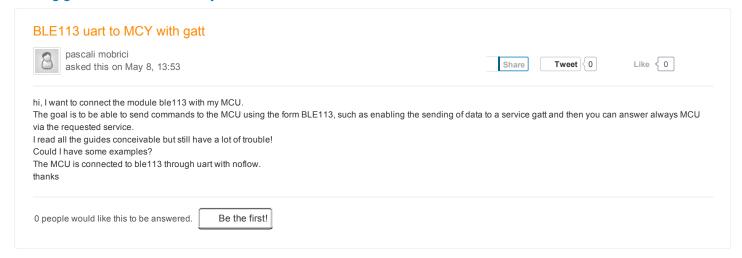
Bluegiga Forums / Community Forums / Bluetooth Smart



Comments



Jeff Rowberg Bluegiga Technologies

Hello Pascali,

The best single resource for this type of development is found here:

• https://bluegiga.zendesk.com/entries/45890933--REFERENCE-BGAPI-BGLib-Implementation-on-BLE-devices

There is an example project attached to that post, plus links to many 3rd-party/unofficial BGLib implementations in various languages.

May 8, 2015, 16:06



pascali mobrici

thank you, I read the guide, but I do not think goes well!

I want to connect my PC (which is installed on a BLED112 using BLE GUI as software) to my device with BLE113 and MCU.

For example I want to create a GATT service on BLE113 named "Temp". When I'm going to read this service from BLE GUI, the BLE113 must send a command to the MCU via uart (eg 0xac 0x53, ie name of the service and stop communication), then the MCU goes to read the temperature from the sensor and re-sent this value to BLE113 (eg send 3 bytes, service name, temperature reading, stop communication ... 0xac 0x19 0x53). Now the BLE113 can update the value of the service GATT "Temp" with the new value (0x19) or 25 ° C and i can read it on the BLE GUI

is possible to have an example?

Thanks in advance

May 11, 2015, 12:39



Jeff Rowberg Bluegiga Technologies

Hello Pascali,

The type of communication that you describe here between the module and MCU (3-byte UART packets) do not use the BGAPI protocol. This kind of comms would require a fully customized BGScript application running on the BLE module to interpret incoming data and send outgoing data back to the MCU. We do not have an example of something like this, although the "uart_echo_packet" project would be the closes starting point to help with UART processing:

https://bluegiga.zendesk.com/entries/28461493--BGScript-uart-echo-packet-UART1-Alt1-loopback-local-echo-with-watermarking-and-packetization

The links that I sent before would have the module configured in BGAPI mode, and then all of the application logic to control connections, GATT transfers, etc. would be present on the MCU instead of in BGScript. But this does not look like what you have in mind.

May 11, 2015, 16:27



another solution?

a guide?

thanks

May 11, 2015, 17:03

Support



Jeff Rowberg Bluegiga Technologies

Hi Pascali,

There are no other guides or solutions besides the two described above. You have to either use BGAPI and and MCU-based application (in which case the first link I sent is the best reference), or a custom BGScript-based implementation. We do not have an example of a BGScript project that does what you discuss here.

May 11, 2015, 17:04



pascali mobrici

now i use the "project_evkit_conn_slave_gatt_server.ble113.xml", in the "cable_replacement" example.

i can send data from BLED112 to BLE113 (so to the MCU), it's perfect.... but i can't send data from the mcu to gatt service!

May 11, 2015, 17:43



pascali mobrici

this code is ok?

Receiving data to UART and forwarding it to Bluetooth connection event system_endpoint_watermark_rx(endpoint, size) if endpoint = system_endpoint_uart1 then in_len = size if in_len > 20 then # Read maximum of 20 bytes in_len = 20 end if

disable RX watermark call system_endpoint_set_watermarks(system_endpoint_uart1, 0, \$ff)

Read data from UART call system endpoint uart1, in len)(result, in len, in(0:in len))

call attributes_write(xgatt_prova, 0, in_len, in(0:in_len))

write data back out the UART1 call system_endpoint_tx(system_endpoint_uart1, in_len, in(0:in_len)) call system_endpoint_set_watermarks(system_endpoint_uart1, 20, \$ff) end if end

i use your project "uart_echo", but not function!!!

i this this product is not so good!

Your guide is not good.

May 13, 2015, 13:21



Jeff Rowberg Bluegiga Technologies

Hello Pascali,

Is there a reason that you are setting a watermark size of 20 bytes? Are you always sending 20-byte packets into the module via the UART interface?

The unmodified "cable_replacement" example project that comes with the SDK has a corresponding application note which you can find under the "Application Notes" section of the BLE module product pages on our website:

 $\bullet \quad \text{https://www.bluegiga.com/en-US/products/ble112-bluetooth-smart-module/\#documentation} \\$

This walks you through how to fully test both RX and TX data using BLEGUI and a BLED112 as the client/master device, and a BLE module running that example firmware as the server/slave device.

May 13, 2015, 23:28

Add a comment	
	•