

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ ОБМЕНА ИНФОРМАЦИЕЙ ПО ИНТЕРФЕЙСУ USB

К.А. Домбровский, В.А. Руднев

INTELLIGENT SYSTEMS OF INFORMATION EXCHANGE VIA USB

K.A. Dombrovsky, V.A. Rudnev

В статье рассматриваются способы организации обмена данными между электронными устройствами с компьютером, используя интерфейс USB. Производится обзор способов программной и аппаратной реализации обмена информацией.

Ключевые слова: интерфейс, USB, обмен данными.

The article considers methods of control for data exchange between electronic devices and computer by means of USB interface. The review of methods for software and hardware implementation of data exchange is given.

Keywords: interface, USB, data exchange.

Введение

В последнее время компьютеры стали неотъемлемой частью нашей жизни. Все больше и больше задач по обработке и хранению информации возлагается на вычислительные средства. Современные языки программирования позволяют написать полноценное приложение даже начинающему программисту за небольшой промежуток времени. Однако существует довольно широкий круг задач, для которых важна не задача хранения или обработки информации, а задача управления внешними устройствами. К примеру, задача управления какими-либо станками на машиностроительных предприятиях или ввод информации о положении элементов управления в компьютер. Классическое решение таких проблем – использование интерфейсов компьютера для подключения внешних устройств, осуществляющих взаимодействие со станками, пультами и т. п. Статья призвана рассказать о проектировании подобных систем и способах взаимодействия внешних устройств с компьютерами. В ней дается описание интерфейса USB, кратко рассматриваются современные интерфейсы, а также предложены способы конструирования программно-аппаратных комплексов на основе этого интерфейса.

Взаимодействие электронных устройств с компьютером, классификация интерфейсов

Сначала поговорим о самом понятии взаимодействия. Под взаимодействием будем понимать обмен данными в одностороннем (от устройства к компьютеру или наоборот) или в двустороннем (в обе стороны) направлениях. Для различных задач может потребоваться различные типы взаимодействия (как по скорости, так и по направленности). Соответственно были разработаны и стандартизованы некоторые *интерфейсы*. Под интерфейсом подразумевается совокупность средств и методов взаимодействия между элементами системы, в нашем случае компьютера и внешнего устройства.

Интерфейсы имеют множество классификаций. Мы рассмотрим только самую распространенную – по количеству линий передачи данных. Классический способ – когда количество сигнальных линий равно разрядности передаваемых данных. То есть, если требуется передавать байты информации, используется 8 линий для передачи. Такие интерфейсы называются *параллельными*. Примером может служить, например, LPT-порт (или Centronics). Параллельные интерфейсы характеризуются относительно высокой скоростью передачи (до 16 Мбит/с у LPT), однако имеют низкую помехоустойчивость (длина линии для LPT не

Домбровский Кирилл Александрович – канд. техн. наук, доцент кафедры электронных вычислительных машин приборостроительного факультета, Южно-Уральский государственный университет; kirill-dombrovsky@yandex.ru

Руднев Владислав Андреевич – аспирант кафедры электронных вычислительных машин приборостроительного факультета, Южно-Уральский государственный университет; megadrivert@smartmcsu.ru

Kirill Aleksandrovich Dombrovsky – Candidate of Science (Engineering), associate professor of the Department of Electronic Computers of Computer Technologies, Control and Radioelectronics Faculty of South Ural State University; kirill-dombrovsky@yandex.ru
Vladislav Andreevich Rudnev – postgraduate student of the Department of Electronic Computers of Computer Technologies, Control and Radioelectronics Faculty of South Ural State University; megadrivert@smartmcsu.ru

более 10 метров). Также параллельные интерфейсы требуют больших затрат оборудования. Из-за этого параллельные интерфейсы уступают позиции **последовательным**. Последовательные интерфейсы используют для передачи данных всего одну линию, а данные передаются побитно. Примером такого интерфейса может служить хорошо известный COM-порт. Последовательные интерфейсы характеризуются значительно большей помехоустойчивостью, нежели параллельные, однако платой за это служит более низкая скорость передачи данных. Но следует отметить, что на сегодняшний день в связи с развитием элементной базы последовательные интерфейсы позволяют передавать данные даже с большей скоростью, чем параллельные. Промежуточное положение занимают **смешанные** интерфейсы, которые можно рассматривать как множество последовательных. То есть сигнальных линий больше одной, однако информация по каждой передается последовательно. К таким интерфейсам относится, к примеру, сетевой интерфейс Ethernet.

Рассмотрим некоторые современные интерфейсы для передачи данных. Сразу следует отметить, что все они являются последовательными, так как современных параллельных интерфейсов, которые получили широкое распространение, не существует.

Одним из современных интерфейсов для высокоскоростной передачи данных является интерфейс **Fire-Wire** (Fi-Wi). Скорость передачи по этому интерфейсу достигает 400 Мбит/с. Этот интерфейс используется, к примеру, для подключения некоторых типов видеокамер, фотоаппаратов и т. п. К его преимуществам следует отнести подачу питающего напряжения на ведомое устройство от компьютера.

Еще одним современным интерфейсом для взаимодействия компьютера и периферийных устройств является беспроводной интерфейс **IrDA**. Здесь применен способ обмена, сходный с COM-портом, за исключением того, что роль проводов здесь выполняют ИК-излучение. Этот интерфейс характеризуется сравнительно высокой скоростью работы (от 2,4 кбит/с до 16 Мбит/с), его простота и дешевизна аппаратной поддержки обусловили его довольно широкое распространение в сфере мобильной техники. К недостаткам следует отнести то, что связываемые устройства должны находиться в области оптической видимости и на расстоянии не более 5 метров.

Широкое распространение в сфере мобильной техники также получил интерфейс **Bluetooth**, в котором применен способ передачи данных по радиоканалу с частотой 2,4 ГГц. Существует несколько спецификаций этого типа связи устройств, и в версии 2.0 (самая последняя версия на сегодняшний день) скорость передачи данных достигает 2,1 Мбит/с при расстоянии между устройствами до 50–100 метров (при прямой видимости и низком уровне помех).

Одним из самых распространенных на сегодняшний день является проводной интерфейс **USB**, который мы рассмотрим более подробно.

Интерфейс USB

История интерфейса USB начинается в 1996 году, когда была опубликована версия 1.0 этого интерфейса, получившего название Universal Serial Bus (USB), а осенью 1998 года вышла версия 1.1, устраняющая недоработки первой версии. Разработчики ориентировались на создание интерфейса, обладающего следующими свойствами:

- легко реализуемое расширение периферии ПК;
- дешевое решение, позволяющее передавать данные с высокой скоростью;
- гибкость протокола передачи;
- интеграция с выпускаемыми устройствами;
- охват всевозможных конфигураций и конструкций ПК;
- обеспечение стандартного интерфейса, способного быстро завоевать рынок.

Физически USB-соединение представляет собой кабель из 4 жил: две из них подают питание (5 В, 500 мА) на устройство, другие две обеспечивают передачу данных. Следует отметить, что интерфейс USB допускает «горячее» подключение – во время работы ПК.

Архитектура USB подразумевает подключение одного или нескольких устройств к компьютеру, который является главным управляющим устройством и называется **хостом**. Подключаемые устройства являются **ведомыми**. Существует 3 скоростных режима передачи данных (по спецификации 2.0):

- низкоскоростной (Low Speed) – до 1,5 Мбит/с;
- полноскоростной (Full Speed) – до 12 Мбит/сек.
- высокоскоростной (High Speed) – до 480 Мбит/с.

В 2008 году появилась спецификация USB3.0, которая увеличила максимальную скорость передачи до 5 Гбит/с, однако для достижения этой скорости необходимы новые типы кабелей (введены дополнительные физические линии). При этом интерфейс сохраняет обратную совместимость как аппаратно, так и программно.

Каждое ведомое устройство имеет несколько **конечных точек** – логических приемников информации (от 1 до 120). Каждое ведомое устройство может содержать от 1 до 128 конечных точек. Передача данных ведется от хоста к конечным точкам. Существует 4 вида передач:

- контрольные передачи (Control). Используются хостом для конфигурирования устройства во время подключения;
- передача массивов данных (Bulk). Используются для передачи больших объемов информации;

– передача по прерыванию (Interrupt). Используется для передачи небольших объемов данных, для которых критично время доставки;

– изохронные передачи (Isochronous). Используются для передачи данных в «реальном времени». Передается строго определенный объем данных через определенный промежуток времени. При этом допускается потеря некоторых пакетов.

Таким образом, с помощью интерфейса USB можно организовать передачу требуемого количества данных с требуемыми условиями доставки, выбрав нужное количество различных конечных точек для устройства. Однако есть стандартные протоколы, которые получили широкое распространение.

– **Mass Storage**. Этот протокол разработан для подключения к компьютеру внешних накопителей информации через интерфейс USB.

– **CDC**. Был создан для совместимости с COM-портом. Установив драйверы, система обращается к ведомому устройству, как к обычному COM-порту.

– **HID**. Human Interface Device. Этот протокол был призван обеспечивать связь человека с компьютером. Он довольно гибок и обширен и позволяет создавать интересные режимы обмена данными с ПК.

Теперь после рассмотрения шины USB можно перейти к вопросу проектирования программно-аппаратных комплексов на его основе. Сначала рассмотрим сторону «железа» – устройств, подключаемых к ПК. Рассмотрим варианты реализации этих устройств.

Способы построения аппаратной части для обмена информацией

Одним из наиболее простых и распространенных решений является применение **микроконтроллеров** – микросхем, имеющих в своем составе процессор, оперативную и постоянную память, а также множество периферийных устройств (таймеры, контроллеры интерфейсов и т. п.). Микроконтроллеры бывают самых различных типов: от микромоощных и миниатюрных до таких, которые могут потягаться в производительности с современным компьютером. Рассмотрим наиболее распространенные на сегодняшний день микроконтроллеры.

Микроконтроллеры фирмы *Microchip* – одни из самых распространенных на сегодняшний день. От «классических» PIC12 и PIC16 до мощных 32-разрядных ds32. Во многие микроконтроллеры встроена аппаратная поддержка интерфейса USB, что значительно сокращает время разработки программ и увеличивает быстродействие системы в целом.

Главным конкурентом компании *Microchip* является компания *ATMEL*, которая производит огромную номенклатуру дешевых и производительных контроллеров. От классических AVR (ATmega и ATtiny) до мощных контроллеров с ARM-ядрами (AT91SAM). Контроллеры ATMEL

сочетают низкое энергопотребление с высокой производительностью и богатой периферией, что делает их весьма привлекательными для разработки устройств. Во многих микроконтроллерах этой фирмы также предусмотрена аппаратная поддержка интерфейса USB, однако даже на AVR-контроллерах можно построить USB-устройства, используя программные способы (как будет рассмотрено далее).

Также существуют контроллеры, главная задача которых – обеспечение связи с компьютером по интерфейсу USB. В таких микроконтроллерах упор сделан на аппаратную поддержку этого интерфейса. Самыми известными контроллерами этого типа являются устройства компании *Cypress*. Их контроллеры поддерживают USB-обмен на скоростях до 480 мБит/с. Недавно вышедшие микроконтроллеры поддерживают даже спецификацию USB3.0.

Помимо микроконтроллеров одним из современных способов обеспечения USB-обмена является применение микросхем ПЛИС (программируемые логические интегральные схемы), которые содержат на кристалле множество логических вентиляей, которые могут соединяться между собой произвольным способом при программировании. Самыми известными на сегодняшний день микросхемами ПЛИС являются микросхемы *Altera* и *Spartan II*.

Однако на сегодняшний день существует множество устаревших устройств, которые ориентированы на COM или LPT-порты, для которых нецелесообразно переделывать аппаратную часть. Специально для таких случаев созданы микросхемы-мосты, которые позволяют создавать виртуальные последовательные или параллельные порты при подключении через USB. Ярким примером таких микросхем являются микросхемы фирмы *FTDI* типа FT232RL. Эти микросхемы подключаются к ПК через USB, создают виртуальный COM-порт, через который осуществляется взаимодействие с внешним устройством. К преимуществам такого решения можно отнести простоту реализации как со стороны ПК, так и со стороны аппаратной части. Однако эти микросхемы стоят подчас в 2–3 раза дороже микроконтроллеров. Таким образом, их применение скорее можно рассматривать как модернизацию устаревшего оборудования, нежели как инструмент для построения нового.

Чаще всего для построения устройств применяют микроконтроллеры. До недавнего времени для реализации USB интерфейса можно было только применять микроконтроллеры с его аппаратной поддержкой (типа AT90USBxxx или PIC18F4xx, STM32F107). Применение этих контроллеров оправдано, когда требуется передавать большие объемы данных или обеспечить максимально возможную скорость передачи. Но такие контроллеры стоят в 2–3 раза дороже аналогичных без USB. Однако на сегодняшний день появились средства программной поддержки USB, которые могут быть портированы практически в любой

современный микроконтроллер [2]. Их суть состоит в том, что сам обмен осуществляется непосредственной сменой логических сигналов на выводах микроконтроллера, именно за счет этого достигается независимость от архитектуры и типа микроконтроллера.

Особенности проектирование программного обеспечения для компьютера

Теперь рассмотрим сторону программного обеспечения, размещаемого на компьютере. Здесь есть несколько путей «общения» с USB-устройствами. Первый из них – использование стандартных протоколов USB вроде CDC, HID, Mass Storage и т.п. Преимущество этого подхода в том, что не требуется написания драйверов для устройства – операционные системы уже включают в себя эти стандартные драйверы, что значительно сокращает время разработки ПО. Однако здесь же кроется и недостаток такого подхода – количество стандартных протоколов фиксировано и зачастую они не удовлетворяют нуждам разработчика. В этом случае применяют другой подход: написание собственного драйвера. Этот процесс довольно сложен и трудоемок. Однако результат оправдывает затраченное время: протокол обмена будет именно такой, какой необходим. Еще один способ, который связан с вышеупомянутыми микросхемами-мостами – взять готовый драйвер, который предлагает использовать фирма-производитель такой микросхемы. Этот способ имеет те же преимущества и недостатки, что и первый: есть готовый драйвер, однако он не всегда удовлетворяет требованиям к обмену.

Если требуется невысокая скорость передачи – наиболее привлекателен протокол HID (Human Interface Device). У этого протокола есть множество плюсов:

- максимальная простота протокола, что сокращает код со стороны «железа»;
- возможность программно пропускать незначительную информацию, что увеличивает надежность и гибкость передач;
- протокол является наращиваемым и устойчивым, что позволяет создавать более новые версии протокола, совместимые с более ранними версиями;
- устройство HID описывает само себя при инициализации, позволяя создавать классовое ПО;
- нет необходимости писать драйверы для устройства – операционная система установит их автоматически.

Стандартные применения HID-протокола – это клавиатуры, мыши, трекболы, джойстики, считыватели штрих-кодов, термометры и т. п. Как уже было отмечено – драйвер для устройства не требуется, поскольку он уже включен в состав ОС. Однако ничто не мешает написать HID-драйвер самому, однако в этом в большинстве случаев нет необходимости.

Для работы с HID-устройством в операционных системах Linux служит библиотека libusb.h, а в сис-

темах Windows – с помощью dll-библиотек HID.dll и SetupAPI.dll или при помощи инструментов DirectInput из пакета DirectX. Рассмотрим более подробно работы в операционной системе Windows.

Для разработки программ для этой операционной системы требуется пакет библиотек WinDDK, который включает в себя все необходимые инструменты для работы с HID-устройствами. Наиболее простым и удобным является использование Feature-репортов. Они имеют длину от 128 до 254 байт и передаются при помощи вызова функций HidD_SetFeature и HidD_GetFeature, которые соответственно отправляют и принимают данные устройства.

Следует отметить, что при грамотном проектировании программы, использование этих функций напрямую нежелательно, так как они работают с указателями и являются небезопасными. Для удобной работы целесообразно завернуть эти функции в классы следующим образом:

1) класс обращений к функциям HidD_SetFeature и HidD_GetFeature. Этот класс обеспечивает вызов «опасных» функций внутри методов, упрощая с ними работу. Он возвращает ошибки, возникшие на самом низком уровне (при помощи исключений или просто возвратом кода ошибки);

2) класс обеспечения корректности протокола. На этом уровне контролируется правильность принятого протокола обмена, обеспечивается правильное формирование управляющих посылок для устройства;

3) класс «завертывания» внешнего устройства. К примеру, для станков это будет станок->поднять_резец.

Такой подход обеспечивает максимальную абстракцию и облегчает переход между операционными системами (потребуется менять только 1 класс).

Выводы

Для построения современных программно-аппаратных комплексов наиболее удобен интерфейс USB. Аппаратная часть такого проекта чаще всего строится на микроконтроллерах с программной или аппаратной поддержкой USB. Также возможно применение специальных микросхем-мостов. Для работы с низкоскоростными устройствами наиболее привлекателен протокол HID. Для обеспечения надежной работы программы на ПК, низкоуровневые функции обращения к USB-устройству лучше реализовывать в классы с гексагональной архитектурой.

Литература

1. Агуров, П.В. *Интерфейс USB. Практика использования и программирования* / П.В. Агуров. – 2-е изд. – СПб.: BHV, 2008. – 256 с.
2. *ATMEGA8535(L) datasheet* – Atmel Corporation. – <http://www.atmel.com/atmel/acrobat/doc2502.pdf>

Поступила в редакцию 23 сентября 2012 г.