

## APPENDIX A. CODE COMPOSER STUDIO (CCS): A BRIEF TUTORIAL FOR THE OMAP-L138

### 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.

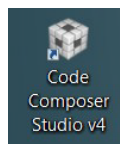
### A.2 Starting Code Composer Studio

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 4.2) supports all of these operating systems, and is available for the Zoom OMAP-L138 development kit.

As long as you are running the CCS version 4.2, 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 a Zoom OMAP-L138 development kit, and the OMAP-L138 board has been reflashed to run the winDSK8 kernel on boot-up. If your OMAP-L138 board has not reflashed, refer to the instructions on the CD to perform this task.

When you are ready...

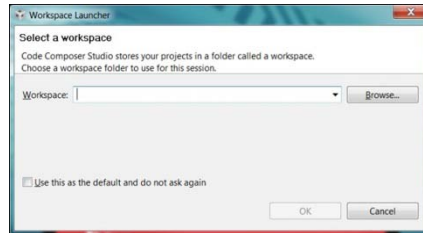
1. Verify that all development board DIP switches (two packages of eight switches, each) are in the "OFF" position. If you look closely, the switch positions are labeled.
2. Power up the Zoom OMAP-L138 development kit by plugging in its power supply to an energized AC outlet, connecting the power supply cable to the development board, and turning on the board using the large black slide switch. A single LED should now be energized on the development board.
3. Start up and allow the computer that you installed CCS on to fully finish its boot-up process. Plug the USB cable into the development board and then plug the other end of the cable into your PC.
4. 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 briefly.



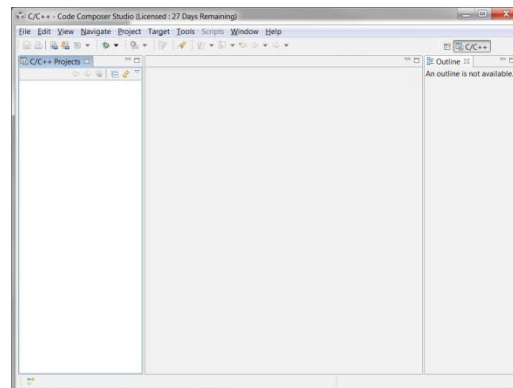
You will then be prompted to select a workspace directory. This directory is where CCS will store your projects. A suggested location would be a directory that is in the same location as the textbook CD-ROM contents were placed.



The CCS integrated development environment (IDE) will then finish loading and a welcome screen similar to the one shown below will appear. Click the CCS icon at the top right to start using CCS.



The CCS main screen will then appear as shown below.



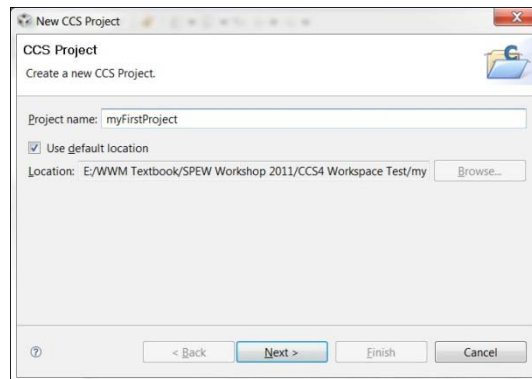
### A.3 Needed Files

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:\CD\common_code`, where `C:\CD` 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 4.2 uses the concept of a workspace to organize groups of projects. A single workspace can be used to manage any number of projects.

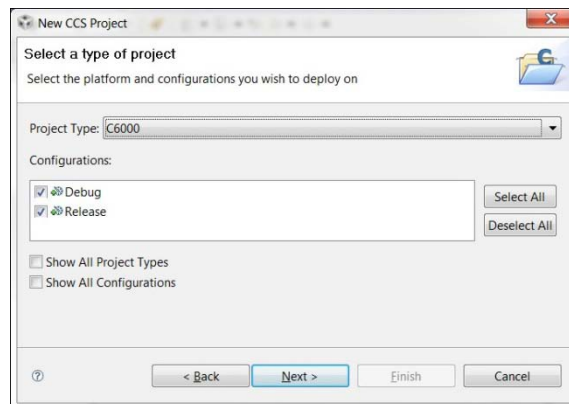
### A.4 Creating a New Project

1. We will now create a project and its associated directory by adding a new project the workspace.

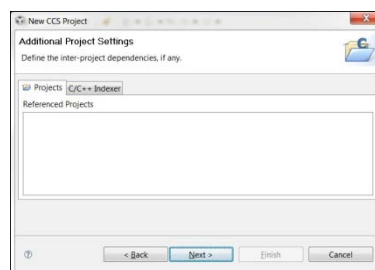
2. To create a new CCS project, click on the “File” pull down menu, then select “New” and “CCS Project”. A dialog box similar to the one shown below will open. By default the project will be created in the default workspace location selected when starting CCS. In the “Project Name:” box, type in **myFirstProject**. Click “Next”.



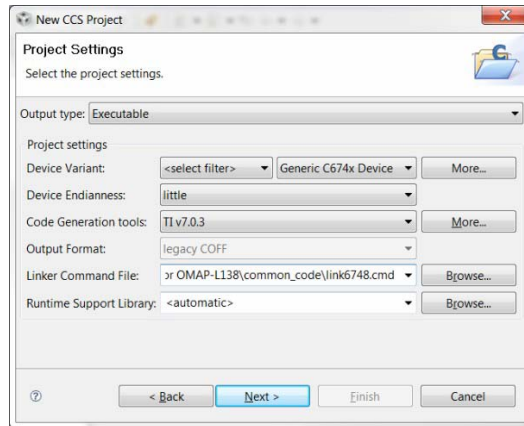
3. In the “Project Type” box, select “C6000”. Click “Next”.



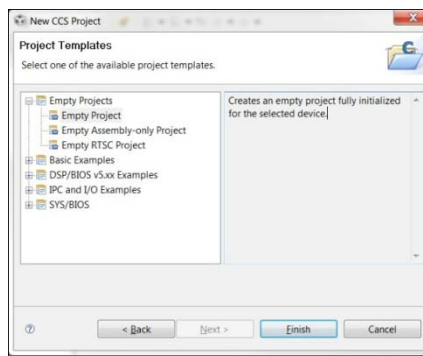
4. CCS supports the concept of inter-project dependencies for more complicated software development work. This is not necessary for our purposes. Click “Next”.



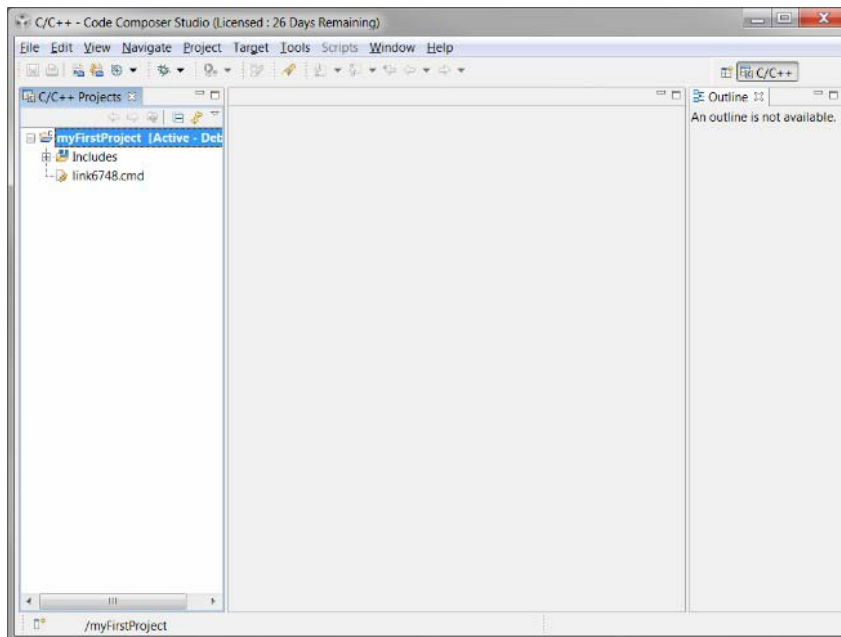
5. In the Project Settings dialog, the type of processor target is selected. For the OMAPL138, select “Generic C674X Device” for the “Device variant”. At the “Linker Command File” box, browse to the **C:\CD\code\common\_code\OMAP** directory and select the **link6748.cmd** file. Leave the other options at their default values. Click “Next”.



6. In the Project Templates dialog, select “Empty Project”. You will add files to the project at a later step. The process you’ve just gone through will be the same for every CCS project you create for the OMAPL138. Click “Finish”.

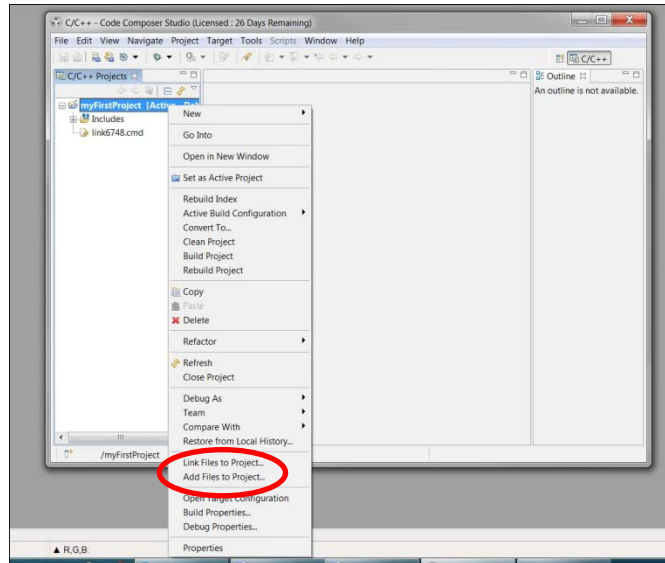


7. CCS will now show your project in the Project window as shown below. Note that the linker command file you selected is displayed in the project tree.

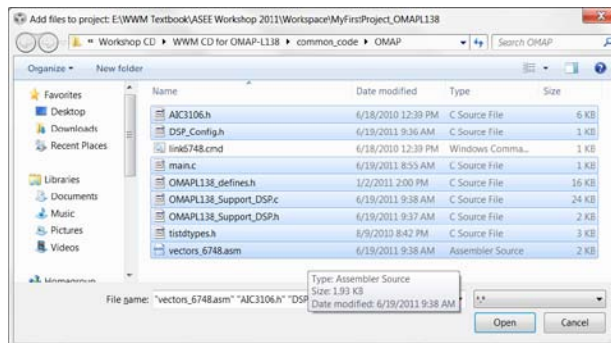


## A.5 Adding Files to a Project

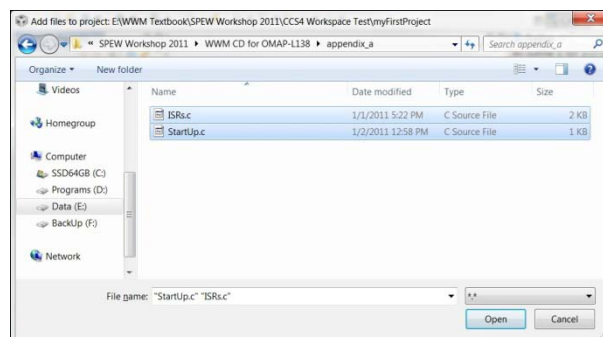
1. CCS will have created a project directory in the workspace and added files to it. The project information is stored in a number of XML files, such as **.ccsproject**. We strongly recommend that you not edit or otherwise modify these files. To add files to your project, right-click **myFirstProject**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



2. Navigate to the **C:\CD\code\common\_code\OMAP** directory and select all the files there except *link6748.cmd* and *vectors\_EDMA.asm*. Click “Open”.

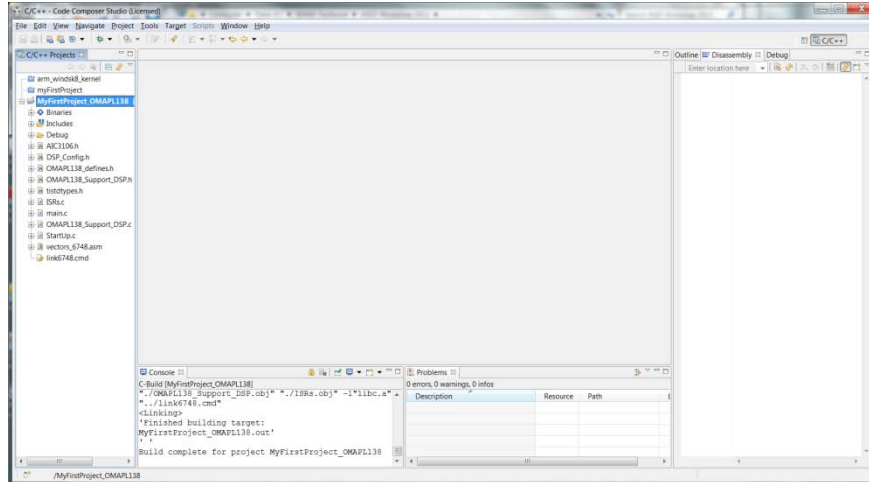


3. Navigate to the **C:\CD\code\appendix\_a** directory and select both files there. Click “Open”.



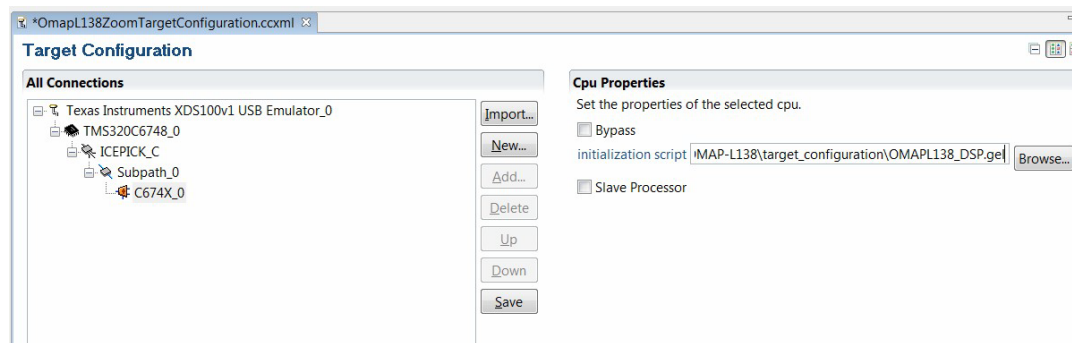
## A.6 Building the Project

1. To build the project, select the Project from the main menu and click “Build Active 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 complete for project...”. 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.

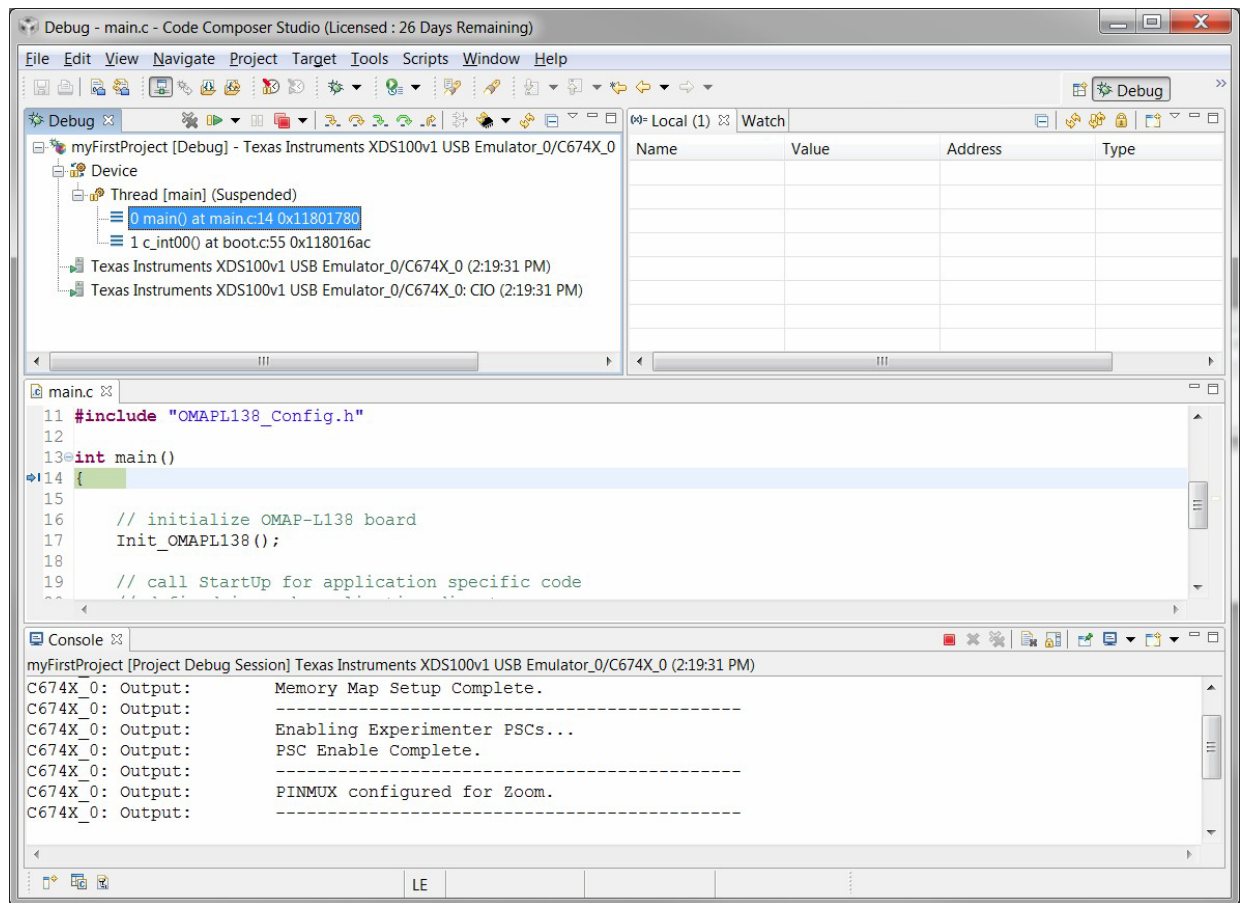


## A.7 Loading the Program onto the Target Hardware

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:\CD\code\target\_configuration\OMAP** directory. Select the **OmapL138ZoomTargetConfiguration.ccxml** file, and click “Open”.
2. In the CCS Projects window, double-click the **OmapL138ZoomTargetConfiguration.ccxml** file to open it. Select the **Advanced** tab, and then click the **C674X\_0** entry. Under **Cpu Properties**, click the “Browse” button and navigate to the **C:\CD\code\target\_configuration\OMAP** directory. Select the **OMAPL138\_DSP.gel** file, and click “Open”. This is a script file that controls the initialization of the target hardware. Save and close the **OmapL138ZoomTargetConfiguration.ccxml** file.



3. From the CCS menu, select **Target** and click **Debug Active Project**. The target hardware is initialized; the configuration script in **OMAPL138\_DSP.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

1. Now that the project is loaded onto the target hardware, select **Target** from the main menu and click **Run**. 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 halt and restart the program as desired. Note that the key items from the target menu are available as buttons in the debug window.
2. To terminate the debug session, select **Target** from the main menu and click **Terminate All**.
3. 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

1. 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 OMAP-L138 development board.