Задачи, программы и процедуры в контроллерах Logix5000

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 и TechConnect 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

Управление задачами

Введение 7
Выбор задач контроллера 7
Осторожность при выборе количества используемых задач 10
Назначение приоритетов периодических задач и событийных задач 10
Дополнительные рекомендации 11
Выделение достаточного времени для незапланированного обмена данными 13
Избежание наложений 14
Проверка наличия наложений в ручном режиме 15
Проверка наличия наложений программным путем 16
Конфигурирование обработки выходных данных для задачи . . . 18
Конфигурирование обработки выходных данных в ручном режиме 20
Конфигурирование обработки выходных данных программным путем 21
Запрет выполнения задачи 22
Запрет и разрешение выполнения задачи в ручном режиме . 22
Запрет и разрешение выполнения задачи программным путем 23
Выбор триггера для событийной задачи 24
Использование триггера изменения состояния входных данных модуля 27
Как модуль ввода/вывода запускает событийную задачу 27
Убедитесь, что модуль может запускать событийную задачу . 30
Контрольный перечень для задачи, запускаемой событием ввода 31
Оценка времени цикла 33
Оценка времени цикла 35
Дополнительные рекомендации 36
Использование триггера группы перемещения. 36
Контрольный перечень для задачи группы перемещения . . . 37
Использование триггера регистрации оси. 38
Контрольный перечень для задачи регистрации оси 39
Использование триггера отслеживания работы оси. 42
Контрольный перечень для задачи отслеживания работы оси 43
Использование триггера потребляемого тега 46
Обеспечение целостности данных 48
Синхронизация нескольких контроллеров 49
Контрольный перечень для производящего контроллера . . . 50
Контрольный перечень для потребляющего контроллера . . 51
Производящий контроллер 52

Свойства производимого тега 52
Релейная логика 52
Потребляющий контроллер 53
Свойства событийной задачи 53
Релейная блок-схема в событийной задаче 53
Использование триггера инструкции по событию. 54
Программное определение того, запустила ли инструкция по событию задачу. 55
Контрольный перечень для задачи, запускаемой инструкцией по событию. 55
Свойства событийной задачи 56
Релейная блок-схема в программе A (Program_A) 56
Релейная блок-схема в программе B 56
Создание задачи 57
Создание событийной задачи 57
Создание периодической задачи. 58
Переключение языков. 59
Определение времени ожидания для событийной задачи 60
Определение значения времени ожидания для событийной задачи 60
Конфигурирование времени ожидания программным путем 61
Программное определение превышения времени ожидания. 62
Настройка кванта времени на системные операции. 64
Настройка кванта времени на системные операции 66
Настройка времени сторожевого таймера 67
Настройка времени сторожевого таймера для задачи 67

Назначение данного руководства

Данное руководство описывает, как конфигурировать и использовать производимые и потребляемые теги. Это руководство входит в комплект руководств, описывающих методику программирования и работу с контроллерами Logix5000. Полный перечень руководств по общей методике находится 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:

- контроллеры CompactLogix;
- контроллеры ControlLogix;
- контроллеры DriveLogix;
- контроллеры FlexLogix;
- контроллеры Logix5000.

Как использовать данное руководство

В руководстве встречается текст с различным форматированием.

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:

Управление задачами

Введение

Проект RSLogix 5000 по умолчанию предлагает одну задачу для всей логики. Несмотря на то, что этого бывает достаточно для большинства приложений, возникают ситуации, когда может потребоваться более одной задачи.

Выбор задач контроллера

Контроллер Logix5000 позволяет использовать многозадачный режим для планирования выполнения программ и назначения им приоритетов на основе определенных критериев. Это позволяет сбалансировано распределять время выполняемой контроллером обработки данных контроллера.

- В каждый момент времени контроллер выполняет только одну задачу.
- Другая задача может прервать выполнение текущей задачи и взять управление на себя.
- В каждый момент времени в каждой задаче выполняется только одна программа.

Контроллер Logix5000 поддерживает три типа задач.

Если вы хотите, чтобы часть вашей логики выполнялась:	То используйте этот тип задачи:	Описание:
Все время	Непрерывная задача (Continuous Task)	Непрерывная задача выполняется в фоновом режиме. Все время центрального процессора, не занятое другими операциями (например, перемещение, обмен данными и периодические задачи или событийные задачи), используется для выполнения программ непрерывной задачи. • Непрерывная задача выполняется непрерывно. Когда непрерывная задача завершает полный цикл сканирования, она немедленно перезапускается. • Для проекта не требуется непрерывная задача. Если такая задача используется, то она может быть только одна.
• С постоянной периодичностью (например, каждые 100 мс) • Несколько раз в процессе сканирования другой логики	Периодическая задача (Periodic Task)	Периодическая задача выполняет определенную функцию с заданной периодичностью. По истечении заданного для периодической задачи времени эта задача: • прерывает все задачи с более низким приоритетом; • однократно выполняется; • возвращает управление в место, где было прервано выполнение предшествующей задачи. Можно задать период времени от 0,1 мс до 2000 с. • Значение по умолчанию – 10 мс. • Выполнение периодической задачи зависит от типа контроллера Logix5000 и от используемой в задаче логики.
Мгновенно после появления события	Событийная задача (Event Task)	Событийная задача выполняет определенную функцию только при появлении события (триггера). При появлении триггера событийной задачи эта задача: • прерывает все задачи с более низким приоритетом; • однократно выполняется; • возвращает управление в место, где было прервано выполнение предшествующей задачи. В качестве триггера может служить: • изменение дискретного входа; • новая выборка аналоговых данных; • определенные операции перемещения; • потребляемый тег; • инструкция по событию (EVENT). **Важно:** Некоторые контроллеры Logix5000 поддерживают не все триггеры.

Ниже приведены примеры возможных ситуаций для задач:

Для данной ситуации:	**Используйте этот тип задачи:**
Заполнение резервуара до максимального уровня с последующим открытием спускного клапана.	Непрерывная задача
Сбор и обработка параметров системы и их вывод на экран.	Непрерывная задача
Выполнение шага 3 в управляющей последовательности – перестановка отводящей перегородки бункера.	Непрерывная задача
Ваша система должна проверять положение механического рычага каждые 0,1 с и вычислять среднюю скорость изменения его положения. Это используется для определения давления торможения.	Периодическая задача
Считывание толщины рулона бумаги каждые 20 мс.	Периодическая задача
На упаковочной линии производится склеивание закрытых коробок. При поступлении коробки на операцию склеивания контроллер должен немедленно выполнить процедуру склеивания.	Событийная задача
При выполнении высокоскоростной сборки оптический детектор обнаруживает брак определенного типа. При обнаружении детектором брака он должен быть немедленно отброшен.	Событийная задача
На испытательном стенде двигателя нужно получить и архивировать все аналоговые данные сразу после каждой выборки данных.	Событийная задача
Загрузка данных в станцию сразу после получения новых производственных данных.	Событийная задача
На линии упаковки конфет необходимо обеспечить перфорацию каждой конфеты в заданном месте. При каждом обнаружении датчиком регистрации соответствующей отметки следует проверить точность оси и при необходимости выполнить ее корректировку.	Событийная задача
Станция склеивания должна корректировать количество наносимого клея для компенсации изменений скорости перемещения оси. После выполнения задачи планировщика перемещения необходимо проверить заданную командой скорость перемещения оси и при необходимости изменить количество клея.	Событийная задача
Если какая-либо программа обнаружит опасное состояние, вся производственная линия должна быть остановлена. Процедура останова не зависит от типа опасного состояния.	Событийная задача

Количество поддерживаемых задач зависит от контроллера:

Контроллер:	**Поддерживает следующее число задач:**	**Примечания:**
ControlLogix SoftLogix5800	32	Только одна задача может быть непрерывной.
CompactLogix DriveLogix FlexLogix	8	

Осторожность при выборе количества используемых задач

Как правило, каждая задача забирает время контроллера, выделяемое на другие задачи. Если у вас слишком много задач, то:

- Непрерывная задача может выполняться слишком долго.
- Может произойти наложение других задач. Если задача прерывается слишком часто или на слишком длительное время, она может не завершиться до ее очередного запуска.

Назначение приоритетов периодических задач и событийных задач

Несмотря на то, что проект может включать несколько задач, в каждый момент времени контроллер выполняет лишь одну задачу. Если во время выполнения какой-либо задачи запускается периодическая задача или событийная задача, приоритет каждой из задач определяет действия котроллера.

Количество уровней приоритета зависит от контроллера:

Контроллер Logix5000	Имеет столько уровней приоритета:
CompactLogix	15
ControlLogix	15
DriveLogix	15
FlexLogix	15
SoftLogix5800	3

Для определения приоритета задачи воспользуйтесь следующими указаниями.

Если нужно, чтобы:	Примите следующие меры:	Примечания:
Данная задача прерывала другую задачу.	Установите номер приоритета, меньший (более высокий приоритет), чем номер приоритета другой задачи.	• Задача с более высоким приоритетом прерывает все задачи, приоритет которых ниже. • Задача с более высоким приоритетом может многократно прерывать задачу, приоритет которой ниже.
Другая задача прерывала данную задачу.	Установите номер приоритета, больший (более низкий приоритет), чем номер приоритета другой задачи.	
Данная задача совместно использовала время контроллера с другой задачей.	Установите одинаковый номер приоритета для обеих задач.	Контроллер переключается между задачами и выполняет каждую из них в течение 1 мс.

Дополнительные рекомендации

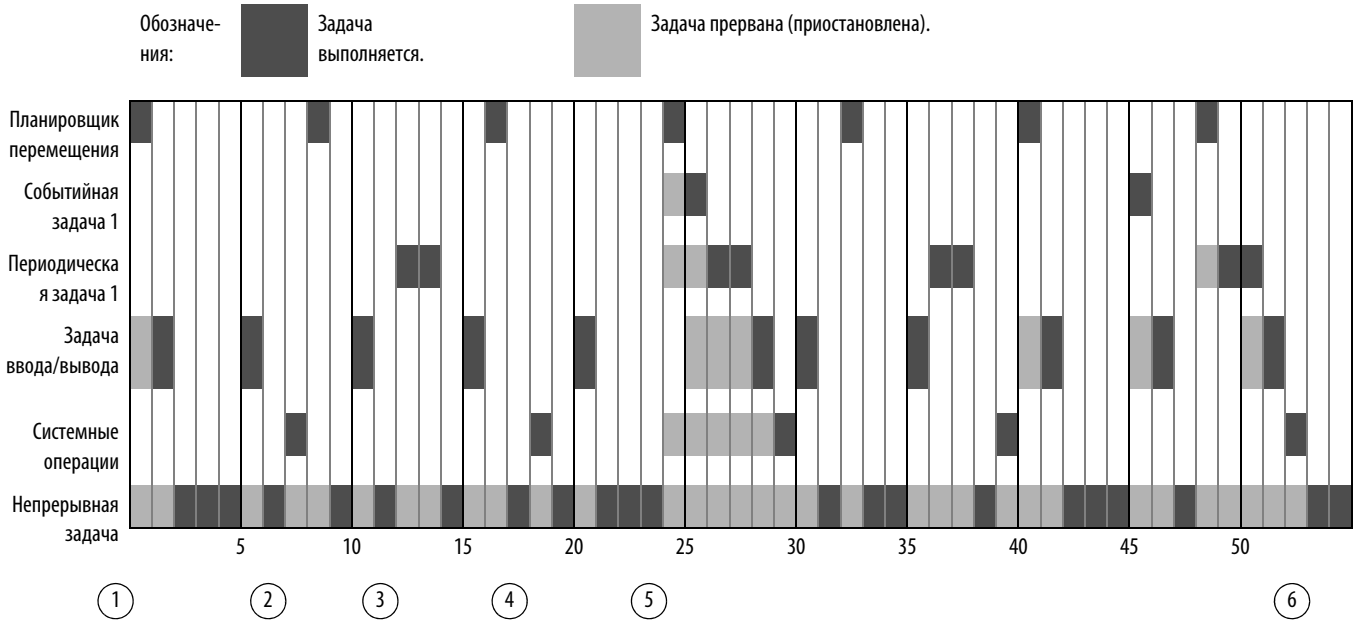
При оценке количества прерываний выполнения какой-либо задачи следуйте данным рекомендациям:

Фактор	Описание
Планировщик задачи управления перемещением	Планировщик перемещения прерывает все другие задачи независимо от их приоритета. • Количество осей и время обновления предварительной обработки группы перемещения влияют на продолжительность и частоту выполнения планировщика перемещения. • Если планировщик перемещения выполнятся во время начала задачи, то задача ожидает завершения работы планировщика перемещения. • Если обновление предварительной обработки выполняется во время выполнения задачи, задача останавливается, чтобы выполнить действия планировщика перемещения.
Задача ввода/вывода	Контроллеры CompactLogix, FlexLogix и DriveLogix используют специализированную периодическую задачу для обработки данных ввода/вывода. Такая задача ввода/вывода: • Не указывается в папке Tasks (Задачи) контроллера. • Не учитывается при подсчете допустимого количества задач для данного контроллера. • Работает с приоритетом 7. • Выполняется с максимальным интервалом передачи пакетов (RPI), запланированным для данной системы. • Выполняется столько времени, сколько требуется для сканирования. При присвоении приоритетов для ваших задач необходимо учитывать задачу ввода/вывода:
	Если нужно, чтобы задача: — **Тогда следует назначить один из следующих приоритетов:**
	прерывала или задерживала обработку данных ввода/вывода — от 1 до 6
	разделяла время контроллера с обработкой данных ввода/вывода — 7
	позволяла обработке ввода/вывода прерывать или задерживать задачу — от 8 до 15
Системные операции	Временные издержки за счет системных операций – это время, затрачиваемое контроллером на незапланированный обмен данными. • Незапланированный обмен данными – это всякий обмен данными, который не конфигурируется посредством папки конфигурации ввода/вывода (I/O configuration) проекта, как например, инструкции Message (MSG) и обмен данными с HMI (человеко-машинным интерфейсом) или рабочими станциями. • Системные операции прерывают только непрерывную задачу. • Значение кванта времени контроллера на системные операции определяет процент времени контроллера (за исключением времени на выполнение периодических задач или событийных задач), которое выделяется контроллером для незапланированного обмена данными. • Контроллер выполняет незапланированный обмен данными в течение максимум 1 мс, после чего возобновляет выполнение непрерывной задачи.
Непрерывная задача	Непрерывная задача не требует назначения приоритета. Она всегда работает с наименьшим приоритетом. Все другие задачи прерывают непрерывную задачу.

ПРИМЕР

В следующем примере показано выполнение проекта с тремя пользовательскими задачами.

Задача	Приоритет	Период	Время выполнения	Продолжи-тельность
Планировщик задачи управления перемещением	Не применяется	8 мс (период обновления данных)	1 мс	1 мс
Событийная задача 1	1	Не применяется	1 мс	от 1 до 2 мс
Периодическая задача 1	2	12 мс	2 мс	от 2 до 4 мс
Задача ввода/вывода не применяется для контроллеров ControlLogix и SoftLogix. Смотрите странице 11.	7	5 мс (самый быстрый RPI)	1 мс	от 1 до 5 мс
Системные операции	Не применяется	Квант времени = 20%	1 мс	от 1 до 6 мс
Непрерывная задача	Не применяется	Не применяется	20 мс	48 мс



Описание	
1	Сначала контроллер выполняет планировщика перемещений и задачу ввода/вывода (если она имеется).
2	После выполнения непрерывной задачи в течение 4 мс контроллер запускает системные операции.
3	Период для периодической задачи 1 (12 мс) истекает, поэтому эта задача прерывает непрерывную задачу.
4	После повторного выполнения непрерывной задачи в течение 4 мс контроллер запускает системные операции.
5	Появляется триггер событийной задачи 1. Событийная задача 1 ожидает завершения действий планировщика перемещений. Задачи с более низкими приоритетами задерживаются дольше.
6	Непрерывная задача автоматически перезапускается.

Программное обеспечение RSLogix 5000 включает в себя инструмент мониторинга нагрузки (Task Monitor Tool) на сопутствующем диске. Можно использовать этот инструмент для анализа выполнения задач.

Выделение достаточного времени для незапланированного обмена данными

Незапланированный обмен данными происходит лишь в то время, когда не работает ни периодическая задача, ни событийная задача. Если вы используете несколько задач, убедитесь в том, что их время сканирования и промежутки времени между выполнениями обеспечивают достаточное время для незапланированного обмена данными.

1. Время выполнения задачи с самым высоким приоритетом значительно меньше частоты ее обновления.
2. Общее время выполнения всех задач значительно меньше частоты обновления задач с наименьшим приоритетом.

Например, для следующей конфигурации задач:

Задача	Приоритет	Время выполнения	Частота обновления
1	выше	20 мс	80 мс
2	ниже	30 мс	100 мс
	общее время выполнения:	50 мс	

- Время выполнения задачи с наивысшим приоритетом (Задача 1) значительно меньше частоты ее обновления (20 мс меньше 80 мс).
- Общее время выполнения всех задач значительно меньше частоты обновления задачи с наименьшим приоритетом (50 мс меньше 100 мс).

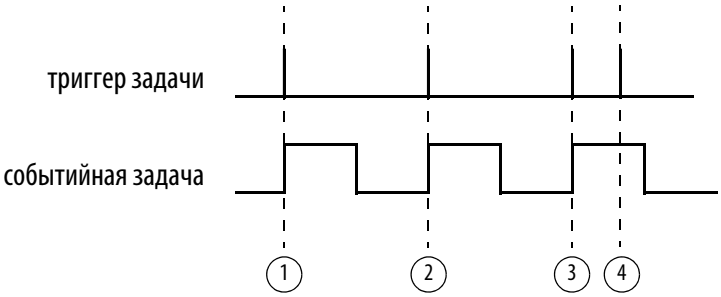
При этом, как правило, остается достаточно времени для незапланированного обмена данными.

- Установите частоты обновления задач таким образом, чтобы добиться наилучшего компромисса между выполнением логики и обслуживанием незапланированного обмена данными.
- Если в вашем проекте есть непрерывная задача, то незапланированный обмен данными происходит в течение определенного процента времени контроллера (за исключением времени выполнения периодических задач или событийных задач).

Избежание наложений

Наложение – это состояние, когда задача (периодическая или событийная) запускается во время выполнения задачи, запущенной предыдущим триггером.

ВАЖНО В случае наложения контроллер игнорирует триггер, вызвавший это наложение. Иными словами, можно пропустить важное выполнение задачи.



	Описание
1	Появляется триггер задачи. Задача выполняется.
2	Появляется триггер задачи. Задача выполняется.
3	Появляется триггер задачи. Задача выполняется.
4	Происходит наложение. Задача снова вызывается во время ее выполнения. Триггер не перезапускает задачу. Триггер игнорируется.

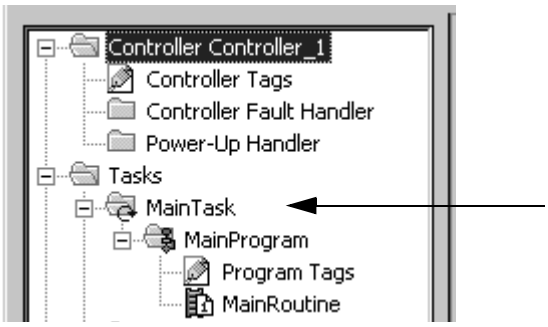
Каждой задаче требуется достаточное время для ее завершения перед очередным запуском. Убедитесь в том, что время сканирования задачи значительно меньше частоты появления триггера. При возникновении наложения уменьшите частоту запуска задачи.

Для этого типа задачи:	Примите следующие меры:
Периодический	Следует увеличить период задачи.
Событийный	Установите конфигурацию вашей системы таким образом, чтобы данная задача запускалась с меньшей частотой.

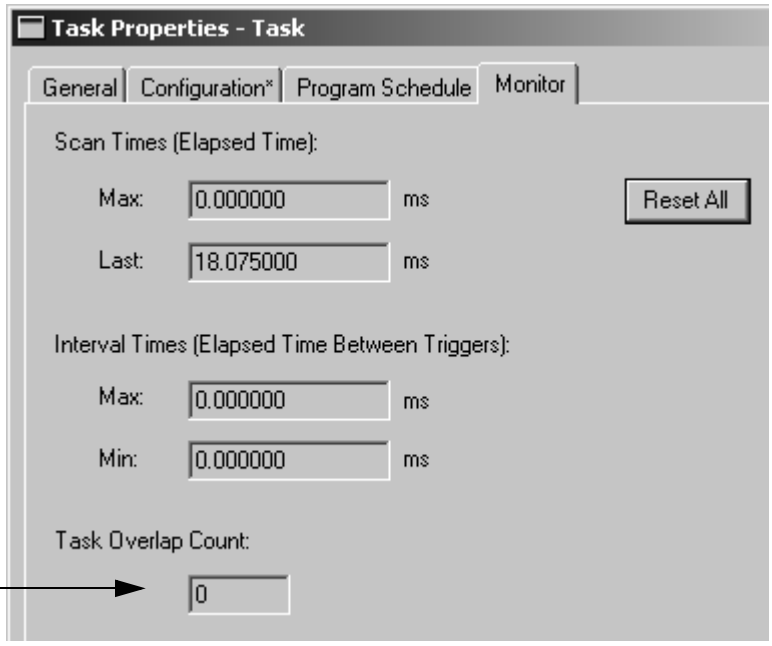
Проверка наличия наложений в ручном режиме

Чтобы вручную проверить, происходят ли наложения для какой-либо задачи:

1. В органайзере контроллера щелкните правой кнопкой мыши по соответствующей задаче и выберите Properties (Свойства).



2. Щелкните по закладке Monitor (Отслеживание).



Количество наложений поле последнего сброса счетчика

3. Нажмите OK.

Проверка наличия наложений программным путем

При появлении наложений контроллер:

- Заносит неосновную ошибку в объект FAULTLOG («Журнал ошибок»).
- Сохраняет информацию о наложении в объекте TASK («Задача») для данной задачи.

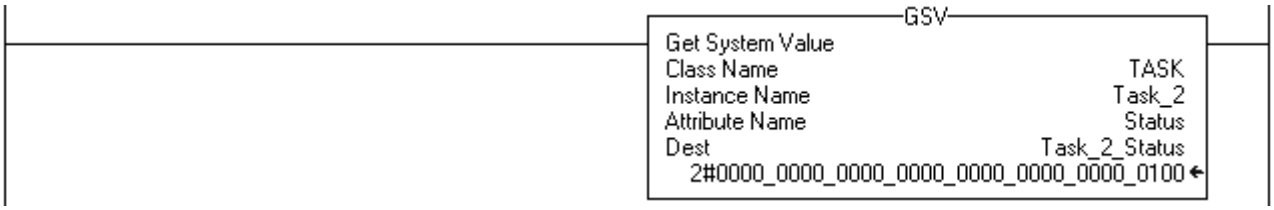
Чтобы написать алгоритм, который проверяет наличие наложений, используйте инструкцию Get System Value (GSV) (Получить значение системы) для контроля одного из следующих объектов:

Если требуется:	**То обратитесь к следующему объекту и параметру:**			
	Объект	**Параметр**	**Тип данных**	**Описание**
Определить, произошло ли наложение для какой-либо задачи.	FAULTLOG	MinorFaultBits	DINT	Отдельные биты, указывающие на неосновную ошибку.
				Чтобы определить, имеет ли место следующее: / **Проверьте этот бит:**
				Инструкция произвела неосновную ошибку. / 4
				Произошло наложение задачи. / 6
				Последовательный порт произвел неосновную ошибку. / 9
				Батарея питания отсутствует или требует замены. / 10
Определить, произошло ли наложение для определенной задачи.	TASK	Status	DINT	Параметр Status («Статус») информирует о состоянии задачи. Когда контроллер установит один из следующих битов, необходимо вручную сбросить соответствующий бит.
				Чтобы определить, имеет ли место следующее: / **Проверьте этот бит:**
				Инструкция по событию EVENT запустила задачу (только для событийных задач). / 0
				Превышение времени задержки запустило задачу (только для событийных задач). / 1
				Произошло наложение задачи. / 2
Определить, сколько раз произошло наложение.	TASK	OverlapCount	DINT	Действует для событийной или периодической задачи. Для сброса счетчика установите этот параметр в 0.

ПРИМЕР

Проверка наличия наложений программным путем.

1. Инструкция GSV устанавливает параметр Task_2_Status =Status («Состояние») для задачи Task_2 (значение DINT).



2. Если Task_2_Status.2 = 1, то произошло наложение, и следует получить значение счетчика наложений: Инструкция GSV устанавливает параметр Task_2_Overlap_Count (тег DINT) =OverlapCount («Подсчет наложений») для задачи Task_2.

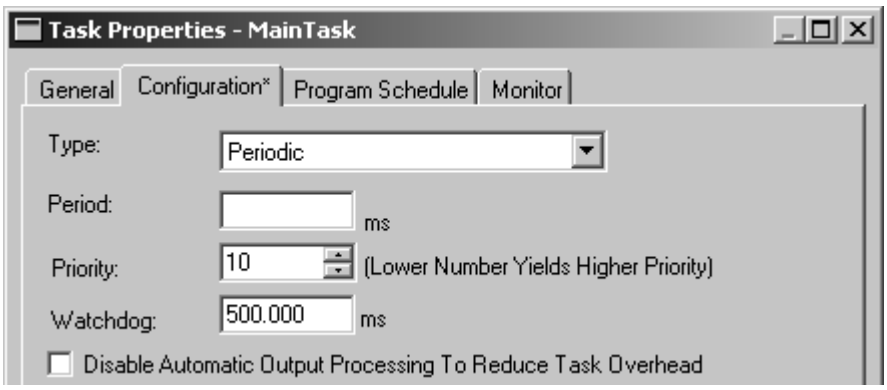
3. Если Condition_1 = 1, то необходимо сбросить биты параметра Status («Состояние») для задачи Task_2: Инструкция SSV устанавливает параметр Status задачи Task_2 = Zero («Ноль»). Zero («Ноль») – это тег DINT, имеющий значение 0.

Конфигурирование обработки выходных данных для задачи

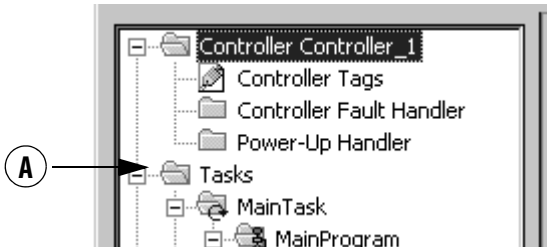
В конце выполнения задачи контроллер выполняет системные операции (обработку выходных данных) для модулей ввода/вывода вашей системы. Хотя это не то же самое, что обновление этих модулей, такая обработка выходных данных может оказывать влияние на обновление модулей ввода/вывода вашей системы.

У вас есть возможность отключить такую обработку выходных данных для определенной задачи, что уменьшает время выполнения данной задачи.

Разрешите или запретите обработку выходных данных в конце выполнения задачи

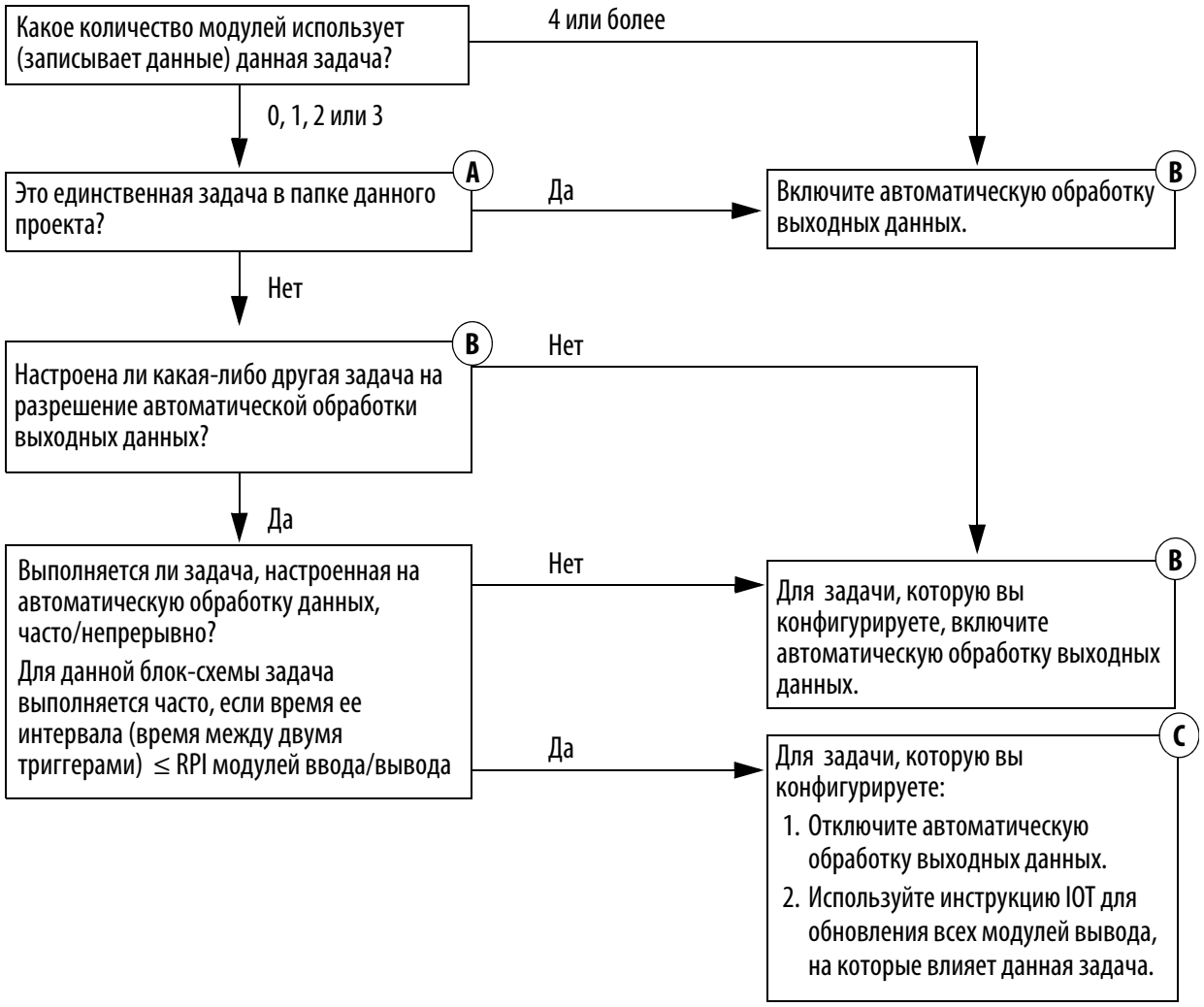


Выберите настройку обработки выходных данных задачи.



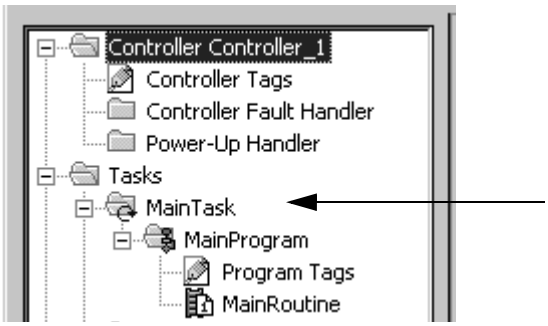
B ☐ Отключите автоматическую обработку выходных данных (Automatic Output Processing), Чтобы уменьшить издержки задачи.

C ☑ Отключите автоматическую обработку выходных данных (Automatic Output Processing), Чтобы уменьшить издержки задачи

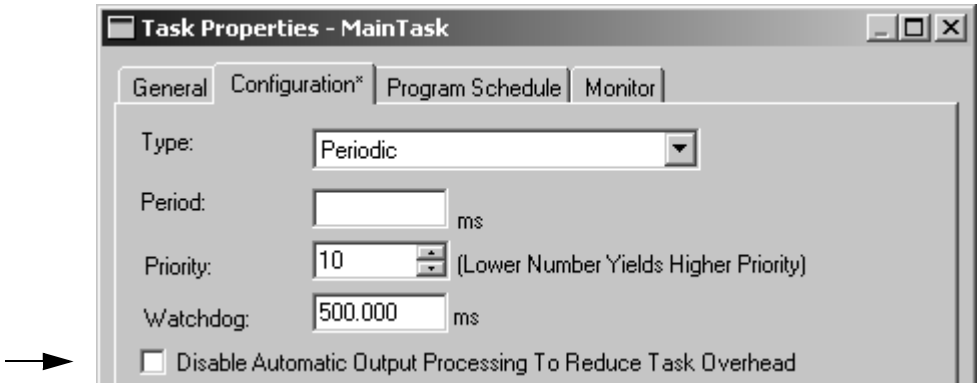


Конфигурирование обработки выходных данных в ручном режиме

1. В органайзере контроллера щелкните правой кнопкой мыши по соответствующей задаче и выберите Properties (Свойства).



2. Щелкните по закладке Configuration (Конфигурация).



3. Настройте обработку выходных данных для задачи.

Если требуется:	Примите следующие меры:
Разрешить обработку выходных данных в конце выполнения задачи.	Снимите флажок Disable Automatic Output Processing To Reduce Task Overhead (Запрет автоматической обработки вывода для уменьшения системных операций задачи) (настройка по умолчанию).
Запретить обработку выходных данных в конце выполнения задачи.	Установите флажок Disable Automatic Output Processing To Reduce Task Overhead (Запрет автоматической обработки вывода для уменьшения системных операций задачи).

4. Нажмите OK.

Конфигурирование обработки выходных данных программным путем

Чтобы написать логику для конфигурирования обработки выходных данных задачи, используйте инструкцию Set System Value (SSV) (Установить значение системы). Обращайтесь к следующему параметру объекта TASK для данной задачи:

Если требуется:	То обратитесь к следующему параметру:	Тип данных:	Инструкция:	Описание:	
Разрешить или запретить обработку выходных данных в конце выполнения задачи.	DisableUpdateOutputs («Запретить обновление выходных данных»)	DINT	GSV SSV	**Чтобы:**	**Установите параметр в:**
				Разрешить обработку выходных данных в конце выполнения задачи.	0
				Запретить обработку выходных данных в конце выполнения задачи.	1 (или любое значение, отличное от нуля)

ПРИМЕР Конфигурирование обработки выходных данных программным путем

Если Condition_1 = 0, то задаче Task_2 разрешается обрабатывать выходные данные по ее завершении.

1. Инструкция ONS ограничивает действительное выполнение инструкции SSV одним сканированием.
2. Инструкция SSV устанавливает параметр DisableUpdateOutputs («Запретить обновление выходных данных») задачи Task_2 = 0. Это позволяет задаче автоматически обрабатывать выходные данные после завершения выполнения.

Если Condition_1 = 1, то задаче Task_2 не разрешается обрабатывать выходные данные по ее завершении.

1. Инструкция ONS ограничивает действительное выполнение инструкции SSV одним сканированием.
2. Инструкция SSV устанавливает параметр DisableUpdateOutputs («Запретить обновление выходных данных») задачи Task_2 = 1. Это запрещает задаче автоматически обрабатывать выход после завершения ее выполнения.

Запрет выполнения задачи

По умолчанию каждая задача (событийная, периодическая или непрерывная) выполняется от своего триггера. У вас есть возможность предотвратить выполнение задачи при срабатывании соответствующего триггера (т.е. запретить задачу). Это полезно при тестировании, диагностике и запуске проекта.

Если требуется:	Примите следующие меры:
Разрешить выполнение задачи при срабатывании соответствующего триггера.	Снимите запрет с данной задачи (настройка по умолчанию).
Запретить выполнение задачи при срабатывании соответствующего триггера.	Запретите выполнения задачи.

ПРИМЕР

Запрет выполнения задачи

При вводе в эксплуатацию системы, использующей несколько задач, можно сначала протестировать каждую задачу в отдельности.

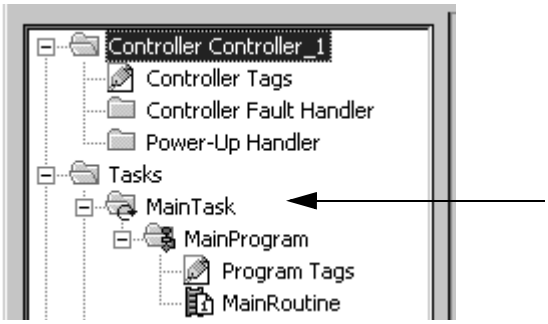
1. Запретите все задачи, кроме одной, и протестируйте эту задачу.
2. Когда задача будет отвечать всем вашим требованиям, запретите ее и снимите запрет с какой-либо другой задачи.
3. Продолжайте этот процесс до тех пор, пока вы не протестируете все свои задачи.

Если задача запрещена, контроллер все равно выполняет предварительное сканирование этой задачи при переходе контроллера из режима программирования в режим выполнения или тестирования.

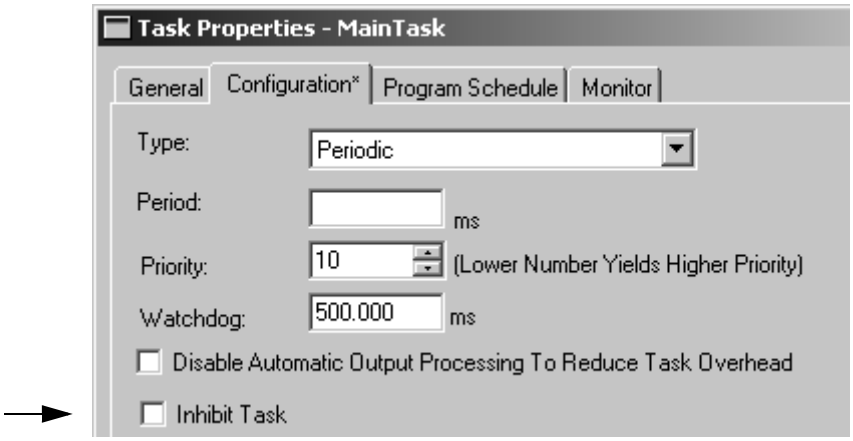
Запрет и разрешение выполнения задачи в ручном режиме

Чтобы вручную запретить выполнение задачи или снять запрет с выполнения задачи:

1. В органайзере контроллера щелкните правой кнопкой мыши по соответствующей задаче и выберите Properties (Свойства).



2. Щелкните по закладке Configuration (Конфигурация).



3. Запретите или снимите запрет на выполнение задачи.

Если требуется:	Примите следующие меры:
Разрешить выполнение задачи при срабатывании соответствующего триггера.	Снимите флажок Inhibit Task (Запрет задачи) (настройка по умолчанию).
Запретить выполнение задачи при срабатывании соответствующего триггера.	Установите флажок Inhibit Task (Запрет задачи).

4. Нажмите OK.

Запрет и разрешение выполнения задачи программным путем

Чтобы написать логику для запрета или отмены запрета какой-либо задачи, используйте инструкцию Set System Value (SSV) для обращения к следующему параметру объекта TASK для данной задачи:

Параметр	Тип данных	Инструкция	Описание	
Запрет задачи (InhibitTask)	DINT	GSV SSV	Предотвращает выполнение задачи.	
			Чтобы:	**Установите параметр в:**
			Разрешить выполнение задачи	0 (настройка по умолчанию)
			Запретить выполнение задачи	1 (или любое значение, отличное от нуля)

ПРИМЕР Запрет и отмена запрета выполнения задачи программным путем.

Если Condition_1 = 0, то выполнение задачи Task_2 разрешается.

1. Инструкция ONS ограничивает действительное выполнение инструкции SSV одним сканированием.
2. Инструкция SSV устанавливает параметр InhibitTask («Запрет задачи») задачи Task_2 = 0. Это снимет запрет с выполнения задачи.

Если Condition_1 = 1, то выполнение задачи Task_2 запрещается.

1. Инструкция ONS ограничивает действительное выполнение инструкции SSV одним сканированием.
2. Инструкция SSV устанавливает параметр InhibitTask («Запрет задачи») задачи Task_2 = 1. Это запрещает выполнение задачи.

Выбор триггера для событийной задачи

Если событийная задача сконфигурирована правильно, она прерывает все остальные задачи на минимальное время, требуемое для отклика на событие. Каждой событийной задаче требуется конкретный триггер, определяющий, когда задача должна выполняться.

Чтобы запустить событийную задачу, когда:	Используйте этот триггер:	С учетом следующих рекомендаций:
Включается или выключается дискретный вход.	Module Input Data State Change (Изменение состояния входных данных модуля)	• Только один модуль ввода может запускать конкретную событийную задачу. • Модуль ввода запускает событийную задачу в соответствии с конфигурацией изменения состояния (COS) этого модуля. Конфигурация изменения состояния определяет, какие точки при включении или выключении заставляют модуль производить данные. Такое производство данных, вызванное изменением состояния, запускает событийную задачу. • Как правило, изменение состояния следует разрешать только для одной точки модуля. Если разрешить изменение состояния для нескольких точек, может произойти наложение событийной задачи.

Чтобы запустить событийную задачу, когда:	Используйте этот триггер:	С учетом следующих рекомендаций:
Аналоговый модуль производит дискретизацию данных.	Module Input Data State Change (Изменение состояния входных данных модуля)	• Только один модуль ввода может запускать конкретную событийную задачу. • Аналоговый модуль запускает событийную задачу после каждой дискретизации в реальном времени (RTS) каналов. • Все каналы модуля используют одно значение RTS.
Контроллер получает новые данные посредством потребляемого тега.	Consumed Tag (Потребляемый тег)	• Только один потребляемый тег может активировать соответствующую событийную задачу. • Как правило, в производящем контроллере следует использовать инструкцию IOT для оповещения о производстве новых данных. Инструкция IOT определяет триггер по событию в производящем теге. Этот триггер переходит к потребляемому тегу и запускает событийную задачу. • Когда потребляемый тег активирует событийную задачу, событийная задача начинает свое выполнение только после поступления всех данных.
Включается (или выключается) вход регистрации оси.	Axis Registration (Регистрация оси) 1 или 2	• Чтобы вход регистрации активировал событийную задачу, сначала выполните инструкцию Motion Arm Registration (MAR) (Регистрации рычага перемещения). Это позволяет оси обнаруживать входные данные регистрации и в свою очередь запустить событийную задачу. • Как только входные данные регистрации запустили событийную задачу, выполните еще раз инструкцию MAR, чтобы изменить положение оси для следующего ввода регистрации. • Если время сканирования обычной логики недостаточно быстрое, чтобы изменить положение оси для следующего ввода регистрации, обратитесь к инструкции MAR во время событийной задачи.
Ось достигает положения, контрольной точки.	Axis Watch (Отслеживание оси)	• Чтобы ввод регистрации активировал событийную задачу, сначала выполните инструкцию Motion Arm Watch (MAW) (Отслеживание рычага перемещения). Это позволяет оси обнаружить контрольную точку и в свою очередь запустить событийную задачу. • Как только контрольная точка запускает событийную задачу, выполните еще раз инструкцию MAW, чтобы изменить положение оси для следующей контрольной точки. • Если время сканирования обычной логики недостаточно быстрое, чтобы изменить положение оси для следующей контрольной точки, обратитесь к инструкции MAW во время событийной задачи.
Завершается выполнение планировщика задачи управления перемещением.	Motion Group Execution (Выполнение группы перемещения)	• Время обновления данных группы перемещения запускает выполнение как планировщика перемещения, так и событийной задачи. • Поскольку планировщик задачи управления перемещением прерывает все другие задачи, он выполняется в первую очередь. Если вы установите высокий приоритет событийной задачи, она будет выполнена после завершения работы планировщика задачи управления перемещением.
Наступает определенное условие или условия логики или программы.	Инструкция EVENT	Одна и та же задача может запускаться несколькими инструкциями EVENT. Это позволяет выполнять задачи из других программ.

Ниже приведены примеры возможных ситуаций для событийных задач и соответствующих им триггеров:

Для данной ситуации:	**Используйте событийную задачу со следующим триггером:**
На упаковочной линии производится склеивание закрытых коробок. При поступлении коробки на операцию склеивания контроллер должен немедленно выполнить процедуру склеивания.	Module Input Data State Change (Изменение состояния входных данных модуля)
На производственной линии используется датчик присутствия для определения наличия детали. Поскольку датчик присутствия находится во включенном состоянии лишь очень короткое время (импульс), непрерывная задача может пропустить переход датчика из выключенного состояния во включенное.	Module Input Data State Change (Изменение состояния входных данных модуля)
На испытательном стенде двигателя нужно получить и архивировать все аналоговые данные сразу после каждой выборки данных.	Module Input Data State Change (Изменение состояния входных данных модуля)
Контроллер A производит массив производственных данных для контроллера B. Вы хотите, чтобы контроллер B не использовал значения в то время, когда контроллер A обновляет массив.	Consumed Tag (Потребляемый тег)
На линии упаковки конфет необходимо обеспечить перфорацию каждой конфеты в заданном месте. При каждом обнаружении датчиком регистрации соответствующей отметки следует проверить точность оси и при необходимости выполнить ее корректировку.	Axis Registration (Регистрация оси) 1 или 2
На участке наклейки этикеток на бутылки вам нужно проверить положение этикетки на бутылке. Проверка этикетки осуществляется по достижении осью положения контрольной точки.	Axis Watch (Отслеживание оси)
Станция склеивания должна корректировать количество наносимого клея для компенсации изменений скорости перемещения оси. После выполнения задачи планировщика перемещения необходимо проверить заданную командой скорость перемещения оси и при необходимости изменить количество клея.	Motion Group Execution (Выполнение группы перемещения)
Если какая-либо программа обнаружит опасное состояние, вся производственная линия должна быть остановлена. Процедура останова не зависит от типа опасного состояния.	Инструкция по событию (EVENT)

Триггеры, которые можно использовать для событийной, зависят от типа вашего контроллера Logix5000.

ВАЖНО Программное обеспечение RSLogix 5000 может позволить вам сконфигурировать не поддерживаемый вашим контроллером триггер для событийной задачи. При этом проект пройдет верификацию и успешно загрузится, но соответствующая событийная задача выполняться не будет.

Если ваш контроллер:	Вы можете использовать эти триггеры событийной задачи :					
	Module Input Data State Change (Изменение состояния входных данных модуля)	Consumed Tag (Потребляемый тег)	Axis Registration (Регистрация оси) 1 или 2	Axis Watch (Отслеживание оси)	Motion Group Execution (Выполнение группы перемещения)	Инструкция по событию (EVENT)
CompactLogix		√				√
FlexLogix		√				√
ControlLogix	√	√	√	√	√	√
DriveLogix		√	√	√	√	√
SoftLogix5800	√(1)	√(2)	√	√	√	√

(1) Требуется модуль ввода/вывода 1756 или виртуальная задняя шина.

(2) Контроллер SoftLogix5800 производит и потребляет теги только в сети ControlNet.

Использование триггера изменения состояния входных данных модуля

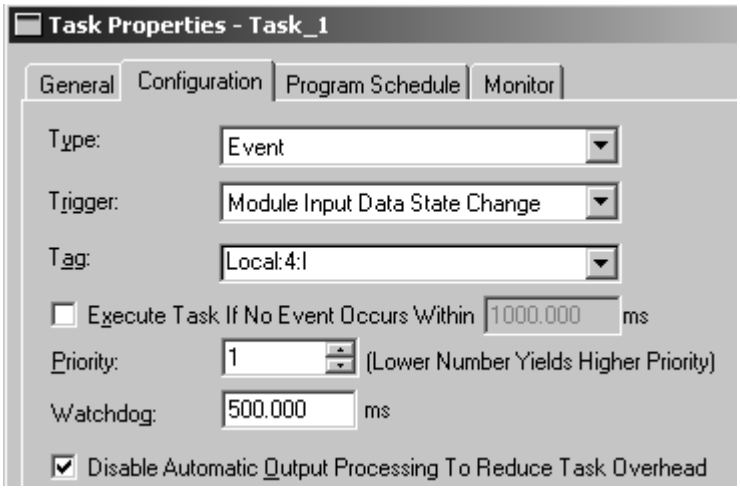
Для запуска событийной задачи в соответствии с данными, поступающими от модуля ввода, используйте триггер Module Input Data Change (Изменение данных модуля ввода).

Данная задача запускается триггером-событием

Данная задача запускается в соответствии с данными модуля ввода

Данная задача запускается этим тегом ввода

По завершении выполнения задачи не следует обновлять дискретные выходы в локальном шасси



Как модуль ввода/вывода запускает событийную задачу

Термины, применяющиеся при описании работы модуля ввода:

Термин:	Определение:
Многоадресная передача	Механизм, при котором модуль отправляет по сети данные, получаемые одновременно несколькими принимающими устройствами. Характеризует особенность линии ввода/вывода Logix5000, поддерживающей одновременное получение входных данных несколькими контроллерами от одного модуля ввода/вывода.
Требуемый интервал передачи пакетов (RPI)	RPI определяет интервал, с которым модуль осуществляет многоадресную передачу своих данных. Например, модуль ввода направляет данные в контроллер с RPI, заданным вами для этого модуля. • Допустимый диапазон – от 0,2 мс (200 микросекунд) до 750 мс. • По истечении заданного времени модуль осуществляет многоадресную передачу данных. Это также называется циклическим обновлением.

Термин:	Определение:
Дискретизация в реальном времени (RTS)	RTS определяет, когда аналоговый модуль должен сканировать свои каналы и осуществлять многоадресную передачу данных (обновлять буфер ввода, а затем осуществлять многоадресную передачу). • RPI определяет, когда модуль должен осуществлять многоадресную передачу текущего содержимого буфера ввода без сканирования (обновления) каналов. • Модуль каждый раз сбрасывает таймер RPI, и происходит передача RTS.
Изменение состояния	Параметр изменения состояния (COS) указывает модулю дискретного ввода осуществлять многоадресную передачу данных всякий раз, когда определенная точка входа переходит из включенного состояния в выключенное, или из выключенного состояния во включенное. • Изменение состояния устанавливается для каждой точки в отдельности. • Когда любая точка, для которой установлен параметр изменения состояния, получает соответствующее изменение, модуль осуществляет многоадресную передачу данных для всех своих точек. • По умолчанию изменение стояния активируется для всех точек как при переходе из включенного состояния в выключенное, так и при переходе из выключенного состояния во включенное. • Необходимо указать RPI независимо от того, активируете ли вы изменение состояния. Если в пределах интервала RPI не происходит никакого изменения, то модуль передает данные по истечении RPI.

В следующей таблице указывается, в каких случаях модуль ввода осуществляет многоадресную передачу данных и запускает событийную задачу в пределах своего шасси.

Если модуль ввода:	И:	То он осуществляет многоадресную передачу:	И запускает событийную задачу:
Дискретный	Параметр изменения состояния активирован для какой-либо точки модуля.	• При получении заданного изменения любой точкой, для которой активирован параметр изменения состояния. • С интервалом RPI.	При получении заданного изменения любой точкой, для которой активирован параметр изменения состояния.
	Параметр изменения состояния не активирован ни для одной точки модуля.	С интервалом RPI	Никогда
Аналоговый	RTS ≤ RPI	С интервалом RTS (последнее обновление данных канала).	С интервалом RTS для данного модуля.
	RTS > RPI	• С интервалом RTS (последнее обновление данных канала). • С интервалом RPI (не содержит обновленные данные каналов).	С интервалом RTS для данного модуля.

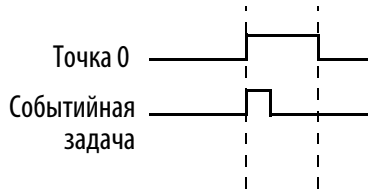
Если модуль находится в удаленном шасси, то только RPI определяет, когда данный контроллер получит по сети данные и триггер-событие.

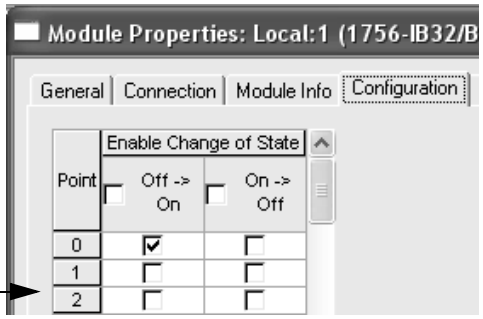
Через сеть:	Контроллер получает данные:
EtherNet/IP	С интервалом, в среднем близким к RPI.
ControlNet	С фактическим интервалом передачи пакетов (≤ RPI).

Далее приводится несколько примеров использования параметра изменения состояния COS и RTS:

ВАЖНО Если вы используете дискретный модуль для запуска событийной задачи, сконфигурируйте только одну точку модуля для параметра изменения состояния. Если вы сконфигурируете несколько точек, то может произойти наложение задач.

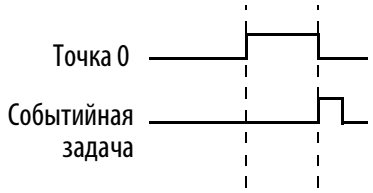
Если нужно, чтобы:	**То сконфигурируйте модуль ввода следующим образом (точка 0 используется лишь для примера):**

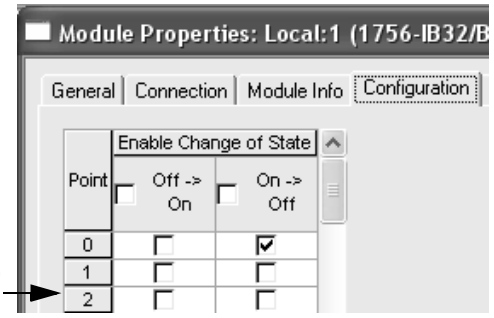




Изменение состояния

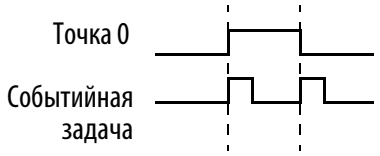
Для остальных точек изменение состояния отключено

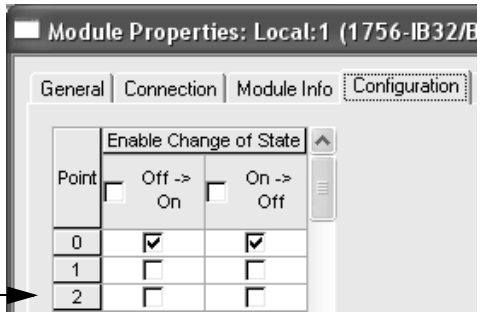




Изменение состояния

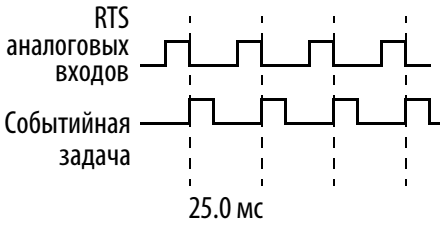
Для остальных точек изменение состояния отключено

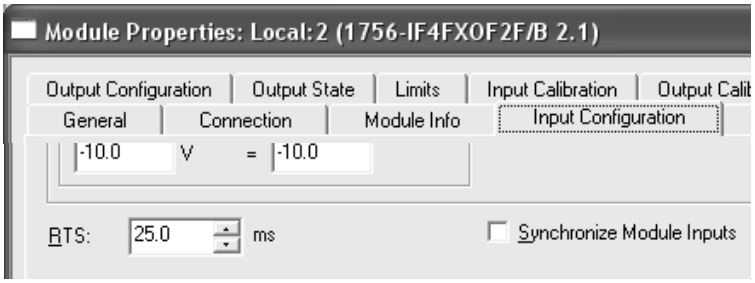




Изменение состояния

Для остальных точек изменение состояния отключено





Дискретизация входных данных в реальном времени

Убедитесь, что модуль может запускать событийную задачу

Чтобы можно было использовать модуль ввода для запуска событийной задачи, этот модуль должен поддерживать запуск событийных задач. Если данный модуль находится в удаленном положении, соответствующие коммуникационные модули также должны поддерживать запуск событийных задач.

В следующей таблице перечисляются модули Rockwell Automation, протестированные нами на предмет запуска событийных задач. Некоторые модули других поставщиков также могут поддерживать запуск событийных задач. Прежде чем использовать модуль сторонних производителей, убедитесь в его пригодности.

Категория	Модули	
Дискретные модули ввода/вывода, которые поддерживают изменение состояния.	1756-IA8D 1756-IA16I 1756-IB16 1756-IB16I 1756-IB32 1756-IG16 1756-IH16ISOE 1756-IN16 1756-IV32	1756-IA16 1756-IA32 1756-IB16D 1756-IB16ISOE 1756-IC16 1756-IH16I 1756-IM16I 1756-IV16
Аналоговые модули ввода/вывода, которые поддерживают дискретизацию в реальном времени.	1756-IF16 1756-IF6CIS 1756-IF8 1756-IT6I	1756-IF4FXOF2F/A 1756-IF6I 1756-IR6I 1756-IT6I2
Коммуникационные модули, которые обеспечивают соединения, оптимизированные для рэка.	1756-CNB/A 1756-CNB/D 1756-CNBR/B 1756-DNB 1756-SYNCH/A	1756-CNB/B 1756-CNBR/A 1756-CNBR/D 1756-ENBT/A 1784-PCIDS/A
Универсальные модули ввода/вывода, поддерживающие события CIP.	1756-MODULE 1789-MODULE	

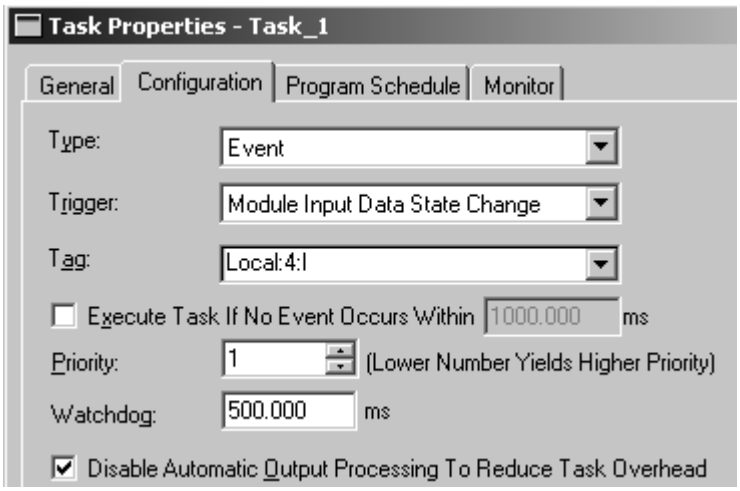
Контрольный перечень для задачи, запускаемой событием ввода

Для этого параметра:	Обеспечьте выполнение следующих требований:
☐ 1. Тип модуля ввода	Для максимального быстродействия используйте следующие модули: • Для максимального дискретного быстродействия используйте модуль 1756-IB32/B. • Для максимального аналогового быстродействия используйте модуль 1756-IF4FXOF2F.
☐ 2. Положение модуля ввода/вывода	Поместите модуль, запускающий событие, и модули, откликающиеся на это событие (выводы), в то же шасси, где находится контроллер. Сетевой обмен данными удаленных модулей влияет на быстродействие.
☐ 3. Количество удаленных модулей	Ограничьте количество модулей в локальном шасси. Дополнительные модули увеличивают вероятность задержек в задней шине.
☐ 4. Изменение состояния	Если событие запускается дискретным устройством, то активируйте изменение состояния лишь для той точки, которая запускает соответствующую событийную задачу. • Разрешите изменение состояния для типа перехода, запускающего задачу: Выкл → Вкл, Вкл → Выл или обоих. • Если вы сконфигурируете изменение состояния и для перехода Выкл→ Вкл, и для Вкл → Выкл, то данная точка будет запускать событийную задачу при всяком ее включении и выключении. Убедитесь в том, что продолжительность ввода больше времени сканирования задачи. В противном случае может произойти наложение. • Отключите изменение состояния (снимите флажок) для остальных точек модуля ввода. Если вы сконфигурируете для параметра изменения состояния несколько точек модуля, то каждая из них сможет запускать событийную задачу. Это может привести к наложению.
☐ 5. Приоритет задачи	Установите для событийной задачи самый высокий приоритет. Если какая-либо периодическая задача имеет более высокий приоритет, событийной задаче придется ожидать завершения выполнения такой периодической задачи.
☐ 6. Планировщик задачи управления перемещением	Планировщик перемещения прерывает все другие задачи независимо от их приоритета. • Количество осей и время обновления данных группы перемещения влияют на продолжительность и частоту выполнения планировщика задачи управления перемещением. • Если планировщик задачи управления перемещением выполнятся во время начала задачи, то задача ожидает завершения планировщика перемещения. • Если обновление данных выполняется во время выполнения задачи, задача останавливается, чтобы выполнить действия планировщика задачи управления перемещением.
☐ 7. Количество событийных задач	Следует ограничить количество событийных задач. Каждая дополнительная задача уменьшает время обработки, отводимое другим задачам. Это может привести к наложению.
☐ 8. Автоматическая обработка выходных данных	Для событийной задачи вы, как правило, можете отключить автоматическую обработку вывода (настройка по умолчанию). Это уменьшает время выполнения задачи. Для принятия решения по этой настройке воспользуйтесь рисунком на странице 19.
☐ 9. Инструкция IOT	Используйте инструкцию IOT для каждого модуля вывода, к которому вы обращаетесь в событийной задаче. Инструкция IOT отменяет RPI для данного модуля и осуществляет немедленную передачу данных.

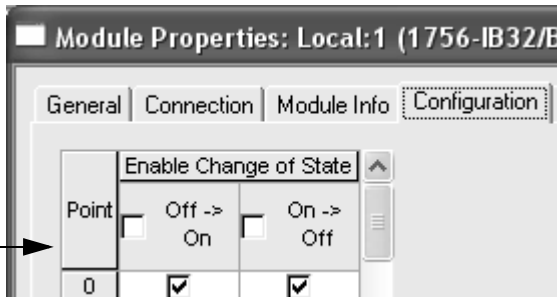
ПРИМЕР

При прохождении деталей мимо сбрасывателя контроллер должен принять решение о том, следует ли включать сбрасыватель. Если сбрасыватель включен, контроллер также должен выключить его до поступления на это место следующей детали. За счет скорости линии событийная задача управляет сбрасывателем.

Фотоэлектрическое устройство, находящееся в месте установки сбрасывателя, фиксирует, когда деталь достигает положения сбрасывателя. Вход подключен к модулю через слот 4 локального шасси.



Фотоэлектрическое устройство сбрасывателя (точка 0) настроено на изменение состояния и для состояния Вкл. и для Выкл. За счет этого фотоэлектрическое устройство Запускает событийную задачу, при включении и выключении.



Эта событийная задача использует следующую логику для управления сбрасывателем:

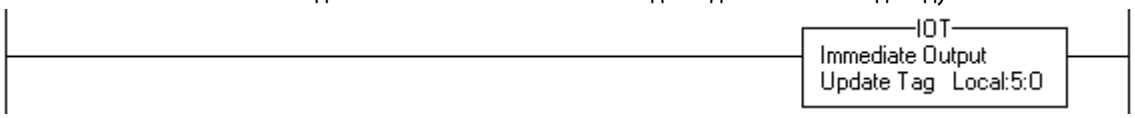
If Diverter_Photoeye = 1 (деталь находится в месте положения сбрасывателя)

And Divert_Part = 1 (сбросить эту деталь)

Then Diverter = 1 (включить сбрасыватель)

Otherwise Diverter = 0 (выключить сбрасыватель)

Немедленно отсылает значения выходных данных на выход модуля в слоте 5.



Оценка времени цикла

Для оценки времени цикла от входа до выхода воспользуйтесь следующей таблицей:

Фактор	Значение
1. Каково время входного фильтра для модуля, запускающего данную событийную задачу? Обычно оно указывается в миллисекундах. Переведите его в микросекунды (мкс).	мкс
2. Каково время аппаратного отклика для модуля ввода, запускающего данную событийную задачу? Убедитесь в том, что вы используете надлежащий тип перехода (Выкл ? Вкл или Вкл ? Выкл). См. «Номинальный аппаратный отклик модуля ввода 1756, который наиболее часто используется с событийными задачами», страница 34.	мкс
3. Каково время коммуникации через заднюю шину?	мкс

Если размер шасси:	Используйте следующее значение (в худшем случае):
4 слота	13 мкс
7 слотов	22 мкс
10 слотов	32 мкс
13 слотов	42 мкс
17 слотов	54 мкс

4. Каково общее время выполнения программ событийной задачи?	мкс
5. Каково время обмена данными через заднюю шину? (То же значение, что и в п.3).	мкс
6. Каково время аппаратного отклика для модуля вывода?	мкс
7. Сложите значения по пунктам с 1 по 6. Это минимальная оценка времени цикла без учета задержек и прерываний событийной задачи из-за выполнения планировщика задачи управления перемещением или других задач.	мкс
8. Каково время сканирования группы перемещения?	мкс
9. Каково общее время сканирования задач, приоритет которых выше приоритета данной событийной задачи (если таковые имеются)?	мкс
10. Сложите значения по пунктам с 7 по 9. Это оценка номинального времени цикла с учетом задержек и прерываний событийной задачи из-за выполнения планировщика перемещения или других задач.	мкс

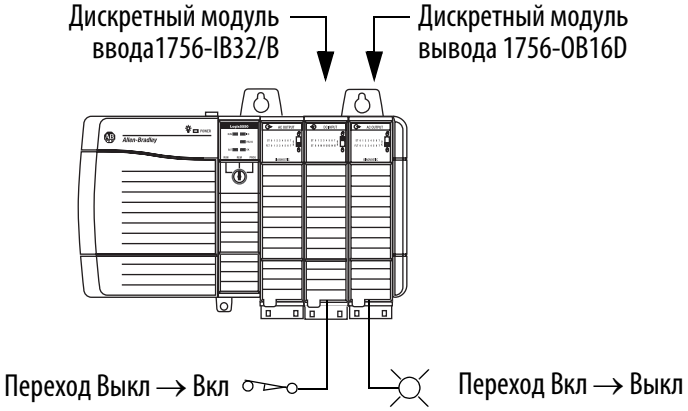
Номинальный аппаратный отклик модуля ввода 1756, который наиболее часто используется с событийными задачами

Модуль	Номинальное время отклика, мкс			
	25° C		60° C	
	Выкл → Вкл	Вкл → Выкл	Выкл → Вкл	Вкл → Выкл
1756-IB16	265	582	265	638
1756-IB16D	303	613	305	673
1756-IB32/B	330	359	345	378
1756-IV16	257	435	254	489
1756-IV32	381	476	319	536
1756-OB16D	48	519	51	573
1756-OB16E	60	290	61	324
1756-OB32	38	160	49	179
1756-OV16E	67	260	65	326
1756-OV32E	65	174	66	210

ПРИМЕР

Оценка времени цикла

В следующем примере время цикла оценивается для приведенной ниже системы. В этом примере временем цикла является время от включения входа до включения выхода.



Фактор	Значение
1. Каково время входного фильтра для модуля, запускающего данную событийную задачу? Обычно оно указывается в миллисекундах. Переведите его в микросекунды (мкс).	0 мкс
2. Каково время аппаратного отклика для модуля ввода, запускающего данную событийную задачу? Убедитесь в том, что вы используете надлежащий тип перехода (Выкл → Вкл или Вкл → Выкл). См. «Номинальный аппаратный отклик модуля ввода 1756, который наиболее часто используется с событийными задачами», страница 34.	330 мкс
3. Каково время коммуникации через заднюю шину?	
Если размер шасси: — **Используйте следующее значение (в худшем случае):**	
4 слота — 13 мкс	
7 слотов — 22 мкс	
10 слотов — 32 мкс	
13 слотов — 42 мкс	
17 слотов — 54 мкс	13 мкс
4. Каково общее время выполнения программ событийной задачи?	400 мкс
5. Каково время коммуникации через заднюю шину? Каково время обмена данными через заднюю шину? (То же значение, что и в п.3).	13 мкс
6. Каково время аппаратного отклика для модуля вывода?	51 мкс
7. Сложите значения по пунктам с 1 по 6. Это минимальная оценка времени цикла без учета задержек и прерываний событийной задачи из-за выполнения планировщика задачи управления перемещением или других задач.	807 мкс
8. Каково время сканирования группы перемещения?	1130 мкс
9. Каково общее время сканирования задач, приоритет которых выше приоритета данной событийной задачи (если таковые имеются)?	0 мкс
10. Сложите значения по пунктам с 7 по 9. Это минимальная оценка времени цикла с учетом задержек и прерываний событийной задачи из-за выполнения планировщика задачи управления перемещением или других задач.	1937 мкс

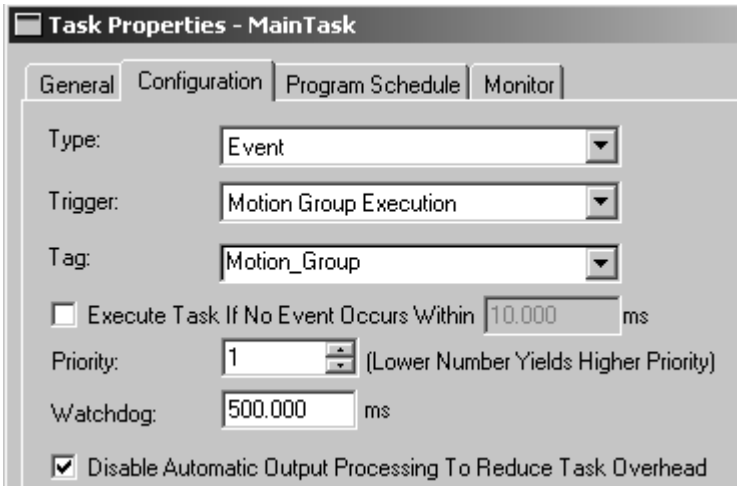
Дополнительные рекомендации

Следующие факторы влияют на время сканирования событийной задачи, что в свою очередь влияет на скорость ее отклика на входной сигнал.

Фактор	Описание
Размер программы событийной задачи	Каждый элемент логики (цепочка, инструкция, элемент структурного текста и т.д.) увеличивает время сканирования задачи.
Приоритет задачи	Если данная событийная задача не является самой высокоприоритетной, то задача с более высоким приоритетом может задержать или прервать выполнение данной задачи.
Инструкции CPS и UID	Если активна одна из этих инструкций, то событийная задача не может прерывать выполняющуюся в данный момент задачу. (Задача с CPS или UID).
Прерывания обмена данных	Следующие действия контроллера прерывают задачу независимо от ее приоритета: • Обмен данными с модулями ввода/вывода. Модули с большими пакетами данных, такие как 1756-DNB, оказывают большее влияние. • Обмен данными через последовательный порт.

Использование триггера группы перемещения

Чтобы совместить выполнение событийной задачи с выполнением планировщика задачи управления перемещением, используйте триггер Motion Group Execution (Выполнение группы перемещения).

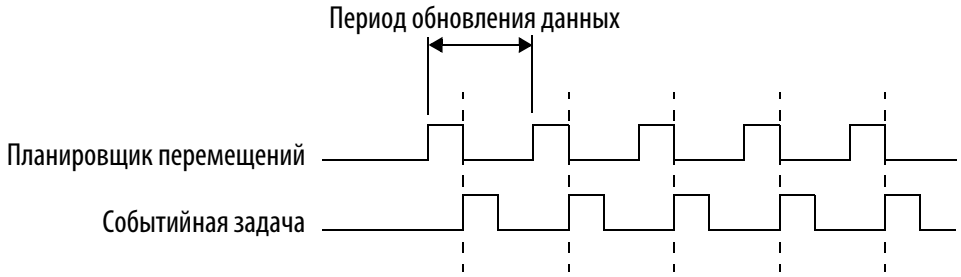


Триггер Motion Group Execution работает следующим образом:

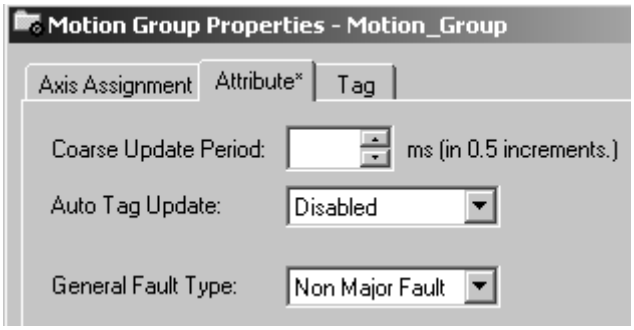
- Время обновления данных группы перемещения запускает выполнение, как планировщика перемещения, так и событийной задачи.

- Поскольку планировщик задачи управления перемещением прерывает все другие задачи, он выполняется в первую очередь. Если вы установите высокий приоритет задачи для событийной задачи, она будет выполнена после завершения программы планирования перемещения.

На следующей временной диаграмме показано соотношение между планировщиком перемещения и событийной задачей.



Период обновления данных запускает как планировщика перемещения, так и событийную задачу.

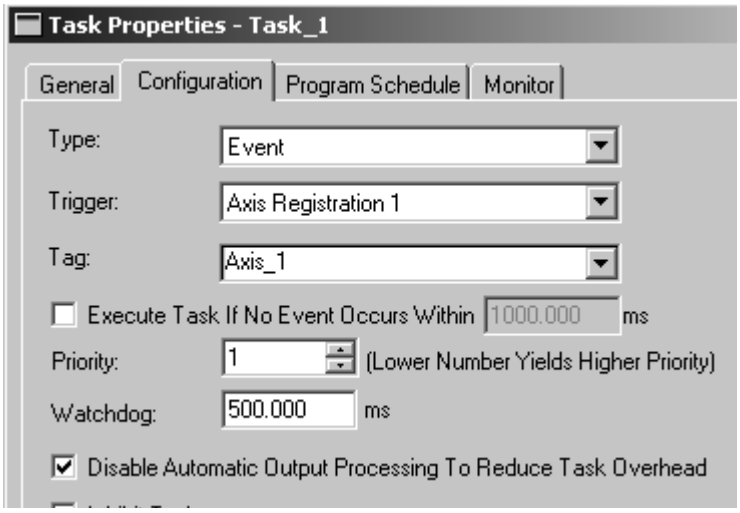


Контрольный перечень для задачи группы перемещения

Для этого параметра:	Обеспечьте выполнение следующих требований:
☐ 1. Время сканирования	Убедитесь в том, что время сканирования задачи значительно меньше периода обновления данных группы перемещения. В противном случае может произойти наложение задачи.
☐ 2. Приоритет задачи	Установите для событийной задачи самый высокий приоритет. Если какая-либо периодическая задача имеет более высокий приоритет, событийной задаче придется ожидать завершения выполнения такой периодической задачи.
☐ 3. Количество событийных задач	Следует ограничить количество событийных задач. Каждая дополнительная задача уменьшает время обработки, отводимое другим задачам. Это может привести к наложению.
☐ 4. Автоматическая обработка выходных данных	Для событийной задачи вы, как правило, можете отключить автоматическую обработку вывода (настройка по умолчанию). Это уменьшает время выполнения задачи.

Использование триггера регистрации оси

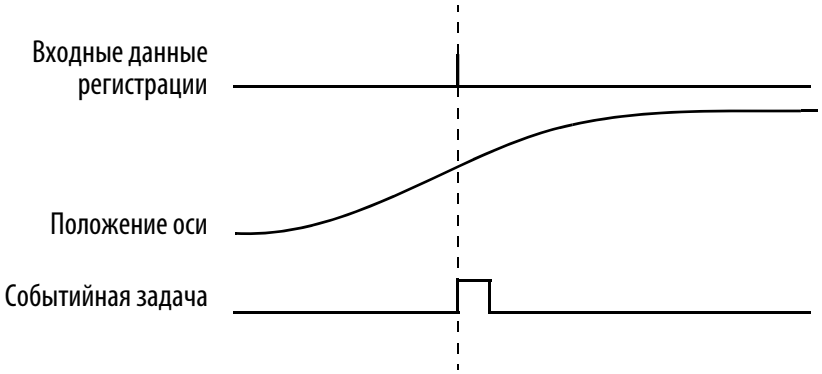
Чтобы входные данные регистрации оси запускали событийную задачу, используйте триггер Axis Registration (1 или 2) (Регистрация оси).



По достижении указанным регистрационным входом условий его срабатывания он запускает соответствующую событийную задачу.

- В настройках событийной задачи укажите, какие входные данные регистрации должны запускать эту задачу. Выберите Axis Registration 1 (Регистрация оси 1) или Axis Registration 2 (Регистрация оси 2).
- Сначала вы должны активизировать входные данные регистрации при помощи инструкции Motion Arm Registration (Регистрация рычага перемещения) (MAR).
- В инструкции MAR операнд Trigger Condition (Условия срабатывания) определяет, какой переход входных данных регистрации (Выкл → Вкл или Вкл → Выкл) будет запускать данную событийную задачу.

На следующей временной диаграмме показано соотношение между регистрационным входом и событийной задачей.



Контрольный перечень для задачи регистрации оси

Для этого параметра:	Обеспечьте выполнение следующих требований:
☐ 1. Входные данные регистрации	Активизируйте входные данные регистрации (инструкция MAR). Это позволяет оси обнаруживать входные данные регистрации и запускать событийную задачу. • Сначала активизируйте входные данные регистрации для обнаружения первого состояния срабатывания. • Повторно активизируйте входные данные регистрации после каждого выполнения событийной задачи. • Активизируйте входные данные регистрации достаточно быстро для обнаружения каждого состояния срабатывания.
	Если ваша обычная логика: / **То:**
	Работает достаточно быстро для повторной активизации входных данных регистрации между состояниями срабатывания. Например, ваша обычная логика всегда выполняет как минимум 2 сканирования между входными данными регистрации. / При желании можно активизировать входные данные регистрации в вашей обычной логике.
	Работает недостаточно быстро для повторной активизации входных данных регистрации. / Активизируйте входные данные регистрации в соответствующей событийной задаче.
☐ 2. Приоритет задачи	Установите для событийной задачи самый высокий приоритет. Если какая-либо периодическая задача имеет более высокий приоритет, событийной задаче придется ожидать завершения выполнения такой периодической задачи.
☐ 3. Количество событийных задач	Следует ограничить количество событийных задач. Каждая дополнительная задача уменьшает время обработки, отводимое другим задачам. Это может привести к наложению.
☐ 4. Автоматическая обработка выходных данных	Для событийной задачи вы, как правило, можете отключить автоматическую обработку вывода (настройка по умолчанию). Это уменьшает время выполнения задачи.

ПРИМЕР

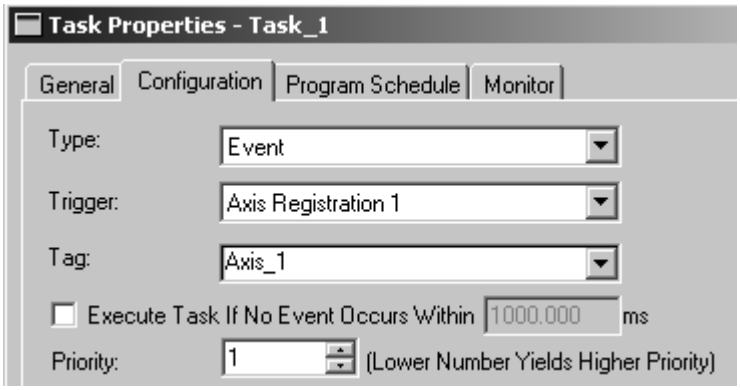
На линии упаковки конфет необходимо обеспечить перфорацию каждой конфеты в заданном месте.

- При каждом обнаружении датчиком регистрации соответствующей отметки следует проверить точность оси и при необходимости выполнить ее корректировку.
- Из-за скорости линии входные данные регистрации должны активизироваться в событийной задаче.

Датчик регистрации подключен в качестве входных данных регистрации 1…

…для оси с именем Axis_1.

Эта событийная задача прерывает все остальные задачи.



Следующая логика активизирует и повторно активизирует входные данные регистрации:

Непрерывная задача

Если Arm_Registration = 1 (система готова к поиску регистрационной отметки) то

Инструкция ONS ограничивает выполнение инструкции по событию (EVENT) одним сканированием.

Инструкция по событию (EVENT) запускает выполнение задачи Task_1 (событийной задачи).

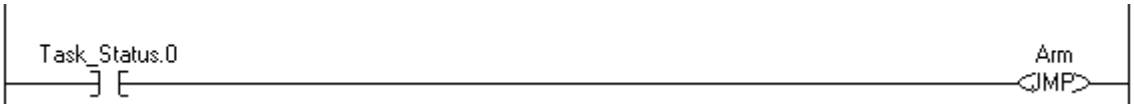
Task_1 (событийная задача)

Инструкция GSV для задачи Task_2 устанавливает параметр Task_Status («Состояние задачи») (тег DINT) =Status («Состояние»). В параметре Instance Name («Имя события») THIS означает объект TASK для задачи, в которой находится данная инструкция (т.е. Task_1).

Продолжение на следующей странице

Если Task_Status.0 = 1, то инструкция по событию запускает событийную задачу. В непрерывной задаче инструкция по событию выполняется для первоначальной активизации регистрации.

Инструкция JMP вызывает переход контроллера к выполнению инструкции Arm LBL. При этом пропускается вся логика данной процедуры, кроме цепочки, активизирующей регистрацию данной оси.



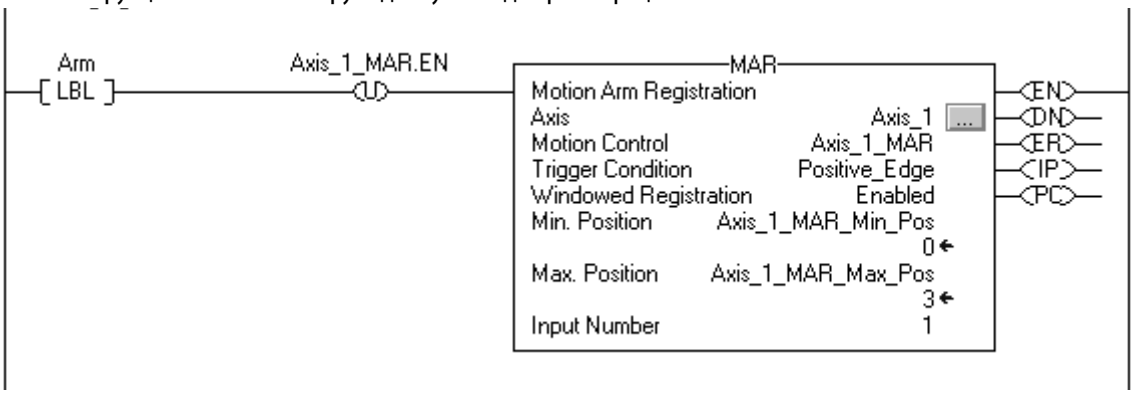
-
- Цепочки релейной логики.
-

Инструкция MAR выполняется при каждом выполнении данной задачи и активизирует регистрацию оси Axis_1.

Инструкция OUT устанавливает бит EN инструкции MAR = 0.

- Инструкция MAR является инструкцией перехода.
- Для выполнения инструкции MAR входное состояние цепочки для нее должно перейти из состояния «ложь» в состояние «истина».
- При первом сбросе бита EN инструкция реагирует таким же образом, как если бы входное состояние цепочки изменилось с «ложь» на «истина».

Инструкция MAR активизирует данную ось для регистрации.

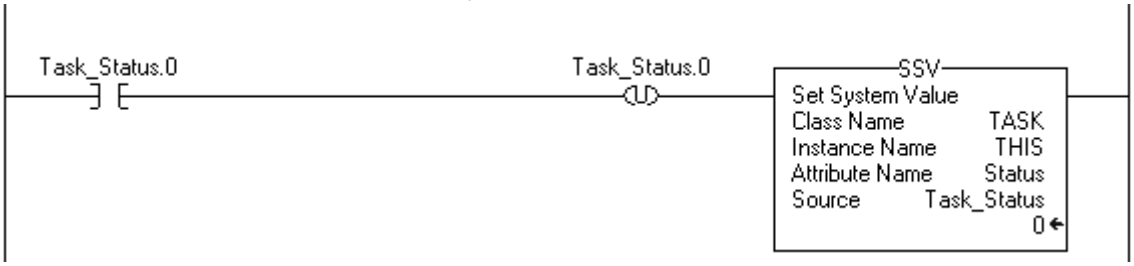


Контроллер не сбрасывает установленные биты параметра Status («Состояние»). Чтобы использовать бит для новой информации о состоянии, необходимо вручную сбросить соответствующий бит.

Если Task_Status.0 = 1, тогда сбросьте этот бит.

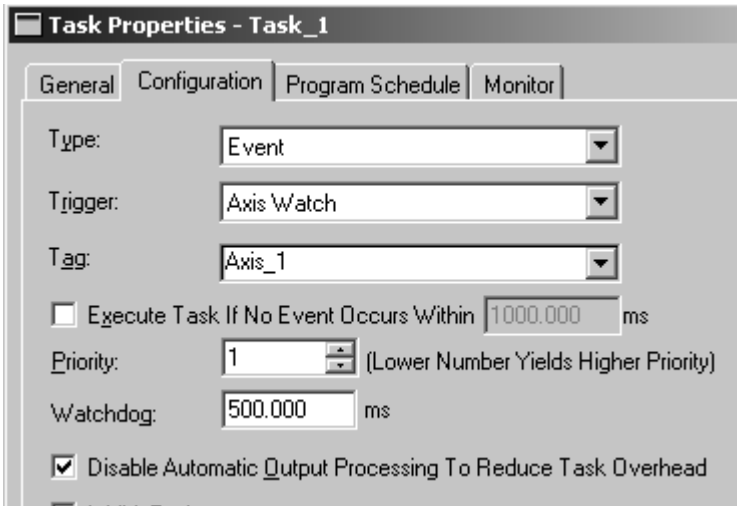
Инструкция OTU устанавливает Task_Status.0 = 0.

Инструкция SSV устанавливает параметр Status задачи THIS («Данная задача») (Task_1) = Task_Status («Состояние задачи»). Сюда входит и сброшенный бит.



Использование триггера отслеживания работы оси

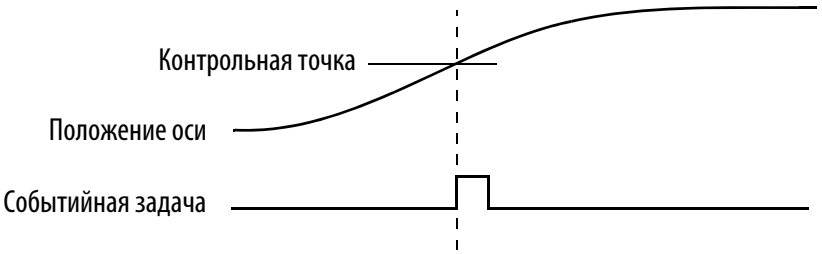
Чтобы контрольное положение оси запускало событийную задачу, используйте триггер Axis Watch (Отслеживание положения оси).



Данная событийная задача будет запущена по достижении осью указанной контрольной точки.

- Сначала вам необходимо активизировать ось для контрольного положения при помощи инструкции отслеживания положения оси (MAW).
- Операнд Trigger Condition (Условие триггера) инструкции MAW задает направление, в котором должна перемещаться ось, чтобы была запущена данная событийная задача.
- После достижения осью контрольного положения и запуска событийной задачи вы должны вновь активизировать ось для следующего контрольного положения.

На следующей временной диаграмме показано соотношение между контрольной точкой и событийной задачей.



Контрольный перечень для задачи отслеживания работы оси

Для этого параметра:	Обеспечьте выполнение следующих требований:	
☐ 1. Контрольное положение	Используйте инструкцию MAR для задания контрольного положения. Это позволяет оси запускать событийную задачу по достижении ею заданного контрольного положения. • Сначала активизируйте ось для обнаружения первого контрольного положения. • После достижения осью контрольного положения и запуска событийной задачи вновь активизируйте ось для следующего контрольного положения. • Повторно активизируйте ось достаточно быстро для обнаружения каждого контрольного положения.	
	Если ваша обычная логика:	**То:**
	Работает достаточно быстро для повторной активизации оси между контрольными положениями. Например, ваша обычная логика всегда выполняет как минимум 2 сканирования между контрольными положениями.	При желании вы можете активизировать ось в вашей обычной логике.
	Работает недостаточно быстро для повторной активизации оси.	Активизируйте ось в соответствующей событийной задаче.
☐ 2. Приоритет задачи	Установите для событийной задачи самый высокий приоритет. Если какая-либо периодическая задача имеет более высокий приоритет, событийной задаче придется ожидать завершения выполнения такой периодической задачи.	
☐ 3. Количество событийных задач	Следует ограничить количество событийных задач. Каждая дополнительная задача уменьшает время обработки, отводимое другим задачам. Это может привести к наложению.	
☐ 4. Автоматическая обработка выходных данных	Для событийной задачи вы, как правило, можете отключить автоматическую обработку вывода (настройка по умолчанию). Это уменьшает время выполнения задачи.	

ПРИМЕР

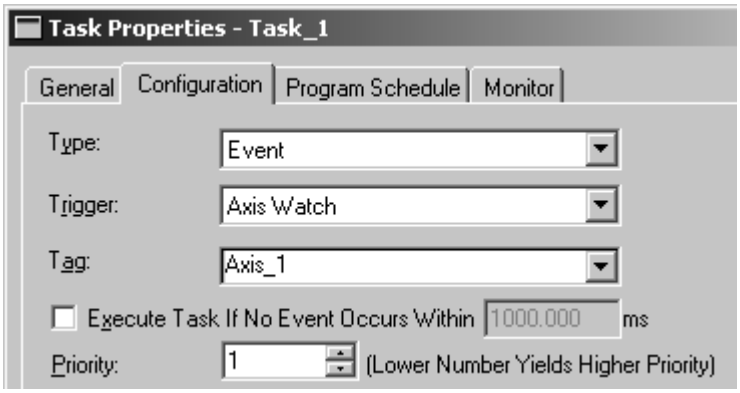
На участке наклейки этикеток на бутылки вам нужно проверить положение этикетки на бутылке.

- По достижении осью положения, заданного в качестве контрольной точки, осуществляется проверка этикетки и, при необходимости, выполняется корректировка.
- Из-за скорости линии вы должны активизировать ось для контрольного положения в событийной задаче.

Эта событийная задача запускается контрольным положением.... →

...для оси с именем Axis_1. →

Эта событийная задача прерывает все остальные задачи. →



Следующая логика активизирует и повторно активизирует указанную ось для заданного контрольного положения:

Непрерывная задача

Если Arm_Watch = 1 (система готова к установке контрольного положения), то

Инструкция ONS ограничивает выполнение инструкции по событию одним сканированием.

Инструкция по событию (EVENT) запускает выполнение задачи Task_1 (событийной задачи).

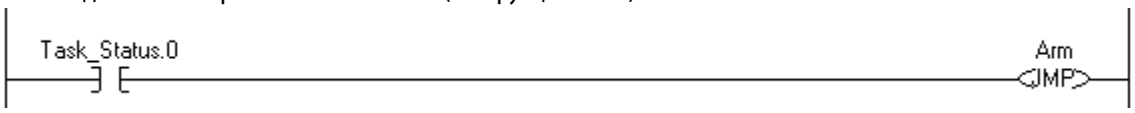
Task_1 (событийная задача)

Инструкция GSV для задачи Task_2 устанавливает параметр Task_Status («Состояние задачи») (тег DINT) =Status («Состояние»). В параметре Instance Name («Имя события») THIS означает объект TASK для задачи, в которой находится данная инструкция (т.е. Task_1).

Продолжение на следующей странице

Если Task_Status.0 = 1, то инструкция по событию запускает событийную задачу. В непрерывной задаче инструкция по событию выполняется для первоначальной установки контрольного положения.

Инструкция JMP вызывает переход контроллера к выполнению инструкции Arm LBL. При этом пропускается вся логика данной процедуры, кроме цепочки, активизирующей данную ось для заданного контрольного положения (инструкция MAW).



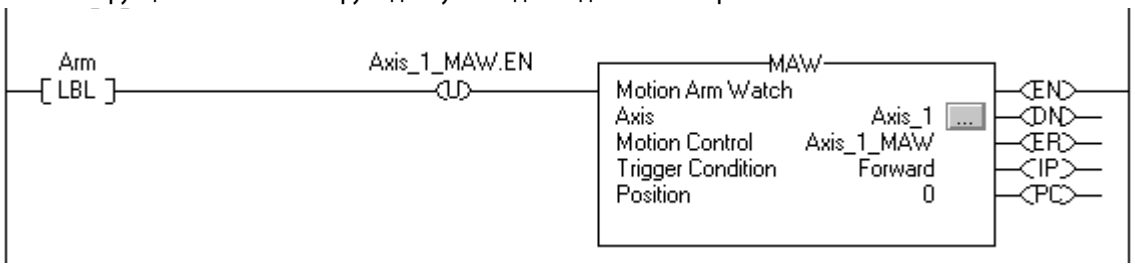
-
- Цепочки релейной логики.
-

Инструкция MAW выполняется при каждом выполнении данной задачи и активизирует ось Axis_1 для заданного контрольного положения.

Инструкция OTU устанавливает бит EN инструкции MAW = 0.

- Инструкция MAW является инструкцией перехода.
- Для выполнения инструкции MAW состояние цепочки для нее должно перейти из состояния «ложь» в состояние «истина».
- При первом сбросе бита EN инструкция реагирует таким же образом, как если бы входное состояние цепочки изменилось с «ложь» на «истина».

Инструкция MAW активизирует данную ось для заданного контрольного положения.

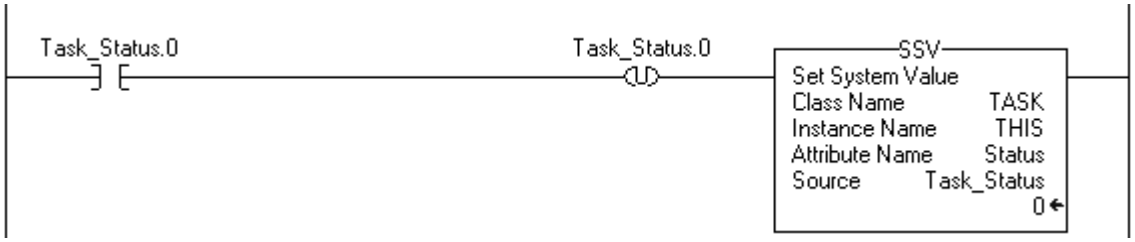


Контроллер не сбрасывает установленные биты параметра Status («Состояние»). Чтобы использовать бит для новой информации о состоянии, необходимо вручную сбросить соответствующий бит.

Если Task_Status.0 = 1, тогда сбросьте этот бит.

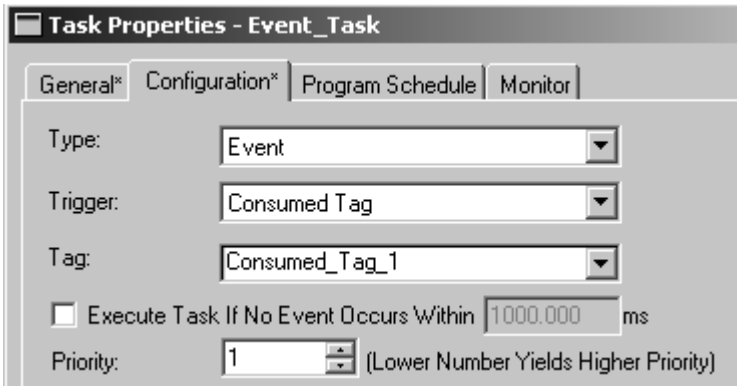
Инструкция OTU устанавливает Task_Status.0 = 0.

Инструкция SSV устанавливает параметр Status задачи THIS («Данная задача») (Task_1) = Task_Status («Состояние задачи»). Сюда входит и сброшенный бит.



Использование триггера потребляемого тега

Для запуска событийной задачи в соответствии с данными потребляемого тега используйте триггер Consumed Tag (Потребляемый тег).

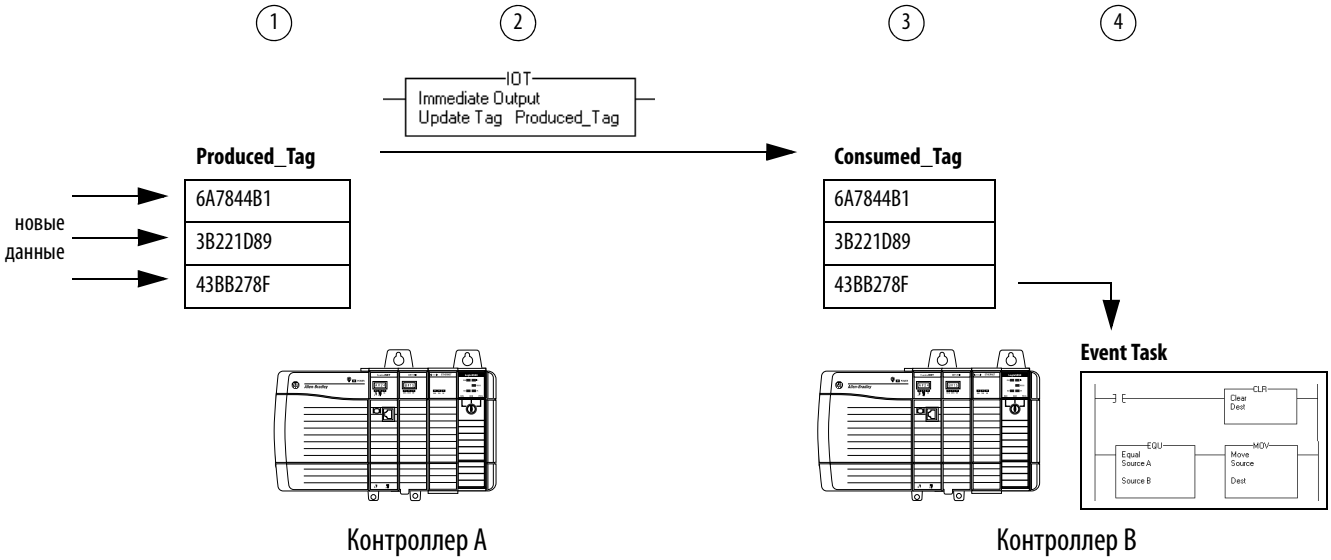


Эта задача запускается событием.

Эта задача запускается потребляемым тегом.

Эта задача запускается этим потребляемым тегом.

Взаимоотношение между производимым и потребляемым тегом может обеспечивать передачу триггера событийной задачи вместе с данными в потребляющий контроллер. Как правило, для отправки триггера событийной задачи в потребляющий контроллер вы используете инструкцию Immediate Output (IOT) (Мгновенные выход).



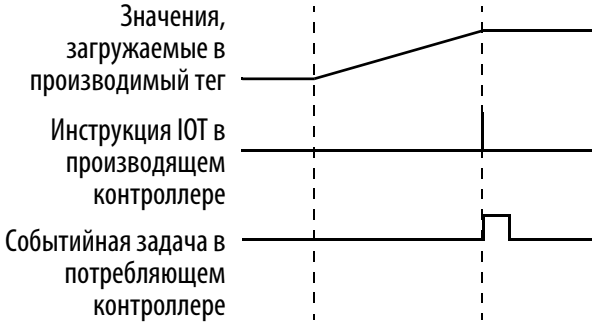
Описание:	
1	В контроллере A логика обновляет значения производимого тега.
2	По завершении обновления контроллер A выполняет инструкцию IOT для отправки соответствующих данных и триггера событийной задачи в контроллер B.
3	Контроллер B получает новые данные.
4	После обновления потребляемого тега контроллер B выполняет событийную задачу.

Тип сети, соединяющей, определяет время получения потребляющим контроллером новых данных и триггера событийной задачи посредством инструкции IOT.

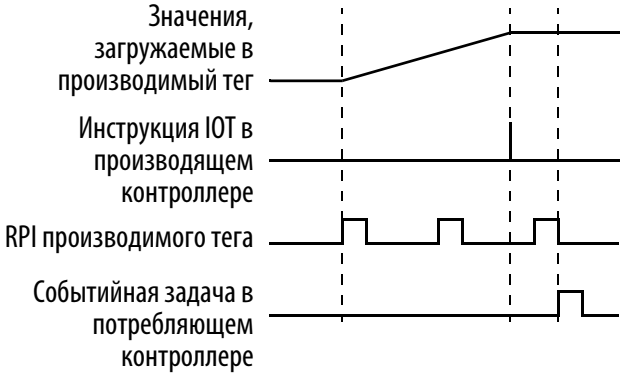
С данным контроллером:	В этой сети:	Устройство-потребитель получает данные и событие-триггер:
ControlLogix	Задняя шина	Немедленно
	Сеть EtherNet/IP	Немедленно
	Сеть ControlNet	Через фактический интервал пакетной передачи (API) для потребляемого тега (соединения).
SoftLogix5800	Можно производить и потреблять теги только в сети ControlNet.	Через фактический интервал пакетной передачи (API) для потребляемого тега (соединения).

Следующие диаграммы позволяют сравнить получение данных при помощи инструкции IOT в сетях EtherNet/IP и ControlNet.

Сеть EtherNet/IP (контроллер ControlLogix)

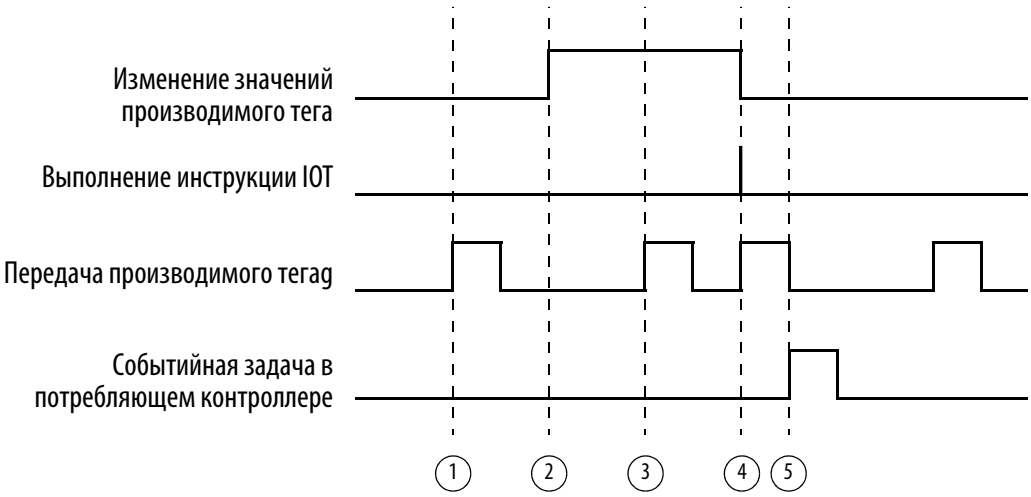


Сеть ControlNet



Обеспечение целостности данных

Событийная задача с триггером потребляемого тега обеспечивает простой механизм передачи данных контроллеру, гарантируя, что контроллер не будет использовать данные во время их изменения.

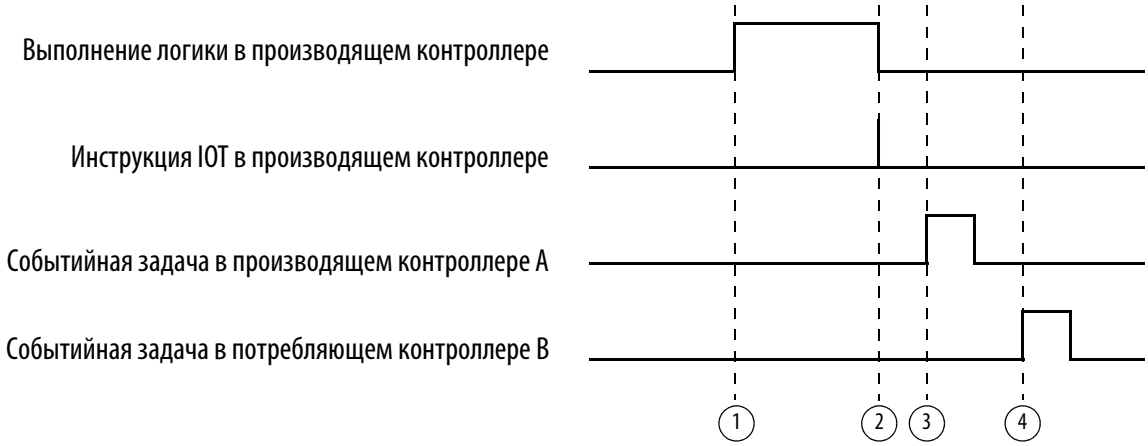


Описание:	
1	Для производимого тега проходит интервал RPI. Производимый тег передает старые данные потребляющему контроллеру.
2	Производящий контроллер начинает обновлять значения производимого тега.
3	Для производимого тега вновь проходит интервал RPI. Производимый тег передает и старые и новые данные потребляющему контроллеру.
4	Производящий контроллер заканчивает обновление значений производимого тега. Производящий контроллер выполняет инструкцию Immediate Output (IOT) (Немедленные выход). Производимый тег немедленно передает все новые данные потребляющему контроллеру.
5	По получении всех данных потребляющий контроллер выполняет событийную задачу.

Хотя производящий контроллер выполняет инструкцию IOT сразу же после загрузки в него новых данных, событийная задача (в потребляющем контроллере) будет запущена лишь тогда, когда потребляющий контроллер получит все новые данные. Это обеспечивает работу контроллера с полным пакетом новых данных.

Синхронизация нескольких контроллеров

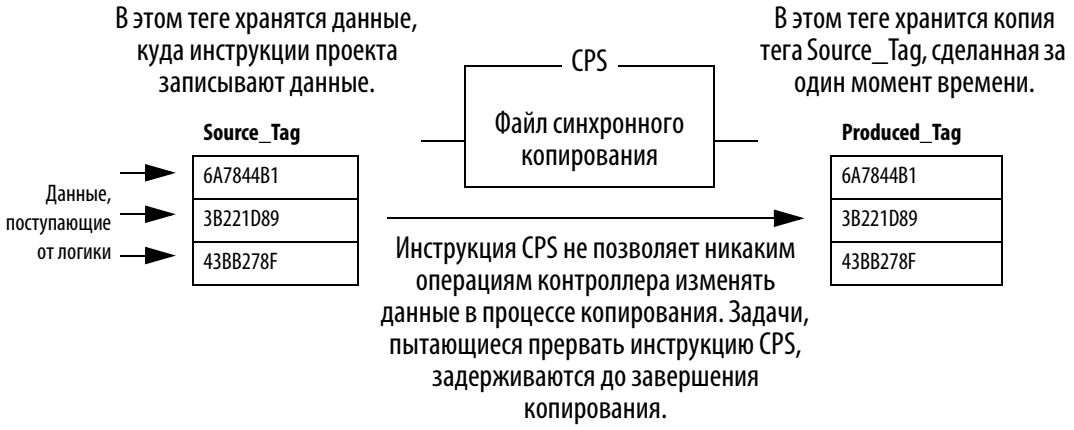
Взаимоотношение между производимым и потребляемым тегом можно также использовать и для синхронизации контроллеров. В этом случае производимый/потребляемый тег служит лишь в качестве пускового механизма.



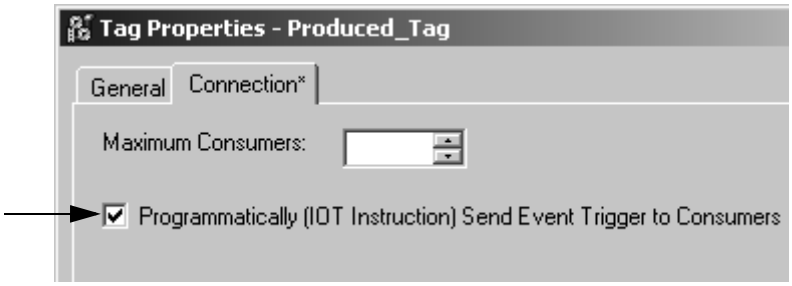
Описание:	
1	Первый контроллер выполняет действие, с которым должны синхронизироваться другие контроллеры.
2	По завершении выполнения этого действия контроллер выполняет инструкцию IOT. Эта инструкция использует производимый тег в качестве своего адресата.
3	По получении производимого тега контроллер A выполняет свою событийную задачу.
4	По получении производимого тега контроллер B выполняет свою событийную задачу.

Контрольный перечень для производящего контроллера

Для следующего элемента:	Проверьте следующее:
☐ 1. Буфер данных	Если вы хотите за один момент времени отправить полный образ данных, создайте копию этих данных, как показано ниже:



☐ 2. Свойства производимого тега	В свойствах соединения (Connection) производимого тега установите следующий флажок:

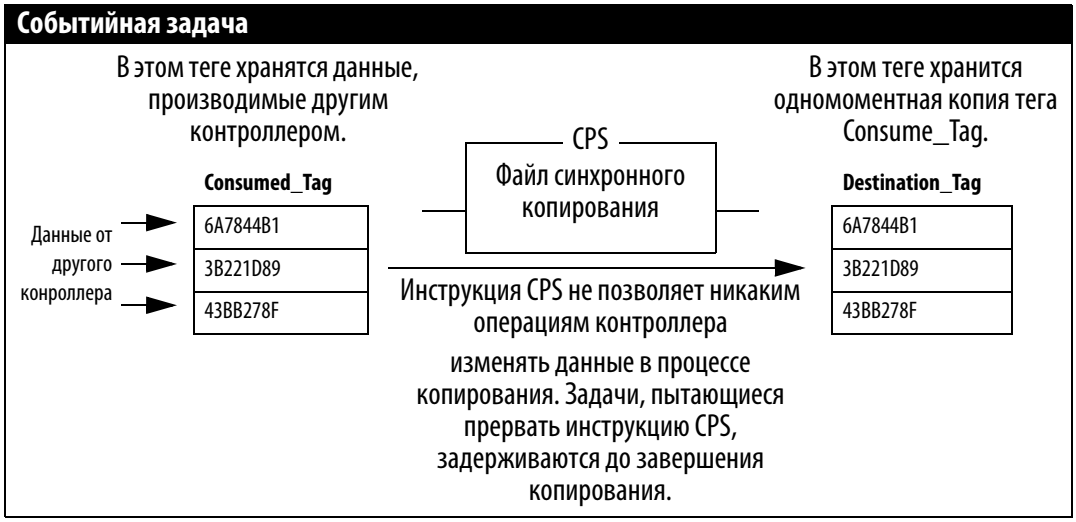


Если этот флажок не будет установлен, производящий контроллер будет запускать событийную задачу в конце всякой задачи, автоматически обновляющей локальные выходы. Иными словами, сканирование задачи будет запускать событие вместе с инструкцией IOT.

☐ 3. Инструкция IOT	Используйте инструкцию IOT в той точке логики, где хотите запускать событийную задачу. Инструкция IOT отменяет RPI для данного тега и осуществляет немедленную передачу триггера событийной задачи и данных тега.

Контрольный перечень для потребляющего контроллера

Для этого параметра:	Обеспечьте выполнение следующих требований:
☐ 1. Буфер данных	Чтобы убедиться, что контроллер не использует данные потребляемого тега в процессе изменения этих данных, используйте копию потребляемого тега. Для копирования данных используйте событийную задачу, как показано ниже:



☐ 2. Приоритет задачи	Установите для событийной задачи самый высокий приоритет. Если какая-либо периодическая задача имеет более высокий приоритет, событийной задаче придется ожидать завершения выполнения такой периодической задачи.
☐ 3. Количество событийных задач	Следует ограничить количество событийных задач. Каждая дополнительная задача уменьшает время обработки, отводимое другим задачам. Это может привести к наложению.
☐ 4. Автоматическая обработка выходных данных	Для событийной задачи вы, как правило, можете отключить автоматическую обработку вывода (настройка по умолчанию). Это уменьшает время выполнения задачи. Для проверки решения обратитесь к рисунку на странице 19.

ПРИМЕР

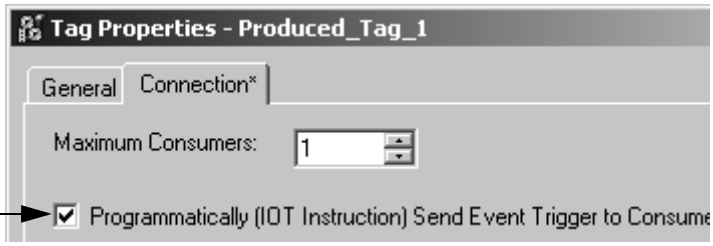
При продвижении детали по производственной линии каждой станции требуются технические условия на данную деталь, относящиеся к конкретной станции. Чтобы исключить возможность действий станции на основе старых данных, событийная задача оповещает о поступлении новых данных для очередной детали.

Производящий контроллер

Этот контроллер управляет станцией 24 и производит данные для следующей станции (станция 25). Чтобы оповестить о передаче новых данных, контроллер использует следующие элементы:

Свойства производимого тега

Производимый тег (Produced_Tag) настроен на обновление своего триггера событийной задачи посредством инструкции IOT.

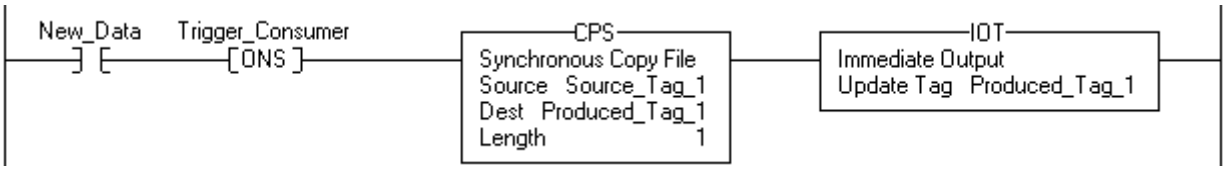


Релейная логика

Если New_Data = on, то для одного сканирования происходит следующее:

Инструкция CPS устанавливает производимый тег, равный тегу-источнику: Produced_Tag_1= Source_Tag_1.

Инструкция IOT обновляет производимый тег Produced_Tag_1 и отправляет обновленные данные в потребляющий контроллер (станция 25). По получении потребляющим контроллером этого обновления он запускает соответствующую событийную задачу.



Продолжение на следующей странице

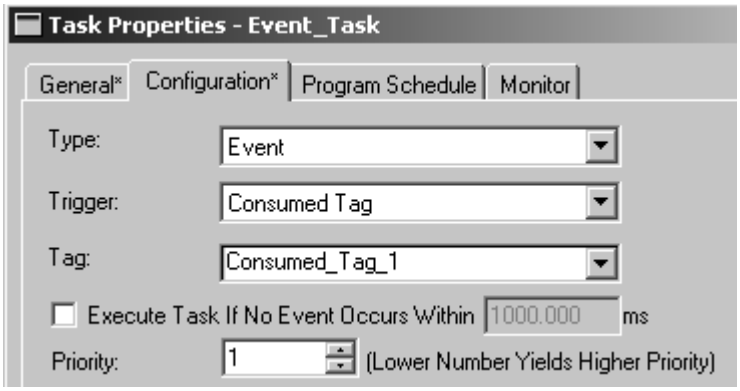
Потребляющий контроллер Контроллер на станции 25 использует данные, произведенные станцией 24. Для определения поступления новых данных этот контроллер использует событийную задачу.

Свойства событийной задачи

Эта задача запускается событием. →

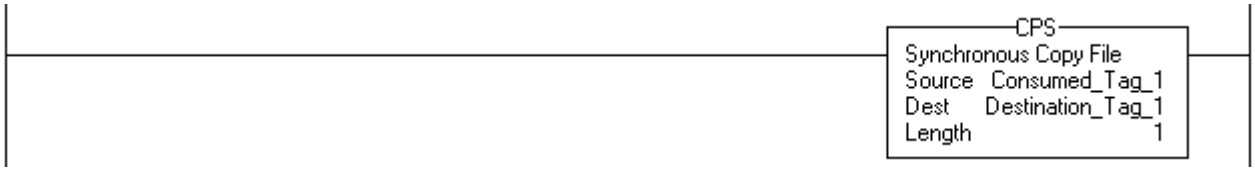
Эта задача запускается потребляемым тегом. →

Эта задача запускается этим потребляемым тегом. →



Релейная блок-схема в событийной задаче

При выполнении событийной задачи инструкция CPS устанавливает принимающий тег, равный потребляемому тегу: Destination_Tag_1= Consumed_Tag_1 (значения, полученные от производящего контроллера). Остальная логика в этом контроллере использует значения из тега Destination_Tag_1.



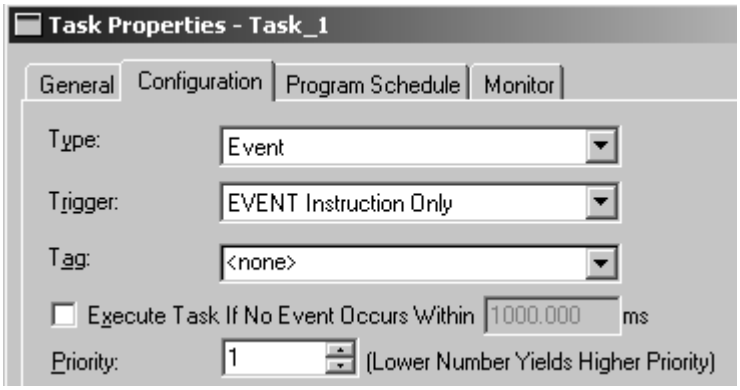
Использование триггера инструкции по событию

Чтобы запустить событийную задачу на основе условий вашей логики, используйте триггер EVENT Instruction Only (Только инструкция по событию).

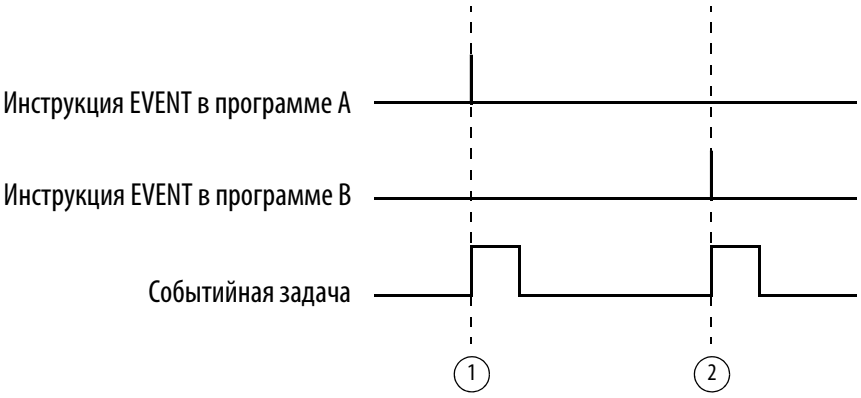
Эта задача запускается событием.

Эта задача запускается инструкцией по событию.

Никакого тега не требуется.



Триггер EVENT Instruction Only (Только инструкция по событию) требует использования инструкции Trigger Event Task (EVENT) (Триггера событийной задачи) для запуска задачи. Можно использовать инструкцию по событию в нескольких точках вашего проекта. При каждом выполнении этой инструкции она будет запускать указанную событийную задачу.



Описание:	
1	Программа A выполняет инструкцию по событию. Событийная задача, заданная данной инструкцией по событию, выполняется один раз.
2	Программа B выполняет инструкцию по событию. Событийная задача, заданная данной инструкцией по событию, выполняется один раз.

Программное определение того, запустила ли инструкция по событию задачу

Чтобы определить, запустила ли инструкция по событию событийную задачу, используйте инструкцию Get System Value (GSV) (Получить значение системы) для проверки параметра Status («Состояние») соответствующей задачи.

Параметр Status объекта TASK (Задачи)

Параметр:	Тип данных:	Инструкция:	Описание:	
Status	DINT	GSV SSV	Предоставляет информацию о состоянии задачи. После установки бита контроллером требуется вручную сбросить бит, чтобы обнаружить, не появилась ли другая ошибка такого типа.	
			Чтобы определить, имеет ли место следующее:	**Проверьте этот бит:**
			Инструкция по событию запустила задачу (только для событийной задачи).	0
			Превышение времени ожидания запустило задачу (только для событийной задачи).	1
			Для данной задачи произошло наложение.	2

Контроллер не сбрасывает установленные биты параметра Status («Состояние»).

- Чтобы использовать бит для новой информации о состоянии, необходимо вручную сбросить соответствующий бит.
- Используйте инструкцию Set System Value (SSV) (Установить значение системы) для установки другого значения этого параметра.

Контрольный перечень для задачи, запускаемой инструкцией по событию

Для этого параметра:	Обеспечьте выполнение следующих требований:
☐ 1. Инструкция по событию (EVENT).	Используйте инструкцию Trigger Event Task (EVENT) (Запустить событийную задачу) в каждой точке логики, где нужно запускать событийную задачу.
☐ 2. Приоритет задачи	Установите для событийной задачи самый высокий приоритет. Если какая-либо периодическая задача имеет более высокий приоритет, событийной задаче придется ожидать завершения выполнения такой периодической задачи.
☐ 3. Количество событийных задач	Следует ограничить количество событийных задач. Каждая дополнительная задача уменьшает время обработки, отводимое другим задачам. Это может привести к наложению.
☐ 4. Автоматическая обработка выходных данных	Для событийной задачи вы, как правило, можете отключить автоматическую обработку вывода (настройка по умолчанию). Это уменьшает время выполнения задачи.

ПРИМЕР

Контроллер использует несколько разных программ, но общую процедуру останова. Каждая из программ использует тег с именем Shut_Down_Line (Остановить линию) в области видимости программы, который включается при обнаружении программой условия, требующего останова.

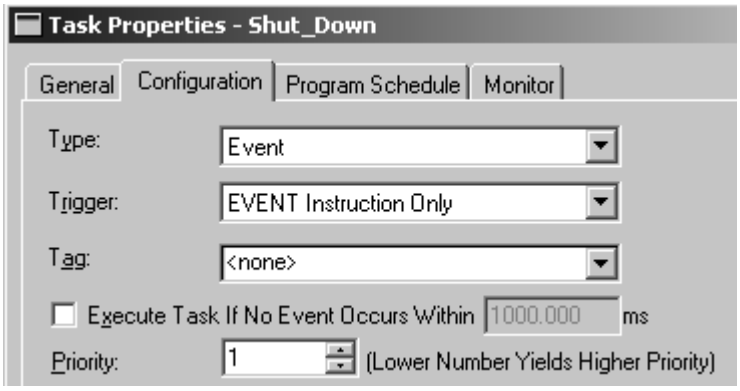
Свойства событийной задачи

Эта задача запускается событием.

Эта задача запускается инструкцией по событию.

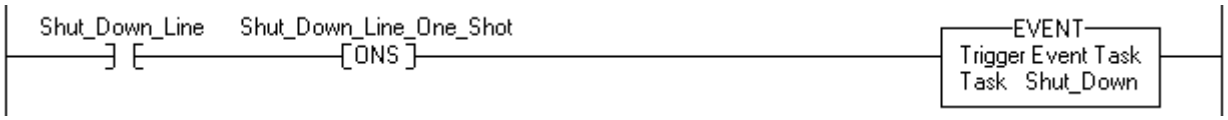
Никакого тега не требуется.

Все остальные задачи прерываются.



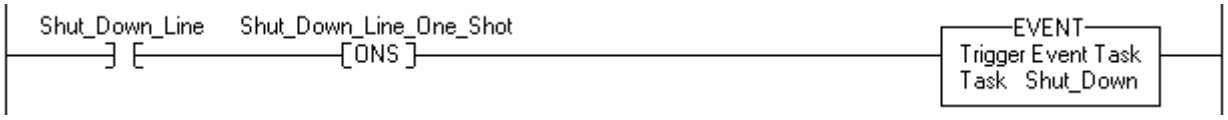
Релейная логика в программе A (Program_A)

Если Shut_Down_Line = on (условия требуют останова), то задача Shut_Down (Остановить) выполняется один раз.



Релейная логика в программе B

Если Shut_Down_Line = on (условия требуют останова), то задача Shut_Down (Остановить) выполняется один раз.

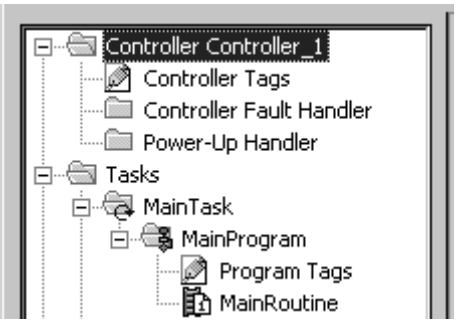


Создание задачи

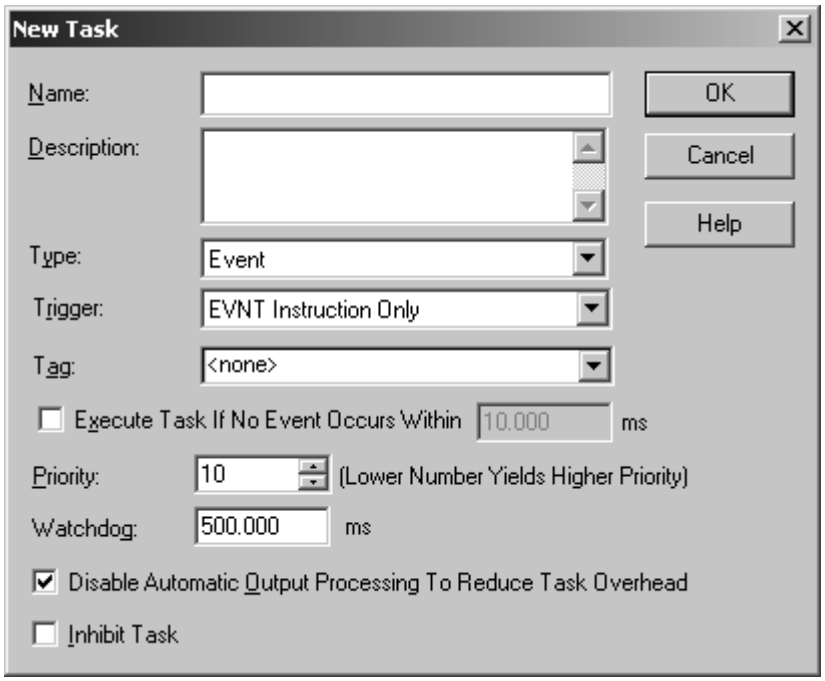
Создание событийной задачи

Чтобы создать событийную задачу, следует:

1. В органайзере контроллера щелкнуть правой кнопкой мыши по папке Tasks (Задачи) и выбрать New Task (Новая задача).



2. Ввести имя задачи.
3. В типе задаче выбрать Event (Событие), выбрать триггер задачи, выбрать тег данных запуска задачи, выбрать приоритет задачи и время таймера схемы безопасности.



4. Щелкнуть OK.

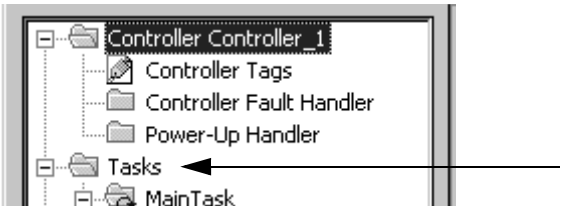
Создание периодической задачи

Периодическая задача выполняет определенную функцию с заданным темпом.

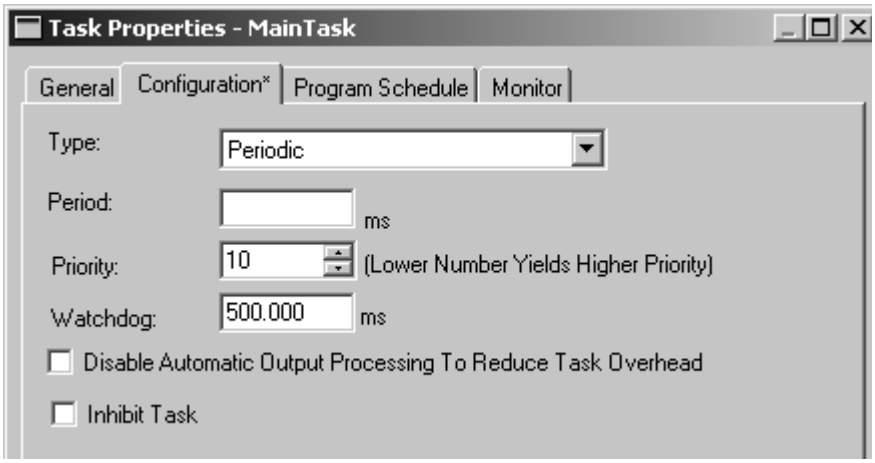
ВАЖНО Временной период задачи должен быть больше суммы времен выполнения всех программ этой задачи.

- Если контроллер обнаружит появление триггера периодической задачи для задачи, которая уже работает, то возникнет неосновная ошибка (наложение).
- Приоритеты и значения времени выполнения других задач также могут приводить к наложению.

1. В органайзере контроллера щелкните правой кнопкой мыши по папке Tasks (Задачи) и выберите New Task (Новая задача).



2. Введите имя задачи, выберите тип задачи Periodic (Периодическая) (установка по умолчанию), введите период выполнения задачи, введите приоритет задачи и введите время таймера схемы безопасности. .



3. Щелкните OK.

Переключение языков

В версии17 программного обеспечения RSLogix 5000 возможно выводить на экран документацию проекта, например, описания тега и комментарии к цепочкам на любом из поддерживаемых языков перевода. Можно 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.

Введите 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.

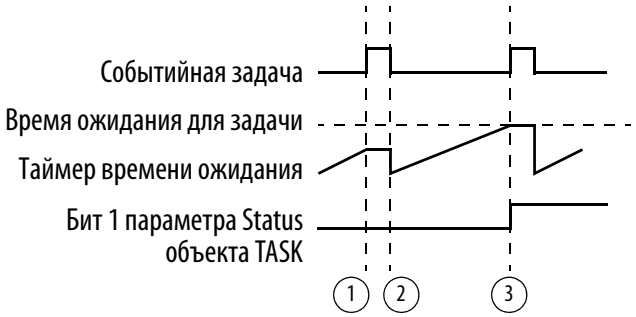
Документация по проекту, для которой поддерживаются несколько переводов в рамках одного проекта, включает:

- Описания компонентов тегов, процедур, программ, пользовательских типов данных и создаваемых пользователем инструкций (Add-On Instructions).
- Фазы оборудования.
- Тренды.
- Контроллеры.
- Сообщения о тревогах (в конфигурации ALARM_ANALOG и ALARM_DIGITAL).
- Задачи.
- Описание свойств модулей в органайзере контроллера.
- Комментарии к цепочкам, текстовые окна ПФС и ФБС.

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

Определение времени ожидания для событийной задачи

Если вы хотите, чтобы событийная задача автоматически выполнялась, если соответствующий триггер не появится в течение определенного времени, то установите для этой задачи время ожидания. После выполнения событийной задачи значение ее таймера ожидания начинает дискретно увеличиваться. Если таймер достигает заданного для него значения до запуска соответствующей событийной задачи, эта событийная задача выполняется автоматически.

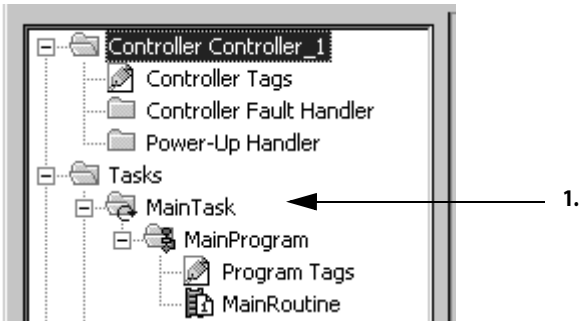


	Описание
1	Выполняется событийная задача. Время ожидания прекращает увеличиваться.
2	Событийная задача завершена. Таймер времени ожидания сбрасывается и начинает увеличиваться.
3	Таймер времени ожидания достигает значения времени ожидания. Событийная задача автоматически начинает выполнение. Устанавливается бит 1 в параметре Status («Состояние») объекта TASK.

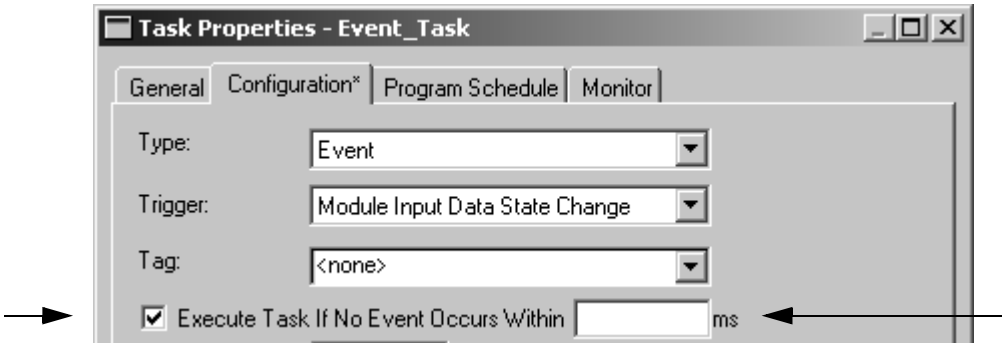
Определение значения времени ожидания для событийной задачи

Чтобы установить значения времени ожидания для событийной задачи:

1. В органайзере контроллера щелкните правой кнопкой мыши по событийной задаче и выберите Properties (Свойства).



2. Щелкните по закладке Configuration (Конфигурация).



3. Установите флажок Execute Task If No Event Occurs Within (Выполнять задачу, если событие не возникает в течение) и введите значение времени ожидания в миллисекундах.

4. Нажмите OK..

Конфигурирование времени ожидания программным путем

Чтобы программно сконфигурировать время ожидания, используйте инструкцию Get System Value (GSV) (Получить значение системы) для обращения к параметрам задачи.

Параметр Status объекта TASK (Задачи)

Параметр	Тип данных	Инструкция	Описание	
Rate	DINT	GSV SSV	**Если тип задачи:**	**То параметр Rate определяет:**
			Периодическая	Период для данной задачи. Время указывается в микросекундах.
			Событийная	Значение времени ожидания для данной задачи. Время указывается в микросекундах.
EnableTimeOut	DINT	GSV SSV	Запрещает или разрешает функцию времени ожидания для событийной задачи.	
			Чтобы:	**Установите значение этого параметра:**
			Запретить функцию времени ожидания.	0 (по умолчанию)
			Разрешить функцию времени ожидания.	1 (или любое значение, отличное от нуля)

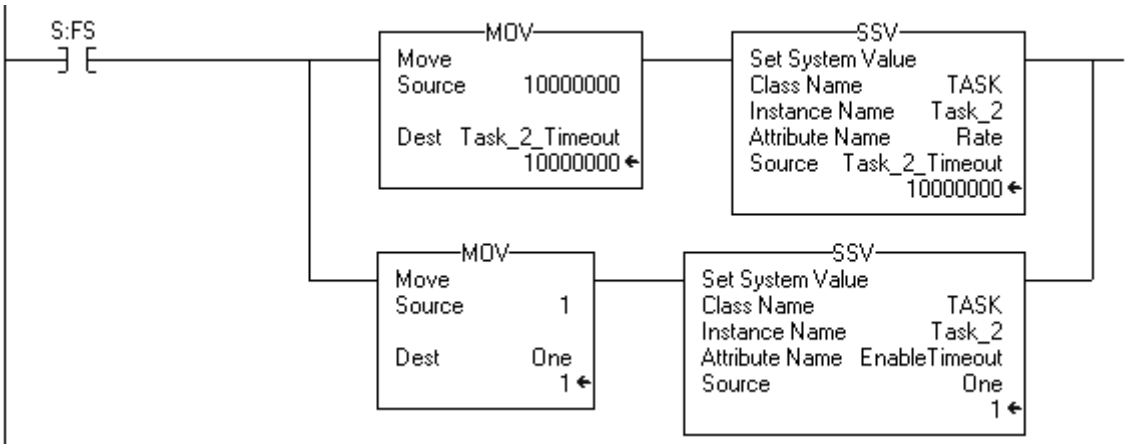
ПРИМЕР

Конфигурирование времени ожидания программным путем

Чтобы значение времени ожидания всегда было определено и разрешено для событийной задачи, следующая логика конфигурирует время ожидания при переходе контроллера в режим выполнения.

Если S:FS = 1 (первое сканирование), то устанавливается значение времени ожидания для задачи Task_2 и разрешается функция времени ожидания:

1. Первая инструкция MOV устанавливает время ожидания задачи Task_2_Timeout = 10000000 мкс (значение DINT). Затем инструкция SSV устанавливает параметр Rate для задачи Task_2 = Task_2_Timeout. Это устанавливает значение время ожидания для данной задачи.
2. Вторая инструкция MOV устанавливает One = 1 (значение DINT). Затем инструкция SSV устанавливает параметр EnableTimeout для задачи Task_2 = One. Это разрешает функцию времени ожидания для данной задачи.



Программное определение превышения времени ожидания

Чтобы определить, выполняется ли событийная задача вследствие превышения времени ожидания, используйте инструкцию Get System Value (GSV) (Получить значение системы) для проверки параметра Status данной задачи.

Параметр Status объекта TASK (Задачи)

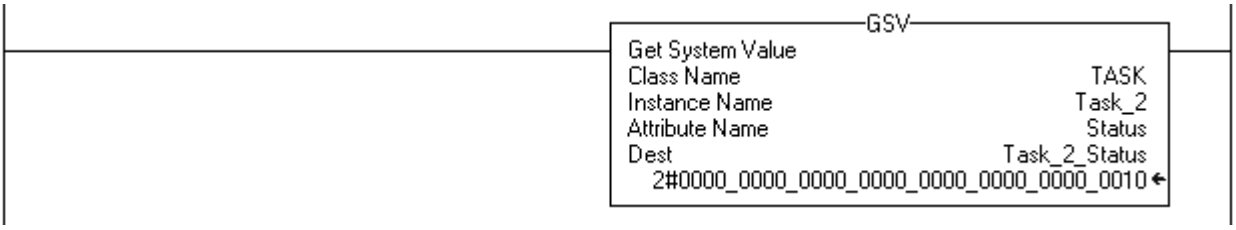
Параметр	Тип данных	Инструкция	Описание	
Status	DINT	GSV SSV	Предоставляет информацию о состоянии задачи. После установки бита контроллером требуется вручную сбросить бит, чтобы обнаружить, не появилась ли другая ошибка такого типа.	
			Чтобы определить, имеет ли место следующее:	**Проверьте этот бит:**
			Инструкция по событию запустила задачу (только для событийной задачи)	0
			Превышение времени ожидания запустило задачу (только для событийной задачи).	1
			Для данной задачи произошло наложение.	2

ПРИМЕР

Определение времени ожидания для событийной задачи

Если для событийной задачи срабатывает таймер времени ожидания, это может означать нарушение связи с запускающим устройством. Это требует остановки процесса. Для остановки работы контроллера событийная задача вызывает процедуру ошибки для соответствующей программы и передает заданный пользователем код ошибки (в этом примере 999).

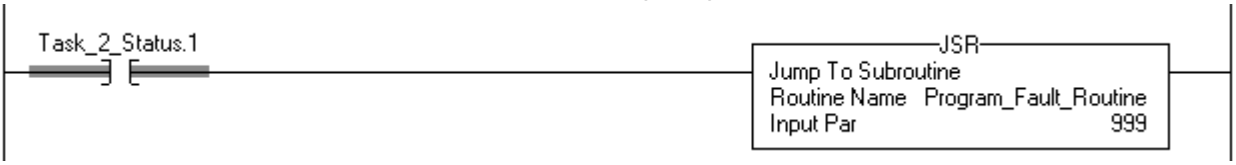
1. Инструкция GSV устанавливает параметр Task_2_Status =Status («Состояние») для задачи Task_2 (значение DINT).



2. Если Task_2_Status.1 = 1, то произошел срабатывание таймера времени ожидания, поэтому контроллер останавливается и код основной ошибки устанавливается на 999:

   Инструкция JSR вызывает процедуру ошибки для данной программы. Это приводит к основной ошибке.

   Код основной ошибки = 999 (значение входного параметра 999).



3. Если Condition_1 = 1, то необходимо сбросить биты параметра Status («Состояние») для задачи Task_2:

   Инструкция SSV устанавливает параметр Status задачи Task_2 = Zero («Ноль»). Zero («Ноль») – это тег DINT, имеющий значение 0.

Настройка кванта времени на системные операции

Контроллер Logix5000 связывается с другими устройствами (модулями ввода/вывода, контролерами, терминалами HMI и т.д.) либо с заданным интервалом (запланированный обмен данными), либо, когда доступно время обработки для выполнения связи (незапланированный обмен данными).

Данный тип связи:	Представляет собой:
Обновление данных ввода/вывода (не включая поблочную передачу).	Запланированный обмен данными
Производство или потребление тегов.	
Связь с программирующими устройствами (например, программное обеспечение RSLogix 5000).	Незапланированный обмен данными
Связь с устройствами HMI.	
Выполнение инструкций Message (MSG) (Сообщения), включая поблочную передачу.	
Отклик на сообщения других контроллеров.	
Синхронизация вторичного контроллера системы с резервированием.	
Восстановление и контроль соединений ввода/вывода (например, при удалении и установке модуля ввода/вывода при подключенном питании); сюда не входит обычное обновление данных ввода/вывода, осуществляемое при выполнении программы.	
Связь с использованием моста между последовательным портом контроллера и другими устройствами ControlLogix через заднюю шину ControlLogix.	

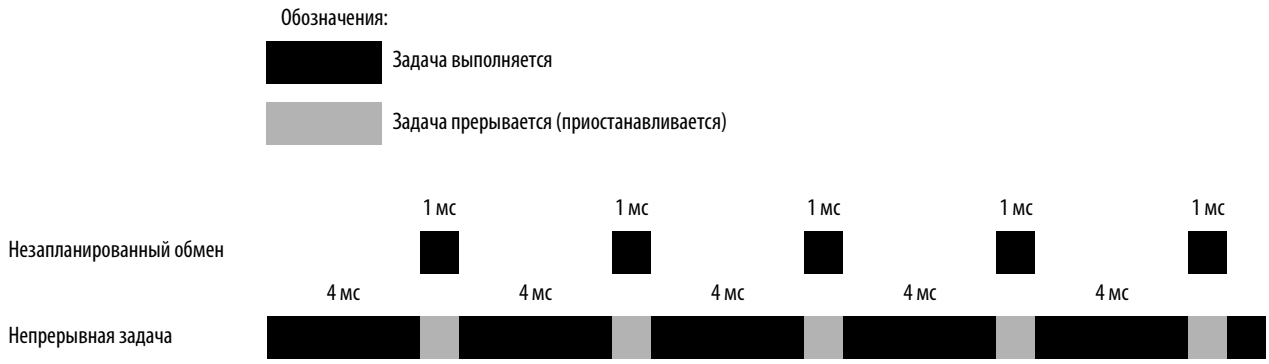
Незапланированный обмен данными – это всякий обмен данными, который вы не настраиваете посредством папки конфигурации ввода/вывода проекта.

- Значение кванта времени на системные операции определяет процент времени контроллера (за исключением времени на выполнение периодических задач или событийных задач), которое выделяется контроллером для незапланированного обмена данными.
- Контроллер выполняет незапланированный обмен данными в течение максимум 1 мс, после чего возобновляет выполнение непрерывной задачи.

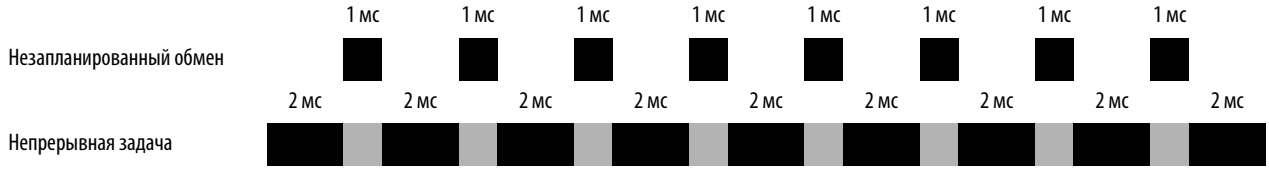
Нижеприведенная таблица показывает отношение времени выполнения непрерывной задачи и времени незапланированного обмена данными в связи со служебными системными операциями.

При таком кванте времени:	Непрерывная задача выполняется в течение:	А системные операции выполняются максимум:
10%	9 мс	1 мс
20%	4 мс	1 мс
33%	2 мс	1 мс
50%	1 мс	1 мс

При кванте времени до 20% незапланированные служебные системные операции прерывают выполнение непрерывной задачи на 1 мс каждые 4 мс..

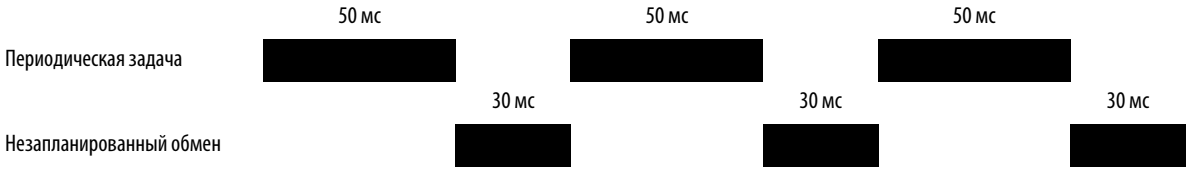


Если увеличить квант времени служебных системных операций до 33 %, незапланированный обмен прерывает выполнение непрерывной задачи на 1 мс каждые 2 мс.



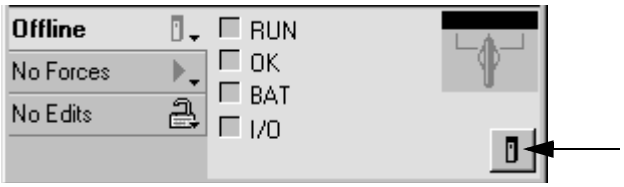
Если контроллер содержит только периодическую задачу или задачи, то значение кванта времени на служебные системные операции не будет оказывать никакого влияния. Незапланированный обмен данными происходит лишь в то время, когда периодическая задача не работает.

Например, если время выполнения задачи составляет 50 мс, причем вы настраиваете частоту обновления, равную 80 мс, контроллер имеет в запасе 30 мс из каждых 80 мс на незапланированный обмен.

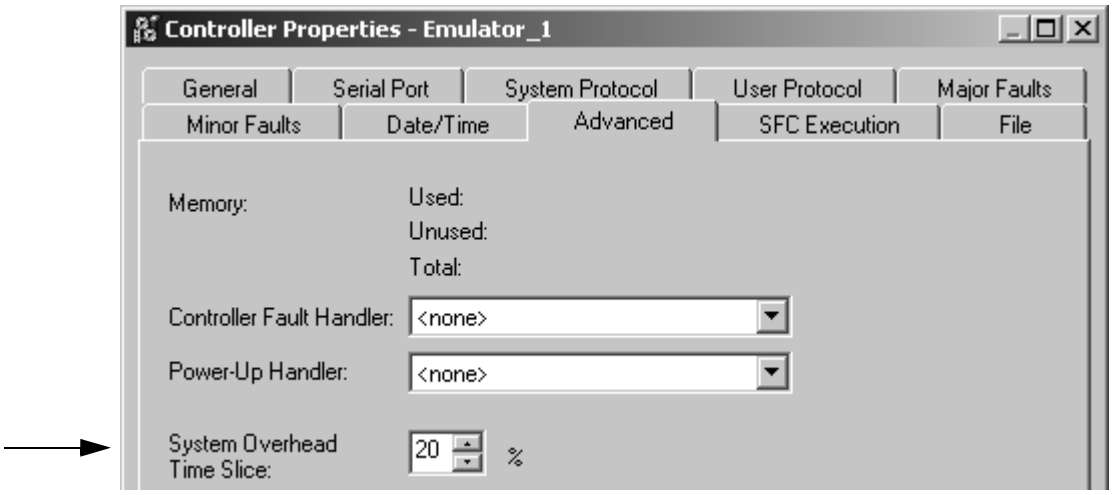


Настройка кванта времени на системные операции

1. На интерактивной панели инструментов щелкните кнопку Properties (Свойства).



2. Выберите вкладку Advanced (Дополнительно) и введите значение кванта времени на служебные системные операции.



3. Нажмите OK.

Настройка времени сторожевого таймера

В каждой задаче имеется таймер схемы безопасности (сторожевой таймер), который определяет, как долго может выполняться задача до появления основной ошибки.

ВНИМАНИЕ

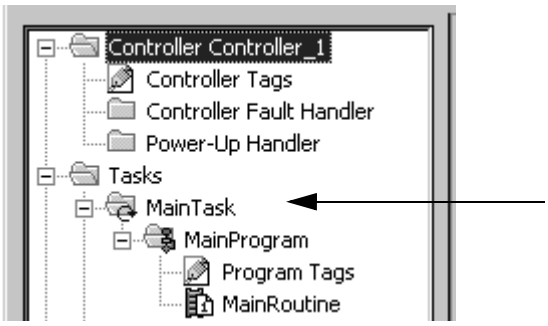
При достижении сторожевым таймером установленного значения времени возникает основная ошибка. В зависимости от программы обработки ошибок контроллера может произойти остановка контроллера.

- Диапазон значений таймера схемы безопасности составляет от 1 мс до 2 000 000 мс (2000 секунд). Значение по умолчанию – 500 мс.
- Таймер схемы безопасности начинает отсчет времени при запуске задачи и останавливается, когда будут выполнены все программы в данной задаче.
- Если время выполнения задачи превышает время таймера схемы безопасности, возникает основная ошибка. (Время включает прерывания другими задачами).
- Ошибка по превышению контрольного времени (основная ошибка) также возникает при повторном запуске задачи в процессе ее выполнения (наложение задач). Это может произойти в том случае, если задача с более низким приоритетом прерывается более высокоприоритетной задачей, что задерживает выполнение задачи с более низким приоритетом.
- Вы можете сбросить ошибку превышения контрольного времени при помощи программы обработки ошибок контроллера. Если при сканировании той же самой логики вновь возникнет такая же ошибка превышения контрольного времени, контроллер перейдет в режим ошибки независимо от того, сбросила ли программа обработки ошибок контроллера эту ошибку превышения контрольного времени.

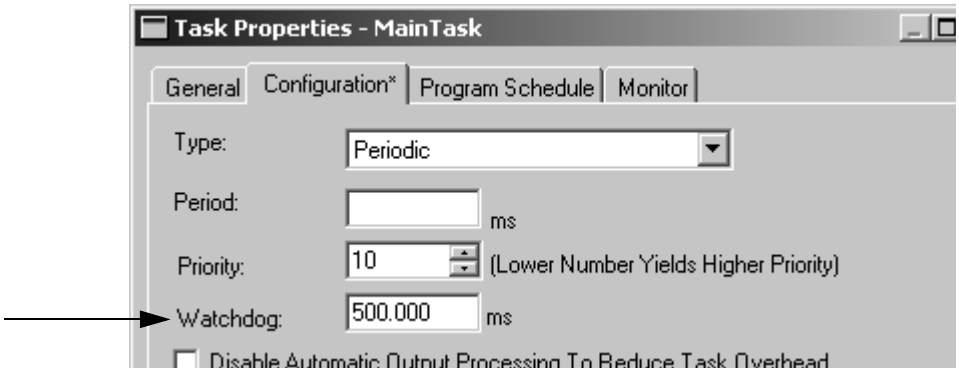
Настройка времени сторожевого таймера для задачи

Чтобы изменить время таймера схемы безопасности для задачи, обратитесь к диалоговому окну Properties (Свойства) задачи.

1. В органайзере контроллера щелкните правой кнопкой мыши по соответствующей задаче и выберите Properties (Свойства).



2. Выберите закладку Configuration (Конфигурация) и выберите контрольное время таймера схемы безопасности задачи в микросекундах.



3. Нажмите OK.

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-PM005B-RU-P - Июль 2008

Замещает публикацию 1756-PM005A-EN-P - Июль 2007

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