

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ**

**Донецкий национальный технический университет  
Факультет вычислительной техники и информатики  
Кафедра ЭВМ**

**ОТЧЕТ О НИРС**

***“Цифровое устройство индикации событий на базе  
микроконтроллера семейства AVR”***

*Руководитель*  
*старший преподаватель* \_\_\_\_\_ *О. Г. Шевченко*  
(подпись) (дата)

*Разработал*  
*Студент группы СП-07м* \_\_\_\_\_ *С.С. Лоцманов*  
(подпись) (дата)

**Донецк 2007**

## **РЕФЕРАТ**

Отчет о НИРС:

стр. 35, рис. 8, табл. 5

Целью работы является изучение основных принципов реализации интерфейса USB в микроконтроллерных устройствах, разработка функциональной схемы устройства сопряжения и программы управления обменом данных между микроконтроллером и ПК.

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС, EEPROM, USB, МИКРОКОНТРОЛЛЕР, МИКРОСХЕМА, FIFO, ATMEL, ИНТЕРФЕЙС, FTDI.

## СОДЕРЖАНИЕ

Введение.....	4
1. Описание проектируемого устройства.....	5
1.1. Описание USB интерфейса.....	7
1.1.1 Структура и взаимодействие системы USB.....	7
1.1.2 Физический интерфейс.....	10
1.1.3 Системное конфигурирование.....	13
1.1.4 USB 2.0.....	14
1.2. Микросхема FT245BM: преобразователь интерфейса USB в параллельный FIFO.....	16
1.3. Описание микроконтроллера Atmega128.....	18
2. Основные варианты реализации интерфейса USB в микроконтроллерных устройствах.....	23
3. Реализованные элементы программно-аппаратного комплекса.....	28
Вывод.....	31
Список использованной литературы.....	32
Приложение 1. Листинг программы с подробными комментариями.....	33

## ВВЕДЕНИЕ

Цифровые устройства плотно интегрировались в нашу жизнь. Сложно уже представить современный уровень жизни без таких устройств как: мобильный телефон, карманный компьютер, электронный органайзер. Каждое из этих устройств, призвано повысить оперативность и комфортность условий работы. Ту же цель преследует предлагаемое устройство индикации событий. Зачастую пользователь ПК может не заметить появление какого-либо, даже самого важного, фоновое события, а с настройкой оповещения о появлении такового возникает и ряд неудобств: теряется фокус ввода, уменьшается рабочая область экрана и т.п. Поэтому предлагается разработка такого устройства оповещения событий, которое не прерывает ход работы на ПК, но в тоже время отслеживает события.

## 1. ОПИСАНИЕ ПРОЕКТИРУЕМОГО УСТРОЙСТВА

В разрезе поставленной задачи разрабатывается программно-аппаратный комплекс, включающий в себя: управляющую программу на ПК, схему сопряжения с микроконтроллером, управляющий микроконтроллер и непосредственно светодиодный экран оповещения о событиях.

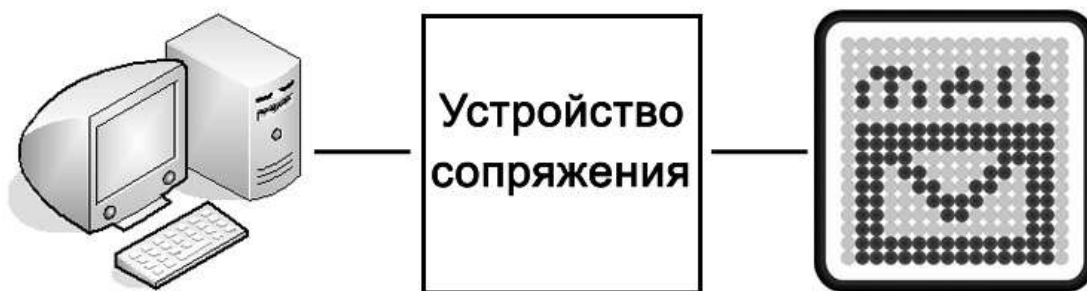


Рис. 1.1. Структурная схема

Алгоритм работы комплекса представляет собой следующее:

1. Управляющая программа на ПК отслеживает появление интересующих пользователя событий.
2. При появлении события программа создает пакет данных для отображения и формирует прерывание контроллера.
3. Схема сопряжения с контроллером прерывает работу контроллера и записывает новую информацию для отображения.
4. На светодиодном экране отображается интересующая пользователя информация.

Среди всех представленных на рисунке 1.1. элементов разработана функциональная электрическая схема устройства сопряжения, а в рамках магистерской работы будет создан весь программно-аппаратный комплекс.

Устройство сопряжения состоит из: микросхемы согласования USB и контроллера (FT245BM), электрически стираемого перепрограммируемого ПЗУ (EEPROM) и AVR микроконтроллера Atmega128.

Функциональное назначение элементов схемы:

1. В качестве интерфейса между цифровым устройством индикации и ПК принят интерфейс USB. Шина USB обеспечивает передачу данных устройству, а также его питание.

2. Микросхема FT245BM передает пакеты данных от USB к микроконтроллеру, а также формирует запрос на получение максимального тока потребления на одно USB устройство (500 мА). Такой запрос избавляет от подключения дополнительных источников тока.

3. EEPROM (Electrically Erasable Programmable Read-Only Memory) хранит информацию об устройстве и конфигурации узла USB.

4. Микроконтроллер Atmega128 управляет процессом отображения информации на светодиодном экране.

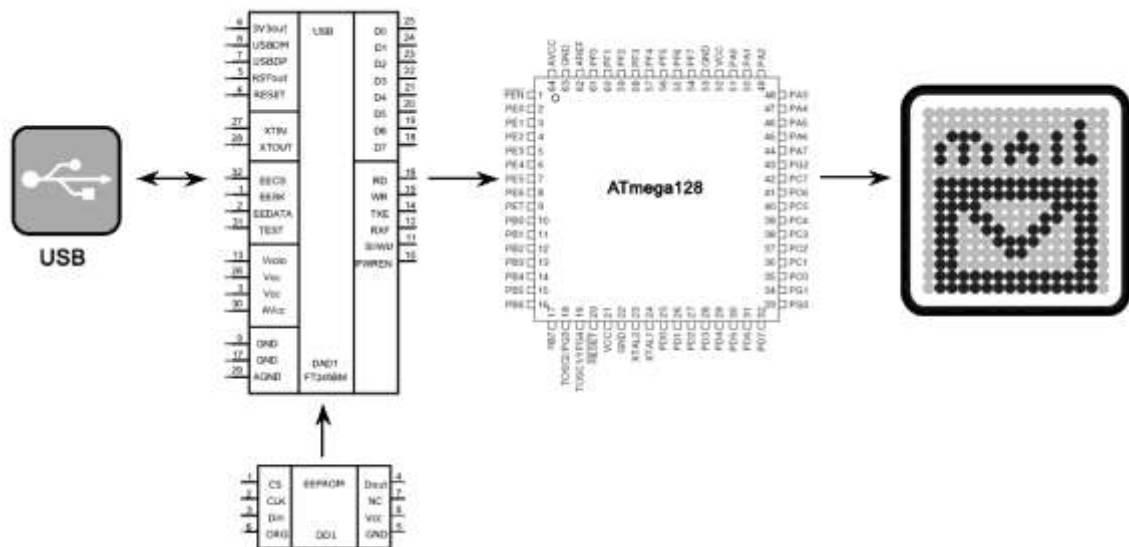


Рис. 1.2. Взаимодействие элементов схемы

Пакет с индицируемой информацией формируется управляющей программой на ПК и через USB интерфейс попадает в микросхему FT245BM, где переводится из последовательного представления в параллельное и передается контроллеру. Контроллер записывает пакет в память емкостью 4 Кбайта. Дальнейшие действия с записанной в память информацией выполняет программа, прошитая в микроконтроллер, которая собственно и обеспечивает индикацию на светодиодном экране.

## 1.1. ОПИСАНИЕ USB ИНТЕРФЕЙСА

USB( Universal Serial Bus - универсальная последовательная шина) является промышленным стандартом расширения архитектуры РС, ориентированным на интеграцию с телефонией и устройствами бытовой электроники. Спецификация USB 1.0 была опубликована в январе 1996.

Архитектура USB определялась следующими критериями:

- Легко реализуемое расширение периферии ПК.
- Дешевое решение, поддерживающее скорость передачи до 12 Мбит/с.
- Полна поддержка в реальном времени передачи аудио и сжатых видео данных.
- Гибкость протокола для смешанной передачи изоморфных данных и асинхронных сообщений.
- Интеграция в технологию выпускаемых устройств.
- Доступность в ПК всех конфигураций и размеров.
- Открытие новых классов устройств, расширяющих ПК.

С точки зрения пользователя привлекательны такие черты USB:

- Простота кабельной системы подключений.
- Изоляция подробностей электрических подключений от пользователя.
- Самоидентифицирующаяся периферия, автоматическая связь устройств с драйверами и конфигурирование.
- Возможность динамического подключения и реконфигурирования периферии.

### 1.1.1 СТРУКТУРА И ВЗАИМОДЕЙСТВИЕ СИСТЕМЫ USB

USB обеспечивает обмен данными между хост-компьютером и множеством одновременно доступных периферийных устройств. Распределение пропускной способности шины между подключенными устройствами планируется хостом и реализуется им с помощью посылки маркеров. Шина позволяет подключать, конфигурировать, использовать и отключать устройства во время работы хоста и самих устройств - динамическое ("горячее") подключение и отключение.

Устройства (Device) USB могут являться хабами, "функциями" или их комбинацией:

- Хаб (Hub) обеспечивает дополнительные точки подключения устройств к шине.
- "Функции" (Function) USB предоставляют системе дополнительные возможности - например, подключение к ISDN,

цифровой джойстик, акустические колонки с цифровым интерфейсом и т.д.

Устройство USB должно иметь интерфейс USB, обеспечивающий поддержку протокола USB, выполнение стандартных операций (конфигурирование и сброс) и стандартное представление информации, описывающей устройство. Многие устройства, подключаемые к USB, имеют в своем составе и "функции" и хабы.

Работой всей системы USB управляет хост-контроллер, являющийся программно-аппаратной подсистемой хост-компьютера.

Физическое соединение устройств осуществляется по топологии многоярусной звезды. Центром каждой звезды является хаб, каждый кабельный сегмент соединяет две точки - хаб с другим хабом или хаб с функцией. В системе USB имеется только один хост-контроллер, расположенный в вершине пирамиды устройств и хабов USB. Хост-контроллер интегрируется с корневым хабом (root hub), обеспечивающим одну или несколько точек подключения - портов. Контроллер USB, входящий в состав чипсетов многих современных системных плат обычно имеет двухпортовый хаб.

Логически устройство подключенной к любому хабу и сконфигурированное может рассматриваться как подключенное напрямую к хост-контроллеру.

"Функции" представляют собой устройства USB, способные принимать или передавать данные или управляющую информацию по шине. Физически в одном корпусе может быть несколько "функций" со встроенным хабом обеспечивающим их подключение к одному порту.

Каждая "функция" предоставляет конфигурационную информацию, описывающую его возможности и требования к ресурсам. Перед использованием функция должна быть сконфигурирована хостом - ей должна быть выделена полоса в канале, выбраны специфические опции конфигурации.

Хаб - ключевой элемент системы Plug-and-Play в архитектуре USB. Хаб является кабельным концентратором, точки подключения называются портами хаба. Каждый хаб преобразует одну точку подключения в их множество. Архитектура подразумевает возможность соединения нескольких хабов.

У каждого хаба имеется один восходящий порт (upstream port), предназначенный для подключения к хосту и ли к хабу верхнего уровня. Остальные порты являются нисходящими (downstream) и предназначены для подключения функций и хабов нижнего уровня. Хаб может распознать подключение или отключение устройств к этим портам и управлять подачей питания на их сегменты. Каждый из этих портов индивидуально может быть разрешен или запрещен и сконфигурирован на полную или



ограниченную скорость обмена. Хаб обеспечивает изоляцию сегментов с низкой скоростью от высокоскоростных.

Хабы могут иметь возможность управления подачей питания на нисходящие порты, предусмотрена управляемая установка ограничения на ток, потребляемый каждым портом.

Система USB разделяется на три уровня с определенными правилами взаимодействия. Устройство USB делится на интерфейсную часть, логическую часть устройства и функциональную часть. Хост тоже делится на три части - интерфейсную, системную и ПО устройства. Каждая часть отвечает только за определенный круг задач, взаимодействие между ними показано на рисунке 1.3.

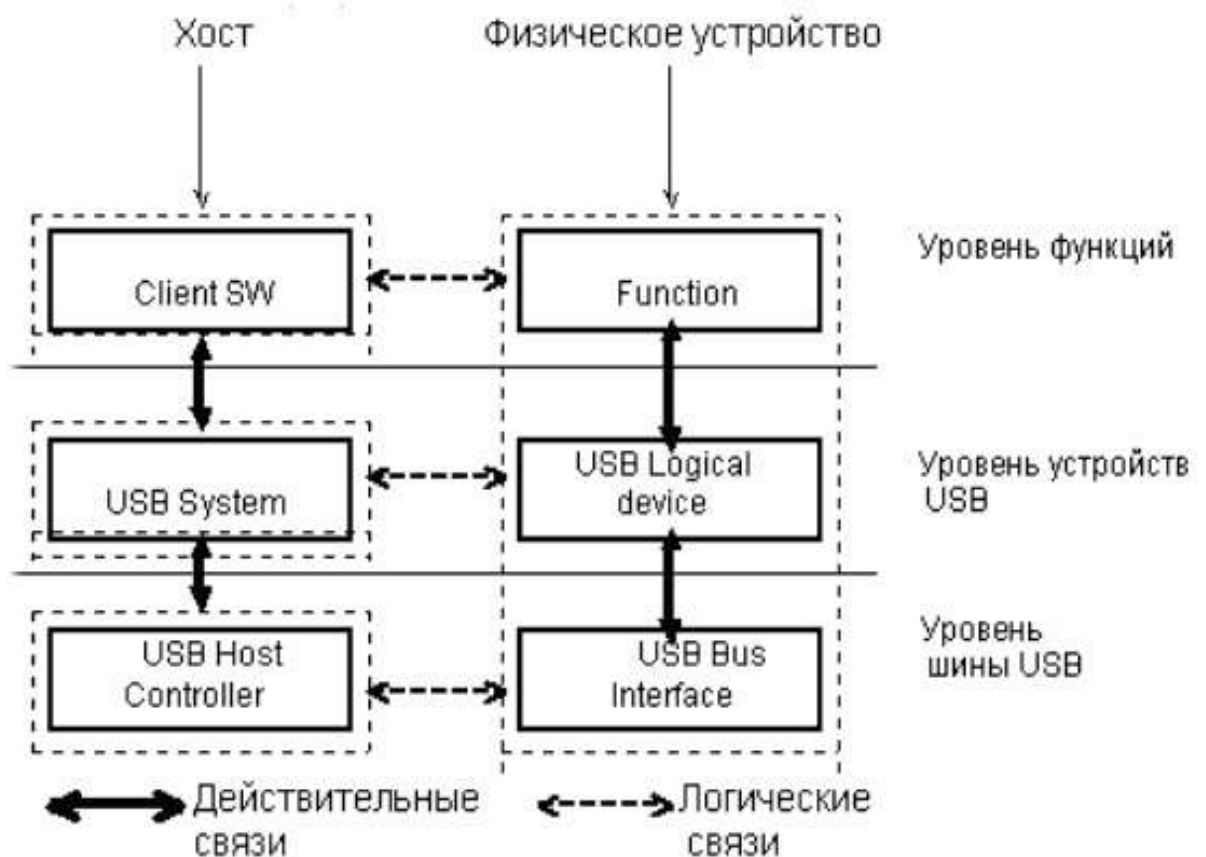


Рис. 1.3. Взаимодействие компонентов USB

1. Физическое устройство USB - устройство на шине, выполняющее функции, интересующие пользователя.
2. Client SW - программное обеспечение, соответствующее конкретному устройству, исполняемое на хост-компьютере. Может являться составной частью ОС или специальным продуктом.
3. USB System SW - системная поддержка USB операционной системой, независимая от конкретных устройств и клиентского ПО.

USB Host Controller - аппаратные и программные средства, обеспечивающие подключение устройств USB к хост-компьютеру.

### 1.1.2 ФИЗИЧЕСКИЙ ИНТЕРФЕЙС

Информационные сигналы и питающее напряжение 5В передаются по четырехпроводному кабелю. Для сигнала используется дифференциальный способ передачи по двум проводам D+ и D-. Уровни сигналов передатчиков в статическом режиме должны быть ниже 0.3 В (низкий уровень) или выше 2.8 В (высокий уровень). Приемники должны выдерживать входное напряжение в пределах -0.5...+3.8 В. Передатчики должны иметь возможность перехода в высокоимпедансное состояние для обеспечения двунаправленной полудуплексной передачи данных по одной паре проводов.

Передача по двум проводам не ограничивается лишь дифференциальными сигналами. Кроме дифференциального приемника, каждое устройство имеет и линейные приемники сигналов D+ и D- , а передатчики этих линий управляются индивидуально. Это позволяет различать множество состояний линии, используемых для организации аппаратного интерфейса. Состояния Diff0 и Diff1 определяются по разности потенциалов на линиях D+ и D- более 200 мВ при условии, что на одной из них потенциал выше порога срабатывания VSE. Состояние, при котором на обоих входах D+ и D- присутствует низкий уровень, называется линейным нулем (SE0 - single-ended zero). Интерфейс определяет следующие состояния:

- Data J State и Data K State - состояния передаваемого бита (определяются через состояния Diff0 и Diff1).
- Idle State - пауза на шине.
- Resume State - сигнал "пробуждения" для вывода устройства из спящего режима.
- Start of Packet (SOP) - начало пакета (переход из "Idle" в "K").
- End of Packet (EOP) - конец пакета.
- Disconnect - устройство отключено от порта.
- Connect - устройство подключено к порту.
- Reset - сброс устройства.

Состояния определяются сочетаниями дифференциальных и линейных сигналов, для полной и низкой скоростей состояния Diff0 и Diff1 имеют противоположное назначение. В декодировании состояние Disconnect, Connect и Reset принимается во внимание и время нахождения линий (более 2.5 мс) в определенных состояниях.

Шина имеет два режима передачи. Полная скорость передачи сигналов USB составляет 12 Мбит/с, низкая - 1.5 Мбит/с. Для полной скорости используется экранированная витая пара с импедансом 90 Ом и

длиной сегмента до 5 м, для низкой - невитой и неэкранированный кабель при длине сегмента до 3 м. Одна и та же система может использовать оба режима, переключение для устройств осуществляется прозрачно. Низкая скорость предназначена для работы с небольшим количеством устройств, не требующих высокой пропускной способности канала.

Скорость, используемая устройством, подключенным к конкретному порту, определяется хабом по уровням сигналов на линиях D+ и D-, смещаемых нагрузочными резисторами R2 приемопередатчиков (рисунки 1.4, 1.5). Сигналы кодируются по методу NRZI (Non Return To Zero Invert) - при переходе сигнала из 0 в 1 сигнал NRZI не изменяется, а при переходе из 1 в 0 - изменяется на противоположный. Каждому пакету предшествует поле SYNC, позволяющее приемнику настроиться на частоту передатчика.

Кроме сигнальной пары кабель имеет линии VBus и GND для передачи питающего напряжения 5В к устройствам.

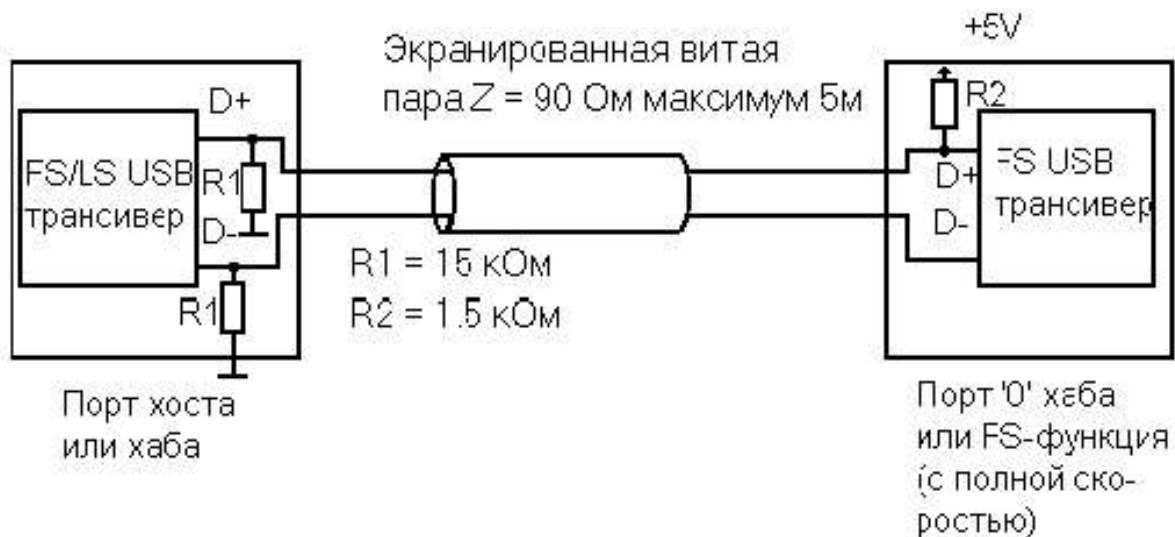


Рис. 1.4. Подключение полноскоростного устройства

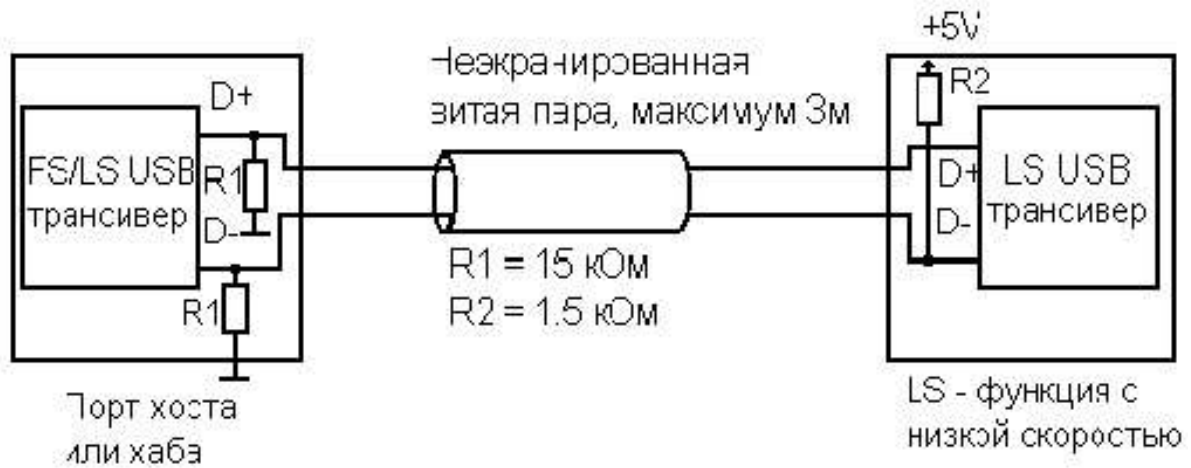


Рис. 1.5. Подключение низкоскоростного устройства

Табл. 1.1. Назначение выводов разъема USB

Контакт	Цепь
1	VCC
2	-Data
3	+Data
4	Ground

Разъемы для подключения к хабам и для подключения к устройствам различаются механически, что исключает возможность неверного соединения.

Для облегчения распознавания разъема USB на корпусе устройства ставится обозначение, приведенное на рисунке 1.6.



Рис 1.6. Обозначение разъема USB

Питание устройства USB возможно как от кабеля, так и от собственного блока питания.

Хост обеспечивает питанием непосредственно подключенные к нему устройства. Каждый хаб обеспечивает питание устройств, подключенных к его нисходящим портам.

USB имеет развитую систему управления энергопотреблением. Хост-компьютер может иметь собственную систему управления энергопотреблением, к которой логически подключается одноименная система USB. Программное обеспечение USB взаимодействуя с этой системой, поддерживает такие события как приостанов (SUSPEND) или восстановление (RESUME). Кроме того, устройства USB могут сами являться источниками событий, обрабатываемых системой управления энергопотреблением.

### 1.1.3 СИСТЕМНОЕ КОНФИГУРИРОВАНИЕ

USB поддерживает подключение и отключение устройств во время работы шины. Нумерация устройств шины является постоянным процессом, отслеживающим динамические изменения физической топологии.

Все устройства USB подключаются через порты хабов. Хабы определяют подключение и отключение устройств к своим портам и сообщают состояние портов в ответ на запрос от контроллера. Хост разрешает работу порта и адресует к устройству через канал управления, используя свой нулевой адрес - USB default address. Все устройства адресуются этим адресом при начальном подключении или после сброса.

Хост определяет, является подключенное устройство хабом или функцией и назначает ему уникальный адрес USB. Хост устанавливает с этим устройством канал управления, используя назначенный адрес и нулевой номер точки назначения.

Если новое устройство является хабом, хост определяет подключенные к нему устройства, устанавливает каналы и назначает для них адреса. Если новое устройство является "функцией" уведомление о подключении передается диспетчером USB соответствующему ПО.

Когда устройство отключается, хаб автоматически запрещает использование соответствующего порта и сообщает об отключении контроллеру, который удаляет сведения о данном устройстве из всех структур данных. Если отключается хаб, то процесс удаления повторяется для всех подключенных к нему устройств.

Нумерация устройств, подключенных к шине, осуществляется динамически по мере подключения или отключения их питания без

какого-либо вмешательства пользователя или клиентского ПО. Процедура нумерации выполняется следующим образом:

1. Хаб, к которому подключилось устройство, информирует хост о смене состояния своего порта ответом на опрос состояния. С этого момента устройство переходит в состояние "Attached" ("присоединено"), а порт, к которому оно присоединено, в состояние "Disabled".

2. Хост уточняет состояние порта.

3. Узнав порт, к которому подключилось новое устройство, хост дает команду сброса и разрешения порта.

4. Хаб формирует сигнал RESET для данного порта (10 мс) и переводит его в состояние "Enabled". Подключенному устройству позволяется потреблять от шины ток питания в пределах 100 мА. Устройство переходит в состояние Powered, все его регистры переводятся в исходное состояние, и оно отзывается на обращение по нулевому адресу.

5. До тех пор пока устройство не получит уникальный адрес, оно доступно по дежурному каналу, по которому хост-контроллер может определять максимально допустимый размер поля данных пакета.

6. Хост сообщает устройству его уникальный адрес, и оно переходит в состояние "Addressed".

7. Хост считывает все конфигурации устройства, включая и заявленный ток потребления от шины.

8. Исходя из полученной информации, хост конфигурирует все имеющиеся конечные точки данного устройства, которое переводится в состояние Configured. Теперь хаб позволяет устройству потреблять от шины полный ток, заявленный в конфигурации. С точки зрения устройства оно становится готовым к использованию.

Когда устройство отделяется от шины, хаб уведомляет об этом хост и работа порта запрещается, а хост обновляет свою текущую топологическую информацию.

#### 1.1.4 USB 2.0

В октябре 1999 года разработчики аппаратных средств, ранее опубликовавшие спецификацию USB 1.1 (Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC и Philips), представили спецификацию USB 2.0, в которой предусмотрено повышение быстродействия шины в 40 (480 Мбит/с) раз по сравнению с предыдущими версиями. USB 2.0 полностью совместима с USB 1.1, и использует те же самые кабели и соединители. Ранее объявлялось повышение быстродействия в 10 - 20 раз, но испытания показали, что пропускная способность 480 Мбит/с может быть достигнута без ущерба для совместимости с версией USB 1.1.

Пропускной способности 12 Мбит/с вполне хватает таким периферийным устройствам как телефоны, клавиатуры, мыши, цифровые

джойстики, приводы гибких дисков, цифровые колонки, и принтеры нижнего уровня. Возможность подключения этой периферии в USB 2.0 сохранилась. Более высокая полоса пропускания позволила использовать современные устройства, такие как: видеокамеры высокого разрешения, сканеры и принтеры, скоростные внешние накопители.

Для применения в системах требующих экономного расхода электроэнергии (таких как ноутбуки и т.д) в USB 2.0 как и в USB 1.1 предусмотрена мощная система управления питанием, что, как ожидается, откроет для USB 2.0 рынок мобильных компьютеров.

Для USB 2.0 существует лишь один солидный конкурент, это IEEE 1394, но как отмечается некоторыми обозревателями 1394 это хорошая шина, выброшенная разработчиками на произвол судьбы в то время, как USB 2.0 усиленно поддерживается и проталкивается на рынок такими гигантами как Compaq, Hewlett-Packard, Intel, Microsoft и т.д.

## 1.2 МИКРОСХЕМА FT245BM: ПРЕОБРАЗОВАТЕЛЬ ИНТЕРФЕЙСА USB В ПАРАЛЛЕЛЬНЫЙ FIFO

Английская компания Future Technology Devices Int. выпускает микросхемы преобразователей интерфейсов. В линейку таких многофункциональных микросхем входят популярные модели FT8U232AM (FT232AM), FT232BM, FT8U245AM (FT245AM), FT245BM, FT8U100AX.

FT245BM – микросхема второго поколения, предназначенная для преобразования USB протокола в параллельный FIFO. По сравнению с предыдущим поколением (FT8U245AM), данная микросхема отличается повышенной функциональностью, что позволяет сократить количество используемых внешних компонентов. Кроме того, FT245BM имеет высокую степень совместимости с предыдущими версиями, что делает переход на новые изделия быстрым и удобным, одновременно расширяя области применения изделий.

Микросхема FT245BM является эффективным и недорогим способом организации передачи данных от периферии к головному компьютеру и обратно со скоростью до 8 Мбит/с (1 Мбайт/с). Ее простой дизайн, представляющий собой буфер FIFO, позволяет организовывать взаимодействие с практически любым микроконтроллером или микропроцессором через порты входа/выхода.

Передача данных от периферийных устройств осуществляется по восьмиразрядной параллельной двунаправленной шине данных (D0-D7). Для передачи данных от периферии к головному компьютеру достаточно записать байт данных в буфер при неактивном бите состояния. Если объем буфера (384 байта) переполнен или идет процесс сохранения предыдущего байта, устройство приостановит запись, уровень сигнала TXE останется высоким, пока информация не будет передана по USB на компьютер.

При передаче данных от головного устройства периферийному по USB, микросхема сообщит периферийному устройству о наличии в приемнике данных для передачи. Чтение данных из буфера FIFO продолжается до тех пор, пока бит активен.

Компания-производитель предлагает использовать бесплатные виртуальные драйвера COM порта. Периферийное устройство при этом выглядит как стандартный последовательный порт, но скорость передачи данных поддерживается на максимальном уровне.



#### Описание микросхемы FT245BM:

- Интегрированная микросхема двусторонней передачи данных USB - параллельный FIFO
- Скорость передачи данных 1 Мбайт/с (при использовании драйверов D2XX)
- Скорость передачи данных до 300 кбайт/с (при использовании виртуальных драйверов VCP)
- Легкость подключения к микроконтроллерам, ПЛЮ, FPGA через 4-проводной синхронизирующий интерфейс
- Интегрированный на чипе USB протокол, не требующий специального программирования
- 384 байт буфер передатчика FIFO
- 128 байт буфер приемника FIFO
- Регулируемая задержка буфера передатчика
- Встроенный преобразователь уровней для FIFO и контрольных сигналов для взаимодействия с 5 и 3.3 В логикой
- 3.3 В регулятор для входа/выхода USB
- Схема перезапуска питания
- 6 – 48 МГц ФАПЧ
- Питание 4.35...5.25 В
- Совместимость с головными контроллерами UHCI / OHCI / EHCI
- Совместимость с USB 1.1 и USB 2.0
- EEPROM программирование через USB

Микросхемы FT245BM могут применяться в USB ISDN и ADSL модемах, карт ридерах, КПК, в цифровых камерах и MP3 плеерах и другой цифровой технике для организации высокоскоростного обмена данными.

### 1.3. ОПИСАНИЕ МИКРОКОНТРОЛЛЕРА ATMEGA128

ATmega128 – маломощный 8-разр. КМОП микроконтроллер, основанный на расширенной AVR RISC-архитектуре. За счет выполнения большинства инструкций за один машинный цикл ATmega128 достигает производительности 1 млн. операций в секунду, что позволяет проектировщикам систем оптимизировать соотношение энергопотребления и быстродействия.

Отличительные особенности:

- Высокопроизводительный, маломощный 8-разрядный AVR-микроконтроллер.
- Развитая RISC-архитектура:
  - 133 мощных инструкций, большинство из которых выполняются за один машинный цикл.
  - 32 8-разр. регистров общего назначения + регистры управления встроенной периферией.
  - Полностью статическая работа.
  - Производительность до 16 млн. операций в секунду при тактовой частоте 16 МГц.
  - Встроенное умножающее устройство выполняет умножение за 2 машинных цикла.
- Энергонезависимая память программ и данных:
  - Износостойкость 128-ми кбайт внутрисистемно перепрограммируемой флэш-памяти: 1000 циклов запись/стирание.
  - Опциональный загрузочный сектор с отдельной программируемой защитой.  
Внутрисистемное программирование встроенной загрузочной программой.  
Гарантированная двухоперационность: возможность чтения во время записи.
  - Износостойкость 4 кбайт ЭСППЗУ: 100000 циклов запись/стирание.
  - Встроенное статическое ОЗУ емкостью 4 кбайт.
  - Опциональная возможность адресации внешней памяти размером до 64 кбайт.
  - Программируемая защита кода программы.
  - Интерфейс SPI для внутрисистемного программирования.
- Интерфейс JTAG (совместимость со стандартом IEEE 1149.1):
  - Граничное сканирование в соответствии со стандартом JTAG.
  - Обширная поддержка функций встроенной отладки.

- Программирование флэш-памяти, ЭСППЗУ, бит конфигурации и защиты через интерфейс JTAG.
- Отличительные особенности периферийных устройств:
  - Два 8-разр. таймера-счетчика с отдельными предделителями и режимами сравнения.
  - Два расширенных 16-разр. таймера-счетчика с отдельными предделителями, режимами сравнения и режимами захвата.
  - Счетчик реального времени с отдельным генератором.
  - Два 8-разр. каналов ШИМ.
  - 6 каналов ШИМ с программируемым разрешением от 2 до 16 разрядов.
  - Модулятор выходов сравнения.
  - 8 мультиплексированных каналов 10-разрядного аналогово-цифрового преобразования:
    - 8 несимметричных каналов
    - 7 дифференциальных каналов
    - 2 дифференциальных канала с выборочным усилением из 1х, 10х и 200х
  - Двухпроводной последовательный интерфейс, ориентированный на передачу данных в байтном формате
  - Два канала программируемых последовательных УСАПП
  - Последовательный интерфейс SPI с поддержкой режимов ведущий/подчиненный
  - Программируемый сторожевой таймер со встроенным генератором
  - Встроенный аналоговый компаратор
- Специальные возможности микроконтроллера
  - Сброс при подаче питания и программируемая схема сброса при снижении напряжения питания
  - Встроенный калиброванный RC-генератор
  - Внешние и внутренние источники прерываний
  - Шесть режимов снижения энергопотребления: холостой ход (Idle), уменьшение шумов АЦП, экономичный (Power-save), выключение (Power-down), дежурный (Standby) и расширенный дежурный (Extended Standby)
  - Программный выбор тактовой частоты
  - Конфигурационный бит для перевода в режим совместимости с ATmega103
  - Общее выключение подтягивающих резисторов на всех линиях портов ввода-вывода

- Ввод-вывод и корпуса
  - 53 – программируемые линии ввода-вывода
  - 64-выв. корпус TQFP
- Рабочие напряжения
  - 2.7 - 5.5В для ATmega128L
  - 4.5 - 5.5В для ATmega128
- Градации по быстродействию
  - 0 - 8 МГц для ATmega128L
  - 0 - 16 МГц для ATmega128

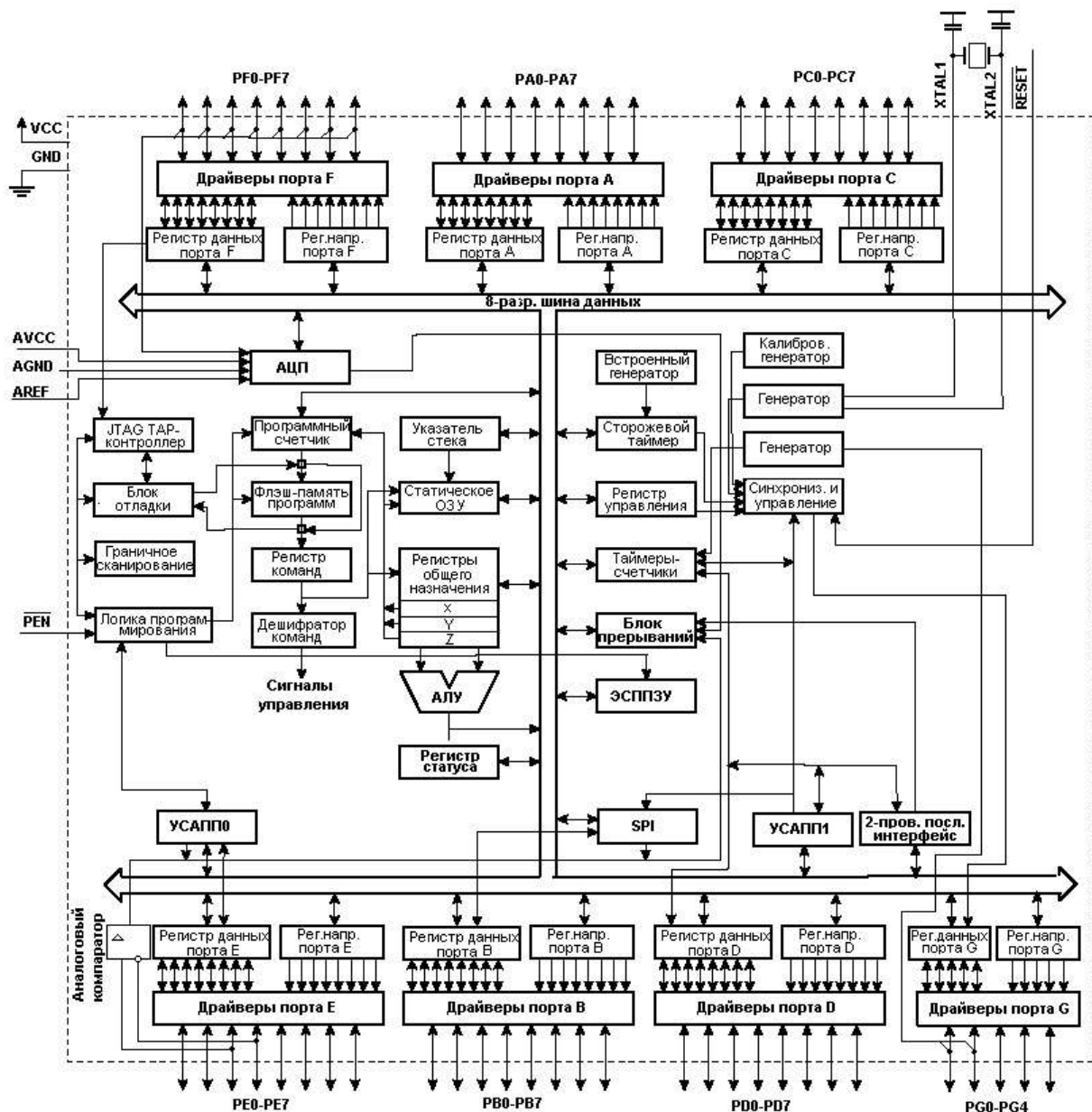


Рис. 1.7. Функциональная схема

Ядро AVR сочетает богатый набор инструкций с 32 универсальными рабочими регистрами. Все 32 регистра непосредственно подключены к арифметико-логическому устройству (АЛУ), который позволяет указать два различных регистра в одной инструкции и выполнить ее за один цикл. Данная архитектура обладает большей эффективностью кода за счет достижения производительности в 10 раз выше по сравнению с обычными CISC-микроконтроллерами.

ATmega128 содержит следующие элементы (рисунок 1.7): 128 кбайт внутрисистемно программируемой флэш-памяти с поддержкой чтения во время записи, 4 кбайт ЭСППЗУ, 4 кбайт статического ОЗУ, 53 линии универсального ввода-вывода, 32 универсальных рабочих регистра, счетчик реального времени (RTC), четыре гибких таймера-счетчика с режимами сравнения и ШИМ, 2 УСАПП, двухпроводной последовательный интерфейс ориентированный на передачу байт, 8-канальный 10-разр. АЦП с опциональным дифференциальным входом с программируемым коэффициентом усиления, программируемый сторожевой таймер с внутренним генератором, последовательный порт SPI, испытательный интерфейс JTAG совместимый со стандартом IEEE 1149.1, который также используется для доступа к встроенной системе отладки и для программирования, а также шесть программно выбираемых режимов уменьшения мощности. Режим холостого хода (Idle) останавливает ЦПУ, но при этом поддерживая работу статического ОЗУ, таймеров-счетчиков, SPI-порта и системы прерываний. Режим выключения (Powerdown) позволяет сохранить содержимое регистров, при остановленном генераторе и выключении встроенных функций до следующего прерывания или аппаратного сброса. В экономичном режиме (Power-save) асинхронный таймер продолжает работу, позволяя пользователю сохранить функцию счета времени в то время, когда остальная часть контроллера находится в состоянии сна. Режим снижения шумов АЦП (ADC Noise Reduction) останавливает ЦПУ и все модули ввода-вывода, кроме асинхронного таймера и АЦП для минимизации импульсных шумов в процессе преобразования АЦП. В дежурном режиме (Standby) кварцевый/резонаторный генератор продолжают работу, а остальная часть микроконтроллера находится в режиме сна. Данный режим характеризуется малой потребляемой мощностью, но при этом позволяет достичь самого быстрого возврата в рабочий режим. В расширенном дежурном режиме (Extended Standby) основной генератор и асинхронный таймер продолжают работать.

Микроконтроллер производится по технологии высокоплотной энергонезависимой памяти компании Atmel. Встроенная внутрисистемно

программируемая флэш-память позволяет перепрограммировать память программ непосредственно внутри системы через последовательный интерфейс SPI с помощью простого программатора или с помощью автономной программы в загрузочном секторе. Загрузочная программа может использовать любой интерфейс для загрузки прикладной программы во флэш-память. Программа в загрузочном секторе продолжает работу в процессе обновления прикладной секции флэш-памяти, тем самым поддерживая двухоперационность: чтение во время записи. За счет сочетания 8-разр. RISC ЦПУ с внутрисистемно-самопрограммируемой флэш-памятью в одной микросхеме ATmega128 является мощным микроконтроллером, позволяющим достичь высокой степени гибкости и эффективной стоимости при проектировании большинства приложений встроенного управления.

ATmega128 поддерживается полным набором программных и аппаратных средств для проектирования, в т.ч.: Си-компиляторы, макроассемблеры, программные отладчики/симуляторы, внутрисистемные эмуляторы и оценочные наборы.

## 2. ОСНОВНЫЕ ВАРИАНТЫ РЕАЛИЗАЦИИ ИНТЕРФЕЙСА USB В МИКРОКОНТРОЛЛЕРНЫХ УСТРОЙСТВАХ

Основные варианты реализации интерфейса USB в микроконтроллерных устройствах – на микроконтроллерах, имеющих поддержку интерфейса на аппаратном уровне, и с применением микросхем реализующих преобразование интерфейсов:

- с помощью микроконтроллера, у которого интерфейс USB реализован аппаратно. В этом случае необходимо написать программу для микроконтроллера, реализующую обмен по USB. Кроме того, если операционная система не поддерживает стандартные классы USB или они не удовлетворяют требованиям разработчика, то необходимо написать драйвер обеспечивающий взаимодействие с USB;
- использование универсального преобразователя интерфейсов. В этом случае разработка специальной прошивки не потребуется, нет надобности разбираться в тонкостях работы интерфейса USB и нет необходимости написания драйвера, т.к. производитель преобразователя предлагает свой драйвер. Недостаток – увеличенные габариты готового изделия. Основная проблема такого подхода заключается в сложности достижения высокой скорости обмена. В данной НИРС исследование начато с рассмотрения этого подхода, в рамках же магистерской работы будут проанализированы оба варианта.

Для реализации интерфейса USB выпускается широкий ассортимент микросхем, различающихся скоростями обмена (LS, FS или HS), числом и возможностями конечных точек (тип передач, размер буфера).

Производители наиболее распространённых микроконтроллеров имеют в линейке своей продукции семейство микроконтроллеров с аппаратной поддержкой интерфейса USB. Рассмотрим ассортимент представленных на украинском рынке фирм: Atmel, Microchip, FTDI.

### **Atmel.**

С портом USB фирма Atmel выпускает микроконтроллеры на ядре MCS51, AVR и ARM. Они различаются объёмом памяти (оперативной и энергонезависимой), производительностью, напряжением питания, потреблением, наличием встроенных устройств АЦП/ЦАП и т.д. Таким образом, разработчик всегда может подобрать себе микроконтроллер, имеющий в своём составе все необходимые ему встроенные устройства. Микроконтроллеры с интегрированным интерфейсом USB фирмы Atmel с архитектурой MCS51 приведены в таблице 2.1. Микроконтроллеры с интегрированным интерфейсом USB фирмы Atmel с ядром AVR приведены в таблице 2.2. Данные микроконтроллеры содержат интегрированный приёмопередатчик, соответствующий спецификации шины USB 2.0, FIFO – буфер, поддерживают одновременную работу с

несколькими устройствами. Для работы интерфейса USB используется единый блок генерации тактового сигнала – один кварц для ядра микроконтроллера и USB. Также следует отметить малое число внешних компонентов для реализации интерфейса.

Данные микроконтроллеры наряду с поддержкой спецификации USB содержат большое количество разнообразных интегрированных периферийных устройств.

Микроконтроллеры поддерживаются полным набором программ и средствами для проектирования, в т.ч.: Си компиляторы, макроассемблеры, программные отладчики/симуляторы, внутрисхемные эмуляторы и оценочные наборы. Для разработчиков на базе микроконтроллеров с интерфейсом USB разработана мезонинная плата STK525, снабжённая набором полезных программ и примеров в исходных кодах.

Табл. 2.1. Микроконтроллеры с архитектурой MCS51

Тип	Характеристики							Функции
	интерфейс	ядро	Flash, Кб	EEPROM, Кб	RAM, байт	напряжение питания, В	количество оконечных устройств	
AT89C5130A-M	USB	C51	16	1	1280	2,7...5,5	7	TWI, SPI, UART, PCA
AT89C5131A-L	USB	C51	32	1	1280	3,0...3,6	7	TWI, SPI, UART, PCA
AT89C5131A-M	USB	C51	32	1	1280	2,7...5,5	7	TWI, SPI, UART, PCA
AT89C5132	USB	C51	64		2304		4	SPI, UART, IDE, TWI, MultiMediaCard, DataFlash, I2S, 10-разрядный АЦП

Табл. 2.2. Микроконтроллеры с ядром AVR

Тип	Характеристики							
	USB	Самопрограммируемая Flash-память программ, Кб	SRAM, Кб	EEPROM Кб	АЦП	MIPS	напряжение питания, В	JTAG для внутрисхемного программирования и отладки
AT90USB1286	Есть	128	8	4	8-канальный 10-разрядный	До 16 при 16 МГц	2,7...5,5	Есть
AT90USB1287	USB On-The-Go функция	128	8	4	8-канальный 10-разрядный	До 16 при 16 МГц	2,7...5,5	Есть
AT90USB646	Есть	64	4	2	8-канальный 10-разрядный	До 16 при 16 МГц	2,7...5,5	Есть
AT90USB647	USB On-The-Go функция	64	4	2	8-канальный 10-разрядный	До 16 при 16 МГц	2,7...5,5	Есть

Микроконтроллеры с интегрированным интерфейсом USB фирмы Atmel AT91 с ядром ARM приведены в таблице 2.3. Данные высокопроизводительные микроконтроллеры с большим числом разнообразных интегрированных периферийных устройств (USB 2.0 Full Speed порт, USARTs, SPI, SSC, TWI, АЦП, DMA-контроллер) являются идеальным решением для перехода с 8-разрядных микроконтроллеров для



увеличения производительности, увеличения объема интегрированной памяти и реализации USB 2.0 Full Speed.

Таблица 3. Микроконтроллеры AT91 с ядром ARM

Тип	Характеристики										
	Flash, Кб	SRAM, Кб	USB Host (Full Speed)	USB Device (Full Speed)	10-разрядный АЦП (каналов)	F <sub>max</sub> МГц	ШИМ, каналов	Число линий ввода/вывода	Число 16-разрядных таймеров	SPI	Ядро
AT91RM3400	-	96	-	1	-	66	-	63	6	1	ARM7TDMI
AT91RM9200	-	16	2	1	-	180	-	94	6	1	ARM920T
AT91SAM7A3	256	32	-	1	16	60	8	62	9	2	ARM7TDMI
AT91SAM7S128	128	32	-	1	8	55	4	32	3	1	ARM7TDMI
AT91SAM7S256	256	64	-	1	8	55	4	32	3	1	ARM7TDMI
AT91SAM7S321	32	8	-	1	8	55	4	32	3	1	ARM7TDMI
AT91SAM7S64	64	16	-	1	8	55	4	32	3	1	ARM7TDMI
AT91SAM7SE256	256	32	-	1	8	48	4	88	3	1	ARM7TDMI
AT91SAM7SE32	32	8	-	1	8	48	4	32	3	1	ARM7TDMI
AT91SAM7SE512	512	32	-	1	8	48	4	88	3	1	ARM7TDMI
AT91SAM7X128	128	32	-	1	8	55	4	60	3	2	ARM7TDMI
AT91SAM7X256	256	64	-	1	8	55	4	60	3	2	ARM7TDMI
AT91SAM7XC128	128	32	-	1	8	55	4	60	3	2	ARM7TDMI
AT91SAM7XC256	256	64	-	1	8	55	4	60	3	2	ARM7TDMI
AT91SAM9260	-	2*4	2	1	4	180	-	96	6	2	ARM926EJ-S
AT91SAM9261	-	160	2	1	-	180	-	96	3	2	ARM926EJ-S

### Microchip.

Фирма Microchip Technology выпускает микроконтроллеры PIC16C745/765, которые имеют встроенный периферийный модуль USB, соответствующий спецификации USB v1.1. Модуль USB поддерживает передачи управления и прерывания для скорости 1,5 Мбит/с. PIC16C745 – микроконтроллер с архитектурой PIC. Имеет встроенную USB-функцию, работающую в LS-режиме, 256 байт ОЗУ, 14 336 байт ПЗУ, 22 программируемых вывода, последовательный интерфейс, 5-канальный 8-битный АЦП. PIC16C765 отличается тем, что имеет 33 программируемых вывода и 8-канальный 8-битный АЦП. Данные микроконтроллеры предназначены для реализации устройств, не требующих высоких скоростей передачи данных, – разнообразных устройств управления и автоматики, систем сбора данных с датчиков и т.п.

Микроконтроллеры с поддержкой USB High Speed представлены в таблице 2.4. Данные микроконтроллеры сочетают в себе интегрированный USB интерфейс и производительный контроллер PIC18F. Данное решение позволяет не вникая в тонкости работы USB-интерфейса перевести устройства с порта RS-232 на современный высокоскоростной USB.

Для помощи разработчикам в реализации проектов фирма Microchip Technology предоставляет набор библиотечных функций, поддерживающих USB-интерфейс. Эти библиотеки позволяют освободить основное программное обеспечение от необходимости обрабатывать сложный протокол USB. Выполнение большинства функций USB обеспечивается просто вызовом интерфейсных функций Put/Get (передать/принять).

Библиотеки также обеспечивают приложения операциями при нумерации и конфигурировании.

Кроме того, компания Microchip предлагает разнообразные демонстрационные и отладочные средства для работы с интерфейсом USB.

Таблица 2.4. Микроконтроллеры с поддержкой USB High Speed

Тип	Характеристики									
	Память программ, Кб	Память данных, байт		Число линий ввода/вывода	АЦП	ШИМ (ССР/ЕСРР)	SPI	I2C	Компараторы	Таймеры 8/16 бит
		RAM	EEPROM							
PIC18F2455	24	2048	256	24	10	2/0	+	+	2	1/3
PIC18F2550	32	2048	256	24	10	2/0	+	+	2	1/3
PIC18F4455	24	2048	256	35	13	1/1	+	+	2	1/3
PIC18F4550	32	2048	256	35	13	1/1	+	+	2	1/3
PIC18F2450	16	768	-	24	10	1/0	-	-	-	1/2
PIC18F4450	16	768	-	35	13	1/0	-	-	-	1/2

### FTDI.

Компания FTDI (Future Technology Devices International) специализируется на производстве микросхем сопряжения микропроцессорных устройств с USB. Решения FTDI позволяют организовать обмен данными по USB простейшим образом благодаря аппаратной реализации протокола и наличию бесплатных драйверов для Windows 98/2000/ME/XP, MAC OS X, Linux.

Микросхемы FTDI являются мостом между USB и такими микропроцессорными интерфейсами, как UART, FIFO, JTAG, SPI, I2C и т.д. В режиме Bit Bang они могут использоваться для конфигурирования микросхем программируемой логики через USB или для ввода/вывода по USB цифровых логических сигналов без использования дополнительного микроконтроллера.

Использование микросхем FTDI является идеальным решением для быстрой замены интерфейса RS232 на современный USB в разработанных устройствах. Использование компонентов FTDI не требует от разработчика глубоких знаний интерфейса USB благодаря наличию на сайте производителя драйверов, богатого инструментария разработчика и примеров.

Продукция компании представлена рядом микросхем. Хост-контроллер Vinculum имеет два порта USB, один из которых может быть сконфигурирован как хост, другой – как периферийный (slave). Кроме того, микросхема имеет дополнительные порты ввода/вывода, которые могут работать в режимах SPI, UART или параллельного 8-разрядного интерфейса. Микросхема FT232R является преобразователем USB – UART с интегрированным внутренним тактовым генератором, который можно использовать в качестве задающего для внешних устройств, при этом значение выходной частоты может быть задано равным 6, 12, 24 или 48 МГц. Также FT232R имеет интегрированную энергонезависимую память

EEPROM и пассивные компоненты, интегрированные в кристалл. Таким образом, для подключения данной микросхемы к микроконтроллеру практически не потребуется дополнительных элементов обвязки.

Также следует отметить наличие уникального идентификационного номера (FTDIChip-IDT), который программируется производителем в процессе изготовления кристаллов и доступен для чтения по шине USB.

Существует несколько типов драйверов для преобразователей интерфейсов FTDI, разработанных непосредственно FTDI или другими компаниями. Это VCP-драйверы: при подключении к хосту устройств на базе FT2XX каждому из них может назначаться свой виртуальный COM порт либо могут использоваться D2XX Direct Drivers. Архитектура D2XX-драйвера состоит из Windows WDM-драйвера, работающего с устройством посредством Windows USB стека и DLL, который связывает приложение (написанное на Visual C++, C++ Builder, Delphi, VB и т.д.) с WDM-драйвером.

Особенностью драйверов является их максимальная простота при установке и использовании, доступность (все они выложены на сайте компании) и широкое разнообразие, обеспечивающее возможность применения практически на любой операционной системе. Все драйверы сопровождаются руководствами и техническими описаниями.

Эти драйверы вместе с микросхемами преобразователей интерфейсов разработаны для тех случаев, когда необходимо быстро и без особых затрат сделать свою продукцию совместимой с USB, сохраняя при этом совместимость и с ранее разработанным программным обеспечением под COM-порт. При установке драйвера в операционной системе добавляется виртуальный COM-порт (дополнительно к существующим аппаратно), и программное обеспечение обращается к USB-устройству так же, как к стандартному COM-порту, используя стандартные вызовы VCOMM API или с использованием библиотек виртуального порта.

### 3. РЕАЛИЗОВАННЫЕ ЭЛЕМЕНТЫ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

В рамках данной НИРС разработана функциональная схема устройства сопряжения и программный модуль управления работой микросхемой FT245BM. Схема имеет следующий вид (рисунок 3.1):

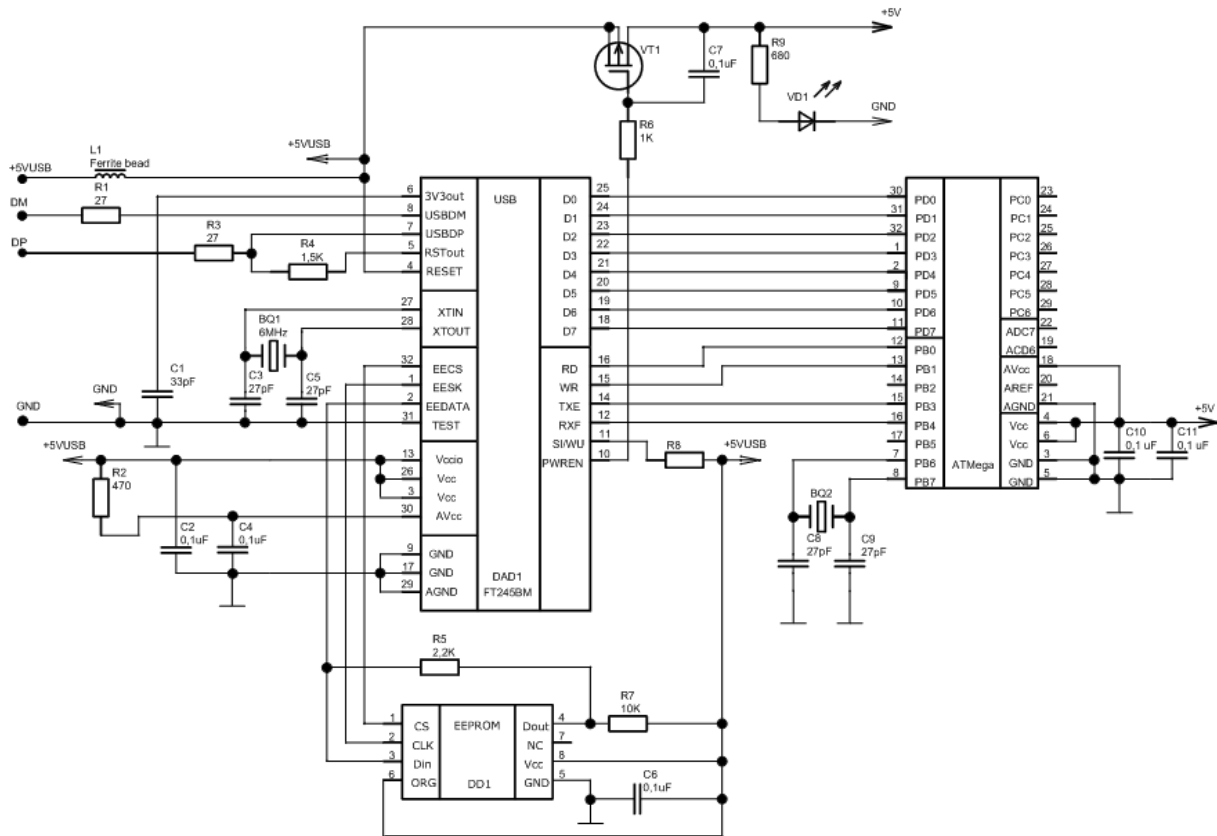


Рис. 3.1. Функциональная схема устройства сопряжения

Устройство сопряжения (рисунок 3.1) построено по принципу “USB Bus Powered device” (устройство питающиеся от USB шины). Основные правила для устройств, построенных по этому принципу следующие:

1. При подключении, устройство должно потреблять не больше, чем 100 мА.
2. При приостановке устройство, должно потреблять не больше, чем 500  $\mu$ А.
3. Устройство, потребляющее более 100 мА должно использовать PWREN# pin, чтобы удерживать 100 мА при подключении и 500  $\mu$ А при приостановке.
4. Устройство, потребляющее больше 100 мА нельзя подключать к хабу.
5. Не одно устройство не может потреблять больше 500мА от USB.

Данная схема позволяет подключить и запитать внешнюю логику, которая может быть необходима на стадии налаживания работы устройства. Для примера на схеме подключен светодиод VD1, который в зависимости от режима работы устройства “выбирает” источник питания.

Подробно выбор элементов схемы обрaмления и их номиналов обоснованы в спецификации “Datasheet FTDI245BM” (EEPROM Configuration [стр. 16], USB Bus Powered Configuration [стр. 17], Bus Powered Circuit with Power Control [стр. 21]). Рассмотрим лишь те элементы схемы, которые требуют детального описания:

- Для организации развязки между выводами Dout (EEPROM) и EEDATA (FT245BM) был установлен резистор 2.2K(R5).
- При использовании кристалла 6MHz (BQ1) необходимо добавить конденсаторы 27 pF (C3 и C4).
- Резистор 10 K (R7) является токоограничивающим, подключенный через него источник питания устанавливает начальное значение высокого уровня на выводе Dout. Такое решение позволяет различить случай возникновения значения низкого уровня на выходе Dout, которым схема EEPROM может сигнализировать об ошибке.
- Конденсаторы C2, C6, C10, C11 выполняют функции стабилизации питания.
- Резисторы R1, R3, R4 - ограничители входного тока.

Разработка программной части велась в интегрированной среде разработки WinAVR.

Рассмотрим программу, написанную для микроконтроллера. Программный модуль управления работой микросхемой FT245BM реализует следующие функции:

1. Инициализация USB  
(void Init\_USB (void))
2. Отправка данных по шине USB  
(void sendtoUSB (uint8\_t \*dataout, uint8\_t cou))
3. Получение данных по шине USB  
(uint8\_t GetFromUSB (uint8\_t \*datain, uint8\_t couread, bool wait))

С целью тестирования была так же разработана программа usb\_try.c, которая позволяет проверить работу всех вышеперечисленных функций.

Что бы получить результаты, которые можно оценить, необходимо сгенерировать прошивку микроконтроллера по следующему алгоритму:

1. Создать проект в «Programmers notepad» и включить в него файлы def.h, ft245bm.h, ft245bm.c, usb\_try.c.

2. После чего создать makefile, используя программу MFile. Выставить тип микроконтроллера Atmega128, подключить Source File ft245bm.c и сохранить makefile в каталоге проекта.
3. Скомпилировать проект и получить файл прошивки для микроконтроллера. Программирование микроконтроллера в makefile настроено на использование прошивающей программы PonyProg.

## ВЫВОД

На рассмотрение представлен программно-аппаратный комплекс, оригинальность которого в первую очередь просматривается в структуре аппаратных средств, где с одной стороны фигурирует современный ПК, а с другой светодиодное устройство индикации.

Устройство сопряжения построено на популярных и общедоступных элементах и интерфейсе, что дает возможность его заводского производства, а также делает его конкурентно способным.

Применяемое USB сопряжение обеспечивает комфортность работы с устройством: избавляет пользователя от необходимости дополнительных коммуникаций с системным блоком; дает возможность его “горячего” подключения; исключает необходимость подключения дополнительных источников тока.

Применение данного комплекса ориентировано на широкую аудиторию пользователей, которым необходимо оперативно отслеживать поступающие в систему события.

Расширение сферы применения разработанного устройства возможно за счет его преобразования из устройства индикации в устройство управления.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ATMEL AVR ATmega128 Datasheet, *www.atmel.com*
2. “Применение микроконтроллеров AVR. Схемы, алгоритмы, программы.”, Баранов В.Н., Москва, 2004. - 288 с.
3. Future Technology Devices International FT245BM Datasheet, *www.ftdichip.com*
4. ATMEL AVR AT90USB1287 Datasheet, *www.atmel.com*
5. Архитектура 8-разрядных микроконтроллеров семейства AVR ATmega128, *www.gaw.ru*
6. FT245BM USB FIFO ( USB - Parallel ) Datasheet, I.C. Future Technology Devices International Ltd. *www.ftdichip.com*
7. “Реализация интерфейса USB в микроконтроллерных устройствах”, Современная электроника, № 2 2007, СТА-ПРЕСС.



## ПРИЛОЖЕНИЕ 1. ЛИСТИНГ ПРОГРАММЫ С ПОДРОБНЫМИ КОММЕНТАРИЯМИ

### Файл def.h:

```
#ifndef __DEFINES__
#define __DEFINES__

#include <avr/io.h>
#include <inttypes.h>
#include <avr/interrupt.h>

#define false 0
#define true 1
#define set_bit(port,mask) ((port) |= _BV(mask)) //макрос устаноки бит по
//маске
#define clear_bit(port,mask) ((port) &= ~_BV(mask)) //макрос сброса бит
//по маске
#define NOP() __asm__ __volatile__ ( "nop\n" )

#endif // __DEFINES__
```

### Файл ft245bm.h:

```
#ifndef __FT245BM_DEF__
#define __FT245BM_DEF__

#define CONTROL_USB PORTB // Порт управления FT245BM
#define CONTROL_USB_CMD DDRB // Регистр настройки порта для RD и WR

#define READ_USB DDB0 // Сигнал управления Read
#define WRITE_USB DDB1 // Сигнал управления Write

#define TRANSMITENABLED DDB3 // Сигнал передатчик пуст
#define READNEED DDB4 // Сигнал приёмник полон

#define INFO_USB DDRB //Регистр настройки порта для #TXE и #RXF
#define INFO_USB_1 PINB

#define USBREADREADY bit_is_set( INFO_USB_1, READNEED )
#define USBTRANSMITREADY bit_is_set( INFO_USB_1, TRANSMITENABLED )

#define DATA_USB_IN PIND // Порт получения данных
#define DATA_USB_OUT PORTD // Порт передачи данных
#define DATA_CONTROL_USB DDRD // Регистр управления портом данных

#endif
```

### Модуль управления работой FT245BM ft245bm.c:

```
#include <inttypes.h>
#include <avr/io.h>
#include <avr/sfr_defs.h>
#include "def.h"
#include "ft245bm.h"

uint8_t flgBusy = false;

void Init_USB( void ) // функция инициализации USB
{
    flgBusy = false;

    clear_bit( INFO_USB, TRANSMITENABLED ); // Порт на вход
    clear_bit( INFO_USB, READNEED );
```

```

// Настраиваем на выход линии для управления FT245BM
set_bit( CONTROL_USB_CMD, READ_USB);
set_bit( CONTROL_USB_CMD, WRITE_USB);

// Не записываем и не читаем из USB
clear_bit( CONTROL_USB, WRITE_USB );
set_bit( CONTROL_USB, READ_USB);
}

uint8_t SendToUSB( uint8_t *dataout, uint8_t couSend )
// функция отправка данных по шине USB,
// где couSend - количество посылаемых uint8_t данных,
// dataout - указатель на массив посылаемых данных
{
    if( flgBusy == true ) return 0;
    flgBusy = true;

    DATA_CONTROL_USB = 0xFF; // Настройка линий на вывод

    uint8_t cou = couSend;
    while( couSend > 0 )
    {
        while( USBTRANSMITREADY > 0 );
        // Ожидаем пока не освободится буфер для передачи
        DATA_USB_OUT = *dataout;
        set_bit( CONTROL_USB, WRITE_USB );
        NOP();
        NOP();
        clear_bit( CONTROL_USB, WRITE_USB );
        dataout ++;
        couSend --;
    }
    flgBusy = false;
    return cou;
}

uint8_t GetFromUSB( uint8_t *datain, uint8_t couread, uint8_t wait )
// функция получения данных по шине USB,
// где couread - количество получаемых uint8_t данных,
// cou - количество полученных uint8_t данных,
// datain - указатель на массив получаемых данных,
// wait - флаг ожидания
{
    uint8_t cou = 0; // Количество прочитанных байт

    if( flgBusy == false )
    {
        flgBusy = true;
        DATA_CONTROL_USB = 0x00; // Настройка линий на ввод

        while( couread > cou ) // читаем, пока не достигнут максимум
        {
            if( USBREADREADY > 0 )
            {
                if( wait == false ) break;
                else continue;
            }
            clear_bit( CONTROL_USB, READ_USB ); NOP();NOP();
            *datain++ = DATA_USB_IN;
            set_bit( CONTROL_USB, READ_USB );
            cou ++;
        }
        flgBusy = false;
    }
    return cou;
}

```

## Файл usb\_try.c:

```
#include <avr/io.h>
#include <util/delay.h>
#include "def.h"
#include "ft245bm.h"

EMPTY_INTERRUPT( __vector_default );
#define T16_TIME_MS      2 // Время вызова прерывания переполнения таймера
#define TCNT16_Init
    (uint16_t) (~(( uint64_t)T16_TIME_MS*(uint64_t)F_CPU/1024000)+1)

void TrySendReceive()
{
    register uint8_t  datacou;
    uint8_t data[64];

    clear_bit(TIMSK,TOIE1);

    datacou = GetFromUSB( data, sizeof(data), false );
    // Получение пакета по USB

    SendToUSB( data, datacou );
    // Отправка эха полученный данных

    if( datacou > 0 )PORTC ^= _BV(PC5);
    TCNT1 = TCNT16_Init;
    set_bit(TIMSK,TOIE1);
}

void inline Init_Timer ( void )
// Инициализация таймера и настройка прерываний от таймера.
{
    TCCR1A = 0;
    TCCR1B = _BV (CS10) | _BV(CS12);
    TCNT1 = TCNT16_Init;
    TIMSK |= _BV(TOIE1);
}

void inline Init (void)
{
    cli();
    DDRC = 0xFF;      // направление линий порта "C" на выход для
    Init_USB();      // тестирования работы программы можно использовать
                    // светодиод,который вводится в отладочную схему
    Init_Timer();
    sei();
}

int main(void)
{
    uint8_t  couIn=0;

    Init();
    TrySendReceive();
    while(100) // Цикл сигнализирования нормальной работы (если будет
              //введен в схему светодиод – он будет мигать)
    {
        couIn = 0;
        PORTC ^= _BV(PC3);
        _delay_ms(100.00);
    }
    return 0;
}
}
```