RT58x SOC Platform

Getting Started

V1.4

About this Document

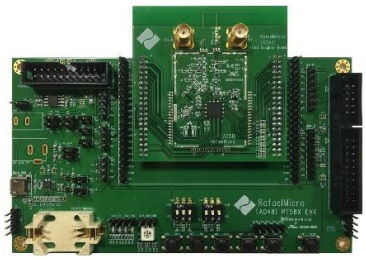
This document supports at least "***Rafael RT58x SDK v0.3.1***". For any version older than v0.3.1, there is no guaranteed it can work properly.

Table of Contents

1. Introduction

The purpose of this document is to teach users about the installation and configuration of Rafael RT58x SDK, Keil MDK-ARM, and Visual Studio Code. The SDK provides examples and projects to develop applications on development kit.

2. Development Environment Setup



Rafael RT58x EVK

ARM J-Link Adapter

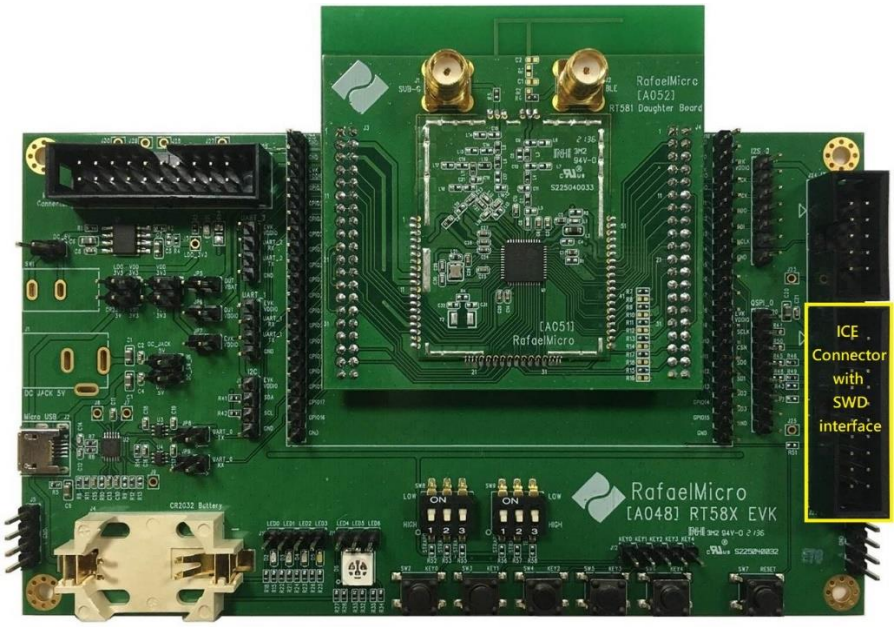
IDE/CMSIS

2.1 Debugger

- Install the ARM J-Link driver.
- Connect J-Link Adapter to RT58x EVK board via JTAG/SWD bus.
- Connect J-Link Adapter to PC via USB.

2.2 Development Board

The RT58x EVK provides an SWD interface with connector for use with an ICE debugger (J-Link Adapter) via 20pin IDE cable.



For detailed description of RT58x EVK board, please refer to RT58X EVK User Guide.

2.3 KEIL MDK-ARM

The MDK-ARM is a complete software development environment for Cortex-M, Cortex-R4, ARM7, and ARM9 processor-based devices. MDK-ARM is specifically designed for microcontroller applications, it is easy to learn and use, yet powerful enough for the most demanding embedded applications.

For detailed introduction and download, please click the following link:
http://www.keil.com/arm/mdk.asp

Note: MDK-Lite Edition is available for download. It does not require a serial number or license key. It features software development for microcontrollers based on ARM Cortex-M processors. It is intended for product evaluation, small projects, and the educational market. It is restricted to 32Kbyte code size.

2.4 Visual Studio Code

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging. Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute command in the console. Visual Studio Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster.

For detailed introduction and download, please click the following link:
https://code.visualstudio.com

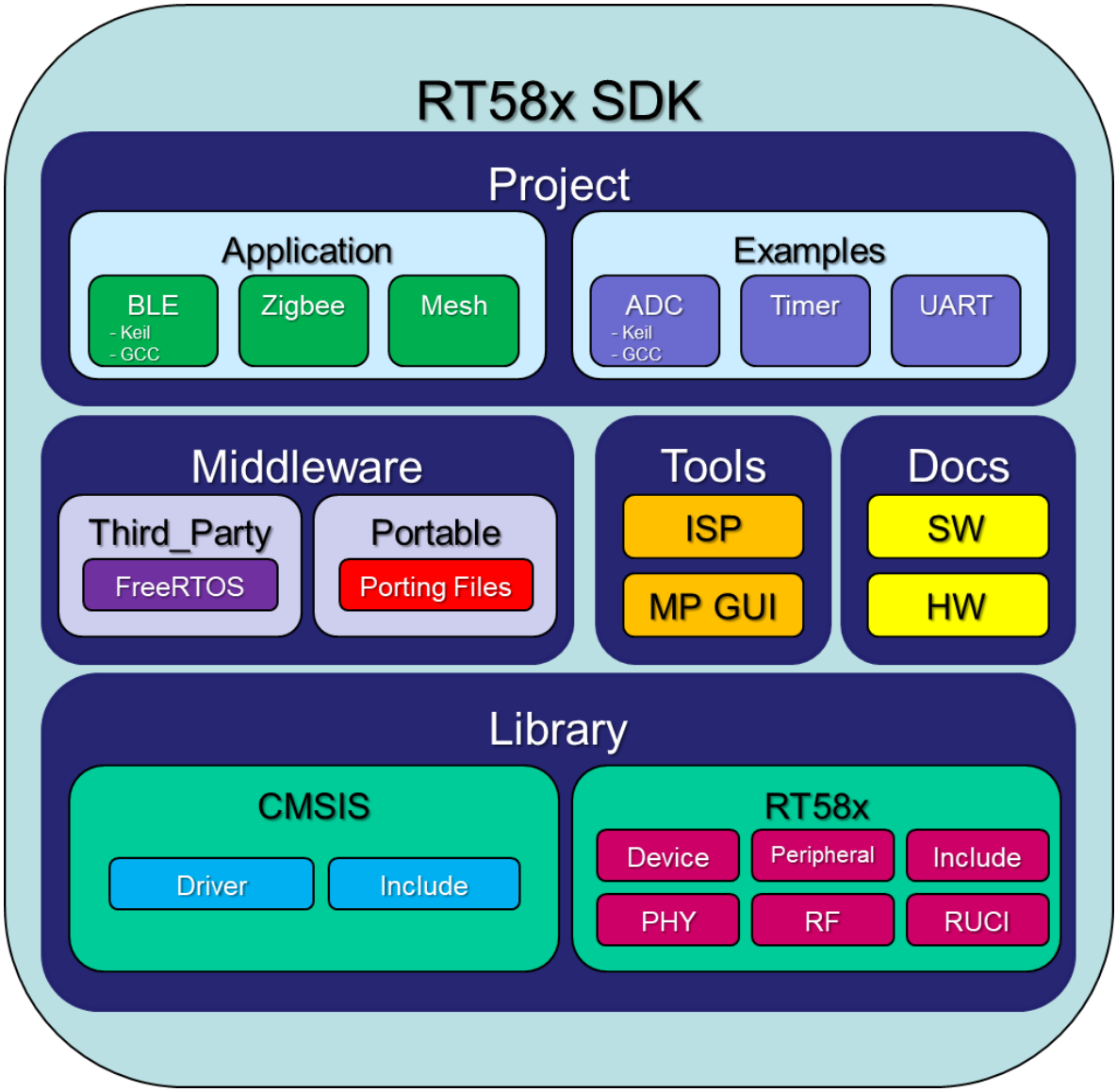
Note: VS Code is available and free for download. It does not require a serial number or license key. For detailed installation of VS Code, please refer to VS Code User Setup Readme.

3. Software Development Kit

Rafael RT58x SDK is a complete software development kit for application development such as BLE, Zigbee, Mesh, and Sub-G. Rafael RT58x SDK is specifically designed for Rafael's SoC with the ability to perform high-performance Cortex-M microcontroller and support to driver powerful RF and peripheral features. It is easy to install the Rafael RT58x SDK package by unzipping the RT58x_SDK_version.zip to the specified development directory.

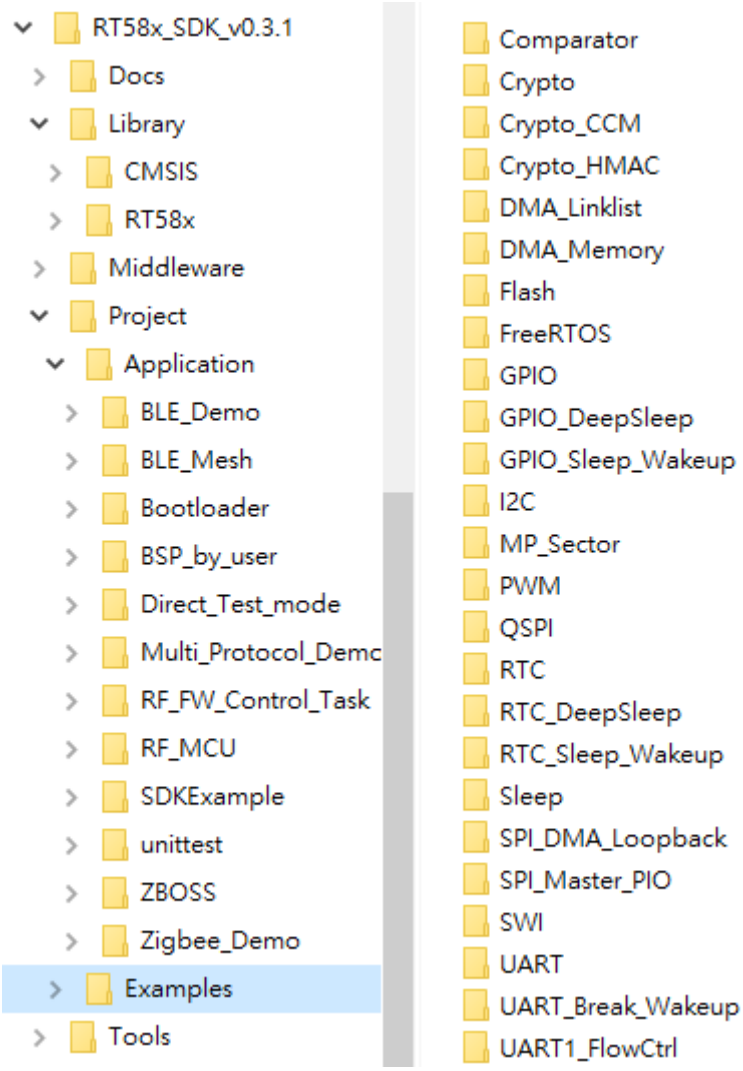
3.1 SDK Software Architecture

The following figure shows the software architecture in the Rafael RT58x SDK. This illustration will guide users to use RT58x to quickly develop structured applications.



3.2 SDK Directory Structure

The following figure shows the related files and directories in the Rafael RT58x SDK. These files and directories contain project, middleware, library, tools, and documents. Rafael RT58x SDK also provides complete examples for user reference. Users can add and modify related files to the corresponding directory to develop the required applications.



4. Develop

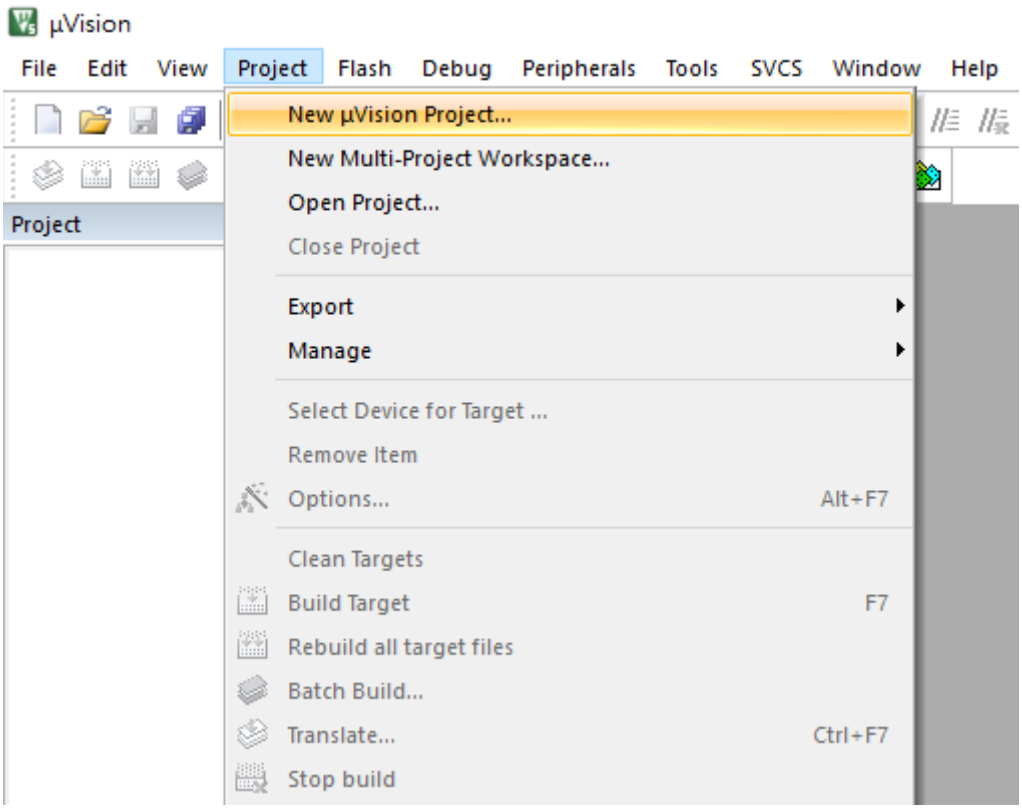
The uVision IDE and Debugger is the central part of the Keil development toolchain and has numerous features that help the programmer to develop embedded applications quickly and successfully. uVision offer a Build Mode for creating applications and a Debug Mode for debugging applications.

4.1 Creating Applications

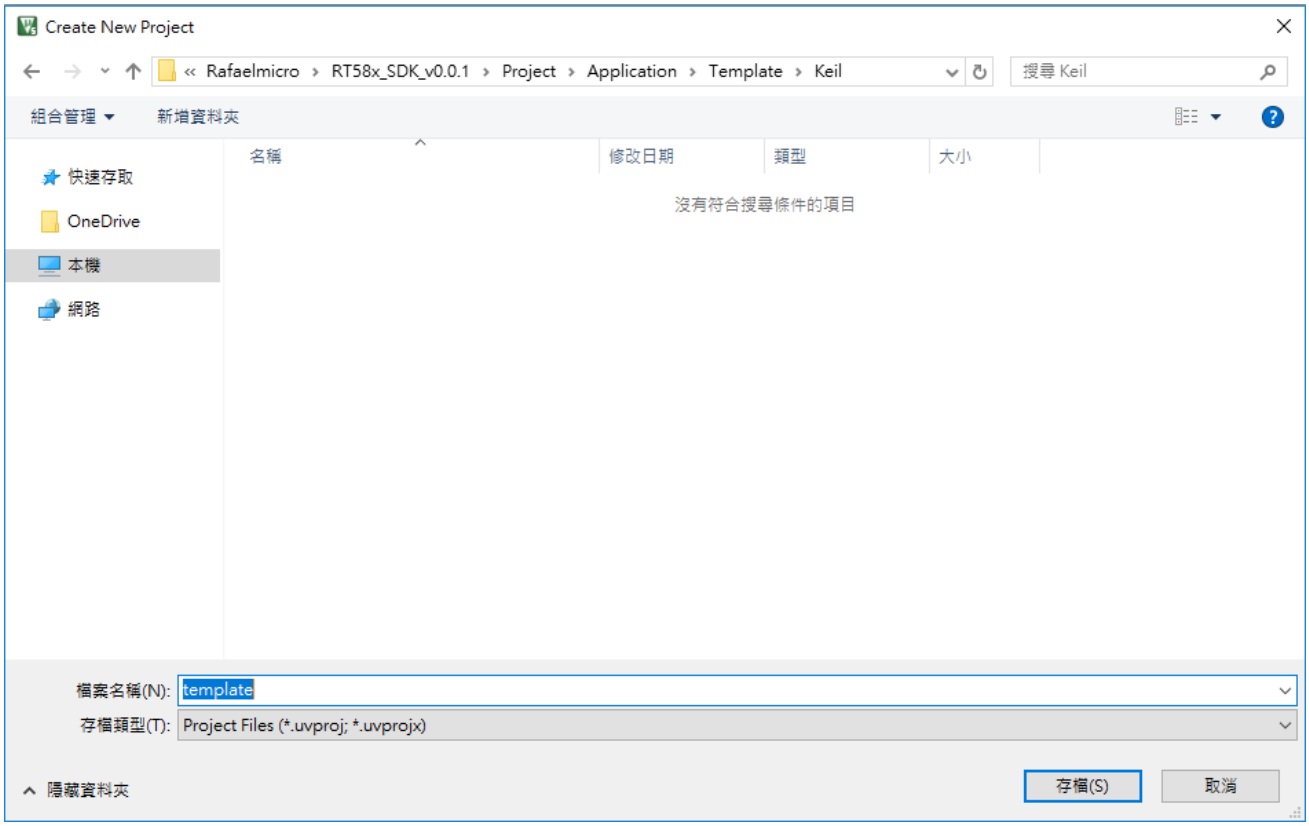
This chapter describes the creation of projects using Software Components, shows features to edit and compile source files, fix errors and warnings, and generate executable code.

The following will provide a step-by-step tutorial that creates a simple project. The Project Manager makes it easy to create a new project and to design the embedded application.

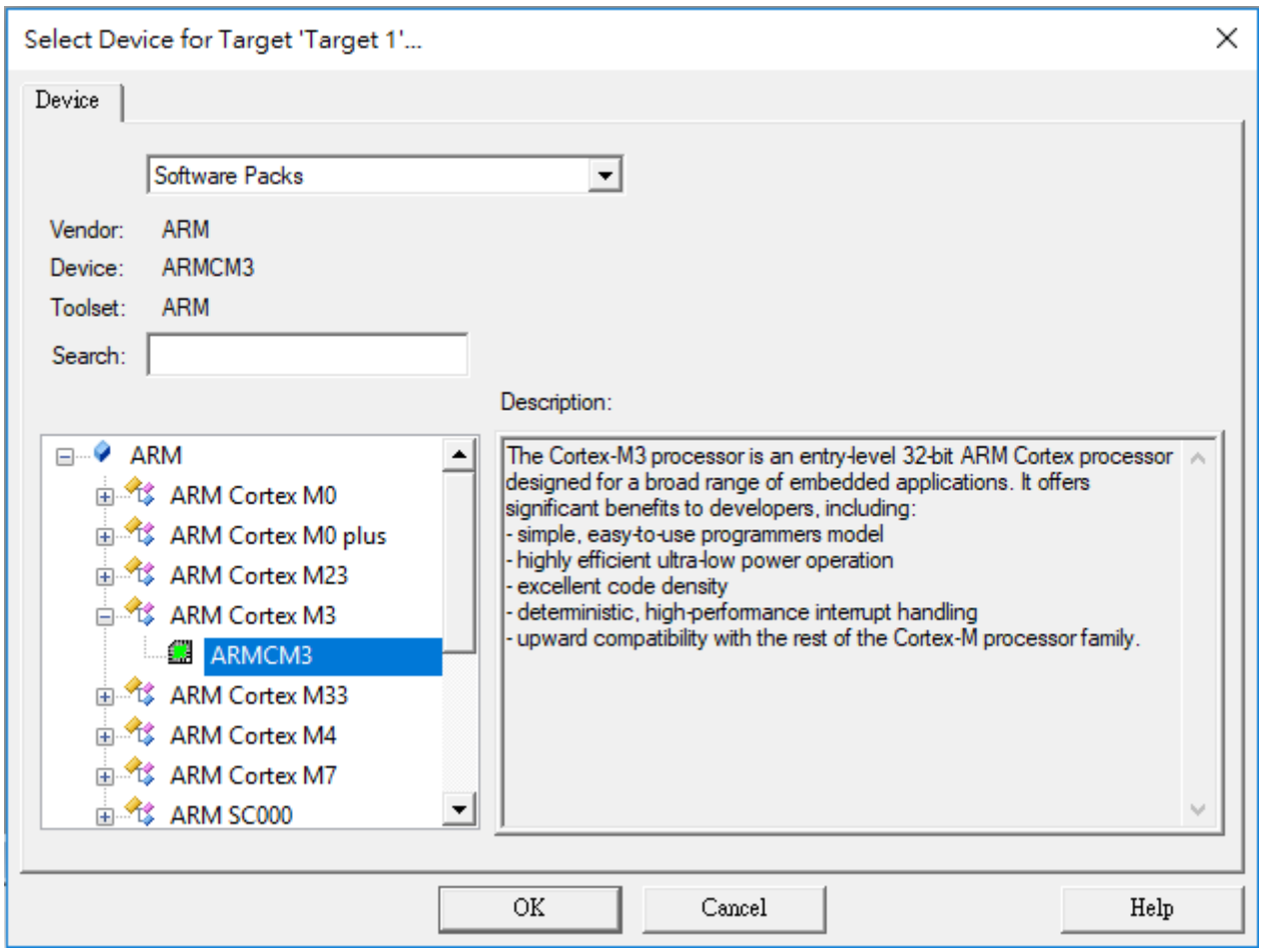
Step 1: Setup the Project. The menu Project → New uVision Project creates a new project.



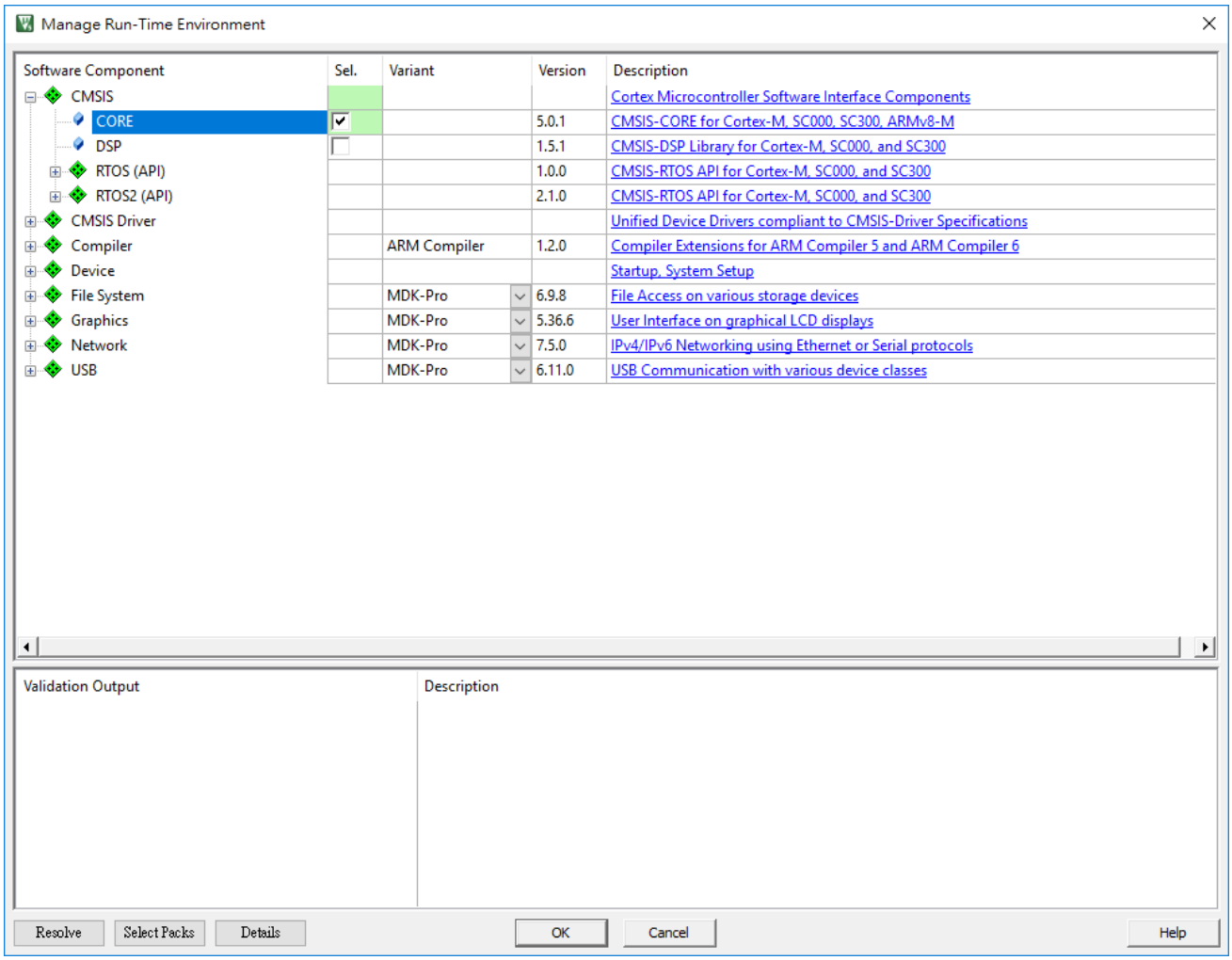
Step 2: Select an empty folder and enter the project name, for example *template*.



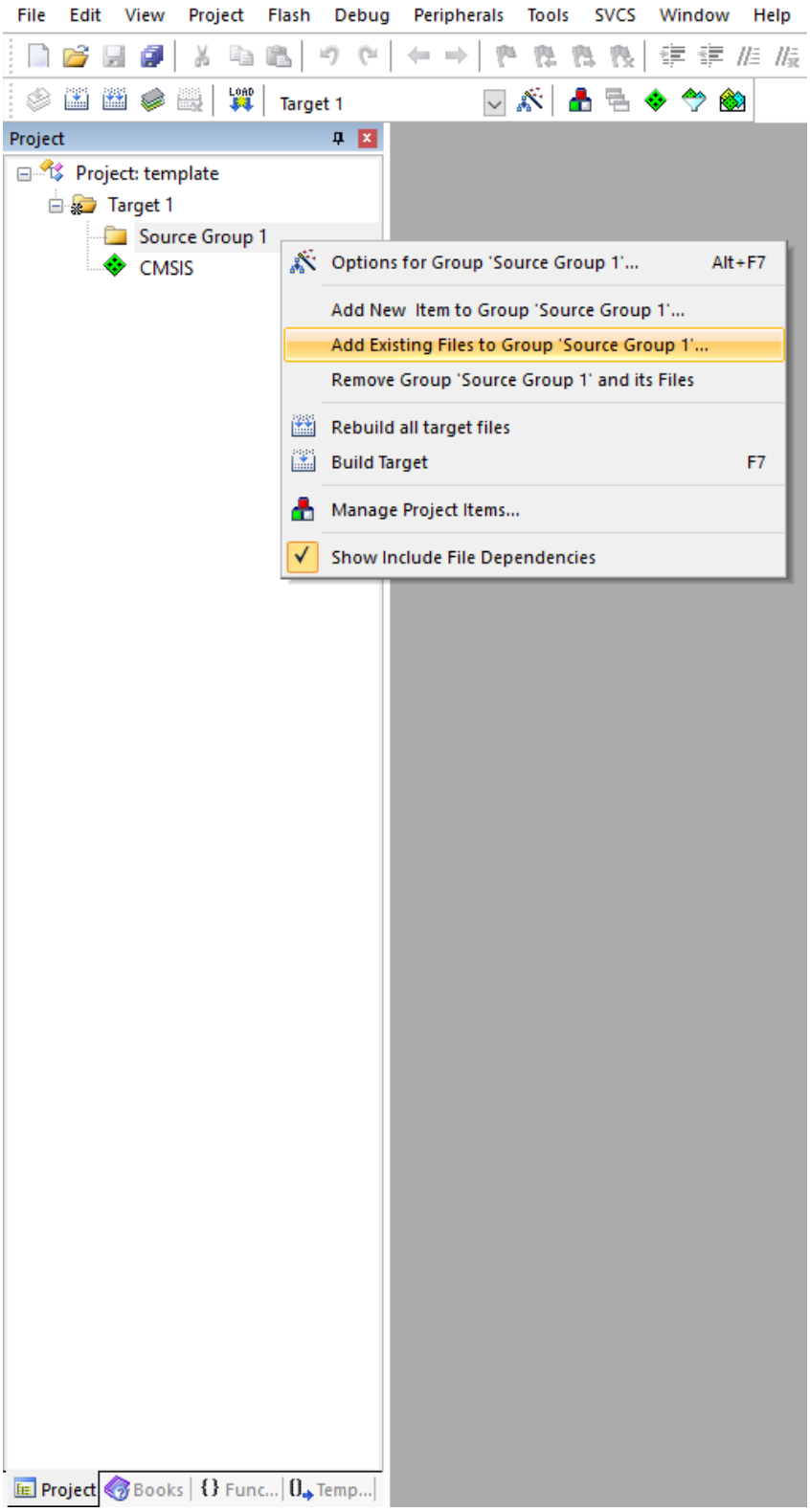
Step 3: Select ARM Cortex M3 → ARMCM3 as the device database.



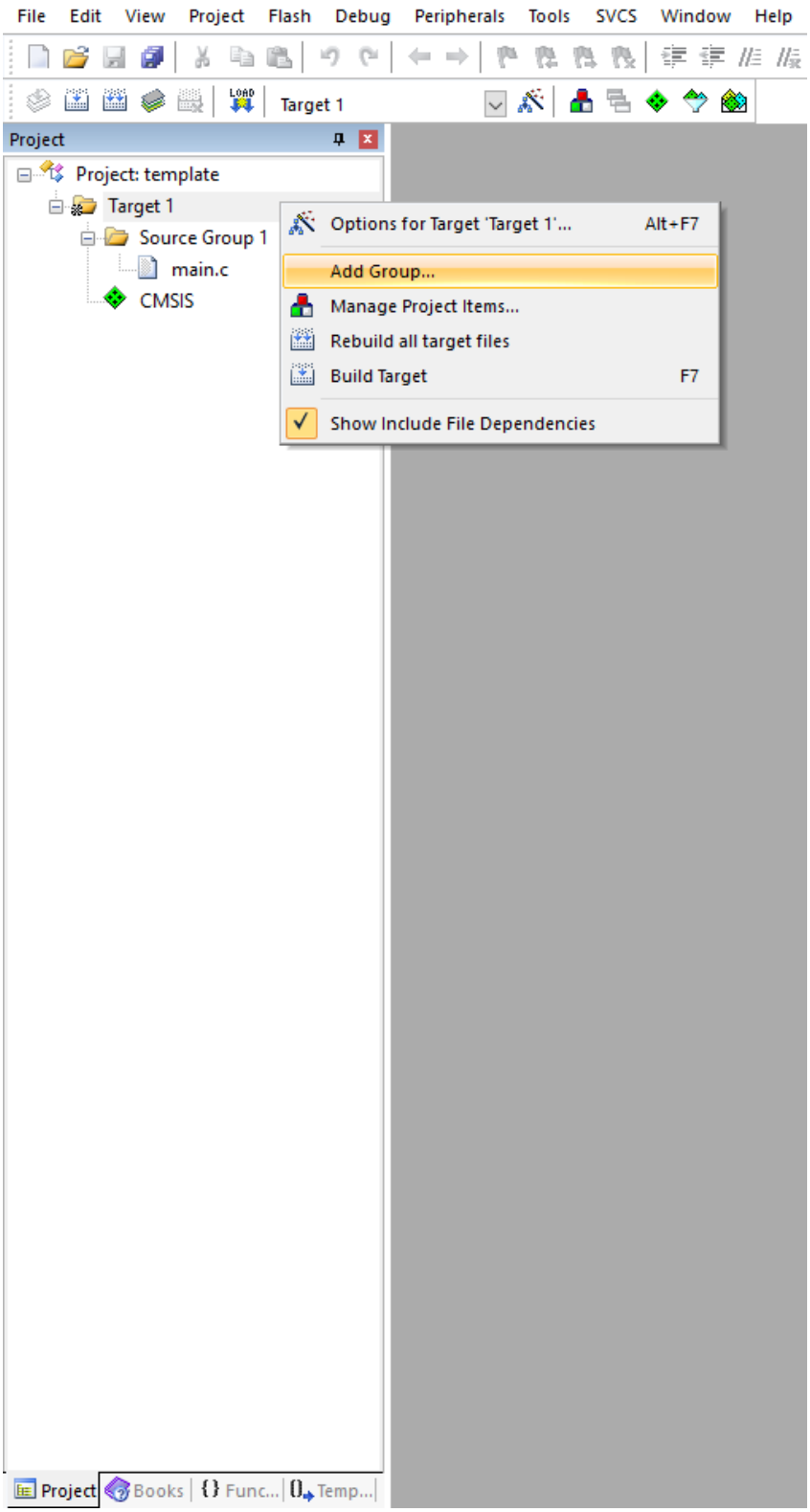
Step 4: Select Software Components. Add CMSIS → CORE to the project.

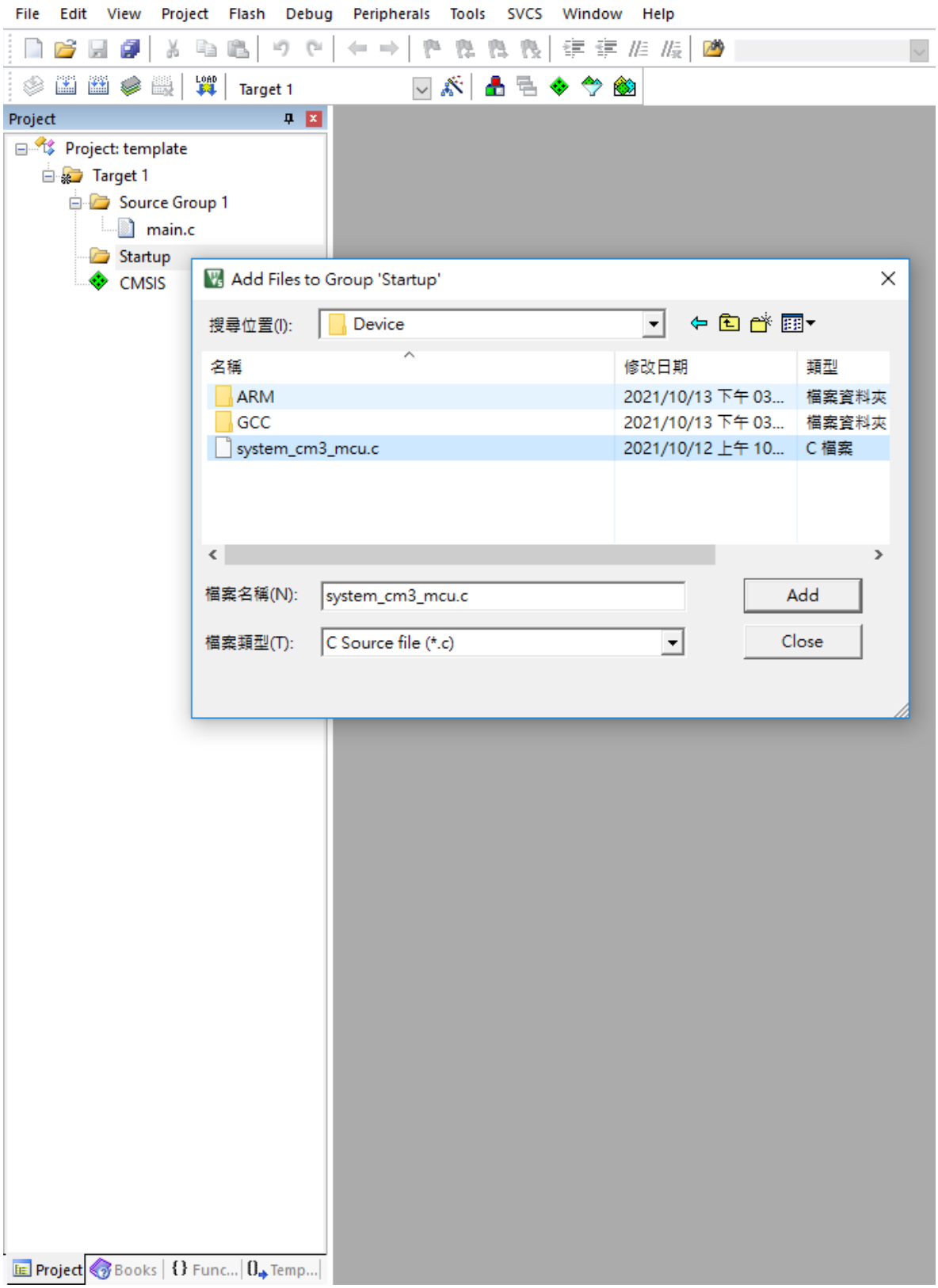


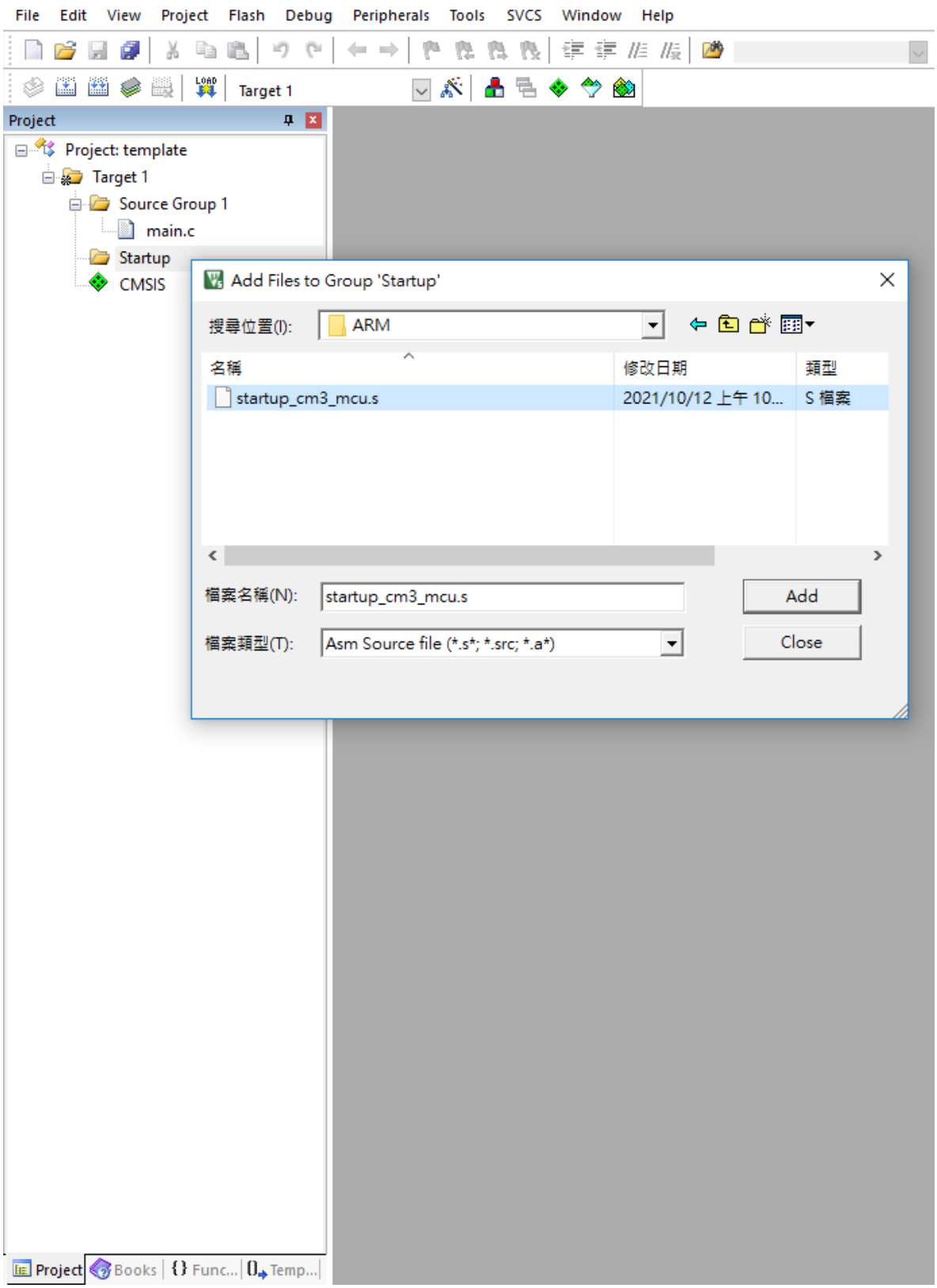
Step 5: Add Source Files to Project. Click on a file group in the window Project and use the context menu “Add New Item to Group” or “Add Existing Files to Group”.

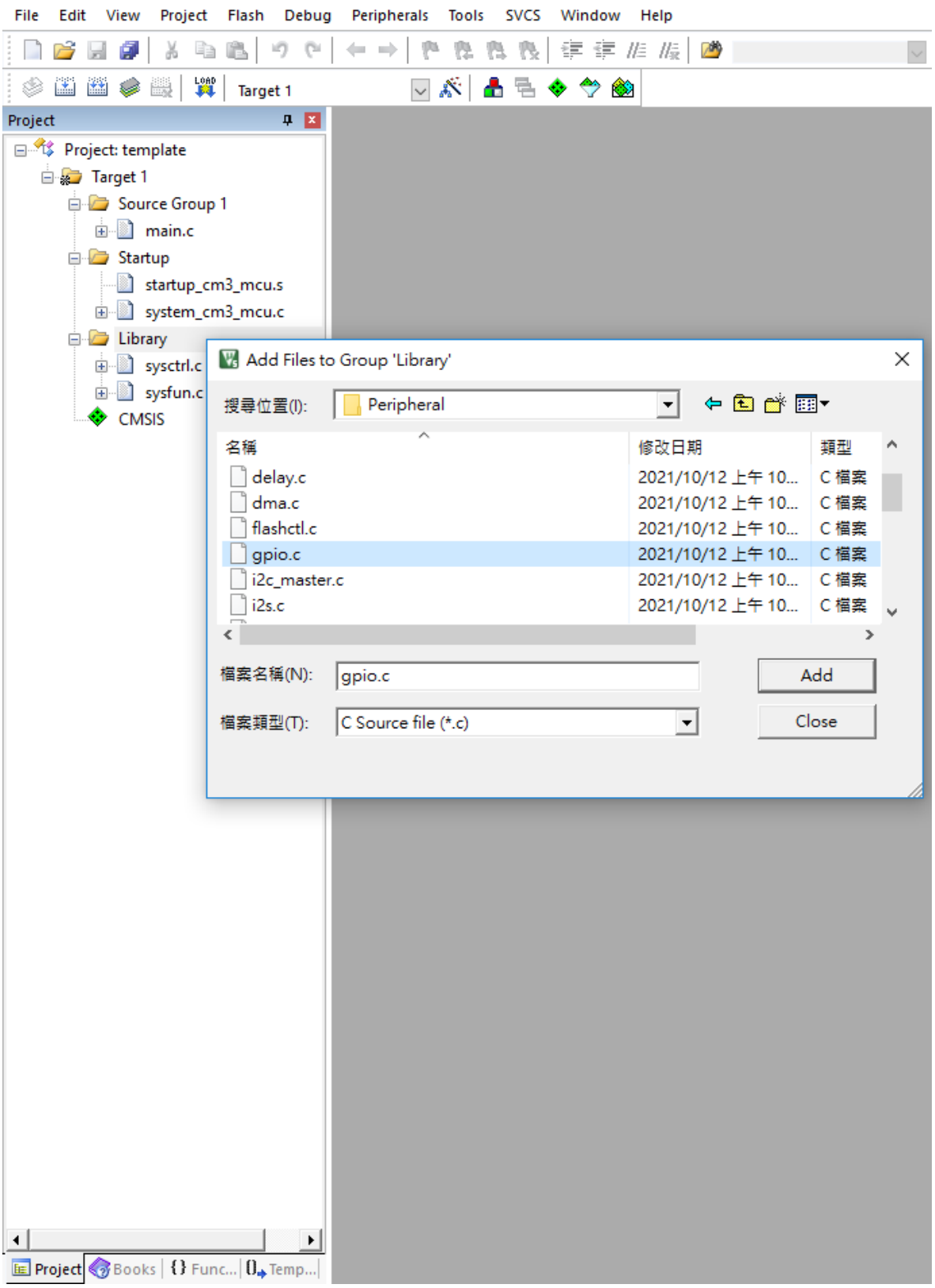


Step 6: Create file groups to simplify the project maintenance. Right-click the target name and select “Add Group” to add “Startup” group and “Library” group, and add startup related files and library related files to the corresponding groups.







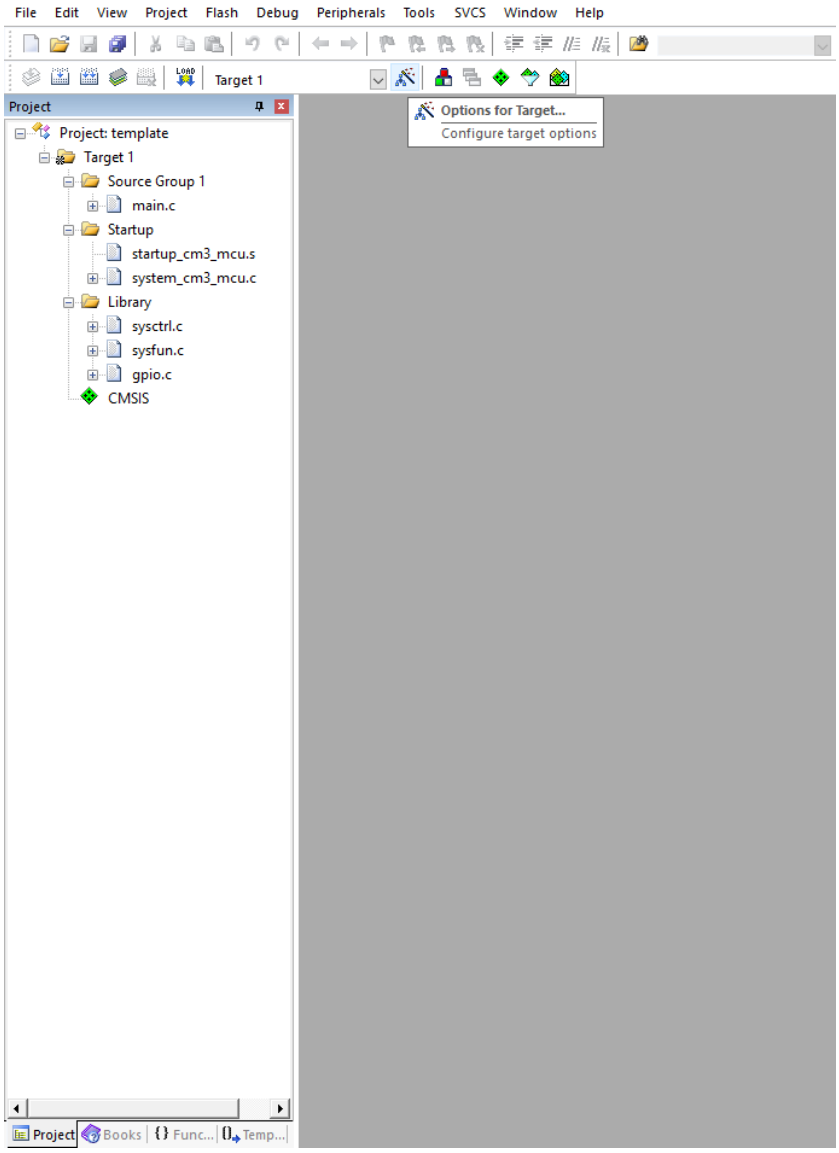


4.2 Debugging

This chapter describes the Debugger for controlling the target device using a debug adapter. While debugging, developers have full access to the source code and can control as well as analyze program execution on C or Assembly level.

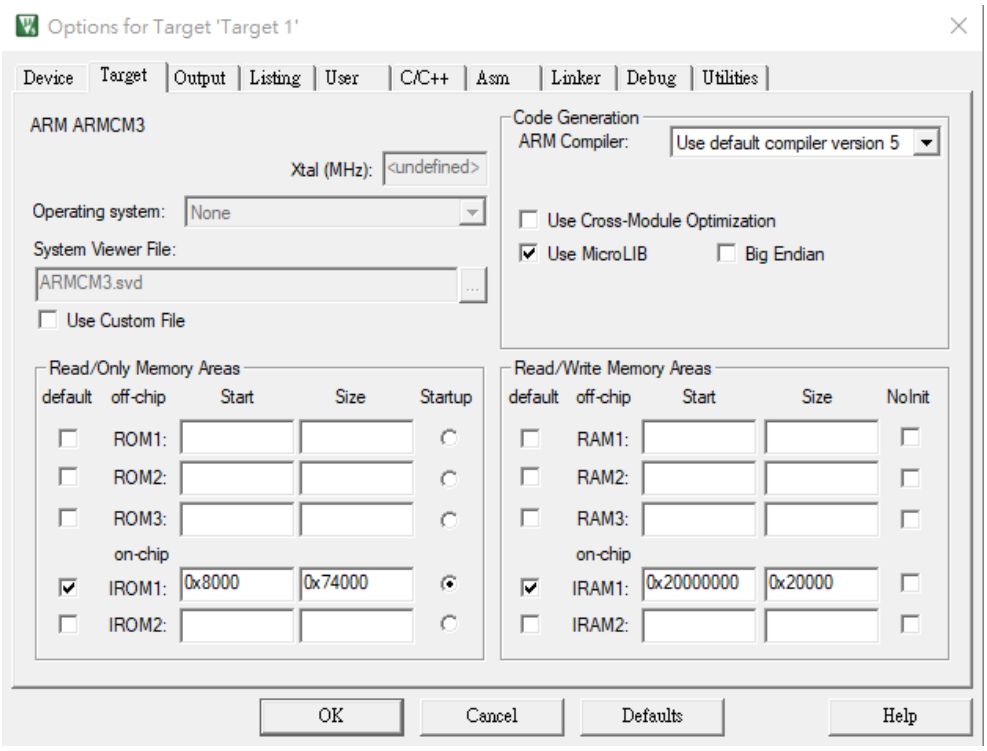
The following will provide a step-by-step tutorial that uses Target Debugger connects to a debug adapter.

Step 1: Set the Options for Target. The options configure the target hardware and the development environment.

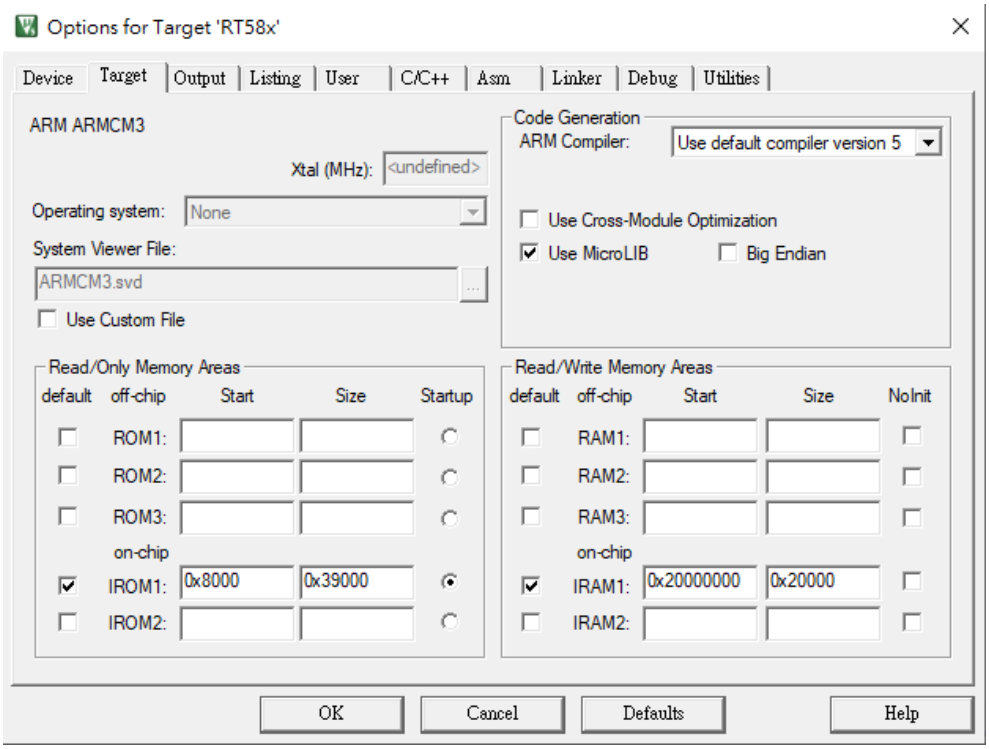


Step 2: Set Xtal, IROM, and IRAM options depend on the RT58x specifications.

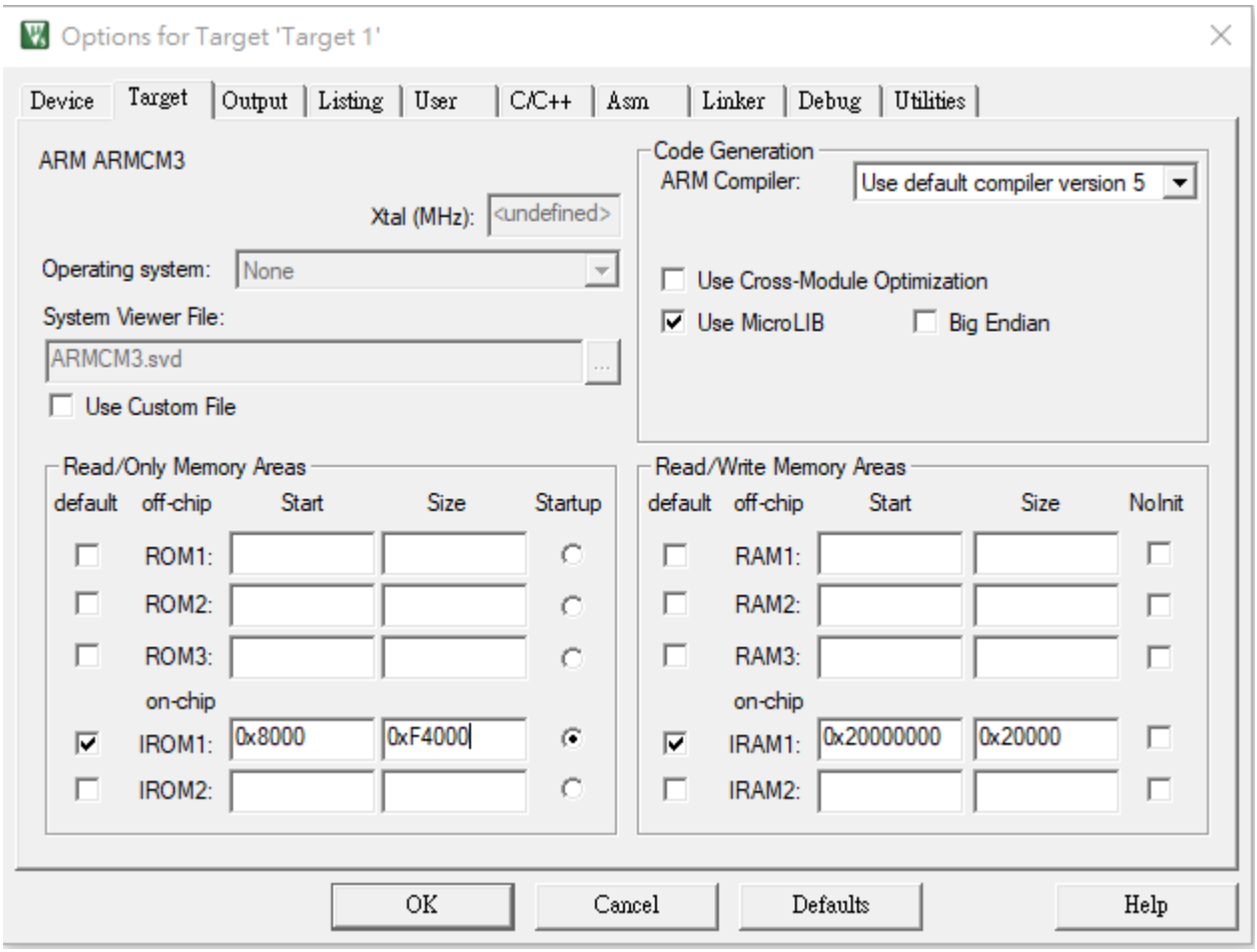
a. RT58x (1MB Flash size) IROM and IRAM setting



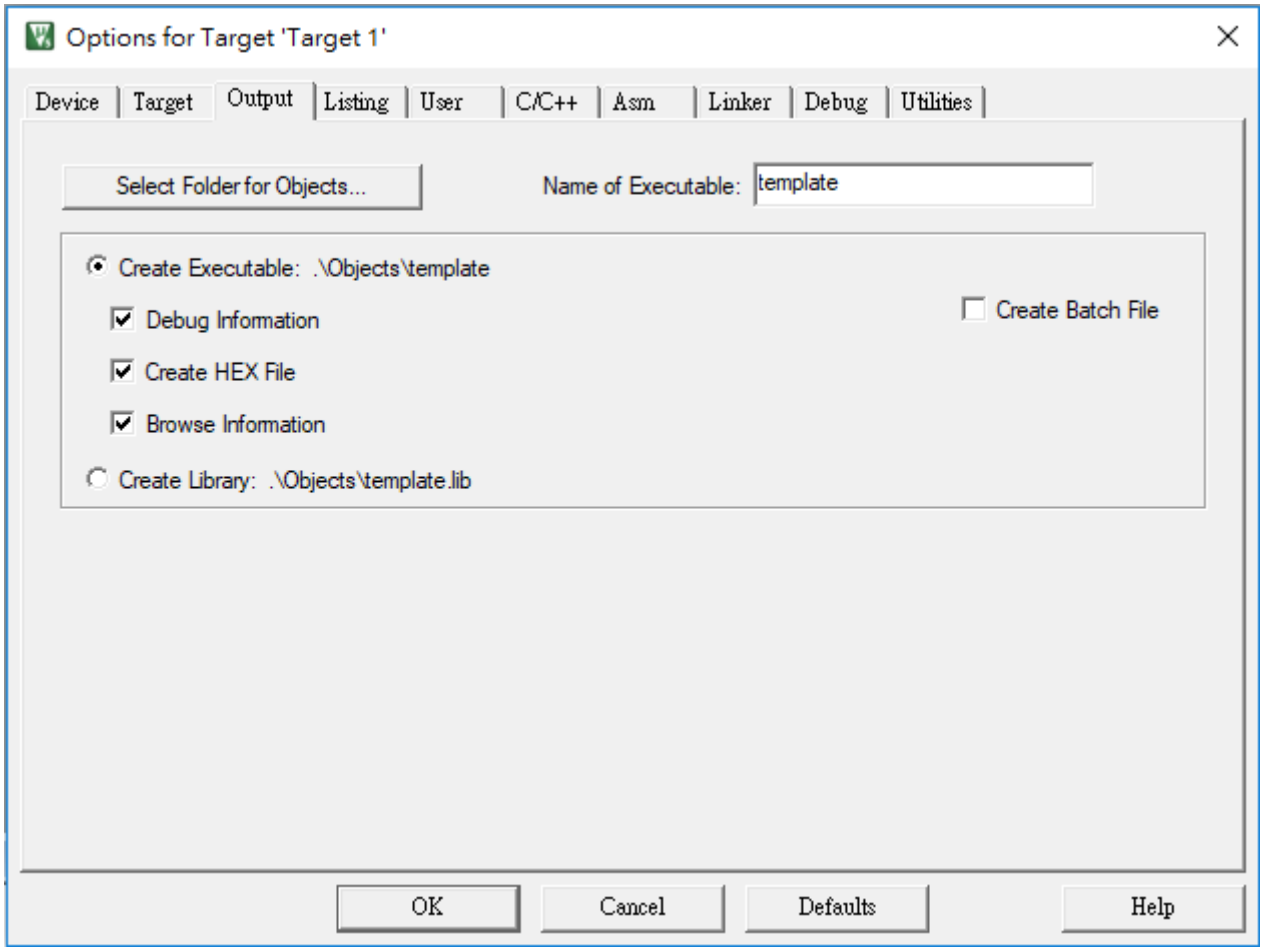
b. RT58x (512KB Flash size) IROM and IRAM setting



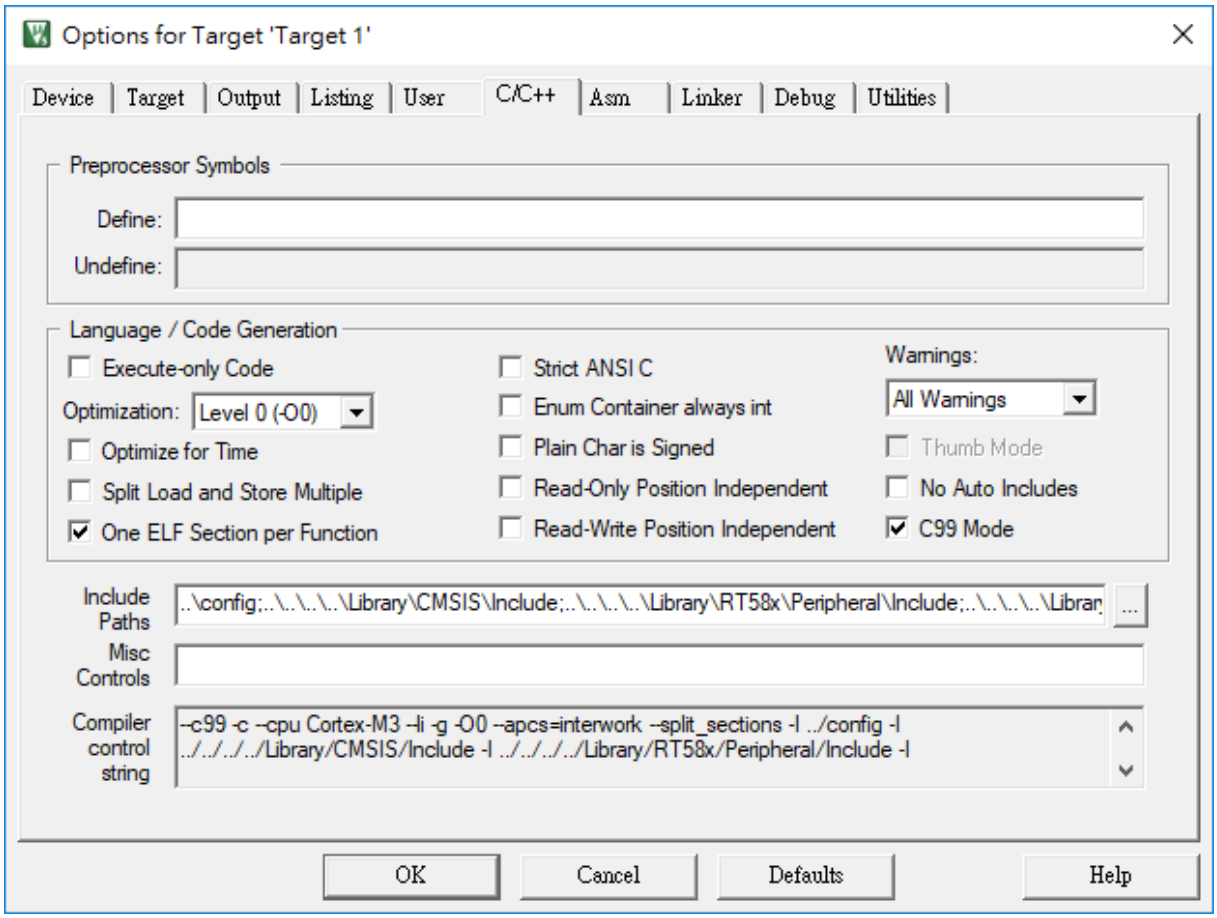
c. RT58x 2MBB Flash size) IROM and IRAM setting

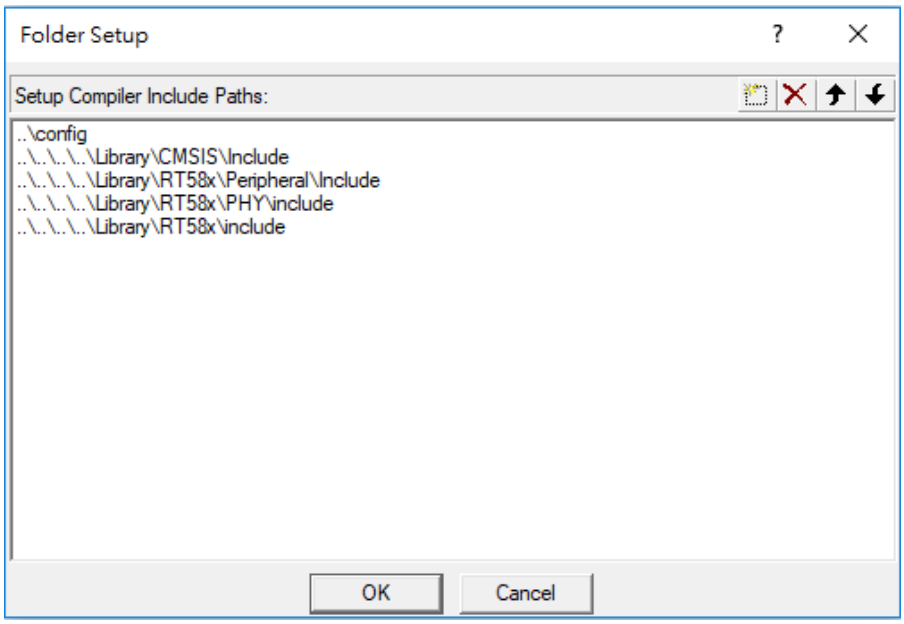


Step 3: Set the Create HEX File. HEX files are used to download the software to programmer memory.

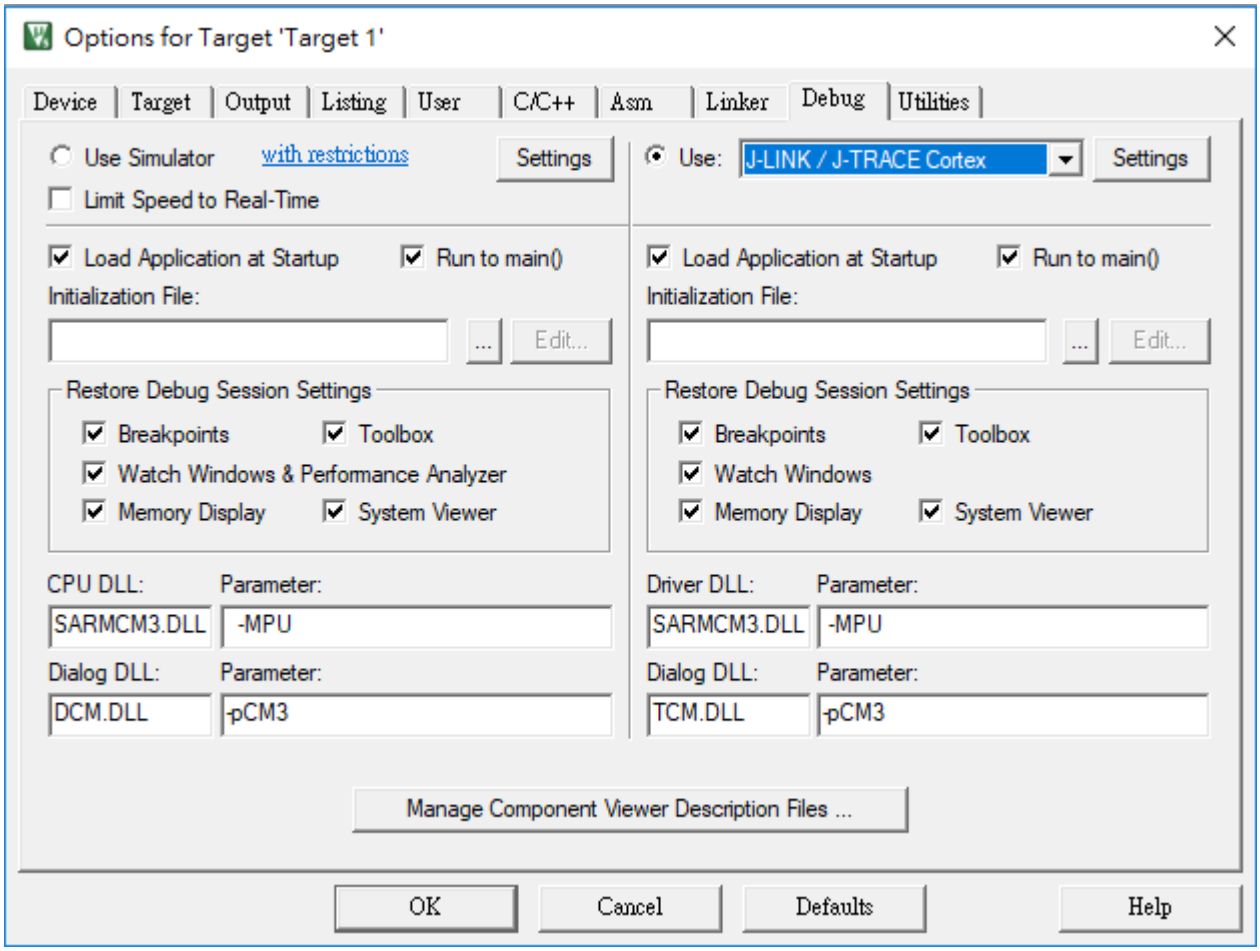


Step 4: Set the Include Paths. Supply one or more paths to search for header files. Add CMSIS file path to Include Paths.

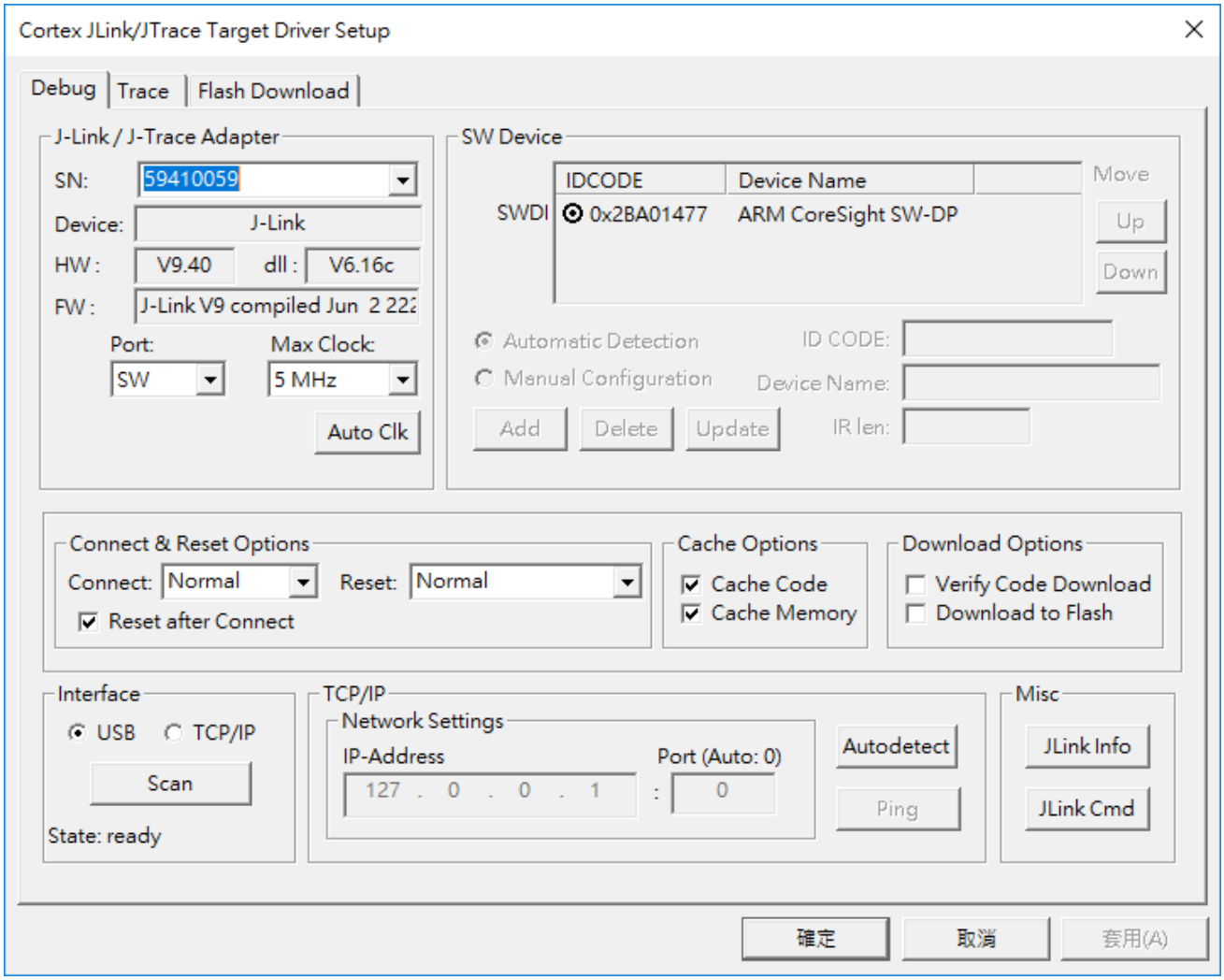




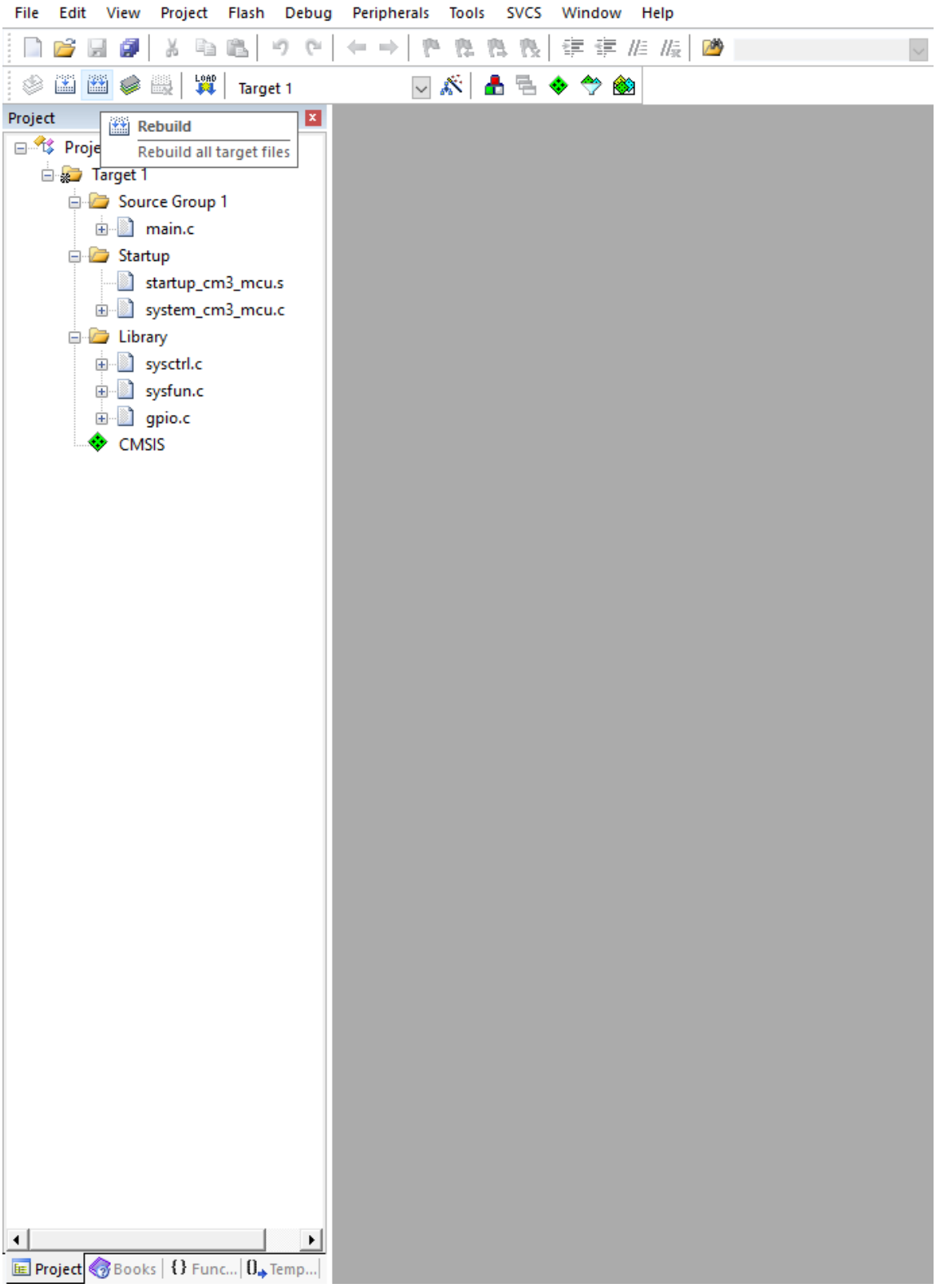
Step5: Set to use debug driver. When J-Link Adapter is available, select J-Link driver for debugging the target hardware.



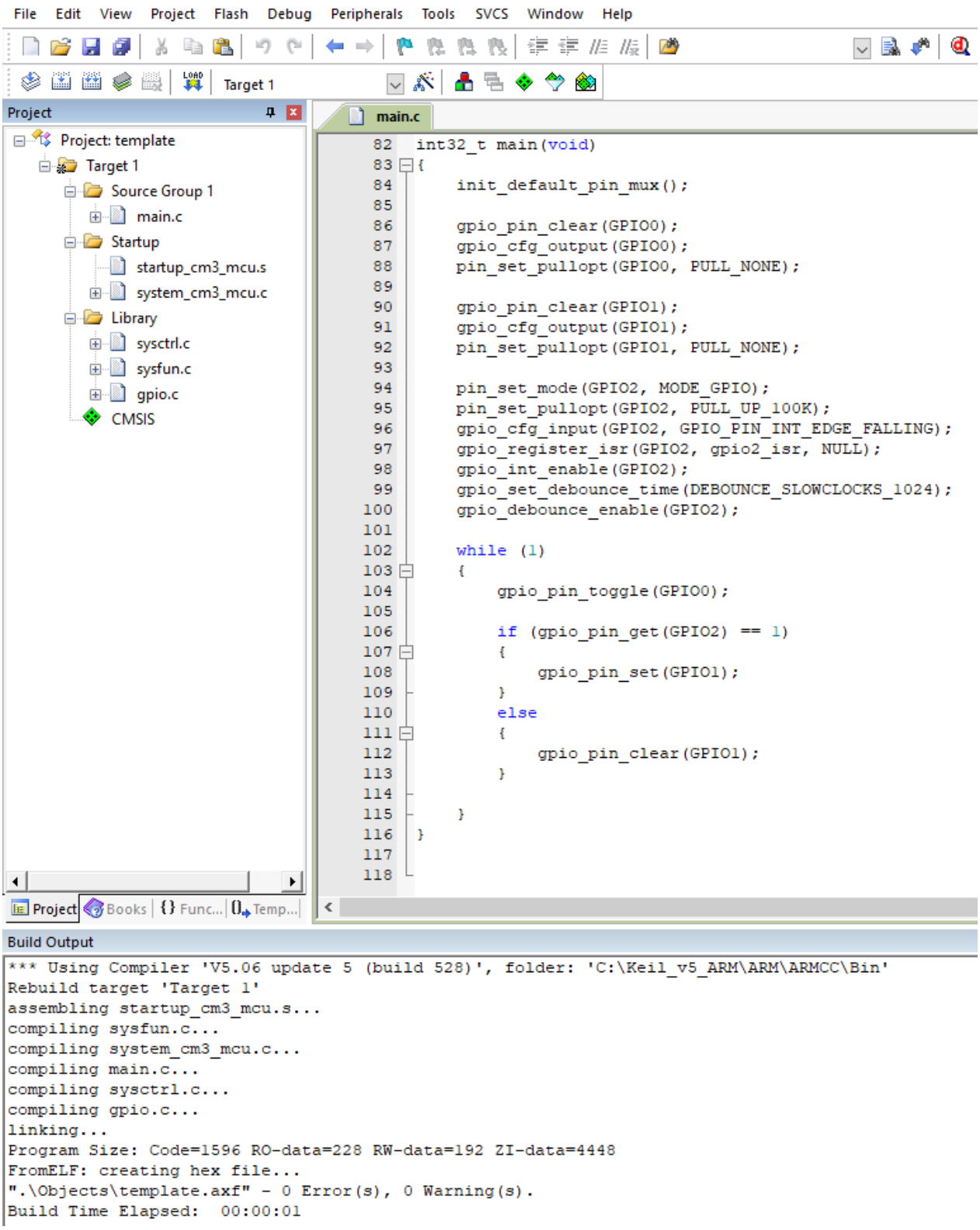
Click the Settings button to display the dialog Target Driver Setup, and select the port as SW will display IDCODE and Device Name when J-Link Adapter is available.



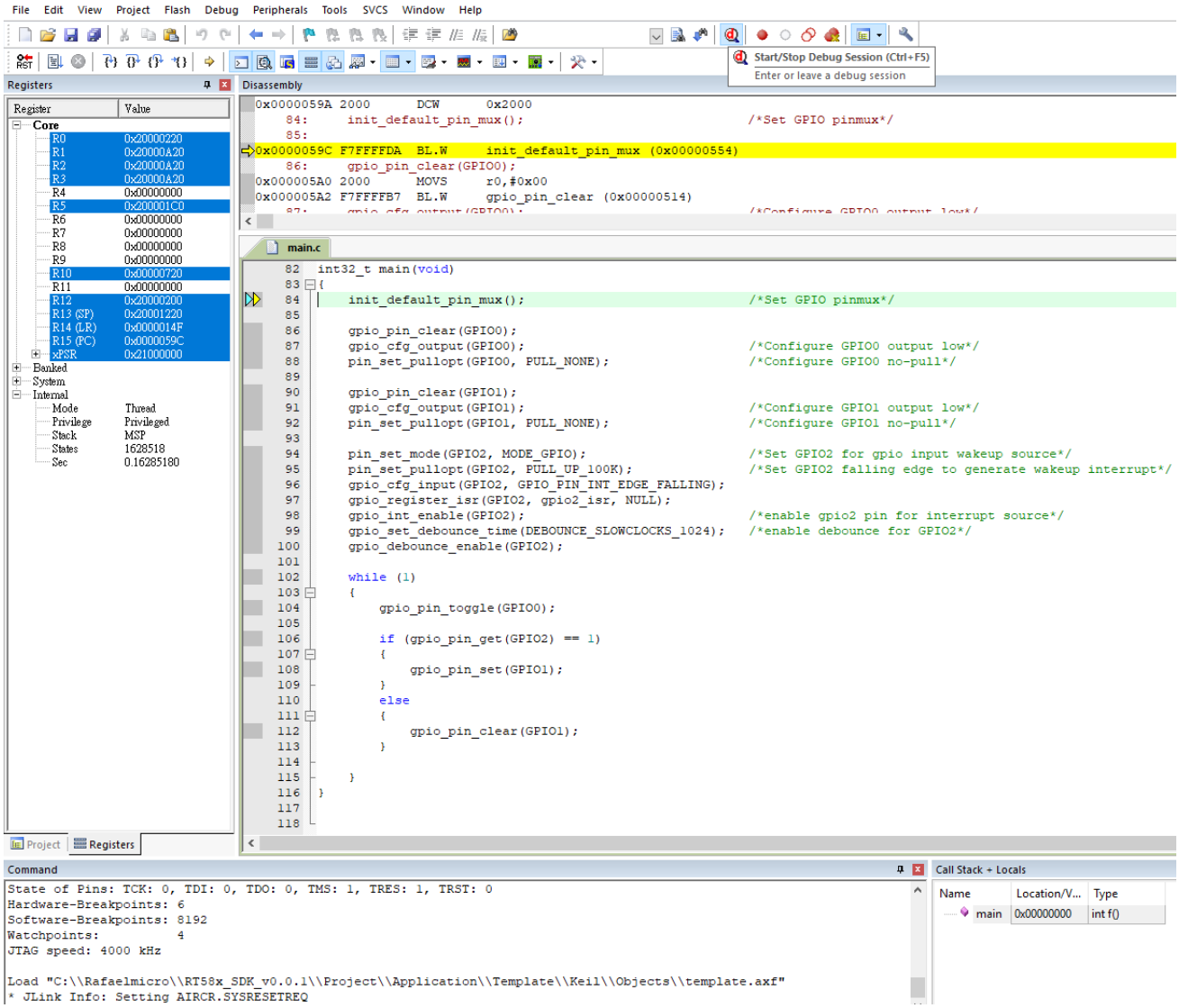
Step 6: Build the Project. Click the Build or Rebuild button to translate all source files and link the application, and generates the executable file.



Step 7: The Build Output window will display errors, warnings, and build messages during the build process.



Step 8: Start Debugging. Click the Start/Stop Debug Session button to load the application program and executes the startup code, and Debug Menu and Commands are available, such as Reset CPU, Run, Stop, Step, Step Over, Step Out, Breakpoint, etc.



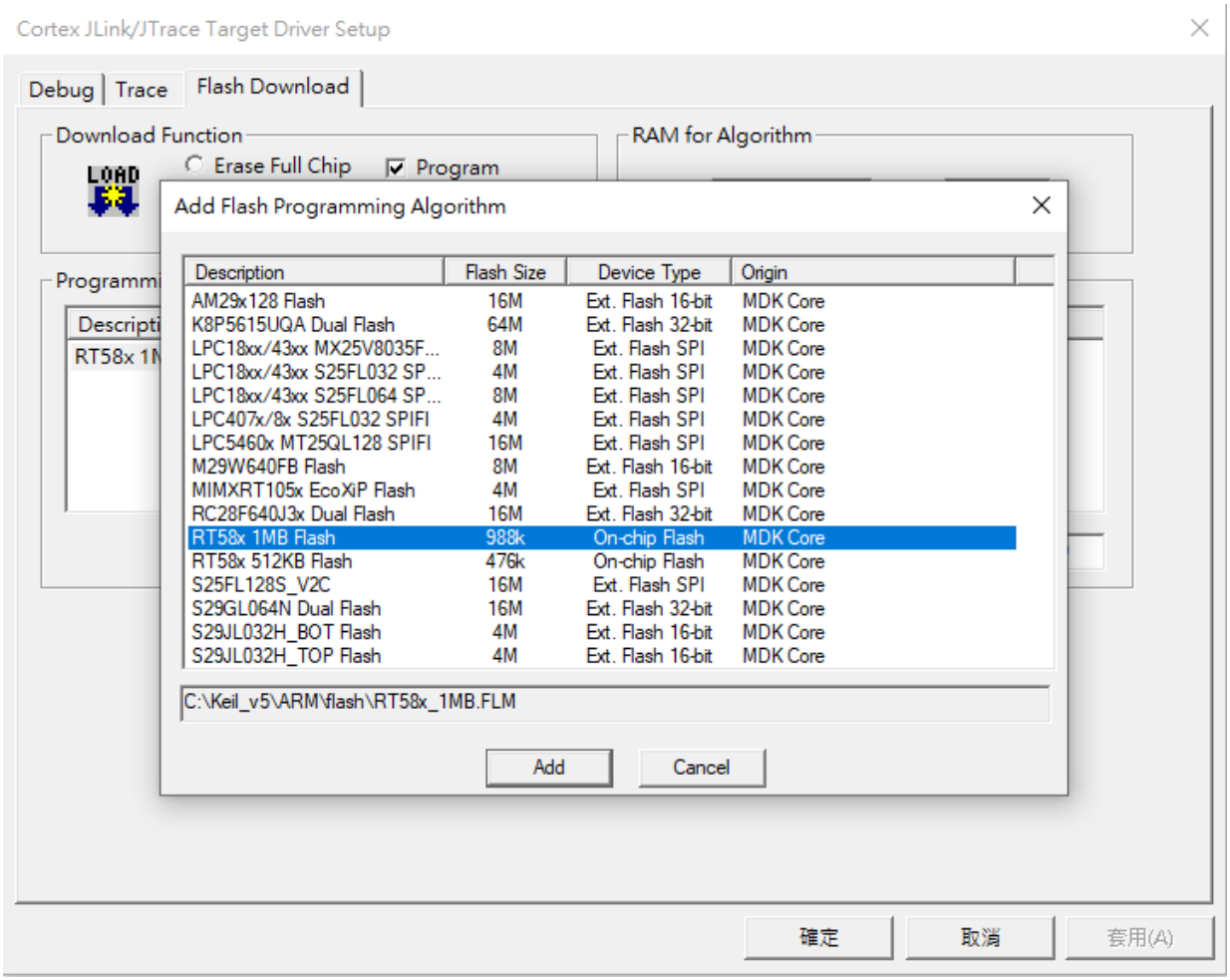
Note: The Reset command in the Debug Menu only resets the Cortex-M microcontroller, not for the RF and peripheral modules in the RT58x.

4.3 Flash Programming

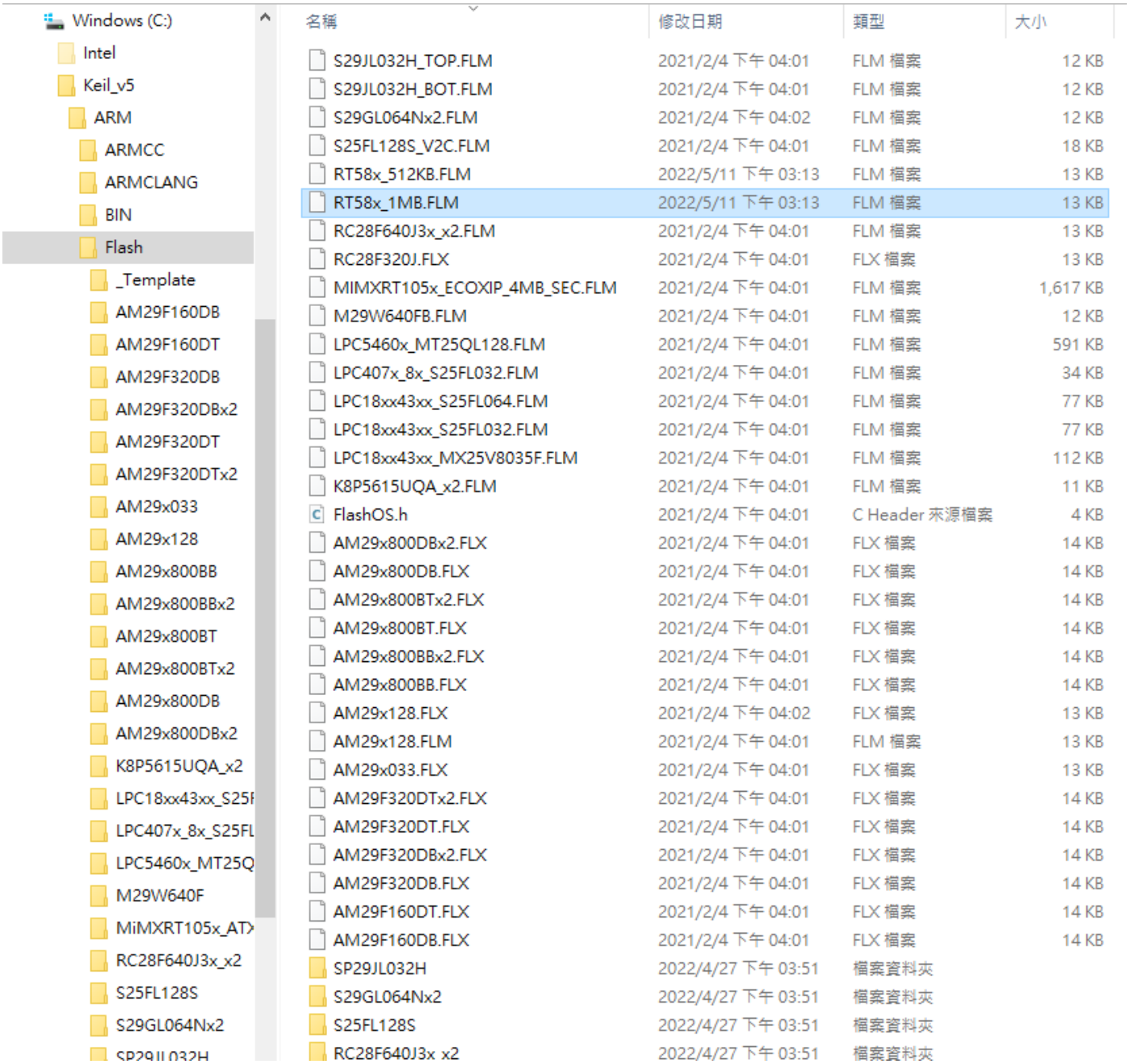
This chapter describes the configuration options for downloading applications to Flash, provides information about Flash programming algorithms.

The following will provide a step-by-step tutorial that configures a debug adapter for Flash programming.

Step 1: Flash Download Configuration. Select the Options for Target → Debug → Settings → Flash Download. Click the Add button in the Programming Algorithm section to include a new device and programming algorithm (*RT58x 1MB Flash/RT58x 512K Flash*) from a list of supplied algorithm.

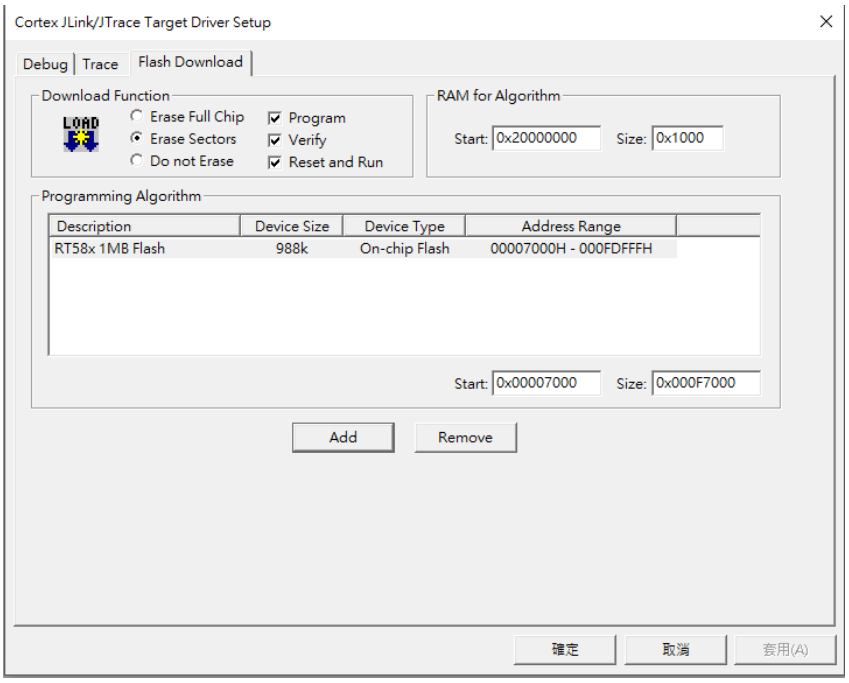


Note: RT58x Flash programming algorithm (RT58x_1MB.FLM/RT58x_512K.FLM) must be copied to the folder [MDK_Install_Path]\ARM\Flash\ to support RT58x in the list of supplied algorithm.

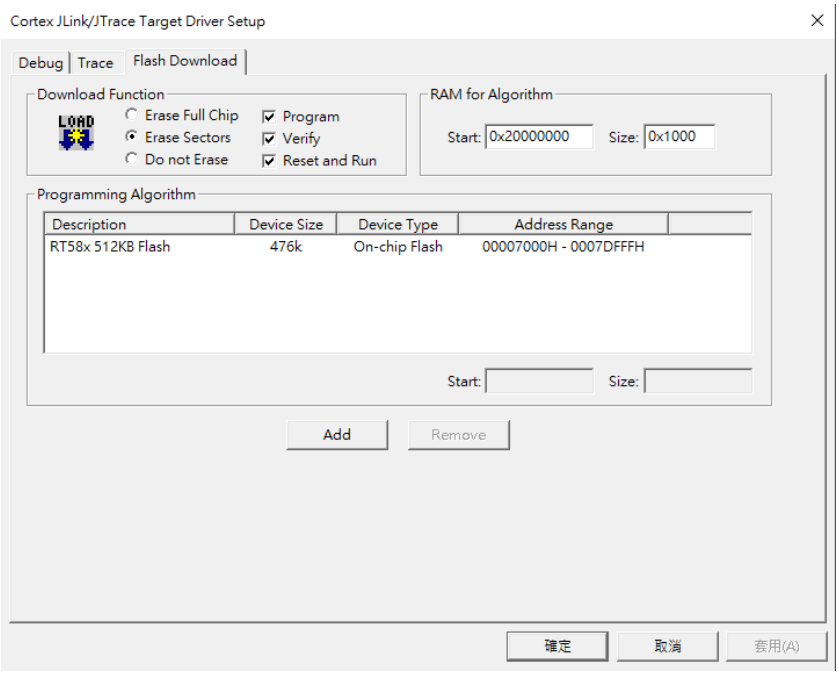


Step 2: Select the Erase Sectors, Program, Verify, Reset and Run options in the Download Function section.

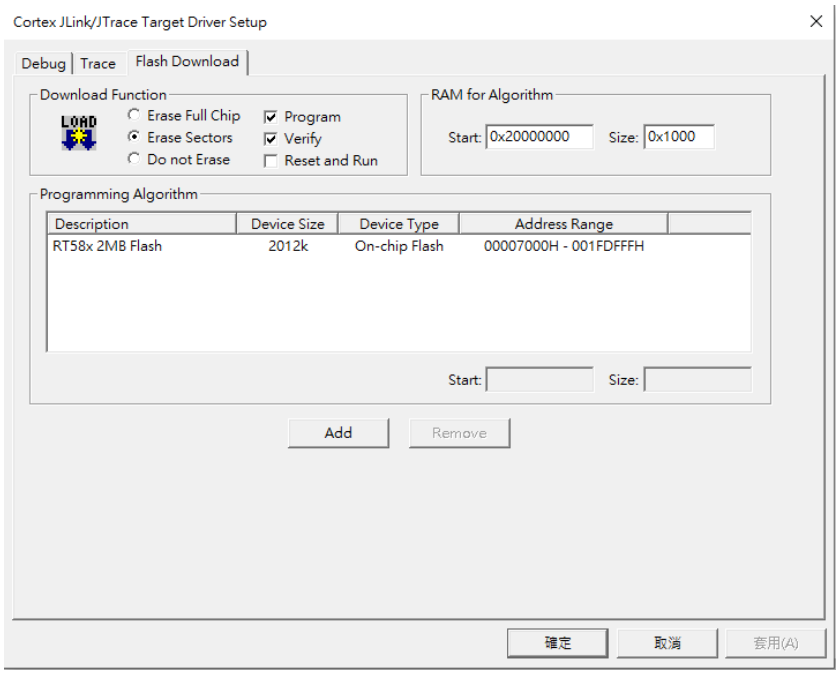
a. RT58x (1MB Flash size) Flash Programming Algorithm select



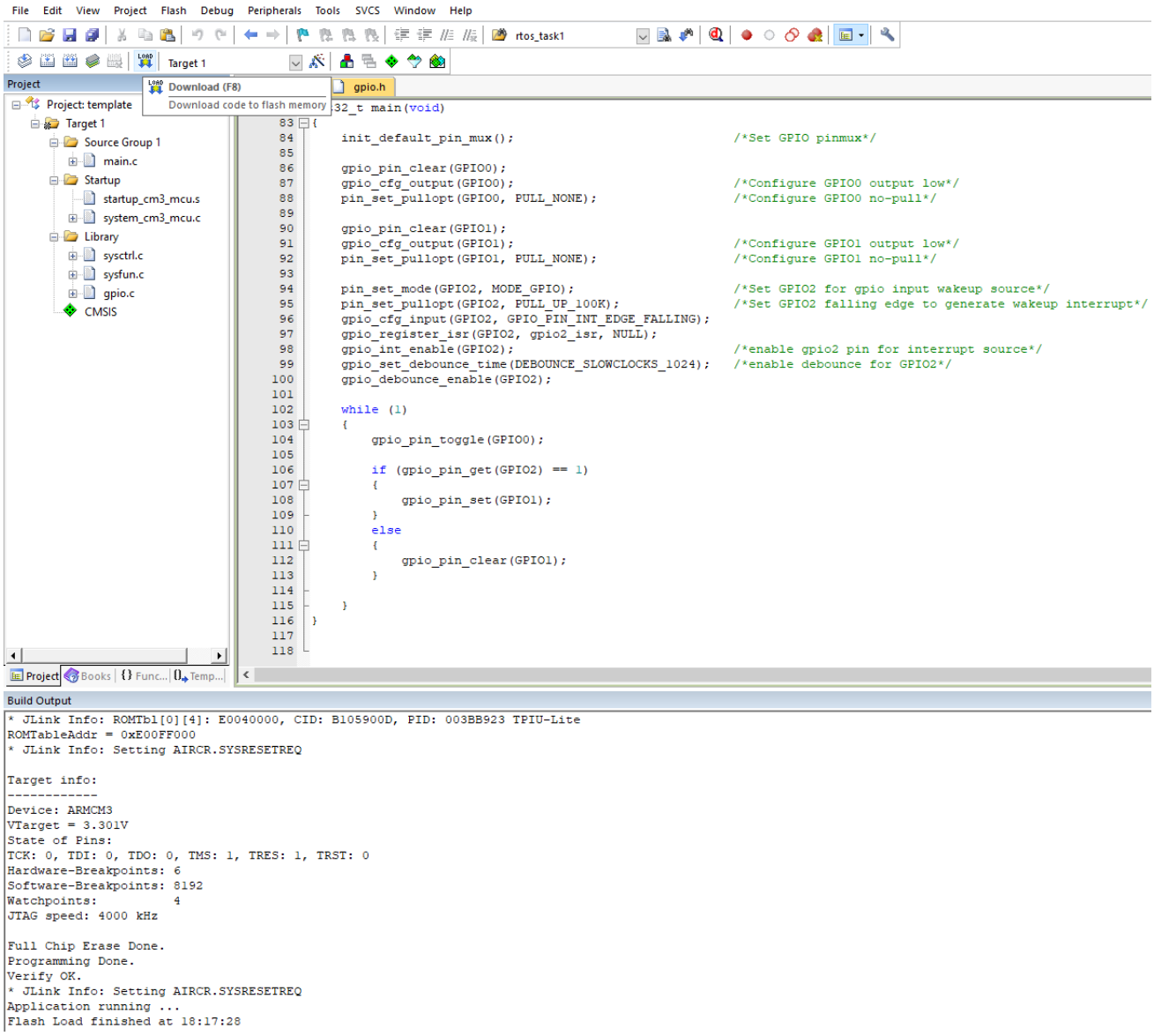
b. RT58x (512K Flash size) Flash Programming Algorithm select



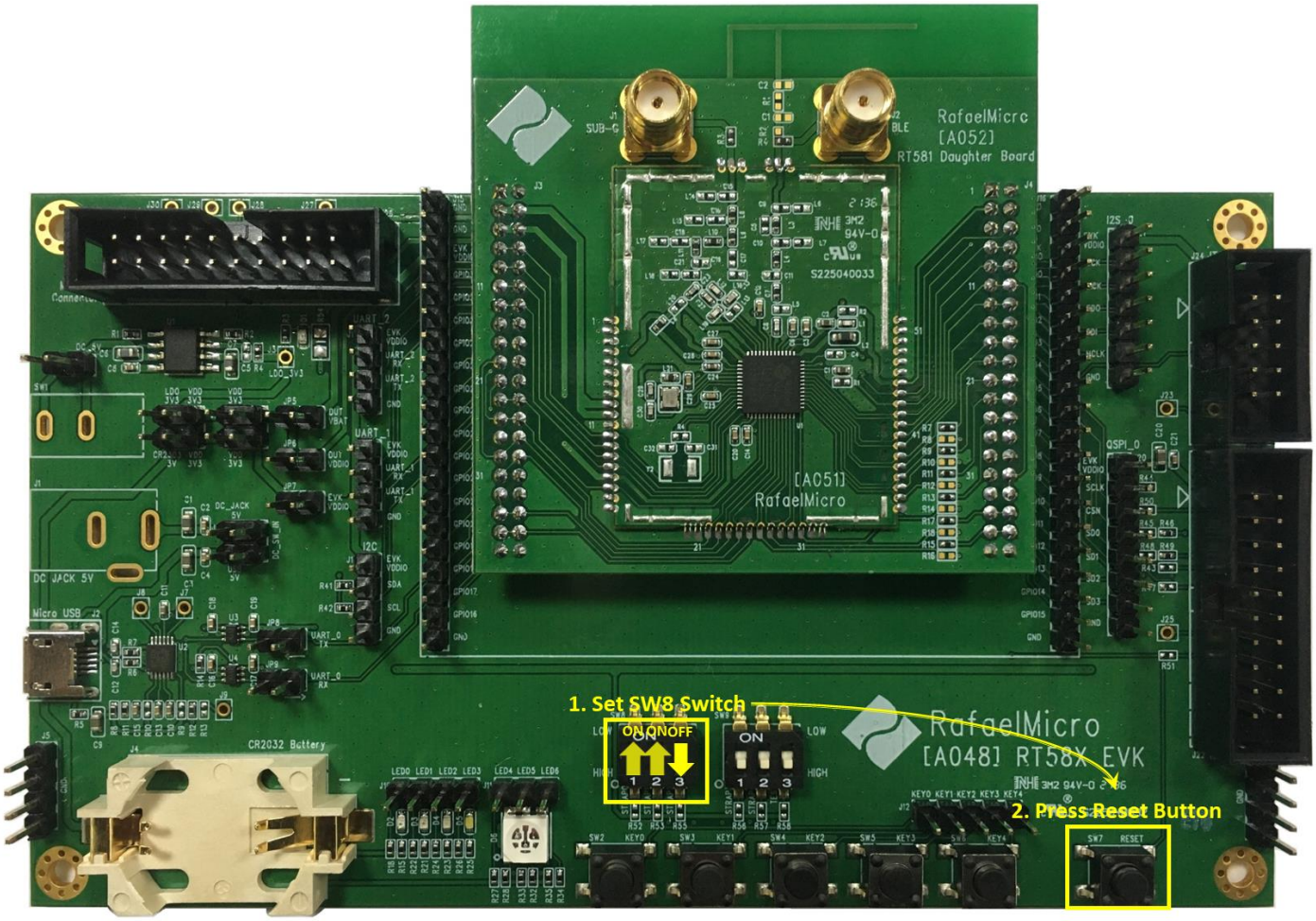
c. RT58x (2MB Flash size) Flash Programming Algorithm select



Step 3: Click the Download button to download code to Flash memory.



Note: RT58x Flash download will fail when RT58x has entered low power mode (Sleep mode or Deep Sleep mode). User can download code to Flash memory in low power mode according to the following RT58x EVK settings. RT58x EVK settings must be restored after downloading, and reset the RT58x EVK to execute the downloaded code.



4.4 Developing Applications

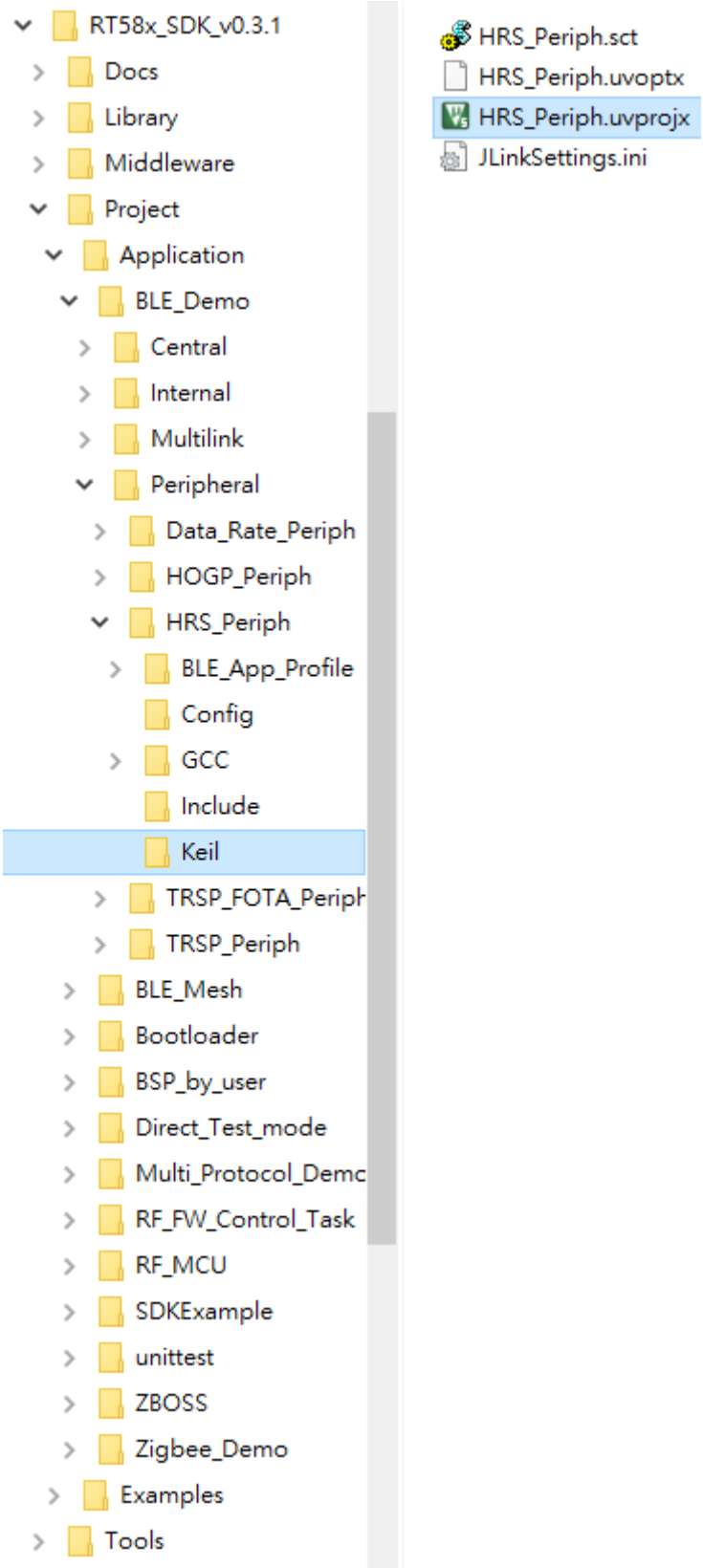
This chapter describes the development of applications using project provided by Rafael as a reference design. In addition to the introduction in the previous Creating Applications chapter, this chapter shows how to easily duplicate existing applications and examples in the Project directory of the RT58x SDK to add and modify related files to the corresponding directory to develop the required applications.

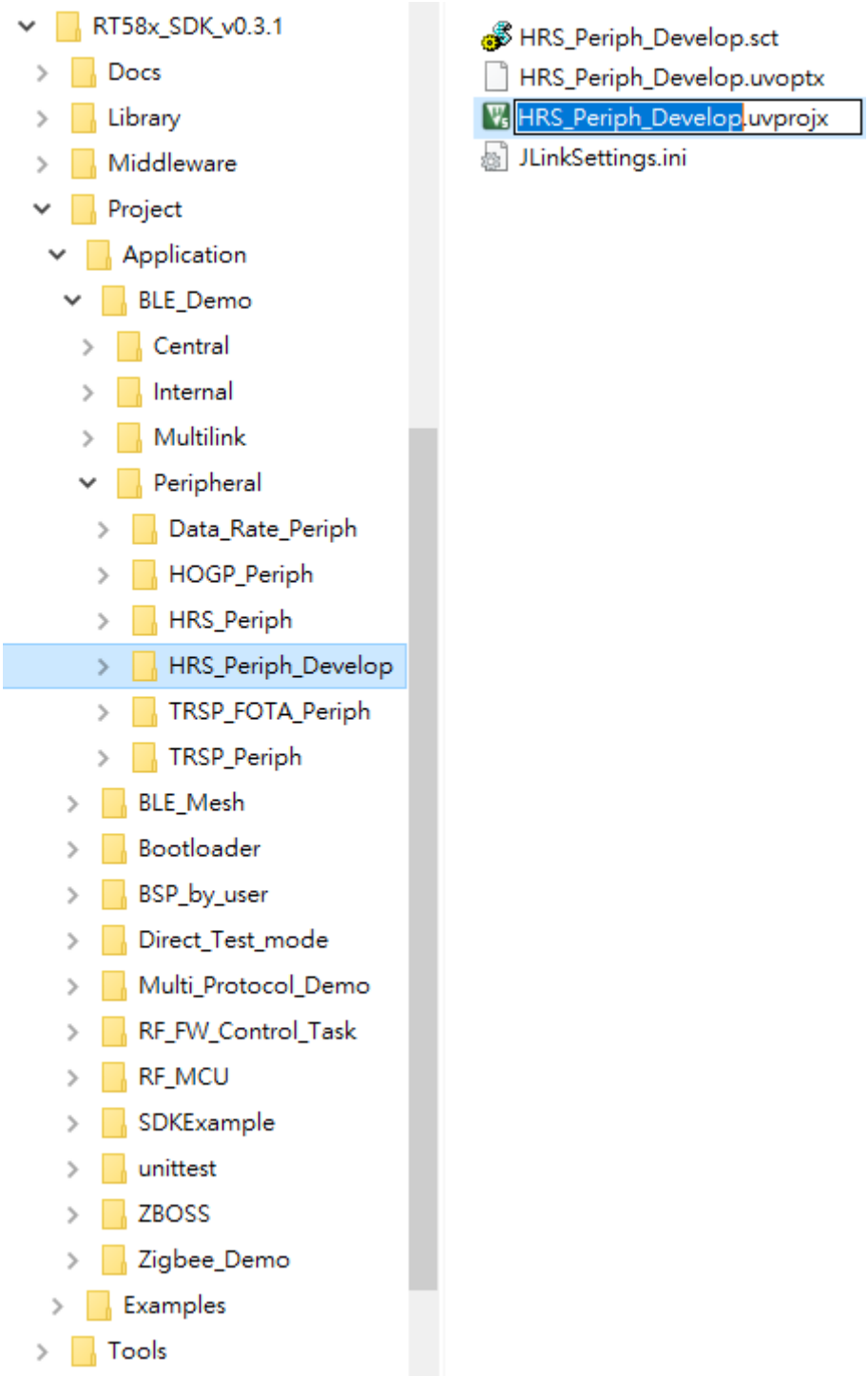
4.4.1 Bootloader

All applications and examples in the Rafael RT58x SDK support OTA feature. Before developing and executing applications and examples, the Bootloader in the \Middleware\Bootloader\ directory of RT58x SDK must be downloaded to RT58x via ISP tool or MP tool to make the applications and examples work properly.

4.4.2 Keil Projects

Rafael RT58x SDK provides rich and complete applications and examples to meet various development requirements. According to the application requirements, select the appropriate project of application or example in the Project directory and open it directly to edit the program (for example, open a HRS_Periph application in the Project directory as shown below), or duplicate it to the corresponding directory and rename the appropriate project name to develop a practical application program (for example, duplicate a HRS_Periph_Develop application from the HRS_Periph application to the same Project directory as shown below).





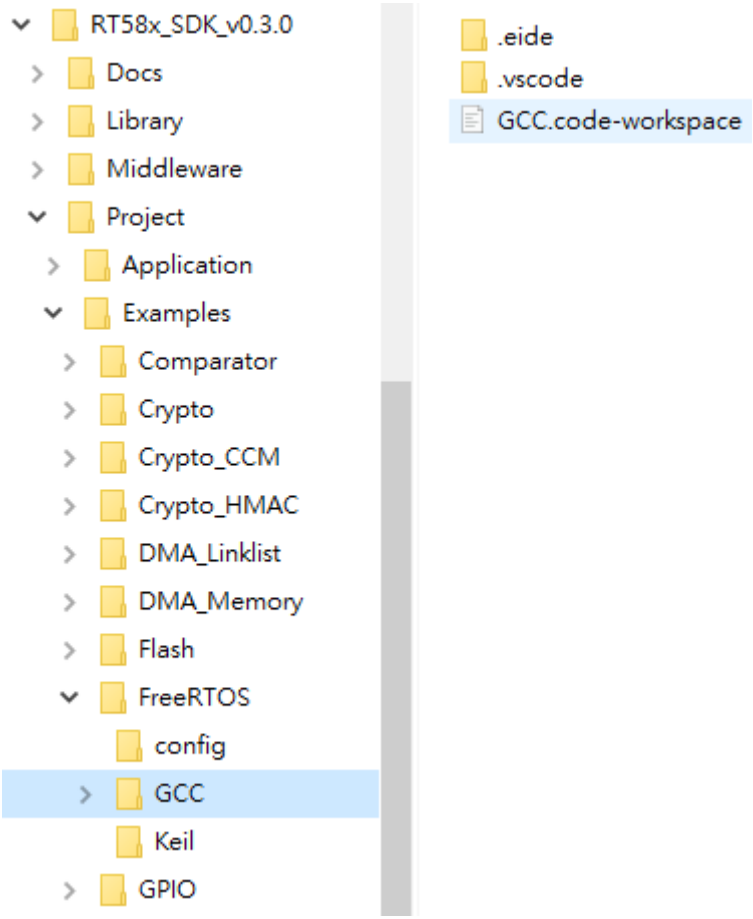
Note: If the path of the application or example in the Project directory has been moved, it may have to modify the Include Paths in the Options for Target window as described in the previous Debugging chapter.

4.4.3 GCC Projects

Rafael RT58x SDK also provides GCC projects by the Visual Studio Code corresponding to the Keil projects for applications and examples to meet various development requirements.

Before opening the GCC project, users must setup the Visual Studio Code installer for Windows. For detailed installation of Visual Studio Code, please refer to VS Code User Setup Readme.

According to the application requirements, select the appropriate project of application or example in the Project directory and open it directly to edit the program (for example, open a FreeRTOS example in the Project directory as shown below), or duplicate it to the corresponding directory and rename the appropriate project name to develop a practical application program.

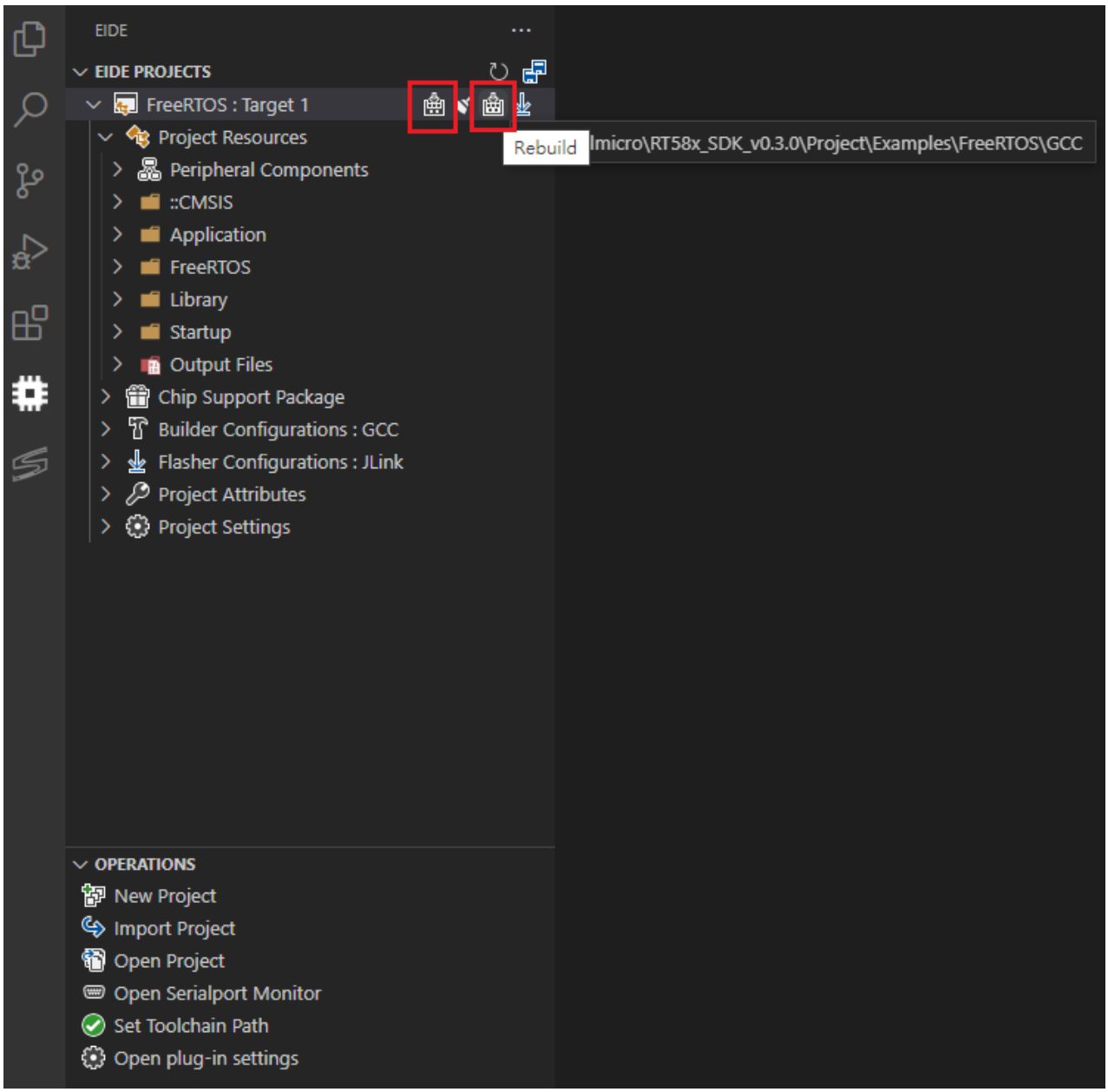


Note: If the path of the application or example in the Project directory has been moved, it may have to modify the Include Paths, please refer to VS Code User Setup Readme.

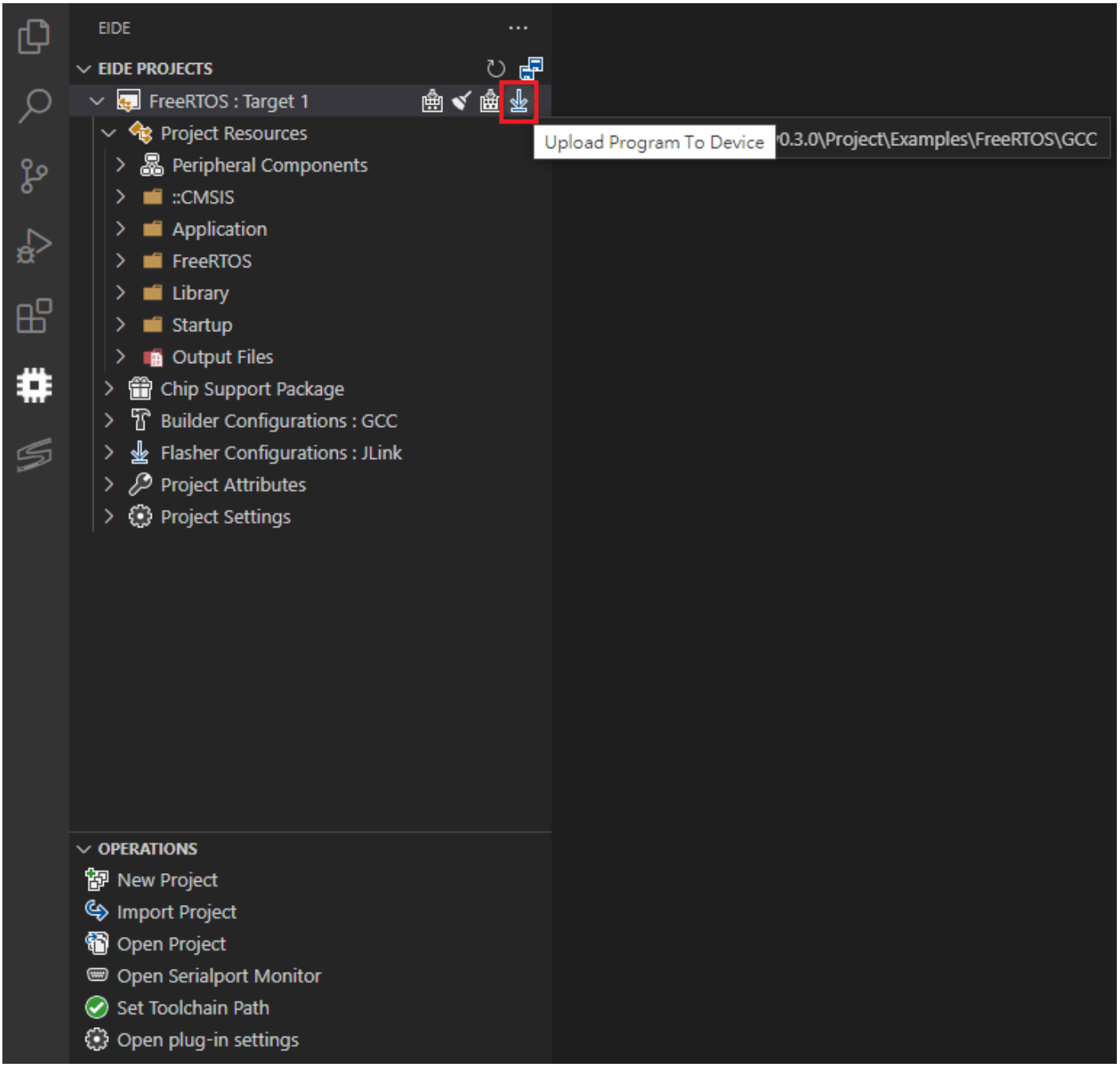
An embedded development environment (EIDE) for Cortex-M microcontroller on VS Code will

provide project development, compilation, download, and ICE debugger functions, the following will provide a step-by-step tutorial for using these functions on EIDE.

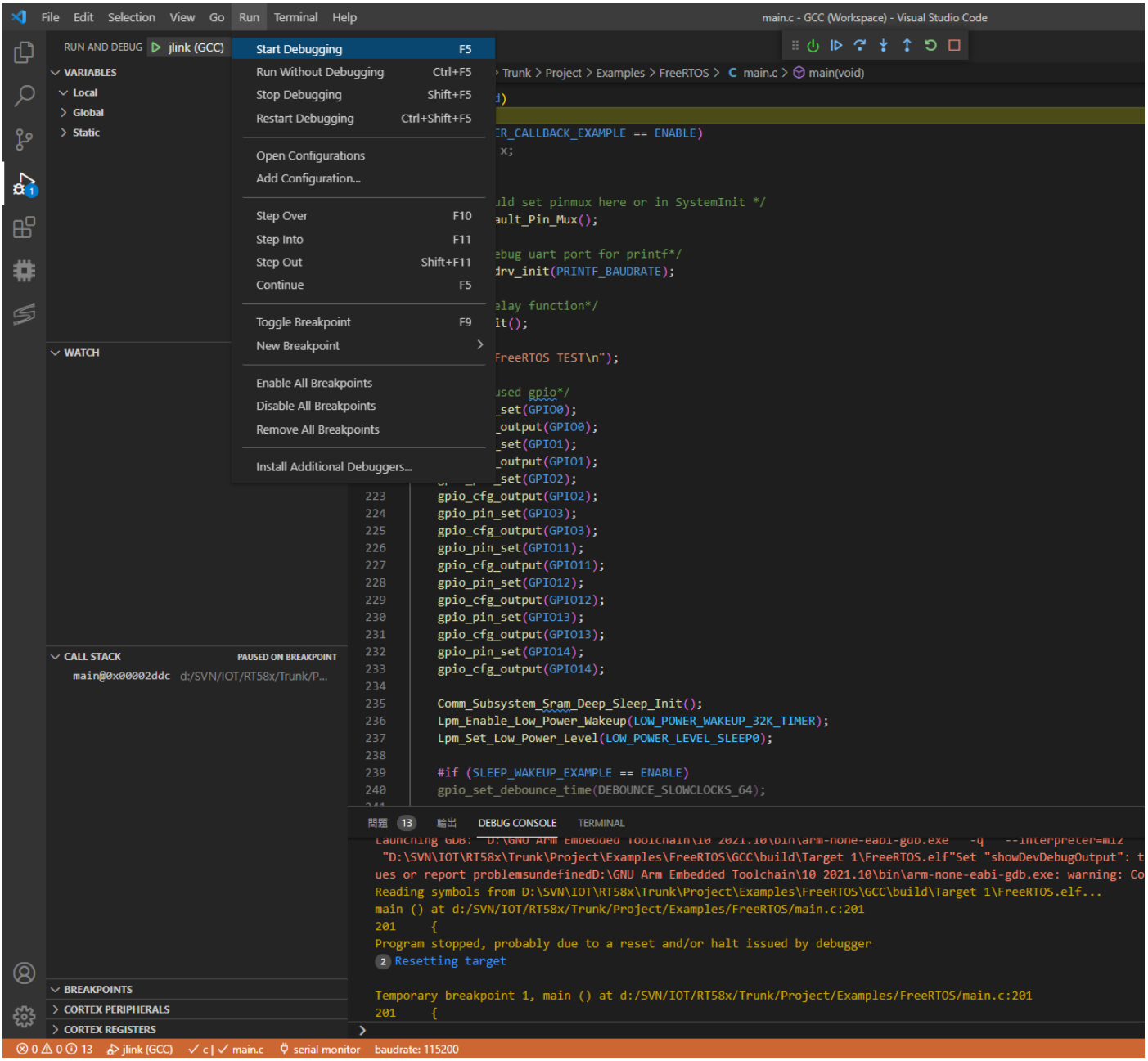
Step 1: Build the Project. Click the Build or Rebuild button to translate all source files and link the application, and generates the executable file.



Step 2: Download the Project. Click the Upload Program To Device button to download code to Flash memory.



Step 3: Start Debugging. Click the Run ➔ Start Debugging (F5) button to executes the startup code and application program, and Debug Menu and Commands are available, such as Reset device, Continue, Step Over, Step Into, Step Out, Restart, Stop, and Breakpoint, etc.



Note: The Reset command in the Debug Menu only resets the Cortex-M microcontroller, not for the RF and peripheral modules in the RT58x.

Revision History

Revision	Description	Owner	Date
1.0	Initial version.	Chiaho Hu	2021/11/05
1.1	1. Added GCC project compilation instructions. 2. Added description of Flash download in low power mode.	Chiaho Hu	2022/02/15
1.2	Revised VS Code setup note	Chiaho Hu	2022/03/28
1.3	Add Keil project select flash program algorithm	Ives Lee	2022/07/15
1.4	Add Keil project select flash program algorithm	Ives Lee	2022/05/02