Getting Started with Keil μVision IDE

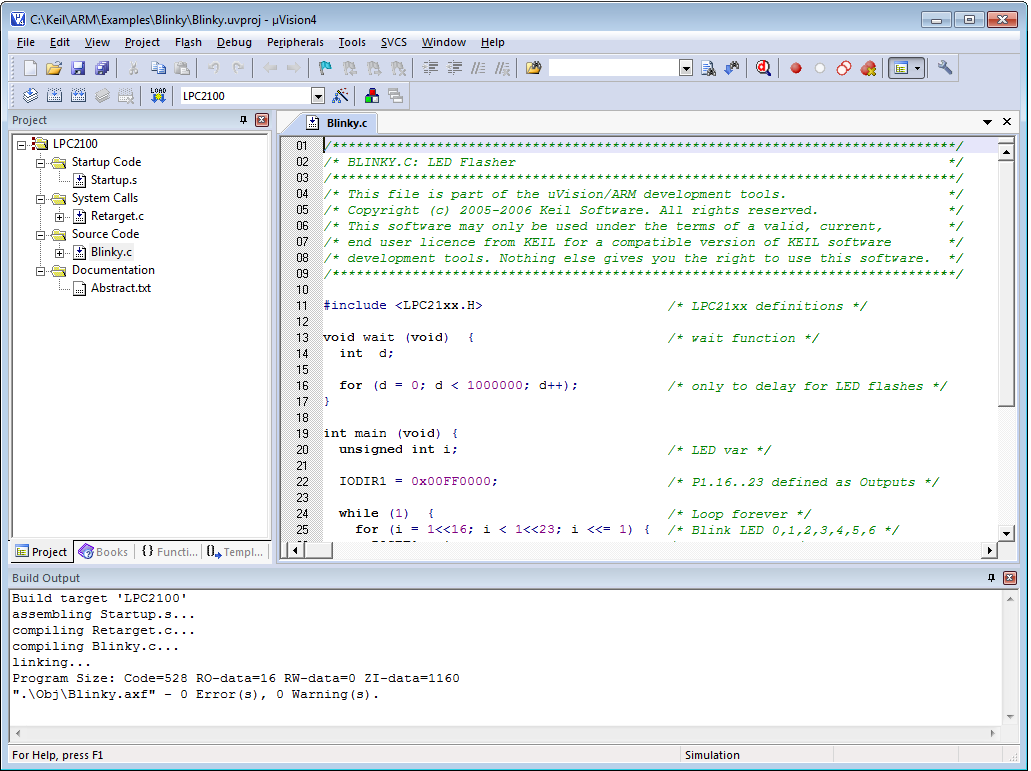
# KEIL μVision IDE OVERVIEW

Keil μVision has an integrated development environment (IDE), which allows users to build a project easily and quickly. The IDE includes:

* Project management
* Make facilities
* Source code editing
* Program debugging
* Complete simulation

A series of Arm-based tools are integrated in Keil μVision, including:

* Compiler
* Assembler
* Linker
* Format converter
* Libraries



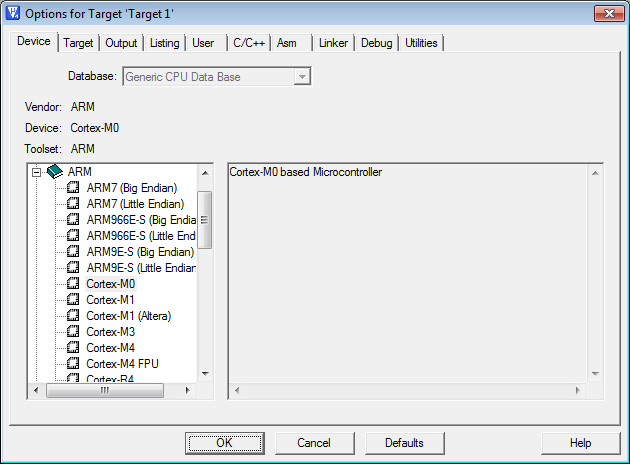
A Snapshot of Keil μVision IDE

# Create a New Project

1. Open Keil uVision from your Desktop.
2. Create a new project (Dummy).
3. In Select Device Target, choose Arm 🡪 Cortex-M0.
4. Right click on Target 1 in project navigator and click “Options for Target 1”.
5. This opens up the configuration window for your project.

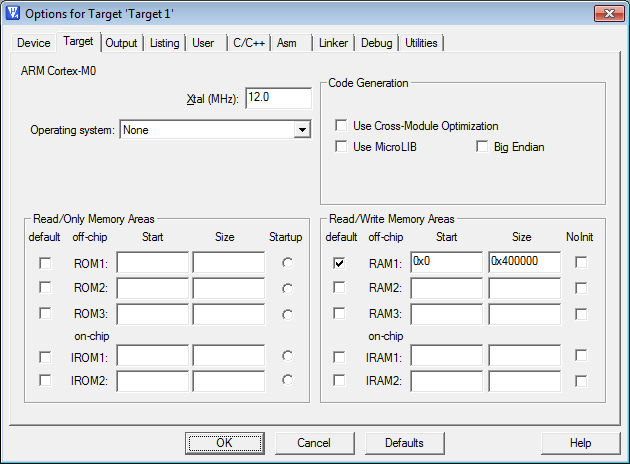
# Changing the configuration options

1. Device: Select the device from the µVision Device Database.



Configuration in Device Tab

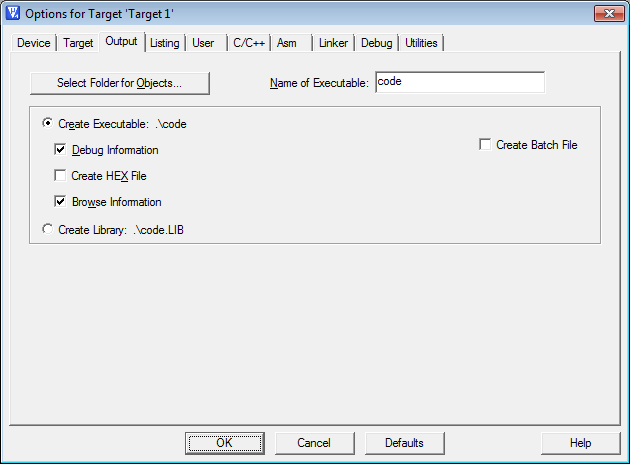
1. Target: Specify the hardware of your application.



Configuration in Target Tab

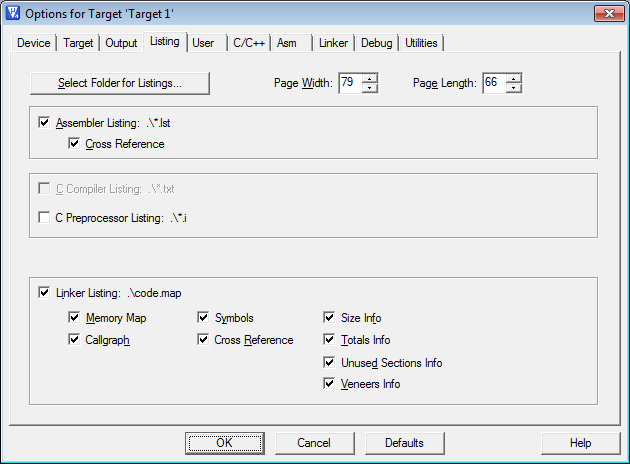
NOTE: The settings for Read/Only Memory Areas and Read/Write Memory Areas are used to create a linker scatter file. This requires that the setting Use Memory Layout from the Target Dialog is enabled in the Linker dialog.

1. Output: Define the output files of the tool chain and allows you to start user programs after the build process.



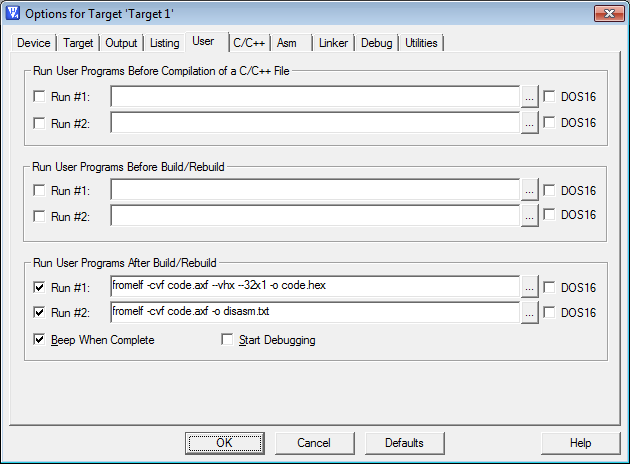
Configuration in Output Tab

1. Listing: Specify all listing files generated by the tool chain.



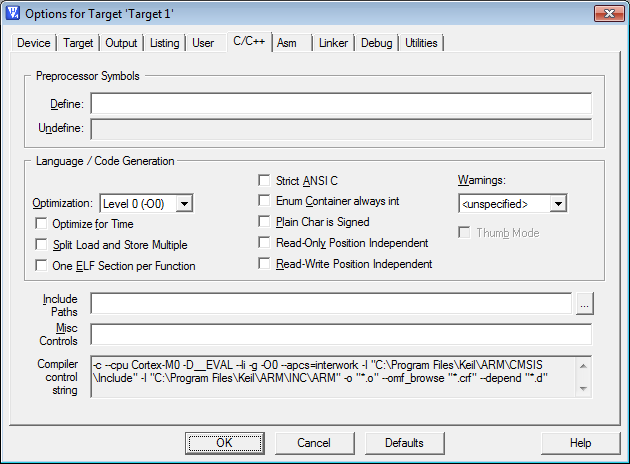
Configuration in Listing Tab

1. User: Specify user programs executed before compilation/build or after build.



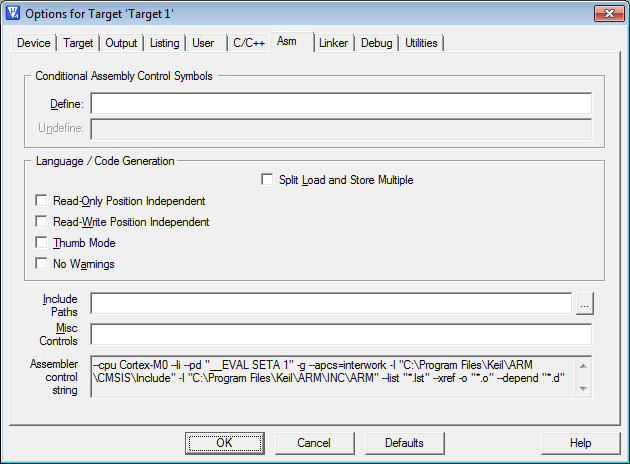
Configuration in User Tab

1. C/C++: Set C/C++ compiler-specific tool options like code optimization or variable allocation.



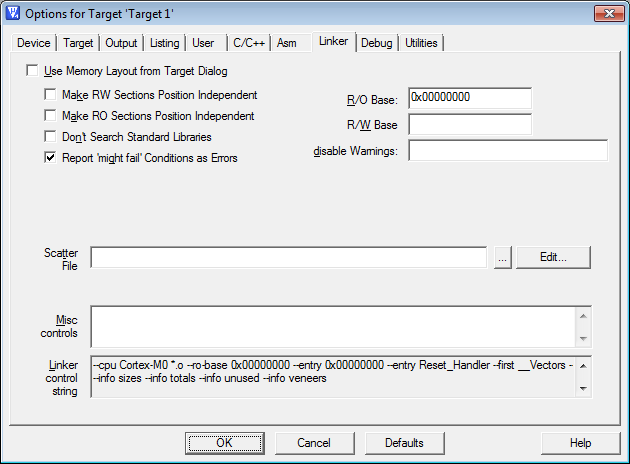
Configuration in C/C++ Tab

1. Asm: Set assembler-specific tool options like macro processing.



Configuration in Arm Tab

1. Linker: Set linker-related options. Typically, linker settings are required to configure the physical memory layout of the target system. Define the location of memory classes and sections in here.



Configuration in Linker Tab

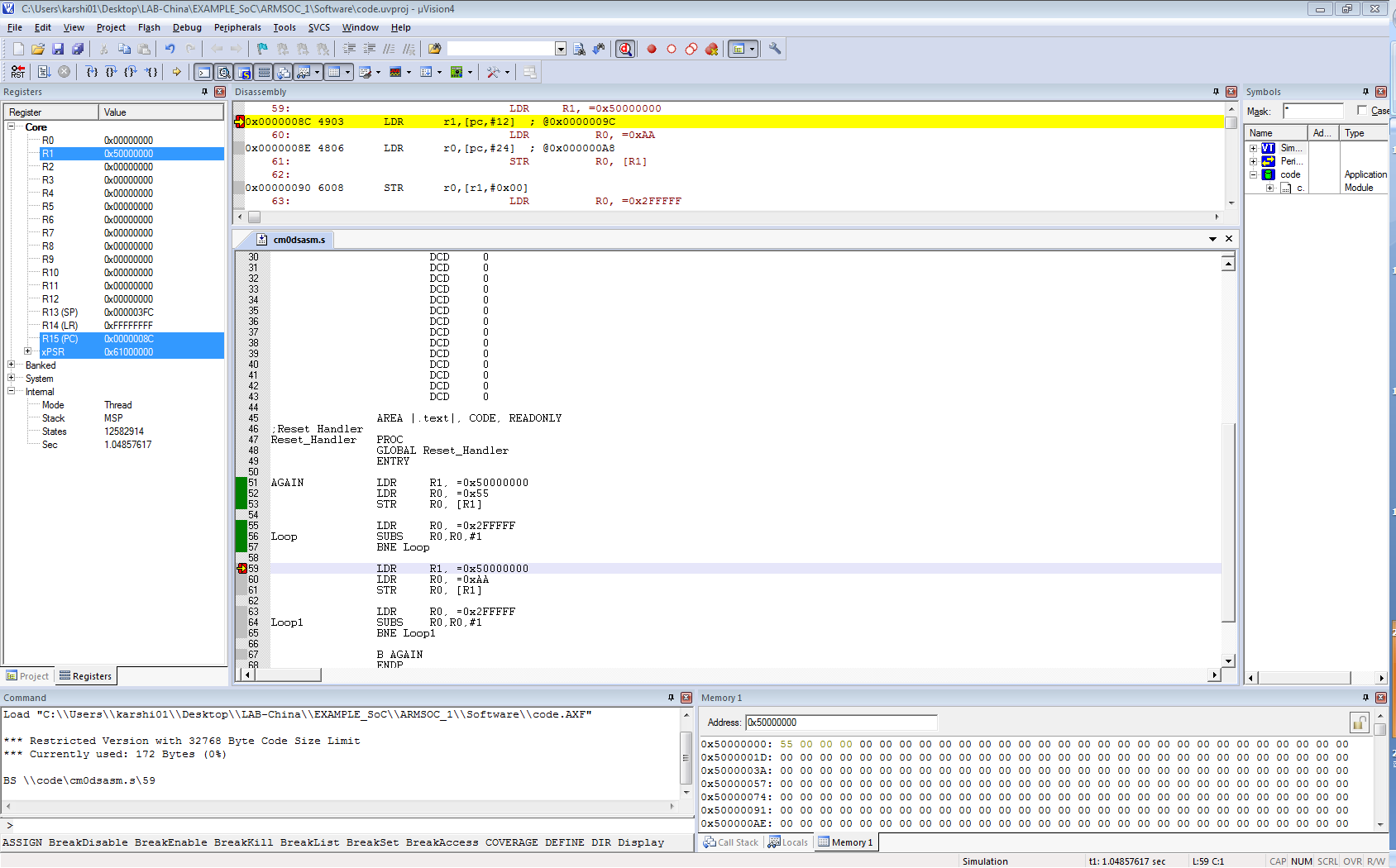
1. Debug: Settings for the µVision4 Debugger (Leave as default.)
2. Utilities: Configure utilities for Flash programming. (Leave as default.)

# Generating Binary from ASM

1. Add the assembly file “cm0dsasm.s” to your project by right clicking the source group and selecting “add files to group.”
2. Analyze the Vector Table and Reset Handler.
3. Build Target (Project 🡪 Build Target).
4. Analyze cm0dsasm.lst, code.map, and code.hex (executable).

# Using the Simulator

1. After you have compiled, go to Debug 🡪 Start/Stop Debug Session.
2. Ignore the warning message.
3. Note that PC is already pointing to the Reset\_Handler. (This matches the entry point flag set during compilation.)
4. Goto Debug 🡪 Memory Map and add the LED peripheral memory information. Note the region should be Read and Write.
5. Execute the image using Single Step.
6. View the contents of 0x5000\_0000 in Memory1 window.



Using μVision Simulator