

Реализация интерфейса RS-485 в микроконтроллерах

Олег Вальпа (Челябинская обл.)

В статье приведено краткое описание интерфейса RS-485, библиотечных функций среды разработки mikroC для поддержки данного интерфейса микроконтроллером и схемы его подключения к персональному компьютеру. Даны примеры программ на языке Си для организации связи между устройствами.

ВВЕДЕНИЕ

Интерфейс RS-485 предназначен для организации последовательного канала связи между несколькими устройствами. Он обеспечивает обмен дан-

ными по одной двухпроводной линии связи в полудуплексном режиме.

Благодаря своей простоте, низкой стоимости и надёжности, данный интерфейс давно и активно использу-

ется в АСУ ТП, системах контроля доступа и другом ответственном оборудовании. Даже в настоящее время интерфейс RS-485 является наиболее популярным промышленным стандартом связи.

КРАТКОЕ ОПИСАНИЕ ИНТЕРФЕЙСА

Интерфейс RS-485 обеспечивает передачу данных со скоростью до нескольких десятков мегабит в секунду. Дальность связи зависит от скорости передачи информации в обратно пропорциональной зависимости. Для обмена данными по интерфейсу RS-485 используется одна витая пара проводников с волновым сопротивлением 120 Ом, обеспечивающая передачу информации на расстояние до 1200 м. Протокол интерфейса позволяет поддерживать многоточечные соединения, обеспечивая создание сетей из 256 устройств.

Количество устройств, подключаемых к одной линии интерфейса, зависит от типа применённых в устройстве приёмопередатчиков. Один передатчик рассчитан на управление 32 стандартными приёмниками. Выпускаются приёмники с входным сопротивлением 1/2, 1/4, 1/8 от сопротивления стандартного приёмника. При использовании таких приёмников общее число устройств может быть увеличено соответственно до 64, 128 или 256. Использование повторителей RS-485 позволяет увеличить расстояние передачи информации и количество устройств в сети.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Среда разработки mikroC [1] предоставляет набор готовых функций для создания программ, обеспечивающих совместную работу устройств в сети связи RS-485 на базе микроконтроллеров (МК) различных семейств. Совместная работа устройств такой сети обеспечивается одним ведущим (master) и множеством ведомых (slave) устройств. Ведущее и ведомое устройства обмениваются пакетами информации, каждый из которых может содержать отдельные байты синхрониза-

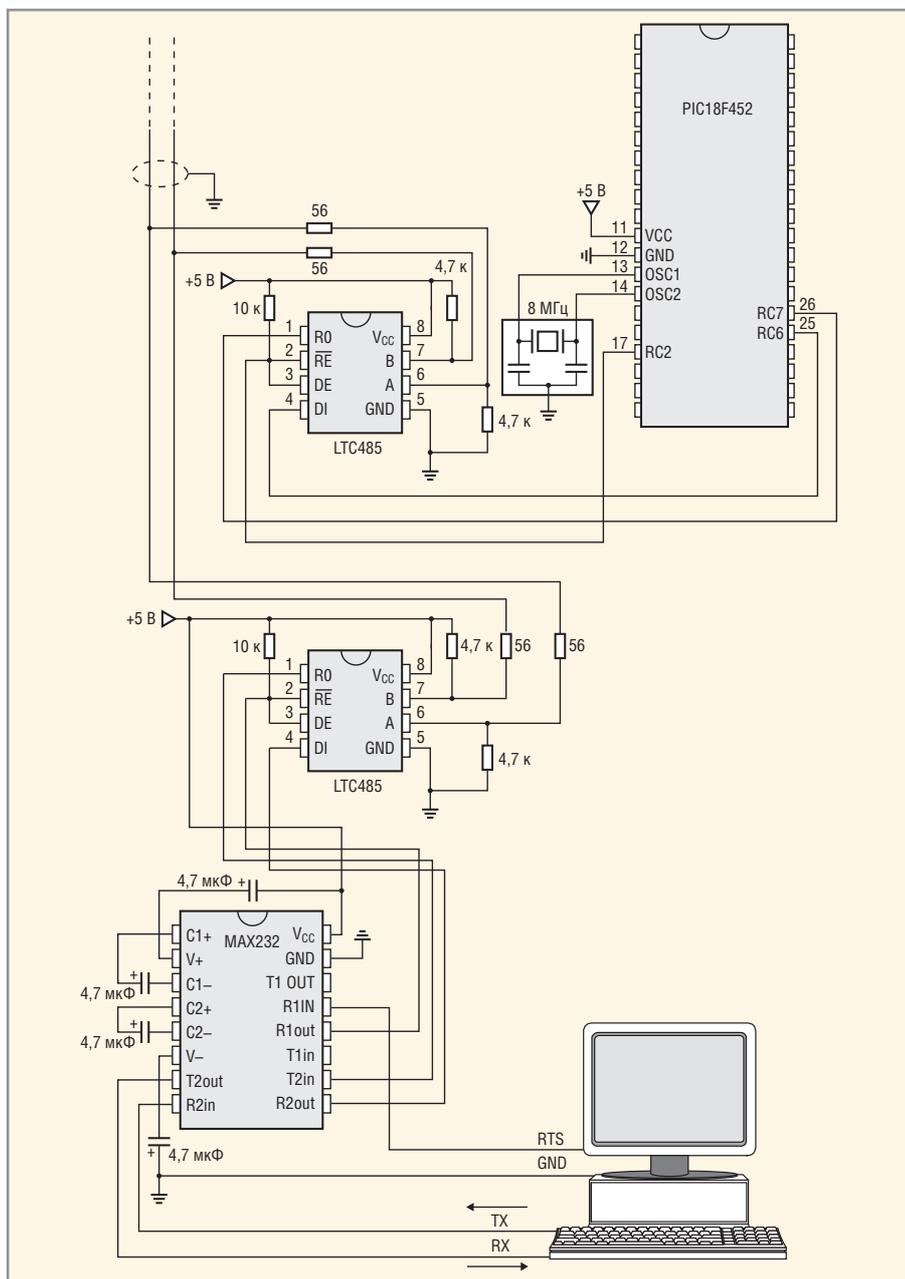


Схема подключения микроконтроллера PIC к персональному компьютеру по интерфейсу RS-485

Таблица 1. Описание функции *RS485Master_Init*

Прототип	<code>void Rs485Master_Init(unsigned short * port, unsigned short pin);</code>
Результат	Нет
Описание	Инициализирует PIC МК как ведущее устройство для обмена по интерфейсу RS-485
Требования	Аппаратура модуля USART должна быть заранее инициализирована с помощью функции <code>Usart_Init</code>
Пример	<code>RS485Master_Init(&PORTC, 2);</code>

ции, контрольной суммы, адреса и данных. Каждое ведомое устройство имеет свой уникальный адрес и принимает пакеты, направляемые по этому адресу. Кроме того, каждое ведомое устройство принимает пакеты с ширококестельным (общим для всех ведомых устройств) адресом. Ведомое устройство никогда не является инициатором обмена. Ответственность за то, чтобы только одно устройство вело передачу в конкретный момент времени по шине RS-485, ложится на программиста.

Рассмотрим пример реализации интерфейса RS-485 на базе МК семейства PIC компании microchip [2]. Функции обмена по интерфейсу RS-485 требуют наличия аппаратного модуля USART на порту PORTC микроконтроллера PIC.

Выводы USART должны быть подключены к микросхеме – драйверу приёмопередатчика RS-485, например, LTC485 или аналогичному. Для управления приёмопередатчиком с помощью сигналов *RE* (Receiver Output Enable) и *DE* (Driver Outputs Enable) используется второй бит порта PORTC (см. рисунок).

Адрес 50 является ширококестельным адресом, т.е. общим адресом для всех ведомых устройств. Пакеты с таким адресом будут приняты всеми ведомыми устройствами. Исключениями являются ведомые устройства с адресами 150 и 169.

Перед инициализацией интерфейса RS-485 должна быть вызвана функция *Usart_Init* для инициализации USART.

БИБЛИОТЕКА ДЛЯ РАБОТЫ С RS-485

Библиотечные функции RS-485 включают в себя следующие функции для программной поддержки интерфейса RS-485:

- *Master_Init*;
- *RS485Master_Receive*;
- *RS485Master_Send*;
- *RS485Slave_Init*;
- *RS485Slave_Receive*;
- *RS485Slave_Send*.

Подробное описание этих функций с требованиями и примером вызова приведено в таблицах 1 – 6 соответственно.

ПРИМЕР ПРОГРАММЫ

Рассмотрим две короткие программы на языке высокого уровня Си, де-

Таблица 2. Описание функции RS485Master_Receive

Прототип	void RS485Master_Receive(unsigned short *data);
Результат	Нет
Описание	Принимает любое сообщение, отправленное ведомым устройством. Сообщение состоит из нескольких байтов, поэтому данная функция должна вызываться для приёма каждого байта (см. примеры программ). По окончании приёма сообщения буфер заполнен следующими значениями: data[0..2] собственно сообщение data[3] количество байтов в принятом сообщении, от 1 до 3 data[4] установлено значение 255, когда всё сообщение принято data[5] установлено значение 255, если при приёме обнаружена ошибка data[6] адрес ведомого устройства, которое отправило данное сообщение Функция автоматически устанавливает data[4] и data[5] после приёма каждого сообщения. Эти флаги должны очищаться программно
Требования	МК должен быть проинициализирован как ведущее устройство для обмена по интерфейсу RS-485, чтобы назначать адрес. См. функцию RS485Master_Init
Пример	unsigned short msg[8]; ... RS485Master_Receive(msg);

Таблица 3. Описание функции RS485Master_Send

Прототип	void RS485Master_Send(unsigned short *data, unsigned short datalen, unsigned short address);
Результат	Нет
Описание	Посылка сообщения data из буфера ведомому устройству (устройствам) с заданным адресом по интерфейсу RS-485; datalen – количество байтов в сообщении (от 1 до 3)
Требования	МК должен быть проинициализирован как Master для обмена по интерфейсу RS-485, чтобы назначать адрес. См. функцию RS485Master_Init. Ответственность за то, чтобы только одно устройство вело передачу в конкретный момент времени по шине RS-485, ложится на программиста
Пример	unsigned short msg[8]; ... RS485Master_Send(msg, 3, 0x12);

Таблица 4. Описание функции RS485Slave_Init

Прототип	void Rs485slave_Init(unsigned short *port, unsigned short pin, char address);
Результат	Нет
Описание	Инициализирует микроконтроллер PIC как ведомое устройство с заданным адресом (address) для обмена по интерфейсу RS-485. Адрес ведомого устройства может принимать любые значения от 0 до 255, исключая 50, который является общим для всех ведомых устройств
Требования	Аппаратура модуля USART должна быть заранее инициализирована. См. функцию Usart_Init [3]
Пример	Инициализация МК как ведомого устройства с адресом 160: RS485Slave_Init(PORTC, 2, 160);

Таблица 5. Описание функции RS485Slave_Receive

Прототип	void RS485Slave_Receive(unsigned short *data);
Результат	Нет
Описание	Принимает сообщение, адресованное ведомому устройству. Сообщение состоит из нескольких байтов, поэтому данная функция должна вызываться для приёма каждого байта (см. примеры программ). По окончании приёма сообщения буфер заполнен следующими значениями: data[0..2] собственно сообщение data[3] количество байтов в принятом сообщении, от 1 до 3 data[4] установлено значение 255, когда всё сообщение принято data[5] установлено значение 255, если при приёме обнаружена ошибка data[6] адрес ведомого устройства, которое отправило данное сообщение Функция автоматически устанавливает data[4] и data[5] после приёма каждого сообщения. Эти флаги должны очищаться программно
Требования	МК должен быть проинициализирован как ведомое устройство для обмена по интерфейсу RS-485. См. функцию RS485Slave_Init
Пример	unsigned short msg[8]; ... RS485Slave_Read(msg);

Таблица 6. Описание функции RS485Slave_Send

Прототип	void RS485Slave_Send(unsigned short *data, unsigned short datalen);
Результат	Нет
Описание	Посылка сообщения (data) из буфера ведущему устройству по интерфейсу RS-485; datalen – количество байтов в сообщении (от 1 до 3)
Требования	МК должен быть проинициализирован как ведомое устройство для обмена по интерфейсу RS-485, чтобы ему был назначен адрес. См. функцию RS485Slave_Init. Ответственность за то, чтобы только одно устройство вело передачу в конкретный момент времени по шине RS-485, ложится на программиста
Пример	unsigned short msg[8]; ... RS485Slave_Send(msg, 2);

монстрирующие применение описанных функций на практике.

Первая программа, приведённая в листинге 1 (см. дополнительные материалы к статье на сайте журнала), демонстрирует работу микроконтроллера PIC, подключенного к персональному компьютеру (ПК) в качестве ведомого устройства при обмене данными по интерфейсу RS-485. Схема подключения микроконтроллера PIC к ПК по интерфейсу RS-485 приведена на рисунке.

Микроконтроллер PIC с помощью данной программы принимает данные, направляемые только ему, по адресу 160, или адресованные всем устройствам – по адресу 50. Принятые данные отображаются в порт PORTB и транслируются обратно ведущему устройству.

В приведённой программе организован байтовый буфер сообщений *dat*, с помощью которого производится обмен информацией между устройствами и осуществляется анализ принятой информации. Внимательно прочитайте комментарии к строкам программы, чтобы понять все действия, производимые в ней.

Следует обратить внимание на то, что скорости обмена данными ведущего и всех ведомых устройств по интерфейсу RS-485 должны совпадать. В противном случае устройства не смогут связаться друг с другом. В программе скорость обмена 9600 бод задаётся с помощью функции инициализации модуля USART (строка: *Usart_Init(9600)*).

Вторая программа, приведённая в листинге 2 (см. дополнительные материалы к статье на сайте журнала), демонстрирует работу микроконтроллера PIC в качестве ведущего устройства при обмене данными по интерфейсу RS-485. Такой МК может быть включен в схему, изображённую на рисунке, вместо ПК, чтобы выполнять функции организатора сети.

В этой программе МК организует связь с ведомым устройством, посылая ему данные по адресу 160. Ведомое устройство должно принять данные и вернуть обратно ведущему устройству. Полученные от ведомого устройства данные отображаются в порт PORTB, а ошибка приёма (0xAA) и количество последовательных неудачных повторных попыток записы-

ваются в порт PORTD. После десятой неудачной попытки получить данные от ведомого устройства, программа посылает данные по широковещательному адресу и обнуляет счётчик ошибок.

ЗАКЛЮЧЕНИЕ

Модифицируя описанные выше программы, можно изменять адрес устройства и алгоритм работы программы для решения конкретной задачи. Аналогично рассмотренным здесь вариантам использования МК семейства PIC, можно применять для работы с интерфейсом RS-485 и микроконтроллеры других семейств. В частности, существует среда разработки *mikroC for 8051* для поддержки МК семейства 8051 и *mikroC for AVR* для МК семейства AVR [3].

ЛИТЕРАТУРА

1. www.mikroe.com.
2. www.microchip.com.
3. *Вальна О.* Современная среда разработки mikroC для программирования микроконтроллеров на языке высокого уровня Си. Современная электроника. 2010. № 6. С. 64.

