

USB для начинающих

Печатается с разрешения журнала Chip News Украина (<http://www.chipnews.com.ua>)

Евгений Рахно (Киев)

Если задать разработчику вопрос: «Используете ли вы в своих проектах USB?», он, скорее всего, ответит: «Да». Но если попросить его уточнить, что и как он использует, то наверняка получите ответ: «Я использую специализированную микросхему». В этой статье мы рассмотрим два подхода к реализации USB – специализированную микросхему-преобразователь USB – USART MCP2200 и микроконтроллеры с USB на борту.

С точки зрения взаимодействия с компьютером, существуют два типа устройств: «стандартные» устройства, требующие подключения лишь для конфигурации и/или обмена данными, но большинство времени проводящие вдали от компьютера, и устройства, полностью зависящие от компьютера (специализированные клавиатуры, манипуляторы, устройства ввода-вывода данных и т.п.).

Для первого типа, особенно если это существующее серийное изделие, наиболее приемлемым вариантом является использование специализированной микросхемы MCP2200.

Несомненным плюсом данного способа является простота использования – достаточно подключить USB с одной стороны, а микроконтроллер – с другой, и MCP2200 сможет передавать данные, принятые по USART от микроконтроллера в виртуальный COM-порт, появившийся в компьютере. Вам понадобится лишь модернизация печатной платы и небольшая корректировка кода программы.

Основные параметры и характеристики MCP2200:

- интерфейс USB 2.0;
- поддержка Full Speed USB 2.0 (12 Мбит/с);
- буфер приёма и передачи 128 байт;
- полностью конфигурируемые значения VID и PID строковых дескрипторов;
- возможность питания от USB-интерфейса;
- Class 02h – CDC;
- Class 03h – HID;
- интерфейс UART;
- аппаратный контроль потока;
- возможность динамической смены скорости обмена;
- восемь линий ввода/вывода общего назначения;

- 256 байт EEPROM;
- выходы для подключения светодиодов статуса (RX/TX);
- рабочее напряжение 3,0...5,5 В;
- ESD-защита более 4 кВ;
- используются стандартные драйверы виртуального COM-порта;
- корпус – 20-выводный QFN, SOIC, SSOP.

Альтернативой специализированным USB-USART-преобразователям могут служить микроконтроллеры с модулем USB на борту. Такой подход требует больших временных затрат, но это сполна компенсируется гибкостью конечного продукта.

«Для чего это нужно, – спросите вы, – ведь есть специализированная микросхема?»

Во-первых, специализированная микросхема стоит денег.

Во-вторых, наличие специализированной микросхемы не отменяет использования микроконтроллера, а микроконтроллер тоже стоит денег.

В-третьих, хороший инженер старается не просто решить задачу, а решить её красиво. Например, настоящий инженер никогда не будет «городить огород» из АЦП, источника опорного напряжения, микроконтроллера с поддержкой USART и специализированной микросхемы USART-USB, если измеряемое напряжение можно подать на ножку микроконтроллера, произвести измерение встроенным модулем АЦП и, используя опять-таки встроенный модуль USB, передать данные в компьютер. Одна микросхема вместо четырёх!

Красиво?

При этом необходимо учесть, что на сегодняшний день компания Microchip серийно выпускает микроконтроллер PIC18F14K50, цена которого сопоста-

вила с ценой специализированной микросхемы, а возможности ограничены лишь вашими знаниями языков программирования. Ниже приведены краткие характеристики данного микроконтроллера:

- MIPS – 12;
- память программ (Flash) – 16 Кб;
- ОЗУ – 768 Кб;
- EEPROM – 256 Кб
- А/Е/UASRT – 1, MSSP (SPI/I²C) – 1;
- ECCP – 1;
- таймеры – 1 × 8, 3 × 16;
- АЦП – 9 каналов, 10 бит;
- два компаратора;
- Cap Touch – 9 каналов;
- корпус SOIC 20.

Что же останавливает разработчиков, не позволяя им использовать микроконтроллеры с аппаратной поддержкой USB? С нашей точки зрения, основным сдерживающим фактором является высокий уровень «абстракции» данного протокола.

Например, если вы решили освоить SPI или I²C, то вам необходимо будет прочесть 5 – 10 страниц документации, а затем спаять схему и запрограммировать микроконтроллер. Не получилось с первого раза – получится со второго, не получилось со второго – получится с третьего. Не получилось и с третьего? Тогда на помощь придёт любой осциллограф. И это максимальный набор необходимого оборудования! Минимальный же – это либо просто паяльник, либо паяльник и недорогой логический анализатор (к примеру, PICkit Serial Analyzer).

Если же вы приступаете к изучению USB, то на вас сразу же сваливается целая гора документации – документация на микроконтроллер, документация на встроенный USB-модуль, спецификация USB, спецификации CDC, HID, MSD и т.д., и т.п.

А проверить как? Осциллограф тут не особо-то и поможет – даже если удастся захватить несколько кадров, то проверить «что, где и как» без понимания всей механики работы USB не получится. А откуда такое понимание у новичка? Вот и возникает желание взять USB-to-COM-преобразователь и не мучить себя.

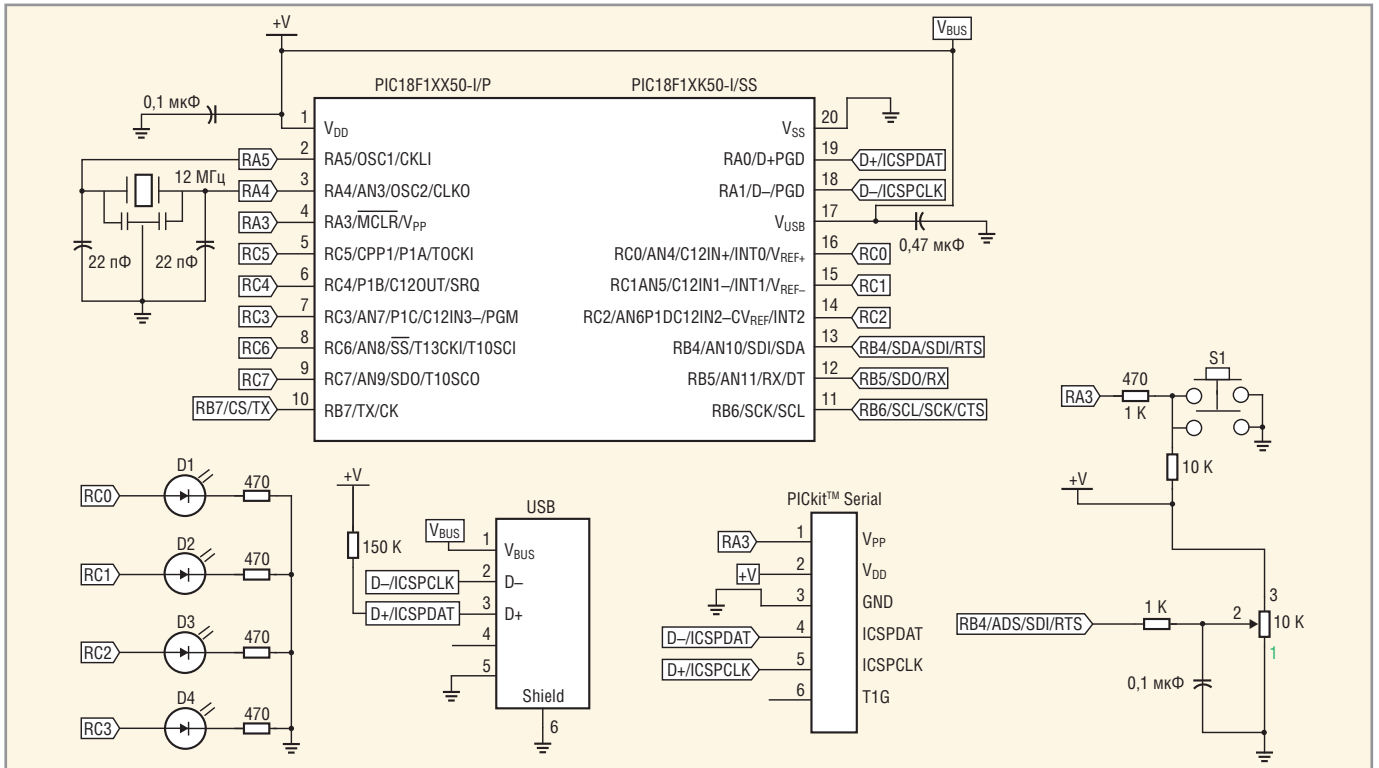


Рис. 1. Схема подключения микроконтроллера PIC18F14K50

Что мы советуем людям, попавшим в подобную ситуацию?

Первое – всегда надо помнить, что компания Microchip не только выпускает продукт, но и обеспечивает полную техническую поддержку.

Второе – USB, действительно, довольно «абстрактный» протокол, но со стороны компьютера всё довольно-таки чётко стандартизировано: написаны драйверы для большинства классов USB-устройств и вероятность того, что

Второе – USB, действительно, довольно «абстрактный» протокол, но со стороны компьютера всё довольно-таки чётко стандартизировано: написаны драйверы для большинства классов USB-устройств и вероятность того, что

Второе – USB, действительно, довольно «абстрактный» протокол, но со стороны компьютера всё довольно-таки чётко стандартизировано: написаны драйверы для большинства классов USB-устройств и вероятность того, что

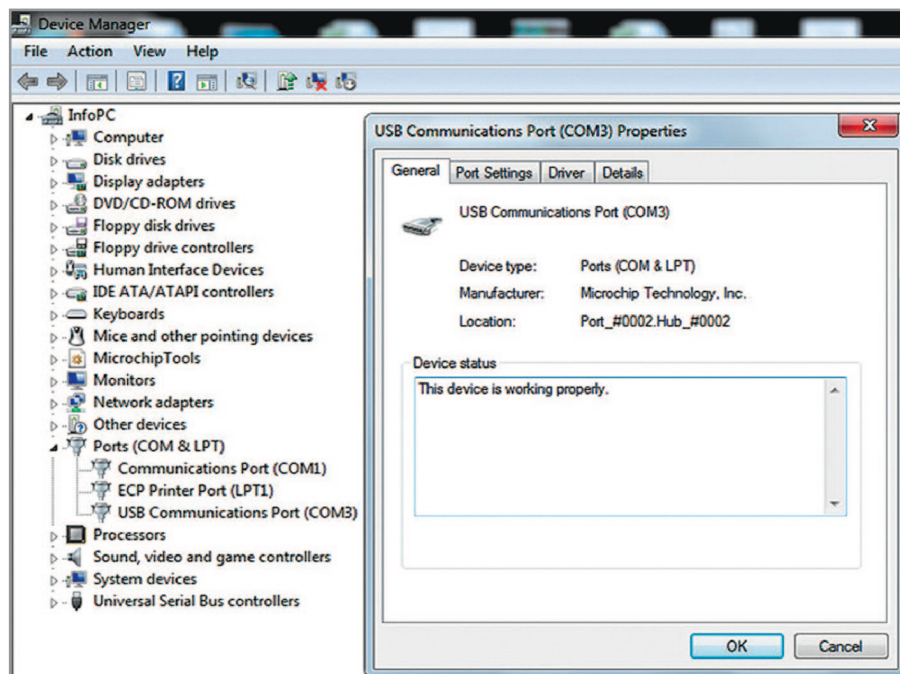


Рис. 2. Отображение виртуального COM-порта во вкладке «Устройства»

вам придётся писать какой-либо драйвер самостоятельно, равна нулю (более того – драйверы для такого класса, как HID (Human Interface Devices – мыши, клавиатуры, джойстики, считыватели штрих-кодов), входят в Windows и не требуют установки, что гарантирует работоспособность вашего устройства на любом из компьютеров с данной операционной системой).

Третье, вы не первый – всегда найдутся люди, прошедшие этот путь до вас. Форумы по электронике для того и созданы, чтобы сообща бороться с проблемами, если они возникают.

С чего начать?

Начать предлагаем с поднятия самооценки, а ничто так не поднимает самооценку (ну и настроение), как работающий проект у вас на столе! Поэтому:

- перейдите по ссылке www.microchip.com/usb;
- прочтите краткое введение в USB;
- вкратце ознакомьтесь с FAQ;
- перейдите по ссылке *USB Software & Tools*, затем выберите *MCHPFSUSB Framework* → *Microchip Application Libraries Download Page*;
- скачайте Microchip Application Libraries и Microchip Application Libraries Help Files;

Корпуса USB микроконтроллеров

Корпус	Микроконтроллер
SOIC20	PIC18F14K50
DIP28; SOIC28	PIC18F2455; PIC18F2550; PIC18F27J53
DIP40; TQFP44	PIC18F4550; PIC18F4553; PIC18F46J50
TQFP64; TQFP80; TQFP100	PIC24FJ64GB106; PIC24FJ128GB106; PIC32MX440F128H; PIC32MX440F512H; PIC32MX695F512L; PIC32MX795F512L

- установите библиотеку на свой компьютер.

Для начала работы с USB мы рекомендуем приобрести либо изготовить самостоятельно демо-плату на основе PIC18F14K50 (рис. 1).

Начать, пожалуй, можно с самого простого – эмуляции COM-порта: C:\Microchip Solutions\USB\Device – CDC – Serial Emulator → USB Device – CDC – Serial Emulator – C18 – Low Pin Count USB Development Kit. После ознакомления с документацией необходимо произвести компиляцию и программирование микроконтроллера.

При подключении микроконтроллера к компьютеру будет найдено новое устройство и Windows попросит вас установить драйверы. Лучше не полагаться на Windows и установить драйверы самостоятельно – C:\Microchip Solutions\USB\Device – CDC – Serial Emulator\inf. После завершения установки драйверов появится дополнительный COM-порт (рис. 2).

На этом ваши проблемы с USB заканчиваются. Так как исходники полностью открыты, вы можете попрактиковаться в программировании, например, добавить в код программы фрагмент, передающий в компьютер

показания АЦП (необходимо использовать функции mUSBUSARTIsTxTrfReady и mUSBUSARTTxRam).

Теперь создадим более интересное устройство – HID. Для этого нам понадобится демо-плата с микроконтроллером PIC18F14K50 (рис. 1).

Компания Microchip приготовила для вас три заготовки: джойстик, клавиатуру и мышку. Советуем начать с клавиатуры: C:\Microchip Solutions\USB\Device – HID – Keyboard → USB Device – HID – Keyboard – C18 – Low Pin Count USB Development Kit.

После компиляции проекта можно увидеть, что остались свободными 60% памяти программ и 60% памяти ОЗУ – этого более чем достаточно для написания вполне увесистого приложения обработки клавиатуры.

Далее стандартно – запрограммировали микроконтроллер, подключили его к USB и наблюдаем результат.

На этот раз всё пройдет без проблем, так как драйвер HID клавиатуры уже установлен на ваш компьютер. Так как исходники открыты, вы без труда модернизируете данный проект под свои нужды.

Как же компьютер узнает, какое именно устройство подключилось к USB? За это отвечает специальное «описание», передаваемое устройством, которое называется «дескриптором». От того, какие именно поля вы в него включите, будет зависеть то, как компьютер будет интерпретировать данные, получаемые от вашего устройства. Написать дескриптор можно и самостоятельно, но проще воспользоваться программой HID Descriptor Tool, которую можно скачать с сайта www.usb.org. С помощью этой программы можно не только создать дескриптор с нуля, но и воспользоваться готовыми дескрипторами, входящими в комплект, модифицировав их под свои требования.

Надеемся, что после прочтения данной статьи у вас появится не только демо-плата с микроконтроллером PIC18F14K50, но и уверенность в том, что USB – это не так страшно, как казалось ранее.

В заключение отметим, что микроконтроллер PIC18F14K50 был выбран нами исключительно из тех соображений, что на сегодняшний день он является самым миниатюрным микроконтроллером со встроенным USB-модулем. Выбрать контроллер, соответствующий вашим запросам, поможет таблица.

