

ПАВЕЛ АГУРОВ



ИНТЕРФЕЙС USB

**ПРАКТИКА ИСПОЛЬЗОВАНИЯ
И ПРОГРАММИРОВАНИЯ**

USB 1.1/2.0, HID-УСТРОЙСТВА

НАПИСАНИЕ USB-ДРАЙВЕРОВ

РАБОТА В DOS, WINDOWS 98/ME/NT/2000/XP

ПРИМЕРЫ НА ЯЗЫКЕ PASCAL

ПРИМЕРЫ РЕАЛИЗАЦИИ ДЛЯ ATMEL AT89C5131

РАБОТА С МИКРОСХЕМАМИ FTDI



+CD 

**АППАРАТНЫЕ
СРЕДСТВА**

Павел Агуров

ИНТЕРФЕЙС USB

**ПРАКТИКА ИСПОЛЬЗОВАНИЯ
И ПРОГРАММИРОВАНИЯ**

Санкт-Петербург

«БХВ-Петербург»

2004

УДК 681.3.06
ББК 32.973
А27

Агуров П. В.

А27 Интерфейсы USB. Практика использования и программирования. — СПб.: БХВ-Петербург, 2004. — 576 с.: ил.

ISBN 5-94157-202-6

Изложены базовые сведения по интерфейсу USB для ПК: примеры USB-устройств и советы по их выбору, правила установки и конфигурирования устройств, методы решения возникающих проблем. Описаны кабели, разъемы, принципы питания устройств и другое аппаратное обеспечение. Приведено внутреннее устройство USB и его физическая реализация, обсуждены общие вопросы написания драйверов для операционной системы Microsoft Windows 2000/XP с примерами на языке Borland Pascal и в среде Delphi. Рассмотрен процесс создания USB-устройства: от выбора микросхем и схемотехники до написания программы микроконтроллера и WDM-драйвера. В книге содержится большое количество практических советов и примеров программ. Для удобства читателей все исходные коды приводятся на прилагаемом компакт-диске.

Для пользователей ПК, разработчиков аппаратуры и программистов

УДК 681.3.06
ББК 32.973

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Юрий Рожко</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.09.04.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 46,44.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953 Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-202-6

© Агуров П., 2004
© Оформление, издательство "БХВ-Петербург", 2004

Содержание

Введение	1
Для кого эта книга.....	2
Что вы найдете в книге.....	2
Программные требования.....	3
Аппаратные требования.....	4
О программном коде.....	4
Краткое описание глав.....	4
Обозначения.....	6
Благодарности.....	7
Часть I. Введение в USB	9
Глава 1. Что такое USB	11
1.1. История USB.....	11
1.2. Сравнение USB с другими интерфейсами.....	14
1.3. Основные понятия USB.....	16
1.3.1. Общая архитектура шины.....	16
1.3.2. Физическая и логическая архитектура шины.....	16
1.3.3. Составляющие USB.....	18
1.3.4. Свойства USB-устройств.....	18
1.3.5. Свойства хабов.....	19
1.3.6. Свойства хоста.....	20
1.4. Примеры USB-устройств.....	20
1.4.1. Мышь и клавиатура.....	21
1.4.2. Мониторы.....	21
1.4.3. Переходники USB-to-COM и USB-to-LPT.....	22
1.4.4. Сканеры.....	23
1.4.5. Модемы.....	23
1.4.6. Звуковые колонки.....	24
1.4.7. Флеш-диски.....	25
1.4.8. Хабы.....	28
1.4.9. Измерительная техника.....	28
1.4.10. Экзотические устройства.....	29
1.5. Сетевое соединение через USB.....	30
1.5.1. Конвертер USB-Ethernet.....	31
1.5.2. Прямое соединение через USB-порт.....	31
1.6. Передача данных.....	31
1.6.1. Принципы передачи данных.....	32
1.6.2. Механизм прерываний.....	32
1.6.3. Интерфейсы хост-адаптера.....	32
1.6.4. Возможность прямого доступа к памяти.....	34
1.6.5. Режимы передачи данных.....	34

1.7. Установка и конфигурирование USB-устройств	35
1.7.1. Настройки BIOS для USB	38
1.7.2. Устранение проблем.....	41
1.8. Ограничения USB	45
1.9. Если вы покупаете компьютер	46
1.9.1. HS и USB 2.0 — не одно и то же!.....	46
1.9.2. Системная плата	47
1.9.3. Корпус	48
1.9.4. USB для "старых" моделей компьютеров.....	48
1.10. Интернет-ресурсы к этой главе	49
Глава 2. Аппаратное обеспечение USB.....	51
2.1. Кабели и разъемы.....	51
2.1.1. Типы кабелей	52
2.1.2. Длина кабеля.....	53
2.1.3. Разъемы.....	53
2.2. Физический интерфейс	55
2.2.1. Кодирование данных.....	57
2.2.2. Идентификация устройств	58
2.3. Питание.....	59
2.3.1. Типы питания USB-устройств	59
2.3.2. Управление энергопотреблением.....	60
2.3.3. Вход в режим низкого энергопотребления.....	61
2.4. Интернет-ресурсы к этой главе.....	61
Часть II. Внутренняя организация USB	63
Глава 3. Внутренняя организация шины	65
3.1. Логические уровни обмена данными.....	65
3.1.1. Уровень клиентского ПО	66
3.1.2. Уровень системного драйвера USB.....	67
3.1.3. Уровень хост-контроллера интерфейса.....	68
3.1.4. Уровень шины периферийного устройства	68
3.1.5. Уровень логического USB-устройства	69
3.1.6. Функциональный уровень USB-устройства	69
3.2. Передача данных по уровням	69
3.3. Типы передач данных	71
3.4. Синхронизация при изохронной передаче.....	73
3.5. Кадры.....	77
3.6. Конечные точки	78
3.7. Каналы.....	79
3.8. Пакеты.....	81
3.8.1. Формат пакетов-маркеров IN, OUT, SETUP и PING	83
3.8.2. Формат пакета SOF	83
3.8.3. Формат пакета данных.....	84
3.8.4. Формат пакета подтверждения	84
3.8.5. Формат пакета SPLIT.....	84

3.9. Контрольная сумма.....	85
3.9.1. Алгоритм вычисления CRC.....	86
3.9.2. Программное вычисление CRC.....	87
3.10. Транзакции.....	90
3.10.1. Типы транзакций.....	91
3.10.2. Подтверждение транзакций и управление потоком.....	92
3.10.3. Протоколы транзакций.....	93
Глава 4. Внутренняя организация устройства.....	96
4.1. Запросы к USB-устройствам.....	96
4.1.1. Конфигурационный пакет.....	96
4.1.2. Стандартные запросы к устройствам.....	99
4.1.3. Дескрипторы устройства.....	105
Глава 5. Внутренняя организация хоста и хабов.....	123
5.1. Хабы.....	123
5.1.1. Взаимодействие хост-контроллера с хабом.....	126
5.1.2. Дескриптор хаба.....	127
5.1.3. Запросы хабов.....	129
5.1.4. Запрос <i>CLEAR_HUB_FEATURE</i>	130
5.1.5. Запрос <i>CLEAR_PORT_FEATURE</i>	130
5.1.6. Запрос <i>GET_BUS_STATE</i>	131
5.1.7. Запрос <i>GET_HUB_DESCRIPTOR</i>	131
5.1.8. Запрос <i>GET_HUB_STATUS</i>	131
5.1.9. Запрос <i>GET_PORT_STATUS</i>	132
5.1.10. Запрос <i>SET_HUB_DESCRIPTOR</i>	134
5.1.11. Запрос <i>SET_HUB_FEATURE</i>	134
5.1.12. Запрос <i>SET_PORT_FEATURE</i>	134
5.2. Совместная работа устройств с разными скоростями.....	135
Глава 6. USB без ПК.....	137
6.1. Разъемы OTG.....	138
6.2. Типы OTG-устройств.....	138
6.3. Дескриптор OTG-устройства.....	139
6.4. Интернет-ресурсы к этой главе.....	140
Часть III. ПРАКТИКА ПРОГРАММИРОВАНИЯ.....	141
Глава 7. Поддержка USB в Windows.....	143
7.1. Модель WDM.....	144
7.2. Взаимодействие с USB-драйвером.....	146
Глава 8. HID-устройства.....	149
8.1. Свойства HID-устройства.....	149
8.2. Порядок обмена данными с HID-устройством.....	151
8.3. Установка HID-устройства.....	152

8.4. Идентификация HID-устройства	152
8.4.1. Идентификация загрузочных устройств	153
8.4.2. Deskриптор конфигурации HID-устройства	153
8.4.3. HID-deskриптор	154
8.4.4. Deskриптор репорта	156
8.5. Структура deskриптора репорта	156
8.5.1. Структура элементов репорта	156
8.5.2. Типы элементов репорта	157
8.5.3. Примеры deskрипторов	165
8.6. Запросы к HID-устройству	168
8.6.1. Запрос <i>GET_REPORT</i>	169
8.6.2. Запрос <i>SET_REPORT</i>	169
8.6.3. Запрос <i>GET_IDLE</i>	170
8.6.4. Запрос <i>SET_IDLE</i>	170
8.6.5. Запрос <i>GET_PROTOCOL</i>	171
8.6.6. Запрос <i>SET_PROTOCOL</i>	171
8.7. Инструментальные средства	171
8.8. Взаимодействие с HID-драйвером	172
Глава 9. Введение в WDM	181
9.1. Драйверные слои	183
9.2. Символьные имена устройств	184
9.3. Основные процедуры драйвера WDM	189
9.3.1. Процедура <i>DriverEntry</i>	190
9.3.2. Процедура <i>AddDevice</i>	192
9.3.3. Процедура <i>Unload</i>	194
9.3.4. Рабочие процедуры драйвера	196
9.3.5. Обслуживание запросов IOCTL	203
9.4. Загрузка драйвера и обращение к процедурам драйвера	209
9.4.1. Процедура работы с драйвером	209
9.4.2. Регистрация драйвера	210
9.4.3. Обращение к рабочим процедурам	217
9.4.4. Хранение драйвера внутри исполняемого файла	218
9.5. Инструменты создания драйверов	220
9.5.1. NuMega Driver Studio	220
9.5.2. Jungo WinDriver	220
9.5.3. Jungo KernelDriver	220
Глава 10. Спецификация PnP для USB	221
10.1. Общие сведения о системе Plug and Play	221
10.1.1. Задачи и функции Plug and Play	221
10.1.2. Запуск процедуры PnP	222
10.1.3. Программные компоненты PnP	224
10.2. Plug and Play для USB	225
10.2.1. Конфигурирование устройств USB	226
10.2.2. Нумерация устройств USB	226
10.2.3. PnP-идентификаторы устройств USB	228

10.3. Получение списка USB-устройств	229
10.4. INF-файл	234
10.4.1. Структура INF-файла	234
10.4.2. Секция <i>Version</i>	235
10.4.3. Секция <i>Manufacturer</i>	237
10.4.4. Секция <i>DestinationDirs</i>	239
10.4.5. Секция описания модели	241
10.4.6. Секция <i>xxx.AddReg</i> и <i>xxx.DelReg</i>	242
10.4.7. Секция <i>xxx.LogConfig</i>	244
10.4.8. Секция <i>xxx.CopyFiles</i>	244
10.4.9. Секция <i>Strings</i>	245
10.4.10. Связи секций	246
10.4.11. Создание и тестирование INF-файлов	247
10.4.12. Установка устройств с помощью INF-файла	248
10.5. Ветки реестра для USB	249
Глава 11. Функции BIOS	251
11.1. Сервис BIOS IAH	251
11.1.1. Функция V101H — определение наличия PCI BIOS	252
11.1.2. Функция V102H — поиск PCI-устройства по идентификаторам устройства и производителя	253
11.1.3. Функция V103H — поиск PCI-устройства по коду класса	254
11.1.4. Функция V108H — чтение регистра конфигурации (Byte)	255
11.1.5. Функция V109H — чтение регистра конфигурации (Word)	256
11.1.6. Функция V10AH — чтение регистра конфигурации (DWord)	256
11.1.7. Функция V10BH — запись регистра конфигурации (Byte)	257
11.1.8. Функция V10CH — запись регистра конфигурации (Word)	257
11.1.9. Функция V10DH — запись регистра конфигурации (DWord)	258
11.2. Пример использования	259
Часть IV. СОЗДАНИЕ USB-УСТРОЙСТВ	283
Глава 12. USB-периферия	285
12.1. Микросхемы Atmel	286
12.1.1. Микроконтроллеры с архитектурой MSC-51	286
12.1.2. Контроллеры хабов	289
12.1.3. Микропроцессоры-хабы с ядром AVR	289
12.1.4. Другие микросхемы Atmel	290
12.2. Микросхемы Cygnal	291
12.2.1. Микропроцессоры C8051F320 и C8051F321	291
12.2.2. Другие микросхемы Cygnal	293
12.3. Микросхемы FTDI	296
12.3.1. Микросхемы FT232AM и FT232BM	297
12.3.2. Микросхемы FT245AM и FT245BM	298
12.3.3. Микросхема FT2232BM	299
12.3.4. Микросхема FT8U100AX	300

12.3.5. Отладочные комплекты и модули.....	301
12.3.6. Драйверы	302
12.3.7. Дополнительные утилиты	303
12.3.8. Другие модули.....	304
12.4. Микросхемы Intel.....	304
12.5. Микросхемы Microchip.....	308
12.6. Микросхемы Motorola	308
12.7. Микросхемы Philips.....	309
12.7.1. Микросхемы USB.....	310
12.7.2. Хабы	311
12.7.3. Другие микросхемы Philips.....	313
12.8. Микросхемы Texas Instruments	314
12.9. Микросхемы Trans Dimension.....	317
12.10. Микросхемы защиты питания	318
12.11. Интернет-ресурсы к этой главе	319
Глава 13. HID-устройство на основе Atmel AT89C5131	322
13.1. Структурная схема AT89C5131.....	322
13.2. USB-регистры AT89C5131	324
13.2.1. Регистр <i>USBCON</i>	324
13.2.2. Регистр <i>USBADDR</i>	326
13.2.3. Регистр <i>USBINT</i>	327
13.2.4. Регистр <i>USBIEN</i>	328
13.2.5. Регистр <i>UEPNUM</i>	329
13.2.6. Регистр <i>UEPCONX</i>	330
13.2.7. Регистр <i>UEPSTAX</i>	331
13.2.8. Регистр <i>UEPRST</i>	334
13.2.9. Регистр <i>UEPINT</i>	335
13.2.10. Регистр <i>UEPIEN</i>	336
13.2.11. Регистр <i>UEPDATX</i>	337
13.2.12. Регистр <i>UBYCTLX</i>	337
13.2.13. Регистр <i>UFNUML</i>	338
13.2.14. Регистр <i>UFNUMH</i>	338
13.3. Схемотехника AT89C5131	338
13.4. Инструменты программирования.....	339
13.4.1. Компилятор.....	341
13.4.2. Программатор	342
13.5. Программа для микропроцессора	349
13.5.1. Первая версия программы для AT89C5131	349
13.5.2. Добавляем строковые дескрипторы.....	369
13.5.3. Добавление конечных точек.....	374
13.5.4. Создание HID-устройства	377
13.5.5. Обмен данными с HID-устройством.....	381
13.6. Чтение репортов в Windows	388
13.7. Дополнительные функции Windows XP.....	396
13.8. Устройство с несколькими репортами.....	397

Глава 14. Создание USB-устройства на основе ATMEL AT89C5131	402
14.1. Не-NID-устройство.....	402
14.2. Создание драйвера с помощью Driver Studio.....	405
14.2.1. Несколько слов о библиотеке Driver Studio.....	407
14.2.2. Другие классы Driver Studio.....	411
14.2.3. Создание шаблона драйвера с помощью Driver Studio.....	412
14.2.4. Доработка шаблона драйвера.....	422
14.2.5. Базовые методы класса устройства.....	423
14.2.6. Реализация чтения данных.....	426
14.2.7. Установка драйвера.....	428
14.2.8. Программа чтения данных.....	429
14.2.9. Чтение данных с конечных точек других типов.....	438
14.2.10. "Чистый" USB-драйвер.....	439
Глава 15. Использование микросхем FTDI	457
15.1. Функциональная схема FT232BM.....	457
15.2. Схемотехника FT232BM.....	460
15.3. Функции D2XX.....	460
15.4. Переход от COM к USB.....	465
15.4.1. Описание схемы преобразователя.....	465
15.4.2. Установка скорости обмена.....	467
Часть V. СПРАВОЧНИК	469
Глава 16. Базовые функции Windows	471
16.1. Функции <i>CreateFile</i> и <i>CloseHandle</i> : открытие и закрытие объекта.....	471
16.1.1. Дополнительные сведения.....	472
16.1.2. Возвращаемое значение.....	472
16.1.3. Пример вызова.....	472
16.2. Функция <i>ReadFile</i> : чтение данных.....	473
16.2.1. Дополнительные сведения.....	474
16.2.2. Возвращаемое значение.....	474
16.2.3. Пример вызова.....	474
16.3. Функция <i>WriteFile</i> : передача данных.....	475
16.3.1. Дополнительные сведения.....	476
16.3.2. Возвращаемое значение.....	476
16.3.3. Пример вызова.....	476
16.4. Функция <i>ReadFileEx</i> : APC-чтение данных.....	477
16.4.1. Возвращаемое значение.....	479
16.4.2. Дополнительные сведения.....	479
16.4.3. Пример вызова.....	479
16.5. Функция <i>WriteFileEx</i> : APC-передача данных.....	480
16.5.1. Возвращаемое значение.....	481
16.5.2. Пример вызова.....	481

16.6. Функция <i>WaitForSingleObject</i> : ожидание сигнального состояния объекта.....	482
16.6.1. Возвращаемое значение.....	482
16.7. Функция <i>WaitForMultipleObjects</i> : ожидание сигнального состояния объектов.....	483
16.7.1. Возвращаемое значение.....	484
16.8. Функция <i>GetOverlappedResult</i> : результат асинхронной операции.....	484
16.8.1. Возвращаемое значение.....	485
16.9. Функция <i>DeviceIoControl</i> : прямое управление драйвером.....	485
16.9.1. Возвращаемое значение.....	487
16.10. Функция <i>QueryDosDevice</i> : получение имени устройства по его DOS-имени.....	487
16.10.1. Возвращаемое значение.....	488
16.10.2. Пример вызова.....	488
16.11. Функция <i>DefineDosDevice</i> : операции с DOS-именем устройства.....	489
16.11.1. Возвращаемое значение.....	490
16.11.2. Пример вызова.....	490
Глава 17. Функции <i>HID API</i>.....	492
17.1. Функция <i>HidD_Hello</i> : проверка библиотеки.....	492
17.2. Функция <i>HidD_GetHidGuid</i> : получение GUID.....	492
17.3. Функция <i>HidD_GetPreparedData</i> : создание описателя устройства.....	493
17.4. Функция <i>HidD_FreePreparedData</i> : освобождение описателя устройства.....	493
17.5. Функция <i>HidD_GetFeature</i> : получение FEATURE-репорта.....	494
17.6. Функция <i>HidD_SetFeature</i> : передача FEATURE-репорта.....	494
17.7. Функция <i>HidD_GetNumInputBuffers</i> : получение числа буферов.....	495
17.8. Функция <i>HidD_SetNumInputBuffers</i> : установка числа буферов.....	495
17.9. Функция <i>HidD_GetAttributes</i> : получение атрибутов устройства.....	495
17.10. Функция <i>HidD_GetManufacturerString</i> : получение строки производителя.....	496
17.11. Функция <i>HidD_GetProductString</i> : получение строки продукта.....	497
17.12. Функция <i>HidD_GetSerialNumberString</i> : получение строки серийного номера.....	497
17.13. Функция <i>HidD_GetIndexedString</i> : получение строки по индексу.....	498
17.14. Функция <i>HidD_GetInputReport</i> : получение INPUT-репорта.....	498
17.15. Функция <i>HidD_SetOutputReport</i> : передача OUTPUT-репорта.....	499
17.16. Функция <i>HidP_GetCaps</i> : получение свойств устройства.....	499
17.17. Функция <i>HidP_MaxDataListLength</i> : получение размеров репортов.....	500
Глава 18. Хост-контроллер UCN.....	502
18.1. Регистры управления хост-контроллером.....	502
18.1.1. Регистр команды USB (<i>USBCMD</i>).....	504
18.1.2. Регистр состояния USB (<i>USBSTS</i>).....	506
18.1.3. Регистр управления прерываниями (<i>USBINTR</i>).....	506
18.1.4. Регистр номера кадра (<i>FRNUM</i>).....	507

18.1.5. Регистр базового адреса кадра (<i>FLBASEADD</i>)	508
18.1.6. Регистр модификатора начала кадра (<i>SOFMOD</i>)	508
18.1.7. Регистр состояния и управления порта (<i>PORTSC</i>)	509
18.2. Структуры данных хост-контроллера UCH	510
18.2.1. Список кадров	510
18.2.2. Дескриптор передачи	511
18.2.3. Заголовок очереди	514
18.3. Обработка списка дескрипторов UCH	516
Глава 19. Инструменты	518
19.1. Средства Microsoft Visual Studio	518
19.1.1. Depends	518
19.1.2. Error Lookup	518
19.1.3. GuidGen	518
19.2. Средства Microsoft DDK	520
19.2.1. DeviceTree	520
19.2.2. DevCon	521
19.2.3. ChkInf и GenInf	526
19.3. Средства CompuWare Corporation	527
19.3.1. Monitor	527
19.3.2. SymLink	527
19.3.3. EzDriverInstaller	527
19.3.4. WdmSniff	527
19.4. Средства SysInternals	528
19.4.1. WinObj	528
19.5. Средства USB Forum	531
19.5.1. HID Descriptor Tool	531
19.6. Средства HDD Software	533
19.7. Средства Sourceforge	533
ПРИЛОЖЕНИЯ	535
Приложение 1. Дополнительные функции	537
Приложение 2. Таблица идентификаторов языков (LangID)	539
Приложение 3. Таблица кодов производителей (Vendor ID, Device ID)	543
Приложение 4. Описание компакт-диска	546
Литература	548
Предметный указатель	549

Введение

Последовательные интерфейсы интересны тем, что позволяют объединить множество устройств, используя всего одну (или две) пары проводов. До 1996 года последовательные интерфейсы персонального компьютера были представлены коммуникационным портом, работающим согласно спецификации RS-232. Хотя RS-232 сохраняет все преимущества последовательной связи, она имеет и ряд недостатков. Самыми существенными из них являются плохая помехозащищенность и отсутствие гальванической развязки. Первое мешает использованию высоких скоростей обмена, а второе — "горячему" подключению устройств. Кроме того, стандарт RS-232 подразумевает подключение только одного устройства к одному последовательному порту.

Еще один недостаток внешних интерфейсов персонального компьютера — строгая предопределенность их использования. Так, COM-порт используется для подключения "мыши" или модема, LPT-порт — для подключения принтера (ну, еще сканера или плоттера), порт клавиатуры предназначен для подключения клавиатуры, и т. д. Кроме необходимости иметь в персональном компьютере множество различных, но, чаще всего, неиспользуемых, разъемов, такое многообразие несет и другие проблемы: для каждого интерфейса необходимо выделять аппаратное прерывание (IRQ), "пустые" разъемы занимают место, что особенно актуально для ноутбуков.

В 1996 году была опубликована версия 1.0 нового интерфейса, названного USB (Universal Serial Bus, универсальная последовательная шина), а осенью 1998 — спецификация 1.1, исправляющая проблемы, обнаруженные в первой редакции. В 2000 году была опубликована версия 2.0, в которой предусматривалось 40-кратное повышение пропускной способности шины.

Шина USB ориентирована на устройства, подключаемые к PC. Изохронные передачи USB позволяют передавать огромные потоки данных, такие как аудиосигналы, а шина USB 2.0 позволяет передавать и видеосигналы. Спецификация USB подразумевает прозрачное подключение устройств к шине и позволяет иметь несколько устройств на одном порту.

Для кого эта книга

Эта книга для вас, если:

- протирая пыль с компьютера, вы обнаружили два непонятных разъема и хотите узнать, что это такое;
- вам надо объяснить в бухгалтерии, что "все то же самое, но без USB" дешевле не будет;
- вам надо подключить к компьютеру два принтера, сканер и цифровой фотоаппарат одновременно;
- подключив к первому порту USB — мышь, а ко второму USB — клавиатуру, вы хотите понять, куда же подключать обещанные в USB-спецификации 127 устройств;
- ваш шеф утверждает, что изготовленный вами прибор не работает, т. к. его нельзя подключить к его новому ноутбуку, в котором нет COM-порта;
- вам обидно, что Windows находит новые устройства, но не говорит об этом вашей программе;
- вам хочется узнать, что нужно Windows, чтобы ваше устройство было названо по имени;
- скорость COM-порта вас не удовлетворяет, поэтому хочется использовать USB, но времени на переделку нет.

Другими словами, мы адресуем эту книгу тем, кто хочет использовать в своей работе современный протокол USB. Книга будет интересна тем, кто занимается или собирается заниматься программированием микроконтроллеров. Книга будет полезна разработчикам Windows-драйверов, а также тем, кто хочет повысить свой профессиональный уровень.

Что вы найдете в книге

В этой книге вы сможете найти ответы на некоторые вопросы, касающиеся разработки и программирования USB-устройств.

- Что такое USB-интерфейс и зачем он нужен:
 - вы хотите узнать, что такое USB? Эта книга ориентируется на практическое использование USB-интерфейса и содержит все сведения, необходимые для разработки своего проекта.
- В каких случаях нужен USB-интерфейс, и какие преимущества он дает:
 - вы используете COM-интерфейс и считаете, что этого достаточно? Прочитайте эту книгу, — возможно, вы измените свое мнение.

- Как USB-устройства взаимодействуют с компьютером:
 - вы хотите понять, как работает USB-шина? что нужно сделать, чтобы ваше устройство было опознано системой? Ответы на эти вопросы содержат главы, посвященные архитектуре USB-шины и стандарту Plug and Play.
- Как разработать приложение, взаимодействующее с USB-контроллером:
 - вы решили использовать USB в своем проекте, вам потребуются сведения о необходимых для работы функциях. Эта книга подробно и просто расскажет обо всех функциях, предоставляемых операционной системой Microsoft Windows. Справочная часть книги позволит быстро найти нужную информацию.
- Как выбрать микросхему USB-приемопередатчика при разработке своего контроллера:
 - существует множество микросхем для организации USB-интерфейса. Выбрать нужную достаточно сложно. Мы предоставим вам краткий обзор основных микросхем приемопередатчиков, присутствующих на российском рынке.
- Как создать USB-устройство на основе микропроцессора 8051:
 - мы предлагаем вместе пройти полный путь создания USB-устройства на основе микропроцессора 8051. После прочтения этих глав использование USB-интерфейса станет простым делом, ничуть не сложнее использования обычного COM-порта.

Программные требования

Все программы мы будем реализовывать на языках Borland Delphi 6 и Visual Studio 6. Так как мы не будем использовать никаких специфических функций, присущих именно этим версиям языков, то все примеры могут быть скомпилированы в других версиях практически без модификации.

Версия Windows должна быть или Windows 2000 или Windows XP. Желательно при этом наличие всех доступных пакетов обновлений (service pack). Возможно использование Windows 98, но многие примеры, связанные с написанием драйверов, могут не работать.

Для компиляции драйверов, приведенных в книге, потребуется Windows 2000 DDK или Windows XP DDK, в соответствии с вашей версией Windows.

В качестве дополнительного источника информации мы рекомендуем установить MSDN. Список полезных для работы утилит и программ приводится в справочной части книги, в *главе 19*.

На компакт-диске содержатся полные исходные тексты и скомпилированные модули.

Аппаратные требования

Достаточно обычного домашнего компьютера, на котором компиляция программы в Delphi и Visual Studio занимает приемлемое для вас время.

Установка Delphi 6 потребует примерно 300 Мбайт на жестком диске, установка Visual Studio — 240 Мбайт, MSDN — 1,5 Гбайт, Windows DDK — 700 Мбайт.

Для тестирования программ необходимо иметь одно или несколько USB-устройств. Для создания своих устройств будет необходим соответствующий инструментарий.

О программном коде

Книга содержит полные исходные коды всех программ, однако многие листинги содержат только изменения кода относительно предыдущего листинга. Такое сокращение позволяет не только экономить место, но и улучшить понимание кода, делая акцент только на новой функциональности. Код на компакт-диске содержит тексты без сокращений.

В программах на Delphi мы не приводим код самого проекта (DPR-файл) и код формы (DFM-файлы). Все исходные коды можно найти на компакт-диске, и, мы думаем, желающих набирать модули форм "с листа" найдется немного.

Еще раз повторим, что такие сокращения не означают отсутствие возможности скомпилировать и попробовать приведенные примеры на своем компьютере. Все исходные коды и необходимые модули находятся на компакт-диске.

Краткое описание глав

Первая часть книги содержит главы 1—2. В них приводится спецификация USB, описание USB-интерфейса и его составляющие. В этих главах мы не будем глубоко заглядывать внутрь USB: регистры контроллера, команды и тонкости аппаратной организации мы оставим на потом:

- глава 1 ("Что такое USB") содержит описание USB-интерфейса с точки зрения пользователя и потенциального покупателя компьютера; примеры USB-устройств и тонкости, которые необходимо знать при их покупке и покупке компьютера; правила установки и конфигурирования устройств, основные проблемы, возникающие при этом, и методы их решения. Кроме того, глава содержит определение основных понятий USB: хабы, концентраторы и т. д.;

- ❑ глава 2 ("Аппаратное обеспечение USB") содержит небольшое описание аппаратной части USB. До схемотехники и микросхем приемопередатчиков мы доберемся в следующих главах, а в этой описываются кабели, разъемы, принципы питания устройств и другие сведения, необходимые не только разработчику аппаратуры, но и обычному пользователю персонального компьютера для грамотного использования интерфейса USB.

Вторая часть книги содержит главы 3—6, которые описывают внутреннюю организацию USB:

- ❑ глава 3 ("Внутренняя организация шины USB") описывает внутреннее устройство шины, организацию и типы передач данных, методы синхронизации данных и правила вычисления контрольных сумм;
- ❑ глава 4 ("Внутренняя организация устройства") описывает структуры и запросы, используемые для обращения к USB-устройствам, дескрипторы устройств и функции их получения;
- ❑ глава 5 ("Внутренняя организация хоста и хабов") описывает структуры и запросы, используемые для взаимодействия хоста, хабов и устройств. Кроме того, глава содержит информацию об организации одновременной работы устройств с разными скоростями;
- ❑ глава 6 ("USB без ПК") описывает расширение спецификации USB, позволяющей соединять USB-устройства между собой без персонального компьютера.

Третья часть книги содержит главы 7—10, которые описывают реализацию поддержки USB-интерфейса в операционной среде Microsoft Windows:

- ❑ глава 7 ("Поддержка USB в Windows") содержит общие сведения о драйверной модели и методах взаимодействия с драйверами;
- ❑ глава 8 ("HID-устройства") содержит описание одного из классов USB-устройств, называемых HID;
- ❑ глава 9 ("Введение в WDM") дает общие сведения о модели драйверов Windows 2000/XP;
- ❑ глава 10 ("Спецификация PnP для USB") содержит описание спецификации Plug and Play для USB, структуры и функции Windows для системы PnP, а также общее описание структуры INF-файла;
- ❑ глава 11 ("Функции BIOS") содержит описание сервисов BIOS, используемых для работы с USB в DOS. Эта информация будет полезна для программистов, использующих промышленные контроллеры с установленным DOS или его клоном.

Четвертая часть книги содержит главы 12—14, которые описывают процесс создания USB-устройства:

- ❑ глава 12 ("USB-периферия") дает обзор микросхем USB-приемопередатчиков. Акцент делается на микросхемах, доступных на российском рынке;

- глава 13 ("HID-устройство на основе ATMEL AT89C5131") описывает схемотехнику и процесс разработки HID-устройства на основе микропроцессора AT89C5131 и программы, использующей HID-функции;
- глава 14 ("Создание USB-устройства на основе ATMEL AT89C5131") рассматривает процесс разработки USB-устройства на основе микропроцессора AT89C5131 и драйвера для Microsoft Windows;
- глава 15 ("Использование микросхемы FT232BM") представляет схемотехнику и процесс разработки USB-устройства на основе преобразователя FT232BM, а также содержит описание схемы преобразователя из COM-интерфейса в USB.

Часть пятая включает *главы 16—19*, в которых содержатся справочные материалы.

Обозначения

При описании некоторых данных мы будем пользоваться битовым представлением, заключая число разрядов каждого поля в квадратные скобки. Например:

- [5] поле А;
- [2] поле Б.

Такое описание означает, что поле А содержит 5 битов, а поле Б — 2 бита. Еще одно представление битовых полей — указание конкретных диапазонов битов, с помощью знака ":", например:

- [16:5] зарезервированы;
- [4:0] индекс.

Такое описание означает, что биты с 16 по 5 включительно зарезервированы, а биты с 4 по 0 включительно представляют собой индекс. Отличить первое описание от второго обычно легко по контексту изложения.

При написании чисел мы будем придерживаться следующих правил:

- шестнадцатеричные числа будут иметь префикс "\$", например "\$45";
- шестнадцатеричные числа могут иметь префикс "0x" или постфикс "H", если того требует контекст изложения или формат строки, например, "INT 3FH";
- битовые последовательности заключены в угловые скобки, например, "<0010>", либо, при написании двоичного числа, обозначены символом b в конце, например, 10101111b.

Для описания версий протоколов, структур и т. д. будет использоваться специальный тип чисел BCD (Binary-Coded Decimal). Такие числа записываются в шестнадцатеричном виде 0xJJMN для обозначения версии JJ.M.N, т. е.

JJ обозначает старший номер версии, M — младший номер версии и N — номер подверсии. Например, версия 2.1.3 будет представлена числом 0x213, а версия 2.0 будет записана числом 0x0200.

При описании регистров мы будем пользоваться следующими обозначениями режимов доступа:

- RO (read only) — регистр только для чтения, запись в него невозможна;
- WO (write only) — регистр только для записи, чтение значения невозможно;
- R/W (read/write) — возможны и чтение, и запись значения;
- R/W2 (read/write word) — возможны чтение и запись слова;
- R/WC (read/write clear) — разрешено как чтение, так и считывание значения, однако при записи в некоторый разряд регистра приводит к его сбросу в ноль.

При необходимости указания версии Windows мы будем использовать следующие сокращения:

- Windows 9x будет означать семейство Windows 95/98/ME;
- Windows NT в общем случае будет обозначать семейство Windows NT/2000/XP;
- при необходимости указания конкретной версии мы будем писать номер этой версии без сокращений, например, Windows NT4 или Windows 98.

Благодарности

В создании этой книги прямо или косвенно участвовало очень много людей. Прежде всего, хочется поблагодарить заместителя главного редактора издательства Евгения Рыбакова, автора идеи создания этой книги.

Нельзя не отметить тех, чьи материалы помогли наполнить книгу полезной и нужной информацией.

Информация о HID-устройствах (*глава 8*), идентификации устройств (*глава 2*) и некоторых микросхемах (*глава 12*) предоставлена Константином Вовк, компания KS Labs (www.is.svitonline.com/vks).

Автор выражает благодарность к. т. н. Малыгину И. В., Лысенко А. А., Назмутдинову Р. Ф. за предоставленные материалы о микросхемах FTDI (*глава 14*), опубликованные на сайте Института радиотехники (www.institute-rt.ru).

Материалы *главы 1* были бы неинтересны без иллюстраций, предоставленных компьютерным супермаркетом NIX (www.nix.ru).

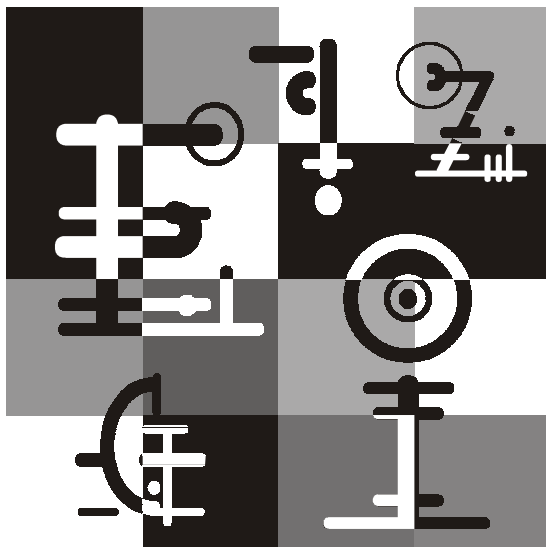
В составлении обзора микросхем автору помогали Сергей Воробьев (ООО "Автоматика-М", vesta.pvasoft.com), Геннадий Курзаев, Сергей Гудков, Игорь Кривченко (компания ООО "Эфо", www.efo.ru).

Спасибо Эрику Тинлоту (Eric Tinlot) — инженеру службы поддержки Atmel — за помощь при разработке примеров для микросхем Atmel (www.atmel.com).

При написании программ к *главе 11* существенную помощь оказывал Максим Локтюхин (компания Intel).

Разработку схем для микропроцессора AT89C5131 производил Сергей Малов. Без его участия и помощи книга лишилась бы практической части.

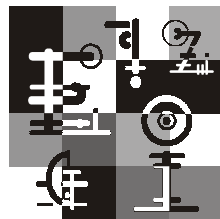
Автор благодарит всех друзей и родных, которые терпели его в процессе написания книги, а также коллектив издательства "БХВ-Петербург".



ЧАСТЬ I

ВВЕДЕНИЕ В USB

Глава 1



Что такое USB

USB — Unusable Serial Interface
(непригодный последовательный интерфейс).

Расшифровка аббревиатуры времен 1998 года

1.1. История USB

Увеличение числа устройств, подключаемых к персональному компьютеру, и, соответственно, развитие внешних интерфейсов привело к довольно неприятной ситуации: с одной стороны, компьютер должен иметь множество различных разъемов, а с другой — большая часть из них не используется. Такая ситуация определяется историческим развитием интерфейсов ПК — каждый интерфейс имел свой специализированный разъем. Например, к последовательному порту можно подключить мышь или модем, к параллельному — принтер или сканер, для клавиатуры стало необходимо иметь два порта — старый клавиатурный и PS/2 и т. д. Более того, к одному порту можно подключить только одно устройство (если не считать подключение "прозрачных" ключей защиты, но это, скорее, исключение). Кроме этой проблемы, многочисленность разнообразных подключений добавляет и другие "радости":

- ❑ практически для каждого из устройств необходимо выделение аппаратного прерывания (IRQ);
- ❑ большая часть устройств требует наличия внешнего блока питания;
- ❑ каждое устройство имеет свой, придуманный разработчиком, протокол обмена, многократно увеличивая необходимое количество драйверов, как в памяти, так и в инсталляции операционной системы;
- ❑ конфигурирование огромного числа устройств, многие из которых не поддерживают спецификации Plug and Play, — практически невыполнимая работа для обычного пользователя;
- ❑ огромное число разнокалиберных шлейфов, тянущихся от компьютера, превращает его перестановку в сложную проблему.

Естественно, что производители компьютерного "железа" задумались о создании единого и универсального интерфейса. В начале 1996 года была опубликована версия 1.0 нового интерфейса, названного USB (Universal Serial Bus, универсальная последовательная шина), а осенью 1998 — спецификация 1.1, исправляющая проблемы, обнаруженные в первой редакции. Весной 2000 года была опубликована версия 2.0, в которой предусматривалось 40-кратное повышение пропускной способности шины. Так, спецификации 1.0 и 1.1 обеспечивают работу на скоростях 12 Мбит/с и 1,5 Мбит/с, а спецификация 2.0 — на скорости 480 Мбит/с. При этом предусматривается обратная совместимость USB 2.0 с USB 1.x, т. е. "старые" USB 1.x устройства будут работать с USB 2.0 контроллерами, правда, на скорости 12 Мбит/с. Скорость 480 Мбит/с достигается только при одновременном использовании USB 2.0 контроллера и USB 2.0 периферии.

Изначально в группу разработчиков входили компании Compaq, DEC, IBM, Intel, Microsoft, NEC и Northern Telecom, а затем количество заинтересованных участников стало расширяться. Шина USB разрабатывалась для обеспечения механизма взаимодействия компьютерных и телефонных систем (CTI, Computer Telephony Integration), однако вскоре члены комитета разработки поняли, что USB может удовлетворить потребности многих приложений и все сферы компьютерной телефонии.

Разработчики шины ориентировались на создание интерфейса, обладающего следующими свойствами:

- легко реализуемое расширение периферии ПК;
- дешевое решение, позволяющее передавать данные со скоростью до 12 Мбит/с (480 Мбит/с для USB 2.0);
- полная поддержка в реальном времени голосовых, аудио- и видеопотоков;
- гибкость протокола смешанной передачи изохронных данных и асинхронных сообщений;
- интеграция с выпускаемыми устройствами;
- охват всевозможных конфигураций и конструкций ПК;
- обеспечение стандартного интерфейса, способного быстро завоевать рынок;
- создание новых классов устройств, расширяющих ПК.

Спецификация USB определяет следующие функциональные возможности интерфейса:

- простота использования для конечного пользователя:
 - простота кабельной системы и подключений;
 - скрытие подробностей электрического подключения от конечного пользователя;

- самоидентифицирующиеся устройства с автоматическим конфигурированием;
 - динамическое подключение и переконфигурирование периферийных устройств;
- широкие возможности работы:
- пропускная способность от нескольких Кбайт/с до нескольких Мбайт/с;
 - поддержка одновременно как изохронной, так и асинхронной передачи данных;
 - поддержка одновременных операций со многими устройствами (multiple connections);
 - поддержка до 127 устройств на шине;
 - передача разнообразных потоков данных и сообщений;
 - поддержка составных устройств (т. е. периферийное устройство, выполняющее несколько функций);
 - низкие накладные расходы передачи данных;
- равномерная пропускная способность:
- гарантированная пропускная способность и низкие задержки голосовых и аудиоданных;
 - возможность использования всей полосы пропускания;
- гибкость:
- поддержка разных размеров пакетов, которые позволяют настраивать функции буферизации устройств;
 - настраиваемое соотношение размера пакета и задержки данных;
 - управление потоком (flow control) данных на уровне протокола;
- надежность:
- контроль ошибок и восстановление на уровне протокола;
 - динамическое добавление и удаление устройств прозрачно для конечного пользователя;
 - поддержка идентификации неисправных устройств;
 - исключение неправильного соединения устройств;
- выгода для разработчиков:
- простота реализации и внедрения;
 - объединение с архитектурой Plug and Play;

- ❑ дешевая реализация:
 - дешевые каналы со скоростью работы до 1,5 Мбайт/с;
 - оптимизация для интеграции с периферией;
 - применимость для реализации дешевой периферии;
 - дешевые кабели и разъемы;
 - использование выгодных товарных технологий;
- ❑ возможность простого обновления.

Практически все поставленные задачи были решены, и весной 1997 года стали появляться компьютеры, оборудованные разъемами для подключения USB-устройств. Иконкой, показанной на рис. 1.1, официально обозначается шина USB, как в Windows, так и на USB-разъемах (подробнее о логотипах и некоторых тонкостях, которые с ними связаны, мы расскажем в *разд. 1.9*).

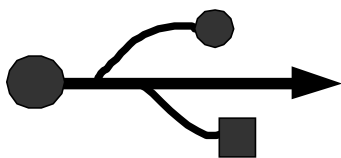


Рис. 1.1. Иконка USB-шины

В феврале 2004 года корпорация Intel совместно с Agere, Systems, HP, Microsoft Corporation, NEC, Philips Semiconductors и Samsung Electronics объявила о создании группы Wireless USB Promoter Group (группа продвижения беспроводного USB). В задачу консорциума входит продвижение первой высокоскоростной технологии для беспроводного подключения внешних устройств Wireless USB на скорости 480 Мбит/с (что сопоставимо с характеристикой стандарта USB 2.0) с дальностью действия при низком энергопотреблении до 10 метров.

1.2. Сравнение USB с другими интерфейсами

В табл. 1.1 приведено сравнение интерфейса USB с другими интерфейсами персонального компьютера. Видно, что достойной альтернативы USB не существует (пожалуй, кроме "изначального" конкурента — FireWire, но у этой шины принципиально другая система соединения¹). Интерфейсы,

¹ FireWire реализует систему "мастер—мастер", а USB — "мастер—ведомый".

сравнимые с USB по скорости обмена, требуют специальных преобразователей. Интерфейсы, не требующие дополнительных элементов, либо низкоскоростные, либо узконаправленные. Кроме того, к несомненным плюсам USB относятся организация помехозащищенности на уровне аппаратного и шинного протоколов и "встроенная" поддержка Plug and Play, а также отсутствие дополнительных элементов для подключения устройств (как, например, терминаторы для SCSI-интерфейса). Пожалуй, единственным минусом можно считать довольно короткое кабельное соединение, но следует помнить, что шина USB разрабатывалась как шина для домашних устройств, стоящих на столе, и дальние соединения не закладывались в нее изначально.

Таблица 1.1. Сравнение USB с другими интерфейсами

Интерфейс	Число устройств / Число проводов / Длина провода (м)	Скорость	Использование
Последовательные			
USB 2.0	127/3/10	1,5 Мбит/с, 12 Мбит/с, 480 Мбит/с	Любые устройства с USB 1.x/2.0 (USB-порт)
RS-232	1/6/50-100	115,2 Кбит/с	Модем, мышь, ключи за- щиты (COM-порт)
RS-485	32/2/4000	10 Мбит/с	Промышленные устройства (COM-порт через преобра- зователь)
FireWire (IEEE-1394)	64/3/15	400 Мбит/с	Видеоданные, дисковые массивы (FireWire-порт)
Ethernet	1024/3/1600	10 Мбит/с 100 Мбит/с 1 Гбит/с	Сетевые соединения ПК (сетевая карта)
Токовая петля			
MIDI	1/3/50	31,5 Кбит/с	Музыкальные устройства
Параллельные			
LPT	1/9/10-30	От 800 Кбит/с до 16 Мбит/с	Принтеры, сканеры, диско- вые устройства

1.3. Основные понятия USB

В этом разделе мы приведем основные понятия, которые используются при работе с шиной USB.

1.3.1. Общая архитектура шины

Обычная архитектура шины USB подразумевает подключение одного или нескольких *USB-устройств* к компьютеру (рис. 1.2). Компьютер в такой конфигурации является главным управляющим устройством и называется *хостом*. Подключение устройств к хосту производится с помощью *кабелей*. Для соединения компьютера и устройства используется *хаб*. Компьютер имеет встроенный хаб, называемый *корневым хабом*.

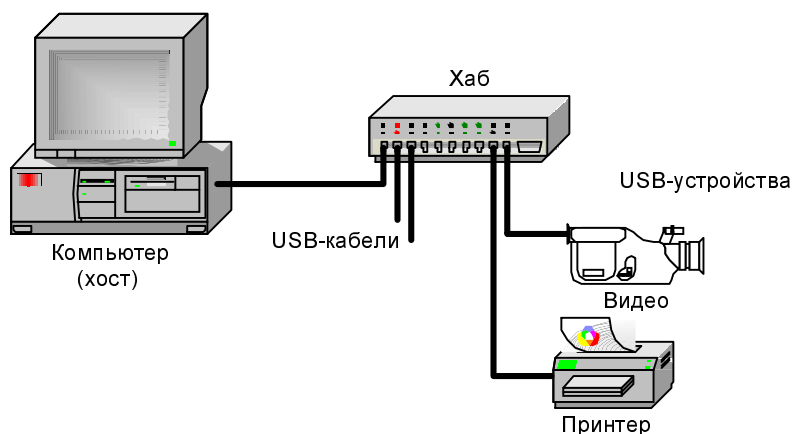


Рис. 1.2. Обычная архитектура USB

1.3.2. Физическая и логическая архитектура шины

Физическая архитектура USB-шины определяется следующими правилами (рис. 1.3):

- устройства подключаются к хосту;
- физическое соединение устройств между собой осуществляется по топологии многоярусной звезды, вершиной которой является корневой хаб;
- центром каждой звезды является хаб;

- ❑ каждый кабельный сегмент соединяет между собой две точки: хост с хабом или функцией (см. далее), хаб с функцией или другим хабом;
- ❑ к каждому порту хаба может подключаться периферийное устройство или другой хаб, при этом допускается до 5 уровней каскадирования хабов, не считая корневого.

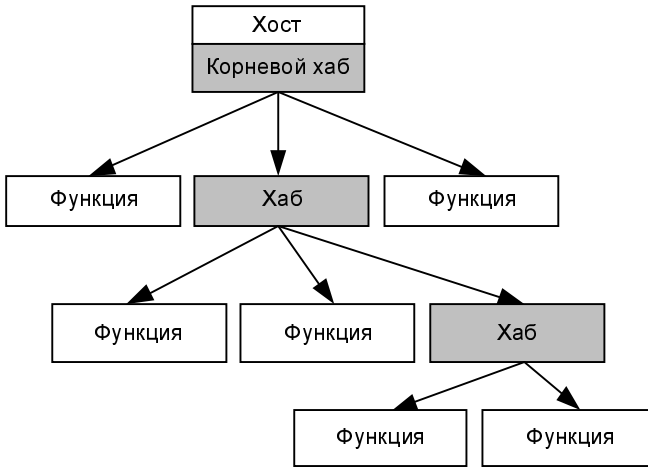


Рис. 1.3. Физическая архитектура USB

Детали физической архитектуры скрыты от прикладных программ в системном ПО, поэтому *логическая архитектура* выглядит как обычная звезда, центром которой является прикладное ПО, а вершинами — набор *конечных точек* (рис. 1.4).

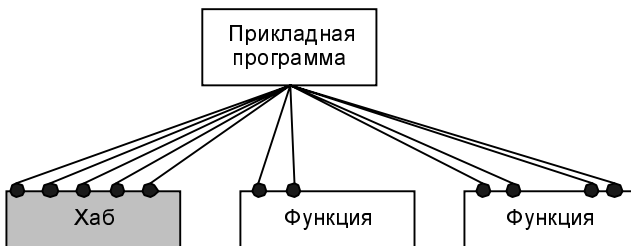


Рис. 1.4. Логическая архитектура USB

Прикладная программа ведет обмен информацией с каждой конечной точкой.

1.3.3. Составляющие USB

Шина USB состоит из следующих элементов.

Хост-контроллер (Host Controller) — это главный контроллер, который входит в состав системного блока компьютера и управляет работой всех устройств на шине USB. Для краткости мы будем писать просто "хост". На шине USB допускается наличие только одного хоста. Системный блок персонального компьютера содержит один или несколько хостов, каждый из которых управляет отдельной шиной USB. В *главе 6* мы рассмотрим одно исключение — соединение двух USB-устройств без персонального компьютера.

Устройство (Device) может представлять собой хаб, функцию или их комбинацию (Compound Device). Примеры USB-устройств приведены в *разд. 1.4*.

Порт (Port) — точка подключения.

Хаб (Hub, другое название — *концентратор*) — устройство, которое обеспечивает дополнительные порты на шине USB. Другими словами, хаб преобразует один порт (*восходящий порт*, Upstream Port) во множество портов (*нисходящие порты*, Downstream Ports). Архитектура допускает соединение нескольких хабов (не более 5). Хаб распознает подключение и отключение устройств к портам и может управлять подачей питания на порты. Каждый из портов может быть разрешен или запрещен и сконфигурирован на полную или ограниченную скорость обмена. Хаб обеспечивает изоляцию сегментов с низкой скоростью от высокоскоростных. Хаб может ограничивать ток, потребляемый каждым портом.

Корневой хаб (Root Hub) — это хаб, входящий в состав хоста.

Функция (Function) — это периферийное устройство (ПУ) или отдельный блок периферийного устройства, способный передавать и принимать информацию по шине USB. Каждая функция предоставляет конфигурационную информацию, описывающую возможности ПУ и требования к ресурсам. Перед использованием функция должна быть сконфигурирована хостом — ей должна быть выделена полоса в канале и выбраны опции конфигурации.

Логическое устройство (logical device) USB представляет собой набор конечных точек.

1.3.4. Свойства USB-устройств

Спецификация USB достаточно жестко определяет набор свойств, которые должно поддерживать любое USB-устройство:

- **адресация** — устройство должно отзываться на назначенный ему уникальный адрес, и только на него;

- ❑ **конфигурирование** — после включения или сброса устройство должно предоставлять нулевой адрес для возможности конфигурирования его портов;
- ❑ **передача данных** — устройство имеет набор конечных точек для обмена данными с хостом. Для конечных точек, допускающих разные типы передач, после конфигурирования доступен только один из них;
- ❑ **управление энергопотреблением** — любое устройство при подключении не должно потреблять от шины ток, превышающий 100 мА. При конфигурировании устройство заявляет свои потребности тока, но не более 500 мА. Если хаб не может обеспечить устройству заявленный ток, устройство не будет использоваться;
- ❑ **приостановка** — устройство USB должно поддерживать *приостановку* (Suspended Mode), при которой его потребляемый ток не превышает 500 мкА. Устройство должно автоматически приостанавливаться при прекращении активности шины;
- ❑ **удаленное пробуждение** — возможность *удаленного пробуждения* (Remote Wakeup) позволяет приостановленному устройству подать сигнал хосту, который тоже может находиться в приостановленном состоянии. Возможность удаленного пробуждения описывается в конфигурации устройства. При конфигурировании эта функция может быть запрещена.

1.3.5. Свойства хабов

Хаб выполняет коммутацию сигналов и выдачу питающего напряжения, а также отслеживает состояние подключенных к нему устройств, уведомляя хост об изменениях. Хаб состоит из двух частей — *контроллера* (Hub Controller) и *повторителя* (Hub Repeater).

Контроллер содержит регистры для взаимодействия с хостом. Доступ к регистрам осуществляется по специфическим командам обращения к хабу. Команды позволяют конфигурировать хаб, управлять нисходящими портами и опрашивать их состояние.

Повторитель представляет собой управляемый ключ, соединяющий выходной порт со входным. Он имеет средства сброса и приостановки передачи сигналов.

Нисходящие порты хабов могут находиться в следующих состояниях:

- ❑ **Питание отключено** (Powered off) — на порт не подается питание (возможно только для хабов, коммутирующих питание). Выходные буферы переводятся в высокоимпедансное состояние, входные сигналы игнорируются;
- ❑ **Отсоединен** (Disconnected) — порт не передает сигналы ни в одном направлении, но способен обнаружить подключение устройства;