Lesson 4: Using the EZ-PD Configuration Utility

Hello. I'm Alan Hawse. Welcome back to Cypress Academy. In this video, I'll change the default behavior of the CCG3 device that's on the CY4531 kit using the EZ-PD Configuration Utility.

Programmability – in fact RE-programmability – is one of the hallmark features of the Cypress Type-C products. This programmability allows you to change your device's functionality when the USB spec changes, or when you need to control a different AC to DC adapter circuit, or a different multiplexor, or when you need to support multiple device capabilities and Vendor or Product IDs in a single design.

Cypress is embedded in tomorrow. Your tomorrow. We don't know what's going to happen and our Type-C programmability provides you protection against a changing world.

First, I'll change jumper 3 to short pins 2 and 3 so that the kit will be powered from the USB mini connector and then I'll connect the kit to a USB port on my laptop using that connector. This is going to allow me to re-program the CCG3 device using the Cypress USB to serial bridge chip that's functioning as a programmer in this dev kit. The bridge connects the USB mini connector on one side and the CCG3 chip on the other side and allows the CCG3 to be reprogrammed.

Now I'll open the EZ-PD Configuration Utility. Note that the software sees the CCG device that's already connected.

The first thing I'll do in the utility is click the "Read from Device" icon. I'll select "NOTEBOOK". When I do that, notice that on the right side of the window it shows me that there's two versions of the firmware – the running firmware and the alternate firmware. This is because the CCG3 device contains two firmware images. This protects you from bad things happening, by allowing the Type-C functionality to continue to operate while an alternate firmware is updated. It also allows for failsafe operation if a firmware update fails or for any other stupid reason.

Note that in your kit, the running firmware may be either FW1 or FW2. It doesn't matter which one is currently running in your case since we are going to update the alternate version, whichever one that is.

Also notice that by default if I read the device configuration information it will read the alternate device firmware configuration. This operation can be done while the primary firmware continues to run. If instead I want to read the running firmware, I just click on the "Bootloader Read" check box first. This will read the running firmware configuration but it will also cause the chip to enter the bootloader mode so the Type-C functionality will no longer be operating. I won’t check the box so it will read the alternate firmware configuration.

Once the configuration is read, I can see all of the settings for the chosen firmware image. There are lots of configurable settings under various menus such as the Device IDs which you may need to change for your specific product’s requirements.

For now, let’s look at the power data objects - PDOs. Oh great, another TLA (three-letter acronym). Here I can see all of the power profiles that the kit can supply – these are called Source PDOs - and all of the power profiles that it can receive – these are called Sink PDOs. Note that there are 5 Sink PDOs specified. In this case, they are: 5V/900mA, 9V/900mA, 14.8V/900mA, 15V/900mA, and 20V/900mA. They are specified in 50mV increments and 10mA increments. That means 10V will show 200 and 1000mA will show 100.

If you remember from the last video, the Type-C power adapter can supply two power profiles: 5V at 1.01A and 14.8V at 1.4A. Since the kit can accept 14.8V and requests 900mA, which is obvously less than the 1.4A max that the adapter can supply, that is what it requests when I connect the power adapter – it asks for the higher voltage since that will allow faster charging.

But, what would happen if I change the 14.8V at 900mA Sink PDO on the kit to be 12V instead of 14.8V? Let’s find out. First click on the “Sink PDO 2” which is the one for 14.8V at 900mA. The voltage is shown in units of 50mV and the current is shown in units of 10mA, so for 12V I’ll change it to 240 from 296.

Now, all I need to do is update the device with this new setting. First I save the new configuration. Then, I click on the “Configure Device” icon (or use Tools > Configure Device). Then I select “NOTEBOOK” and again I decide which version of the firmware to update. I’ll use the default setting of “Normal” to update the alternate firmware and then I’ll click “Program”.

Once programming is finished, I will reset the board and then click on “Read from Device” again. Notice that the running firmware and the alternate firmware have swapped. This is because the CCG3 automatically runs the most recent valid firmware image which in this case is the one I just updated. At this point, I could repeat the Configure Device steps again to update the other firmware image, but I won’t do that now since the updated one is already running. In a production design you would typically want both versions of the firmware to be the same.

Now, I’ll test out the change. First, disconnect the USB Mini-B connector and change J3 back to pins 1 and 2, or Run Mode. Then, connect the EZ-PD Protocol Analyzer to the PC and to the Type-C port on the kit. Run the EZ-PD Analyzer Utility and start capturing data. Finally, plug in the Type-C power adapter and see what happens. Notice that the power contract is established to be 5V this time because the kit no longer accepts the 14.8V power profile.

OK, now let’s change the firmware back. Disconnect the Type-C power adapter, move J3 to pins 2 and 3, and then reconnect the USB Mini. Use the EZ-PD Configuration Utility to read from the device, but this time read the running firmware instead of the alternate firmware configuration. Go to “Sink PDO 2” and change the voltage back to 14.8V by entering 296. Save the changes and configure the device. I want to update the running firmware so I select “Bootloader Flashing”.

Once programming is done, disconnect the USB Mini-B, and change J3 back to pins 1 and 2. Now if I plug in the Type-C power adapter, it will establish a power contract at 14.8V again. Cool!

So now I have changed the configuration of a Cypress CCG3 device.

The bottom line is that the EZ-PD Configuration utility allows you to customize your CCGx solution to reflect the requirements of your system without a bunch of software changes.

However, sometimes you need to do something truly custom, so in the next video, I’ll show you how customize the solution even more by adding new functionality using the software development kit. As always, you are welcome to email me at alan_hawse@cypress.com or tweet me @askiotexpert with your comments, suggestions, criticisms, and questions. Thank you.