**PSoC Creator 101: CY8CKIT-049 Prototyping Kit New Bootloadable**

My name is Alan Hawse; I'm Executive Vice President of Software at Cypress Semiconductor.

In the introduction to the 049 Prototyping Kit lesson, you successfully modified a bootloadable application, downloaded it to the dev kit and got familiar with the bootloader host tool. Now, you will create a bootloadable application from scratch. To program the kit with this application, you will use the bootloader host to talk to the already programmed bootloader on the PSoC. That bootloader will reprogram the flash with your newly created bootloadable application.

To start, create a new project for the CY8CKIT-049 kit, select an empty schematic, and save it into a suitable location. Once you have created the bootloadable project, add the bootloadable component from the component catalog. Then associate the component with the bootloader that is already programmed into the device by using the customizer.

There are two tabs on the bootloader customizer; in the first one you can enter custom values to identify your application. These values are written into the flash memory and are typically used by the bootloader to learn about the currently loaded bootloadable application. For instance, you might want to verify that the application your loading has a newer version than is already on the device. The kit bootloader does not use the information and so we can safely leave these fields unmodified. But if in the future you would like to make your own bootloader that verifies this information, you should feel free to do so. You can also choose a specific load address for your application, but we're going to ignore that too, because PSoC Creator will auto-magically calculate those values for you.

On the bootloadable component, dependencies tab, there are two fields that are used to identify the bootloader. This information is needed to determine the load address of your application and to ensure the system settings will match between your bootloader and the bootloadable program. These fields can be quickly populated by clicking the little dot, dot, dot button and finding the HEX file and the ELF file for your bootloader. You can find these files in the example project installed along with the kit. They will be in *program files* for a 32-bit installation, or *program files x86* for a 64-bit Windows installation. Under that folder the path is *Cypress*; the kit name folder, in this case *CY8CKIT-049-42xx* or *41xx*; the version folder, *1.0*; the *Firmware* folder, the workspace folder, *SCB_Bootloader*; the project folder, *UART_Bootloader.cydsn*; the architecture folder, *CortexM0*; the compiler folder, *ARM_GCC* and the build configuration folder, *Debug*. When you select the HEX file, it automatically populates the second field for the ELF. The ELF contains symbol data required to correctly assemble your project.

Now you can add your custom design; a blinking LED, Capsense, or whatever else you can dream up, to your bootloadable application. Then you click "build" that will create the CYACD file and then use the bootloader host to bootload your new application. That's all you need to do in order to create a brand new bootloadable application.

If you have questions about PSoC Creator or PSoC in general, you're welcome to email me alan_hawse@cypress.com and I will make sure your questions are answered.