ПАВЕЛ АГУРОВ

bhv®

ПРАКТИКА ПРОГРАММИРОВАНИЯ USB

USB 1.1/2.0

КЛАССЫ HID, CDC
И ПРИМЕРЫ ИХ РЕАЛИЗАЦИИ

ФУНКЦИИ РАБОТЫ С USB
ДЛЯ WINDOWS 98/NT/2000/XP

РАЗРАБОТКА USB-ДРАЙВЕРОВ
ДЛЯ WINDOWS 2000/XP

ПРИМЕРЫ ПРОГРАММ
НА ЯЗЫКАХ DELPHI, C, C#

СПЕЦИАЛЬНЫЕ ФУНКЦИИ
RAW INPUT, DIRECT INPUT,
SETUP API

PRO
ПРОФЕССИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

+CD

УДК 681.3.06
ББК 32.973.26-018.1
А27

Агуров П. В.

А27 Практика программирования USB. — СПб.: БХВ-Петербург, 2006. — 624 с.: ил.

ISBN 978-5-94157-851-1

В книге собрана информация, необходимая для создания USB-устройств и драйверов для операционной системы Microsoft Windows 2000/XP. Рассмотрен процесс создания USB-устройства: от написания программы микроконтроллера (примеры реализованы для микропроцессора AT89C5131) до разработки собственного WDM-драйвера. Содержится описание специальных классов устройств: HID-класс, позволяющий обойтись без разработки драйвера, и класс CDC, позволяющий работать с USB как с обычным COM-портом. Рассмотрено использование функций Raw Input, Direct Input и Setup API, содержится большое количество практических советов и примеров программ на языках Delphi, C и C#. Для удобства читателей все исходные коды приводятся на прилагаемом компакт-диске.

Для программистов и разработчиков аппаратуры

УДК 681.3.06
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Константин Костенко*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Наталия Першакова*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.03.06.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 50,31.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

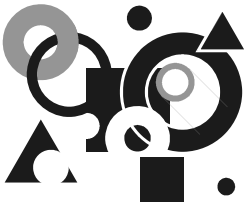
Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04
от 11.11.2004 г. выдано Федеральной службой по надзору
в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-94157-851-1

Оглавление

Глава 1



Спецификация USB

Подробное описание USB [4] мы рассматривать не будем, а приведем лишь минимум сведений, необходимых для практической работы.

1.1. Что такое USB и зачем это надо

Шина USB (Universal Serial Bus, универсальная последовательная шина) появилась в начале 1996 года как попытка решения проблемы множественности интерфейсов. К тому времени персональные компьютеры (ПК) были оснащены большим количеством разнообразных внешних интерфейсов, полезных и необходимых, но обладающих одним недостатком: все они требовали своего специального разъема и, чаще всего, выделенного аппаратного прерывания (IRQ, Interrupt ReQuest).

Первая спецификация (версия 1.0) USB была опубликована в начале 1996 года, а осенью 1998 года появилась 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, спецификация 1.0 и 1.1 обеспечивает 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.

Разработчики шины ориентировались на создание интерфейса, обладающего следующими свойствами:

- легкореализуемое 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 Кбит/с до нескольких Мбит/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 (периферийных устройств, выполняющих 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 Мбит/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.

1.1.1. Общая архитектура USB

Обычная архитектура USB подразумевает подключение одного или нескольких *USB-устройств* к компьютеру (рис. 1.1), который в такой конфигурации является главным управляющим устройством и называется *хостом*. Подключение USB-устройств к хосту производится с помощью *кабелей*. Для соединения компьютера и USB-устройства используется *хаб*. Компьютер имеет встроенный хаб, называемый *корневым хабом*.

1.1.2. Физическая и логическая архитектура USB

Физическая архитектура USB определяется следующими правилами (рис. 1.2):

- 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 USB-устройство или другой хаб, при этом допускается до 5 уровней каскадирования хабов, не считая корневого.

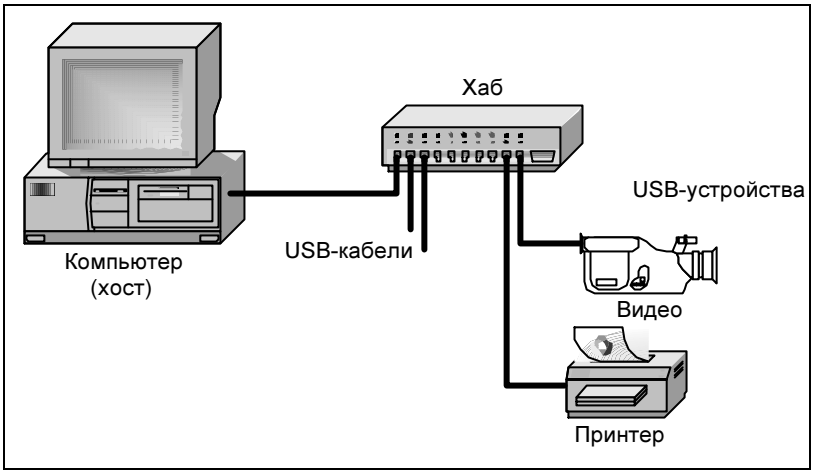


Рис. 1.1. Обычная архитектура USB

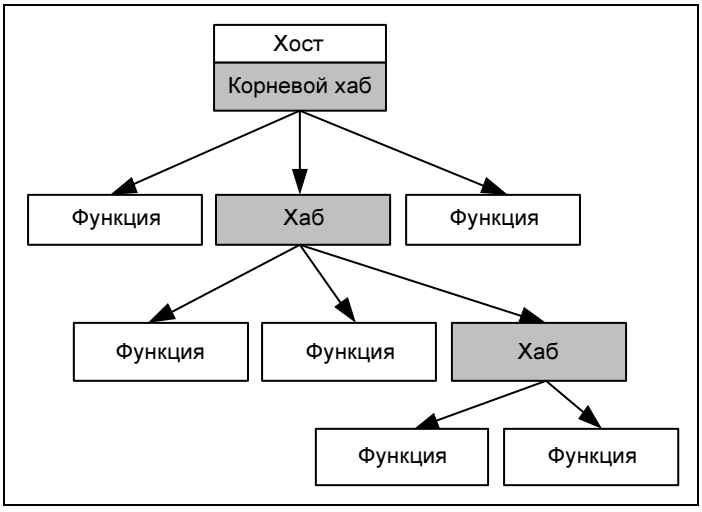


Рис. 1.2. Физическая архитектура USB

Детали физической архитектуры скрыты от прикладных программ в системном программном обеспечении (ПО), поэтому *логическая архитектура* выглядит как обычная звезда, центром которой является прикладное ПО, а вершинами — набор *конечных точек* (рис. 1.3).

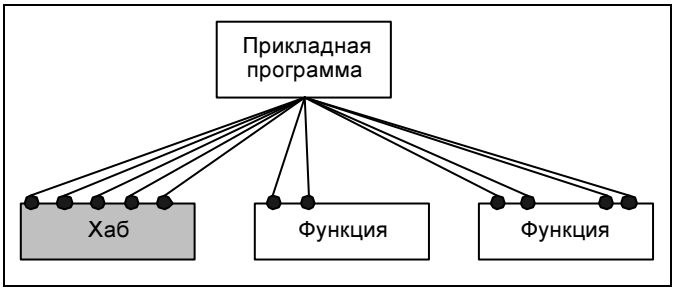


Рис. 1.3. Логическая архитектура USB

Прикладная программа ведет обмен информацией с каждой конечной точкой.

1.1.3. 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;
- *устройство* (device) может представлять собой хаб, функцию или их комбинацию (compound device);
- *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 USB-устройство или его отдельный блок, способный передавать и принимать информацию по шине USB. Каждая функция предоставляет конфигурационную информацию, описывающую возможности периферийного USB-устройства и требования к ресурсам. Перед использованием функция должна быть сконфигурирована хостом — ей должна быть выделена полоса в канале и выбраны опции конфигурации;
- *логическое USB-устройство* (logical device) представляет собой набор конечных точек.

1.1.4. 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 — USB-устройство должно поддерживать приостановку (suspended mode), при которой его потребляемый ток не превышает 500 мкА. USB-устройство должно автоматически приостанавливаться при прекращении активности шины;
- удаленное пробуждение — возможность удаленного пробуждения (remote wakeup) позволяет приостановленному USB-устройству подать сигнал хосту, который тоже может находиться в приостановленном состоянии. Возможность удаленного пробуждения описывается в конфигурации USB-устройства. При конфигурировании эта функция может быть запрещена.

1.1.5. Принципы передачи данных

Механизм передачи данных является асинхронным и блочным. Блок передаваемых данных называется *USB-фреймом* или *USB-кадром* (*см. разд. 1.1.11*) и передается за фиксированный временной интервал. Оперирование командами и блоками данных реализуется при помощи логической абстракции, называемой *каналом* (*см. разд. 1.1.13*). Внешнее устройство также делится на логические абстракции, называемые *конечными точками* (*см. разд. 1.1.12*). Таким образом, канал является логической связкой между хостом и конечной точкой внешнего устройства. Канал можно сравнить с открытым файлом.

Для передачи команд (и данных, входящих в состав команд) используется канал по умолчанию, а для передачи данных открываются либо *потоковые каналы*, либо *каналы сообщений* (*см. разд. 1.1.13*).

Все операции по передаче данных по шине USB инициируются хостом. Периферийные USB-устройства сами начать обмен данными не могут. Они могут только реагировать на команды хоста.

1.1.6. Механизм прерываний

Для шины USB настоящего механизма прерываний (как, например, для последовательного порта) не существует. Вместо этого хост опрашивает подключенные устройства на предмет наличия данных о прерывании. Опрос происходит в фиксированные интервалы времени, обычно каждые 1—32 мс. Устройству разрешается посылать до 64 байт данных.

С точки зрения драйвера, возможности работы с прерываниями фактически определяются хостом, который и обеспечивает поддержку физической реализации USB-интерфейса.

1.1.7. Режимы передачи данных

Пропускная способность шины USB, соответствующей спецификации 1.1, составляет 12 Мбит/с (т. е. 1,5 Мбайт/с). Спецификация 2.0 определяет шину с пропускной способностью 400 Мбайт/с. Полоса пропускания делится между всеми устройствами, подключенными к шине.

Шина USB имеет три режима передачи данных:

- низкоскоростной (LS, Low-speed);
- полноскоростной (FS, Full-speed);
- высокоскоростной (HS, High-speed, только для USB 2.0).

1.1.8. Логические уровни обмена данными

Спецификация USB определяет три *логических уровня* с определенными правилами взаимодействия. USB-устройство содержит интерфейсную, логическую и функциональную части. Хост тоже делится на три части — интерфейсную, системную и ПО. Каждая часть отвечает только за определенный круг задач. Логическое и реальное взаимодействие между ними показано на рис. 1.4.

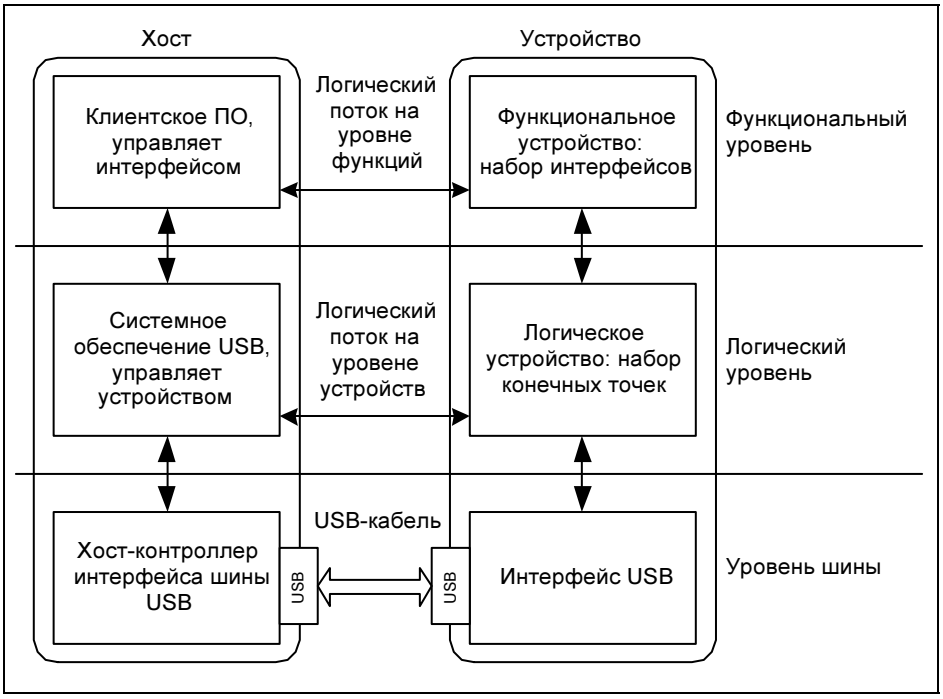


Рис. 1.4. Взаимодействие компонентов USB

Таким образом, операция обмена данными между прикладной программой и шиной USB выполняется путем передачи буферов памяти через следующие уровни:

- ❒ уровень клиентского ПО в хосте:
  - обычно представляется драйвером USB-устройства;
  - обеспечивает взаимодействие пользователя с операционной системой с одной стороны и системным драйвером с другой;

- ❒ уровень системного драйвера USB в хосте (USBD, Universal Serial Bus Driver):
  - управляет нумерацией устройств на шине;
  - управляет распределением пропускной способности шины и мощности питания;
  - обрабатывает запросы пользовательских драйверов;
- ❒ уровень хост-контроллера интерфейса шины USB (HCD, Host Controller Driver):
  - преобразует запросы ввода/вывода в структуры данных, по которым выполняются физические транзакции;
  - работает с регистрами хоста.

Рассмотрим каждый уровень более подробно.

1.1.8.1. Уровень клиентского ПО

На уровне клиентского программного обеспечения определяется тип передачи данных, необходимый для выполнения затребованной прикладной программой операции. После определения типа передачи данных на системный уровень передается следующее:

- ❒ буфер памяти, называемый клиентским буфером;
- ❒ *пакет запроса на ввод/вывод* (IRP, Input/output Request Packet), указывающий тип необходимой операции.

IRP содержит только сведения о запросе (адрес и длина буфера в оперативной памяти). Непосредственно обработкой запроса занимается системный драйвер USB. *Подробную информацию о запросах клиентского ПО см. в разд. 1.2, о формировании пакетов IRP — в гл. 12.*

1.1.8.2. Уровень системного драйвера USB

Уровень системного драйвера USB необходим для управления ресурсами USB. Он отвечает за выполнение следующих действий:

- ❒ распределение полосы пропускания шины USB;
- ❒ назначение логических адресов каждому физическому USB-устройству;
- ❒ планирование транзакций.

Распределение полосы пропускания

До установления каналов передач из хоста в конечную точку какого-либо USB-устройства системный драйвер должен сначала определить, может ли шина обеспечить требуемую полосу пропускания для данной точки. В каждом USB-устройстве есть специальная таблица, содержащая дескрипторы

конечных точек, в которых хранится значение минимально допустимой полосы пропускания для нее.

В фазе начальной инициализации системный драйвер читает эти дескрипторы и определяет необходимую суммарную полосу пропускания для USB-устройства. Хранящееся в дескрипторе значение определяет, какая доля пропускной способности шины необходима для работы конечной точки. При этом, однако, не учитываются никакие накладные расходы. Определяя общую потребность для поддержки канала к каждой конечной точке, системный драйвер USB учитывает следующее:

- число байтов данных;
- тип передачи данных;
- время восстановления хоста;
- время заполнения битами;
- уровень вложенности топологии.

Назначение логических адресов

Логическое USB-устройство представляет собой набор независимых конечных точек (*см. разд. 1.1.12*), с которыми клиентское ПО обменивается информацией. Каждому логическому USB-устройству (как функции, так и хабу) назначается свой адрес (1—127), уникальный на данной шине USB. Каждая конечная точка логического USB-устройства идентифицируется своим номером (0—15) и направлением передачи (IN — передача к хосту, OUT — от хоста).

Планирование транзакций

Транзакция на шине USB — это последовательность обмена пакетами между хостом и ПУ, в ходе которой может быть передан или принят один пакет данных. Когда клиентское ПО передает IRP системному драйверу USB, то он преобразует их в одну или несколько транзакций шины и затем передает получившийся перечень транзакций драйверу контроллера хоста.

Архитектура системного драйвера USB

Системный драйвер USB состоит из *драйвера USB* и *драйвера контроллера хоста*. Драйвер контроллера хоста принимает от системного драйвера перечень транзакций и выполняет следующие действия:

- планирует исполнение полученных транзакций, добавляя их к списку транзакций;
- извлекает из списка очередную транзакцию и передает ее на уровень хост-контроллера интерфейса шины USB;
- отслеживает состояние каждой транзакции вплоть до ее завершения.

При выполнении всех связанных с командным пакетом транзакций системный уровень уведомляет об этом клиентский уровень.

1.1.8.3. Уровень хост-контроллера интерфейса

Уровень хост-контроллера интерфейса шины USB получает отдельные транзакции от драйвера контроллера хоста (в составе уровня системного обеспечения USB) и преобразует их в соответствующую последовательность операций шины. В результате этого USB-пакеты передаются вдоль всей физической иерархии хабов (на рис. 1.4 мы изобразили последовательность хабов как одну логическую линию, но физически это может быть как один USB-кабель, так и последовательность хабов) до периферийного USB-устройства (правая часть рис. 1.4).

1.1.8.4. Уровень шины периферийного USB-устройства

Нижний уровень периферийного USB-устройства называется уровнем интерфейса шины USB. Он взаимодействует с интерфейсным уровнем шины USB на стороне хоста и передает пакеты данных от хоста в формате, определяемом спецификацией USB. Затем он передает пакеты вверх — уровню логического USB-устройства.

1.1.8.5. Уровень логического USB-устройства

Средний уровень периферийного USB-устройства называется уровнем логического USB-устройства. Каждое логическое USB-устройство представляется набором своих конечных точек, с которыми может взаимодействовать системный уровень хоста. Эти точки являются источниками и приемниками всех коммуникационных потоков между хостом и периферийными USB-устройствами.

1.1.8.6. Функциональный уровень USB-устройства

Самый верхний уровень периферийного USB-устройства называется функциональным уровнем. Этот уровень соответствует уровню клиентского обеспечения хоста. С точки зрения клиентского уровня, нижележащие уровни нужны для организации между ним и конечными точками прямых *каналов данных* (*см. разд. 1.1.13*), которые идут вплоть до функционального уровня периферийного USB-устройства. А с точки зрения нашей схемы функциональный уровень выполняет следующие действия:

- получает данные, посылаемые клиентским уровнем хоста из конечных точек каналов данных нижележащего уровня логического USB-устройства;
- посылает данные клиентскому уровню хоста, направляя их в конечные точки каналов данных нижележащего уровня логического USB-устройства.

1.1.9. Передача данных по уровням

Логически передача данных между конечной точкой и ПО производится с помощью выделения канала и обмена данными по этому каналу, а с точки зрения представленных уровней передача данных выглядит следующим образом (рис. 1.5).

1. Клиентское ПО посылает IPR-запросы на уровень системного драйвера USB.
2. Системный драйвер USB разбивает запросы на транзакции по следующим правилам:
   - выполнение запроса считается законченным, когда успешно завершены все транзакции, его составляющие;
   - все подробности отработки транзакций (такие как ожидание готовности, повтор транзакции при ошибке, неготовность приемника и т. д.) до клиентского ПО не доводятся;
   - ПО может только запустить запрос и ожидать или выполнения запроса, или выхода по тайм-ауту;
   - устройство может сигнализировать о серьезных ошибках (*см. разд. 1.1.14*), что приводит к аварийному завершению запроса, о чем уведомляется источник запроса.

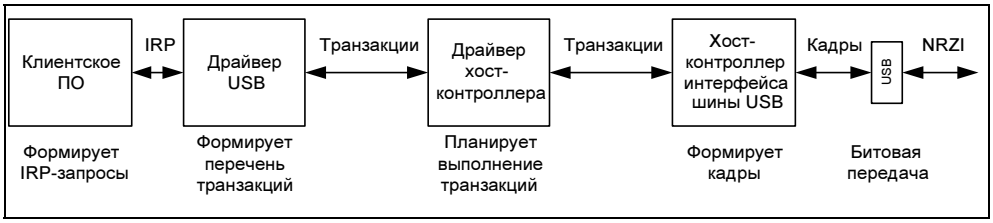


Рис. 1.5. Уровни передачи данных

3. Драйвер контроллера хоста принимает от системного драйвера USB перечень транзакций и выполняет следующие действия:
   - планирует исполнение полученных транзакций, добавляя их к списку транзакций;
   - извлекает из списка очередную транзакцию и передает ее уровню хост-контроллера интерфейса шины USB;
   - отслеживает состояние каждой транзакции вплоть до ее завершения.
4. Хост-контроллер интерфейса шины USB формирует кадры.

5. Кадры передаются последовательной передачей бит по методу, называемому *NRZI with bit stuffing* (Non Return to Zero Invert, метод возврата к нулю с инвертированием единиц).

Таким образом, можно сформировать следующую *упрощенную* схему (рис. 1.6):

- каждый кадр состоит из наиболее приоритетных посылок, состав которых формирует драйвер контроллера хоста;
- каждая передача состоит из одной или нескольких транзакций (*см. разд. 1.1.16*);
- каждая транзакция состоит из пакетов;
- каждый пакет состоит из идентификатора пакета, данных (если они есть) и контрольной суммы.

В следующих разделах мы рассмотрим все составляющие передачи более подробно.

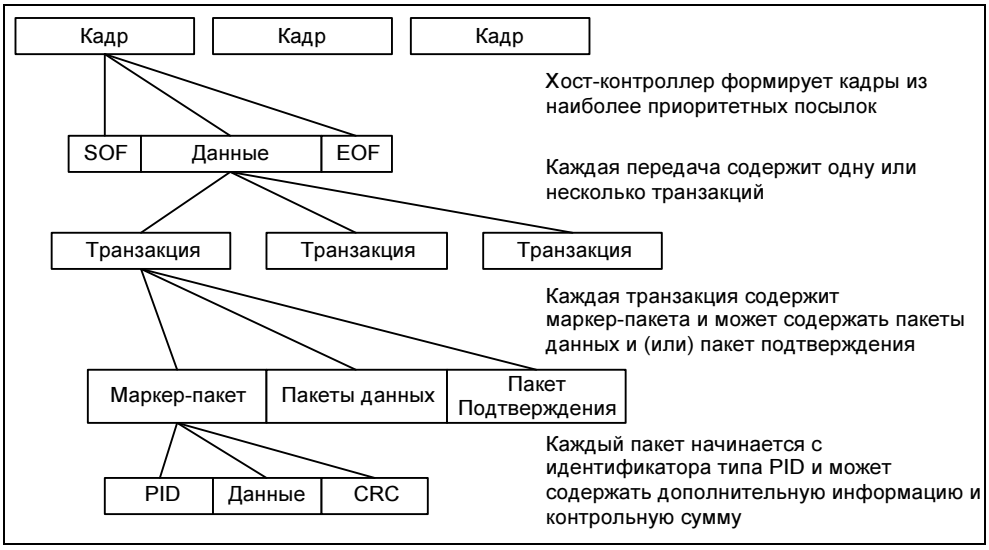


Рис. 1.6. Общая схема составляющих USB-протокола

1.1.10. Типы передач данных

Спецификация шины определяет четыре различных типа передачи (transfer type) данных для конечных точек (табл. 1.1):

- *управляющие передачи* (control transfers) — используются хостом для конфигурирования устройства во время подключения, для управления уст-

ройством и получения статусной информации в процессе работы. Протокол обеспечивает гарантированную доставку таких посылок. Длина поля данных управляющей посылки не может превышать 64 байта на полной скорости и 8 байтов на низкой. Для таких посылок хост гарантированно выделяет 10% полосы пропускания;

- *передачи массивов данных* (bulk data transfers) — применяются при необходимости обеспечения гарантированной доставки данных от хоста к функции или от функции к хосту, но время доставки не ограничено. Такая передача занимает всю доступную полосу пропускания шины. Пакеты имеют поле данных размером 8, 16, 32 или 64 байт. Приоритет у таких передач самый низкий, они могут приостанавливаться при большой загрузке шины. Допускаются только на полной скорости передачи. Такие посылки используются, например, принтерами или сканерами;
- *передачи по прерываниям* (interrupt transfers) — используются в том случае, когда требуется передавать одиночные пакеты данных небольшого размера. Каждый пакет требуется передать за ограниченное время. Операции передачи носят спонтанный характер и должны обслуживаться не медленнее, чем того требует устройство. Поле данных может содержать до 64 байтов при передаче на полной скорости и до 8 байтов на низкой. Предел времени обслуживания устанавливается в диапазоне 1—255 мс для полной скорости и 10—255 мс — для низкой. Такие передачи используются в устройствах ввода, таких как мышь и клавиатура;
- *изохронные передачи* (isochronous transfers) — применяются для обмена данными в "реальном времени", когда на каждом временном интервале требуется передавать строго определенное количество данных, но доставка информации не гарантирована (передача данных ведется без повторения при сбоях, допускается потеря пакетов). Такие передачи занимают предварительно согласованную часть пропускной способности шины и имеют заданную задержку доставки. Изохронные передачи обычно используются в мультимедийных устройствах для передачи аудио- и видеоданных, например, цифровая передача голоса. Изохронные передачи разделяются по способу синхронизации конечных точек — источников или получателей данных — с системой: различают асинхронный, синхронный и адаптивный классы устройств, каждому из которых соответствует свой тип канала USB.

Таблица 1.1. *Типы передач по шине USB*

Тип передачи	Направление	Частота запуска	Гарантия доставки
Управляющие посылки	Двунаправленные	Не гарантируется	Есть
Изохронные (только для высокоскоростных устройств)	Однонаправленные	Каждые 1 мс	Нет

Таблица 1.1 *(окончание)*

Тип передачи	Направление	Частота запуска	Гарантия доставки
Передача по прерываниям	Однонаправленные	Определяется частотой опроса	Есть
Передача массивов данных	Двунаправленные	Не гарантируется	Есть

Все операции по передаче данных инициируются только хостом независимо от того, принимает ли он данные или пересылает в периферийное USB-устройство. Все невыполненные операции хранятся в виде четырех списков по типам передач. Списки постоянно обновляются новыми запросами. Планирование операций по передаче информации в соответствии с упорядоченными в виде списков запросами выполняется хостом с интервалом один кадр. Обслуживание запросов выполняется по следующим правилам:

- наивысший приоритет имеют изохронные передачи;
- после отработки всех изохронных передач система переходит к обслуживанию передач прерываний;
- в последнюю очередь обслуживаются запросы на передачу массивов данных;
- по истечении 90% указанного интервала хост автоматически переходит к обслуживанию запросов на передачу управляющих команд независимо от того, успел ли он полностью обслужить другие три списка или нет.

Выполнение этих правил гарантирует, что управляющим передачам всегда будет выделено не менее 10% пропускной способности шины USB. Если передача всех управляющих пакетов будет завершена до истечения выделенной для них доли интервала планирования, то оставшееся время будет использовано хостом для передач массивов данных. Таким образом:

- изохронные передачи гарантированно получают 90% пропускной способности шины;
- передачи прерываний занимают оставшуюся часть этой доли;
- под передачу данных большого объема выделяется все время, оставшееся после изохронных передач и передач прерываний (по-прежнему в рамках 90% пропускной способности);
- управляющим передачам гарантируется 10% пропускной способности шины;
- если передача всех управляющих пакетов будет завершена до завершения выделенного для них 10-процентного интервала, то оставшееся время будет использовано для передач данных большого объема.

1.1.11. Кадры

Любой обмен по шине USB инициируется хостом. Он организует обмены с устройствами согласно своему плану распределения ресурсов.

Контроллер циклически (с периодом 1,0±0,0005 мс) формирует *кадры* (frames), в которые укладываются все запланированные передачи (рис. 1.7). Каждый кадр начинается с посылки маркер-пакета SOF (Start Of Frame, начало кадра, *см. разд. 1.1.14*), который является синхронизирующим сигналом для всех устройств, включая хабы. В конце каждого кадра выделяется интервал времени EOF (End Of Frame, конец кадра), на время которого хабы запрещают передачу по направлению к контроллеру. Если хаб обнаружит, что с какого-то порта в это время ведется передача данных, то он отключит этот порт.

В режиме высокоскоростной передачи (*см. разд. 1.1.10*) пакеты SOF передаются в начале каждого *микрокадра* (период 125±0,0625 мкс). Хост планирует загрузку кадров так, чтобы в них всегда находилось место для наиболее приоритетных передач, а свободное место кадров заполняется низкоприоритетными передачами больших объемов данных. Спецификация USB позволяет занимать под периодические транзакции (изохронные и прерывания) до 90% пропускной способности шины.

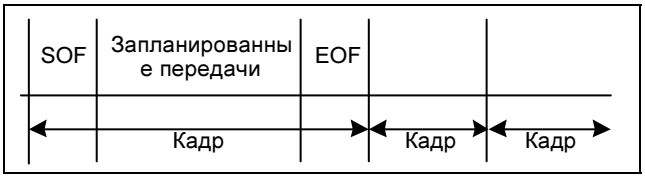


Рис. 1.7. Поток кадров USB

Каждый кадр имеет свой номер. Хост оперирует 32-битным счетчиком, но в маркере SOF передает только младшие 11 бит (*см. разд. 1.1.14*). Номер кадра циклически увеличивается во время EOF.

Для изохронной передачи (*см. разд. 1.1.10*) важна синхронизация устройств и контроллера. Есть три варианта синхронизации:

- синхронизация внутреннего генератора устройства с маркерами SOF;
- подстройка частоты кадров под частоту устройства;
- согласование скорости передачи (приема) устройства с частотой кадров.

В каждом кадре может быть выполнено несколько транзакций, их допустимое число зависит от скорости, длины поля данных каждой из них, а также от задержек, вносимых кабелями, хабами и USB-устройствами. Все транзак-