Stellaris® Development and Evaluation Kits for Code Composer Studio™

Stellaris Development and Evaluation Kits provide a low-cost way to start designing with Stellaris microcontrollers using Texas Instruments' Code Composer Studio development tools. The evaluation boards can function as either a complete evaluation target or as a debugger interface to any external Stellaris device.

Requirements

- You have a PC with a USB interface, running Microsoft® Windows XP (SP2 or greater) or Vista
- You have the Stellaris Evaluation Kit Documentation and Software DVD or the standalone Code Composer Studio DVD found in the Development Kit



CAUTION: There is a known electrical issue with the FT2232 device that is used in the on-board In Circuit Debug Interface (ICDI). Some USB hubs can cause the device to misbehave, with symptoms ranging from failed enumeration to corrupt data transfers. If you experience trouble when using the on-board ICDI, try connecting the USB cable directly to one of the USB ports on your PC or laptop.

Code Composer Studio

This quickstart shows you how to install the Code Composer Studio development tool and how to use it to build and run an example application on your Stellaris Evaluation or Development Board.

Step 1: Install Code Composer Studio

- 1. Insert the Evaluation Kit Documentation and Software DVD or the standalone Code Composer Studio DVD into the DVD-ROM drive of your computer. If Autoplay is enabled on your PC, the index.htm file automatically opens in your default web browser. If Autoplay is not enabled, use Windows Explorer to open the DVD manually.
- 2. With the Evaluation Kit DVD, click the Tools button and then click the Code Composer Studio logo to start the setup program.
 - **NOTE:** The CCS installer executable relies on additional installer executables on the DVD. Your web browser must support running files without first saving them to your hard drive to successfully run the CCS installer by clicking the CCS logo. If your web browser does not support this feature, you will need to use Windows Explorer to browse to and run the CCS installer (setup_CCS_n.n.n.nnnnn.exe) from the \Tools\CCS\ directory on the DVD.

With the standalone DVD, follow the installer dialog.



3. Follow the instructions in the Code Composer Studio installation program. Select to install the Platinum Edition or the Microcontroller Edition with the Stellaris Cortex-M3 MCUs ISA selected at a minimum.

Step 2: Install the StellarisWare® Package

A full set of C-based peripheral drivers is provided, covering all peripherals and functionality of the Stellaris devices. The StellarisWare package includes various example applications with project files for all major tool vendors that support Stellaris, including Code Composer Studio. To install StellarisWare components, follow these steps:

1. Navigate to the *Tools* tab on the Evaluation Kit Documentation and Software DVD or Development Kit Documentation and Software DVD.

NOTE: If you are navigating the DVD using Windows Explorer (or a similar application), go to the Tools/StellarisWare/ directory.

- 2. Click on the 'Install' link next in the StellarisWare section of the DVD and run the StellarisWare installer. If you prefer to manually install StellarisWare, the installer is a self-extracting zip file that is located in the Tools/StellarisWare directory. You can use a zip file extraction utility such as WinZip to manually extract the contents.
- 3. To view the StellarisWare documentation, navigate to the installation directory and click on the *Stellaris Peripheral Driver Library User's Guide* PDF.

NOTE: For the most recent version of the StellarisWare, check the <u>www.ti.com/stellaris</u> web site for the latest software updates.

Step 3: Start Code Composer Studio and Open a Workspace

1. Start the Code Composer Studio IDE by selecting it from the Windows Start menu or double-clicking the icon installed on your desktop. When the IDE loads, it asks you where to open the workspace folder:



2. The Workspace launcher defaults to the following path:



C:\Documents and Settings \<username>\My Documents \work
space

Click OK to use this default workspace location.

- **Note:** If you installed the Code Composer Studio IDE from the DVD, the license was automatically installed. If you downloaded the Code Composer Studio IDE, you may have to go through an extra licensing step.
 - 3. The Code Composer Studio IDE may now open with the welcome page. If so, close out the welcome page by clicking the 'X' on the tab. You should now have an empty workspace.



Step 4: Import Libraries

1. Select the "Import Existing CCS/CCE Eclipse Project" option from the Project menu in the IDE.





2. The "Import" dialog box appears. Browse to the root directory of driver library (StellarisWare\driverlib). Be sure that "driverlib" is selected in the listed projects and click "Finish."



🕸 Import 🛛 🔀
Import Projects Select a directory to search for existing Eclipse projects.
Select root directory: C:\StellarisWare\driverlib Browse Select archive file: Browse Projects:
Image: Special Select All Image: Special Select All Image: Deselect All Image: Regresh
Copy projects into workspace
⑦ Einish Cancel

IMPORTANT: If you have installed the StellarisWare Firmware Development package for a board that does not use the USB library or Graphics library, the following steps do not apply. If the directories mentioned below do not exist in your StellarisWare installation, then cancel out of the import projects dialog.

3. Select the "Import Existing CCS/CCE Eclipse Project" option from the Project menu in the IDE again. Browse to the root directory of USB library (StellarisWare\usblib). Be sure that "usblib" is selected in the projects list and click "Finish."



🕸 Import	
Import Projects Select a directory to search for exist	ing Eclipse projects.
Select roo <u>t</u> directory: C:\Stella Select <u>a</u> rchive file: Projects:	arisWare\usblib
usblib	Select All Deselect All Refresh
<u>C</u> opy projects into workspace	
0	<u> </u>

4. Select the "Import Existing CCS/CCE Eclipse Project" option from the Project menu in the IDE again. Browse to the root directory of the graphics library (StellarisWare\grlib). Be sure that "grlib" is selected in the projects list and click "Finish."



🕸 Import		
Import Projects Select a directory to search	n for existing Eclipse projects.	
 Select root directory: Select archive file: 	C:\StellarisWare\grlib	B <u>r</u> owse
Projects:		Select All Deselect All Refresh
Copy projects into wo	rkspace	
0	Einish	Cancel

Step 5: Import Board Examples

1. Select the "Import Existing CCS/CCE Eclipse Project" option from the Project menu in the IDE.



🜍 C/C++ - Code Compo	ser Studio (Licensed)	
File Edit View Navigate	Project Target Tools Scripts Wind	low Help
: □ □ □ □ □ 0 · · · · · · · · · · · · · ·	Rebuild Active Project	Ctrl+Shift+P Alt+Shift+P
C ⊂ ⊂	Import Existing CCS/CCE Edipse P Import Legacy CCSv3.3 Project	roject
ie.⊶ <mark>⊱ grlib [Actīve - De</mark> ie.⊶⊖ usblib	Build All Rebuild All Clean Selected File(s) Build Selected File(s) Exclude File(s) from Build Active Build Configuration Build Project Build Working Set Clean Build Automatically	Ctrl+B Alt+B •
	Link Files to Active Project Add Files to Active Project Properties	

2. The "Import" dialog box appears. Browse to the root directory for your chosen board. The example screen shot below has the EK-LM3S9B90 board as the chosen board. (StellarisWare\boards\ek-lm3s9b90). Be sure that the examples are selected in the listed projects and click "Finish."

💱 Import		×
Import Projects Select a directory to search	n for existing Eclipse projects.	
 Select root directory: Select archive file: Projects: 	C: \StellarisWare \boards \ek-lm3s9b90	Browse
····♥ aes_expanded_ke ···♥ aes_set_key ···♥ bitband ···♥ blinky ···♥ enet_lwip	ey	Select All Deselect All Refresh
Copy projects into wo	'kspace	
0	Einish	Cancel

3. All of the projects now show in the Projects Explorer Window.





Step 6: Building and Debugging a Project

1. Right click on the hello project and select "Set as Active Project."



2. Click on the Project pull-down menu and select "Rebuild Active Project." The build may take a few moments. As the project builds, messages scroll by in the console window. When the build is complete, the words "Build complete for project hello" appear in the console window.



🕸 C/C++ - Code Compos	er Studio (Licensed :	29 Days Remaining)	
File Edit View Navigate	Project Target Tools	Scripts Window Help	
i 🖪 📤 🗟 🗞 🔹	🗟 Build Active Project	Ctrl+Shift+P	E 6 C/C++
	😪 Rebuild Active Proje	ct Alt+Shift+P	
C/C+++ Projects X C/C+++ Projects X C C C C C C C C C C C C C C C C C C C	Import Existing CCS, Import Legacy CCSv Build All Clean Selected File(s) Exclude File(s) from Active Build Configur Build Project Build Working Set Clean Build Automatically Link Files to Active P Add Files to Active P Properties	/CCE Eclipse Project :3.3 Project Ctrl+B Alt+B :) Build ration roject	E Outline 🛛 🖓 🖓 🖓

3. Click on the Target pull-down menu and select "Debug Active Project." Alternatively, you can simply click the "Debug Launch" icon on the toolbar.



💱 C/C++ - Code Composer Stud	dio (Licensed : 29 Days Remaining)	
File Edit View Navigate Project	Target Tools Scripts Window Help	
i 🗔 🖻 🗟 🍇 🗞 • i 🏇 •	📓 New Target Configuration] - 🌾 🔶 - 🔶 - 🗈 🖬 🖬 C/C++	
C/C L Drojecte Y	Sa Debug Active Project □ □ E Outline 😒	- 8
	An outline is not available.	
I aes_expanded_key	Debug History	
aes_expanded_key ⊕ ⊕ aes_set_key	Debug	
■ 🚰 bitband		
∎ 🖻 blinky		
🖻 🖻 driverlib		
🗎 😂 enet_lwip		
🗄 😂 enet_ptpd		
itag itag itag itag		
Hello [Active - Debug]		
🖻 🖻 mpu_fault		
🕮 😂 qs-adventure		
⊞ 🔁 timers		
udma_demo ⊕…⊖ usb_dev_bulk		
usb_dev_buik ⊞…⊖ usb_dev_keyboard		
	Console 🛛 🖓 🖓 🖓 🖓	
⊕ 🖻 usb_dev_serial		
🕮 😂 usb_host_keyboard	C-Build [hello] 0 errors, 0 warnings, 0 infos Description	
⊕ 🔁 usb_host_mouse		
🕀 🔁 usb_host_msc	exe" "C:/Program	
	Files/Texas	
tusb_suck_update	Instruments/ccsv4/utils/t	
■ 🖆 watchdog	iobj2bin/mkhex4bin.exe"	
-	Build complete for	
	project hello	
∏ [♦] /hello		

4. The Code Composer Studio debugger automatically connects to your evaluation board, programs the Flash memory, and runs to the beginning of the main() function. From here, you can examine and modify memory, program variables and processor registers, set breakpoints, step, and perform other typical debugging activities. To run the program, select "Run" from the Target pull-down menu.



🐼 Debug - hello.c - Code Comp	ooser Studio (Licensed)								
File Edit View Navigate Project		Help							
: 🖃 🗁 🗟 🗞 : 💶 🎭 😃		Ctrl+Alt+L Ctrl+Alt+R	{- <- <->				E 3	S Debug 📴 C/	(C++
🎋 Debug 🗙 🛛 🎉 🕪	Load Symbols	CUTHAILTR	· 🤣 🗸 🗖	🗱= Local (1) 💥 🛛 W	atch		😑 🔗 🖗	' 🔒 📑 🏹 '	
🖃 灖 hello [Debug] - Stellaris In-Ciri	\$ Add Symbols		Session]	Name	Value		Address	Type	
Device Device	Connect Target	Alt+C Alt+D							
Stellaris In-Circuit Debug I			1						
📲 Stellaris In-Circuit Debug I	💞 Restart 🔍 Go to Main								
	w so to Main I≥ Run	F8							
	III Halt								
	Terminate All	Ctrl+Alt+T							
	Step Into	F5	1						
	🔿 Step Over	F6		<					>
	Assembly Step Into								
l͡c hello.c ⊠	Assembly Step Over								-
65 //***********************************	RStep Return	F7	********	*********	****				
67 [©] main(void)	leset 🏀	+	1						
➡1 68 {	⇒[Run to Line	Ctrl+R							
69 //	Clock	•							
70 // Set the clo 71 //	Advanced	•	rystal.						
72 ROM SysCtlCloc	R New Target Configuration		E OSC L SY	SCTL_OSC_MAIN					
73	Note: The two project the two projects are two			'					a -1
74	🎋 Launch TI Debugger								
75 // 76 // Initialize	Debug History	•							
77 //	Debug								~
<			-					>	
🗐 Console 🔀							🗶 💥 📑 🚮 🖻	' 📮 🕶 📑 🕶 '	
hello [Project Debug Session] Stellaris I									
CORTEX_M3_0: GEL: Warni:				struments\ccsv4	\emulation\	gel\CortexM3_	util.gel', line	68: the fu	in (📥
CORTEX_M3_0: GEL Output CORTEX M3 0: GEL Output			piete						
CORTEX_M3_0: GEL Output									
									~
<)			>
i 🕈 🖬 🖻			1	.E		1			

Step 7: Build and Run Additional Example Programs

There are several additional example projects listed in the workspace. If you would like to build another example project, follow the above instructions. The quickstart application that came preloaded on the evaluation board is the qs-xxxxx project listed with the examples.

Creating a New Project

Once you have gone through the StellarisWare example applications, you may want to create your own project to start development. While you can always start with an existing, simple project, sometimes you may want to start fresh.

The example provided below creates a fresh project, copies code from an existing project, and builds the new project.

1. To add a new project to your workspace, go to File > New > CCS Project.



ile	Edit V	ew Na	vigate Pro	ject Target	Tools	Scripts	s Window Help
	New			A	lt+Shift-	HN 🕨 🕨	CCS Project
	New File.						🕅 Standard Make Project
	Open File						RTSC Configuration Project
	Close			C	trl+W		😂 Folder
	Close All			C	trl+Shift	+W	💣 Source File
Ū	Save				trl+S		💦 Header File
	Save As.						📑 File
ĥ	Save All			C	trl+Shift	+S	🞯 Class
-uau	Revert						😭 Target Configuration File
							배 DSP/BIOS v5.xx Configuration File
	Move Rename			F	-		🐴 RTSC Configuration File
	кепате Refresh			-	2 5		=\$ other
	Kerresn Convert l	ine Delin		F	5		Cther

2. Code Composer Studio prompts you with a dialog box asking you to name the project and specify the location of the project. Name your project, specify the location, and click "Next."





3. The next dialog box asks for the project type and configurations. Select "ARM" as the project type, check both the "Debug" and "Release" configurations, and click "Next."

🕸 New CCS Project	X
Select a type of project Select the platform and configurations you wish to deploy on	G
Project Type: ARM Configurations:	~
 ✓ W Debug ✓ W Release 	Select All Deselect All
Show All Project Types	
(?) < <u>Back</u> <u>Next</u> > Einish	Cancel

4. The next dialog box allows you to define any inter-project dependencies. If your project will be using driverlib, usblib, or grlib, now is a good time to define that dependency.



🕸 New CCS Project	
Additional Project Settings Define the inter-project dependencies, if any.	G
<pre> Projects C/C++ Indexer Referenced Projects Ccass_expanded_key Ccass_expanded_key Ccass_ext_key Ccass_set_key Ccass_blinky Ccass_blinky Ccass_constant Projects Ccass_constant Ccass_constant Projects Ccass_constant Projects Ccass_constant Projects Projects Ccass_constant Projects Projects Ccass_constant Projects Ccass_constant Projects Projects Projects Projects Projects Projects Projects Projects</pre>	
? < <u>Back</u> <u>Next</u> > Einish	Cancel

5. On the next dialog, select the appropriate Stellaris device, little endian, the TI code generation tool, ELF output format, and the rtsv7M3_T_le_eabi.lib runtime library. Click "Finish."



💱 New CCS Project		
CCS Project Settings Select the CCS project sett	ngs.	G
Output type: Executable		~
Project settings		
Device Variant:	Cortex M 💉 Stellaris LM3S9B90	More
Device Endianness:	little	~
Code Generation tools:	TI v4.6.1	✓ More
Output Format:	eabi (ELF)	~
Linker Command File:	lm3s9b90.cmd	Browse
Runtime Support Library:	rtsv7M3_T_le_eabi.lib	Browse
Treat as an Assembly-	only project	
Target content None		
O Use DSP/BIOS v5.xx		<u>M</u> ore
O Enable RTSC support (equired for DSP/BIOS v6.xx, Codec Engine v3.xx, e	etc.)
0	< <u>B</u> ack <u>N</u> ext > <u>Finish</u>	Cancel

- 6. A new project has now been created, but the project does not include any source code. The next step is to add some startup code to the project. Copy the startup_ccs.c file from an existing example directory in StellarisWare (such as \StellarisWare\boards\<board>\blinky) to your new project directory created in step 2 above.
- 7. Create a new C source file by going to File > New > Source File.



e Edit View Navig	gate Project Target Too	ls Scripts Window Help
New	Alt+Sh	ift+N 🔸 🔂 CCS Project
New File		🛤 Standard Make Project
Open File		RTSC Configuration Project
Close	Ctrl+W	😂 Folder
Close All	Ctrl+S	nift+W 💣 Source File
Save	Ctrl+S	h Header File
Save As		📑 File
Save All	Ctrl+S	nift+S GClass
Revert		😭 Target Configuration File
		배송 DSP/BIOS v5.xx Configuration File
Move		🐴 RTSC Configuration File
Rename	F2	

8. In the dialog box that pops up, name the file main.c and click "Finish."

🕸 New Source File		×
Create a new so	ource file.	C
Source <u>F</u> older:		Browse
Source File:	main.c	
0	Einish	Cancel



9. Add a main() function to main.c. Save the file. See the screenshot below.



10. Both files, main.c and startup_ccs.c, should have automatically been added to the project. If not, right-click the project, select "Add Files to Project" and browse to the files.



11. Build the new project by selecting Project > Rebuild Active Project.

With your project created, all you really need to do is add your own code. Use the existing StellarisWare board examples as a reference.

NOTE: To set up your project to output a binary file (.bin) to be used with tools such as LM Flash Programmer, copy the post-build step from an existing StellarisWare example project. See the following screenshot.



Properties for mpu_f	ault
type filter text	$C/C++$ Build $\Leftrightarrow - \Leftrightarrow +$
 Info Builders C/C++ Build C/C++ Documentation C/C++ Hie Types C/C++ Indexer CCS Debug Project References Refactoring History 	Active configuration Project Type: ARM Configuration: Debug Configuration: Manage Configuration: Manage Configuration: Manage Configuration: Manage Configuration: Manage Pre-build Stetp: Command: Command: Description: [*s[CCE_TINSTALL_ROOT)/utils/tobj2bin/bobj2bin/bob; 2bin.bet" *s[BuildArtifactFileName)."s[CuildArtifactFileBaseName].bin" *s[CG_TOOL_ROOT)/bin/ofd470.exe" *s[CCE_TOSL_ROOT)/bin/ofd470.exe" *s[CCE_TOSL_ROOT]/bin/ofd470.exe" *s[CCE_TOSL_ROOT]/bin/ofd470.exe" *s[CCE_TOSL_ROOT]/bin/ofd470.exe" *s[CCE_TOSL_ROOT]/bin/ofd470.exe" *s[CCE_TINSTALL_ROOT]/utils/tobj2bin/bob; 2bin.bet" *s[BuildArtifactFileName].bin" *s[CCE_TOOL_ROOT]/bin/ofd470.exe" *s[CCE_TINSTALL_ROOT]/bin/ofd470.exe" *s[CCE_TINSTALL_ROOT]/bin/of
0	OK Cancel

NOTE: If you use standard C I/O functions such as printf or fread, you must increase the stack and heap allocations for your project. Right-click on your project and then select Properties to open a dialog containing the project options. Select C/C++ Build in the pane on the left side of the dialog box. This displays the build options. Under the Linker node in the Tool Settings tab, select Basic Options from the scroll list. Enter 0x800 for both the system stack and the heap. See the following screenshot.





Conclusion

You have now installed the Code Composer Studio development tools and used them to build and load an example application on your Stellaris Evaluation Board. From here, you can experiment with the debugger or start creating your own application using the example projects as a reference. For further information on Code Composer Studio, go to the CCS Developer Site.





References

The following references are included on the Stellaris Evaluation Kit Documentation and Software DVD and are also available for download at <u>www.ti.com/stellaris</u>:

- Stellaris Evaluation Kit User's Manual
- StellarisWare Software, Order Number SW-LM3S
- StellarisWare Peripheral Driver Library User's Guide, Order Number SW-DRL-UG

In addition, the following website may be useful:

Code Composer Studio website at <u>http://www.ti.com/ccstudio</u>

Copyright © 2009–2010 Texas Instruments, Inc. All rights reserved. Stellaris and StellarisWare are registered trademarks of Texas Instruments. ARM and Thumb are registered trademarks, and Cortex is a trademark of ARM Limited. Other names and brands may be claimed as the property of others.

Texas Instruments 108 Wild Basin Rd., Suite 350 Austin, TX 78746 http://www.ti.com/stellaris





