Bluegiga Forums / Community Forums / Bluetooth Smart

Multiple Connections over SPP over BLE Example

Answered

Felipe Herranz
asked this on February 17, 2014, 00:31

| Share | **Tweet** 0 | Like 0 |

Hello everyone.

I am trying to adapt the SPP over BLE Example to handle two connections working as a Master. Besides modifying config.xml file to support multiple connections I am not totally sure how to proceed.

In the **event gap_scan_response(line 428 of the bgscript)** there are these two calls(lines 474 and 475):

call hardware_set_soft_timer(0, HANDLE_TIMER_PINGPONG, 1)
call gap_end_procedure()

I guess these calls will deactivate the scanning and the (ATP)Ping pong mode (Although I am using ATD for the master and ATA for slaves). Should I remove them in order to keep scanning working? Is there something more I should know about this firmware to handle multiple connections?

This example is working really fine and we are using the AT commands so I would wish to keep it as close as possible to the original.

Thank you very much,

One person would like this to be answered.   Me too!

Comments

Felipe Herranz

Hello again.

I have tested removing the **call gap_end_procedure()** and even I could see it is the only call of that kind executed within a connection process still I can not connect more devices(event gap_scan_response does not happen either)

As I said previously I modified the config.xml to allow multiple connections and connection interval is connected as follows:

config_page_3(8:2) = $0010 # connection interval, default 20ms (16 * 1.25ms)

Maybe I am forgetting something... any clue? I will keep trying anyways.

Thanks in advance

February 19, 2014, 02:23

Jeff Rowberg
Bluegiga
Technologies

Hi Felipe,

Answer

The SPP-over-BLE project cannot really be modified for multiple connections without significant effort, and by definition it would no longer be a transparent data pipe, since you would have to modify it to allow simultaneous connection sending and receiving, similar to MUX mode in iWRAP on our classic Bluetooth module. There isn't enough RAM on the module to receive and buffer incoming data from a non-active channel while an active one is being used; you would need to have all channels active at the same time (i.e. MUX mode).

The **gap_end_procedure** call is used for ATH and if the connection times out; otherwise the connection attempt will go forever. You have to manually re-start scanning after a connection is established, if you want to resume scanning.

But fundamentally this is a very complex modification that will have to change many parts of the project code to work reliably, including the actual data stream.

February 19, 2014, 03:12

Felipe Herranz

Thank you very much Jeff,

I will give it a try anyways

February 19, 2014, 11:15

Hi Felipe,

Have you managed to make any progress regarding this issue?