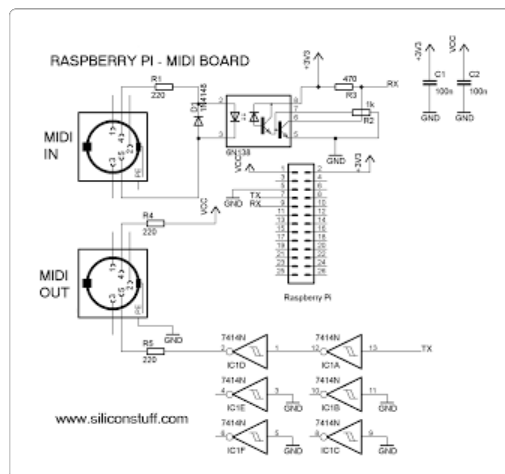


# Silicon Stuff

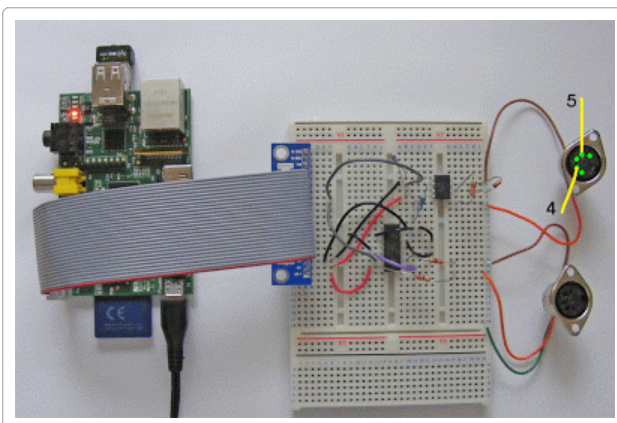
Friday, 31 August 2012

## Serial Port MIDI on the Raspberry Pi

I noticed a project at Raspberry Pi was having to use a microcontroller to buffer 38400 Baud serial port data down to the Midi standard of 31250. As part of an ongoing larger project I solved this problem by changing `init_uart_clock = 2441406` in `/boot/config.txt` on the raspberry Pi.



My proto-type MIDI hardware is the usual 6N138 opto isolator for the MIDI Input and a 7414 for the output shown below. The output from the 6N138 is 3.3v and the output from the 7414 is at 5V. This was chosen because although the MIDI specification usually talks about the signal as a current loop, there are a few devices that are MIDI powered and require 5v. MIDI powered devices work by doing away with the opto-isolator and the safety it provides.



There are a couple of changes that are needed as by default, Raspbian "wheezy" Linux distro, sets it up a serial port tty. Using a console make these changes can be made as follows.

```
sudo nano/boot/cmdline.txt
```

Edit the line

### Blog Archive

▼ 2012 (3)

▼ August (3)

[FLUIDSYNTH on the Raspberry Pi](#)

[TTYMIDI on the Raspberry Pi](#)

[Serial Port MIDI on the Raspberry Pi](#)

### Contributors

[Unknown](#)

[Silicon Stuff](#)

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcbkl0p2 rootfstype=ext4 elevator=deadline rootwait
to
```

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcbkl0p2 rootfstype=ext4 elevator=deadline
rootwait
```

exit saving the changes

Next by changing `init_uart_clock=3000000` to `init_uart_clock=2441406`, all baud rates will be reduced by about 81%. This is not an ideal solution as if you wanted to use the serial port for other projects all the standard baud rate timings would be out by 19%. However, from a MIDI point of view you can simply open `/dev/ttyAMA0` at 38400 baud and you are up and running.

Again using a consol.

```
sudo nano /boot/config.txt
```

Add these lines to the end of the file

```
# change uart clock to 2441406 for midi 31250 baud rate
```

```
init_uart_clock=2441406
```

```
init_uart_baud=38400
```

save and exit, and reboot the Pi.

The Raspberry Pi can now talk MIDI by opening `/dev/ttyAMA0` at 38400 baud or connected to [ALSA](#) using `ttymidi` driving [FluidSynth](#).

Posted by [Silicon Stuff](#) at 10:47

## 11 comments:



**7thheaven** 24 October 2012 at 05:38

Hey, I'm looking to find the best (as far as reliability/timing) midi interface for my RPi, I'll be running a few signals in and a lot of signals out. What's the benefit of using this over a USB MIDI interface?

[Reply](#)



**Silicon Stuff** 3 November 2012 at 02:21

I was getting between 2.26 - 3.94 ms latency using FluidSynth at the time when the RPi was having problems with a USB overhead of 20% CPU.

With the MIDI shield, it's just tidier, with one of the USB ports containing a WiFi Dongle and the other free for a sound card/USB flash etc. Which in the near future will be used as a midi controller.

[Reply](#)



**LOVEIT** 4 November 2012 at 16:53

[this is still me I'm just signed in elsewhere]

Not sure that I quite understand, So you still intend to use USB for midi? And you were using a soft synth within the rPi, not sending midi out?

[Reply](#)



**Silicon Stuff** 11 November 2012 at 00:50

I will not be using USB MIDI for no other reason than less cables.

I only used the soft synth for testing the hardware. Ultimately it will be a MIDI controller.

Leaving latency and jitter aside, the MIDI Baud rate of 31250 bps format is 1 start bit, 8 data, no parity, 1 stop bit. This gives us 3125 bytes of data or ~1000 MIDI messages a second in and ~1000 out (for the RPi).

This is not a lot for the RPi to handle so long as the RPi is not busy running to many apps.

The ALSA MIDI latency test will probably be of help to you.  
<https://github.com/koppi/alsa-midi-latency-test/>

Which you can also use by connecting just RPi midi in and out together.

[Reply](#)



**James Myers** 22 December 2012 at 14:04

this is exactly what I wanted thank you. and thank you for covering the things relating to midi serial baud rate. (it's an important consideration.) I am using my raspberry pi as a midi interface with pure data running in -nogui mode this is the next part of the build for me.

[Reply](#)



**Erik P Beamster** 7 January 2013 at 05:54

hi silicon stuff, i have tried to use my raspberry for midi applications on its uart as you described, but the clockrate on the Tx signal remains 38400 baud, though the vcgencmd measure\_clock uart wil prompt: 224100. What image did you use for this midi-project? There should be a Kernel to support this uart-change, isn't it?

[Reply](#)



**Silicon Stuff** 19 January 2013 at 09:38

Hi Erik.

Sorry, I havn't access to my R-Pi at the moment but I would have been using the current distro in August 2012.

Did config.txt save? as I seem to recall this sometimes being an issue?

I will recheck next week and post any updates.

[Reply](#)

[Replies](#)



**Erik Beamster** 6 February 2013 at 11:50

thaks for the reply....finally i got it working: apart from the additions in config.txt regarding the new clockrate, the following has to be added to the existing adapted line in cmdline.txt: bcm2708.uart\_clock=3000000. So no it's working fine...i had allready made an circuitboard with a PIC controller on it to translate the 38400 to 31250.....but that's for other purposes now..once againg thanks and regards..erik

---

[Reply](#)



**Anonymous** 26 January 2013 at 10:35

Thanks for this. I managed to get a simple MIDI in/out interface working with these instructions. I even managed to "recycle" an opto isolator chip from an old modem circuit board that I had, which saved me a couple of pounds. One word of warning though - some opto isolators don't work fast enough for Midi, so check that the on/off time is at most about 15 microseconds.

I also wrote a simple Midi message monitor in Python, which I'll upload to <http://www.stevegutteridge.com/raspberry> in a moment.

Steve.

[Reply](#)



**Anonymous** 31 December 2013 at 06:32

Very good work, looking for just such a solution for a project of mine.

[Reply](#)

**Florent Frindel** 16 June 2014 at 16:46



Hello. Nice work. Thanks for sharing it. There are two capacitors on the diagram, but i haven't seen them on the photo. Did i miss something ?

[Reply](#)

Enter your comment...

Comment as: Google Accour ▼

[Publish](#)

[Preview](#)

[Newer Post](#)

[Home](#)

Subscribe to: [Post Comments \(Atom\)](#)