# 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.



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)

💱 Workspace Launcher		×
Select a workspace		
Code Composer Studio stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.		
Workspace: D:\EE091189\	~	<u>B</u> rowse
Use this as the default and do not ask again	OK	Cancel

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

😵 New CCS Pr	oject				-	- 🗆	×
CCS Project Create a new C	CS Project.						
Project name: Output type: ☑ Use default	 Executable location						~
Location:	C:\Users\0	wner\workspace	e_temp			Brows	e
Device Family: Variant:	C6000 <select or<="" th=""><th>type filter text&gt;</th><th>~</th><th>Generic</th><th>C62xx Device</th><th></th><th><b>&gt;</b></th></select>	type filter text>	~	Generic	C62xx Device		<b>&gt;</b>
<ul> <li>Advanced set</li> </ul>	ettings						
Device endiar	nness:	little			~		
Compiler ver	sion:	TI v7.4.4			$\sim$	More	
Output forma	at:	legacy COFF			~		
Linker comm	and file:				~	Browse	
Runtime supp	port library:	<automatic></automatic>			~	Browse	
<ul> <li>Project temp</li> </ul>	plates and e	kamples					
?		< Back	Next >		Finish	Can	cel

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

Project name:	Ехрб	
Output type:	Executable	~
🗹 Use default	location	
Location:	C:\Users\Owner\workspace_temp\Expб	Browse
Device		
Family:	C6000	~
Variant:	TMS320C67XX ~ DSK6713	~
Connection:	Spectrum Digital DSK-EVM-eZdsp onboard USB Emulator	~

- 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

<ul> <li>Advanced settings</li> </ul>		
Device endianness:	little	~
Compiler version:	TI v7.4.4	<ul> <li>✓ <u>M</u>ore</li> </ul>
Output format:	legacy COFF	~
Linker command file:	C:\ECCB311\Support\C6713dsk.cmd	✓ Browse
Runtime support library:	<automatic></automatic>	✓ B <u>r</u> owse

- Device endianness : little
  Compiler version : TL v7
- 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

type filter text       Creates an empty project fully initialized for the selected device.         Empty Project       For the selected device.         Empty Project (with main.c)       For the selected device.         Empty Assembly-only Project       For the selected device.         Empty RTSC Project       For the selected device.         For the selected device.       For the selected	<ul> <li>Project templates and examples</li> </ul>		
	type filter text	Creates an empty project fully initialized for the selected device.	< >

Then click **Finish** 

# Section 3: Preparing Build Settings

On Menu Bar, Select <b>Project</b> → Show Build Settings.	You	will be	prompted	with Build
Properties window				

une filter text	<b>6</b>			<u> </u>
General	General			<b>···</b> ·································
<ul> <li>Build</li> <li>C6000 Compiler</li> <li>Processor Options</li> </ul>	Configuration: Debug [ A	ctive ]	∽ Manag	je Configurations
Debug Options	Main			
Performance Advisor > Advanced Options	Output type: Executable			~
> C6000 Linker	Device			
	Family: C6000			~
	Variant: <select or<="" td=""><td>type filter text&gt; ~</td><td>DSK6713</td><td>~</td></select>	type filter text> ~	DSK6713	~
	Connection: Spectrum [ Manage	Digital DSK-EVM-eZdsp onboard the project's target-configuratio	USB Emulator v (applies n automatically	to whole project)
	<ul> <li>Advanced settings</li> </ul>			
	Device endianness:	little	$\checkmark$	
	Compiler version:	TI v7.4.4	~	More
	Output format:	legacy COFF	~	
	Linker command file:	C6713dsk.cmd	~	Browse
	Runtime support library:	<automatic></automatic>	$\sim$	Browse

## 1. Under C600 Compiler > Processor options:

Target processor version ( -- silicon\_versino, -mv) : 6700

Processor Opti	ons	← ▼ ⇒ ▼
Configuration:	Debug [Active]	<ul> <li>✓ Manage Configurations</li> </ul>
Target processor	version (silicon_version, -mv)	6700
Generate big e	ndian code [See 'General' page to edit] (big_endian, -me)	
Application binar	y interface (coffabi, eabi) [See 'General' page to edit] (abi)	coffabi $\sim$

## 2. Under C600 Compiler > Optimization

Optimization le Optimize for co	evel (opt_leve ode size (opt_	l, -O) for_space, -ms)	: 1 : 1	
Optimization				← ▼ ⇒ ▼
Configuration:	Debug [Active]			✓ Manage Configurations
Optimization lev	el (opt_level, -O)	1		~
Optimize for coo	le size (opt_for_space, -ms)	1		~

### 3. Under C6000 Compiler > Debug Options

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

Debug Options 🗘 🕆 🗘					
Configuration:	Debug [ Active ]	~	Manage Configurations		
Debugging mode	el	Full symbolic debug (symdebug:dwarf, -g)	~		
Optimize fully in	the presence of debug directives (optimize_with_debug, -mn) pop optimization. (disable_push_pop)		~		
Specify DWARF v	ersion (symdebug:dwarf_version)		~		

### 4. Under C6000 Compiler > Include Options

Include Options	⇔ • ⇒ • •
Configuration: Debug [Active]	Manage Configurations
Add dir to #include search path (include_path, -I)	🗐 🗐 😼 🖓 🖓
Specify a preinclude file (preinclude)	🛃 🔊 🗟 한 한

Ð	button and	VOII
	oution and	you

Under **dir to #include search path (--include\_path, -l)**, click on the will be prompted with **Add directory path** window.

💱 Add dire	ctory path			×
Directory:				
	ОК	Cancel	Workspace	File system

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

Include Option	15	⇐ • ⇒ • •
Configuration:	Debug [ Active ]	✓ Manage Configurations
Add dir to #incl "S(CG_TOOL_RO "C:\Users\Owne "C:\Users\Owne "C:\Users\Owne	lude search path (include_path, -I) )OT}/include" r\Desktop\DSK6713\c6000\bios\include" r\Desktop\DSK6713\c6000\dsk6713\include" r\Desktop\DSP Lab files\FIR\Support"	🛃 🗟 불 处
Specify a preinc	:lude file (preinclude)	● 图 상 상

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

Predefined Syr	nbols		$\leftarrow \bullet \bullet \bullet \bullet$
Configuration:	Debug [Active]	~	Manage Configurations
Pre-define NAN CHIP_6713	IE (define, -D)		🗐 🗐 🞯 🖓 🖓

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

Data access model (--mem\_model:data) :far

Runtime Mode	el Options		↓ ↓ ↓ ▼
Configuration:	Debug [ Active ]	~	Manage Configurations
Place each functi	ion in a separate subsection (gen_func_subsections, -mo)		~
Do not reorde	r floating point operations (fp_not_associative, -mc)		
Allow reassociati	on of sat arithmetic (sat_reassoc)		~
Speculate load	ds with unbounded address ranges (speculate_unknown_loa	ads)	
Specify maximur	n disabled interrupt cycles (interrupt_threshold, -mi)		
Disable softwa	are pipelining (disable_software_pipeline, -mu)		,
Allow reassociati	on of FP arithmetic (fp_reassoc)		~
Always use DF	P-relative addressing for data (ELF only) (dprel)		
Const access mo	del (mem_model:const)		~
Assume no irr	egular alias or loop behavior (no_bad_aliases, -mt)		
Data access mod	el (mem_model:data)	far	~
Set the size (in bi	ts) of the C/C++ type wchar_t (16,32) (wchar_t)		~
Prevent loop I	buffer reload-related error detection (no_reload_errors)		
Specify speculati	ve load byte count threshold or 'auto' (speculate_loads, -m	h)	
Compile for b	reakpoint-based profiling (profile:breakpt)		
Compile for p	ower profiling (profile:power)		
Use const to d	lisambiguate pointers. (use_const_for_alias_analysis, -ox)		
Generate verb	ose software pipelining information (debug_software_pipel	ine, -mw)	
Enable 62xx co	ompatibility (DEPRECATED) (target_compatibility_6200, -mb	)	
Disable comp	ression for 6400+ (no_compress)		

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

Basic Options		<	▶ • ⇔ • •
Configuration:	Debug [ Active ]	∽ Manage Co	nfigurations
Specify output fil	e name (output_file, -o)	"\${ProjName}.out"	<u>B</u> rowse
Set C system stac	k size (stack_size, -stack)	800	
Input and output	sections listed into <file> (map_file, -m)</file>	"\${ProjName}.map"	<u>B</u> rowse
Heap size for C/C	++ dynamic memory allocation (heap_size, -heap)		

C system stack size ( --stack\_size, -stack) : 800

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

File Search Pat	h	¢	• 🔶 • •
Configuration:	Debug [ Active ] V	Manage Confi	gurations
Include library fi "libc.a"	le or command file as input (library, -l)	£ 8	🗟 실 한
A.1. P PI			
Add <dir> to lib</dir>	rary search path (search_path, -i) OTV/ib"		🗑 🖓 🔛
"\${CG_TOOL_RO	OT}/include"		
Reread librarie	s; resolve backward references (reread_libs, -x)		
Search libraries	s in priority order (priority, -priority)		
Disable autom	atic RTS selection (disable_auto_rts)		

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.

💱 Add file path			×
File:			
ОК	Cancel	Workspace	File system

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 •

File Search Pat	h	↓ ↓ ↓ ↓
Configuration:	Debug [ Active ]	✓ Manage Configurations
Include library fi	ile or command file as input (library, -l)	🗐 🍙 😭 🖓
"C:\Users\Owner	\Desktop\DSK6713\c6000\dsk6713\lib\dsk6713bsl.lib"	
Add <dir> to lib</dir>	vrary search path (search_path, -i)	🛃 📾 🖗 🖓 🖓
"\${CG_TOOL_RO	OT}/lib"	
Reread librarie	s; resolve backward references (reread_libs, -x)	
Disable autom	s in priority order (priority, -priority) atic RTS selection (disable auto its)	
	and his selection (disable_auto_its)	

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

File Operation		×
Select how files should be imported into the Copy files	e project:	
Create link locations relative to:	PROJECT_LOC	~
Configure Drag and Drop Settings		
?	ОК	Cancel

\*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)

💱 CCS Debug - Exp6/sine8_buf.c - Code Composer Studio				-	σx
<u>File Edit View Project Iools Run Scripts Window H</u> elp					
📫 • 🔛 🔞 🛍 🏘 • 🔜 🍬 🤬 • 🕅 🕅 🗳 • 🔗 •				🖹 🎭 CCS Debug	🕞 ( »
🕸 Debug 🛛 🦉 🦓 👘 💷 🔳 🚴 👁 🧟 🏟 🗸 🖶 🧔	🗱 🗱 Variables 🛛 🖓 Expressions	888 Registers	🖢 🍂 🕞 🕻	🔉 🖗 🗶 💥 📘 🚺	2 - 1
✓	Name	Туре	Value	Location	
<u>a</u> ⊆ sine8_buf.c ⊠					
<pre>13{     output_sample(sine_table[loop]*gain); //output sine_values 15 out_buffer[i] = sine_table[loop]*gain; //output to buffer 16 i++; //increment buffer count 17 if(i=BUFFELENOTH) i=0; //if(@ bottom reinit count 18 if(loop &lt; 7) ++loop; //heck for end of table 19 else loop = 0; //reinit table index 20 return; //return from interrupt 21 22 23 void main() 24 25 comm_intr(); //init DSK, codec, McBSP 26 while(1); //infinite loop 27 27 27 27 27 27 27 27 27 27 27 27 27</pre>					
<					>
E Console 🛛				🗟 🛃 🛃 🖛	📑 🕶 🗆
Ехрб					
TMS320C671X: GEL Output: GEL StartUp Complete.					~ ~ ~
🛯 🕆 🖻 Licensed LE					

After the program has been built successfully, load the program by going to

**Run**  $\rightarrow$  Load  $\rightarrow$  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  $Run \rightarrow Load \rightarrow Reload Program$  if you rebuild the project after making changes)

# Section 7: Running the Executable file exp6.out.

On the debug window, click is 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 III to **HALT** the programme.
- 2. Click Los **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  $Run \longrightarrow Suspend$  or the  $\square$  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**  $\rightarrow$  **Graph** $\rightarrow$  **Single Time**. Change the graph Property Dialog as follows. Click **OK** and time domain graph will be plotted

raph Properties	×
Property	Value
<ul> <li>Data Properties</li> </ul>	
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
<ul> <li>Display Properties</li> </ul>	
Axis Display	✓ 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	□ false
	mport Export OK Cancel

### 2. Display FFT using CCS.

Select **Tools**  $\rightarrow$  **Graph** $\rightarrow$  **FFT Magnitude**. Change the graph Property Dialog as follows. Click **OK** and time domain graph will be plotted

Graph Properties	×
Property	Value
<ul> <li>Data Properties</li> </ul>	
Acquisition Buffer Size	256
Dsp Data Type	16 bit signed integer
Index Increment	1
Q_Value	0
Sampling Rate Hz	8000
Signal Type	Real
Start Address	out_buffer
<ul> <li>Display Properties</li> </ul>	
Axis Display	✓ true
Data Plot Style	Line
Frequency Display Unit	Hz
Grid Style	No Grid
Magnitude Display Scale	Linear
✓ FFT	
FFT Frame Size	32
FFT Order	5
FFT Window Function	Rectangular
Import Export OK Cancel	