Bluegiga Forums / Community Forums / Bluetooth Smart

Clearing RX/TX buffer in BGScript

Answered

A C
asked this on August 12, 2014, 21:36

Hello,

I am working on send in sensor data from the MPU-9150 sensor chip to a BLE112 dongle using the BLE112 connected to a microcontroller over UART.

I need each 20 byte packet sent to the dongle to be in a very specific format because software on the dongle-side needs to process it; for example the first 2 bytes represent x-axis gyro data, the next 2 are y-axis gyro, and so on.

I have set the TX watermark on the BLE112 module to 20 bytes as my intention is to send all the sensor data in a single packet to the dongle.

My problem is that before I've sent anything over UART to the BLE112 module, there may be junk in the buffer and hence my format is messed up.

Is there any way to clear the buffer before sending anything over UART to the BLE112 module? I assume so, but where should this be done? I am using the cable replacement demo BGScript code to begin. Or is there a better way to ensure packet format integrity?

Comments

Jeff Rowberg
Bluegiga Technologies

Hi Andrew,

There is another topic that deals with this:

- https://bluegiga.zendesk.com/entries/28275056-How-to-reset-tx-rx-buffers-on-BLE112-3-or-how-to-set-watermark-position-to-0

Specifically check out the comment right before the last one in the thread for an answer.

August 12, 2014, 22:25

A C

Hello Jeff,

Thanks for the reply. I looked at the article, and you mentioned the commands:

```
call system_reg_write($70f8, $80)
```

...and then enable again when you are ready:

```
call system_reg_write($70fb, $c0)
```

I am unsure where would be the best place to call these commands; preferably during setup once connection is established. I tried various places such as

before the call to system_endpoint_set_watermarks, but still I am getting data all over the place.

Thanks,

Andrew

August 12, 2014, 23:13

Hi Andrew,

Have you also tried "**system_endpoint_rx(system_endpoint_uart1, 64)**" just before enabling the RX watermark? (Assuming of course that you are using UART1)

Support Jeff Rowberg Bluegiga Technologies

When you say "data all over the place," does this mean that it is not being packetized correctly into the 20-byte chunks that you want, or that there is actual incorrect data?

August 13, 2014, 00:29

A C

Hello Jeff,

I just tried your suggestion; to test I am using a microcontroller to send a fixed sequence over UART to the BLE112 module. Each sequence

is 20 bytes, the length of my watermark. When I step through the UART sends by the microcontroller one at a time, the sequence is good and

I am not getting any other junk as previously after adding the suggested command before the call to "system_endpoint_set_watermarks" for Rx. However when I run outside of the debugger at full speed, the chunks are no longer aligned. I noticed that one byte from the previous sequence is repeated as the first byte of the next which makes everything go out of alignment. I defined two 20-byte sequences to be sent to run in an infinite while loop.

Also, in your command, is 64 bytes the the size of the Rx buffer in the BLE112?

Sorry I was not specific; when I mentioned "data all over the place" I am referring to data not being packetized into the 20-byte chunk that I want.

Thanks.

August 13, 2014, 18:24

Jeff Rowberg Bluegiga Technologies

Hi Andrew,

This kind of alignment/duplication issue sounds like something that would be a flow control issue. Are you using flow control here, and more importantly, are you 100% sure that the RTS/CTS signal timing is being followed **very** precisely? See this KB article for information on what can happen with certain UART chipsets even when flow control is enabled:

- https://bluegiga.zendesk.com/entries/23143152-Using-flow-control-for-reliable-UART-communication

August 13, 2014, 20:19

A C

Hi Jeff,

Thanks for the article. I am currently running the cable replacement demo BGScript code and not using any flow control, but I should add it in any case as I will eventually increase my baud rate to as high as possible. My MSP430 MCU does not inherently support flow control and thus I will use two GPIO pins with interrupt support. I have been unable to find any example code for the RTS/CTS protocol, but it appears I will simply be testing and driving the pins HIGH or LOW.

For example I could test the CTS pin on the MCU side to make sure it is safe to send data to the BLE112 and if so use my current method to send bytes over UART and drive RTS LOW (safe for BLE112 to send to MCU) while the MCU is sleeping and waiting for data over UART.

Thanks.

August 14, 2014, 03:52

Jeff Rowberg Bluegiga Technologies

Hi Andrew, **Answer**

Running cable replacement without flow control is *extremely* risky unless you are sending very, very small amounts of data infrequently. You can use the GPIO method to implement "half flow control" basically as you describe; normally the MCU doesn't have trouble receiving data, and so the BLE module's CTS pin can be pulled or driven directly to GND all the time. The important thing is to check the state of the BLE module's RTS pin prior to sending each byte. If it is low, go ahead; if it is high, wait.

August 14, 2014, 17:44

Add a comment

Save comment