

MT-101 Lesson 1-2: ModusToolbox Blinky LED

Welcome back to Cypress Academy, ModusToolbox 101. In this video I'll show you how to build your first PSoC 6 MCU application using ModusToolbox and the PSoC 6 WiFi-BT Prototyping kit. This video will put you on track for designing awesome IoT applications using ModusToolbox, PSoC 6 MCUs, and WICED wireless connectivity devices.

As customary, we're going to start off with a Hello World application to make sure that everything is working. In MCUs, the Hello World application is the blinking LED so that's what we're going to do here.

First, open ModusToolbox.

Select an existing workspace or enter your own name to create a new one. I like to create a new workspace for each new application under the default folder, which is mtw in your home directory. For this example, I'll use the name MT101-blinky and I'll put it under the mtw directory.

Before we create a new application, we'll take a quick look at the information available on the Welcome page.

The Welcome page has four main sections:

#1: Create a New Application

#2: Configure a PSoC Device

#3: Add in Middleware

And #4: Build and Debug your Application

And interestingly enough, those are like the four steps you'll be going through.

Each of these sections has a short description, a link to the appropriate section in the Help guide, a link to a short video covering these portions of the design flow.

Once you close this window by clicking on the X, it won't show up again until you create a new Workspace or install a new version of ModusToolbox. You can also view it again at any time by selecting Welcome from the Help menu.

Next let's select New Application from the Quick Panel on the left-side of the screen to launch the wizard.

Select Dev/Eval Kit to choose the CY8CPROTO-062-4343W kit, click Next.

Now you will see a list of starter applications. This gives you a variety of choices as a starting point ranging from an empty starter design all the way up to applications with some pretty complex functionality like CapSense, Amazon Web Services using MQTT, and even combination Bluetooth/WiFi applications. There is a Browse button where you can import other code examples that we posted on GitHub. We will talk about that more in detail in a later video. For our application, let's choose the BlinkyLED and click Next.

The final window in the new application wizard is just a summary of what you've already chosen. Review the summary and when you're happy, click Finish.

After clicking Finish, the IDE reads a file called modus.mk to identify all the source files including all of the Middleware being used in your application. During this step all the build dependencies are also set up automatically for you.

In a few moments you will have an Application consisting of a set of Eclipse projects shown in the Project Explorer on the top-left portion of your screen. In general, you'll only need to worry about the `_mainapp` project. That's where you will add your source code to build your specific project.

Now is a good time to talk about the little Quick Panel that's in the bottom left portion of the screen. First, make sure you click the BlinkyLED_mainapp in the Project Explorer. Notice in the Quick Panel that there's three regions:

First is the Start Region which contains Application-independent quick links such as creating a new ModusToolbox application or locating an on-line example. We will cover on-line code examples in detail in a later video, but we're super excited that we're going to be deploying a bunch of them in GitHub.

Next, is the selected Application (or Project) region, which is device specific. It contains pre-configured links for building or cleaning your full ModusToolbox Application. There are also links to adjust the build settings for the selected project, to configure your device, and to select from a list of available middleware such as CapSense, HTTP, cryptography, etc.

Finally, there is the Launches Region – this area has Eclipse run and debug configurations pre-made for you. Notice that there are four configurations for this example.

We will be using the KitProg3 which is built into the kit and we'll be using that for all of these videos. Nothing but a USB cable is required to program and debug.

Now, let's open up the source code to take a look. The first thing your'e going to notice is that there's several projects. There may be as few as 2 or 3, or there may be more like 10 or possibly 20 projects. All of these projects are individual Eclipse projects that get combined by our build system into one application. So, we'll always use the word application to mean the combination of the projects that you see in the project explorer.

Alright, in the project explorer, navigate to BlinkyLED_mainapp->Source->main.c.

As you can see, we have comments at the top explaining what this application does, then some includes to get the PDL API functions and the device configuration. In main, we initialize everything, enable interrupts, and then in the infinite loop we just toggle the LED, which we've named LED_RED, once per second.

In the panel on the right you can see that the includes and functions are shown. Variables and macros would also be shown there if we had included them in the project. You can click on items to jump to that location in your source file. This isn't really necessary for such a simple example as the blinking LED, but as you start working on more complex projects, this can help you navigate your source code.

Now let's talk about the device configurator. The configurators are used for setting up things like pins, the global resources like the clocks and peripherals such as PWMs, UARTs, CapSense and so on.

This starter example can work with a number of different Cypress kits. For our kit, we want to blink LED4 which is on Port 13, Pin 7. If you have an electron microscope, you can actually see on the board silkscreen that LED4 is labeled "USER (13.7)". Yes, I'm working with the kit team – I promise you in the future we'll have bigger silkscreen. Actually, they tell me it's not the silkscreen that's the problem – it's me getting old and not being able to see it very well. So, there it is.

Alright, we will use the device configurator to select that pin.

Click on Configure Device in the Quick Panel.

This example is set up by default to work for another kit where the LED is on Port 0, Pin 3. We need to move it to Port 13, Pin 7 to work with this exact kit. So, in the configurator, go to the Pins tab and find P0[3]. Notice that it has an alias of LED_RED. Let's de-select that pin.

Now find P13[7] - also known as Port 13, Pin 7 - and select it. Make sure the Drive Mode in the Parameters selection is set to Strong Drive, Input buffer off, and the Initial Drive State is High.

In the text area to the right of P13[7], give the pin an alias to help with the code readability. Enter in a value there. Maybe something like MYLED.

Notice the Code Preview area on the right. It shows you how ModusToolbox will set up all of your code for you. Now change the alias to LED_RED since that's what main.c is expecting and then look – watch the code update in the Code Preview window.

Save the configuration with control-S or File->Save and exit the configurator.

Now click on Build BlinkyLED Application in the Quick Panel.

After the build is complete make sure your kit is connected to the KitProg USB connector.

You notice on this board there are two connectors. The one on this side is KitProg, the one on this side is connected to the PSoC 6. If you plug into this port and try to program you're going to find yourself frustrated so make sure and plug into this one.

Then, select BlinkyLED_mainapp in the Project Explorer, and click on the “BlinkyLED Program (KitProg3)”.

You should eventually see a message that says **** Program operation completed successfully ****

The LED4, located towards the center of the board, should be blinking now. If you like, you can change the value of the delay and program it again.

If you have any questions or general comments, please post them in the ModusToolbox community on cypress.com. If you have a personal note or just want to chat, please feel free to email me at alan_hawse@cypress.com or tweet me [@askiotexpert](https://twitter.com/askiotexpert). Thank you.