
INTEGRATING KEIL 8051 TOOLS INTO THE SILICON LABS IDE

1. Introduction

This application note describes how to integrate the Keil 8051 Tools into the Silicon Labs IDE (Integrated Development Environment). Integration provides an efficient development environment with compose, edit, build, download and debug operations integrated in the same program.

The latest version of the IDE and the Keil C51 Tools can be downloaded from the Silicon Labs website as part of the 8-bit microcontroller studio package: www.silabs.com/8bit-software. More information about the Keil tools and their options can be found on the Keil website: www.keil.com/support.

2. Key Points

- The Intel OMF-51 absolute object file generated by the Keil 8051 tools enables source-level debug from the Silicon Labs IDE.
- Once Keil Tools are integrated into the IDE, they are called by simply clicking the **Assemble/Compile Current File** button or the **Build/Make Project** button.
- See the `...\\Silabs\\MCU\\Examples` directory for examples created for use with the Keil tools.
- Information in this application note applies to Version 1.90 and later of the Silicon Labs IDE and the Keil 8051 Tools included with Silicon Labs kits.
- Evaluation versions of the Keil 8051 toolset included with some Silicon Labs kits are initially limited to a code size of 2 kB and programs start at code address 0x0800. To upgrade the Keil tool limitations to a time and code unlimited version, see Section 9. "Registering the Evaluation Keil Toolset," on page 7.

3. Create a Project in the Silicon Labs IDE

A project is necessary in order to link assembly files created by the compiler and build an absolute **OMF-51** output file. Follow these steps to create a project:

1. Under the **Project** menu, click **New Project...** Select a device family or click **Cancel** to start with an entirely blank project.
2. Select **Project**→**Add Files to Project...** or right-click on **New_Project** in the **Project Window** and select **Add files to project New_Project**.
3. Select the desired C source files and click **Open**. Continue adding files until all project files have been added.
4. To add files to the build process, right-click on the file name in the **Project Window** and select **Add filename to build**.
5. Under the **Project** menu, select **Save Project As...** Enter a project workspace name and click **Save**.

4. Configure the Tool Chain Integration Dialog

Under the **Project** menu select **Tool Chain Integration** to bring up the dialog box shown below. First, select **Keil** from the **Preset Name** drop down list. Next, define the Keil assembler, compiler, and linker as shown in the following sections. The executable paths listed are the default locations for the evaluation version of the Keil tools included with Silicon Labs development kits.

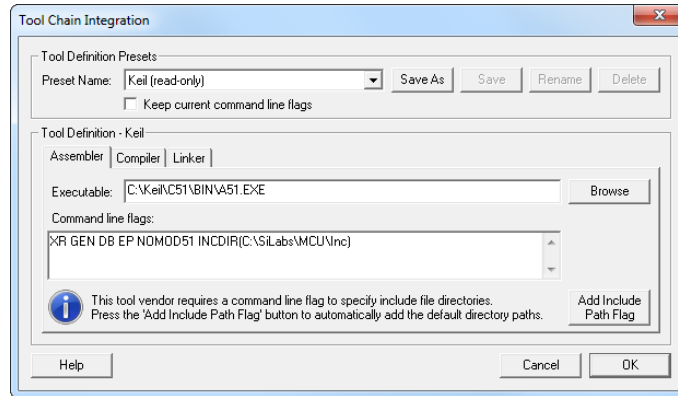


Figure 1. IDE Tool Chain Dialog

4.1. Assembler Definition

The **Keil (read-only)** option in the Tool Definition Presets box will automatically populate the Assembler tab with the default options. To modify these options from the default:

1. Under the **Assembler** tab, click the browse button next to the **Executable** text box and locate the assembler executable. The default location for the Keil assembler is **C:\Keil\C51\BIN\A51.exe**.
2. Enter any additional command line flags directly in the **Command Line Flags** box.
3. See the following figure for the **Assembler** tab with the default Keil settings.

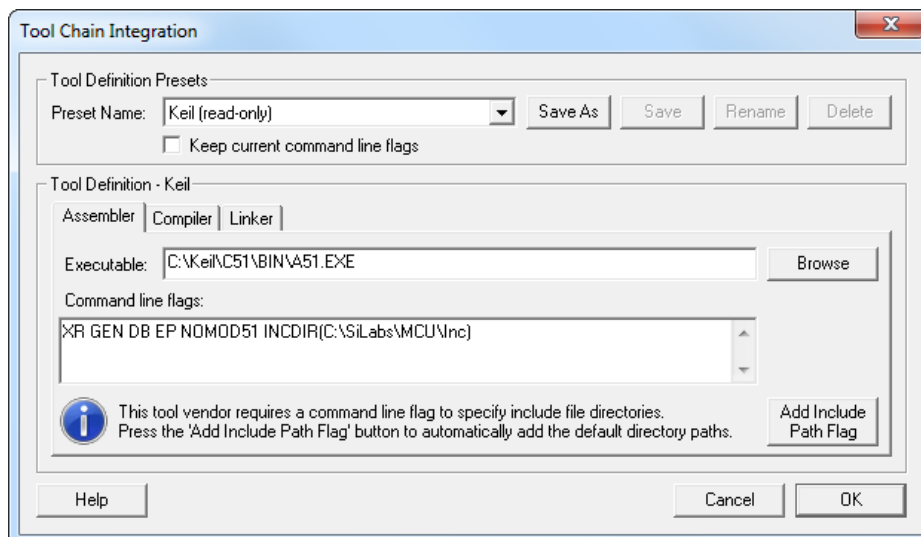


Figure 2. IDE Tool Chain Dialog—Assembler Tab

4.2. Compiler Definition

The **Keil (read-only)** option in the Tool Definition Presets box will automatically populate the **Compiler** tab with the default options. To modify these options from the default:

1. Under the **Compiler** tab, if the compiler executable is not already defined, click the browse button next to the **Executable** text box and locate the compiler executable. The default location for the Keil compiler is **C:\Keil\C51\BIN\C51.exe**.

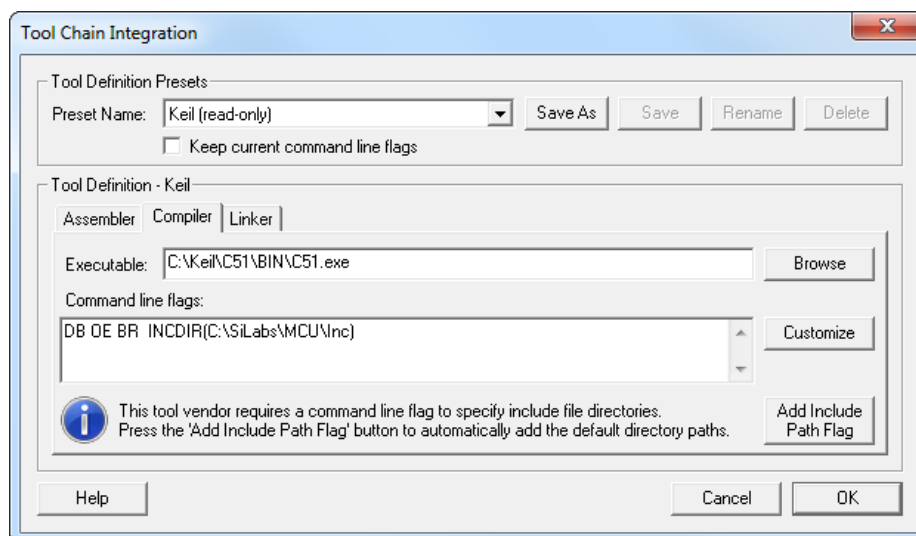


Figure 3. IDE Tool Chain Dialog—Compiler Tab

2. Enter any additional command line flags directly in the **Command Line Flags** box or click on the **Customize** button to display the dialog box with additional options. To enable source-level debugging **Include debug information** and **Include Extended debug information** should be selected.

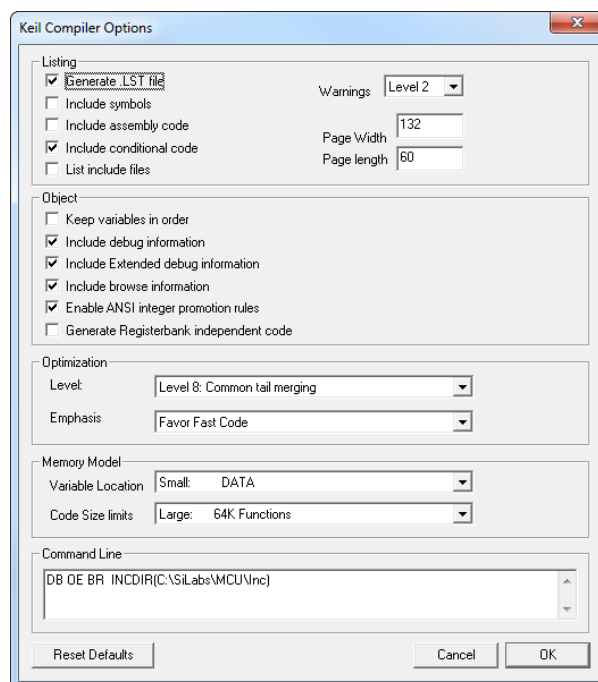


Figure 4. IDE Tool Chain Dialog—Compiler Customization Options

4.3. Linker Definition

The **Keil (read-only)** option in the Tool Definition Presets box will automatically populate the **Linker** tab with the default options. To modify these options from the default:

1. Under the **Linker** tab, if the linker executable is not already defined, click the browse button next to the **Executable** text box, and locate the linker executable. The default location for the Keil linker is **C:\Keil\C51\BIN\BL51.exe**.

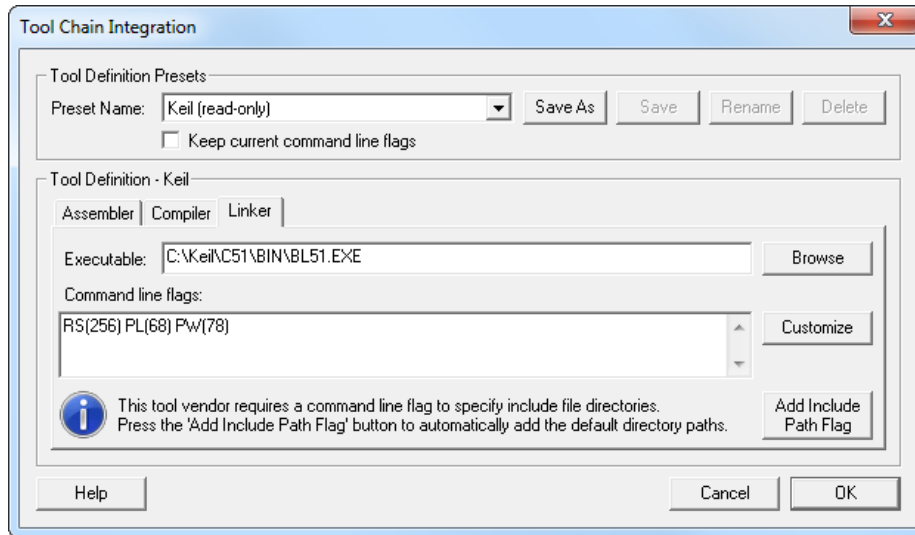


Figure 5. IDE Tool Chain Dialog—Linker Tab

2. Enter any additional command line flags directly in the **Command line flags** box or click on the **Customize** button to display additional options.

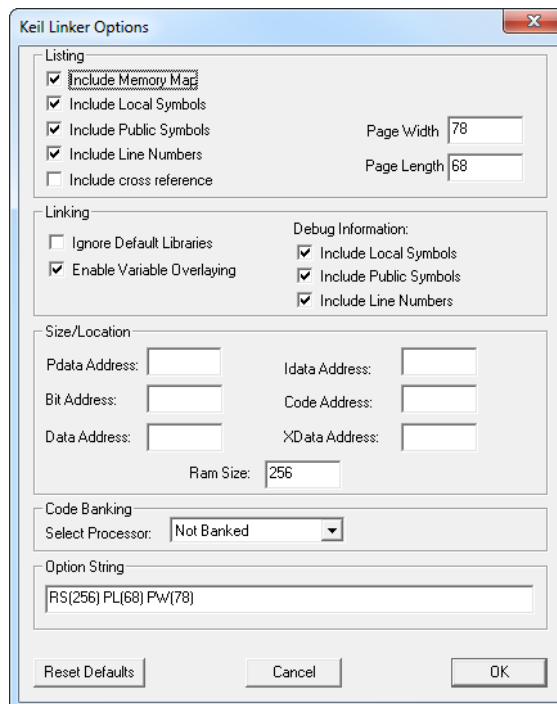


Figure 6. IDE Tool Chain Dialog—Linker Customization Options

5. Target Build Configuration

Under the **Project** menu, select **Target Build Configuration** to bring up the detailed build configuration settings dialog.

The **Target Build Configuration** dialog has several options:

- To customize a default filename or to create a new filename, click the browse button next to the **Absolute OMF file name** edit box. Select a path and enter an output filename with no extension (ex. blinky).
- Define the project build configuration by clicking the **Define Build Process Customize** button. More information can be found in Section 5.1.
- Generate a HEX file output automatically after a successful build by enabling the **Generate hex file** check box. This option calls the object-to-hex file converter specified by the **Generation Executable** field.
- Automatically save all files when the **Build/Make project** is pressed by enabling the **Enable automatic save for project files before build** check box.
- Automatically download the firmware image to the target when the **Build/Make project** button is pressed by enabling the **Enable automatic connect/download after build** check box.
- Halt the target on the first line in main() when the **Download code** button is pressed by enabling the **Run to main() on code download** check box.

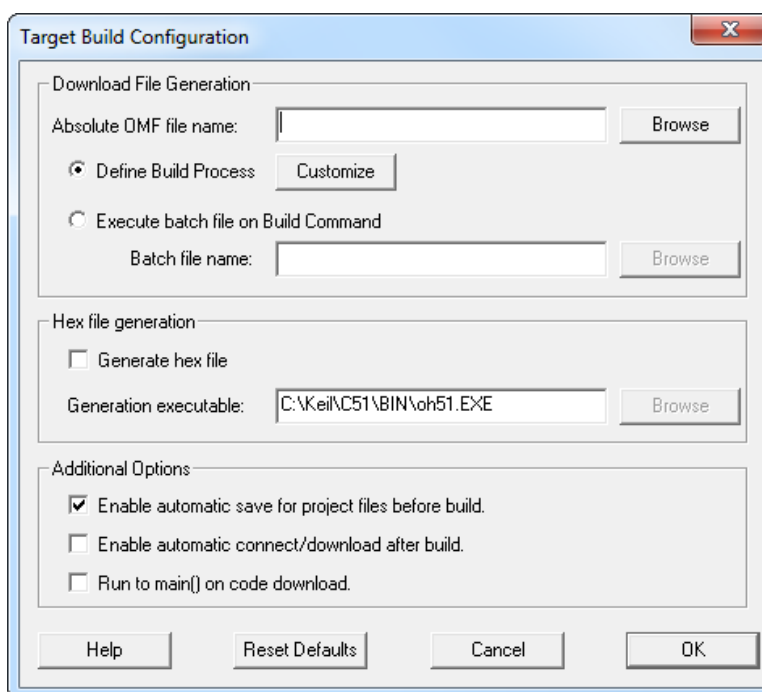


Figure 7. IDE Target Build Configuration

5.1. Project Build Definition

Click the **Define Build Process Customize** button to bring up the **Project Build Definition** dialog. The IDE creates default assemble and compile selections based on the files included in the build using the **Project Window**. The build process customization dialog allows selection of the additional files to be included in the build process, like libraries or assembly files. Under each tab, add files to assemble or compile by selecting the desired file and clicking the **Add** button. To add an external library, click the **Files to Link** tab and click the **Add External OBJ** button. Files are removed in the same manner.

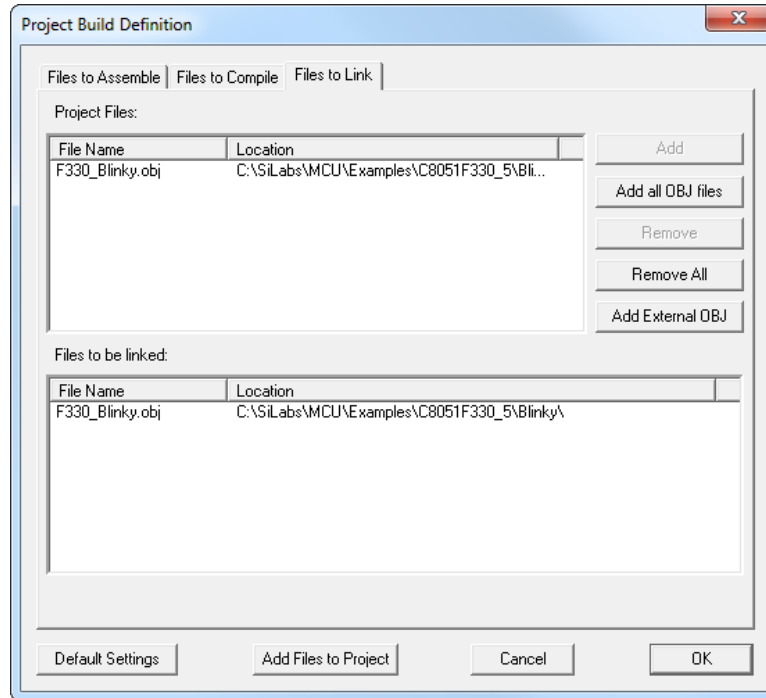


Figure 8. IDE Project Build Definition

6. File Backups

When saving an edited file, the IDE will automatically save the previous revision as a backup file. Backups are saved as the name of the file with the extension #1, #2, #3, and so on up to the number of backups (N) selected by the **Options**→**File Backup Setting...** dialog. The file ending in #1 is the most recent backup, and the file ending in N is the least recent backup.

7. Building the Project

To build the project:

- Click the **Assemble/Compile current file** button to compile just the current open and selected file.
- Click the **Build/Make project** button to compile and link all the files in the project.

After building the file or project, review the errors and warnings generated during the build process located in the **Build** tab of the **Output** window typically found at the bottom of the IDE. Double-clicking on an error that is associated with a line number will automatically move the cursor to the line number in the source file that generated the error.

8. Source/Include File Examples

Example source files and include files for each of the Silicon Labs devices have been created for use with the Keil tools. The default location for these examples is **C:\Silabs\MCU\Examples**. In addition, examples can be found in many Silicon Labs Application Notes, which are available at www.silabs.com/appnotes.

9. Registering the Evaluation Keil Toolset

The Silicon Labs 8-bit microcontroller studio package includes the latest version of the evaluation Keil 8051 tools. These tools are initially limited to a code size of 2 kB and programs start at code address 0x0800. After registration, the code size limit is removed entirely (no time limit) and programs will start at code address 0x0000. The Silicon Labs version of the PK51 Keil tools:

- Can be used for commercial applications.
- Includes μ Vision4 with build-support only (no debugging) and can only be used with Silicon Labs devices. Debugging can be accomplished with the Silicon Labs IDE.

To register the Keil tools:

1. Fill out a form to get the FREE product key <https://pages.silabs.com/lp-keil-pk51.html>

Silicon Labs » Products

8-bit Microcontroller Studio

FREE Keil® PK51 Developer's Kit Available for Silicon Labs' 8-bit MCUs and Studio.

Silicon Labs now offers a free version of the Keil PK51 Professional Developer's Kit, including the compiler/linker/assembler for use with Silicon Labs' 8-bit microcontrollers and Studio. This free tool comes without a time or code size limit!

Simply follow the steps below to get your free key and unlock the tools:

- Download the [8-bit microcontroller studio](#) and install the Keil C51 tools.
- Complete the form on the right to get a FREE product key.
- Unlock your Keil tools using the steps described in [Application Note 104](#).

Get Your Key!

Simply fill out this short form to get your FREE key

* Email Address:

* First Name:

* Last Name:

* Company Name:

* Phone Number:

* Country:

* Postal Code:

* Target MCU:

Figure 9. Obtain Keil PK51 Professional Developer's Kit Key

2. Open the Keil μ Vision4 IDE from the installation directory with administrative privileges.

3. Select **File**→**License Management** to open the License Management window.

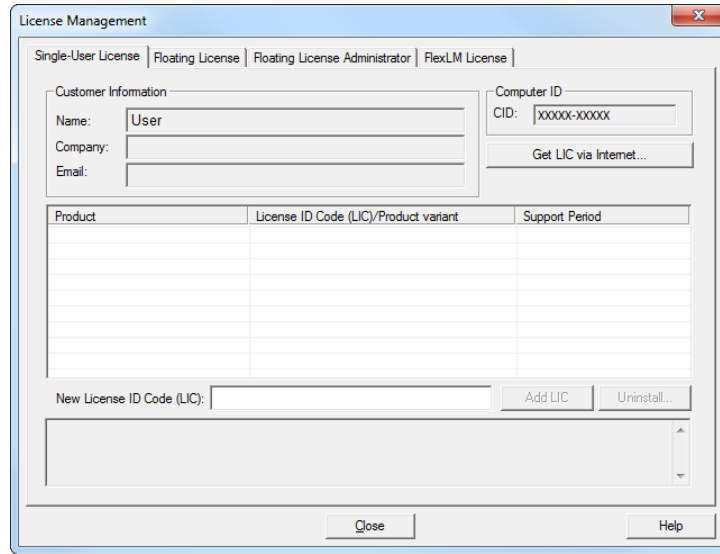


Figure 10. Keil μVision4 IDE License Management Window

4. Click on the **Get LIC via Internet...** button to open the Obtaining a License IDE Code (LIC) window.

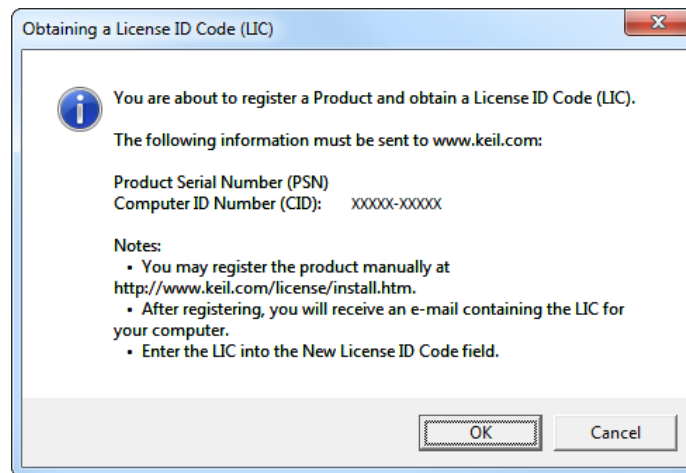


Figure 11. Keil μVision4 IDE License Management Window

5. Press **OK** to open a browser window to the Keil website. If the window doesn't open, navigate to www.keil.com/license/install.htm.
6. Enter the Silicon Labs **Product Serial Number** printed on the CD-ROM or the **Product Key** from the website, along with any additional required information.

The screenshot shows a web browser window with the URL <https://www.keil.com/license/install.htm?C=CESXM-RSUFM&E=taparker@silabs.com&F=Tabitha&L=Parker&O=Silicon>. The page features the Keil logo and the ARM logo. The main heading is "Single-User License". Below the heading, there is a form titled "Enter Your Contact Information Below" with the following fields:

- Computer ID (CID):
- Product Serial # (PSN):
- PC Description: (with a note: "Enter a description of the PC on which this license is registered. For example: LAB PC, Office Computer, Laptop, John's PC, etc.")
- First Name:
- Last Name:
- E-mail:
- Company:
- Address:
- City:
- State/Province:
- Zip/Postal Code:
- Country:
- Phone:
- My job is:

There is also a "Read More" link and an "accept and hide this message" button in the bottom left corner of the form area.

Figure 12. Keil Registration Website

7. Once the form is complete, click the **Submit** button. An email will be sent to the provided email address with the license activation code.
8. Copy the License ID Code (LIC) from the email.

Thank you for licensing your Keil product. Your License ID Code (LIC) is printed below. Print a copy of this e-mail to save with your installation CD.

PK51 Professional Developer's Kit
For Silicon Labs Devices Only
Support Ends 31 Mar 2016

PC Description : XXXXXXX
Computer ID (CID): XXXXX-XXXXX

License ID Code (LIC): XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX

To activate your Keil product, copy the License ID Code (LIC) and paste it into the New License ID Code input field on the Single-User License Tab in the uVision4 License Manager Dialog (available from the File menu).

*** DO NOT REPLY TO THIS EMAIL: For licensing problems or questions, please contact Keil Technical Support.

Thank You,
Technical Support

Figure 13. Keil Registration Email

AN104

9. Paste the LIC into the **New License ID Code (LIC)** text box at the bottom of the License Management window in μ Vision4.
10. Press the **Add LIC** button. The window should now list the **PK51 Prof. Developers Kit for Silabs** as a licensed product.

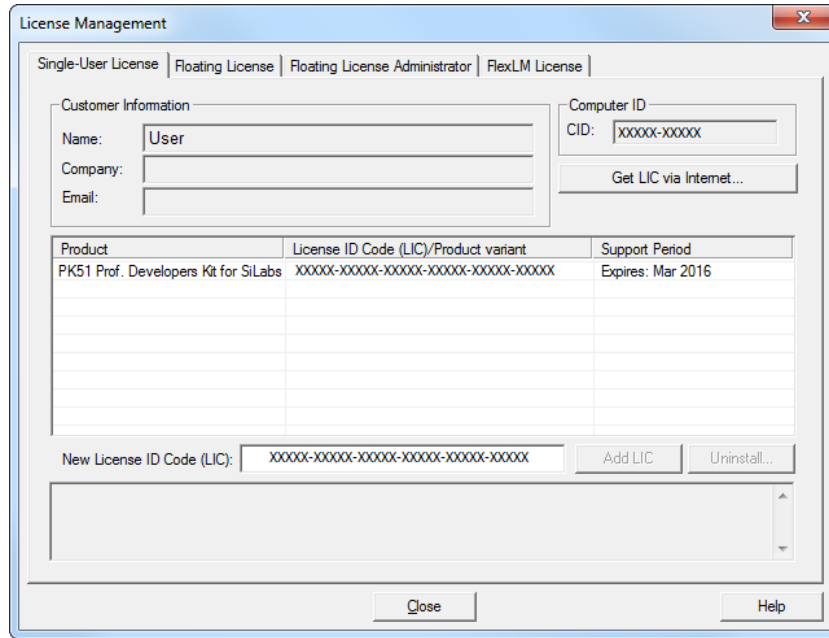


Figure 14. Adding the LIC to μ Vision4

11. Click the **Close** button.

DOCUMENT CHANGE LIST

Revision 2.4 to Revision 2.5

- Introduction updated.
- Example path updated from C:\Cygnaal\Examples to C:\Silabs\MCU\Examples.
- Target Build Configuration and Project Build Definition windows screenshots updated to reflect the new examples path.
- Key Points updated to include Silicon Labs and Keil tools version information.

Revision 2.5 to Revision 2.6

- Section "Registering the Evaluation Keil Toolset" added.
- Bullet added to Section "Key Points" referring to new Section.

Revision 2.6 to Revision 2.7

- Instructions added to retrieve Keil Serial Number from ToolStick kit boxes.
- Instructions updated due to Keil evaluation tools no longer being included with the installation of the Silicon Labs IDE.

Revision 2.7 to Revision 2.8

- Corrected link to Keil website in Section 9. "Registering the Evaluation Keil Toolset," on page 7.

Revision 2.8 to Revision 2.9

- Updated screenshots.
- Added instructions for new Tool Chain presets.

Revision 2.9 to Revision 3.0

- Updated screenshots.
- Added instructions for new registration process.

Revision 3.0 to Revision 3.1

- Updated new instruction to obtain Keil PK51 Professional Developer's Kit key on page 7.

Smart. Connected. Energy-Friendly.



IoT Portfolio
www.silabs.com/products



Quality
www.silabs.com/quality



Support & Community
www.silabs.com/community

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and “Typical” parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A “Life Support System” is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.

Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information

Silicon Laboratories Inc.[®], Silicon Laboratories[®], Silicon Labs[®], SiLabs[®] and the Silicon Labs logo[®], Bluegiga[®], Bluegiga Logo[®], EFM[®], EFM32[®], EFR, Ember[®], Energy Micro, Energy Micro logo and combinations thereof, “the world’s most energy friendly microcontrollers”, Redpine Signals[®], WiSeConnect[®], n-Link, ThreadArch[®], EZLink[®], EZRadio[®], EZRadioPRO[®], Gecko[®], Gecko OS, Gecko OS Studio, Precision32[®], Simplicity Studio[®], Telegesis, the Telegesis Logo[®], USBXpress[®], Zentri, the Zentri logo and Zentri DMS, Z-Wave[®], and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

www.silabs.com