

УДК 681.3

Ковыркин С.В., Буторин Д.В., Попов А.В., Коденёв К.Ф.

Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация

ПЕРЕДАЧА ДАННЫХ ПО ИНТЕРФЕЙСУ RS485 И ПРОТОКОЛУ MODBUS RTU МЕЖДУ МИКРОПРОЦЕССОРНОЙ ПЛАТОЙ ARDUINO И ПРОМЫШЛЕННЫМ ЛОГИЧЕСКИМ КОНТРОЛЛЕРОМ

***Аннотация.** В статье описывается практический опыт выбора и настройки протокола передачи данных между микропроцессорной платой Arduino и промышленным логическим контроллером (ПЛК). Данная задача рассматривалась при создании макета автоматизированного крана мостового типа, разрабатываемого в рамках гранта РНФ*. Проводился эксперимент с замерами скорости передачи данных в двух режимах: Arduino выступает в роли мастера, а ПЛК - в роли слейва и наоборот. Основными требованиями к системе передачи данных были скорость, максимально близкая к реальному времени и отсутствие ошибок при передаче. Нарушение хотя бы одного из них сильно сказывалось на качестве управления технологическим процессом (успокоения колебаний груза). В статье представлено описание тестируемой системы, описан алгоритм подсчета времени на прием сообщения и приведены результаты тестирования, представляющие интерес для разработчиков промышленных автоматизированных систем. Полученные результаты позволяют сделать выводы о возможности использования рассматриваемого протокола и интерфейса для передачи данных в задачах управления быстропротекающими технологическими процессами.*

***Ключевые слова:** промышленная сеть, промышленный логический контроллер, Arduino, передача данных, интерфейс RS485, протокол ModBus RTU*

Исследование выполнено за счет гранта Российского научного фонда №23-29-00654, <https://rscf.ru/project/23-29-00654/>

S.V. Kovyrrshin, D.V. Butorin, A.V. Popov, K.F. Kodenyyov

Irkutsk State Transport University, Irkutsk, Russian Federation

DATA TRANSFER VIA RS485 INTERFACE AND MODBUS RTU PROTOCOL BETWEEN ARDUINO MICROPROCESSOR BOARD AND INDUSTRIAL LOGIC CONTROLLER

***Abstract.** This article describes practical experience in selecting and configuring the data transmission protocol over the RS485 interface and Modbus RTU protocol between the Arduino microcontroller board and an industrial logic controller (PLC). The experiment was conducted using a sensor that measures parameters of a fast-flowing process. The main technical requirements were to achieve a transmission speed as close to real-time as possible and to ensure error-free communication. Violating any of these requirements affects the quality of process control. The article provides a description of the tested system, an algorithm for calculating the message reception time, and the results of the testing, which are of interest to developers of industrial automated systems. The obtained results allow conclusions to be drawn regarding the suitability of the chosen protocol and interface for data transmission between Arduino and the industrial logic controller.*

***Keywords:** industrial network, industrial logic controller, Arduino, data transmission, RS485 interface, ModBus RTU protocol*

This work was supported by the Russian Science Foundation, project no. 23-29-00654; <https://rscf.ru/project/23-29-00654/>

Введение

Промышленные системы автоматизации технологических процессов обладают определенной спецификой, в частности они могут собирать данные с большого количества информационных устройств, обрабатывать ее, производить диагностику и удалённое конфигурирование датчиков и исполнительных механизмов, осуществлять передачу информации на верхний уровень управления – уровень человеко-машинного интерфейса и/или компьютер верхнего уровня. Такая специфика привела к созданию специализированных промышленных сетей и протоколов передачи данных (Fieldbus). В настоящее время имеется большое количество как специализированных промышленных сетей, например, для автоматизации зданий,

автомобилей, так и универсальных. Подключение информационных и управляющих устройств к промышленной сети осуществляется, с помощью сетевых интерфейсов. Они состоят из устройств аппаратной реализации, т.е. набора электронных компонентов и связанного с ними программного обеспечения. Наибольшее распространение в настоящее время получили последовательные интерфейсы RS-485, RS-232, RS-422, Ethernet, ProfiBus, EtherCAT и др. [1]. Кроме этого, для обмена информацией взаимодействующие устройства должны иметь одинаковый протокол, то есть набор правил, по которым осуществляется обмен информацией. В протоколе определяется синтаксис и семантика сообщений, операции управления, синхронизация и состояния при коммуникации [2-6], это значит, что если надо подключить устройство к имеющейся сети, то необходим соответствующий интерфейс и протокол обмена.

При создании макетов мехатронных и робототехнических устройств, часто используются микропроцессорные платы семейства Arduino (далее Arduino) [7]. Они дешевы, доступны и имеют необходимый для реализации не сложных задач функционал. Большинство начинающих осваивать программирование микропроцессоров для обучения и построения макетов выбирают именно эти платы. По мере усложнения проектов возникают задачи передачи информации другими вычислителями. Связать две и более Arduino между собой можно через имеющиеся штатные интерфейсы (I2C, SPI, UART), это не вызывает особых затруднений, для этого имеются готовые библиотеки. Но в случае, если необходимо организовать обмен данными между Arduino и промышленным логическим контроллером (ПЛК), у которого свои интерфейсы и протоколы, то задача становится не такой простой, необходимо предусмотреть в системе дополнительные преобразователи интерфейсов и адаптировать протоколы. Этот вопрос требует детального изучения и один, частный случай рассматривается в статье.

Описание исследуемой системы

Создана установка (макет крана мостового типа) для исследования законов гашения колебаний грузов на гибком подвесе [8, 9]. Необходимо передавать данные от Arduino с микроконтроллером ATmega328 к ПЛК XinJe XSLH-30A32 с максимально возможной скоростью и минимальным количеством ошибок.

В ПЛК обычно реализуются определенные интерфейсы для передачи, из цифровых интерфейсов наибольшее распространение получили RS-232 [10], RS-485[11], EtherCAT [12] и др. У Arduino имеется UART (Universal Asynchronous Receiver-Transmitter), который относится к последовательному порту уровня TTL, и напрямую подключить к интерфейсу стандарта RS-232 невозможно из-за разницы логических уровней сигнала. Уровень TTL составляет 3,3 В, а RS232 - отрицательный логический уровень, определяется диапазоном напряжения + 5 ~ + 12В как низкий уровень, а -12 ~ -5V – как высокий уровень.

Для преобразования логических уровней применяются специальные согласующие микросхемы, наиболее известные MAX232, для преобразования уровней TTL в уровни стандарта RS-232 и MAX485, для преобразования в уровни интерфейса RS-485.

Для проведения тестирования будем использовать интерфейс RS-485, так как он является промышленным стандартом, имеет хорошую помехоустойчивость и повышенную скорость передачи до 4 мегабит в секунду. Структура тестируемой системы представлена на рис. 1. В качестве протокола передачи данных будет использоваться самый распространенный в промышленности протокол – ModBus RTU.

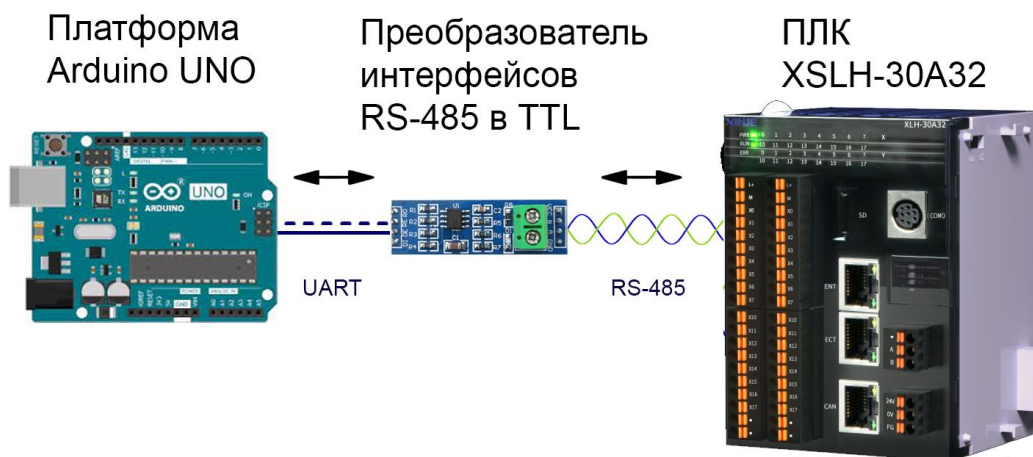


Рис. 1. Схема структурная тестируемой системы

ModBus интерфейс и одноименная сеть, строится принципу одно ведущее (мастер) и 247 ведомых (слейвов) устройств. Мастер инициирует обмен данными в сети посылая запрос и ожидает ответ в течение некоторого промежутка времени («время таймаута»). Если в этот промежуток времени ответ не получен, то считается, что связь с ведомым отсутствует. Сами ведомые устройства не могут самостоятельно инициировать передачу данных, а только посылают ответы на запросы.

Тестирование

Тестирование интерфейса RS-485 будет производиться в двух режимах.

В первом режиме, Arduino выступает в роли мастера, а ПЛК - в роли слейва.

Во втором режиме, наоборот, Arduino является слейвом, а ПЛК - мастером, процесс происходит аналогично, но с другой логикой передачи данных.

Целью тестирования является подсчет количества принятых сообщений (messageCounter) и вычисления среднего времени на прием одного сообщения (timMsPerMessage) в обоих режимах работы интерфейса RS-485. Блок-схема алгоритма для теста представлена на рис. 2.

Сам алгоритм заключается в следующем. При включении таймера с задержкой на 10 с (startTimer), код проверяет текущее значение переменной modbusVal и сравнивает его с предыдущим значением (modbusValOld). Если значения отличаются, увеличивается счетчик принятых сообщений (messageCounter) на 2. По истечении времени таймера, вычисляется среднее время на прием одного сообщения (timMsPerMessage).

Стоит отметить, что с Arduino отправляет постоянно изменяющееся значение (0/1), поэтому данный метод измерения времени передачи – корректный.

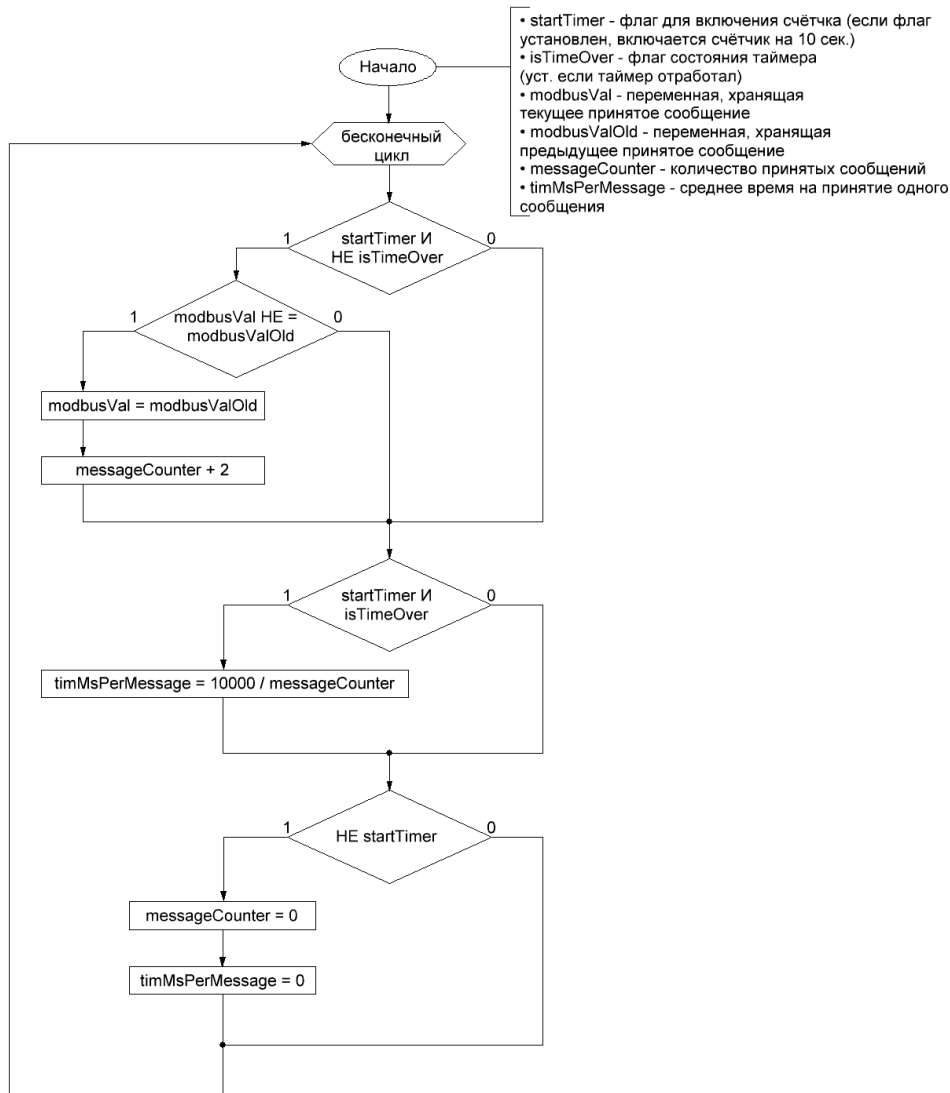


Рис. 2. Блок-схема алгоритма подсчёта времени на прём сообщения

Таблица 1. Формат сообщений запросов и ответов

Режим 1: ПЛК – MASTER, Arduino – SLAVE	Режим 2: Arduino – MASTER, ПЛК - SLAVE
Запрос: Адрес slave - 1 байт Код функции (0x03 чтение нескольких holding registers) - 1 байт Начальный адрес регистра - 2 байта Количество регистров N* - 2 байта CRC16 - 2 байта Ответ: Адрес slave - 1 байт Код функции (0x03) - 1 байт Количество передаваемых байт данных - 1 (2N*) Данные - (2N*) байт CRC16 - 2 байта	Запрос: Адрес slave - 1 байт Код функции (0x10 запись нескольких holding registers) - 1 байт Начальный адрес - 2 байта Количество регистров - 2 байта Байтовое количество данных - 1 байт Данные - до 0xFF байт CRC16 - 2 байта

* примечание: N - количество запрошенных регистров

Конфигурация последовательного порта: 115200 Бод, 8N1.

Тестирование проводилось в среде CODESYS V3.5 SP16 Patch 3 при помощи описанного выше алгоритма. Результат очередного измерения можно увидеть на рисунке 3. Была произведена серия из 10 измерений, результаты которых записаны в таблице 2.

Device.Application.PLC_PRG			
Выражение	Тип	Значение	Подготовленное ...
Timer	TON		
startTimer	BOOL	TRUE	FALSE
isTimeOver	BOOL	TRUE	
modbusVal	WORD	0	
modbusValOld	WORD	1	
messageCounter	WORD	3284	
timMsPerMessage	REAL	3.045067	

Рис. 3. Окно с переменными

Таблица 2. Среднее время на передачу сообщения

№ измерения	Время на передачу сообщения, мс	
	Режим 1	Режим 2
1	3,533	3,046
2	7,288	3,045
3	5,01	3,046
4	3,533	3,046
5	2,997	3,045
6	4,295	3,045
7	2,997	3,045
8	5,351	3,045
9	6,176	3,045
10	5,856	3,046

Заключение

Полученные в тестах измерения позволяют сделать следующие выводы:

В режиме 1 (Arduino в роли мастера, ПЛК в роли слейва):

- среднее время на прием одного сообщения составляет примерно 4,422 мс.
- значения отличаются из-за ошибок в передаче данных, что может быть связано с некорректной работой интерфейса RS-485, например, не готовностью слейва отправить данные. Настроить безошибочную передачу данных в этом режиме не удалось, они периодически появлялись при различных настройках протокола.

В режиме 2 (Arduino в роли слейва, ПЛК в роли мастера):

- среднее время на прием одного сообщения составляет примерно 3,046 мс.
- значения в данном режиме стабильны и не отличаются.

Таким образом, если на шине не будет других устройств, кроме Arduino, которые необходимо подключить к ПЛК по RS-485 и важна скорость передачи данных, рекомендуется использовать первый режим, так как он работает быстрее и стабильнее.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Зимин В. В. Промышленные сети : учеб. пособие для студентов вузов/ В. В. Зимин ; Министерство науки и высшего образования РФ. – Н. Новгород : Нижегородский государственный технический университет, 2006. – 252 с.
2. ГОСТ Р МЭК 870-5-1-95 Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 1. Форматы передаваемых кадров.
3. ГОСТ Р МЭК 870-5-2-95 «Устройства и системы телемеханики. Часть 5. Протоколы передачи. Раздел 2. Процедуры в каналах передачи».
4. ГОСТ Р МЭК 870-1-1-93 Устройства и системы телемеханики. Часть 1. Основные положения. Раздел 1. Общие принципы.
5. ГОСТ Р МЭК 870-3-93 «Устройства и системы телемеханики. Часть 3. Интерфейсы (электрические характеристики)».
6. ГОСТ Р МЭК 870-4-93 «Устройства и системы телемеханики. Часть 4. Технические требования».
7. S.P. Kruglov, S.V. Kovyrshin, "Adaptive trailer backup control, " 2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 2018, pp. 1-6.
8. Круглов С.П., Ковыршин С.В., Аксаментов Д.Н. Адаптивное управление двухмаятниковым подвесом мостового крана. Мехатроника, автоматизация, управление, 2022, т. 23, № 9, с. 451-461.
9. Круглов С.П., Ковыршин С.В. Адаптивное управление мостовым краном при переносе длинномерного вертикально расположенного груза. Известия высших учебных заведений. Машиностроение, 2023, № 11, с. 34–44, doi: 10.18698/0536-1044-2023-11-34-44.
10. EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Washington, USA: Electronic Industries Association, Engineering Department. 1969. — URL: http://bitsavers.trailing-edge.com/pdf/datapro/communications_standards/2740_EIA_RS-232-C.pdf (дата обращения: 03.09.2023).
11. Electronic Industries Association. Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems. EIA Standard RS-485. — URL: http://www.softelectro.ru/rs485_en.html (дата обращения: 03.09.2023).
12. Janssen, D.; Buttner, H. Real-time Ethernet: the EtherCAT solution, Computing & Control Engineering Journal. 2004. Vol. 15. P. 16–21.

REFERENCES

1. Zimin V. V. Promyshlennye seti : ucheb. posobie dlya studentov vuzov/ V. V. Zimin; Ministerstvo nauki i vysshego obrazovaniya RF. – N. Novgorod : Nizhegorodskij gosudarstvennyj tekhnicheskij universitet, 2006. – 252 s.
2. GOST R MEK 870-5-1-95 Ustrojstva i sistemy telemekhaniki. CHast' 5. Protokoly peredachi. Razdel 1. Formaty peredavaemyh kadrov.
3. GOST R MEK 870-5-2-95 «Ustrojstva i sistemy telemekhaniki. CHast' 5. Protokoly peredachi. Razdel 2. Procedury v kanalah peredachi».
4. GOST R MEK 870-1-1-93 Ustrojstva i sistemy telemekhaniki. CHast' 1. Osnovnye polozheniya. Razdel 1. Obshchie principy.
5. GOST R MEK 870-3-93 «Ustrojstva i sistemy telemekhaniki. CHast' 3. Interfejsy (elektricheskie harakteristiki)».
6. GOST R MEK 870-4-93 «Ustrojstva i sistemy telemekhaniki. CHast' 4. Tekhnicheskie trebovaniya».S.P. Kruglov, S.V. Kovyrshin, "Adaptive trailer backup control, " 2018 International

Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 2018, pp. 1-6.

7. S.P. Kruglov, S.V. Kovyrshin, "Adaptive trailer backup control, " 2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), Vladivostok, Russia, 2018, pp. 1-6.

8. Kruglov S.P., Kovyrshin S.V., Aksamentov D.N. Adaptivnoe upravlenie dvuhmayatnikovym podvesom mostovogo krana. Mekhatronika, avtomatizaciya, upravlenie, 2022, t. 23, № 9, s. 451-461.

9. Kruglov S.P., Kovyrshin S.V. Adaptivnoe upravlenie mostovym kranom pri perenose dlinnomernogo vertikal'no raspolozhennogo gruzha. Izvestiya vysshih uchebnyh zavedenij. Mashinostroenie, 2023, № 11, s. 34–44, doi: 10.18698/0536-1044-2023-11-34-44. EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Washington, USA: Electronic Industries Association, Engineering Department. 1969. — URL: http://bitsavers.trailing-edge.com/pdf/datapro/communications_standards/2740_EIA_RS-232-C.pdf (дата обращения: 03.09.2023).

10. EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange. Washington, USA: Electronic Industries Association, Engineering Department. 1969. — URL: http://bitsavers.trailing-edge.com/pdf/datapro/communications_standards/2740_EIA_RS-232-C.pdf (дата обращения: 03.09.2023).

11. Electronic Industries Association. Electrical Characteristics of Generators and Receivers for Use in Balanced Multipoint Systems. EIA Standard RS-485. — URL: http://www.softelectro.ru/rs485_en.html (дата обращения: 03.12.2023).

12. Janssen, D.; Buttner, H. Real-time Ethernet: the EtherCAT solution, Computing & Control Engineering Journal. 2004. Vol. 15. P. 16–21.

Информация об авторах

Ковыршин Сергей Владимирович - к.т.н., доцент кафедры «Автоматизация производственных процессов», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: sergkow@mail.ru

Буторин Денис Витальевич - к.т.н., доцент кафедры «Автоматизация производственных процессов», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: den_butorin@mail.ru

Попов Алексей Владимирович – студент, направление подготовки «Мехатроника и робототехника», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: popov.lxg18@mail.ru

Коденёв Кирилл Федорович – студент, направление подготовки «Мехатроника и робототехника», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: kirill_kodenev@mail.ru

Information about the authors

Sergej Vladimirovich Kovyrshin – Ph.D. in Engineering Science, Associate Professor, the Subdepartment of Automation of Production Processes, Irkutsk State Transport University, Irkutsk, e-mail: sergkow@mail.ru

Denis Vital'evich Butorin – Ph.D. in Engineering Science, Associate Professor, the Subdepartment of Automation of Production Processes, Irkutsk State Transport University, Irkutsk, e-mail: den_butorin@mail.ru

Aleksej Vladimirovich Popov – student, specialty “Mechatronics and Robotics”, Irkutsk State University of Transport, Irkutsk, e-mail: popov.lxg18@mail.ru

Kirill Fedorovich Kodenov – student, specialty “Mechatronics and Robotics”, Irkutsk State University of Transport, Irkutsk, e-mail: kirill_kodenev@mail.ru