

## Bluegiga Forums / Community Forums / Bluetooth Smart

## How do I get a buzzer to make noise on BLE113?

Answered



Razvan Mocanu

asked this on December 17, 2013, 17:24

Share

Tweet 0

Like 0

I have a buzzer connected to P1\_0, and I'm trying to make it make noise by doing this:

```
freq=32768/10000
```

```
event hardware_soft_timer(handle)
```

```
  if handle=BUZZ then
```

```
    isBuzzing=1-isBuzzing
```

```
    call hardware_io_port_write(1, 1, isBuzzing)
```

```
  end if
```

```
  call hardware_set_soft_timer(freq, BUZZ, 1)
```

```
end
```

However, nothing happens on the buzzer side. Is that not how you buzz something???

0 people would like this to be answered.

Be the first!

## Comments



Greg Rowberg

Razvan,

Answer

You are likely having a problem because the `hardware_soft_timer` event in BGScript isn't designed for the PWM output your buzzer is looking for. If you check section 5.2.5 of the BGScript dev guide, it states: "In order to generate PWM signals output compare mode needs to be used. PWM output signals can be generated using the timer modulo mode and when channels 1 and 2 are in output compare mode 6 or 7."

Take a look at these two projects for reference:

<https://bluegiga.zendesk.com/entries/23508197--BGScript-pwm-4channel-1khz-Fast-4-channel-PWM-output-at-1-KHz>

<https://bluegiga.zendesk.com/entries/23498301--BGScript-pwm-4channel-1hz-Slow-4-channel-PWM-output-at-1-Hz>

Those projects illustrate how to use hardware PWM output rather than the software timers in BGScript, which is too slow to output a square wave at the speed you need for your buzzer.

December 17, 2013, 21:21



Razvan Mocanu

I have tried implementing the `hardware_timer_comparator` method, but I am unsure as to what I need to change in `hardware.xml` to get the square wave to 'output' on the port that I need it to.

In the example it talks about ports p0.0, 0.1, 0.6 and 0.7, but the 'enabled\_channels' property says 1f(00011111) so ports 0-4. I've tried to look for an explanation to what goes into the `<timer>` tag for `hardware.xml`, but cannot find anything.

Could you please enlighten me as to what I'm not doing right?

December 19, 2013, 18:26



Greg Rowberg

Razvan,

Answer

The only line you need to add to your `hardware.xml` file is:

```
<timer index="1" enabled_channels="0x1f " divisor="0" mode="2" alternate="2" />
```

The "`hardware_timer_comparator`" command has the ability to output 4 PWM channels simultaneously. For your project, you only need one output. The example shows 5 call commands, using channels 0-4 (5 total). Channel 0 establishes the baseline frequency for your PWM output. In the case of the 1khz example, it explains that default speed on the timer is 32MHz.. For an audible tone out of your buzzer, a common frequency is the 440Hz range. For a solid tone, you'll probably want to utilize the 50% duty cycle shown with channel 1. Assuming a ~440Hz desired tone, your BGScript file should include:

```
#channel 0 establishes the baseline frequency (32,000,000 / 73000 = 438.35 Hz)
call hardware_timer_comparator(1, 0, 6, 73000)
```

```
#channel 1 is 50% duty cycle (36500 / 73000 = 0.5)
call hardware_timer_comparator(1, 1, 6, 36500)
```

If you check the BLE113 Datasheet, you'll see which pins are mapped where on the devkit using the pin-out on page 7. Channels 1, 2, 3, and 4 output to pins p1.1, p1.0, p0.7, and p0.6, respectively. You can output to any of those 4 pins, depending on which channel you use. In the case above, we're using channel 1, so your output will be on p1.1 (Pin 23) on the devkit.

December 20, 2013, 04:04



Razvan Mocanu

Hello again,

I have followed your instructions, and I now know what the 5 channels are about.

We have the pin1.0 as the output for the buzzer, so I'm using channel 2 for the output (pin 24). The specifications for our buzzer require 4800Hz ± 500Hz. This is the code I'm using:

```
# channel 0 establishes the baseline frequency (32,000,000 Hz / 6400 = 5,000 Hz)
call hardware_timer_comparator(1, 0, 6, 6400)

# 'output' the square wave needed to generate noise on channel 2 (P1.0)
call hardware_timer_comparator(1, 2, 6, 3200)
```

This code is in **system\_boot**, so it should run as soon as the device boots up, right? Well, we can't hear anything coming out of the device.

I also tried running it with your example, at 440Hz, and still no sound.

December 20, 2013, 15:59



Greg Rowberg

Razvan,

Answer

After doing a bit of research, it turns out the fourth parameter in **hardware\_timer\_comparator** is a uint16 value, which means my previous recommendation of 73000 wouldn't work (values > 65535 I believe will just overflow). On the other hand, 5000hz is an extremely high frequency tone, and it's a little bit surprising to see a buzzer with a range that high. Do you have a link to product documentation for the buzzer you're using?

I would recommend trying it again at 1Khz:

```
# channel 0 establishes the baseline frequency (32,000,000 Hz / 32000 = 1,000 Hz)
call hardware_timer_comparator(1, 0, 6, 32000)

# 'output' the square wave needed to generate noise on channel 2 (P1.0)
call hardware_timer_comparator(1, 2, 6, 16000)
```

And see if you can get any tone out of your buzzer. A few other simple and practical troubleshooting recommendations:

- If you have a logic analyzer or analog scope, try testing the the output on P1.0 to see if it's behaving as it should. If you don't have access to either of those, a simple volt meter should at least tell you whether you're getting any data sent to the pin.
- If you have another PWM output system, you could test the buzzer with that to make sure it's functioning properly.
- You could check to make sure the buzzer's positive pin is connected to P1.0, negative to ground, and that it's compatible with a 3.3V system (which is what the BLE113 is).

Yes, placing the code in **system\_boot** should cause it to run on start up. If you'd like, you could paste your BGScript file, if you'd like confirmation on the structure.

December 23, 2013, 08:50



Razvan Mocanu

We've tested our hardware with an arduino board and it works fine. Frequencies up to 5.2kHz  
Here is the product page for our buzzer

<http://www.puiaudio.com/product-detail.aspx?categoryId=5&partnumber=AB1548B>

January 13, 2014, 17:20

Hi Greg, did you have a chance to look at this again? I am trying to find an answer...

January 15, 2014, 22:33



Ray Sark



Razvan Mocanu

Hi Greg and Ray

Answer

Update on the buzzer not buzzing:

It now buzzes properly, and I can make it stop buzzing by writing 0 values for the mode and frequency values.

January 17, 2014, 13:15



KRUKUSA

Would you mind sharing the code? @ Razvan

April 17, 2014, 02:41

#### Add a comment

Save comment