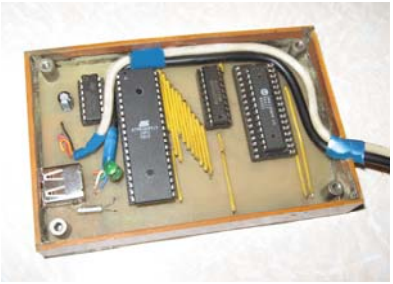
Анализатор USB



- **Аппаратный анализ протокола USB**
- **Возможность анализа на уровне пакетов**
- **Не требует установки драйверов**
- **Может работать с любой операционной системой**
- **Питание от шины USB**
- **Простая схема, доступные комплектующие**

Сейчас практически все периферийные устройства подключаются к компьютеру через USB интерфейс. Порты COM, LPT и PS/2, традиционно использовавшиеся для этой цели, постепенно исчезают из стандартной конфигурации компьютера. Наряду с простотой использования для конечного пользователя, реализация протокола обмена по шине USB достаточно сложна для программистов и разработчиков периферийных устройств. Для отладки обычно используются программные средства, например USB-Monitor [1], позволяющие следить за информацией, передаваемой по шине. Это существенно упрощает отладку программного обеспечения.

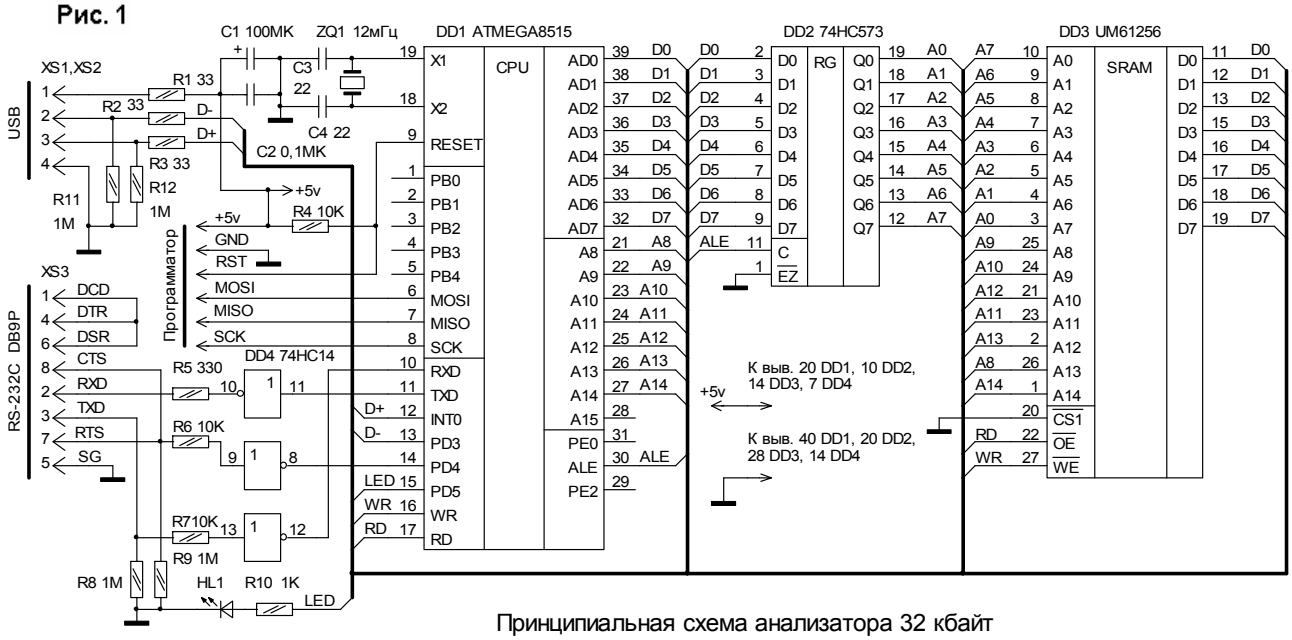
Однако отследить обмен данными между хост-контроллером и устройством на уровне передаваемых блоков данных, называемых пакетами, таким образом невозможно. Дело в том, что каждый пакет содержит много служебной информации, которая добавляется на аппаратном уровне в контроллере USB, как на стороне хоста, так и на стороне устройства. Это идентификатор пакета, адрес, контрольная сумма и др. Таким образом, программно можно отследить, что передается, но нельзя увидеть, как это делается. А для разработчика устройства это может быть очень важно, особенно при программной реализации протокола на контроллерах, не имеющих аппаратного порта USB.

Существуют аппаратные анализаторы протокола USB, которые подключаются между устройством и компьютерам и перехватывают всю информацию, передаваемую по шине. Однако стоимость таких устройств очень высока и далеко не каждая лаборатория может позволить себе иметь такую экзотику. Не говоря уже о радиолюбителях, для которых устройства подобного рода практически недоступны.

Восполнить этот пробел призван предлагаемый анализатор USB. Это простой и дешевый прибор, подключаемый к шине параллельно отлаживаемому устройству. К сожалению, быстродействие контроллеров AVR позволяет отслеживать только низкоскоростной (Low-speed) режим передачи данных. Однако этот режим довольно широко используется, например, в клавиатуре, мыши, джойстике. Практически для любого устройства, которое работает с COM портом, достаточно возможностей режима Low-speed.

Принципиальная схема анализатора показана на рис. 1. Его основа – контроллер DD1 ATMEGA8515. Анализатор подключается парал-

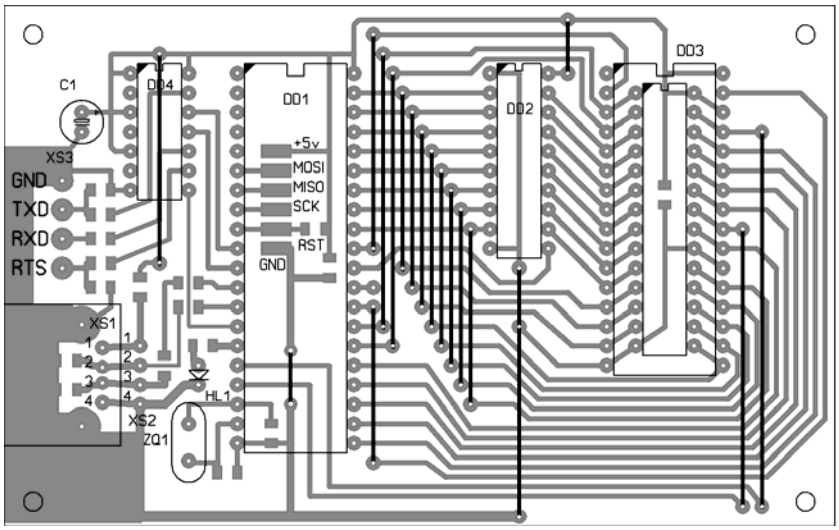
Рис. 1



Принципиальная схема анализатора 32 кбайт

лельно шине и перехватывает всю передаваемую по ней информацию. Эта информация временно сохраняется во внешнем буферном ОЗУ DD3 типа UM61256, а затем передается в компьютер через COM порт. Такое решение обусловлено тем, что тактовая частота на шине USB 1,5 МГц, скорость же работы COM порта обычно не превышает 115200 бод (бит/сек). Это вынуждает накапливать информацию в ОЗУ, а затем, в паузах между пакетами обрабатывать ее и передавать в компьютер. Объема встроенного ОЗУ контроллера для этого недостаточно. Если в компьютере нет COM порта, можно использовать конвертер USB-COM.

Для согласования уровней используется микросхема DD4. Уровни сигналов интерфейса RS232 отличаются от стандартных, но если длина кабеля не превышает нескольких метров, проблем не возникает. Дело в том, что по стандарту уровень единицы должен быть в пределах +(3...12 В), а уровень нуля –(3...12 В). Однако, практически для всех современных интерфейсных микросхем, граница между уровнями нуля и единицы находится в пределах +(1...2 В). Светодиод HL1 сигнализирует о готовности устройства к работе, он гаснет при переполнении ОЗУ.



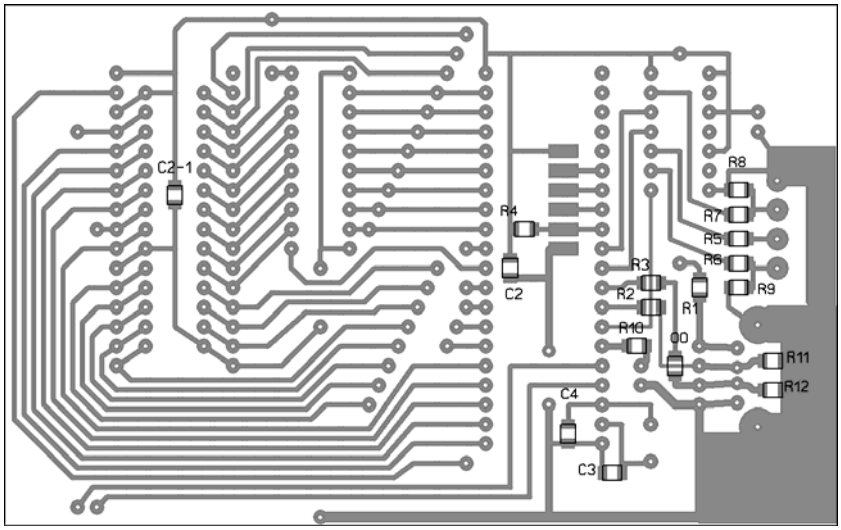


Рис. 2

Устройство смонтировано на односторонней печатной плате размером 110x70 мм. Ее чертеж показан на рис. 2. Верхняя часть рисунка – это вид со стороны установки деталей, а нижняя – со стороны печатных проводников. Для подключения отлаживаемого устройства на плате установлен разъем XS1 - USB розетка типа «А». Параллельно ей распаивается кабель с вилкой USB типа «А» для подключения к компьютеру. Таким образом, устройство оказывается подключенным параллельно шине.

Как видно из схемы, подключение адресных входов DD3 к адресной шине контроллера не стандартно. Это никак не сказывается на работе, но упрощает разводку платы. Ведь безразлично, в какую именно ячейку памяти записывается информация, главное, чтобы при чтении обращение происходило к той же самой ячейке. Светодиод HL1 может быть любого типа, DD2 можно заменить на 74AC573, а DD4 – на 74AC14, 74HC04, 74AC04.

Объем ОЗУ UM61256 32 кбайт. Микросхемы подобного типа широко использовались в материнских платах старых компьютеров. 32 кбайт обычно достаточно для отслеживания процесса нумерации любого устройства, но на плате предусмотрена возможность установки ОЗУ на 128 кбайт типа UT621024 в «широком» корпусе DIP32. Его адресные выводы A15 и A16 подключаются к выводам A15 и PE0 контроллера соответственно. На печатной плате это соединение предусмотрено. Но для того, чтобы контроллер смог использовать эту дополнительную память, потребуется доработка программы.

Во FLASH память контроллера должен быть запрограммирован файл **an.hex**, а FUSE биты установлены следующим образом (не забывайте, что 0 – значит «запрограммирован», а 1 – нет):

S8515C =1, WDTON =1, SPIEN =0, CKOPT =0, EESAVE =1, BOOTSZ 1:0 =11 (Boot Flashsize=128 words Boot address=$0F80), BOOTRST =1, BODEN =0, BODLEVEL =1 (Brown-out detection at Vcc=2.7V), SUT 1:0 =11, CKSEL 3:0 =1111 (Ext. Crystal/Resonator Hih Freq; Start-up time: 16K CK+64ms).

Порядок работы с устройством следующий. Вначале необходимо запустить на компьютере какую-либо терминальную программу, выбрать в ней реальный (или виртуальный, при работе через конвертер USB-COM) порт и подключить к нему COM разъем анализатора. Параметры порта должны быть установлены такие: скорость 115200 бод, режим 8N1.

Теперь нужно подключить USB вилку анализатора к свободному

USB порту компьютера. Устройство никак себя не проявляет, поэтому компьютер не обнаружит этого подключения. В терминальной программе выведется на экран сообщение о готовности анализатора к работе, а на нем засветится светодиод HL1. Теперь можно подключить кабель отлаживаемого устройства к розетке анализатора. При этом оно определится компьютером и начнется процедура нумерации. В терминальной программе будет выводиться на экран все, что происходит на шине USB. Эту информацию можно сохранить в файл для последующего анализа. Вот так примерно будет выглядеть начало процесса нумерации устройства:

```
0000: 2D 00 10
0010: C3 80 06 00 01 00 00 40 00 DD 94
0020: D2
0030: 69 00 10
0040: 5A
0050: 69 00 10
0060: 5A
0070: 69 00 10
0080: 5A
0090: 69 00 10
00A0: 5A
00B0: 69 00 10
00C0: 5A
00D0: 69 00 10
00E0: 5A
00F0: 69 00 10
0100: 4B 12 01 10 01 00 00 00 08 11 77
0110: D2
0120: 69 00 10
0130: C3 C0 16 DC 05 01 00 01 02 9A 94
0140: D2
0150: 69 00 10
0160: 5A
0170: 69 00 10
0180: 5A
0190: 69 00 10
01A0: 5A
01B0: 69 00 10
01C0: 4B 00 01 3F 8F
01D0: D2
01E0: E1 00 10
01F0: 4B 00 00
0200: D2
RESET
```

Для анализа этой информации программист должен иметь представление о том, что такое интерфейс USB. Эти сведения на русском языке можно найти в [2].

Каждая строка – это пакет, переданный по шине. В начале выводится адрес, по которому строка была записана в ОЗУ. Он позволяет убедиться, что анализатор работает нормально, без пропусков информации, а также облегчает восприятие, являясь порядковым номером строки. Затем после двоеточия идут байты информации. Направление передачи невозможно определить аппаратно, но оно может быть определено из предыстории и по идентификатору пакета – это всегда первый байт в пакете и, соответственно, в строке. Собственно идентификатор (PID) – это 4 младших бита первого байта, 4 старших бита – это инверсное значение младших бит, они служат для контроля.

Любая активность на шине начинается всегда по инициативе компьютера (хоста), устройство может только ответить, но ничего не может передать по своей инициативе. В строке с адресом 0000 компьютер передал первый байт $2D, что означает, что это пакет типа «SETUP», в его следующих двух байтах содержится адрес устройства и номер конечной точки, для которой будет передаваться информация в следующем пакете, а также контрольная сумма CRC5. После пакета «SETUP» компьютер передает пакет данных, который начинается байтом $C3 - это «DATA0». Затем в пакете передаются 8 байт данных и 2 байта контрольной суммы CRC16. В данном случае данные – это стандартный запрос дескриптора устройства.

Компьютер передал запрос и должен получить подтверждение, что запрос принят. В следующей строке устройство так и делает - подтверждает прием пакетом из одного байта «ACK». Теперь компьютер ждет ответ «по существу», посылая запрос «IN» (строка 0030). Но устройству на подготовку ответа требуется время, поэтому оно отвечает «NAK» (строка 0040). Это значит, что ответ еще не готов. Наконец, в строке 0100 устройство передает первые 8 байт своего дескриптора.

Компьютер в строке 0110 подтверждает прием, посылая «ACK», затем просит продолжить передачу дескриптора. Ведь длина дескриптора устройства 18 байт, а в одном пакете не может быть передано больше 8, поэтому передача ведется частями. В строке 01C0 устройство передает последние 2 байта своего дескриптора, компьютер подтверждает прием, затем посылает в строке 01E0 пакет типа «OUT», за которым следует в строке 01F0 так называемый «NULL data packet». Устройство подтверждает его прием в строке 0200.

На этом первая фаза нумерации закончена, компьютер получил минимум необходимой информации об устройстве и производит сброс шины. Это строка «RESET». После этого процесс нумерации продолжается.

Таким образом, имея представление о том, что должно происходить на шине и отслеживая, что реально там происходит, можно вести отладку программного обеспечения. После заполнения ОЗУ запись прекращается, о чем сигнализирует погасание светодиода. Возобновить запись с начала можно, нажав в любой момент клавишу «Пробел» на клавиатуре компьютера. Можно было бы реализовать кольцевой буфер в ОЗУ, но при неблагоприятных условиях не исключено его переполнение. Кроме того, отслеживание положения указателей записи и чтения в кольцевом буфере требует больше времени, а ресурсы контроллера и так используются почти полностью на программное декодирование пакетов USB.

Облегчить интерпретацию записанной информации могла бы специальная программа, запущенная на компьютере. Приглашаю заинтересованных ра-

диолюбителей, которые имеют необходимые знания, опыт, время и желание, взяться за ее создание.

Литература.

1. USB-Monitor – http://www.hhdsoftware.com
2. Агуров П. В. Практика программирования USB – СПб.: БХВ-Петербург, 2006.

Прошивку контроллера, исходный текст программы, чертеж печатной платы в формате Sprint Layout 4.0, схему в формате Orcad 9.1, а также вариант схемы устройства с использованием ОЗУ на 128 кб можно загрузить с сайта автора по адресам:

http://ra4nal.qrz.ru
http://ra4nal.lanstek.ru
http://ra4nalr.tut.ru

Разработка 2008 г.