Comipling Code on the Stellaris LaunchPad

Abstract
The purpose of this application note is to provide help in compiling C+ code onto the Stellaris LaunchPad using Code Composer Studio. This code can be used to control key components and features of the Stellaris. In this document the user will get a brief background on Code Composer Studio (CCS) as well as the Stellaris LaunchPad. There is also a step by step process on how to use CCS to start a new project or import an existing project.
**Background**

**Code Composer Studio**

The software used to program the Stellaris Launchpad is Code Composer Studio. This software is an integrated development environment that was created and developed by Texas Instruments for their embedded processor families. The program has applications for development, debugging, compiling, and simulation. This will give the user access to Texas Instruments’ entire device families, essentially any device that has a processor implanted on the printed circuit board will be pre-downloaded on the software. This software is a great tool. It will allow the user to compile, debug, develop, and simulate all Texas Instruments’ devices currently being implemented. Knowing exactly how the printed circuit boards will react before actually programming the device will save the user a great deal of time. One of the biggest advantages this software has is the ability to run situations based on downloaded content. As mentioned the software has access to Texas Instruments’ device families, so the specific device and functionality are available once Code Composer Studio is running. There will be no need for the actually device to be programmed until the functionality of the program code is correct. This will be useful because if there is a crucial error in the user code it can be caught in simulation and not allowed to potentially harm the Stellaris.

**Stellaris LaunchPad**

The dimensions of the Stellaris LaunchPad are 2.0” x 2.25” x 0.425”. This microcontroller was made and provided by Texas Instruments. The Stellaris Launchpad is a 32-KB ARM microcontroller with 256-KB flash memory, 32-KB SRAM, 80-MHz operation, and USB device. The Launchpad also has an on board in-circuit debug interface. This device can do a range of tasks that range from lighting up LED’s in a sequence to sending data to another device via a USB port.

**Installation**

Before proceeding the user should ensure that Code Composer Studio and Stellaris software is properly installed to the computer. Code Composer Studio can be purchased from Texas Instruments at [http://www.ti.com/tool/ccstudio#buy](http://www.ti.com/tool/ccstudio#buy). A critical folder called StellarisWare comes with the program. This folder contains sample projects that can be loaded on the board for different applications. Follow the step by step instructions for installation.

Also the Stellaris LaunchPad software can be downloaded from the Texas Instruments site at [http://www.ti.com/tool/EK-LM4F120XL](http://www.ti.com/tool/EK-LM4F120XL). Some of this software includes but not limited to the Stellaris ICDI Drivers folder and the LaunchPad Evaluation Board software. Download the folders and follow the step by step instructions for installation.
Making the USB Connection

Connection must be made from the user computer to the Stellaris LaunchPad. This is done by a USB to Micro USB cord that is provided with the Stellaris. The Stellaris LaunchPad is equipped with two micro USB ports. The port on the upper left side, connector device, can be used to power the board as well as send data to and from the Stellaris. While the port on the top left, power/ICDI, is used to power the board and compile code onto the board. They are activated by a power select switch on the top left of the board. Refer to figure 1 for help. Knowing the difference between these micro USB ports are crucial since C+ code can only be compiled on the power/ICDI port.

Making USB connection:

- Connect USB cable to user computer
- Connect USB cable to micro USB (power/ICDI) port.

![Figure 1: USB connection from computer to Stellaris LaunchPad](image)
After connection is made the Stellaris should be recognized by the computer. You can check this connection by checking your computer’s device manager.

Checking Device Manager (on computer):

- click start menu
- my computer
- right click on the my computer page
- click on properties
- click on Device Manager

![Device Manager](image)

Figure 2: Device Manager

Look for the Stellaris Virtual Serial Port and the Stellaris ICDI JTAG/SWD Interface, as seen in figure 2. If these ports cannot be found, then either your drivers software has not been installed or needs to be updated. Again you can find the Stellaris software at [http://www.ti.com/tool/EK-LM4F120XL](http://www.ti.com/tool/EK-LM4F120XL)
Opening a New Project

Now that the connection is made from the Stellaris LaunchPad to the user computer Code Composer Studio can now be opened.

Opening Code Composer Studio:

- click start menu
- all programs
- ccs_v5

After opening Code Composer Studio the user will be prompted to select a workspace. This is the default folder that the user’s projects will be saved automatically. Select a location for the folder and be sure to remember where this is. Opening Code Composer Studio for the first time will take you to the welcome page where you can find useful links for beginners with the program. From here the user needs to start a new project for the Stellaris LaunchPad.

New Project:

- click file
- New
- CCS Project

![Figure 3: Opening a new project](image)
The CCS Project window will appear, as shown in figure 4. Here the user needs to enter information about the TI device the user is using to ensure code will be compiled correctly. In the window enter the following:

- Project name: first project
- Family: Arm
- Variant: 120
  - Stellaris LM4F120H5QR
- Connection: Stellaris In-Circuit Debug Interface
- From the projects list select:
  - Empty Project (with main.c)
- Click finish

Figure 4: CCS Project Window

Your project should open and begin adding code or importing code.
Importing a Existing Project

For comiling C+ code on the Stellaris LaunchPad there are two ways of accomplishing this. First the user can write your own C+ code to run the specific functions that you want. Or for new users can import existing projects to use or to edit for addition functions.

Importing Project:

- Click Projects
- Import Exisiting CCS Eclipse Project

The Select Existing CCS Eclipse Project window should appear, as seen in figure 5. The user can then browse through their personal folders to find CCS projects. For new users you will not have projects to import, but Code Composer Studio comes with pre-loaded example projects. This was downloaded with the Stellaris software and is called StellarisWare. This folder must be located in order to continue, location of this folder will vary based on where the user saves the file but is general found in the TI folder of your C: Drive. Project0 will be the project being imported.

Figure 5: Import Exisiting CCS Eclipse Project Window
StellarisWare Folder:

• Click StellarisWare
• Boards
• Ek-Im4f120xl
• Project0
• Click ok

Project0 should appear in the project explorer tap, as seen in figure 6. The next step will be to build the project.

Build Project:

• Project tab
• Built All

Debug Mode

Now that the project is built, the C+ code is ready to be compiled onto the Stellaris LaunchPad.

Debug Mode

• Run tab
• Debug

Figure 6: Debugging Project0
After this process is finished the C+ code is not complied on the Stellaris LaunchPad and is ready to run.

Run Program:

- Run tab
- Resume

Figure 7: Running Project0

Congratulations!!! Project0 had just been successfully downloaded on the Stellaris LuanchPad and the user should now see the LED’s flashing.

Figure 8: LED flashing