

Tiva™ C Series Development and Evaluation Kits for Keil™ RealView® MDK

Tiva C Series and Evaluation Kits provide a low-cost way to start designing with Tiva microcontrollers using Keil's RealView Microcontroller Development Kit (MDK) for ARM® controllers. The boards can function as either a complete evaluation target or with minor modifications as a debugger interface to any external Tiva C Series device.

1 Requirements

- PC with a USB interface, running Microsoft Windows® 2000, Windows XP, Windows 7, or Windows 8 operating systems (OSs).
- 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](#)).

2 Keil RealView MDK

This quick start guide shows how to install the evaluation version of the Keil RealView MDK, and then how to use it to build and run an example application on your Tiva C Series Evaluation or Development Board.

2.1 Installation

2.1.1 Install the RealView MDK Tools

1. 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/tool/sw-<kit_name>).
2. With the Evaluation Kit USB flash drive, click the Keil Logo to download a .zip file containing the installer. If you prefer or if you are using downloaded software, you can use Windows Explorer to view the files on the USB flash drive and double-click the MDKxxx.exe file in the *Tools\Keil* directory.

Tiva, TivaWare are trademarks of Texas Instruments.
Stellaris is a registered trademark of Texas Instruments.
Keil, Cortex are trademarks of ARM limited.
RealView, ARM are registered trademarks of ARM limited.
Microsoft Windows is a registered trademark of Microsoft Corporation.
All other trademarks are the property of their respective owners.

2.1.2 Install the TivaWare™ C Series Software Package

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

NOTE: If you are navigating the USB flash drive using Windows Explorer (or a similar application), go to the *Tools/TivaWare* directory.

2. Run the TivaWare installer. To manually install TivaWare, the installer is a self-extracting .zip file that is located in the *Tools/TivaWare* directory. Use a .zip file extraction utility such as WinZip to manually extract the contents.
3. To view the TivaWare documentation, navigate to the *Tools/TivaWare/docs* directory and click the *Tiva Peripheral Driver Library User's Guide* PDF.

NOTE: For the most recent version of TivaWare, check the www.ti.com/tiva-c website

2.2 Start the Keil μVision IDE and Load the Hello Project

1. Start the Keil μVision IDE by double-clicking the icon on your desktop or by selecting it from the Windows Start Menu. When the IDE loads, it was a blank screen (see [Figure 1](#)).

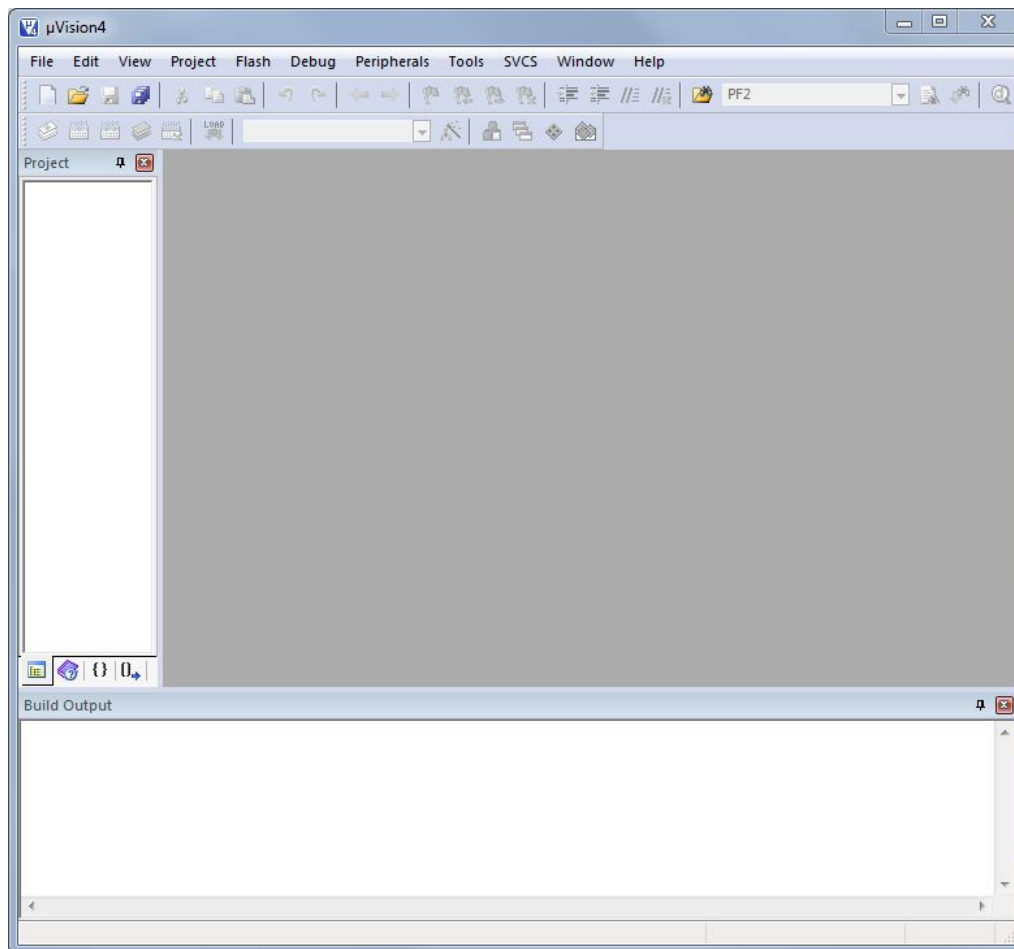


Figure 1.

- From the Project menu, select Open Project... (see Figure 2).

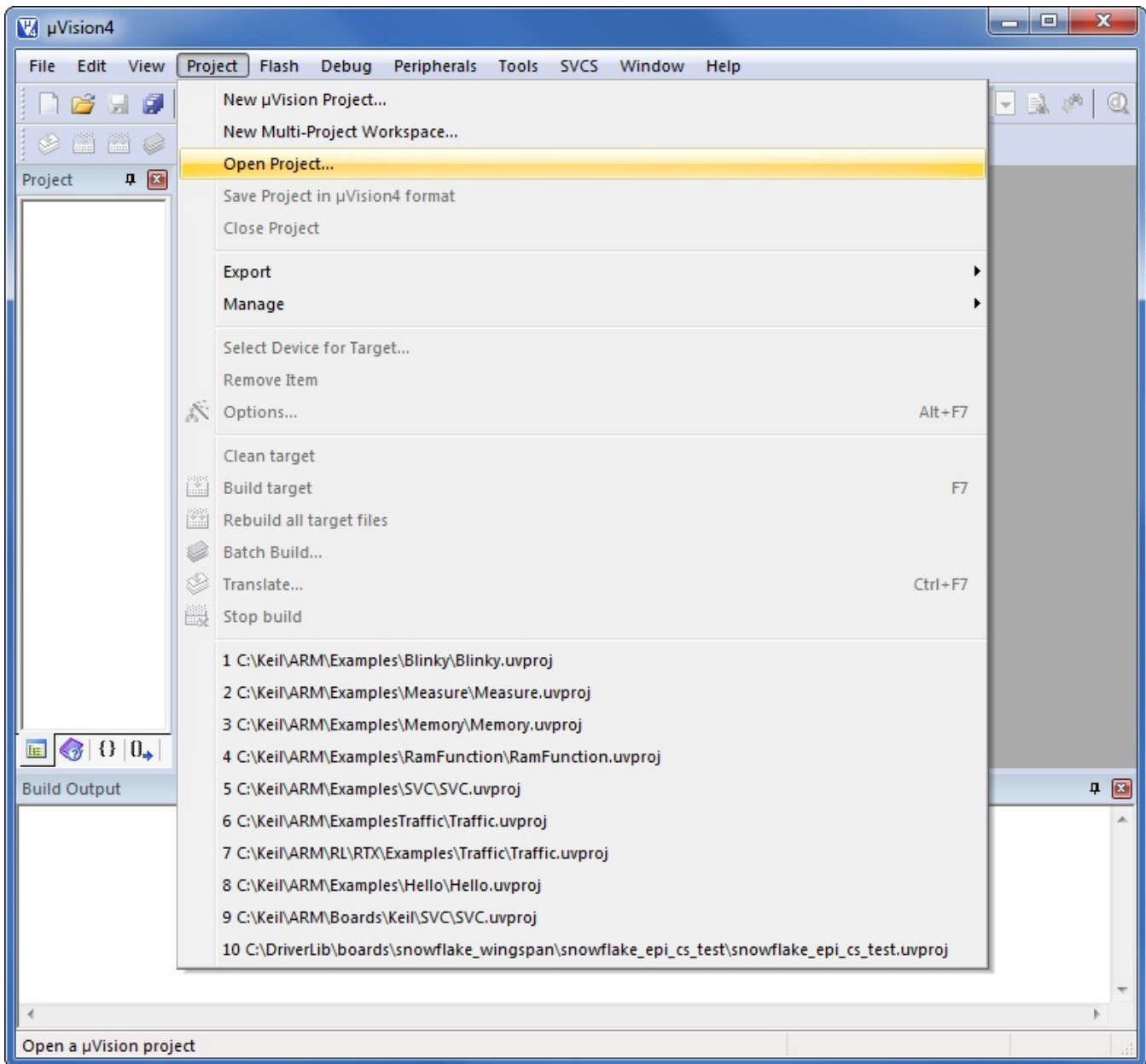


Figure 2.

- Use the dialog box to navigate to the Hello program in the directory appropriate for your board. From the location where you installed TivaWare, the Hello project is located in: *ti\TivaWare_C_Series-n.n\examples\boards\{board_name}\hello*

NOTE: The Keil tools also installs TivaWare as part of the default installation, but the version may be older than what is currently available in the development or evaluation kit. You can find TivaWare in the Keil tree by looking in: *C:\Keil\ARM\Boards\Luminary\{board_name}*

Important: For the most recent version of the TivaWare example projects, check www.ti.com/tiva-c for the latest software updates.

4. Select the hello.uvproj project file and click Open. The project opens in the IDE (see [Figure 3](#)).

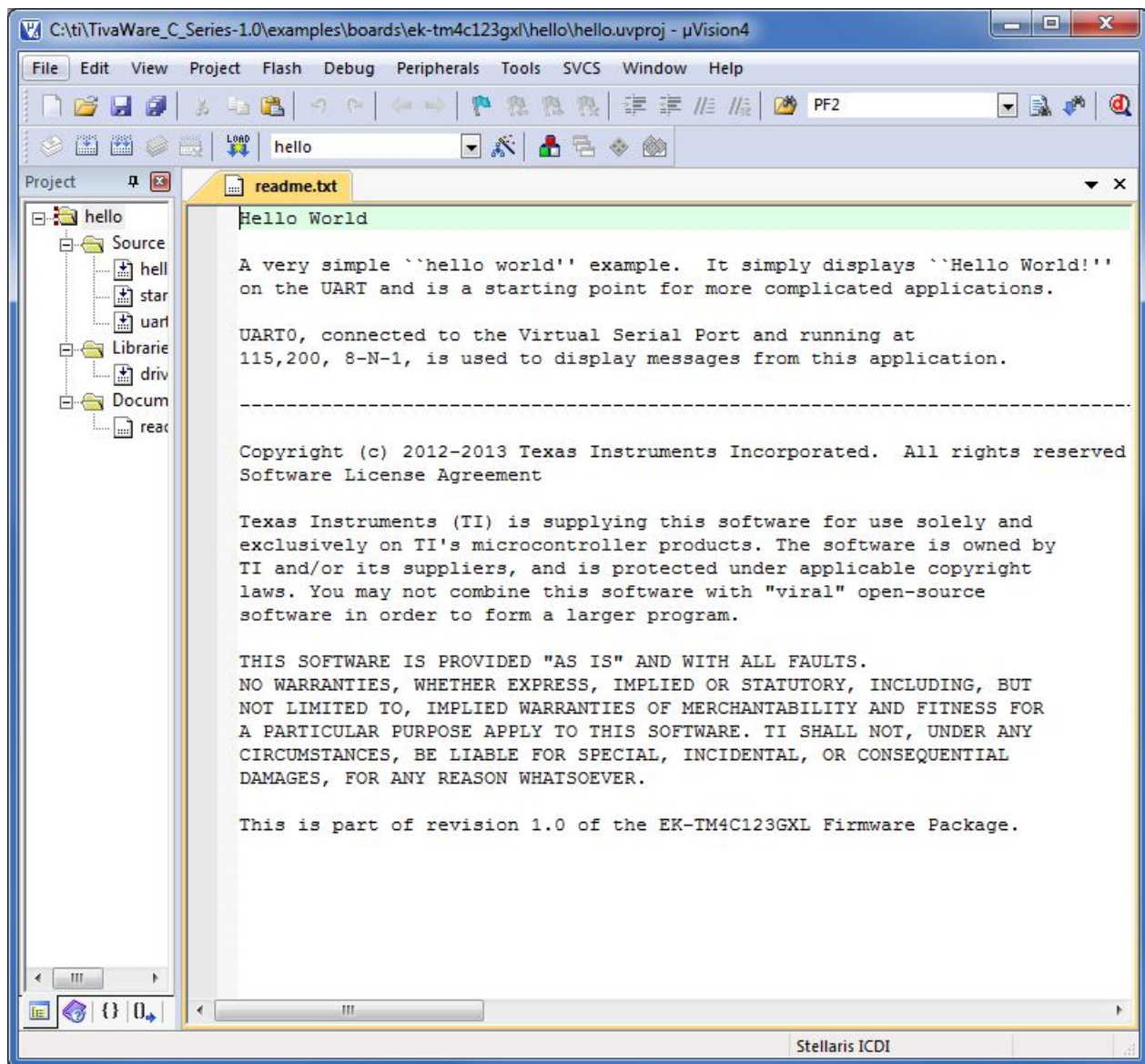


Figure 3.

5. You can view source files in the project by double-clicking a filename in the Project Workspace pane on the left. For example, double-click hello.c, and the source file opens in the editor (see [Figure 4](#)).

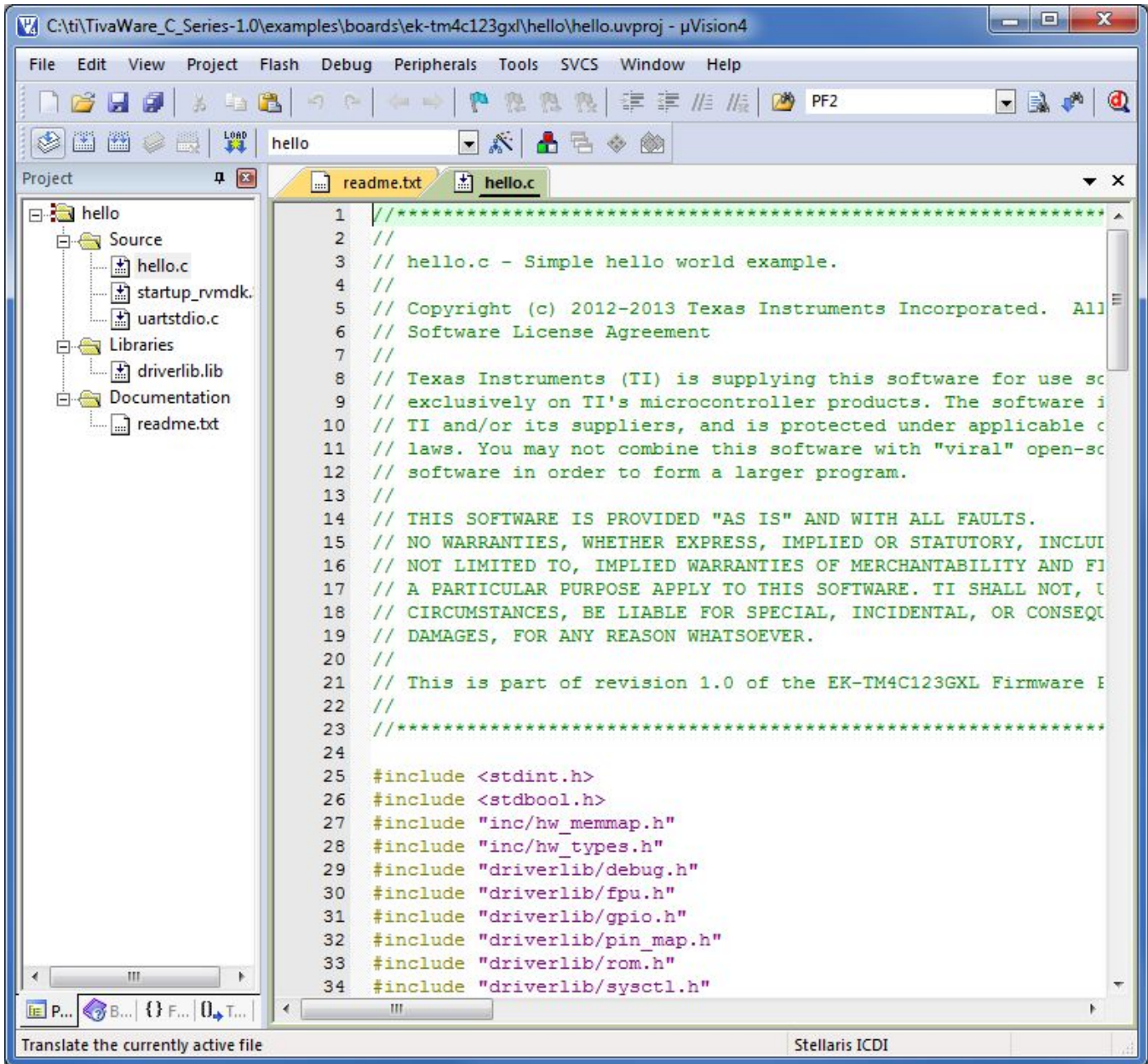


Figure 4.

2.3 Build the Hello Project

1. Select Project → Rebuild all target files, or click the Rebuild all button (icon) (see Figure 5).

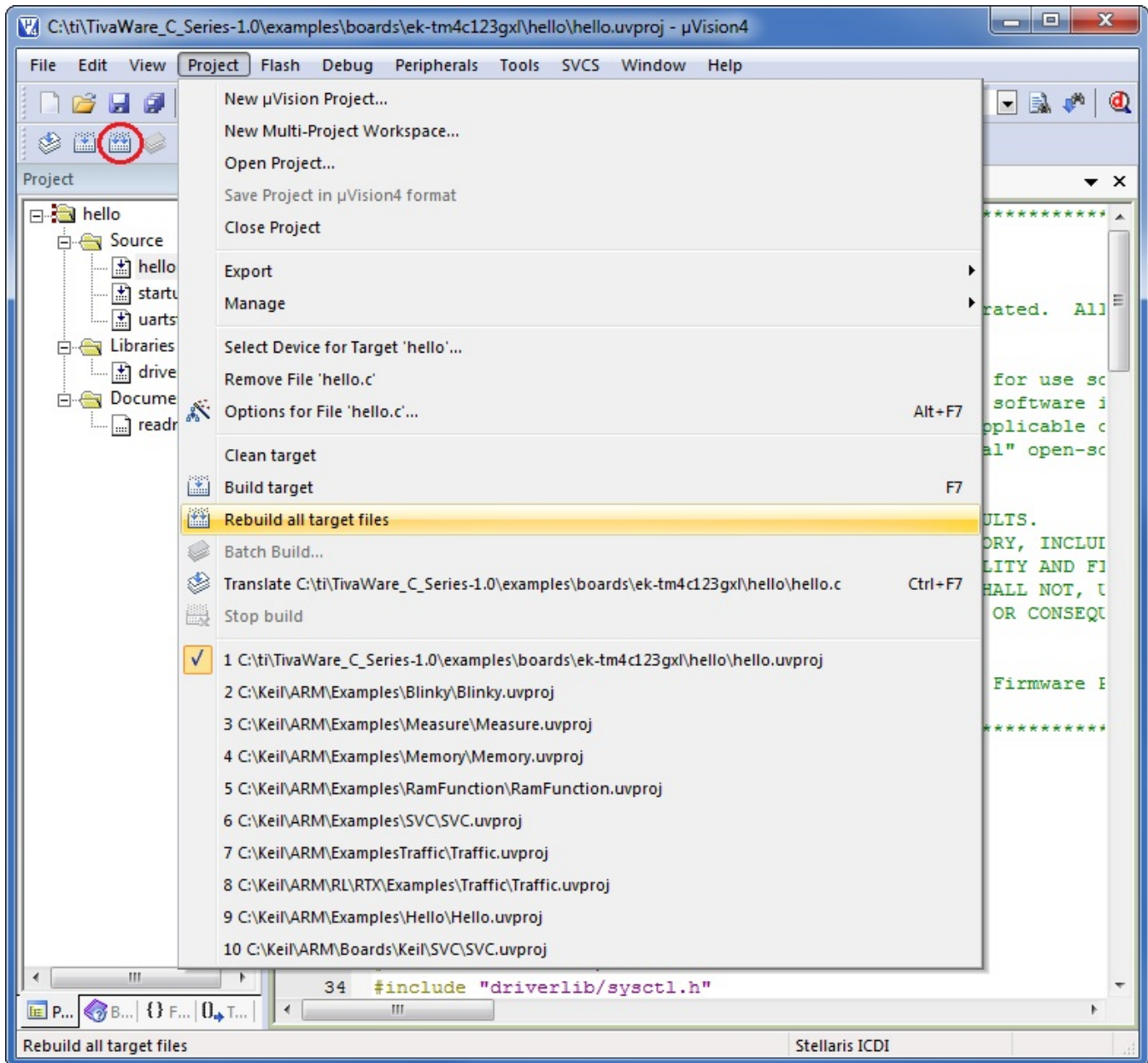


Figure 5.

All of the source files are compiled and linked. The activity can be seen in the Build window at the bottom of the μVision IDE. The process completes with an application named hello.axf built with no errors and no warnings (see Figure 6).

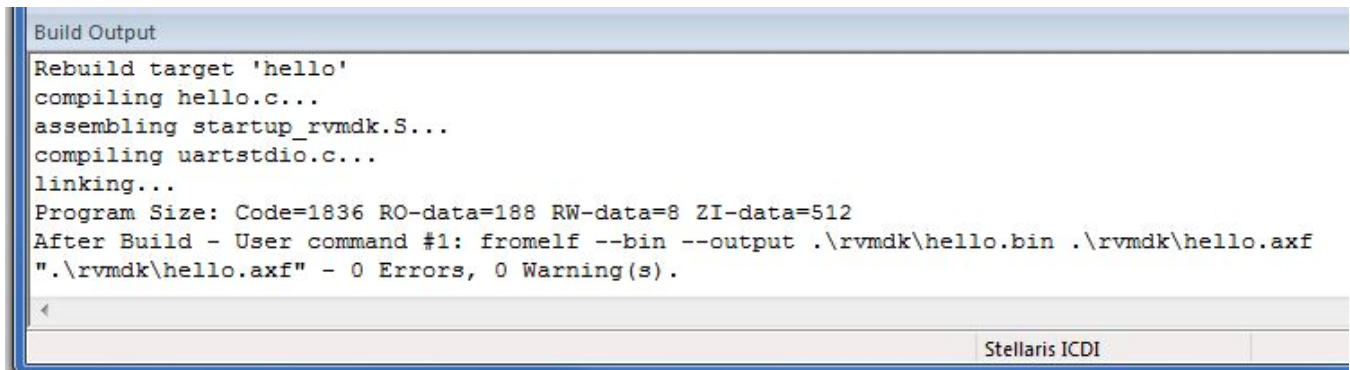


Figure 6.

2.4 Load the Hello Program into the Flash Memory

You can debug with either the on-board ICDI or you can use the Keil ULINK debug probe.

1. Select Download from the Flash menu, or click the Download button (icon) (see Figure 7).

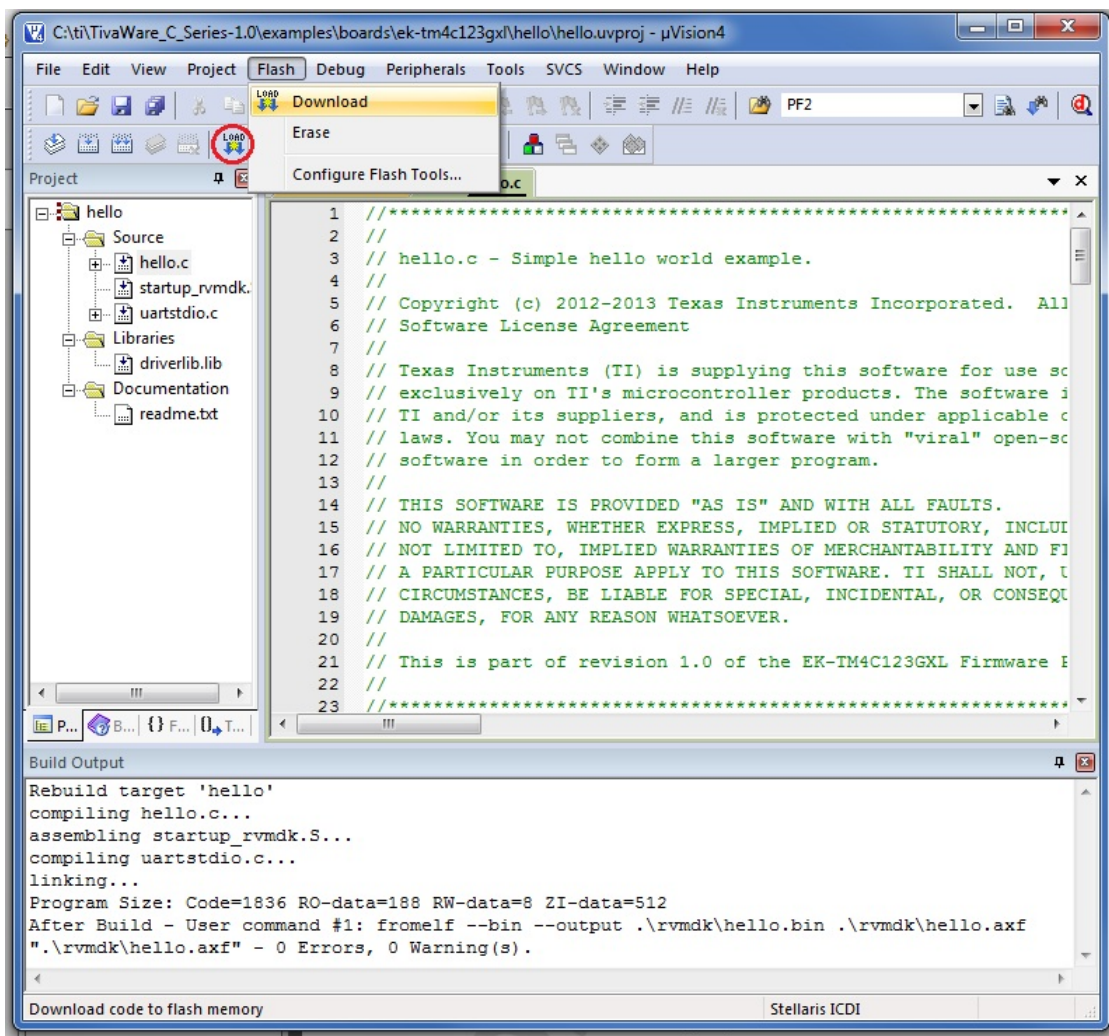


Figure 7.

The process takes a few seconds. A progress bar will show at the bottom of the IDE window as the device is programmed. When it is finished, the Build window will show that the device was erased, programmed, and verified OK.

The Hello application is now programmed into the flash memory of the Stellaris microcontroller on the Evaluation Board.

2.5 Debug and Run the Hello Program

1. Select Start/Stop Debug Session from the Debug menu or click the Debug button (icon) (see [Figure 8](#)).

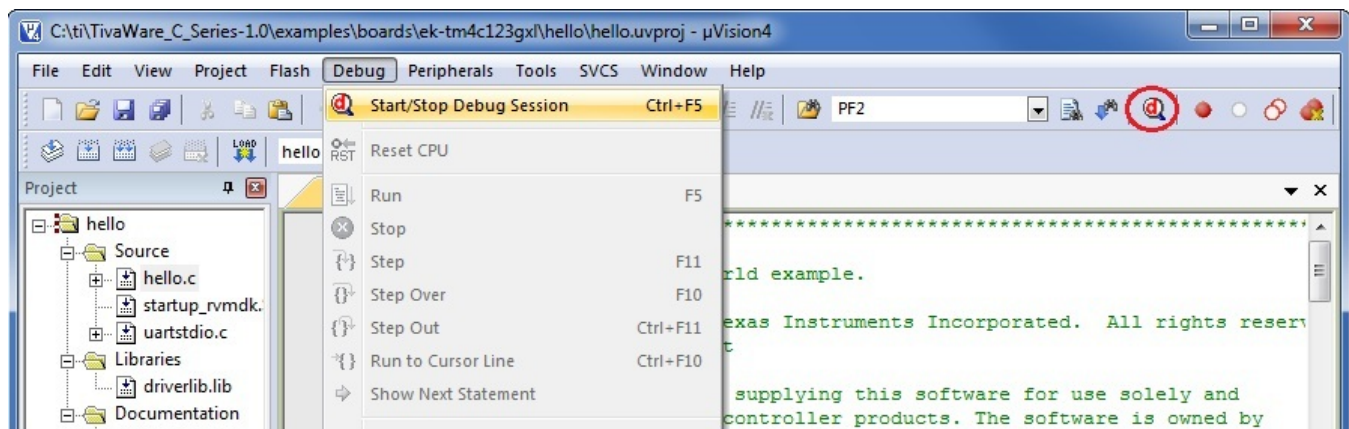


Figure 8.

The IDE switches to debugging mode. The processor registers show in a window on the left, the debugger command window is visible at the bottom, and the main window shows the source code being debugged. The debugger automatically stops at main (see [Figure 9](#)).

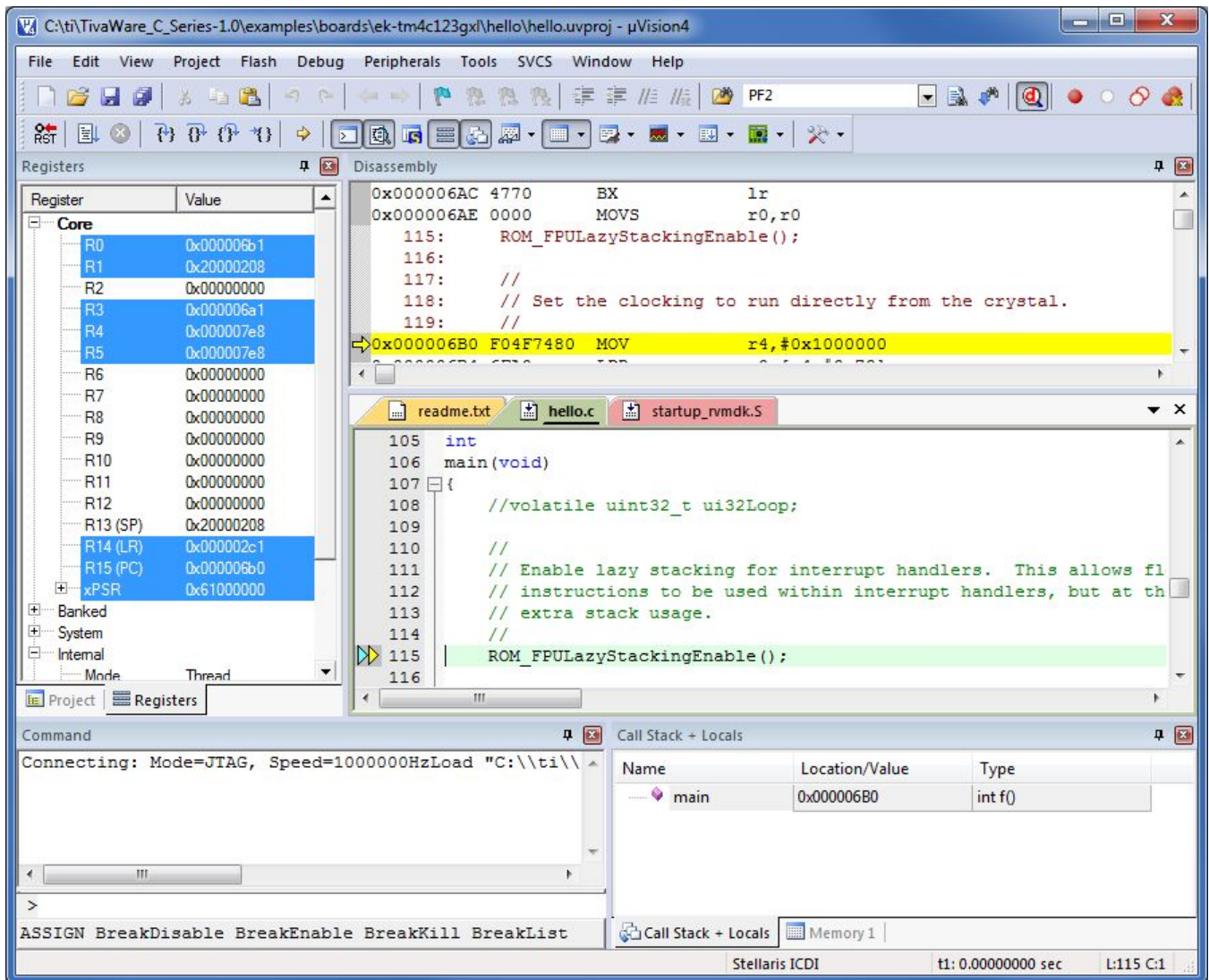


Figure 9.

From here, you can:

- Examine and modify memory.
- Program variables and processor registers.
- Set breakpoints.
- Single step through a program.
- Perform other typical debugging activities.

- To run the program, select Run from the Debug menu, or click the Run button (icon) (see [Figure 10](#)).

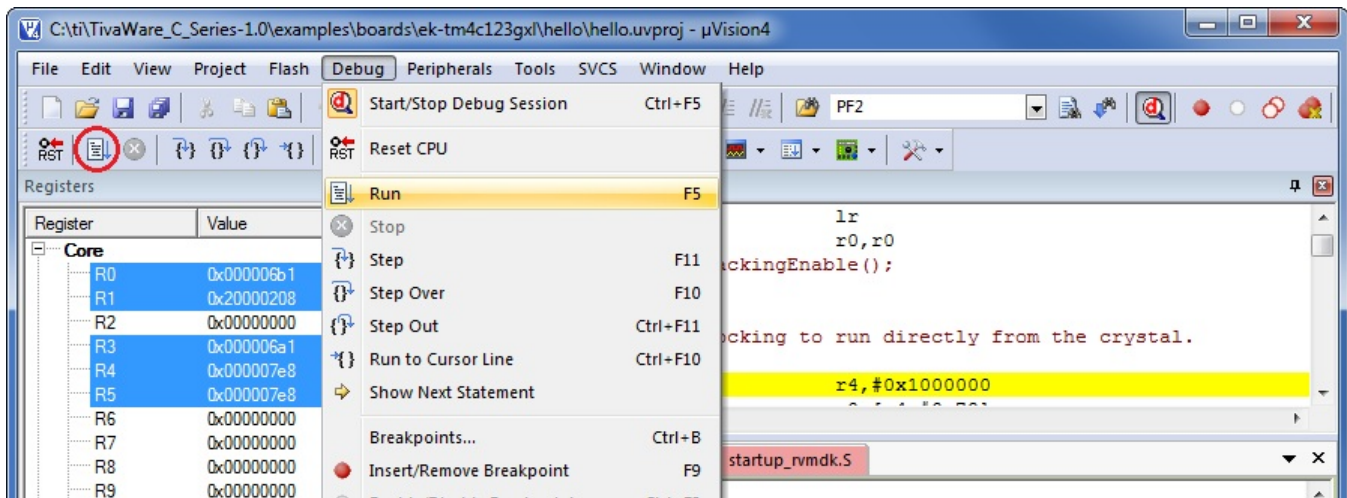


Figure 10.

The application runs and the text Hello World! displays on the PC via the UART or on the display of the evaluation board.

2.6 Build and Run Additional Example Programs

There are several additional example project folders under the folder: *ti\TivaWare_C_Series-n.n\examples\boards\{board_name}*.

These additional example programs are discussed in the *TivaWare Peripheral Driver Library User's Guide* (found on the Tiva C Series Development Kit Software USB flash drive). You can open, build, and run any of these programs in the same way by referring to [Section 2.3](#) and opening a different project, as long as it fits within the 32-KB code limit of the evaluation tools. All projects larger than 32-KB have binaries that can be downloaded with the LMFLASH utility.

3 Create a New Project

Once you have gone through the TivaWare 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 create a new project.

To create a new project, perform the following steps:

- In the Project menu, select Project → New uVision Project... (see [Figure 11](#)).

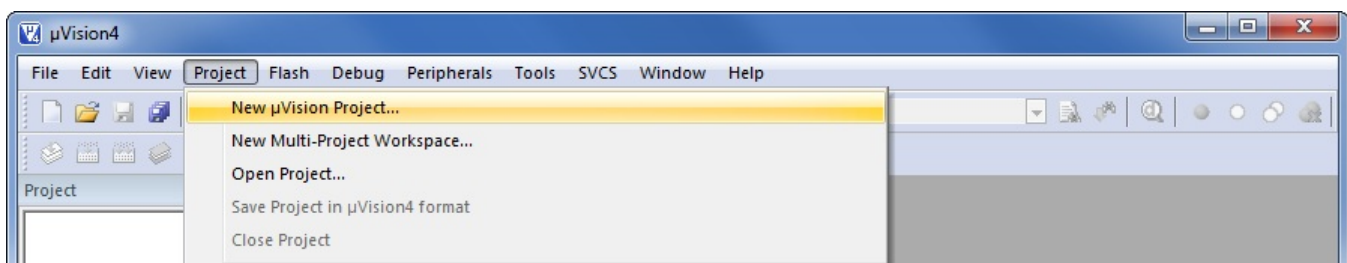


Figure 11.

- The Create New Project dialog box appears (see [Figure 12](#)). Create the project within the TivaWare tree by either saving it in the existing *ti\TivaWare_C_Series-n.n\examples\boards* directory, or creating a new item in the boards directory that corresponds to your specific board or development.
- If you create a new item, create a new directory called *my_board* in *ti\TivaWare_C_Series-n.n\examples\boards*, and create a new project called *my_project*.

The TivaWare tree now looks like: `C:\ti\TivaWare_C_Series-1.0\examples\boards\my_board\my_project` (see [Figure 12](#)).

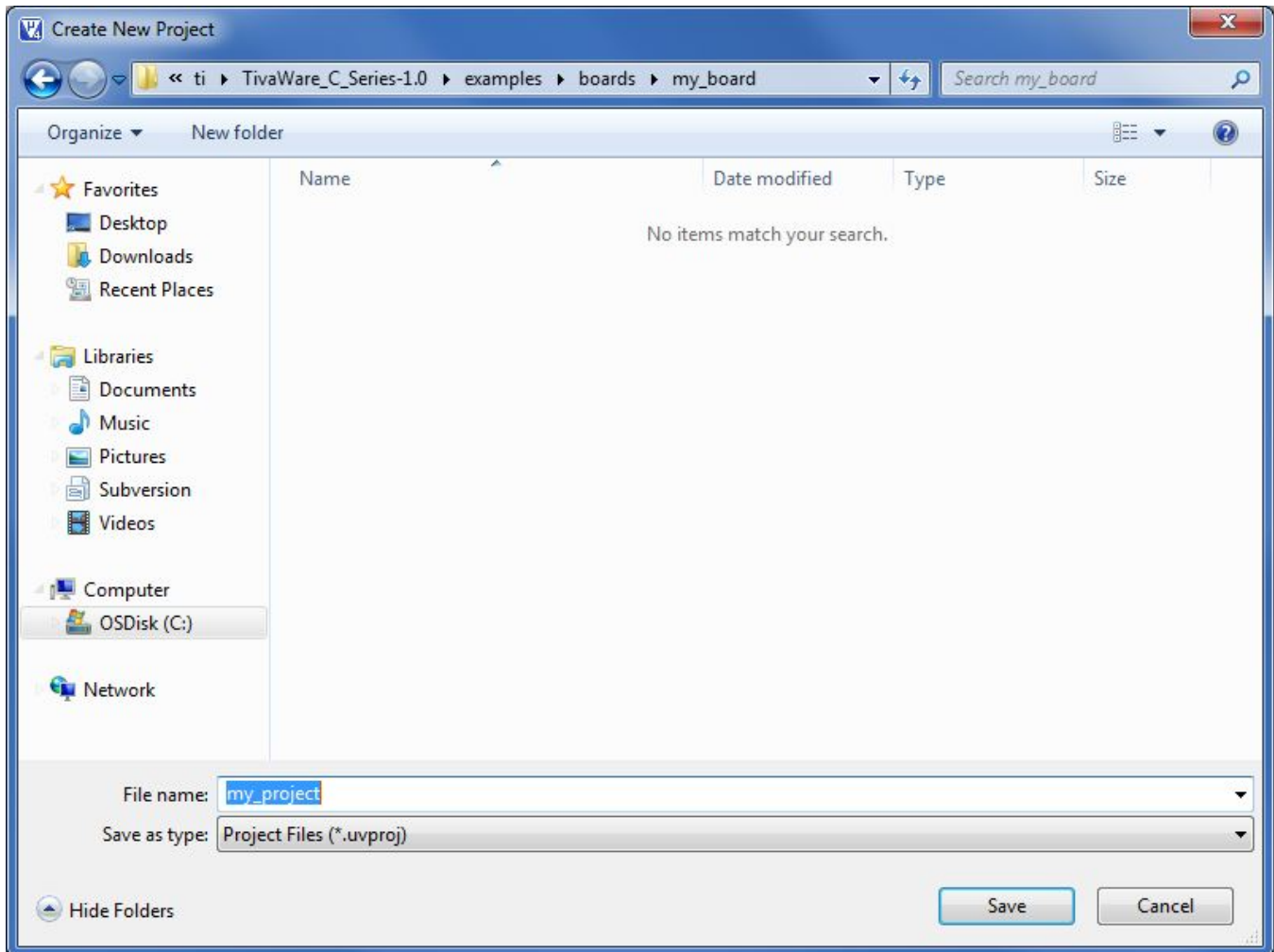


Figure 12.

- Once the project file (.uvproj) is saved, a dialog window appears asking you to select the device that you are using (see [Figure 13](#)). Select the appropriate device under the Texas Instruments list.

Important: The device selection is used primarily to configure the memory sizes (linker). If your device is not on this list, select a similar device.

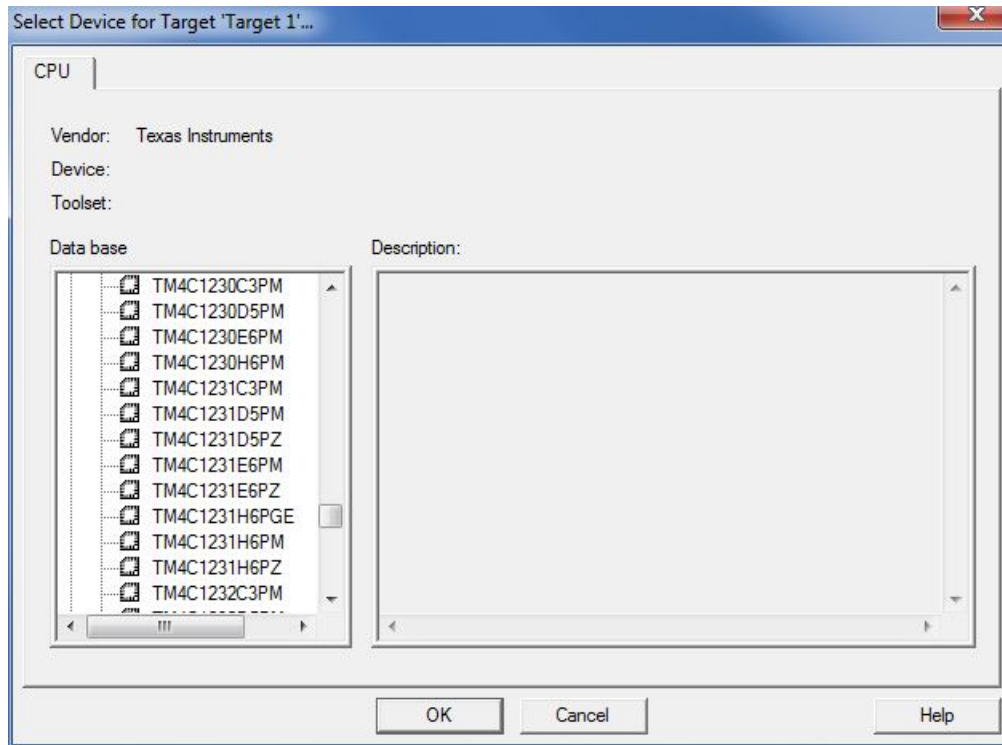


Figure 13.

- The tool asks whether you want to add startup code to the project (see [Figure 14](#)). You can click Yes or No, but just know that you do need to have startup code whether it is provided by Keil or you get it elsewhere. For this example, we will add the startup code provided by the tool.

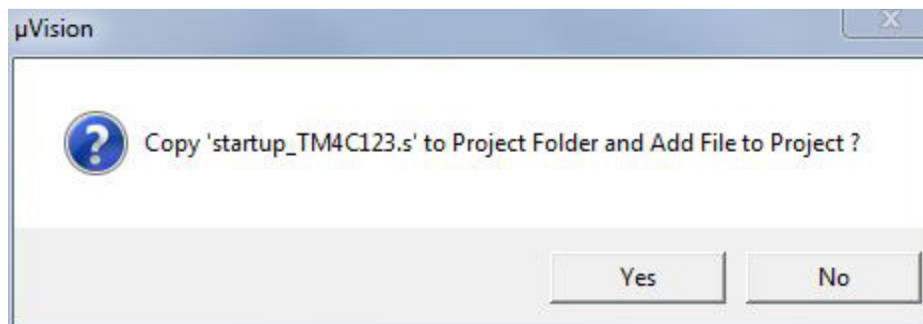


Figure 14.

The empty project appears as shown in [Figure 15](#).

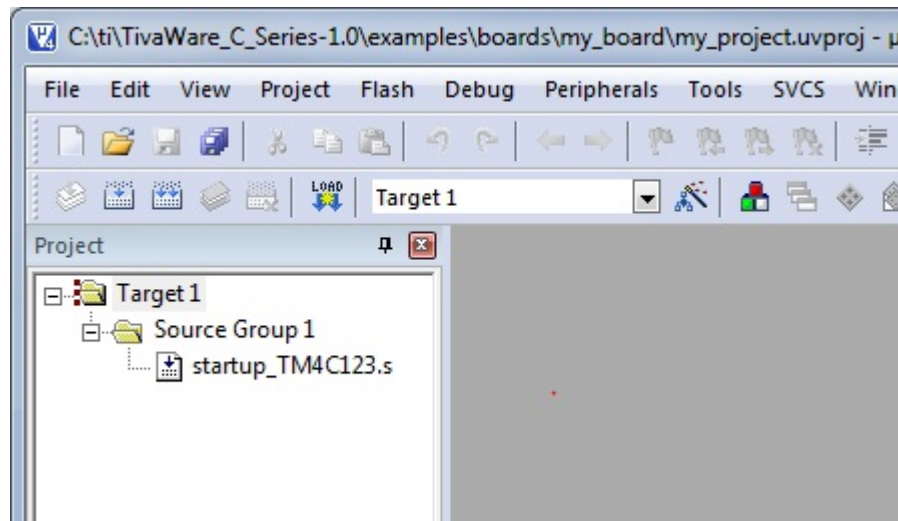


Figure 15.

The startup code adds the vector table for you, so you need to create a C file with a main function.

6. Create a new file (from File → New...) and add a simple main function with a while(1) loop.
7. Save the new file as a *.c file in your project directory.
8. Right-click the Source Group 1 folder and select Add Files to Group 'Source Group 1' (see [Figure 16](#)). When the dialog box pops up to find the file, browse to your project directory and select main.c.

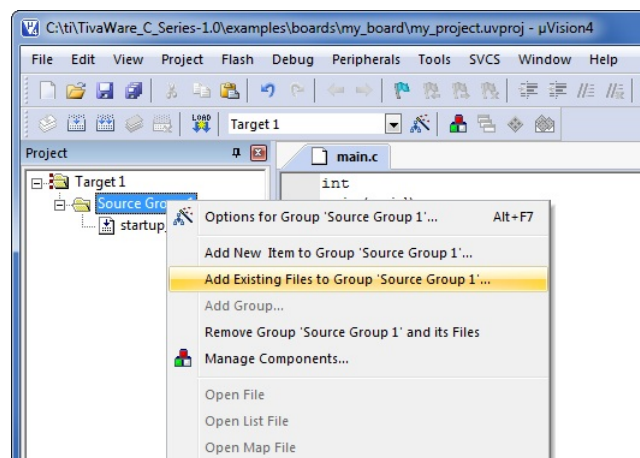


Figure 16.

This process gives you the most basic project that will run on a Tiva microcontroller. The next steps include configuring your project for your specific board, adding the hooks for TivaWare, and setting up the debug and flash programming environment.

3.1 Configure Your Hardware

To set up the project for your specific board:

Go to Project → Options for Target 'Target 1.' The Options for Target 'Target 1' dialog box appears (see [Figure 17](#)).

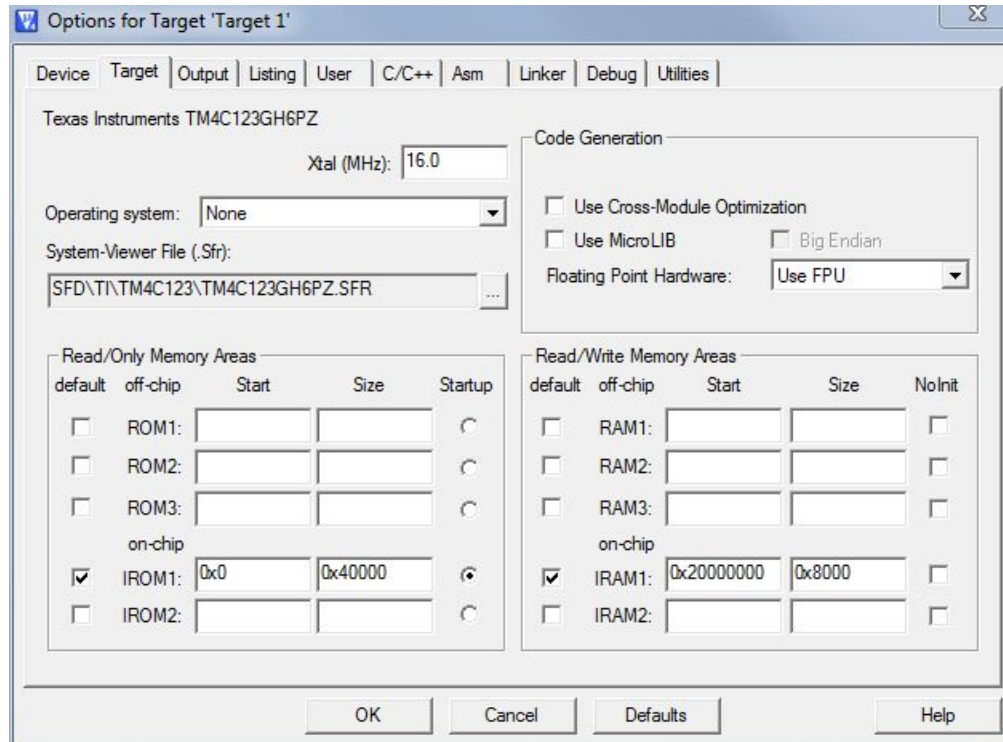


Figure 17.

Here you must set up the Xtal frequency for your specific board. This step is important because the value of the Xtal field is used to calculate flash programming timing later on. Also check the ROM/RAM sizes to make sure that it corresponds to the device you have.

3.2 Set up Debug

1. In the same project options dialog shown in [Figure 17](#), select the Debug tab (see [Figure 18](#)).
The default configuration for a new project is to use the simulator, which is not going to allow you to debug real hardware.
2. Choose the Use radio button on the right side with the ULINK Cortex™ Debugger next to it (this selection is the default in the drop-down box).
3. If you are using a Tiva C Series evaluation or development board as your ICDI, choose the Stellaris ICDI option from the drop-down box (see [Figure 18](#)).

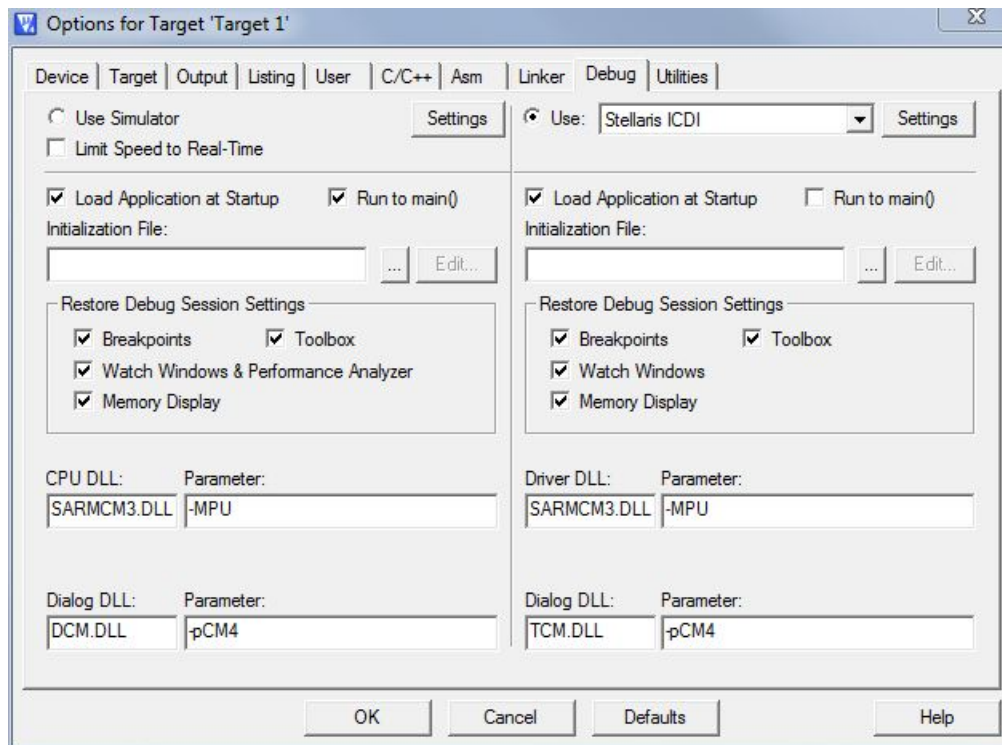


Figure 18.

3.3 Set up Flash Programming

Just as you did in [Section 3.2](#), select the appropriate debug interface for flash programming.

Browse to the Utilities tab of the project options and select the Stellaris ICDI option from the drop-down box (see [Figure 19](#)).

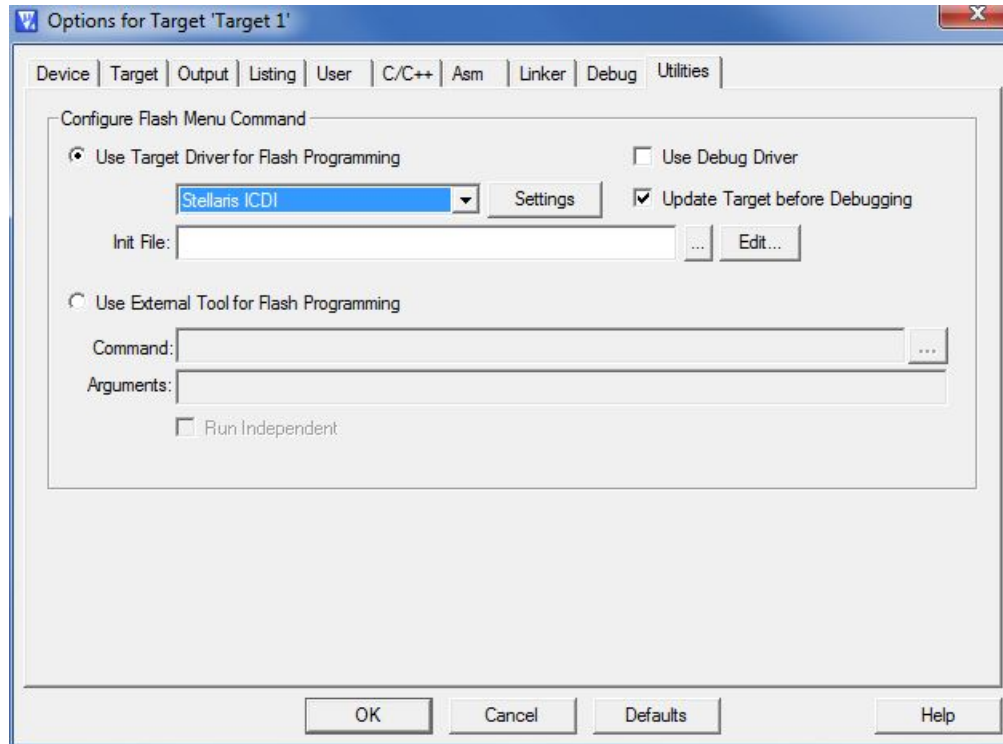


Figure 19.

With these settings complete, you can download and debug your simple application.

3.4 Add TivaWare Hooks

The easiest way to see how to configure your project to use the TivaWare drivers and utilities is to look at the examples. The basic things you will need to do are:

1. Add the driverlib.lib file to your project. This gives the application the entire driver source at compile/link time. To use any of the functions, you must include the appropriate header files.
 - (a) Right-click the Source Group 1 folder.
 - (b) Select Add Existing Files to Group 'Source Group 1.'
 - (c) Browse to `ti\TivaWare_C_Series-n.n\driverlib\rvmdk` and select the driverlib.lib file.

NOTE: You must tell the file browser to look for *.lib file types, so change the Files of type drop-down from C Source file (*.c) to Library file (*.lib) or All files.

2. In your project options, select the C/C++ tab (see [Figure 20](#)). Here you need to tell the project that you are using the ARM compiler, so define rvmdk. This definition is used in TivaWare to compile the Keil specific sections correctly. Define the part you are using as well as add the top-level TivaWare directory to your include path.

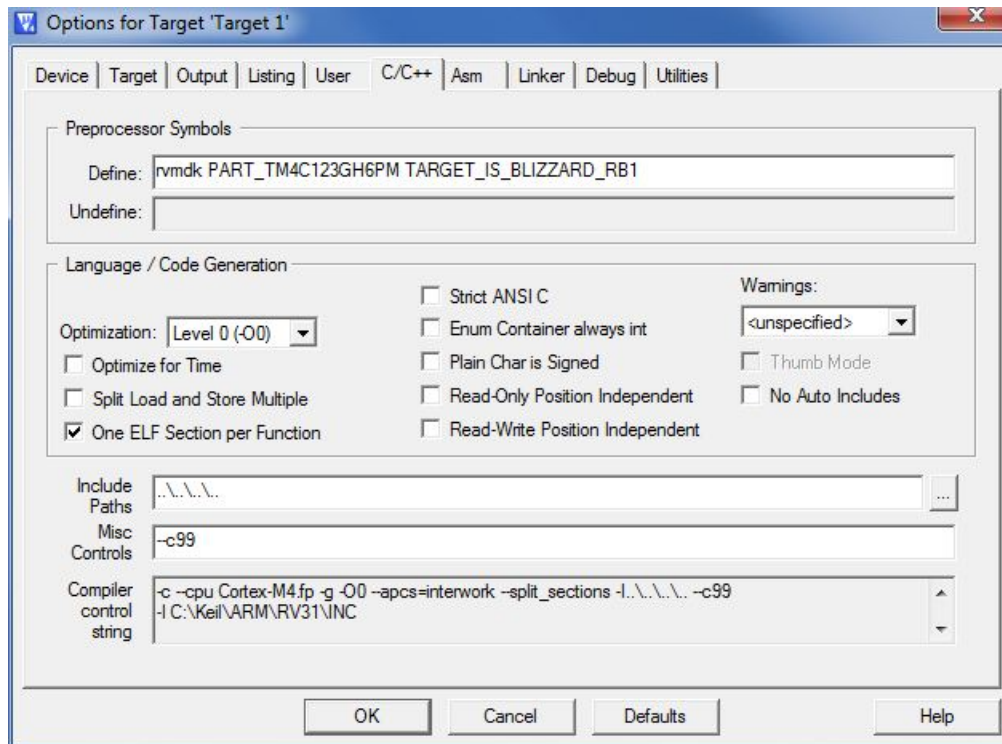
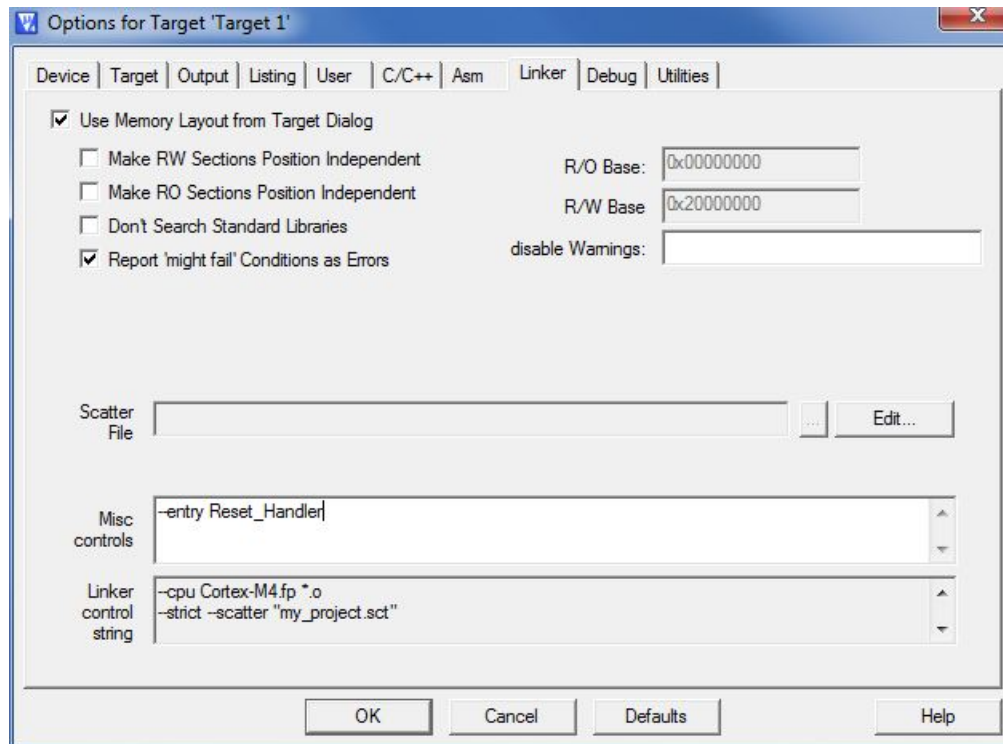


Figure 20.

3. In the linker section, you must add **—entry Reset_Handler** to the Miscellaneous controls box (see [Figure 21](#)). If your startup code changes, adjust this accordingly.

NOTE: The default linker configuration is to use an automatically generated linker/scatter file. If you want to use your own scatter file, uncheck Use Memory Layout from Target Dialog and point the tool to a scatter file.


Figure 21.

These steps get the basic TivaWare hooks into your project. The best way to make sure that you have done everything correctly is to review some of the examples that we provide in the TivaWare package.

4 Conclusion

You have now installed the Keil RealView Microcontroller Development Kit and used it to build, load, and run a demonstration application on your Tiva Evaluation Board. You have also learned how to create a new project. From here, you can experiment with the debugger or create your own application or use the Hello program as an example.

5 References

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

- *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*, Order Number SW-DRL-UG (literature [SPMU298](#)).
- In addition, the following website may be useful: RealView MDK website at <http://www.keil.com/arm/rvmdkkit.asp>

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

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 Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
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
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com