

А. А. Гайнуллина, А. Д. Байтимиров

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ПЕРЕДАЧИ ДАННЫХ МЕЖДУ ПРОГРАММИРУЕМЫМИ ЛОГИЧЕСКИМИ КОНТРОЛЛЕРАМИ ПО ПРОТОКОЛУ MODBUS

Ключевые слова: протокол Modbus, PCY Centum VP, контроллер Stardom FCN, передача данных.

В данной статье рассматриваются основы протокола Modbus и его базовые принципы работы, а также приводится конкретный пример работы с ним.

Keywords: protocol Modbus, DCS Centum VP, controller Stardom FCN, the transfer of data.

In this article describe the basics of the Modbus protocol and its basic operating principles, as well as a specific example of working with him.

На данный момент, справочная документация, однозначно описывающая последовательность подключения и разработку алгоритмов передачи данных между контроллером PCY (распределенной системы управления) и ПЛК (программируемым логическим контроллером) по протоколу *Modbus* в свободном доступе отсутствует. По этой причине авторами статьи предлагается рассмотреть указанную процедуру, на примере взаимодействия контроллера PCY *Centum VP* и ПЛК *Stardom FCN*.

Протокол *Modbus* разработан для использования в программируемых логических контроллерах и в настоящее время является достаточно распространенным.

Протокол *Modbus* имеет два режима передачи данных: *Modbus ASCII* (*American Standart Code for information Interchange*), *Modbus RTU* (*Remote Terminal Unit*). На основе международного стандарта и с учетом того факта, что режим *ASCII* в России практически не используется, авторами будет рассматриваться только режим *RTU*.

Преимуществами протокола *Modbus* являются отсутствие необходимости в специальных интерфейсных контроллерах и простота программной реализации. Он позволяет унифицировать команды обмена благодаря стандартизации номеров (адресов) регистров и стандартизации функций их чтения и записи.

Главным недостатком данного протокола является сетевой обмен по типу «ведущий/ведомый», что не позволяет ведомым устройствам передавать данные ведущему по мере их появления. Это потребует увеличения интенсивности опроса ведомых устройств ведущим.

Протокол *Modbus* широко используется для передачи данных по последовательной линии связи, основанной на интерфейсах *RS-485*, *RS-422*, *RS-232*. В настоящее время интерфейс *RS-485* позволяет объединять несколько устройств в единую сеть и добиться более высокой скорости передачи на большие расстояния.

Последовательный интерфейс имеет возможность передачи данных по двухпроводной или четырехпроводной схеме.

Двухпроводный способ соединения используется для полудуплексной передачи, когда информация может передаваться в обоих направле-

ниях, но в разное время. Для полнодуплексной (дуплексной) передачи используют четыре линии связи: по двум информация передается в одном направлении, по двум другим – в обратном.

Недостатком четырехпроводной схемы является необходимость жесткого указания ведущего и ведомых устройств на стадии проектирования системы, в то время как в двухпроводной схеме любое устройство может быть как в роли ведущего, так и ведомого. Достоинством четырехпроводной схемы является возможность одновременной передачи и приема данных, что бывает необходимо при реализации некоторых сложных протоколов обмена.

Магистральный кабель протокола *Modbus* должен содержать три проводника в общем экране, два из которых представляют собой витую пару, а третий соединяет общие («земляные») выводы всех интерфейсов *RS-485* в сети.

Протокол *Modbus* предполагает, что только одно ведущее устройство и до 247 ведомых могут быть объединены в промышленную сеть. Обмен данными инициируется ведущим. Ведомые устройства не начинают передачу данных до тех пор, пока не получат запрос от ведущего. Ведомые устройства также не могут обмениваться данными друг с другом. Поэтому в любой момент времени в сети *Modbus* может происходить только один акт обмена.

Сообщения по *Modbus RTU* передаются в виде кадров, для каждого из которых известно начало и конец. Признаком начала кадра является пауза (тишина) продолжительностью не менее 3,5 шестнадцатеричных символов (14 бит). Кадр должен передаваться непрерывно. Если при передаче кадра обнаруживается пауза продолжительностью более 1,5 шестнадцатеричных символов (6 бит), то считается, что кадр содержит ошибку и должен быть отклонен принимающим модулем. Формат кадра показан на рис. 1.

Адрес - это поле, содержащее номер устройства, которому адресован запрос. Каждое устройство в сети должно иметь уникальный адрес. Устройство отвечает только на те запросы, которые поступают на его адрес, во избежание конфликтов. При этом ведомое устройство в своем ответе так же посылает поле «Адреса».

«Код функции» содержит номер функции протокола *Modbus*. Функция может запрашивать

данные или давать команду на определенные действия. Коды функций являются числами в диапазоне от 1 до 127.



Рис. 1 – Формат кадра протокола Modbus RTU; PDU (Protocol Data Unit) – элемент данных протокола; ADU (Application Data Unit) – элемент данных приложения

В поле «Данные» содержится информация, которую ведущее устройство передает ведомому. Длина этого поля зависит от типа передаваемых данных.

Поле «Контрольная сумма» является важным элементом протокола: в нем содержится информация, необходимая для проверки целостности сообщения и отсутствия ошибок передачи. Поле «Контрольная сумма» содержит контрольную сумму длиной 2 байта.

На рис. 2 показана структура данных в режиме RTU.

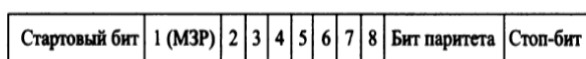


Рис. 2 – Последовательность битов в режиме RTU. МЗР – младший значащий разряд. При отсутствии бита паритета на его место записывается второй стоп-бит

По умолчанию в режиме RTU бит паритета устанавливаются равным 1, если количество двоичных единиц в байте нечетное, и равным 0, если оно четное. Такой паритет называют четным (even parity), а метод контроля называют контролем четности.

При четном количестве двоичных единиц в байте бит паритета может быть равен 1. В этом случае говорят, что паритет является нечетным (odd parity) [1].

Контроль четности может отсутствовать вообще. В этом случае вместо бита паритета должен использоваться второй стоповый бит.

Для передачи данных по протоколу Modbus используются устройства со скоростью обмена 9600бит/с или 19200бит/с, при этом скорость обмена 19200 бит/с устанавливается «по умолчанию».

Далее рассмотрим работу указанного протокола на примере организации передачи данных с контроллера Stardom FCN в контроллер PCY Centum VP, с целью отображения полученных данных в среде визуализации Centum VP [2].

Centum VP – система управления производством реального времени, включающая такие компоненты как станция оператора и станция управления (FCS - field control station), связанные посредством сети Vnet/IP [3].

Stardom является базированной на сетевых средствах управляющей системой, которая пози-

ционируется как промежуточная между PLC (Programmable Logic Controller) / SCADA (supervisory control and data acquisition) системами и PCY системами управления.

Stardom состоит из следующих полностью независимых компонентов:

- Автономный контроллер FCN и FCJ;
- Программного обеспечения многоцелевого сервера данных VDS;
- Папок приложений.

FCJ (Field Control Junction) – малогабаритный автономный контроллер с небольшим фиксированным количеством аналоговых и дискретных входов/выходов и коммуникационными портами для подключения удаленных устройств ввода/вывода, PLC и Fieldbus.

FCN (Field Control Node) – каркасно-монтажный блок, расширяемый до трех каркасов, для установки модулей ввода/вывода также включает в себя коммуникационные модули для удаленного ввода/вывода, подключения PLC контроллеров и Fieldbus.

VDS (Versatile Data Server) – пакет программного обеспечения, работающего под ОС Windows и оснащенного человеко-машинным интерфейсом, функциями OPC-сервера, сбора данных от различных контроллеров и устройств, а также управления историческими данными.

Stardom может работать со SCADA системами сторонних производителей. SCADA HMI, FAST/TOOLS обеспечивает аналогичную VDS функциональность.

Также в качестве SCADA системы можно использовать операторский интерфейс PCY Centum VP, организовав передачу данных с контроллера Stardom FCN в контроллер PCY Centum VP по протоколу Modbus (рис. 3).

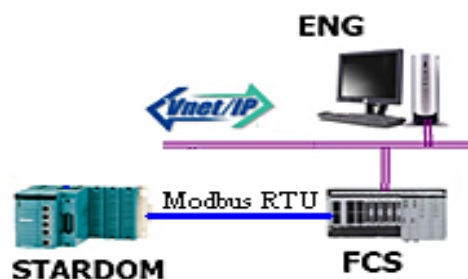
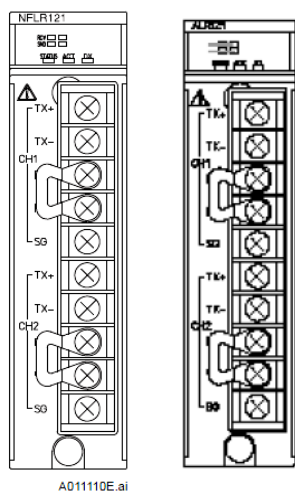


Рис. 3 - Интеграция Stardom FCN с PCY Centum VP по протоколу Modbus

Связь контроллера Stardom FCN с контроллером PCY Centum VP по протоколу Modbus осуществляется с помощью соответствующих коммуникационных модулей NFLR121 и ALR121. Внешние интерфейсы этих модулей изображены на рис. 4. Их соединение между собой осуществляется по четырехпроводному способу связи.

Перед организацией передачи данных, необходимо осуществить настройку модулей. Настройка модуля ALR121 производится в программе System View PCY Centum VP, а модуля NFLR121 - в программе Resource Configuration контроллера Stardom FCN. В настройках указывается бит паритета,

стоповый бит, способ связи, время ожидание ответа, интервал подключения, номер станции, адрес устройств, скорость передачи данных, и при этом оба модуля должны иметь одинаковые режимы передачи данных. При этом важно изменить названия портов в *Stardom FCN* с «COM1» и «COM2», на любые другие сочетания, например «*MODBUS1*» и «*MODBUS2*», так как «*COM1*» и «*COM2*» зарезервированы под уже существующие на корпусе контроллера порты. Без замены названий портов передача данных осуществляться не будет.



Название клеммы	Название сигнала
TX+	Посылка данных
TX-	Посылка данных
RX+	Прием данных
RX-	Прием данных
SG	Сигнальное заземление

Рис. 4 – Коммуникационный модуль

Связь *Stardom FCN* со станцией управления *Centum VP (FCS)* осуществляется с использованием пакета поддержки связи *Modbus*. При этом используется последовательный режим передачи данных *RTU*, где контроллер *Centum VP* выступает в качестве ведущего устройства (*Master*), а *Stardom FCN* – в роли ведомого (*Slave*).

Система записи в память допускает использование четырех разных типов данных: группа логических ячеек, состояние дискретного входа, входные регистры и регистры хранения временной информации.

Для сбора и обработки информации, поступающей по каналам последовательной связи, в *Stardom FCN* предусмотрен специализированный пакет поддержки связи *Modbus – Modbus Communication Portfolio*, в котором содержится библиотека с функциональным блоком, с помощью которого осуществляется создание и конфигурирование связи между контроллером *Stardom FCN* и станцией управления *FCS*.

Функциональный блок *SD_CMDBSM_BS_OPEN_1*, с помощью которого осуществляется передача данных с контроллера

Stardom FCN на станцию управления, представлен на рисунке 5, а в таблице 1 указаны основные входные/выходные сигналы данного блока.

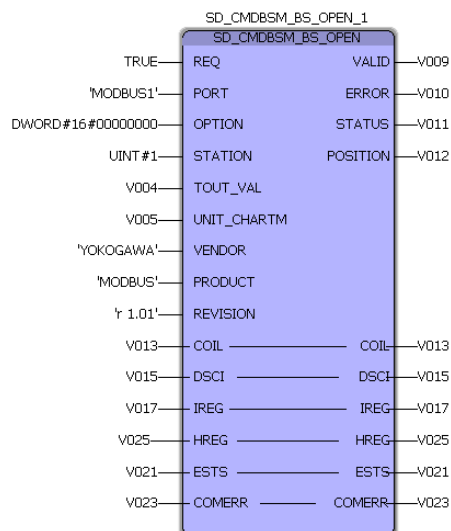


Рис. 5 - Функциональный блок поддержки связи *Modbus*

Таблица 1 – Обозначение входных/выходных значений функционального блока *SD_CMDBSM_BS_OPEN_1*

Наименование	Тип данных	Описание
1	2	3
<i>REQ</i>	<i>BOOL</i>	Всегда указывается значение TRUE
<i>PORT</i>	<i>STRING</i>	Указывается название последовательного порта
<i>OPTION</i>	<i>DWORD</i>	Не указывается
<i>STATION</i>	<i>UNIT</i>	Адрес ведомой станции (1-247)
<i>TOUT_VAL</i>	<i>REAL</i>	Указывает значение времени прерывания (от 0.0 до 100000.0 мс)
<i>UNIT_CHARTM</i>	<i>BOOL</i>	Указывает на ед. измерения времени прерывания. FALSE указывает на то, что используются миллисекунды.
<i>VENDOR</i>	<i>STRING</i>	Указывает название фирмы-производителя, до 80 символов
<i>PRODUCT</i>	<i>STRING</i>	Указывает название продукта, до 80 символов
<i>REVISION</i>	<i>STRING</i>	Указывает тип редакции, до 80 символов

Окончание табл. 1

1	2	3
<i>VALID</i>	<i>BOOL</i>	Указывает на нормальное состояние коммуникационного порта. TRUE указывает на нормальное состояние.
<i>ERROR</i>	<i>BOOL</i>	Указывает на состояние коммуникационного порта. TRUE указывает на ошибку работы порта.
<i>STATUS</i>	<i>WORD</i>	Указывает на ошибку во время передачи данных
<i>POSITION</i>	<i>STRING</i>	Указывает позицию коммуникационного порта в следующем формате 'UNIT-SLOT-PORT'
<i>COIL</i>	<i>SD_CMDBSM_BIT9999</i>	Указывает состояние группы логических ячеек
<i>DSCI</i>	<i>SD_CMDBSM_BIT9999</i>	Указывает состояние дискретного входа
<i>IREG</i>	<i>SD_CMDBSM_WORD9999</i>	Указывает входные регистры
<i>HREG</i>	<i>SD_CMDBSM_WORD9999</i>	Указывает регистры временного хранения
<i>ESTS</i>	<i>SD_CMDBSM_BIT8</i>	Указывает состояние исключительного события
<i>COMERR</i>	<i>SD_CMDBSM_BS_COMERR</i>	Информация об ошибке буфера

Сигналы, которые необходимо вывести на экран станции управления, указываются в регистре временного хранения *HREG*. Для этого в программе *Logic Designer* (рисунок 6) коды аналого-цифрового преобразования, с помощью функционального блока (*NPAS_AI_ANLG*) преобразовываются в аналоговые значения, а в блоке *REAL_TO_WORD* в 16 битное слово. В таблице 2 указаны типы передаваемых данных, которые поддерживает протокол *Modbus RTU*.

Далее преобразованные значения измеренных величин подсоединяются к одной из ячеек массива переменных *V025[1-9999]* функционального блока *SD_CMDBSM_BS_OPEN*.

После чего осуществляется загрузка проекта в контроллер и передача измеренных величин в среду визуализации станции управления *Centum VP*.

Данная статья может быть использована в качестве руководства при проведении работ по организации обмена данными между контроллером РСУ (распределенной системы управления) *Centum VP* (модуль *ALR121*) и ПЛК (программируемый ло-

гическим контроллером) *Stardom FCN* (модуль *HNLR121*), с помощью протокола *Modbus RTU*.

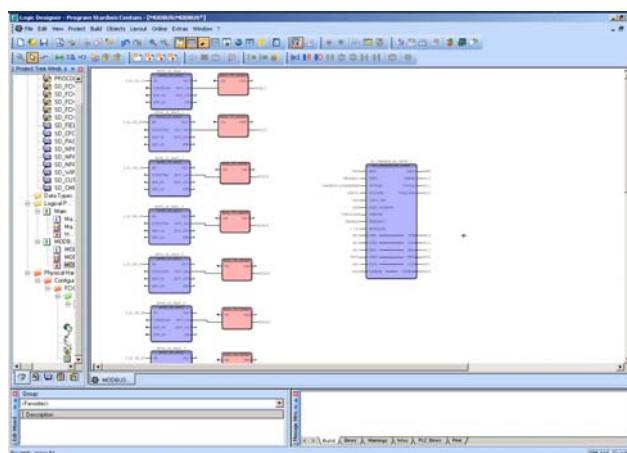


Рис. 6 – Внешний интерфейс программной среды *Logic Designer*

Таблица 2 - Типы передаваемых данных

	Тип данных	Тип доступа
Дискретные входы (<i>Discrete Inputs</i>)	один бит	только чтение
Логическая ячейка (<i>Coils</i>)	один бит	чтение и запись
Регистры ввода (<i>Input Registers</i>)	16-битное слово	только чтение
Регистры хранения (<i>Holding Registers</i>)	16-битное слово	чтение и запись

Также данный материал послужит основой для методического указания, в котором будут рассмотрены следующие вопросы:

- изучение протокола *Modbus RTU*, распределенной системы управления *Centum VP*, контроллера *Stardom FCN*;
- осуществление настройки контроллера *CENTUM VP* на запрос и считывание данных с контроллера *STARDOM FCN*;
- разработка среды визуализации для отображения считываемых данных, настройке контроллера *STARDOM FCN* с помощью программного продукта *Resource Configuration*,
- разработка алгоритмов передачи данных и программированию контроллера с помощью программного компонента *Logic Designer*.

Методическое указание может быть применено для выполнения лабораторных работ, студентами направления 220400.62 «Управление в технических системах», по следующим дисциплинам:

- «Вычислительные машины, системы и сети»,
- «Вычислительная техника в САУТП»,
- «Системы связи УВМ с объектом».

Литература

1. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием / В.В. Денисенко. – М.: Горячая линия – Телеком, 2009. – с.105-110

2. Нургалиев Р.К. Лабораторный стенд для изучения систем автоматизации узлов коммерческого учета жидких продуктов / Р.К. Нургалиев, В.В. Кузьмин, Ю.А. Куликов, А.В. Чупаев, Р.Р. Галямов, А.А. Гайнуллина // Вестник Казан.технол.ун-та. –2013. –Т. 16, № 1. – С. 67-70.
3. Нургалиев Р.К. Промышленные сети передачи данных / Р.К. Нургалиев, Р.Н. Зарипов, Д.Б. Флак, Э.У. Даутова // Вестник Казан.технол.ун-та. –2013. –Т. 16, № 1. – С. 252-255.

© А. А. Гайнуллина - асс. каф. САУТП КНИТУ, alinagainullina@rambler.ru; А. Д. Байтимиров – магистрант той же кафедры.