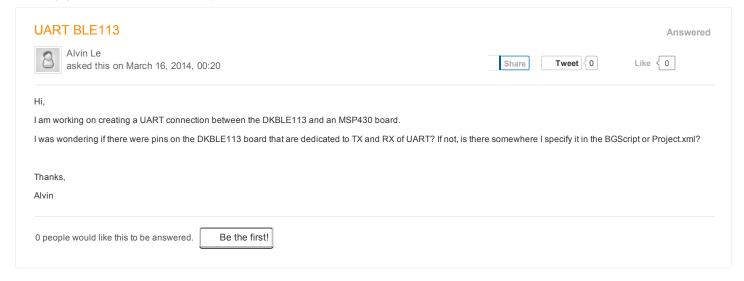
## Bluegiga Forums / Community Forums / Bluetooth Smart



## Comments



Jeff Rowberg Bluegiga Technologies

Hi Alvin,

Answer

The UART pin selection (or whether the UARTs are available at all) is configured inside the project's **hardware.xml** file using the **<usart>** tag. This tag and its attributes are documented in the latest **Bluetooth Smart Configuration Guide** which you can find in the Documentation section of the BLE113 product page. Also, in the BLE113 datasheet, you will find a peripheral pin configuration table which shows the actual pin assignments for the selected **channel/alternate** configuration.

March 17, 2014, 15:21



Why am I getting errors when using UART together with HID over GATT to emulate keyboard? I am using BLE113 UART with configuration <usart channel="1" alternate="2" baud="57600" endpoint="usart" flow="false" />

 $When \ not \ using \ \textbf{call} \ \textbf{gap\_set\_mode} (\textbf{gap\_general\_discoverable}, \textbf{gap\_undirected\_connectable}) \\$ 

everything (uart echo works great) but when using HID (connect with android device to start emulate keyboard), i started to receiving errors on UART... please see screenshot

first part is when not connected, second when starting with emulation... so my question is how to achieve keyboard emulation with data received from uart?

uart\_missing\_wrong\_data.png (quick view)

March 17, 2014, 18:51



Jeff Rowberg Bluegiga Technologies

Hi Željan,

This sounds like a flow control issue. Without flow control, it is difficult to guarantee data integrity. Also, if you are sending data very quickly, there may not be enough space available in the outgoing UART TX buffer (which is 64 bytes). The BLE connection has the highest interrupt priority, and sometimes the UART interrupts cannot be serviced in real time. To ensure that incoming data is reliable, you must use flow control.

See this article for more information:

• https://bluegiga.zendesk.com/entries/23143152-REFERENCE-Using-or-bypassing-flow-control-with-UART-communication

At the very lease, you should try to absolutely minimize the amount of information that has to flow over UART. If you can compress or otherwise encode the important data into only a few bytes instead of a full 12-byte "GRW2512528\r\n" block, the performance may improve.

Also, your **<usart>** tag has an incorrect value for the **endpoint** attribute; the correct value for streaming data via BGScript is **endpoint="none"**.

March 17, 2014, 19:06

Hi Jeff,

Support



Alvin I e

Great, thanks for the information

March 17 2014 22:29



Željan Alduk

Hi Jeff,

thank you for your help. I set hw flow control and now i am sending data, byte by byte and it seem that now works great! :)

I would like to share my project with you and others, maybe someone finds it useful.

You can see it running at: https://www.youtube.com/watch?v=-UMMRQpcmYI

and you can find source attached.

I have also one problem... if sending data to module before module is connected to for example Nexus 7, it will stuck, and reboot of module is needed... can you maybe detect why is this happening?

and also... i want to use this on iPad... is possible to setup some kind of auto reconnect/auto pair after connection is lost? Or is it maybe possible to do this from application on iPad?

rfid\_reader\_hid\_over\_gatt\_keyboard.zip

March 18, 2014, 20:13



Jeff Rowberg Bluegiga Technologies

Hello Željan,

In your code, you have the following:

# disable RX watermark on UART1 while we are processing existing data
call system\_endpoint\_set\_watermarks(system\_endpoint\_uart1, 0, \$ff)

...followed by this further down:

# module is not connected skip further processing
if connected = 0 then return end if

...which means that this code even further down is never executed if you are not connected

# set RX watermark on UART1
call system\_endpoint\_set\_watermarks(system\_endpoint\_uart1, 1, \$ff)

You are disabling the RX watermark and then leaving it in that state, which will cause the data to build up and overflow the UART endpoint RX buffer, causing the behavior you are seeing. You have to ensure that data sent to the module is still processed and discarded, or else not sent to the module at all in the first place.

As for automatic iPad reconnection, something like this would need to be initiated from the iPad side since it will (presumably) be operating as the BLE master (central) device. Your iOS app will need to scan and reconnect when necessary. The BGScript code on the module already resumes advertising automatically when a connection is lost, which is all it can do.

March 19, 2014, 16:21



Željan Alduk

Hi Jeff!

Thank you for clarification, after correction, everything works great!

Is it possible to protect module from unauthorized access with some kind of PIN? I don't want to allow other devices then iPad to be able to connect/pair with module.

March 20, 2014, 13:06



Jeff Rowberg Bluegiga Technologies

Hi Željan,

That depends on how exactly the protection needs to work. I would recommend reading the following KB article and looking at the example project linked here:

- https://bluegiga.zendesk.com/entries/22882472--REFERENCE-Bonding-encryption-and-MITM-protection-with-BLE-modules
- https://bluegiga.zendesk.com/entries/36943196--BGScript-whitelist-peripheral-BLE-peripheral-with-pairing-bonding-and-whitelisting

If you want a fixed PIN that only allows one specific app or device to use the peripheral, then this will probably need to be implemented in your application. You could create a special characteristic which must have a unique value written to it with a normal GATT write operation, and keep track of

## UART BLE113 : Bluegiga Technologies

whether this has happened or not in your BGScript code. The rest of your code should be written to check this variable and then *not* do what it is otherwise supposed to do if the correct value hasn't been written.

March 20, 2014, 14:53



Hi Jeff,

Great suggestion! I will go with custom characteristic. Thanx!

March 20, 2014, 15:26