

PSoC Academy: How to Create a PSoC BLE iOS App

Lesson 2: Configure the CapSense Component

1

00:00:09 Hello, I'm Alan Hawse. Welcome to Cypress Academy. In this video we're going to show you how to create a project that we'll use to connect to the IOS app. So first we need to make firmware that we'll be able to connect to. I'm going to build the project, so that it starts with a blinking blue LED when the device is not connected. It's got a red LED that you can turn on and off from the app and CapSense slider that you'll be able to read the values of.

00:00:39 All right. First you need to start by creating a new project. So let's see here. File, new, project. And this is going to be "PSoC 4100BLE / PSoC 4200BLE". And we'll call this the "capsenseled" project. Okay, we'll start with an empty schematic. Go. Once we got the project created we'll start adding the components that we'll need for this project in the schematic.

00:01:08 We'll start by putting the BLE component from the component catalogue into our project. Then we'll add the CapSense component, which we'll use for the CapSense slider. So we'll grab the CapSense component and put it on the screen. Then I'll need two pins. One pin for the red LED and one pin for the blue LED. So I'll get a digital pin. So we have a digital output pin.

00:01:38 And I'll copy it on the screen. Change its name to red. And set its initial drive state to high because the LED is active low and we would like the LED to be off when the chip turns on. Then I'll get

PSoC Academy: How to Create a PSoC BLE iOS App
Lesson 2: Configure the CapSense Component

2

another digital pin – digital output pin. Copy it on the screen. And this one we'll call blue. Then I'll need a PWM to drive the blue output. So I'll grab the UDB based PWM from the catalog.

00:02:13 And because the blue LED is active low I'm going to want to invert the output of the PWM. So I'll grab a Not gate out of the library. Put it on the screen. Connect it to the output of the PWM and then connect the blue LED to that output. All right. Great. So then I'm going to change the configuration of PWM, so that it's one output PWM.

00:02:45 And then I'll need a clock to drive the PWM. So I'll grab a clock component out of the library. We'll set this clock to be driving this PWM, and I'll configure it to one kilohertz. So this will give us a blinking LED. I would like to not leave the reset input of this PWM just hanging. So I'll attach it to a logic low, so it's an active high.

00:03:16 Logic – grab a logic low. Connect it to reset pin. All right. I'm going to change the name of the PWM so I've got a sensible name to interface with it. Okay. Good. This circuit is the PWM interface circuit. It drives the blue LED when you're not connected, and when you're connected it will stop driving it. The red LED is going to be the LED that we'll switch from our app. I'm going to disable the hardware connection as I'll only interface to this pin from the software.

00:03:49 When you flip it inside of your app it will turn on, and when you

PSoC Academy: How to Create a PSoC BLE iOS App
Lesson 2: Configure the CapSense Component

3

flip it the other way it will turn off. All right. So now I need to configure the CapSense component. And I'm going to start by giving it the name capsense because I like to type less, and I'm going to add a linear slider to this project. Okay, cool. Five sensors because the board has got five sensors on it. That's good. Oh, I'm going to change the name of BLE to make that just a hair simpler.

00:04:20 All right. At this point we have our schematic completely configured. In the next lesson we're going to configure the BLE, which is to set up the profile, set up the services and set up the characteristics that you'll use to interface to your app.