ShortStack Goes Raspberry Pi Wiring Instructions

Revision 13-Feb-2015, Bernd Gauweiler

Wiring instructions between a Raspberry Pi Model B+ and an Echelon FT 5000 EVK follow, followed by troubleshooting tips. The wiring instructions shown here match the default GPIO assignments used in the application and driver example for ShortStack on Raspberry Pi. You can change the GPIO assignments in software, and the wiring, according to your application's requirements later if you wish. We recommend that you begin using the default assignments documented here.

The same wiring instructions hold for a Raspberry Pi 2 Model B. The example application and driver were successfully tested on both Raspberry Pi Model B+ V 1.2 and Raspberry Pi 2 Model B V 1.1.

RPi: Raspberry Pi or compatible board. J8 is the big GPIO expansion connector. Pin J8:2 is closest to the nearest corner of the PCB in top view.

EVB: An FT 5000 or FT 6050 EVB. Note that the Raspberry Pi uses 3.3V GPIO. Connections to 5V boards require level shifters.

Begin by removing all jumpers from EVB JP31, JP32, JP201, JP203. Find six or eight short female-to-female jumper wires.

RPi GPIO	RPi J8	µS Signal Name	µS I/O	EVB P201	Additional connection
14	8	RxD	8	9	
15	10	TxD	10	11	JP201:7-8
9	21	CTS	0	1	
10	19	RTS	4	5	JP203:5-6
11	23	HRDY	1	2	JP24:2 - JP31:9 (*)
GND	6	GND		20	
		SBR0	5		JP203:1-2 (**)

		SBR1	6		JP203:3-4 (**)
		SCI	3		JP32:11-12
					JP31:2-11 (***)
					JP24 remains available for logic analyzer connections.

Notes:

(*) This connection is optional. It drives LED1 with the HRDY signal. The LED is lit when the driver is ready to receive data from the Micro Server.

(**) Use SBR0, SBR1 to select the correct link layer bit rate. The current default is 38400bps, which has both jumpers JP203:1-2 and 3-4 removed (SBR0 = SBR1 = 1, using on-board pull-up resistors).

(***) This connection is optional. It drives LED2 with the CTS signal. The LED is lit when the Micro Server receives data from the host.

Figure 1: A Raspberry Pi Model B+ V1.2.

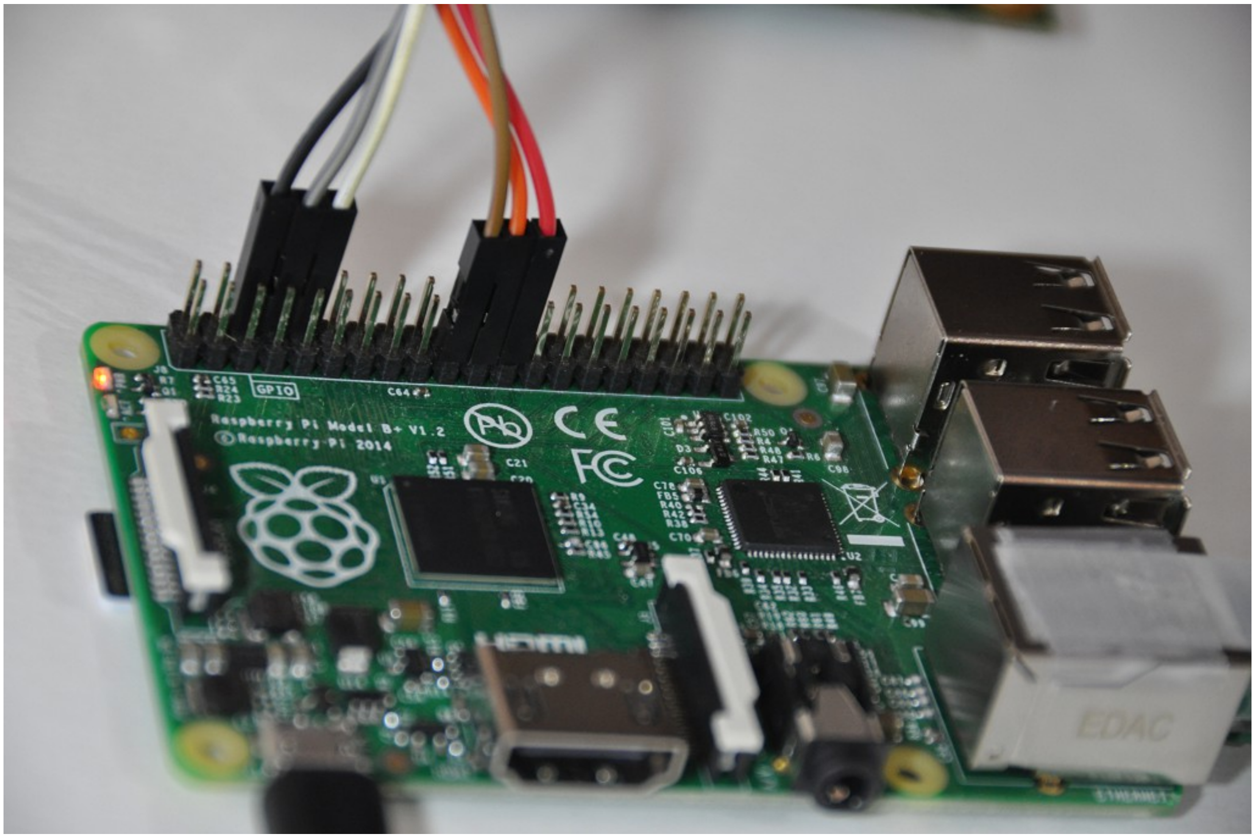


Figure 2: An FT x0y0 EVB. Pinout and relevant jumpers are the same for FT 5000 and FT 6050 EVB. <u>Do not forget the GND connection</u>.

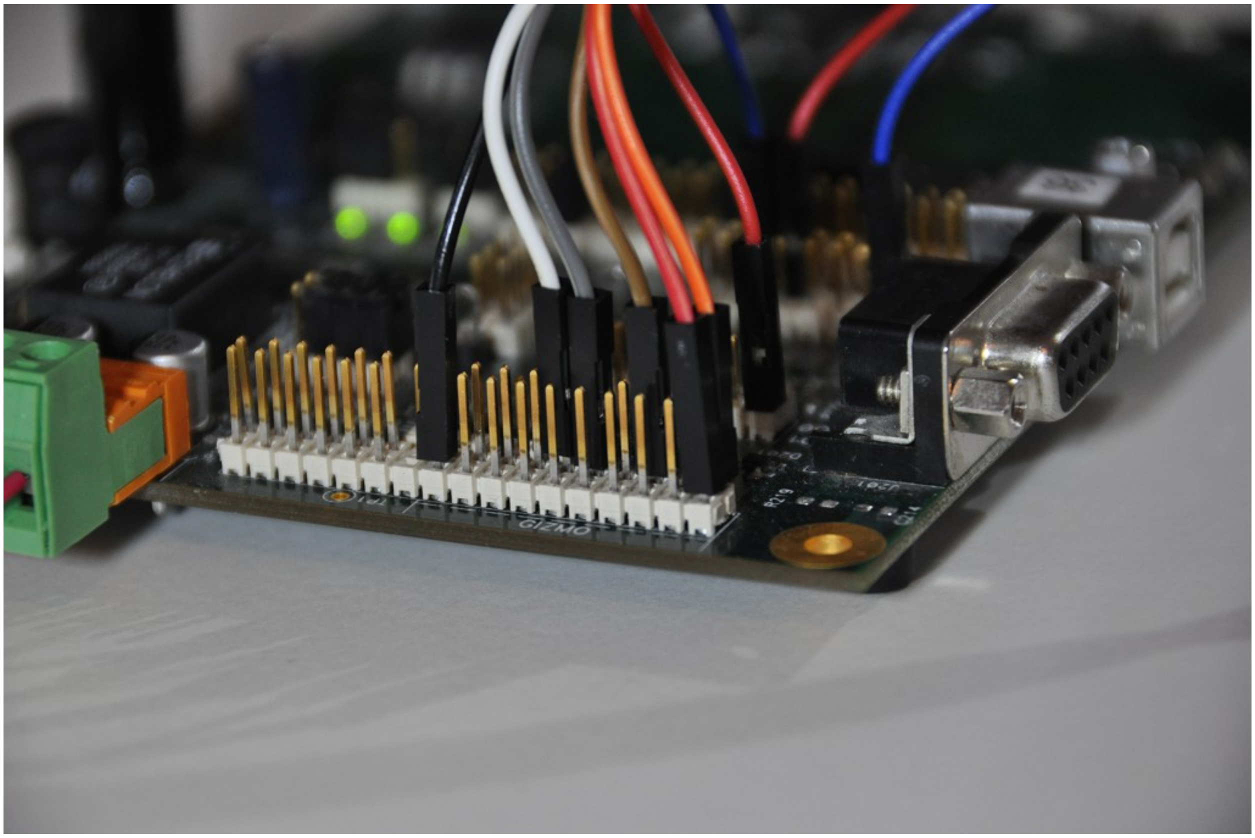
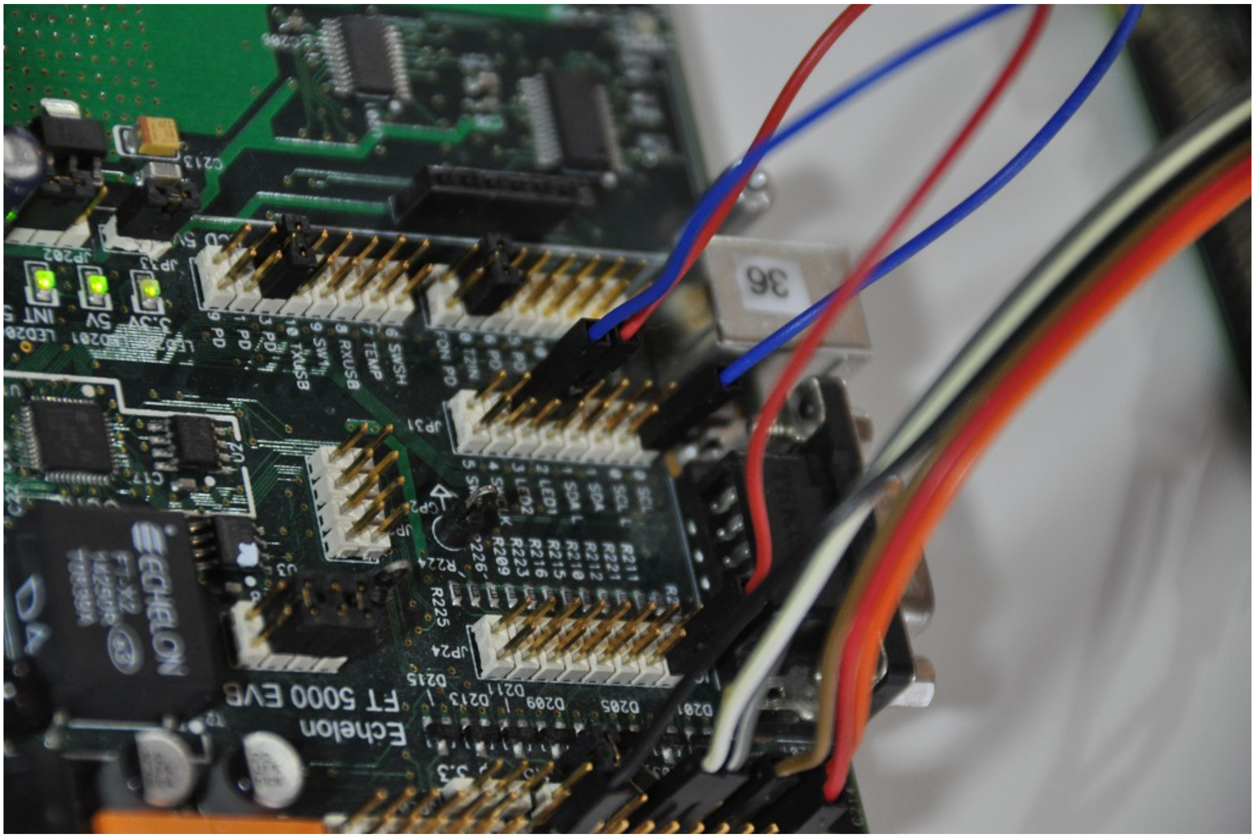


Figure 3: FT x0y0 detail, showing optional wiring (red, blue) to drive LED1 and LED2



Troubleshooting

As always, the cardinal rule is to take one small step at a time. Do not try to plug everything together, load all software, and expect everything to work at once. Maybe it does (it is supposed to), but chances are that some error occurred in the wiring or some other configuration step.

You can try to drive the RTS~ signal from the Raspberry Pi *by hand* using simple console commands. To do so, close the running ShortStack application example (enter *x* on its console), then enter the following commands:

```
cd /sys/class/gpio
echo “10” > export
cd gpio10
echo “high” > direction
```

You should now verify the signal level of RTS~ (RPi GPIO10) using an oscilloscope or multimeter while disconnected from the Micro Server.

Your reading should be 3.3 V.

```
echo "0" > value
```

Your reading should now be 0 Volts. Now connect the RTS~ signal to the Micro Server and verify the signal level. You should still see 0 Volts.

A reading of approximately 1.7 Volts suggests that you may have connected two outputs against each other; disconnect the link and re-examine your wiring. Do not forget to check that you have a solid Ground connection between your Micro Server and host boards.

With the Micro Server running, you should now observe it's CTS~ signal, or its RESET~ signal or LED. The Micro Server should acknowledge your manual RTS~ request with a CTS~ response (low-active, so your reading should be 0V). Because you only assert RTS~ but no start-bit is sensed on the serial line, the Micro Server will experience a watchdog timer reset every 840 ms. This is good an as-designed.

Now try to confirm that your Raspberry Pi board can sample the CTS~ signal correctly using the console:

```
cd /sys/class/gpio
echo "9" > export
cd gpio9
echo "in" > direction
cat value
```

You may need to repeat the last command a few times, because the CTS~ signal stays low only for approximately 840 ms at a time, then is pulled high using one of the pull-up resistors on the EVB for a short period of time.

Finally, confirm that your Raspberry Pi board can correctly drive the HRDY~ signal (GPIO11):

```
cd /sys/class/gpio
echo "11" > export
cd gpio11
echo "high" > direction
```

You should now verify the signal level of HRDY~ (RPi GPIO11) using an oscilloscope or multimeter while disconnected from the Micro Server.

Your reading should be 3.3 V.

```
echo "0" > value
```

Your reading should now be 0 Volts. Now connect the HRDY~ signal to the Micro Server and verify the signal level. You should still see 0 Volts.

Now re-connect the serial data lines, RxD and TxD, using the wiring instructions above. Note that the above table lists Micro Server-centric signal names, the Micro Server's transmit data signal (TxD) must be connected to the Raspberry Pi's RXD0 port, RxD to TXD0.

Start the application example and monitor the serial data lines using an oscilloscope, a serial analyzer or a digital logic analyzer. Note the link layer signals use 3.3V signals.

The application example is configured for a bitrate of 38400 bits per second. Assuming the Micro Server's HRDY~ input is driven low, you should see the Micro Server's uplink reset notification message on the TxD line, and you should be able to confirm the bitrate.

Verify the wiring of the Micro Server's SBR0, SBR1 signals if the bitrate is incorrect.

Verify the wiring of the Micro Server's SPI/SCI~ signal if the uplink reset message is missing.

If all goes well, the ShortStack for Raspberry Pi application example should show console output like so:

```
      Shortstack Goes Raspberry Pi Sample Application 1.00.00
      Copyright (C) 2014 Echelon Corporation


      Enter 'x' to exit or ? for help.
      Connected to /dev/ttyAMA0
      DN.00001 H:0x1c.08
      DN.00001 H:0x1c.08
      P:0x49.3f.9f.ff.ff.06.00.0a.04.01.37.2e.16.09.1a.2b.2c.00.04.00.fc.00.00.
      00.00.00.20.02
      DN.00002 H:0x05.0b
      DN.00002 H:0x05.0b P:0x00.02.02.00.60
      DN.00003 H:0x00.50
UP.00001 H:0x10.50 P:0x00.03.83.0a.b0.14.00.07.00.10.63.a1.00.0f.02.7f
```