

MT-101 Lesson 2-5: ModusToolbox PWM

Hello. My name is Alan Hawse. Welcome to Cypress Academy, ModusToolbox 101.

In this video I'm going to show you how to add some additional functionality to the BlinkyLED project.

First make a new application, choose the CY8CPROTO-062-WIFI-BT devkit, and the BlinkyLED template.

You might recall from an earlier video that this project was built for a different development kit so let's fix that. Run the configurator by double clicking design.modus and go to the pins tab.

Deselect Port0/pin3, enable Port 13/pin7, give it the alias LED_RED and change it to strong drive output with the input buffer off. Hit save and then close the configurator.

I want to make sure that things are working, so go ahead and press program.

Allright... good I got a blinking LED. Notice that it's blinking at $\frac{1}{2}$ Hz. It inverts the LED, then waits 1000 milliseconds – let's see – 1000 milliseconds – oh yeah, 1 second. That means it goes through the loop once every 2 seconds.

Instead of blinking this LED with firmware, how about I use one of the hardware blocks in the PSoC - namely the TCPWM - which means the timer counter pulse width modulator.

Augusto our CTO got a little bit out of control when he named it, but I suppose its accurate.

In order to do that I'll use our GUI for configuring the PSoC 6 – that's a new Cypress word – well, actually that's an Alan word. Alright, so let's configure the TCPWM.

Start by opening up the _config project then click on design.modus to start the configurator again.

Next, select TCPWM Counter 3, which happens to connect to the pin that the LED is on. Give it the alias RED_PWM. Give it a period of 1000 and a compare of 500. Then select the clock as 16-bit divider 0. Finally pick the pin that we want the PWM_n signal to drive which is the LED pin of this development kit which is connected to Port13, Pin 7.

Notice that Modus created a macro for us called RED_PWM_HW, remember, that's the alias that we gave it in the configurator. That #define has the base address of the PWM that you are using, RED_PWM_NUM which has the counter number, and RED_PWM_MASK which has the counter mask. All three of those macros are used as inputs to the functions.

Now we have to set the input clock to give us a 1kHz input so that our PWM divider of 1000 will give us a 1Hz output.

Click on the peripheral clocks tab. Notice that the 16-bit divider is 0 is already selected. That's pretty convenient. Click on that divider and setup the divider to be 50000. The customizer tells you that the peripheral clock is setup to be 50MHz, so let's see here 50MHz divided by 50000 – let's see here – Georgia Tech – yeah, yeah, yeah. OK, that will give you a 1kHz output. Perfect.

Now press save. The magic starts now as Modus Toolbox turns your configuration that you've been setting up with our GUI into usable source code.

Back in the IDE, you can initialize the PWM using the Cypress PDL functions. Remember, you can read about those in Cypress PDL API which is available on the help menu.

In main.c call the initialization function, the enable multiple function and the trigger start. On the screen you'll be able to see the actual parameters that I used. The macros that the configurator created for us are a big help here.

Then empty out the while(1) loop so that the MCU won't be doing anything.

Now when you program the development kit... look, once again you have a blinking LED but this time the work is being done in hardware with no MCU involvement – the MCU is now free to do other work in your systems. Note that the PWM has a period of 1Hz and a duty cycle of 50% so it toggles every 500ms. You can see that it's now blinking twice as fast as our original firmware, so we know that things are working.

That's all for this time. If you have any questions or general comments, by all means post them in the ModusToolbox community. If you have a personal note or really if you just want to chat, please feel free to email me at alan_hawse@cypress.com or tweet me @askiotexpert. Go forth and configurate! Thank you guys for your time.