Application Note

Document No.: AN1178

Quick Start Guide to G32F031_G32M31xx_DAL_SDK

Version: V1.1

© Geehy Semiconductor Co., Ltd.

1. Introduction

G32F031 DAL SDK (Software Development Kit) is a comprehensive set of development drivers and documents designed specifically for the G32F031, G32M3122, G32M3114, G32M3115 series MCUs. It includes a low-level peripheral driver library, a collection of examples for different development boards, and a rich set of middleware. The SDK aims to significantly improve developer productivity by reducing development workload, time, and cost.

The materials referenced in this application note are:

AN1080_APM32 Series Toolchain User Guide

Eclipse Development Tutorial under AN1093_APM32 Arm MCU Windows System

Note: These materials can be obtained from the official website www.geehy.com.

Contents

1. Introduction 1

2. Content about SDK 3

3. Content about Boards 4

4. Content about Documents 5

5. Content about Examples 6

6. Content about Libraries 9

6.1. G32F031_DAL_Driver 9

6.2. CMSIS 11

6.3. Device 11

7. Content about Middlewares 12

7.1. CMSIS-FreeRTOS 12

8. Content about Package 13

8.1. SVD 13

8.2. Pack Package 13

9. Quick Start 15

9.1. Template Project 16

9.2. SDK-related Macro Definitions 17

9.3. Device Initialization 17

9.4. Compilation and Download 18

10. Revision History 24

11. Annex 25

11.1. Uninstalling and Installing Keil pack 25

2. Content about SDK

The complete "SDK" directory is as follows:

Figure 1 "SDK" Directory



3. Content about Boards

The "Boards" folder contains the BSP (Board Support Package) for the G32F031 development board. It can help to quickly drive the peripheral circuits or components on the board. The BSP can be found in the ～/Boards directory.

The provided BSP is built for the G32F031 mainboard. For other user-developed boards, some minor modifications may be required.

The directory structure of the "G32F031 DAL SDK Boards" folder is as follows:

- "Board_G32F031_EVB" folder
- "Components" folder

4. Content about Documents

The "Documents" folder contains a link file that can redirect to Geehy Semiconductor's Technical Support Center. The library support documentation can be found in the ~/Documents directory. This file can be used to view all supported function explanations, parameter functions, and return values.

The directory structure of the "G32F031 DAL SDK Documents" folder is as follows:

- DATASHEET.pdf
- G32F031_um.chm
- 快速上手指南.docx
- Quick Start Guide.docx

5. Content about Examples

Example applications can be found in the ～/Examples_DDL directory.

The provided examples are built for the G32F031 EVB development board. For use on other user-developed boards, appropriate adjustments are required.

The directory structure of the "G32F031 DAL SDK Example" folder is as follows:

- Config
- Include
- Project
  - Eclipse
  - IAR
  - MDK
- Source
- Clean Project.bat
- readme.txt

All example applications are tested using G32F031 DAL SDK V1.0.0, including the following examples:

Table 1 List of Example Programs Supported by the Development Board

IP / Module	Example	G32F031_EVB	NA	NA	NA
ADC	ADC_ContinuousConversion	√			
	ADC_DisContinuousConversion	√			
	ADC_DMA	√			
	ADC_MultiChannelScan	√			
	ADC_MultiChannelScanTimerTrigger	√			
	ADC_SequentialSectionSampling	√			
COMP	COMP_Output	√			
CRC	CRC_Calculation	√			
DIV	DIV_Unsigned	√			
DMA	DMA_FMCToRAM	√			

IP / Module	Example	G32F031_EVB	NA	NA	NA
EINT	EINT_Config	√			
FMC	FLASH_EEPROM	√			
	FLASH_Read_Write	√			
	FLASH_Write	√			
GPIO	GPIO_Toggle				
I2C	I2C_TwoBoardsPolling	√			
IAP	IAP_Application1	√			
	IAP_Application2	√			
	IAP_Bootloader	√			
IWDT	IWDT_Reset	√			
LPTMR	LPTMR_WakeUp	√			
NVIC	NVIC_WFI	√			
OPA	OPA_Standalone	√			
PMU	PMU_PVD	√			
	PMU_SLEEP	√			
	PMU_STOP	√			
RCC	RCC_ClockConfig	√			
RTOS	CMSIS_FreeRTOS				
SysTick	SysTick_TimeBase	√			
Template	Template	√			
TMR	TMR_BTMROCToggle	√			
	TMR_BTMRPWMOutput	√			
	TMR_DMA	√			
	TMR_InputCapture	√			
	TMR_OCActive	√			
	TMR_OCInactive	√			

IP / Module	Example	G32F031_EVB	NA	NA	NA
	TMR_OCToggle	√			
	TMR_PWMInput	√			
	TMR_PWMOutput	√			
	TMR_SinglePulse	√			
	TMR_Synchronize	√			
	TMR_TimeBase	√			
USART	UART_EchoInterrupt	√			
	UART_Printf	√			
	UART_TwoBoardsDMA	√			
	UART_TwoBoardsInterrupt	√			
	UART_TwoBoardsPolling	√			
USART	USART_EchoInterrupt	√			
	USART_Printf	√			
	USART_TwoBoardsDMA	√			
	USART_TwoBoardsInterrupt	√			
	USART_TwoBoardsPolling	√			
WWDT	WWDT_OverTime	√			

6. Content about Libraries

The "Libraries" folder contains a series of library files. It can provide support for G32F031 MCU, such as device support, startup files, linker files, and DDL peripheral support. The library files can be found in the ～/Libraries directory.

The "G32F031 DAL SDK Libraries" folder includes the following materials:

- G32F031_DAL_Driver
- CMSIS
- Device

6.1. G32F031_DAL_Driver

This folder contains all DDL peripheral library driver files.

Table 2 List of DDL Drivers Supported by the Device

IP / Module	G32F031xx	NA	NA	NA
g32_assert.h	√			
g32f031_ddl_adc	√			
g32f031_ddl_atmr	√			
g32f031_ddl_btmr	√			
g32f031_ddl_comp0	√			
g32f031_ddl_comp1	√			
g32f031_ddl_cortex	√			
g32f031_ddl_crc	√			
g32f031_ddl_div	√			
g32f031_ddl_dma	√			
g32f031_ddl_eint	√			
g32f031_ddl_flash	√			
g32f031_ddl_gpio	√			
g32f031_ddl_gtmr	√			
g32f031_ddl_i2c	√			
g32f031_ddl_iwdt	√			

IP / Module	G32F031xx	NA	NA	NA
g32f031_ddl_lptmr	√			
g32f031_ddl_opa	√			
g32f031_ddl_pmu	√			
g32f031_ddl_rcc	√			
g32f031_ddl_scu	√			
g32f031_ddl_spi	√			
g32f031_ddl_uart	√			
g32f031_ddl_usart	√			
g32f031_ddl_utils	√			
g32f031_ddl_wwdt	√			

6.1.1. DDL

The device driver library (DDL) provides peripheral development interfaces in the form of inline functions and regular functions.

Inline functions are defined in the DDL header files and are responsible for providing bit-field level operations, while regular functions offer more complex Init/DeInit operations.

The common naming characteristics of inline functions are shown in the table below:

Table 3 Naming Characteristics of DDL Inline Functions

Scenario	Function	Naming Characteristic
General configuration	Set	DDL_PPP_SetXXX
	Get	DDL_PPP_GetXXX
	Enable	DDL_PPP_EnableXXX
	Disable	DDL_PPP_DisableXXX
Flag	Get Flag status	DDL_PPP_IsActiveFlag_XXX
	Clear Flag status	DDL_PPP_ClearFlag_XXX
Interrupt	Enable interrupt	DDL_PPP_EnableIT_XXX
	Disable interrupt	DDL_PPP_DisableIT_XXX
	Check interrupt	DDL_PPP_IsEnabledIT_XXX
DMA	Enable DMA	DDL_PPP_EnableDMAReq_XXX
	Disable DMA	DDL_PPP_DisableDMAReq_XXX

	Check DMA request	DDL_PPP_IsEnabledDMAReq_XXX

The common naming characteristics of regular functions are shown in the table below:

Table 4 Naming Characteristics of DDL Regular Functions

Function	Naming Characteristic
Initialization	DDL_PPP_Init
Deinitialization	DDL_PPP_DeInit
Initialize the initialization structure	DDL_PPP_StructInit

6.2. CMSIS

This folder contains header files related to the Arm® Cortex®-M0+ core, which define the registers and functions of the core.

6.3. Device

This folder contains device-specific files related to the G32F031. Wherein:

- The "Geehy\G32F031\Include" folder contains all G32F031 device header files, which define the device registers and interrupt vectors.
- The "Geehy\G32F031\Source" folder contains startup files and linker files for the MDK, IAR, and GCC platforms, including the reset vector table and startup code.

7. Content about Middlewares

The "Middlewares" folder contains a series of third-party middleware. Such middleware can be found in the ~/Middlewares directory.

The middleware used in the "G32F031 DAL SDK Middlewares" folder includes the following materials:

- CMSIS-FreeRTOS

7.1. CMSIS-FreeRTOS

The "CMSIS-FreeRTOS folder" contains the FreeRTOS operating system compliant with the CMSIS-RTOS2 standard.

8. Content about Package

The "Package" folder includes the Geehy G32F031 DFP Package. This package can be found in the ~/Package directory.

The "G32F031 DAL SDK Libraries" folder includes the following materials:

- Geehy.G32F031_DFP.x.y.z.pack

8.1. SVD

This folder contains SVD files, which allow users to simulate and view register usage.

8.2. Pack Package

This folder stores the latest version of the Keil pack package, for use in the MDK-ARM environment to support the G32F031 series chips. The latest version of the Keil pack package can also be obtained from the following link:

https://www.keil.arm.com/packs/

https://www.geehy.com/design/software

9. Content about Utilities

The "Utilities" folder includes some utilities. These utilities can be found in the ~/Utilities directory.

The "G32F031 DAL SDK Utilities" folder includes the following materials:

- FLM
- IAR_AddOn
- pyOCD
- SVD

9.1. FLM

This folder stores the FLM files.

9.2. IAR_AddOn

This folder stores the IAR AddOn exe files

9.3. pyOCD

This folder stores the python script files for pyOCD.

9.4. SVD

This folder contains SVD files, which allow users to simulate and view register usage.

10. Quick Start

To quickly EVBuate the G32F031 SDK, you may need to prepare the following environment or content:

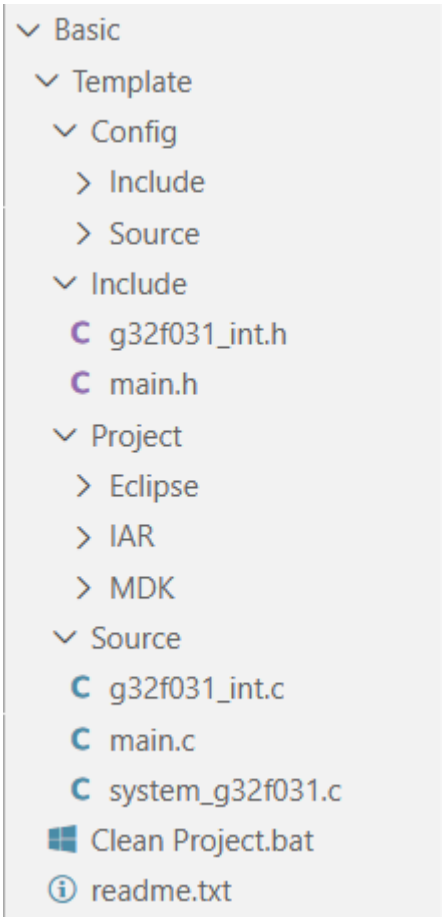
- Window 10/11
- MDK-ARM v5.40 or updated version
- IAR EW for ARM 8.50.5 or updated version
- Eclipse 4.24 or updated version
- arm-gnu-toolchain 10.3.1 or updated version
- Emulate debugger (e.g. Geehy-Link or J-Link)
- Any of the following development boards (depending on the chip used in the project):
  - G32F031 EVB Development Board

Note: The content of this section takes the G32F031 EVB Development Board as an example.

10.1. Template Project

The template project is stored in ~/Examples_DDL/Board_G32F031_EVB/Template, and its contents are shown in Figure 2. It includes template projects for MDK, IAR, and Eclipse. The "Config" folder contains the initialization code for the devices used, while the "Include" and "Source" folders store the header files and source files used by the application code in the template project, respectively.

Figure 2 Contents of the Template Folder

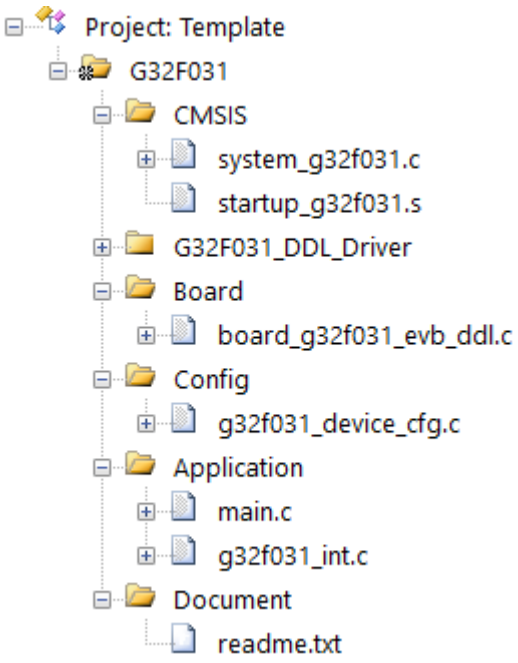


Open the folder of the corresponding project and click the project file to open the IDE project. Shown in Figure 3 is the MDK template project. The files in the project are described as follows:

1. system_g32f031.c – System initialization configuration file.
2. startup_g32f031xx.s – Startup file.
3. g32f031_ddl_ppp.c in the "G32F031_DDL_Driver" folder: Low-level driver files for various peripherals.
4. board_g32f031_EVB_ddl.c – Onboard support file.
5. g32f031_ppp_cfg.c in the "Config" folder – Initialization configuration files for various peripherals, providing initialization entry functions for different peripherals.

6. g32f031_device_cfg.c – Global peripheral initialization file, providing the main entry that includes initialization entry functions of other peripherals.
7. g32f031_int.c – Interrupt file, containing kernel and peripheral interrupt service functions.
8. main.c – Main source code file of the template project.
9. readme.txt – Project description file.

Figure 3 MDK Template Project



10.2. SDK-related Macro Definitions

When creating a new project, in addition to importing the startup file (startup_g32f031xx.s) into the project, it is also necessary to enable the relevant macro definitions according to the MCU model. The relationships of the macro definitions used in the SDK are shown in the table below.

Table 5 Macro Definition Relationships

Macro definition	Description
USE_FULL_DDL_DRIVER	Enable full support for low-level library drivers
G32F031xx	Enable support for the G32F031x8 model
BOARD_G32F031_EVB	Enable BSP support for the G32F031 EVB development board
USE_FULL_ASSERT	Enable support for assertion functions

10.3. Device Initialization

When initializing peripherals, it is recommended to place the device initialization programs in

the “Config” directory of the project, and to configure a separate initialization file for each peripheral.

For example, when initializing USART, you can add g32f031_usart_cfg.h to the "Config/Include" directory and g32f031_usart_cfg.c to the "Config/Source" directory.

In g32f031_usart_cfg.h, declare the initialization routines for all USARTs to be initialized, and define them in g32f031_usart_cfg.c.

The USART initialization routines should include enabling the USART clock, initialization configuration of the GPIOs used by the USART, initialization configuration of the USART itself, and interrupt configuration of the USART.

10.4. Compilation and Download

For installation and usage of IDEs and simulation/download toolchains, refer to the following documents:

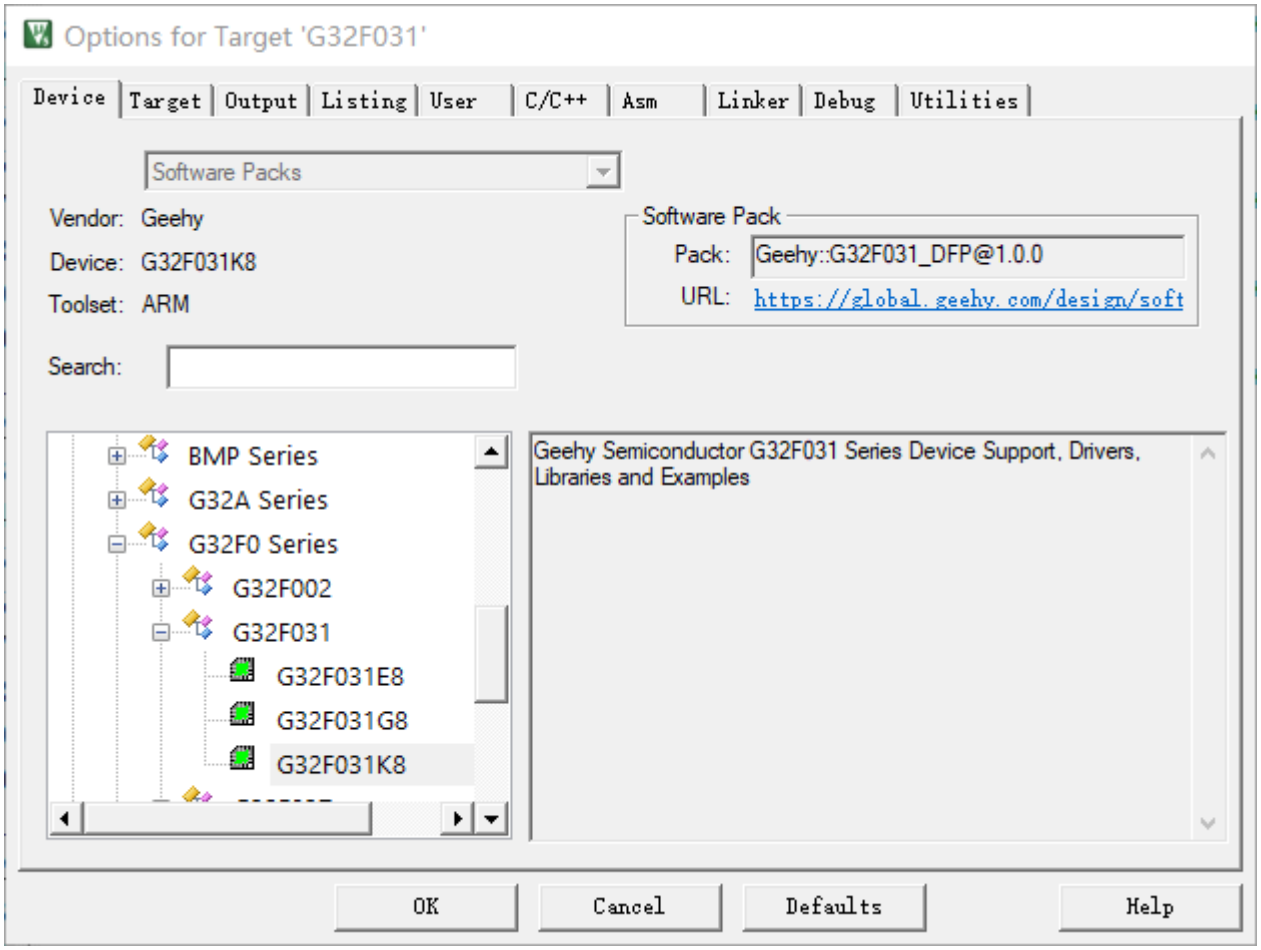
AN1080_G32 Series Toolchain User Guide

Eclipse Development Tutorial under AN1093_G32 Arm MCU Windows System

10.4.1. Configuration Project

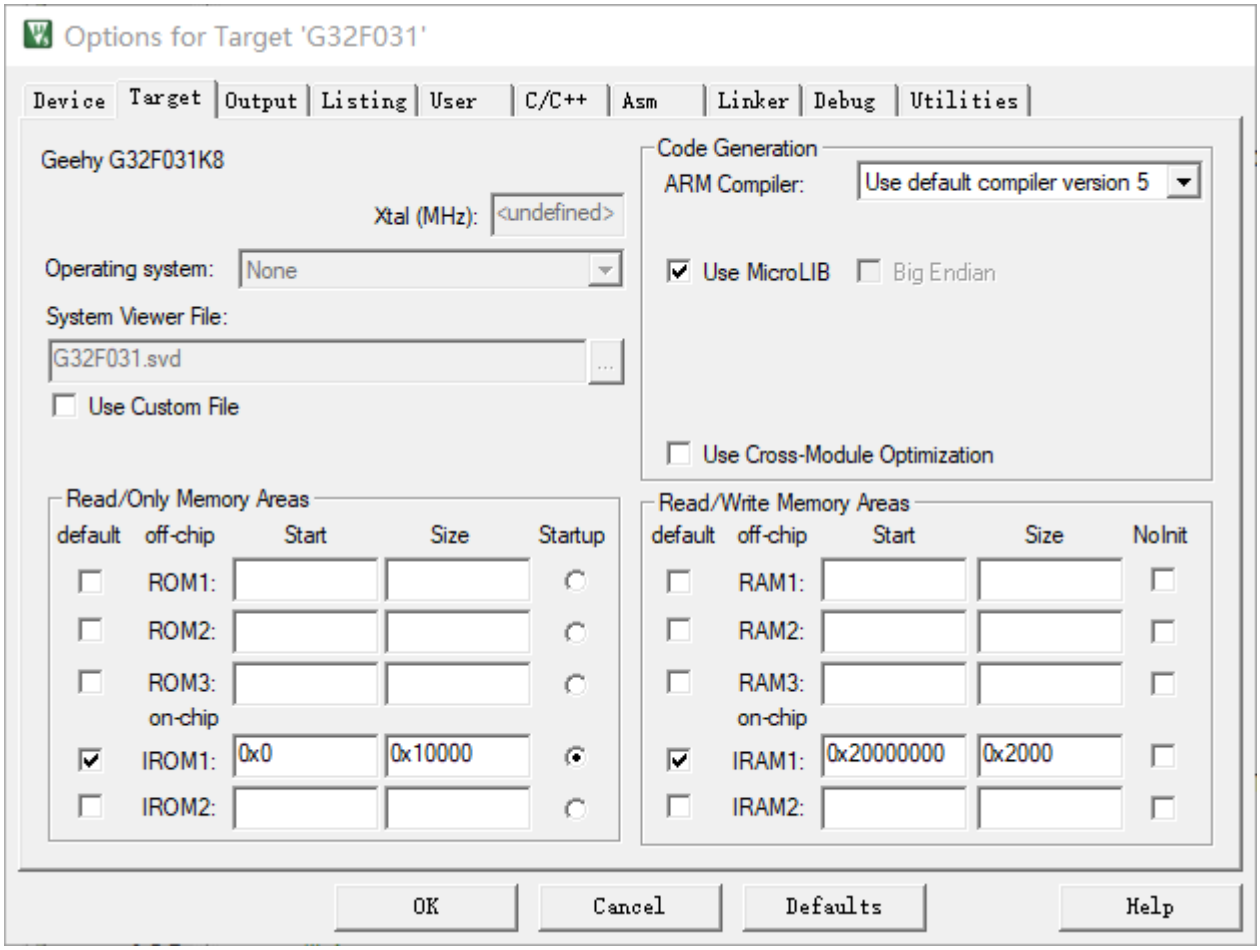
After installing the Keil pack toolchain as described in the above documents, open the Template MDK project and click the “Magic Wand” to confirm the chip selection. As shown in Figure 4, select G32F031K8 for development.

Figure 4 Viewing Chip Selection



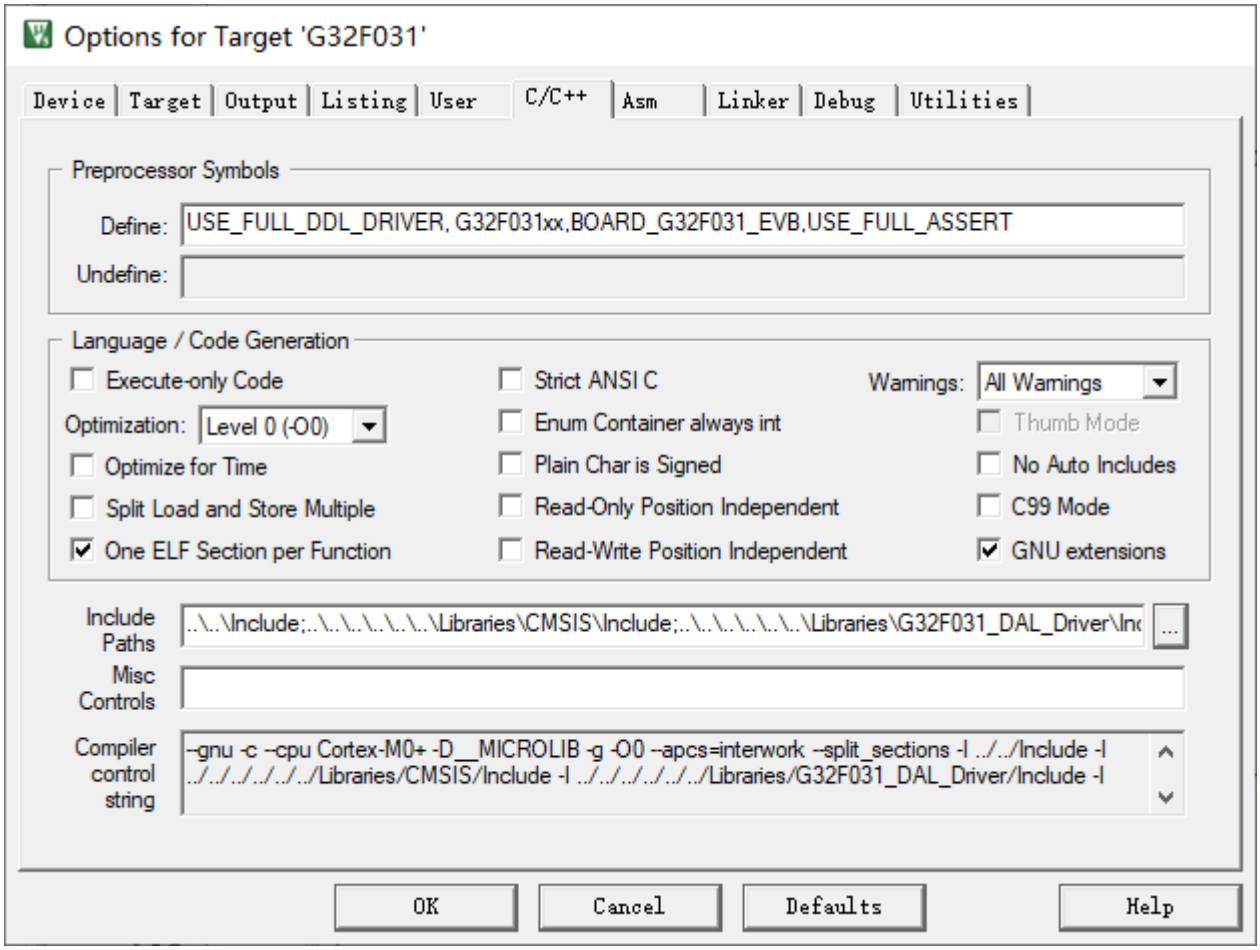
By switching to the "Target" tab, you can view the address and size of the ROM and RAM configured for the K8.

Figure 5 ROM and RAM



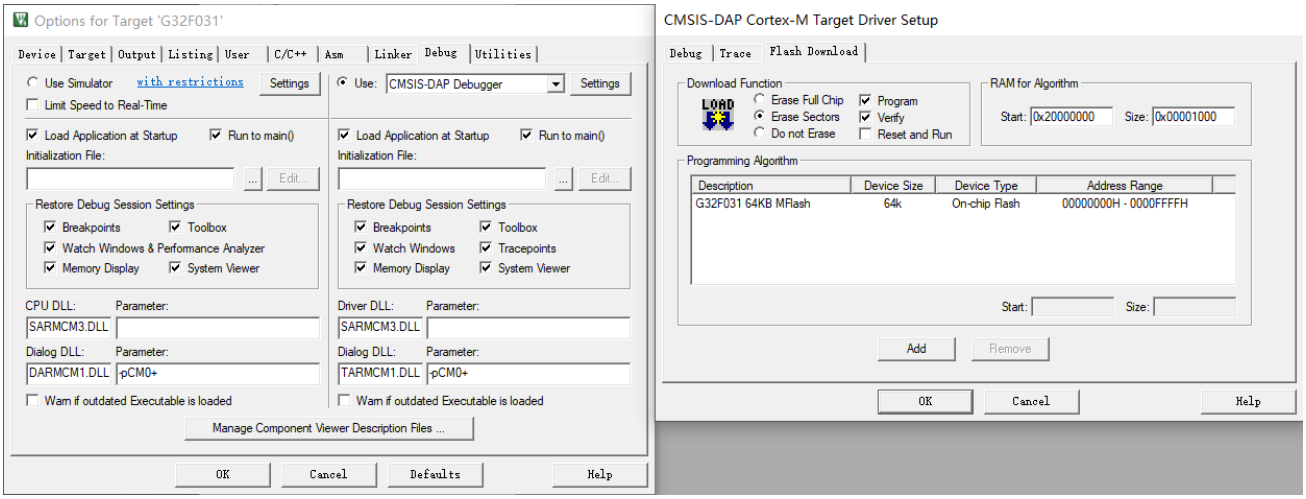
Switch to the C/C++ tab and confirm that the G32F031xx macro definition is enabled to support the development of G32F031K8.

Figure 6 C/C++ Configuration



Switch to the "Debug" tab, select the *CMSIS-DAP* Debugger emulator, and click the "Settings" button to enter the "Flash Download" tab. Confirm that the G32F031 64KB Flash download algorithm is selected. Note that the algorithm usage range is 0x00000000H ~ 0x0000FFFFH, as shown in Figure 7.

Figure 7 Debug



10.4.2. Compilation and Download

Click the "Build" button on the MDK interface for compilation. After confirming that the compilation results are correct, click the "Download" button to perform the download. The download result is as shown in Figure 10.

Figure 8 Compilation Project

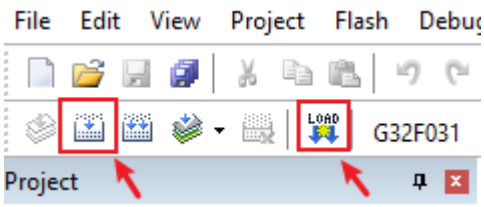


Figure 9 Compilation Results

```
Build Output
compiling g32f031_int.c...
linking...
Program Size: Code=580 RO-data=236 RW-data=4 ZI-data=1028
".\Objects\G32F031\Template.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:05
```

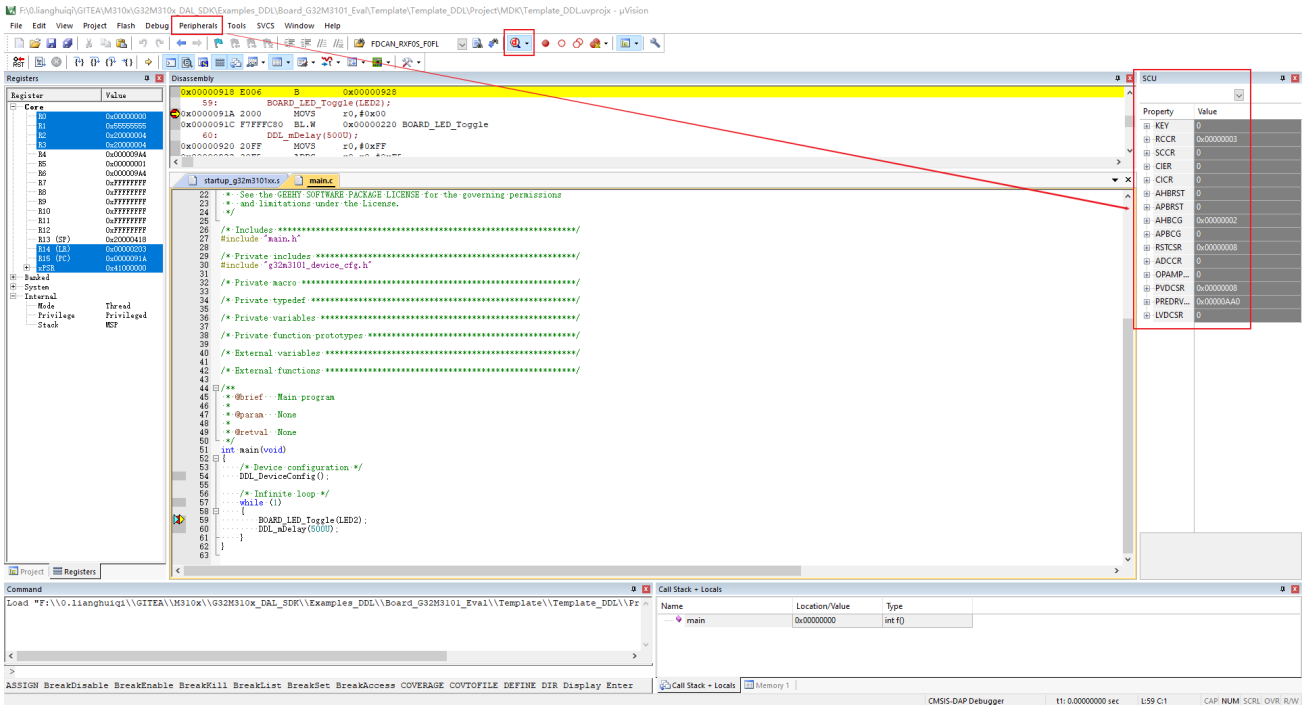
Figure 10 Download Results

```
Build Output
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 16:19:14
```

10.4.3. Simulation Project

Click the "Start/Stop Debug Session" button on the interface to start simulation. On the simulation interface, you can use the "Peripherals", "Memory", and "Registers" tabs to view information such as the MCU core, peripheral registers, and memory data.

Figure 11 Simulation Project



11. Revision History

Table 6 Revision History

Date	Version	Revision History
1,2026	V1.0	New
7,2026	V1.0	Rename the SDK and add supported chip series

12. Annex

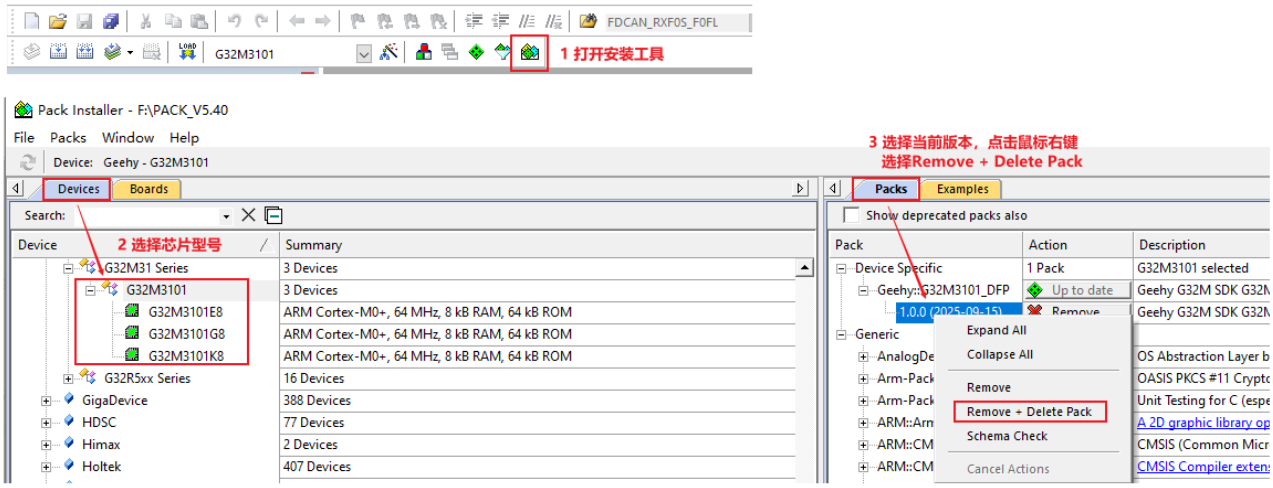
12.1. Uninstalling and Installing Keil pack

The following shows how to uninstall all versions of the pack and install the latest version of the pack.

Steps:

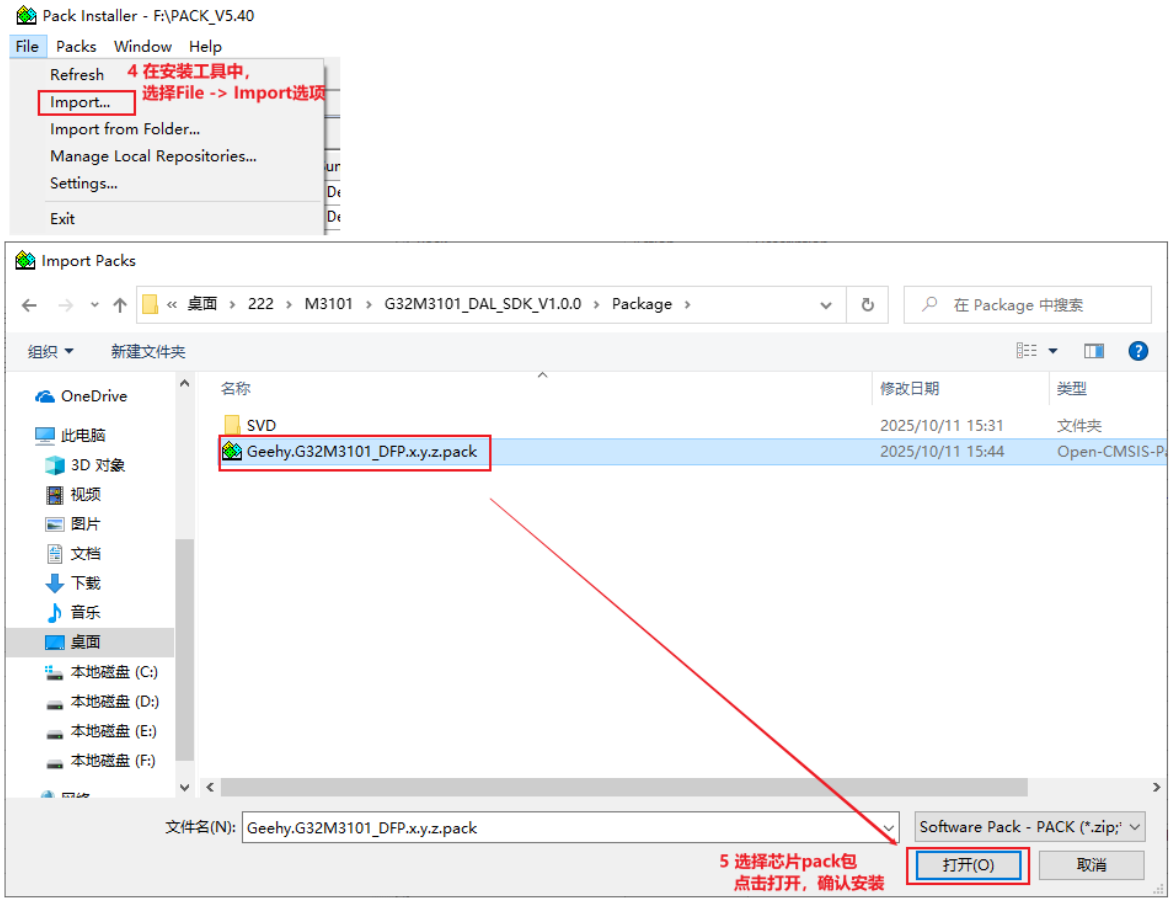
1. Open the pack installer.
2. In the Devices tab, select the corresponding chip model.
3. In the Packs tab, right-click and perform the delete operation.
4. In the pack installer, click File → Import to import and install the pack.
5. Select the pack and click Open to start the installation.
6. Check the installation directory of the pack to see if the pack contents have been updated.

Figure 12



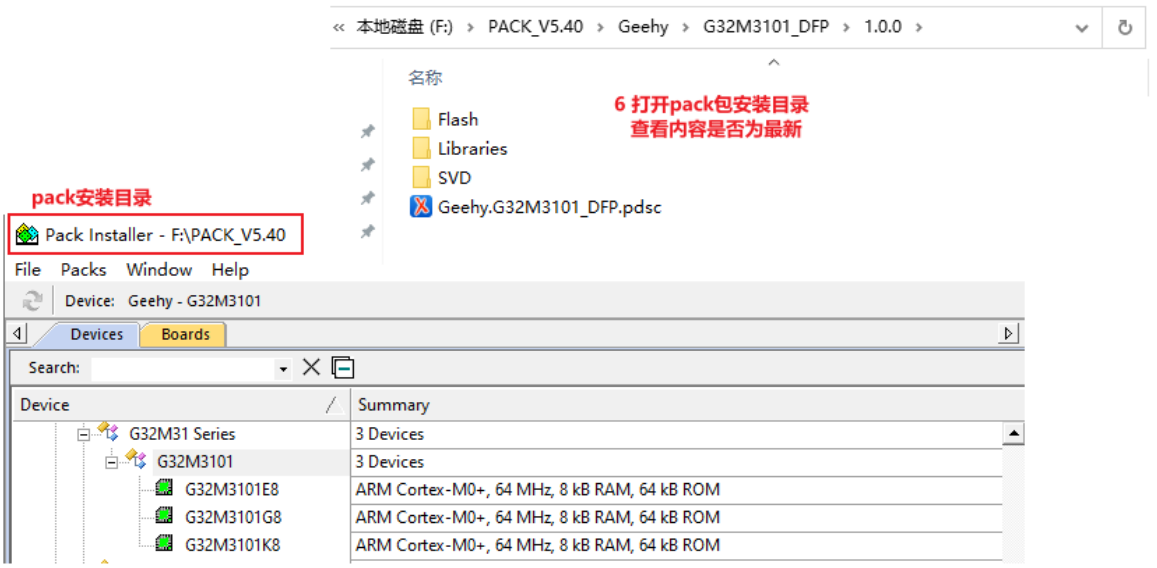
1. Open the pack installer
2. Select the chip model
3. Select the current version, right-click the mouse, and select “Remove + Delete Pack”

Figure 13



4. In the pack installer, select File → Import.
5. Select the pack and click Open for installation.

Figure 14



6. Check the pack to see if it is the newest version.

Statement

This manual is formulated and published by Zhuhai Geehy Semiconductor Co., Ltd. (hereinafter referred to as "Geehy"). The contents in this manual are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to correct and modify this manual at any time. Please read this manual carefully before using the product. Once you use the product, it means that you (hereinafter referred to as the "users") have known and accepted all the contents of this manual. Users shall use the product in accordance with relevant laws and regulations and the requirements of this manual.

1. Ownership of rights

This manual can only be used in combination with chip products and software products of corresponding models provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this manual for any reason or in any form.

The "Geehy" or "Geehy" words or graphics with "®" or "TM" in this manual are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

2. No intellectual property license

Geehy owns all rights, ownership and intellectual property rights involved in this manual.

Geehy shall not be deemed to grant the license or right of any intellectual property to users

explicitly or implicitly due to the sale and distribution of Geehy products and this manual.

If any third party's products, services or intellectual property are involved in this manual, Geehy shall not be deemed to authorize users to use the aforesaid third party's products, services or intellectual property, nor shall it be deemed to provide any form of guarantee for third-party products, services, or intellectual property, including but not limited to any non-infringement guarantee for third-party intellectual property, unless otherwise agreed in sales order or sales contract of Geehy.

3. Version update

Users can obtain the latest manual of the corresponding products when ordering Geehy products.

If the contents in this manual are inconsistent with Geehy products, the agreement in Geehy sales order or sales contract shall prevail.

4. Information reliability

The relevant data in this manual are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment are unavoidable. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this manual. The relevant data in this manual are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If losses are caused to users due to the user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

5. Compliance requirements

Users shall abide by all applicable local laws and regulations when using this manual and the matching Geehy products. Users shall understand that the products may be restricted by the export, re-export or other laws of the countries of the product suppliers, Geehy, Geehy distributors and users. Users (on behalf of itself, subsidiaries and affiliated enterprises) shall agree and undertake to abide by all applicable laws and regulations on the export and re-export of Geehy products and/or technologies and direct products.

6. Disclaimer

This manual is provided by Geehy on an "as is" basis. To the extent permitted by applicable laws, Geehy does not provide any form of express or implied warranty, including without limitation the warranty of product merchantability and applicability of specific purposes.

Geehy products are not designed, authorized, or guaranteed to be suitable for use as critical components in military, life support, pollution control, or hazardous substance management systems, nor are they designed, authorized, or guaranteed to be suitable for applications that may cause injury, death, property, or environmental damage in case of product failure or malfunction.

If the product is not labeled as "Automotive grade", it means it is not suitable for automotive applications. If the user's application of the product is beyond the specifications, application fields, and standards provided by Geehy, Geehy will assume no responsibility.

Users shall ensure that their application of the product complies with relevant standards, and the requirements of functional safety, information security, and environmental standards. Users are fully responsible for their selection and use of Geehy products. Geehy will bear no responsibility for any disputes arising from the subsequent design and use of Geehy products by users.

7. Limitation of liability

In any case, unless required by applicable laws or agreed in writing, Geehy and/or any third party providing this manual and the products on an "as is" basis shall not be liable for

damages, including any general or special direct, indirect or collateral damages arising from the use or no use of this manual and the products (including without limitation data loss or inaccuracy, or losses suffered by users or third parties), which cover damage to personal safety, property, or environment, for which Geehy will not be responsible.

8. Scope of application

The information in this manual replaces the information provided in all previous versions of the manual.

©2025 Zhuhai Geehy Semiconductor Co., Ltd. All Rights Reserved