

SIEMENS

SIMATIC

Загружаемый драйвер для PtP-соединения с протоколом MODBUS RTU-формата для модулей CP (в случае, когда S7 - ведущее устройство)

Руководство

Данное руководство является
частью пакета документации
с заказным номером:

6ES7870-1AA00-0YA0

Предисловие, содержание

Описание 1

Инсталляция 2

Настройка драйвера 3

Протокол передачи данных 4

Функциональные коды 5

Интерфейс CPU - CP 6

Диагностика драйвера 7

Пример использования 8

Приложения

Технические данные A

Схемы многоточечного соединения B

Список литературы C

Глоссарий,
предметный указатель

Указания по технике безопасности

Данное руководство содержит указания, которые вы должны соблюдать для обеспечения собственной безопасности, а также защиты от повреждений оборудования. Эти замечания выделены в руководстве символами, как показано ниже, в соответствии с уровнем опасности:



Опасность

указывает, что несоблюдение надлежащих мер предосторожности может привести к гибели людей, тяжким телесным повреждениям или существенному имущественному ущербу.



Предупреждение

указывает, что несоблюдение надлежащих мер предосторожности может привести к гибели людей, тяжким телесным повреждениям или к существенному имущественному ущербу.



Предостережение

указывает, что несоблюдение надлежащих мер предосторожности может привести к легким телесным повреждениям и нанесению незначительного имущественного ущерба.

Предостережение

указывает, что несоблюдение надлежащих мер предосторожности может привести к нанесению незначительного имущественного ущерба.

Замечание

привлекает ваше внимание к особо важной информации о продукте, обращении с ним, или к соответствующей части документации.

Квалифицированный персонал

К монтажу и работе на этом оборудовании должен допускаться только квалифицированный персонал. Квалифицированный персонал – это люди которые, имеют право вводить в эксплуатацию, заземлять и маркировать электрические цепи, оборудование и системы в соответствии с установленной практикой и стандартами техники безопасности.

Надлежащее использование

Примите во внимание следующее:



Предупреждение

Это устройство и его компоненты могут использоваться только для применений, описанных в каталоге или технической документации, и совместно только с теми устройствами или компонентами других производителей, которые были одобрены или рекомендованы фирмой Siemens.

Этот продукт может правильно и надежно функционировать только в том случае, если он правильно транспортируется, хранится, устанавливается и монтируется, а также если эксплуатируется и обслуживается в соответствии с рекомендациями.

Товарные знаки

SIMATIC®, SIMATIC HMI® and SIMATIC NET® - это зарегистрированные товарные знаки SIEMENS AG.

Некоторые другие обозначения, использованные в этих документах, также являются зарегистрированными товарными знаками; если они используются третьей стороной для своих собственных целей, то соответствующие права собственности могут быть нарушены.

Copyright © Siemens AG 2003 Все права защищены

Воспроизведение, передача или использование этого документа или его содержания не допускаются без письменного разрешения правообладателя. Нарушители будут нести ответственность за нанесенный ущерб. Все права, включая права на патенты, практические модели и конструкции, защищены.

Отказ от ответственности

Мы проверили содержание этого руководства на соответствие с описанным аппаратным и программным обеспечением. Так как отклонения не могут быть полностью исключены, то мы не можем гарантировать полного соответствия. Однако данные, приведенные в этом руководстве, регулярно пересматриваются и все необходимые исправления вносятся в последующие издания. Мы будем благодарны за предложения по улучшению содержания.

Содержание

1	Описание	1-1
1.1	Возможности приложения	1-1
1.2	Требования к программному и аппаратному обеспечению	1-3
1.3	Резюме по протоколу GOULD-MODBUS	1-4
2	Инсталляция	2-1
2.1	Использование защитного ключа-заглушки	2-1
2.2	Подключение к интерфейсу	2-1
3	Настройка драйвера	3-1
3.1	Инсталляция драйвера в STEP 7 в программаторе / в ПК	3-1
3.2	Деинсталляция драйвера	3-2
3.3	Конфигурирование данных коммуникаций	3-2
3.3.1	Конфигурирование данных коммуникаций для CP 341	3-2
3.3.2	Конфигурирование данных коммуникаций для CP 441-2	3-3
3.4	Назначение параметров для CP	3-4
3.4.1	Назначение параметров для CP 341	3-4
3.4.2	Назначение параметров для CP 441-2	3-5
3.5	Конфигурирование данных коммуникаций проекта	3-7
3.6	Назначение параметров для загружаемого драйвера	3-9
3.6.1	Протокол для ведущего устройства Modbus	3-10
3.6.2	Интерфейс X27 (RS 422/485)	3-13
3.7	Загрузка данных конфигурации и параметризации для CP 341	3-15
3.8	Загрузка данных конфигурации и параметризации для CP 441-2	3-16
3.9	Характеристики запуска CP	3-17
3.10	Назначение параметров запуска ("Startup of the CPU")	3-18

4	Протокол передачи	4-1
5	Функциональные коды	5-1
5.1	Функциональный код 01 - Read Output Status (считывание состояния выхода)	5-1
5.2	Функциональный код 02 - Read Input Status (считывание состояния входа)	5-3
5.3	Функциональный код 03 - Read Output Registers (считывание выходных регистров)	5-5
5.4	Функциональный код 04 - Read Input Registers (считывание входных регистров)	5-7
5.5	Функциональный код 05 - Force Single Coil (назначение для одной катушки)	5-9
5.6	Функциональный код 06 - Preset Single Register (предустановки для одного регистра)	5-10
5.7	Функциональный код 07 - Read Exception Status (считывание ошибочного состояния)	5-11
5.8	Функциональный код 08 - Loop Back Diagnostic Test (тестирование коммуникационного соединения)	5-12
5.9	Функциональный код 11 - Fetch Communications Event Counter (считывание значения счетчика коммуникационных событий)	5-13
5.10	Функциональный код 12 - Fetch Communications Event Log (считывание протокола коммуникационных событий)	5-14
5.11	Функциональный код 15 - Force Multiple Coils (назначения для нескольких катушек)	5-16
5.12	Функциональный код 16 - Preset Multiple Registers (предустановки для нескольких регистров)	5-18
6	Интерфейс CPU - CP	6-1
6.1	Интерфейс CPU - CP для модуля CP 341	6-1
6.1.1	Передача данных от CPU к CP посредством P_SND_RK (CP 341)	6-2
6.1.2	Передача данных от CP к CPU посредством P_RCV_RK (CP 341)	6-4
6.2	Интерфейс CPU - CP для модуля CP 441-2	6-5
6.2.1	Передача данных от CPU к CP посредством BSEND (CP 441-2)	6-5
6.2.2	Передача данных от CP к CPU посредством BRCV (CP 441-2)	6-8
7	Диагностика драйвера	7-1
7.1	Средства диагностики модуля CP 341	7-2
7.1.1	Диагностика с использованием элементов индикации CP 341	7-2
7.1.2	Диагностические сообщения функциональных блоков CP 341	7-3
7.2	Функции диагностики CP 441-2	7-4
7.2.1	Диагностика с использованием элементов индикации CP 441-2	7-4

7.2.2	Диагностические сообщения системных функциональных блоков CP 441-2	7-5
7.2.3	Диагностика посредством области для сообщений об ошибках SYSTAT	7-6
7.3	Таблицы диагностических событий (Events) / ошибок (Errors)	7-8
7.3.1	Номера ошибок в области SYSTAT для "CPU Job Errors" ("Ошибки в работе CPU")	7-8
7.3.2	Номера ошибок в области SYSTAT для "Receive Errors" ("Ошибки при приеме")	7-9
7.3.3	Номера ошибок в области SYSTAT для "General Processing Errors" ("Общие ошибки обработки")	7-11
8	Примеры применения	8-1
8.1	Пример для модуля CP 341	8-1
8.1.1	Используемые блоки	8-1
8.1.2	Описание программы	8-3
8.1.3	Пример программирования	8-4
8.2	Пример для модуля CP 441-2	8-9
8.2.1	Используемые блоки	8-9
8.2.2	Описание программы	8-12
8.2.3	Пример программирования	8-14
A	Технические данные	A-1
B	Схемы многоточечных соединений	B-1
C	Список литературы	C-1
	Словарь терминов	G-1
	Предметный указатель	I-1

Предисловие

Назначение

Информация, которую Вы найдете в данном руководстве, поможет Вам устанавливать и настраивать канал обмена данными между CP и системой управления с протоколом обмена "Modbus".

Рекомендуемый базовый уровень подготовки пользователей

Для того, чтобы материал данного руководства был Вам понятен, Вы должны иметь общие знания по технике автоматизации.

Кроме того, Вы должны уметь обращаться с компьютерами или аналогичными устройствами (например, с программаторами) под управлением операционной системы Windows 95/98/2000/NT или XP. Так как загружаемый драйвер базируется на программном обеспечении STEP 7, то Вы также должны уметь использовать ПО STEP 7. Информацию по данному вопросу Вы можете найти в руководстве "Programming with STEP 7 V5.2" ("Программирование в STEP 7 версии 5.2").

Применимость данного руководства

Данное руководство может быть применимо для следующих пакетов ПО:

ПО	Заказной номер	Начиная с версии
Loadable Driver for Point-to-Point CPs (Загружаемый драйвер для PtP-соединений с использованием CP)	6ES7 870-1AA00-0YA0	3.0

Примечание

Описание драйвера в данном руководстве корректно на момент данной публикации. В описании дается информация об изменениях по сравнению с предыдущими версиями.

Содержание руководства

В этом руководстве описываются функции загружаемого драйвера и то, как его связать с программным обеспечением и аппаратурой коммуникационных процессоров CP 341 и CP 441-2.

Руководство состоит из глав, посвященных инструкциям, и из справочных разделов (приложений).

В руководстве рассматриваются следующие темы:

- Описание ПО / Инсталляция
- Настройка драйвера / Инсталляция / Назначение параметров
- Интерфейс CPU - CP
- Протокол передачи / Функциональные коды
- Диагностика драйвера

Соглашения

В документации используется сокращение CP для обозначения коммуникационных процессоров CP 341 и/или CP 441-2.

Структура данного руководства

Для быстрого поиска нужной информации руководство имеет следующие особенности:

- В основных разделах в левом поле на каждой странице представлено название темы, в котором обобщается содержание соответствующего раздела.
- В последующем за приложениями глоссарии объяснены важные термины, используемые в руководстве.
- Подробный предметный указатель поможет Вам быстро найти нужную информацию, касающуюся отдельных объектов.

Примечание

Драйвер, описанный в данном руководстве, может применяться как загружаемый протокол обмена для CP.
Он может использоваться вместо стандартных протоколов 3964R, RK512, ASCII и протокола обмена для принтера.

Предупреждение

Для описанного здесь драйвера допускаются модификации или расширения последовательностей данных обмена между CP и CPU.

Указанные модификации или расширения могут касаться в частности классов событий или номеров событий в системе диагностики.

Более того, необходимо иметь в виду, что данное руководство описывает только модификации или расширения в сравнении со стандартными функциями. Общая информация по конкретным модулям CP может быть найдена в руководствах по использованию соответствующих модулей CP.

Для корректного использования рассматриваемого драйвера Вы должны обладать подробными сведениями о том, как функционирует выбранный Вами коммуникационный процессор.

Дополнительная помощь

Если у Вас возникли любые технические вопросы, то обратитесь к местному представителю фирмы Siemens. Вы можете также получить нашу поддержку с помощью Интернета, используя следующую ссылку:

<http://www.siemens.com/automation/partner>

Учебные курсы

Мы предоставляем множество курсов, чтобы помочь Вам начать использовать программируемые контроллеры SIMATIC S7. Вы можете поддерживать контакт с местным учебным центром или с центральным учебным центром в Нюрнберге:

Nuremberg, D-90327 Germany, тел. +49 (911) 895-3200.

Интернет: <http://www.sitrain.com>

Обслуживание и поддержка с помощью Интернета

В дополнение к документации, Вы можете также получить нашу поддержку с помощью Интернета используя следующую ссылку:

<http://www.siemens.com/automation/service&support>

Здесь Вы найдете:

- Новую информацию, которая постоянно обновляется, с последними данными по продуктам, которые Вы используете.
- Правовую документацию, необходимую Вам для поиска в службе обслуживания и поддержки с помощью Интернета (Service & Support).
- Форум для обмена информацией и опытом между специалистами и пользователями.
- Контактные реквизиты для ответов на вопросы по автоматизации и приводам (Automation & Drives) в нашей базе данных.
- Информацию по местным отделениям обслуживания, по ремонту и запчастям, а также многое другое под заголовком "Services" ("Обслуживание").

Техническая поддержка A&D (A&D Technical Support)

Круглосуточно, по всему миру:



<p>Всемирная (Нюрнберг) техническая поддержка</p> <p>Мест. вр. : 0:00 - 24:00 / 365 дней Тлф: +49 (0) 180 5050-222 Факс: +49 (0) 180 5050-223 E-mail: adsupport@siemens.com GMT: +1:00</p>		
<p>Европа / Африка (Нюрнберг) Авторизация</p> <p>Мест. вр. : Пнд.-Птн. 8:00 - 17:00 Тлф: +49 (0) 180 5050-222 Факс: +49 (0) 180 5050-223 E-mail: adautorisierung@siemens.com GMT: +1:00</p>	<p>США (Джонсон-Сити) Техническая поддержка и авторизация</p> <p>Мест. вр. : Пнд.-Птн. 8:00 - 17:00 Тлф: +1 (0) 423 262 2522 Факс: +1 (0) 423 262 2289 E-mail: simatic.hotline@sea.siemens.com GMT: -5:00</p>	<p>Азия / Австралия (Пекин) Техническая поддержка и авторизация</p> <p>Мест. вр. : Пнд.-Птн. 8:00 - 17:00 Тлф: +86 10 64 75 75 75 Факс: +86 10 64 74 74 74 E-mail: adsupport.asia@siemens.com GMT: +8:00</p>
<p>Язык общения для технической поддержки и авторизации: английский или немецкий.</p>		

1 Описание

1.1 Функциональные возможности

Положение в системе автоматизированного управления

Драйвер, описанный в данном руководстве, является программным продуктом, предназначенным для работы с коммуникационными процессорами CP 341 (S7-300) и CP 441-2 (S7-400).

Коммуникационные процессоры CP 341 и CP 441-2 могут использоваться в автоматизированных системах для обеспечения обмена данными с партнерскими системами.

Функции драйвера

Указанный драйвер позволяет установить коммуникационное соединение между коммуникационным процессором CP 341 или CP 441-2 и системой управления, поддерживающей Modbus (например, Modicon или Honeywell TDC 3000).

Используемый протокол обмена **GOULD - MODBUS** имеет **RTU**-формат. Обмен данными происходит в соответствии с принципом Master - Slave (Ведущее устройство - Ведомое устройство).

Ведущее устройство (Master в SIMATIC S7) имеет активный статус во время передачи.

Функциональные коды (Function Codes) 01, 02, 03, 04, 05, 06, 07, 08, 11, 12, 15 и 16 могут использоваться для коммуникаций между CP и основной системой.

Пригодные для использования интерфейсы и протоколы

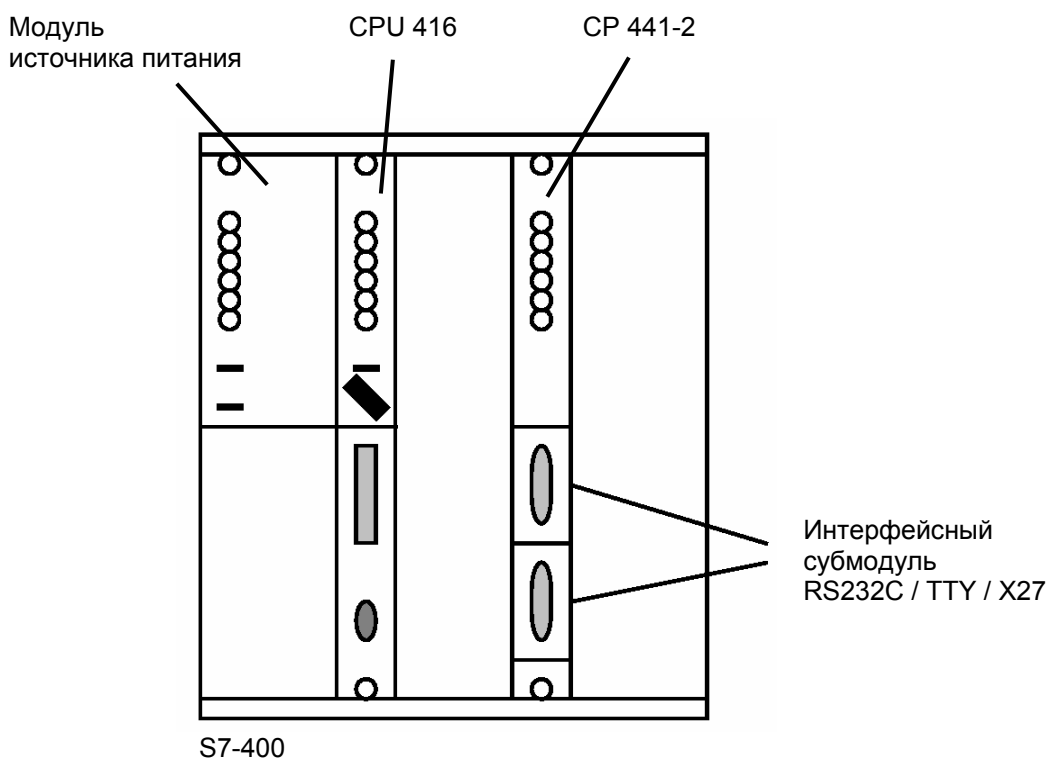
Два последовательных интерфейса коммуникационного процессора CP 441-2 могут работать независимо, используя разные стандартные протоколы или протоколы загружаемых драйверов.

Вы можете использовать интерфейсные submodule RS 232, TTY или RS 422/485 (X27) в качестве интерфейса CP.

С рассматриваемым здесь драйвером возможно использование интерфейсного submodule RS 422/485 (X27) как с двухпроводной, так и с четырехпроводной линией связи. В двухпроводном режиме Вы можете подключить до 32 ведомых (slave) устройств к одному ведущему (master) (полудуплексный режим), тем самым обеспечивая многоточечное соединение (сеть).

Возможная конфигурация системы

На следующем рисунке показана возможная конфигурация системы.



1.2 Требования к программному и аппаратному обеспечению

Используемые модули

Драйвер работает с модулями CP 341 и CP 441-2 с заказными номерами 6ES7 441-2AA02-0AE0 или с более новыми.

Модули CP 441-1 с номером MLFB 6ES7 441-1AA0х-0AE0 и CP 441-2 с номером MLFB 6ES7 441-2AA00-0AE0 или 6ES7 441-2AA01-0AE0 не могут быть использованы с загружаемыми драйверами.

Защитный ключ-заглушка (dongle)

Для использования в CP загружаемых драйверов Вам необходим защитный ключ-заглушка. Ключ-заглушка поставляется с драйвером.

Загрузочная память (Loading Memory) CPU (карта памяти)

Если Вы используете CP 441-2, то загружаемые драйвера загружаются в загрузочную память (loading memory) CPU после назначения параметров и пересылаются в память CP при запуске. Поэтому CPU должен иметь загрузочную память (loading memory) достаточного размера. По этой причине необходимо использовать карту памяти типа RAM или FLASH (заказной номер MLFB 6ES7 952-...).

Для каждого интерфейса CP, для которого назначается данный загружаемый драйвер, требуется загрузочная память в CPU в размере около 25 кбайтов.

Если Вы используете CP 341, то загружаемые драйвера загружаются непосредственно в CP 341. Для этого Вам не требуется загрузочная память (loading memory) в S7-300 CPU. Однако Вы должны иметь в виду, что это значит, что Вы не сможете менять модуль без программатора.

Требования к программному обеспечению

В системе должно быть установлено ПО STEP 7 Standard версии 4.02.1 или выше.

В системе должно быть установлено ПО *Point-to-Point Communication, Parameter Assignment (CP 341: PtP-соединение, назначение параметров)* (CP PtP Param) версии 4.1 или выше

Структуры данных

Перед конфигурированием в проекте структур S7-данных Вы должны убедиться, что они совместимы с пользовательскими программами ведомых MODBUS-систем (корректность функциональных кодов и Modbus-адресов).

1.3 Резюме по протоколу GOULD-MODBUS

Функциональные коды

Вид обмена данными между MODBUS-системами определяется функциональными кодами (FC).

Обмен данными

Для обеспечения побитного обмена данными могут использоваться следующие функциональные коды (FC):

FC 01 Read coil (output) status (считывание состояния выхода)

FC 02 Read input status (считывание состояния входа)

FC 05 Force single coil (назначение для одной катушки)

FC 15 Force multiple coils (назначения для нескольких катушек)

Для обеспечения порегистрового обмена данными могут использоваться следующие функциональные коды (FC):

FC 03 Read holding registers (считывание выходных регистров)

FC 04 Read input registers (считывание входных регистров)

FC 06 Preset single register (предустановки для одного регистра)

FC 16 Preset multiple registers (предустановки для нескольких регистров)

Области данных

Как правило, отдельные функциональные коды FC работают в соответствии с представленной ниже таблицей:

Функциональный код	Данные	Тип данных		Вид доступа
01, 05, 15	Состояние катушки (coil) (выход)	Бит	Выход	Чтение/Запись
02	Состояние входа	Бит	Вход	Только чтение
03, 06, 16	Выходной регистр	Регистр (16 битов)	Выходной регистр	Чтение/Запись
04	Входной регистр	Регистр (16 битов)	Входной регистр	Только чтение

Представление адресов

Аналогично разделению областей с доступом для чтения/записи и с доступом только для чтения, данные на уровне пользователя могут быть представлены как показано в следующей таблице:

Функция	Тип данных	Представление адреса на пользовательском уровне
01, 05, 15	Выходной бит	0xxx
02	Входной бит	1xxx
04	Входной регистр	3xxx
03, 06, 16	Выходной регистр	4xxx

В сообщениях при передаче данных (**transmission messages**) в последовательной линии передаче адреса в пользовательской системе MODBUS указываются с 0.

Собственно в пользовательской системе MODBUS эти адреса считаются, начиная с 1.

Пример: Первый выходной удерживающий (holding) регистр в пользовательской системе представляется как регистр 40001. В сообщениях при передаче данных (transmission messages) 0000 Hex передается как адрес регистра при использовании FC 03, 06, или 16.

127-я катушка (coil) представляется как катушка 00127 в пользовательской системе и в сообщениях при передаче данных имеет адрес 007E Hex.

2 Инсталляция

2.1 Использование защитного ключа-заглушки

Введение

Для того чтобы запускать CP с загружаемыми драйверами, необходим защитный ключ-заглушка. Если ключ-заглушка установлен в порт CP, то драйвер может быть загружен. Оба интерфейса CP 441-2 могут работать с загружаемыми драйверами.

Установка защитного ключа-заглушки

Для того чтобы вставить ключ-заглушку, необходимо снять модуль CP с монтажной шины. На задней панели модуля над разъемами для подключения к системной шине есть слот, в который может быть установлен ключ-заглушка.

2.2 Интерфейсы

Интерфейс RS232C / TTY

Интерфейс RS232C / TTY позволяет создать PtP-соединение с ведомой (slave) системой.

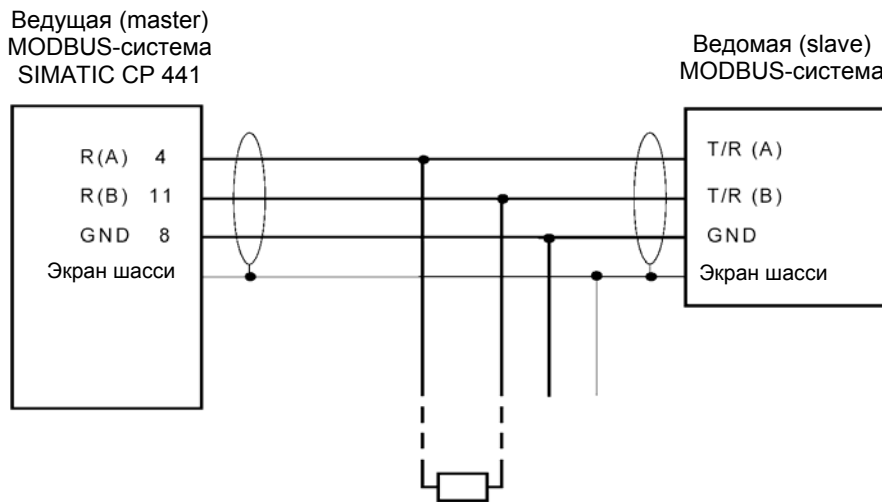
Дополнительную информацию по подключению к интерфейсам Вы можете получить в руководстве "CP 341 or CP 441-2 Point-to-Point Communication" ("PtP-соединение с CP 341 или CP 441-2").

Интерфейс X27 / RS485 (двухпроводный режим)

Интерфейс X27 / RS485 (2-хпроводный режим) позволяет установить многоточечное коммуникационное соединение (сеть) одной ведущей (master) системы с несколькими (до 32) ведомых (slave) устройств.

Драйвер CP обеспечивает переключение двухпроводной линии в режим приема или передачи.

Ниже представлена схема соединения 1 ведущая система - 1 ведомая система на шине:



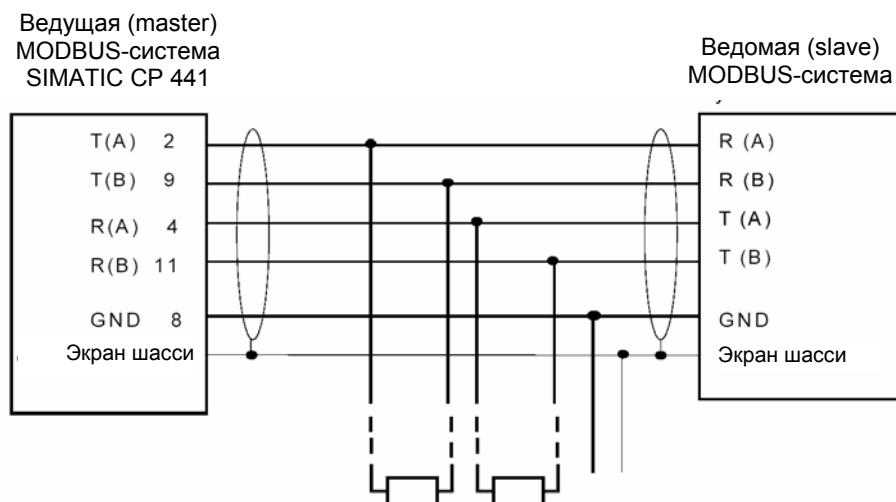
Интерфейс X27 / RS422 (четырёхпроводный режим)

Интерфейс X27 / RS422 (4-хпроводный режим) позволяет установить PtP-соединение ведущей (master) системы с ведомой (slave) системой.

Возможно прямое установление многоточечного соединения (сети) более чем с одним ведомыми (slave) устройствами с использованием оборудования ведомых Modbus систем.

Также ведомые Modbus системы должны поддерживать переключение своих передатчиков в состояние высокого выходного сопротивления при отсутствии передачи.

Ниже представлена схема соединения 1 ведущая система - 1 ведомая система на шине:



Дополнительную информацию Вы можете найти в главе В данного руководства и в руководстве "CP 341 or CP 441-2 Point-to-Point Communication" ("PtP-соединение с CP 341 или CP 441-2").

Загружаемый драйвер для PtP-соединений CP
(S7 - ведущее устройство)
A5E00218410-04

3 Настройка драйвера

Общая информация

Весь материал в последующих разделах данного руководства, посвященный STEP 7, относится к ПО STEP 7 версии 4.02.

Последовательности операций, имена и имена каталогов (директорий) могут быть изменены в последующих версиях ПО.

3.1 Инсталляция драйвера в STEP 7 в программаторе/ПК

Для инсталляции драйвера, состоящего из программного модуля и определяемых им файлов, выполните следующие действия:

1. Вставьте компакт-диск Modbus Master CD в привод CD-ROM.
2. В панели управления Windows откройте диалоговое окно для установки программного обеспечения двойным щелчком на значке "Add / Remove Software" ("Установка / удаление программ").
3. В диалоговом окне выберите привод CD-ROM и запустите на выполнение программу инсталляции Setup.EXE.
4. Последовательно шаг за шагом выполняйте инструкции по установке, которые Вы увидите на экране.

Результат:

Драйвер и интерфейс назначения параметров (parameter assignment) устанавливаются в следующий раздел: Step7\S7ftp\S7Driver.

Раздел содержит следующие файлы:

- S7wfpa1a.dll
- S7wfpa1x.cod
- S7wfpa2x.cod

3.2 Деинсталляция драйвера

Драйвер может быть удален из STEP 7 под управлением Windows посредством утилиты из панели управления для установки и удаления программ ("Add / Remove Software").

Пользователь может проконтролировать, все ли файлы S7wfra1?.*, S7wfra2?.*, S7wfra3?* были успешно удалены из раздела Step7\S7ftp\S7Driver.

Примечание

До деинсталляции программного пакета *Point-to-Point Communication, Parameter Assignment (CP 341: PtP-соединение, назначение параметров)* Вы сначала должны удалить (деинсталлировать) все загружаемые драйверы.

3.3 Конфигурирование данных коммуникаций

Введение

Конфигурирование данных коммуникаций (Data Link) заключается в размещении оборудования в таблице конфигурации (configuration table) с использованием утилиты конфигурирования HW config. Конфигурация может выполняться с помощью средств STEP 7.

3.3.1 Конфигурирование данных коммуникаций для CP 341

Проект в S7

Перед выполнением конфигурирования Вы должны создать S7-проект средствами ПО STEP 7.

Компоненты проекта

Вставьте необходимые компоненты в открытый проект, используя SIMATIC Manager: станция S7-300.

Перед каждой вставкой Вы должны выбрать требуемый проект:

Insert (Вставка) -> Station (Станция) -> SIMATIC 300 Station

Загружаемый драйвер для PtP-соединений CP
(S7 - ведущее устройство)
A5E00218410-04

Конфигурация оборудования

Конфигурация оборудования заключается собственно в определении компонентов оборудования, а также в определении свойств этих компонентов.

Для запуска конфигурирования оборудования выберите станцию "SIMATIC 300 Station" и дважды щелкните на пункте "Hardware" ("Оборудование") или выберите опции меню:
Edit (Правка) -> Open Object (Открыть объект).

Используйте опции меню:

Insert (Вставка) -> Hardware Components (Компоненты оборудования) для вставки объекта стойки станции "RACK 300", модуля источника питания "PS-300", процессора "CPU-300" из "SIMATIC 300" и модуля "CP PtP" из "CP-300" с подходящим номером компонента.

Подробное описание конфигурирования модулей S7-300 Вы можете найти в руководстве пользователя по STEP 7.

3.3.2 Конфигурирование данных коммуникаций для CP 441-2

Для обеспечения данных коммуникаций для PtP-соединения Вы должны сконфигурировать станцию S7-400, связанную с ней партнерскую станцию, PtP-узлы и PtP-сеть.

Проект в S7

Перед выполнением конфигурирования Вы должны создать S7-проект средствами ПО STEP 7.

Компоненты проекта

Вставьте необходимые компоненты в открытый проект, используя SIMATIC Manager: станцию S7-400, партнерскую станцию и PtP-сеть.

Перед каждой вставкой Вы должны щелчком выбрать требуемый проект:

Insert (Вставка) -> Station (Станция) -> SIMATIC 400 Station

(для Вашей собственной S7-программы (Rack, PS, CPU, CP 441-2, ...),

Insert (Вставка) -> Station (Станция) -> Other Station (Другая станция)

(Для партнерской станции)

Insert (Вставка) -> Subnet (Подсеть) -> PtP

(Для PtP-подсети между станцией S7-400 и партнерской станцией)

Конфигурация оборудования

Конфигурация оборудования заключается собственно в определении компонентов оборудования, а также в определении свойств этих компонентов.

Для запуска конфигурирования оборудования выберите станцию "SIMATIC 400 Station" и дважды щелкните на пункте "Hardware" ("Оборудование") или выберите опции меню:
Edit (Правка) -> Open Object (Открыть объект).

Используйте опции меню:

Insert (Вставка) -> Hardware Components (Компоненты оборудования) для вставки объекта стойки станции "RACK 400", модуля источника питания "PS-400", процессора "CPU-400" из "SIMATIC 400" и модуля "CP PtP" из "CP-400" с подходящим номером компонента.

Подробное описание конфигурирования модулей S7-400 Вы можете найти в руководстве пользователя по STEP 7.

3.4 Назначение параметров для CP

Назначение параметров для модуля CP

После того, как Вы расположили модули в Вашей стойке, используя утилиту конфигурирования HW Config, Вы должны назначить для них параметры.

Для запуска утилиты назначения параметров дважды щелкните на модуле CP в "Hardware Configuration" ("Конфигурация оборудования") или щелкните на модуле CP и выберите опции меню:

Edit (Правка) -> Object Properties (Свойства объекта).

3.4.1 Назначение параметров для CP 341

- 1) Окно "Properties - CP" ("Свойства CP") -> вкладка "Basic Parameters" ("Основные параметры")

Щелкнув один раз на кнопке "Parameters" ("Параметры"), откройте интерфейс для выбора протокола "Parameter Assignment of Point-to-Point Connection" ("Назначение параметров для PtP-соединения"). Здесь Вы можете выбрать требуемый протокол обмена данными.

После выбора протокола Вы можете выполнить назначение параметров драйвера ("Parameter Assignment of the Driver") (запуск осуществляется двойным щелчком на символе "почтовый ящик").

Подробное описание процесса выбора протокола и назначения параметров для загружаемых драйверов Вы можете найти в разделе "Назначение параметров для загружаемого драйвера".

После назначения параметров Вы возвращаетесь в диалоговое окно "Properties - CP" ("Свойства CP").

- 2) Окно "Properties - CP" ("Свойства CP") -> вкладка "Addresses" ("Адреса")

Никаких установок не требуется выполнять на вкладке "Addresses" ("Адреса") диалогового окна "Properties - CP" ("Свойства CP").

- 3) Окно "Properties - CP" ("Свойства CP") -> вкладка "General" ("Общие свойства")

Никаких установок не требуется выполнять на вкладке "General" ("Общие свойства") диалогового окна "Properties - CP" ("Свойства CP").

Вы можете завершить параметризацию модуля CP щелчком на кнопке "OK" в диалоговом окне "Properties - CP" ("Свойства CP"). После этого Вы возвращаетесь в диалоговое окно "Hardware Configuration" ("Конфигурация оборудования").

Сохраните назначенные параметры и закройте диалоговое окно "Hardware Configuration" ("Конфигурация оборудования"). После этого Вы возвращаетесь в основное меню проекта STEP 7.

3.4.2 Назначение параметров для CP 441-2

- 1) Окно "Properties - CP 441-2" ("Свойства CP 441-2") -> вкладка "Basic Parameters" ("Основные параметры")

Определите требуемый интерфейс ("Interface") для модуля CP 441-2 (1= верхний (upper), 2= нижний (lower)) на вкладке "Basic Parameters" ("Основные параметры"). Выберите установленный интерфейсный подмодуль как "Module" ("Модуль").

Щелкнув один раз на кнопке "Parameters" ("Параметры"), откройте интерфейс для выбора протокола "Parameter Assignment of Point-to-Point Connection" ("Назначение параметров для PtP-соединения"). Здесь Вы можете выбрать требуемый протокол обмена данными.

После выбора протокола Вы можете выполнить назначение параметров драйвера ("Parameter Assignment of the Driver") (запуск осуществляется двойным щелчком на символе "почтовый ящик").

Подробное описание процесса выбора протокола и назначения параметров для загружаемых драйверов Вы можете найти в разделе "Назначение параметров для загружаемого драйвера".

После назначения параметров Вы возвращаетесь в диалоговое окно "Properties - CP 441-2" ("Свойства CP 441-2").

- 2) Окно "Properties - CP 441-2" ("Свойства CP 441-2") -> вкладка "Addresses" ("Адреса")

Никаких установок не требуется выполнять на вкладке "Addresses" ("Адреса") диалогового окна "Properties - CP 441-2" ("Свойства CP").

3) Окно "Properties - CP 441-2" ("Свойства CP 441-2") -> вкладка "General" ("Общие свойства")

На вкладке "General" ("Общие свойства") диалогового окна "Properties - CP 441-2" ("Свойства CP 441-2") Вы определяете PtP-сеть, к которой подключаются интерфейсы модуля CP.

PtP(1) соответствует верхнему интерфейсу, а PtP(2) - соответствует нижнему интерфейсу модуля CP.

Щелчок на PtP(1) или PtP(2) открывает диалоговое окно для конфигурирования соответствующей подсети проекта.

Выберите требуемую подсеть (subnet) и активируйте элемент управления checkbox "Partner is connected to the selected network" ("Партнер подсоединен к выбранной подсети").

Щелкните на кнопке "OK" для возврата в диалоговое окно "Properties - CP 441-2" ("Свойства CP 441-2"). Щелчком на кнопке "OK" в этом диалоговом окне Вы можете завершить параметризацию модуля CP 441-2, после чего Вы возвращаетесь в диалоговое окно "Hardware Configuration" ("Конфигурация оборудования").

Сохраните назначенные параметры и закройте диалоговое окно "Hardware Configuration" ("Конфигурация оборудования"). После этого Вы возвращаетесь в основное меню проекта STEP 7.

Назначение параметров для коммуникационного партнера

После вставки в S7-проект партнерской станции как описано выше, Вы должны определить свойства для этой станции.

В открытом проекте Вы можете щелчком выбрать станцию-коммуникационного партнера (other station).

Выберите опции меню:

Edit (Правка) -> Object Properties (Свойства объекта).

При этом откроется диалог "Properties - Other Station" ("Свойства - Другая станция")

1) Окно "Properties - Other Station" ("Свойства - Другая станция") -> вкладка "Node List" ("Список узлов")

Щелкните на кнопке "New" ("Новый") на вкладке "Node List" ("Список узлов"). В окне "Select Type" ("Выбор типа") выберите "PtP Nodes" ("PtP-узлы") и щелкните затем на кнопке "OK".

После этого появится диалоговое окно "Network Connection" ("Подключение к сети").

Выберите требуемую подсеть, которая соединяет интерфейс модуля CP с интерфейсом коммуникационного партнера, и активируйте элемент управления checkbox "Node is connected to the selected network" ("Узел подключен к выбранной подсети").

После этого щелкните затем на кнопке "OK" для возвращения на вкладку "Node List" ("Список узлов").

- 2) Окно "Properties - Other Station" ("Свойства - Другая станция") -> вкладка "General" ("Общие свойства")

Никаких установок не требуется выполнять на вкладке "General" ("Общие свойства").

Вы можете завершить параметризацию станции партнера щелчком на кнопке "ОК". После этого Вы возвращаетесь в основное меню проекта STEP 7.

Партнерская станция может иметь несколько интерфейсов (другими словами, несколько PtP-узлов) и может быть посредством этих интерфейсов подключена к различным PtP-подсетям.

3.5 Конфигурирование данных коммуникаций проекта

Данный раздел касается только модуля CP 441-2. Если Вы используете модуль CP 341, то Вы можете пропустить данный раздел.

Коммуникационные связи

В коммуникационном соединении S7-CPU с коммуникационным партнером / в подключении шины посредством PtP-соединения коммуникационный процессор CP играет роль элемента, обеспечивающего связь. Для каждого последовательного интерфейса, подключаемого к коммуникационному партнеру / к шине, Вы должны выполнить конфигурирование данных коммуникаций.

Конфигурирование данных коммуникаций

Выберите CPU в проекте STEP 7 в открытой станции S7-400 и двойным щелчком на "Connection" ("Соединение") откройте диалог "Project configuration of data link" ("Конфигурирование данных коммуникаций проекта"). При этом появится диалоговое окно "Carry out Project Configuration of Connections" ("Выполнение конфигурирования соединений проекта").

Выберите опцию меню:

Insert (Вставка) -> Connection (Соединение) ,

чтобы открыть окно диалога "New Connection" ("Новое соединение"). Здесь Вы можете выбрать коммуникационного партнера ("other station" - другую станцию) для нового соединения и выбрать в качестве типа соединения ("connection type") - тип "S7 Point-to-Point Connection" (S7 PtP-соединение).

Подтвердите свой выбор щелчком на кнопке "ОК".

После этого откроется диалоговое окно "Connection Properties" ("Свойства соединения").

Диалоговое окно "Connection Properties" ("Свойства соединения")

- Вы получаете идентификатор ID соединения, который при необходимости Вы можете изменить.
- Выберите "Communication Direction 3: Local <-> Partner" (Направление соединения 3: локальный <-> партнер)
- Отображается rooting для параметризованного соединения.
- Оба обозначения номера CPU некорректны для обработки данным драйвером.

Подтвердите сделанные установки щелчком на кнопке "ОК".

Сохраните "Project configuration of data link" ("Конфигурирование данных коммуникаций проекта") и закройте диалоговое окно.

Примечание

Вы должны помнить, что идентификатор соединения (Local ID) должен также использоваться при вызове функций SFB в пользовательской программе.

3.6 Назначение параметров для загружаемого драйвера

Открытие утилиты для назначения параметров CP-PtP (Parameter Assignment Tool CP-PtP)

Для запуска конфигурирования оборудования выберите станцию "SIMATIC Station" и дважды щелкните на пункте "Hardware" ("Оборудование") или выберите опции меню:

Edit (Правка) -> Open Object (Открыть объект).

Щелкните на CP и выберите опции меню:

Edit (Правка) -> Object Properties (Свойства объекта).

После того, как Вы выбрали интерфейс (только для CP 441-2) и интерфейсный submodule (только для CP 441-2), щелкните на кнопке "Parameters" ("Параметры"); при этом откроется интерфейс для выбора протокола "Parameter Assignment of Point-to-Point Connection" ("Назначение параметров для PtP-соединения").

Выбор протокола

Кроме стандартных протоколов в окне со списком также отображаются все установленные загружаемые драйверы. Выберите для загружаемого драйвера опцию "MODBUS Master" ("Ведущее MODBUS-устройство"). После этого дважды щелкните кнопкой манипулятора "мышь" на символе протокола передачи (значок "почтовый ящик"); при этом откроется диалоговое окно, в котором устанавливаются параметры, которые зависят от драйвера.

Параметры, зависящие от драйвера

Параметры, описываемые далее, могут быть заданы для данного драйвера в отдельных диалоговых окнах.

3.6.1 Протокол для ведущего устройства Modbus

Обзор параметров для обмена данными

Скорость, Фрейм символа			
Параметр	Описание	Диапазон значений	Значение по умолчанию
Скорость передачи (Transmission Rate)	Скорость передачи данных, измеряемая количеством битов в секунду	300	9600
		600	
		1200	
		2400	
		4800	
		9600	
		19200	
		38400	
		76800	
Дополнительная скорость передачи для CP 341 (№ 6ES7 341-1xH01-0AE0)		57600	
Дополнительная скорость передачи для CP 441-2 (№ 6ES7 441-2AA03-0AE0)		57600 115200	
Биты данных (Data Bits)	Битов на символ	8	8
Стоповые биты (Stop Bits)	Количество стоповых битов	1 2	1
Четность (Parity)	Бит четности не передается	Нет	Четный
	Количество битов данных заканчивается нечетным	Нечетный	
	Количество битов данных заканчивается четным	Четный	

Скорость передачи (Transmission Rate)

Скорость передачи данных, измеряемая количеством битов в секунду

Биты данных (Data Bits)

Количество битов данных в передаваемом символе

Стоповые биты (Stop Bits)

Количество стоповых битов определяет наименьший временной интервал между двумя передаваемыми символами.

Четность (Parity)

Бит четности служит для контроля цельности данных; в зависимости от заданных параметров, он завершает количество передаваемых битов четным или нечетным номером.

Если выбрано значение “no” (“нет”), то это означает, что бит четности не передается, при этом уменьшается надежность сохранения цельности передаваемых данных.

Обзор параметров протокола обмена данными

Параметры протокола			
Параметр	Описание	Диапазон значений	Значение по умолчанию
Время ожидания ответного сообщения (Reply Monitoring Time)	Контролируемое время, в течение которого ожидается начало передачи ответного сообщения ведомым (slave) устройством	5 ... 65500	2000
Рабочий режим (Operating Mode)	Нормальный режим (“Normal Operation”) Подавление помех (“Interference Suppression”)	Нормальный режим (“Normal Operation”) Подавление помех (“Interference Suppression”)	Нормальный режим (Normal)
Множитель для времени ожидания символа (Multiplier Character Delay Time)	Множитель для заданного времени ожидания символа (character delay time), зависящего от скорости передачи данных	1 ... 10	1

Время ожидания ответного сообщения (Reply Monitoring Time)

Время ожидания ответного сообщения - это время, в течение которого ожидается начало передачи сообщения ведомым (slave) устройством в ответ на сообщение-запрос ведущего (master) устройства.

Нормальный режим ("Normal Operation")

В данном рабочем режиме все распознанные ошибки при передаче данных и/или отказы, вызванные обрывом связи (BREAK) до и после принятых сообщений от ведомого устройства, приводят к соответствующим сообщениям об ошибках (error message).

Подавление помех ("Interference Suppression")

В данном рабочем режиме в случае распознавания отказа, вызванного обрывом линии связи ("BREAK"), при входящем сообщении или, если интерфейсный блок CP обнаружил ошибки при передаче, то драйвер рассматривает принятое сообщение ошибочным и игнорирует его.

Начало входящего сообщения от ведомого (slave) устройства распознается посредством корректно принятого адреса ведомого устройства (slave address).

Распознанные ошибки при передаче данных и/или отказы, вызванные обрывом связи (BREAK), также игнорируются, если они происходят после окончания приема сообщения (CRC-код).

Множитель для времени ожидания символа (Multiplier Character Delay Time)

Если коммуникационный партнер не может обеспечить временных параметров, соответствующих требованиям спецификаций MODBUS, то возможно использование множителя f_{MUL} для контролируемых периодов времени задержки символа (character delay time) ZVZ.

Параметр "время ожидания (задержки) символа" (character delay time) должен корректироваться, только если коммуникационный партнер не может обеспечить требуемых временных параметров.

В результате "время ожидания (задержки) символа" (character delay time) t_{ZVZ} рассчитывается следующим образом:

$$t_{ZVZ} = t_{ZVZ_TAB} \times f_{MUL};$$

t_{ZVZ_TAB} : табличное значение для ZVZ (см. главу "Протокол обмена данными")

f_{MUL} : множитель (коэффициент).

3.6.2 Интерфейс X27 (RS 422/485)

Общий обзор

Интерфейс X27 (RS 422/485)			
Параметр	Описание	Диапазон значений	Значение по умолчанию
Presets of the Receiving Line (Предустановки для приемной линии)	No presets (нет предустановок) Preset "Break" (Предустановка "Обрыв") Preset "High" (Предустановка "Высокий уровень")	None R(A)5V,R(B)0V R(A)0V,R(B)5V	R(A)5V,R(B)0V
X27 Operating Mode (Рабочий режим X27)	Использование линии передачи T(A), T(B) для передачи данных, использование линии R(A), R(B) для приема данных. Переключение входной (приемной) линии R(A), R(B) при переключении между режимами передачи (send) и приема (receive).	Полный дуплексный режим (RS422), 4-хпроводный режим работы Полу-дуплексный режим (RS485), 2-хпроводный режим работы	Полный дуплексный режим (RS422), 4-хпроводный режим работы

Полный дуплексный режим (RS422), 4-хпроводный режим работы ("Full-Duplex (RS422) Four-Wire Operation")

В этом рабочем режиме для передачи данных используются линии передачи T(A), T(B), а для приема данных используются входные линии (для приема) R(A), R(B).

Обработка ошибок выполняется в соответствии с установленной функцией в параметре "Driver Operating Mode" ("Режим работы драйвера"): "нормальный режим" ("Normal") или "режим с подавлением помех" ("Interference Suppression").

Полудуплексный режим (RS485), 2-хпроводный режим работы ("Half-Duplex (RS485) Two-Wire Operation")

В этом рабочем режиме драйвер обеспечивает переключение 2-хпроводной приемной линии интерфейса R(A), R(B) при переключении между режимами передачи (send) и приема (receive).

В данном рабочем режиме все распознанные ошибки передачи и/или BREAK до и после входных сообщений игнорируются. Уровень обрыва (BREAK) в линиях в паузах между сообщениями также игнорируется.

Начало входных сообщений от ведомого (slave) устройства распознается при приеме корректного адреса ведомого устройства (slave address).

Рекомендуется использование предустановки R(A) 0V, R(B) 5V (High) (R(A) 0B, R(B) 5B (высокое)) для входных (приемных) линий интерфейса.

Предустановки для входных линий

"None" (плавающее значение)

При данном значении параметра нет предустановок для 2-хпроводной приемной линии интерфейса R(A),R(B). При этом текущие (мгновенные) значения сигналов в линиях определяются и обеспечиваются коммуникационным партнером.

Предустановка "R(A) 5V, R(B) 0V" (BREAK) ("Обрыв")

Такая предустановка для 2-хпроводной приемной линии интерфейса обеспечивается в CP следующим образом:

$R(A) \rightarrow +5B, R(B) \rightarrow 0B (V_A - V_B \geq +0.3V)$.

Это значит, что в линии возникает "уровень обрыва" (BREAK level) в CP при реальном событии - при обрыве линии.

Предустановка "R(A) 0V, R(B) 5V" (High) ("Высокое")

Такая предустановка для 2-хпроводной приемной линии интерфейса обеспечивается в CP следующим образом:

$R(A) \rightarrow 0B, R(B) \rightarrow +5B (V_A - V_B \leq -0.3V)$.

Это значит, что в линии возникает "высокий уровень" (HIGH level) в CP при реальном событии - при обрыве линии (и/или когда CP находится в состоянии ожидания, то есть, когда ведомое (slave) устройство не передает данных).

Состояние линии "BREAK" ("Обрыв") не может быть распознано.

Выбор параметров

Выберите требуемые значения параметров коммуникационных данных и выходите из соответствующего диалога, нажав кнопку "OK."

3.7 Загрузка данных конфигурации и параметризации для CP 341

Управление данными (Data Management)

При закрытии "Hardware Configuration" ("Конфигурация оборудования") данные автоматически сохраняются в Вашем проекте STEP 7.

Загрузка данных конфигурации и параметров

Заданные значения данных конфигурации и параметров теперь могут быть загружены в интерактивном (online) режиме из программатора в CPU. Для передачи данных в CPU используйте опции меню: *PLC -> Download (Загрузить)*.

Во время запуска CPU и всякий раз при переключении из режима STOP (СТОП) в RUN (ВЫПОЛНЕНИЕ) параметры модуля CP автоматически пересылаются в CP, как только модуль CP становится доступен посредством системной шины S7-300.

Программный модуль драйвера (коды) не сохраняются в CPU, однако непосредственно с помощью утилиты назначения параметров (parameter assignment tool) сохраняется в реманентной памяти модуля CP 341. Однако Вы должны иметь в виду, что это значит, что Вы не можете заменить модуль без программатора.

Дополнительная информация

Вы можете использовать руководство пользователя по программному обеспечению STEP 7 ("User Manual for STEP 7") для получения более подробной информации по следующим темам:

- Сохранение данных конфигурации и параметров.
- Загрузка (load) данных конфигурации и параметров в CPU.
- Считывание, изменение, копирование и распечатка данных конфигурации и параметров.

3.8 Загрузка данных конфигурации и параметризации для CP 441-2

Управление данными (Data Management)

При закрытии "Hardware Configuration" ("Конфигурация оборудования") и/или "Project Configuration of Connection" ("Конфигурация соединений в проекте") данные (включая данные модуля CP и программный (кодový) модуль драйвера) автоматически сохраняются в Вашем проекте STEP 7.

Загрузка данных конфигурации и параметров

Заданные значения данных конфигурации и параметров теперь могут быть загружены в интерактивном (online) режиме из программатора в CPU. Для передачи данных в CPU используйте опции меню:
PLC -> Download (Загрузить).

Во время запуска CPU и всякий раз при переключении из режима STOP (СТОП) в RUN (ВЫПОЛНЕНИЕ) параметры модуля CP автоматически пересылаются в CP, как только модуль CP становится доступен посредством системной шины S7-400.

Дополнительная информация

Вы можете использовать руководство пользователя по программному обеспечению STEP 7 ("User Manual for STEP 7") для получения более подробной информации по следующим темам:

- Сохранение данных конфигурации и параметров.
- Загрузка (load) данных конфигурации и параметров в CPU.
- Считывание, изменение, копирование и распечатка данных конфигурации и параметров.

3.9 Характеристики запуска CP

Введение

Запуск модуля CP происходит в два этапа:

- Инициализация (включение питания CP)
- Назначение параметров

Инициализация (Initialization)

Как только подается напряжение на модуль CP и после отработки программы тестирования оборудования, микропрограмма в CP подготавливается к выполнению.

Назначение параметров (Parameter Assignment)

Во время назначения параметров модуль CP получает параметры, соответствующие слоту, в котором он установлен.

После этого коммуникационный процессор CP готов к выполнению своих функций.

3.10 Назначение параметров запуска ("Startup of the CPU" - "Запуск CPU")

Этот раздел относится только к коммуникационным процессорам CP 441-2. Если в качестве коммуникационного процессора Вы используете модуль CP 341, то Вы можете не принимать данный раздел во внимание.

Конфигурация оборудования

Чтобы не возникало проблем при запуске CPU-CP, при конфигурировании оборудования в диалоге "Hardware Configuration" ("Конфигурация оборудования") при назначении параметров CPU ("Parameter Assignment of the CPU") необходимо выполнить следующие установки.

После запуска утилиты для назначения параметров кнопкой манипулятора "мышь" дважды щелкните на CPU или один раз щелкните на CPU и затем выберите опции меню:

Edit (Правка) -> Object Properties (Свойства объекта);

после этого появится диалоговое окно "Properties - CPU" ("Свойства - CPU").

На вкладке "Startup" ("Запуск") установите минимальное значение контролируемого времени в окне ("Monitoring Time for"), равное 3000 (что означает 300 с) в пункте "Transfer of Parameters to Module (100ms):" ("Передача параметров в модуль (100 мс)").

Смысл:

Когда для интерфейса модуля CP 441-2 назначаются параметры путем загрузки драйвера, программный модуль (коды) драйвера передается в коммуникационный процессор CP, как и назначенные значения параметров. Процедура загрузки параметров в целом контролируется на предмет превышения заданного временного интервала, упомянутого выше; при этом значение этого параметра должно быть задано достаточно большим.

4 Протокол обмена данными

4.1 Инсталляция драйвера в STEP 7 в программаторе / ПК

Общая информация

Используемая процедура "прозрачна" с точки зрения кодов, она является асинхронной и предполагает полудуплексный режим.

Передача данных выполняется без квитирования установления связи.

Связь ведущего и ведомого устройств (Master-Slave-связь)

Модуль CP инициирует передачу (CP - ведущее устройство - master), и после выдачи сообщения запроса он ожидает ответное сообщение от ведомого (slave) устройства в течение установленного времени ожидания ответа (reply monitoring time).

Структура сообщения

Передаваемые данные "Master-Slave" и/или "Slave-Master" начинаются с адреса ведомого устройства (Slave Address), за которым следует функциональный код (Function Code). Затем следуют пересылаемые данные. Структура полей данных зависит от используемого кода функции. Данные для CRC-проверки передаются в конце сообщения.

Адрес	Функция	Данные	CRC
1 байт	1 байт	n байтов	2 байта

Адрес	Адрес ведомого MODBUS-устройства (MODBUS Slave Address)
Функция	Функциональный код MODBUS (MODBUS Function Code)
Данные сообщения:	Byte_Count (счетчик байтов), Coil_Number (число катушек), Data (данные)
CRC	Контрольная сумма сообщения

Адрес ведомого устройства (Slave Address)

Значение для адреса ведомого устройства лежит в диапазоне от 1 до 255. Используемый адрес - это шинный адрес определенного ведомого устройства.

Рассылка сообщений

Ведущее (master) устройство использует нулевой адрес (zero) в качестве адреса ведомого устройства при адресации одновременно всех ведомых устройств шины. Рассылка сообщений (**Broadcast Messages**) разрешается только для функциональных кодов **05, 06, 15** и **16**.

Рассылка сообщений не сопровождается ответными сообщениями ведомых устройств.

Функциональный код (Function Code)

Функциональный код определяет значение, а также структуру сообщения. Модулем CP поддерживаются следующие функциональные коды:

Функциональный код	Функция в соответствии со спецификацией MODBUS
01	Считывание состояния выхода (Read Output Status)
02	Считывание состояния входа (Read Input Status)
03	Считывание выходных регистров (Read Output Registers)
04	Считывание входных регистров (Read Input Registers)
05	Назначение для одной катушки (Force Single Coil)
06	Предустановки для одного регистра (Preset Single Register)
07	Считывание состояния исключения (Read Exception Status)
08	Тестирование обратной связи (Loop Back Diagnostic Test)
11	Считывание значения счетчика коммуникационных событий (Fetch Communications Event Counter)
12	Считывание протокола коммуникационных событий (Fetch Communications Event Log)
15	Назначения для нескольких катушек (Force Multiple Coils)
16	Предустановки для нескольких регистров (Preset Multiple Registers)

Поле данных DATA

Поле данных DATA определяется функциональным кодом, как например: Bytecount (счетчик байтов), Coil_Start Address (начальный адрес катушки), Register_Start Address (начальный адрес регистра); Number_of_Coils (число катушек), Number_of_Registers (число регистров) и т.д.. См. также главу "Функциональные коды".

Проверка четности (CRC Check)

Конец сообщения идентифицируется с помощью контрольной суммы CRC 16, состоящей из 2 байтов. Она рассчитывается как полином:

$$x^{16} + x^{15} + x^2 + 1.$$

Первым передается младший байт, затем - старший байт.

Конец сообщения

Загружаемый драйвер распознает конец сообщения, затем передачи нет в течение периода времени, требуемого для передачи трех с половиной символов ($3.5 \times \{\text{время ожидания символа (character delay time)}\}$) (см. справочное руководство по протоколу MODBUS (*MODBUS Protocol Reference Guide*)).

Следовательно, задержка TIME_OUT зависит от скорости передачи.

Скорость передачи, бит/с	TIME_OUT, мс
76800	0,5
38400	1
19200	2
9600	4
4800	8
2400	16
1200	32
600	64
300	128

После истечения задержки TIME_OUT по окончании передачи проверяется формат ответного сообщения, принятого ведомым (slave) устройством.

Сообщения об отказах (Exception Responses)

При распознавании ошибки в сообщении запроса от ведущего (master) устройства (например, некорректный адрес регистра), ведомое (slave) устройство устанавливает старший значащий бит в функциональном коде ответного сообщения.

За этим следует передача одного байта кода ошибки (Exception Code), который описывает причину ошибки.

Подробное описание значений вышеупомянутых параметров может быть найдено в руководстве по протоколу "GOULD MODICON Modbus Protocol".

Коды ошибок в сообщениях

Сообщение с кодом ошибки (error code) от ведомого устройства (slave) имеет следующую структуру: например, адрес ведомого устройства 5, функциональный код 5, код ошибки 2

Коды ошибок в сообщениях ведомого устройства EXCEPTION_CODE_хх:

05H	адрес ведомого устройства
85H	функциональный код
02H	код ошибки (1 ... 7)
ххH CRC	младший байт контрольного кода ("Low")
ххH CRC	старший байт контрольного кода ("High")

При получении драйвером сообщения с кодом ошибки текущее задание завершается с ошибкой.

Номер ошибки соответствует принятому коду ошибки (Exception Code 1-7) вводится в область SYSTAT.

При этом нет ввода в целевой блок данных BRCV.

Следующие коды ошибок определены в соответствии со спецификациями MODBUS:

Код ошибки	Значение в соответствии со спецификациями MODBUS	Причина (краткое описание) *
1	Некорректная функция	Некорректный функциональный код (function code)
2	Некорректный адрес данных	Указан некорректный адрес данных ведомого устройства
3	Некорректное значение данных	Некорректное значение данных ведомого устройства
4	Отказ соответствующего устройства	В ведомом устройстве обнаружена внутренняя ошибка
5	Квитирование	Функция выполняется
6	Занято: отбрасывание сообщения	Ведомое устройство не готово к приему данных
7	Негативное квитирование	Функция не может быть выполнена

* Детальная проверка ведомого устройства.

Вспомогательные сигналы RS 232C (Secondary Signals)

Следующие вспомогательные сигналы RS 232C могут использоваться в CP при использовании интерфейсного модуля RS 232C:

- DCD (входной) **Data Carrier Detect** - Обнаружение несущего сигнала
- DTR (выходной) **Data Terminal Ready** - CP готов к работе
- DSR (входной) **Data Set Ready** - партнер готов к работе
- RTS (выходной) **Request To Send** - CP готов к передаче (send)
- CTS (входной) **Clear To Send** - партнер может принять данные от CP 341 (отклик на сигнал RTS = ON от CP)
- RI (входной) **Ring Indicator** - индикация входящего звонка

Если CP 441 включен, то выходные сигналы в состоянии OFF (выкл) (неактивен).

Вы должны задать параметры для способа использования управляющих сигналов DTR/DSR и RTS/CTS с помощью интерфейса параметризации: *CP 441: Point-to-Point Communication, Parameter Assignment (CP 341: PtP-соединение, назначение параметров)* или управляя ими с помощью вызова функций (FC) в пользовательской программе.

Использование вспомогательных сигналов RS 232C

Вспомогательные сигналы RS 232C могут быть использованы:

- при параметризации автоматического использования вспомогательных сигналов RS 232C
- посредством FC-функций V24_STAT и V24_SET

Примечание

При параметризации автоматического использования вспомогательных сигналов RS 232C невозможны ни RTS/CTS-управление потоками данных, ни RTS- и DTR-управление посредством FC-функции V24_SET. Однако с другой стороны, всегда возможно считывать все вспомогательные сигналы RS 232C с помощью FC V24_STAT.

В следующих разделах описывается как управлять вспомогательными сигналами RS 232C и как их проверять.

Автоматическое использование вспомогательных сигналов

Автоматическое использование вспомогательных сигналов RS 232C в CP производится следующим образом:

- Как только CP переключается с помощью параметризации в рабочий режим с автоматическим использованием вспомогательных сигналов RS 232C, он переключает линию RTS в выключенное состояние (OFF), а линию DTR - во включенное состояние (ON) (это означает, что CP 341 готов к работе (режим - ready)).
Фреймы сообщений не могут посылаться и приниматься до тех пор, пока линия DTR находится в ON-состоянии. Пока DTR находится в OFF-состоянии, никакие данные не принимаются с помощью интерфейса RS 232C. Если выполняется запрос на передачу (send), то он отбрасывается с соответствующим сообщением об ошибке.
- Если выполняется запрос на передачу (send), то RTS устанавливается в ON-состояние и начинается отсчет заданного параметром времени ожидания вывода данных (data output waiting time). Если время ожидания вывода данных заканчивается и CTS = ON, то данные посылаются через интерфейс RS 232C.
- Если линия CTS не устанавливается в состояние ON в течение времени ожидания вывода данных (data output waiting time) для разрешения посылки данных или, если CTS переключается в состояние OFF во время передачи данных, то запрос на передачу (send) отбрасывается и генерируется сообщение об ошибке.
- После выполнения передачи данных линия RTS устанавливается в OFF после истечения заданного параметром времени для перехода RTS в OFF-состояние ("time to RTS OFF"). CP 340 не ждет, пока CTS переключится в OFF.
- Данные могут приниматься с помощью RS 232C-интерфейса, как только линия DSR устанавливается в ON. Если возникает угроза переполнения приемного буфера CP, то CP не выдает ответного сообщения.
- Запрос на передачу (send) или прием данных отбрасывается с соответствующим сообщением об ошибке, если DSR переключается из состояния ON в OFF. При этом в диагностический буфер CP вводится сообщение "DSR = OFF (automatic use of V24 signals)" ("DSR = OFF (автоматическое использование сигналов V24)").

Примечание

Автоматическое использование вспомогательных сигналов RS 232C возможно только в полудуплексном режиме.

При выборе режима автоматического использования вспомогательных сигналов RS 232C невозможны ни RTS/CTS-управление потоками данных, ни RTS- и DTR-управление посредством FC-функции V24_SET.

Примечание

Параметр "Время для перехода RTS в OFF" ("time to RTS OFF") должен быть задан с помощью интерфейса параметризации так, чтобы партнер мог принять последние символы фрейма сообщения во всей их полноте до того, пока RTS и запрос на передачу будут выключены. Параметр "Время ожидания вывода данных" ("data output waiting time") должно выбираться так, чтобы партнер мог перейти в состояние готовности для приема данных до истечения этого времени.

Временная диаграмма

На рисунке 4-1 представлена обработка во времени запроса на передачу (send request):

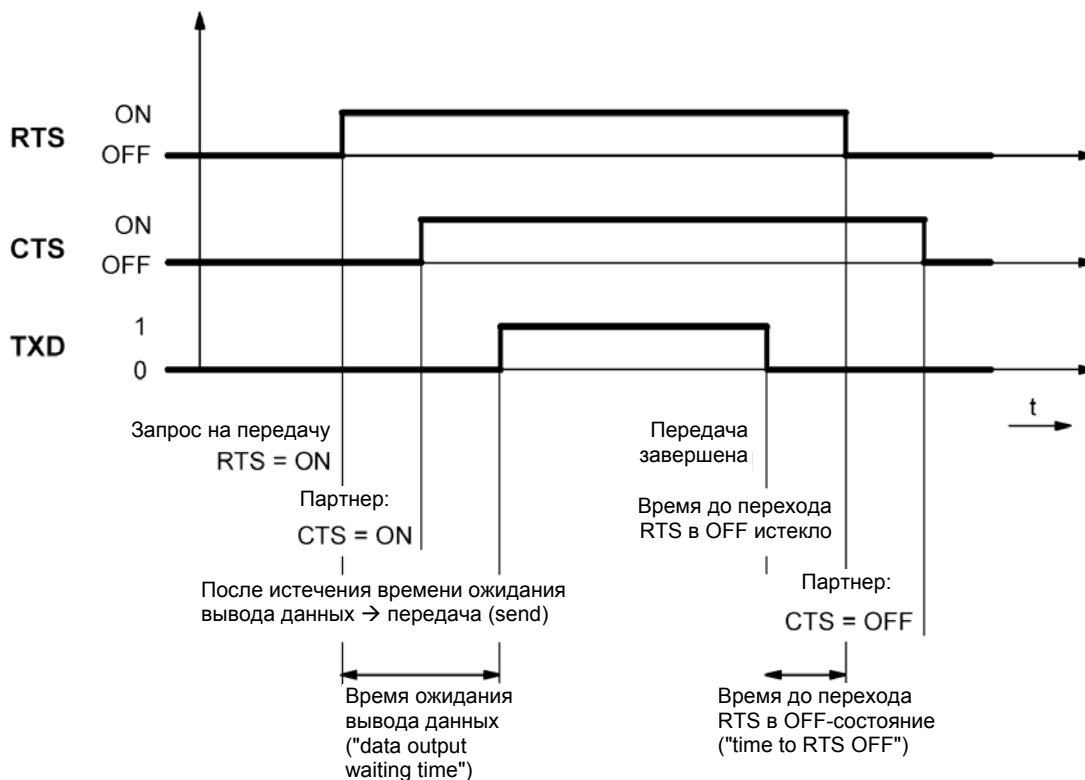


Рис. 4-1 Схема автоматического использования вспомогательных сигналов RS 232C

5 Функциональные коды

5.1 Функциональный код 01 - Read Output Status (считывание состояния выхода)

Функция

Данная функция служит для считывания отдельных битов из памяти ведомого (slave) устройства.

Начальный адрес

Параметр "начальный битовый адрес" (**bit start address**) драйвером не проверяется и передается неизменным.

Количество битов

Допускается любое количество битов от 1 до 2040 указывать в этом параметре (**amount of bits**) (число катушек - "coils").

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого (slave) устройства
+1.0	function	BYTE	B#16#1	Функциональный код
+2.0	bit_startadr	WORD	W#16#0040	Начальный битовый адрес
+4.0	bit_anzahl	INT	16	Количество битов

Пример сообщения-запроса для функции 01:

05H Адрес ведомого устройства
 01H Функциональный код
 00H Начальный битовый адрес (старший байт)
 40H Начальный битовый адрес (младший байт)
 00H Число битов (старший байт)
 10H Число битов (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 01:

05H Адрес ведомого устройства
 01H Функциональный код
 02H Байтовый счетчик
 01H <Данные>
 17H <Данные>
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#1701	Данные

Драйвер вводит данные ответного сообщения в блок данных назначения DV слово за словом.

Первый принятый байт вводится как младший байт первого слова "data[1]", третий принятый байт вводится как младший байт второго слова "data[2]" и т.д..

Если количество данных меньше, чем 9 битов, или если был считан только один младший байт, то значение **00H** вводится в оставшийся старший байт последнего слова.

5.2 Функциональный код 02 - Read Input Status (считывание состояния входа)

Функция

Данная функция служит для считывания отдельных битов из памяти ведомого (slave) устройства.

Начальный адрес

Параметр "начальный битовый адрес" (**bit start address**) драйвером не проверяется и передается неизменным.

Количество битов

Допускается любое количество битов от 1 до 2040 указывать в этом параметре (**amount of bits**) (число катушек - "coils").

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого (slave) устройства
+1.0	function	BYTE	B#16#2	Функциональный код
+2.0	bit_startadr	WORD	W#16#0120	Начальный битовый адрес
+4.0	bit_anzahl	INT	24	Количество битов

Пример сообщения-запроса для функции 02:

05H Адрес ведомого устройства
 02H Функциональный код
 01H Начальный битовый адрес (старший байт)
 20H Начальный битовый адрес (младший байт)
 00H Число битов (старший байт)
 18H Число битов (младший байт)
 ххН Код CRC-проверки (младший байт)
 ххН Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 02:

05H Адрес ведомого устройства
 02H Функциональный код
 03H Байтовый счетчик
 04H <Данные>
 26H <Данные>
 48H <Данные>
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#2604	Данные
+2.0	data[2]	WORD	W#16#0048	Данные

Драйвер вводит данные ответного сообщения в блок данных назначения DB слово за словом.

Первый принятый байт вводится как младший байт первого слова "data[1]", третий принятый байт вводится как младший байт второго слова "data[2]" и т.д..

Если количество меньше, чем 9 битов, или если только один младший байт был считан, то значение 00H вводится в оставшийся старший байт последнего слова.

5.3 Функциональный код 03 - Read Output Registers (считывание выходных регистров)

Функция

Данная функция служит для считывания отдельных регистров из памяти ведомого (slave) устройства.

Начальный адрес

Параметр "начальный адрес регистра" (**Register Start Address**) драйвером не проверяется и передается неизменным.

Количество регистров

Допускается максимальное количество регистров 127 в этом параметре (1 регистр = 2 байта).

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого (slave) устройства
+1.0	function	BYTE	B#16#3	Функциональный код
+2.0	reg_startadr	WORD	W#16#0040	Начальный адрес регистра
+4.0	reg_anzahl	INT	2	Количество регистров

Пример сообщения-запроса для функции 03:

05H Адрес ведомого устройства
 03H Функциональный код
 00H Начальный адрес регистра (старший байт)
 40H Начальный адрес регистра (младший байт)
 00H Число регистров (старший байт)
 02H Число регистров (младший байт)
 ххH Код CRC-проверки (младший байт)
 ххH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 03:

05H Адрес ведомого устройства
 03H Функциональный код
 04H Байтовый счетчик
 21H Адрес регистра 40H (старший байт)
 23H Адрес регистра 40H (младший байт)
 25H Адрес регистра 41H (старший байт)
 27H Адрес регистра 41H (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#2123	Данные
+2.0	data[2]	WORD	W#16#2527	Данные

5.4 Функциональный код 04 - Read Input Registers (считывание входных регистров)

Функция

Данная функция служит для считывания отдельных регистров из памяти ведомого (slave) устройства.

Начальный адрес

Параметр "начальный адрес регистра" (**Register Start Address**) драйвером не проверяется и передается неизменным.

Количество регистров

Допускается максимальное количество регистров 127 в этом параметре (1 регистр = 2 байта).

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого (slave) устройства
+1.0	function	BYTE	B#16#4	Функциональный код
+2.0	reg_startadr	WORD	W#16#0050	Начальный адрес регистра
+4.0	reg_anzahl	INT	3	Количество регистров

Пример сообщения-запроса для функции 04:

05H Адрес ведомого устройства
 04H Функциональный код
 00H Начальный адрес регистра (старший байт)
 50H Начальный адрес регистра (младший байт)
 00H Число регистров (старший байт)
 03H Число регистров (младший байт)
 ххH Код CRC-проверки (младший байт)
 ххH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 04:

05H Адрес ведомого устройства
 04H Функциональный код
 04H Байтовый счетчик
 31H Адрес регистра 50H (старший байт)
 32H Адрес регистра 50H (младший байт)
 33H Адрес регистра 51H (старший байт)
 34H Адрес регистра 51H (младший байт)
 35H Адрес регистра 52H (старший байт)
 36H Адрес регистра 52H (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#3132	Данные
+2.0	data[2]	WORD	W#16#3334	Данные
+4.0	data[3]	WORD	W#16#3536	Данные

5.5 Функциональный код 05 - Force Single Coil (назначение для одной катушки)

Функция

Данная функция служит для установки или сброса отдельных битов в памяти ведомого (slave) устройства.

Начальный адрес

Параметр "битовый адрес" (**bit address**) драйвером не проверяется и передается неизменным.

Состояние битов

Допускается следующие два значения для состояния битов:

FF00H --> установка бита

0000H --> сброс бита

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#5	Функциональный код
+2.0	bit_address	WORD	W#16#0019	Адрес бита
+4.0	bit_state	WORD	W#16#FF00	Состояние бита

Пример сообщения-запроса для функции 05:

05H Адрес ведомого устройства
 05H Функциональный код
 00H Адрес бита (старший байт)
 19H Адрес бита (младший байт)
 FFH Установка бита
 00H
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 05:

05H Адрес ведомого устройства
 05H Функциональный код
 00H Адрес бита (старший байт)
 19H Адрес бита (младший байт)
 FFH Состояние бита (старший байт)
 00H Состояние бита (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

5.6 Функциональный код 06 - Preset Single Register (предустановки для одного регистра)

Функция

Данная функция служит для записывания нового значения в регистр ведомого (slave) устройства.

Адрес регистра

Параметр "адрес регистра" (**Register Address**) драйвером не проверяется и передается неизменным.

Значение регистра

Допускается любое значение в этом параметре (**Register Value**).

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#6	Функциональный код
+2.0	reg_address	WORD	W#16#0180	Адрес регистра
+4.0	reg_value	WORD	W#16#3E7F	Значение регистра

Пример сообщения-запроса для функции 06:

05H Адрес ведомого устройства
 06H Функциональный код
 01H Адрес регистра (старший байт)
 80H Адрес регистра (младший байт)
 3EH Значение регистра (старший байт)
 7FH Значение регистра (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 06:

05H Адрес ведомого устройства
 06H Функциональный код
 01H Адрес регистра (старший байт)
 80H Адрес регистра (младший байт)
 3EH Значение регистра (старший байт)
 7FH Значение регистра (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

5.7 Функциональный код 07 - Read Exception Status (считывание ошибочного состояния)

Функция

Данная функция служит для считывания 8 событийных битов (event bit) ведомого устройства. Начальный номер событийного бита определяется подключенным устройством, а не определяется в программе пользователя.

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#7	Функциональный код

Пример сообщения-запроса для функции 07:

05H Адрес ведомого устройства
 07H Функциональный код
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 07:

05H Адрес ведомого устройства
 07H Функциональный код
 3EH <Данные>
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#3Exx	Данные

Драйвер вводит отдельные биты ответного сообщения в старший байт в блоке данных назначения DB "data[1]".
 Младший байт слова "data[1]" остается без изменений.
 Значение 1 отображается как размер (длина) в параметре LEN функции BRCV.

5.8 Функциональный код 08 - Loop Back Diagnostic Test (тестирование коммуникационного соединения)

Функция

Данная функция служит для проверки коммуникационного соединения.

Диагностический код

Поддерживается только диагностический код (**Diagnostic Code**) 0000.

Количество битов

Допускается любое значение в качестве тестового значения (**Test Value**).

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#8	Функциональный код
+2.0	diag_code	WORD	W#16#0000	Диагностический код
+4.0	test_value	WORD	W#16#A5C3	Тестовое значение

Пример сообщения-запроса для функции 08:

05H Адрес ведомого устройства
 08H Функциональный код
 00H Диагностический код (старший байт)
 00H Диагностический код (младший байт)
 A5H Тестовое значение (старший байт)
 C3H Тестовое значение (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 08:

05H Адрес ведомого устройства
 08H Функциональный код
 08H Функциональный код
 00H Диагностический код (старший байт)
 00H Диагностический код (младший байт)
 A5H Тестовое значение (старший байт)
 C3H Тестовое значение (младший байт)
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Ведомое устройство должно возвращать сообщение неизменным (эхо) в ведущее устройство. Ответное сообщение не вводится в RCV DB.

5.9 Функциональный код 11 - Fetch Communications Event Counter (считывание значения счетчика коммуникационных событий)

Функция

Данная функция служит для считывания слова состояния (**Status Word**) (2 байта) и значения счетчика коммуникационных событий (**Event Counter**) (2 байта) из памяти ведомого (slave) устройства. Значения данных параметров подробно представлены в описании протокола GOULD MODICON ("GOULD MODICON Modbus Protocol").

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#0B	Функциональный код

Пример сообщения-запроса для функции 11:

05H Адрес ведомого устройства
 0BH Функциональный код
 ххH Код CRC-проверки (младший байт)
 ххH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 11:

05H Адрес ведомого устройства
 0BH Функциональный код
 FEH Слово состояния (старший байт)
 DCH Слово состояния (младший байт)
 01H Счетчик событий (старший байт)
 08H Счетчик событий (младший байт)
 ххH Код CRC-проверки (младший байт)
 ххH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#FEDC	Слово состояния
+2.0	data[2]	WORD	W#16#0108	Счетчик событий

5.10 Функциональный код 12 - Fetch Communications Event Log (считывание протокола коммуникационных событий)

Функция

Данная функция служит для считывания:

- 2 байтов слова состояния ("**Status Word**")
- 2 байтов счетчика событий ("**Event Counter**")
- 2 байтов счетчика сообщений ("**Message Counter**")
- 64 событийных байтов ("**Event Bytes**")

Значения данных параметров подробно представлены в описании протокола GOULD MODICON ("GOULD MODICON Modbus Protocol").

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#0C	Функциональный код

Пример сообщения-запроса для функции 12:

05H Адрес ведомого устройства
 0CH Функциональный код
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 12:

05H Адрес ведомого устройства
 0CH Функциональный код
 46H Байтовый счетчик
 87H Слово состояния (старший байт)
 65H Слово состояния (младший байт)
 01H Счетчик событий (старший байт)
 08H Счетчик событий (младший байт)
 02H Счетчик сообщений (старший байт)
 20H Счетчик сообщений (младший байт)
 01H Событийный байт 1
 12H Событийный байт 2
 ::
 C2H Событийный байт 63
 D3H Событийный байт 64
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Область назначения для приема данных (RCV):

Структура области назначения для приема данных (RCV):

Адрес	Имя	Тип	Фактическое значение	Комментарий
+0.0	data[1]	WORD	W#16#8765	Слово состояния
+2.0	data[2]	WORD	W#16#0108	Счетчик событий
+4.0	data[3]	WORD	W#16#0220	Счетчик сообщений
+6.0	bytedata[1]	BYTE	B#16#01	Событийный байт
+7.0	bytedata[2]	BYTE	B#16#12	Событийный байт
:	:	:	:	:
+68.0	bytedata[63]	BYTE	B#16#C2	Событийный байт
+69.0	bytedata[64]	BYTE	B#16#D3	Событийный байт

5.11 Функциональный код 15 - Force Multiple Coils (назначения для нескольких катушек)

Функция

Данная функция служит для изменения состояния нескольких битов (общим числом до 2040) в памяти ведомого (slave) устройства.

Начальный адрес

Параметр "битовый начальный адрес" (**Bit Start Address**) драйвером не проверяется и передается неизменным.

Количество битов

Допускается любое количество битов от 1 до 2040 указывать в этом параметре (**amount of bits**) (число катушек - "coils"). Параметр показывает, какое количество битов должно быть перезаписано.

Параметр "счетчик байтов" ("**byte counter**") в сообщении-запросе генерируется драйвером исходя из заданного значения параметра "количество битов".

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	V#16#5	Адрес ведомого устройства
+1.0	function	BYTE	V#16#0F	Функциональный код
+2.0	bit_startadr	WORD	W#16#0058	Адрес бита
+4.0	bit_anzahl	INT	10	Состояние бита
+6.0	coil_state[1]	WORD	W#16#EFCD	Состояние катушки (Coil) 5FH..58H / 57H..50H

Пример сообщения-запроса для функции 15:

05H Адрес ведомого устройства
 0FH Функциональный код
 00H Адрес бита (старший байт)
 50H Адрес бита (младший байт)
 00H Количество битов (старший байт)
 0AH Количество битов (младший байт)
 02H Байтовый счетчик
 CDH Состояние катушки (Coil) 50H..57H
 EFH Состояние катушки (Coil) 58H..59H
 xxH Код CRC-проверки (младший байт)
 xxH Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 15:

05H	Адрес ведомого устройства
0FH	Функциональный код
00H	Адрес бита (старший байт)
50H	Адрес бита (младший байт)
00H	Количество битов (старший байт)
0AH	Количество битов (младший байт)
xxH	Код CRC-проверки (младший байт)
xxH	Код CRC-проверки (старший байт)

5.12 Функциональный код 16 - Preset Multiple Registers (предустановки для нескольких регистров)

Функция

Данная функция служит для одновременного перезаписывания значений нескольких регистров (общим числом до 127) в памяти ведомого (slave) устройства.

Начальный адрес

Параметр "начальный адрес регистра" (**Register Start Address**) драйвером не проверяется и передается неизменным.

Количество регистров

Допускается максимальное количество регистров 127 в этом параметре (1 регистр = 2 байта).

Параметр "счетчик байтов" ("**byte counter**") в сообщении-запросе генерируется драйвером исходя из заданного значения параметра "количество регистров".

Исходный блок DB для передачи (SEND)

Структура исходной области данных для передачи (SEND):

Адрес	Имя	Тип	Начальное значение	Комментарий
+0.0	address	BYTE	B#16#5	Адрес ведомого устройства
+1.0	function	BYTE	B#16#10	Функциональный код
+2.0	reg_startadr	WORD	W#16#0060	Начальный адрес регистра
+4.0	reg_anzahl	INT	3	Количество регистров
+6.0	reg_data[1]	WORD	W#16#41A1	Данные регистра
+8.0	reg_data[2]	WORD	W#16#42A2	Данные регистра
+10.0	reg_data[3]	WORD	W#16#43A3	Данные регистра

Пример сообщения-запроса для функции 04:

05H	Адрес ведомого устройства
10H	Функциональный код
00H	Начальный адрес регистра (старший байт)
60H	Начальный адрес регистра (младший байт)
00H	Число регистров (старший байт)
03H	Число регистров (младший байт)
06H	Байтовый счетчик
41H	Данные регистра <reg_data[1]> (старший байт)
A1H	Данные регистра <reg_data[1]> (младший байт)
42H	Данные регистра <reg_data[2]> (старший байт)
A2H	Данные регистра <reg_data[2]> (младший байт)
43H	Данные регистра <reg_data[3]> (старший байт)
A3H	Данные регистра <reg_data[3]> (младший байт)
xxH	Код CRC-проверки (младший байт)
xxH	Код CRC-проверки (старший байт)

Пример ответного сообщения ведомого устройства для функции 04:

05H	Адрес ведомого устройства
10H	Функциональный код
00H	Начальный адрес регистра (старший байт)
60H	Начальный адрес регистра (младший байт)
00H	Число регистров (старший байт)
03H	Число регистров (младший байт)
xxH	Код CRC-проверки (младший байт)
xxH	Код CRC-проверки (старший байт)

6 Интерфейс CPU - CP

6.1 Интерфейс CPU - CP для модуля CP 341

Используемые SFB

Передача данных между CP и CPU выполняется с помощью FB **P_SND_RK** и **P_RCV_RK**.

FB **P_SND_RK** активируется фронтом сигнала на входе **REQ**, если требуется вывод данных.

Готовность FB **P_RCV_RK** к операции приема данных обеспечивается при **EN_R = 1**.

Блок **P_RCV_RK** требуется при всех функциональных кодах считывания.

Параллельная обработка запросов

В каждый момент времени только один блок FB **P_SND_RK** и один блок FB **P_RCV_RK** могут быть вызваны CP 341 в пользовательской программе.

6.1.1 Передача данных от CPU к CP посредством P_SND_RK (CP 341)

Активация

Выполнение функционального кода MODBUS активируется с помощью SFB P_SND_RK при наличии фронта сигнала на входе REQ.
 Введите 'S' для операции пересылки SEND в параметр SF.
 Адрес логического модуля вводится в параметр LADDR.
 Вы должны ввести 'X' при использовании дополнительного блока данных как типа области данных партнерского CPU. Никакие значения не должны быть определены для других параметров партнерского CPU (R_...).
 Это обеспечивает пересылку драйверу функциональных кодов, необходимых для выполнения.

Исходные данные

Если активирован блок P_SND_RK, то информация об области исходных данных, определенная параметрами DB_NO и DBB_NO, пересылается CP с параметром длины LEN.

Индикация размера (Length)

Размер (длина) LEN зависит от используемого функционального кода.

Функциональный код	Длина LEN (в байтах)
01	6
02	6
03	6
04	6
05	6
06	6
07	2
08	6
11	2
12	2
15	>6
16	>6

Если количество передаваемых данных отличается от количества данных, указанного выше для отдельных функциональных кодов, то соответствующее задание не выполняется и P_SND_RK отбрасывается с фронтом сигнала на выходе ERROR (Ошибка).

Исходный блок DB (SEND)

Параметры, требуемые для выполнения функционального кода должны быть введены как пользовательские данные в область исходных данных.

Подробное описание каждого исходного блока DB для P_SND_RK может быть найдено в главе "Функциональные коды"

Генерация сообщений

Сообщения-запросы в адрес ведомых (slave) устройств генерируются в соответствии с передаваемыми P_SND_RK исходными данными и посылаются модулем CP.

Прежде всего драйвер проверяет длину LEN на соответствие значения длины, определенной в P_SND_RK, длине необходимой для данного кода. Если результат проверки отрицательный, то задание не выполняется и завершается с выдачей фронта сигнала на выход ERROR блока P_SND_RK.

При использовании других функциональных кодов, отличающихся от перечисленных, то задание не выполняется и завершается с выдачей фронта сигнала на выход ERROR блока P_SND_RK.

Элементы "байтовый счетчик" ("byte counter") и "CRC-проверка" ("CRC check") в запросе-сообщении генерируются модулем CP: их ввод в исходный блок DB для P_SND_RK не требуется.

Завершение задания для функций записи

Для функций записи активированный обработка блока P_SND_RK завершается после того, как без ошибок принимается ответное сообщение.

Об этом сообщается в пользовательскую SIMATIC-программу с помощью фронта сигнала на выходе DONE блока P_SND_RK.

Если возникли ошибки во время обмена сообщениями, или если ведомое (slave) устройство передает код ошибки в ответном сообщении, то об этом сообщается посредством выдачи фронта сигнала на выход ERROR.

Завершение задания для функций считывания

Для функций считывания активированный обработка блока P_SND_RK завершается после того, как без ошибок принимается ответное сообщение и завершается пересылка принятых данных в CPU.

Об этом сообщается в пользовательскую SIMATIC-программу с помощью фронта сигнала на выходе DONE блока P_SND_RK.

Это значит, что принятые данные уже доступны для CPU.

Если возникли ошибки во время обмена сообщениями, или если ведомое (slave) устройство передает код ошибки в ответном сообщении, то об этом сообщается посредством выдачи фронта сигнала на выход ERROR. В этом случае никакие принятые данные не пересылаются в CPU.

Ввод информации о выполнении задания в параметр STATUS

Для случаев, когда задание завершается с ошибкой в блоке P_SND_RK, то дополнительный код ошибки вводится в параметр состояния STATUS.

Точная причина ошибки может быть определена по этому коду ошибки.

6.1.2 Передача данных от CP к CPU посредством P_RCV_RK (CP 341)

Для всех функциональных кодов считывания требуется блок P_RCV_RK.

Область назначения данных

Если блок FB P_RCV_RK готов к приему, то он принимает входящие данные от CP и вводит их в область данных назначения, определенную параметрами DB_N0 и DBB_N0.

Индикация приема данных

Пользователь информируется о приеме данных в CPU с помощью фронта сигнала на выходе NDR.

При этом длина принятого блока данных отображается в параметре LEN.

Как указано выше, информация о завершении в целом задания Modbus может быть получена из параметра DONE блока FB P_SND_RK.

Обработка ошибок

В случае ошибок при приеме или при передаче, данные не передаются в CPU. В этом случае обработка блока P_SND_RK прекращается с выдачей фронта сигнала на выход ERROR.

Область назначения (целевая область) P_RCV_RK

Пользовательские данные, принятые с помощью функционального кода считывания, вводятся в целевую область данных блока P_RCV_RK.

Подробное описание каждого целевого блока DB для блока P_RCV_RK может быть найдено в главе "Функциональные коды"

Длина введенных в целевую область данных отображается в параметре LEN блока P_RCV_RK.

6.2 Интерфейс CPU - CP для модуля CP 441-2

Используемые SFB

Передача данных между CP и CPU выполняется с помощью SFB-блоков BSEND и BRCV.

SFB BSEND активируется фронтом сигнала на входе REQ, если требуется вывод данных. Готовность SFB BRCV к операции приема данных обеспечивается при EN_R=1.

Блок BRCV требуется при всех функциональных кодах считывания.

6.2.1 Передача данных от CPU к CP посредством BSEND (CP 441-2)

Коммуникационные соединения

Параметр ID описывает уникальное коммуникационное соединение с коммуникационным партнером. В этом параметре должен быть записан локальный ID из коммуникационных данных проекта.

Связи блока

Параметр R_ID описывает уникальные характеристики соединения блока.

Для рассматриваемого драйвера в R_ID может быть записано только значение от 0 до 255 для блока BSEND.

В случае задания считывания назначенный параметр для BRCV должен иметь то же значение R_ID, что и в BSEND.

Активация

Выполнение функционального кода MODBUS активируется с помощью SFB BSEND при наличии фронта сигнала на входе REQ.

Это обеспечивает пересылку драйверу функциональных кодов, необходимых для выполнения.

Исходные данные

Если активирован блок BSEND, то информация об области исходных данных, определенная параметром SD_1, пересылается CP с параметром длины LEN.

Индикация размера (Length)

Размер (длина) LEN зависит от используемого функционального кода.

Функциональный код	Длина LEN (в байтах)
01	6
02	6
03	6
04	6
05	6
06	6
07	2
08	6
11	2
12	2
15	>6
16	>6

Если количество передаваемых данных отличается от количества данных, указанного в таблице для отдельных функциональных кодов, то соответствующее задание не выполняется и BSEND отбрасывается с фронтом сигнала на выходе ERROR (Ошибка).

Исходный блок DB (BSEND)

Параметры, требуемые для выполнения функционального кода должны быть введены как пользовательские данные в область исходных данных.

Подробное описание каждого исходного блока DB для BSEND может быть найдено в главе "Функциональные коды"

Генерация сообщений

Сообщения-запросы в адрес ведомых (slave) устройств генерируются в соответствии с передаваемыми BSEND исходными данными и посылаются модулем CP.

Прежде всего драйвер проверяет длину LEN на соответствие значения длины, определенной в BSEND, длине необходимой для данного кода. Если результат проверки отрицательный, то задание не выполняется и завершается с выдачей фронта сигнала на выход ERROR блока BSEND.

При использовании других функциональных кодов, отличающихся от перечисленных, то задание не выполняется и завершается с выдачей фронта сигнала на выход ERROR блока BSEND.

Элементы "байтовый счетчик" ("byte counter") и "CRC-проверка" ("CRC check") в запросе-сообщении генерируются модулем CP: их ввод в исходный блок DB для BSEND не требуется.

Завершение задания для функций записи

Для функций записи активированный блок BSEND завершается после того, как без ошибок принимается ответное сообщение.

Об этом сообщается в пользовательскую SIMATIC-программу с помощью фронта сигнала на выходе DONE блока BSEND.

Если возникли ошибки во время обмена сообщениями, или если ведомое (slave) устройство передает код ошибки в ответном сообщении, то об этом сообщается посредством выдачи фронта сигнала на выход ERROR.

Завершение задания для функций считывания

Для функций считывания активированный блок BSEND завершается после того, как без ошибок принимается ответное сообщение и завершается пересылка принятых данных в CPU.

Об этом сообщается в пользовательскую SIMATIC-программу с помощью фронта сигнала на выходе DONE блока BSEND.

Это значит, что принятые данные уже доступны для CPU.

Если возникли ошибки во время обмена сообщениями, или если ведомое (slave) устройство передает код ошибки в ответном сообщении, то об этом сообщается посредством выдачи фронта сигнала на выход ERROR. В этом случае никакие принятые данные не пересылаются в CPU.

Ввод информации о выполнении задания в параметр SYSTAT

Для случаев, когда задание завершается с ошибкой в блоке BSEND, то дополнительный код ошибки вводится в параметр состояния SYSTAT.

Точная причина ошибки может быть определена по этому коду ошибки.

6.2.2 Передача данных от CP к CPU посредством BRCV (CP 441-2)

Коммуникационные соединения

Параметр ID описывает уникальное коммуникационное соединение с коммуникационным партнером. В этом параметре должен быть записан локальный ID из коммуникационных данных проекта.

Связи блока

Параметр R_ID описывает уникальные характеристики соединения блока. Для всех функциональных кодов считывания требуется блок BRCV. Для рассматриваемого драйвера в R_ID может быть записано только значение от 0 до 255 для блока BRCV. В случае задания считывания назначенный параметр для BRCV должен иметь то же значение R_ID, что и в BSEND.

Таким образом Вы можете запрограммировать обработку нескольких пар блоков BSEND / BRCV в своей программе. Ответные сообщения, принятые от ведомого Modbus-устройства, будут при этом храниться в различных целевых областях, в зависимости от используемого в задании значения R_ID.

Область назначения данных

Если блок SFB BRCV готов к приему, то он принимает входящие данные от CP и вводит их в область данных назначения, определенную в параметре RD_1. Это значит, что целевая область данных - переменная.

Индикация приема данных

Пользователь информируется о приеме данных в CPU с помощью фронта сигнала на выходе NDR.

При этом длина принятого блока данных отображается в параметре LEN. Как указано выше, информация о завершении в целом задания Modbus может быть получена из параметра DONE блока SFB BSEND.

Обработка ошибок

В случае ошибок при приеме или при передаче, данные не передаются в CPU. В этом случае обработка блока BSEND прекращается с выдачей фронта сигнала на выход ERROR.

Область назначения (целевая область) BRCV

Пользовательские данные, принятые с помощью функционального кода считывания, вводятся в целевую область данных блока BRCV.

Подробное описание каждого целевого блока DB для блока BRCV может быть найдено в главе "Функциональные коды"

Длина введенных в целевую область данных отображается в параметре LEN блока BRCV.

7 Диагностика драйвера

Функции диагностики

Функции диагностики модуля CP позволяют оперативно локализовать возникающие ошибки. Пользователю доступны следующие диагностические функции:

- диагностика посредством элементов индикации модуля CP
- диагностика посредством выходного параметра "состояний" STATUS функциональных блоков
- диагностика посредством сообщений об ошибках в области SYSTAT (только для CP 441-2)
- диагностика посредством диагностического буфера CP

Диагностика посредством элементов индикации модуля (светодиоды)

Элементы индикации показывают рабочее состояние и/или возможные ошибочные состояния модуля CP. Элементы индикации дают предварительную обзорную информацию внутренних или внешних ошибок, а также ошибок интерфейса.

Диагностика посредством выходного параметра "состояний" STATUS функциональных блоков

Каждый функциональный блок / системный функциональный блок имеет выходной параметр STATUS для диагностики ошибок в модуле CP. Считывая значения этого параметра, пользователь может получить информацию по ошибкам, которые возникают в соединениях. Выходной параметр STATUS может быть проверен в пользовательской программе.

Диагностика посредством сообщений об ошибках в области SYSTAT (только для CP 441-2)

Область SYSTAT в модуле CP 441-2 предназначена для хранения подробной информации о диагностических событиях и об обнаруженных ошибках. Вы можете считывать эту информацию из пользовательской программы, запрограммировав системный функциональный блок SYSTAT.

Диагностика посредством диагностического буфера CP

Все ошибки и диагностические события, описанные в разделе 7.3, вводятся в диагностический буфер CP. В руководстве по CP Вы можете найти информацию о том, как считывать записи из диагностического буфера.

7.1 Средства диагностики модуля CP 341

7.1.1 Диагностика с использованием элементов индикации CP 341

Введение

Элементы индикации на передней панели CP 341 выдают информацию о его состоянии. Имеются следующие функции индикации:

- **Индикатор групповых сообщений (Group alarm)**
 - SF (светится красным при отказе или при повторной параметризации)
- **Специальные индикаторы (Special)**
 - TXD (светится зеленым при передаче; загорается, когда CP 341 передает пользовательские данные через интерфейс)
 - RXD (светится зеленым при приеме; загорается, когда CP 341 принимает пользовательские данные через интерфейс)

Индикатор групповых сообщений (Group alarm) SF

Индикатор групповых сообщений (Group alarm) SF всегда загорается после включения питания (POWER ON) и выключается после инициализации. Если данные параметризации были сгенерированы для CP 341, светодиод SF вновь загорается на короткое время во время повторной параметризации.

Индикатор групповых сообщений (Group alarm) SF загорается при следующих событиях:

- Отказ оборудования
- Сбой микропрограммы
- Ошибка при параметризации или отсутствие параметризации
- Обрыв линии (размыкание сигнального кабеля между CP 341 и коммуникационным партнером)

7.1.2 Диагностические сообщения функциональных блоков CP 341

Введение

Каждый функциональный блок имеет параметр STATUS для диагностики ошибок. Номера сообщений об одних и тех же ошибках STATUS всегда имеют одинаковые значения, независимо от конкретного блока.

Схема нумерации классов и номеров событий:

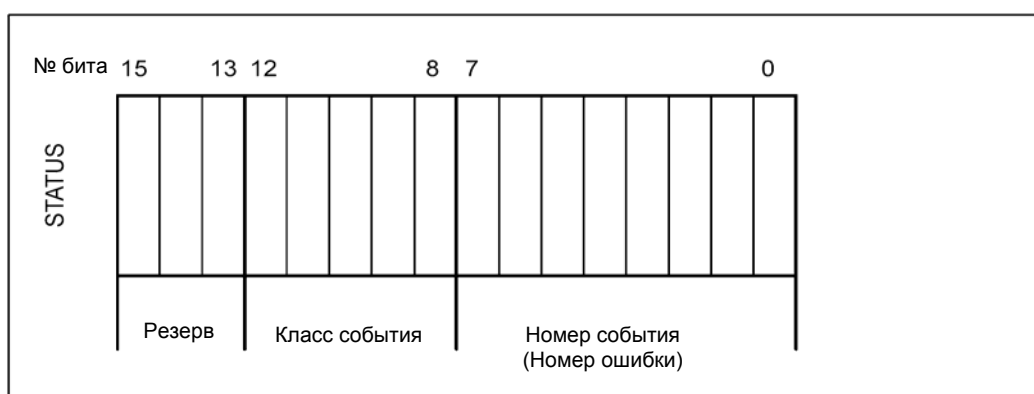


Рис. 7.1 Структура параметра STATUS

В разделе 7.3 представлен список отдельных диагностических событий / ошибок.

7.2 Средства диагностики модуля CP 441-2

7.2.1 Диагностика с использованием элементов индикации CP 441-2

Функции индикации

Элементы индикации на передней панели модуля CP 441-2 выдают информацию о его состоянии. Имеются следующие функции индикации:

- **Индикатор групповых сообщений (Group Error Displays)**
 - INTF (индикатор внутренней ошибки)
 - EXTf (индикатор внешней ошибки)
- **Специальные индикаторы (Special Displays)**
 - TXD (индикатор выполнения задания BSEND в CP)
 - RXD (индикатор передачи принятых CP данных в CPU)
- **Индикатор групповых сообщений (Group Error Displays)**
 - FAULT (индикатор ошибки интерфейса)

Индикация сообщений об ошибках

Сообщение об ошибке	Описание ошибки / диагностическое событие	Действия
Светодиод INTF светится	CP сообщает о внутренней ошибке; ошибка аппаратного или программного обеспечения	Запрограммируйте SFB STATUS для считывания подробной информации
Светодиод EXTf светится	CP 441-2 сообщает о внешней ошибке; например, об обрыве входных линий связи (BREAK)	Запрограммируйте SFB STATUS для считывания подробной информации
Светодиод FAULT не светится	Интерфейс готов к работе или интерфейсный submodule не вставлен в модуль	-
Светодиод FAULT редко мигает	Интерфейс инициализирован и готов к работе, но невозможно соединение посредством системной шины S7-400	Проверьте общую конфигурацию и коммуникацию данных в конфигурации проекта
Светодиод FAULT часто мигает	Некорректен параметр или сбой интерфейсного submodule (не совпадают параметры submodule и интерфейса)	Проверьте установки параметров в утилите для назначения параметров или в интерфейсном submodule
Светодиод FAULT светится	Нет доступа к параметрам интерфейса или существенная ошибка submodule (ошибка оборудования)	Задайте параметры с утилитой для назначения параметров или проверьте интерфейсный submodule

7.2.2 Диагностические сообщения системных функциональных блоков CP 441-2

Введение

В каждом системном функциональном блоке имеется параметр STATUS, предназначенный для диагностики ошибочных состояний. Номера сообщений об одних и тех же ошибках STATUS всегда имеют одинаковые значения, независимо от используемого системного блока. Наиболее важные для модуля CP STATUS-сообщения описываются в следующей ниже таблице. Вы можете найти полное описание STATUS-сообщений в справочном руководстве *"Системное программное обеспечение для S7-300/400, Системные и стандартные функции"* (*"System Software for S7-300/400, System and Standard Functions"*).

Сообщения в выходном параметре STATUS блоков SFB

STATUS	Описание
0	Нет ошибок
1	Проблемы в соединении CP - CPU
2	Негативное квитирование, функция не может быть выполнена (например, партнер не отвечает или выдает негативное квитирование)
3	Неизвестный идентификатор R-ID; устройство недоступно
4	Некорректны число данных или отдельные типы данных
5	Принят запрос на выполнение сброса (RESET)
6	Удаленный блок недоступен (disabled)
7	Удаленный партнер находится в некорректном состоянии
8	Нет доступа к удаленному объекту; ошибка доступа в сервере (GET/PUT)
9	Предупреждение о переполнении (ERROR = 0): принятые данные были перезаписаны новыми значениями
10	Нет доступа к локальной памяти пользователя (например, удален DB)
11	Предупреждение (ERROR = 0): новое задание неактивно, так как пока не завершено предыдущее.
12	Экземпляр не совместим с системным вызовом; вызван не экземплярный, а обычный блок DB
13	Ошибка в описании формата
14	Указанная связь данных (относящаяся к приложению) не существует. Неизвестный ID (Вы должны задать локальный ID из данных коммуникаций в конфигурации проекта)
15	Связь данных сгенерирована в соответствии с указанным ID.
16	Связь данных не генерируется из-за недостатка ресурсов.

Отображение и проверка информации в выходном параметре STATUS

Информация в параметре STATUS системного функционального блока может быть отображена и считана с использованием таблицы переменных STEP 7.

Примечание

Считывание информации из области SYSTAT с помощью задания STATUS предоставит Вам подробную информацию по ошибкам / событиям, которые имели место во время коммуникационных связей CP, соответствующего CPU и подключенного коммуникационного партнера.

7.2.3 Диагностика посредством области для сообщений об ошибках SYSTAT

Введение

Область для сообщений об ошибках SYSTAT - это область данных в CP 441-2, в которую вводятся все распознанные CP ошибки / диагностические события в деталях. Область SYSTAT содержит поля для информации о 6 событиях для каждого интерфейса, а также информацию о рабочих состояниях CP и состоянии собственно области SYSTAT.

Считывание информации из области SYSTAT

Информация из области SYSTAT может быть считана с помощью SFB STATUS. Необходимые коммуникации данных, для которых текущие события должны быть считаны, должны быть введены в ID параметра (Parameter ID). 16 байтов диагностических данных передаются в связанные данные (data link) на выход LOCAL; параметры PHYS и LOG не используются при PtP-коммуникациях связанных данных (data link).

Примечание

Благодаря тому, что задание STATUS выполняется асинхронно по отношению к другим заданиям в сети, не предоставляется возможным разместить SFB для заданного R_ID. Это означает, что даже если можно отобразить информацию о произошедших ошибках в связанных данных (data link), но невозможно вычислить, какой из вызовов SFB стал их причиной.

Структура области SYSTAT

Информация о первых шести распознанных CP ошибках / событиях вводится в область SYSTAT. Последующие ошибки / события могут быть введены в область SYSTAT, только после того как область SYSTAT будет очищена.

Ошибки / события вводятся в параметр LOCAL следующим образом:

- Байт 0 Рабочее состояние CP (02H для RUN, 05H для отказа)
- Байт 1 Зарезервирован
- Байт 2 Бит 0 - F ошибка введена в SYSTAT
Бит 1 - U переполнение (Error overflow)
Бит 2 - B обрыв (BREAK)
- Байт 3 Зарезервирован
- Байты 4/5 Событие 1
- Байты 6/7 Событие 2
- Байты 8/9 Событие 3
- Байты 10/11 Событие 4
- Байты 12/13 Событие 5
- Байты 14/15 Событие 6

Очистка (Deleting) области SYSTAT

После того, как SYSTAT-информация была считана с помощью SFB STATUS, все SYSTAT-сообщения автоматически удаляются.

Схема нумерации

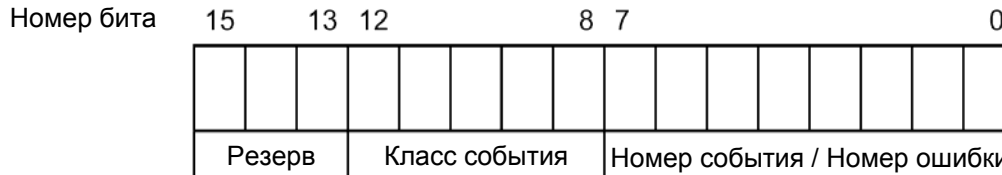


Рис. 7.2 Схема нумерации событий в области SYSTAT

Классификация классов и номеров событий представлена в следующих разделах, а также в руководстве "PtP-коммуникации CP 441-2" ("CP 441-2 Point-to-Point Communication").

Примечание

В отличие от стандартных драйверов классы и номера событий для загружаемых драйверов для SYSTAT-области частично изменены. В следующих разделах представлены все **измененные классы и номера событий** для их использования с загружаемыми драйверами. Если событие не представлено в данном руководстве, Вы можете рассматривать его соответствующим стандартному применению и найти в руководстве по CP 441-2.

7.3 Таблицы событий (Events) / ошибок (Errors)

Классы событий

Определены следующие классы событий:

Классы событий	Описание	Справочник
1	Ошибки аппаратуры CP	Руководство по CP
2	Ошибки при инициализации	Руководство по CP
3	Ошибки при параметризации PBK	Руководство по CP
4	Ошибки при обмене данных CP - CPU	Руководство по CP
5	Ошибки в работе CPU	Руководство по CP / Описание драйвера
6	Ошибки при обработке задания партнера	Руководство по CP
7	Ошибки при передаче	Руководство по CP
8	Ошибки при приеме	Описание драйвера
9	Сообщение с кодом ошибки принято от коммуникационного партнера	Не используется
10	Ошибки распознаны CP в ответном сообщении партнера	Не используется
14	Общие ошибки обработки загружаемого драйвера	Описание драйвера

7.3.1 Коды ошибок в области SYSTAT для "CPU Job Errors" ("Ошибки в работе CPU")

Класс события 5 (05H): "Ошибки в работе CPU" ("CPU Job Errors ")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(05)18H	24	Transmission length during transmission is too large (> 4 KB), or transmission length for SEND is too small. (Для передачи задан размер > 4 kb - слишком большой или слишком малый для параметров интерфейса)	Проверьте параметр LEN для функции SEND.

7.3.2 Номера ошибок в области SYSTAT для "Receive Errors" ("Ошибки при приеме")

Класс события 8 (08H): "Ошибки при приеме CPU" ("CPU Receive Errors ")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(08)06H	6	Character delay time (ZVZ) exceeded. (Превышено время ожидания (задержки) символа)	Устраните ошибку в партнерском устройстве или помехи в линии передачи.
(08)0CH	12	Transmission error (parity error, overflow error, stop bit error (frame)) recognized in a character. (Обнаружена ошибка при передаче - в фрейме символа (ошибка четности, ошибка переполнения, ошибка стоповых битов)	Проверьте наличие помех в линии передачи. Если требуется, измените структуру системы и/или прокладку кабелей. Проверьте, одинаковы ли в CP и в партнере параметры протокола: скорость передачи, количество битов данных, четность, количество стоповых битов.
(08)0DH	13	BREAK Receiving line to partner device is interrupted. (Обрыв - нарушена целостность приемной линии связи с партнерским устройством)	Восстановите соединение между устройствами или включите устройство-партнер. При использовании TTY-соединения, проверьте ток в линии в состоянии покоя. При использовании RS422 / 485 (X27) - соединения, проверьте (измените, если требуется) назначение соединительных контактов 2-х проводной приемной линии R(A), R(B).

Класс события 8 (08H): "Ошибки при приеме CPU" ("CPU Receive Errors ")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(08)30H	48	A request message has been sent and the reply monitoring time has elapsed without the start of a reply message being recognized. (После отправки сообщения-запроса было превышено время ожидания ответного сообщения - не было зафиксировано начала передачи ответного сообщения)	Проверьте линию передачи (можно использовать интерфейсный анализатор). Проверьте, одинаковы ли в CP и в партнере параметры: скорость передачи, число битов данных, четность, количество стоповых битов. Проверьте с PtP-Param достаточно ли велико значение времени ожидания. Проверьте адрес партнера.
(08)31H	49	The first character in the reply message from the slave is different from the slave address sent in the request message (for operating mode "Normal"). (Первый символ в ответном сообщении партнера отличается от адреса в запросе (в нормальном рабочем режиме)).	Ответ пришел не от того партнера, который ожидается. Проверьте линию передачи (для этого может быть использован интерфейсный анализатор).
(08)32H	50	Overflow of receive buffer in CP during reception of the reply message. (Переполнение приемного буфера CP во время приема ответного сообщения)	Проверьте параметры протокола, заданные в партнерском устройстве.

7.3.3 Номера ошибок в области SYSTAT для "General Processing Errors" ("Общие ошибки обработки")

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)01H	1	Error during initialization of the driver-specific SCC process (Ошибка при инициализации определенного драйвером SCC-процесса)	Переопределите параметры драйвера и перезагрузите его
(0E)02H	2	Error during startup of driver: Wrong SCC process active (SCC driver). The driver cannot function with this SCC driver. (Ошибка при запуске драйвера: активен не тот процесс (SCC-драйвер). Драйвер не работает с этим SCC-драйвером)	Переопределите параметры драйвера и перезагрузите его
(0E)03H	3	Error during startup of driver: Wrong data transfer process active (interface to SFBs). The driver cannot function with this data transfer process. (Ошибка при запуске драйвера: активен не тот процесс передачи (интерфейс с SFC). Драйвер не работает с этим процессом передачи)	Переопределите параметры драйвера и перезагрузите его
(0E)04H	4	Error during startup of driver: Illegal interface submodule. The driver cannot run with the parameterized interface submodule. (Ошибка при запуске драйвера: недопустимый интерфейсный submodule. Драйвер не работает с этим submodule)	Проверьте и исправьте назначенные параметры
(0E)05H	5	Error with driver dongle: No dongle plugged in, or inserted dongle is faulty. The driver is not ready to run. (Ошибка из-за защитного ключа-заглушки драйвера: ключ-заглушка отсутствует или неисправен. Драйвер не готов к работе)	Проверьте наличие защитного ключа-заглушки для драйвера в CP. Замените ключ-заглушку на исправный.
(0E)06H	6	Error with driver dongle: The dongle has no valid contents. The driver is not ready to run. (Ошибка из-за защитного ключа-заглушки драйвера: ключ-заглушка имеет некорректное содержание. Драйвер не готов к работе)	Получите подходящий защитный ключ-заглушку загружаемого драйвера для CP из представительства Siemens.
:	:		

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)10H	16	Internal error procedure: default branch in procedure automatic device. (Внутренняя ошибка процедуры: ошибка ветви процедуры в автоматическом устройстве)	Перезапустите CP (включите электропитание)
(0E)11H	17	Internal error procedure: default branch for procedure status Send / Receive. (Внутренняя ошибка процедуры: ошибка ветви процедуры для состояния передачи/приема)	Перезапустите CP (включите электропитание)
(0E)12H	18	Internal error active automatic device: default branch. (Внутренняя ошибка активного автоматич. устройства: ошибка ветви)	Перезапустите CP (включите электропитание)
(0E)13H	19	Internal error passive automatic device: default branch. (Внутренняя ошибка пассивного автоматич. устройства: ошибка ветви)	Перезапустите CP (включите электропитание)
(0E)20H	32	For this data link, the amount of data bits must be set to 8. The driver is not ready to run. (В этих данных связи количество битов данных должно быть 8. Драйвер не готов к работе)	Проверьте и исправьте назначенные параметры драйвера
(0E)21H	33	The multiplication factor set for the character delay time is not within the value range of 1 to 10. The driver is operating with a default setting of 1. (Множитель для времени ожидания символа вне пределов 1...10. Драйвер работает со значением по умолчанию 1)	Проверьте и исправьте назначенные параметры драйвера
(0E)22H	34	The operating mode set for the driver is illegal. "Normal operation" or "Interference Suppression" must be specified. The driver is not ready to run. (Рабочий режим для драйвера недопустим: должен быть задан "нормальный" режим или режим "с подавлением помех". Драйвер не готов к работе)	Проверьте и исправьте назначенные параметры драйвера
(0E)23H	35	An illegal value for the reply monitoring time has been set: Valid values are 5 to 65500ms. The driver is not ready to run. (Задано неверное значение для времени мониторинга ответного сообщения: значение должно быть от 5 до 65500 мс. Драйвер не готов к работе)	Проверьте и исправьте назначенные параметры драйвера
:	:	:	:

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)2EH	46	An error occurred when reading the interface parameter file. The driver is not ready to run. (Ошибка при считывании файла параметров интерфейса. Драйвер не готов к работе)	Перезапустите CP (включите электропитание)

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors <CPU - CP>")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)30H	48	Internal error during data transfer to CPU: Unexpected acknowledgment Passive. (Внутренняя ошибка при передаче в CPU: неожиданный ответ "Пассивное состояние")	Игнорируйте, если это состояние не является постоянным
(0E)31H	49	Timeout during data transfer to CPU. (Прерывание при передаче данных в CPU)	Проверьте интерфейс CP - CPU
(0E)32H	50	Error occurred during data transfer to CPU with RCV: Exact failure reason (detailed error) is in SYSTAT before this entry. (Ошибка при передаче данных в CPU посредством RCV: Точная причина ошибки (подробно) в области SYSTAT перед данным вводом)	Проверьте интерфейс CP - CPU
(0E)33H	51	Internal error during data transfer to CPU: Illegal status of automatic device. (Внутренняя ошибка при передаче данных в CPU: недопустимое состояние автоматического устройства)	Проверьте интерфейс CP - CPU
:	:	:	:
(0E)3CH	60	Illegal job with this driver. (Недопустимое задание для драйвера)	Допустимы только задания: SFB BSEND, RCV, STATUS (только для CP 441-2).

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors <BSEND>")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)40H	64	Value specified for parameter LEN at SFB SEND too small. (Значение для длины данных LEN в SFB слишком мало)	Минимальное значение длины = 2 байта.
(0E)41H	65	Value specified for parameter LEN at SFB SEND too small. A greater length is required for the transferred function code. (Значение для длины данных LEN в SFB слишком мало. Длина должна быть больше для данного функционального кода)	Минимальное значение длины для этого функционального кода = 6 байтов.
(0E)42H	66	Transferred function code is illegal. (Некорректный функциональный код)	Список допустимых кодов - в главе "Функциональные коды"
(0E)43H	67	Slave Address 0 (= Broadcast) not permitted with this function code. (Адрес ведомого устройства = 0 (режим рассылки) недопустим с этим функциональным кодом)	Используйте адрес ведомого устройства 0 только с допустимыми функциональными кодами
(0E)44H	68	The value of the transferred parameter "Amount of Bits" is not within the range 1 to 2040. (Значение параметра "количество битов" за пределами диапазона 1...2040)	Задайте значение параметра из диапазона 1...2040
(0E)45H	69	The value of the transferred parameter "Amount of Registers" is not within the range 1 to 127. (Значение параметра "число регистров" за пределами диапазона 1...127)	Задайте значение параметра из диапазона 1...127.
(0E)46H	70	Function codes 15 or 16: The values of the transferred parameters "Amount of Bits" and/or "Amount of Registers" are not within the range 1 to 2040 and/or 1 to 127. (Функциональный код 15 или 16: Значения параметров "количество битов" и/или "число регистров" не в диапазоне 1...2040 и/или 1...127)	Задайте значения соответствующих параметров из диапазонов 1...2040 / 1...127

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors <BSEND>")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)47H	71	Function codes 15 or 16: The parameter LEN for SFB BSEND does not correspond to the transferred parameters "Amount of Bits" and/or "Amount of Registers." Parameter LEN is too small. (Функциональный код 15 или 16: Значение параметра LEN для SFB BSEND не соответствует значениям переданных параметров "количество битов" и/или "число регистров". Значение длины данных LEN слишком мало)	Увеличивайте значение LEN для SEND пока не будет передаваться в CP достаточное количество данных пользователя. Больше количество данных должно передаваться в CP из-за значений "Amount of Bits" и/или "Amount of Registers."
(0E)48H	72	Function code 5: The code specified in SEND source DB for "Set Bit" (FF00H) or "Delete Bit" (0000H) is wrong. (Функциональный код 5: неверен код, заданный в исходном (SEND) DB для "Установить бит" (FF00H) или "сбросить бит" (0000H))	Единственно здесь разрешенный код: 0000H.
(0E)49H	73	Function code 8: The code specified in SEND source DB for "Diagnostic Code" is wrong. (Функциональный код 8: неверен код, заданный в исходном (SEND) DB для "кода диагностики" (0000H))	Единственно здесь разрешенный код: "Diagnostic Code" 0000H.
:	:	:	:
(0E)4FH	79	CP 441: The R_ID specified for SFB SEND is illegal with this driver. (Недопустимое задание R_ID для SFB SEND для данного драйвера)	Используйте для R_ID только значения из диапазона 0...255 (00000000-000000FFH)
		CP 341: The R_TYP specified for SFB SEND RK is illegal with this driver. (Недопустимое задание R_TYP для SFB SEND RK для данного драйвера)	"X" вводится в R_TYP.

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors <BSEND>")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)50H	80	Slave address incorrect: The received slave address is different from the sent slave address. (Некорректный адрес ведомого (Slave) устройства: Принятый адрес ведомого устройства отличается от запрошенного)	Отвечает не то ведомое (Slave) устройство. Проверьте линию передачи (возможно с интерфейсным анализатором)
(0E)51H	81	Function code incorrect: The function code received in the reply message is different from the sent function code. (Некорректный функциональный код: Функциональный код в ответном сообщении отличается от запрошенного функционального кода)	Проверьте ведомое (Slave) устройство
(0E)52H	82	Byte Underflow: Amount of characters received is less than should have resulted from the byte counter of the reply message, or is less than expected with this function code. (Количество принятых символов меньше, чем указано в байтовом счетчике ответного сообщения или меньше, чем ожидается для данного кода)	Проверьте ведомое (Slave) устройство
(0E)53H	83	Byte Overflow: Amount of characters received is more than should have resulted from the byte counter of the reply message, or is more than expected with this function code. (Количество принятых символов больше, чем указано в байтовом счетчике ответного сообщения или больше, чем ожидается для данного кода)	Проверьте ведомое (Slave) устройство
(0E)54H	84	Byte counter wrong: The byte counter received in the reply message is too small. (Ошибка байтового счетчика: Количество байтов, указанное в байтовом счетчике ответного сообщения слишком мало)	Проверьте ведомое (Slave) устройство
(0E)55H	85	Byte counter wrong: The byte counter received in the reply message is wrong. (Ошибка байтового счетчика: Количество байтов, указанное в байтовом счетчике ответного сообщения некорректно)	Проверьте ведомое (Slave) устройство

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors <BSEND>")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)56H	86	Echo wrong: The data of the reply message (amount of bits, ...) echoed from the slave are different from the data sent in the request message. (Ошибка в ответном тестовом коде (ошибка эха): Данные ответного сообщения от ведомого устройства отличаются от исходного кода)	Проверьте ведомое (Slave) устройство
(0E)57H	87	CRC Check incorrect: An error has occurred on CRC 16 checksum of the reply message from slave. (Несовпадение при проверке четности)	Проверьте ведомое (Slave) устройство

Класс события 14 (0EH): "Загружаемый драйвер - Общие ошибки обработки" ("Loadable Driver - General Processing Errors <ERROR code>")			
Класс / номер события (HEX)	Класс / номер события (десятич.)	Текст события	Действия
(0E)61H	97	Reply message with Exception Code 01: Illegal Function (Ответное сообщение с кодом ошибки 01: Некорректная функция)	См. руководство по ведомому (Slave) устройству
(0E)62H	98	Reply message with Exception Code 02: Illegal Data Address (Ответное сообщение с кодом ошибки 02: Некорректный адрес данных)	См. руководство по ведомому (Slave) устройству
(0E)63H	99	Reply message with Exception Code 03: Illegal Data Value (Ответное сообщение с кодом ошибки 03: Некорректное значение данных)	См. руководство по ведомому (Slave) устройству
(0E)64H	100	Reply message with Exception Code 04: Failure in associated device (Ответное сообщение с кодом ошибки 04: Ошибка в связанном устройстве)	См. руководство по ведомому (Slave) устройству
(0E)65H	101	Reply message with Exception Code 05: Acknowledge (Ответное сообщение с кодом ошибки 05: Квитирование)	См. руководство по ведомому (Slave) устройству
(0E)66H	102	Reply message with Exception Code 06: Busy, Rejected message (Ответное сообщение с кодом ошибки 06: Занято, Сообщение отбрасывается)	См. руководство по ведомому (Slave) устройству
(0E)67H	103	Reply message with Exception Code 07: Negative Acknowledgment (Ответное сообщение с кодом ошибки 07: Квитирование)	См. руководство по ведомому (Slave) устройству

8 Пример применения

8.1 Пример для модуля CP 341

Общая информация

В представленном ниже простом примере программирования показано использование блоков FB P_SND_RK и P_RCV_RK.

После инсталляции вспомогательной программы Modbus master файл с примером программы сохраняется в STEP 7 в директории EXAMPLES с именем Modma.

S7-программа предназначена только для информирования пользователя и не должна восприниматься как готовое решение для конфигурирования конкретной установки.

Для иллюстрации базовой структуры мы намеренно сохранили простоту и не применяли символьное отображение.

8.1.1 Используемые блоки

Используемые блоки

В примере программирования используются следующие блоки:

Блок	Символ	Комментарий
OB 1	Cycle Execution	Циклическое выполнение программы
OB 100	Complete Restart	Стартовый блок (Startup OB) для перезапуска
FC 10	Initiation	Функция FC для блока (Startup OB)
FC 21	Execute Send Jobs	FC, вызывающий SFB P_SND_RK
FC 23	Execute Receive Jobs	FC, вызывающий SFB P_RCV_RK
DB 50	IDB_P_SND_RK	Экземплярный DB для P_SND_RK
DB 70	IDB_P_RCV_RK	Экземплярный DB для P_RCV_RK
DB 40	Work DB Send	Рабочий (Work) DB для FC21 и P_SND_RK
DB 41	Work DB Receive	Рабочий (Work) DB для FC23 и P_RCV_RK
DB 42	SOURCE_DB	Исходный DB P_SND_RK для передачи
DB 43	DESTINATION_DB	Целевой DB P_RCV_RK для приема

Используемые данные

В примере программирования используются следующие операнды (меркеры, биты данных, слова данных):

Блок	Символ	Комментарий
M 120.7		Бит запуска (Trigger bit) для выполнения задания P_SND_RK
DB40.DBX 0.0		Управляющий параметр запроса REQ: для активации задания P_SND_RK
DB40.DBX 0.1		Управляющий параметр сброса Reset: для прерывания задания P_SND_RK
DB40.DBX 0.4		Параметр индикации состояния DONE: показывает, что задание P_SND_RK завершено без ошибок
DB40.DBX 0.5		Параметр индикации состояния ERROR: показывает, что задание P_SND_RK завершено с ошибкой
DB40.DBW 6		Счетчик успешных завершений (Success) задания P_SND_RK
DB40.DBW 8		Счетчик завершений с ошибкой (Error) задания P_SND_RK
DB40.DBW 10		Длина LEN исходных данных для задания P_SND_RK передаваемых в CP (в байтах)
DB40.DBW 12		Состояние (STATUS) в P_SND_RK
DB40.DBW 14		Сохраненное состояние (Stored STATUS) в P_SND_RK
DB41.DBX 0.0		Управляющий параметр EN_R: "P_RCV_RK готов к приему"
DB41.DBX 0.4		Параметр индикации состояния NDR: показывает, что текущий P_RCV_RK принял новые данные от CP
DB41.DBX 0.5		Параметр индикации состояния ERROR: показывает, что текущий P_RCV_RK обработан с ошибкой
DB41.DBW 4		Сохраненная длина (Stored length) LEN в P_RCV_RK
DB41.DBW 6		Счетчик успешных завершений (Success) задания P_RCV_RK
DB41.DBW 8		Счетчик завершений с ошибкой (Error) задания P_RCV_RK
DB41.DBW 10		Длина LEN целевой области данных для задания P_RCV_RK, принятая CP (в байтах)
DB41.DBW 12		Состояние (STATUS) в P_RCV_RK
DB41.DBW 14		Сохраненное состояние (Stored STATUS) в P_RCV_RK

8.1.2 Описание программы

Общая информация

Пример программы содержит:

- Блок запуска OB100, FC10
- Циклически выполняемую часть программы, вызываемую в OB1
- Функциональный блок FC21 для передачи данных от CPU в CP (Send)
- FC23 для приема данных из CP в CPU

Параметры для запрограммированных системных функциональных блоков P_SND_RK, P_RCV_RK сохраняются в рабочих (work) DB-блоках: DB40 и DB41.

Данные для передачи (область исходных данных "SEND source area") располагаются в DB42.

Данные, принятые от коммуникационного партнера вводятся в DB43 (целевая область для принятых данных "RCV destination area").

Задание P_SND_RK

Задание P_SND_RK может быть активировано в циклической части программы при установке меркера ("memory bit") M 120.7 (например, посредством переменной).

Данные длины LEN, содержащиеся в исходной области (P_SND_RK) блока DB42, передаются в модуль CP.

После этого меркер M 120.7 немедленно сбрасывается.

После завершения задания P_SND_RK без ошибок, счетчик успешных завершений задания инкрементируется; после завершения задания с ошибкой инкрементируется счетчик завершений с ошибкой.

Задание P_RCV_RK

SFB P_RCV_RK программируется в FC23, где индикатор "прием разрешен" (Receive Enable) всегда равен "1" для приема данных от коммуникационного партнера.

Принятые данные вводятся в целевую область (P_RCV_RK), при этом количество введенных данных отражается в параметре длины LEN.

Если данные приняты без ошибок, то счетчик успешных завершений задания инкрементируется; после завершения задания с ошибкой инкрементируется счетчик завершений с ошибкой.

Для заданий P_SND_RK и P_RCV_RK выходной параметр состояния STATUS сохраняется, если его значение отлично от 0.

8.1.3 Пример программирования

Пример программирования

В примере программирования используются следующие блоки:

Блок	Символ	Комментарий
OB 100	Complete Restart	Стартовый блок (Startup OB) для перезапуска
FC 10	Initiation	Функция FC для блока (Startup OB)
OB 1	Cycle Execution	Циклическое выполнение программы
FC 21	Execute Send Jobs	FC, вызывающий SFB P_SND_RK
FC 23	Execute Receive Jobs	FC, вызывающий SFB P_RCV_RK

OB 100	Complete Restart	Стартовый блок (Startup OB) для перезапуска
--------	------------------	---

```

L      272                //логический адрес
T      DB40.DBW  16       //для операций SEND
T      DB41.DBW  16       //и RCV
UC     FC 10              //вызов FC для инициализации

```

FC 10	Initiation	Функция FC для блока (Startup OB)
-------	------------	-----------------------------------

```

//-----
Reset Control Bits      //Сброс управляющих битов
//-----
L      B#16#0
T      DB40.DBB  0       //SEND- рабочий (Work)-DB
T      DB41.DBB  0       //RCV- рабочий (Work)-DB
//-----
Reset counters/STATUS  //Сброс счетчиков/парам.состояния
//-----
L      W#16#0
T      DB40.DBW  6       //SEND- рабочий (Work)-DB
T      DB40.DBW  8
T      DB40.DBW  12
T      DB40.DBW  14
T      DB41.DBW  6       //RCV- рабочий (Work)-DB
T      DB41.DBW  8
T      DB41.DBW  12
T      DB41.DBW  14

```


Циклическая часть программы

ОВ 1	Cycle Execution	Циклическое выполнение программы
------	-----------------	----------------------------------

```
UC      FC 21      //Вызов функции пересылки SEND
UC      FC 23      //Вызов функции приема RCV
```

FC 21	Execute Send Jobs	FC, вызывающий SFB P_SND_RK
-------	-------------------	-----------------------------

```
//      -----
//      Подключение для SEND
//      -----
U      M      120.7      //запуск SEND
UN     DB40.DBX 0.0      //SEND_REQ
UN     DB40.DBX 0.4      //SEND_DONE
UN     DB40.DBX 0.5      //SEND_ERROR
R      M      120.7      //сброс меркера запуска SEND
S      DB40.DBX 0.0      //установка SEND_REQ

//      -----
//      Генерация фронта сигнала SEND_REQ
//      -----
U(
O      DB40.DBX 0.4      //SEND_DONE
O      DB40.DBX 0.5      //SEND_ERROR
)
U      DB40.DBX 0.0      //SEND_REQ
R      DB40.DBX 0.0      //SEND при REQ=0

//      -----
//      Задание длины LEN
//      -----
L      W#16#20      //длина данных для SEND
T      DB40.DBW 10      //SEND-LEN

//      -----
//      Передача SEND с экземплярным DB
//      -----
CALL FB 8 ,      DB50
SF:=
REQ:=DB40.DBX0.0
R:=DB40.DBX0.1
LADDR:=DB40.DBW16
DB_NO:=42
DBB_NO:=10
LEN :=DB40.DBW10
```

```

R_CPU_NO:=
R_TYP:='x'
R_NO:=
R_OFFSET:=
R_CF_BYT:=
R_CF_BIT:=
DONE :=DB40.DBX0.4
ERROR :=DB40.DBX0.5
STATUS:=DB40.DBW12

// -----
// Проверка завершения без ошибок
// -----
U   DB40.DBX    0.4   //выполнено без ошибок (DONE) ?
SPBN CON1
//если нет
L   DB40.DBW    6     //"выполнено без ошибок"
+1
//инкрементирование счетчика
T   DB40.DBW    6

:
// :
:
// другие пользовательские функции
:
// :

SPA LEAV

// -----
// Проверка завершения с ошибкой
// -----
CON1: U   DB40.DBX    0.5   //выполнено с ошибкой (ERROR) ?
SPBN CON2
//если нет
L   DB40.DBW    8     //"выполнено с ошибкой"
+1
//инкрементирование счетчика
T   DB40.DBW    8

:
// :
:
// обработка ошибок
:
// :

L   0
L   DB40.DBW    12   //если STATUS <>0
==I
SPB LEAV
T   DB40.DBW    14   //сохранение STATUS
SPA LEAV

```

```

// -----
//      Проверка "Ошибка в STATUS"
// -----
CON2:  L    0
      L   DB40.DBW    12    //если STATUS <>0
      ==I
      SPB  LEAV
      T   DB40.DBW    14    //сохранение STATUS
:
:      // :
:      // обработка ошибок
:      // :
LEAV:  CLR

```

FC 23	Execute Receive Jobs	FC, вызывающий SFB P_RCV_RK
-------	----------------------	-----------------------------

```

// -----
//      Разрешение приема данных (Enable Receive Data)
// -----
      SET
      =   DB41.DBX    0.0    //прием (RCV) с EN_R=1
// -----
//      прием (RCV) с экземплярным DB
// -----
CALL  FB 7 ,    DB70
EN_R :=DB41.DBX0.0
R:=
LADDR:=DB41.DBW16
DB_NO:=43
DBB_NO:=0
L_TYP:=
L_NO:=
L_OFFSET:=
L_CF_BYT:=
L_CF_BIT:=
NDR :=DB41.DBX0.4
ERROR :=DB41.DBX0.5
LEN :=DB41.DBW10
STATUS:=DB41.DBW12

```

```

// -----
// Проверка завершения без ошибок
// -----
U      DB41.DBX   0.4   //выполнено без ошибок (NDR) ?
SPBN   CON1      //если нет
L      DB41.DBW   6     //"выполнено без ошибок"
+1     //инкрементирование счетчика
T      DB41.DBW   6
L      DB41.DBW   10    //сохранено
T      DB41.DBW   4     //длина принятых данных LEN
SPA    LEAV

// -----
// Проверка завершения с ошибкой
// -----
CON1:  U      DB41.DBX   0.5   //выполнено без ошибок (ERROR) ?
SPBN   CON2      //если нет
L      DB41.DBW   8     //"выполнено с ошибкой"
+1     //инкрементирование счетчика
T      DB41.DBW   8
L      0
L      DB41.DBW   12    //если STATUS <>0
==I
SPB    LEAV
T      DB41.DBW   14    //сохранение STATUS
SPA    LEAV

// -----
// Проверка "Ошибка в STATUS"
// -----
CON2:  L      0
L      DB41.DBW   12    //если STATUS <>0
==I
SPB    LEAV
T      DB41.DBW   14    //сохранение STATUS

LEAV: CLR

```

8.2 Пример для модуля CP 441-2

Общая информация

В представленном ниже простом примере программирования показано использование блоков SFB BSEND, BCRV и STATUS.

После инсталляции вспомогательной программы Modbus Master файл с примером программы сохраняется в STEP 7 в директории EXAMPLES с именем *Modma*.

Программа для S7-400 предназначена только для информирования пользователя и не должна восприниматься как готовое решение для конфигурирования конкретной установки.

Для иллюстрации базовой структуры мы намеренно сохранили простоту и не применяли символьное отображение.

8.2.1 Используемые блоки

Используемые блоки

В примере программирования используются следующие блоки:

Блок	Символ	Комментарий
OB 1	Cycle Execution	Циклическое выполнение программы
OB 100	Complete Restart	Стартовый блок (Startup OB) для перезапуска
FC 100	Initiation	FC для блока (Startup OB)
FC 210	Execute BSEND Jobs	FC, вызывающий SFB BSEND
FC 230	Execute BRCV Jobs	FC, вызывающий SFB BRCV
FC 250	Execute STATUS Jobs	FC, вызывающий SFB STATUS
DB 22	IDB_STATUS	Экземплярный DB для BSEND
DB 50	IDB_BSEND	Экземплярный DB для BSEND
DB 70	IDB_BRCV	Экземплярный DB для BRCV
DB 400	Work DB BSEND	Рабочий (Work) DB для FC210 и BSEND
DB 401	Work DB BRCV	Рабочий (Work) DB для FC230 и BRCV
DB 410	SOURCE_DB	Исходный DB BSEND для передачи
DB 430	DESTINATION_DB	Целевой DB BRCV для приема
DB 450	Work DB STATUS	Рабочий DB и DB SYSTAT для приема для FC 250 и STATUS

Используемые данные

В примере программирования используются следующие операнды (меркеры, биты данных, слова данных):

Блок	Символ	Комментарий
M 119.7		Бит запуска (Trigger bit) для выполнения задания STATUS
M 120.7		Бит запуска (Trigger bit) для выполнения задания BSEND
DB400.DBX 0.0		Управляющий параметр запроса REQ: для активации задания BSEND
DB400.DBX 0.1		Управляющий параметр сброса Reset: для прерывания задания BSEND
DB400.DBX 0.4		Параметр индикации состояния DONE: показывает, что задание BSEND завершено без ошибок
DB400.DBX 0.5		Параметр индикации состояния ERROR: показывает, что задание BSEND завершено с ошибкой
DB400.DBW 6		Счетчик успешных завершений (Success) задания BSEND
DB400.DBW 8		Счетчик завершений с ошибкой (Error) задания BSEND
DB400.DBW 10		Длина LEN исходных данных для задания BSEND передаваемых в CP (в байтах)
DB400.DBW 12		Состояние (STATUS) в BSEND
DB400.DBW 14		Сохраненное состояние (Stored STATUS) в BSEND
DB400.DBW 16		Параметр R_ID в BSEND
DB401.DBX 0.0		Управляющий параметр EN_R: "BRCV готов к приему"
DB401.DBX 0.4		Параметр индикации состояния NDR: показывает, что текущий BRCV принял новые данные от CP
DB401.DBX 0.5		Параметр индикации состояния ERROR: показывает, что текущий BRCV обработан с ошибкой
DB401.DBW 4		Сохраненная длина (Stored length) LEN в BRCV
DB401.DBW 6		Счетчик успешных завершений (Success) задания BRCV
DB401.DBW 8		Счетчик завершений с ошибкой (Error) задания BRCV
DB401.DBW 10		Длина LEN целевой области данных для задания BRCV, принятая CP (в байтах)
DB401.DBW 12		Состояние (STATUS) в BRCV
DB401.DBW 14		Сохраненное состояние (Stored STATUS) в BRCV
DB401.DBW 16		Параметр R_ID в BRCV

Блок	Символ	Комментарий
DB450.DBX 0.0		Управляющий параметр запроса REQ: для активации задания STATUS
DB450.DBX 0.4		Параметр индикации состояния NDR: показывает, что в задании STATUS новые данные SYSTAT приняты от CP
DB450.DBX 0.5		Параметр индикации состояния ERROR: показывает, что задание STATUS завершено с ошибкой
DB450.DBW 6		Счетчик успешных завершений (Success) задания STATUS
DB450.DBW 8		Счетчик завершений с ошибкой (Error) задания STATUS
DB450.DBW 12		Состояние (STATUS) в задании STATUS
DB450.DBW 14		Сохраненное состояние (Stored STATUS)
DB450.DBW 16		Параметр PHYS: физическое состояние (статус) интерфейса (кроме случаев использования PtP-соединения)
DB450.DBW 18		Параметр LOG: логическое состояние (статус) интерфейса (кроме случаев использования PtP-соединения)
DB450.DBW 20 ... 34		Параметр LOCAL: целевая область (область назначения) для принятых данных SYSTAT интерфейса
DB450.DBW 40 ... 54		Сохраненная область (Stored) SYSTAT

8.2.2 Описание программы

Общая информация

Пример программы содержит:

- Блок запуска OB100, FC100
- Циклически выполняемую часть программы, вызываемую в OB1
- Функциональный блок FC210 для передачи данных от CPU в CP (Send)
- FC230 для приема данных из CP в CPU
- FC250 для считывания данных из области SYSTAT

Параметры для запрограммированных системных функциональных блоков BSEND, BRCV и STATUS сохраняются в рабочих (work) DB-блоках: DB400 (BSEND), DB401 (BRCV) и DB450 (STATUS).

Данные для передачи (область исходных данных "BSEND source area") располагаются в DB410.

Данные, принятые от коммуникационного партнера вводятся в DB430 (целевая область для принятых данных "BRCV destination area").

Значение 1000 (Hex) определяется как ID для SFB BSEND, BRCV и STATUS. Идентификаторы ID получают номера, начиная с 1000 (Hex) в конфигурации связей данных проекта. Если в конфигурации связей данных проекта ID различны, то такие ID должны быть определены для соответствующих блоков SFB. (См. также раздел "Связи данных (Data Link)").

Задание BSEND

Задание BSEND может быть активировано в циклической части программы при установке меркера ("memory bit") M 120.7 (например, посредством переменной). Данные длины LEN, содержащиеся в исходной области (BSEND) блока DB410, передаются в модуль CP. После этого меркер M 120.7 немедленно сбрасывается.

После завершения задания BSEND без ошибок, счетчик успешных завершений задания инкрементируется; после завершения задания с ошибкой инкрементируется счетчик завершений с ошибкой.

Задание BRCV

SFB BRCV программируется в FC230, где индикатор "прием разрешен" (Receive Enable) всегда равен "1" для приема данных от коммуникационного партнера. Принятые данные вводятся в целевую область (BRCV), при этом количество введенных данных отражается в параметре длины LEN.

Если данные приняты без ошибок, то счетчик успешных завершений задания инкрементируется; после завершения задания с ошибкой инкрементируется счетчик завершений с ошибкой.

Задание STATUS

Задание считывания области SYSTAT может быть активировано в циклической части программы при установке меркера ("memory bit") M 119.7 (например, посредством переменной). После этого меркер M 119.7 немедленно сбрасывается.

Задание STATUS программируется в FC250. При выполнении этого задания данные из области SYSTAT модуля CP вводятся в целевую область, определенную в блоке STATUS.

Если данные приняты без ошибок, то счетчик успешных завершений задания инкрементируется; после завершения задания с ошибкой инкрементируется счетчик завершений с ошибкой.

Для заданий BSEND, BRCV и STATUS выходной параметр состояния STATUS сохраняется, если его значение отлично от 0.

8.2.3 Пример программирования

Пример программирования

В примере программирования используются следующие блоки:

Блок	Символ	Комментарий
OB 100	Complete Restart	Стартовый блок (Startup OB) для перезапуска
FC 100	Initiation	FC для блока (Startup OB 100)
OB 1	Cycle Execution	Циклическое выполнение программы
FC 210	Execute BSEND Jobs	FC, вызывающий SFB BSEND
FC 230	Execute BRCV Jobs	FC, вызывающий SFB BRCV
FC 250	Execute STATUS Jobs	FC, вызывающий SFB STATUS

Блок	Символ	Комментарий
OB 100	Complete Restart	Стартовый блок (Startup OB) для перезапуска

```
CALL FC 100 //Инициализация
```

Блок	Символ	Комментарий
FC 100	Initiation	FC для блока (Startup OB 100)

```
// -----
Reset Control Bits //Сброс управляющих битов
// -----
L   B#16#0
T   DB400.DBB 0 //BSEND-Work-DB
T   DB401.DBB 0 //BRCV-Work-DB
T   DB450.DBB 0 //STATUS-Work-DB
// -----
Reset counters/STATUS //Сброс счетчиков/парам.состояния
// -----
L   W#16#0
T   DB400.DBW 6 //BSEND- Work -DB
T   DB400.DBW 8
T   DB400.DBW 12
T   DB400.DBW 14
T   DB401.DBW 6 //BRCV- Work -DB
T   DB401.DBW 8
T   DB401.DBW 12
T   DB401.DBW 14
T   DB450.DBW 6 //STATUS- Work -DB
T   DB450.DBW 8
T   DB450.DBW 12
T   DB450.DBW 14
```

OB 1	Cycle Execution	Циклическое выполнение программы
------	-----------------	----------------------------------

```

UC    FC 210          //Вызов BSEND
UC    FC 230          //Вызов BRCV
UC    FC 250          //Вызов STATUS

```

FC 210	Execute BSEND Jobs	FC, вызывающий SFB BSEND
--------	--------------------	--------------------------

```

// -----
// Подключение для SEND
// -----
U      M 120.7          //запуск SEND
UN     DB450.DBX 0.0    //REQuest STATUS
UN     DB400.DBX 0.0    //BSEND_REQ
UN     DB400.DBX 0.4    //BSEND_DONE
UN     DB400.DBX 0.5    //BSEND_ERROR
R M    120.7          //сброс меркера запуска BSEND
S      DB400.DBX 0.0    //установка BSEND_REQ
// -----
// Генерация фронта сигнала BSEND_REQ
// -----
U(
O      DB400.DBX 0.4    //BSEND_DONE
O      DB400.DBX 0.5    //BSEND_ERROR
)
U      DB400.DBX 0.0    //BSEND_REQ
R      DB400.DBX 0.0    //BSEND при REQ=0
// -----
// Задание R_ID, LEN
// -----
L      DW#16#1          //Использование R_ID = 1
T      DB400.DBD 16     //в качестве BSEND-R_ID
L      W#16#6           //длина BSEND-Data
T      DB400.DBW 10     //BSEND-LEN
// -----
// BSEND с экземплярным DB
// -----
CALL  SFB 12 ,  DB50
REQ  :=DB400.DBX0.0
R    :=DB400.DBX0.1
ID   :=W#16#1000
R_ID :=DB400.DBD16
DONE :=DB400.DBX0.4
ERROR :=DB400.DBX0.5
STATUS:=DB400.DBW12
SD_1 :=P#DB410.DBX 10.0 WORD 1

```

```

LEN :=DB400.DBW10

// -----
// Проверка завершения без ошибок
// -----
U    DB400.DBX 0.4 //выполнено без ошибок DONE ?
SPBN CON1          //если нет
L    DB400.DBW 6  //“выполнено без ошибок”
+1                               //инкрементирование счетчика
T    DB400.DBW 6

:                               // :
:                               // другие пользовательские функции
:                               // :

SPA  LEAV

// -----
// Проверка завершения с ошибкой
// -----
CON1: U    DB400.DBX 0.5 //выполнено с ошибкой (ERROR) ?
SPBN CON2          //если нет
L    DB400.DBW 8  //“выполнено с ошибкой”
+1                               //инкрементирование счетчика
T    DB400.DBW 8

:                               // :
:                               // обработка ошибок
:                               // :

L    0
L    DB400.DBW 12 //если STATUS <>0
==I
SPB  LEAV
T    DB400.DBW 14 //сохранение STATUS
SPA  LEAV

// -----
// Проверка "Ошибка в STATUS"
// -----
CON2: L    0
L    DB400.DBW 12 //если STATUS <>0
==I
SPB  LEAV
T    DB400.DBW 14 //сохранение STATUS

:                               // :
:                               // обработка ошибок
:                               // :
LEAV: CLR

```

FC 230	Execute BRCV Jobs	FC, вызывающий SFB BRCV
--------	-------------------	-------------------------

```

// -----
// Задание R_ID
// -----
L    DW#16#1          //использование BRCV-R_ID = 1
T    DB401.DBD 16    //(также как BSEND-R_ID)
// -----
// Разрешение приема данных
// -----
SET
= DB401.DBX 0.0      //BRCV при EN_R=1
// -----
// BRCV с экземплярным DB
// -----
CALL SFB 13 , DB70
EN_R :=DB401.DBX0.0
ID :=W#16#1000
R_ID :=DB401.DBD16
NDR :=DB401.DBX0.4
ERROR :=DB401.DBX0.5
STATUS:=DB401.DBW12
RD_1 :=P#DB430.DBX 0.0 WORD 128
LEN :=DB401.DBW10
// -----
// Проверка завершения без ошибок
// -----
U    DB401.DBX 0.4    //выполнено без ошибок (NDR) ?
SPBN CON1            //если нет
L    DB401.DBW 6      //"принято без ошибок"
+1                               //инкрементирование счетчика
T    DB401.DBW 6
L    DB401.DBW 10     //сохранено
T    DB401.DBW 4      //длина принятых данных LEN
SPA LEAV
// -----
// Проверка завершения с ошибкой
// -----
CON1: U    DB401.DBX 0.5 //выполнено с ошибкой (ERROR) ?
SPBN CON2            //если нет
L    DB401.DBW 8      //"выполнено с ошибкой"
+1                               //инкрементирование счетчика
T    DB401.DBW 8
L    0
L    DB401.DBW 12     //если STATUS <>0

```

```

==I
SPB LEAV
T DB401.DBW 14 //сохранение STATUS
SPA LEAV
// -----
// Проверка "Ошибка в STATUS"
// -----
CON2: L 0
L DB401.DBW 12 //если STATUS <>0
==I
SPB LEAV
T DB401.DBW 14 //сохранение STATUS
LEAV: CLR

```

FC 250	Execute STATUS Jobs	FC, вызывающий SFB STATUS
--------	---------------------	---------------------------

```

// -----
// Подключение STATUS
// -----
U M 119.7 //запуск STATUS
UN DB400.DBX 0.0 //BSEND-REQ активен?
UN DB450.DBX 0.0 //STATUS_REQ активен?
R M 119.7 //сброс запуска STATUS
S DB450.DBX 0.0 //активация STATUS_REQ
// -----
// Генерация фронта сигнала STATUS_REQ
// -----
U (
O DB450.DBX 0.4 //STATUS_NDR
O DB450.DBX 0.5 //STATUS_ERROR
)
U DB450.DBX 0.0 //STATUS_REQ
R DB450.DBX 0.0 //STATUS при REQ=0
// -----
// STATUS с экземплярным DB (= считывание области SYSTAT)
// -----
CALL SFB 22 , DB22
REQ :=DB450.DBX0.0
ID :=W#16#1000
NDR :=DB450.DBX0.4
ERROR :=DB450.DBX0.5
STATUS:=DB450.DBW12
PHYS :=P#DB450.DBX 16.0 BYTE 2
LOG :=P#DB450.DBX 18.0 BYTE 2
LOCAL :=P#DB450.DBX 20.0 BYTE 16

```

```

//      -----
//      Проверка "новые данные приняты"
//      -----
U      DB450.DBX  0.4  //выполнено без ошибок (NDR) ?
SPBN   CON1                //если нет
L      DB450.DBW  6    //"выполнено без ошибок"
+1                    //инкрементирование счетчика
T      DB450.DBW  6
U      DB450.DBX  22.0 //бит 0: есть ошибки ?
SPBN   LEAV                //если нет
//      -----
//      Сохранение SYSTAT
//      -----
L      DB450.DBW  22
T      DBW  42
L      DBD  24
T      DBD  44
L      DBD  28
T      DBD  48
L      DBD  32
T      DBD  52
SPA    LEAV
//      -----
//      Проверка завершения с ошибкой
//      -----
CON1:  U      DB450.DBX  0.5  //выполнено без ошибок (ERROR) ?
SPBN   CON2                //если нет
L      DB450.DBW  8    //"выполнено с ошибкой"
+1                    //инкрементирование счетчика
T      DB450.DBW  8
L      0
L      DB450.DBW  12  //если STATUS <>0
==I
SPB    LEAV
T      DB450.DBW  14  //сохранение STATUS
SPA    LEAV
//      -----
//      Проверка "Ошибка в STATUS"
//      -----
CON2:  L      0
L      DB450.DBW  12  //если STATUS <>0
==I
SPB    LEAV
T      DB450.DBW  14  //сохранение STATUS
LEAV:  CLR

```


А Технические данные

Значения времени пересылки

Далее в таблицах указываются измеренные значения времени, требуемые для передачи данных, для различных функциональных кодов: Для измерения временных характеристик передачи были использованы станция S7-300 с CPU 315-2DP (6ES7 315-2AF01-0AB0) и CP 341 и как партнер - станция S7-400 с CPU 414 (6ES7 414-1XG01-0AB0) и CP 441-2.

В каждом эксперименте время обработки измерялось:

- от момента инициации задания в пользовательской программе и включая время обработки в ведущем (master) устройстве,
- как время, требуемое для пересылки данных партнеру через последовательный интерфейс,
- как время обработки в ведомом (slave) устройстве,
- как время, требуемое для пересылки квитирования партнеру через последовательный интерфейс.

При вычислении полного времени передачи все указанные выше компоненты складываются. Если в качестве партнерского устройства Вы используете другое (ведущее (master) или ведомое (slave)), при вычислении полного времени передачи Вы должны использовать те компоненты времени, которые соответствуют варианту вместо табличных значений. Значения времени для выполнения задания и квитирования не изменяются; они зависят только от используемой скорости передачи.

Ведущее (master) устройство CP 341
Функциональный код 1 (Read) - считывание выхода (в мс)

Скорость передачи (бит/с)	300			
Пользовательские данные	Ведущий CP 341	Задание (Job)	Ведомый CP 441-2	Квитиро- вание
1 байт	236	257	188	184
10 байтов	236	257	190	515
20 байтов	238	257	190	882
50 байтов	244	257	193	1986
100 байтов	280	257	199	3824
200 байтов	286	257	207	7502
255 байтов	288	257	216	9487

Скорость передачи (бит/с)	9600			
Пользовательские данные	Ведущий CP 341	Задание (Job)	Ведомый CP 441-2	Квитиро- вание
1 байт	33	8	40	6
10 байтов	33	8	43	16
20 байтов	35	8	44	28
50 байтов	42	8	45	62
100 байтов	56	8	56	120
200 байтов	75	8	64	235
255 байтов	82	8	77	296

Скорость передачи (бит/с)	76800			
Пользовательские данные	Ведущий CP 341	Задание (Job)	Ведомый CP 441-2	Квитиро- вание
1 байт	35	1	23	1
10 байтов	36	1	25	2
20 байтов	37	1	26	3
50 байтов	46	1	27	8
100 байтов	61	1	30	15
200 байтов	82	1	39	29
255 байтов	92	1	48	37

Ведущее (master) устройство CP 341
Функциональный код 15 (Write) - запись в несколько выходов (в мс)

Скорость передачи (бит/с)	300			
Пользовательские данные	Ведущий CP 341	Задание (Job)	Ведомый CP 441-2	Квитиро- вание
1 байт	225	331	223	257
10 байтов	227	662	224	257
20 байтов	227	1030	228	257
50 байтов	227	2132	232	257
100 байтов	229	3971	236	257
200 байтов	230	7648	243	257
255 байтов	237	9634	255	257

Скорость передачи (бит/с)	9600			
Пользовательские данные	Ведущий CP 341	Задание (Job)	Ведомый CP 441-2	Квитирование
1 байт	64	11	62	8
10 байтов	64	21	63	8
20 байтов	69	32	64	8
50 байтов	69	67	68	8
100 байтов	72	124	70	8
200 байтов	75	239	76	8
255 байтов	75	301	86	8

Скорость передачи (бит/с)	76800			
Пользовательские данные	Ведущий CP 341	Задание (Job)	Ведомый CP 441-2	Квитирование
1 байт	60	1	56	1
10 байтов	60	3	58	1
20 байтов	62	4	58	1
50 байтов	64	9	60	1
100 байтов	67	16	67	1
200 байтов	72	30	77	1
255 байтов	77	38	84	1

Ведущее (master) устройство CP 441
Функциональный код 1 (Read) - считывание выхода (в мс)

Скорость передачи (бит/с)	300			
Пользовательские данные	Ведущий CP 441-2	Задание (Job)	Ведомый CP 341	Квитирование
1 байт	229	257	179	184
10 байтов	229	257	179	512
20 байтов	229	257	180	882
50 байтов	232	257	182	1986
100 байтов	236	257	192	3842
200 байтов	243	257	208	7501
255 байтов	251	257	214	9487

Скорость передачи (бит/с)	9600			
Пользовательские данные	Ведущий CP 441-2	Задание (Job)	Ведомый CP 341	Квитирование
1 байт	74	8	18	6
10 байтов	75	8	19	16
20 байтов	77	8	19	27
50 байтов	83	8	24	62
100 байтов	90	8	34	119
200 байтов	92	8	48	235
255 байтов	95	8	56	296

Скорость передачи (бит/с)	76800			
Пользовательские данные	Ведущий CP 441-2	Задание (Job)	Ведомый CP 341	Квитиро- вание
1 байт	73	1	13	1
10 байтов	74	1	13	2
20 байтов	76	1	13	4
50 байтов	86	1	20	8
100 байтов	93	1	29	15
200 байтов	95	1	45	29
255 байтов	97	1	50	37

Ведущее (master) устройство CP 441

Функциональный код 15 (Write) - запись в несколько выходов (в мс)

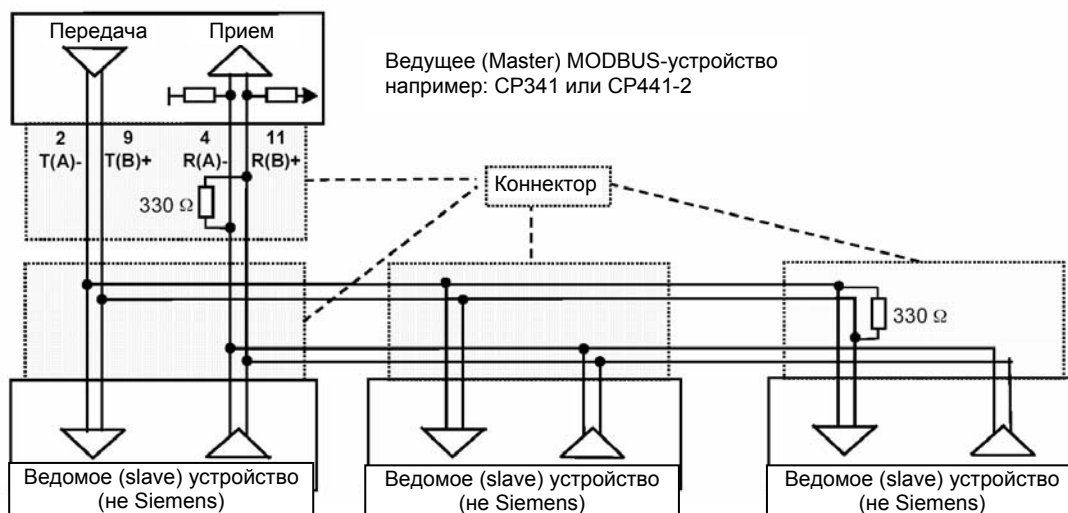
Скорость передачи (бит/с)	300			
Пользовательские данные	Ведущий CP 441-2	Задание (Job)	Ведомый CP 341	Квитиро- вание
1 байт	205	331	199	257
10 байтов	206	662	200	257
20 байтов	206	1028	201	257
50 байтов	208	2132	212	257
100 байтов	211	3971	223	257
200 байтов	217	7648	238	257
255 байтов	221	9634	243	257

Скорость передачи (бит/с)	9600			
Пользовательские данные	Ведущий CP 441-2	Задание (Job)	Ведомый CP 341	Квитиро- вание
1 байт	48	10	41	8
10 байтов	48	20	41	8
20 байтов	50	32	43	8
50 байтов	52	67	48	8
100 байтов	55	124	56	8
200 байтов	63	239	74	8
255 байтов	67	301	88	8

Скорость передачи (бит/с)	76800			
Пользовательские данные	Ведущий CP 441-2	Задание (Job)	Ведомый CP 341	Квитиро- вание
1 байт	58	1	40	1
10 байтов	61	3	43	1
20 байтов	62	4	43	1
50 байтов	63	8	44	1
100 байтов	64	15	50	1
200 байтов	66	30	69	1
255 байтов	68	38	85	1

В Схемы многоточечных соединений

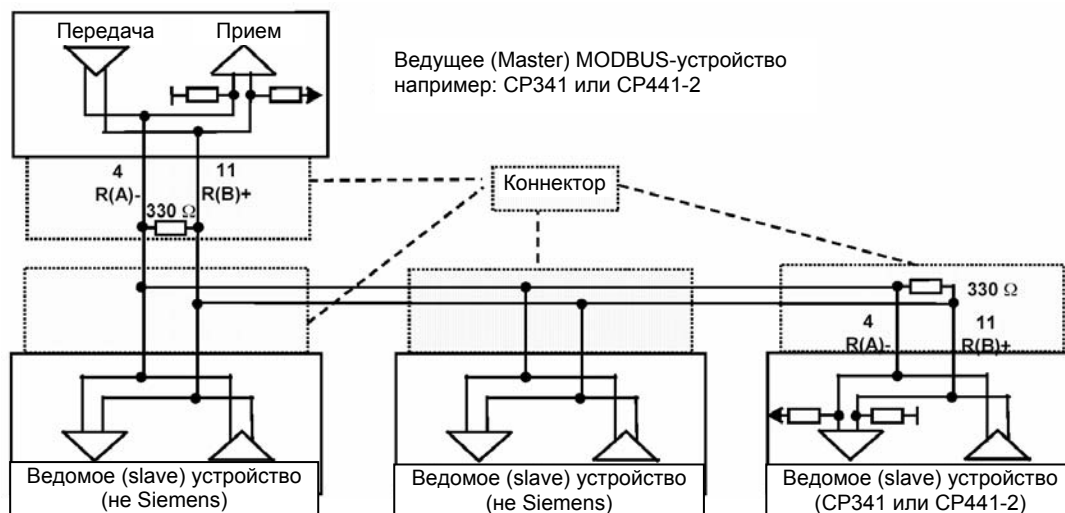
Схема многоточечного соединения RS422 (MODBUS)



Внимание

В режиме RS422 модули CP341 и CP441-2 могут использоваться только как ведущие (Master) устройства, потому что входные (receive) линии могут иметь три состояния ("Tri State")!

Схема многоточечного соединения RS485 (MODBUS)



Следующее касается модулей обоих типов:

- "Земляной" контакт (GND) (контакт 8 для CP341 / CP441-2) должен быть всегда подключен на обоих концах кабеля.
- Везде должны быть использованы экранированные кобуха.
- На коннекторе должен быть распаян оконечный резистор (примерно 330 Ом) в последнем приемном устройстве (receiver) всей последовательности узлов.
- Рекомендуется использовать следующий тип кабеля:
LIYCY 3 x 2 x 0,14 R(A)/R(B) и T(A)/T(B) (витая пара).
- Схема с фиктивным модулем ("Stub") не допускается.

С Справочная литература

Протокол Modbus (Modbus Protocol)

/1/ Gould Modbus Protocol Reference Guide PI-MBUS-300 Rev B GOULD Electronics

Справочное руководство *Протокол Gould Modbus PI-MBUS-300 выпуск B*
(GOULD Electronics)

Словарь терминов

А

Автономный режим (Офлайн)

Термин обозначающий режим работы с данными. Режим офлайн (или автономный режим) означает отсутствие обмена данными между PLC и программатором.

Адрес

Адрес определяет физическое место расположения данных. Если адрес известен, то возможен прямой доступ к операнду, на который адрес указывает.

Б

Блок

Блоки - это элементы пользовательской программы, которые характеризуются своей функцией, структурой или назначением. В STEP 7 различают

- кодовые блоки (FB, FC, OB, SFB, SFC)
- блоки данных (DB, SDB)
- типы данных, определенные пользователем (UDT)

Блок данных (DB)

Блоки данных - это блоки, содержащие данные и параметры, которые используются в пользовательской программе. В отличие от других блоков, блоки данных не содержат инструкций. Они подразделяются на глобальные (общие) блоки данных и экземплярные блоки данных. Доступ к данным, содержащимся в этих блоках, может быть абсолютным или символьным. Сложные данные могут храниться в структурированных формах.

В

Время цикла (Cycle Time)

Время цикла (cycle time) - это время, необходимое CPU для однократного сканирования пользовательской программы.

Вызов блока

Вызов блока происходит, когда ход выполнения основной программы временно оставляется для того, чтобы выполнить программу вызванного блока.

Д

Диагностические события (Diagnostic Events)

Диагностические события - это, например, ошибки в модуле или системные ошибки в CPU, которые вызваны, скажем, ошибкой в программе или изменением рабочего режима.

Диагностические функции

Диагностические функции охватывают целую систему средств диагностики, включающую в себя функции детектирования, анализа и генерации отчетов об ошибках в PLC.

Диагностический буфер

Каждый CPU имеет диагностический буфер, в котором хранится подробная информация о диагностических событиях в порядке их возникновения. Модуль CP 341 имеет свой диагностический буфер, в который вводятся все диагностические события CP 341 (ошибки оборудования и микропрограммы, ошибки инициализации и параметризации, ошибки при приеме и при передаче).

З

Загрузка (Download)

Термин загрузка означает загрузку загружаемых объектов (например, кодовых блоков) из программатора в загружаемую память (load memory) CPU.

И

Интерактивная справка (On-line Help)

STEP 7 обеспечивает пользователя контекстно отображаемой справочной информацией при работе с программным обеспечением.

Интерактивный режим (On-line)

Термин обозначающий режим работы с данными. Режим онлайн (или интерактивный режим) означает наличие обмена данными между PLC и программатором.

Интерфейсный submodule

Интерфейсный submodule модуля CP 441-2 отвечает за физическое преобразование сигналов. Заменяя интерфейсный submodule на другой делает возможным совместимость модуля CP с коммуникационным партнером.

Загружаемый драйвер для PtP-соединений CP
(S7 - ведущее устройство)
A5E00218410-04

Интерфейс параметризации CP: *Point-to-Point Communication, Parameter Assignment*

Интерфейс параметризации CP: *Point-to-Point Communication, Parameter Assignment* используется для параметризации submodule коммуникационного процессора CP.

К

Коммуникационный процессор

Коммуникационные процессоры - это модули, обеспечивающие PtP-соединения и соединения по шине.

Конфигурация связанных данных проекта (Project Configuration of Data Link)

Конфигурация связанных данных проекта - это термин относящийся к размещению ID соединений в системном функциональном блоке. ID соединений позволяет отдельным системным функциональным блокам обеспечивать связь между двумя коммуникационными узлами.

Конфигурирование

Конфигурирование - это установка отдельных модулей PLC в таблице конфигурации (configuration table).

М

Модули

Модули - это вставляемые в PLC печатные платы, расширяющие его функциональность.

О

Образ процесса (Process Image)

Образ процесса - это специальная область памяти в PLC. В начале выполнения циклической программы состояния сигналов входных модулей пересылается в таблицу входов области отображения процесса (process image input table). В конце выполнения циклической программы значения из таблицы выходов области отображения процесса (output table) пересылаются в выходные модули и формируют состояния выходов модулей вывода сигналов.

Онлайн/Офлайн (On-line/Off-line)

Термины обозначающие режим работы с данными. Режим онлайн (или интерактивный режим) означает наличие обмена данными между PLC и программатором. Режим офлайн (или автономный режим) означает отсутствие обмена данными.

Операнд

Операнд - это часть инструкции и выражений в STEP 7, с которой процессор выполняет действия. Операнд может быть адресован символьным или абсолютным способом.

Операционная система CPU

Операционная система CPU организует все функции и операции CPU, которые не связаны с пользовательской задачей управления.

П**Параметр**

Параметры - это значения, которые могут быть назначены. Различают параметры блоков и параметры модулей.

Параметризация

Термин означает процесс задания определенного поведения модуля.

Параметр блока

Параметры блока - это переменные внутри блоков многократного использования, которые заменяются текущими значениями, когда вызывается соответствующий блок.

Параметр модуля

Параметры модулей устанавливаются для задания их определенных реакций. Различают статические и динамические параметры.

Перезапуск (RESTART)

PLC при переходе от режима STOP к режиму RUN проходит режим перезапуска (RESTART).

Переменная

Переменная - это операнд (например, E 1.0), который может иметь символьное имя и может, следовательно, быть адресован с помощью этого символьного имени.

Прерывание (Interrupt)

Прерывание происходит, когда обработка программы в процессоре PLC прерывается внешним сигналом.

Программируемый логический контроллер (PLC)

Программируемый логический контроллер (PLC) - это электронное устройство управления с по крайней мере одним центральным процессором, различными модулями ввода/вывода и устройствами операторского управления и мониторинга.

Протокол

Коммуникационные партнеры, участвующие в обмене данными, должны руководствоваться определенными правилами для управления и обработки потока данных. Такие правила называются протоколами.

Процедура

Выполнение операции обмена данными в соответствии с определенным протоколом называется процедурой.

Р

Рабочая память (Work Memory)

Рабочая память - это RAM-память в CPU, к которой процессор обращается, когда обрабатывает программу пользователя.

Рабочий режим

Программируемые контроллеры SIMATIC S7 имеют три рабочих режима: STOP (Стоп), RESTART (Перезапуск) и RUN (Выполнение). Функции CPU изменяются в зависимости от выбранного рабочего режима.

С

Системный блок

Системные блоки отличаются от других блоков тем, что они уже интегрированы в систему S7-300 и уже доступны для уже определенных системных функций. Системные блоки подразделяются на системные блоки данных, системные функции (SFC) и системные функциональные блоки (SFB).

Системные функции (SFC)

Системные функции (SFC) - это модули, не имеющие собственной памяти, которые уже интегрированы в операционную систему CPU и могут быть вызваны из пользовательской программы.

Системные функциональные блоки (SFB)

Системные функциональные блоки (SFB) - это модули, имеющие свою память, которые уже интегрированы в операционную систему CPU и могут быть вызваны из пользовательской программы.

Стойка

Стойка (rack) - это базовый конструктив, имеющий слоты для монтажа модулей контроллера.

Т**Тип данных (Data Type)**

Типы данных позволяют пользователям определять, как значение переменной или константы должно использоваться в пользовательской программе. Типы данных подразделяются на простые и структурированные типы.

У**Установки по умолчанию (Default Setting)**

Установки по умолчанию - практические базовые установки для параметров, которые всегда используются, если для параметров не заданы другие значения.

Ф**Функциональный блок (FB)**

Функциональные блоки - это компоненты пользовательской программы, являющиеся в соответствии со стандартом IEC "блоками с памятью". Памятью для функционального блока является назначенный экземплярный блок данных ("instance data block"). Функциональные блоки могут иметь, а могут и не иметь параметров.

Ц**Циклическое выполнение программы**

При циклическом выполнении пользовательская программа обрабатывается в постоянно повторяющемся цикле.

Э**Экземпляр DB (Instance Data Block)**

Экземпляр DB - блок данных, назначенный функциональному блоку, содержащий данные для этого специального блока.

C

CPU

Центральный процессор программируемого контроллера S7, имеющий в своем составе АЛУ, память, операционную систему и интерфейсы для модулей ввода/вывода (I/O).

CRC

Cyclic Redundancy Check - циклическая проверка целостности данных методом сверки контрольной суммы с гарантированной точностью распознавания ошибки.

Cycle Time (Время цикла)

Время цикла - это время, необходимое CPU для однократного сканирования пользовательской программы.

D

Data Block (DB) (Блок данных)

Блоки данных - это блоки, содержащие данные и параметры, которые используются в пользовательской программе. В отличие от других блоков блоки данных не содержат инструкций. Они подразделяются на глобальные (общие) блоки данных и экземплярные блоки данных. Доступ к данным, содержащимся в этих блоках может быть абсолютным или символьным. Сложные данные могут храниться в структурированных формах.

Data Type (Тип данных)

Типы данных позволяют пользователям определять, как значение переменной или константы должно использоваться в пользовательской программе. Типы данных подразделяются на простые и структурированные типы.

Default Setting (Установки по умолчанию)

Установки по умолчанию - практические базовые установки для параметров, которые всегда используются, когда для параметров не заданы другие значения.

Diagnostic Events (Диагностические события)

Диагностические события - это, например, ошибки в модуле или системные ошибки в CPU, которые вызваны, скажем, ошибкой в программе или изменением рабочего режима.

Download (Загрузка)

Термин загрузка означает загрузку загружаемых объектов (например, кодовых блоков) из программатора в загружаемую память (load memory) CPU.

F**Function Block (FB) (Функциональный блок)**

Функциональные блоки - это компоненты пользовательской программы, являющиеся в соответствии со стандартом IEC "блоками с памятью". Памятью для функционального блока является назначенный экземплярный блок данных ("instance data block"). Функциональные блоки могут иметь, а могут и не иметь параметров.

H**Hardware (Оборудование)**

Термин обозначающий все физическое и техническое оборудование (аппаратная часть) PLC.

I**Instance Data Block (Экземпляр DB)**

Экземпляр DB - блок данных, назначенный функциональному блоку, содержащий данные для этого специального функционального блока.

Interrupt (Прерывание)

Прерывание происходит, когда обработка программы в процессоре PLC прерывается внешним сигналом.

O**On-line/Off-line (Онлайн/Офлайн)**

Термины обозначающие режим работы с данными. Режим онлайн (или интерактивный режим) означает наличие обмена данными между PLC и программатором. Режим офлайн (или автономный режим) означает отсутствие обмена данными.

On-line Help (Интерактивная справка)

STEP 7 обеспечивает пользователя контекстно отображаемой справочной информацией при работе с программным обеспечением.

P**PtP-соединение**

Используя PtP-соединения (point-to-point - "точка к точке"), модуль CP формирует интерфейс между PLC и коммуникационным партнером.

Process Image (Образ процесса)

Образ процесса - это специальная область памяти в PLC. В начале выполнения циклической программы состояния сигналов входных модулей пересылается в таблицу входов области отображения процесса (process image input table). В конце выполнения циклической программы значения из таблицы выходов области отображения процесса (output table) пересылаются в выходные модули и формируют состояния выходов модулей вывода сигналов.

Project Configuration of Data Link

Project Configuration of Data Link (конфигурация связанных данных проекта) - это термин относящийся к размещению ID соединений в системном функциональном блоке. ID соединений позволяет отдельным системным функциональным блокам обеспечивать связь между двумя коммуникационными узлами.

PLC

Программируемый логический контроллер (PLC) - это электронное устройство управления с по крайней мере одним центральным процессором, различными модулями ввода/вывода и устройствами операторского управления и мониторинга.

R

RESTART (Перезапуск)

PLC при переходе от режима STOP к режиму RUN проходит режим перезапуска (RESTART).

S

Software (Программное обеспечение)

Программное обеспечение (Software) - это термин, объединяющий все программы, используемые в компьютерной системе. Сюда входят также операционная система и пользовательские программы.

STEP 7

STEP 7 - это программное обеспечение для программируемых контроллеров SIMATIC S7.

U

Upload (Выгрузка)

Выгрузка означает загрузку загружаемых объектов (например, кодовых блоков) из загружаемой памяти (load memory) CPU в программатор. Программа пользователя содержит все инструкции и объявления, касающиеся обработки сигналов, с помощью которых системой или процессом можно управлять. Пользовательская программа для системы SIMATIC S7 структурируется и подразделяется на более мелкие части, называемые блоками.

W

Work Memory (Рабочая память)

Рабочая память - это RAM-память в CPU, к которой процессор обращается, когда обрабатывает программу пользователя.

Предметный указатель

А

Автоматическое использование вспомогательных сигналов 4-6

Адрес ведомого устройства (Slave Address) 4-2

Б

Биты данных (Data Bits) 3-10

В

Ввод информации о выполнении задания в параметр STATUS 6-3

Возможная конфигурация системы 1-2

Время ожидания ответного сообщения (Reply Monitoring Time) 3-11

Вспомогательные сигналы RS 232C (Secondary Signals) 4-5

Выбор протокола 3-9

Д

Деинсталляция драйвера 3-2

Диагностика драйвера 7-1

Диагностика посредством выходного параметра "состояний" STATUS функциональных блоков 7-1

Диагностика посредством диагностического буфера CP 7-1

Диагностика посредством области для сообщений об ошибках SYSTAT 7-6

Диагностика посредством сообщений об ошибках в области SYSTAT (только для CP 441-2) 7-1

Диагностика посредством элементов индикации модуля 7-1

Диагностика с использованием элементов индикации CP 341 7-2

Диагностика с использованием элементов индикации CP 441-2 7-4

Диагностические сообщения системных функциональных блоков CP 441-2 7-5

Диагностические сообщения функциональных блоков CP 341 7-3

З

Завершение задания для функций записи 6-3

Завершение задания для функций считывания 6-3

Загружаемая память (Loading Memory) 1-3

Загрузка данных конфигурации и параметризации для CP 341 3-15

Загрузка данных конфигурации и параметризации для CP 441-2 3-16

Задание BRCV 8-12

Задание BSEND 8-12

Задание P_RCV_RK 8-3

Задание P_SND_RK 8-3

Задание STATUS 8-13

Защитный ключ-заглушка 1-3, 2-1

Значения времени пересылки A-1

И

Индикатор групповых сообщений (Group alarm) SF 7-2

Индикатор групповых сообщений EXTf 7-4

Индикатор групповых сообщений FAULT 7-4

Индикатор групповых сообщений INTf 7-4

Индикация приема данных 6-4

Индикация сообщений об ошибках 7-4

Инициализация (Initialization) 3-17

Инсталляция 2-1

Инсталляция драйвера 4-1

Интерфейс CPU - CP 6-1

Интерфейс CPU - CP для модуля CP 341 6-1

Интерфейс CPU - CP для модуля CP 441-2 6-5

Интерфейс RS232C / TTY 2-1

Интерфейс X27 (RS 422/485) 3-13

Интерфейс X27 / RS422 (четырёхпроводный режим) 2-2

Интерфейс X27 / RS485 (двухпроводный режим) 2-1

Интерфейсы 2-1

Использование вспомогательных сигналов RS 232C 4-5

Используемые блоки 8-1, 8-9

К

Коды ошибок в области SYSTAT для "CPU Job Errors" ("Ошибки в работе CPU") 7-8

Коды ошибок в сообщениях 4-4

Конец сообщения 4-3

Конфигурация оборудования 3-4

Конфигурация оборудования 3-18

Конфигурирование данных коммуникаций 3-2

Конфигурирование данных коммуникаций проекта 3-7

М

Множитель для времени ожидания символа (Multiplier Character Delay Time) 3-12

Н

Назначение параметров (Parameter Assignment) 3-17

Назначение параметров для CP 3-4

Назначение параметров для CP 341 3-4

Назначение параметров для CP 441-2 3-5

Назначение параметров для загружаемого драйвера 3-9

Назначение параметров для коммуникационного партнера 3-6

Назначение параметров запуска 3-18

Настройка драйвера 3-1

Номера ошибок в области SYSTAT для "General Processing Errors" ("Общие ошибки обработки") 7-11

Номера ошибок в области SYSTAT для "Receive Errors" ("Ошибки при приеме") 7-9

Нормальный режим ("Normal Operation") 3-12

О

Обзор параметров протокола обмена данными 3-11

Области данных 1-4

Оконечный резистор B-2

Описание программы 8-3, 8-12

Отображение и проверка информации в выходном параметре STATUS 7-6

Очистка (Deleting) области SYSTAT 7-7

П

Параллельная обработка запросов 6-1

Передача данных от CP к CPU посредством BRCV (CP 441-2) 6-8

Загружаемый драйвер для PtP-соединений CP (S7 - ведущее устройство) A5E00218410-04

- Передача данных от CP к CPU посредством P_RCV_RK (CP 341) 6-4
- Передача данных от CPU к CP посредством BSEND (CP 441-2) 6-5
- Передача данных от CPU к CP посредством P_SND_RK (CP 341) 6-2
- Подавление помех ("Interference Suppression") 3-12
- Полный дуплексный режим (RS422) 3-13
- Полудуплексный режим (RS485) 3-14
- Представление адресов 1-5
- Предустановка "R(A) 0V, R(B) 5V" (High) ("Высокое") 3-14
- Предустановка "R(A) 5V, R(B) 0V" (BREAK) ("Обрыв") 3-14
- Предустановки для входных линий 3-14
- Пример для модуля CP 341 8-1
- Пример для модуля CP 441-2 8-9
- Пример применения 8-1
- Пример программирования 8-4, 8-14
- Проверка четности (CRC Check) 4-3
- Протокол GOULD-MODBUS 1-4
- Протокол для ведущего устройства Modbus 3-9
- Протокол обмена данными 4-1
- Р**
- Рассылка сообщений (Broadcast Messages) 4-2
- С**
- Связь ведущего и ведомого устройств 4-1
- Скорость передачи (Transmission Rate) 3-10
- Сообщения в выходном параметре STATUS блоков SFB 7-5
- Специальные индикаторы (Special) 7-2
- Загружаемый драйвер для PtP-соединений CP (S7 - ведущее устройство) A5E00218410-04
- Специальный индикатор RXD 7-4
- Специальный индикатор TXD 7-4
- Средства диагностики модуля CP 341 7-2
- Средства диагностики модуля CP 441-2 7-4
- Стоповые биты (Stop Bits) 3-10
- Структура области SYSTAT 7-7
- Структура параметра STATUS 7-3
- Структура сообщения 4-1
- Схема автоматического использования вспомогательных сигналов RS 232C 4-7
- Схема многоточечного соединения RS422 (MODBUS) B-1
- Схема многоточечного соединения RS485 (MODBUS) B-2
- Схема нумерации классов и номеров событий 7-3
- Схема нумерации событий в области SYSTAT 7-7
- Схемы многоточечных соединений B-1, B-2
- Считывание информации из области SYSTAT 7-6
- Т**
- Таблицы событий (Events) / ошибок (Errors) 7-8
- Технические данные A-1
- Требования к программному и аппаратному обеспечению 1-3
- У**
- Установка защитного ключа-заглушки 2-1
- Ф**
- Функции диагностики 7-1
- Функции драйвера 1-1
- Функциональные коды 1-1, 1-4, 4-2, 5-1

- Функциональный код 01 - Read Output Status (считывание состояния выхода) 5-1
- Функциональный код 02 - Read Input Status (считывание состояния входа) 5-3
- Функциональный код 03 - Read Output Registers (считывание выходных регистров) 5-5
- Функциональный код 04 - Read Input Registers (считывание входных регистров) 5-7
- Функциональный код 05 - Force Single Coil (назначение для одной катушки) 5-9
- Функциональный код 06 - Preset Single Register (предустановки для одного регистра) 5-10
- Функциональный код 07 - Read Exception Status (считывание ошибочного состояния) 5-11
- Функциональный код 08 - Loop Back Diagnostic Test (тестирование коммуникационного соединения) 5-12
- Функциональный код 11 - Fetch Communications Event Counter (считывание значения счетчика коммуникационных событий) 5-13
- Функциональный код 12 - Fetch Communications Event Log (считывание протокола коммуникационных событий) 5-14
- Функциональный код 15 - Force Multiple Coils (назначения для нескольких катушек) 5-16
- Функциональный код 16 - Preset Multiple Registers (предустановки для нескольких регистров) 5-18

Х

Характеристики запуска CP 3-17

Ч

Четность (Parity) 3-11

Загружаемый драйвер для PtP-соединений CP
(S7 - ведущее устройство)
A5E00218410-04

D

Dongle (Защитный ключ-заглушка) 1-3

F

Function Codes 1-1

G

GOULD-MODBUS (Протокол) 1-4

L

Loading Memory (Загружаемая память)
1-3

P

*Point-to-Point Communication, Parameter
Assignment (CP 341: PtP-
соединение, назначение
параметров)* 3-2

R

RTU-формат 1-1

S

S7wfp1a.dll 3-1

S7wfp1x.cod 3-1

S7wfp2x.cod 3-1

