

Stellaris® Development and Evaluation Kits for Sourcery CodeBench™ Development Tools

This Quickstart shows you how to install the Sourcery CodeBench™, compile StellarisWare®, and flash/debug an application using the Sourcery CodeBench IDE and GDB. The Stellaris Development and 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 2000, XP, Vista, or Windows 7.
- You have the Stellaris Evaluation Kit Documentation and Software CD or the standalone Sourcery CodeBench CD found in the Development Kit.
- If you are using a command line interface, you must have an installed command-line environment that includes make and the other associated utilities described in the *StellarisWare User's Guide*. The UnxUtils package is included on the kit CD.



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.

Sourcery CodeBench™

This quickstart shows you how to install the Sourcery CodeBench tools and how to build and run an example application on your Stellaris 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.

Step 1: Install UnxUtils (if necessary)

If you want to perform builds using the command line rather than the Integrated Development Environment and your PC does not already have a Unix-like command line available, you can install the UnxUtils package found on the Evaluation Kit Documentation and Software CD.

1. On the Evaluation Kit Documentation and Software CD, navigate to the *Tools* tab and click the *UnxUtils* link in the center of the page. On the Development Kit CD, this is located on the *Software* tab.

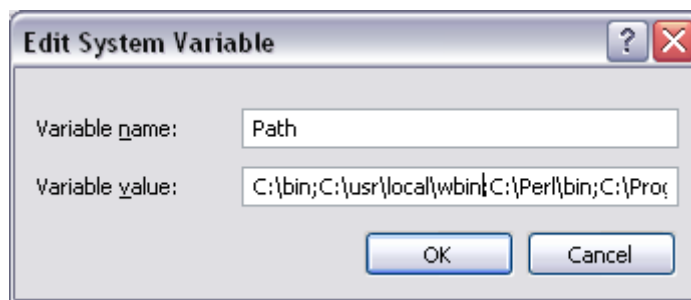
NOTE: If you are navigating the CD using Windows Explorer (or a similar application), go to the Tools\UnxUtils directory.



QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Download the .zip file to your PC and extract the contents to your “C:\” drive. You now have a “bin” and “usr” directory under “C:\”.

2. Make sure that you have “C:\usr\local\wbin” and “C:\bin” in your path. To do this, right-click on My Computer and select Properties. Select the *Advanced* tab and click the Environment Variables button. Scroll down in the *System variables* box to see the Path variable. Double-click on *Path* and add “C:\bin; C:\usr\local\wbin;” to the *beginning* of the string.



3. Open a command-prompt (shell) window by clicking on the sh.exe file located in “C:\bin.” Type “which make” to verify that Windows is looking at the correct version of make. You should see this:

```
MYPC# which make
c:/usr/local/wbin/make
```

NOTE: It is a good idea to create a shortcut to sh.exe and place it somewhere convenient (like your desktop).

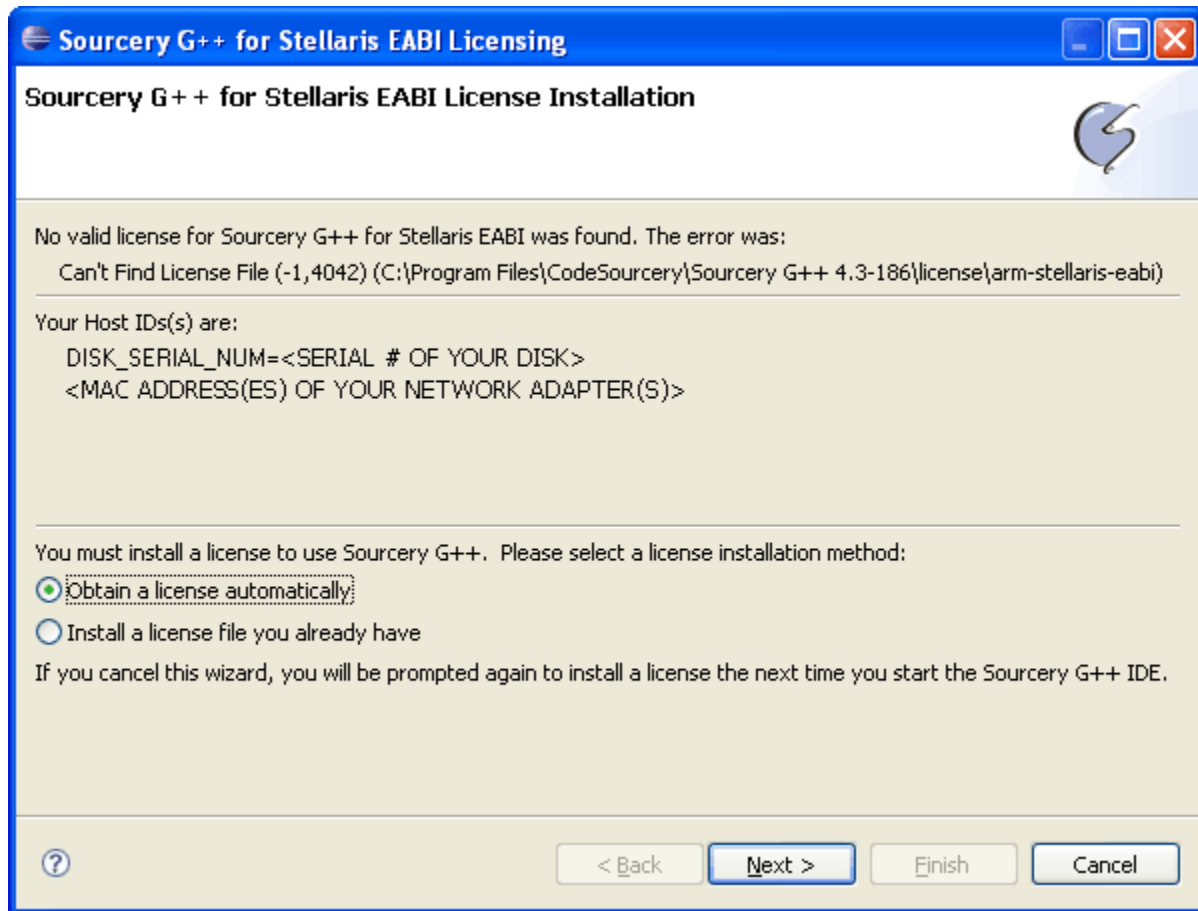
Step 2: Install Sourcery CodeBench

1. Insert the Evaluation Kit Documentation and Software CD into the CD-ROM drive of your PC. Navigate to the *Tools* section of the CD and click on the CodeSourcery logo to run the installer. Depending on your web browser, you may be able to run the installer directly from the CD or you may have to download it to your PC first.

NOTE: If you are installing manually, click on the sourceryg++-4.x-xx-arm-stellaris-eabi.exe file in the “Tools\CodeSourcery” directory of the CD.

2. Follow the instructions detailed in the installer until it indicates that the installation is complete.
3. Once installation is complete, you must register with CodeSourcery to obtain an evaluation license. To do so, start the Sourcery CodeBench IDE. You will be prompted with the Sourcery CodeBench License Installation window.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS



4. 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 CodeSourcery 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 clicking “Next” twice then clicking the “Sourcery G++ Portal” link on the “Obtain License” screen.
5. 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\share\doc\sourceryg++-arm-stellaris-eabi`. Chapter two of CodeSourcery’s getting started guide goes into further detail concerning the installation and configuration process.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Step 3: 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 CodeSourcery. To install StellarisWare, follow these steps:

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

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

2. Click on the 'Install' link next in the StellarisWare section (under Tools) of the CD 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 the *StellarisWare User's Guide* PDF file.

Building and Debugging the Software

There are two ways to build and debug 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 StellarisWare. To use the IDE interface, see [Building using the Integrated Development Environment on page 6](#).

Building using the command line

After installing UnxUtils, Sourcery CodeBench, and StellarisWare, you may choose to build and debug the software from a command line by following these steps.

Step 1: Building the Driver Library with GCC

For all examples, the UnxUtils package is used as the command-line interface.

1. Using your command-line utility (shell), navigate to the StellarisWare directory. For this example, assume that StellarisWare was extracted to C:\StellarisWare.

```
MYPC# cd C:/StellarisWare
```

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

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

```
MYPC# make
```

3. You see the compiler going through the build process for each file and build the output files (*.axf and *.bin).

Step 2: Running GDB

The GDB stub is installed with the Sourcery CodeBench tools and is located in the “CodeSourcery\Sourcery G++\bin” directory (which is installed to C:\Program Files by default).

In this example, the “Hello” example from “StellarisWare\ev-lm3s811\hello” is loaded into flash, although there is a “Hello” example for all Stellaris boards.

1. To run GDB, open a shell window and navigate to the ev-lm3s811\hello\gcc directory of StellarisWare. Again, assume that StellarisWare is located at C:\StellarisWare.

```
MYPC# cd C:/StellarisWare/boards/ev-lm3s811/hello/gcc
```

2. Now, type arm-stellaris-eabi-gdb hello.axf (this assumes that you have already built the examples as described in Step 4). After pressing Enter, you see:

```
MYPC# arm-stellaris-eabi-gdb hello.axf
GNU gdb (Sourcery G++ x.x-xx) x.x.xx.xxxxxxxx-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)
```

3. GDB is now running. To connect to the board, type target extended-remote | arm-stellaris-eabi-sprite armusb:?speed=2 <target>, where <target> should be replaced with the Stellaris device on the board. For this example, <target> is replaced with lm3s811. A full list of the supported target strings can be found in the Sourcery CodeBench installation directory at CodeSourcery\share\doc\sourceryg++-arm-stellaris-eabi\html\getting-started\sec-sprite-supported-boards.html. You see the following after pressing Enter:

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

```
(gdb) target extended-remote | arm-stellaris-eabi-sprite
armusb:?speed=2 lm3s811
Remote debugging using | arm-stellaris-eabi-sprite armusb:?speed=2
lm3s811
```

4. To download `hello.axf` to the flash, simply type `load`. You see something similar to this after pressing Enter:

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

5. Now that the image has been loaded into flash, the application can be restarted by typing `continue`. You should see the words “Hello World!” on the LCD panel once the application begins executing.

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

If you encounter any difficulties when working with the Sourcery CodeBench command line debugger, further documentation on the subject can be found 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 goes into further detail about running applications via `gdb`.

Step 3: Simplifying GDB Commands

It is possible (and easy) to simplify the commands that you send GDB by defining custom commands in a `.gdbinit` file and placing it in your `$HOME` directory. To find out where `$HOME` is, type “`echo $HOME`” in a shell window. For Windows XP users, the directory is “`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 Step 5, item 3. As an example, after starting GDB, typing “`connect`” would have the same effect as typing “`target extended-remote | arm-stellaris-eabi-sprite -s 2 -f lmi.dll stdio.`”

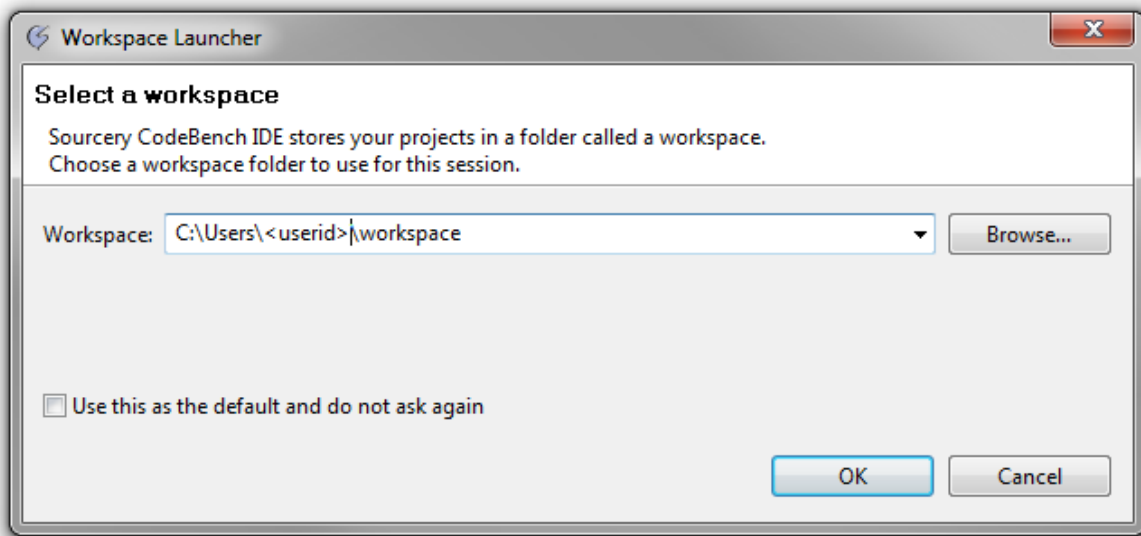
Building using the Integrated Development Environment

UnxUtils is not required for building and debugging in the IDE. Follow the steps below to use the IDE. To use the command line interface, see [Building using the command line on page 4](#).

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Step 1: Start the Sourcery CodeBench IDE

Start the IDE using the shortcut found under Start/All Programs/CodeSourcery/Sourcery CodeBench for Stellaris EABI/Sourcery CodeBench IDE. You will be presented with the following dialog box. Enter the path in which you want Sourcery CodeBench to store the source files for the project that you will import in the following steps.



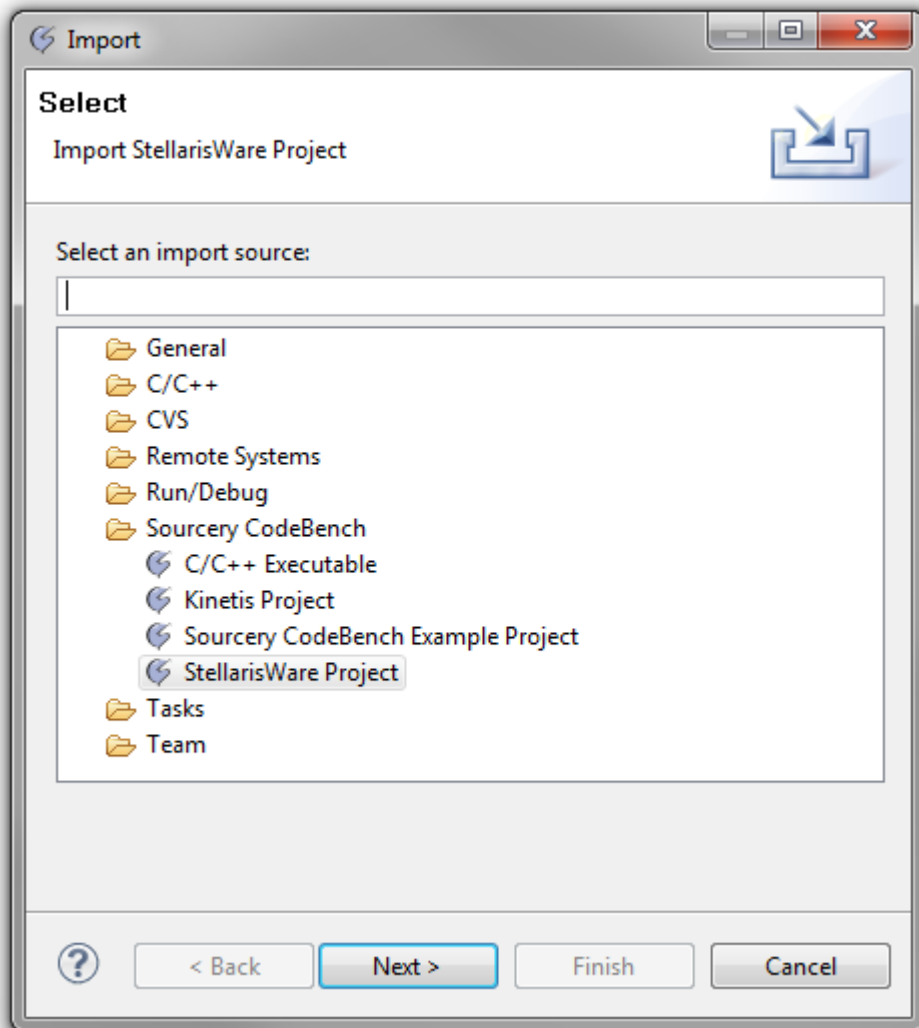
If you check the box marked, “Use this as the default and do not ask again,” this workspace opens each time you start the IDE in future.

Click “OK” to proceed. If you see the License Installation screen, obtain a license for the toolchain by following the instructions given earlier in this document.

Step 2: Import the StellarisWare release for your board

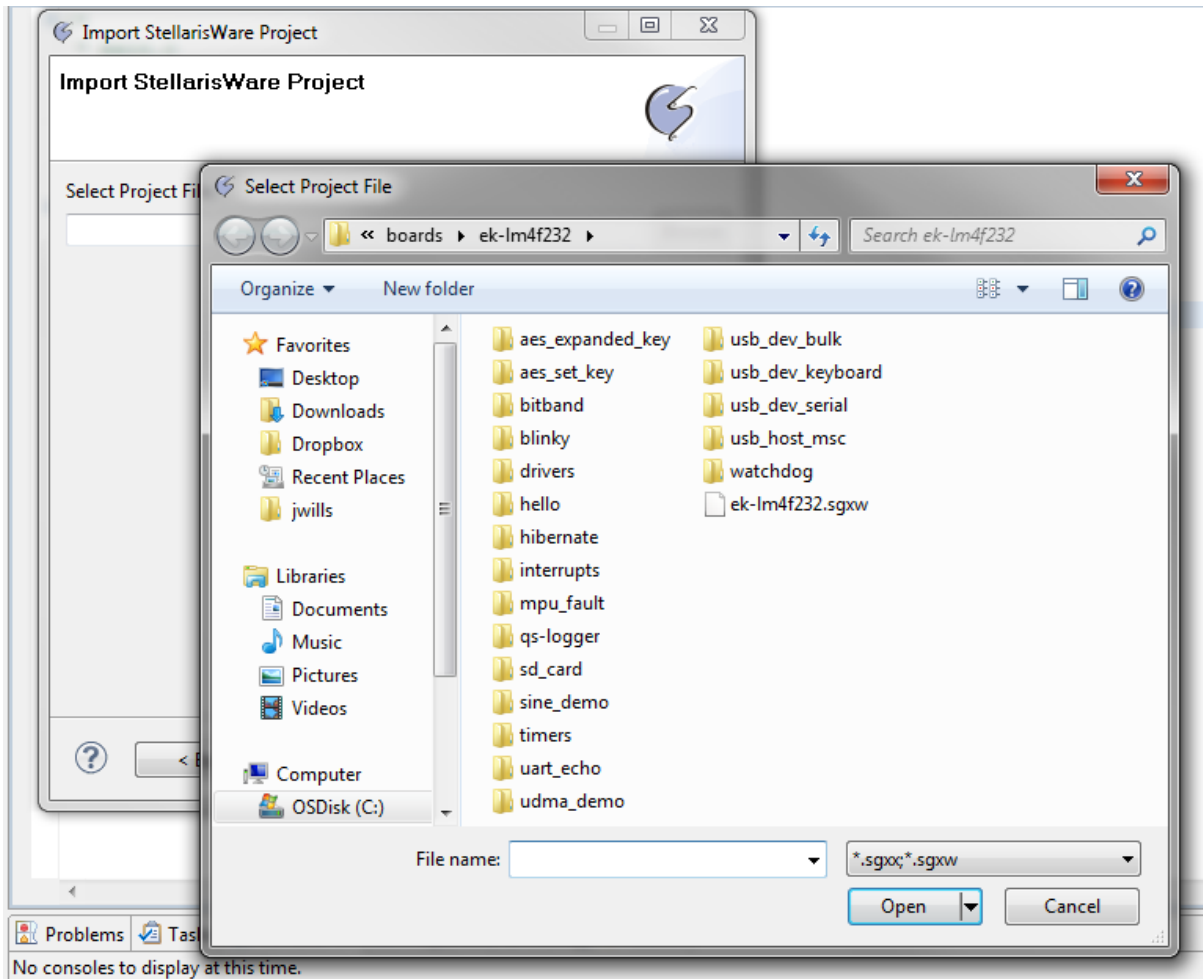
Select “File/Import...” from the Sourcery CodeBench IDE menu, then highlight “StellarisWare Project” in the “Import” dialog which appears.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS



Click “Next” and browse to the directory containing the StellarisWare release for your board. If you installed StellarisWare in the default directory, this file structure appears under C:\StellarisWare\boards\<board name>. Select the file named “<board>.sgxw,” then click “Open.” The following example assumes that you are importing the ek-lm3s9b90 examples but the same procedure applies to all other StellarisWare releases.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

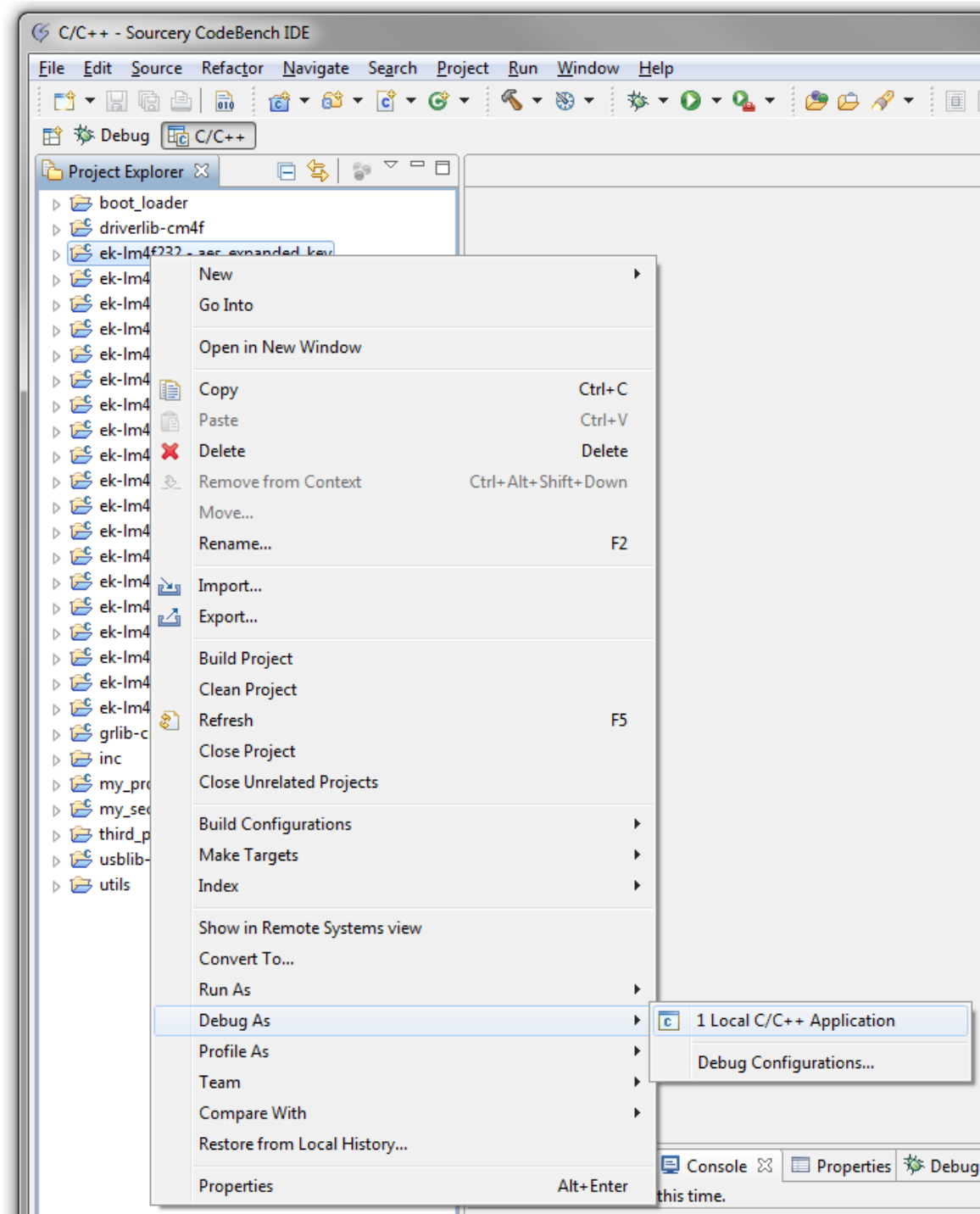


Press the “Finish” button to start importing the project files into your workspace. Once the files are imported, the IDE builds each of the libraries and example applications.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Step 3: Flash and debug an example application

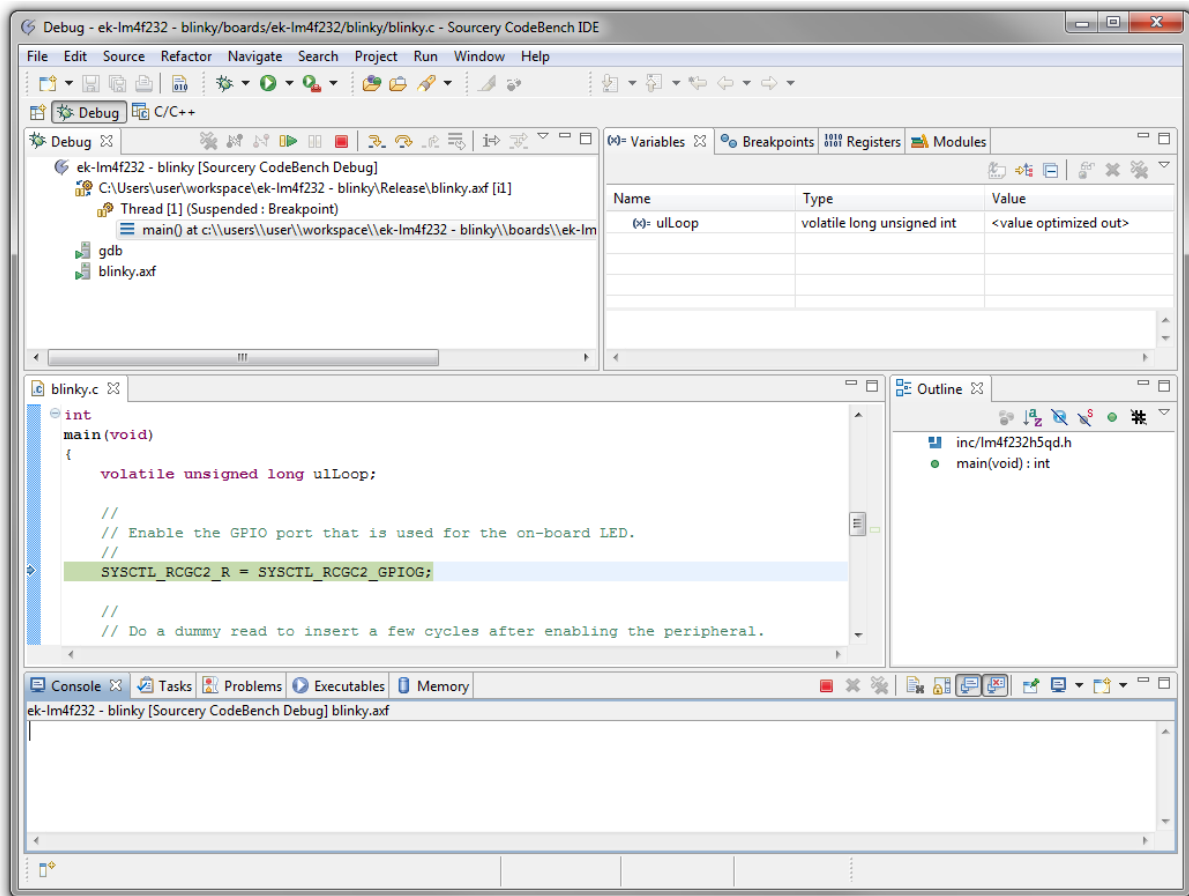
Once the imported examples have finished building, they may be downloaded to the target board for debugging.



QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Point to the example application you wish to download and click the right mouse button to show a context menu. From this menu select “Debug As...” then select “C/C++ Application” from the menu which pops up. The application downloads to your board and is ready for debugging.

At this point, you may see a dialog box asking whether you want to switch to the debug perspective (or view). Select “Yes” and, optionally, check the box indicating that you wish the switch to be automatic in the future. When the image download is complete, you should see a screen similar to the following:



The icons on the top edge of the debug pane (in the top left of the window in this example) can be used to start and stop program execution.

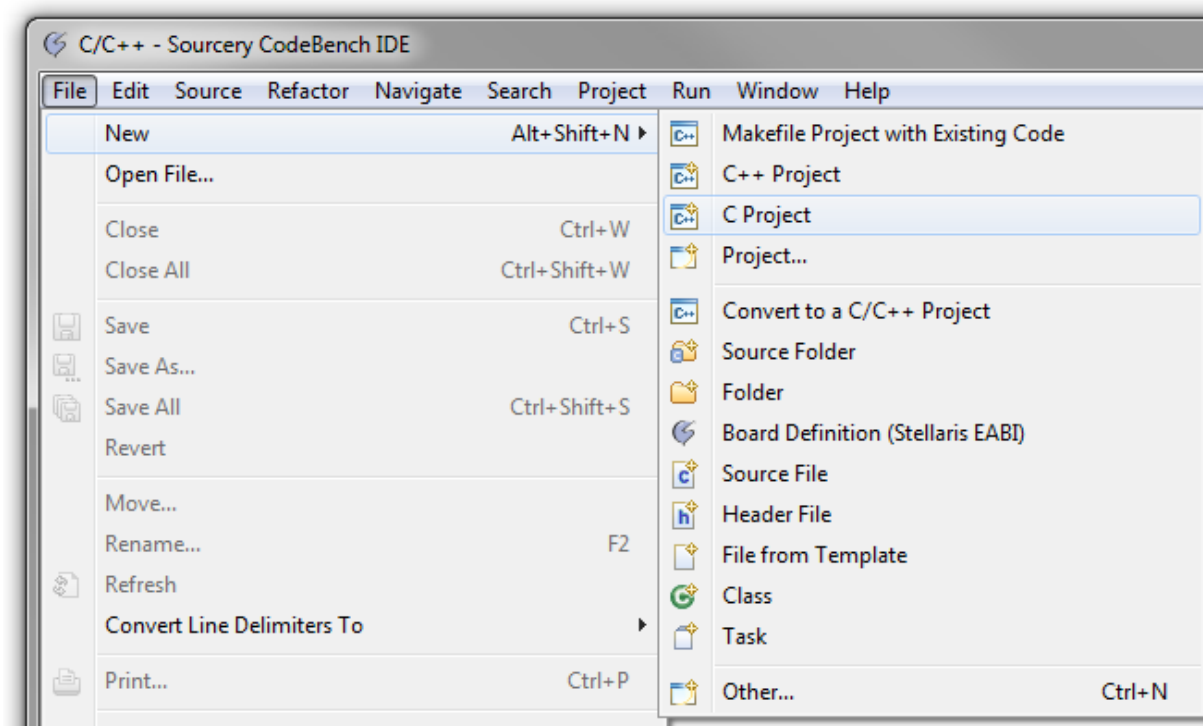
When execution completes or you terminate the debug session using the stop button (the red rectangle), you can switch back to the editor perspective by selecting “Window/Open Perspective.../C/C++...” from the IDE menu.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Creating a New Project

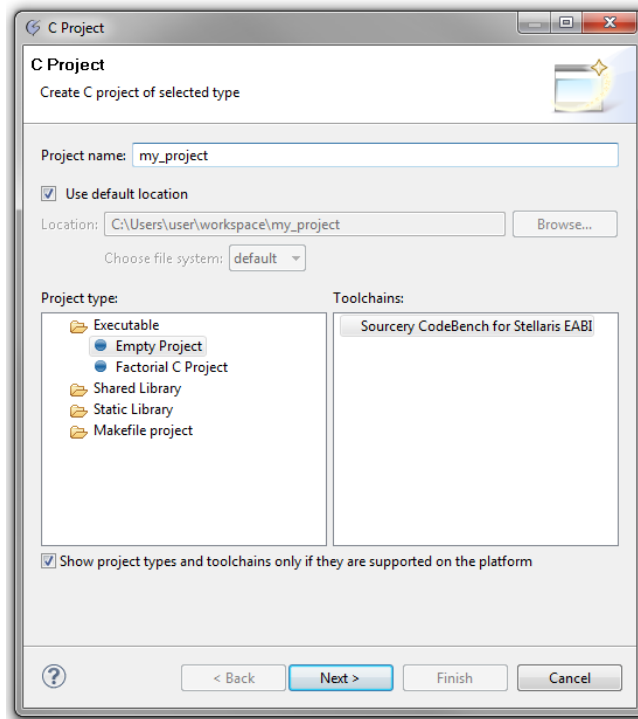
Once you have gone through the StellarisWare example applications, you may want to create your own project to start development. While you can always start with an existing, simple project, sometimes you may want to start fresh.

To add a new project to your workspace (assuming you're still using the example described above), go to File > New > C Project.

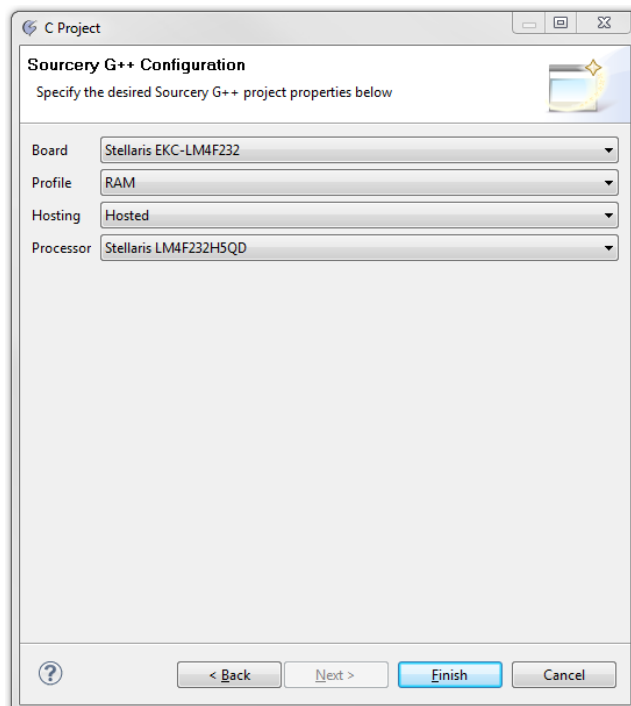


The tool prompts you with a dialog box asking for the type of project you want to create. We'll focus on the executable project types for this example. In the Executable menu, select "Empty Project" and name it.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS



Clicking on the Next button takes you to a dialog box that lets you select the device you're using. Choose the device from the Board menu, or keep the default Stellaris option.

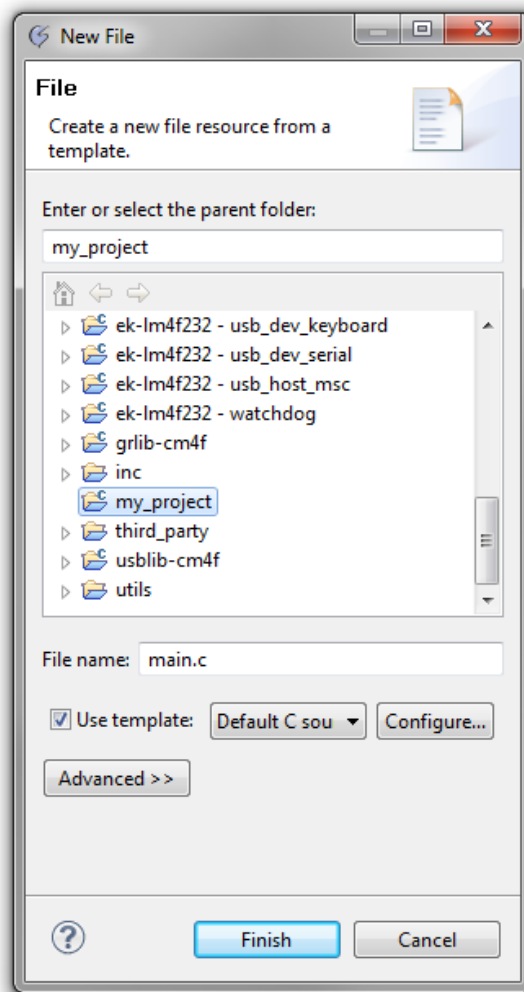


QUICKSTART – SOURCERY CODEBENCH™ TOOLS

The project that is created will be empty, so you will need to do a few things such as add source files, setup include paths, and set up the debug interface. Let's go through these step by step.

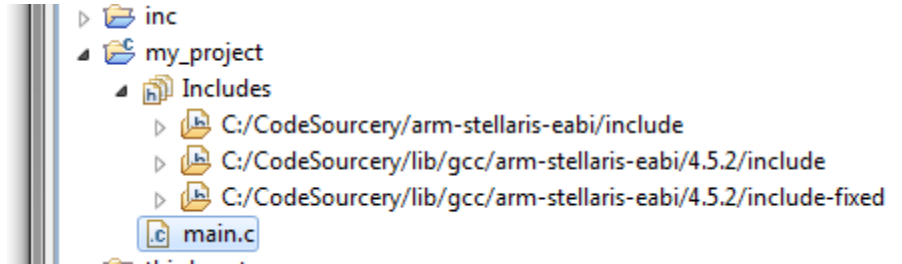
Adding Source Files:

For a basic project, you'll need to create a main function. In this example, you'll create a main file using the Sourcery CodeBench templates. To create the main file, right click on the project and select New > File From Template. In the dialog box that pops up, type "main.c" in the File Name field, and then click the Finish button. Make sure 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.



QUICKSTART – SOURCERY CODEBENCH™ TOOLS

In the end, the project should look like this:



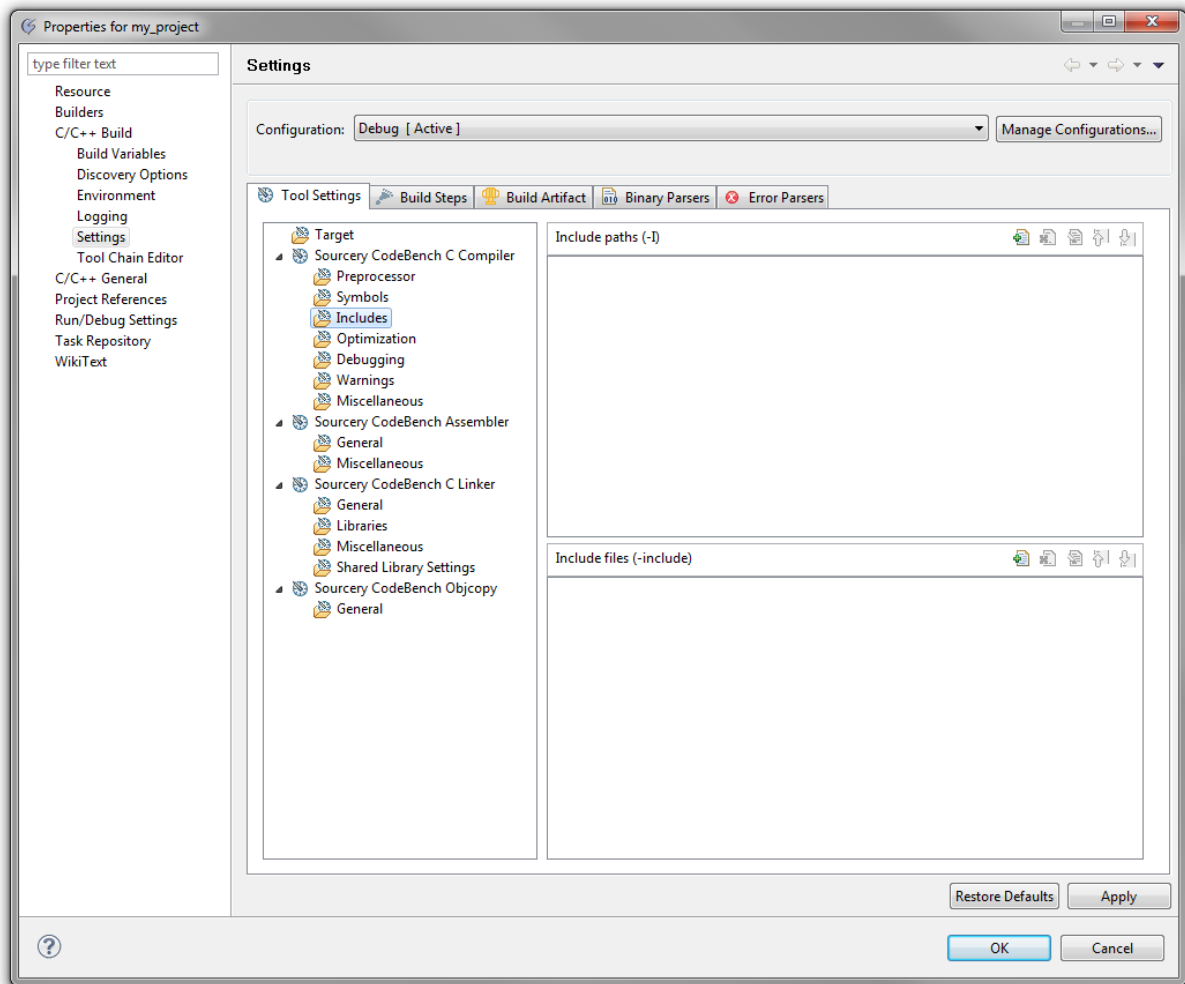
In the main.c file, add a simple main function.

Adding Include Paths, Libraries and Output:

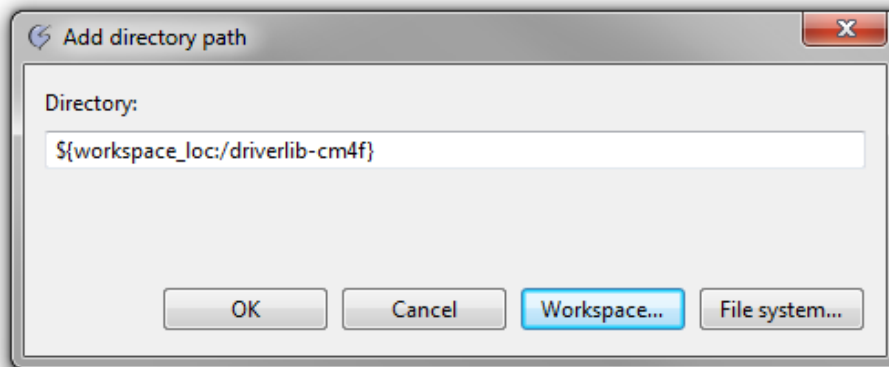
The tool has many options, but the basic set up must include the hooks to StellarisWare, primarily the Driver Library (driverlib). To get to the project options dialog, right click the project and select Properties. Expand the C/C++ Build menu and select the Settings item.

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 “+”.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

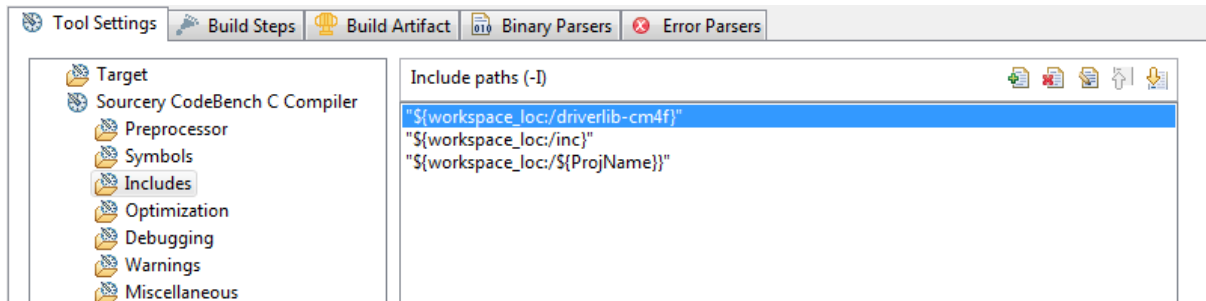


A window pops up to browse for the new path. Click on the “Workspace...” button to browse through your existing workspace. When the workspace browser appears, expand the “driverlib” project and select the Release folder.

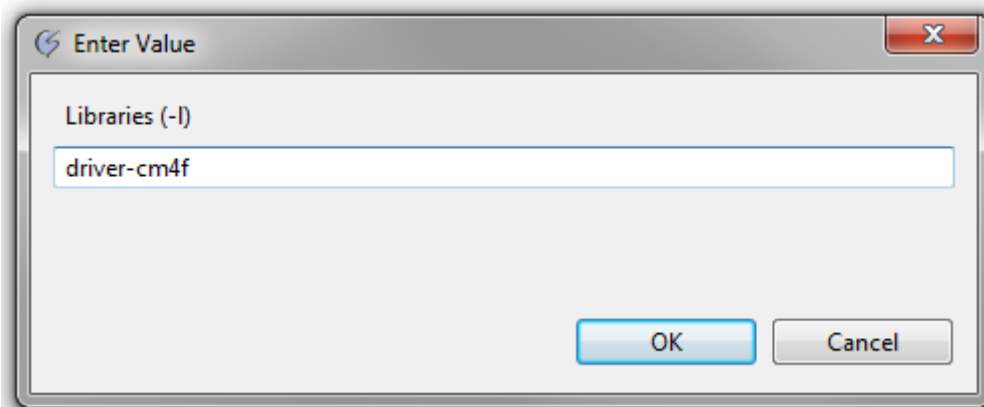


QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Repeat this process two more times, adding the “inc” project and your working project (in this case “my_project” to the include paths.

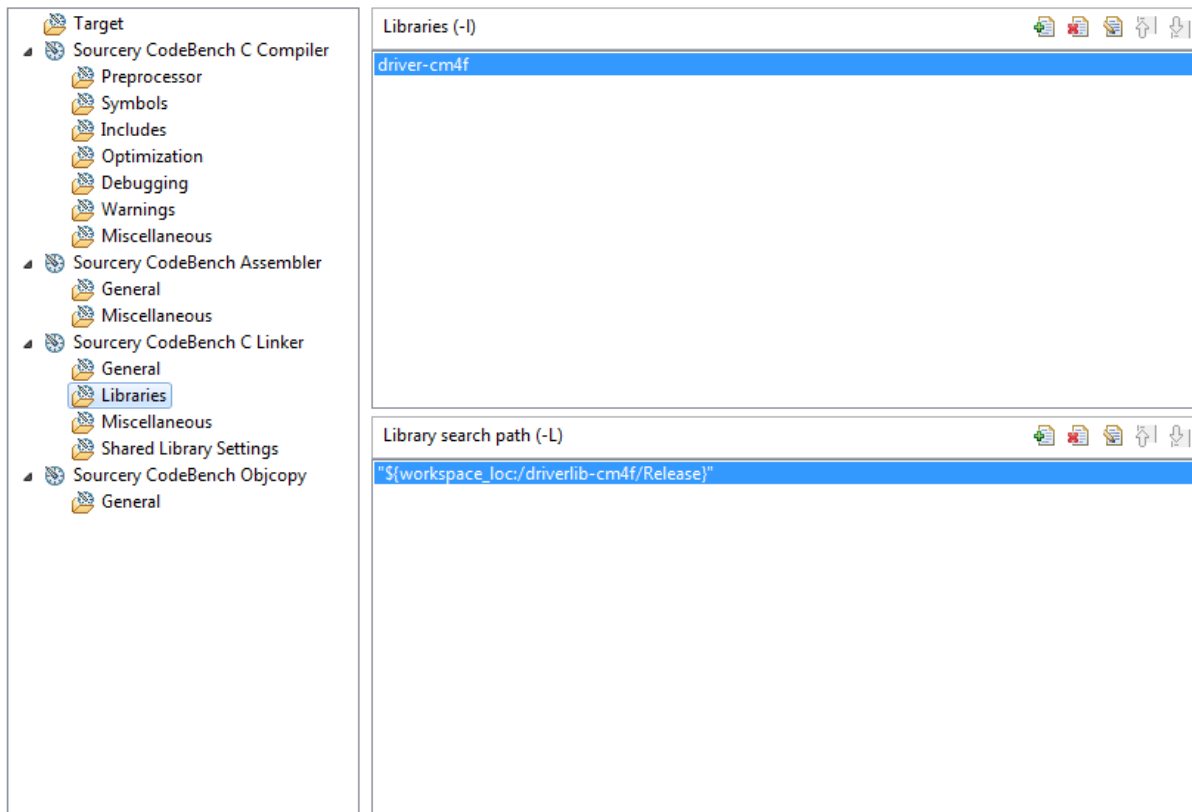


Next you’ll add a pointer to the StellarisWare Driver Library. Select the Libraries item under Sourcery CodeBench C Linker. Again, to add a new item, click the little icon with the green “+”. In the Libraries field, type the name of the Driver Library to match the project that was included in the previous step. For this example, our Driver Library project was named driverlib-cm4f, so we simply type driver-cm4f in the Libraries field.

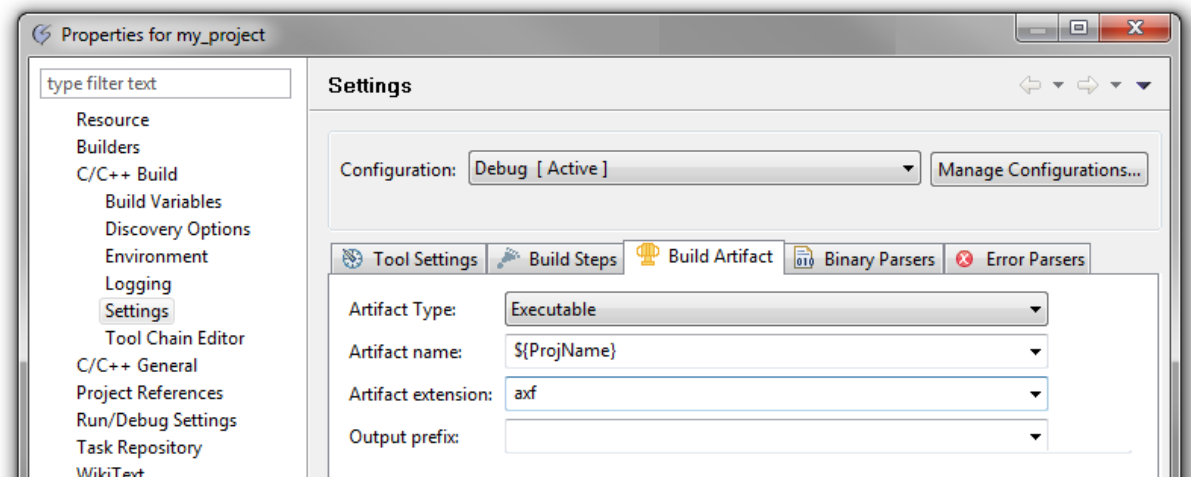


In the Library search path, again point the tool to the driverlib\Release directory, just like you did for the C include paths.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS



The last step is to tell the tool how to format the output executable. The default configuration generates an output file in ELF format, but without an extension. To change it to an AXF file, go to the Build Artifact tab and type “axf” in the Artifact extension box.

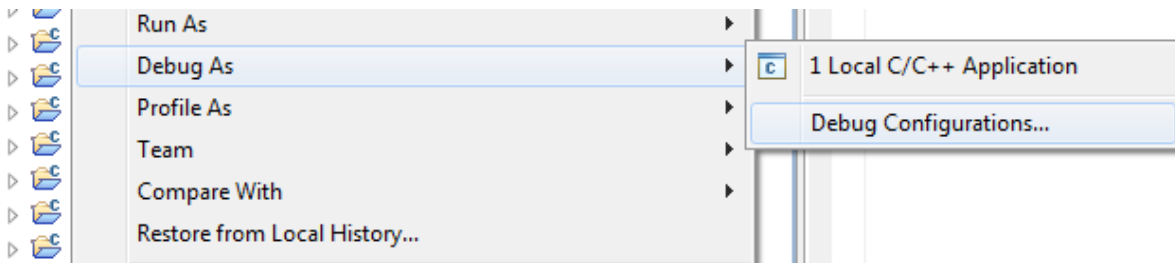


Now you'll be able to build your project.

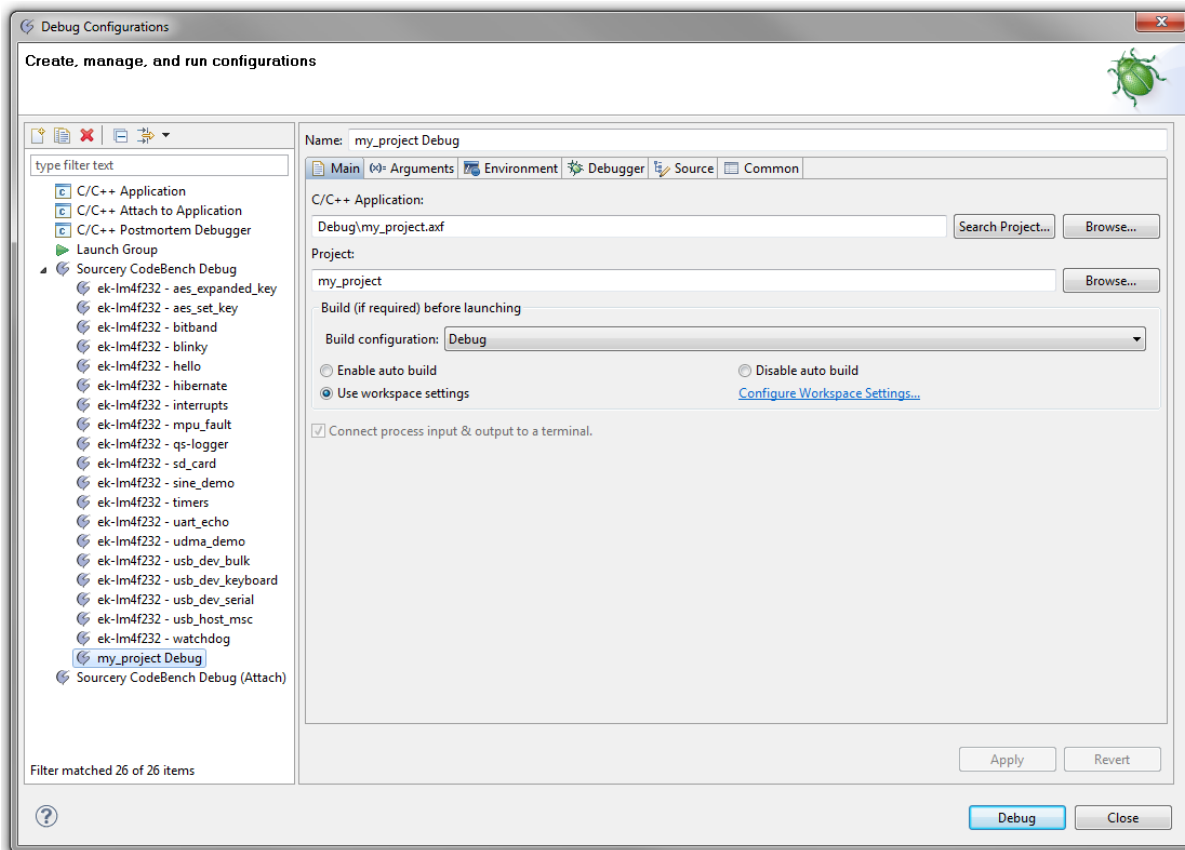
QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Adding a Debug Configuration:

Before debugging the application, you must create a debug configuration. To do this, right click on the project and select **Debug As > Debug Configurations...**



In the dialog box that appears, simply double click on the Sourcery CodeBench Debug item, and the tool automatically creates a new debug configuration for you (the new configuration is called “my_project Debug”).



After applying the changes, click on the **Debug** button. From there, follow the steps described earlier in this document about using the debugger.

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

Conclusion

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

QUICKSTART – SOURCERY CODEBENCH™ TOOLS

References

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

- *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:

- Sourcery CodeBench website at <http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/>

Copyright © 2006–2011 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>



IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Mobile Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Transportation and Automotive	www.ti.com/automotive
Video and Imaging	www.ti.com/video

TI E2E Community Home Page

e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated