

## APPENDIX A. CODE COMPOSER STUDIO (CCS) v5.2.1: A BRIEF TUTORIAL FOR THE LCDK

### A.1 Introduction

Code Composer Studio™ (CCS) is Texas Instruments' integrated development environment (IDE) for developing routines on a wide variety of their DSPs. In CCS, the editing, code generation, and debugging tools are all integrated into one unified environment. You can select the target DSP, adjust the optimization parameters, and set the user preferences as you desire.

An application is developed based on the concept of a project, where the information in the project determines what source code is used and how it will be processed. Learning to use Code Composer Studio is a necessary step in bridging the gap between DSP theory and real-time DSP. We recommend that you devote some time to getting to know CCS. You can practice on your own or you can use this appendix to help get you started.

### A.2 Starting Code Composer Studio

This tutorial assumes that CCS is properly installed on a Windows XP, Windows Vista, or Windows 7-based computer. As of this printing, the latest version of CCS (version 5.2.1) supports all of these operating systems, and is available for the TI OMAP-L138 Low Cost Development Kit (LCDK).

As long as you are running CCS version 5.2.1, this tutorial will sequence you through the basic steps involved in creating, compiling, loading, and running (executing) a project. If CCS is not installed, please install it now. The following discussion assumes you are using an OMAP-L138 LCDK with an XDS100 emulator module, and the LCDK board has been reflashed to run the winDSK8 kernel on boot-up. If your LCDK board has not reflashed, refer to the instructions posted at [rt-dsp.com](http://rt-dsp.com) to perform this task.

When you are ready...

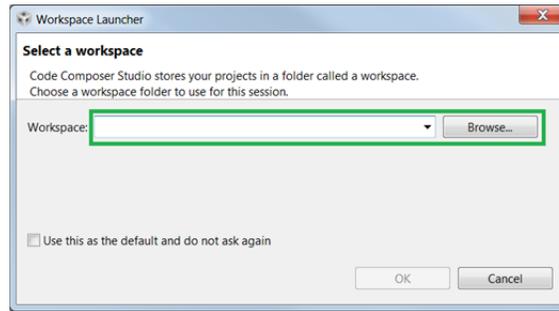
1. Verify that the LCDK board DIP switch SW1 is set with switch 1 OFF and switches 2-4 ON. If you look closely, the switch positions are labeled.
2. Connect the XDS100 emulator board into the 14-pin header on the LCDK board. Connect a USB cable to the XDS100 emulator, but do not plug it into your PC yet.
3. Power up the OMAP-L138 LCDK board by plugging its power supply into an energized AC outlet, and then connecting the power supply cable to the development board. The four user LEDs on the development board should flash in a sequential pattern then turn off.
4. Start up and allow the computer that you installed CCS on to fully finish its boot-up process. Plug the USB cable from the XDS100 emulator into your PC. Be sure that you allow the computer to finish loading the device drivers before you move to the next step.
5. Launch CCS by double clicking on its icon. After a standard CCS install, this icon will be located on your computer's "desktop". You are looking for the icon shown below.



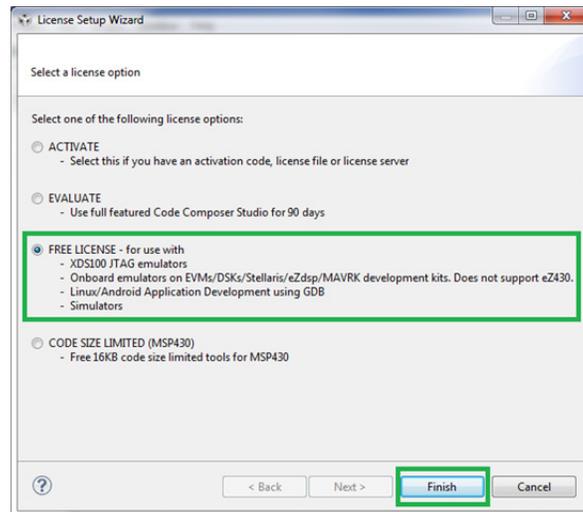
As CCS starts up, a splash screen similar to one shown below will appear.



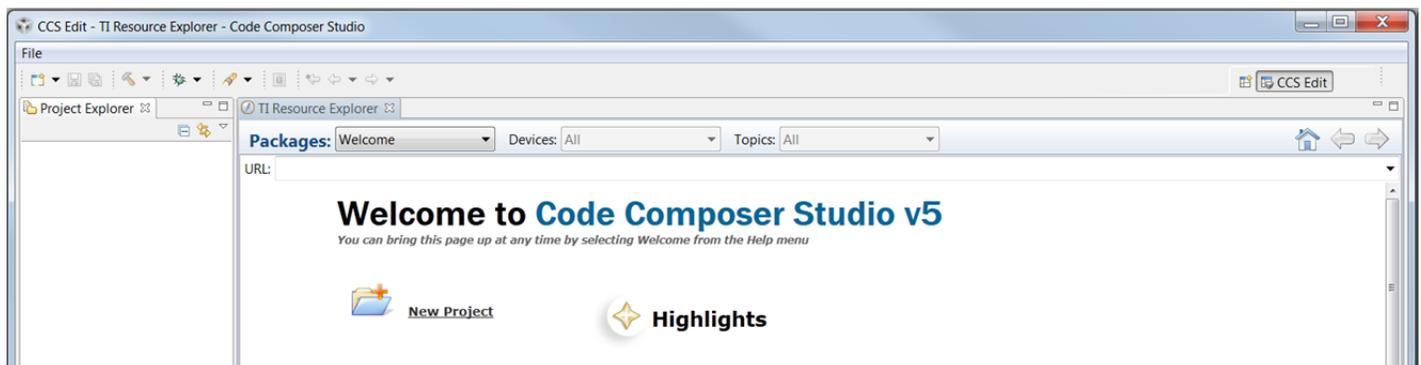
You will then be prompted to select a workspace directory. This directory is where CCS will store your projects. A suggested location would be a directory that is in the same location as the textbook CD-ROM contents were placed (C:\CD2ndEdition is the default directory for the textbook support files). You may store your files wherever you desire and please realize that some of this document's screen captures use slightly different directories.



If prompted about a software license, select the “FREE LICENSE...” option for the time being. This license supports the XDS100 JTAG emulation. Click **Finish**.



The CCS integrated development environment (IDE) will then finish loading and a welcome page similar to the one shown below will appear as shown below. If you are returning to CCS your display will be different.

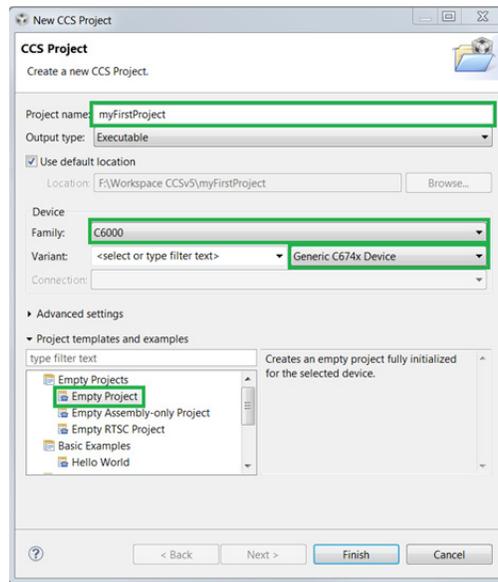


### A.3 Needed Files

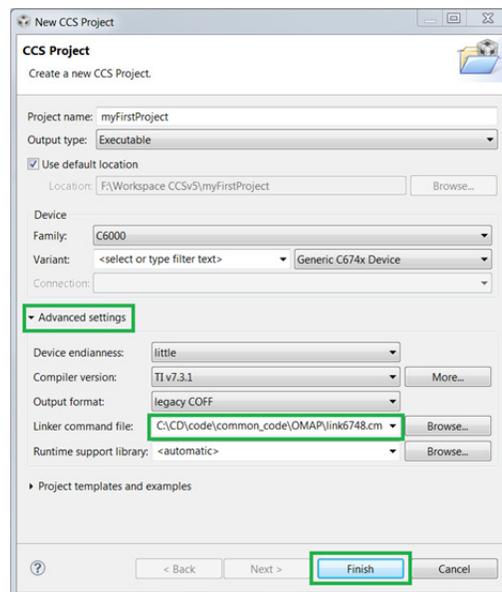
1. The following sections of the CCS tutorial assume that you have installed the software that was included on the provided CD-ROM.
2. Use a file management program such as Windows Explorer to verify that the directory *common\_code* exists at **C:\CD2ndEdition\code\common\_code**, where C:\CD2ndEdition may be some other name or disk location that you specified when you installed the provided software. If this directory structure does not exist, you either need to create it or install the provided software. There should be several files in the common\_code directory. If these files are not already in the directory, they can be copied from the common\_code directory of the provided CD-ROM.
3. Some form of organization is required for your projects and the dozens of files you will eventually create. CCS uses the concept of a workspace to organize groups of projects. A single workspace can be used to manage any number of projects.
4. If you copied the files from a CD-ROM, the files most likely have a “read-only” attribute. Removing this attribute may prove helpful later in this process.

## A.4 Creating a New Project

1. We will now create a project and its associated directory by adding a new project to the workspace.
2. To create a new CCS project, click on the “File” pull down menu, then select “New” and “CCS Project”. A dialog box similar to the one shown below will open. By default, the project will be created in the workspace location that was selected when you started CCS.
  - a. In the “Project Name:” box, type in **myFirstProject**.
  - b. In the “Family” box, select **C6000**.
  - c. In the “Variant” box, the type of processor target is selected. For the OMAPL138, select **Generic C674x Device**.
  - d. Under “Project templates and examples”, select **Empty Project**. You will add files to the project at a later step.

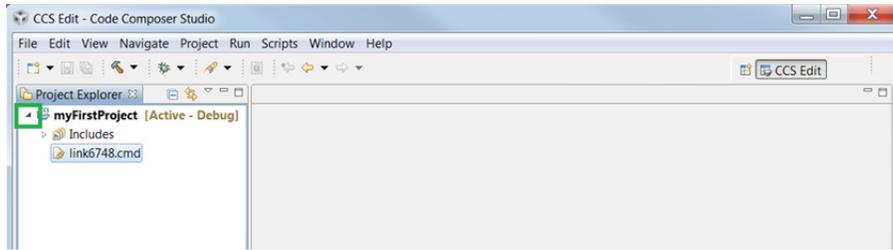


3. Click “Advanced settings”.
  - a. At the “Linker command file” box, browse to the **C:\CD2ndEdition\code\common\_code\LCDK** directory and select the **link6748.cmd** file. Leave the other options at their default values.



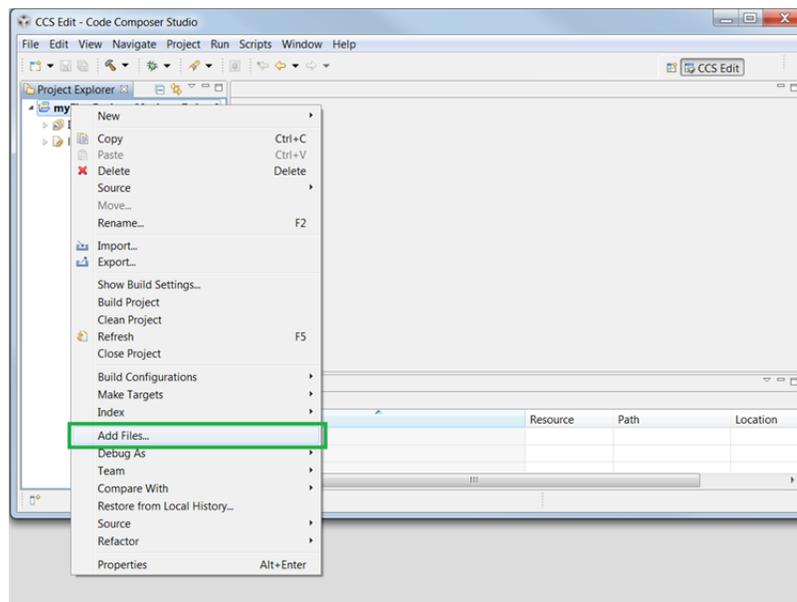
- b. The process you’ve just gone through will be the same for every CCS project you create for the OMAPL138. Click **Finish**.

4. CCS will now show your project in the Project Explorer window as shown below. Note that the linker command file you selected is displayed in the project tree. You may need to click on the small triangle symbol to expand the project view. If the project contains a *main.c* file, right click the file and select **Delete** to remove it from the project.

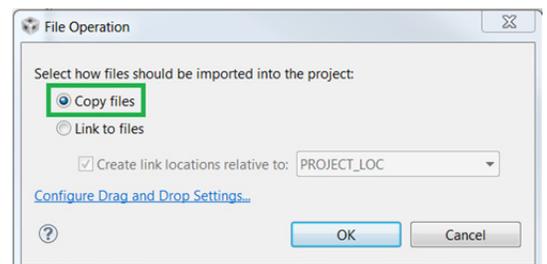
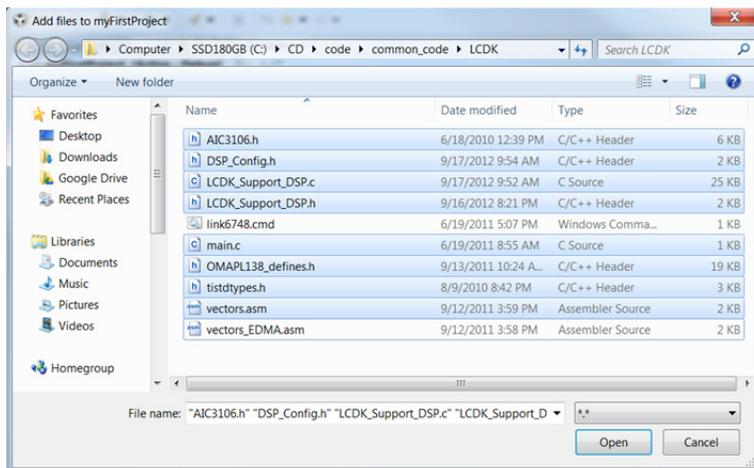


## A.5 Adding Files to a Project

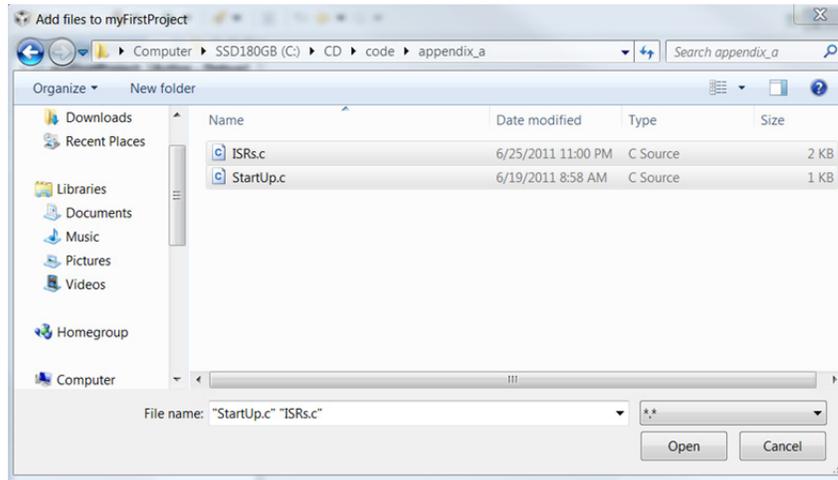
1. CCS will have created a project directory in the workspace and added files to it. The project information is stored in a number of XML files, such as *.ccsproject*. We strongly recommend that you not edit or otherwise modify these files. To add files to your project, right-click **myFirstProject** in the Project Explorer window and select “**Add files...**”.



2. “**Add Files...**”. Navigate to the `C:\CD2ndEdition\code\common_code\LCDK` directory and select all the files there *except* *link6748.cmd* and *vectors\_EDMA.asm*. Click **Open**. In the dialog that pops up, ensure that **Copy files** is selected, and click **OK**.

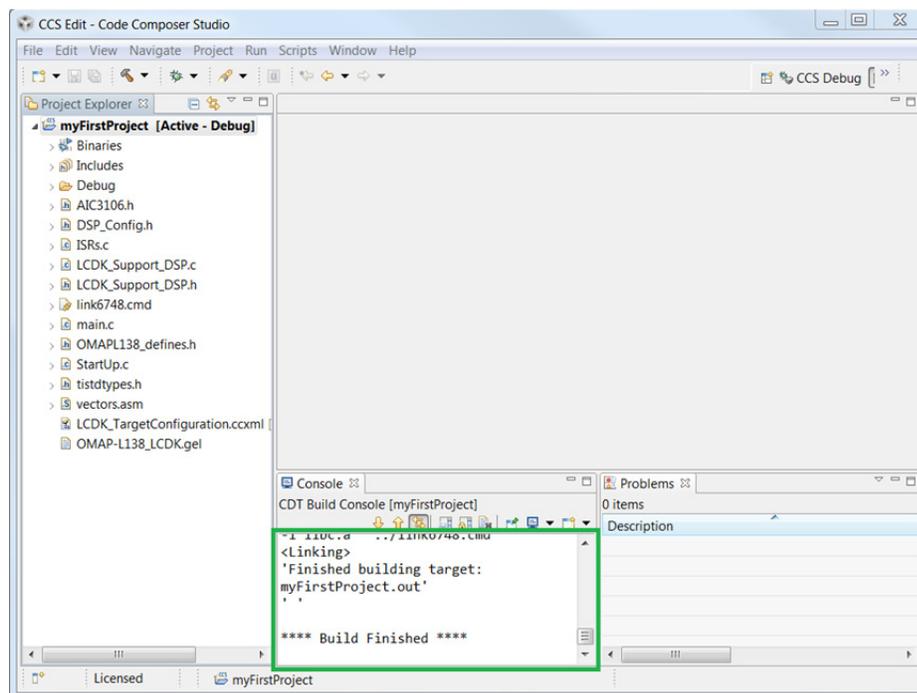


3. “Add Files...”. Navigate to the `C:\CD2ndEdition\code\appendix_a` directory and select both files there. Click **Open**. In the dialog that pops up, ensure that **Copy files** is selected, and click **OK**.



## A.6 Building the Project

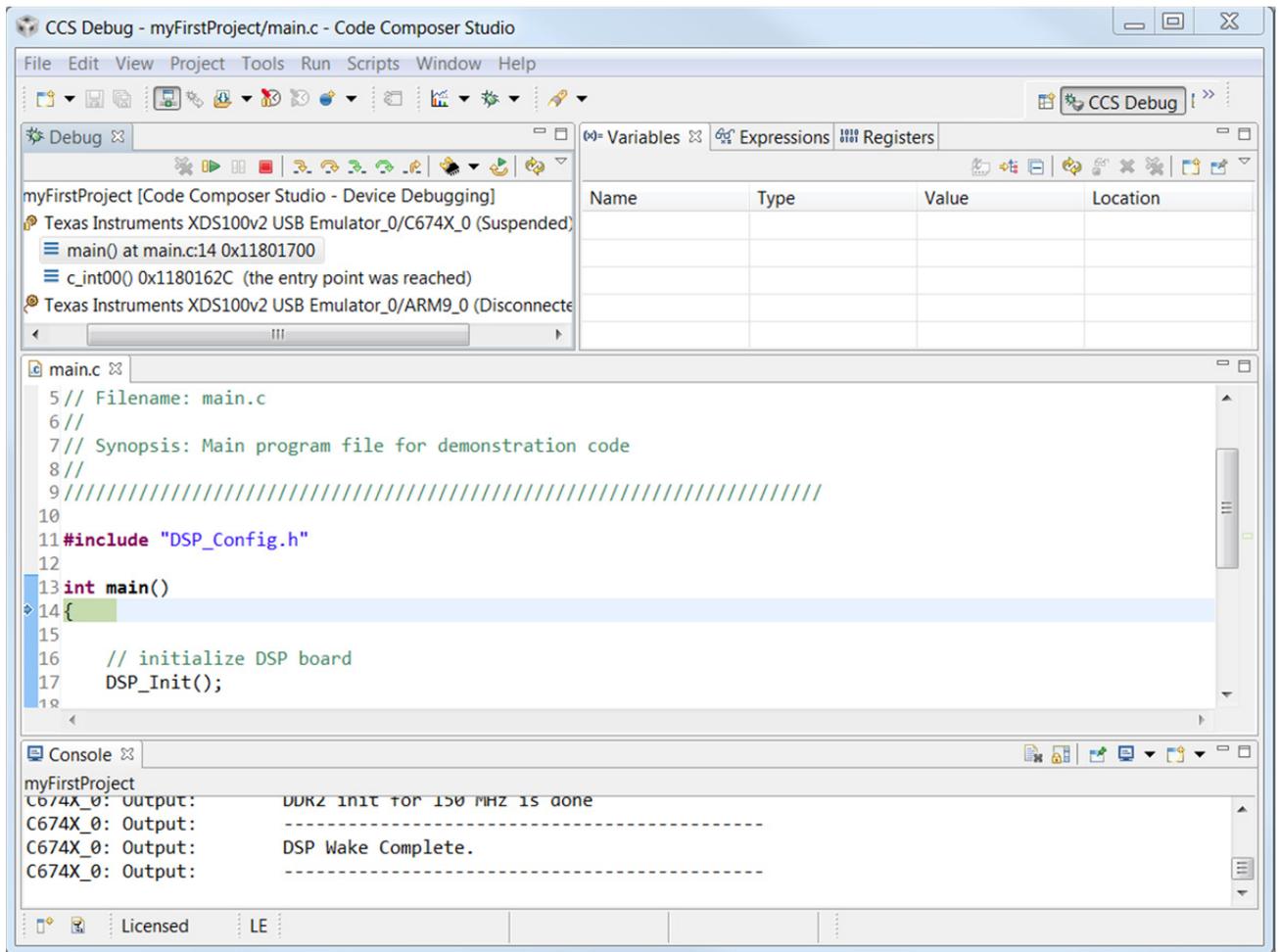
1. To build the project, select **Project** from the main menu and click **Build Project**. CCS will now compile and link the project files to produce an executable file. If all is well, the console window will show “**Build Finished**”. Note that CCS has many menu buttons that have the same function as the menu commands. Getting to know where the frequently used ones are can be very useful and speed up development.



## A.7 Loading the Program onto the Target Hardware

1. Now that the project is built, you need to load the project onto the target hardware, and then run it. The hardware interface is controlled by a target configuration file which we will add to the project. As before, right-click the project to **Add Files...** and navigate to the `C:\CD2ndEdition\code\target_configuration\LCDK` directory. Select the **LCDK\_TargetConfiguration.ccxml** and **OMAP-L138\_LCDK.gel** files, and click **Open**. In the dialog that pops up, ensure that **Copy files** is selected, and click **OK**.
2. In the Project Explorer window, double-click the **LCDK\_TargetConfiguration.ccxml** file to open it. Select the **Basic** tab and then select **LCDKOMAPL138**. Select the **Advanced** tab, and then click the **C674X\_0** entry. Under **Cpu Properties** (in the right column), note that the initialization script field should be set to **OMAP-L138\_LCDK.gel**. This is a script file that controls the initialization of the target hardware. Close and save the **LCDK\_TargetConfiguration.ccxml** file.

- In the Project Explorer window, click on your project. Then, from the CCS menu, select **Run** and click **Debug**. The target hardware is initialized; the configuration script in *OMAP-L138\_LCDK.gel* is run (as shown in the Console window), the program is loaded into the DSP's memory, and the DSP halted at the beginning of the main function.



### A.8 Running the Program on the Target Hardware

- Now that the project is loaded onto the target hardware, select **Run** from the main menu and click **Resume**. Assuming you have connected a line-level audio source and speakers/headphones to the board, you should hear audio being fed through the hardware. You can suspend and resume the program as desired. Note that the key items from the target menu are available as buttons in the debug window.
- To terminate the debug session, select **Run** from the main menu and click **Terminate**.
- You may modify and rebuild your code even while the hardware is running. You must rebuild and reload a program after any changes are made in the code if you wish to see the results of these changes!

### A.9 Getting to Know CCS

- As we mentioned at the start of this appendix, we highly recommend that you invest some time to become familiar with the CCS interface and its operation. It is the cornerstone of real-time DSP using the LCDK board.