Introduction

This document is intended to help developers develop ABOV 32-bit microcontrollers (Arm Cortex-M0 / M0+ / M3 / M4) in the Keil MDK Version 5 integrated development environment.

This document explains how to configure the initial development environment from the preparation of the Keil project to the environment variable setting, debugger setting, compilation, and debugging.

This document applies to the part numbers listed in Table 1.

Table 1. Applicable Devices

Base Product	Part Number
A31G21x	A31G213CL, A31G213SQ, A31G213KN, A31G213GR, A31G212CL, A31G212SQ, A31G212KN, A31G212GR
A31G22x	A31G226ML, A31G226MM, A31G226RM, A31G226RL, A31G226RL A31G224ML, A31G224MM, A31G224RM, A31G224RL, A31G224RL
A31G31x	A31G316MM, A31G316ML, A31G316RM, A31G316RL, A31G314MM, A31G314ML, A31G314RM, A31G314RL, A31G314CL, A31G314CU, A31G314SN, A31G313RM, A31G313RL, A31G313CL, A31G313CU, A31G313SN
A31G32x	A31G324RL, A31G324CL, A31G324CU, A31G323RL, A31G323CL, A31G323CU
A33G52x	A33G527VQ, A33G527VL, A33G527MM, A33G527RL, A33G527ML, A33G527RM, A33G526VQ, A33G526VL, A33G526MM, A33G526RL, A33G526ML, A33G526RM, A33G524MM, A33G524RL, A33G524ML, A33G524RM
A33G53x	A33G539VQ, A33G539VL, A33G539MM, A33G539RL, A33G538VQ, A33G538VL, A33G538MM, A33G538RL
A31M22x	A31M223CL, A31M223KN
A33M11x	A33M116RL, A33M116RM, A33M116CL, A33M114RL, A33M114CL
A34M41x	A34M418YL, A34M418VL, A34M418RL, A34M416VL, A34M416RL, A34M414VL, A34M414RL
A34M420	A34M420YL, A34M420VL
A34M456	A34M456VL, A34M456RL, A34M456RK
A34L716	A34L716VL, A34L716RL

Reference Document

The following documents are available on www.abovsemi.com.

- Reference software URL in µVision User's Guide
- User's Manual, Datasheet, Application Note for products in Table 1
- Reference software package for products in Table 1

Contents

List of Tables

List of Figures

1. Keil Project Preparation

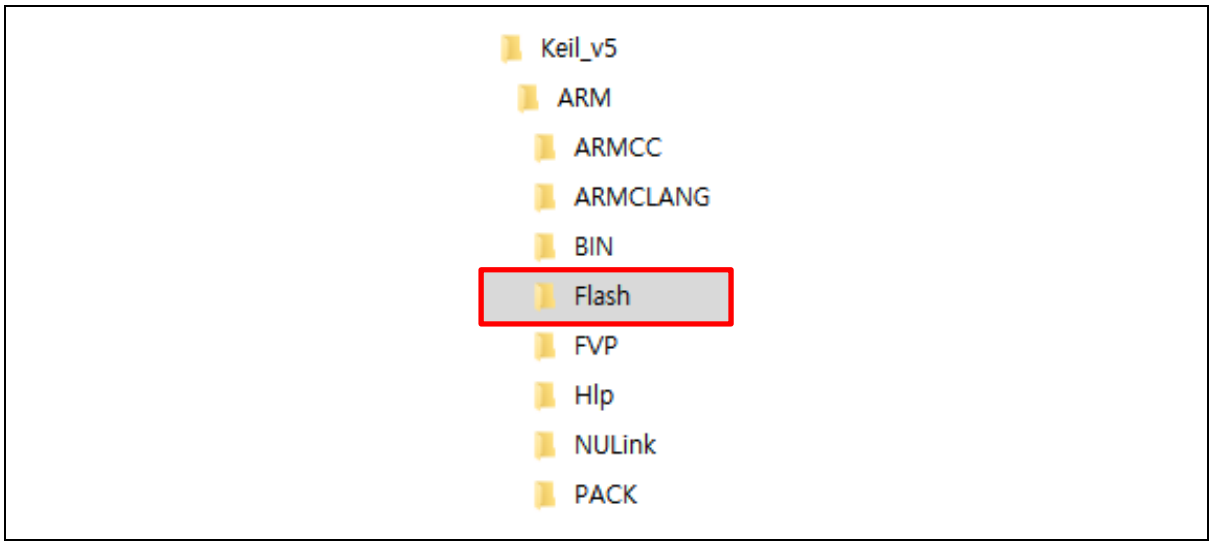
1.1 Keil MDK Installation Path

The device's **Flash Programming Algorithm** (PFA) file must exist in the designated folder of the Keil MDK installation path prior to using a project in the Keil MDK integrated development environment.

The Keil installation path and the **Flash Programming Algorithms** folder are as follows:

- Default Keil MDK installation path: \Keil_v5\ARM
- Flash Programming Algorithm path: \Keil_v5\ARM\Flash

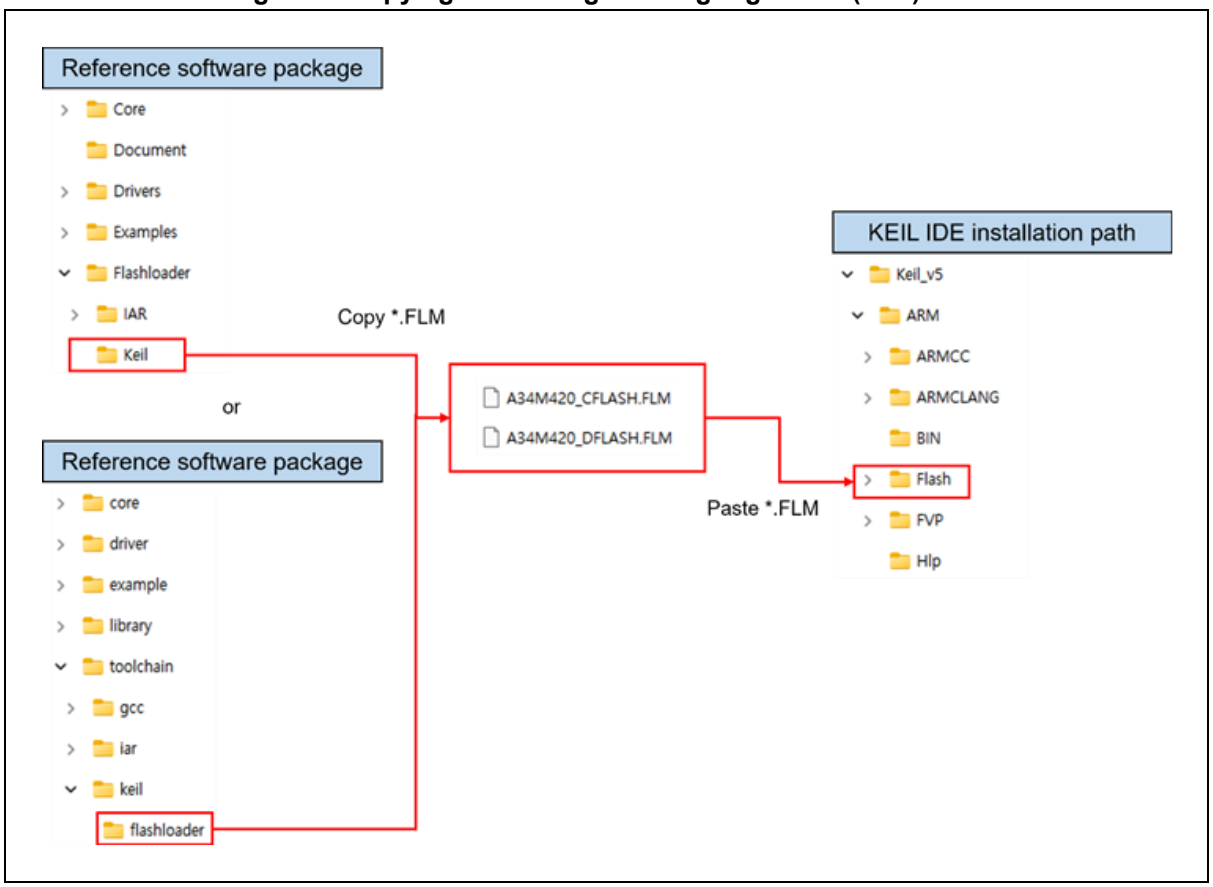
Figure 1. Keil MDK Installation Path



1.2 How to Register Flash Programming Algorithm (FPA)

ABOV 32-bit Reference Software Packages include the **Flash Programming Algorithm (FPA)** files to download firmware in Keil MDK. The component files must be copied manually prior to downloading firmware to the target microcontroller in Keil MDK. First, copy the **Flash Programming Algorithm** (FPA) files from the **Flashloader** folder in the reference software package to the **Flash** folder in the Keil MDK installed folder. Detailed information about each product's configuration is described in the following section.

Figure 2. Copying Flash Programming Algorithm (FPA) files



1.2.1 Flash Programming Algorithm (FPA) Files

The **Flash Programming Algorithm** (FPA) files of the reference software package for KEIL MDK are in the path below:

\Flashloader\Keil or \toolchain\keil\flashloader

Table 2 shows the directory name to be installed in the Keil installation path and the FPA file for the target device. Follow the procedure below to complete the copy-and-paste process:

1. Copy the FPA file provided by Reference Software Package.
2. Move into the '**Flash**' folder in the Keil MDK installation path.
3. Paste the copied FPA file into the created folder flashloader.

Table 2. Target Devices and Flash Programming Algorithm Files

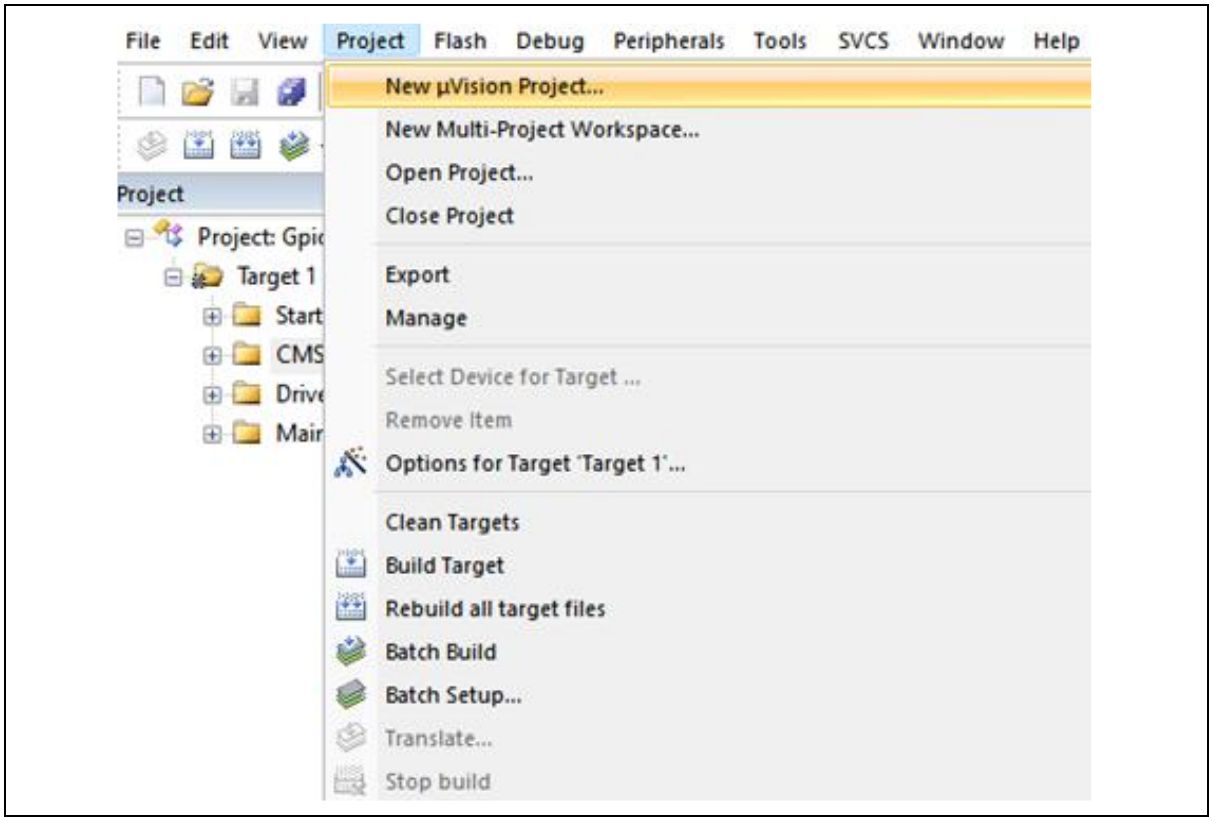
Target Device	Flash Programming Algorithm File	Code Flash Area for Algorithm	RAM Area for Algorithm
A31G213 (64 KB)	A31G213_64.FLM	0x0000_0000 ~ 0x0000_FFFF (64 KB – Size: 0x10000)	0x2000_0000 ~ 0x2000_17FF (6 KB – Size: 0x1800)
A31G212 (32 KB)	A31G212_32.FLM	0x0000_0000 ~ 0x0000_7FFF (32 KB – Size: 0x8000)	
A31G226 (256 KB)	A31G226.FLM	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	0x2000_0000 ~ 0x2000_4FFF (20 KB – Size: 0x5000)
A31G224 (128 KB)	A31G224.FLM	0x0000_0000 ~ 0x0001_FFFF (128 KB – Size: 0x20000)	
A31G316 (256 KB)	A31G316_256.FLM	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	0x2000_0000 ~ 0x2000_3FFF (16 KB – Size: 0x4000)
A31G314 (128 KB)	A31G314_128.FLM	0x0000_0000 ~ 0x0001_FFFF (128 KB – Size: 0x20000)	
A31G313 (64 KB)	A31G313_64.FLM	0x0000_0000 ~ 0x0000_FFFF (64 KB – Size: 0x10000)	
A31G324 (128 KB)	A31G324_128.FLM	0x0000_0000 ~ 0x0001_FFFF (128 KB – Size: 0x20000)	0x2000_0000 ~ 0x2000_3FFF (16 KB – Size: 0x4000)
A31G323 (64 KB)	A31G323_64.FLM	0x0000_0000 ~ 0x0000_FFFF (64 KB – Size: 0x10000)	
A33G527 (384 KB)	A33G527_Code_Flashloader.flm A33G527_Data_Flashloader.flm	0x0000_0000 ~ 0x0005_FFFF (384 KB – Size: 0x60000)	0x2000_0000 ~ 0x2000_5FFF (24 KB – Size: 0x6000)
A33G526 (256 KB)	A33G526_Code_Flashloader.flm A33G526_Data_Flashloader.flm	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	
A33G524 (128 KB)	A33G524_Code_Flashloader.flm A33G524_Data_Flashloader.flm	0x0000_0000 ~ 0x0001_FFFF (128 KB – Size: 0x20000)	
A33G539 (768 KB)	A33G539_Code_Flashloader.flm A33G539_Data_Flashloader.flm	0x0000_0000 ~ 0x000B_FFFF (768 KB – Size: 0xC0000)	0x2000_0000 ~ 0x2000_5FFF (24 KB – Size: 0x6000)
A33G538 (512 KB)	A33G538_Code_Flashloader.flm A33G538_Data_Flashloader.flm	0x0000_0000 ~ 0x0007_FFFF (512 KB – Size: 0x80000)	
A31M223 (64 KB)	a31m223_code.flm	0x0000_0000 ~ 0x0000_FFFF (64 KB – Size: 0x10000)	0x2000_0000 ~ 0x2000_1FFF (8 KB – Size: 0x2000)
A33M116 (256 KB)	A33M116_CFLASH.FLM A33M116_DFLASH.FLM	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	0x2000_0000 ~ 0x2000_3FFF (16 KB – Size: 0x4000)
A33M114 (128 KB)	A33M116_CFLASH.FLM A33M116_DFLASH.FLM	0x0000_0000 ~ 0x0001_FFFF (128 KB – Size: 0x20000)	
A34M418 (512 KB)	a34m41x_cflash.flm a34m41x_dflash.flm	0x0000_0000 ~ 0x0007_FFFF (512 KB – Size: 0x80000)	0x2000_0000 ~ 0x2000_FFFF (64 KB – Size: 0x10000)
A34M416 (256 KB)	a34m41x_cflash.flm a34m41x_dflash.flm	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	
A34M414 (128 KB)	a34m41x_cflash.flm a34m41x_dflash.flm	0x0000_0000 ~ 0x0001_FFFF (128 KB – Size: 0x20000)	0x2000_0000 ~ 0x2000_7FFF (32 KB – Size: 0x8000)
A34M420 (1024 KB)	A34M420_CFLASH.FLM A34M420_DFLASH.FLM	0x0000_0000 ~ 0x000F_FFFF (1024 KB – Size: 0x100000)	0x2000_0000 ~ 0x2000_FFFF (64 KB – Size: 0x10000)
A34M456 (256 KB)	A34M456_code.FLM A34M456_data.FLM	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	0x2000_0000 ~ 0x2000_7FFF (32 KB – Size: 0x8000)
A34L716 (256 KB)	A34L716_code.FLM A34L716_data.FLM	0x0000_0000 ~ 0x0003_FFFF (256 KB – Size: 0x40000)	0x2000_0000 ~ 0x2000_5FFF (24 KB – Size: 0x6000)

2. Create New Keil µVision Project

To create a new Keil MDK project, follow the procedure below:

1. Select the **New µVision Project** in the **Project** menu, then the file browser dialog box appears.
2. Navigate to where you want to create the new project.
3. Click New folder to create the new project folder if necessary.
4. Type the project name according to the purpose.
5. Click the **Save** button to complete the project creation.

Figure 3. New µVision Project Menu



6. Select the target processor in the **Device** tab when the **Options for Target** dialog box appears. For example, the target processor of A34M420 devices is **ARMCM4_FP** of the '**ARM Cortex M4'** in the **ARM** category (See the red box in Figure 4).

Figure 4. Selection of Target Device

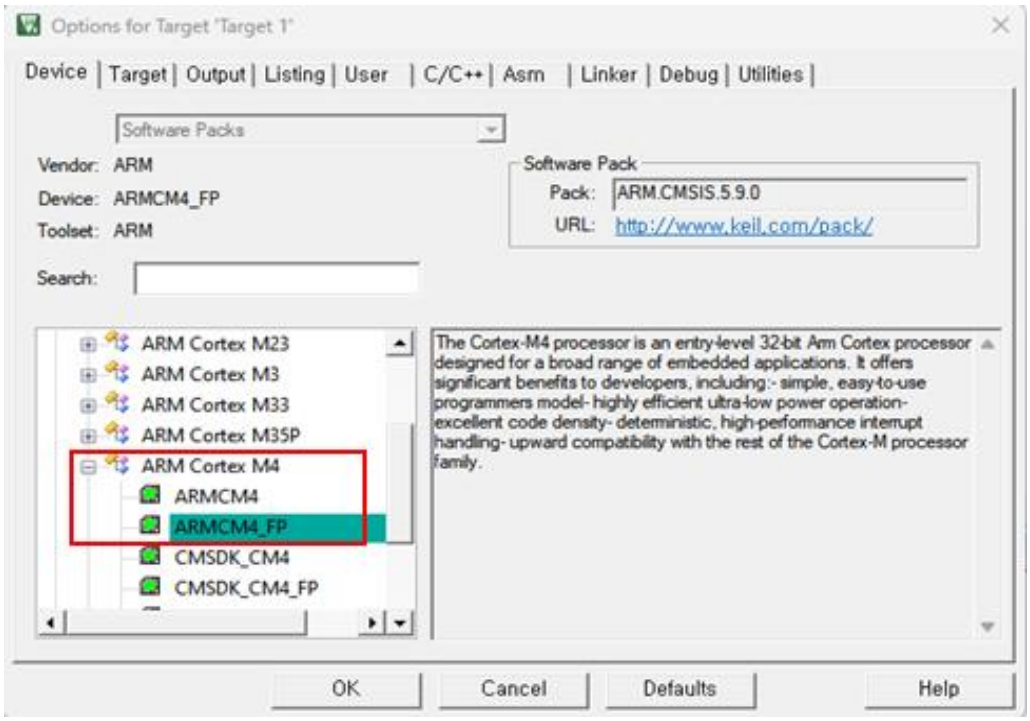
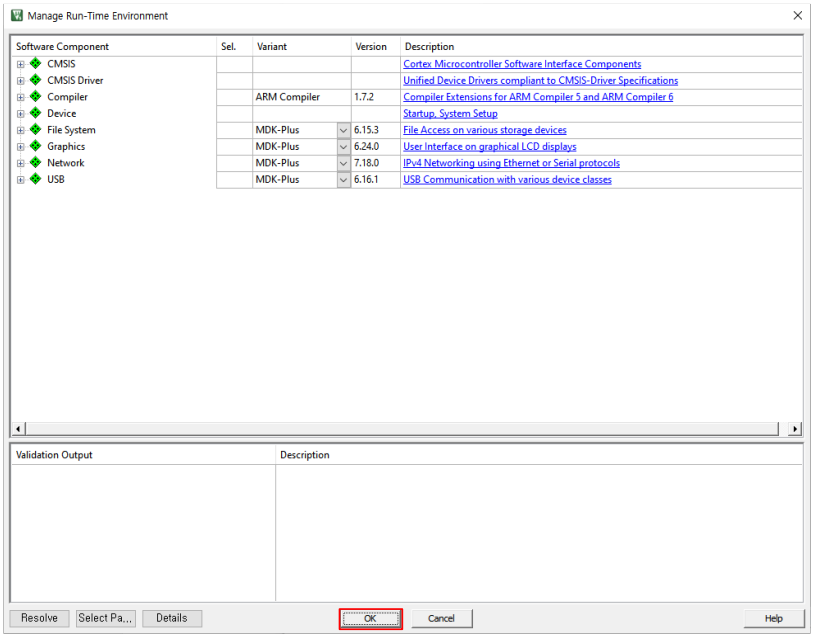


Table 3. Device Selection based on Base Product

Base Product	Category	ARM Device Name
A31G21x	ARM Cortex M0 plus	ARMCM0P
A31G22x	ARM Cortex M0 plus	ARMCM0P
A31G31x	ARM Cortex M0 plus	ARMCM0P
A31G32x	ARM Cortex M0 plus	ARMCM0P
A33G52x	ARM Cortex M3	ARMCM3
A33G53x	ARM Cortex M3	ARMCM3
A31M22x	ARM Cortex M0 plus	ARMCM0P
A33M11x	ARM Cortex M3	ARMCM3
A34M41x	ARM Cortex M4	ARMCM4_FP
A34M420	ARM Cortex M4	ARMCM4_FP
A34M456	ARM Cortex M4	ARMCM4_FP
A34L716	ARM Cortex M4	A4MCM4_FP

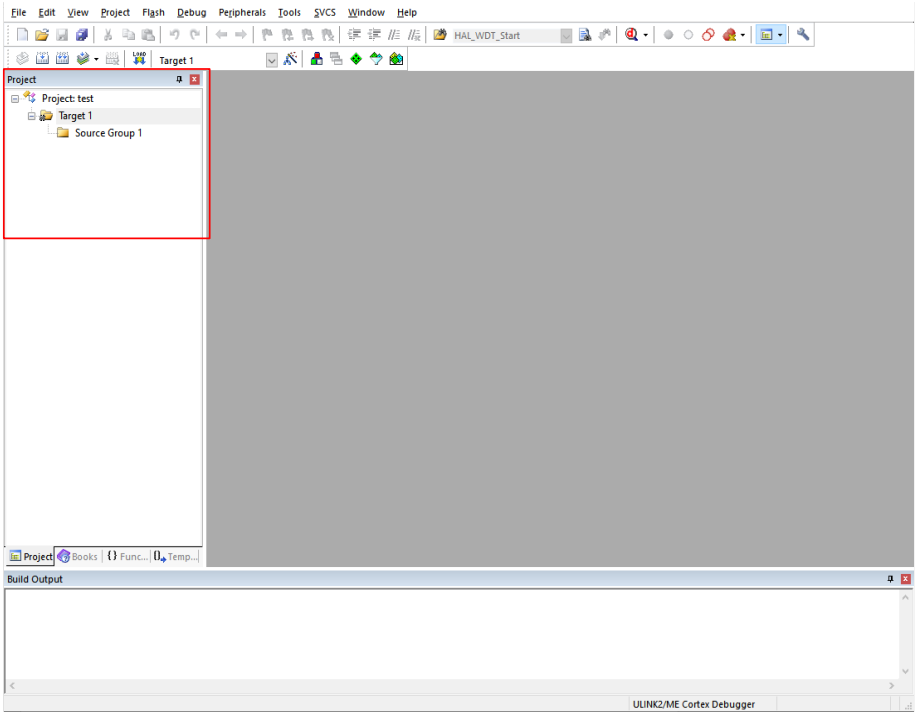
7. If the Manage Run-Time Environment window appears right after the project setup, click the OK button to complete the project creation.

Figure 5. Manage Run-Time Environment Window



8. When the project is successfully created, the project name and target name will appear in the **Project** pane.

Figure 6. Project Window and Project Pane



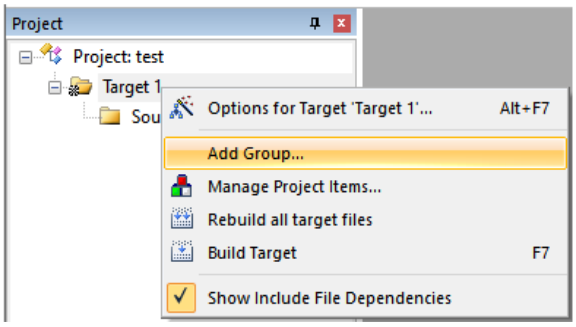
3. Add Group and Source Code to Project

This chapter explains how to add groups and source code files to the newly created Keil MDK project.

3.1 Add Group to Project

1. Right-click where you want to add a group in the Project pane.
2. Add a group by clicking **Add Group…** in the drop-down menu.

Figure 7. Add Group… in the Drop-down Menu



3. Add the group names of '**Main**', '**Startup**', '**Drivers**', and '**CMSIS**' below the target name, as shown in Figure 8. After that, source codes should be located in the appropriate group according to the purposes when adding a source code to the target. The properties of the groups are described in Table 4.

Figure 8. Adding Source Files to Group

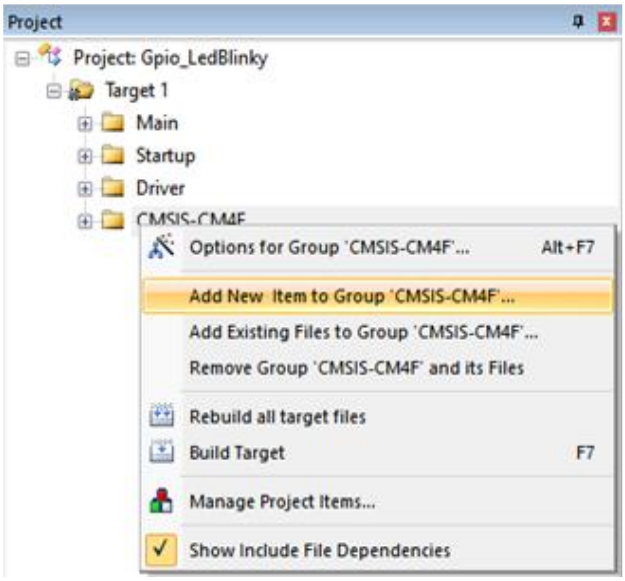


Table 4. Description of Groups in Project

Group	Description
Main	Source code that performs the main or user code
StartUp	Vector tables that are first executed immediately after boot and before entering main()
Drivers	HAL driver source code for peripheral control
CMSIS-CMx	System initialization source code

3.2 Add Source Code to the Project

To add source code to the project, double-click on the group name or right-click on the group name in the Project pane and select the one shown below from the drop-down menu.

- Add New Item to Group ‘Group name’…: to add a new item or create a file to the project.
- Add Existing Files to Group ‘Group name’…: to add a file that already exists to the project.

For example, Figure 9 shows the dialog box for adding a new item to the ‘CMSIS-CM4F’ group. To add a new file, select the file type, specify the file name and location, and click the Add button to create and add the new file to the project.

Figure 9. Add New Item to Group…

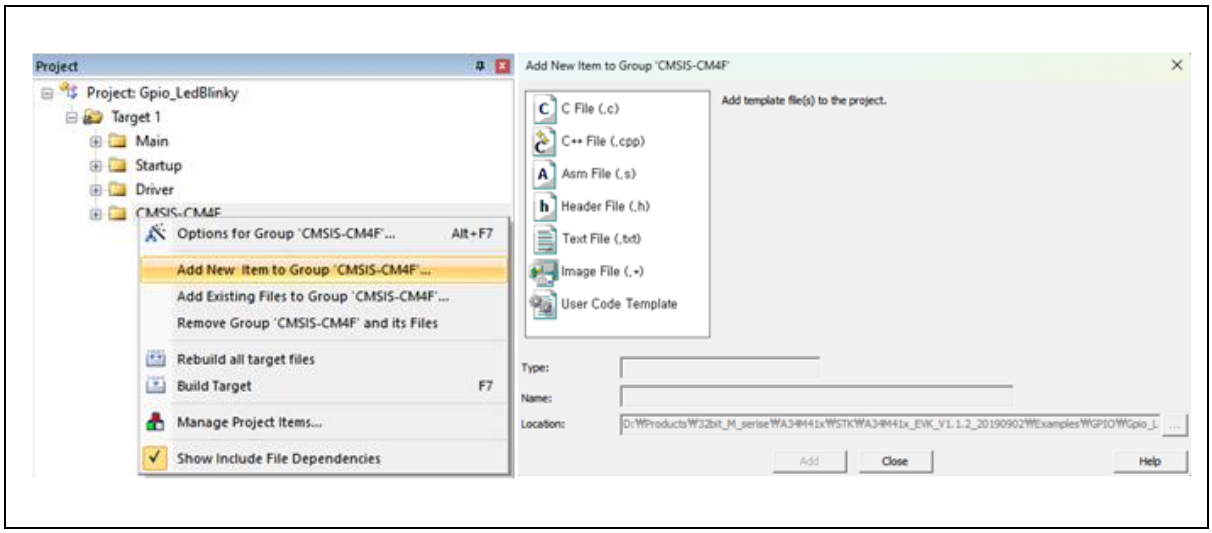
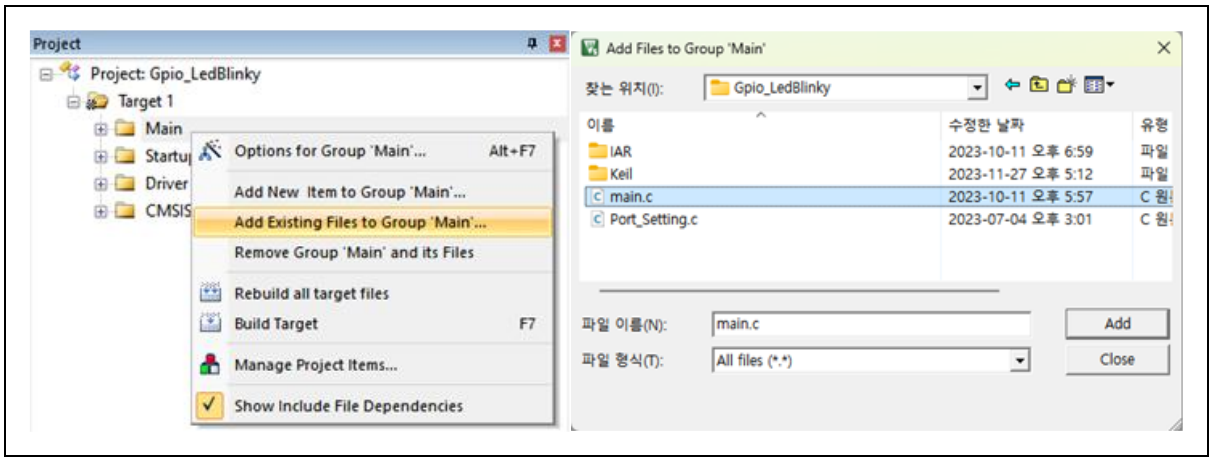


Figure 10 shows the file browser for adding the existing file to the ‘app’ group. To add existing files to the target, select the files in the file system and click the **Add** button to add the files to the project. This file browser allows you to select and add multiple files simultaneously.

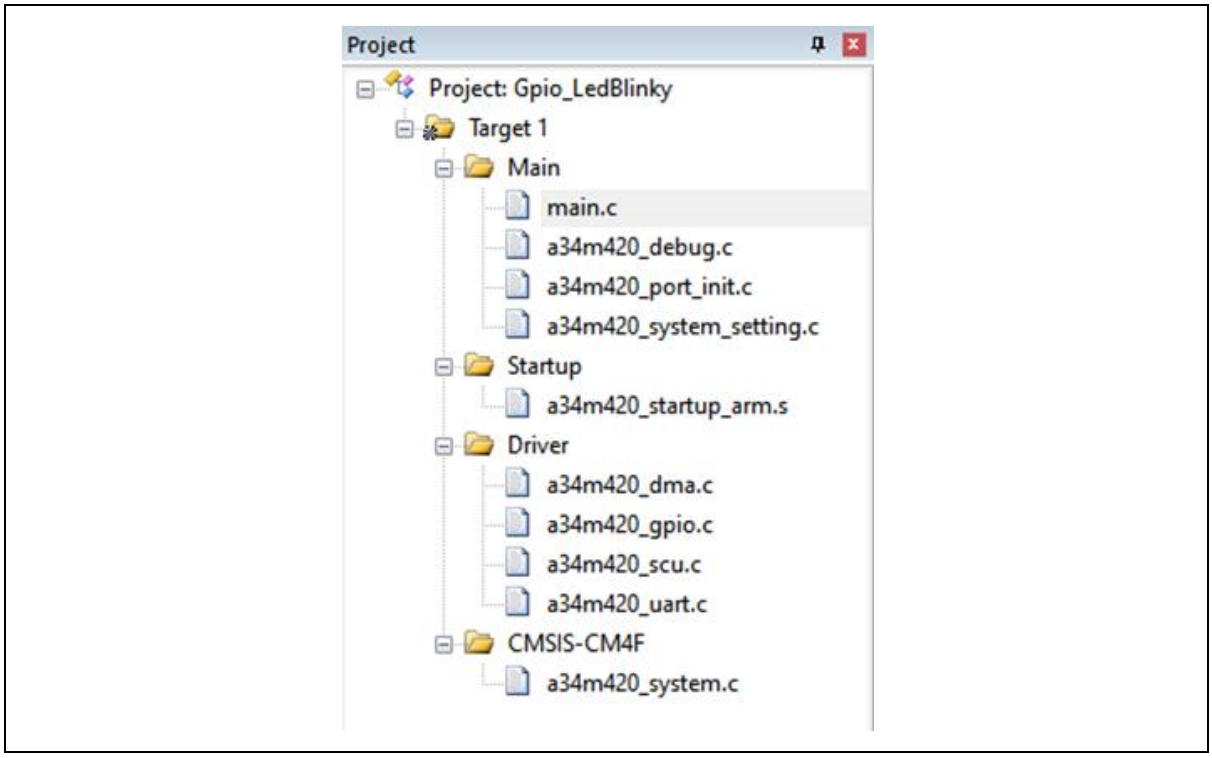
Figure 10. Add Existing Files to Group…



3.3 Target Project Example

Figure 11 shows the group and file configuration of the 'Gpio_LedBlinky' example project provided in the **A34M420 Software Package**.

Figure 11. 'Gpio_LedBlinky' example project



The groups, source codes, and file paths of the 'Gpio_LedBlinky' example project are described in Table 5.

Table 5. Description of Groups in Project

Group	Source Code (Example)	Relative File Path (Example)	Description
Main	main.c a34m420_debug.c a34m420_port_init.c a34m420_system_setting.c	\example\starterkit\Gpio_LedBlinky	Workspace for project development
Startup	a34_m420_startup_arm.s	\core\device\a34m420\source	CMSIS Core Device Startup File for ABOV Target Device
Driver	a34m420_dma.c a34m420_gpio.c a34m420_scu.c a34m420_uart.c	\driver\a34m420\source	Device driver interface files that allow the program to communicate with the ABOV Target Device
CMSIS-CMx	a34m420_system.c	\core\device\a34m420\source	System configuration File for ABOV Target Device

The Keil MDK projects in the **Reference Software Package** consist of the following folders. In Figure 12, the root folder contains nine folders, and a description of each folder is as follows:

- Core
  - The CMSIS header files are provided as a standard for the Cortex-M architecture, and the device-specific C and assembly files are located.
- Document
  - The documents explaining the operation, arguments, and return value of each function in the HAL drivers are located.
- Driver
  - The C source codes and header files of the HAL driver are located.
- Examples
  - The example codes for each peripheral using the HAL driver are located. You can run the example for the peripheral.
- Flashloader
  - Files related to the debugger interface settings and flash downloader for each development environment are located.
- Ini
  - Files for initializing the target device, including ROM, RAM, Stack Pointer, Program Counter, etc., are located.
- SFR and SVD
  - Files used in the system viewer to inspect or modify peripheral registers during debugging are located.
- Release
  - Contains text files related to software version information, bugs, issues, fixes, etc.

Figure 12. Directory Structure of Reference Software Package project

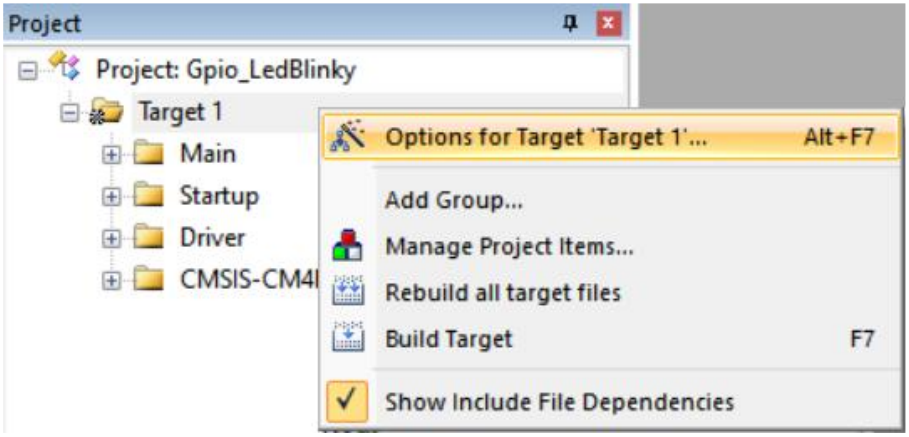
NOTE:

1. The reference software package folder structure may vary in part depending on the target device.

4. Keil MDK Environment Settings

This chapter describes the project option settings for compiling and debugging the Keil MDK project. Select the '**Options for Target…**' in the Project menu to open the '**Options for Target**' dialog box.

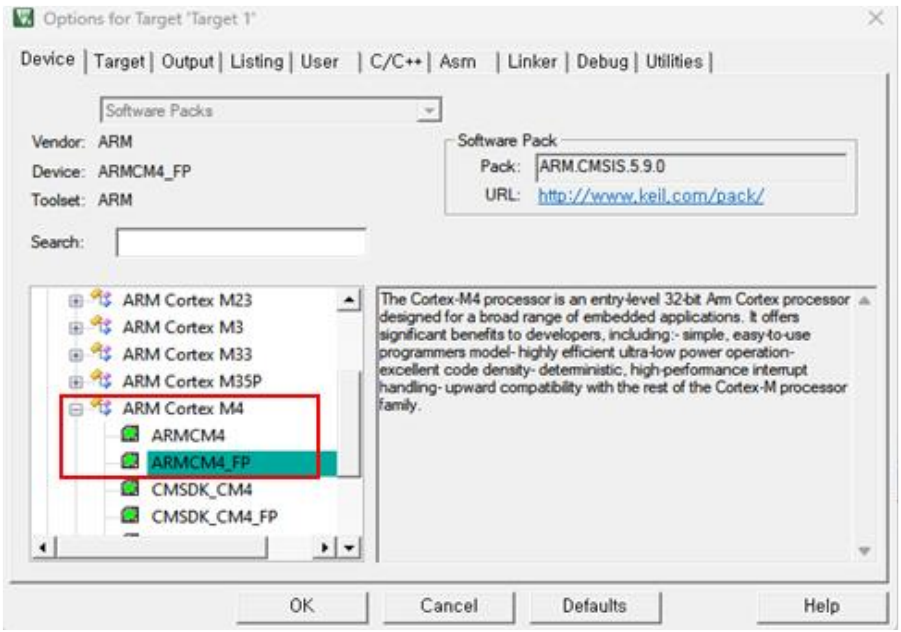
Figure 13. Keil MDK Project Option Menu



4.1 Device

You can select the target device on the 'Device' tab, as shown in Figure 14. If you need to use flash memory separately, check and enter the size of flash memory you need to use as follows.

Figure 14. Target Device Selection on the 'Device' tab



4.2 Target

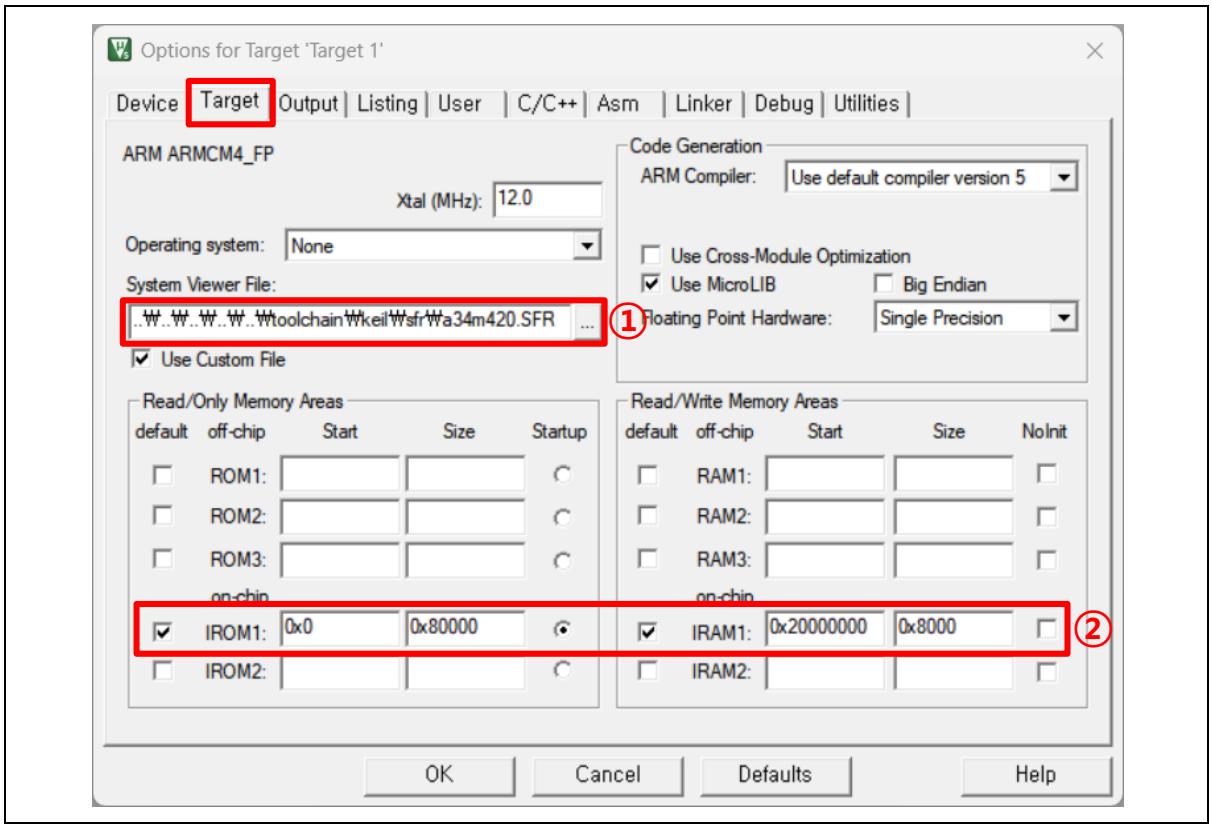
You can set the target device's memory information and SFR file on the **Target** tab, and each setting is described below:

1. System Viewer File:
   A. The SFR file is to be loaded from the path '\sfr'. The SFR file is the information file for displaying and controlling peripheral registers in the **System Viewer** when debugging.
2. Read/Only Memory Areas and Read/Write Memory Areas:
   A. Adjust the size of ROM and RAM according to the microcontroller memory specification.(1)
      i. IROM1: start 0x0000_0000, size 0x0008_0000
      ii. IRAM1: start 0x2000_0000, size 0x0000_8000

NOTES:

1. If the memory setting is incorrect, errors may occur when downloading firmware to the target device. Please refer to Table 2.
2. The size of the IROM is generally the same as the size of the Code Flash area of the target device.
3. However, in the case of a device that can separate the bank, please use it according to the size of the bank. If banks are not divided and used, all code flash areas can be used.
   Ex) A34M420 Code Flash size = 0x0010_0000, Bank size = 0x0008_0000,
   Not used bank function: IROM size: 0x100000, using the bank function: IROM size: 0x80000.)

Figure 15. SFR File and Memory Configurations on the Target tab

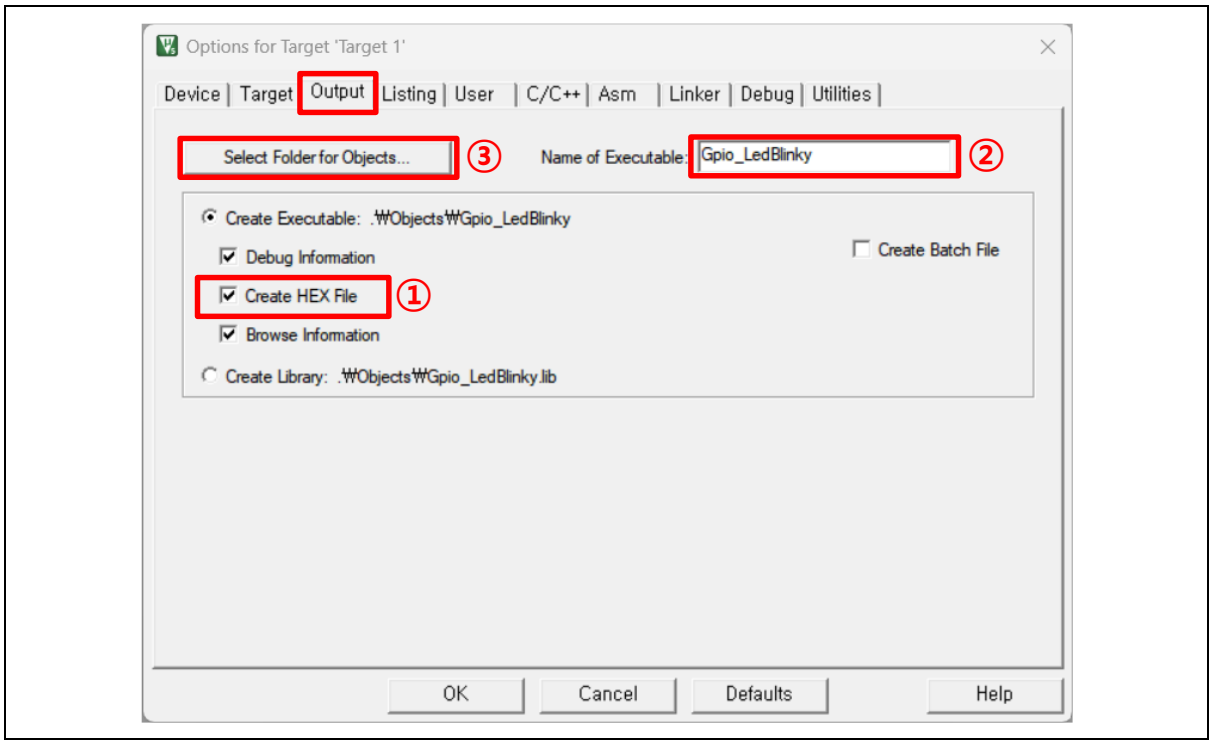


4.3 Output and Listing

You can create an Intel HEX Format file from the output executable of the project on the Output tab by following the procedure below:

1. **Create HEX file**: Check the **'Create HEX File'** option to create a hex extension file if you do not check this option, only binary files with **'*.axf'** extensions are created.
2. **Name of Executable**: Type the name of the executable file of the build output.
3. **Select Folder for Objects…**: Set the path for the object files of the build output. By default, the **'Objects'** and **'Listings'** folders are automatically created after the build. These folders contain files such as objects, maps, and project build results.

Figure 16. Option Settings on the Output tab



4.4 C/C++

This section describes the C/C++ compiler settings in the C/C++ tab shown in Figure 17.

In this tab, you can configure C/C++ compiler settings such as **Preprocessor Symbols, Language/Code Generation, Include Paths,** etc. For the basic settings of the reference software package, follow the procedure below:

1. **Optimization**: There are four levels of optimization options, and the higher the level, the more the compiler optimizes the code with an optimization strategy for size or speed efficiency.
2. **Include Paths**: To add or remove the include path for compiling, click the button on the right of the current include path. Then, the **Folder Setup** dialog box will appear on the right side. The compiler will refer to the **Include Paths** in the list to find header and source files included in the project while compiling.
3. **Preprocessor symbols, Define**: Define the preprocessor symbols for the whole project files. This option is the same with #if, #ifdef, or #ifndef in the source code.

Figure 17. Optimization Level and Include Paths Settings on the 'C/C++' Tab

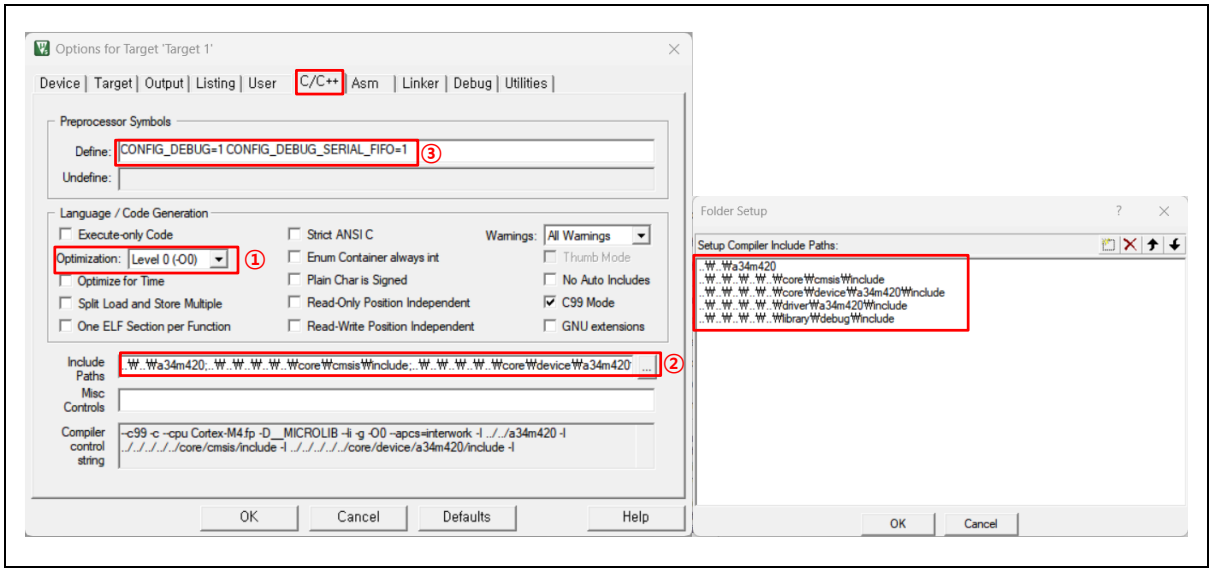


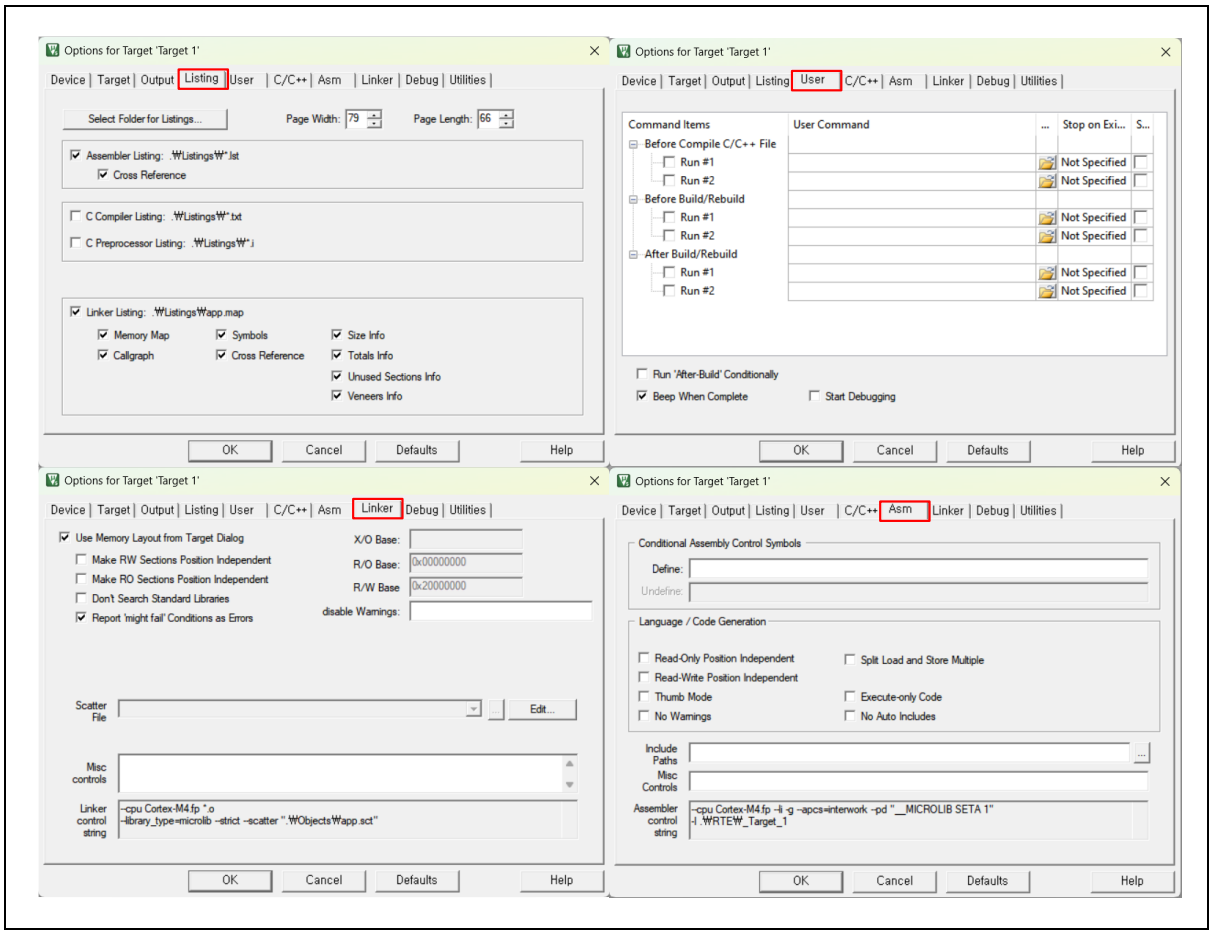
Table 6. Icons in Folder Setup Dialog Box

Icon	Menu	Description
	New (Insert)	Inserts new include paths of the header file.
	Delete	Deletes the selected include paths of the header file.
	Move up	Moves the order of the selected include paths of the header file upwards.
	Move down	Moves the order of the selected include paths of the header file downwards.

4.5 Listings, User, Linker and Asm

You can set the other options, such as the Listings, User, Linker, and Asm tabs, and default values are shown in Figure 18.

Figure 18. Option Tabs of Listing, User, Linker, and Asm



4.6 Debug

To debug the target ABOV 32-bit microcontrollers in the Keil MDK project environment, a debugging tool that supports the CMSIS-DAP interface is required.

- 32-bit Microcontroller Starter Kit Board
- ABOV A-Link / A-LinkPro Debugger(CMSIS-DAP compatible)
- ARM KEIL ULINK2
- Segger J-Link

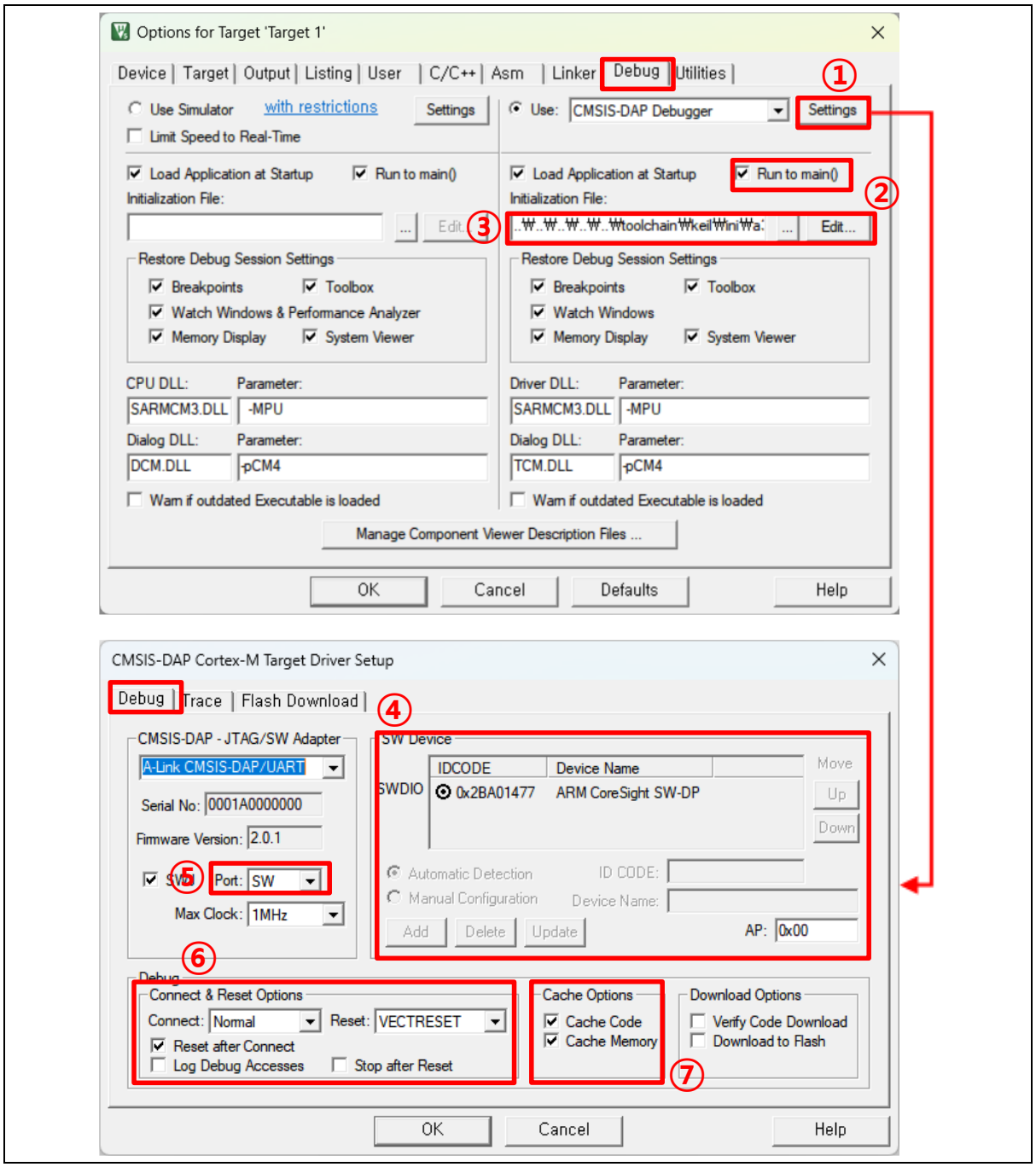
As shown in Figure 19, you can configure the debugging options on the **Debug** tab by following the procedure below:

1. **Use Simulator** or **Use Debugger**: You can choose between **Use Simulator** and **Use Debugger**. Check **Use Simulator** to simulate with a virtual device or **Use Debugger** to debug a target device directly. If you select **Use Debugger**, find your debugger's name from the drop-down menu, and select the debugger.
2. **Run to main ()**: This option determines whether to start with main () or startup code (*.s) at the startup sequence after booting.
3. **Initialization File**: Click the **Edit…** button to set the initialization file (*.ini) of the SP (Stack Pointer) and PC (Program Counter). The *.ini file path is '~/toolchain/keil/ini'.
4. **Settings**: As clicking the **Settings** button, the **CMSIS-DAP Cortex-M Target Driver Setup** dialog box will appear. Then check the **SW Devices** section and if your device appears in that section, the debugger is successfully connected to the target device.
5. **Port**: Select **SW** because 32-bit microcontroller devices support the SWD interface.
6. **Connect & Reset Options**: In the **Connect & Reset Options** section, set the option values appropriately from the **Connect** and **Reset** drop-down lists. Some devices must use **VECTRESET,** and other devices recommend **SYSRESETREQ** in Reset drop-down lists (see Table 7).
7. **Cache Options**: Select the appropriate settings from the **Cache Options** section (See Figure 19).

Table 7. Reset Type Selection Guide by Device

Reset Type	Device
SYSRESETREQ	A31G21x, A31G22x, A31G31x, A31G32x, A33G52x, A33G53x, A31M22x A33M11x, A34M456
VECTRESET	A34M41x, A34M420, A34L716

Figure 19. Option on Debug Tab and Target Driver Setup



4.6.1 Debug: Connect & Reset Options and Cache Options

Table 8 describes the options set on the **Debug** tab in the **CMSIS-DAP Cortex-M Target Driver Setup** dialog box shown in Figure 20. These options are used to configure settings when entering debug mode.

Figure 20. "Connect & Reset Options" and "Cache Options" in Debug Mode

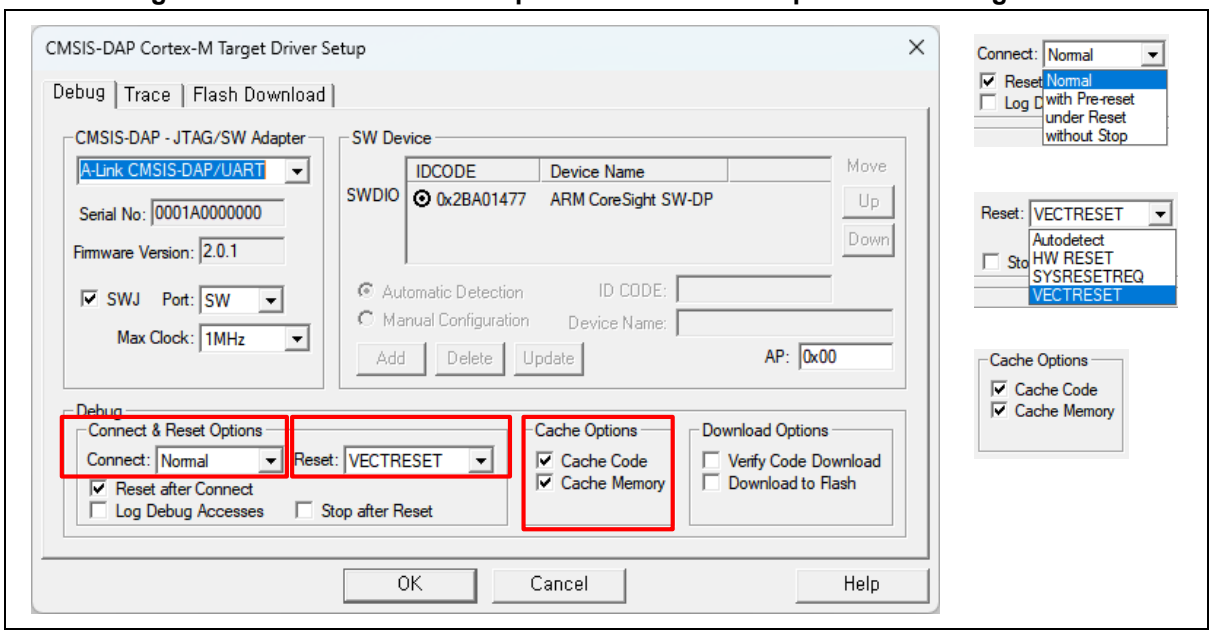


Table 8. Description of "Connect & Reset Options" and "Cache Options"(1)

Option		Description
Connect	Normal	Stops the CPU immediately after connecting the device and the debugger.
	With Pre-reset	Applies the Hardware Reset (HW RESET) before connecting the debugger to the device.
	Under Reset	Holds the Hardware Reset signal active while connecting the debugger to the device.
	Without Stop	Connects to and disconnects from the target device without explicitly stopping the CPU. This option is used to inspect memory or peripheral SFRs.
Reset	Autodetect	Automatically selects one of the Reset methods based on the target device.
	HW RESET	Performs a Hardware Reset by asserting the HW RESET signal.
	SYSRESETREQ	Performs a Hardware Reset by setting the SYSRESETREQ bit. It is typically recommended for most devices.
	VECTRESET	Performs a Software Reset by setting the VECTRESET bit. Some devices must use this mode (See Table 7).
Cache	Cache Code	Informs the debugger that the code area will not be changed.
	Cache Memory	Determines whether to update the memory display while the program stops.

NOTE:

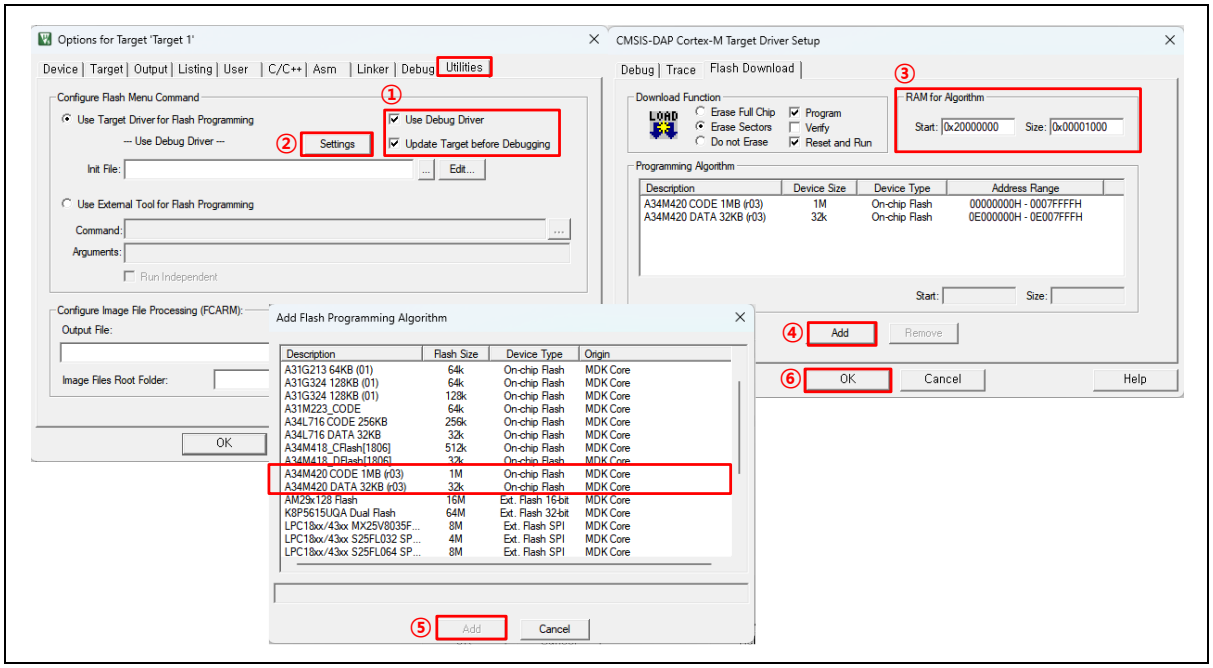
1. For more information, see ARM ULINK2 User's Guide.

4.7 Utilities

Set the **Configure Flash Menu Command** options on the **Utilities** tab by following the procedure below:

1. Check the **Use Debugger Driver** and the **Update Target before Debugging** options on the **Utilities** tab.
2. Click the **Settings** button to open the **Cortex-M Target Driver Setup** dialog box for setting the flash download options.
3. Set the start address and size in the **RAM for Algorithm** field on the **Flash Download** tab according to the device specification. Incorrect values in this field will cause flash programming to fail. (See section 0 for information on the values set in the **RAM for Algorithm** field.)
4. Click the **b** button on the **Flash Download** tab and select the Flash Programming Algorithm file of the target device in the **Add Flash Programming Algorithm** dialog box. After copying the Flash programming algorithm files in section 1.2, you can find the target device name on the **Add Flash Programing Algorithm** dialog box list.
5. Click the **Add** button in the **Add Flash Programming Algorithm** dialog box.
6. On the **Flash Download** tab, ensure that the desired device name is in the **Programming Algorithm** field and click the **OK** button to save the settings.

Figure 21. Flash Download Settings



5. Build and Debug Keil MDK Project

This chapter describes the compilation process, download process, and debugging process of the target device using the Keil MDK project.

Follow the procedure below to build and download programs:

1. Select the **Rebuild All** command shown as ① in Figure 22 to build the target project.
2. Figure 22 shows the result when the build process is completed. Check the build log window to see if there is an error in the source code.
3. When you click this download button, the compiled information is transferred to **ULINK2** or **ABOV A-Link CMSIS-DAP** (Debug interface) via a USB connector. Then, the code is downloaded to the flash memory of the target device.
4. When you click this debug button, the compiler enters debug mode. You can detect some errors and identify the causes by using debug commands.

Table 9. Project Build Command of Keil Project

Icon	Command	Description
	Build	Compiles only the edited source codes and builds the project.
	Rebuild	Builds the entire project.

Figure 22. Keil Project Build Result

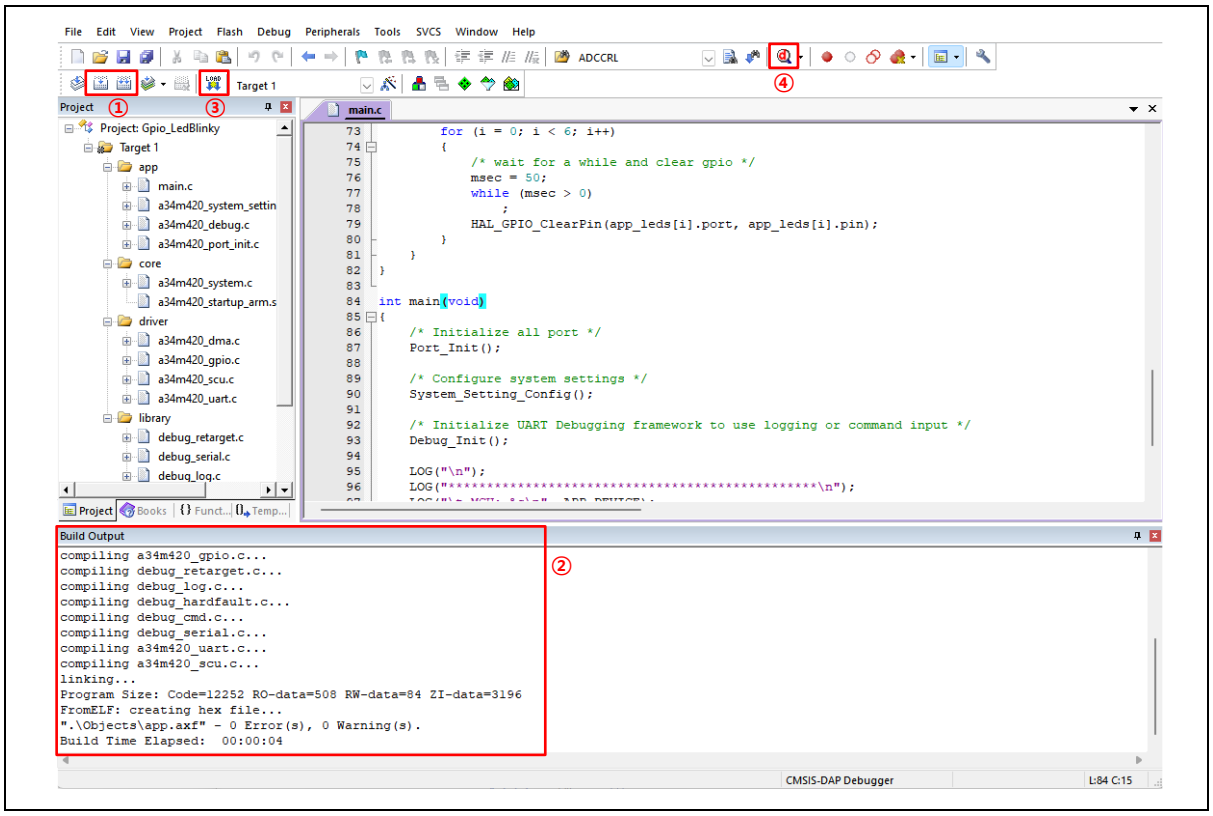


Figure 23 shows the debugging screen.

After downloading the program, you can execute and debug the program from the debugging screen.

Select the **Run** command from the **Debug** menu to run the program and select the **Break** command to stop the program operation. You can also run the program step by step through the **Step** command options.

Figure 23. Keil MDK Project Debugging Screen

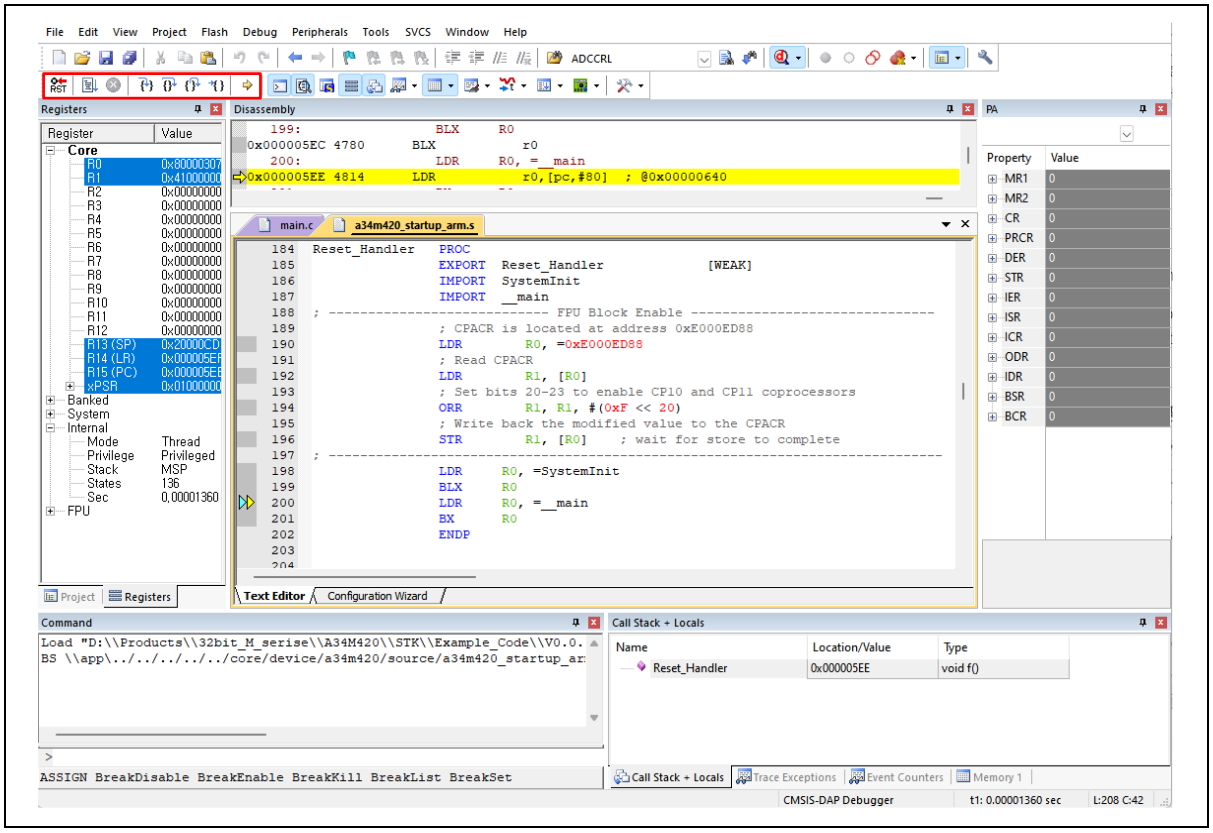


Table 10. Debug Commands of Keil Project

Icon	Command	Description
	Reset	Sets the CPU to RESET state.
	Run	Continues executing the program until the next active breakpoint is reached.
	Stop	Stops the program execution immediately.
	Step	Executes a single step into a function; Executes the current instruction line.
	Step Over	Executes a single step over a function.
	Step Out	Finishes executing the current function and stops afterward.
	Run to the Cursor Line	Executes the program until the current cursor line is reached.
	Show Next Statement	Shows the next executable statement/ instruction.

Revision History

Revision	Date	Notes
1.00	Dec. 20, 2023	Initial release.
1.01	Jul. 15, 2024	Added A34L716 device.
1.02	Nov. 15, 2024	Updated the disclaimer.
1.03	Nov. 22, 2024	Added A34M456 device.

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

HQ, Ochang
R&D, QA, and Test Center
37, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
www.abovsemi.com

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr

Japan Sales Manager
Tel: +81-50-6883-8679
Email: sales_jp@abov.co.kr