

Tiva[™] C Series Development and Evaluation Kits for Sourcery CodeBench[™] Development Tools

This Quickstart provides instructions to install the Sourcery CodeBench, flash and debug an application using the Sourcery CodeBench IDE and GDB, and compile TivaWare[™] for C Series. The Tiva C Series development and evaluation boards can function as either a complete evaluation target with minor modifications or as a debugger interface to any external Tiva C Series device.

1 Requirements

- PC with a USB interface and Microsoft® Windows 2000, XP, Vista, or 7 Operating System
- Tiva C Series Development Kit Software USB flash drive or the downloaded and extracted kit software (found on www.ti.com/tool/sw-<kit_name>)
- ICDI drivers installed following the instructions in Stellaris[®] In-Circuit Debug Interface (ICDI) and Virtual COM Port (SPMU287).
- Installed command-line environment, if using a command line interface. The environment should include make and the other associated utilities described in the TivaWare User's Guide.

2 Sourcery CodeBench

This quickstart provides instructions to install the Sourcery CodeBench tools and how to build and run an example application on a Tiva Development or Evaluation board.

NOTE: The GDB stub included with Sourcery CodeBench is a 30-day time-limited evaluation version. For long-term use, you must purchase a license from CodeSourcery.

Tiva, TivaWare are trademarks of Texas Instruments. Stellaris is a registered trademark of Texas Instruments. Sourcery CodeBench is a trademark of Mentor Graphics. Microsoft is a registered trademark of Microsoft. All other trademarks are the property of their respective owners.

1

Copyright © 2013, Texas Instruments Incorporated



Sourcery CodeBench

2.1 Installation

2.1.1 Install Sourcery CodeBench

- Insert the TIva C Series Development Kit Software USB flash drive and click on the index.html. For kits that don't include a USB flash drive, go to the evaluation board software webpage (www.ti.com/tools/sw-<kit_name>)
- 2. With the Development Kit USB flash drive, click the CodeSourcery logo to start the setup program. Depending on your web browser, either run the installer directly from the USB flash drive or first download it to your PC and then run the installer.

NOTE: If installing CodeBench manually, open the sourceryg++x.x-xx-arm-stellaris-eabi.exe file in the *Tools\CodeSourcery* directory of the USB flash drive.

- 3. Follow the instructions detailed in the installer until the installation is complete.
- Register with CodeSourcery to obtain an evaluation license once installation is complete. To obtain a license, start the Sourcery CodeBench integrated development environment (IDE). The Sourcery G++ for Tiva EABI License Installation window appears. (see Figure 1)

Sourcery G++ for Stellaris EABI Licensing	
Sourcery G++ for Stellaris EABI License Installation	5
No valid license for Sourcery G++ for Stellaris EABI was found. The error was: Can't Find License File (-1,4042) (C:\Program Files\CodeSourcery\Sourcery G++ 4.3-186\license\arm-stellari	is-eabi)
Your Host IDs(s) are: DISK_SERIAL_NUM= <serial #="" disk="" of="" your=""> <mac adapter(s)="" address(es)="" network="" of="" your=""></mac></serial>	
You must install a license to use Sourcery G++. Please select a license installation method: Obtain a license automatically	
Install a license file you already have If you cancel this wizard, you will be prompted again to install a license the next time you start the Sourcery G-	++ IDE.
(?) < Back	ancel

Figure 1.

- 5. If you already have a license for the Sourcery CodeBench tools, select "Install a license file you already have," click "Next" and follow the instructions. If you do not have a license file, you must visit the Sourcery CodeBench web site to register and request an evaluation license key. You can reach this web site by making sure that "Obtain a license automatically" is selected before click "Next" twice then clicking the "Sourcer G++ Portal" link on the "Obtain License" screen.
- 6. Once you have obtained a license key, use the "Back" button to return to the screen shown above, then select "Install a license file you already have" and proceed as directed.

If you encounter any difficulties during the Sourcery CodeBench installation procedure, or would like to learn more about the Sourcery CodeBench IDE, further documentation on the subject can be found in the CodeSourcery installation directory at

\CodeSourcery\Sourcery_CodeBench_for_Stellaris_EABI\share\doc\sourceryg++-arm-stellaris-eabi. Chapter two of CodeSourcery's getting started guide goes into further detail concerning the installation and configuration process.

2.1.2 Install the TivaWare Package

A full set of C-based peripheral drivers is provided, covering all peripherals and functions of the Tiva devices. The TivaWare package includes various example applications with project files for all major tool vendors that support Tiva C Series, including CodeSourcery.

To install TivaWare, follow these steps:

1. On the html index of the Development Kit Software USB flash drive, click on the Firmware Development Package. Depending on your web browser, you have the option to run the TivaWare installer or save it to your drive.

- 2. Run the TivaWare installer. To manually install TivaWare, use a .zip file extraction utility, such as WinZip, to manually extract the contents of the .zip file in the *TivaWare* directory.
- 3. Navigate to the Tools/TivaWare/docs directory and click the *TivaWare Peripheral Driver Library User's Guide* PDF file to view the TivaWare documentation.

2.2 Build and Debug the Software

There are two methods of building and debugging software in the Sourcery CodeBench environment: a command line interface and an Eclipse-based Integrated Development Environment (IDE). To use the command line interface, follow these steps after installing Sourcery CodeBench and TivaWare. To use the IDE interface, see Building using the Integrated Development Environment.

2.2.1 Build Using the Command Line

After installing Sourcery CodeBench and TivaWare, build and debug the software from a command line using the following steps.

2.2.1.1 Build the Driver Library With GCC

1. Use your command-line utility (shell) to navigate to the TivaWare directory. Assume the TivaWare directory was extracted to *C:\ti\TivaWare_C_Series-n.n.*

MYPC# cd C:/TivaWare

2. Type make to build the TivaWare and software examples:

MYPC# make

3. The compiler completes the build process for each file and then builds the output files (*.axf and *.bin).

2.2.1.2 Run GDB

The GDB stub is in the *CodeSourcery\Sourcery* G++\bin directory, which is installed to C:\Program Files by default.

In this example, the *Hello* example from *ti\TivaWare_C_Series-n.n\examples\boards\ek-tm4c123gxl* is loaded into flash, although there is a *Hello* example for all Tiva boards.

1. Open a command-line utility (shell) window and navigate to the *ek-tm4c123gxl\hello\gcc* directory of TivaWare to run GDB. Assume TivaWare is at *C:\ti\TivaWare_C_Series-1.0*.

MYPC# cd C:/ti/TivaWare_C_Series-1.0/examples/boards/ek-tm4c123gx1/hello/gcc

2. Type **arm-stellaris-eabi-gdb hello.axf** and press Enter. Assuming you have already built the example

NOTE: If navigating the USB flash drive with Windows Explorer or a similar application, navigate to the *TivaWare* directory.



as described in Section 2.2.1, you should see the following:

MYPC# arm-stellaris-eabi-gdb hello.axf GNU gdb (Sourcery G++ x.x-xx) x.x.xx.xxxxxx-cvs Copyright (C) 2007 Free Software Foundation, Inc. GDB is free software, covered by the GNU General Public License, and you are welcome to change it and/or distribute copies of it under certain conditions. Type "show copying" to see the conditions. There is absolutely no warranty for GDB. Type "show warranty" for details. This GDB was configured as "--host=i686-mingw32 --target=arm-stellaris-eabi".

GDB is now running.

 Type target extended-remote | arm-stellaris-eabi-sprite armusb: ?speed=2 <target> to connect to the board, where <target> should be replaced with the Tiva device on the board. For this example, <target> is replaced with ek-tm4c123gxl. Find the full list of supported target strings in the Sourcery CodeBench installation directory at CodeSourcery\share\doc\sourceryg++-arm-stellariseabi\html\getting-started\sec\sprite\supported-boards.html. Press Enter and you should see the following:

(gdb) target extended-remote | arm-stellaris-eabi-sprite armusb:?speed=2 tm4cl23gxl Remote debugging using | arm-stellaris-eabi-sprite armusb:?speed=2 tm4cl23gxl

4. Type load to download hello.axf to the flash and press Enter. The following should appear:

(gdb) load Loading section .text, size 0xxxx lma 0x0 Loading section .data, size 0xxxx lma 0xxxxx Start address 0xxxx, load size xxxx Transfer rate: xxxx KB/sec, xxx bytes/write.

5. Type **continue** to restart the application. The words, "Hello World!" should appear on the PC via the UART or to the display once execution of the application starts.

```
(gdb) continue
Continuing.
```

If you encounter difficulties when working with the Sourcery CodeBench command line debugger, find further documentation in the Sourcery CodeBench installation directory at *CodeSourcery\share\doc\sourceryg++-arm-stellaris-eabi.* Section 5.3 and Chapter 7 of Sourcery CodeBench's Getting Started Guide contains more information on running applications through GDB.

2.2.1.3 Simplify GDB Commands

Simplify commands sent to the GDB by defining custom commands in a .gdbinit file and placing it in your \$HOME directory. To find the location of the #HOME directory, type **echo \$HOME** in a shell window. For Windows users, the directory is at *C*:\Documents and Settings\<user name> by default.

An example of a command to include in your .gdbinit file is *connect*. This command replaces the complex command string described in Section 2.2.1.2, item 3. For example, after starting GDB, typing **connect** has the same effect as typing **target extended-remote** | **arm-stellaris-eabi-sprite** –**s 2** –**f Imi.dll stdio**.

2.2.2 Build Using the Integrated Development Environment

Perform the steps in the following subsections to use the IDE. To use the command line interface, see Building using the command line.

2.2.2.1 Start the Sourcery CodeBench IDE

To start the IDE, use the shortcut found under *Start/All Programs/CodeSourcery/Sourcery CodeBench for Stellaris EABI/Sourcery CodeBench IDE* to open a dialog box (see Figure 2). Enter the path where Sourcery CodeBench will store the source files of the project.



6 Workspace	e Launcher		×
Select a w	orkspace		
	deBench IDE stores your projects in a folder called a workspace. orkspace folder to use for this session.		
Workspace:	C:\Users\ <userid>\workspace</userid>	•	Browse
🔲 Use this a	s the default and do not ask again	ОК	Cancel

Figure 2.

If you check the Use this as the default and do not ask again checkbox, this workspace opens each time you start the IDE.

Click OK to proceed. If the License Installation screen appears, obtain a license for the toolchain by following the instructions given in Section 2.1.1.

2.2.2.2 Import the TivaWare or TivaWare Release for Your Board

- 1. Select File/Import... from the Sourcery CodeBench IDE menu.
- 2. Highlight StellarisWare or TivaWare Project in the Import dialog (see Figure 3).



5 Import	
Select Import StellarisWare Project	2J
Select an import source:	
type filter text	
 General C/C++ CVS Remote Systems Run/Debug Sourcery CodeBench C/C++ Executable Kinetis Project Sourcery CodeBench Example Project StellarisWare Project Tasks Team 	
Rext > Fi	nish Cancel

Figure 3.

- Click Next and browse to the directory containing the TivaWare release for the board. If TivaWare is
 installed in the default directory, this directory appears under C:\ti\TivaWare_C_Seriesn.n\examples\\boards\<board name>.
- 4. Select the <board>.sgxw file and click Open (see Figure 4). The following example assumes that you are importing the ek--tm4c123gxl examples, but the same procedure applies to all TivaWare releases.

TEXAS INSTRUMENTS

www.ti.com

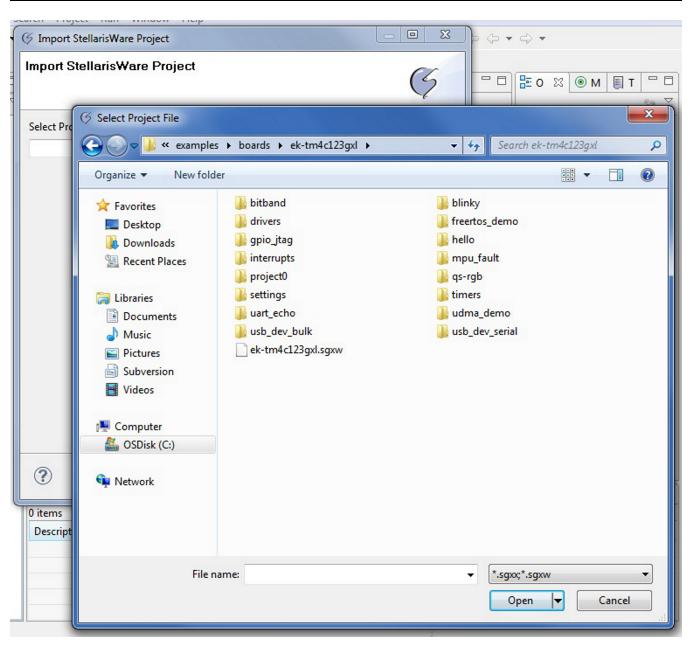


Figure 4.

Press the Finish button to start importing the project files into the workspace. When the files are imported, the IDE builds each library and example application.



2.2.2.3 Flash and Debug an Example Application

After the IDE builds the imported examples, download the examples to the target board for debugging.

1. Point to the example application you wish to download and right-click to show a context menu (see Figure 5). From this menu select Debug As... then select C/C++ Application from the menu that appears. The application is downloaded to the board and is ready for debugging.

(C/C++ - Sourcery CodeBe	ench	IDE				
File Edit Source Refact	tor	Navigate Search Project Run Wi				
	1	* • 🚳 • 🔂 • 🞯 • 🕺 • 🗞 • 📎		🎍 🕶 😂 🌮 🗸 🗐 🗊 🖢 🖛 🖗	- *	↓ ↓ ↓
😭 🛅 C/C++						
Project Explorer						Ео ¤ ⊛ м ∎ т □ □
	e					
driverlib						An outline is not available.
Anvenib B B ek-tm4c123gxl - bitb	hand					An outline is not available.
ek-tm4c123gxl - blin						
⊳ 😂 ek-tm4c123gxl - free		demo				
b 😂 ek-tm4c123gxl - gpic	-	g				
	-	New	•			
▷ 😂 ek-tm4c123gxl - ir ▷ 😂 ek-tm4c123gxl - m		Go Into				
ek-tm4c123gxl - n						
▷ B ek-tm4c123gxl - q		Open in New Window				
⊳ 😂 ek-tm4c123gxl - ti	D	Сору	Ctrl+C			
▷ 😂 ek-tm4c123gxl - u	1	Paste	Ctrl+V			
▷ 😂 ek-tm4c123gxl - u ▷ 😂 ek-tm4c123gxl - u	×	Delete	Delete			
, <u> </u>	3	Remove from Context Ctrl	+ Alt+ Shift+ Down			
▷ 😂 ek-tm4c123gxl - u ▷ 😂 inc		Move				
b 🗁 third_party		Rename	F2			
b 🚰 usblib						
		Import				
E	4	Export				
		Build Project				
		Clean Project				
4	8	Refresh	F5			
		Close Project				
		Close Unrelated Projects				
		Build Configurations				
		Make Targets				
		Index	•			
		Show in Remote Systems view				
		Convert To				
		Run As	•	Description (
		Debug As	*	C 1 Local C/C++ Application		e
		Profile As	•	Debug Configurations	Locat	ion Type
		Team	▶ L	- song consignmentation		· / F -
		Compare With	*			
		Restore from Local History				
		Properties	Alt+Enter			
[[♦]	2			Duildia	aworken	ace: (52%)
🕴 💽 ek-tm4c12	3gxl	- nello		Buildi	ig worksp	······································

Figure 5.

2. If a dialog box asks whether you want to switch to the debug perspective (or view), select Yes and, optionally, check the box indicating you wish to switch automatically in the future. When the image download is complete, the screen similar to the one shown in Figure 6 should appear.



G Debug - ek-tm4c123gxl - hello/examples/boards/ek-tm4c123gxl/hello/hello.c - Sourcery C	odeBench IDE			
File Edit Source Refactor Navigate Search Project Run Window Help				
📑 • 🗑 🖻 📾 🕸 • 🛛 • 隆 😕 😂 🖋 • 🍠 🥹 🖞 • 🥻] • ♥ (> • o	≎.▼		
☆ Debug 昭 C/C++				
🎋 Debug 🛛 🥂 🎉 🗱 🕼 🕪 💷 🔳 🚴 🐟 🚓 🔯 😒 🍸 🖓 🗖	(x)= Variables 🖾	💁 Breakpoints り Reg	jisters 🛋 Module	s 🗖 🗖
6 ek-tm4c123gxl - hello [Sourcery CodeBench Debug]				₺ 🕫 🖻 🥤 🗶 🖄 🏹
C:\Users\a0272201\workspace\ek-tm4c123gxl - hello\Release\hello.axf [i1]	Name	Туре		Value
main() at c:\\users\\a0272201\\workspace\\ek-tm4c123gxl - hello\\exampl				
j⊌j gdb j∰ hello.axf				
s reioan				
4 H	4			+
li hello.c ⊠			🗆 📴 Outline 🖾	
1/		•		🍃 ‡ªz 😿 🍾 ● 💥 ▽
//************************************	*********	******	📕 stdir	
main (void)				ool.h
				hw_memmap.h hw_types.h
<pre>//volatile uint32_t ui32Loop;</pre>				erlib/debug.h
11				erlib/fpu.h
<pre>// Enable lazy stacking for interrupt handlers. This // instructions to be used within interrupt handlers,</pre>				erlib/gpio.h erlib/pin_map.h
<pre>// instructions to be used within interrupt mandrers, // extra stack usage.</pre>	but at the c.	E	📲 drive	erlib/rom.h
<pre>// ROM FPULazyStackingEnable();</pre>				erlib/sysctl.h erlib/uart.h
KOM_FFOLAZYStackingEnable();				/uartstdio.h
11				ror_(char*, uint32_t) : void
<pre>// Set the clocking to run directly from the crystal. //</pre>		-		figureUART(void) : void n(void) : int
		•	• mai	n(vold) : Inc
📮 Console 🖾 🖉 Tasks 🖹 Problems 🕥 Executables 🏮 Memory		= ×	🔌 📄 🔒 🕞	🖉 🛃 - 📑 - 🗆
ek-tm4c123gxl - hello [Sourcery CodeBench Debug] hello.axf				
				*
1				T
<	1			*

Figure 6.

3. Use the icons on the top edge of the debug pane (the top-left of the window in this example) to start and stop program execution.

When execution completes or you terminate the debug session using the stop icon (the red square), you can switch back to the editor perspective by selecting Window/Open Perspective.../C/C__... from the IDE menu.

3 Create a New Project

When you have completed the TivaWare example applications, you may want to create your own project to start development. Create a new project by following these steps:

1. To add a new project to your workspace (assuming you're still using the example described above), go to *File* → *New* → *Project.* (see Figure 7).



File	Edit Source Refactor Navig	gate Search Project	Run	Window Help	
	New	Alt+Shift+N ►	C 3	Project	11 · 10 @
	Open File		F9	Other Ctrl+N	
	Close	Ctrl+W		- it 10	(×)= Variables
ě	Close All	Ctrl+Shift+W	:h D	ebug]	Name
	Save	Ctrl+S			
	Save As		ι.		
0	Save All	Ctrl+Shift+S	ι.		
	Revert				-
	Move				
	Rename	F2			
8	Refresh	F5			
	Convert Line Delimiters To	•			
	Print	Ctrl+P			4
	Switch Workspace				



2. The tool prompts you to select a wizard for creating the new project. Select C Project.

Create a New Project



www.ti.com

(New Project		- - X
Select a wi Create a new		
Wizards:		
type filter tex	t	
C the		
?	< Back Next >	Finish Cancel

Figure 8.

3. The tool prompts you with a dialog box that asks for the desired project type (see Figure 9). For this example, focus on the executable project types. In the Executable menu, select Empty Project and name it.



Create a New Project

www.ti.com

C Project	
Project Create C project of selected type	
Project name: my_project	
Location: C:\Users\a0272201\workspac Choose file system: default Project type:	Toolchains:
 Executable Empty Project Factorial C Project Shared Library Static Library Makefile project 	Sourcery CodeBench for Stellaris EABI
Show project types and toolchains or	nly if they are supported on the platform
< Back	Next > Finish Cancel



- 4. Click the Next button
- 5. The Sourcery G++ Configuration dialog box appears. From this dialog box choose the device used for the project (see Figure 10). Choose the device from the Board menu, or keep the default Tiva option.



5 C Projec								
Sourcery Specify th				iect prop	perties bel	ow		
Board	Stellaris	EKC-LM4	IF232					•
Profile	RAM							•
Hosting	Hosted							•
Processor	Stellaris	LM4F232	H5QD					•

The created project will be empty, so you must complete some additional steps including the following:

- 1. Add source files
- 2. Set up include paths
- 3. Set up the debug interface

4 Add Source Files

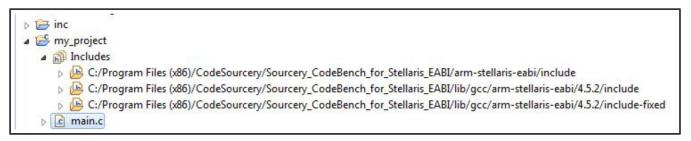
For a basic project, you'll need to create a main function. In this example, create a main file using the Sourcery CodeBench templates. To create the main file, right click on the project and select $New \rightarrow File$ *From Template*. In the dialog box that appears, type **main.c** in the File Name field, and then click the Finish button (see Figure 11). Ensure the correct project is selected in the project browser, otherwise the file will not be added to your project. Notice that the file is added to *my_project*.



🧭 New File	
File Create a new file resource from a template.	
Enter or select the parent folder:	
my_project	ji.
 Image: Second Second	E
File name: main.c Use template: Default C source template Advanced >>	 Configure
•	Finish Cancel



The project should look similar to Figure 12:





In the main.c file, add a simple main function.

5 Add Include Paths, Libraries, and Output

The tool has many options, but the basic set up must include the hooks to TivaWare, primarily the Driver Library (driverlib). To get to the project options dialog, right click the project and select Properties. Expand the C/C++ Build menu and select the Settings item (see Figure 13).

First, add the appropriate include paths. In the Tool Settings tab click on the Includes item, located inside the Sourcery CodeBench C Compiler section. To the right, there is a box that says Include paths. To add a new path, click the small icon with the green +.

G Properties for my_project			
type filter text	Settings		⇔・⇔・・
 Resource Builders C/C++ Build Build Variables Discovery Options Environment Logging Settings Tool Chain Editor C/C++ General Project References 	Configuration: Debug [Active]	I Artifact 🕢 Binary Parsers 🧿 Error Parsers	 ▼ Manage Configurations ● 行 公
Run/Debug Settings ▷ Task Repository WikiText	 Includes Optimization Debugging Warnings Warnings Sourcery CodeBench Assembler Sourcery CodeBench C Linker General Libraries 		
	Miscellaneous Shared Library Settings Sourcery CodeBench Objcopy General	Include files (-include)	
			Restore Defaults Apply
?			OK Cancel

Figure 13.

A window appears, allowing you to browse for the new path (see Figure 14). Click the Workspace... button to browse through your existing workspace. When the workspace browser appears, expand the *driverlib* project and select the Release folder.



х 6 Folder selection Select one or more Workspace Folders a 😂 driverlib ä 👂 🗁 driverlib Release ▷ 🥵 ek-tm4c123gxl - bitband E ek-tm4c123gxl - blinky b 😂 ek-tm4c123gxl - freertos_demo 8 E ek-tm4c123gxl - gpio_jtag k-tm4c123gxl - hello ek-tm4c123qxl - interrupts b 😂 ek-tm4c123gxl - mpu_fault

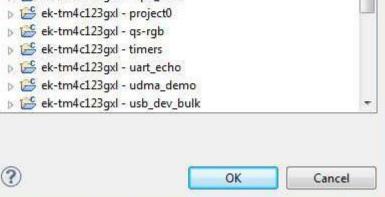
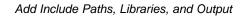


Figure 14.

Repeat this process three more times, adding the *inc* project, the *utils* project and your working project (in this case *my_project*) to the include paths (see Figure 15).





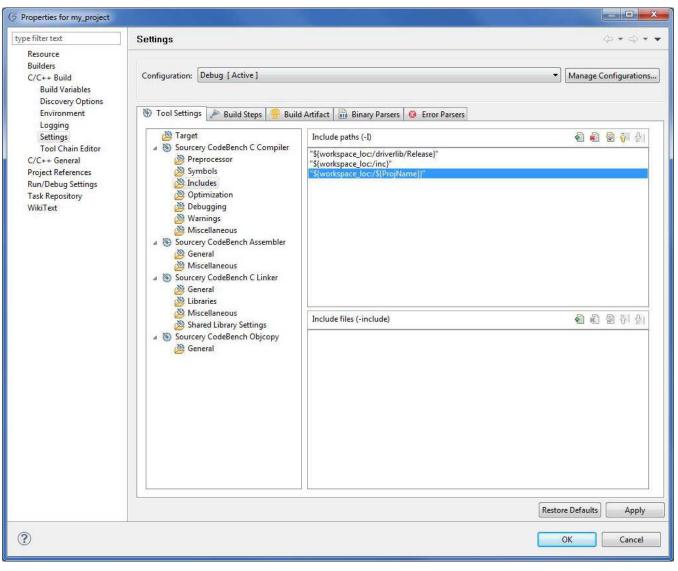


Figure 15.

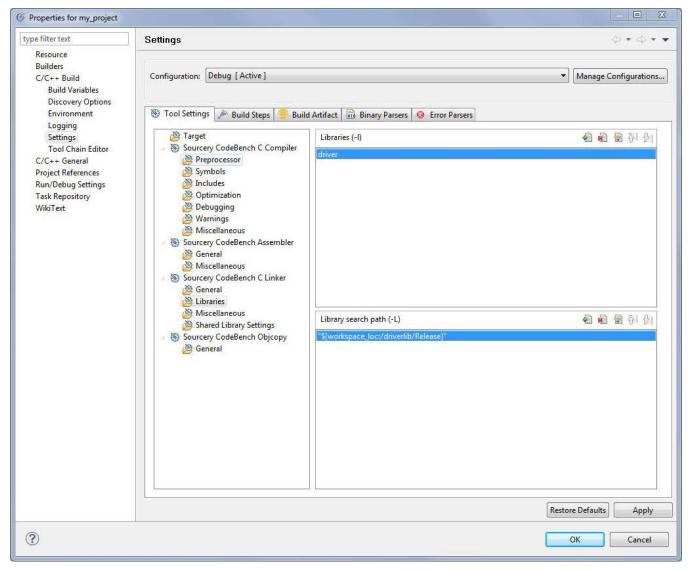
Next, add a pointer to the TivaWare Driver Library. Select the Libraries item under Sourcery CodeBench C Linker. Add a new item by clicking the little icon with the green +. In the Libraries field, type the name of the Driver Library to match the project that was included in the previous step. For this example, the Driver Library project was named *driverlib*, so type **driver** in the Libraries field (see Figure 16).



ibraries (-I)		
driver		



In the Library search path, point the tool to the driverlib\Release directory as done for the C include paths (see Figure 17).





Tell the tool how to format the output executable. The default configuration generates an output file in ELF format without an extension. To change it to an AXF file, go to the Build Artifact tab and type **axf** in the Artifact extension box (see Figure 18).

G Properties for my_project						
type filter text	Settings		↓ • ↓ • ↓			
Resource Builders C/C++ Build Build Variables Discovery Options	Configuration: De	bug [Active]	Manage Configurations			
Environment	🛞 Tool Settings 🎤 Build Steps 😤 Build Artifact 📓 Binary Parsers 👩 Error Parsers					
Logging Settings	Artifact Type:	Executable	*			
Tool Chain Editor C/C++ General	Artifact name:	\${ProjName}	-			
Project References	Artifact extension:	axf	•			
Run/Debug Settings Task Repository WikiText	Output prefix:					



Now you can build your project.

6 Add a Debug Configuration

Before debugging the application, create a debug configuration.

1. Right-click the project and select *Debug* $As \rightarrow Debug$ *Configurations...* (see Figure 19).

Show in Remote Systems view Convert To	
Run As	
Debug As	1 Local C/C++ Application
Profile As	Debug Configurations
Team	• ¹

Figure 19.

2. In the dialog box that appears, double-click on the Sourcery CodeBench Debug item, and the tool creates a new debug configuration called my_project Debug (see Figure 20).



Conclusion

Create, manage, and run c	onfigurations			Ś
Image: Second Secon	Name: my_project Debug Main Main C/C++ Arguments Debug/my_project.axf Search Project Brow Brow Project: my_project my_project Brow Build (if required) before launching Brow Build configuration: Use Active Image: Disable auto build Image: Disable auto build Image: Use workspace settings			
	Connect process input & output to a		Apply	Revert
			Debug	Close

Figure 20.

3. Click Debug after applying the changes. See Section 2.2.2.3 for more information on using the debugger.

7 Conclusion

You have now installed GCC, built the TivaWare package, and flashed the device using either GDB or the Sourcery CodeBench IDE. To learn more about using GDB commands, type **help**. You will find a list of expandable help topics. For example, typing **help data** lists the commands available for doing data operations.



8 References

The following references are included on the Tiva C Series Development Kit Software USB flash drive and are also available for download at http://www.ti.com/tiva-c:

References

- Tiva C Series Evaluation Kit User's Manual
- TivaWare for C Series Software, Order Number SW-TM4C (http://www.ti.com/tool/sw-tm4c)
- TivaWare Peripheral Driver Library User's Guide, Document Number SW-DRL-UG (literature <u>SPMU298</u>)

In addition, the following website may be useful:

Mentor Graphics Sourcery Code Bench website at http://www.mentor.com/embedded-software/sourcery-tools

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have *not* been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		Applications	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	TI E2E Community	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconne	ectivity	

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2013, Texas Instruments Incorporated