

Bluegiga Forums / Community Forums / Bluetooth Smart

Simple LED switching by using button

Answered



Jeeva Suria Rajah

asked this on December 30, 2013, 17:44

Share

Tweet 0

Like 0

Hi, im creating a simple example by switching on the LED by using button. However i can't find the proper code. Below is the code that i created but it seems not working. I'm making P1.3 as input.

```
dim res
dim port
dim data

# System start/boot listener
event system_boot(major,minor,patch,build,hl_version,protocol,hw)

# configure P0.7 as output
call hardware_io_port_config_direction(0, 192)
call hardware_io_port_write(0, 192, 0)
# configure P1.3 as input
call hardware_io_port_config_direction(1, 7)
call hardware_io_port_write(1, 7, 0)

call hardware_io_port_read(1, 7)(res, port, data)

if data = 1 then
  call hardware_io_port_write(0, 192, 64)
end if

end
```

Hope anyone can help in this.

One person would like this to be answered.

Me too!

Comments



Greg Rowberg

Jeeva,

Answer

There are a couple issues I can see with the example code you provided. With pins being zero-based, remember the bits correlate right to left, 0-7. Your "call hardware_io_port_read(1, 7)(res, port, data)" command is reading pins 1.0, 1.1, and 1.2, which you previously configured as outputs. This isn't fundamentally incorrect, but you're not reading pin 1.3 at all, which is where you have your button input configured. You would need parameter values of (1, 8) for that specific pin, or (1, 15) for all of the first 4 pins (P1.0, P1.1, P1.2, and P1.3).

The way the data is returned isn't as a high/low 1/0 result, but as a full byte for the port. In other words, if you want to check if P1.3 specifically is set high, your if statement would want to read:

```
"if (data & 8) > 0 then"
```

That essentially takes the value you want (00001000), and matches it to the value it reads from the port, which, assuming the button is on P1.3 and connecting that pin to VDD, should also be (00001000). That 4th bit, corresponding to P1.3, will then return with a value of 1, and the if statement will evaluate as true.

One further problem, though, is the location of this code in your program. Having it located in the system_boot event means that this will only take place once, on boot/reset. If the button is not pressed and held during boot, this code will never be executed. Assuming your goal is to allow you to press the button and half the LED go on, this code should go in a separate event.

Along the same lines, in order to get the LED to respond each time you press the button, you would need to set up interrupts in your program. You might want to take a look at this project for more information on implementing interrupt requests:

<https://bluegiga.zendesk.com/entries/23132056--BGScript-gpio-demo-GPIO-setup-and-interrupt-detection>

January 1, 2014, 00:40

Support



Jeeva Suria
Rajah

Thanks very much Greg. I have follow the example that you gave. Now it is working. How ever i have another question regarding the connection of ble112. Currently doing a simple project to control dc motor which is interfaced with BLE112 and control it by Raspberry Pi. I can scan, connect and disconnect the BLE112 from my Raspberry Pi without any problem.

However I'm planning to receive the connection just from raspberry pi only and not from other device. This is because, as far i can use raspberry pi to connect, I also manage to connect the BLE112 from my phone.

I have read about the whitelist feature and try it, unfortunately, it is still not working. If i want to use the directed connection, it is not practical for my project since if i not mistake, the directed connection is just open for 1.28sec only. My project required to received the connection for 24 hours.

So, currently my only option is the undirected connection which is i don't understand to implement it.

I hope for a help. Thanks.

January 3, 2014, 19:50

Add a comment