Objective Development Forums

USB MIDI Interface

(#p8061)

Postby **hs4pmh** » Mon Feb 23, 2009 4:06 am

dear horo,

I've studied your project
and now I can implement the USB-MIDI In already

next step,

how Can I do the USB-MIDI-Out

(Host send midi data to AVR
and AVR Generate MIDI out to UART TX @31250bps)

And I've read the MIDI10.pdf
and found that it is possible to do the
multiple midi port on one USB
how can I do this?

thank you

Top

(#p8062)

- Report this post (./report.php?f=8&p=8062)
- Quote (./posting.php?mode=quote&f=8&p=8062)
- (javascript:void(0);)

Postby **horo** » Mon Feb 23, 2009 10:31 am

Hi hs4pmh,

I didn't implement midi out but pat did it - look about 5 postings ^above^. He used a piece of code from http://x37v.info/projects/microcontroll ... idi/files/ (http://x37v.info/projects/microcontroller/avr-midi/files/)

Concerning multi midi the doc midi10.pdf (p.13) says:

> USB-MIDI converters can connect to multiple Embedded MIDI Jacks. Each MIDI Endpoint in a USB-MIDI
> converter can be connected to up to 16 Embedded MIDI Jacks. Each Embedded MIDI Jack connected to one
> MIDI Endpoint is assigned a number from 0 to 15. MIDI Data is transferred over the USB in 32-bit USBMIDI
> Event Packets, with the first 4 bits used to designate the appropriate Embedded MIDI Jack.

So you have to define the number of endpoints and the endpoints in your descriptor starting from section B.4 in my source code - it's a bit trial and error. It would help if you draw the connections from endpoint to internal and external midi jack to understand the data flow (look for bJackID, BaSourceID and baAssocJackID). You connect an embedded midi in to an external midi out and vice versa.

Ciao Martin

Top

Re: USB MIDI Interface (#p8825)

- Report this post (./report.php?f=8&p=8825)
- Quote (./posting.php?mode=quote&f=8&p=8825)
- (javascript:void(0);)

Postby **psc** » Sat Apr 11, 2009 2:02 am

hi martin,

i am very happy to say that i am now using an atmega164p (new version of atmega16). it's cheaper and faster (20 mhz). i want to ask you a question about optimization for the usb communication. here's my function that send the noteon, noteoff:

```
void usbsend(unsigned char note, unsigned char velo) {
   while (!usbInterruptIsReady()) {
            wdt_reset();
            usbPoll();
      }
   midiMsg[0] = 0x09;
   midiMsg[1] = MIDI_NOTEON|MIDI_CHAN;
```

```
    midiMsg[2] = note;
    midiMsg[3] = velo;
    midiMsg[4] = 0x08;
    midiMsg[5] = MIDI_NOTEOFF|MIDI_CHAN;
    midiMsg[6] = note;
    midiMsg[7] = 0;
    sendEmptyFrame = 1;
    usbSetInterrupt(midiMsg, 8);
}
```

this function takes 8ms to finish (i am running the obdev usb firmware at 20mhz). sending all 8 ADC takes 8ms * 8 = 64ms. long enough to miss an hit (i'm making an electronic drum). i was wondering if you have an advice for me? or maybe someone else (hi christian!). is it the limitation of usb-midi or the limitation of my knowledge?

cheers!
pat

Top

USB MIDI Interface updated (#p9028)

- Report this post (./report.php?f=8&p=9028)
- Quote (./posting.php?mode=quote&f=8&p=9028)
- (javascript:void(0);)

Postby **horo** » Wed Apr 22, 2009 10:37 pm

Hi,

I've just uploaded a new version of my project - it's got a new name V-USB-MIDI according to the new driver name.
I made no big changes but I updated to the last version of v-usb (2009-04-15).
V-USB-MIDI-0.2 (http://cryptomys.de/horo/V-USB-MIDI/index.html)

hi pat,

I have no idea how to speed up the communication - but maybe christian knows??

Ciao Martin

Top

Re: USB MIDI Interface (#p9030)

- Report this post (./report.php?f=8&p=9030)
- Quote (./posting.php?mode=quote&f=8&p=9030)
- (javascript:void(0);)

Postby **psc** » Wed Apr 22, 2009 11:14 pm

Hi Horo,

Thank you for your update!

About the speed, there is no way to make it faster when using usbSetInterrupt() (up to 8 bytes may be passed in one call) which take 8ms (homemade measurement). The only way i found was to get rid of MIDI_NOTEOFF since i only want to trig a sound percussion and add a second MIDI_NOTEON instead.

```
void usbsend(unsigned char first_note, unsigned char first_velo, unsigned char
second_note, unsigned char second_velo) {
   while (!usbInterruptIsReady()) {
            wdt_reset();
            usbPoll();
      }
   midiMsg[0] = 0x09;
   midiMsg[1] = MIDI_NOTEON|MIDI_CHAN;
   midiMsg[2] = first_note;
   midiMsg[3] = first_velo;
   midiMsg[4] = 0x09;
   midiMsg[5] = MIDI_NOTEON|MIDI_CHAN;
   midiMsg[6] = second_note;
   midiMsg[7] = second_velo;
   sendEmptyFrame = 1;
   usbSetInterrupt(midiMsg, 8);
}
```

Cheers,
Patrick

Top

USB MIDI Interface in Bulk mode 2 ms (#p9034)

- Report this post (./report.php?f=8&p=9034)
- Quote (./posting.php?mode=quote&f=8&p=9034)
- (javascript:void(0);)

Postby **horo** » Thu Apr 23, 2009 11:21 am

Hi Patrick,

I've read the descriptor section in main.c again and changed some values and suddenly -tada! - my test program sends data (4 or 8 byte) every two milliseconds! Then I started again step by step - it's only necessary to change the values of bmAttributes from 3 to 2 (Interrupt -> Bulk) in section B.5.1 and B.6.1.

```
/* Name: main.c
 * ...
 */

// ...

// B.5 Bulk OUT Endpoint Descriptors
```

```
//B.5.1 Standard Bulk OUT Endpoint Descriptor
   9,          /* bLenght */
   USBDESCR_ENDPOINT,   /* bDescriptorType = endpoint */
   0x1,          /* bEndpointAddress OUT endpoint number 1 */
   2,          /* bmAttributes: 2:Bulk, 3:Interrupt endpoint  <=== CHANGE HERE
TO BULK */
   8, 0,         /* wMaxPacketSize */
   10,          /* bIntervall in ms */
   0,          /* bRefresh */
   0,          /* bSyncAddress */

// B.5.2 Class-specific MS Bulk OUT Endpoint Descriptor
   5,          /* bLength of descriptor in bytes */
   37,          /* bDescriptorType */
   1,          /* bDescriptorSubtype */
   1,          /* bNumEmbMIDIJack  */
   1,          /* baAssocJackID (0) */


//B.6 Bulk IN Endpoint Descriptors

//B.6.1 Standard Bulk IN Endpoint Descriptor
   9,          /* bLenght */
   USBDESCR_ENDPOINT,   /* bDescriptorType = endpoint */
   0x81,          /* bEndpointAddress IN endpoint number 1 */
   2,          /* bmAttributes: 2: Bulk, 3: Interrupt endpoint  <=== CHANGE
HERE TO BULK */
   8, 0,         /* wMaxPacketSize */
   10,          /* bIntervall in ms */
   0,          /* bRefresh */
   0,          /* bSyncAddress */

// B.6.2 Class-specific MS Bulk IN Endpoint Descriptor
   5,          /* bLength of descriptor in bytes */
   37,          /* bDescriptorType */
   1,          /* bDescriptorSubtype */
   1,          /* bNumEmbMIDIJack (0) */
   3,          /* baAssocJackID (0) */
};

// ...
```

My Laptop uses sidux - a debian sid based linux - with new sidux kernel version 2.6.29.1 / or the actual realtime kernel version. I don't know which system do you have - please check it out.

So long
Martin

Top

Re: USB MIDI Interface (#p9040)

- Report this post (./report.php?f=8&p=9040)
- Quote (./posting.php?mode=quote&f=8&p=9040)
- (javascript:void(0);)

Postby **psc** » Thu Apr 23, 2009 10:52 pm

Hi Horo,

How nice! From 8 ms to 2 ms just by changing the bmAttributes. It's working very well, however i have this warning(?) when plugging the device:

```
[ 8848.352583] usb 1-2: new low speed USB device using uhci_hcd and address 14
[ 8848.456365] usb 1-2: config 1 interface 1 altsetting 0 endpoint 0x1 is
Bulk; changing to Interrupt
[ 8848.456379] usb 1-2: config 1 interface 1 altsetting 0 endpoint 0x81 is
Bulk; changing to Interrupt
[ 8848.473328] usb 1-2: configuration #1 chosen from 1 choice
```

Linux mbp 2.6.24-23-rt #1 SMP PREEMPT RT Wed Apr 1 23:40:34 UTC 2009 i686 GNU/Linux
I didn't try on Windows or Mac.

There's something i am wondering:

> Consumes 90% or more of the AVR's CPU time because bulk endpoints are polled aggressively by the host. Real-time applications on the AVR are close to impossible.

Since we are now using bulk endpoint (are we?) is this a concern for us?

```
for (;;) {
      PORTD ^= 0x80;
      wdt_reset();
      usbPoll();
}
```

Gives **2 µs** (the CPU doesn't seem busy).

```
for (;;) {
      PORTD ^= 0x80;
      wdt_reset();
      usbPoll();
      usbSetInterrupt(midiMsg, 8);
}
```

Gives **2 ms** - Wonderful!

Patrick

Top

Re: USB MIDI Interface (#p9092)

- Report this post (./report.php?f=8&p=9092)
- Quote (./posting.php?mode=quote&f=8&p=9092)
- (javascript:void(0);)

Postby **horo** » Tue Apr 28, 2009 9:56 am

> ***psc wrote:***
> ...
> There's something i am wondering:
>
> > Consumes 90% or more of the AVR's CPU time because bulk endpoints are polled aggressively by the host. Real-time applications on the AVR are close to impossible.
>
> Since we are now using bulk endpoint (are we?) is this a concern for us?
> ...
> Patrick

Hi Patrick,

I don't think we're using bulk xfer because:

```
[ 8848.456365] usb 1-2: config 1 interface 1 altsetting 0 endpoint 0x1 is
Bulk; changing to Interrupt
[ 8848.456379] usb 1-2: config 1 interface 1 altsetting 0 endpoint 0x81 is
Bulk; changing to Interrupt
```

The linux driver changed it to int - I see activity on usb lines every 2 ms for about 120 µs (and the SOF "heartbeat" on D- every 1ms). So we have 6% usb load and more than 90% cpu time remaining for our application - that's ok.

Ciao Martin

P.S.: it would be helpful if someone could verify this for win and mac.

Top

Re: USB MIDI Interface (#p9104)

- Report this post (./report.php?f=8&p=9104)
- Quote (./posting.php?mode=quote&f=8&p=9104)
- (javascript:void(0);)

Postby **horo** » Tue Apr 28, 2009 3:20 pm

Hi,

did some "research":
linux:
bmAttributes=2 (bulk), bIntervall don't care -> 2ms poll, low cpu load

bmAttributes=3 (interrupt), bIntervall=1..3 -> 2ms poll, low cpu load
bmAttributes=3 (interrupt), bIntervall=4..7 -> 4ms poll, low cpu load
bmAttributes=3 (interrupt), bIntervall=8..15 -> 8ms poll, low cpu load
bmAttributes=3 (interrupt), bIntervall=16..31 -> 16ms poll, low cpu load
bmAttributes=3 (interrupt), bIntervall=0 or 32..63 -> 32ms poll, low cpu load
bmAttributes=3 (interrupt), bIntervall=64..127 -> 64ms poll, low cpu load
bmAttributes=3 (interrupt), bIntervall=128..255 -> 128ms poll, low cpu load

winxp:
bmAttributes=2 (bulk), bIntervall don't care -> permanent poll, high cpu load
bmAttributes=3 (interrupt), bIntervall don't care -> 8ms poll, low cpu load

So I think it's best to use
bmAttributes=3 (interrupt), bIntervall=2
this gives fastest access for linux and no heavy load for win.

Ciao Martin

P.S.: mac users - please test

Top

Re: USB MIDI Interface (#p9163)

- Report this post (./report.php?f=8&p=9163)
- Quote (./posting.php?mode=quote&f=8&p=9163)
- _ (javascript:void(0);)

Postby **eclipso_1** » Fri May 01, 2009 10:15 pm

Hi Martin,

I was wondering if there is a reason that only 2 midi events AKA 8 bytes are sent at a time. I don't recall this from the doc midi10.pdf. Is it a limit of usb 1.1 or just the default in the v-usb driver? Couldn't a larger packet also increase speed or at least throughput but not latency?

-- EDIT
Ok, after further research I've found that that interrupt packets are limited to 8 bytes at low-speed. That means the max you can transfer is 800 bytes (200 midi events) per second with 10ms "interrupt". These numbers all assume that you abide by the official usb spec. The AVR-CDC project uses bulk over low-speed and this apparently can increase throughput significantly but isn't within official usb spec. on low speed. He even has a driver for vista at http://www.recursion.jp/avrcdc/lowbulk.html (http://www.recursion.jp/avrcdc/lowbulk.html) that might be useful but none of this helps decrease latency. I would recommend one of the full-speed capable usb integrated AVRs for use in a midi drumset.

BTW I tested the default descriptor from your project on Windows 7 beta build 7000 and it enumerates and functions great.

Thanks for your work on this project,
Dave

Top

Re: USB MIDI Interface (#p9322)

- Report this post (./report.php?f=8&p=9322)
- Quote (./posting.php?mode=quote&f=8&p=9322)
- _ (javascript:void(0);)

Postby **horo** » Thu May 14, 2009 10:06 am

Hi Dave,

could you please test my recommendation:
`bmAttributes=3 (interrupt), bIntervall=2`
to verify that it works well on win7.

Ciao Martin

Any Mac users around

Top

Re: USB MIDI Interface (#p9407)

- Report this post (./report.php?f=8&p=9407)
- Quote (./posting.php?mode=quote&f=8&p=9407)
- _ (javascript:void(0);)

Postby **eclipso_1** » Thu May 21, 2009 12:05 am

Martin,

I tested the 2ms interrupt setting on the same Windows 7 install and it works great. Above you stated that under winxp that polling is 8ms regardless of bInterval. How did you come to that conclusion? Are you probing the usb signal lines with a scope? In the win utility "USB View" it lists back this:

Endpoint Descriptor:
bEndpointAddress: 0x01
Transfer Type: Interrupt
wMaxPacketSize: 0x0008 (8)
wInterval: 0x0002
bSyncAddress: 0x00

I'm sure that it is just reporting back the descriptor that we gave it and not reporting what's actually happening.

Laters Dave

Top

Re: USB MIDI Interface (#p9413)

- Report this post (./report.php?f=8&p=9413)
- Quote (./posting.php?mode=quote&f=8&p=9413)
- _ (javascript:void(0);)

Postby **horo** » Thu May 21, 2009 11:29 am

Hi Dave,

I've tested the timing with a program fragment like Patrick's (some posts above ^^). I checked the usb lines and the debug port pins with a scope. Pin D.7 toggles permanently with some inactivity (about 120µs) every 2 ms. So there's more than 90% of cpu activity for user program.

Ciao Martin

Top

Re: USB MIDI Interface (#p9440)

- Report this post (./report.php?f=8&p=9440)
- Quote (./posting.php?mode=quote&f=8&p=9440)
- (javascript:void(0);)

Postby **eclipso_1** » Sat May 23, 2009 7:46 am

Hi Martin,

This is all just great! Your research on this descriptor and the use of these cheap 8bit avr chips has opened the door for so many homebrew midi devices I can't even describe how grateful that I am for your time. All we need now is confirmation from mac users and this will be a universal FREE music interface that is future proof. Cheers to you Martin. If you ever make your way to Texas the beer is on me. I might be American but you make me proud to be German too. Long live open source!

Laters, David Burmeister

--EDIT Your awesome too Christian, sorry

Top

Re: USB MIDI Interface (#p9516)

- Report this post (./report.php?f=8&p=9516)
- Quote (./posting.php?mode=quote&f=8&p=9516)
- (javascript:void(0);)

Postby **muntablues** » Thu May 28, 2009 11:21 pm

Hi horo

First of all, great job what you have done.

I took your example changed the ADC and KEY handling and it is working fine.

Now to my question. If I try to send data to the V-USB device there is no feedback on my debug LED. So I am quiet sure that there comes no data to my V-USB devcie. Have you tried that yourself or could give me a hint what is going wrong.

If I send some data to the Windows Midi device with my testsoftware it is working fine. So I think my testapp is OK.

Thank you for your help.

MB

Top

Display posts from previous: All posts ▼ Sort by Post time ▼ Ascending ▼ Go

Post Reply

- Email topic
- Print view

90 posts

- Previous
- 1
- 2
- 3
- 4
- 5
- 6
- Next

Return to “V-USB”

Jump to

- LaunchBar
- LaunchBar General
- LaunchBar Actions
- LaunchBar Search Templates
- Little Snitch
- Little Snitch General
- Sharity
- Sharity General
- WebYep
- WebYep (english)
- WebYep (deutsch)
- Other Products
- V-USB
- V-USB Project Ratings

Who is online

Users browsing this forum: No registered users and 1 guest