

Code Composer Studio Operation Manual

Contents

Code Composer Studio Operation Manual	1
Contents	1
Section 1: Launching CSS	1
Section 2: Create Project & Preparing Project Setting	3
Section 3: Preparing Build Settings	6
Section 4: Importing Files.....	12
Section 5: Building Your Program.	12
Section 6: Loading Your Program.....	13
Section 7: Running the Executable file exp6.out.....	14
Section 8: Plotting with CCS.	15

****Note: The manual use Experiment 6 as example**

Section 1: Launching CSS

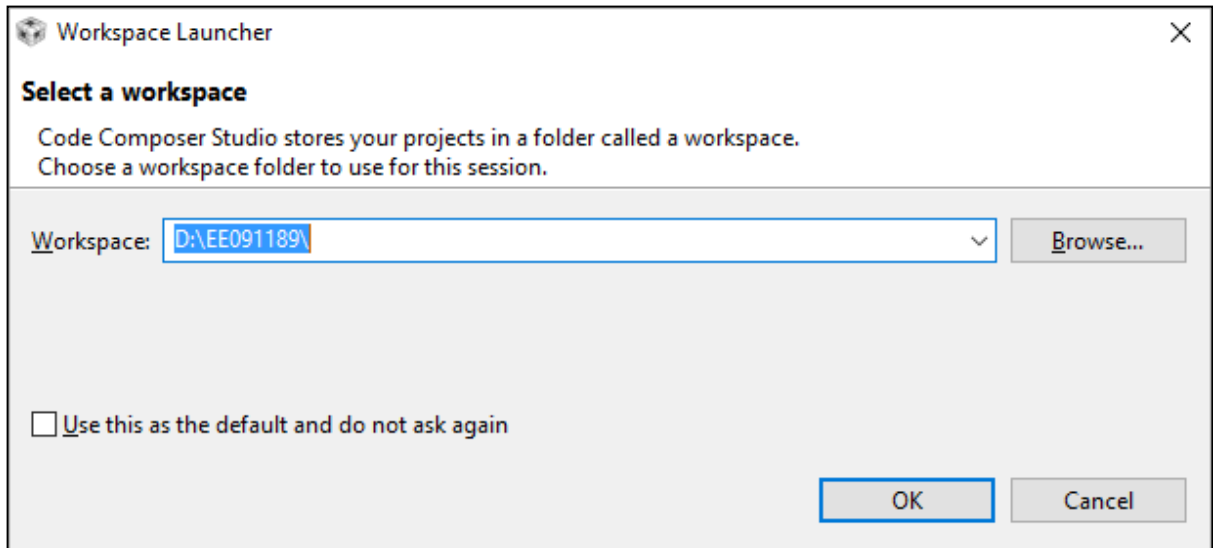
To create a working project, follow these steps:



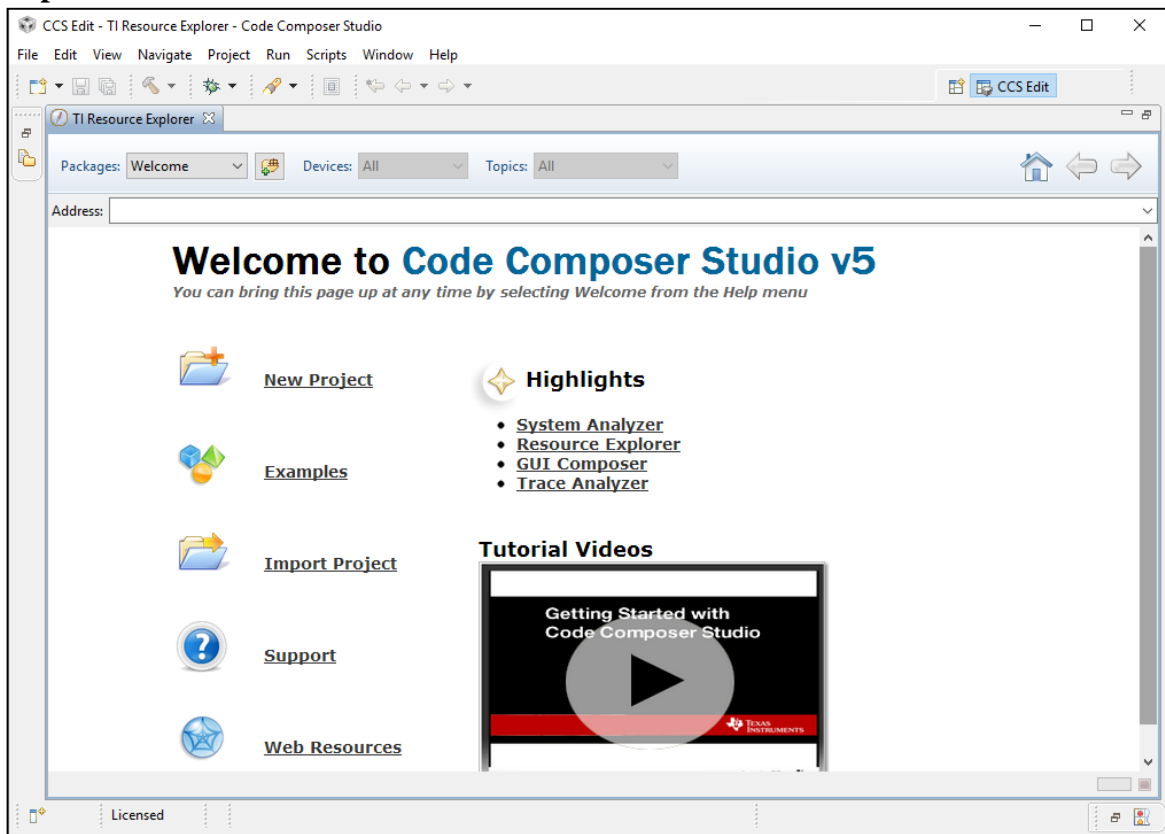
1. **Double-Click** the CCS icon on the Desktop. You will see the CCSv5 splash screen while CCS loads.



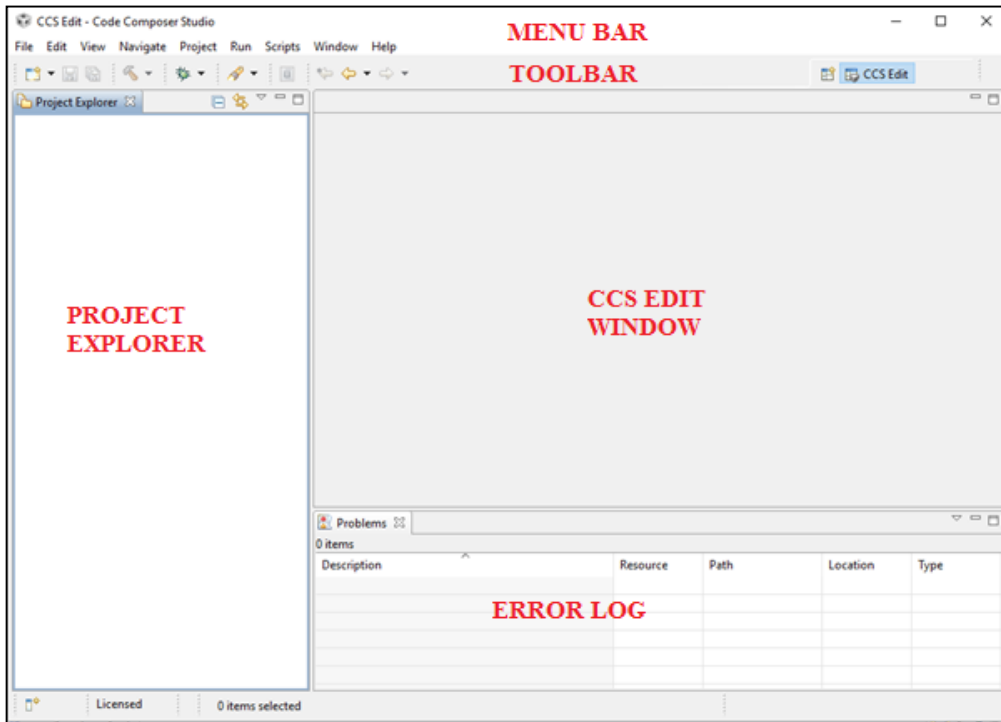
2. The next window will be the **Workspace Launcher** window which will ask you where you want to locate your **CCSv5 workspace**. Replace the location with D:\<Your_Student_ID> (example: D:\EE091189\ as shown in figure below)



3. Upon running the directory for the first time, you will be prompted with **TI Resource Explorer** screen

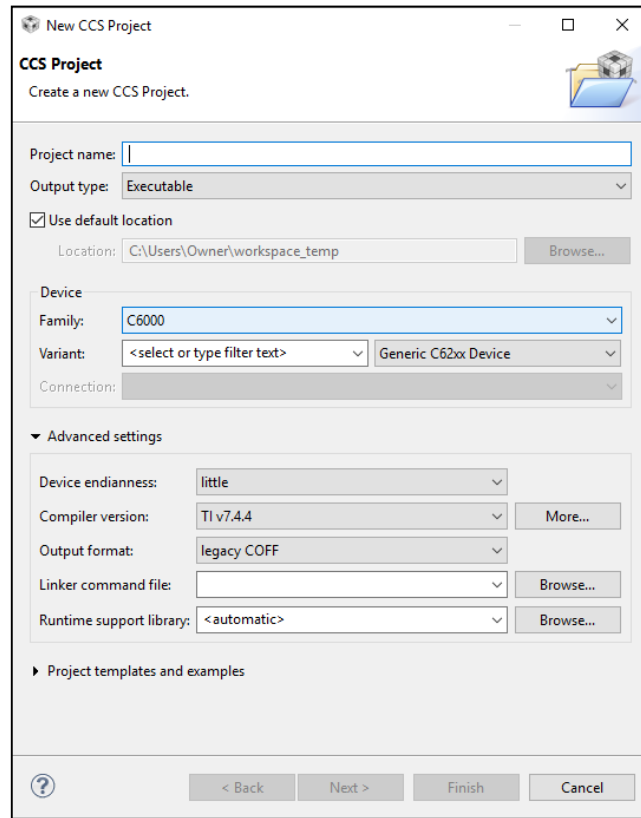


4. This screen is useful when making TI CCS projects which use TI tools. The DSK6713 uses standard features and therefore does not use the TI resources Explorer. Close the **TI Explorer** Screen. You will be left in the **Project Explorer Default Menu**

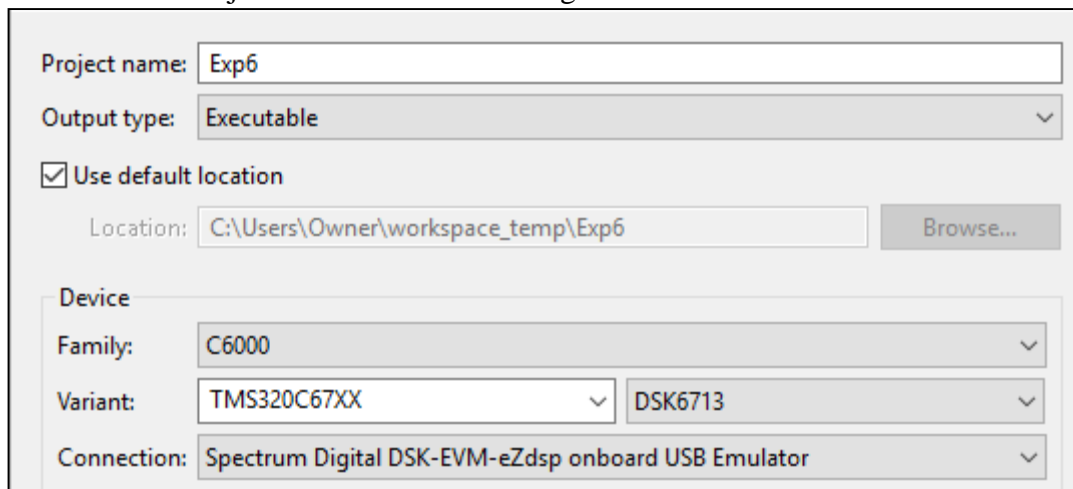


Section 2: Create Project & Preparing Project Setting

On Menu Bar, Select **Project** → **New CCS Project**. You will be prompted with **New CCS Project Window**



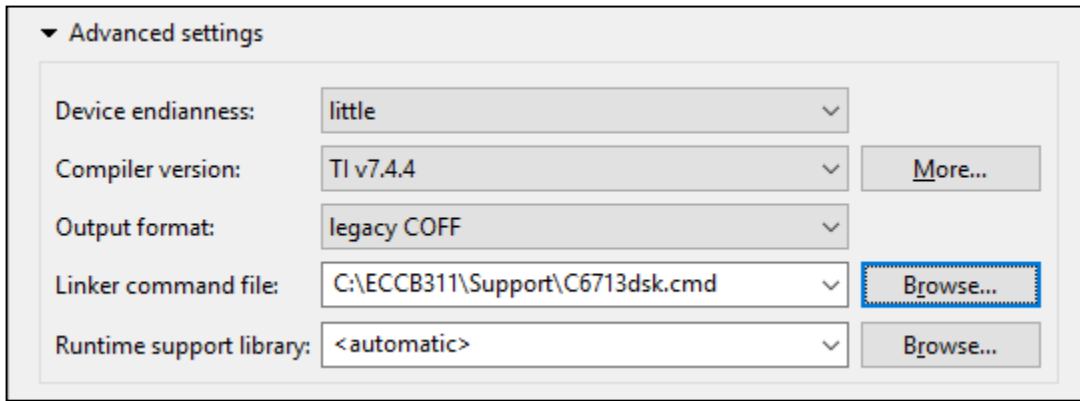
1. Fill in the Project Details as shown in figure below



- Project Name : **ProjectName** (Example: **Exp6**)
- Output type : **Executable**
- Tick use default location

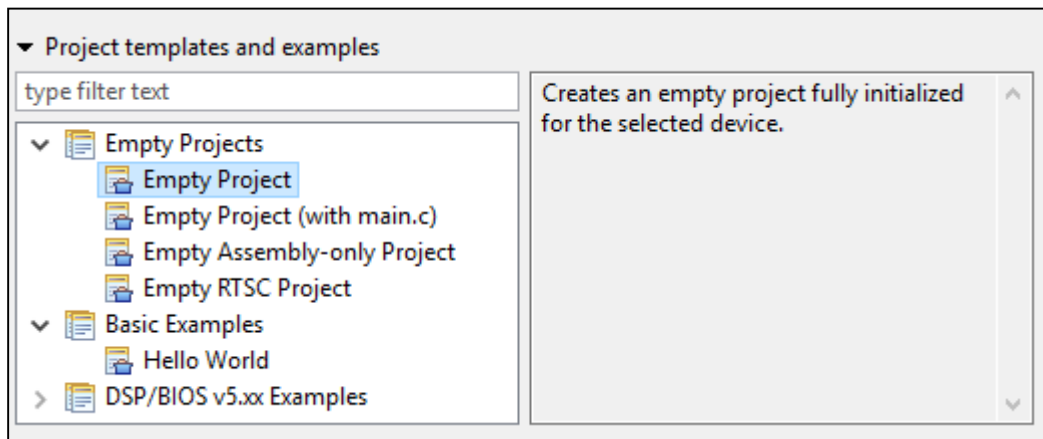
- Device Family: **C6000**
- Variant : **TMS320C67XX** **DSK6713**
- Connection : **Spectrum Digital DSK-EVM-eZdsp onboard USB Emulator**

2. Click on the **Advance Settings** to bring down the menu. Fill in the information as shown



- **Device endianness** : little
- **Compiler version** : TI v7.4.4
- **Output format** : legacy OFF
- **Linker command file**
 1. Click Browse
 2. Browse to “C:\ECCB311 \Support\”
 3. Select **C6713dsk.cmd**
- **Runtime Support Library** : <automatic>

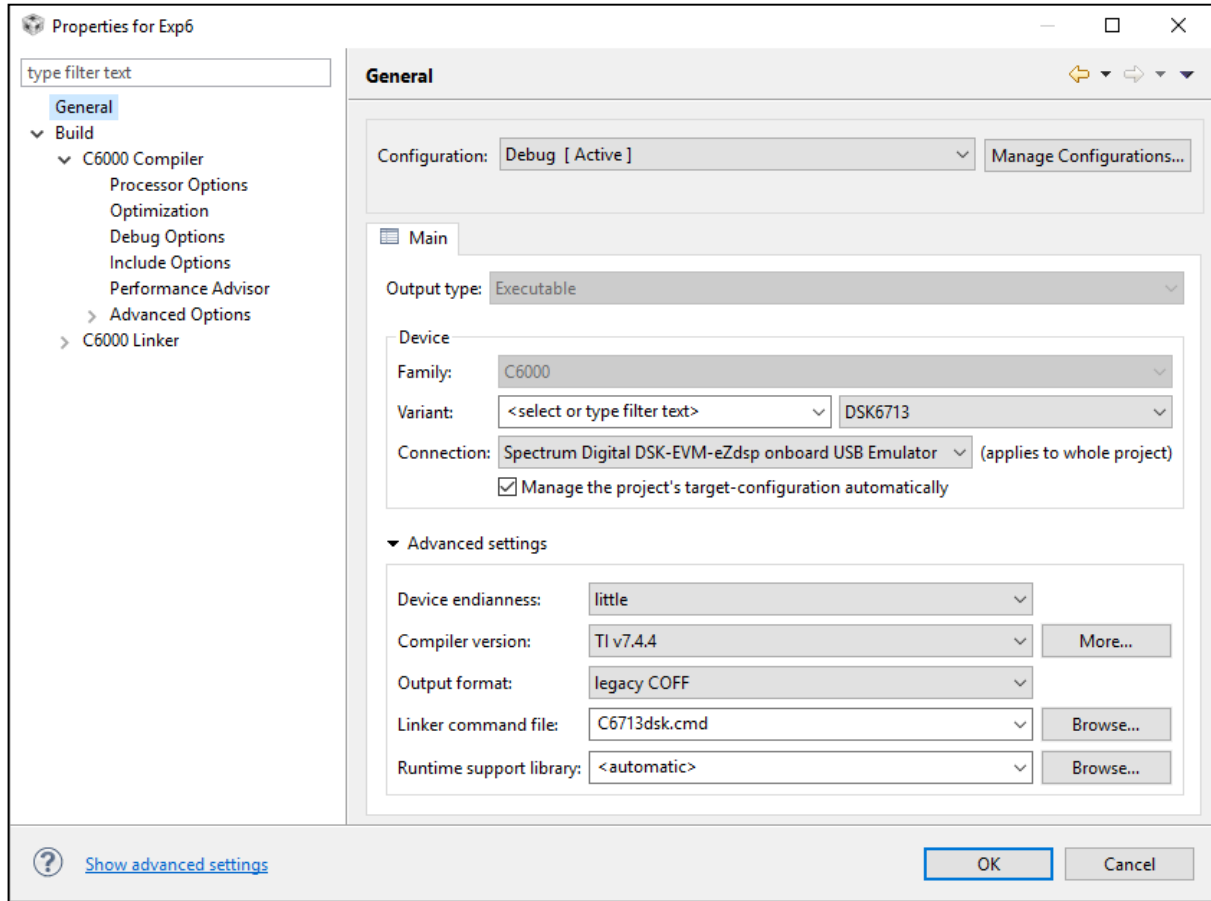
3. Click on the **Project templates and Examples** to bring down the menu. Click on **Empty Project**



Then click **Finish**

Section 3: Preparing Build Settings

On Menu Bar, Select **Project** → **Show Build Settings...** You will be prompted with Build Properties window



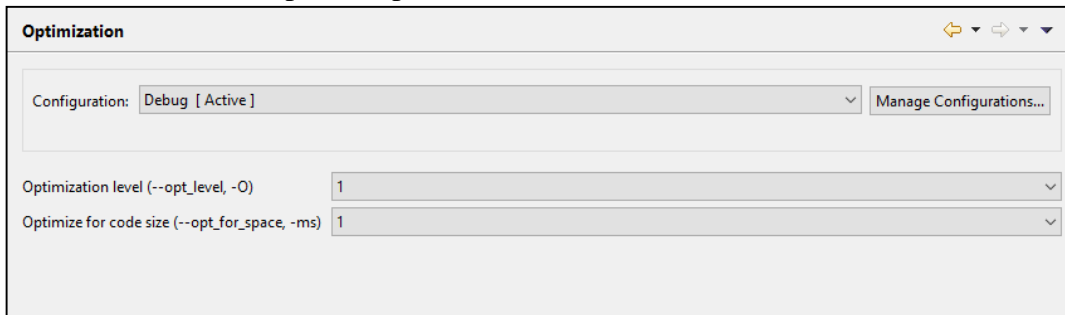
1. Under **C600 Compiler** > **Processor options**:

Target processor version (-- silicon_ versino, -mv) : **6700**



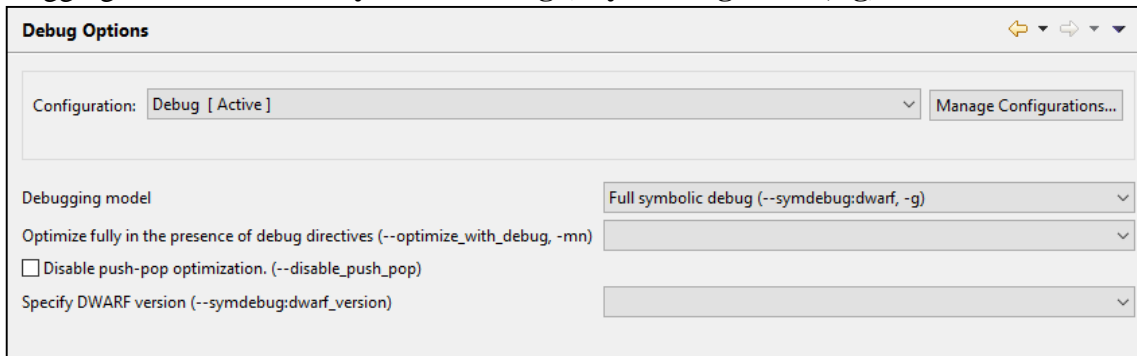
2. Under C600 Compiler > Optimization

Optimization level (--opt_level, -O) : 1
Optimize for code size (--opt_for_space, -ms) : 1

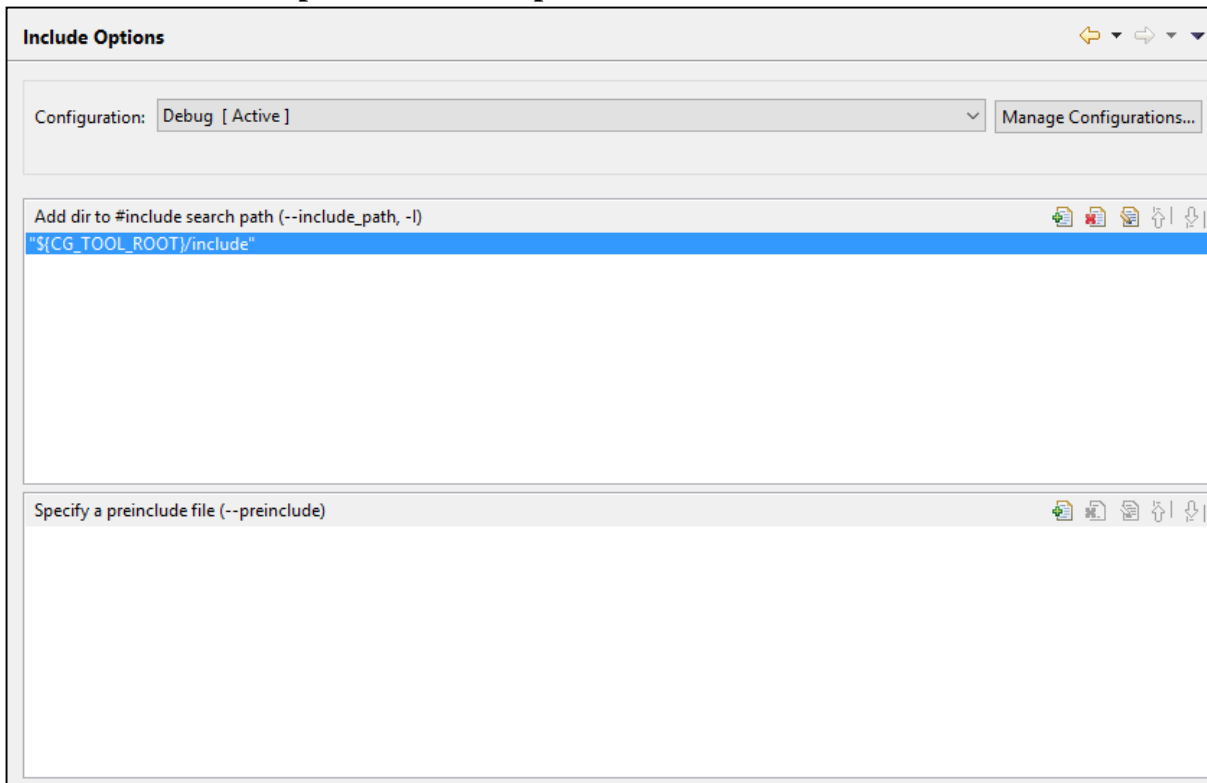


3. Under C6000 Compiler > Debug Options

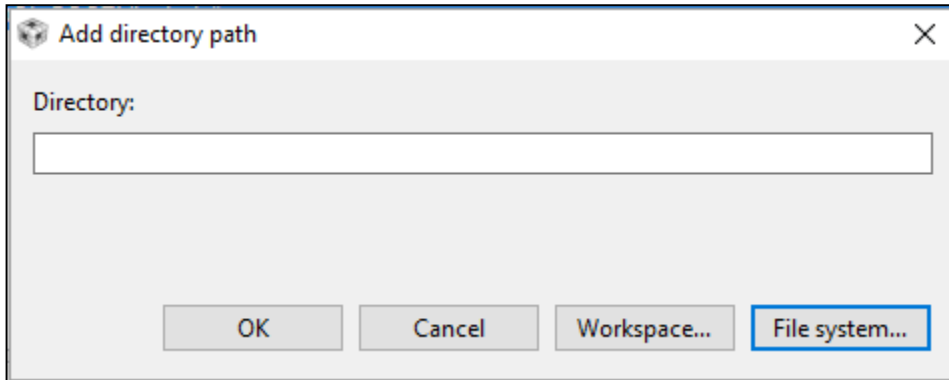
Debugging model : Full symbolic debug (--symdebug:dwarf, -g)



4. Under C6000 Compiler > Include Options



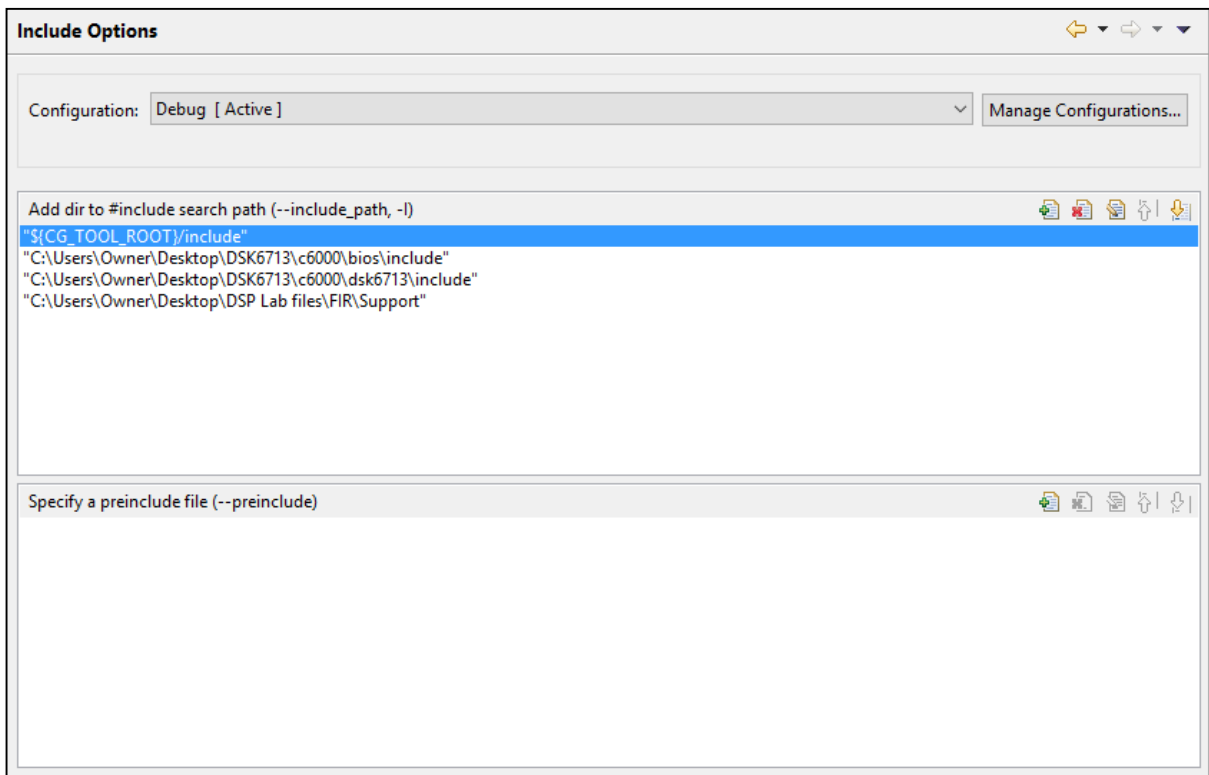
Under **dir to #include search path (--include_path, -I)**, click on the  button and you will be prompted with **Add directory path** window.



Type in the location and press enter. Then repeat for every location as listed below

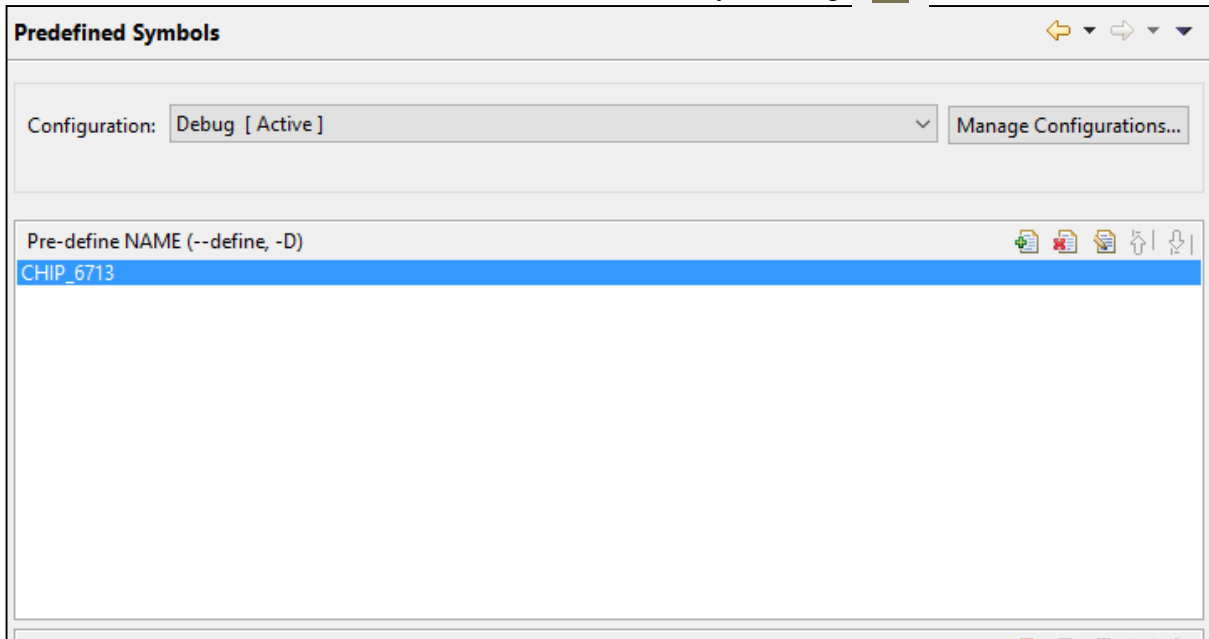
- C:\ECCB311\DSK6713\c6000\bios\include
- C:\ECCB311\DSK6713\c6000\dsk6713\include
- C:\ECCB311\Support

Your **Include options** window should look like figure shown below



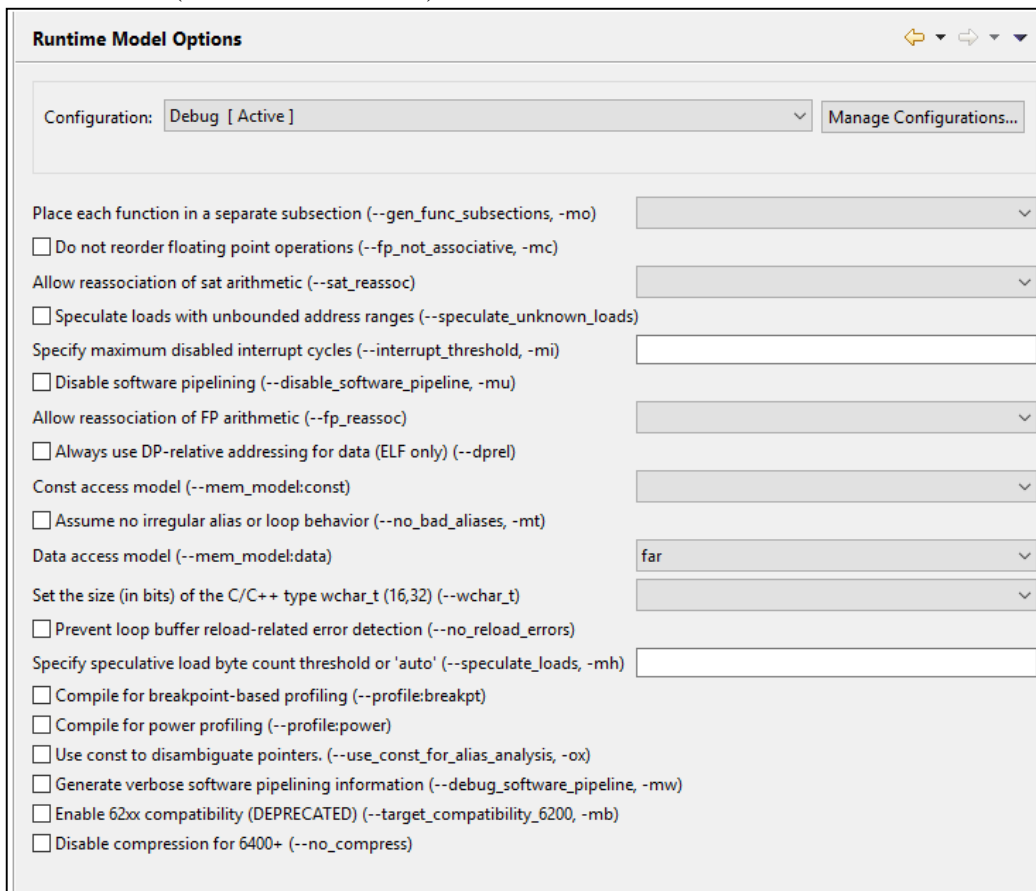
5. Under **C6000 Compiler > Advanced Options > Predefined Symbols**

Add CHIP_6713 to **Pre-define NAME (--define, -D)** by clicking



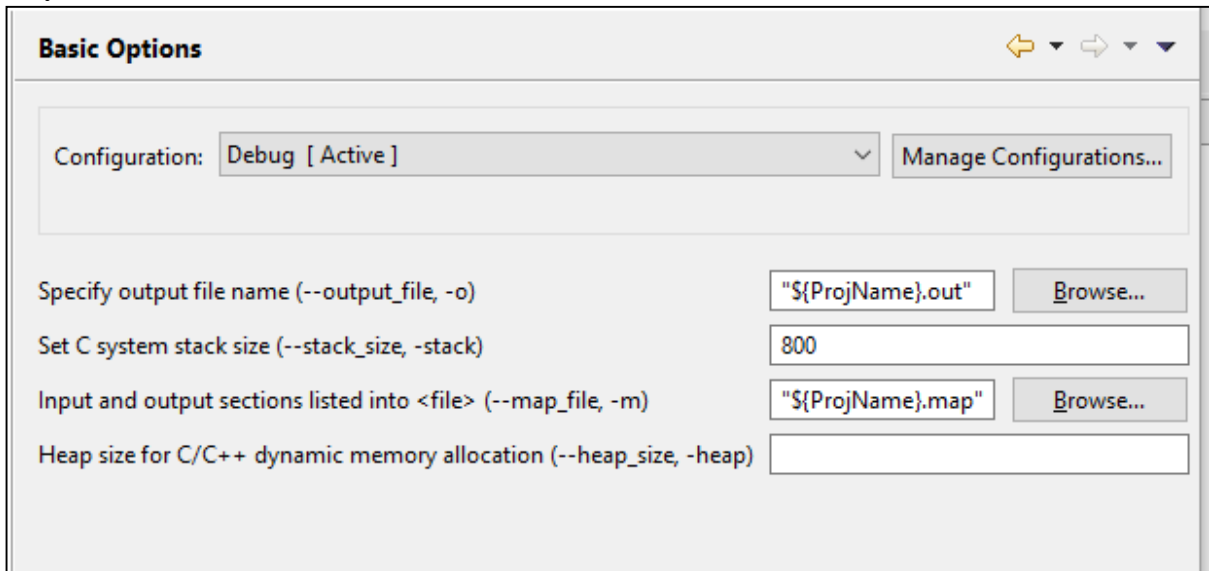
6. Under **C6000 Compiler > Advanced Options > Runtime Model Options**

Data access model (--mem_model:data) **:far**

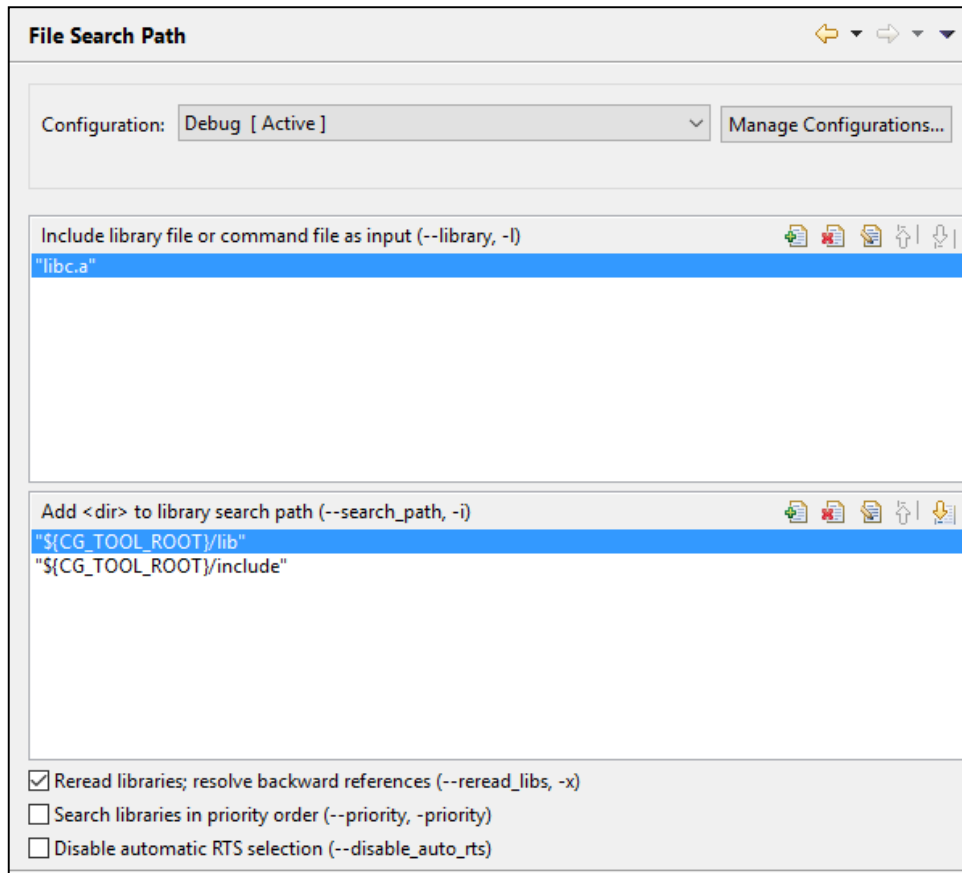


7. Under **C6000 Linker > Basic Options**

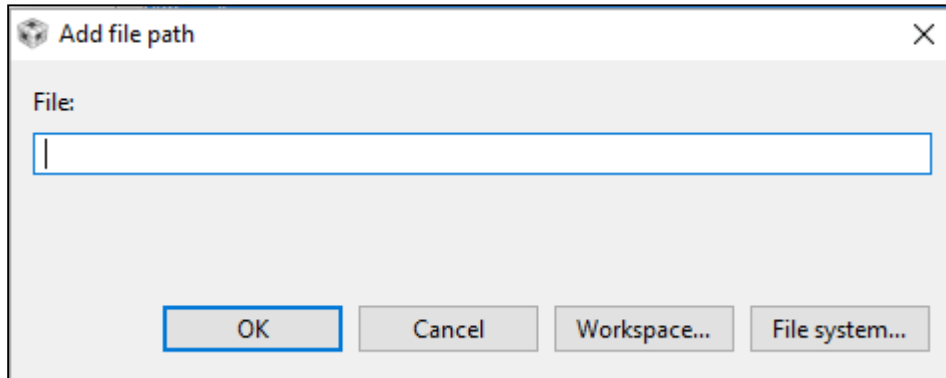
C system stack size (--stack_size, -stack) : **800**



8. Under **C6000 Linker > File Search Path**

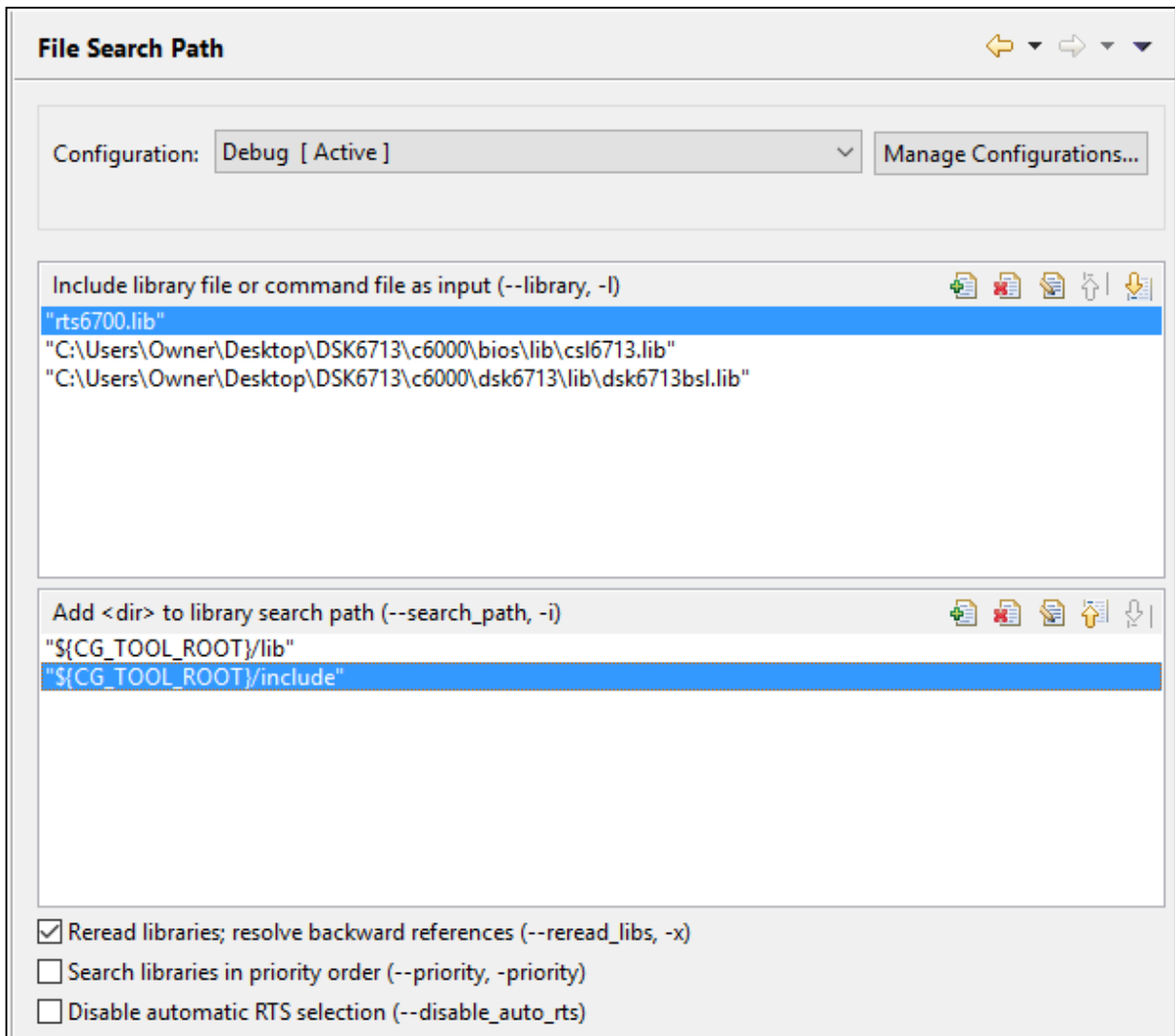


Under **Include library file or command file as input (--library, -l)**, click on the  button and you will be prompted with **Add file path** window.



Type in the location and press **OK**. Then repeat for every location as listed below

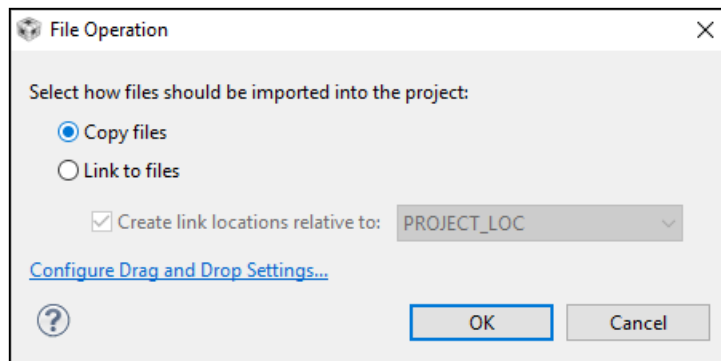
- C:\ECCB311\DSK6713\c6000\bios\lib\csl6713.lib
- C:\ECCB311\DSK6713\c6000\dsk6713\lib\dsk6713bsl.lib



9. Once you finish preparing build settings, press **OK** on the properties window.

Section 4: Importing Files

1. In **Project Explorer Window**, make sure your project is selected
2. Add **sine_8_buf.c**
 - On Menu Bar, Select **Project → Add Files...** You will be prompted with Add Files window
 - Navigate to **C:\ECCB311\sine8_buf** folder
 - Add **sine8_buf.c**



***Note:** When prompted with this window, select **copy files** and press **OK**

3. Add **C6713dskinit.c**
 - Navigate to **C:\ECCB311\Support** folder
 - Add **C6713dskinit.c**
4. Add **vectors_intr.asm**
 - Navigate to **C:\ECCB311\Support** folder
 - Add **vectors_intr.asm**

Section 5: Building Your Program.

Project → Build Project

After you build your project, all the necessary include files will appear in the include tab in the project explorer.

The error log window will show the build process and status. When the build is finished, the error log window will display *****Build Finished***** and you will get *8 warnings*

NOTE:

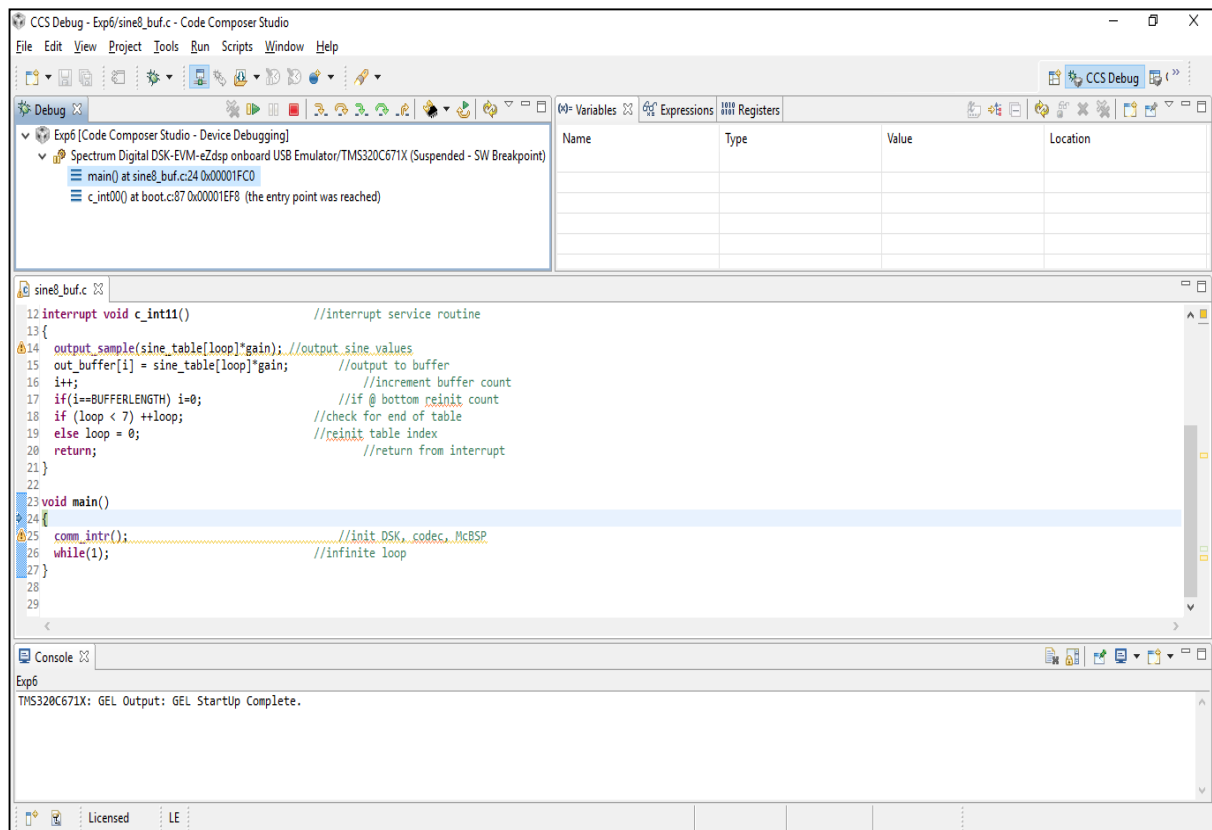
1. Since the DSK Starter kit is an old hardware, it is not fully compatible running on a newer platform. Hence you will get 8 warnings)
2. The **Rebuild All** command is mainly used to rebuild the project when the project options or any files in the project have changed.

Section 6: Loading Your Program.

Select **Run → Debug**.

A new **CCS Debug window** (as shown in figure below) will appear.


(**NOTE:** you need to connect the DSK starter to your PC using USB. If there is no connection, an error will appear and the **CCS Debug window** will be closed)

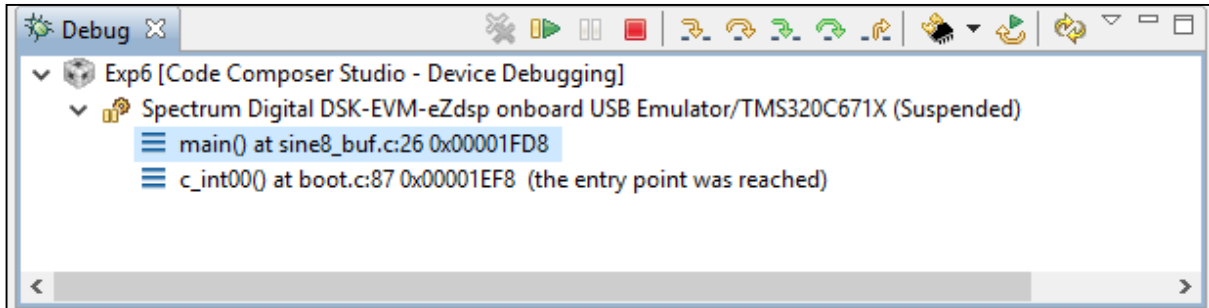


After the program has been built successfully, load the program by going to **Run → Load → Load Program**. By default, Code Composer Studio IDE will create a subdirectory called Debug within your project directory and store the .out file in it. Select **exp6.out** and click **Open** to load the program.



(**Note:** Remember to reload the program by choosing **Run → Load → Reload Program** if you rebuild the project after making changes)

Section 7: Running the Executable file exp6.out.


On the debug window, click  to run program. Connect a speaker or scope to the LINE OUT connector on the DSK. Verify that a 1 KHz tone or sinusoid is generated.



Note:

1. Click  to **HALT** the programme.
2. Click  to **STOP** the programme. By clicking this, the **CCS Debug** window will close and you will return to **CCS Edit** window

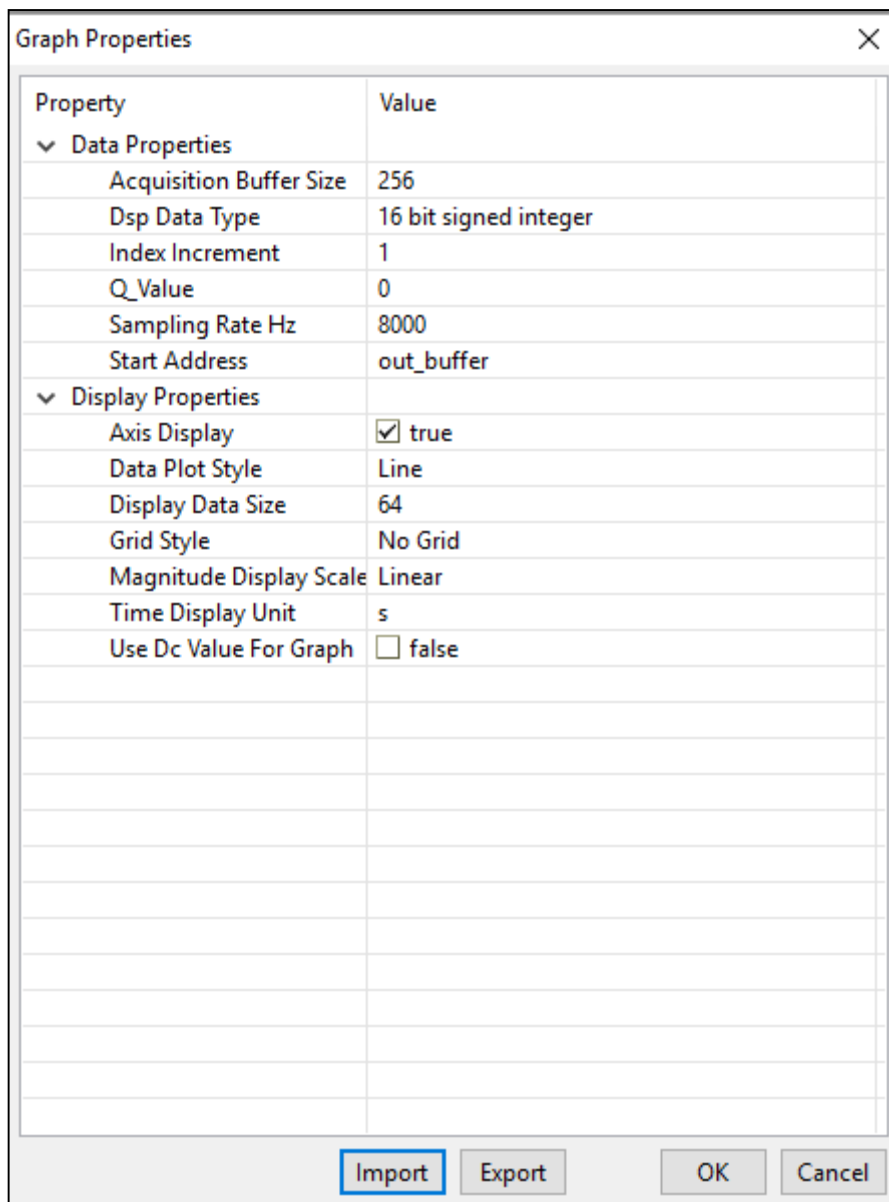
Section 8: Plotting with CCS.

Halt the DSK by selecting **Run** → **Suspend** or the  icon.

The output buffer is continuously being updated every 256 points. We can use the CCS to plot the current output data stored in the out_buffer.

1. Display Single Time graph using CCS.

Select **Tools** → **Graph** → **Single Time**. Change the graph Property Dialog as follows. Click **OK** and time domain graph will be plotted



The image shows a 'Graph Properties' dialog box with a table of settings. The 'Data Properties' section is expanded, showing values for Acquisition Buffer Size (256), Dsp Data Type (16 bit signed integer), Index Increment (1), Q_Value (0), Sampling Rate Hz (8000), and Start Address (out_buffer). The 'Display Properties' section is also expanded, showing values for Axis Display (checked true), Data Plot Style (Line), Display Data Size (64), Grid Style (No Grid), Magnitude Display Scale (Linear), Time Display Unit (s), and Use Dc Value For Graph (unchecked false). At the bottom, there are buttons for 'Import', 'Export', 'OK', and 'Cancel'.

Property	Value
▼ Data Properties	
Acquisition Buffer Size	256
Dsp Data Type	16 bit signed integer
Index Increment	1
Q_Value	0
Sampling Rate Hz	8000
Start Address	out_buffer
▼ Display Properties	
Axis Display	<input checked="" type="checkbox"/> true
Data Plot Style	Line
Display Data Size	64
Grid Style	No Grid
Magnitude Display Scale	Linear
Time Display Unit	s
Use Dc Value For Graph	<input type="checkbox"/> false

