

## Bluegiga Forums / Community Forums / Bluetooth Smart

## UART communication between MCU and BLE113 not working on custom PCB

Answered



Simon Deleersnyder

asked this on December 13, 2014, 19:31

Share

Tweet 0

Like 0

Hi,

I made a PCB where an AVR MCU communicates with a BLE113 over UART. The design of the BLE part of the PCB is attached (it was made in Kicad). The PCB is however not functioning as expected:

- 1) The UART communication from the MCU to the BLE113 works. I tested this by connecting the CC debugger to the BLE programming pins and pressing the reset button on the CC debugger, and the MCU does receive the system boot event.
- 2) The UART communication from the BLE113 to the MCU does NOT work. I tested this by trying to make an LED attached to port P2\_0 on the BLE turn on from the MCU (by sending the appropriate commands `ble_cmd_hardware_io_port_config_direction(2,1)` and `ble_cmd_hardware_io_port_write(2,1,1)` over UART) and the LED did not turn on.
- 3) The TX pin on the MCU (operating at 5V) is connected with the RX pin on the BLE (operating at 3.3V) through a voltage divider. Similarly, the TX on the BLE is connected with the MCU through a 3.3V to 5V level shifter using two npn transistors. Both this level shifter, and the voltage divider circuit work because I tried connecting Jeff Rowberg's breakout board to my PCB's female headers, that is: 3v3 to Vdd, gnd to gnd, P02 to P0\_2 and P03 to P0\_3. In this case the UART communication works in both directions and the BLE113 on the breakout executes all the commands flawlessly.
- 4) So maybe the BLE113 is not connected to the PCB properly? To test this I pulled all ports of the BLE high (in the hardware.xml file) and measured the voltages with my multimeter and they behaved as specified.

So the question is if there is anything that I am clearly doing wrong? It seems that in Jeff's breakout PCB (see attachment) the only pins that are connected to ground are pins 36 and 25, while in my circuit (see other attachment) all of the BLE's GND pins are connected to ground. Is this maybe what I am doing wrong? Another difference is that Jeff Rowberg's breakout uses a BLE113, while I use a BLE113-A-M256K.

Thanks in advance for your help!

[my\\_pcb.pdf](#)
[pcb\\_jeff\\_rowberg.pdf](#)

2 people would like this to be answered.

Me too!

## Comments



Jeff Rowberg  
Bluegiga  
Technologies

Hi Simon,

Given the explanation of symptoms in this case, most likely the issue is that you are not properly driving the wake-up pin (P0\_0, active-high) and/or that you do not have the correct baud rate, and/or that you are using packet mode when it is not actually enabled on the module. I am guessing it is the wake-up pin, since you said you received the `system_boot` event correctly. Note that the breakout board's firmware is **not the same** as the factory default firmware on new bare modules, for a couple of different reasons:

- Breakout uses 38400 baud
- Breakout disables sleep mode so no wake-up pin is needed
- Breakout enables packet mode on BGAPI-enabled UART port

The actual project used on those boards can be found here:

- [https://github.com/jrowberg/bglib/tree/master/BLEFirmware/BGLib\\_U1A1P\\_38400\\_noflow](https://github.com/jrowberg/bglib/tree/master/BLEFirmware/BGLib_U1A1P_38400_noflow)

In contrast, the factory default configuration for new BLE modules is described here:

- <https://bluegiga.zendesk.com/entries/80487657-BLExxx-Factory-default-firmware>

Can you confirm that all of your settings, connections, and MCU signalling behavior are correct in light of whatever firmware is actually running on the module?

December 15, 2014, 16:06



Simon  
Deleersnyder

Hi Jeff,

Thanks for your answer.

I flashed both the BLE113 on the breakout and the BLE113 on the PCB with the same bgproj file. I use 4800 baud, disable sleep (so no wake-up pin) and use packet mode. Here's my hardware.xml file:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

&lt;hardware&gt;

&lt;sleeposc enable="true" ppm="30" /&gt;

&lt;sleep enable="false" /&gt;

&lt;txpower power="15" bias="5" /&gt;

&lt;usart mode="packet" channel="0" alternate="1" baud="4800" endpoint="api" flow="false" /&gt;

&lt;port index="0" pull="down" tristatemask="0" /&gt;

&lt;/hardware&gt;

I agree that it looks like the BLE113 is asleep but I don't get why this is the case. Because the one on the breakoutboard is not asleep and I used the same bgproj file.

December 15, 2014, 17:00



Jeff Rowberg  
Bluegiga  
Technologies

Hi Simon,

If the same firmware works in one case and not in another, I would triple-check the electrical connections between the RX and TX pins on the module and the MCU. Can you monitor both pins on the module side using a logic analyzer? Can you send any commands (like "system\_reset" or "hardware\_io\_port\_write") that produce observable effects, so you can tell whether (1) commands aren't making to the module in the first place, or (2) responses aren't coming back?

December 15, 2014, 21:05

Answer



Simon  
Deleersnyder

Hi Jeff,

I do not own a logic analyzer but the data should arrive since I connect the breakout to the very same RX/TX pins and this works. So I'm thinking the problem might be that the BLE is not connected properly to my PCB (I'm new to hot air rework). If you can confirm that the PCB layout I attached to my first post is in fact correct, I think this is the only possible explanation. My PCB layout is however not exactly the same as yours on the breakout, for instance I connected every GND pin on the BLE to ground whereas you only seem to be grounding two of the GND pins. Could this be the reason it is not working?

I will have to wait to do some more testing since I damaged one of the solder pads on my BLE and I have new ones arriving on thursday.

December 16, 2014, 01:34



Simon  
Deleersnyder

Turns out the BLE113 wasn't connected to the PCB after all. It works now.

Thanks for the help!

December 30, 2014, 17:43

Answer



Simon  
Deleersnyder

I meant that it wasn't \*properly\* connected to the PCB.

December 30, 2014, 17:44

#### Add a comment

Save comment