

**PSoC Academy: How to Create a PSoC BLE iOS App**  
**Lesson 5: Debug the Firmware**

- 00:00:08 Hi. I'm Alan Hawse. This is Cypress Academy. Let's get going. In the last lessons, we created the schematic, set the pins, and wrote the firmware for a CapSense LED PSoC 4 BLE project. Now we'll program it into the board. So, I'll press the program button, flash starts going, and then you look down and, as soon as the programming finishes, just like we said, the blue light starts blinking. The blue light indicates that the stack has gotten going, and has turned on the PWM to drive the blinking process in the LED.
- 00:00:45 So, in this next couple minutes, I'm going to show you how to debug your project. We're going to debug it with two different tools. The first one is the CySmart BLE Dongle. So, let's plug this thing into my computer. All right. It's all good. I'm running VMware so I need to tell it to connect to my Windows installation. Then I'm going to run the program CySmart. There it is. Cool. So CySmart starts up, and, it searches for the Dongle. I found the Dongle, and I want to connect to it. This brings up a lot of options.
- 00:01:22 The first thing I want to do is I want to tell the Dongle to start scanning the universe of BLE devices that it can hear. So I press Scan, and there is my device – capped. You remember I gave it a name and the reason I gave it a name in the GAP settings was so that I would easily be able to find it when it was scanning. I'll

**PSoC Academy: How to Create a PSoC BLE iOS App**  
**Lesson 5: Debug the Firmware**

connect to it.

00:01:44 As soon as I press Connect, it establishes a BLE connection, and if you'll remember, when there's a connection established, we turn off the blinking light, and you can see on my board the blinking light is no longer doing its thing. So the first thing that we need to do is tell CySmart to interrogate the BLE stack and ask it what are all of the attributes that it knows about. So I press Discover All Attributes, and it finds out that I've created a primary service.

00:02:16 That service has got two characteristics. The first characteristic is that blah blah blah one that we talked about, and the second characteristic is the blah blah blah two that we talked about. So. If you'll recall, we set the characteristic user description up so we could find the name. That has the UUID of 2901, and that's assigned by the Bluetooth SIG. So when we read that value, it says, Oh, that characteristic is led uint8.

00:02:47 So we're connected to the correct characteristic. So what we would like to do is actually write into that characteristic. Remember that we said when you write a 1 it will light the red light, and when you write a 0 it will turn it off. So, let's start by reading the characteristic, and we find that the characteristic is 0, which means the light is off. That corresponds to what I can see on the board.

**PSoC Academy: How to Create a PSoC BLE iOS App**  
**Lesson 5: Debug the Firmware**

- 00:03:15     Awesome. Now, let's write a 1 and see what happens. Oh my god! The red light turned on. It actually works. That's excellent. So, when I read it again, it still reads a 1. That means that the GATT database has got a 1 in it, so that's good. The next thing we can do is look at the CapSense characteristic. You recall that we called it blah blah blah blah F2. So I click on the characteristic and I read its value.
- 00:03:43     CySmart comes back with FF FF. You'll recall the CapSense returns FF FF, also known as all ones, when there's no touch on the sensor. So, I'm not touching the sensor, so it should be all Fs. Now, when I put my finger on the sensor and click Read, I should get some number that's not that. And in fact I get 21 hex which is about a third of the way down the sensor. 64 hex is 100. So, 21 hex is about a third of that.
- 00:04:19     So if I move it to the other end and read it again, I'll find that it's 57, so I'm near the end, so that makes sense. I don't touch. Read again. All Fs. That's cool. Now, you'll remember that we set it up so that I could get notifications when that changes. So what I'm going to do is I'm going to tell CySmart to turn on Notifications. And what it does is, it writes a 1 into the Notify flag in the CCCD. After that, our update CapSense function will run every time it gets a new value for CapSense. So, I'll turn on Notifications, and you can

**PSoC Academy: How to Create a PSoC BLE iOS App**  
**Lesson 5: Debug the Firmware**

see the FF FFs are streaming, as they get updates every 20 or 30 milliseconds, on the bottom of my screen. And then, as I touch it, you can see that I get constant updates from the CapSense.

00:05:15 So I'll turn off Notifications, disable all Notifications, and then I'm going to write a 0 to turn off the LED, which I'll do, WriteValue 0. The LED goes off, and then I'll disconnect from that device, and then it will start blinking again, because a disconnection event turns the PWM on. So now we've used the CySmart BLE Dongle to debug our firmware. The other thing you can do is, we provide an application called CySmart which you can get out of the app store, and it runs on your iPhone.

00:05:52 So I'm going to run that. So I'll start CySmart, and I'll tell it to search for devices. The first device it finds is capled. That's good. Remember, in the GAP Advertisement Packet, we advertise the name of the device, and we called it capled. So I'll click on capled and the blue light will go off, because we're connected, and it brings up a GATT database browser.

00:06:20 So, the first thing I'll do is I'll click on the GATT database browser, and it'll say, the GATT database has one unknown service. So when I click on the service, it says, Oh, there's two unknown characteristics. I know that the first characteristic is that LED. So I'll press Write, to write into that LED - 01. Okay. The LED comes

**PSoC Academy: How to Create a PSoC BLE iOS App**  
**Lesson 5: Debug the Firmware**

on. That means this phone is connected to the GATT database on the GATT peripheral on the board called capled.

00:06:54 So this time I'll write a 0 to turn off the LED. Okay. Good. That works. And then the last thing I'll show you is that I can read the CapSense characteristic. So I'll ask the CapSense what's its value? When I press the Read, it says it's all Fs. When I put a finger on, it says it's 25 on that end. I'll come down to this other end. It goes to 09, a lower number. And at the top end, I press it and it's a higher number.

00:07:25 That's it. So I'll disconnect by going backwards. As soon as the disconnect event happens, the LED starts going again, and CySmart says it's disconnected. So you can easily use either the PC-based tool for CySmart, paired with the Dongle, or you can use your iPhone or your Android – we provide CySmart for both of those platforms – to browse your GATT database and write values and read values from the GATT database.