Bluegiga Forums / Community Forums / Bluetooth Smart

How to reset tx/rx buffers on BLE112(3) or how to set watermark position to 0?

Answered

Sergey Nelyub
asked this on October 21, 2013, 19:30

Share Tweet 0 Like 2

Good evening!

Sometimes it is useful to be sure that Tx and Rx buffers are empty or watermark position is set to zero before starting communications.

For example on the slave side there have been "01234" already in Rx buffer and watermark is set to 20 bytes. I send another 20 bytes. So it will be more than 20 bytes in the buffer and the watermark event will come. But the buffer contains wrong content.

01234012345678901234 instead of 01234567890123456789.

Strangely enough that I can't find corresponding commands in the API Reference.

Could you advice me what can be done? Thank you!

Best regards,

SANEL

3 people would like this to be answered. Me too!

Comments

Sergey Nelyub

Good evening!

Do only I face this problem on a regularly basis?

For example I transmit the following packets from the slave to the master:

"S|0123456789012345678" on the odd packet's number and "R|987654321098765432" on the even packet's number. So the sequence is:

S|0123456789012345678 R|987654321098765432 S|0123456789012345678 R|987654321098765432 S|0123456789012345678R

1 2 3 4 5

Normally I see right packets in the master terminal. But! If I reset the slave or power it off and on I could see that the bytes in the packets might come in the wrong order:

|0123456789012345678R |987654321098765432S |0123456789012345678R |987654321098765432S |0123456789012345678R

1 2 3 4
5

Packet |0123456789012345678R comes in the odd packet number and packet |987654321098765432S comes in the even packet number. Let's look at |0123456789012345678R. The idea is that the substring "|0123456789012345678" originates from the odd packets while the substring "R" originates from the even packets.

That is why I am asking how to reset that buffers to be sure that these buffer are empty before starting any communications. But I can't find the function to fulfill this.

Thank you!

Best regards,

SANEL

November 5, 2013, 18:03

Sergey Nelyub

A little unimportant misprint:

Packets are 20 bytes long. So S|012345678901234567 instead of S|0123456789012345678.

November 5, 2013, 18:35

Support

Sami Kaislasuo
Bluegiga
Technologies

Hi Sergey,

Answer

If you read the the buffer before you send / receive data it will clear the buffer, you can discard the content of the data.

Cheers,

Sami

November 8, 2013, 15:56

Sergey Nelyub

Good day!

Sami, thank you for an answer. It's nice that this problem could be solved. Could you provide me with the functions to read the buffer (or with a simple example)? Thanks a lot!

Best regards,

SANEL

November 14, 2013, 13:14

Sergey Nelyub

Good day!

I haven't found these functions. The problem is taking place again and again. The guess is that after reset watermarks position doesn't set to zero.

Have somebody faced with the same problem? What are these functions to read the buffer and how to set watermark position to 0?

Thanks a lot!

Best regards,

SANEL

February 19, 2014, 15:41

Sami Kaislasuo
Bluegiga
Technologies

Hi Sergey,

You can read the data from endpoint with the following command:

```
      call system_endpoint_rx(endpoint, size)(result, data_len, data_data)
```

And if you want to clear the buffer just discard the read data.

Cheers,

March 12, 2014, 10:39

Sergey Nelyub

Good day!

Thank you for the answer.

I have used this function. No result. Are you sure that this could really help?

The function: call **flash_read_data**(address, length)(data_len, data_data) doesn't suits to BLE112/113.

Best regards,

SANEL

March 12, 2014, 15:07

Jeff Rowberg
Bluegiga
Technologies

Hi Sergey,

The best way that I know of to flush the buffer is by disabling the UART and then enabling it again, which is not typically done but can be done in BGScript. You can use the following commands, which use the undocumented **system_reg_write()** command:

Disable UART receiver before using pin for something else, register **U1CSR**:

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

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

```
call system_reg_write($70f8, $c0)
```

Note that these addresses are specific to **UART1**. You can do the same thing with **UART0**, but the register addresses are different. You would likely want to run these two commands perhaps immediately before enabling the RX watermark again.

For detail, see the page 162 of the CC2540 User Guide document from TI:

- http://www.ti.com/lit/ug/swru191e/swru191e.pdf#page=162

These pages describe the U1*** registers. Previous pages discuss the U0*** registers that you would use for UART0.

March 12, 2014, 16:40

Sergey Nelyub

Good day!

Thank you for the answer! I hope that will solve the problem.

Best regards,

SANEL

March 14, 2014, 11:50