Dipl.-Ing. Jürgen Hulzebosch

USB in der Elektronik

FRANZIS
PC+ELEKTRONIK

Dipl.-Ing. Jürgen Hulzebosch

USB in der Elektronik

Die USB-Schnittstelle für praktische Anwendungen am PC einsetzen

Юрген Хульцебош

USB в электронике

2-е издание, исправленное

Санкт-Петербург
«БХВ-Петербург»
2011

УДК 681.3.06
ББК 32.973
Х98

Хульцебош Ю.

Х98 USB в электронике: Пер. с нем. — 2-е изд., испр. — СПб.: БХВ-Петербург, 2009. — 224 с.: ил. + CD-ROM — (Электроника)

ISBN 978-5-9775-0658-8

В книге показано, как с помощью специализированных микросхем USB без интегрированного микроконтроллера создавать различные системы управления и устройства. Рассмотрены основы USB, аппаратное обеспечение (микросхемы, флэш-модули и др.), установка драйверов и разработка программ на Visual Basic. Приведены практические примеры различных устройств от простых (светофор, аварийная сигнализация, устройство для наблюдения за уровнем воды в аквариуме и др.) до более сложных (тестер дистанционного управления, устройство записи EEPROM-памяти, аналого-цифровой преобразователь и др.). Показано, как разработать универсальный интерфейс USB-I^2C, осуществить Flash-программирование AT89LP-микроконтроллера через SPI/ISP-интерфейсы и многое другое. На компакт-диске находятся примеры программ на языке Visual Basic, описания и спецификация электронных компонентов, а также специализированные драйверы и утилиты.

Для профессиональных инженеров-электронщиков и радиолюбителей

УДК 681.3.06
ББК 32.973

Die berechtigte Übersetzung von deutschsprachiges Buch USB in der Elektronik, ISBN: 978-3-7723-4089-5. Copyright © 2008 Franzis Verlag GmbH, 85586 Poing. Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Das Erstellen und Verbreiten von Kopien auf Papier, auf Datenträger oder im Internet, insbesondere als PDF, ist nur mit ausdrücklicher Genehmigung des Verlags gestattet und wird widrigenfalls strafrechtlich verfolgt. Die Russische Übersetzung ist von BHV St. Petersburg verbreitet, Copyright © 2009.

Авторизованный перевод немецкой редакции книги USB in der Elektronik, ISBN: 978-3-7723-4089-5. Copyright © 2008 Franzis Verlag GmbH, 85586 Poing. Все права защищены, включая любые виды копирования, в том числе фотомеханического, а также хранение и тиражирование на электронных носителях. Изготовление и распространение копий на бумаге, электронных носителях данных и публикация в Интернете, особенно в формате PDF, возможны только при наличии письменного согласия Издательства Franzis. Нарушение этого условия преследуется в уголовном порядке. Перевод на русский язык "БХВ-Петербург" © 2009.

ISBN 978-3-7723-4089-5 (нем.)
ISBN 978-5-9775-0658-8 (рус.)

© 2008 Franzis Verlag GmbH, 85586 Poing
© Перевод на русский язык "БХВ-Петербург", 2009

Оглавление

Предисловие **8**

Глава 1. Измерение, управление и регулировка при помощи интерфейсов ПК **11**
1.1. Понятие о данных и единицах информации 14
1.2. Двоичный счет (логические 0 и 1) 15
1.3. "Счет на пальцах" 16
1.4. Биты и байты 16

Глава 2. Основы USB **19**
2.1. USB 1.0, 1.1, 2.0, On-The-Go 19
2.2. Идентификация устройств USB-интерфейса. Ток и напряжение линий электропитания 20
2.3. Последовательная передача данных в интерфейсах USB, RS-232, SPI и I^2C 21
2.4. Кодировка NRZI 22
2.5. Последовательная шина USB 22
2.6. Типы USB-передач 24
2.7. USB-драйвер 25
2.8. Идентификация USB-устройства 26

Глава 3. Аппаратное обеспечение **27**
3.1. USB-адаптер и описание 28
3.2. Дополнительная плата 31
3.3. UM232R-модуль от компании FTDI 33
3.4. Внутренняя структура микросхемы FT232R от FTDI 35
3.5. Функции микросхемы FT232R 36
3.6. Пример последовательного подключения микроконтроллера к USB 37
3.7. Согласование уровней напряжения RS-232/485 39
3.8. Генератор скорости передачи данных 41

Глава 4. Установка драйвера FTDI версии 2.x **43**
4.1. Программа для отображения USB-устройств (утилита USB View) 47
4.2. Удаление FTDI-драйвера 48

Глава 5. Начало работы **49**
5.1. Вызов первой демонстрационной программы на Visual Basic (VB) 50
5.2. Первые обращения программы к FTD2XX.DLL-библиотеке 51
5.3. Пример программы на Visual Basic 51
5.4. Объявление функций драйвера FTD2XX для Visual Basic 52
5.5. Исходный код функций *FT_ListDevices* и *FT_OpenEx* 53
5.6. Другие вызовы функции *FT_ListDevices* 55
5.7. Вызовы функций *FT_OpenEx* и *FT_Close* 56

Глава 6. "Игры" со светом ... 59
6.1. Включение светодиода ... 60
6.2. Переключение светодиода ... 63
6.3. Еще вариант переключения ... 63
6.4. Вспышка светодиода ... 63
6.5. Управление яркостью светодиода ... 66
6.6. Управление двухцветным светодиодом ... 68
6.7. Мигающее светосигнальное устройство ... 71
6.8. Включение выхода TxD ... 72
6.9. Пример схемы светофора с тремя светодиодами ... 73
6.10. Пример схемы USB-осветителя для чтения ... 74

Глава 7. Опрос входов ... 77
7.1. Система сигнализации ... 80
7.2. Счетчик сигналов тревоги ... 82
7.3. Пример схемы системы охранной сигнализации ... 82
7.4. Здесь ли кошка? ... 84
7.5. Осторожно, вода ... 85
7.6. Светло или темно? ... 86
7.7. Применение оптического фотоэлемента в аварийной сигнализации ... 87
7.8. Более точное определение сопротивления фоторезистора ... 88

Глава 8. Управление кварцевыми часовыми механизмами ... 93
8.1. Подключение катушки ... 93
8.2. Программное обеспечение ... 94

Глава 9. Режим Bit Bang ... 97
9.1. Синхронный режим Bit Bang ... 98
9.2. Опрос входных сигналов от D0 до D7 при помощи режима Bit Bang ... 103
9.3. Исходный код для режима Bit Bang ... 105
9.4. Режим Bit Bang и эмуляция других портов ... 107

Глава 10. Простой АЦП с использованием режима Bit Bang ... 111
10.1. Понятие аналого-цифрового преобразователя (АЦП) ... 111
10.2. Электрическая схема АЦП с компаратором ... 113
10.3. Первое тестирование ПО для АЦП ... 114
10.4. Согласование между программным и аппаратным обеспечением USB ... 116
10.5. Исходный код к АЦП ... 119
10.6. Добавочный операционный усилитель ... 122
10.7. Измерение напряжения вольтметром на аналоговом входе E2 ... 124
10.8. Тестер батареек питания ... 128

Глава 11. Измерение температуры при помощи терморезистора с отрицательным ТКС ... 129
11.1. Подключение терморезистора и запуск ПО для измерения температуры ... 130
11.2. Исходный код программы для измерения температуры ... 132

Глава 12. Генерирование сигналов различных частот и их применение ... 135
12.1. Генератор частот для последовательного интерфейса ... 135
12.2. Генератор частот с использованием режима Bit Bang ... 135
12.3. Цифроаналоговый преобразователь с ШИМ ... 137

Глава 13. Хранение данных в EEPROM-памяти ... 141
13.1. Основы EEPROM-памяти ... 141
13.2. Основы интерфейса I^2C ... 141
13.3. Подключение EEPROM-памяти ... 143
13.4. Предварительные размышления ... 144
13.5. Пять шагов к успеху ... 146
13.6. "Выуживание" данных ... 154
13.7. Эксплуатация программы EEPROM-накопителя условного кода ... 157
13.8. Программа EEPROM-накопителя условного кода — фрагменты исходного текста ... 158
13.9. Пример программы двухпроводной связи по интерфейсу I^2C ... 160

Глава 14. Инфракрасное дистанционное управление ... 163
14.1. Инфракрасная передача данных по протоколу RC5 ... 164
14.2. Пример программы тестирования инфракрасного дистанционного управления ... 165
14.3. Исходный код программы тестирования инфракрасного дистанционного управления ... 168
14.4. Управление дополнительной ведомой вспышкой при помощи фотодиода ... 170
14.5. Обработка сигналов с представлением результата в виде временной диаграммы ... 171

Глава 15. Анализатор для цифровых сигналов с частотами до 60 кГц ... 173

Глава 16. 8-канальный логический анализатор ... 175
16.1. Исследование цифровых схем ... 178

Глава 17. Управление шаговыми двигателями ... 179
17.1. Схема подключения униполярного шагового двигателя ... 180
17.2. Пошаговое управление ... 181
17.3. Пример программы для управления шаговым двигателем ... 183

Глава 18. Использование USB для защиты программ от копирования ... 187
18.1. Вызов FTDI-функций в Visual C ... 189

Глава 19. Изменение данных в EEPROM-памяти ... 193

Глава 20. Последовательная запись и чтение без VCP-драйвера ... 197

Глава 21. Подключение набора для изучения микроконтроллера к компьютеру с помощью USB ... 203

Глава 22. Пример флэш-программирования микроконтроллера Atmel-AT89LP ... 207
22.1. ISP-программирование микроконтроллера Atmel AT89LPx052 посредством интерфейса SPI ... 209
22.2. Пример на Visual Basic — чтение 2 Кбайт флэш-памяти ... 212

Приложение. Описание компакт-диска ... 217

Список источников информации ... 220

Предметный указатель ... 221

Предисловие

В последние годы USB стал универсальным интерфейсом. Наряду с компьютерной индустрией вряд ли осталась какая-либо область электроники, не затронутая USB, будь то автомобильный радиоприемник с USB-входом для подключения MP3-плеера, цифровой спутниковый приемник с обновлением через USB или современный сотовый телефон с MP3 и интегрированной камерой, которая имеет USB-порт.

USB — это сокращение Universal Serial Bus — универсальная последовательная шина. Интерфейс USB версии 1.0 был разработан еще в 1995 г. консорциумом нескольких больших предприятий по электронике.

В сравнении с последовательным COM-портом или параллельным LPT-портом персонального компьютера (ПК) USB обладает достаточно высокой скоростью передачи данных. Кроме того, USB поддерживает "горячее" подключение и отключение устройств.

При более детальном рассмотрении USB сложность этого интерфейса сначала может слегка испугать. Однако если раньше для управления собственной электроникой имелась возможность использования только параллельного или последовательного интерфейсов ПК, то сейчас нужно обязательно разбираться в USB, поскольку ПК новейших поколений оборудованы только этим интерфейсом.

Следует отметить, что большое количество литературы по USB можно найти в Интернете, у различных издательских компаний, а также у производителей микроконтроллеров. В большинстве случаев описываемые USB-контроллеры имеют микропроцессорное ядро, вследствие чего кажущийся простым USB-интерфейс снова становится достаточно сложным.

В книге мы не будем очень подробно останавливаться на функционировании USB. Используемая здесь USB-микросхема, произведенная компанией FTDI, не имеет интегрированного микроконтроллера. Однако, несмотря на это, в книге показано, как и без применения микроконтроллера могут быть созданы достаточно интересные USB-системы управления, сбора (и регистрации) данных.

Изначальный мотив изготовителей FTDI-микросхемы заключался в разработке преобразователя интерфейса USB в последовательный и параллельный интерфейсы. С помощью такого преобразователя, при необходимости, современные ПК можно дооборудовать и этими уже устаревшими интерфейсами. Все это послужило

основанием для автора лично разобраться с USB- и FTDI-микросхемами, поскольку новейшие ПК более не имеют последовательного интерфейса, а микроконтроллеры программируются именно через последовательный интерфейс.

Однако скоро было установлено, что хотя простая последовательная передача данных с помощью преобразователя функционировала, переключение отдельных сигналов последовательного интерфейса все же было слишком медленным по сравнению с обычным последовательным интерфейсом (RS-232) устаревшего ПК. Автора заинтересовал вопрос: не является ли будто бы быстрый USB-интерфейс на самом деле достаточно медленным. Из этой постановки вопроса, в конце концов, и родилась эта книга с большим количеством практических примеров.

В повседневной жизни встречается множество ситуаций, в которых можно использовать различные электронные устройства, управляемые при помощи компьютера. Книга начинается с простых экспериментов со светодиодом и светофором, подключенных к ПК с помощью интерфейса USB. Далее рассмотрены примеры схем аварийной сигнализации, устройства для наблюдения за уровнем воды в аквариуме, а также и другие устройства.

К концу книги представленные примеры становятся все сложнее и сложнее. Измерения яркости или температуры при помощи собственноручно построенного аналого-цифрового преобразователя способствуют закладке основ знаний, которые необходимы для создания как программного, так и аппаратного обеспечения для USB. Кое-кто, возможно, удивится, как легко с помощью нескольких строк в прикладном программном обеспечении можно управлять построенным собственными руками аналого-цифровым преобразователем. Тестер инфракрасного дистанционного управления на фотодиоде, вольтметр, запоминающее устройство на электрически программируемом ПЗУ (EEPROM-память) и использование USB-интерфейса для защиты программ от копирования являются дальнейшими практическими примерами.

Поскольку используемый USB-адаптер может служить в качестве преобразователя USB-интерфейса в последовательный интерфейс (COM-порт) микроконтроллера, приводятся дополнительные примеры программного обеспечения (ПО).

Достаточно опытный электронщик научится, каким образом можно использовать линии ввода/вывода (I/O) для более сложных интерфейсов, таких как I^2C (Inter-Integrated Circuit) или SPI (Serial Peripheral Interface). Работа с USB-I^2C-интерфейсом становится по сути "детской игрой", т. к. логический анализатор с помощью программного обеспечения фактически визуализирует сигнал данных и тактовый сигнал в виде генерируемой временной диаграммы. В книге рассмотрена поэтапная разработка интерфейса. Фактически за пять достаточно простых шагов вы сможете разработать универсальный USB-I^2C-интерфейс.

Последний пример в этой книге покажет вам основы того, как можно выполнить более сложные задания, например, флэш-программирование микроконтроллера Atmel AT89LPx052 посредством интерфейса SPI и внутрисистемного программирования ISP (In-System Programming).

Те, кто уже умеет разрабатывать, собирать и программировать электронику на микроконтроллерах, могут использовать описываемый в этой книге адаптер непосредственно с помощью преобразователя USB-интерфейса в последовательный интерфейс (COM-порт) или с помощью программируемого адаптера SPI/RS-232.

Отдельные примеры детально разъясняются и достаточно иллюстрируются, позволяя понять взаимосвязи программного обеспечения, выполненного на языке Visual Basic, с аппаратным обеспечением и USB-микросхемой FT232R. Эти взаимосвязи могут быть использованы и для собственных идей и разработок.

Те, кто любит экспериментировать и желает на практике воплотить некоторые приведенные примеры схем, должны иметь соответствующие навыки и уметь, по крайней мере, пользоваться паяльником.

Актуальную дополнительную информацию вы сможете найти на сайте: **www.minimikro.de**.

ГЛАВА 1

Измерение, управление и регулировка при помощи интерфейсов ПК

При использовании ПК для электронных экспериментов требуется хотя бы один интерфейс для внешней среды или для собственной созданной электроники.

Посредством этого интерфейса ПК и соответствующего программного обеспечения компьютер общается с подключаемой электроникой (рис. 1.1). Во многих случаях для этого использовались (и все еще иногда используются) последовательные или параллельные интерфейсы ПК.

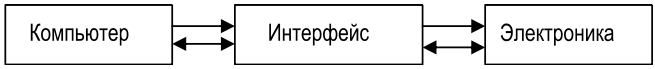

Рис. 1.1

Как можно догадаться из названия, при *последовательном интерфейсе* данные пересылаются последовательно, т. е. наименьшие единицы информации передаются по сигнальному проводу друг за другом. В последовательном интерфейсе ПК для отправки и приема передаваемых данных применяют соответствующие отдельные линии. Кроме того, при последовательном интерфейсе необходимы и другие линии, предназначенные для сигналов управления, с помощью которых осуществляется процедура передачи данных от компьютера к электронике (электронному устройству), или наоборот (рис. 1.2).

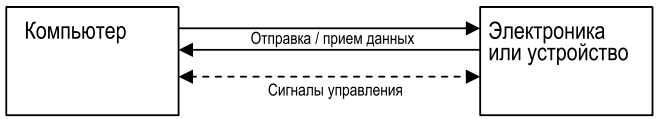

Рис. 1.2

Типичные устройства, которые подключаются к последовательному интерфейсу (COM-порту), — это модемы, измерительные устройства, а также мышь. Для соединения нужны 9- и 25-контактные электрические разъемы D-Sub.

Примечание

В USB используется последовательная передача данных.

В стандартном параллельном интерфейсе ПК одновременно может осуществляться передача восьми состояний на восьми информационных линиях. Как и в случае с последовательным интерфейсом, в параллельном интерфейсе также имеются дополнительные сигналы управления (рис. 1.3).

Рис. 1.3

Параллельный интерфейс использовался в основном для подключения принтеров. Для подсоединения применялись 25-контактные электрические разъемы D-Sub, которые устанавливались на ПК, и 36-контактные разъемы Centronics — на внешних периферийных устройствах.

Обычно к последовательному или параллельному интерфейсу ПК подключается только один прибор. Если один принтер уже подключили к параллельному интерфейсу, то для второго принтера нужен еще один параллельный интерфейс. Если нет в наличии или недостаточно свободных интерфейсов, то тогда ПК можно дооборудовать платами расширения, которые вставляются в различные внутренние слоты. Обусловленные прежними определениями аппаратного и программного обеспечений и операционной системы, начиная от DOS и Windows, вплоть до Windows XP в общем случае поддерживаются два параллельных и до 32 последовательных интерфейсов. Со специальной поддержкой изготовителя приборов возможны до 8 параллельных и 256 последовательных интерфейсов.

Как уже, наверняка, известно большинству читателей, для устройств, подключенных к последовательному или параллельному интерфейсу, необходимо установить соответствующее ПО — *драйвер*. Этот драйвер, как правило, предоставляется производителем устройства для соответствующей операционной системы. Без подобного драйвера устройство не может использоваться или может, но лишь ограниченно, т. к. операционная система ПК не имеет возможности распознать его.

В измерительной технике, напротив, электронщику для этого интерфейса драйвер может потребоваться лишь в очень редких случаях, потому что он имеет дело непосредственно с функциями аппаратного обеспечения компьютера (с адресами ввода и вывода), которое со времен введения их в IBM PC XT до сих пор поддерживается и в современных системах.

При разработке USB-интерфейса учитывался опыт, который уже был получен при подключении к компьютеру дополнительных устройств, что значительно облегчи-

ло пользователям применение этого интерфейса. USB означает Universal Serial Bus (универсальная последовательная шина) и является интерфейсом для многих современных устройств. USB (исходя от одного подключения USB-хоста) может опознать до 127 подключенных устройств, причем адрес 00 зарезервирован. Хост USB управляет всей активностью на USB-линиях и может целенаправленно обращаться к любому конкретному устройству по его адресу.

По сравнению с существующими последовательным и параллельным интерфейсами область применения USB-интерфейса была значительно расширена. USB-интерфейс находит применение не только в принтерах, модемах, мышках, сканерах и внешних жестких дисках, но он также активно используется и в других электронных приборах, таких как MP3-плееры, сотовые телефоны, видеоплееры, USB-карты памяти и т. д. Они зачастую поддерживаются непосредственно операционной системой без инсталляции драйвера. Шина USB имеет древовидную структуру. Если же ПК не обладает достаточным количеством USB-интерфейсов, то можно выполнить разветвление USB на нужное количество портов при помощи так называемых *хабов*.

Исходя от так называемого USB-хост-контроллера (в нашем примере USB-интерфейса компьютера), другие устройства могут быть включены звездообразно с помощью последовательно присоединенного 4-портового хаба. Хаб имеет один или несколько USB-контактов для подключения других USB-приборов. Хост-контроллер, установленный в ПК, управляет обменом с подключенными USB-устройствами и операционной системой.

Вместе с этим к 4-портовому хабу могут быть подключены и другие хабы. Так, например, USB-контакты, предназначенные для Прибора 4 (рис. 1.4), могут быть использованы для другого 7-портового хаба, благодаря чему количество USB-контактов структуры увеличивается на десять. В этом случае говорят о *каскадировании*.

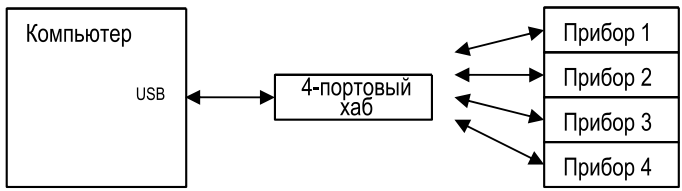

Рис. 1.4

В определенный момент времени только один подключенный USB-прибор может обмениваться с одним USB-хост-контроллером. Отправка сообщения одновременно на несколько подключенных приборов (*Broadcasting* — транслирование сообщений, режим передачи сообщения всем или нескольким абонентам сети одновременно) невозможна. Для повышения ширины полосы пропускания современные ПК имеют зачастую два, четыре или восемь хост-контроллеров. Если несколько USB-устройств должны обмениваться между собой данными, например, копировать данные с одного USB-жесткого диска на другой, и наоборот, имеет смысл подключить жесткие диски к разным USB-хост-контроллерам, а не к одному.

USB-разъем имеет только 4 соединительных контакта. Помимо двух контактов, предназначенных для подключения к источнику постоянного тока, которое не свойственно для последовательного или параллельного интерфейсов, имеется всего лишь две линии для передачи данных, по которым и осуществляется обмен с хост-контроллером. Вместе с тем, USB-разъемы не очень дороги и могут быть миниатюризированы.

Подключенные к USB устройства очень часто автоматически распознаются операционными системами, например, Windows 2000, Windows XP и Vista. Если же устройство подключается впервые, то почти всегда происходит запрос на инсталляцию драйвера. После успешной установки и интеграции драйвера операционная система распознает прибор автоматически.

Иногда, однако, не удается заметить интеграцию драйвера, т. к. операционная система (например, для USB-карты памяти или USB-жестких дисков) уже содержит драйвер в списке совместимости оборудования и автоматически интегрирует его.

1.1. Понятие о данных и единицах информации

Что в компьютерных технологиях или информатике подразумевается под понятием *данные*? Как выглядят данные, которые используются для обработки в компьютерах в соответствии с микроконтроллерами, а также передаются по различным интерфейсам — от электронных устройств к компьютеру или в самом компьютере. В Интернете можно обнаружить примерно следующий ответ:

Данные — это логически сгруппированные единицы информации, которые передаются между (компьютерными) системами. Они состоят из двоичного кода. Требуется наличие кодирующих инструкций, с помощью которых данные могут быть преобразованы.

В информатике и электронной обработке данных под понятием "данные" понимается читаемое (машиной) и обрабатываемое, как правило, цифровое представление информации. Информация для этого сначала кодируется в знаки (или цепочки знаков), структура которых подчиняется строгим правилам.

Данные — это множественное число термина "данное", т. е. наименьшей неделимой единицы информации, которая передается в сообщении. *Сообщение* — это форма представления информации, т. е. совокупность знаков или первичных сигналов, содержащих информацию, которая имеет признаки начала и конца и предназначена для передачи через определенную среду связи. Сообщение имеет определенное значение для получателя и подчиняется фиксированным правилам. В компьютере, работающем в двоичном коде, наименьшими *единицами информации* являются значения 0 и 1. Благодаря расположению в ряд нулей и единиц образуются цепочки знаков и чисел для дальнейшей обработки в программном и аппаратном обеспечении компьютера.

Примечание

Компьютер — это производное от англ. "to compute" (считать или вычислять).

Данные могут быть представлены в аналоговой или цифровой форме и быть соответственно аналоговыми или цифровыми.

Наиболее наглядным примером передачи *аналоговых данных*, наверное, является телефон: если говорить в микрофон телефонной трубки, то звуковые волны речи преобразуются в электрические колебания; эти электрические сигналы (данные) передаются по проводам, а в динамике телефонной трубки получателя они снова преобразуются в звуковые волны.

В противоположность двоичным *цифровым данным* с дискретными значениями 0 или 1 аналоговый сигнал может принять любое произвольное значение внутри установленных предельных значений, например: 0,1; 0,2; 0,2345...

1.2. Двоичный счет (логические 0 и 1)

Сигналы или сигнальные линии в цифровом интерфейсе закодированы в двоичной системе счисления и могут принимать только значения 0 или 1.

Все это становится довольно просто, если представить себе то, как мы считаем при помощи десяти чисел от 0 до 9 в нашей обычной десятичной системе счисления (табл. 1.1).

Таблица 1.1

Десятичное основание 10 (0–9)	Двоичное основание 2 (0,1)	Шестнадцатеричное основание 16 (0–F)
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

При всех системах счисления соответствующие дискретные единицы информации располагаются в ряд, и от этого зависит соответствующее значение позиции внутри информации (данных). Например, десятичное число 14_{10} в разных системах счисления можно представить следующим образом:

14_{10} в десятичной системе $\rightarrow 14 = 1\times10^1 + 4\times10^0 = 10 + 4$

14_{10} в двоичной системе $\rightarrow 1110 = 1\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 = 8 + 4 + 2 + 0$

14_{10} в шестнадцатеричной системе $\rightarrow E = 14\times16^0 = 14$

В приведенных системах счисления различны только основания или количество возможных дискретных значений единиц информации. В десятичной системе ими являются числа от 0 до 9 (до основания 10, десять возможных дискретных значений), а в двоичной системе — два значения 0 и 1 (до основания 2).

1.3. "Счет на пальцах"

Очень просто производить деление двоичного числа на два: нужно сдвинуть все биты на одну позицию направо, а последний бит отбросить. Например, при делении числа 10_{10} (двоичное число 1010_2) на 2_{10} получается двоичное число 101_2 (десятичное 5_{10}). Также и из десятичного числа 11 получится число 5. Умножение на два производится со сдвигом битов налево.

Читатель может проверить следующее умножение двух чисел и попытаться вникнуть в суть дела. Автору этот вид умножения известен под названием "счет на пальцах". Рассмотрим пример умножения 17 на 12 (17×12).

Левый столбец (17) каждый раз делится на 2, правый столбец (12) каждый раз умножается на 2. Остатки при этом не берутся в расчет (табл. 1.2).

Таблица 1.2

17	12
8	24
4	48
2	96
1	192

Сложим только те числа в столбце справа, которые в левом столбце имеют нечетное значение: т. е. 12 + 192 = 204. Ответ подошел, но всегда ли так происходит?

1.4. Биты и байты

Бит (binary digit, двоичный разряд) может принимать логические значения 0 и 1. Если расположить в ряд 8 битов, как в предыдущем примере, то тогда с 8 битами можно досчитать до 255 значений:

$$\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1}\boxed{1} = 1\times2^7 + 1\times2^6 + 1\times2^5 + 1\times2^4 + 1\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0 =$$

$$= 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$$

Итак, 8 битов образуют 1 байт. Байт может принимать 256 значений (от 0 до 255).

Итак, *байт* — это единица информации данных, которая состоит из 8 битов. Первый бит (самый левый, 128) обозначается также MSB (Most Significant Bit, наиболее значащий бит, старший разряд), последний бит (самый правый) — как LSB (Least Significant Bit, наименее значащий бит, младший разряд).

Слово данных (Word) — это единица информации, состоящая из 16 битов, или 2 байтов, которая может принимать 65 536 значений (от 0 до 65 535).

1 Мбайт (1 000 000 байт) равен 10^6 байтов, однако в вычислительной технике под *мегабайтом* принято понимать 2^{20} байт (1 048 576), что приводит к почти 5%-й разнице с первым значением, а под *килобайтом* принято понимать 2^{10} байт.

Иначе: 1 Мбайт = 1024 Кбайт = 2^{10} Кбайт,

1 Кбайт = 1024 байт = 2^{10} байт,

1 Мбайт = 1024×1024 байт = $2^{10}\times2^{10}$ байт = 2^{20} байт.

ГЛАВА 2

Основы USB

2.1. USB 1.0, 1.1, 2.0, On-The-Go

USB 1.0

Известная на сегодняшний момент под названием USB 1.0 последовательная шина была разработана компанией Intel. Спецификации USB 1.0 были обнародованы еще в январе 1996 г.

Интерфейс USB 1.0 имел максимальную скорость передачи, равную 1,5 Мбит/с (низкая скорость — Low-Speed), и поэтому не мог быть полезным для запоминающих устройств сверхбольшой емкости (например, жестких дисков). Однако такие устройства уже поддерживались. Первыми же используемыми устройствами были, например, принтер, сканер, мышь и клавиатура.

USB 1.1

С интерфейсом USB 1.1 стало возможным достигнуть скорости передачи данных, равной 12 Мбит/с (полная скорость — Full-Speed).

USB 2.0

В 2000 г. была обнародована спецификация USB 2.0 с высокоскоростной передачей данных (480 Мбит/с), а также созданы предпосылки для того, чтобы USB стал более привлекателен для периферийных устройств, таких как принтер, жесткий диск, карты памяти и т. д.

USB 2.0 со скоростью передачи данных, большей в 40 раз, чем при высокоскоростном режиме, остался совместим с версиями USB 1.X.

USB On-The-Go

Благодаря USB On-The-Go (OTG), дальнейшего расширения спецификации USB 2.0, оборудованные им USB-устройства могут непосредственно сообщаться между собой. Поэтому от компьютера, который принимает на себя функцию хоста,

можно отказаться. Два USB-устройства могут обмениваться сообщениями без хоста. Это имеет смысл при использовании цифровых камер с USB-разъемом для управления USB-принтерами, чтобы иметь возможность распечатать фотографии непосредственно на принтере, без подключения к компьютеру.

Wireless USB (беспроводная USB)

При упоминании этой спецификации USB речь идет о беспроводном USB. Данная спецификация была выпущена в мае 2005 г. Она делает возможным беспроводную связь между USB-устройствами при полной скорости передачи данных максимум 480 Мбит/с.

2.2. Идентификация устройств USB-интерфейса. Ток и напряжение линий электропитания

Физически USB-разъем имеет в общей сложности 4 линии (рис. 2.1), две для электропитания (земля и +5 В) и две линии для передачи данных (D+ и D–).

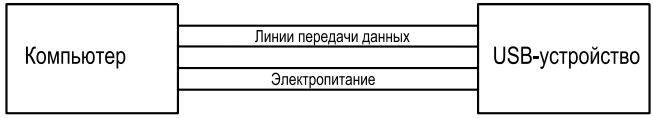

Рис. 2.1

Потребление электроэнергии подключенного USB-устройства не должно превышать 100 мА, если USB-устройство не было сконфигурировано. Максимальное потребление электроэнергии при сконфигурированном USB-устройстве составляет 500 мА. Если необходимо более 100 мА, то этого должно потребовать USB-устройство. Устройства, которые должны питаться непосредственно от шины USB, снабжены опознавательным признаком *bus powered* (питаемый от шины). Само собой разумеется, что USB-устройства также могут быть оснащены своим собственным электропитанием. Такие устройства обладают признаком *self powered* (с собственным источником питания).

Линии передачи данных (D+ и D–) рассчитаны для оптимизации скорости передачи данных. Отдельные состояния сигнала этих линий здесь рассматривать не будем, т. к. они были бы важны только при разработке USB-чипа.

Если USB-устройство подключается к USB-контроллеру или извлекается из него, то линии передачи данных используются по технологии *Plug-and-Play* (досл. англ. — подключи и работай). Если линия передачи данных D+ подключенного USB-устройства относительно напряжения питания имеет сопротивление 1,5 кОм, то устройство в этом случае определяется в качестве полноскоростного или иначе высокоскоростного устройства. Если же то же самое можно сказать для линии D–, то тогда это низкоскоростное устройство.

Все USB 2.0-совместимые устройства при подключении к шине с полной скоростью (Full-Speed) определяются как устройства интерфейса USB 1.1. Только когда хост-контроллер обнаружит, что речь идет об устройстве, приспособленном к интерфейсу USB 2.0, происходит переключение скорости через хост.

Если обе линии для передачи данных D+ и D– остаются на низком логическом уровне дольше 2,5 мкс, то тогда это состояние определяется как *disconnect* (разъединение), и устройство можно отсоединить.

Длина USB-кабеля не должна превышать 5 м. Если потребуется кабель большей длины, то тогда на каждые 5 м кабеля нужно использовать USB-хаб.

2.3. Последовательная передача данных в интерфейсах USB, RS-232, SPI и I^2C

Краткое сопоставление наиболее часто используемых последовательных интерфейсов приведено в табл. 2.1.

Таблица 2.1

Параметр	Интерфейс			
	USB 2.0	**RS-232**	**SPI**	**I^2C**
Количество устройств	127	2	8	40
Скорость передачи данных	до 480 Мбит/с	до 115 Кбит/с	до 2 Мбит/с	до 2 Мбит/с
Максимальное расстояние	5 м, макс. 30 м	до 30 м	до 3 м	до 5 м

USB служит для подключения самых разных внешних устройств. Интерфейс RS-232 можно встретить, например, в модемах, мышах и измерительных приборах.

Интерфейсы SPI (Serial Peripheral Interface) и I^2C (Inter-IC bus) служат преимущественно для внешнего соединения микроконтроллера с другими периферийными микросхемами (узлами) и поэтому поддерживают работу лишь на небольших расстояниях.

При последовательной передаче сообщение расчленяется на минимальные элементы. Таким образом, 1 байт посылки при передаче снова разделяется на 8 отдельных битов. Кроме данных к сообщению добавляется и другая служебная информация: как, например, признак начала и признак конца одной единицы посылаемой информации.

Скорость передачи данных — это скорость, выраженная в количестве бит, передаваемых за единицу времени, которая, по сути, соответствует частоте передаваемых битов. Частота, как известно, вычисляется по формуле: $f = 1/t$, где t — длительность передачи одного бита. Для интерфейса USB 1.1 в режиме Full-Speed скорость пере-