应用笔记

Application Note

文档编号：**AN1178**

G32F031_G32M31xx_DAL_SDK 快速上手指南

版本：**V1.1**

© 珠海极海半导体有限公司

1. 引言

G32F031 DAL SDK（Software Development Kit）是专为 G32F031、G32M3122、G32M3114、G32M3115 系列 MCU 设计的一套全面的开发驱动和文档，其包含底层外设驱动库、针对不同开发板提供的一系列例程以及丰富的中间件，旨在通过减少开发工作量、时间和成本来显著提高开发人员的生产力。

本应用笔记中涉及的资料：

《AN1080_APM32 系列工具链使用指南》

《AN1093_APM32 Arm MCU Windows 系统下 Eclipse 开发教程》

注：资料可在官网 www.geehy.com 获取。

目录

1. 引言 1

2. 关于 SDK 内容 4

3. 关于 Boards 内容 5

4. 关于 Documents 内容 6

5. 关于 Examples 内容 7

6. 关于 Libraries 内容 10

6.1. G32F031_DAL_Driver 10

6.2. CMSIS 12

6.3. Device 12

7. 关于 Middlewares 内容 13

7.1. CMSIS-FreeRTOS 13

8. 关于 Package 内容 14

8.1. Pack 包 14

9. 关于 Utilities 内容 15

9.1. FLM 15

9.2. IAR_AddOn 15

9.3. pyOCD 15

9.4. SVD 15

10. 快速上手 16

10.1. 模板工程 17

10.2. SDK 相关宏定义 18

10.3. 设备初始化 18

10.4. 编译和下载 19

11. 版本历史 24

12. 附录 25

12.1. Keil pack 包的卸载与按照 25

12.2. Keil 关联 SVD 外设调试信息..... 27

2. 关于 SDK 内容

完整的 SDK 目录如下：

图 1 SDK 目录



3. 关于 Boards 内容

Boards 文件夹包含 G32F031 开发板的 BSP 支持包。它可以帮助快速驱动板卡上的外围电路或元件。BSP 可以在~/Boards 目录中找到。

提供的 BSP 是为 G32F031 主板构建的。对于其他用户开发的板卡使用，可能需要进行一些小的修改。

G32F031 DAL SDK Boards 文件夹的目录结构如下：

- Board_G32F031_EVB 文件夹
- Components 文件夹

4. 关于 Documents 内容

Documents 文件夹包含一个链接文件，可以重定向到 Geehy 半导体的技术支持中心。库支持文档可以在～/Documents 目录中找到，该文件可以查看所有支持的函数解释，参数作用和返回值。

G32F031 DAL SDK Documents 文件夹的目录结构如下：

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

5. 关于 Examples 内容

示例应用可以在~/Examples_DDL 目录中找到。

所提供的示例是为 G32F031 EVB 开发板构建的。对于其他用户开发板的使用，需要进行适当的调整。

G32F031 DAL SDK Example 文件夹中例程项目的目录结构如下：

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

所有示例应用程序使用 G32F031 DAL SDK V1.0.0 进行测试，包括以下示例：

表格 1 开发板支持的示例程序列表

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	√			

IP / Module	Example	G32F031_EVB	NA	NA	NA
DMA	DMA_FMCToRAM	√			
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	√			
SPI	SPI_TwoBoards	√			
SysTick	SysTick_TimeBase	√			
Template	Template_DDL	√			
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	√			
UART	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. 关于 Libraries 内容

Libraries 文件夹包含一系列库文件。它可以为 G32F031 MCU 提供支持，如设备支持、启动文件、链接文件和 DDL 外设支持等。库文件可以在~/Libraries 目录中找到。

G32F031 DAL SDK Libraries 文件夹包括以下资料：

- G32F031_DAL_Driver
- CMSIS
- Device

6.1. G32F031_DAL_Driver

该文件夹包含所有 DDL 外设库驱动文件。

表格 2 设备支持的 DDL 驱动程序列表

IP / Module	G32F031xx	NA	NA	NA
apm32_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	√			
g32f031_ddl_lptmr	√			
g32f031_ddl_opa	√			
g32f031_ddl_pmu	√			
g32f031_ddl_rcc	√			
g32f031_ddl_scu	√			

IP / Module	G32F031xx	NA	NA	NA
g32f031_ddl_spi	√			
g32f031_ddl_uart	√			
g32f031_ddl_usart	√			
g32f031_ddl_utils	√			
g32f031_ddl_wwdt	√			

6.1.1. DDL 驱动库

DDL 驱动库是由内联函数加普通函数的形式提供外设开发接口。

内联函数定义在 DDL 驱动库头文件中，负责提供位域级功能操作，普通函数提供更为复杂的 Init/DeInit 操作。

内联函数常见的命名特征如下表所示：

表格 3 DDL 内联函数命名特征

场景	功能	命名特征
通用配置	设置	DDL_PPP_SetXXX
	获取	DDL_PPP_GetXXX
	使能	DDL_PPP_EnableXXX
	失能	DDL_PPP_DIsableXXX
Flag	获取 Flag 状态	DDL_PPP_IsActiveFlag_XXX
	清除 Flag 状态	DDL_PPP_ClearFlag_XXX
中断	使能中断	DDL_PPP_EnableIT_XXX
	失能中断	DDL_PPP_DisableIT_XXX
	检查中断使能	DDL_PPP_IsEnabledIT_XXX
DMA	使能 DMA 请求	DDL_PPP_EnableDMAReq_XXX
	失能 DMA 请求	DDL_PPP_DisableDMAReq_XXX
	检查 DMA 请求使能	DDL_PPP_IsEnabledDMAReq_XXX

普通函数常见的命名特征如下表所示：

表格 4 DDL 普通函数命名特征

功能	命名特征
初始化	DDL_PPP_Init
去初始化	DDL_PPP_DeInit
初始化初始化结构体	DDL_PPP_StructInit

6.2. CMSIS

该文件夹包含与 Arm® Cortex®-M0+ 内核相关的头文件，这些文件定义了内核的寄存器和功能。

6.3. Device

该文件夹包含与 G32F031 设备特定相关的文件。其中：

- Geehy\G32F031\Include 文件夹包含所有 G32F031 设备头文件，该文件定义了设备的寄存器和中断向量。
- Geehy\G32F031\Source 文件夹包含适用于 MDK、IAR 和 GCC 平台的启动文件、链接文件，包含了复位向量表和启动代码。

7. 关于 Middlewares 内容

Middlewares 文件夹包含一系列第三方中间件。这些中间件可以在~/Middlewares 目录中找到。

G32F031 DAL SDK Middlewares 文件夹使用的中间件包括如下资料：

- CMSIS-FreeRTOS

7.1. CMSIS-FreeRTOS

CMSIS-FreeRTOS 文件夹包含 CMSIS RTOS2 标准的 FreeRTOS 操作系统。

8. 关于 Package 内容

Package 文件夹包括 Geehy G32F031 DFP Package。该 Package 可以在~/Package 目录中找到。

G32F031 DAL SDK Package 文件夹包括以下资料：

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

8.1. Pack 包

该文件夹存放最新版本的 Keil pack 包，供用户在 MDK-ARM 环境下使用，以支持 G32F031 系列芯片。最新版本的 Keil pack 包也可以从以下链接获取：

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

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

9. 关于 Utilities 内容

Utilities 文件夹包含了一些工具。相关工具可以在~/Utilities 目录中找到。

G32F031 DAL SDK Utilities 文件夹包括以下资料：

- FLM
- IAR_AddOn
- pyOCD
- SVD

9.1. FLM

该文件夹存放 FLM 下载算法文件。

9.2. IAR_AddOn

该文件夹存放 IAR 的补丁包文件。

9.3. pyOCD

该文件夹存放 pyOCD 的 python 脚本文件。

9.4. SVD

该文件夹存放 SVD 文件。

10. 快速上手

快速评估 G32F031 SDK 您可能需要准备以下环境或内容：

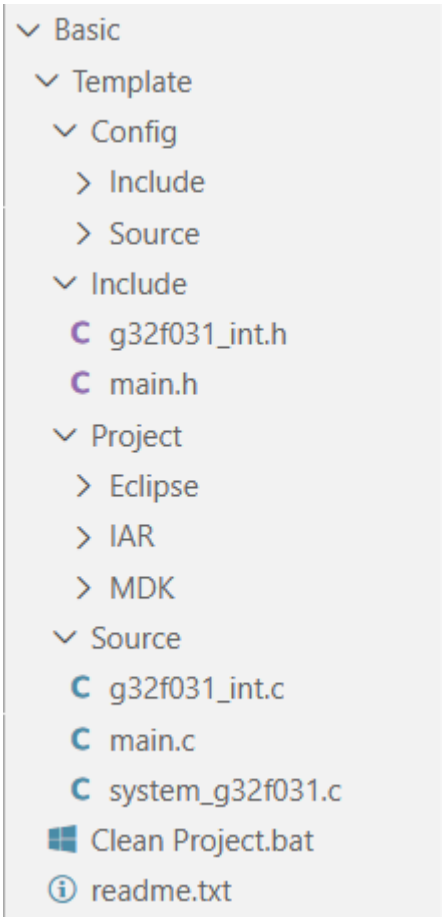
- Window 10/11
- MDK-ARM v5.40 或更高版本
- IAR EW for ARM 8.50.5 或更高版本
- Eclipse 4.24 或更高版本
- arm-gnu-toolchain 10.3.1 或更高版本
- 仿真调试器（如 Geehy-Link 或 J-Link）
- 以下任意开发板（视所开发项目的芯片而定）：
  - G32F031 EVB 开发板

注：本章节内容以 G32F031 EVB 开发板为例。

10.1. 模板工程

模板工程存放在~/Examples_DDL/Board_G32F031_EVB/Basic 中，其内容如图 2 所示。包含 MDK、IAR 和 Eclipse 的模版工程。Config 文件夹存放了所用到的设备的初始化代码，Include 和 Source 文件夹分别存放了模版工程中应用代码所用到的头文件和源文件。

图 2 Template 文件内容

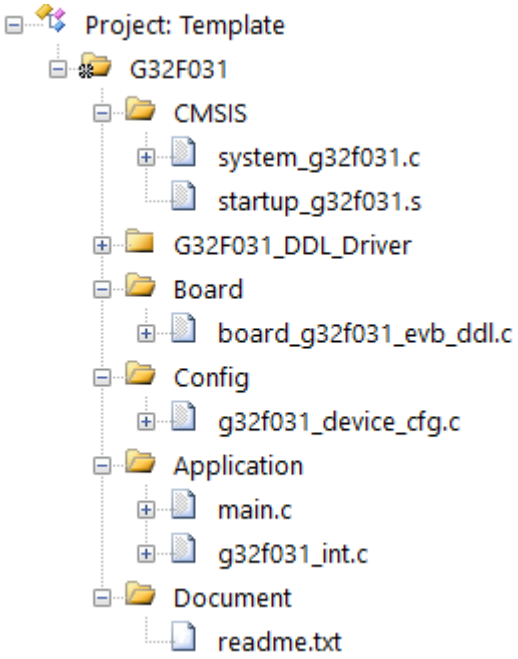


打开对应工程的文件夹并点击工程文件即可打开对应的 IDE 工程。如图 3 所示是 MDK 模板工程。工程中的文件内容描述如下：

1. system_g32f031.c 系统初始化配置文件。
2. startup_g32f031xx.s 启动文件。
3. G32F031_DDL_Driver 文件夹的 g32f031_ddl_ppp.c 文件是各个外设的底层驱动文件。
4. board_g32f031_evb_ddl.c 板载支持文件。
5. Config 文件夹中的 g32f031_ppp_cfg.c 文件是各个外设的初始化配置文件，提供不同外设的初始化程序入口。
6. g32f031_device_cfg.c 总外设初始化文件，提供包含了其他外设初始化程序入口的总入口。

7. g32f031_int.c 中断文件，包含内核和外设中断服务函数。
8. main.c 模板工程的主代码文件。
9. readme.txt 工程的说明文件。

图 3 MDK 模板工程



10.2. SDK 相关宏定义

在创建新工程时，除了需要导入启动文件（startup_g32f031xx.s）到工程中，还需要根据 MCU 型号，开启相关的宏定义。SDK 中使用到的宏定义关系如下表所示。

表格 5 宏定义关系表

宏定义	描述
USE_FULL_DDL_DRIVER	开启全部的底层库驱动支持
G32F031xx	开启 G32F031x8 型号支持
BOARD_G32F031_EVAL	开启 g32f031 eval 开发板的 BSP 支持
USE_FULL_ASSERT	开启断言函数支持

10.3. 设备初始化

在初始化外设时，推荐将设备初始化程序放在工程的 Config 目录下，并为每种外设单独配置初始化程序文件。

例如，在初始化 USART 时，可以分别向 Config/Include 和 Config/Source 目录中添加 g32f031_usart_cfg.h 以及 g32f031_usart_cfg.c。

在 g32f031_usart_cfg.h 中声明所有待初始化的 USART 的初始化程序，并在 g32f031_usart_cfg.c 中进行定义。

USART 的初始化程序中应该包含 USART 时钟使能、USART 所用 GPIO 的初始化配置，USART 的初始化配置，USART 的中断配置。

10.4. 编译和下载

IDE、仿真下载工具链的安装和使用可参考以下文档：

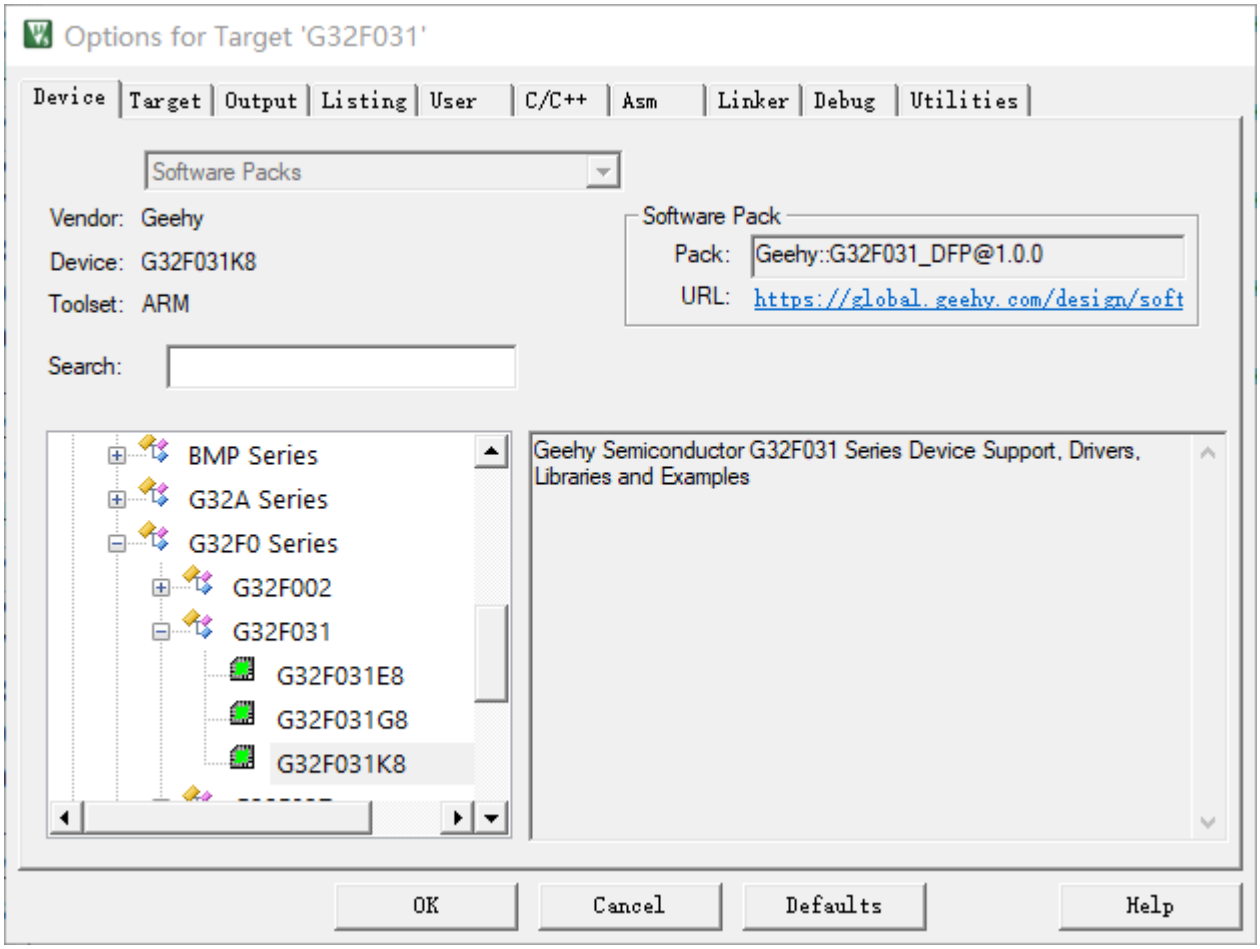
《AN1080_APM32 系列工具链使用指南》

《AN1093_APM32 Arm MCU Windows 系统下 Eclipse 开发教程》

10.4.1. 配置工程

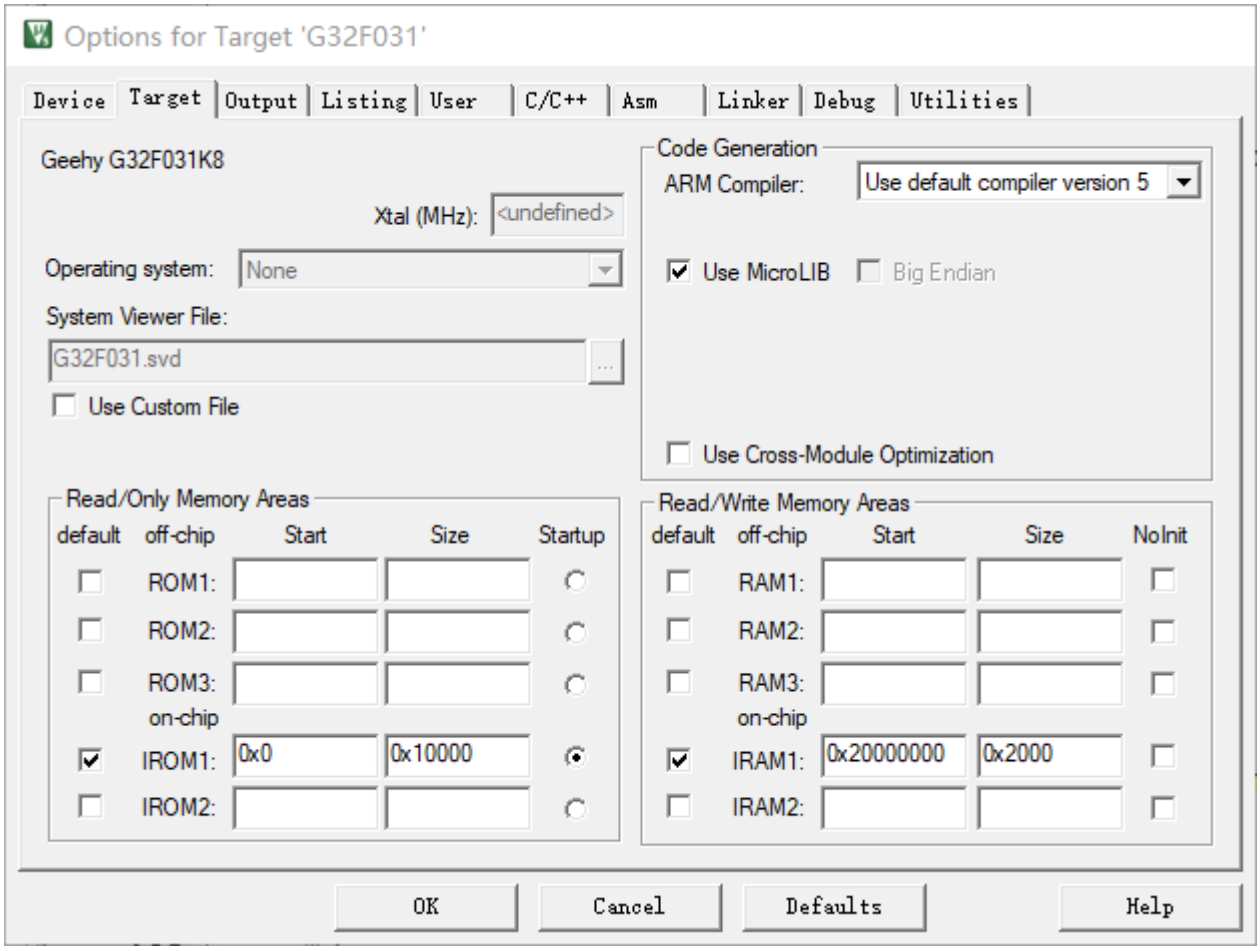
参考上述文档安装 Keil pack 包工具链后，打开 Template MDK 工程，并点击魔术棒确认芯片选型，如图 4 所示选中 G32F031K8 进行开发。

图 4 查看芯片选型



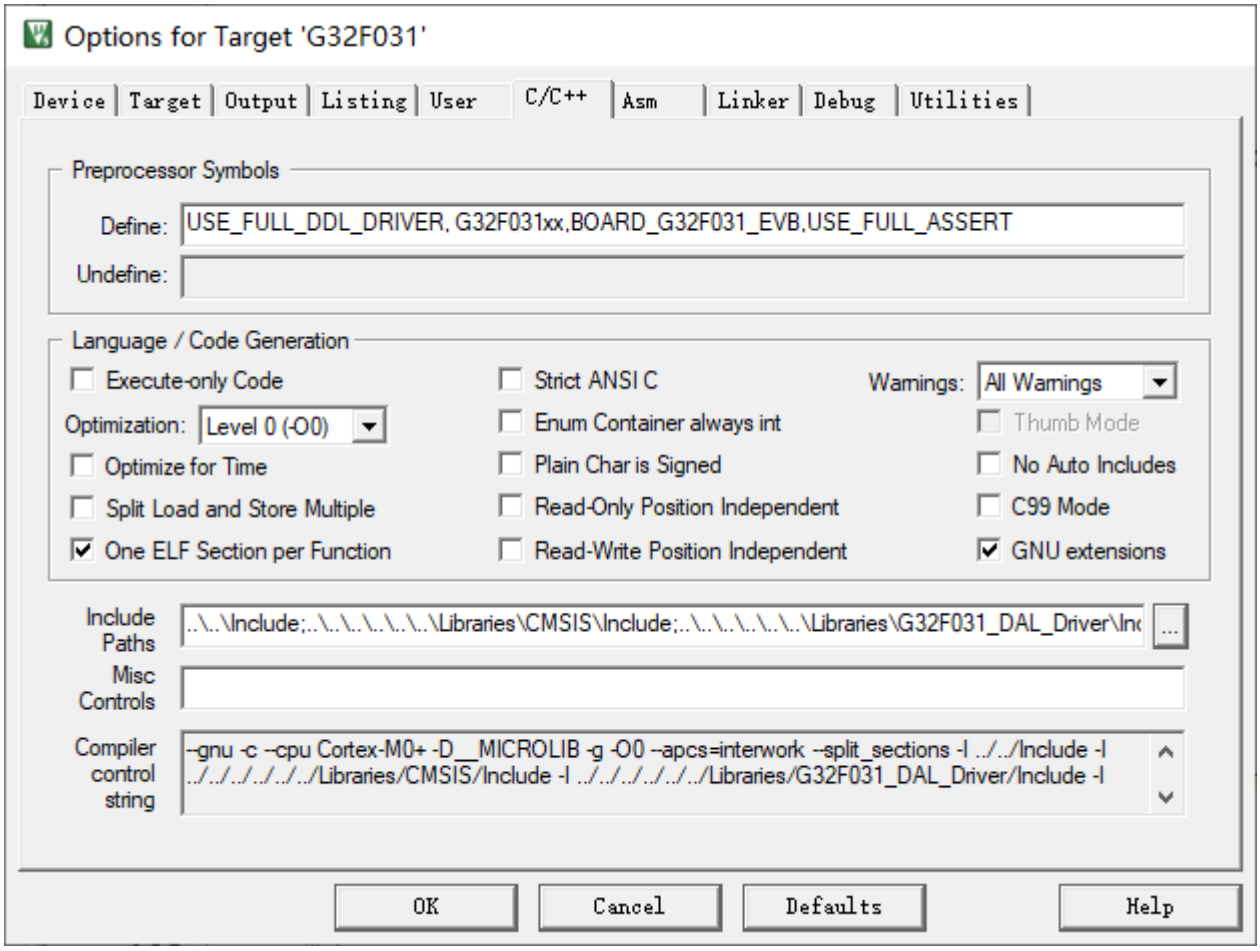
切换 Target 选项卡，可以看到 G32F031K8 配置使用的 ROM、RAM 的地址和大小。

图 5 ROM 和 RAM



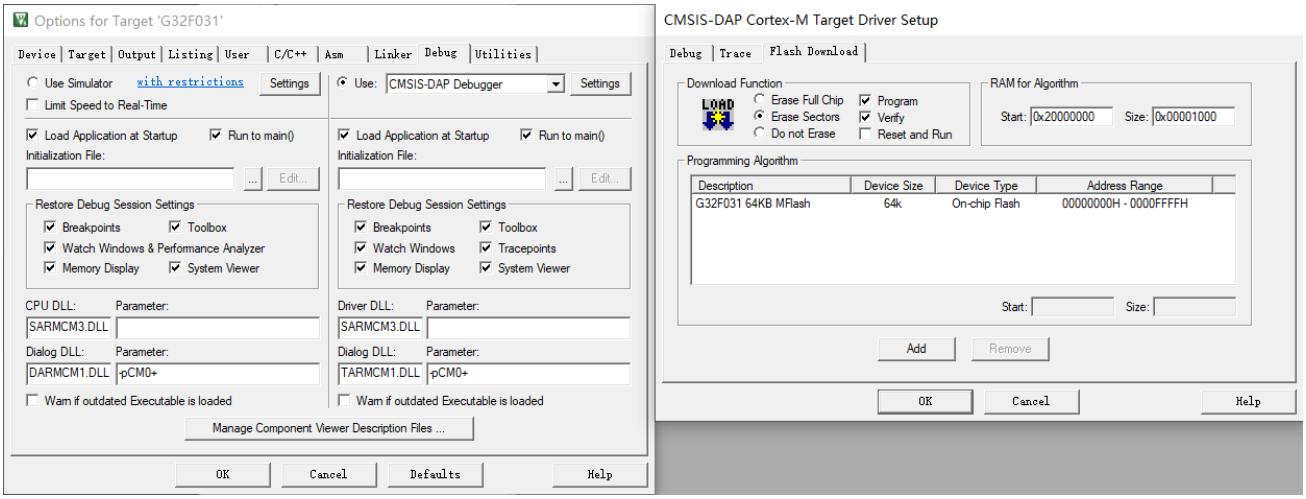
切换 C/C++选项卡，确认开启了 G32F031xx 宏定义，以支持开发 G32F031K8。

图 6 C/C++配置



切换 Debug 选项卡，选中 CMSIS-DAP Debugger 仿真器，并通过 Settings 按钮进入 Flash Download 选项卡，确认选中 G32F031 64KB Flash 下载算法，需要注意算法使用范围为 0x00000000H ~ 0x0000FFFFH，如图 7 所示。

图 7 Debug



10.4.2. 编译下载

点击 MDK 界面上的 Build 按钮进行编译，并确认编译结果无误后，点击 Download 按钮进行下载。下载结果图 10 所示。

图 8 编译工程

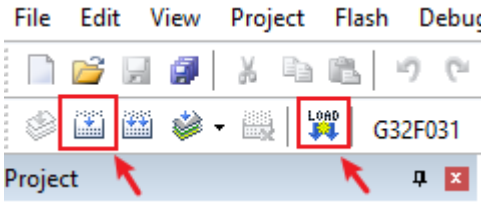


图 9 编译结果

```
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
```

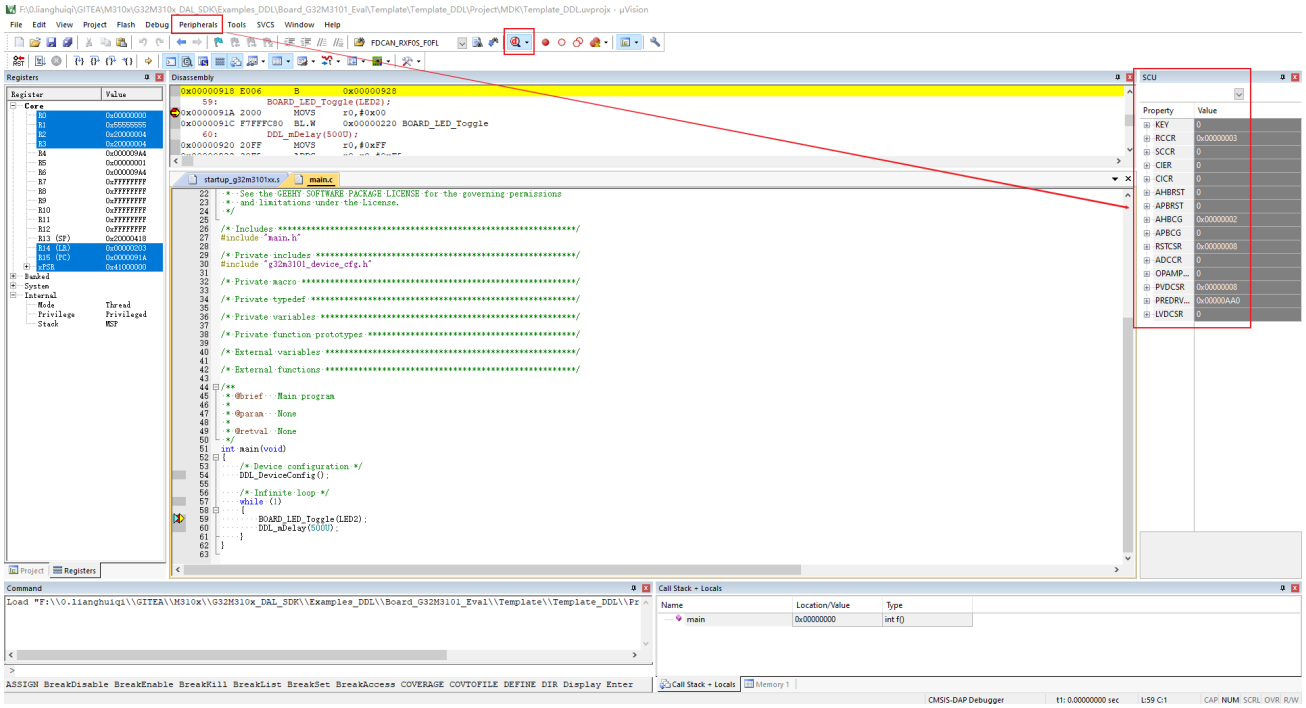
图 10 下载结果

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

10.4.3. 仿真工程

点击界面上的 Start/Stop Debug Session 按钮进行仿真，在仿真界面可以通过 Peripherals、Memory、Registers 选项卡查看 MCU 内核、外设寄存器，内存数据等信息。

图 11 仿真工程


File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
SCU
startup_g32m3101xx.s
main.c
Command
Call Stack + Locals

11. 版本历史

表格 6 文件版本历史

日期	版本	变更历史
2026.1	V1.0	新建
2026.7	V1.1	修改 SDK 命名，增加支持的芯片系列说明

12. 附录

12.1. Keil pack 包的卸载与按照

以下演示如何卸载所有版本的 pack 包，以及按照最新版本的 pack 包。

步骤：

1. 打开 pack 包的安装工具
2. 在 Devices 选项卡中，选择相应的芯片型号
3. 在 Packs 选项卡中，鼠标右击，进行删除操作
4. 在 pack 包的安装工具中，点击 File->Import 选项，进行导入安装 pack 包
5. 选中 pack 包，然后点击打开，进行安装
6. 可在 pack 安装目录下，检查 pack 包内容是否已更新。

图 12

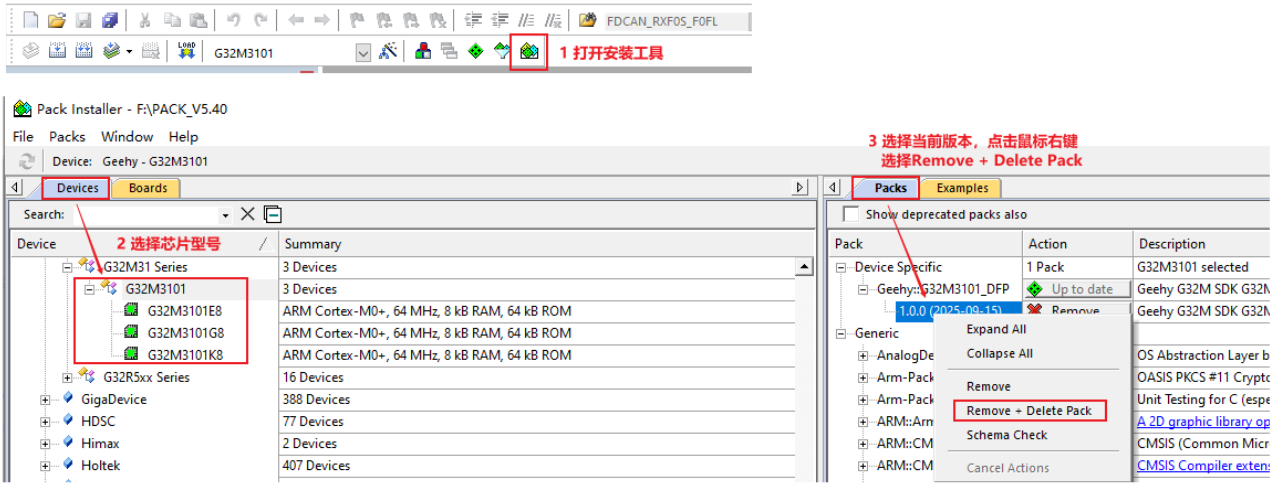


图 13

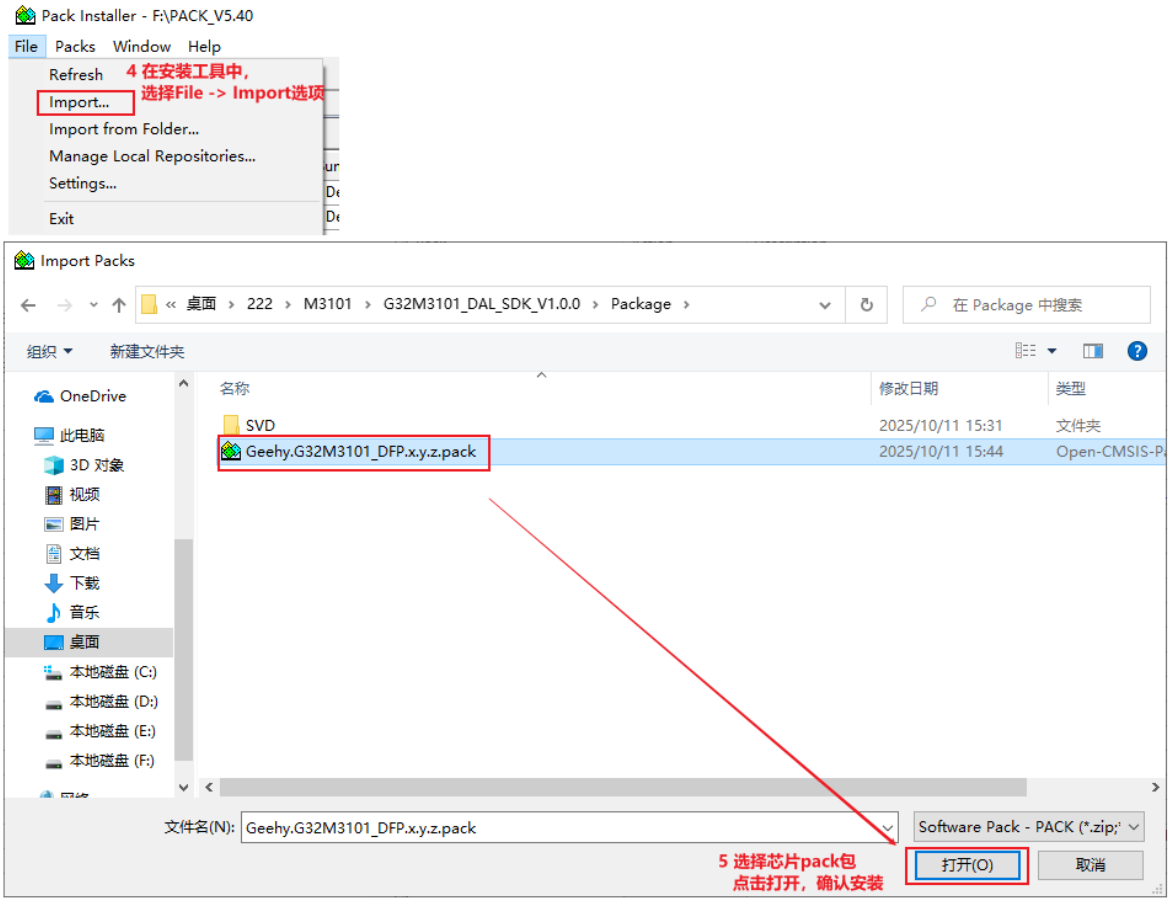
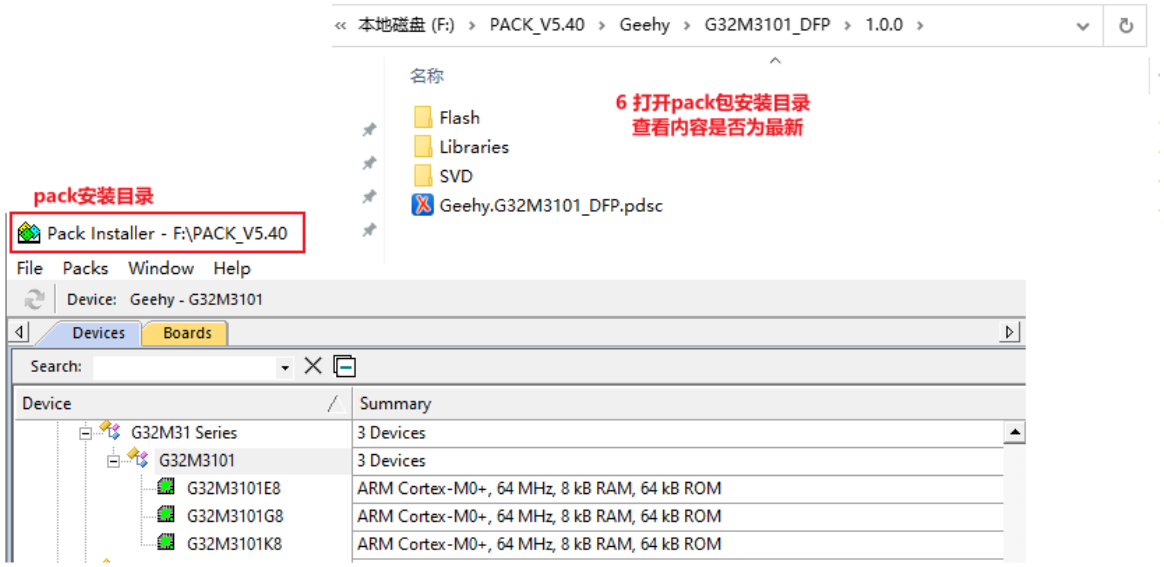

Pack Installer - F:\PACK_V5.40
File Packs Window Help
Refresh
Import...
Import from Folder...
Manage Local Repositories...
Settings...
Exit
4 在安装工具中，
选择File -> Import选项
Import Packs
« 桌面 › 222 › M3101 › G32M3101_DAL_SDK_V1.0.0 › Package ›
在 Package 中搜索
组织
新建文件夹
OneDrive
此电脑
3D 对象
视频
图片
文档
下载
音乐
桌面
本地磁盘 (C:)
本地磁盘 (D:)
本地磁盘 (E:)
本地磁盘 (F:)
名称
修改日期
类型
SVD
2025/10/11 15:31
文件夹
Geehy.G32M3101_DFP.x.y.z.pack
2025/10/11 15:44
Open-CMSIS-P
文件名(N): Geehy.G32M3101_DFP.x.y.z.pack
Software Pack - PACK (*.zip;*
5 选择芯片pack包
点击打开，确认安装
打开(O)
取消

图 14

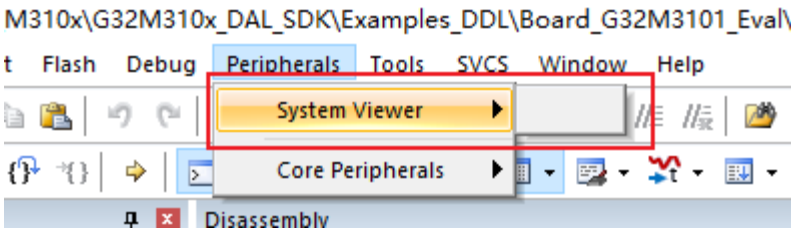

« 本地磁盘 (F:) › PACK_V5.40 › Geehy › G32M3101_DFP › 1.0.0 ›
名称
Flash
Libraries
SVD
Geehy.G32M3101_DFP.pdsc
6 打开pack包安装目录
查看内容是否为最新
pack安装目录
Pack Installer - F:\PACK_V5.40
File Packs Window Help
Device: Geehy - G32M3101
Devices
Boards
Search:
Device
Summary
G32M31 Series
3 Devices
G32M3101
3 Devices
G32M3101E8
ARM Cortex-M0+, 64 MHz, 8 kB RAM, 64 kB ROM
G32M3101G8
ARM Cortex-M0+, 64 MHz, 8 kB RAM, 64 kB ROM
G32M3101K8
ARM Cortex-M0+, 64 MHz, 8 kB RAM, 64 kB ROM

12.2. Keil 关联 SVD 外设调试信息

问题：

安装完 pack 包之后，进入调试时，发现缺少了外设模块的信息，如下图所示：

图 15

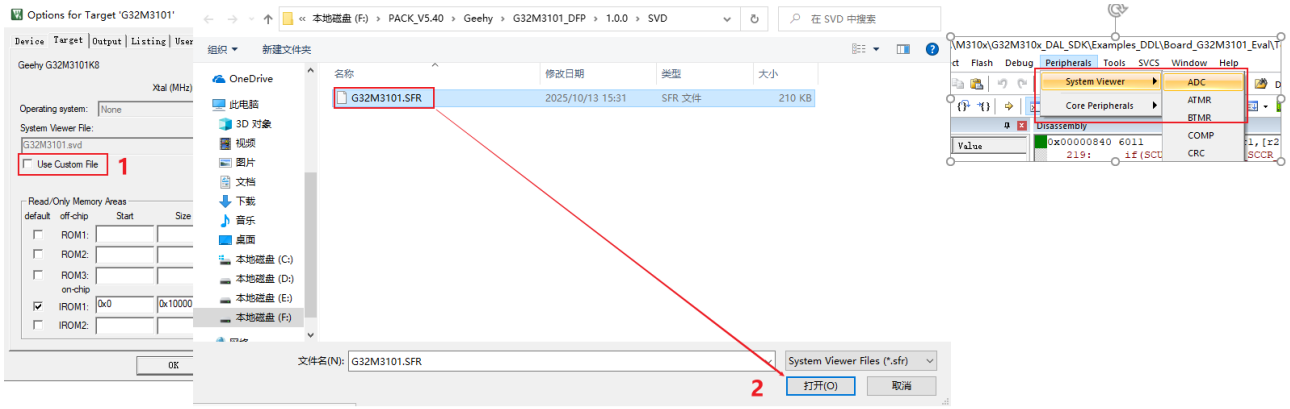


解决方案：

1. 点击魔术棒，选择 Target 一栏”Options for Target -> Target”，勾选 UseCustom；
2. 选中 pack 包安装目录下的 SFR 文件，并点击打开；
3. 再次进入调试，即可显示有外设的调试信息。

步骤如下图所示：

图 16



声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2025 珠海极海半导体有限公司 – 保留所有权利