Author: Addison Elliott

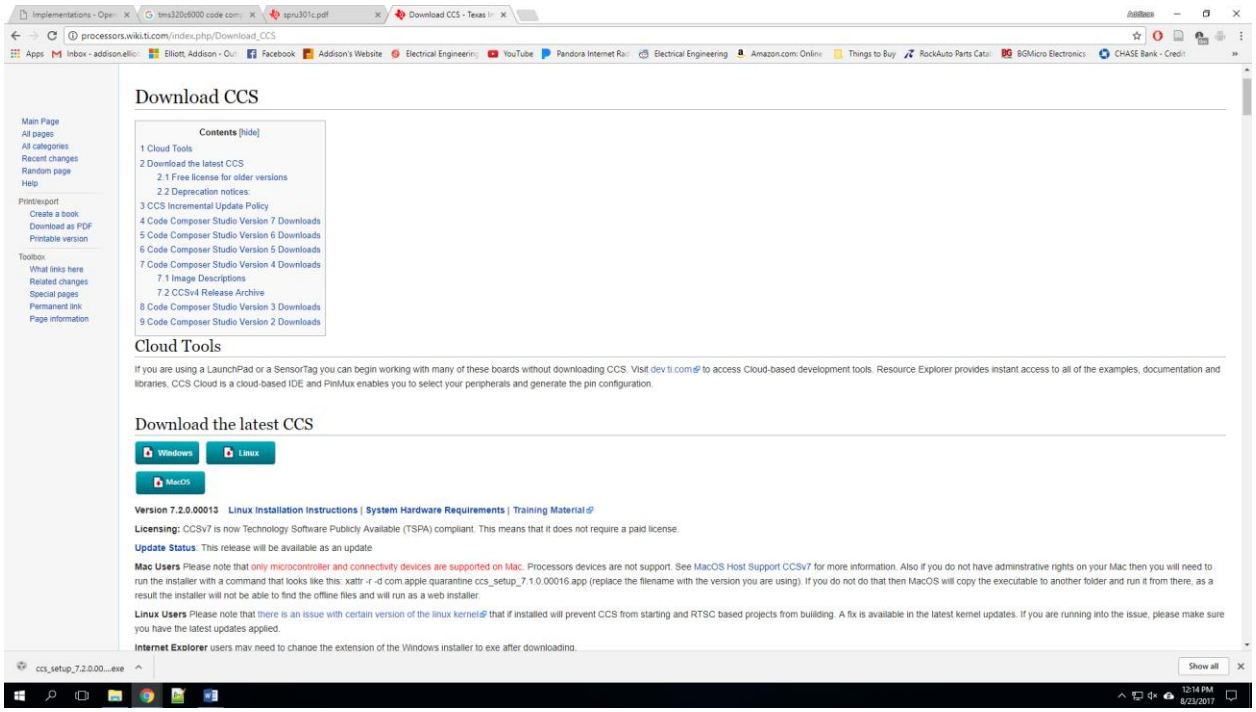
Date Created: 8/23/2017

Last Updated by: Addison Elliott

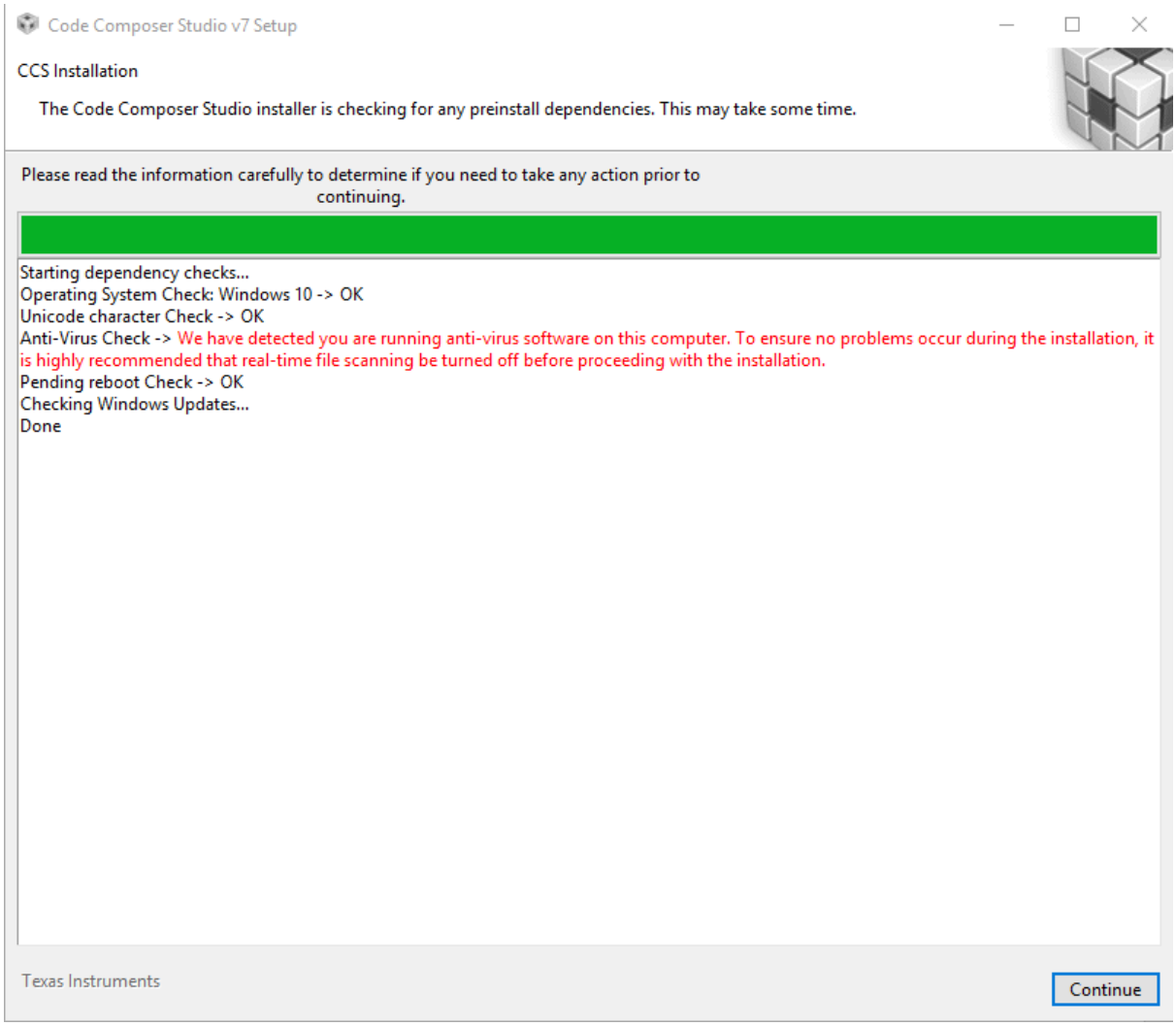
Last Updated: 11/01/2017

Setting up DSP Starter Kit (DSK)

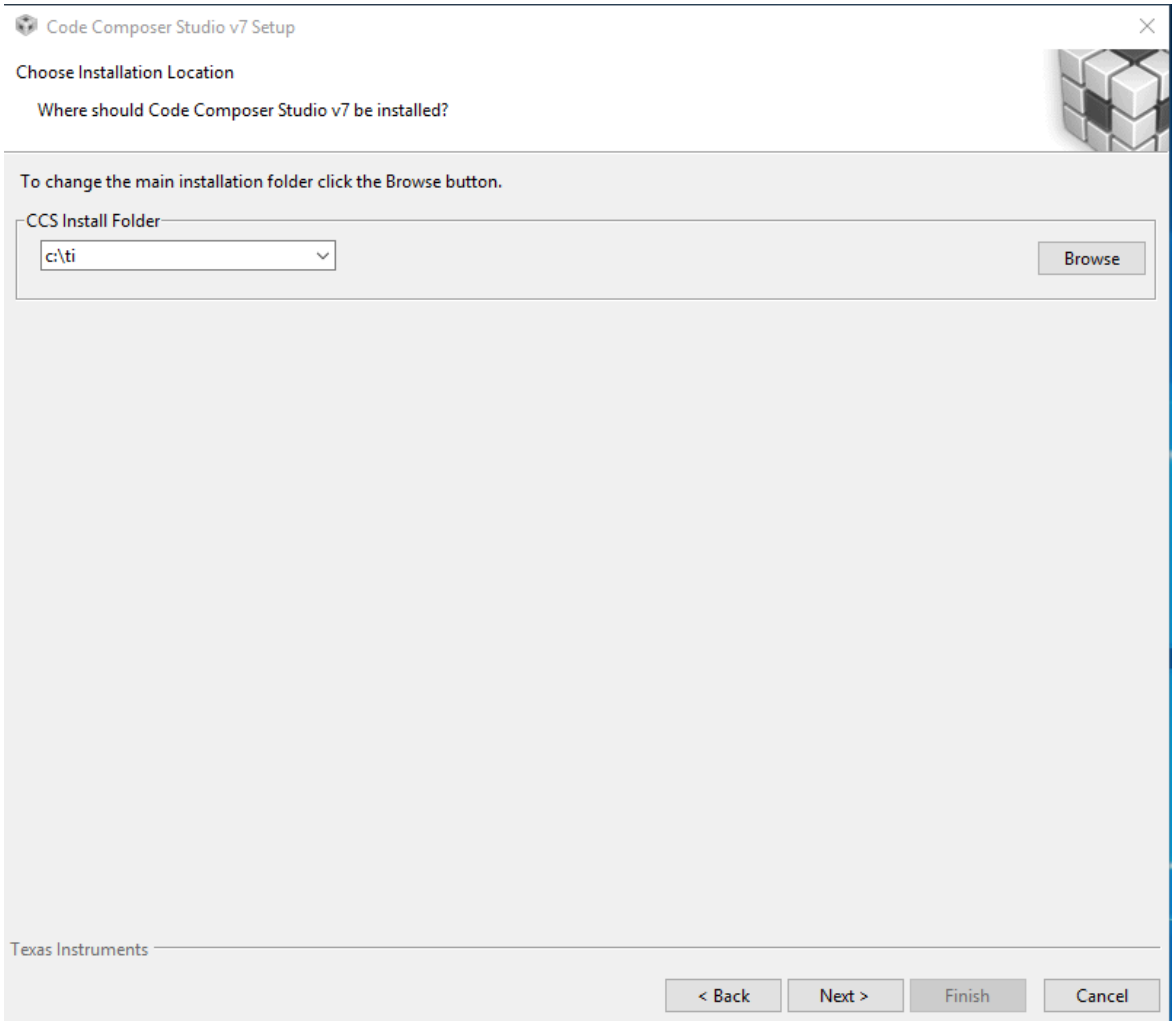
1. Begin by downloading the latest Code Composer Studio (CCS) software from the following link: http://processors.wiki.ti.com/index.php/Download_CCS
    a. CCS v3.1 is available on the CD with the product but this does **NOT** support Windows Vista and higher operating systems. You must use a later version of CCS for these OSes.



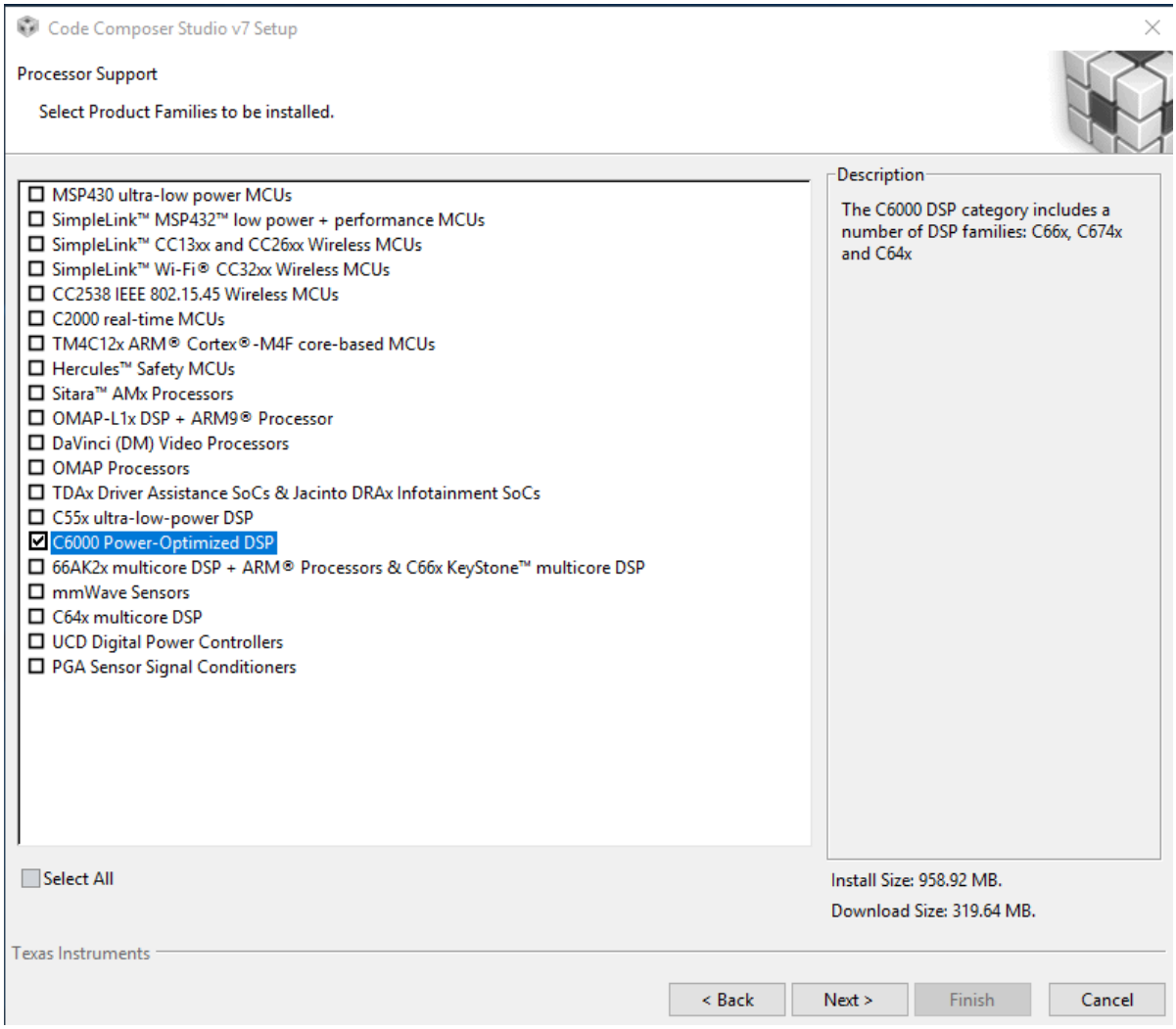
2. After the executable is finished downloading, double-click it to run. A window will open up and check the status of your system. If there is any errors or warnings in the textbox, it is recommended you resolve those issues before continuing



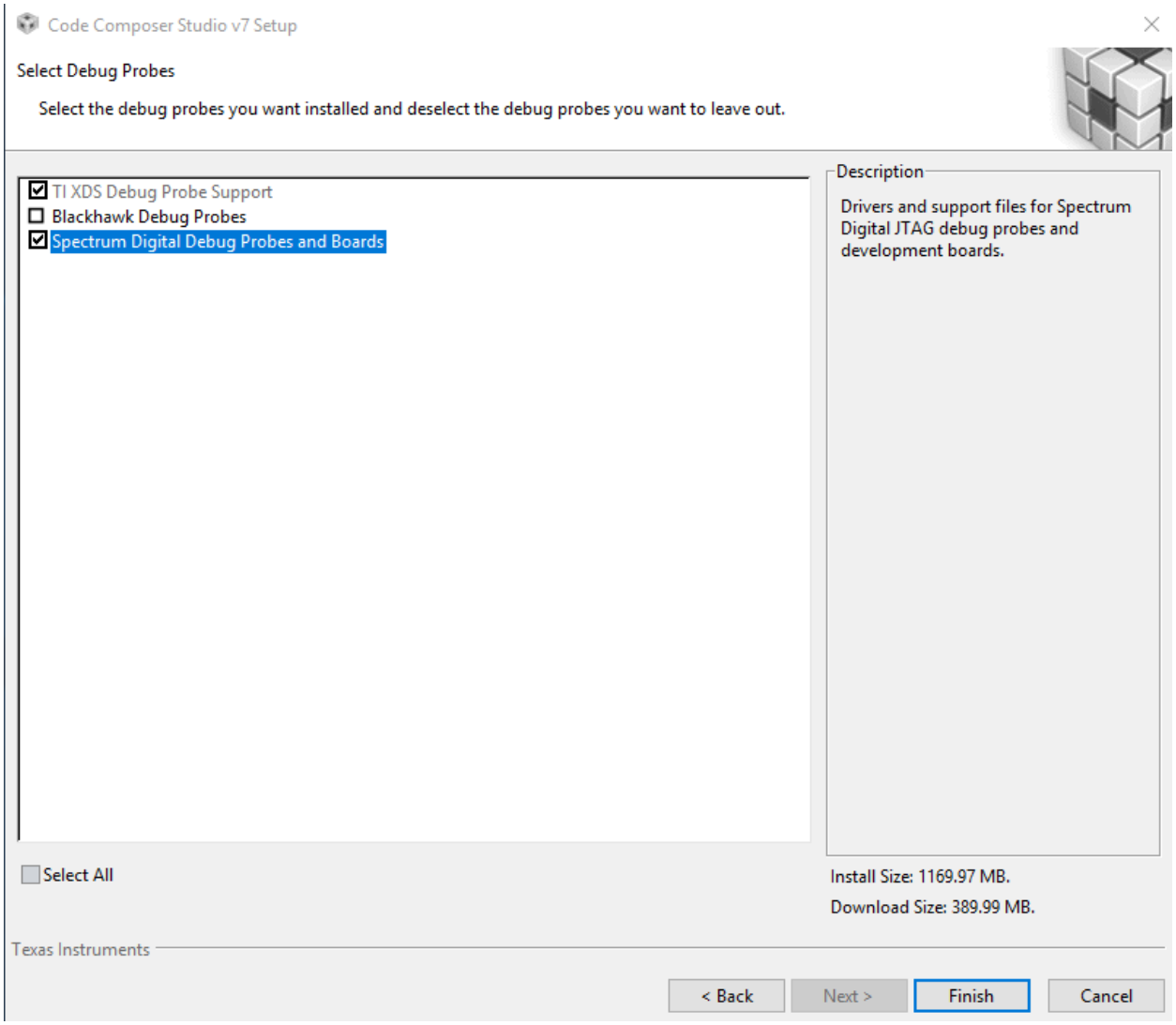
3. Accept the terms and conditions and continue
4. Select the desired installation folder for CCS and continue
    a. If unsure, leave the default folder and continue



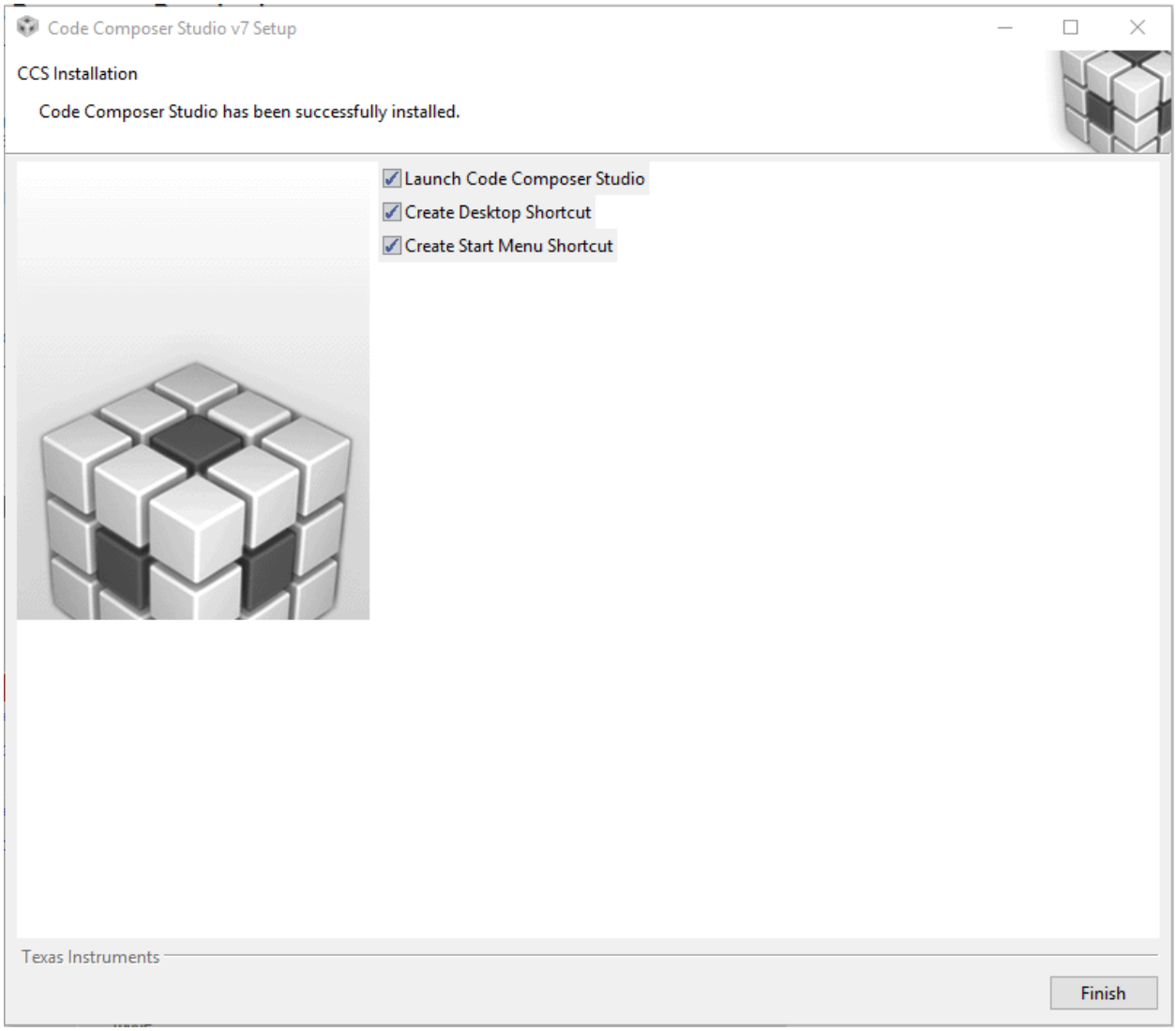
5. Select the Product Families to be installed. For this DSP Starter Kit, the processor is **C6000 Power-Optimized DSP**. Press “Next” to continue



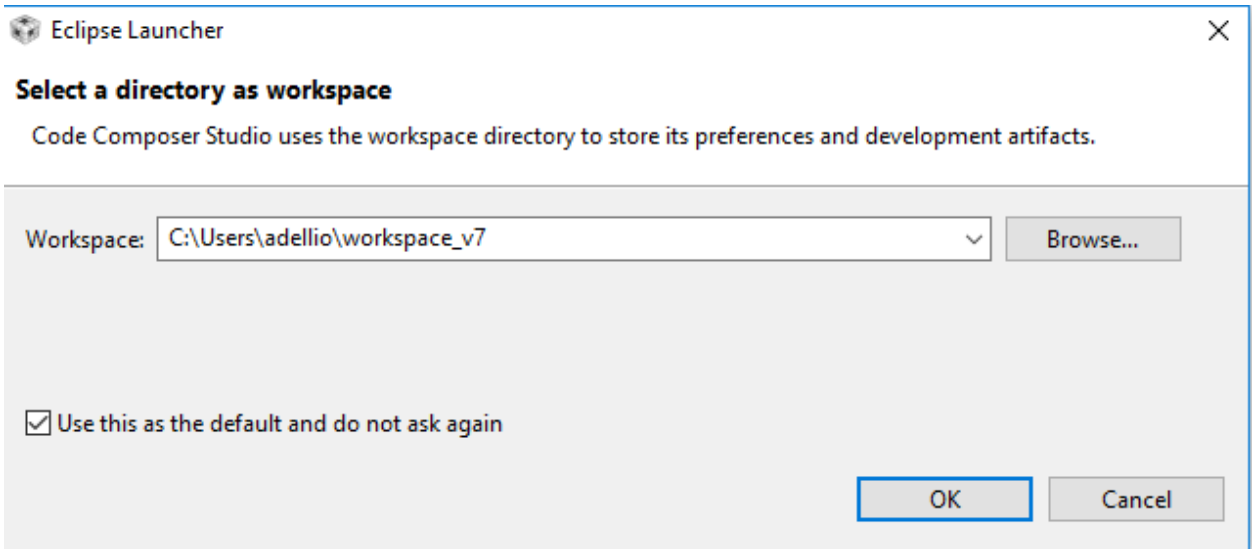
6. Select the **Spectrum Digital Debug Probes and Boards** option and click Finish
    a. This option installs the drivers required to program to the board



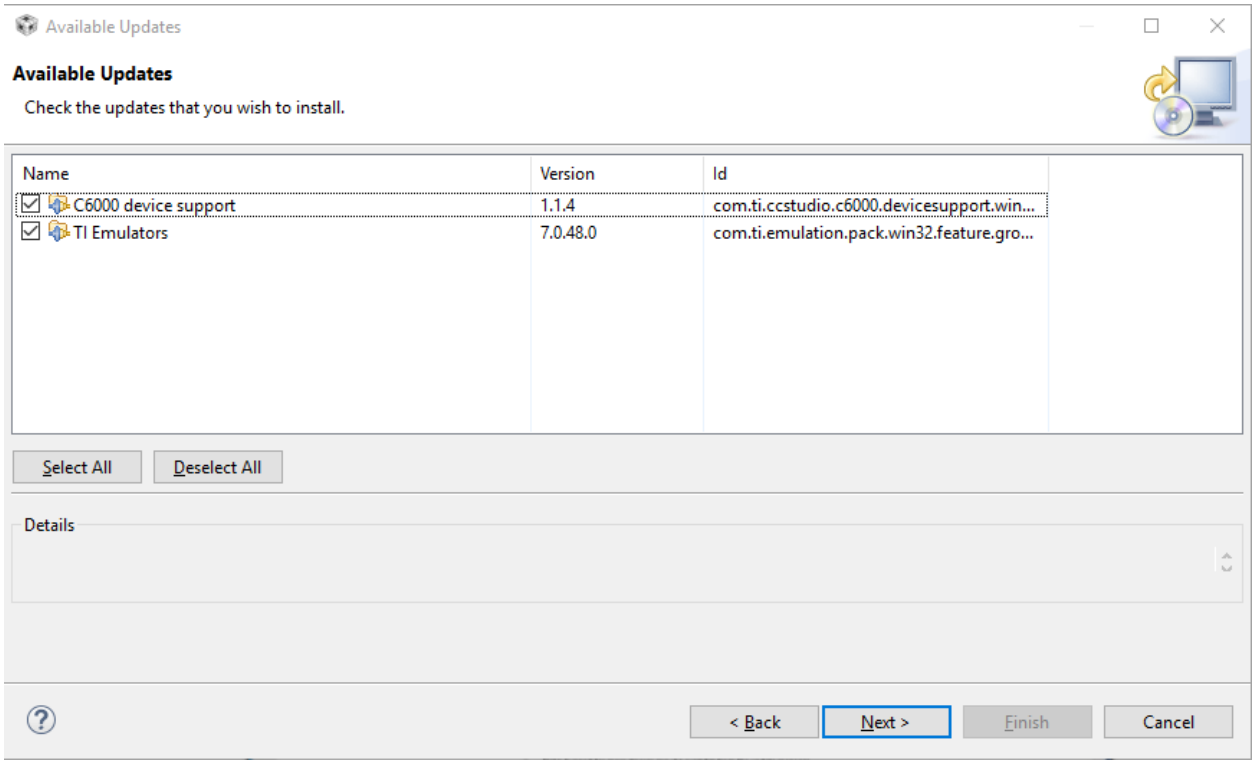
7. CCS will now be downloaded and installed. When finished, you will see a window like the following. Click Finish and CCS is successfully installed!



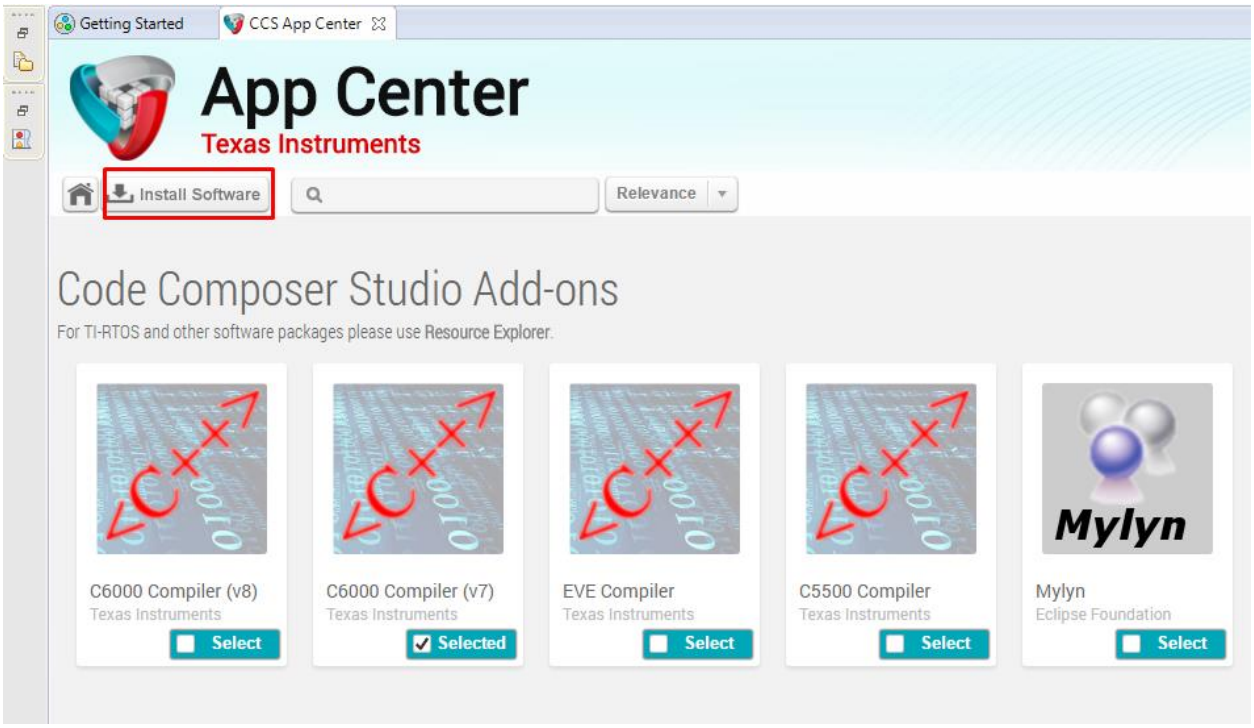
8. Start CCS. Upon first startup, a dialog will appear asking where your workspace directory should be located. The default location is fine but check **Use this as the default and do not ask again** and press OK.



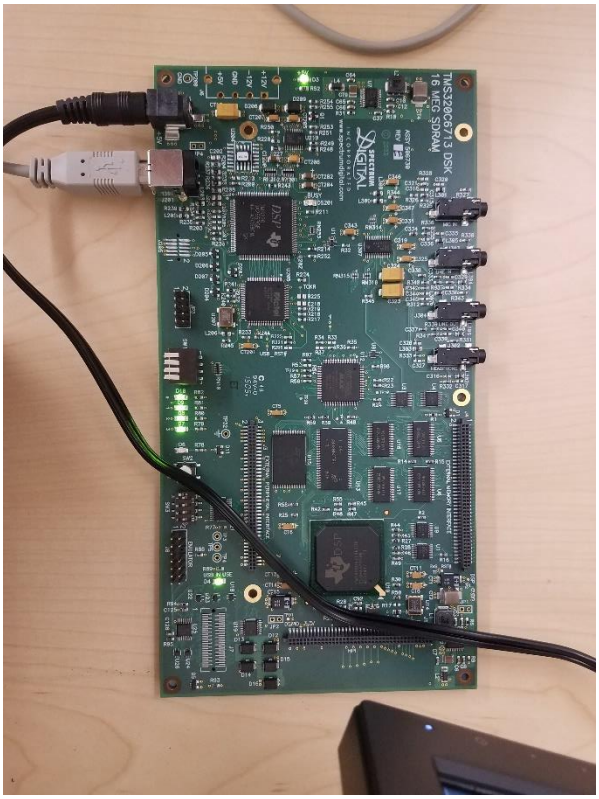
9. Next, we need to install any necessary updates. Go to **Help->Check for Updates**
10. Install any necessary updates. Typically, the C6000 device support will need to be updated



11. Next, we must install the C6000 compile. Select **View->CCS App Center**
12. In the section "Code Composer Studio Add-ons", select **C6000 Compiler (v7)** and then select **Install Software** at the top of the page
    a. The C6000 Compiler v8 and above does **NOT** support this board anymore since it is so old. You must use v7 of the compiler!



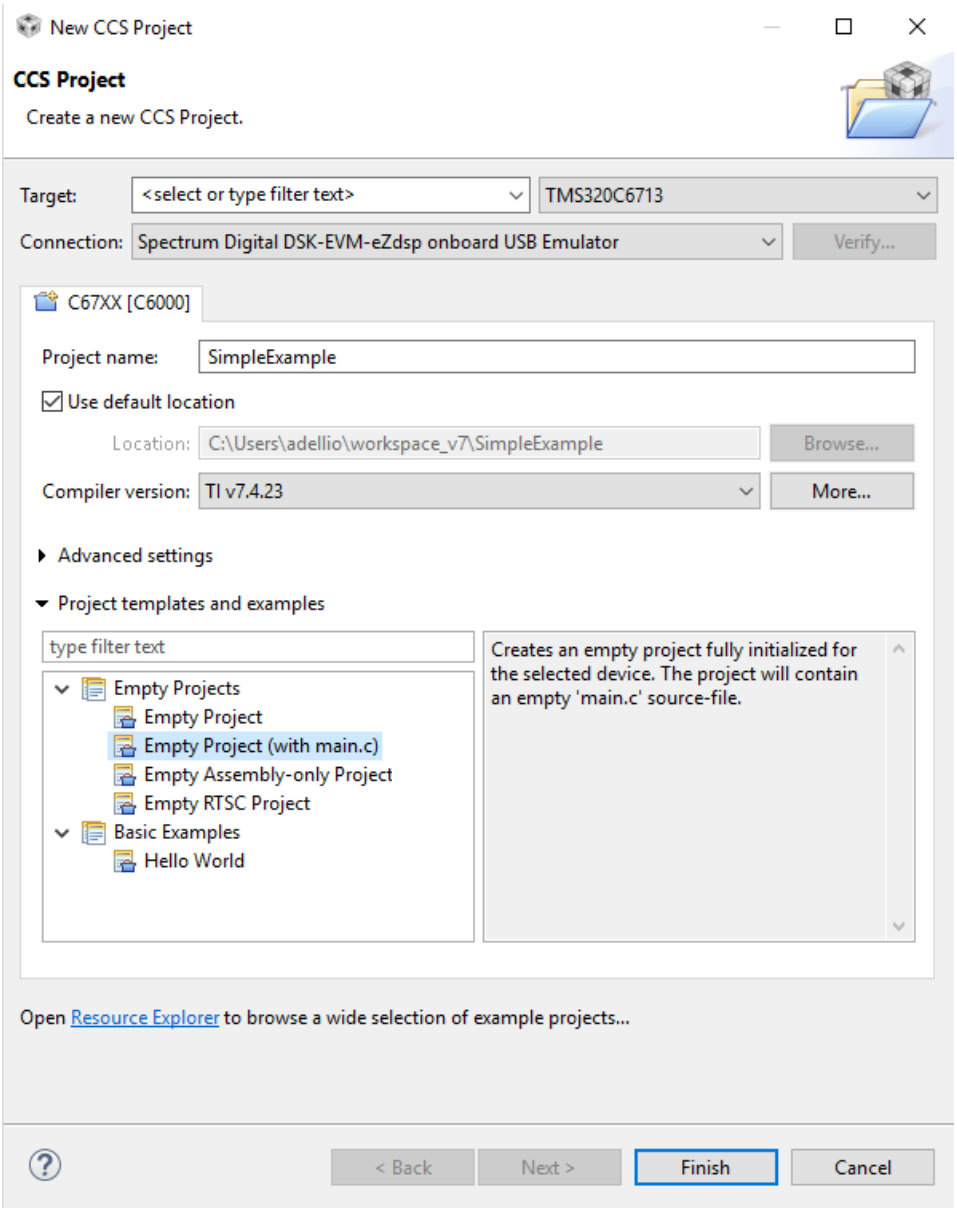
13. Accept the terms of agreement, and follow the prompt to finish the installation. Restart CCS to have the changes take effect
14. The software side of things is complete for installation. Now we move on to the hardware. Plug in the +5V power supply to the DSK board. When power is applied to the board the Power On Self Test (POST) will run. LEDs 0-3 will flash. When the POST is complete all LEDs blink on and off then stay on. At this point your DSK is functional.



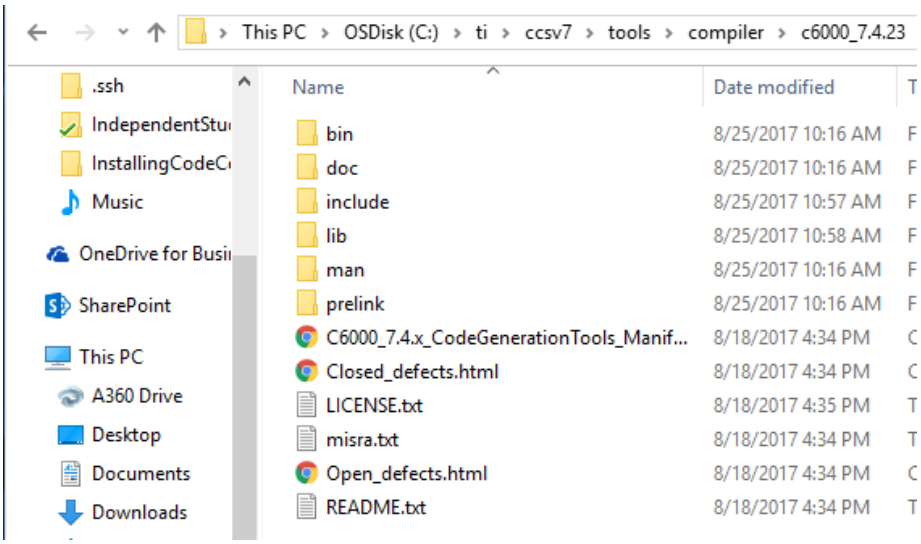
15. At this point, your DSK is functional and you can connect it to your PC with the included USB cord.
    a. You should hear the familiar “beep” noise when plugging in the USB. If not, this may be an issue.
16. You are finished setting up your DSK board. The next steps are to create a project and compile and run it on the board.

Creating Project in CCS

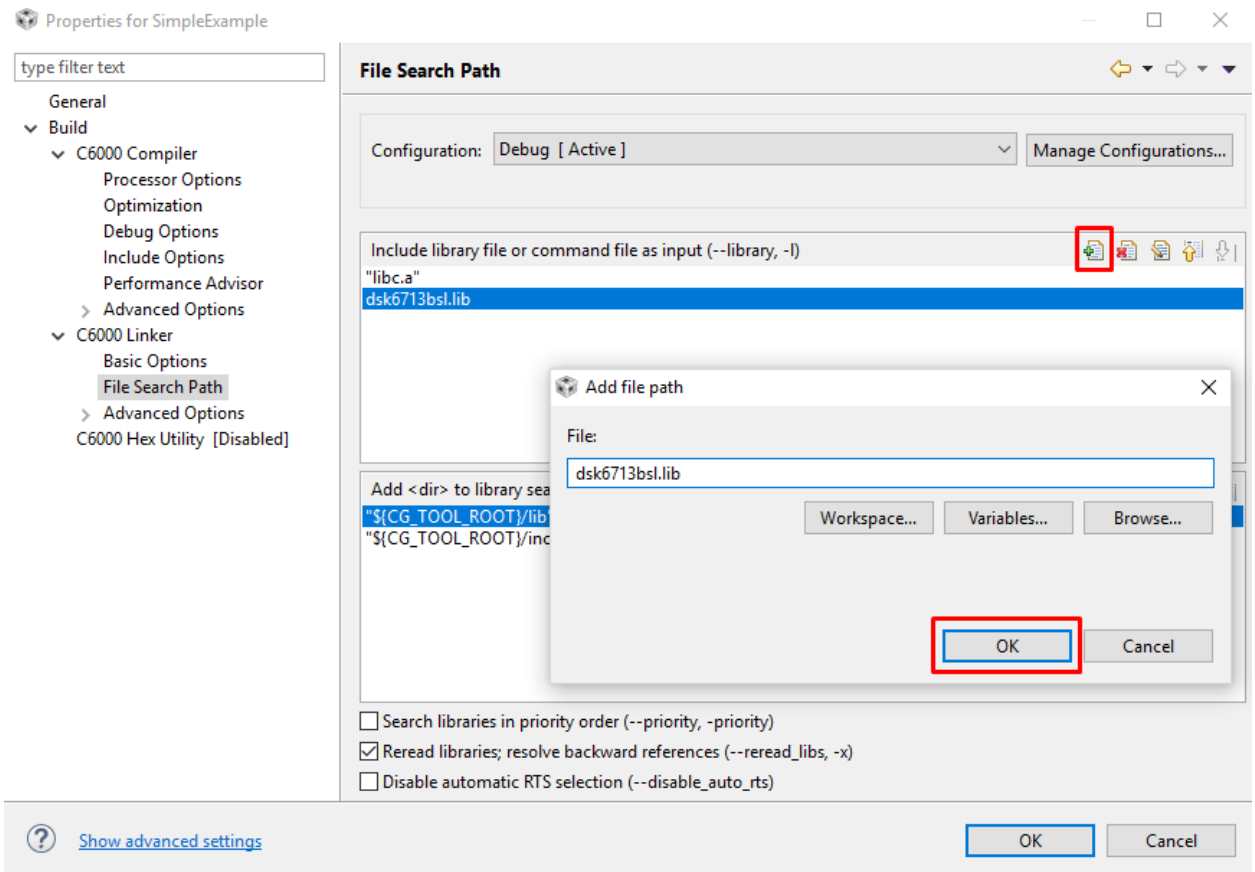
1. To begin creating a new project in CCS, start by selecting **File->New->CCS Project**
    a. You can also select **File->New->Project** and then select **CCS Project** but this is an extra step
2. A new project dialog box will appear. Fill in the following settings:
    a. **Target:** TMS320C6713
    b. **Connection:** Spectrum Digital DSK-EVM-eZdsp onboard USB Emulator
    c. **Project Name:** Your desired project name
    d. **Compiler Version:** TI V7.4.23 (version downloaded from CCS App Center)



3. Select “Finish” to create the project. A new window will appear where you will spend most of your time programming. Make sure you have the “Project Explorer” panel on the left hand side. If not, you can show it by selecting **View->Project Explorer**
4. This next step is a one-time thing. If this has already been completed, then you may move on to the next step. Extract the provided folders **include** and **lib** folders to **C:\ti\ccsv7\tools\compiler\c6000_7.4.23**. There should already be existing **include** and **lib** folders in this directory
    a. These files are used to communicate with the board. They are specific to the board with functions such as read from Audio Codec, read state of switch, or change the state of the LEDs
    b. The provided directory will depend on where you installed your CCS installation. Please adjust accordingly



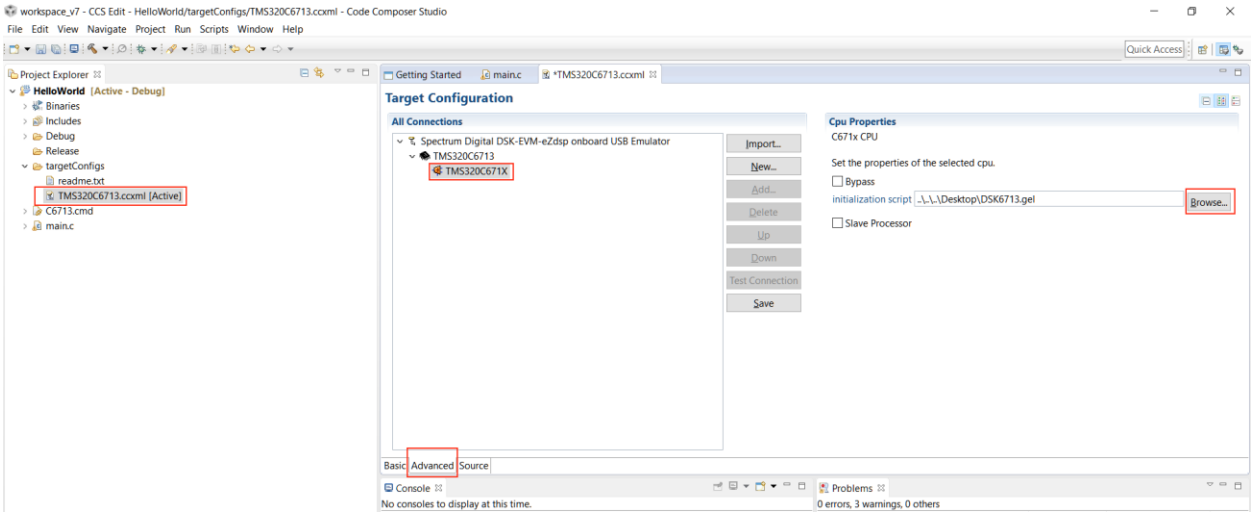
5. Now we need to link the library files that contain the actual source code for the support library functions. In the Project Explorer, right-click your project name and select **Show Build Settings**
6. A new dialog will appear. Select **CCS Build->C6000 Linker->File Search Path** in the list on the left hand side. Under the first textbox, select the paper icon with the green plus to add a new library. Type in **dsk6713bsl.lib** and select OK



7. Repeat Step 7 to add another library named **csl6713.lib**

a. Repeat Step 7 and Step 8 for the Release configuration by selecting the combo box at the top and changing it from Debug to Release

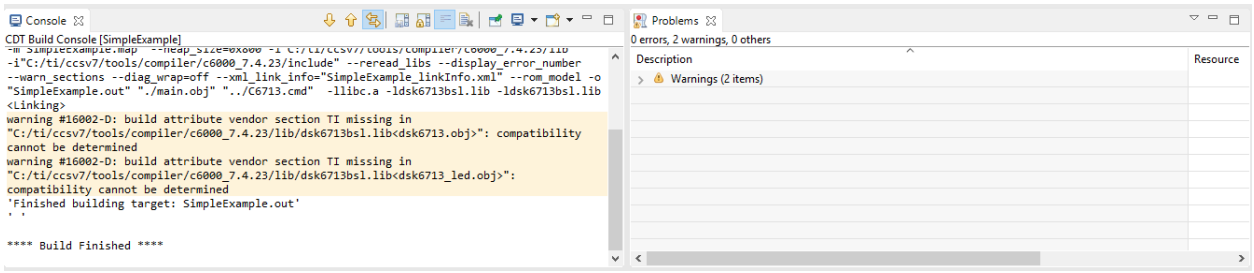
8. Press OK again to exit the build settings and open the file **TMS320C6713.ccxml** (it may also be located in the directory **targetConfigs**). Select the **Advanced** tab at the bottom of the page. Select **TMS320671X** in the connections tree view. Finally, set the **initialization script** under Cpu Properties to the GEL script (DSK6713.gel) provided.
    a. The GEL script contains to flush the cache, reset the board, and other things that are not absolutely required to build your project. However, one issue that arises without the GEL script is running your project on the board requires you to power cycle the board each time.



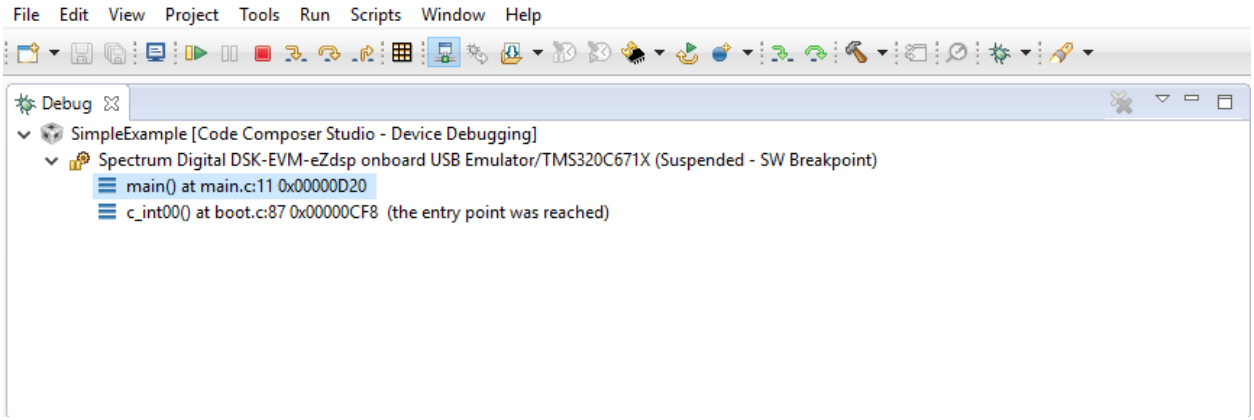
9. Press OK again to exit the build settings and open the file **main.c** from the Project Explorer. This should be familiar to you since this is where the C code will be at. As an initial test, copy/paste the code below into the file
    a. The CHIP_6713 macro is required for **EVERY** project to let the library know what board this is for.
    b. The three include files are also necessary to import the functions dsk6713_XXX
    c. In future projects, **don't** forget to call dsk6713_init() and supplementary init's or the program will not work correctly
    d. In this example, LED 2 (they are labelled on the board) will toggle on/off at a rate of 1Hz

```c
/*
 * main.c
 */
#define CHIP_6713

#include "dsk6713.h"
#include "dsk6713_led.h"
#include "dsk6713_flash.h"

void main(void)
{
    DSK6713_init();
    DSK6713_LED_init();

    while (1)
    {
        DSK6713_LED_toggle(2);
        DSK6713_waitusec(1000000);
    }
}
```

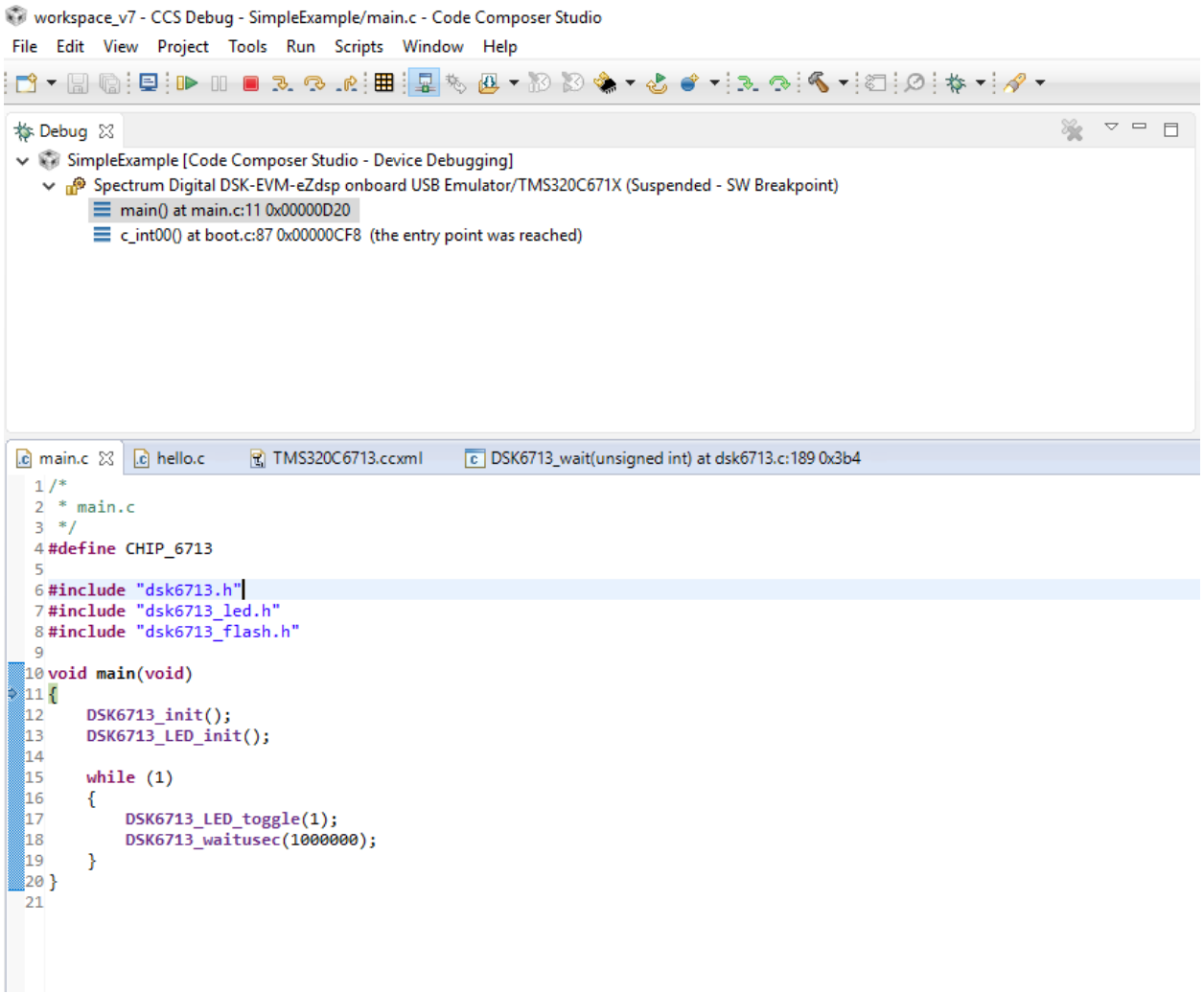
10. Press the hammer icon to build the source code into a binary with extension .out. You will know it built correctly if the two panels at the bottom **Console** and **Problems** show that no errors occurred
    a. If you get an “undefined reference…” error, that means you made a mistake with linking the .lib file
    b. If you get an “cannot find some include .h file” error, that means you may have forgotten to copy the include files into the correct directory



11. Finally, we will upload the binary to the board and run the code. Select **Run->Debug** or press **F11**
    a. CCS will change from edit mode to run mode.
    b. In the **Debug** panel, you should see the emulator name with the following in parenthesis: (Suspended – SW Breakpoint). This means the binary was successfully uploaded and it is waiting to run.



12. Click the **Resume** button at the top or press **F8.** Your code should now be running and the Debug panel should say (Running).
13. Now, doing this every time you want to test some code can be a major pain. Instead, it is easier to press the **Suspend** button or **Alt+F8.**
14. You can now make changes to your code as you see fit. Finally, press the **Build** button. Select Yes to reloading the program automatically. It will be in the same state as step 11. Click **Resume** and you are editing code again.



Frequently Asked Questions

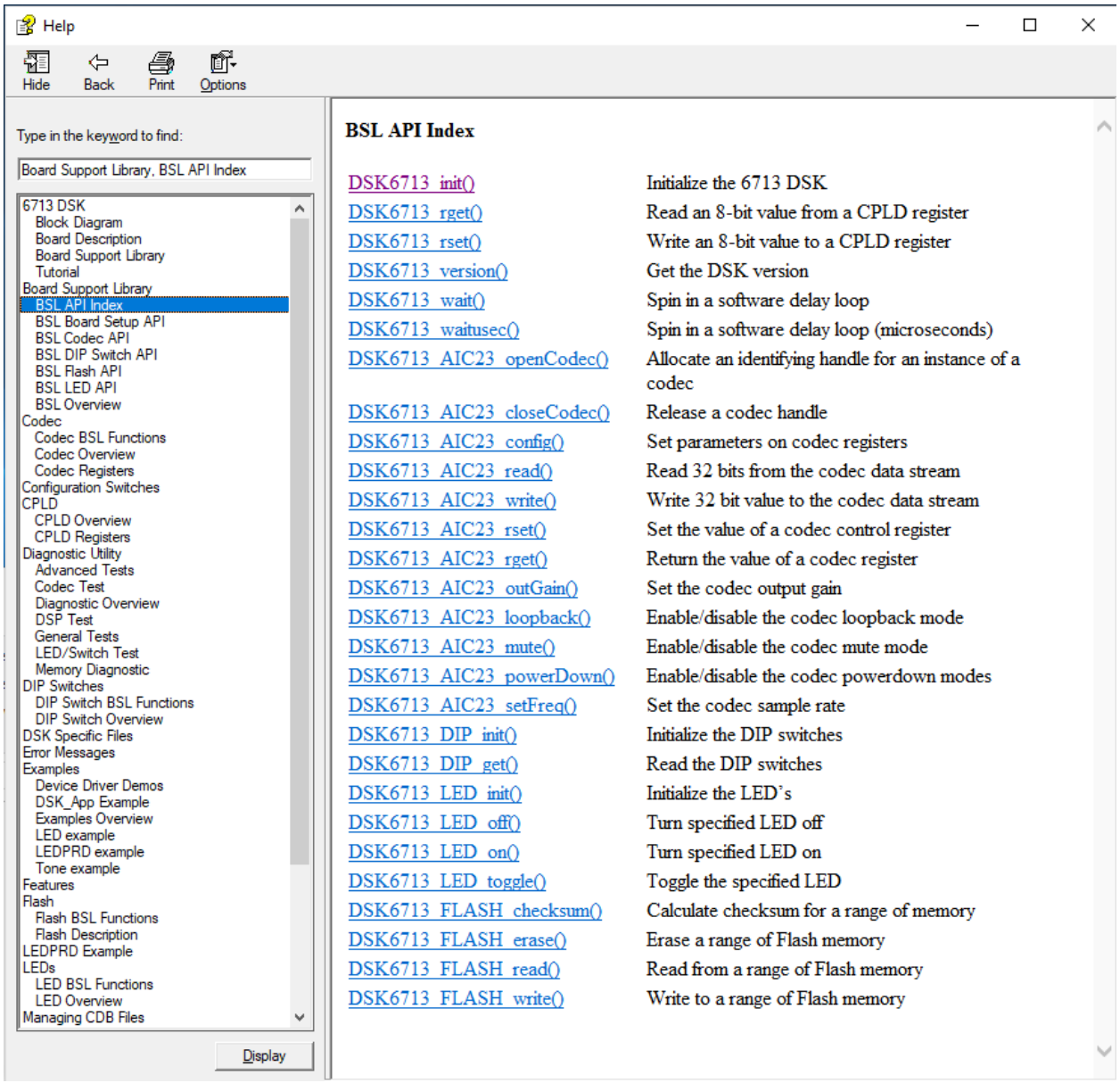
- Where can I find necessary schematics, technical reference manuals, and other documentation for the DSK?
    - Everything can be found here: http://c6000.spectrumdigital.com/dsk6713/revc/

      Additionally, some of these documents are contained in the **docs** directory of the ZIP file provided with the project.
- How can I see what functions are available to use?
    - Well, many of the traditional functions in the C language are available such as *malloc*, *free*, *printf*, etc.

      However, there are some special functions from the Board Support Library (BSL) that was included. The API reference can be seen in the help file **C6713DSK.chm** located inside the **docs** folder in the ZIP file provided with the project. Double-click the CHM file and a menu will open.

This documentation is 10+ years older when CCS was at version 3.1 and thus some of the information may not be applicable. However, the following sections contain information regarding the BSL API:

- BSL API Index
- BSL Board Setup API
- BSL Codec API
- BSL DIP Switch API
- BSL Flash API
- BSL LED API
- Codec BSL Functions



- I received error message "**function _acos in file acos.c: invalid instruction schedule generated …**"
    - This is an odd message that happens the **first** time you build a project with CCS. If you simple clean the project and try to rebuild it, the error message should disappear.