Integrating Silicon Labs SiM3xxxx Devices into the Keil µVision® IDE

1. Introduction

This application note describes how to configure and use the Keil µVision® Integrated Development Environment (IDE) with Silicon Laboratories Precision32™ 32-bit microcontrollers (SiM3xxxx).

2. Key Points

Key points described in this application note include:

- Generating a blank project in Keil µVision
- Configuring a µVision project for use with Silicon Laboratories SiM3xxxx devices
- Using the µVision IDE to build, download, run, and debug a project
- Using the System Viewer Windows and Debug (Print) Viewer

3. Creating a Project

A project is necessary in order to build an example and download the firmware image to the MCU. To create a project in µVision:

1. Under the **Project** menu, select **New µVision Project**. After naming your new project, select **SiLabs SiM3x Devices** in the **Select a CPU Data Base File** dialog and click OK.
2. Expand the Silicon Laboratories data base to open a list of supported MCUs, select the appropriate MCU, and click OK.

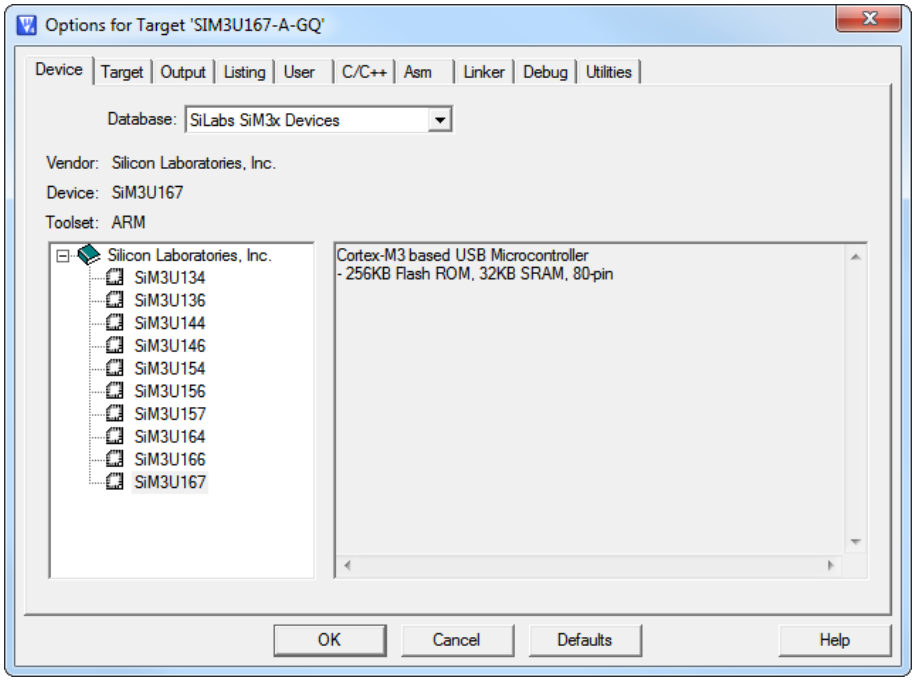


Figure 1. Selecting a SiM3x Device

After creating your blank project, there will be an empty project in the Project Window. The next step is to configure the project options.

4. Configuring Options for Target

Specific configurations are required in order to communicate with the MCU using µVision. Some of the options are preconfigured after selecting a device under the Device tab, but some modifications are required. This section describes the required settings in all of the configuration tabs within the **Project→Options for Target** dialog; tabs that do not require any changes are explicitly noted.

4.1. Target

- Check the **Use MicroLIB** option in the **Code Generation** section.

Note: Adding the correct scatter file in "Section G: Linker" will override the options in Read/Only Memory Areas and Read/Write Memory Areas on this tab.

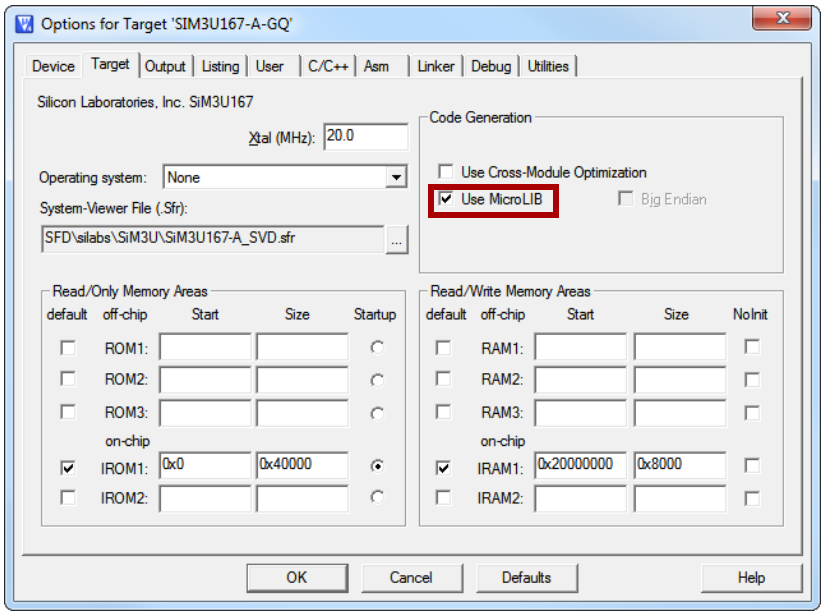


Figure 2. Target Tab

4.2. Output

- No changes needed on this tab.

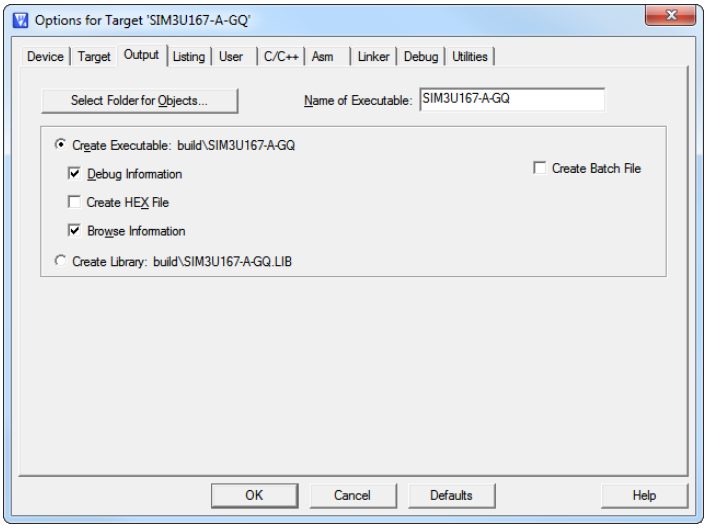


Figure 3. Output Tab

4.3. Listing

- No changes needed on this tab.

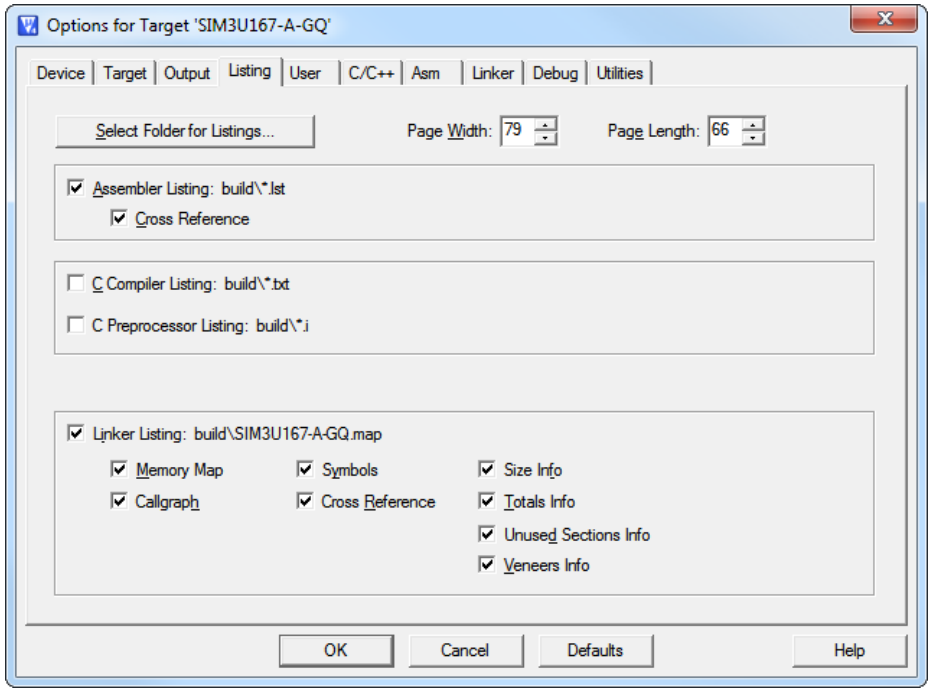


Figure 4. Listing Tab

4.4. User

- No changes needed on this tab.

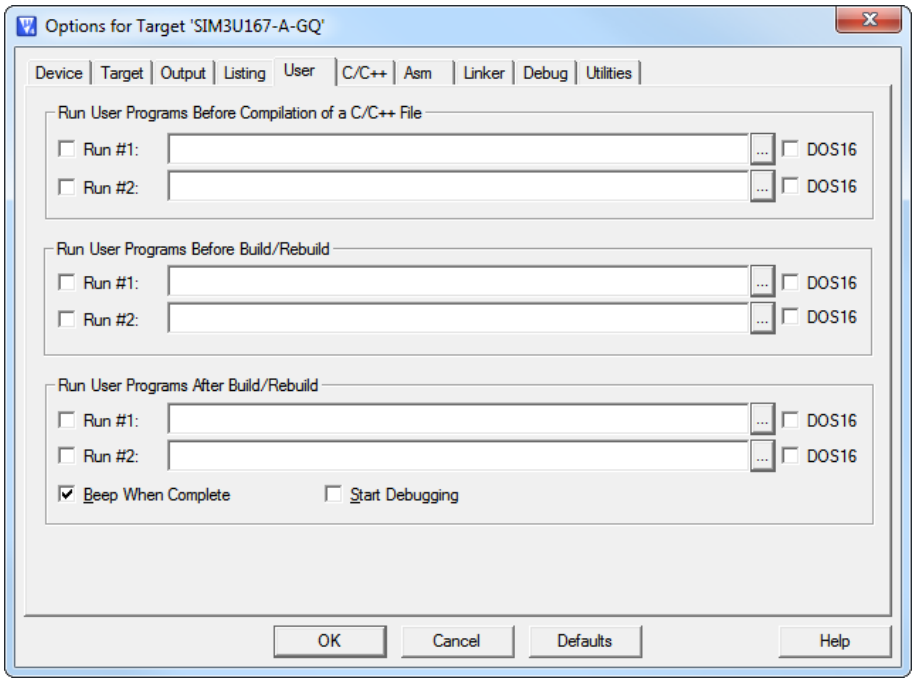


Figure 5. User Tab

4.5. C/C++

- Add **NDEBUG** to the **Define** text box.
- Add include paths by clicking the **"…"** box to the right of the **Include Paths** text box.
- Add **-c99 -gnu** to the **Misc Controls** text box.

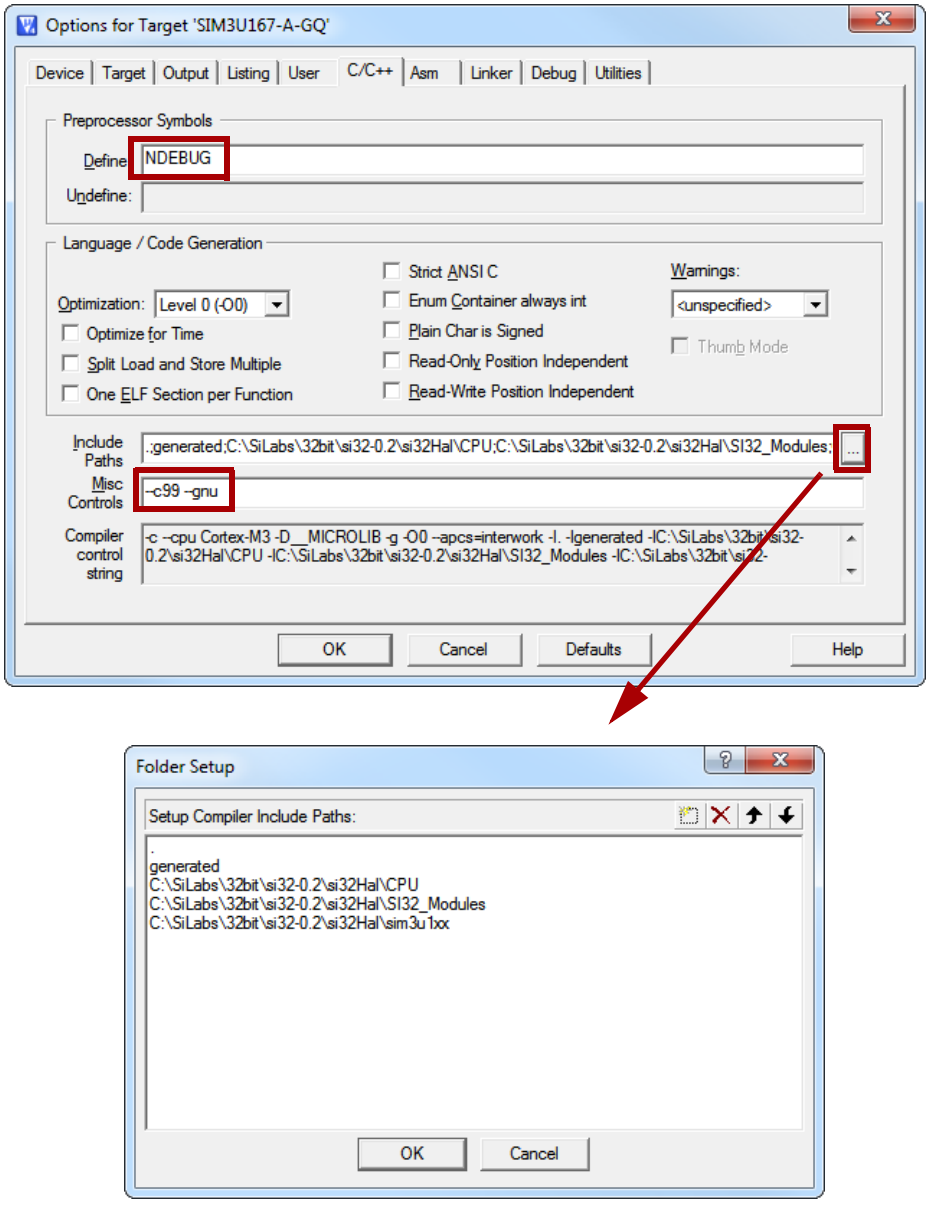


Figure 6. C/C+ and Include Path Tabs

4.6. ASM

- No changes needed on this tab.

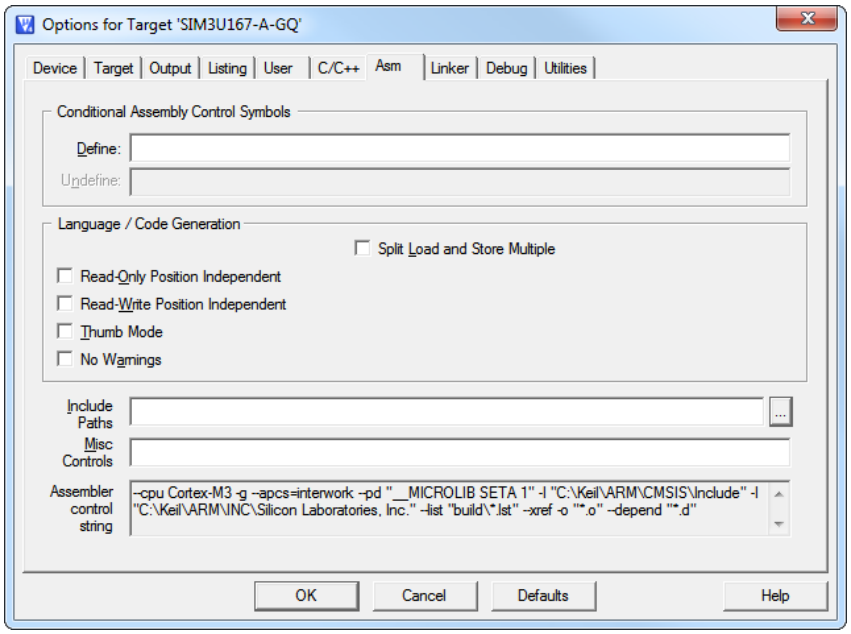


Figure 7. ASM Tab

4.7. Linker

- Uncheck **Use Memory Layout from Target Dialog**.
- Add the correct scatter file by clicking **'...'** to the right of the **Scatter File** text box and selecting the corresponding scatter file.

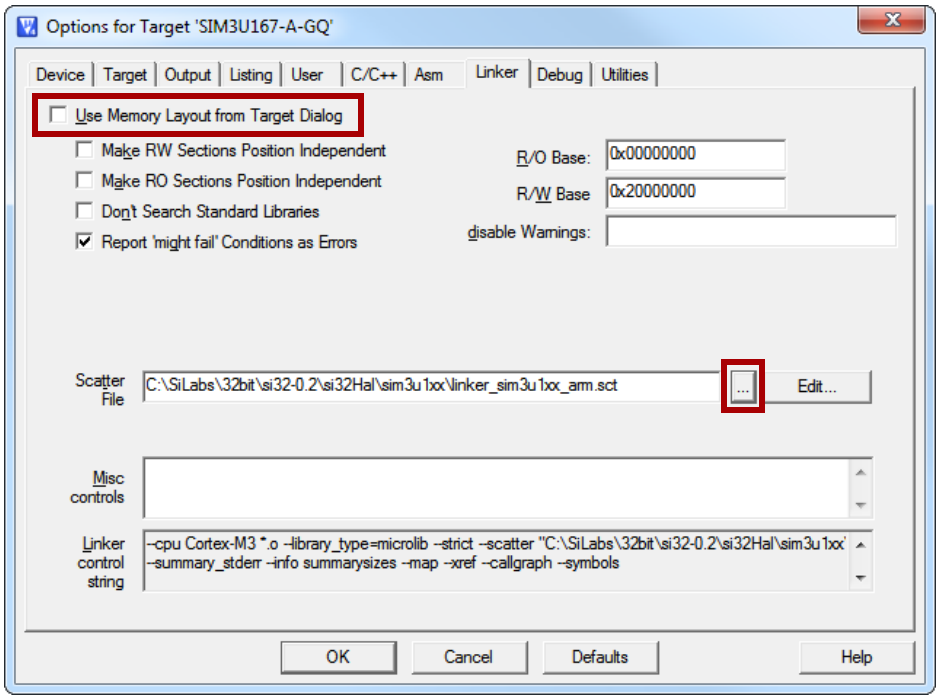


Figure 8. Linker Tab

4.8. Debug

- Select **SiLabs UDA Debugger** in the drop-down menu on the top right-hand side and select **Use**.

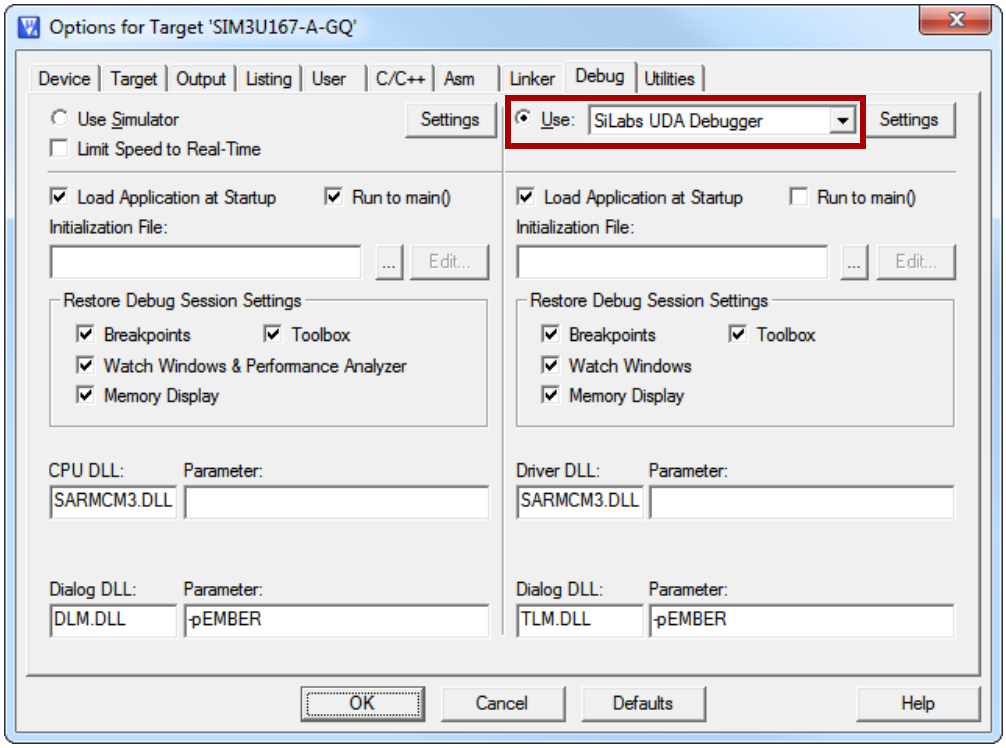


Figure 9. Debug Tab

4.9. Utilities

- Select **SiLabs UDA Debugger** in the drop-down menu under the **Use Target Driver for Flash Programming**.
- Check the **Update Target before Debugging** option.

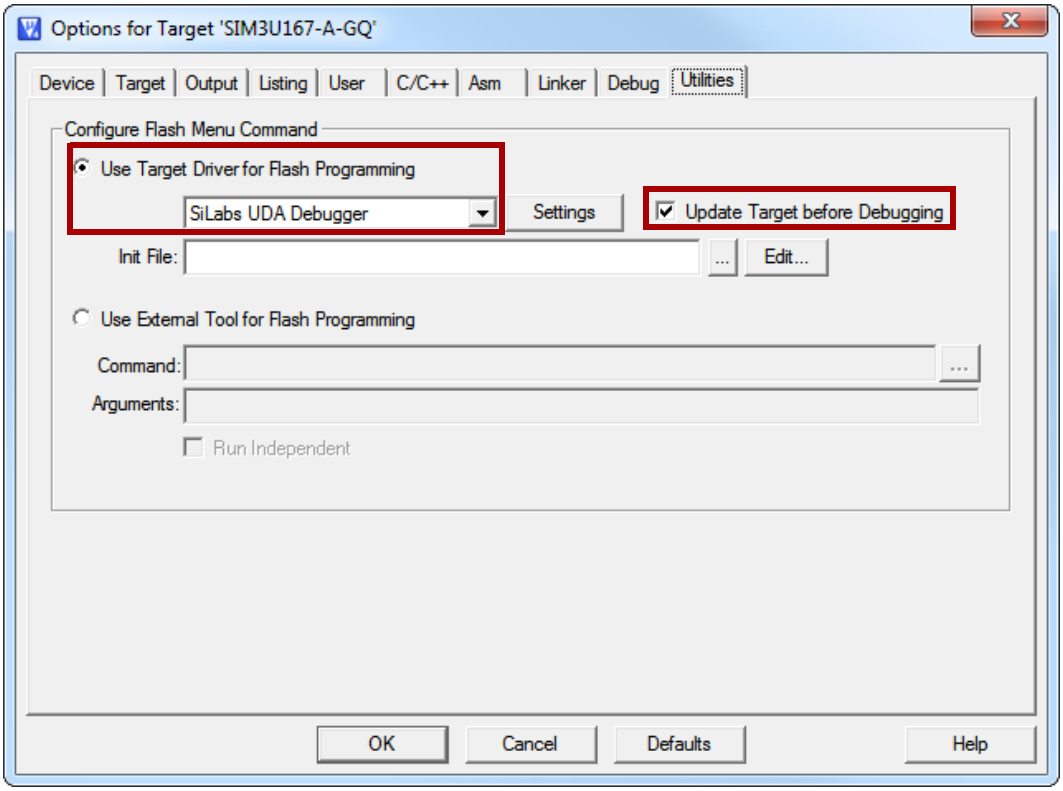


Figure 10. Utilities Tab

4.10. UDA Debug Adapter Settings

The UDA Debug Adapter options can be accessed by clicking the **Settings** button on the right-hand side of **SiLabs UDA Debugger** on the **Utilities** or on the **Debug tab**.

4.10.1. Debug

- Verify that the correct USB Debug Adapter is select in the **Unit** dropdown menu. For more information on the **Debug** section of the tab, see the µVision help. In µVision, select **Help→µVision Help**. On the **Contents** tab of the µVision help file, select ULINK2 User's Guide→Setup ULINK2→Configure µVision for Debugging→Cortex-M Debugging→Debug Driver Configuration→Debug.

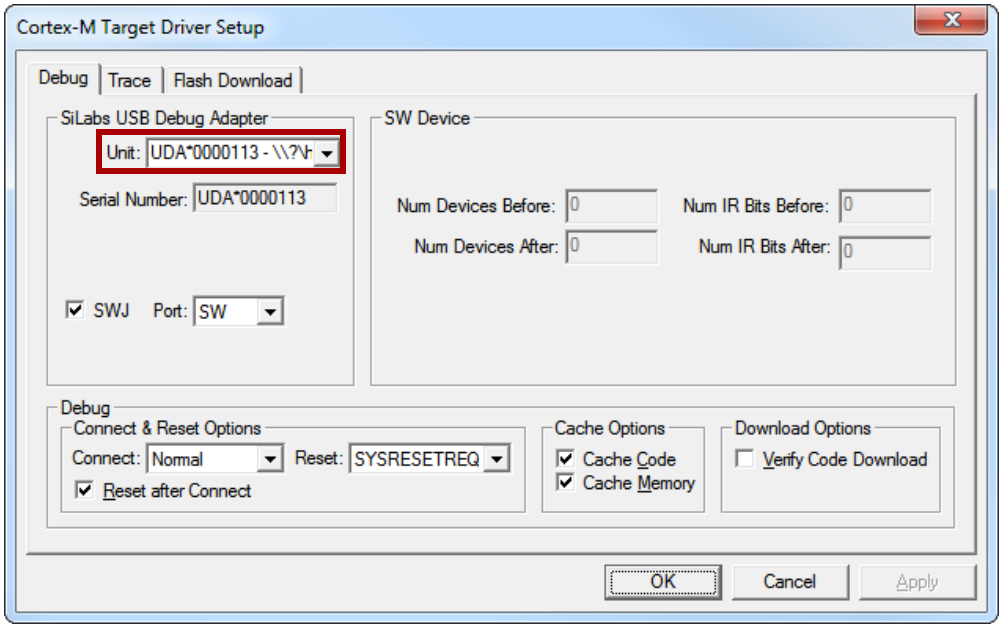


Figure 11. Debug Adapter Settings—Debug Tab

4.10.2. Trace

- If any of the trace tools will be used (including the Debug (Printf) Viewer), check the **Trace Enable** box. In addition, the **Core Clock** setting must be correct in order to use any of the trace tools. For more information on the trace options, see the µVision help. In µVision, select **Help→µVision Help**. On the Contents tab of the µVision help, select ULINK2 User's Guide→Setup ULINK2→Configure µVision for Debugging→Cortex-M Debugging→Trace Configuration.

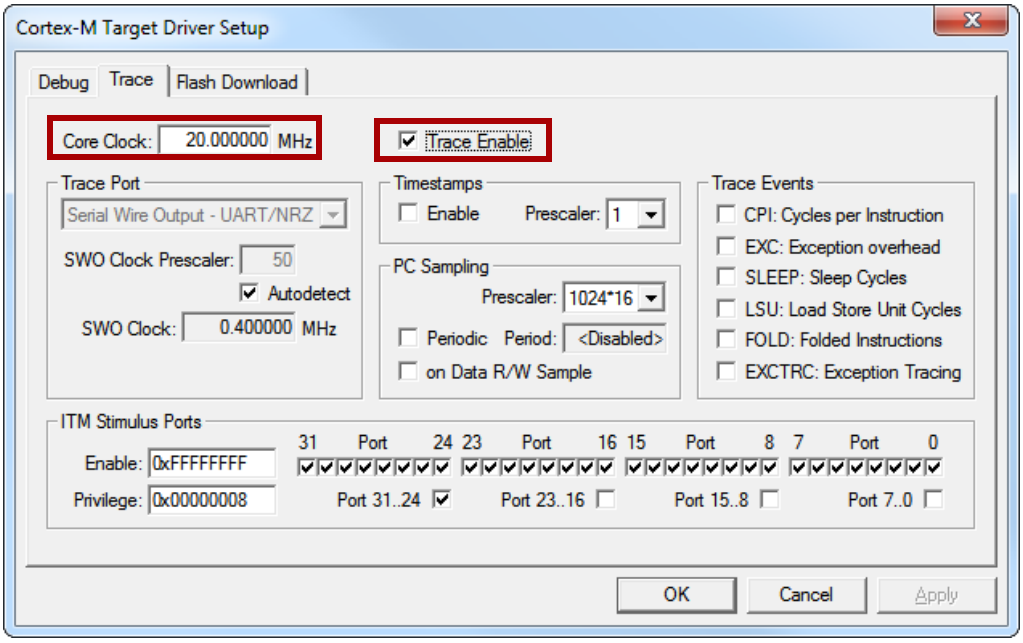


Figure 12. Debug Adapter Settings—Trace Tab

4.10.3. Flash Download

- A Flash download algorithm must be selected to download a code image to the MCU. The Flash programming algorithm temporarily resides in RAM and is used by the Debug Adapter to program the Flash.
- On the **Flash Download** tab, click **Add** and select the appropriate SiM3x algorithm. Next, click **Add** and then click OK.

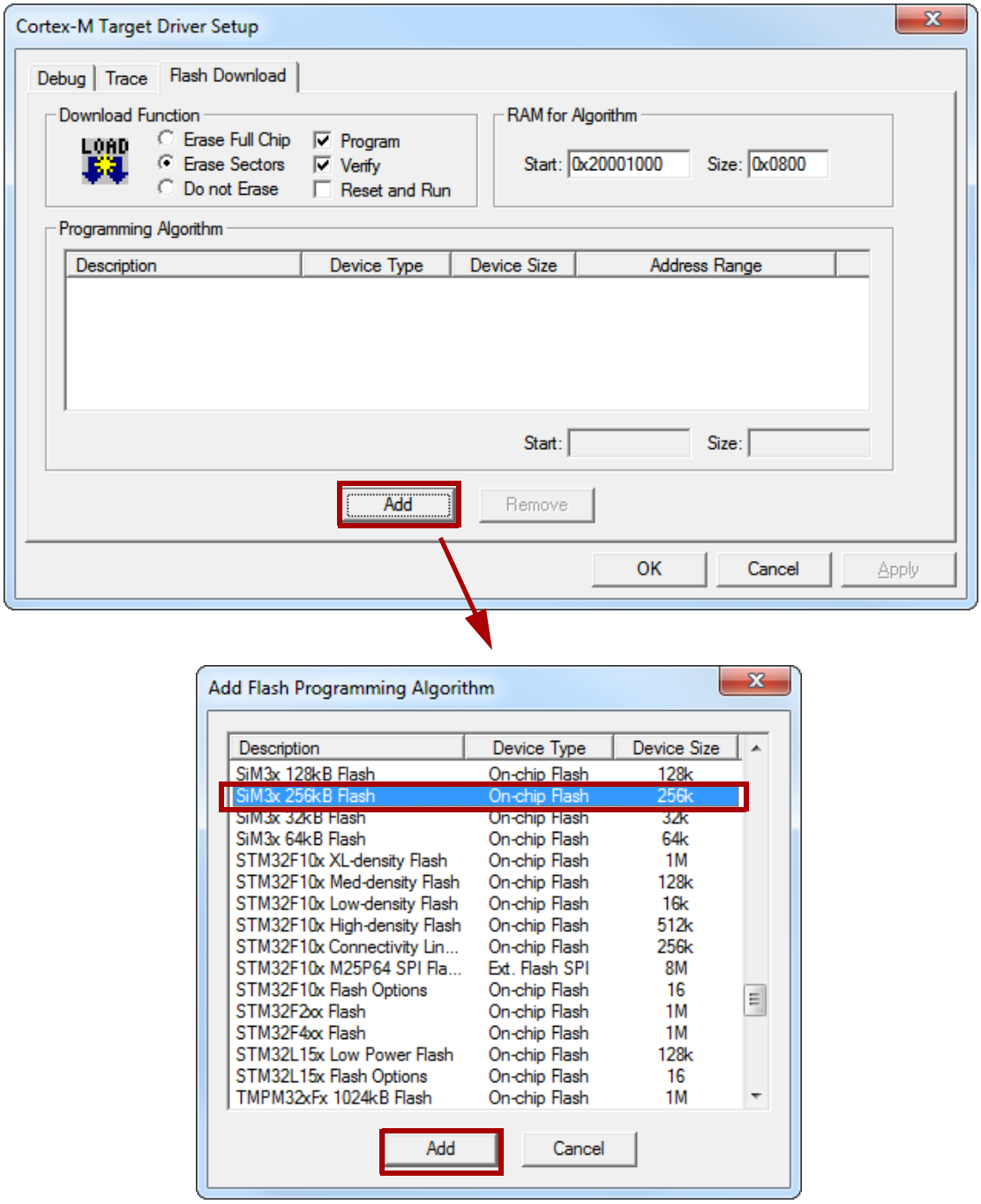


Figure 13. Debug Adapter Settings—Flash Download Tab

5. Managing Files Within the Project

To add files to a project, add a Group in the project tree structure first. To add a Group, right-click on the root folder in the **Project Window** and select **Add Group**. After adding a Group, add files to the group by right-clicking on the **Group** and select **Add Files to Group**. To modify the overall project structure or file ordering, right-click on the root folder in the **Project Window** and select **Manage Components**.

6. Building the Project

Build a project by selecting the **Rebuild** icon in the **Build Toolbar** or by selecting **Project→Rebuild all target files**. To compile individual files, click the **Translate** icon in the **Build Toolbar** or select **Project→Translate**. Any build-related errors or warnings will appear in the **Build Output** window. If the Build Output window is not visible, select **View→Build Output Window**.

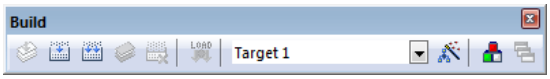


Figure 14. Build Toolbar

7. Running the Code and Debugging

After successfully building a project, clicking the **Start/Stop Debug Session** icon or selecting **Debug→Start/Stop Debug Session** will initiate a debug session. If the project has been changed in any way since the last build, starting a debug session will initiate a download of the updated firmware image to the device. The µVision tools will not automatically download a firmware image when initiating a debug session if the IDE did not detect any changes. If this behavior is not desired, clicking the **Download** icon or selecting **Flash→Download** will manually initiate a Flash download.

After the Flash download, initiate a debug session to start debugging. During a debug session, **Run**, **Step Into**, **Step Over**, **Stop**, or **Reset** the code by clicking the corresponding button in the **Debug Toolbar** or in the **Debug Menu**. To stop a debug session, click the **Start/Stop Debug Session** icon or select **Debug→Start/Stop Debug Session**.

Figure 15. Debug Toolbar

Add variables to the **Watch Window** or **Memory Window** by right-clicking on a variable, selecting **Add to…→Watch** or **Add to…→Memory**. The µVision IDE has two Watch Windows and four Memory Windows. To view a specific memory location, open a memory window by selecting **View→Memory Windows**. After opening a Memory Window, manually enter a location in the **Address** box. This will also work when viewing the contents of Special Function Registers (SFRs).

8. System Viewer

The **System Viewer Window** allows users to easily view and modify SFRs of the MCU. Each modifiable field within the System Viewer Window contains a field description, which can also be helpful for debugging. To open a peripheral's system viewer window, select **View→System Viewer→(Select Peripheral)**. After selecting the desired peripheral, the **System Viewer Window** will appear. Bit fields that changed while stepping through code will be highlighted in green in the System Viewer Window. Figure 16 shows an example of the Clock Control (CLKCTRL) System Viewer Window.

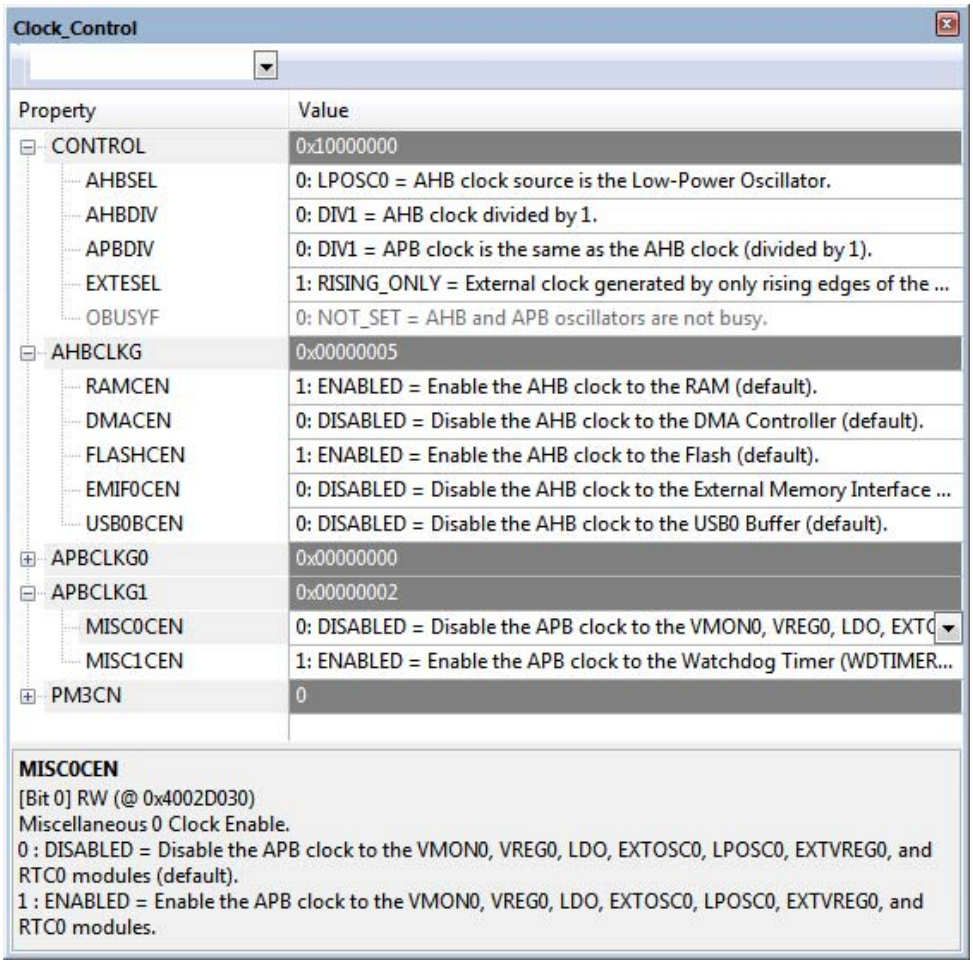


Figure 16. Clock Control System Viewer Window

Note: Individual peripheral bit-fields cannot be modified if the clock control bit for the peripheral is disabled. Before manually modifying bits in the System Viewer Window for a peripheral, enable the corresponding clock enable bit for the peripheral.

9. Debug (Printf) Viewer

The debug printf function allows for firmware to use printf() functions to print characters or strings to the **Debug (Printf) Viewer**. The Debug Viewer uses the Serial Wire Output (SWO) pin of the MCU to send data back to the µVision IDE. To use the Debug Viewer:

- Configure the SWO port pin to push-pull mode (this is done in firmware).
- Select the **Trace Enable** option in the Trace tab within SiLabs UDA Debugger settings (this is done in the project settings; see "4.10.2. Trace" on page 9 for more information).
- Configure the **Core Clock** in the Trace tab within the SiLabs UDA Debugger settings (this is done in the project settings; see "4.10.2. Trace" on page 9 for more information).
- Open the **Debug (Printf) Viewer** window within the µVision IDE. To open the Debug Viewer, select **View→Serial Windows→Debug (Printf) Viewer**.

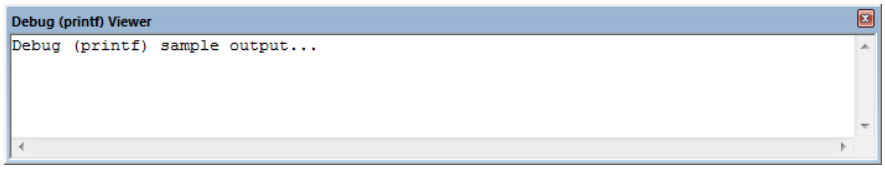


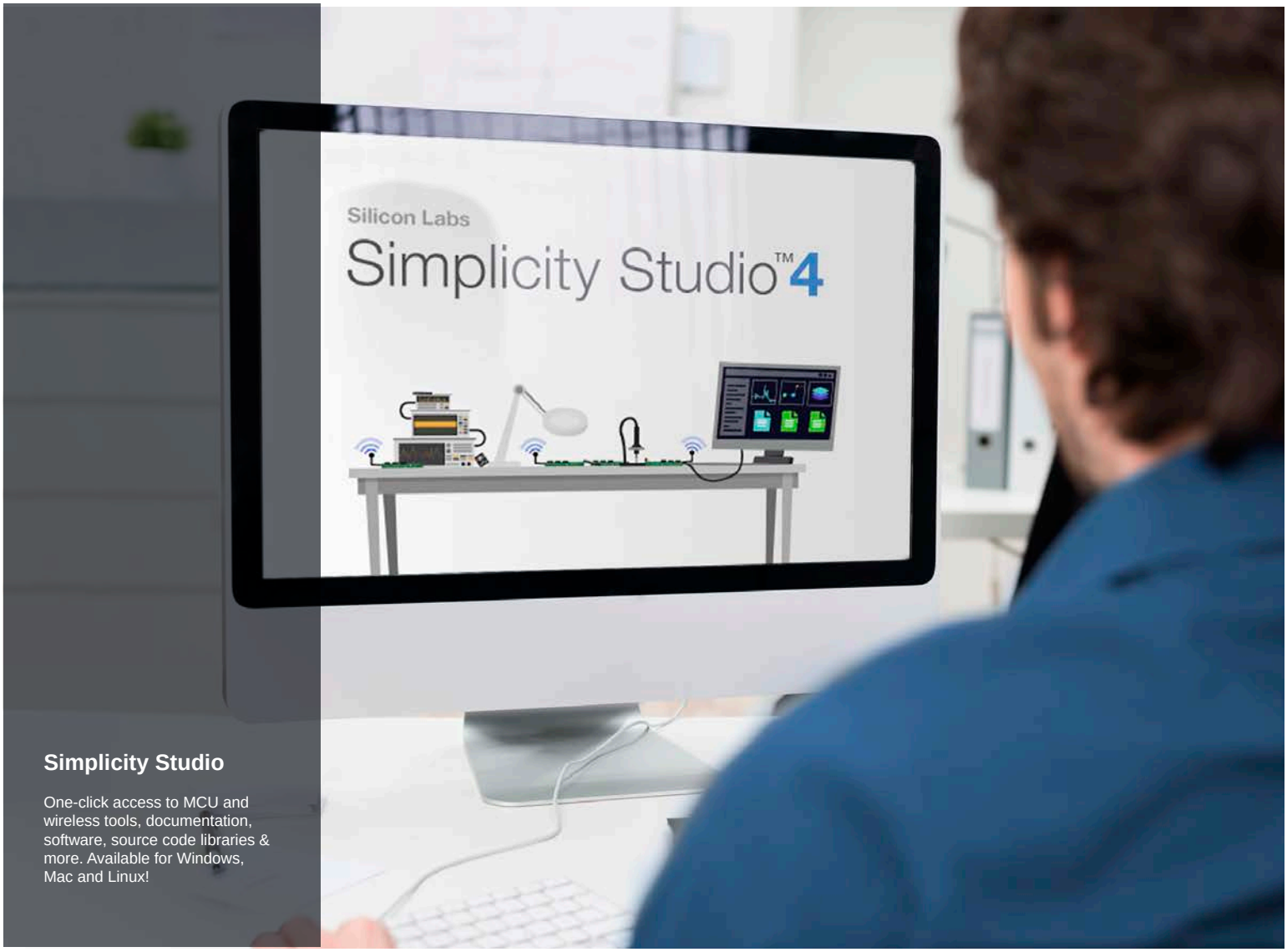
Figure 17. Debug (Printf) Viewer Window

10. Evaluation Tools

An evaluation version of the Keil µVision tool is available for download at https://www.keil.com/demo/eval/arm.htm. The free evaluation tools, called MDK-Lite, are 32 kB limited (debugger, compiler, and assembler-limited). More information on MDK-Lite can be found at http://www.keil.com/arm/selector.asp.

11. Code Examples and Source/Include Files

Examples and source/include files for each of the Silicon Labs devices are installed with the Precision32 IDE. The default location for these examples is **C:\Silabs\32bit\si32-x.y\Examples**, where x is the major revision number and y is the minor revision number. Within each example folder, the folder labeled **ARM** contains the µVision project files. The **src** folder contains the source files for each example.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com