GPIO - Use GPIO Interrupt to Control LED

Preparation

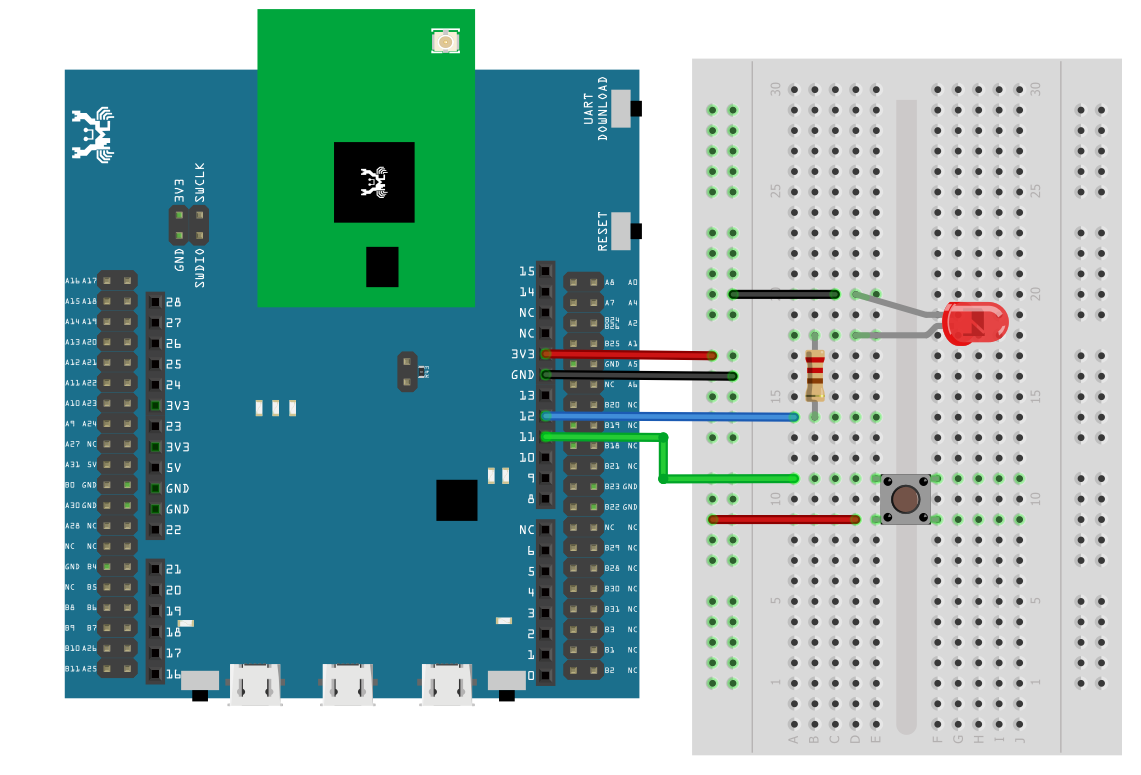
* AmebaD [ [AMB23](https://www.amebaiot.com/amebad/#rtk_amb23) / [AMB21](https://www.amebaiot.com/amebad/#rtk_amb21) / [AMB22](https://www.amebaiot.com/amebad/#rtk_amb22) / [BW16](https://www.amebaiot.com/amebad/#partner_bw16) ] x 1
* LED x 1
* Button x 1

Example

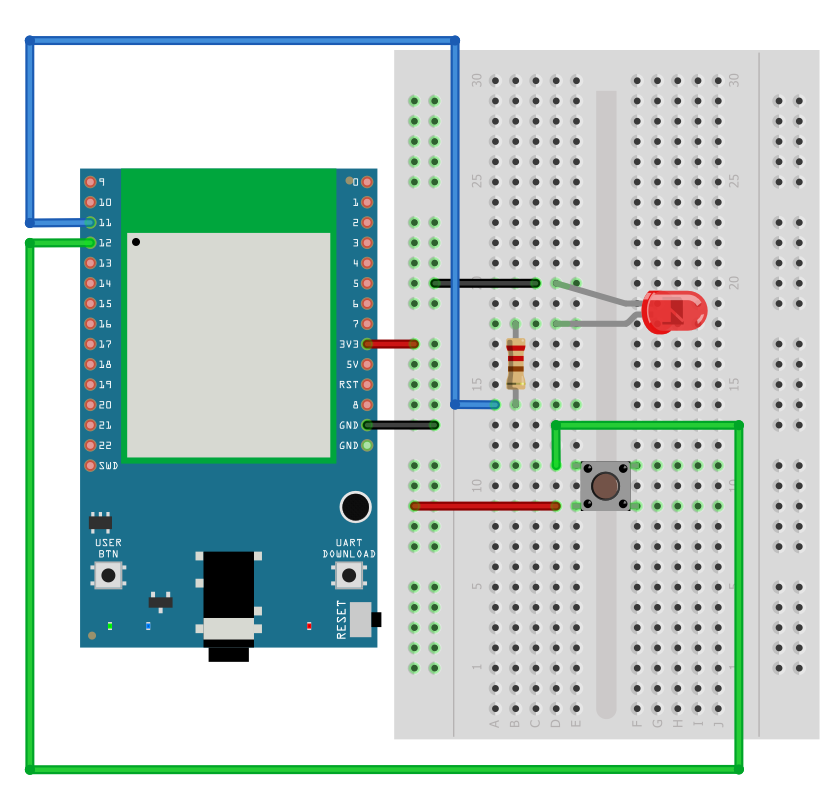
In this example, we use a button to trigger interrupt and control the LED. When we press and release the button, the LED dims, press and release the button again, and the LED lights up.Note that in the Arduino example “Button and LED”, LED only lights up when the button is pressed and hold, when we release the button, the LED dims.

Open the example, “File” -> “Example” -> “AmebaGPIO” -> “LED\_InterruptCtrl”

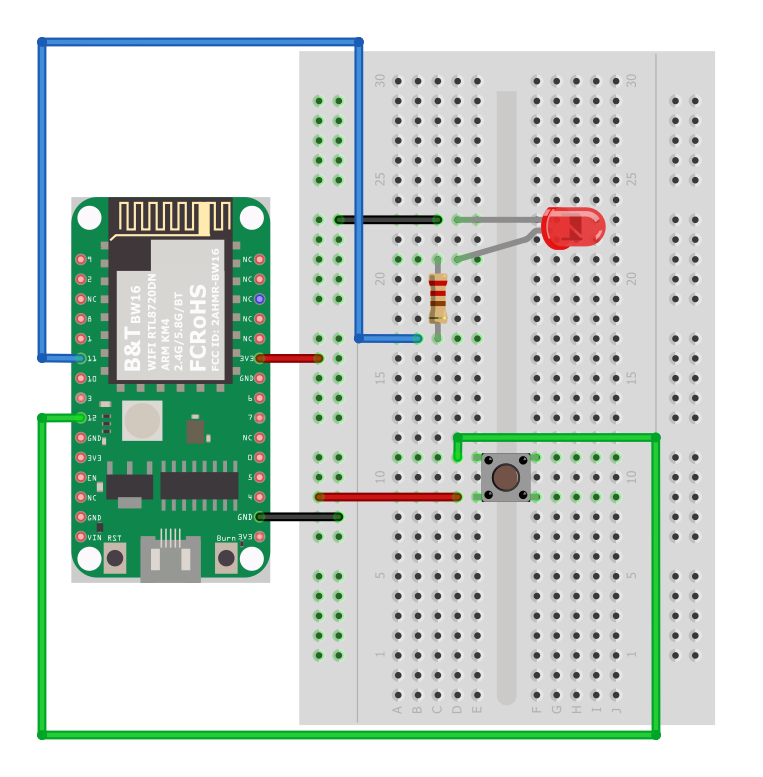
**AMB21/ AMB22**wiring diagram:



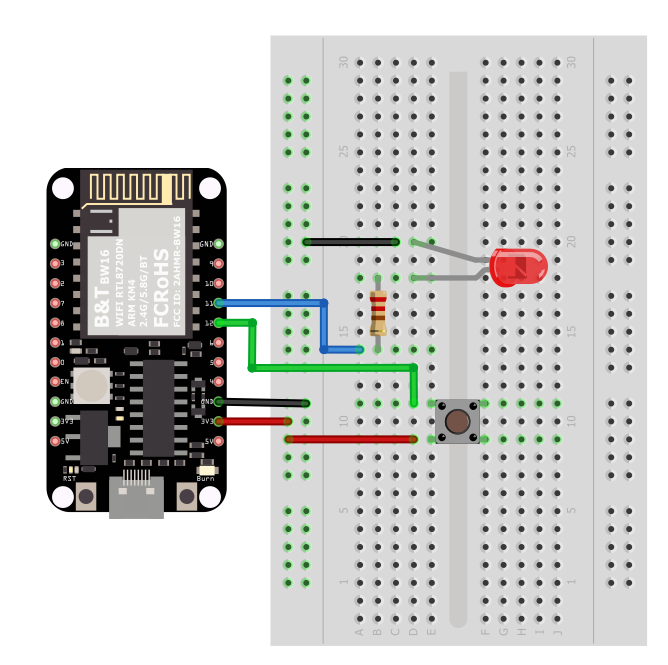
**AMB23**wiring diagram:



**BW16** wiring diagram:



**BW16-TypeC** wiring diagram:



Compile and upload the program, press the reset button on the Ameba.  
The LED will light up at first. Press and release the button, then the LED should dim. Press the button again the LED should light up.

Code Reference

In

setup()

we set Pin 12 to

INPUT\_IRQ\_RISE

, this means that an interrupt occurs when the voltage of this pin changes from GND to 3V3. Therefore, we connect the other side of the button to 3V3, so as to trigger interrupt event when the button is pressed.

pinMode(button, INPUT\_IRQ\_RISE);

On the other hand, we can set pin 12 to

INPUT\_IRQ\_FALL

, this means that an interrupt occurs when the voltage of this pin changes from 3V3 to GND. In this case, the other side of the button is connected to GND. Next, we need to specify the function to be execute to handle the interrupt:

digitalSetIrqHandler(button, button\_handler);

The second parameter is a function pointer, with prototype:

void button\_handler(uint32\_t id, uint32\_t event)

In this handler, every time we press and release the button, we trigger an interrupt, and change the status of the LED.