки подчеркивания (_) · должно начинаться с буквы или знака подчеркивания · может включать не более 40 символов · не должно включать несколько знаков подчеркивания подряд или замыкающий знак подчеркивания (_) · не чувствительно к регистру
❑ 6. Используйте символы разных регистров.	Хотя имена тегов не чувствительны к регистру (буква A верхнего регистра воспринимается так же, как буква a нижнего регистра), имена легче читаются при использовании символов разных регистров.

Эти теги легче читаются:	**Чем эти теги:**
Tank_1	TANK_1
Tank1	TANK1
	tank_1
	tank1

❑ 7. Учитывайте алфавитный порядок тегов.

Программное обеспечение RSLogix5000 отображает на экране теги одной и той же области видимости в алфавитном порядке. Для облегчения контроля взаимосвязанных тегов используйте одинаковые начальные символы для тегов, которые должны отображаться вместе.

Если каждый тег для резервуара будет начинаться с Tank, то эти теги будут отображаться вместе.

Имя тега
Tank_North
Tank_South
…

В противном случае эти теги могут оказаться далеко друг от друга.

Имя тега
North_Tank
…
…
…
South_Tank

Другие теги, начинающиеся с букв o, p, q, и т.д.

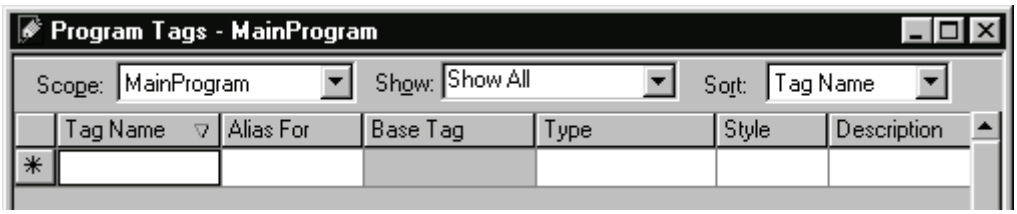
Создание тега

ВАЖНО Программное обеспечение RSLogix 5000 автоматически создает теги, если вы:

- добавляете элемент в последовательную функциональную схему (ПФС);
- добавляете инструкцию функционального блока в функциональную блок-схему.

Окно Tags позволяет создавать и редактировать теги при помощи представления тегов в виде электронной таблицы.

1. В меню Logic (Логика) выберите Edit Tags (Редактирование тегов).
2. Выберите область видимости для тега.



42350

Если вы будете использовать тег:	То выберите:
в нескольких программах внутри проекта	*name_of_controller(контроллер)*
в качестве производителя или потребителя	
в сообщении	
только в одной программе внутри проекта	программу, которая будет использовать этот тег

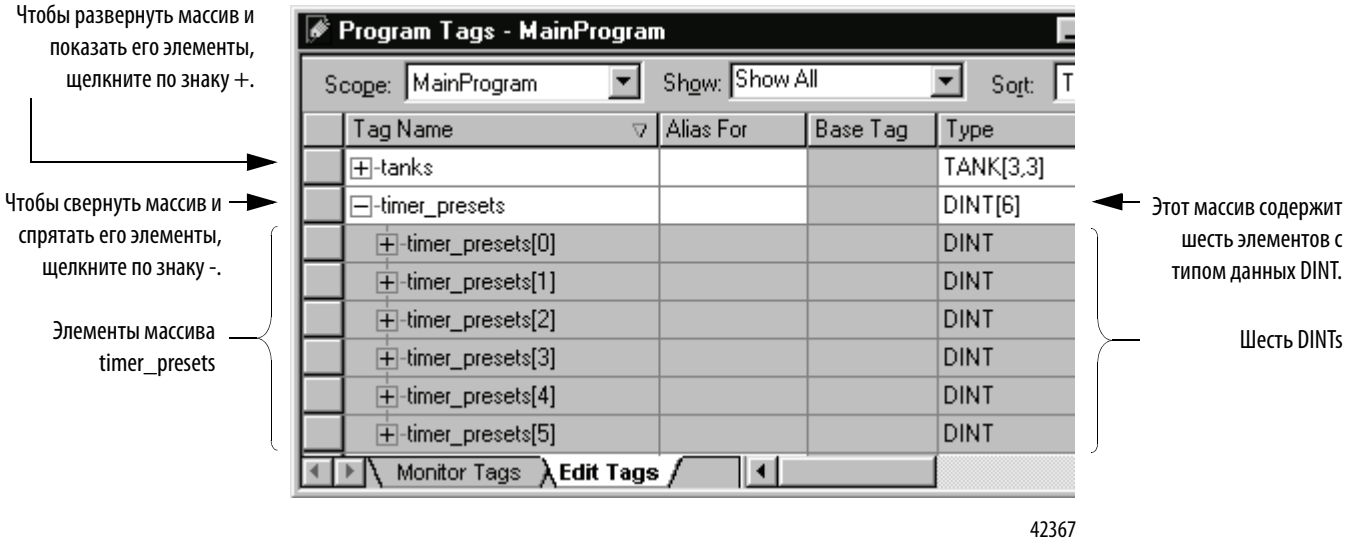
3. Введите имя, тип данных и описание (опционально) для тега.

Создание массива

Контроллеры Logix5000 также позволяют вам использовать массивы для организации данных.

Термин	Определение
Массив	Тег, содержащий блок из многих элементов данных. · Массив аналогичен файлу. · Каждая отдельная часть данных внутри массива называется элементом. · Каждый элемент использует один и тот же тип данных. · Тег массива занимает в контроллере непрерывный блок памяти, один элемент за другим. · Вы можете использовать инструкции массива и секвенсора для выполнения действий с элементами массива или их индексации по всему массиву. · Вы можете организовать данные в одно-, дву- или трехмерный блок.

Индекс идентифицирует каждый отдельный элемент массива. Диапазон индексов – от 0 до количества элементов минус 1 (индексация с нуля).



42367

В следующем примере структура сравнивается с массивом.

Это тег, использующий структуру (тип данных) Timer.

Имя тега	Тип данных
⊟ Timer_1	TIMER
⊞ Timer_1.PRE	DINT
⊞ Timer_1.ACC	DINT
Timer_1.EN	BOOL
Timer_1.TT	BOOL

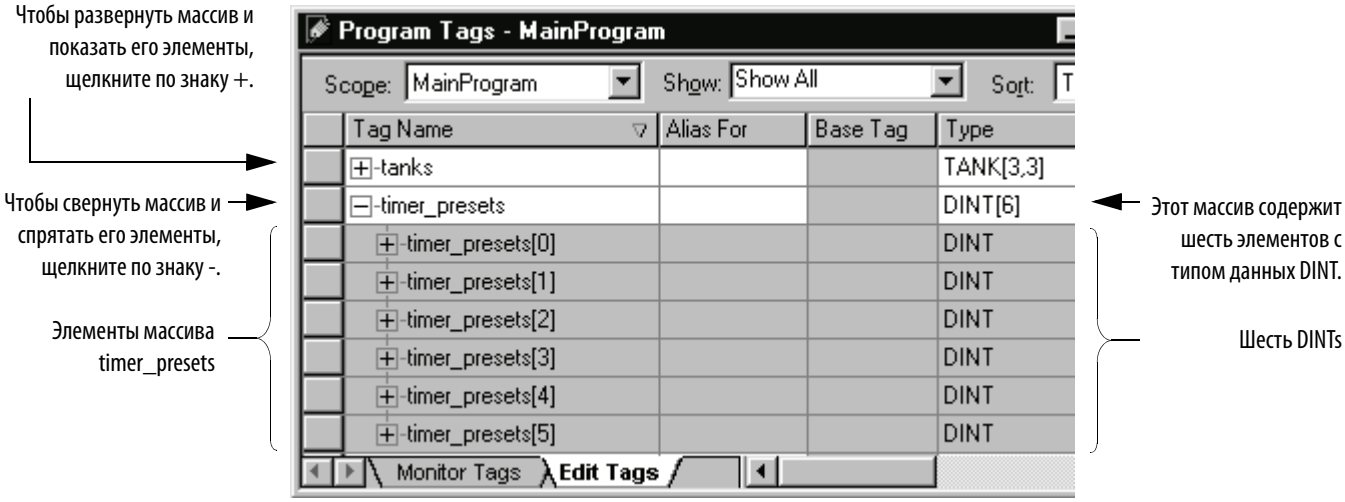
Это тег, использующий массив с типом данных Timer.

Имя тега	Тип данных
⊟ Timers	TIMER[3]
⊞ Timer[0]	TIMER
⊞ Timer[1]	TIMER
⊞ Timer[2]	TIMER

ПРИМЕР

Одномерный массив

В этом примере одна инструкция таймера определяет продолжительность нескольких операций. Для каждой операции требуется свое предварительно заданное значение. Поскольку все значения имеют один и тот же тип данных (DINT), используется массив.



42367

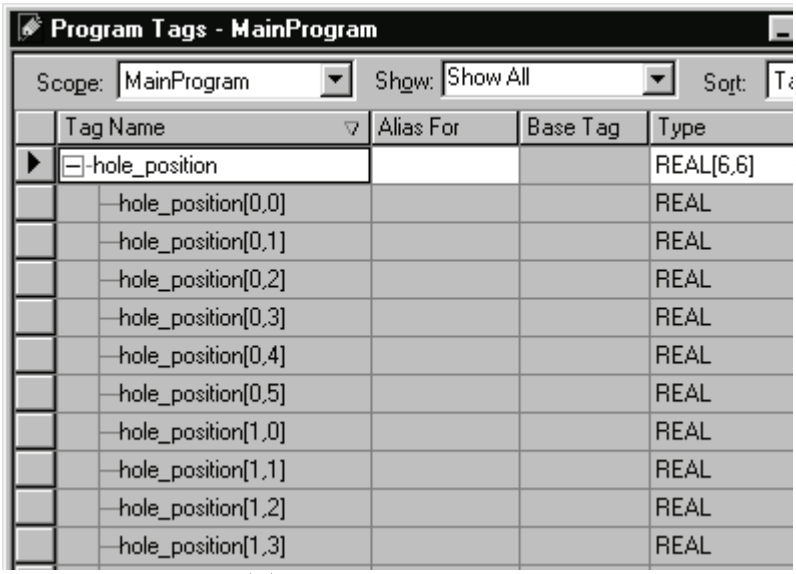
ПРИМЕР

Двумерный массив

Сверлильный станок может высверливать в книжном блоке от одного до пяти отверстий. Станку необходимо значение для положения каждого отверстия относительно передней кромки блока. Для организации этих значений в конфигурации используется двумерный массив. Первый индекс обозначает отверстие, которому соответствует данное значение, а второй указывает на то, сколько отверстий будет просверлено (от одного до пяти).

	Индекс второй размерности						Описание
Индекс первой размерности	0	1	2	3	4	5	
0							
1		1.5	2.5	1.25	1.25	1.25	расстояние первого отверстия от передней кромки блока
2			8.0	5.5	3.5	3.5	расстояние второго отверстия от передней кромки блока
3				9.75	7.5	5.5	расстояние третьего отверстия от передней кромки блока
4					9.75	7.5	расстояние четвертого отверстия от передней кромки блока
5						9.75	расстояние пятого отверстия от передней кромки блока

В окне Tags (Теги) элементы представлены в показанном ниже порядке.



Этот массив содержит двумерную решетку элементов - шесть элементов на шесть элементов.

42367

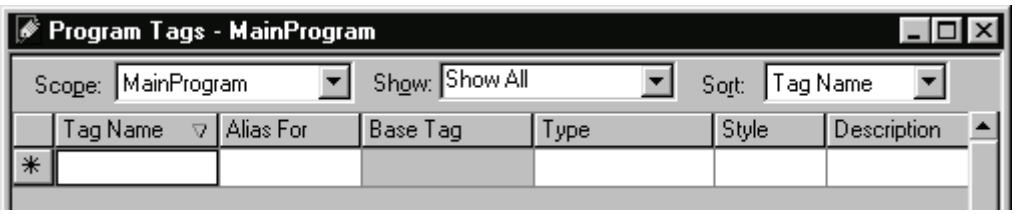
Правая размерность увеличивается на единицу до своего максимального значения, затем отсчет начинается заново.

Когда правая размерность возвращается к исходному значению, левая размерность увеличивается на единицу

Создание массива

Для создания массива вы должны создать тег и задать размерность для соответствующего типа данных.

1. В меню Logic (Логика) выберите Edit Tags (Редактирование тегов).



42350

2. Введите имя тега и выберите область видимости для данного тега:

Если вы будете использовать тег:	То выберите:
в нескольких программах внутри проекта	*name_of_controller(контроллер)*
в качестве производителя или потребителя	
в сообщении	
только в одной программе внутри проекта	программу, которая будет использовать этот тег

3. Задайте размерность массива:

Если тег представляет собой:	То введите:	Где:
одномерный массив	*data_type[x]*	*data_type* – это тип данных, хранимых в теге. x – это количество элементов первого измерения. y – это количество элементов второго измерения. z – это количество элементов третьего измерения.
двумерный массив	*data_type[x,y]*	
трехмерный массив	*data_type[x,y,z]*	

Создание пользовательского типа данных

Пользовательские типы данных (структуры) позволяют вам организовать ваши данные в соответствии с вашей установкой или процессом.

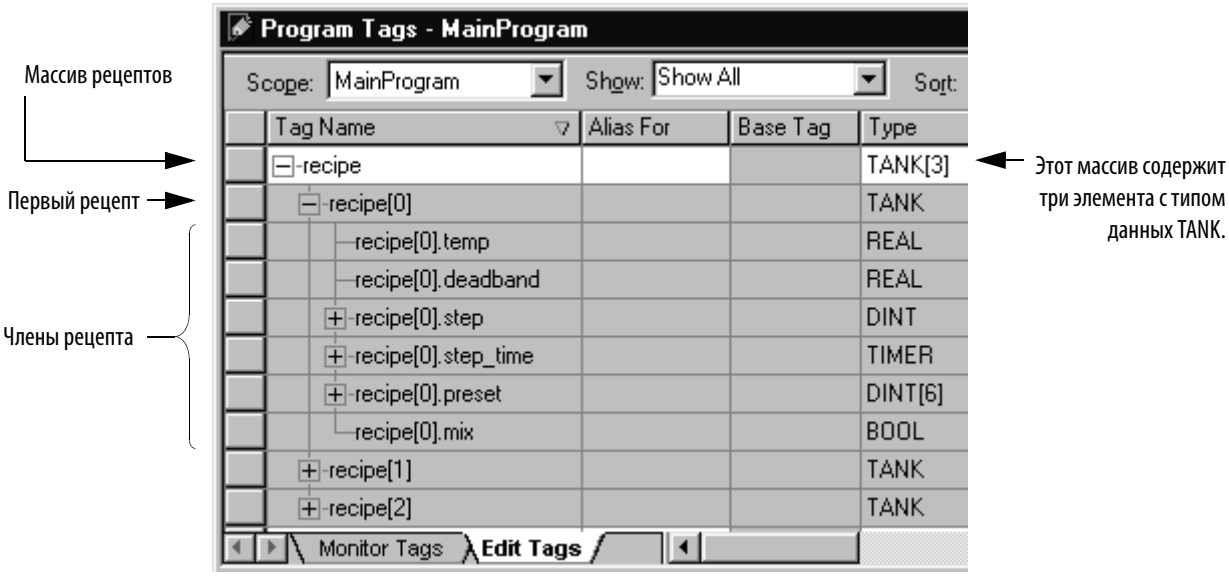
ПРИМЕР

Пользовательский тип данных для хранения рецепта

В системе из нескольких резервуаров для каждого из них могут использоваться разнообразные рецепты. Поскольку для рецепта требуется сочетание различных типов данных (REAL, DINT, BOOL и т.д.), используется пользовательский тип данных.

Имя (типа данных): TANK	
Имя члена	**Тип данных**
Temp	REAL
Deadband	REAL
Step	DINT
Step_time	TIMER
Preset	DINT[6]
Mix	BOOL

Массив, использующий такой тип данных, будет выглядеть следующим образом:



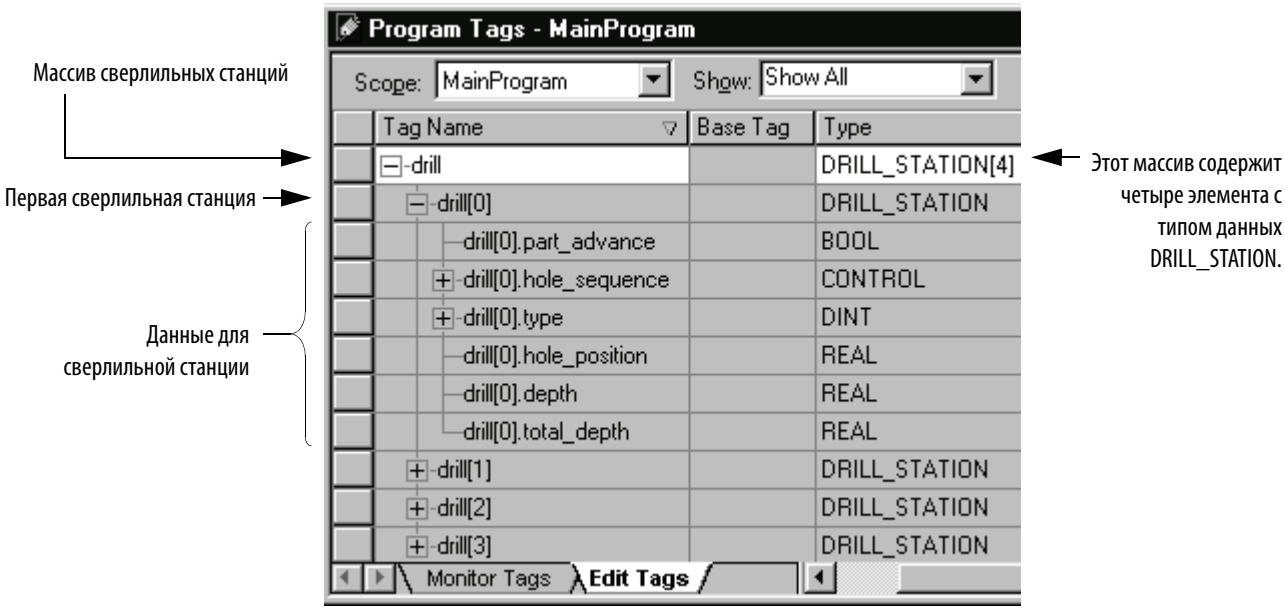
ПРИМЕР

Пользовательский тип данных для хранения данных, необходимых для работы станка

Поскольку для нескольких сверлильных станций требуется следующее сочетание данных, создается пользовательский тип данных.

Имя (типа данных): DRILL_STATION	
Имя члена	**Тип данных**
Part_advance	BOOL
Hole_sequence	CONTROL
Type	DINT
Hole_position	REAL
Depth	REAL
Total_depth	REAL

Массив, использующий такой тип данных, будет выглядеть следующим образом:



Рекомендации по пользовательским типам данных

При создании пользовательского типа данных помните о следующем:

- Если вы включаете члены для устройств ввода/вывода, вы должны использовать логику для копирования данных между членами структуры и соответствующими тегами ввода/вывода. См. раздел «Адресация данных ввода/вывода» на странице 13.
- Если вы включаете массив в качестве члена, ограничьте массив одним измерением. Многомерные массивы не допускаются в пользовательском типе данных.
- При использовании типов данных BOOL, SINT или INT размещайте члены одного типа данных друг за другом:

Более эффективно

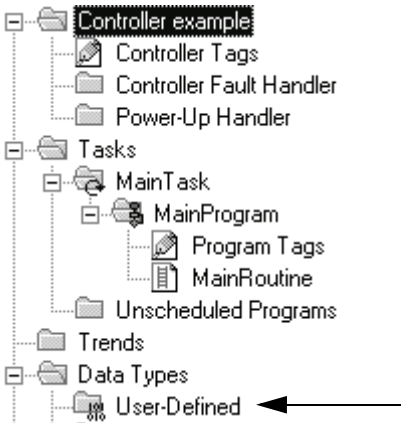
BOOL
BOOL
BOOL
DINT
DINT

Менее эффективно

BOOL
DINT
BOOL
DINT
BOOL

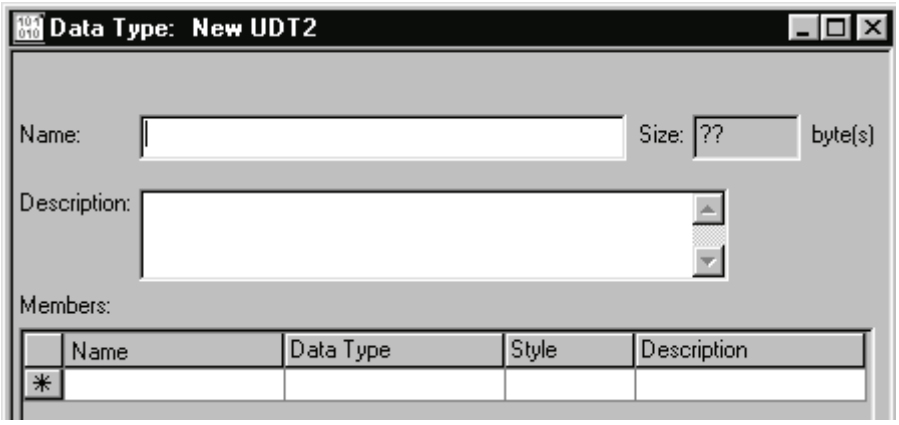
Создание пользовательского типа данных

1. 1.В папке User-Defined (Пользовательский) щелкните правой кнопкой мыши по User-Defined (Пользовательский) и выберите New Data Type (Новый тип данных).



2. Введите имя и описание массива.

3. Для каждого члена массива введите имя, тип данных, стиль и описание (по желанию).



42196

Ограничивайте массивы одним измерением.

Для отображения значения члена в другом стиле (другой системе счисления), выберите стиль

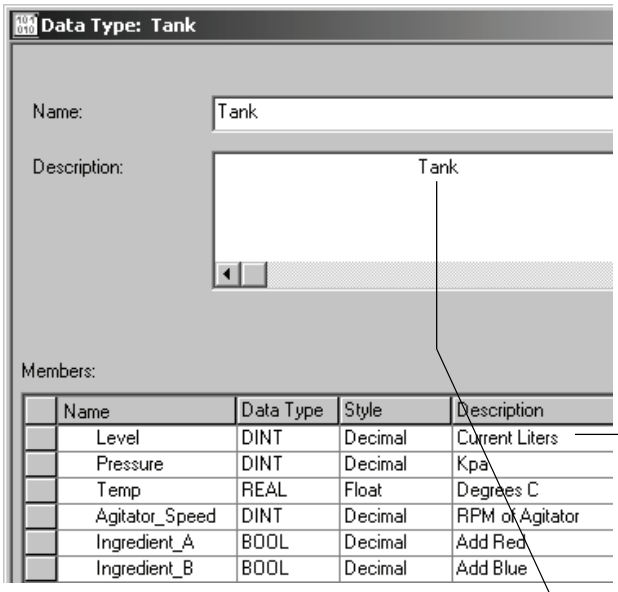
4. Нажмите Apply (Применить).
5. Добавьте столько членов, сколько необходимо.

Описание пользовательского типа данных

RSLogix 5000 версии 13.0 и выше

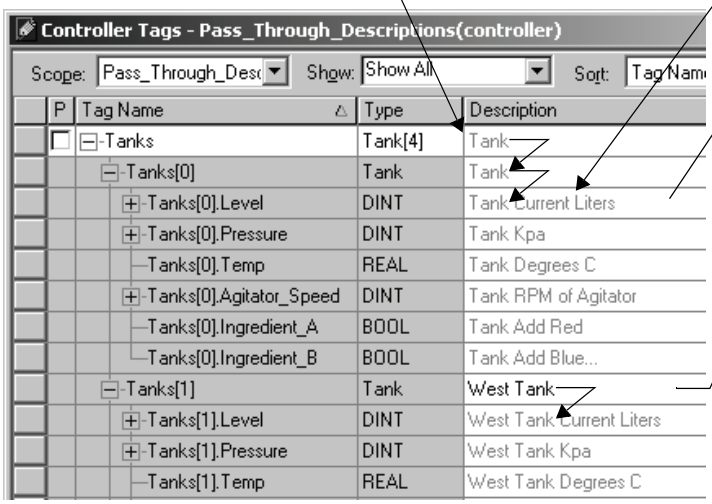
Программное обеспечение RSLogix 5000 позволяет вам автоматически создавать описания для тегов на основе описаний, содержащихся в ваших пользовательских типах данных. Это значительно сокращает время, затрачиваемое вами на документирование вашего проекта.

При организации пользовательских типов данных помните о следующих возможностях программного обеспечения RSLogix 5000:



Сквозное использование описаний - когда это возможно, RSLogix 5000 ищет имеющееся описание для тега, элемента или члена:

- Описания в пользовательских типах данных переносятся на теги, использующие соответствующий тип данных.
- Описание тега массива переносится на элементы и члены этого массива.



Присоединение описания к базовому тегу – RSLogix 5000 автоматически формирует описание для каждого члена тега, использующего пользовательский тип данных. Оно начинается с описания данного тега, к которому присоединяется описание члена, взятое из типа данных.

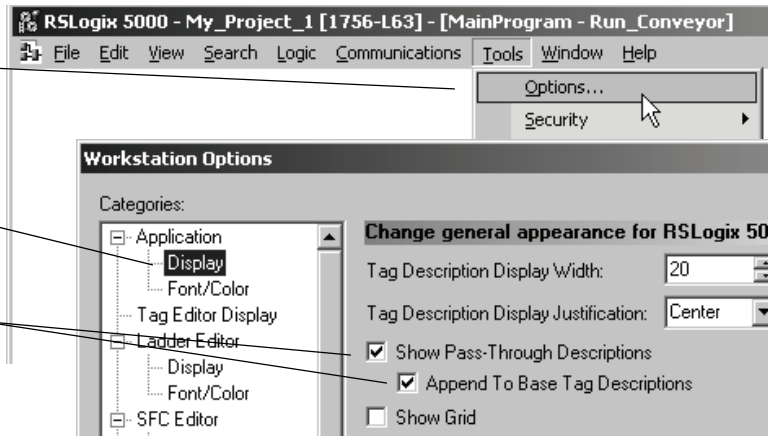
Вставка сквозного описания - Используйте описание типа данных и массива в качестве основы для более конкретных описаний. В данном примере Tank (Резервуар) превратился в West Tank (Западный резервуар).

Для описаний RSLogix 5000 использует различные цвета:

Описание этого цвета:	Соответствует:
серый	сквозному описанию
черный	описанию, введенному вручную

Включение и выключение сквозных (pass-through) и присоединяемых (append) описаний

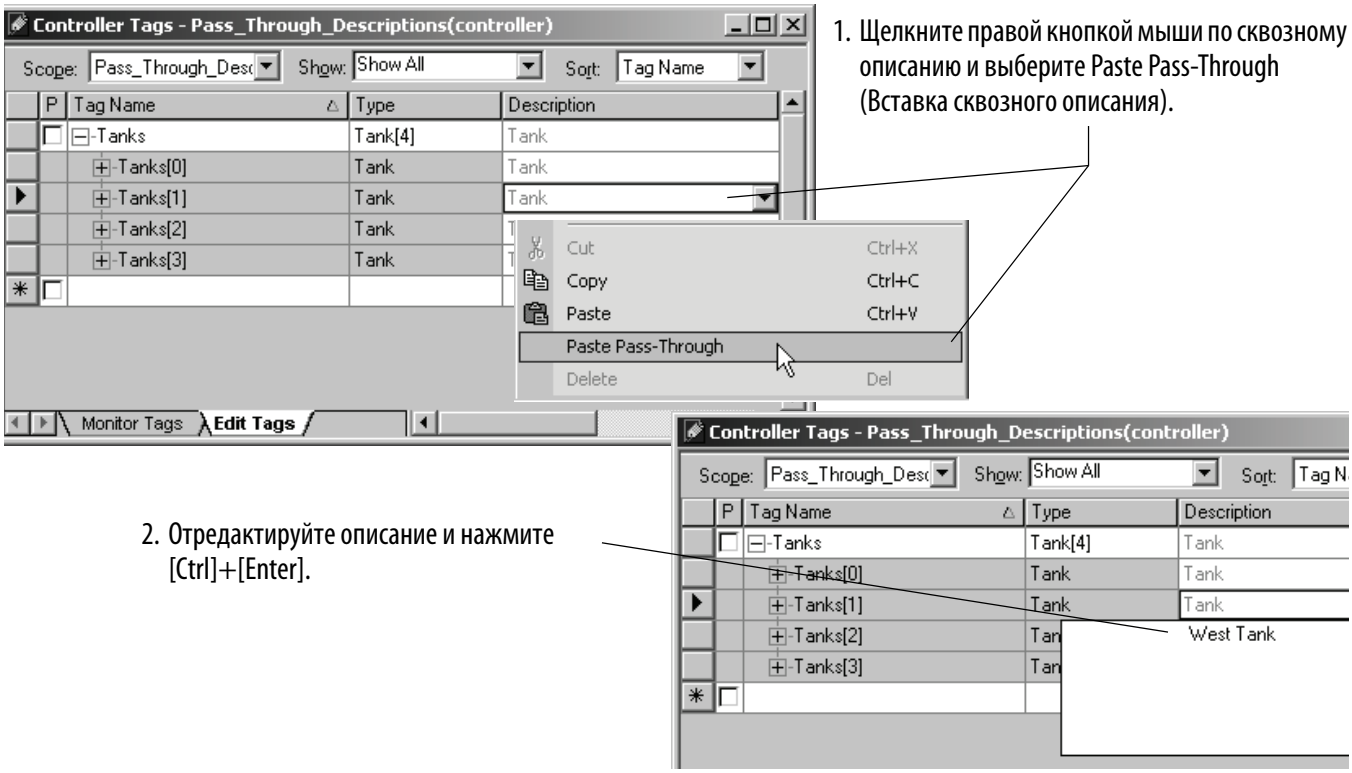
1. В RSLogix 5000 выберите Tools (Инструменты) > Options (Опции).
2. Выберите Application (Приложение) > Display (Отображение).
3. Включите (отметьте) или отключите (снимите отметку) соответствующие опции.



Вставка сквозных описаний

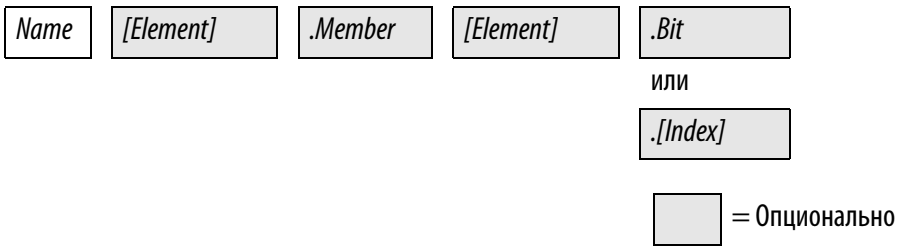
Для использования сквозного описания в качестве основы для более конкретного описания:

1. Щелкните правой кнопкой мыши по сквозному описанию и выберите Paste Pass-Through (Вставка сквозного описания).
2. Отредактируйте описание и нажмите [Ctrl]+[Enter].



Адресация данных в тегах

Имя тега имеет следующий формат:



Где:	Это:
Name	Имя, идентифицирующее конкретный тег.
Element	Индекс или индексы, указывающие на конкретный элемент массива. · Используйте идентификатор элемента лишь в тех случаях, когда тег или член является массивом. · Используйте один индекс для каждого измерения массива. Например: [5], [2,8], [3,2,7]. Для косвенной (динамической) ссылки на элемент используйте тег или численное выражение, дающее номер элемента. · В численном выражении используется комбинация тегов, констант, операторов и функций для вычисления значения. Например, Tag_1 – Tag_2, Tag_3 + 4, ABS (Tag_4). · Значение тега или численного выражения должно находиться в пределах размерности массива. Например, если массив состоит из 10 элементов, то значение тега или численного выражения должно быть в диапазоне от 0 до 9 (10 элементов).
Member	Конкретный член структуры. · Используйте идентификатор члена лишь в тех случаях, когда тег является структурой. · Если структура содержит другую структуру в качестве одного из своих членов, используйте дополнительные уровни формата. Member для идентификации соответствующего члена.
Bit	Конкретный бит целочисленного типа данных (SINT, INT или DINT).
Index	Для косвенной (динамической) ссылки на бит целого числа используйте тег или численное выражение, дающее номер бита. · В численном выражении используется комбинация тегов, констант, операторов и функций для вычисления значения. Например, Tag_1 – Tag_2, Tag_3 + 4, ABS (Tag_4). · Значение тега или численного выражения должно находиться в пределах диапазона битов целочисленного тега. Например, если целочисленным тегом является Dint (32-bits), то значение индекса должно быть в пределах от 0 до 31 (32 бита).

Задание тегов-псевдонимов

Тег-псевдоним позволяет вам создать тег, который будет представлять другой тег.

- Оба тега используют одно и то же значение (значения).
- При изменении значения одного из таких тегов значение другого тега будет так же изменено.

Используйте псевдонимы в следующих случаях:

- при программировании логики до выпуска схем подключения;
- для присвоения описательного имени устройству ввода/вывода;
- для того, чтобы дать более простое имя сложному тегу;
- для использования описательного имени для элемента массива.

Информация о псевдонимах показана в окне тегов.

drill_1_depth_limit - это псевдоним для Local:2:I.Data.3 (точки цифрового ввода). При установке ввода устанавливается и тег-псевдоним.

drill_1_on - это псевдоним для Local:0.0.Data.2 (точки цифрового вывода). При установке тега-псевдонима устанавливается и тег вывода.

north_tank – это псевдоним для tanks[0,1].

Program Tags - MainProgram

Scope: MainProgram Show: Show All Sort: Tag Name

Tag Name	Alias For	Base Tag	Type
⊞-drill_1			DRILL_STAT
drill_1_depth_limit	Local:2:I.Data.3(C)	Local:2:I.Data.3(C)	BOOL
drill_1_forward	Local:0:O.Data.3(C)	Local:0:O.Data.3(C)	BOOL
drill_1_home_limit	Local:2:I.Data.2(C)	Local:2:I.Data.2(C)	BOOL
drill_1_on	Local:0:O.Data.2(C)	Local:0:O.Data.2(C)	BOOL
drill_1_retract	Local:0:O.Data.4(C)	Local:0:O.Data.4(C)	BOOL
⊞-hole_position			REAL[6,6]
machine_on			BOOL
⊞-north_tank	tanks[0,1]	tanks[0,1]	TANK
north_tank_drain			BOOL

42360

(C) указывает на то, что данный тег находится в области видимости контроллера.

Как правило, теги-псевдонимы используются при программировании логики в отсутствие схем подключения:

1. Для каждого устройства ввода/вывода создайте тег с именем, описывающим это устройство, например, conveyor для двигателя конвейера.
2. Запрограммируйте свою логику, используя описательные имена тегов. (Вы даже можете протестировать свою логику без подключения к устройствам ввода/вывода.)
3. Затем, когда будут готовы схемы подключения, добавьте модули ввода/вывода в конфигурацию ввода/вывода данного контроллера.
4. Наконец, преобразуйте описательные теги в псевдонимы для соответствующих точек или каналов ввода/вывода.

Показанная ниже логика была первоначально запрограммирована с использованием описательных имен тегов, таких как stop и conveyor_on. Затем эти теги были преобразованы в псевдонимы для соответствующих устройств ввода/вывода.

stop - это псевдоним для Local:2:I.Data.1 (кнопки останова на пульте оператора).

conveyor_on - это псевдоним для Local:0.0.Data.0

(пускатель двигателя конвейера).

```
        stop                    start
 <Local:2:I.Data.1>    <Local:2:I.Data.0>
 ──────┤ ├──────────┬──────────┤ ├──────────────────────────────
                    │  machine_on          │
                    └──────┤ ├──────────────┘

                   drill_1_on
 machine_on  <Local:0:O.Data.2>  drill_1.part_advance
 ──────┤ ├──────────────( )──────────────────────┤ ├──────────────

    conveyor_on
 <Local:0:O.Data.0>
 ──────┤ ├──────────────────────────────────────────────────────
```

42351

Отображение информации о псевдонимах

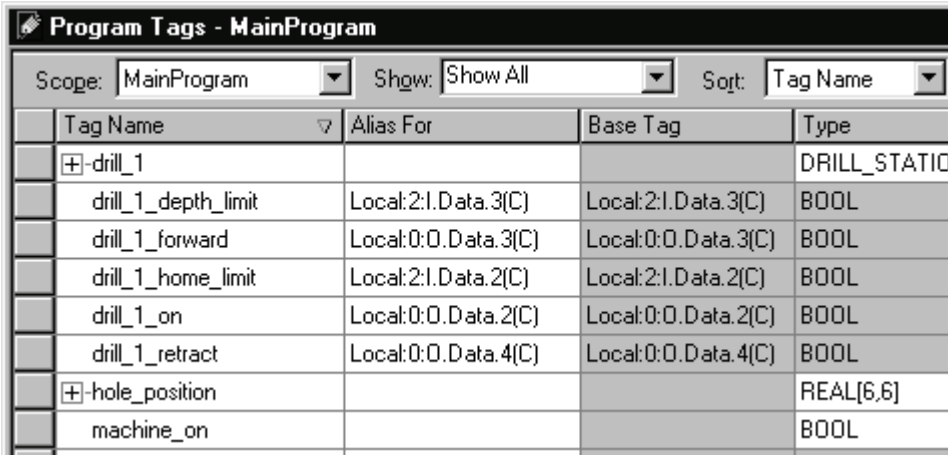
Чтобы показать (в вашей логике) тег, на который указывает псевдоним:

1. В меню Tools (Сервис) выберите Options (Опции).
2. Щелкните по закладке Ladder Display (Экран релейной логики).
3. Установите флажок Show Tag Alias Information (Показать информацию о псевдониме тега).
4. Нажмите OK.

Задание псевдонима

Чтобы задать тег в качестве тега-псевдонима для другого тега:

1. В меню Logic (Логика) выберите Edit Tags (Редактирование тегов).



42360

2. Выберите область видимости данного тега.
3. Щелкните по ячейке Alias For (Псевдоним для) справа от имени тега.

   В ячейке появится значок ▼.
4. Щелкните по значку ▼.
5. Выберите тег, которому будет соответствовать данный псевдоним:

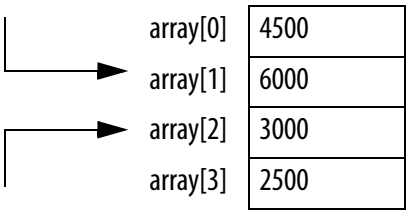
Чтобы:	Выполните следующее:
выбрать тег	Дважды щелкните по имени тега.
выбрать номер бита	A. Щелкните по имени тега. B. Щелкните по значку справа от имени тега ▼. C. Щелкните по требуемому биту.

6. Нажмите на клавишу [Enter] или щелкните по другой ячейке.

Задание косвенного адреса

Если вы хотите, чтобы инструкция обращалась к различным элементам массива, используйте тег в индексе массива (косвенный адрес). Изменяя значение тега, вы изменяете элемент массива, к которому обращается ваша логика.

Когда index равен 1, array[index] указывает сюда.



Когда index равен 2, array[index] указывает сюда.

В следующей таблице указываются некоторые наиболее распространенные применения косвенного адреса:

Чтобы:	Используйте тег в индексе и:
выбрать рецепт из массива рецептов	введите в тег номер соответствующего рецепта
загрузить конкретную настройку станка из массива возможных настроек	введите в тег нужную вам настройку
поочередно, элемент за элементом, загрузить параметры или состояния из массива	A. выполните необходимое действие над первым элементом B. используйте инструкцию ADD для увеличения значения тега и перехода к следующему элементу массива
занести в журнал регистрации коды ошибок	
выполнить несколько действий над элементом массива, а затем перейти к следующему элементу	

В следующем примере осуществляется загрузка ряда предварительно заданных значений в таймер, одно значение (элемент массива) за другим.

ПРИМЕР Обработка массива

В массиве timer_presets хранится ряд предварительно заданных значений для таймера в следующей цепочке. Тег north_tank.step указывает, какой элемент массива следует использовать. Например, когда north_tank.step равен 0, инструкция загружает в таймер timer_presets[0] (60000 мс).

```
                                                  MOV
                                   Move
                                   Source timer_presets[north_tank.step]
                                                                 60000
                                   Dest         north_tank.step_time.PRE
                                                                 60000

north_tank.step_time.DN                           TON
]/[                                Timer On Delay                  (EN)
                                   Timer   north_tank.step_time    (DN)
                                   Preset                  60000
                                   Accum                       0
```

После выполнения north_tank.step_time цепочка увеличивает значение north_tank.step до следующего номера, и в таймер загружается соответствующий элемент массива timer_presets.

```
north_tank.step_time.DN                                   ADD
] [                                                Add
                                                   Source A                1

                                                   Source B north_tank.step
                                                                           0
                                                   Dest     north_tank.step
                                                                           0
```

Когда значение north_tank.step становится больше размера массива, цепочка возвращает данный тег к первому элементу массива. (Массив включает элементы с 0-го по 3-ий).

```
          EQU                                                  MOV
Equal                                                  Move
Source A north_tank.step                               Source              0
                       0
Source B               4                               Dest north_tank.step
                                                                           0
```

42358

Выражения

Вы также можете использовать выражения для задания индекса массива.

- В выражении для вычисления значения используются операторы, такие как + или -.
- Контроллер вычисляет результат выражения и использует его в качестве индекса массива.

Для задания индекса массива вы можете использовать следующие операторы:

Оператор	Описание
+	сложение
-	вычитание/отрицание
*	умножение
/	деление
ABS	абсолютная величина
AND	и
FRD	перевод чисел в двоично-десятичном формате в целые числа
MOD	остаток от деления

Оператор	Описание
NOT	дополнение
OR	или
SQR	квадратный корень
TOD	преобразование целого числа в число в двоично-десятичном формате
TRN	усечение
XOR	исключающее или

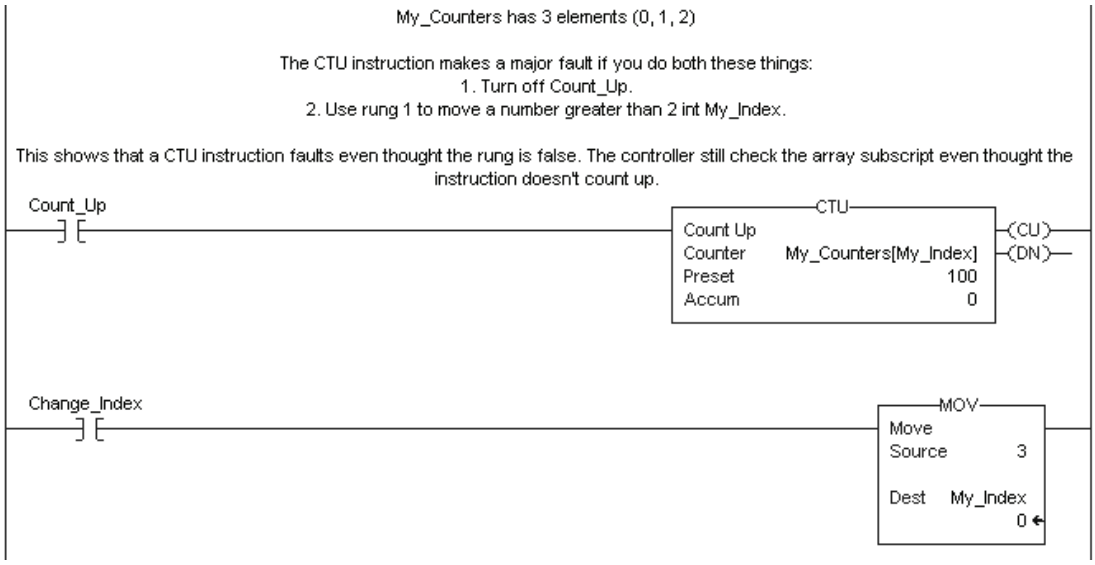
Ваше выражение должно иметь следующий формат:

Если для оператора требуется:	Используйте этот формат:	Примеры:
одно значение (тег или выражение)	*operator(value)*	ABS(tag_a)
два значения (теги, константы или выражения)	*value_a operator value_b*	· tag_b + 5 · tag_c AND tag_d · (tag_e ** 2) MOD (tag_f / tag_g)

Индекс массива за границей диапазона

В случае, если индекс массива выходит за границу диапазона, любая инструкция генерирует основную ошибку. Инструкции перехода также генерируют основную ошибку, даже в случае если цепочка принимает значение "ложь". Контроллер проверяет индекс массива даже при цепочке со значением "ложь" в следующих инструкциях.

ПРИМЕР



Более подробно обработка основных ошибок описывается в руководстве по программированию "Основные и неосновные ошибки в контроллерах Logix5000", публикация 1756-PM014.

Информация по тегам

Поскольку вы можете создавать четыре различных типа тега, описания, которые вы вводите для каждого тега, также будут различаться. Вы можете создавать теги следующих типов:

- Base (Базовый)
- Alias (Псевдоним)
- Produced (Производимый)
- Consumed (Потребляемый)

ВАЖНО

Программное обеспечении RSLogix5000 автоматически присваивает, так называемые, сквозные описания для всех тегов, которые вы создаете. Вы можете использовать или не использовать эти описания.

Информация о базовых тегах

Если вы создаете тег, не указывая его тип, RSLogix 5000 автоматически присвоит этому тегу тип по умолчанию – базовый (Base).

Поскольку при использовании базовых тегов вы можете создавать собственное внутреннее хранилище данных, в описание тега вы можете добавить характер хранимых данных.

Информация о тегах-псевдонимах

При создании тега-псевдонима у вас появляется возможность задания пользовательского имени существующему тегу, члену структуры тега или биту. В описание вашего тега-псевдонима вы можете добавить информацию о теге, к которому относится этот тег-псевдоним.

Информация о производимых тегах

Производимый тег относится к тегу, который потребляется другим контроллером. В описании производимого тега может содержаться информация об удаленных контроллерах, для которых должен быть доступен этот тег посредством обмена данными между контроллерами.

Информация о потребляемых тегах

Потребляемый тег относится к тегу, который производится другим контроллером, и данные которого вы хотите использовать в вашем контроллере. В описании производимого тега может содержаться информация о том, каким образом будут использоваться данные произведенного тега или контроллер, производящий данные.

Переключение языков

В 17 версии программного обеспечения RSLogix 5000 у вас появилась возможность просмотра документации по проекту, такой как описания тегов и комментарии к цепочкам, на любом из поддерживаемых языков перевода. Вы можете хранить документацию по проекту на нескольких языках в одном файле проекта, а не в отдельных файлах для конкретных языков. Вы определяете, какие языки будут поддерживаться в проекте и устанавливаете текущий язык перевода, язык по умолчанию и, при необходимости, пользовательский язык. В программном обеспечении язык по умолчанию используется в том случае, если отсутствует текст для определенного элемента проекта на текущем языке. В тоже время, вы можете использовать пользовательский язык для того, чтобы адаптировать информацию для отдельных пользователей файла проекта.

Вводите в ваш проект RSLogix 5000 перевод описания либо при программировании на данном языке, либо с помощью утилиты импорта/экспорта для автономного перевода документации с ее последующим импортом обратно в проект. Как только вы активируете опцию переключения языков в программном обеспечении RSLogix 5000, вы сможете динамически переключать языки в процессе работы с программным обеспечением.

Документация по проекту, для которой поддерживаются несколько переводов в рамках одного проекта, включает:

- Описания компонентов тегов, процедур, программ, пользовательских типов данных и инструкций по добавлению.
- Фазы оборудования.
- Тренды.
- Контроллеры.
- Сообщения о тревогах (в конфигурациях ALARM _ANALOG и ALARM_DIGITAL).
- Задачи.
- Описания свойств для модулей в органайзере контроллера.
- Комментарии к цепочкам, текстовые окна ПФС и текстовые окна ФБС.

За более подробной информацией по активации поддержки нескольких языков для документации по проекту обращайтесь к интерактивной справочной системе.

Для заметок:

Форсировка входов/выходов

Введение

Используйте форсировку для замены данных, которые используются или производятся вашим алгоритмом. Используйте форсировку, например, в следующих ситуациях:

- при проверке и отладке вашего алгоритма;
- при проверке связи с выходным устройством;
- для временного сохранения функционирования процесса, когда входное устройство уже дало сбой.

Используйте форсировку только как временную меру. Форсировка не предназначена для постоянной работы в составе вашего приложения.

Меры предосторожности

Принимайте следующие меры предосторожности:

ВНИМАНИЕ

Форсировка может привести к неожиданному движению механизмов, что может повлечь за собой травмы персонала. Перед использованием форсировки определите, как она повлияет на ваш механизм или процесс, и удалите людей из зоны действия механизмов.

- Разрешение форсировки ввода/вывода вызывает изменение входных, выходных, производимых или потребляемых значений.
- Разрешение форсировки ПФС вызывает переход механизма или процесса в другое состояние или фазу.
- Даже при удалении форсировок состояние разрешения форсировок может сохраниться.
- Если форсировки разрешены, и вы устанавливаете форсировку, она действует немедленно.

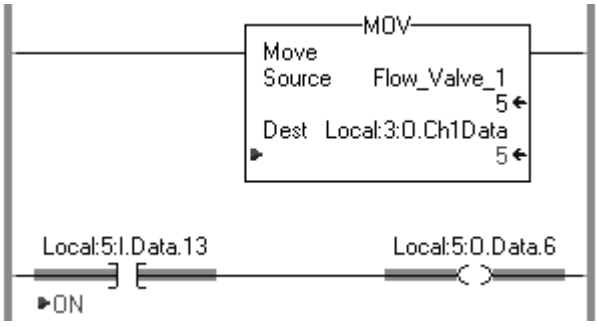
Разрешение форсировок

Для того чтобы форсировка заработала, она должна быть разрешена. Вы можете разрешить или запретить форсировку только на уровне контроллера.

- Вы можете разрешить форсировки ввода/вывода и ПФС по отдельности или одновременно.
- Вы не можете разрешать или запрещать форсировки для отдельного модуля, набора тегов или элемента тега.

ВАЖНО Если вы загружаете проект, в котором есть разрешенные форсировки, программное обеспечение предложит вам разрешить или запретить режим форсировки после окончания загрузки.

Когда форсировка работает (разрешена), значок ► появляется рядом с форсированным элементом.



состояние, до которого элемент форсирован

Запрещение или удаление форсировки

Для выключения форсировки и выполнения проекта в запрограммированном режиме, запретите или удалите форсировку.

- Вы можете запретить или удалить форсировки ввода/вывода и ПФС по отдельности или одновременно.
- Удаление форсировки для тега псевдонима удаляет форсировку для базового тега..

ВНИМАНИЕ

Изменение форсировок может привести к неожиданному движению механизмов, что может повлечь за собой травмы персонала. Перед использованием форсировки определите, как изменение повлияет на ваш механизм или процесс, и удалите людей из зоны действия механизмов.

Проверка состояния форсировок

Перед использованием форсировки проверьте состояние форсировок контроллера. Вы можете проверить состояние форсировок следующими способами:

Для определения состояния:	Используйте:
форсировки ввода/вывода	· панель инструментов Online · светодиод FORCE · инструкцию GSV
форсировки ПФС	панель инструментов Online

Панель инструментов Online демонстрирует состояние форсировок. Статус форсировок ввода/вывода и ПФС отображаются отдельно.

Закладка Forces (Форсировки)

Это:	Означает:
Enabled (Разрешены)	· Если проект содержит форсировки этого типа, они подменяют алгоритм. · Если вы добавляете форсировку этого типа, новая форсировка начинает действовать немедленно.
Disabled (Запрещены)	Форсировки не активны. Если проект содержит форсировки этого типа, *они не* подменяют алгоритм.
Installed (Установлены)	По крайней мере, одна форсировка данного типа содержится в данном проекте.
None Installed (Не установлены)	В проекте нет форсировок данного типа.

Светодиод FORCE

Если ваш контроллер оборудован светодиодом FORCE , используйте его для определения наличия любых форсировок ввода/вывода.

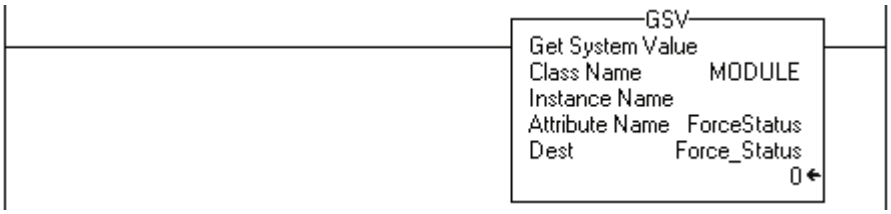
ВАЖНО Светодиод FORCE демонстрирует только наличие форсировок ввода/вывода. Наличие форсировок ПФС не отображается.

Если светодиод FORCE:	То:
не горит	· теги не содержат значения форсировок · форсировки ввода/вывода не активны (запрещены)
мигает	· по крайней мере, один тег содержит значение форсировки · форсировки ввода/вывода не активны (запрещены)
горит не мигая	· форсировки ввода/вывода активны (разрешены) · значения форсировок могут существовать, а могут и не существовать

Инструкция GSV

ВАЖНО Атрибут ForceStatus показывает только наличие форсировок ввода/вывода. Наличие форсировок ПФС не отображается.

Следующий пример показывает, как использовать инструкцию GSV для получения состояния форсировок.



где:

Force_Status – это тег типа DINT.

Для того, чтобы определить, что:	Проверьте этот бит:	На наличие значения:
форсировки установлены (installed)	0	1
остановленные форсировки отсутствуют	0	0
форсировки разрешены	1	1
форсировки запрещены	1	0

Когда использовать форсировку ввода/вывода

Используйте фосировку ввода/вывода для следующих целей:

- Подмена входного значения от другого контроллера (т.е. потребляемого тега).
- Подмена входного значения от устройства ввода.
- Подмена логики и задания выходного значения, предназначенного для другого контроллера (т.е. производимого тега).
- Подмена вашей логики и задания состояния выходного устройства.

ВАЖНО Форсировка увеличивает время выполнения логики. Чем большее количество значений используют форсировку, тем дольше выполняется алгоритм.

ВАЖНО Форсировки хранятся в контроллере, а не в рабочей станции, с которой ведется программирование. Форсировка сохраняется, даже если рабочая станция будет отключена.

Когда используется форсировка ввода/вывода:

- Вы можете вводить форсировку для любых данных ввода/вывода, за исключением данных конфигурации.
- Если тег является массивом или конструкцией, такой как тег ввода/вывода, форсируйте элементы или члены типа BOOL, SINT, INT, DINT или REAL.
- Если данные имеют тип SINT, INT или DINT, вы можете форсировать все значение или отдельные биты в пределах значения. Отдельные биты могут иметь следующие состояния форсировки:
  - no force (нет форсировки);
  - force on (форсировка включена);
  - force off (форсировка отключена).
- Вы можете вводить форсировку для псевдонима элемента структуры ввода/вывода, производимого тега или потребляемого тега.
  - Тег-псевдоним использует значения данных совместно с базовым тегом, поэтому форсировка тега-псевдонима вызывает форсировку связанного с ним базового тега.
  - Снятие форсировки с тега-псевдонима снимает форсировку с соответствующего базового тега.

Форсировка входного значения

Форсировка входного или потребляемого тега:

- Приводит к замене значения независимо от физического устройства или производимого тега.
- Не влияет на значение, получаемое другими контроллерами, отслеживающими этот вход или производимый тег.

Форсировка выходного значения

Форсировка выхода или производимого тега заменяет логику для физического устройства или другого контроллера (контроллеров). Другие контроллеры, отслеживающие этот модуль выхода в режиме слежения, так же увидят это форсированное значение.

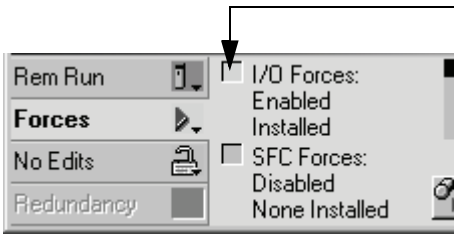
Добавление форсировки ввода/вывода

Используйте форсировку для замены входного значения, выходного значения, производимого тега и потребляемого тега.

ВНИМАНИЕ

Форсировка может привести к неожиданному движению механизмов, что может повлечь за собой травмы персонала. Перед использованием форсировки определите, как она повлияет на ваш механизм или процесс, и удалите людей из зоны действия механизмов.

- Разрешение форсировки ввода/вывода вызывает изменение входных, выходных, производимых и потребляемых значений.
- Если форсировка разрешена, она начинает действовать немедленно.



1. Каково состояние индикатора форсировки ввода вывода?

Если:	То:
не горит	В данный момент не существует форсировок ввода/вывода.
мигает	Форсировки ввода/вывода не активны. Но, по крайней мере, одна форсировка существует в вашем проекте. Когда вы разрешите форсировки ввода/вывода, начнут действовать все форсировки ввода/вывода.
горит не мигая	Форсировки ввода/вывода разрешены (активны). Когда вы устанавливаете (добавляете) форсировку, она начинает действовать немедленно.

2. Откройте процедуру, содержащую тег, который вы хотите форсировать.
3. Щелкните правой клавишей мыши на этом теге и выберите Monitor… Если необходимо, раскройте этот тег, чтобы увидеть значение, которое вы хотите форсировать (напр. Значение BOOL тега типа DINT).
4. Установите значение форсировки:

Для форсировки:	Сделайте следующее:
значение типа BOOL	Щелкните правой клавишей мыши на теге и выберите Force ON или Force OFF
значение не типа BOOL	В колонке Force Mask для данного тега введите с клавиатуры значение, на которое вы хотите форсировать данный тег. Затем нажмите клавишу Enter.

5. Разрешены ли форсировки ввода/вывода? (См. Шаг 1.).

Если:	То:
Нет	В меню Logic(Логика) выберите I/O Forcing >Enable AI I/O Forces. Затем выберите Yes для подтверждения.
Да	Действия не требуются.

Удаление или запрещение форсировок

ВНИМАНИЕ

Изменение форсировок может привести к неожиданному движению механизмов, что может повлечь за собой травмы персонала. Перед использованием форсировки определите, как изменение повлияет на ваш механизм или процесс, и удалите людей из зоны действия механизмов.

Если вы хотите:	И:	То:
остановить отдельную форсировку	сохранить другие форсировки разрешенными и действующими	удалите отдельную форсировку
остановить все форсировки ввода/вывода, но оставить все форсировки ПФС активными	сохранить все форсировки ввода/вывода в вашем проекте	запретите все форсировки ввода/вывода
	удалить все форсировки ввода/вывода в вашем проекте	удалите все форсировки ввода/вывода

Удаление отдельной форсировки

ВНИМАНИЕ

Если вы удаляете отдельную форсировку, другие форсировки остаются в состоянии «разрешены» и новая форсировка начинает действовать немедленно.

Перед удалением форсировки определите, как это изменение повлияет на ваш механизм или процесс, и удалите людей из зоны действия механизмов.

1. Откройте процедуру, которая содержит форсировку, которую вы собираетесь удалить.
2. На каком языке написана данная процедура?

Если на языке:	То:
ПФС	переходите к шагу 4
релейной логики	переходите к шагу 4
функциональных блоков	переходите к шагу 3
структурированного текста	переходите к шагу 3

3. Щелкните правой клавишей мыши на теге, который имеет форсировку и выберите Monitor… Если необходимо, то раскройте этот тег, чтобы видеть значение, которое было форсировано (например, значение BOOL тега DINT).
4. Щелкните правой клавишей мыши на теге или элементе, который имеет форсировку и выберите Remove Force (Удалить форсировку).

Запрещение всех форсировок ввода/вывода

В меню Logic выберите I/O Forcing >Disable All I/O Forces. Затем выберите Yes для подтверждения.

Удаление всех форсировок ввода/вывода

В меню Logic выберите I/O Forcing > Remove All I/O Forces. Затем выберите Yes для подтверждения.

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future.
Thank you for taking the time to provide us feedback.

You can complete this form and mail (or fax) it back to us or email us at RADocumentComments@ra.rockwell.com.

Pub. Title/Type Logix5000 Controllers I/O and Tag Data

Cat. No.	1756 ControlLogix, 1768 CompactLogix, 1769 CompactLogix, 1789 SoftLogix, 1794 FlexLogix, PowerFlex 700S with DriveLogix	Pub. No.	1756-PM004B-EN-P	Pub. Date	July 2008	Part No.	

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness	1	2	3	How can we make this publication more useful for you?
Completeness (all necessary information is provided)	1	2	3	Can we add more information to help you? procedure/step illustration feature example guideline other explanation definition
Technical Accuracy (all provided information is correct)	1	2	3	Can we be more accurate? text illustration
Clarity (all provided information is easy to understand)	1	2	3	How can we make things clearer?
Other Comments				You can add additional comments on the back of this form.

Your Name

Your Title/Function

Location/Phone

Would you like us to contact you regarding your comments?

____No, there is no need to contact me

____Yes, please call me

____Yes, please email me at

____Yes, please contact me via

Return this form to:Rockwell Automation Technical Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Fax: 440-646-3525Email: RADocumentComments@ra.rockwell.com

Publication CIG-CO521D-EN-P- July 2007

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

1 ALLEN-BRADLEY DR
MAYFIELD HEIGHTS OH 44124-9705

Служба поддержки Rockwell Automation

Компания Rockwell Automation предоставляет техническую информацию в интернете с целью поддержки своих клиентов. По адресу http://support.rockwellautomation.com вы найдете технические руководства, ответы на часто задаваемые вопросы, заметки по техническим характеристикам и эксплуатации продукции, коды пользования пакетами ПО для ознакомления и загрузки, а также службу поддержки MySupport, которую Вы можете настроить по своему желанию с целью оптимального использования перечисленных средств клиентской поддержки.

Наша компания также предлагает своим клиентам программы поддержки пользователей по телефону TechConnect по вопросам установки, настройки и разрешения проблем. Более подробные сведения вы можете получить у дистрибьюторов и представителей компании Rockwell Automation в вашем регионе или на сайте http://support.rockwellautomation.com

Содействие при установке

Если у вас возникли проблемы в течении первых 24 часов процесса установки, пожалуйста, обратитесь к информации, содержащейся в настоящем руководстве. Вы также можете позвонить по специальному телефону Службы поддержки клиентов и проконсультироваться по вопросам приобретения и эксплуатации продукции нашей компании.

Соединенные Штаты	1.440.646.3434 Понедельник – Пятница, 8.00 – 17.00 (Восточное стандартное время США)
За пределами США	Пожалуйста, обратитесь к представителю компании Rockwell Automation в вашем регионе по любым интересующим вас вопросам.

Возврат продукции

Компания Rockwell Automation проводит испытания всей своей продукции в целях обеспечения контроля качества и пригодности изделий к использованию при отгрузке продукции с производства. Тем не менее, в случае если изделие не функционирует и подлежит возврату, выполните следующие шаги.

Соединенные Штаты	Для осуществления процесса возврата вы должны предоставить ваш личный идентификационный номер в службе поддержки клиентов (узнать его можно, позвонив по вышеуказанному телефону) дистрибьютору компании в вашем регионе.
За пределами США	Пожалуйста, обратитесь к представителю компании Rockwell Automation в вашем регионе по вопросу возврата изделия.

www.rockwellautomation.com

Штаб-квартира по решениям в энергетике, управлении и информации

Россия и СНГ: Rockwell Automation BV, 115054, Москва, Большой Строченовский пер., 22/25, офис 402, Тел. +7(495)956-0464, факс +7(495)956-0469

Америка: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204 USA, тел.: (1) 414 382-2000, факс: (1) 414 382-4444

Европа/Ближний Восток/Африка: Rockwell Automation, Vorstlaan/Boulevard de Souverain 36, 1170 Brussels, Belgium, тел.: (32) 2 663 0600, факс: (32) 2 663 0640

Тихоокеанский регион: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, тел.: (852) 2887 4788, факс (852) 2508 1846

Публикация 1756-PM004B-EN-P - Июль 2008

Замещает публикацию 1756-PM-004A-EN-P - Июль 2007

Авторское право © 2009 Rockwell Automation, Inc. Все права защищены.