

АНАЛИЗ И СРАВНЕНИЕ ИНТЕРФЕЙСОВ ПЕРЕДАЧИ ДАННЫХ НА МИКРОКОНТРОЛЛЕРАХ

Г.Т. Катукия

giorgi.katukiya@mail.ru

SPIN-код: 9618-0360

А.В. Дорофеев

alexanderzloizebroo@gmail.com

SPIN-код: 4995-1118

Д.Э. Линтварев

lintvarevdenis@gmail.com

SPIN-код: 5322-0792

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Аннотация

Разработан методический материал, содержащий основную информацию о самых популярных протоколах передачи данных, используемых при работе с микроконтроллерами. Подробно представлены такие интерфейсы, как UART/USART, SPI, I²C. Рассмотрены основные характеристики, принцип работы и схема подключения. Составлена таблица сравнения характеристик вышеперечисленных протоколов обмена информацией, включающая в себя рекомендательные комментарии по области применения каждого из них. Статья позволяет в кратчайшие сроки подобрать оптимальный в заданных условиях интерфейс для передачи данных, разобраться в его принципе работы и получить минимально необходимые сведения для его реализации.

Ключевые слова

Интерфейсы передачи данных, микроконтроллер, ведущее/ведомое устройство, пакет данных, UART/USART, SPI, I²C, встраиваемые системы

Поступила в редакцию 30.04.2020

© МГТУ им. Н.Э. Баумана, 2020

Введение. В век информационных технологий передача данных является неотъемлемой частью любой деятельности человека, особенно при работе с электронными устройствами. С развитием встраиваемых систем и IoT (internet of things — интернет вещей) устройства на базе микроконтроллеров (МК) становятся все более востребованными [1–3]. Одним из основных вопросов при разработке таких устройств является выбор протокола передачи данных между МК и внешними устройствами. От выбора наиболее предпочтительного в поставленных условиях протокола передачи данных будет зависеть стабильность, быстродействие и удобство разработки устройства на базе МК [4–19]. В настоящей статье подробно рассмотрим основные протоколы передачи данных и на основе приведенного материала дадим рекомендации по использованию каждого из протоколов в конкретных условиях.

Анализ протокола UART/USART. Основные характеристики и особенности протокола UART/USART. Universal Asynchronous Receiver-Transmitter (UART) — универсальный асинхронный приемо-передатчик, способ асинхрон-

ной полудуплексной (поочередной) передачи данных, который используется для обмена данным между одним приемником и одним передатчиком [10–12].

UART как таковой не имеет стандартов, поскольку во времена его создания не было единой системы стандартизации. Наиболее близкими к UART стандартизованными интерфейсами являются токовая петля и RS-232. Интерфейс RS-232 впервые был представлен в 1960 г. Ассоциацией электронной промышленности (EIA) в качестве рекомендуемого стандарта для использования в телекоммуникационной отрасли.

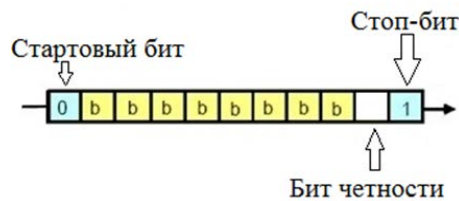


Рис. 1. Структура данных в UART

На устройствах заранее устанавливаются количество передаваемых бит за единицу времени, наличие/отсутствие бита четности, длину стоп-бита, которая может составлять 1 или 2 бита. При передаче больших файлов рекомендуют использовать стоп-бит длиной 2 бита. Второй стоп-бит воспринимается приемником как небольшой перерыв на линии. Стоп-бит необходим для того, чтобы приемник понимал, что передача сведений закончена. По умолчанию «тишиной» на линии является логическая единица. Это сделано для того, чтобы избежать помех. Перед тем как начать передавать полезную информацию, отправляется стартовый бит, значение которого соответствует логическому нулю. Приемник отсчитывает от стартового бита время, равное половине промежутка трансляции бита. Если на входе по окончании этого промежутка времени находится ноль, приемник начинает принимать информацию. Приемник отсчитывает определенное количество таких промежутков (это количество зависит от количества передаваемых битов) и в каждый момент фиксирует напряжение на входе. Последний принимаемый бит должен содержать логическую единицу — стоп-бит (рис. 1). При его отсутствии UART фиксирует ошибку [10].

Поскольку в ходе обмена информацией UART в обязательном порядке использует два дополнительных бита (стартовый бит и как минимум один стоп-бит), фактическая скорость передачи будет несколько ниже. В основном этот интерфейс позволяет передавать данные на определенных скоростях: 9 600, 19 200, 38 400, 57 600, 115 200 бит/с. Другие скорости протоколом не поддерживаются.

Для контроля передачи сведений и ее целостности используют бит четности. Предпоследний бит перед стоп-битом контролируется логикой интерфейса и хранит сведения о четности количества единичных бит в одном блоке полезных сведений. Существует два вида такого контроля — на четность и нечетность. В первом случае сумма количества логических единиц в посылке — число четное, во втором — нечетное [11].

Для передачи данных необходимо использовать две линии: DL — dataline (служит для передачи сигналов) и GND — Ground (необходима для корректного

распознавания сигналов) [10]. Существует также интерфейс USART (Universal Synchronous / Asynchronous Receiver-Transmitter) — универсальный синхронный/асинхронный приемо-передатчик [12]. Основными его особенностями можно считать наличие дополнительной линии синхронизации; отсутствие необходимости сообщать приемнику скорость передачи данных; возможность использовать как UART; несколько выше скорость обмена информацией по сравнению с UART.

Если в UART строго необходимо наличие стартового бита и стоп-бита, то в USART для этого предусмотрена отдельная линия, по которой и происходит синхронизация (рис. 2).

Благодаря синхронизации по отдельной линии скорость передачи может достигать 4,5 мбит/с.

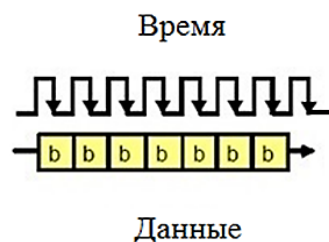


Рис. 2. Структура данных в USART

Сравнительный анализ преимуществ и недостатков протокола UART/USART. Основными преимуществами UART/USART над другими интерфейсами являются:

- 1) простота реализации;
- 2) наличие всего лишь двух проводов для создания соединения;
- 3) хорошая помехозащищенность;
- 4) возможность передавать данные на относительно большие расстояния (порядка 1 км);
- 5) встроенный в протокол способ обнаружения ошибок передачи;
- 6) возможность управления потоком данных приемником в некоторых версиях USART;

К недостаткам можно отнести следующее:

- 1) отсутствие возможности использования нескольких принимающих/передающих устройств;
- 2) довольно низкую скорость обмена информацией;
- 3) необходимость стабильного питания.

Области применения протокола UART/USART. Использование UART находит в основном для управления блоками питания, прошивки микросхем, программирование микроконтроллеров, в коммуникационных устройствах, модемах, для ввода AT-команд и др. AT-команды — набор команд, разработанных в 1977 г. компанией “Hayes” для модема Smartmodem 300 baud. Набор команд состоит из коротких текстовых строк, которые объединяют вместе для формирования полной команды, например набор номера, начало соединения или изменения параметров подключения [13]. Протокол UART также используют в работе COM-порта, который в наши дни в своем физическом представлении встречается довольно редко. В повседневных задачах его практически полностью заменил USB-интерфейс, но иногда нужно использовать более простой и удобный протокол, тогда обращаются к виртуальным COM-портам.

Анализ протокола протокола SPI. Основные характеристики протокола SPI. Serial Peripheral Interface (*SPI, SPIbus*) — последовательный периферийный интерфейс на основе синхронной полнодуплексной передачи данных (рис. 3) [14–17].

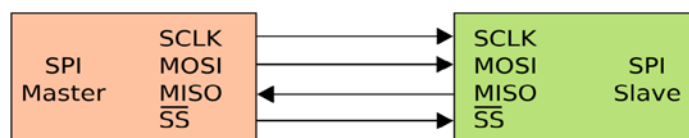


Рис. 3. Интерфейс SPI

Интерфейс разработан компанией “Motorola” в 1980-х годах [14], а теперь многие производители электроники добавляют в свои устройства поддержку SPI. Существует множество реализаций протокола из-за отсутствия общепринятого официального стандарта.

Анализ особенностей интерфейса SPI. В интерфейсе SPI используется всего четыре цифровые линии (см. рис. 3) [15]:

- *SCK/SCLK* — SerialClock, последовательный тактовый сигнал (линия синхронизации);
- *MOSI* — MasterOutSlaveIn, выход ведущего, вход ведомого (линия приема/передачи данных);
- *MISO* — MasterInSlaveOut, вход ведущего, выход ведомого (линия приема/передачи данных);
- $\overline{SS} / \overline{CS}$ — SlaveSelect/ChipSelect, выбор ведомого или выбор микросхемы.

В зависимости от устройства и фирмы изготовителя возможны альтернативные обозначения сигналов, в этой статье приведены лишь самые популярные варианты.

Передача данных происходит отдельными пакетами длиной обычно 1 байт. Процесс передачи одного пакета данных в режиме Mode 0 показан на рис. 4 [16].

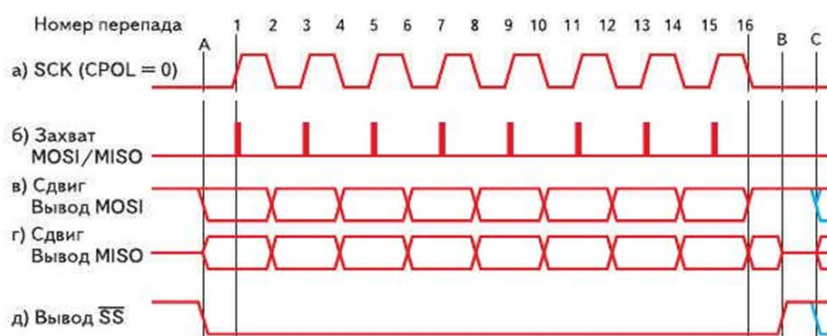


Рис. 4. Процесс передачи пакета данных

До начала передачи пакета сигнал \overline{SS} на выходе ведущего в высоком состоянии, ведомый неактивен. Чтобы активировать ведомое устройство, ведущий

устанавливает на линии \overline{SS} низкий уровень, делая возможным считывание тактовых импульсов ведомым. Затем Master начинает формирование синхроимпульсов на линии SCK , вызывая сдвиг содержимого сдвиговых регистров ведущего и ведомого, а также захват новых значений битов. Данные операции, т. е. установка и чтение данных, происходят по противоположным перепадам (фронтам) тактового сигнала. В рассматриваемом режиме работы SPI захват осуществляется по переднему фронту синхроимпульса, а сдвиг в регистрах — по заднему фронту. Передача пакета данных завершается установкой ведущим линии \overline{SS} в высокое состояние, после чего возможна передача следующего пакета.

Существует четыре режима работы интерфейса SPI , которые различаются уровнем сигнала синхронизации SCK при отсутствии передачи данных и последовательностью выполнения установки и выборки данных. Первый параметр называется полярностью тактовых импульсов и устанавливается битом $CPOL$, второй управляется битом $CPHA$ и называется фазой. При $CPOL = 0$ на линии SCK в неактивном состоянии низкий уровень, а при $CPOL = 1$ — высокий. Если $CPHA = 0$, то по первому фронту SCK происходит считывание данных, при $CPHA = 1$ происходит сдвиг данных. Ниже приведены принятые обозначения режимов работы протокола SPI : Mode 0 ($CPOL = 0, CPHA = 0$), Mode 1 ($CPOL = 0, CPHA = 1$), Mode 2 ($CPOL = 1, CPHA = 0$), Mode 3 ($CPOL = 1, CPHA = 1$).

Отметим, что ведомые устройства обычно работают только в одном режиме, поэтому перед началом обмена данными ведущий должен быть настроен на этот режим.

Группа ведомых устройств может соединяться с ведущим двумя способами (рис. 5): с помощью радиальной или кольцевой структуры связи.

Первый способ (рис. 5, а) основан на независимом подключении ведомых устройств и считается наиболее популярным. В данном случае линии $MISO$, $MOSI$ и SCK ведомых устройств соединены параллельно, но для активации каждого ведомого требуется отдельная линия \overline{SS} .

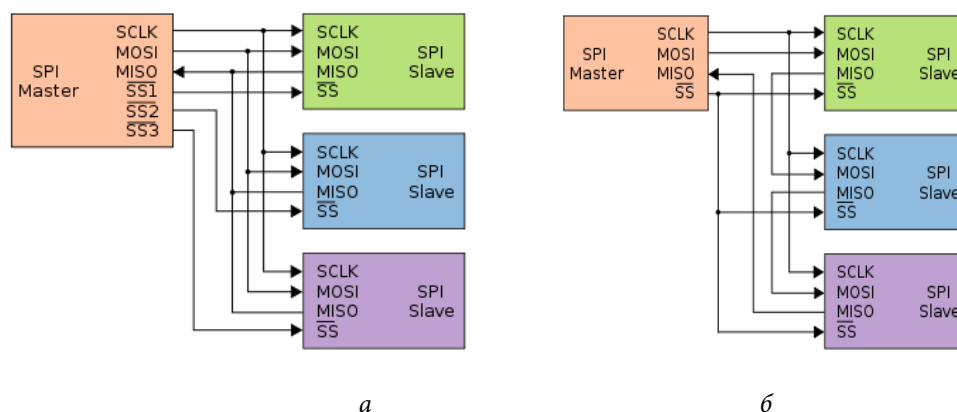


Рис. 5. Способы построения связей по протоколу SPI

Кольцевая структура связи (каскадное подключение, Daisychain) (рис. 5, б) позволяет обойтись лишь одной линией \overline{SS} , при этом линии синхронизации SSC соединяются параллельно, а линии данных — последовательно, образуя замкнутую цепь (кольцо). Пакет данных последовательно проходит через всех ведомых до того устройства, которому он и предназначался. Отметим, что не все периферийные устройства поддерживают кольцевую структуру связи.

Сравнительный анализ преимуществ и недостатков протокола SPI. К преимуществам протокола следует отнести:

- простоту аппаратной и программной реализации;
- относительно высокую скорость передачи (обычно 20 Мб/с, максимум до 100 Мб/с), ограниченная только быстродействием связанных устройств [11];
- полнодуплексную передачу данных по умолчанию;
- возможность произвольного выбора длины пакета;
- возможность использования в системах с низкостабильной тактовой частотой;
- однонаправленный характер сигналов позволяет при необходимости легко организовать гальваническую развязку и преобразование уровней между ведущим и ведомыми устройствами.

Недостатками протокола являются:

- отсутствие официального стандарта и, вследствие этого, наличие множества вариантов реализации интерфейса;
- отсутствие определенного стандартом метода обнаружения ошибок и подтверждения передачи данных [17];
- относительно большое количество сигнальных линий (от трех и больше);
- тот факт, что ведомое устройство не может контролировать передачу данных;
- отсутствие поддержки динамического подключения устройств;
- небольшая дальность передачи данных (менее 1 м, обычно 20 см) [11].

Области применения протокола SPI. Подключение МК (или ПЛИС) к периферийным устройствам: микросхемы памяти, аналого-цифровые и цифро-аналоговые преобразователи, дисплеи, датчики, микросхемы времени, цифровые потенциометры и т. д. Для МК AVR протокол SPI используется для внутрисхемного программирования (ISP).

Чаще всего SPI применяется для связи устройств на одной печатной плате за счет небольшой дальности передачи данных.

Анализ протокола I²C. Основные характеристики протокола I²C. Протокол I²C (Inter-Integrated Circuit — межинтегральная схема) — двухсторонняя полудуплексная (последовательно прием и передача данных) шина связи.

Интерфейс был разработан в 1980-х годах компанией “Philips”. Де-факто является мировым стандартом и применяется более чем в 1000 микросхем по всему миру. Помимо этого шина I²C используется в следующих архитектурах управления [18]:

- System Management Bus (SMBus);
- Power Management Bus (PMBus);

- Intelligent Platform Management Interface (IPMI);
- Display Data Channel (DDC).

Протокол I²C состоит из двух каналов передачи данных: SDA (serial data line) и SCL (serial clock line). Первый используется для передачи данных, в то время как второй отвечает за синхронизацию.

Передача данных начинается с инициализации начального условия (start) ведущим устройством. Для этого уровень сигнала на линии SDA опускается от единицы к нулю при высоком уровне SCL. Конечное условие (stop) задается переходом от нуля к единице на линии SDA при логической единице на линии SCL. Возможен сценарий непрерывной передачи данных, когда вместо остановки (stop) ведущее устройство начинает очередную передачу данных. Такая ситуация называется повторным стартом (repeated start). Передача данных по линии SDA может производиться только в том случае, когда сигнал на линии SCL соответствует нулю. В конце сигнал на линии SCL должен подняться к единичному уровню в знак подтверждения бита. Логика передачи данных наглядно продемонстрирована на рис. 7 [18].

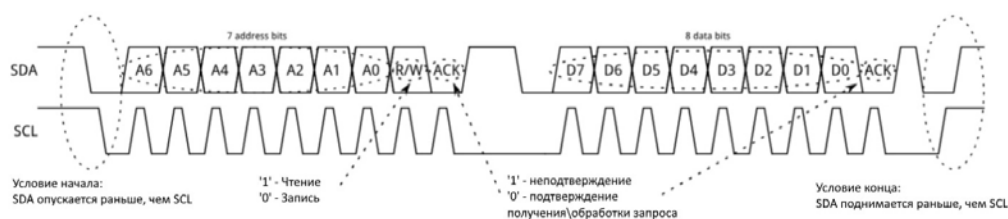


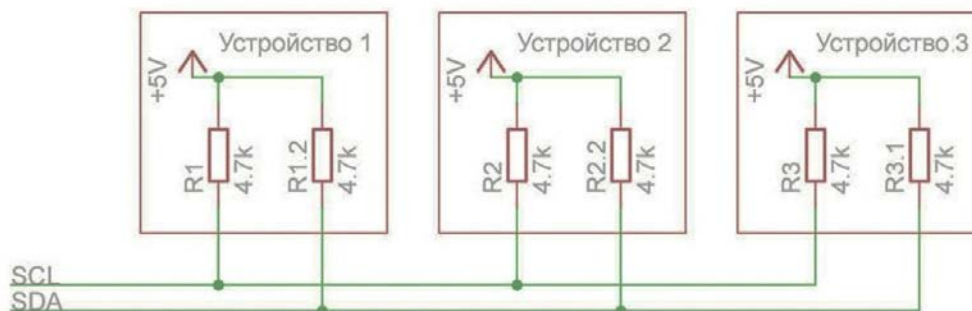
Рис. 7. Логика передачи данных по протоколу I²C:

Выделяют следующие режимы работы:

- стандартный режим (StandardMode) — 100 кбит/с (100 кГц), с возможностью замедления до 0;
- быстрый режим (Fastmode) == 400 кбит/с (400 кГц);
- высокоскоростной режим (High-speedmode) — 3,4 Мбит/с (3,4 МГц);
- суперскоростной режим (UltraFast-mode) — однонаправленный 5 МГц без подтягивающих резисторов [18].

Соединение устройств осуществляется в режиме открытого коллектора. Управление в шине I²C происходит путем понижения сигнала с высокого уровня на низкий. Высокий уровень — логическая единица, а низкий — логический ноль. Поскольку поднимать сигнал управляющее устройство самостоятельно не может, нужно выставить высокий уровень вручную с помощью подтягивающих резисторов. Чем ниже сопротивление, тем выше скорость работы цепи, но больше энергопотребление. Чтобы не перегружать цепь сопротивлением, его следует ставить только на устройстве, подключенном к питанию (рис. 8) [19].

Шина по протоколу I²C использует семи- и десятибитную адресацию, что позволяет размещать на ней до 1024 различных устройств [18].

Рис. 8. Схема подключения устройств по протоколу I²C

Сравнительный анализ преимуществ и недостатков протокола I²C.

Протокол обладает следующими преимуществами:

- используется только два провода для подключения;
- можно на ходу добавлять и убирать устройства из цепи;
- минимальные помехи благодаря встроенному фильтру;
- можно подключить больше 1000 устройств благодаря десятибитной адресации;
- может быть более одного управляющего устройства;
- осуществляется двусторонний обмен данными.

К недостаткам протокола относятся следующие факты:

- емкость линии не превышает 400 пФ;
- емкость шины состоит из общей емкости проводов, подключенных портов и контактов. Чем меньше емкость, тем выше скорость работы цепи;
- поскольку применяется полудуплексный метод передачи данных, скорость цепи будет равна скорости самого медленного устройства;
- при наличии более одного управляющего устройства, возможно возникновение ошибок;
- сложно отыскать ошибку, поскольку неправильно работающее устройство делает неработоспособной сразу всю линию;
- для работы необходимы дополнительные резисторы.

Области применения протокола I²C. Протокол I²C применяют, когда нужно передать данные между устройствами максимально простым и дешевым способом:

- для регулировки параметров цветопередачи мониторов;
- регулировки звука в динамиках;
- получения данных от различных датчиков;
- обмена информации между управляющими устройствами.

Сравнительный анализ интерфейсов передачи данных. Выполним сравнение рассматриваемых протоколов передачи данных с помощью таблицы, в которой указаны основные особенности каждого из интерфейсов для упрощения выбора.

Результаты сравнительного анализа базовых протоколов

Параметр	UART/USART	SPI	I ² C
Скорость	Минимальная скорость 300 бит/с. Сейчас используется крайне редко. Максимальная скорость 115 200 бит/с	Обычно 20 Мб/с. При желании можно получить 100 Мб/с	В стандартном режиме 100 кбит/с. При желании можно получить скорость до 3,4 Мбит/с
Надежность	Присутствует защита от помех. Возможна проверка целостности передачи информации	Малая помехозащищенность; возможность работы с нестабильной тактовой частотой; отсутствует проверка ошибок	Присутствует защита от помех; отсутствует протокол обнаружения ошибок
Удобство	Легко подключать	Неудобно подключать из-за большого количества линий	Легко подключать, но сложно настраивать
Дальность	До 1 км	Менее 1 м, обычно 20 см	Зависит от емкости цепи (до 50 м)
Количество подключаемых устройств	1 ведущее — 1 ведомое	1 ведущее — произвольное число ведомых. На каждое новое ведомое устройство необходима линия выбора SS	Произвольное число ведущих и ведомых. Можно подключать больше 1000 устройств, используя лишь два провода

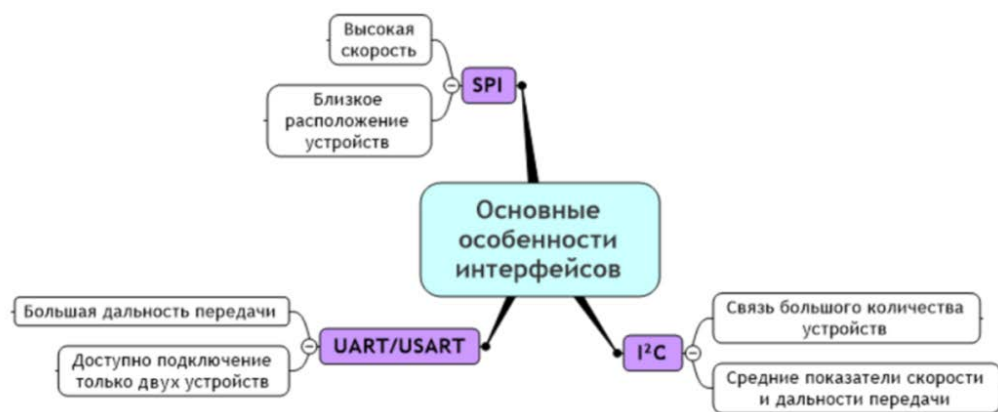


Рис. 9. Классификация базовых протоколов по функциональным особенностям

По результатам анализа можно сделать следующие выводы и рекомендации:

- протокол UART/USART будет наиболее эффективным для соединения двух устройств, удаленных друг от друга на достаточно большое расстояние [20];
- протокол SPI оптимально использовать для взаимосвязи на относительно высоких скоростях небольшого количества устройств, расположенных вблизи друг к другу, в основном микросхемы на одной печатной плате [21];

– протокол I²C станет наиболее подходящим в задачах связывания большого количества устройств [22].

Заключение. В работе рассмотрены основные интерфейсы передачи данных SPI, UART/USART, I²C, используемые в устройствах на базе микроконтроллеров, приведены их основные параметры и области применения. Данная работа ориентирована на помощь разработчикам электронной аппаратуры в выборе протокола передачи данных, наиболее подходящего под поставленные задачи.

Литература

- [1] Санталов Г.Д., Муравьев К.А., Леонидов В.В. Методика проектирования электронных устройств в концепции «Интернета вещей». *Проектирование и технология электронных средств*, 2019, № 1, с. 10–22.
- [2] Vlasov A.I., Yudin A.V., Shakhnov V.A., et al. Design methods of teaching the development of internet of things components with considering predictive maintenance on the basis of mechatronic devices. *Int. J. Appl. Eng. Res.*, 2017, vol. 12, no. 20, pp. 9390–9396.
- [3] Vlasov A.I., Muraviev K.A., Prudius A.A., et al. Load balancing in big data processing systems. *IREACO*, 2019, vol. 12, no. 1, pp. 42–47. DOI: <https://doi.org/10.15866/ireaco.v12i1.16808>
- [4] Муравьев К.А., Цивинская Т.А., Манушян Д.Г. Анализ протоколов передачи данных самоорганизующихся сенсорных сетей. *Тр. межд. симп. Надежность и качество*, 2019, т. 2, с. 142–145.
- [5] Адамов А.П., Адамова А.А., Семенцов С.Г. и др. Протоколы обмена в беспроводных сенсорных сетях регистрации событий. *Надежность и качество сложных систем*, 2018, № 3(23), с. 124–132. DOI: <https://doi.org/10.21685/2307-4205-2018-3-15>
- [6] Власов Д.С., Коновалов Д.С., Муравьев К.А. Исследование качественных показателей беспроводных AD-НОС сетей. *Тр. межд. симп. Надежность и качество*, 2019, т. 1, с. 155–158.
- [7] Адамова А.А., Аваева Л.Г., Лохтуров А.В. Анализ перспектив интеграции беспроводных сенсорных сетей с сетью интернет с использованием стандарта 6LOWPAN. *Надежность и качество сложных систем*, 2019, № 1(25), с. 54–61. DOI: <https://doi.org/10.21685/2307-4205-2019-1-6>
- [8] Зимин Д.В., Муравьев К.А. Анализ проблем энергоэффективности беспроводных сетей передачи данных на базе стека протоколов ZIGBEE. *Тр. межд. симп. Надежность и качество*, 2016, т. 1, с. 195–197.
- [9] Краснобрыжий Б.В., Лавров И.В., Муравьев К.А. и др. Анализ беспроводных сенсорных сетей на основе стандарта ZIGBEE. *Наукоемкие технологии и интеллектуальные системы. XV Молодеж. науч.-тех. конф.*, 2013, с. 306–314.
- [10] Universal asynchronous receiver/transmitter (UART). *ti.com: веб-сайт*. URL: <http://www.ti.com/lit/ug/sprugp1/sprugp1.pdf> (дата обращения: 08.12.2019).
- [11] Frenzel L.E. Jr. Handbook of serial communications interfaces. Elsevier, 2015.
- [12] USART. *ww1.microchip.com: веб-сайт*. URL: <http://ww1.microchip.com/downloads/en/devicedoc/usart.pdf> (дата обращения: 08.12.2019).
- [13] AT command set. *mt-system.ru: веб-сайт*. URL: http://www.mt-system.ru/sites/default/files/documents/simcom_sim5360_atc_en_v0.2.2.pdf (дата обращения: 08.12.2019).

- [14] SPI (serial peripheral interface). *ru.bmstu.wiki: веб-сайт*. URL: [https://ru.bmstu.wiki/index.php?title=SPI_\(Serial_Peripheral_Interface\)&mobileaction=toggle_view_mobile](https://ru.bmstu.wiki/index.php?title=SPI_(Serial_Peripheral_Interface)&mobileaction=toggle_view_mobile) (дата обращения: 08.12.2019).
- [15] SPI block guide V03.06. *web.archive.org: веб-сайт*. URL: <https://web.archive.org/web/20150413003534/http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf> (дата обращения: 08.12.2019).
- [16] Новицкий А. Синхронный последовательный интерфейс SPI в микроконтроллерах «от А до Я» и его реализация на примере ADuC70xx фирмы AnalogDevices. *Компоненты и технологии*, 2009, № 3, с. 53–60.
- [17] SPI. *webhamster.ru: веб-сайт*. URL: <https://webhamster.ru/mytetrashare/index/mtb186/1517214119k1qn5rp0h8> (дата обращения: 08.12.2019).
- [18] UM10204 I2C. *nxp.com: веб-сайт*. URL: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf> (дата обращения: 08.12.2019).
- [19] Using the I2C bus. *robot-electronics.co.uk: веб-сайт*. URL: <https://www.robot-electronics.co.uk/i2c-tutorial/> (дата обращения: 08.12.2019).
- [20] Петров А.А. Протокол передачи данных для UART. *Достижения науки и образования*, 2016, № 1(2), с. 9–10.
- [21] Муравьев К.А., Хуббатулин М.Э. Методика настройки VPN на роутерах MIKROTIK. *Технологии инженерных и информационных систем*, 2019, № 2, с. 22–33.
- [22] Семенов Б.Ю. Шина I2C в радиотехнических конструкциях. М., Солон-Пресс, 2004.

Катукия Гиорги Торникевич — студент кафедры «Проектирование и технология производства электронной аппаратуры», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Дорофеев Александр Вадимович — студент кафедры «Проектирование и технология производства электронной аппаратуры», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Линтварев Денис Эминович — студент кафедры «Проектирование и технология производства электронной аппаратуры», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Глушко Андрей Александрович, кандидат технических наук, доцент кафедры «Проектирование и технология производства электронной аппаратуры», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Катукия Г.Т., Дорофеев А.В., Линтварев Д.Э. Анализ и сравнение интерфейсов передачи данных на микроконтроллерах. *Политехнический молодежный журнал*, 2020, № 07(48). <http://dx.doi.org/10.18698/2541-8009-2020-07-624>

ANALYSIS AND COMPARISON OF DATA TRANSMISSION INTERFACES ON MICROCONTROLLERS

G.T. Katukiya

giorgi.katukiya@mail.ru

SPIN-code: 9618-0360

A.V. Dorofeev

alexanderzloizebroo@gmail.com

SPIN-code: 4995-1118

D.E. Lintvarev

lintvarevdenis@gmail.com

SPIN-code: 5322-0792

Bauman Moscow State Technical University, Moscow, Russian Federation

Abstract

A methodological material has been developed containing basic information about the most popular data transfer protocols used when working with microcontrollers. Such interfaces as UART/USART, SPI, I²C are presented. The main characteristics, principle of operation and connection diagram are considered. A table has been compiled for comparing the characteristics of the above information exchange protocols, including recommendatory comments on the scope of each of them. The article allows to select the interface for data transmission that is optimal under the given conditions, to understand its principle of operation and obtain the minimum necessary information for its implementation.

Keywords

Data interfaces, microcontroller, master / slave, data packet, UART/USART, SPI, I²C, embedded systems

Received 30.04.2020

© Bauman Moscow State Technical University, 2020

References

- [1] Santalov G.D., Murav'yev K.A., Leonidov V.V. Design method for electronic devices using "Internet of things" conception. *Proektirovanie i tekhnologiya elektronnykh sredstv*, 2019, no. 1, pp. 10–22 (in Russ.).
- [2] Vlasov A.I., Yudin A.V., Shakhnov V.A., et al. Design methods of teaching the development of internet of things components with considering predictive maintenance on the basis of mechatronic devices. *Int. J. Appl. Eng. Res.*, 2017, vol. 12, no. 20, pp. 9390–9396.
- [3] Vlasov A.I., Muraviev K.A., Prudius A.A., et al. Load balancing in big data processing systems. *IREACO*, 2019, vol. 12, no. 1, pp. 42–47. DOI: <https://doi.org/10.15866/ireaco.v12i1.16808>
- [4] Murav'yev K.A., Tsivinskaya T.A., Manushyan D.G. [Analysis of data transmission protocols for self-organizing networks]. *Tr. mezhd. simp. Nadezhnost' i kachestvo* [Proc. Int. Symp. Reliability and Quality], 2019, vol. 2, pp. 142–145 (in Russ.).
- [5] Adamov A.P., Adamova A.A., Sementsov S.G., et al. Communication protocols in wireless sensor networks of event registration. *Nadezhnost' i kachestvo slozhnykh sistem* [Reliability & Quality of Complex Systems], 2018, no. 3(23), pp. 124–132. DOI: <https://doi.org/10.21685/2307-4205-2018-3-15> (in Russ.).
- [6] Vlasov D.S., Kononov D.S., Murav'yev K.A. [Study on quality merits of wireless AD-HOC networks]. *Tr. mezhd. simp. Nadezhnost' i kachestvo* [Proc. Int. Symp. Reliability and Quality], 2019, vol. 1, pp. 155–158 (in Russ.).

- [7] Adamova A.A., Avaeva L.G., Lokhturov A.V. Analysis of the prospects for integration of wireless sensor networks with the internet using standard 6LOWPAN. *Nadezhnost' i kachestvo slozhnykh system* [Reliability & Quality of Complex Systems], 2019, no. 1(25), pp. 54–61. DOI: <https://doi.org/10.21685/2307-4205-2019-1-6> (in Russ.).
- [8] Zimin D.V., Murav'yev K.A. [Analysis of energy efficiency problems of wireless data transmission networks based on ZIGBEE protocol stack]. *Tr. mezhd. simp. Nadezhnost' i kachestvo* [Proc. Int. Symp. Reliability and Quality], 2016, vol. 1, pp. 195–197 (in Russ.).
- [9] Krasnobryzhiy B.V., Lavrov I.V., Murav'yev K.A., et al. The review of wireless touch networks on the basis of the ZIGBEE standard. *Naukoemkie tekhnologii i intellektual'nye sistemy. XV Molodezh. nauch.-tekh. konf.* [High Technologies and Intellectual Systems. XV Youth Sci.-Tech. Sys.], 2013, pp. 306–314 (in Russ.).
- [10] Universal asynchronous receiver/transmitter (UART). *ti.com: website*. URL: <http://www.ti.com/lit/ug/sprugp1/sprugp1.pdf> (accessed: 08.12.2019).
- [11] Frenzel L.E. Jr. Handbook of serial communications interfaces. Elsevier, 2015.
- [12] USART. *ww1.microchip.com: website*. URL: <http://ww1.microchip.com/downloads/en/devicedoc/usart.pdf> (accessed: 08.12.2019).
- [13] AT command set. *mt-system.ru: website*. URL: http://www.mt-system.ru/sites/_default/files/documents/simcom_sim5360_atc_en_v0.22.pdf (accessed: 08.12.2019).
- [14] SPI (serial peripheral interface). *ru.bmstu.wiki: website* (in Russ.). URL: [https://ru.bmstu.wiki/index.php?title=SPI_\(Serial_Peripheral_Interface\)&mobileaction=toggle_view_mobile](https://ru.bmstu.wiki/index.php?title=SPI_(Serial_Peripheral_Interface)&mobileaction=toggle_view_mobile) (accessed: 08.12.2019).
- [15] SPI block guide V03.06. *web.archive.org: website*. URL: <https://web.archive.org/web/20150413003534/http://www.ee.nmt.edu/~teare/ee308l/datasheets/S12SPIV3.pdf> (accessed: 08.12.2019).
- [16] Novitskiy A. Serial peripheral interface in microcontrollers from “A to Z” and its realization by the example of ADuC70xx from AnalogDevices. *Komponenty i tekhnologii*, 2009, no. 3, pp. 53–60 (in Russ.).
- [17] SPI. *webhamster.ru: website* (in Russ.). URL: <https://webhamster.ru/mytetrashare/index/mtb186/1517214119k1qn5rp0h8> (accessed: 08.12.2019).
- [18] UM10204 I2C. *nxp.com: website* (in Russ.). URL: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf> (accessed: 08.12.2019).
- [19] Using the I2C bus. *robot-electronics.co.uk: website*. URL: <https://www.robot-electronics.co.uk/i2c-tutorial> (accessed: 08.12.2019).
- [20] Petrov A.A. Data transmission protocol for UART. *Dostizheniya nauki i obrazovaniya* [Achievements of Science and Education], 2016, no. 1(2), pp. 9–10 (in Russ.).
- [21] Murav'yev K.A., Khubbatulin M.E. VPN setup technique on the MIKROTIK routers. *Tekhnologii inzhenernykh i informatsionnykh system* [Technologies of Engineering and Information Systems], 2019, no. 2, pp. 22–33 (in Russ.).
- [22] Semenov B.Yu. Shina I2C v radiotekhnicheskikh konstruktsiyakh [I2C bus in radiotechnical constructions]. Moscow, Solon-Press Publ., 2004 (in Russ.).

Katukiya G.T. — Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University, Moscow, Russian Federation.

Dorofeev A.V. — Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University, Moscow, Russian Federation.

Lintvarev D.E. — Student, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Glushko A.A., Cand. Sc. (Eng.), Assoc. Professor, Department of Electronic Equipment Design and Technology, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Katukiya G.T., Dorofeev A.V., Lintvarev D.E. Analysis and comparison of data transmission interfaces on microcontrollers. *Politekhnicheskiy molodezhnyy zhurnal* [Politechnical student journal], 2020, no. 07(48). <http://dx.doi.org/10.18698/2541-8009-2020-07-624.html> (in Russ.).