Bluegiga Forums / Community Forums / Bluetooth Smart

Multiple attributes_write in an event

Answered

Rafal Piersiak
asked this on June 30, 2014, 18:49

Share Tweet 0 Like 0

I would like to use attributes_write multiple times in an event, but it only lets me do it once. Any other call to attributes_write doesn't work. Why does that happen?

```
event attributes_status(handle, flags)

if (handle = xgatt_data) && (flags = 2) then

#call system_endpoint_tx(endpoint, 57, "\n\r++ Local CCC set by remote side to start indications ++")
#call system_endpoint_tx(endpoint, 46, "\n\r++ Transparent data exchange can start ++\n\n\r")
call attributes_write(xgatt_data, 0, 19, "\rHello_CO2_Snsr_001")
connected = 1
call system_endpoint_set_watermarks(endpoint, rxWatermarkBuffer, $ff) #set RX watermark

if startup = 1 then
call system_endpoint_tx(endpoint,6,"K 2\r\n")(result) #send K 2 for polling

call hardware_set_soft_timer(8000, 3, 1)

end if

if disconnected = 1 then
call hardware_set_soft_timer(TIMER_ONE_SEC, 0, 0) #set time/32768, handle, reoccurring (0)
call hardware_set_soft_timer(TIMER_FIVE_SEC, 1, 1) #set time/32768, handle, one-shot (1)
disconnected = 0

end if

call attributes_write(xgatt_data, 0, 12, "attstat_if\r\n")

end if

call attributes_write(xgatt_data, 0, 9, "attstat\r\n")

end
```

0 people would like this to be answered. Be the first!

Comments

Jeff Rowberg
Bluegiga Technologies

Hi Rafal,

[THIS RESPONSE NOT APPLICABLE TO ABOVE ISSUE, SEE NEXT RESPONSE]

This will not work because each one of these write operations must be **acknowledged** before you can send another one. The acknowledgement will come in the form of the `attclient_procedure_completed` event, along with relevant connection and attribute handle data. You will need to cascade these write operations so that the 2nd is triggered when the 1st completes, the 3rd is triggered when the 2nd completes, and so on.

June 30, 2014, 19:00

Rafal Piersiak

Does the cascading happen in the attclient_procedure_completed event?

In other embedded systems, if an interrupt occurs, it remembers where it was and comes back to the place where it was interrupted, when the interrupt finishes. Is this true for the BGScript?

Meaning, when I call the first attributes_write, does the code continue to finish executing in the event, so the next thing would be "connected = 1" and so on?

June 30, 2014, 19:04

Hi Rafal,

Answer

Support

Jeff Rowberg
Bluegiga
Technologies

My mistake--while what I said above is correct when using the **attclient_attribute_write** command when writing a value to a remote attribute as a GATT client, and the same concept applies here, the specific API call is different in this case. You are operating here as a GATT server, using **attributes_write** to attempt to push data to a GATT client using **indications**.

Indications are not queued and can only be pushed one at a time, in series, to any given target device. There are three API packets at play here:

1. `attributes_write` command: sent from your MCU, which updates a value in the local GATT database
2. `attributes_write` response: sent back from the BLE module, telling whether the local update succeeded or not
3. `attclient_indicated` event: sent from the BLE module, telling that the remote end received and acknowledged an indication

Because of the way indications are handled, you can send `attributes_write` commands all day long very rapidly and receive back `attributes_write` responses with a **0x0000** result code (no error) every time, but still only have a small portion of those updates end up being pushed to the remote device. The "result" code in the `attributes_write` response packet concerns the local GATT update, not the subsequent indication which *may* happen if the client has subscribed. Instead, because of the one-at-a-time nature of indications, **you must wait for the** `attclient_indicated` **event to come back after each update** before sending using `attributes_write` to push a new value.

To answer your other question: in BGScript, events are executed atomically (all at once) an in the order they are queued. An **attclient_indicated** event which is generated while other code inside your **attributes_status** event handler is executing will be queue, but it will not be executed until after the rest of the **attributes_status** event finishes.

June 30, 2014, 19:22

Add a comment

Save comment