

开发环境学习手册

Keil 篇

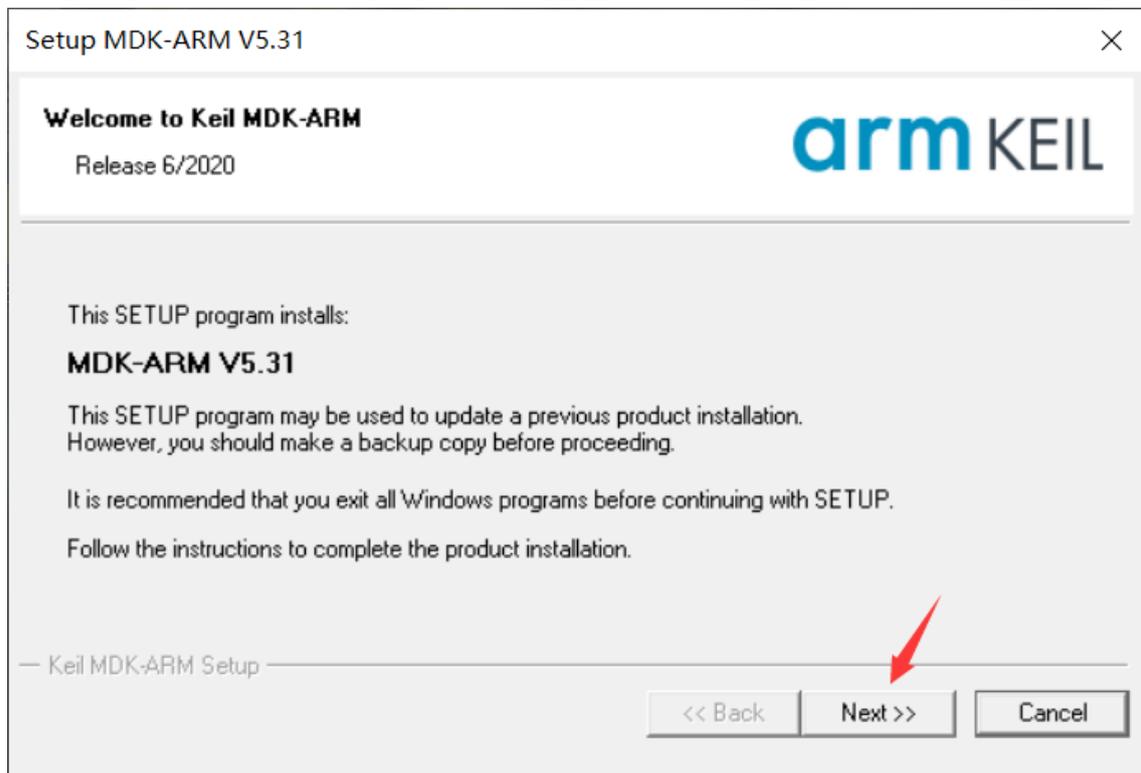
1.1 Keil 安装步骤

安装包链接: <https://cloud.tencent.com/developer/article/1605195> (keil 推荐在 MDK5.31 (含) 以上)。

一、下载并解压安装包, 并按步骤完成安装

名称	修改日期	类型	大小
今天 (1)			
 MDK531.EXE	2022/9/26 13:47	应用程序	896,983 KB
上周 (5)			
 TencentDocsSetup.exe	2022/9/21 11:19	应用程序	132,287 KB
 Setup_EmbeddedStudio_ARM_v634...	2022/9/20 13:47	应用程序	775,584 KB

二、运行安装程序, 点击 next



三、勾选 accept, 点击 next



四、选择安装路径, 点击 next (最好选择默认路径, 如果自行修改需要注意不要添加中文)



注意事项:

- 安装路径不能有中文
- 安装目录不能和 Keil4 以及 KeilC51 重合

五、信息随意填写，点击 next

Setup MDK-ARM V5.31 ×

Customer Information **arm KEIL**

Please enter your information.

Please enter your name, the name of the company for whom you work and your E-mail address.

First Name:

Last Name:

Company Name:

E-mail:

— Keil MDK-ARM Setup —

<input type="button" value=" Next >> " /><input type="button" value=" Cancel " />

六、等待安装

Setup MDK-ARM V5.31 ×

Setup Status **arm KEIL**

MDK-ARM Setup is performing the requested operations.

Install Files ...

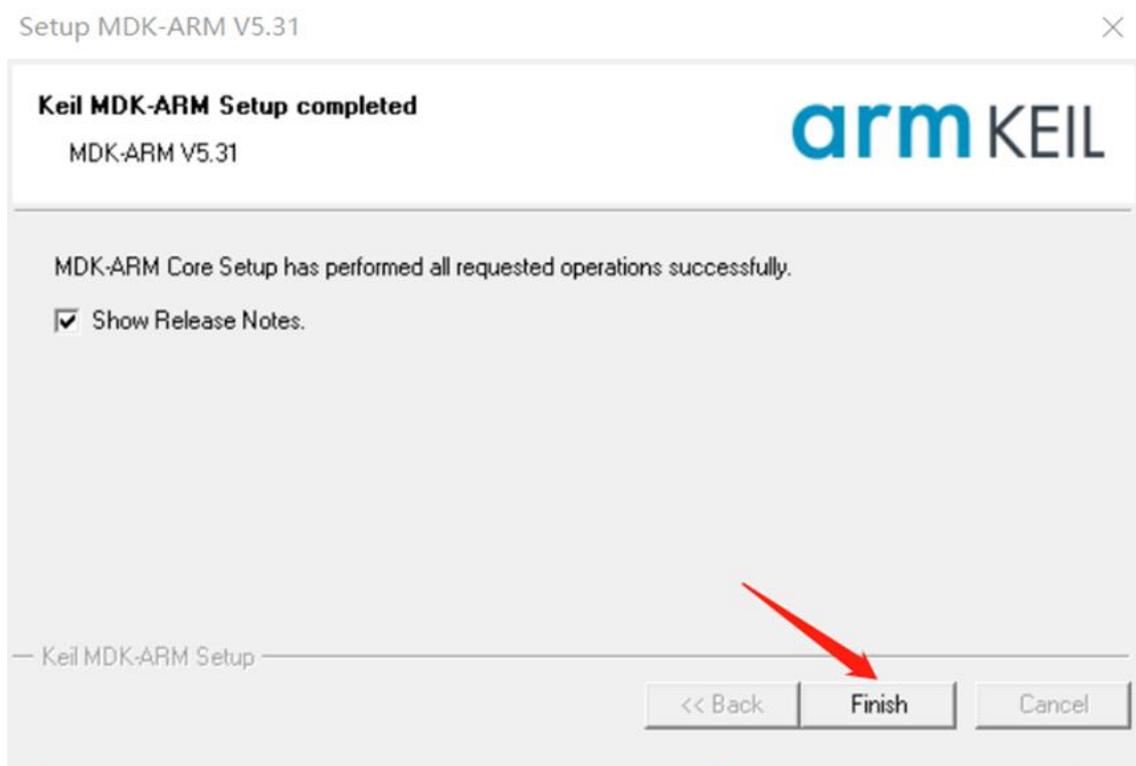
Installing cpp_wsenu.b.



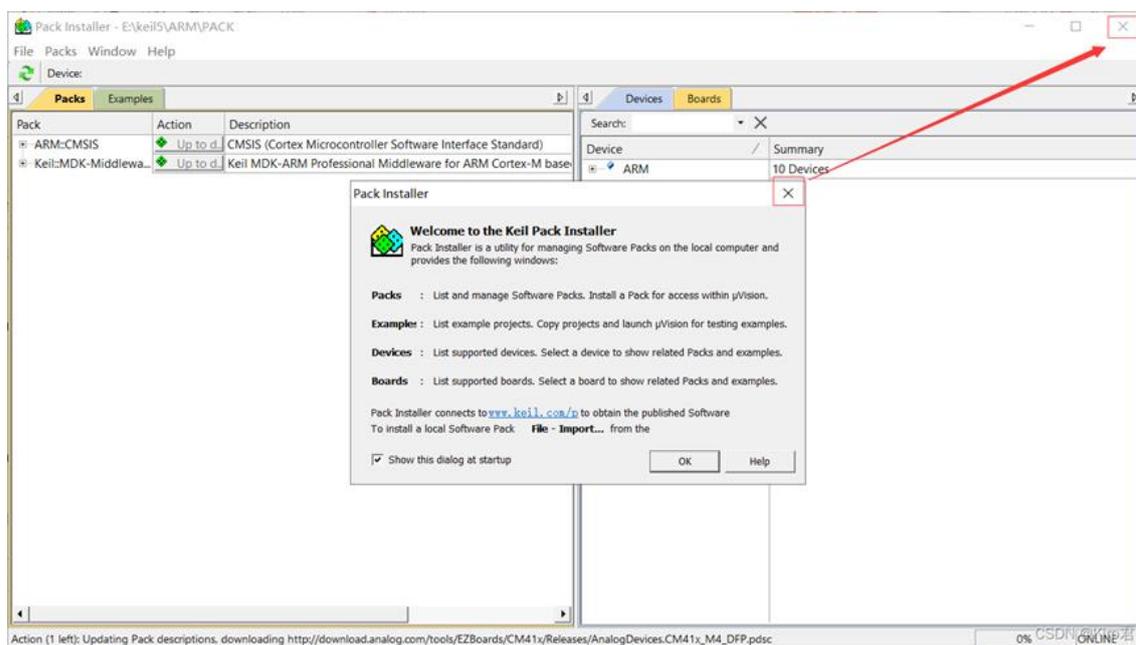
— Keil MDK-ARM Setup —

<input type="button" value=" Next >> " /><input type="button" value=" Cancel " />

七、点击 finish，完成安装

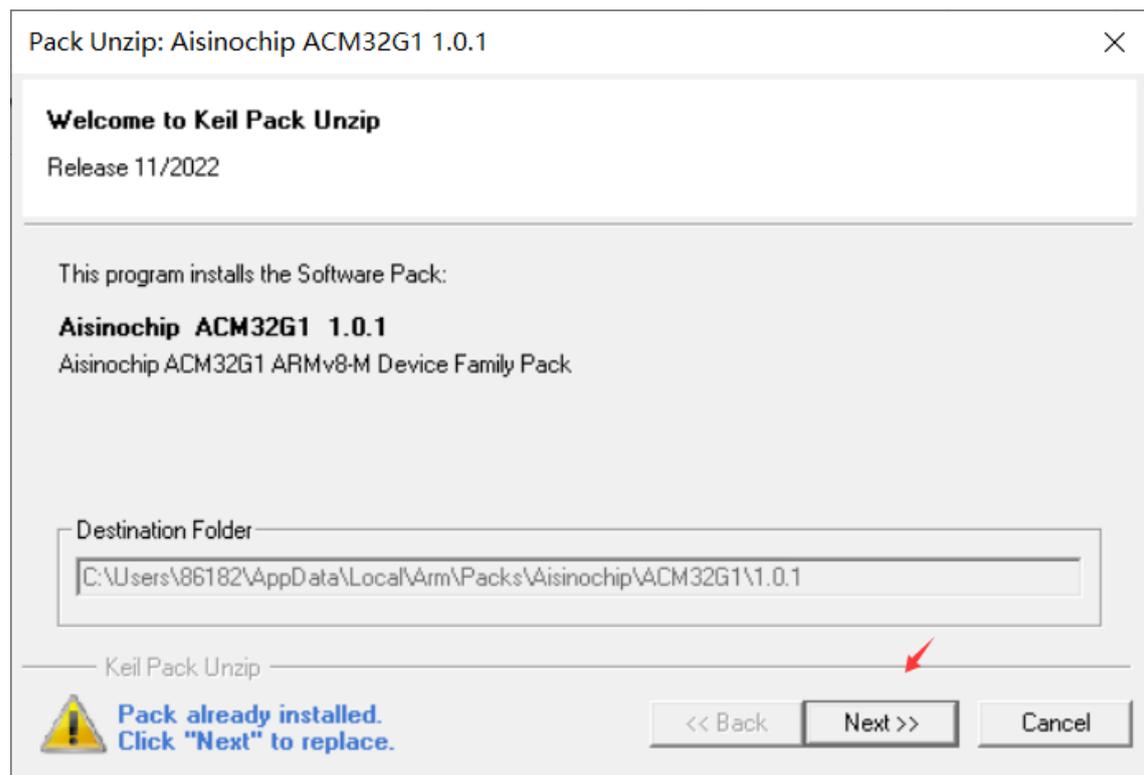
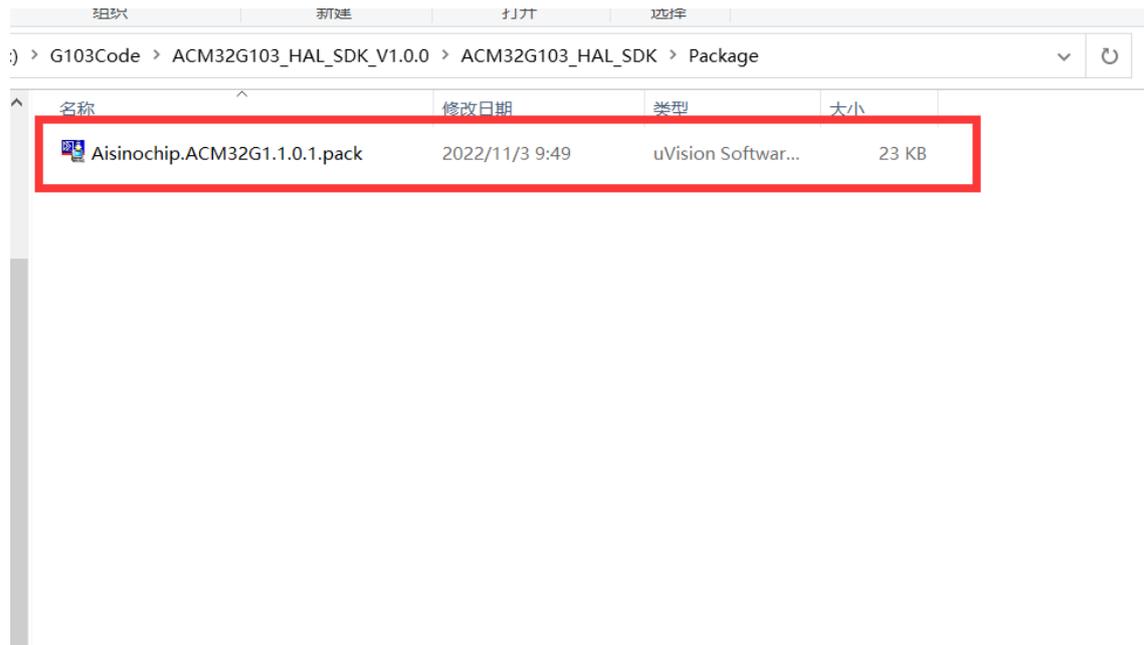


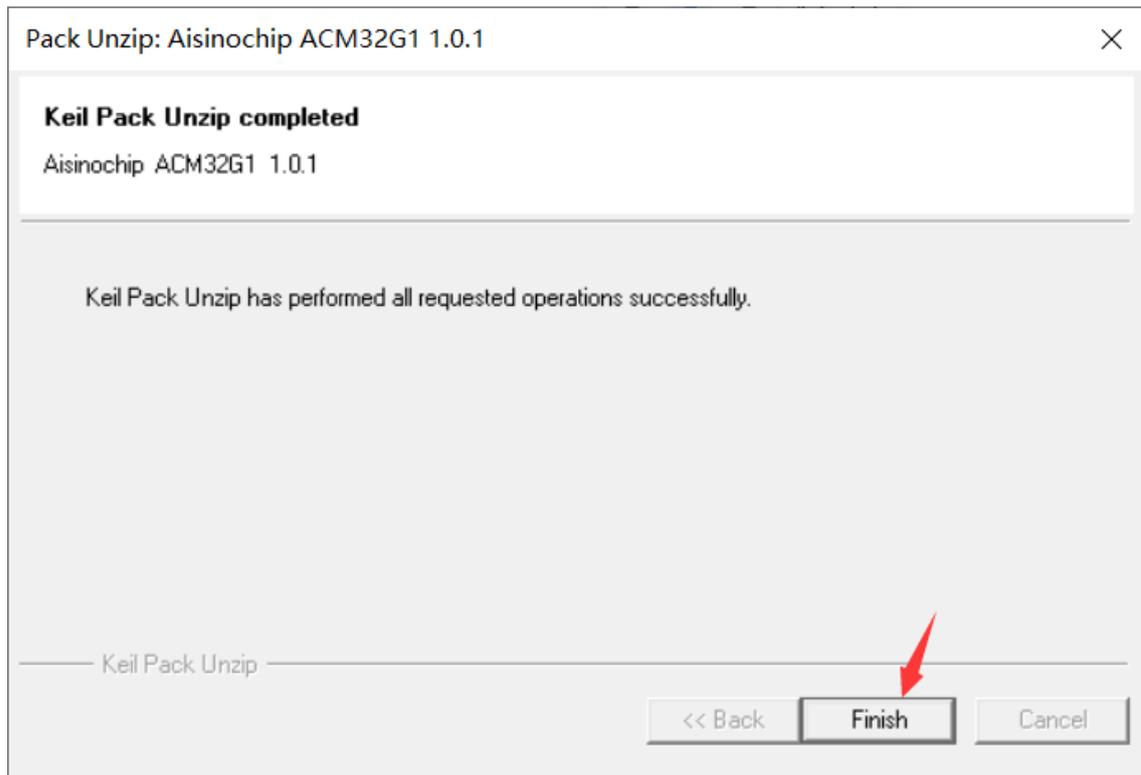
八、突然会弹出“Pack installer”页面，先后关闭两个窗口



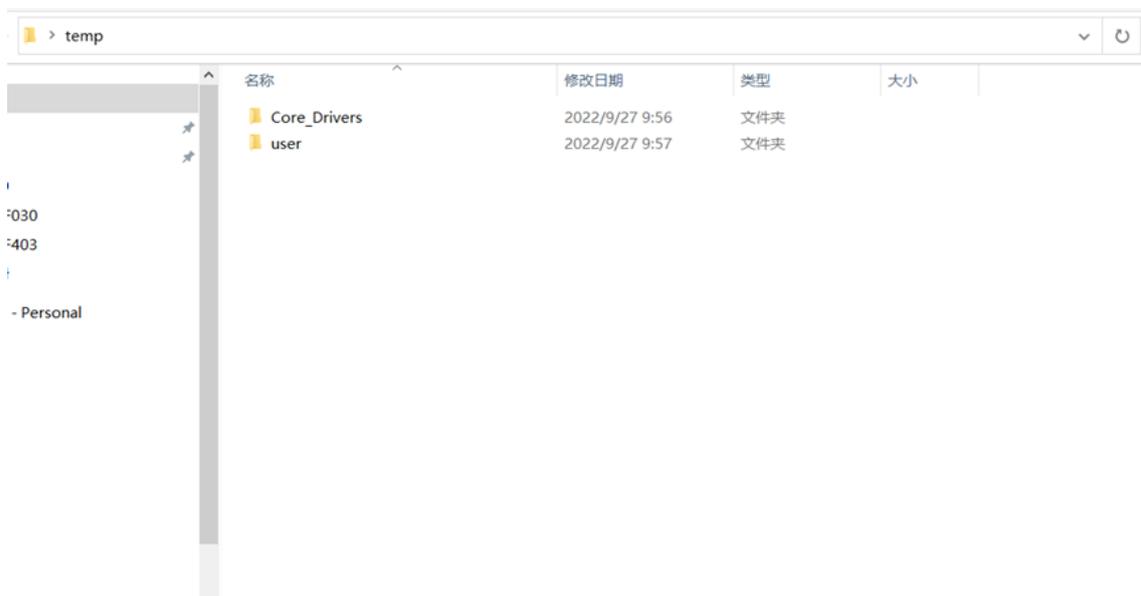
1.2 Keil 使用流程（以 ACM32G1XX 系列为例）

一、安装航芯的芯片包

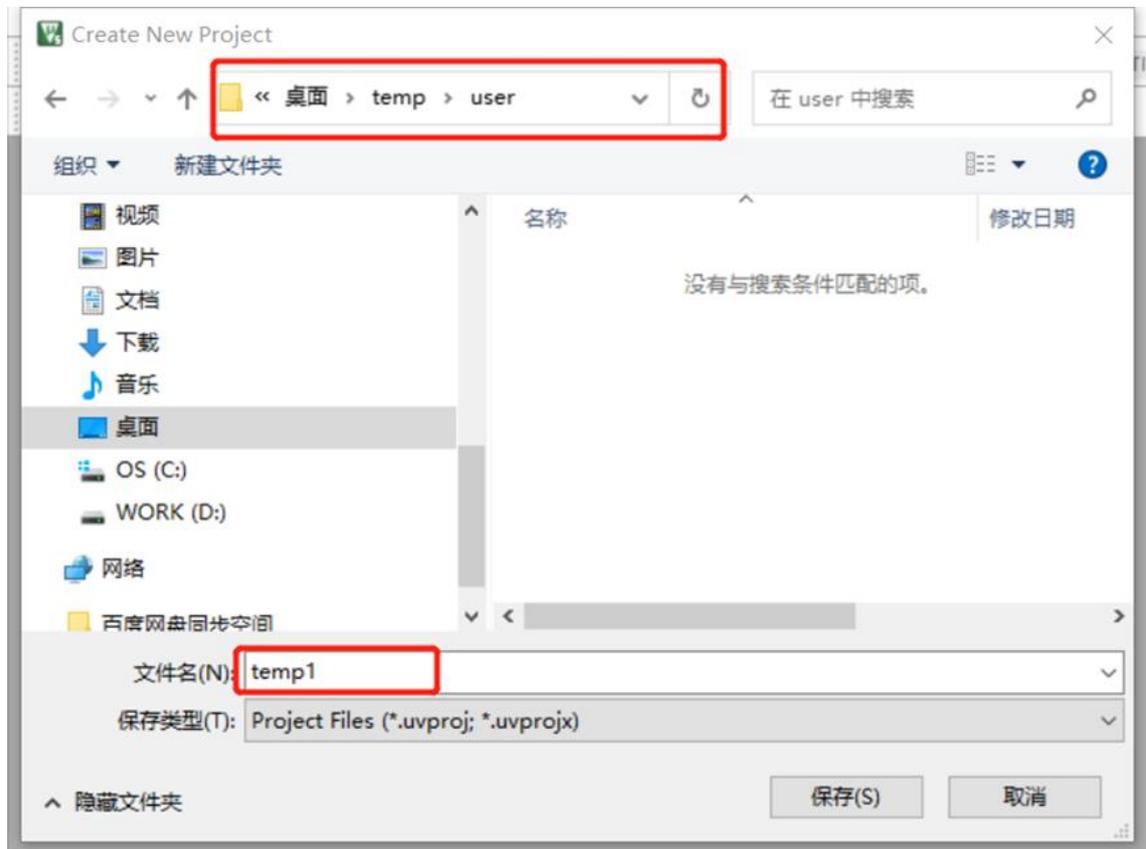
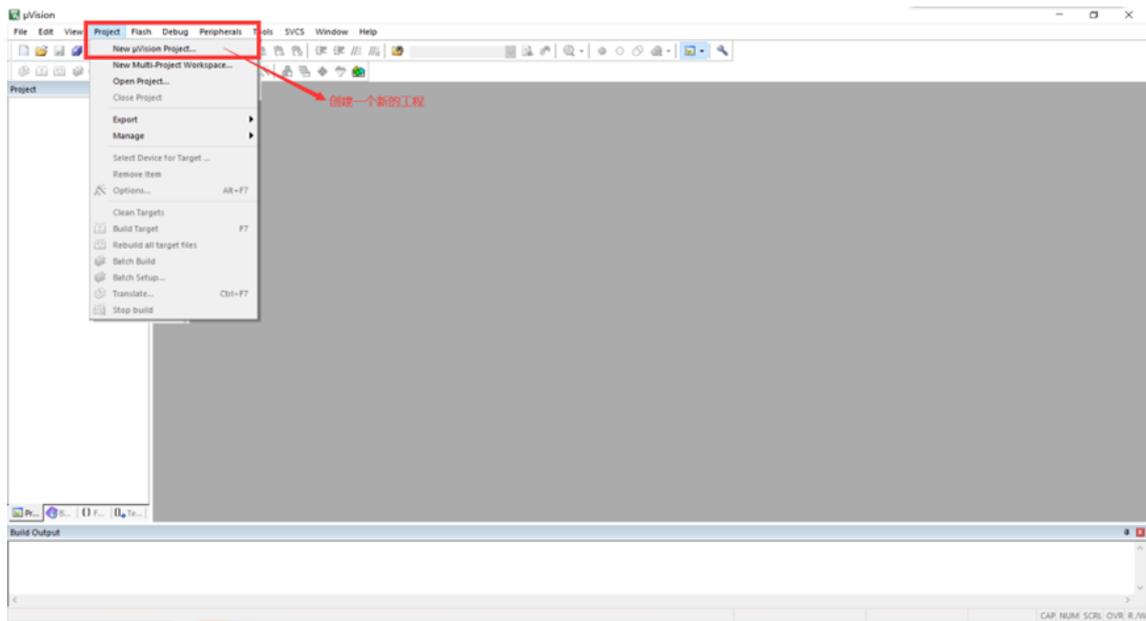




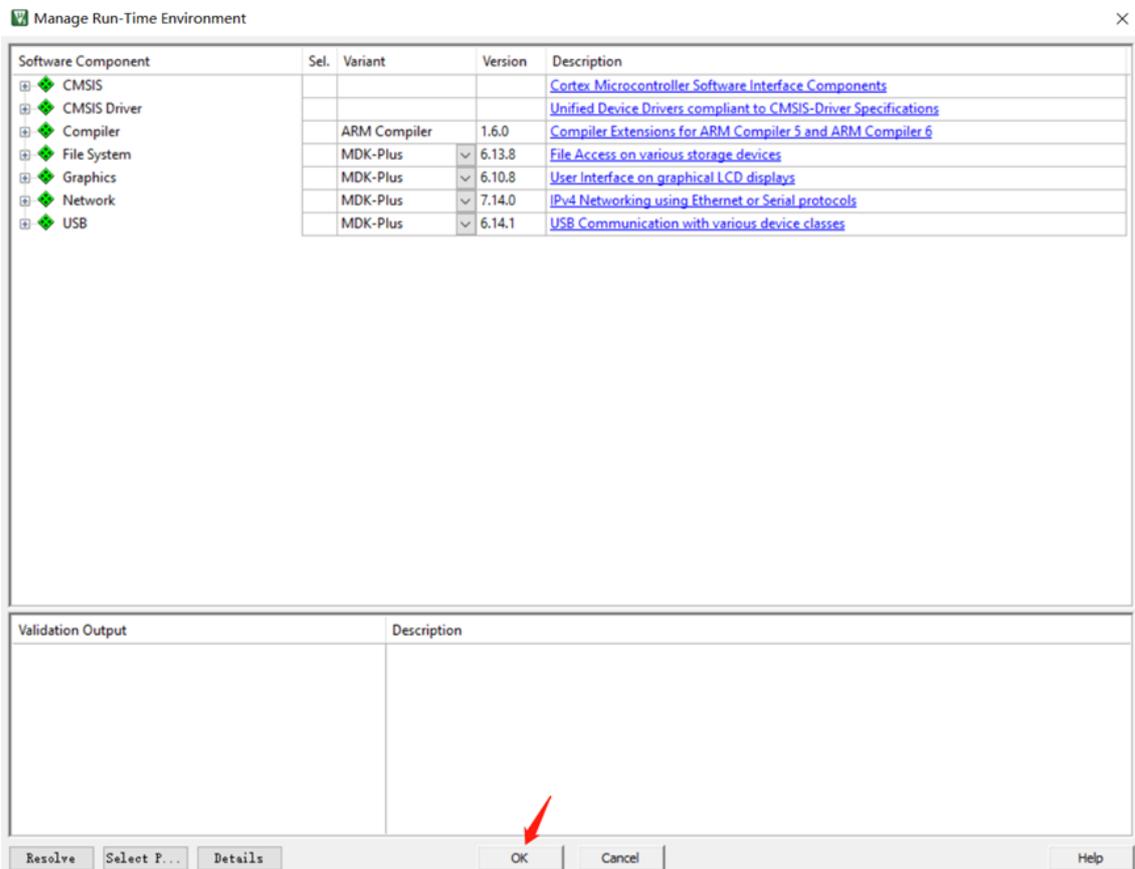
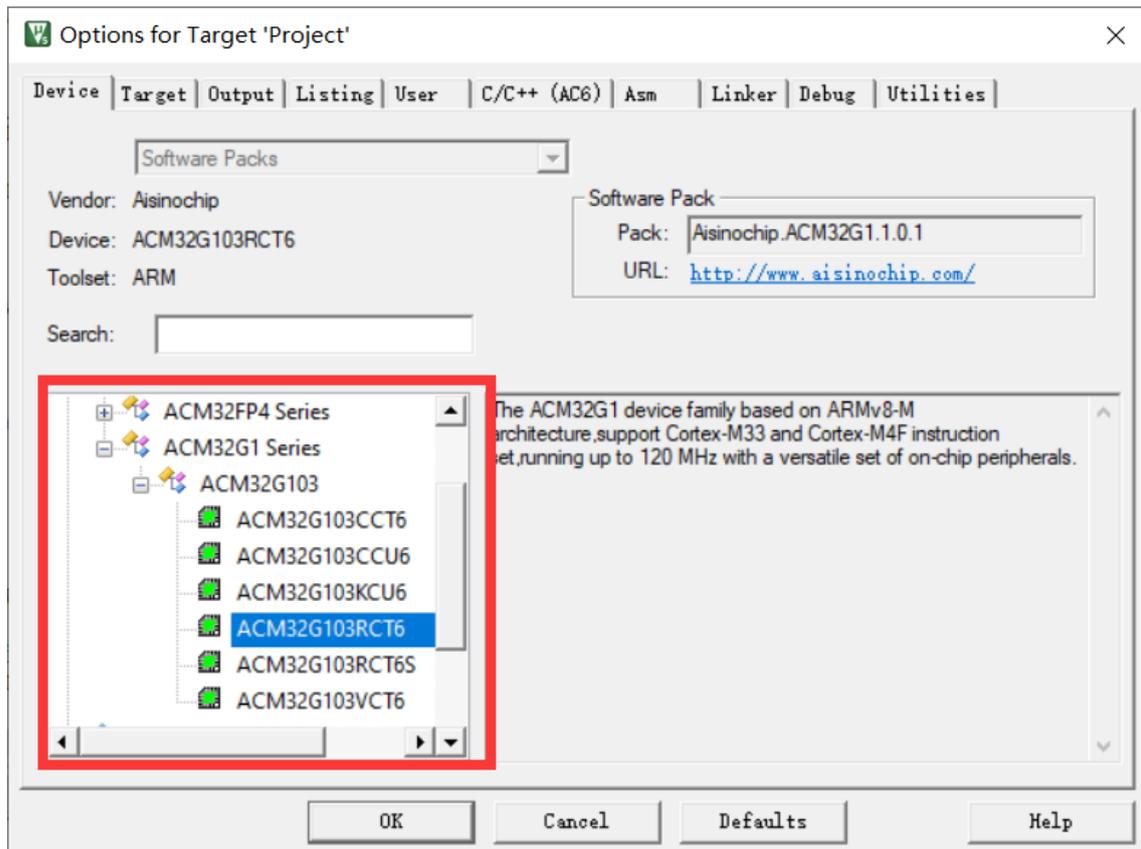
二、创建一个新的文件夹 **temp**，放入航芯提供的 **Core_Drivers** 文件，同时在里面创建一个新的文件夹 **user**。



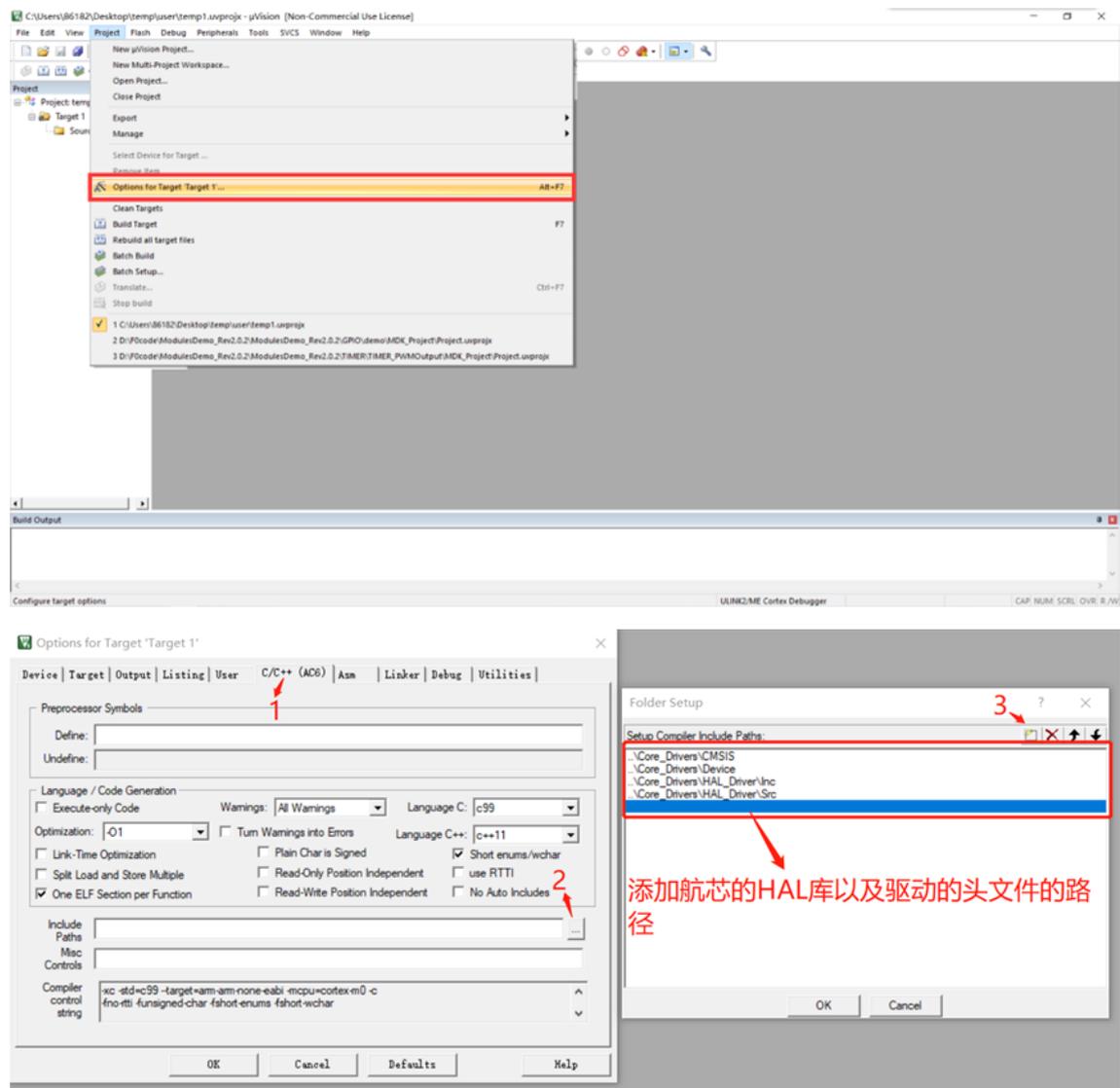
三、打开 keil，建立一个新的工程



四、安装好了航芯的芯片包后，可以选择对应的设备号（没有找到对应的设备号可能是芯片包未安装好）

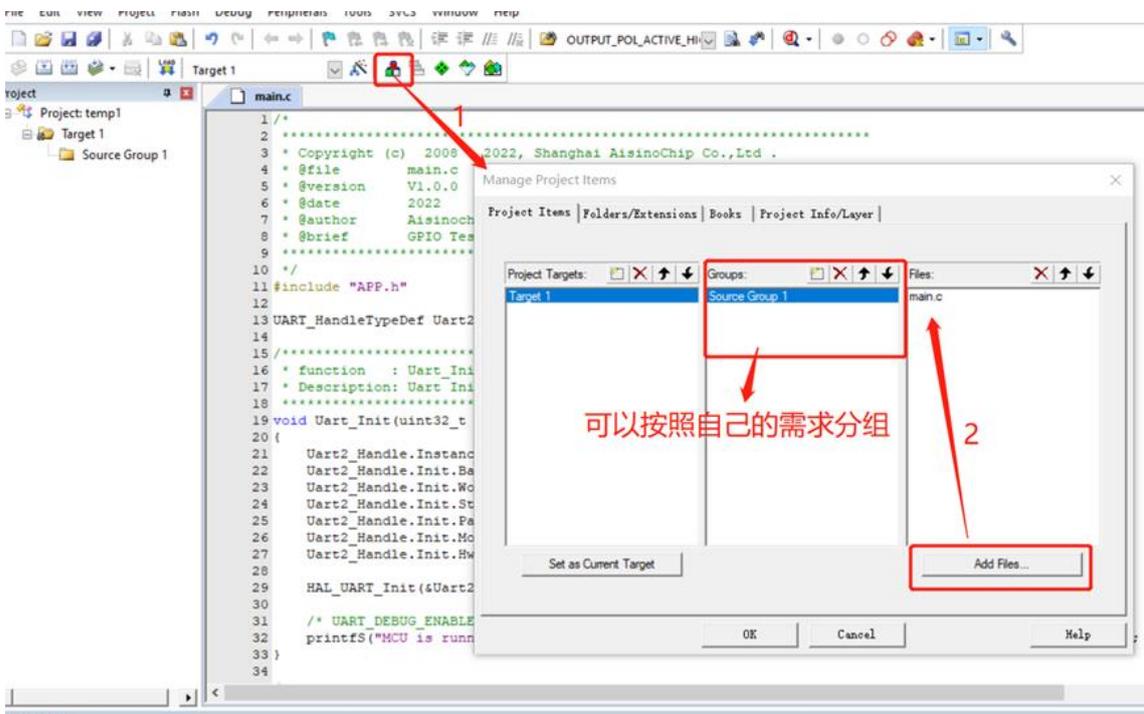
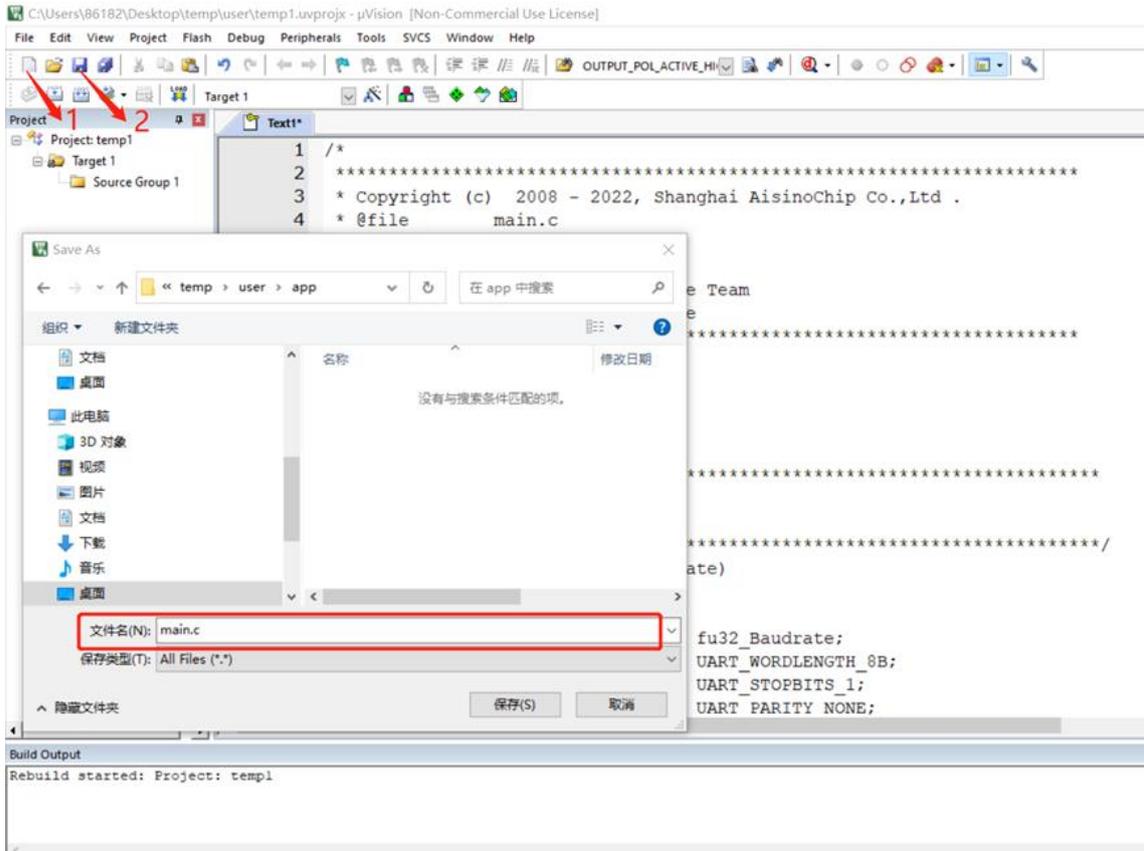


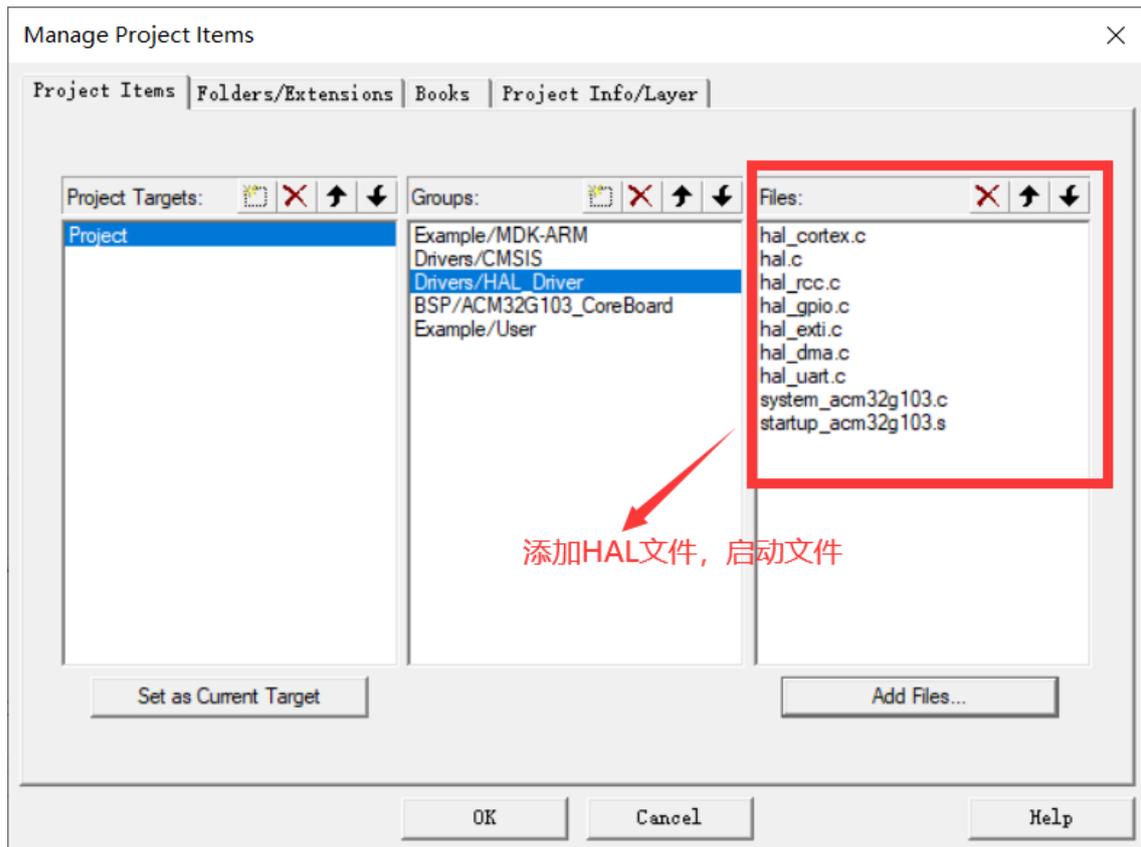
五、添加头文件（.h 文件）



六、添加源文件（.C 文件）

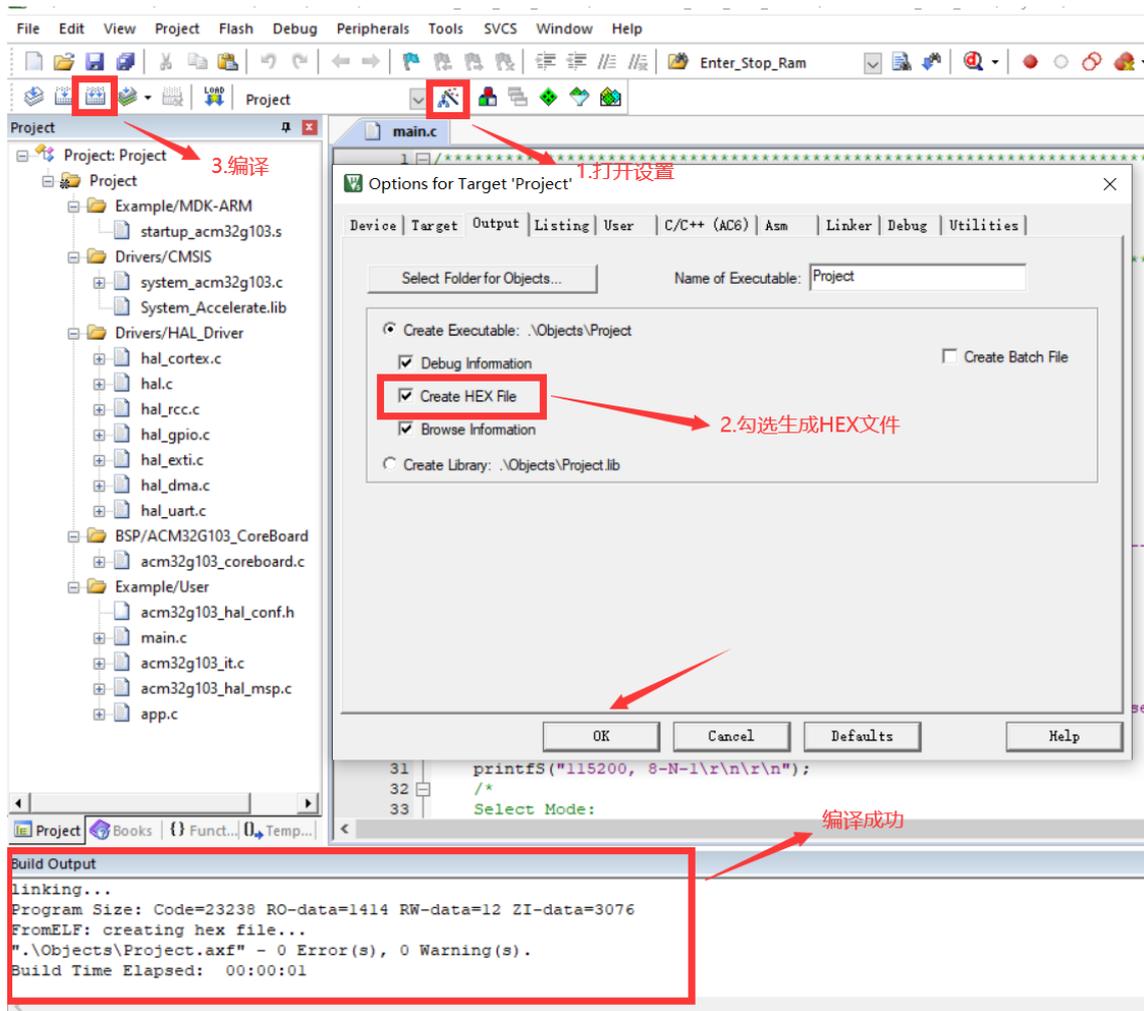
第一步打开空白页，编写程序（可以使用航芯提供的 demo 程序或者自己编写的程序）后，第二步保存到对应的工程文件夹。



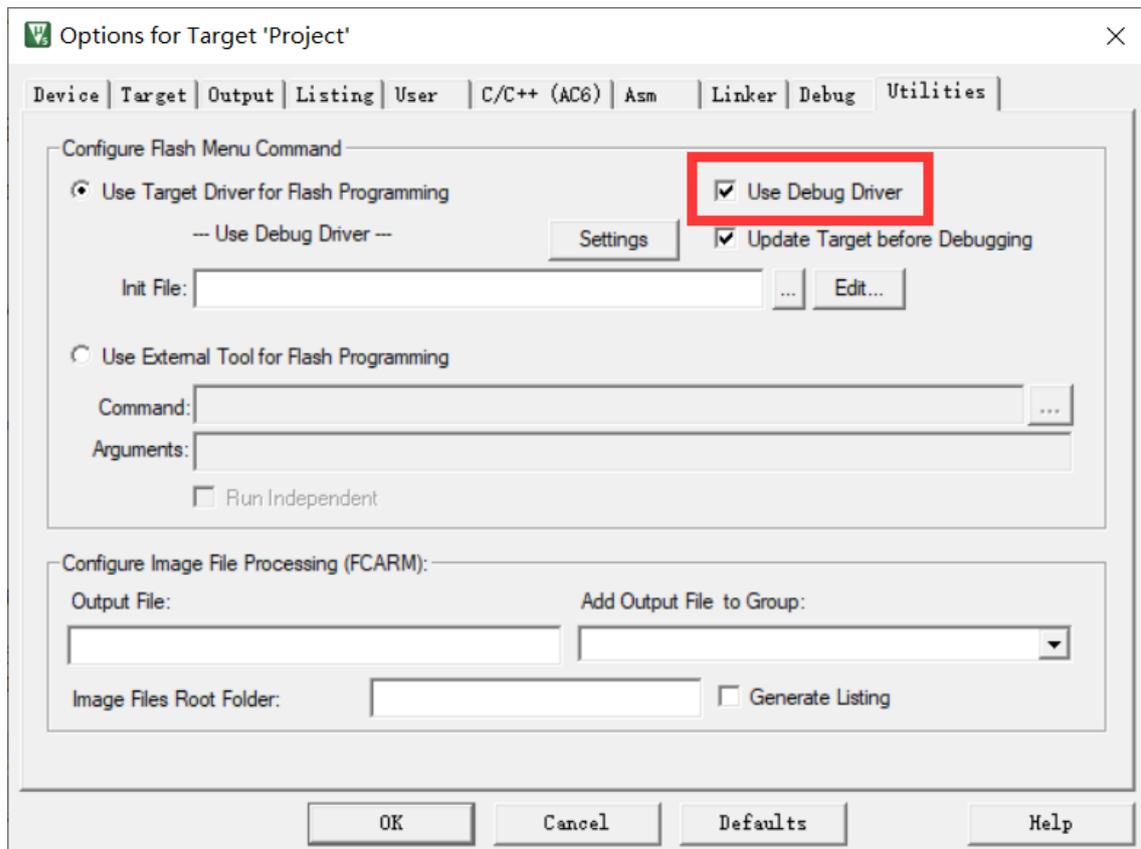
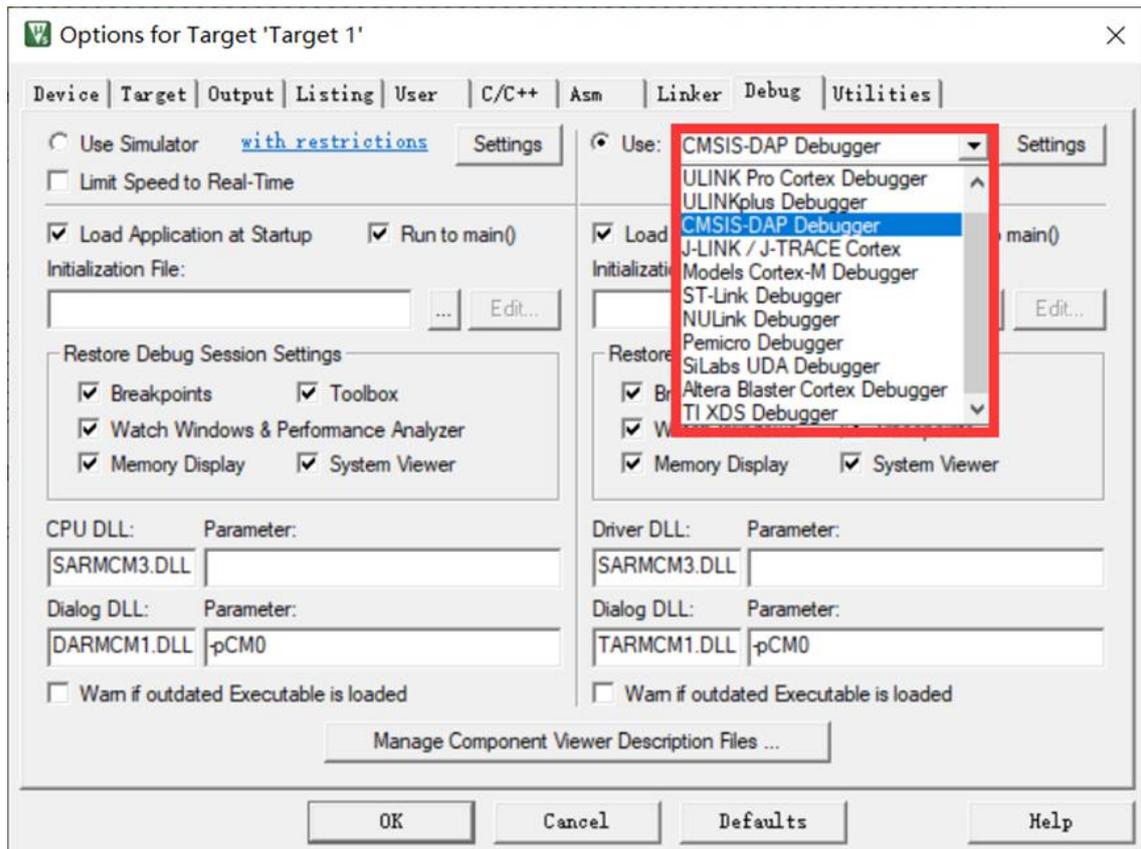


1.3 keil 编译、下载、运行

一、工程编译

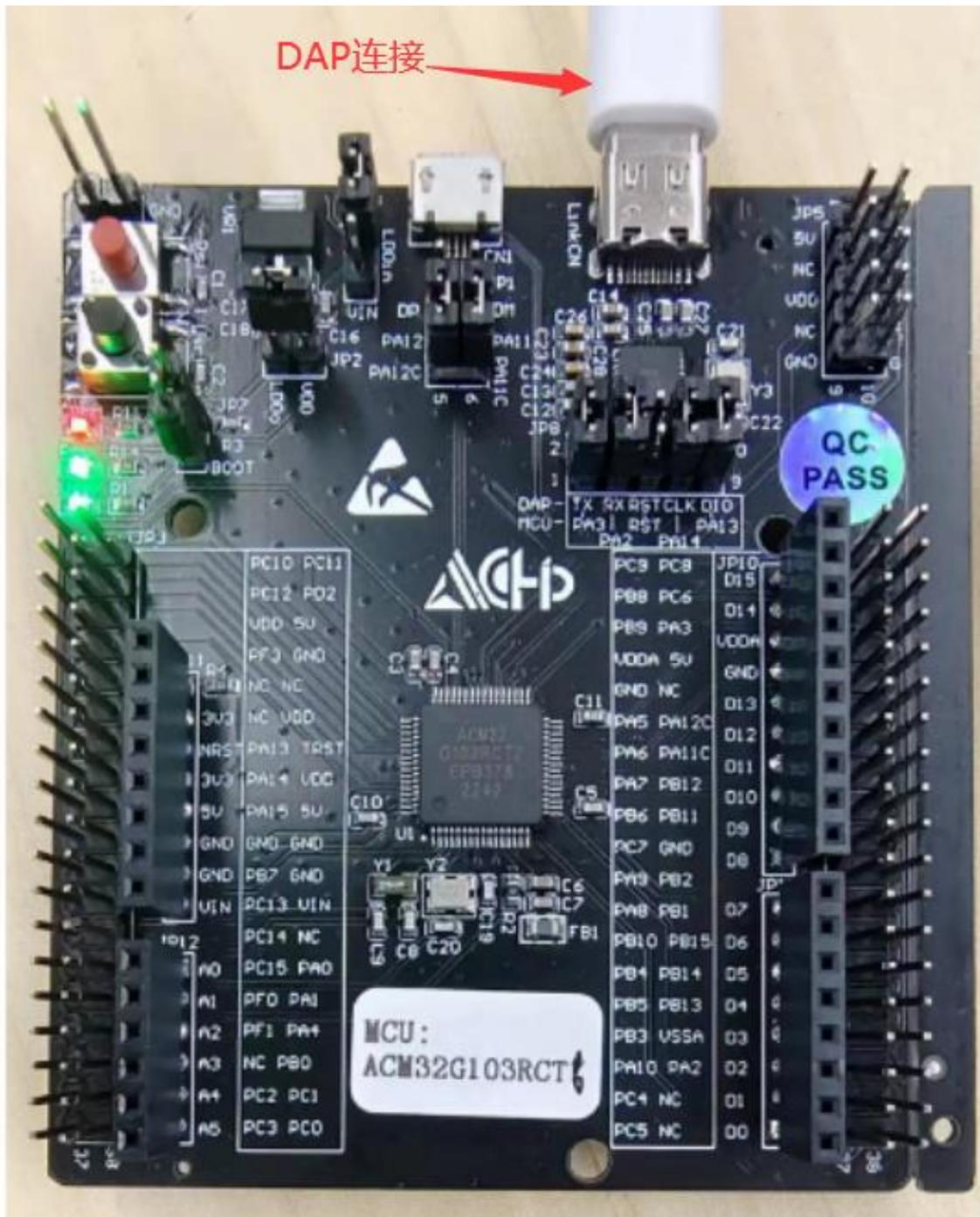


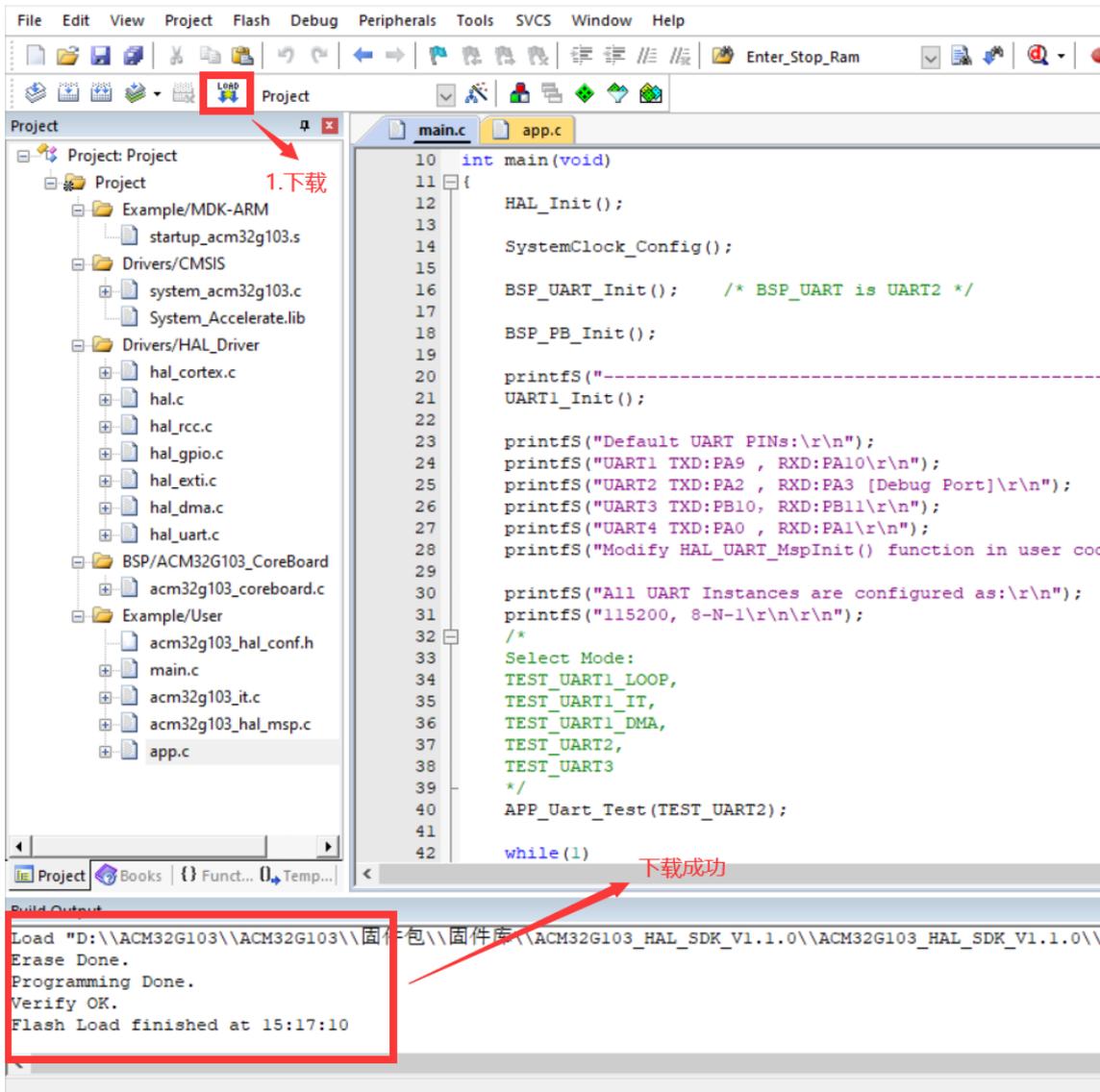
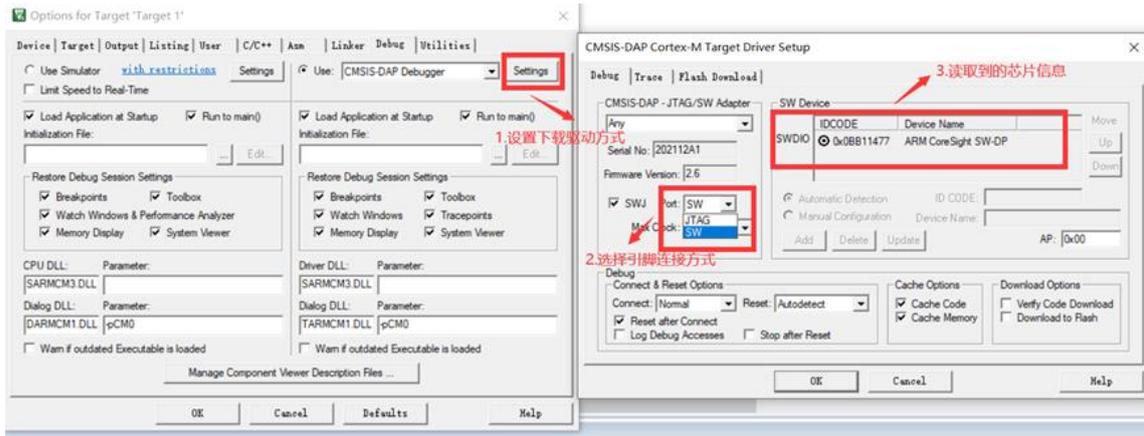
二、下载程序（ACM32G1XX 系列支持的在线仿真器包括：J-Link-V9（含）以上、U-Link2、CMSIS-DAP 等，使用 J-Link 在线调试时，Keil 推荐在 MDK5.33（含）以上，J-Link 驱动建议在 V6.70e（含）以上）



航芯的 core 开发板是带有 DAP 连接的，所以可以直接选择 CMSIS-DAP Debugger，通过 Settings 查看是否正常连接。如果需要使用 Jlink 方式连接，可以参考《航芯通用

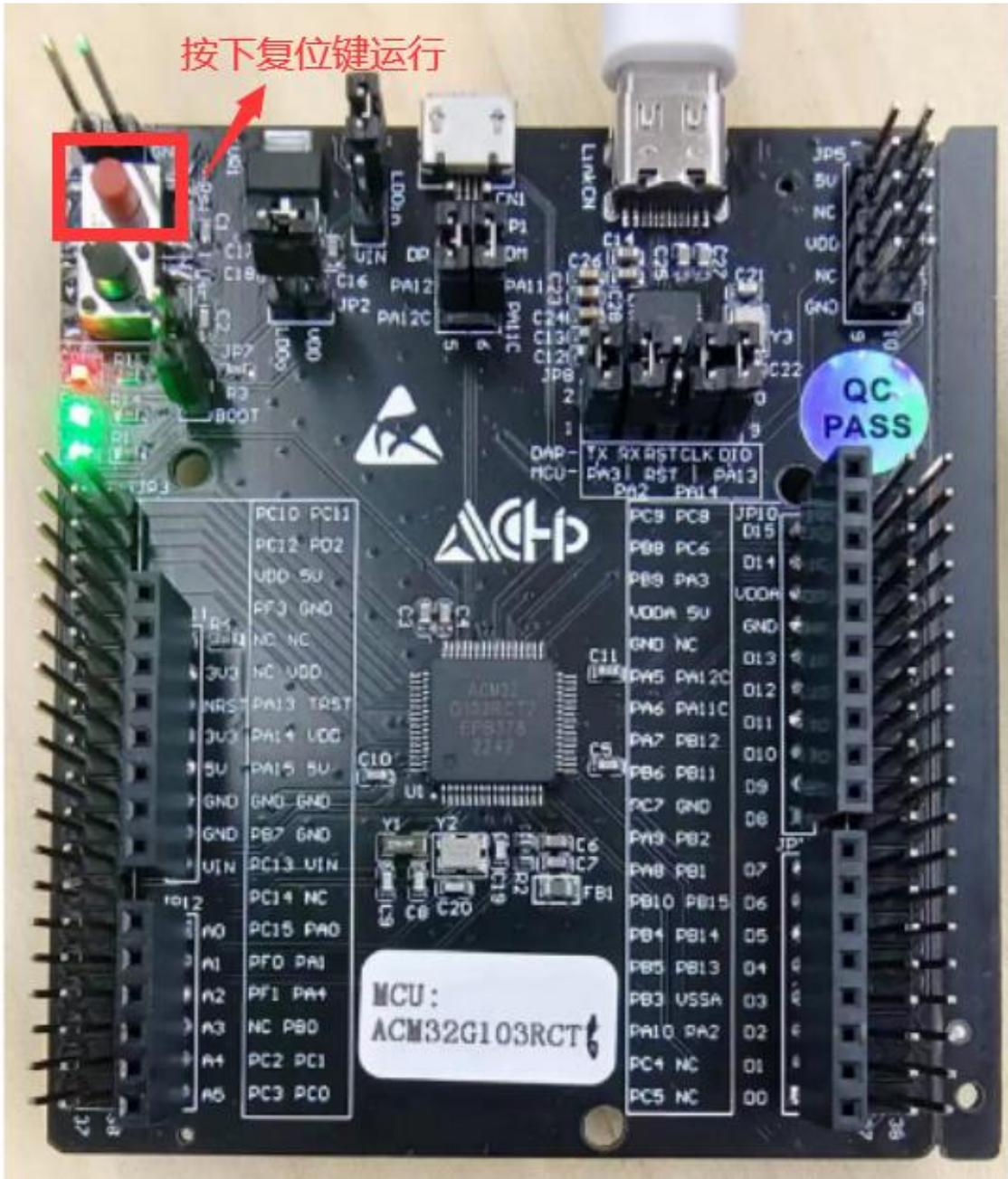
MCU 使用 JFlash 烧录程序的方法说明.pdf》。





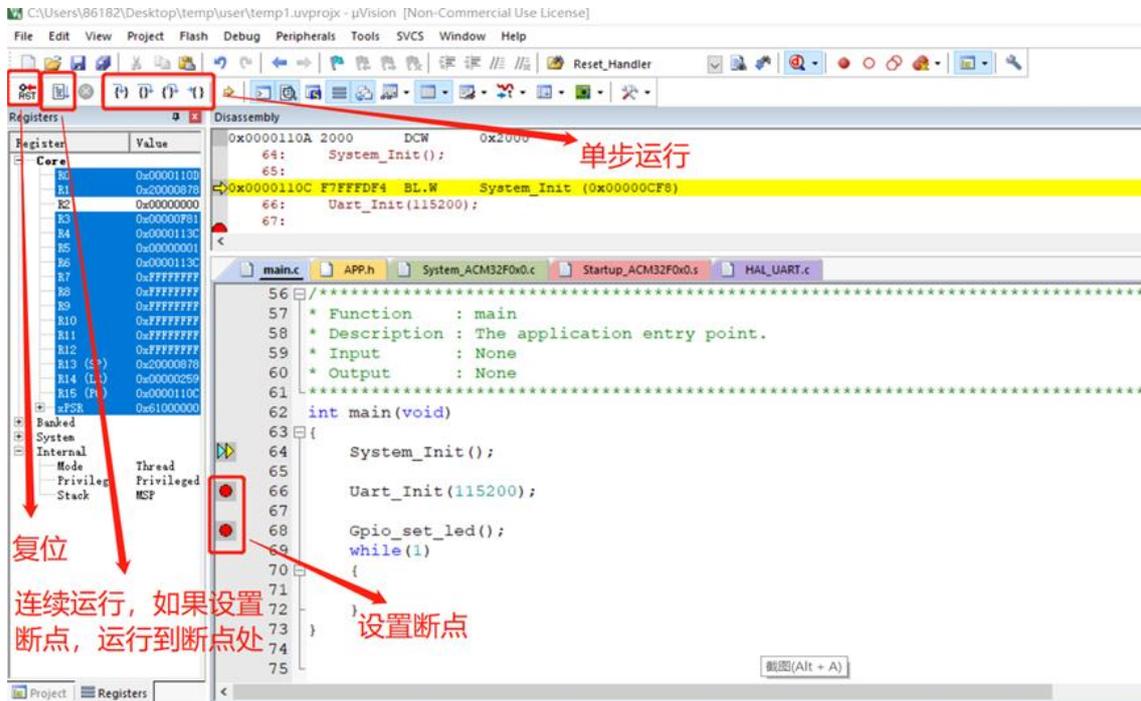
三、运行（目前芯片不支持下载后自启动，需要按下 RESET 键后，程序才能运行），以 UART_Printf_Demo 为例，部分示例代码（完整代码见 ACM32G103 的 SDK 库）和现象如下：

按下复位键运行

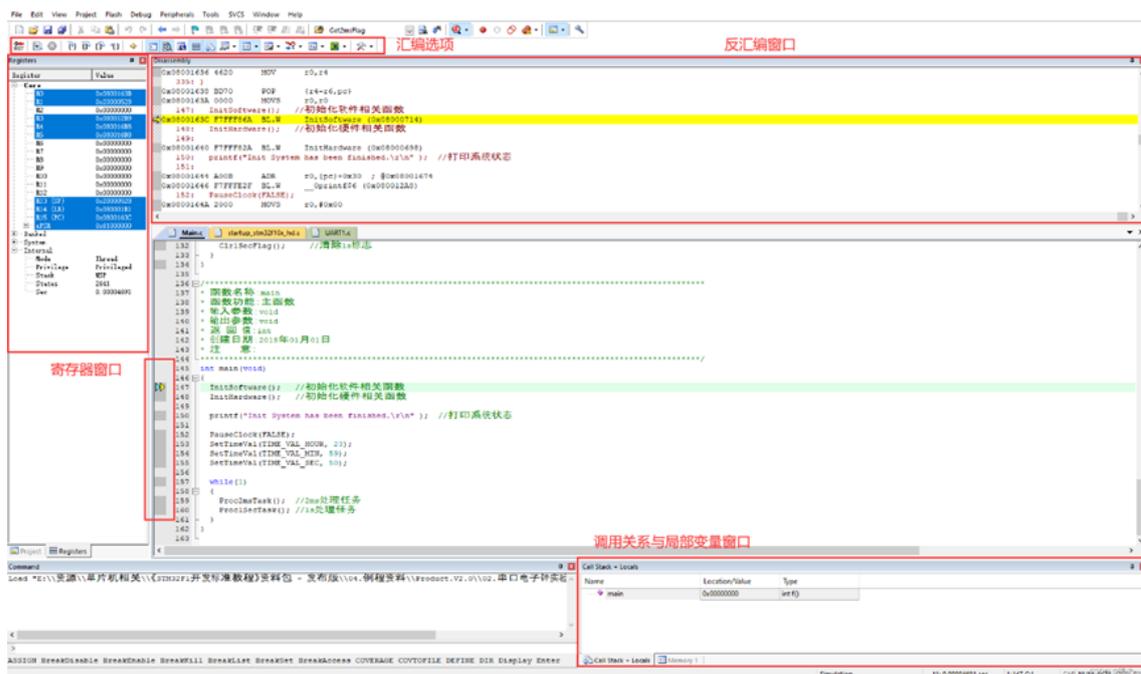


二、调试工具栏介绍

断点是调试器的功能之一，可以让程序停止在设置断点的语句。在调试过程中，可在程序的某一处设置断点再点击 Run 运行，此时当程序运行至该位置时自动停止

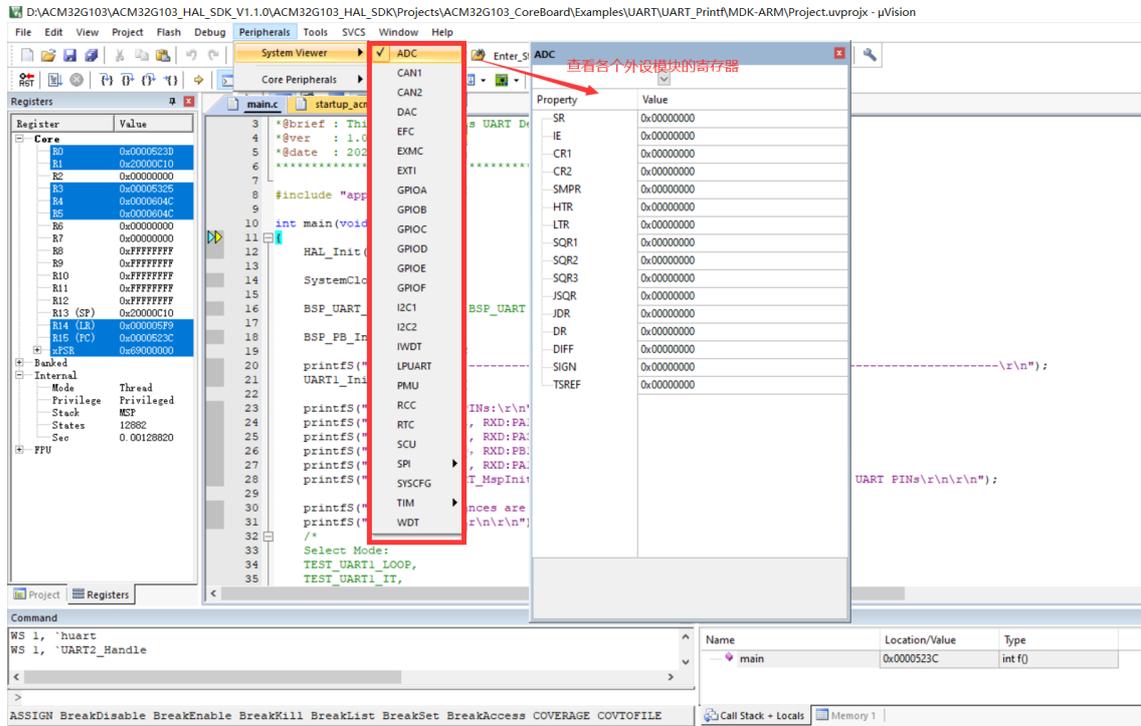


调试选项	名称	介绍
	Reset	实现程序复位，以重新运行
	Run	使程序开始运行
	Stop	使程序停止运行（程序运行状态时才有效）
	Step	运行当前行代码（若为函数则进入函数）
	Step Over	运行至下一行代码（若为函数会将函数执行完毕）
	Step Out	跳出当前函数
	Run to Cursor line	运行至蓝色箭头处
	Show Next Statement	跳转至当前代码（方便在打开多个文件时找到代码运行处）

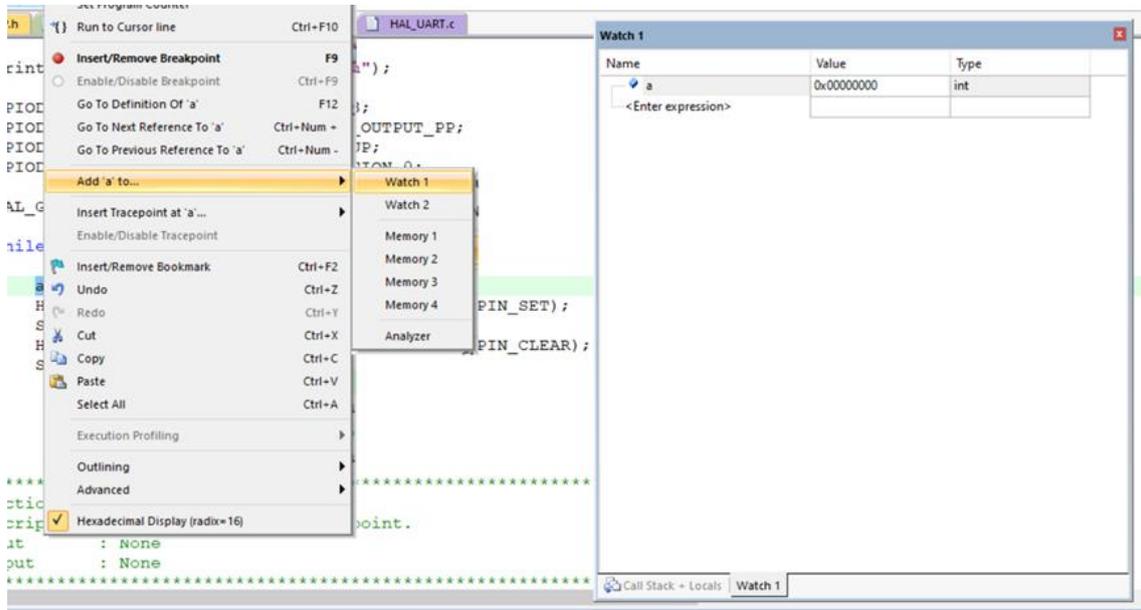


三、调试窗口介绍

1 查看外设寄存器的值，参考《航芯 ACM32G103_用户手册_V1.1.pdf》查看各个外设寄存器的功能。

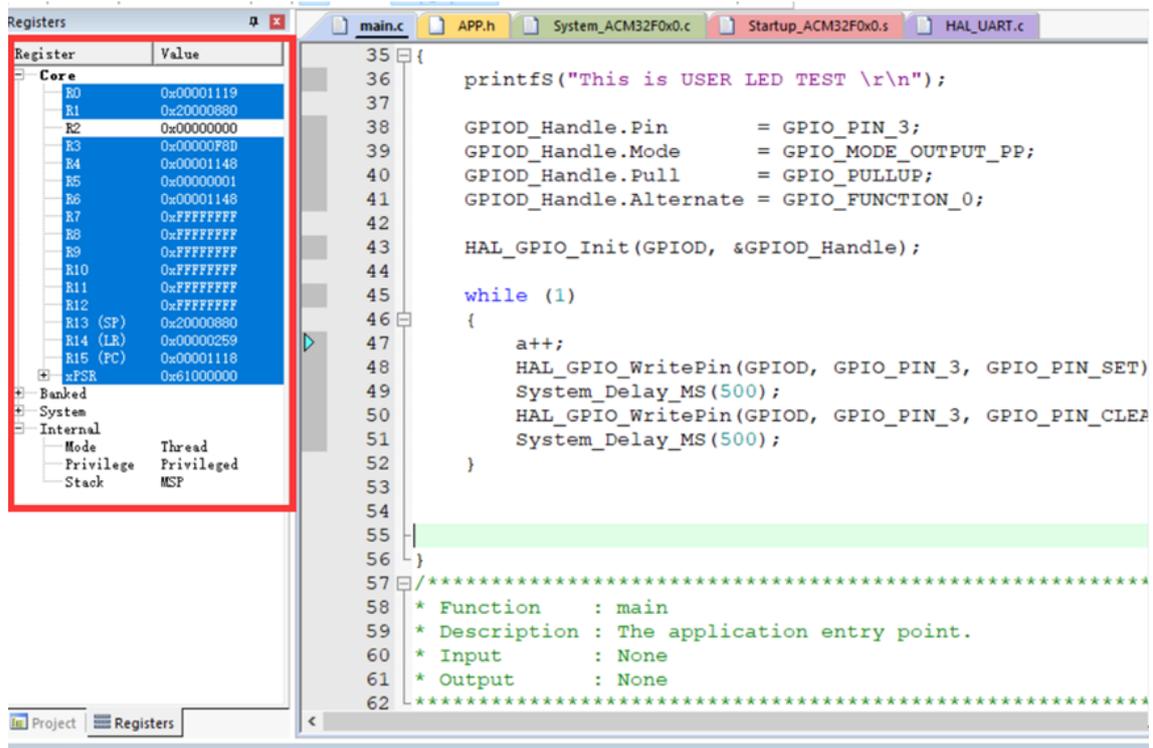


2 选中需要查看的参数，单击鼠标右键，可以将参数放入观察窗口，实时查看参数的变化



3 内核寄存器组

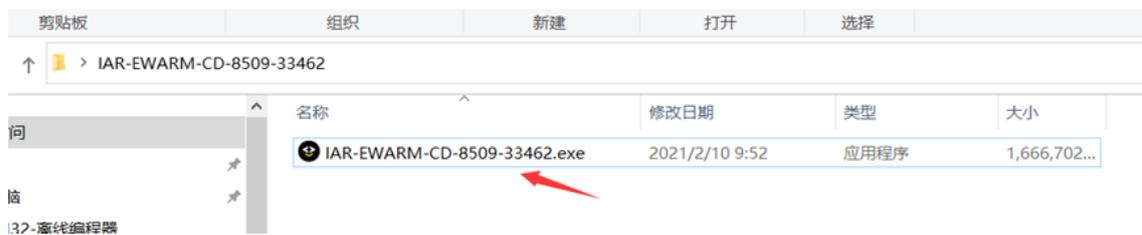
寄存器窗口，如下图所示，该窗口用于显示 R0~R15、xPSR 等内核寄存器的值（不是外设寄存器），关于内核寄存器的介绍可以查看 [ARM 架构基本寄存器](#)。

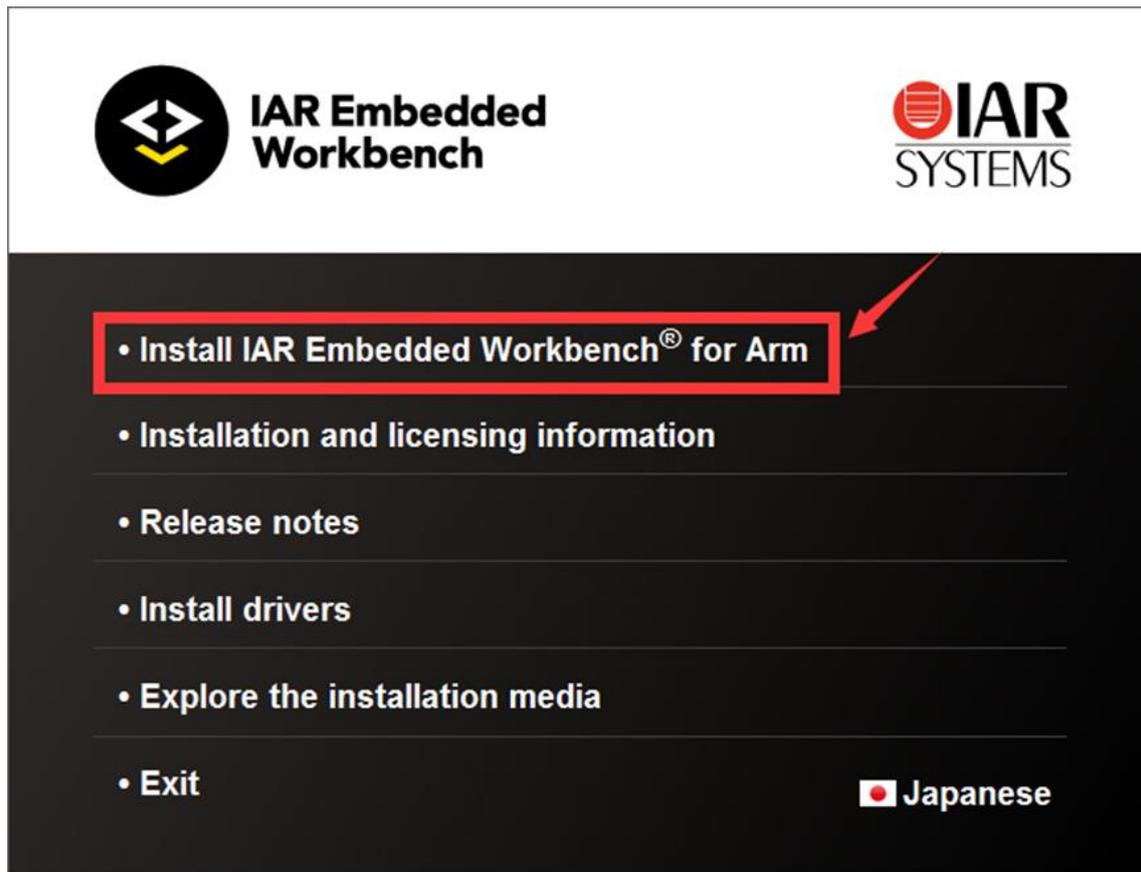


IAR 篇

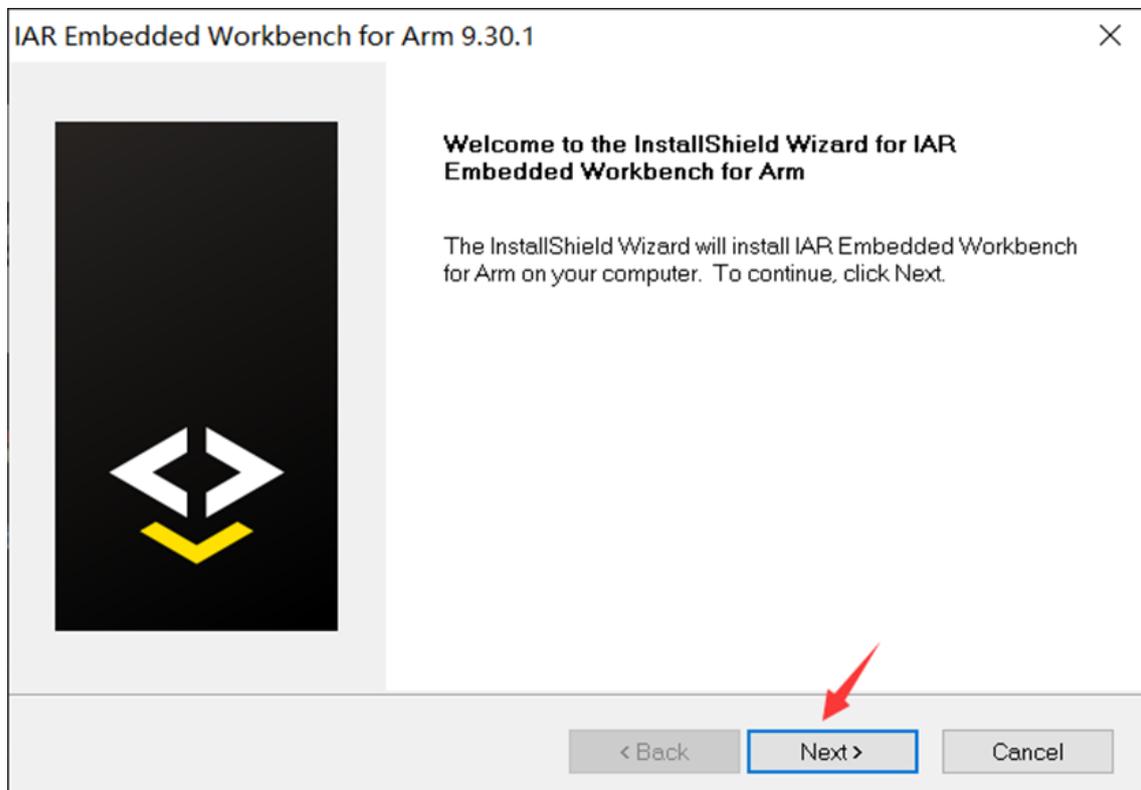
1.1 IAR 安装步骤

一、下载并解压安装包，并按步骤完成安装

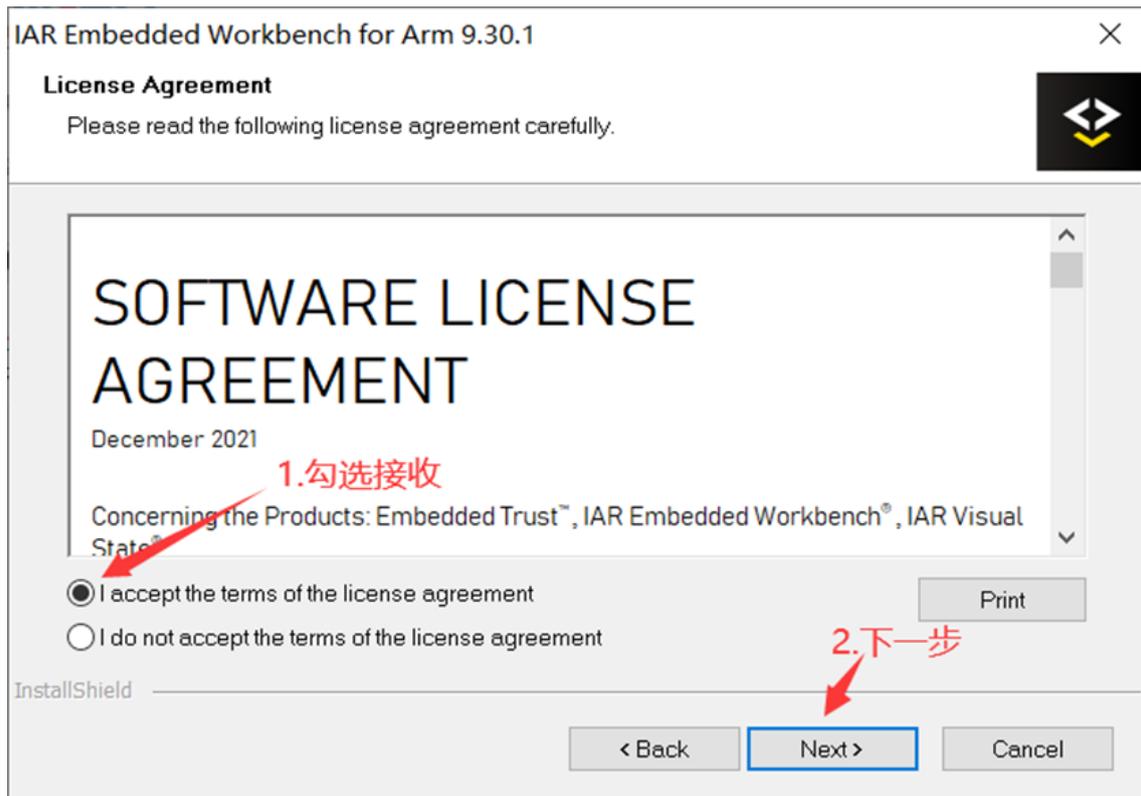




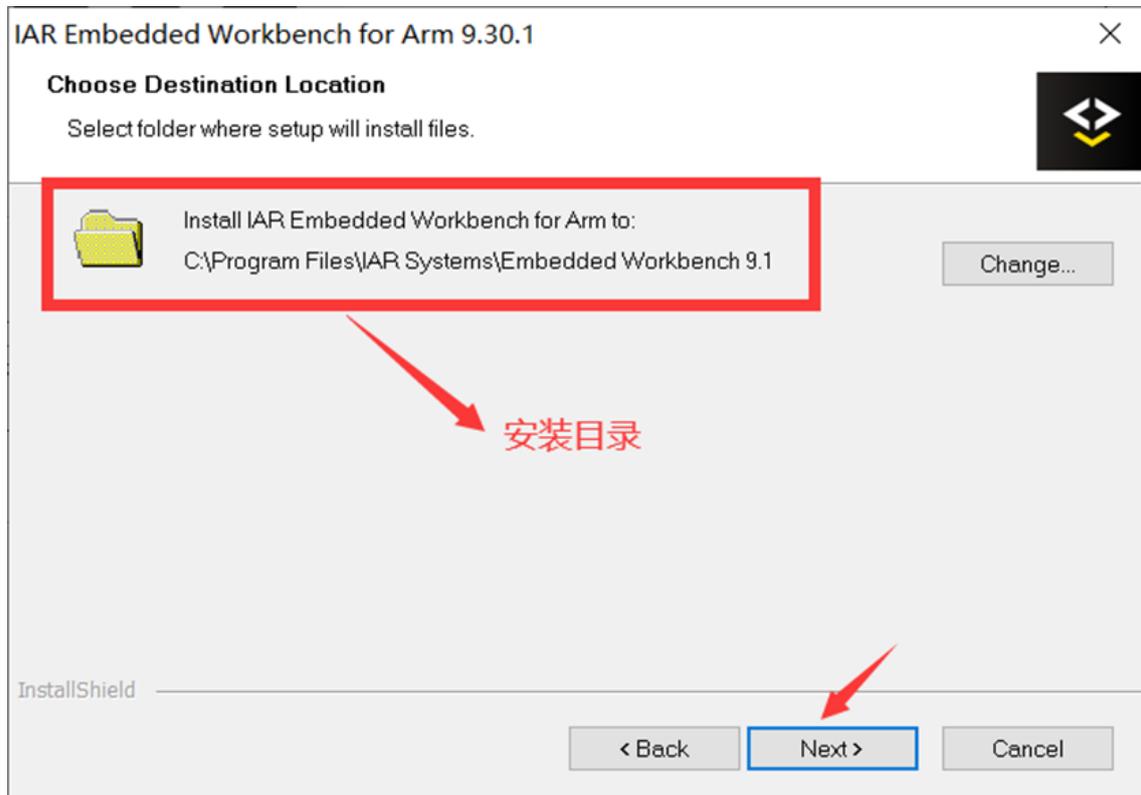
二、运行安装程序，点击 next



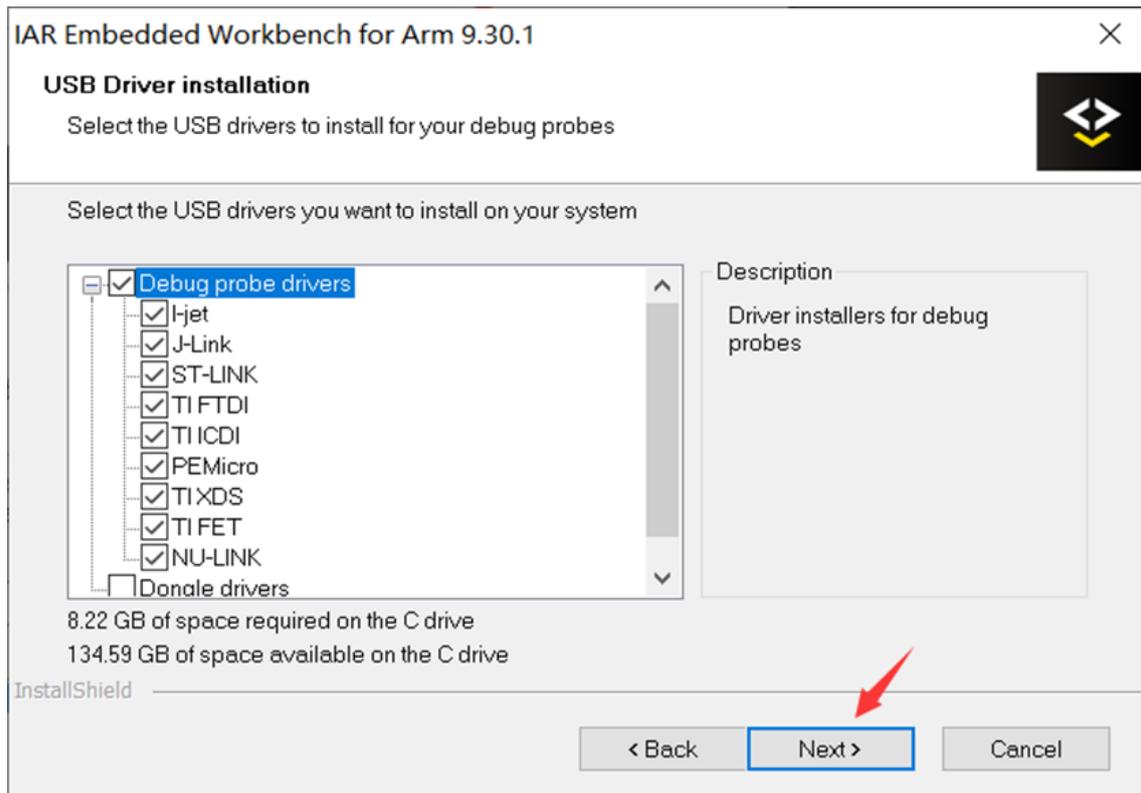
三、勾选 accept，点击 next



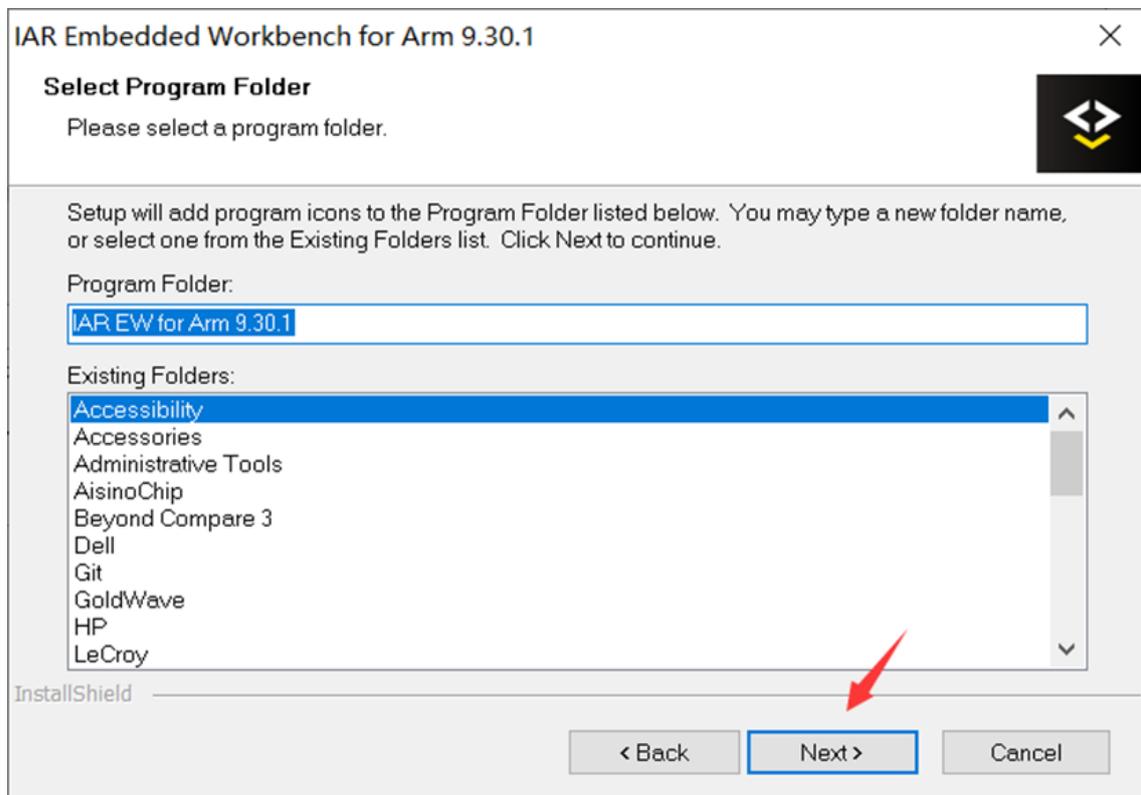
四、选择安装路径，点击 next



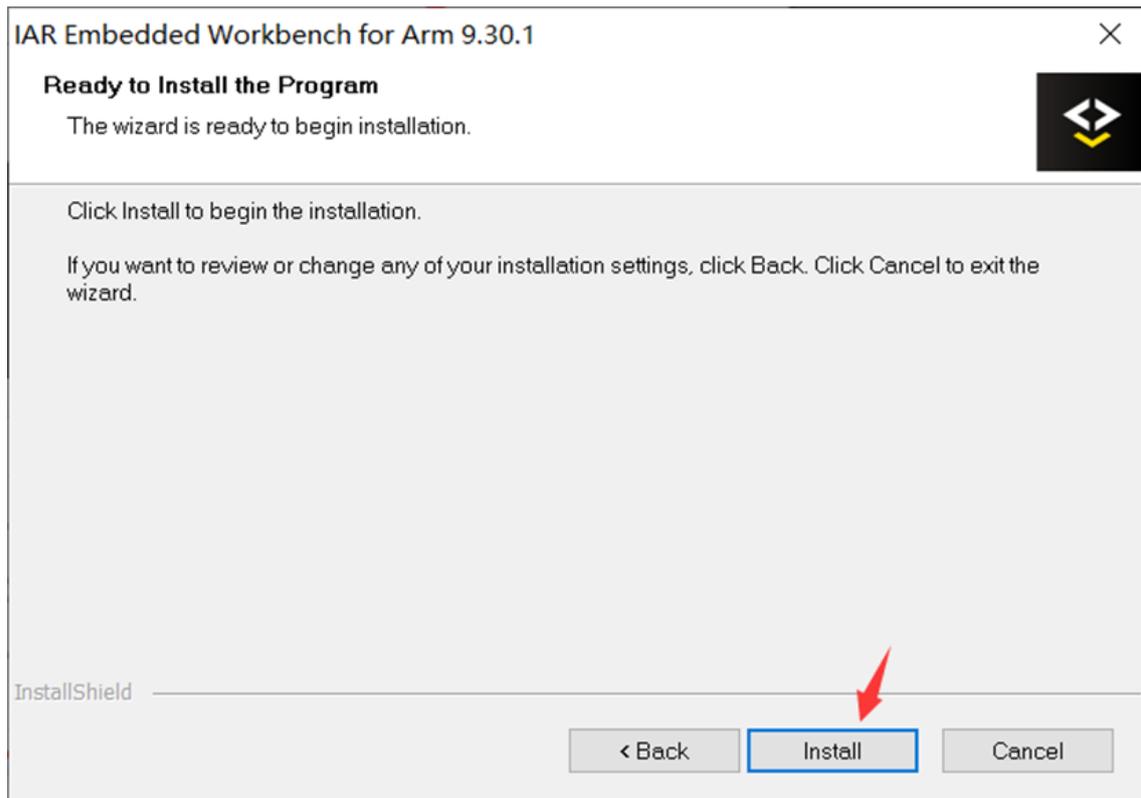
五、安装驱动，为避免以后可能使用到其他下载器，默认即可，会自动安装一些驱动



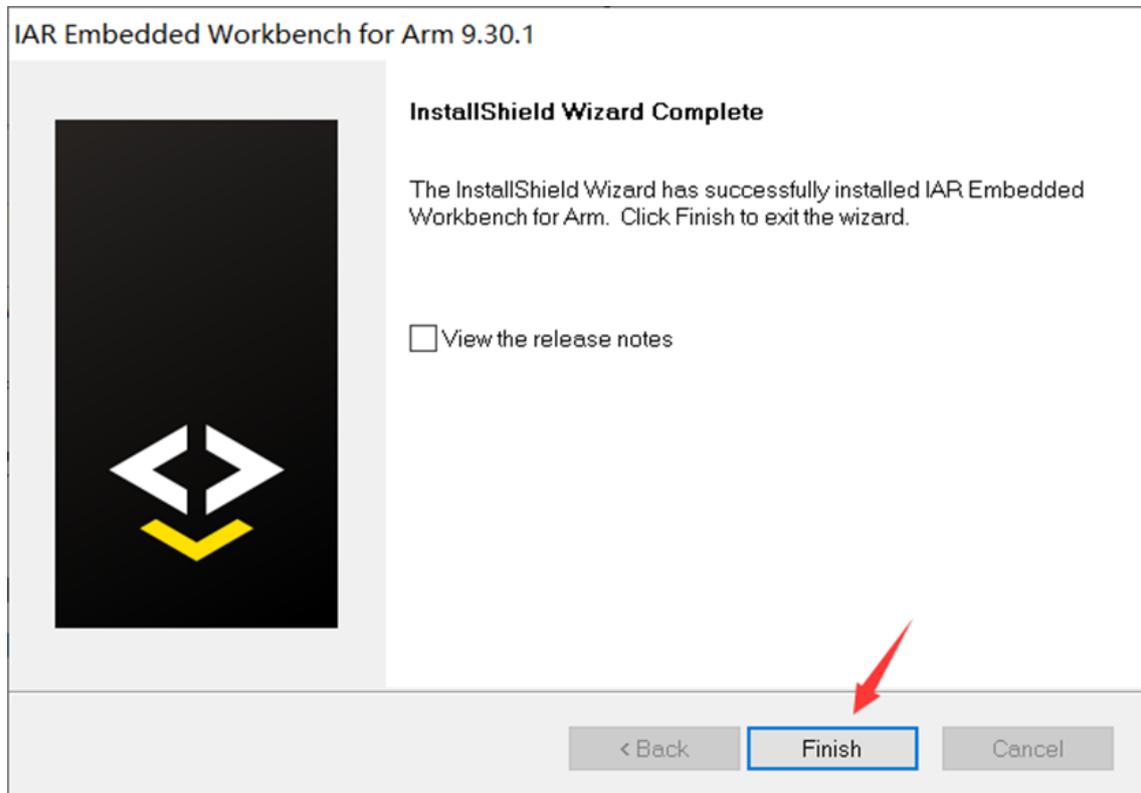
六、直接选择"NEXT"



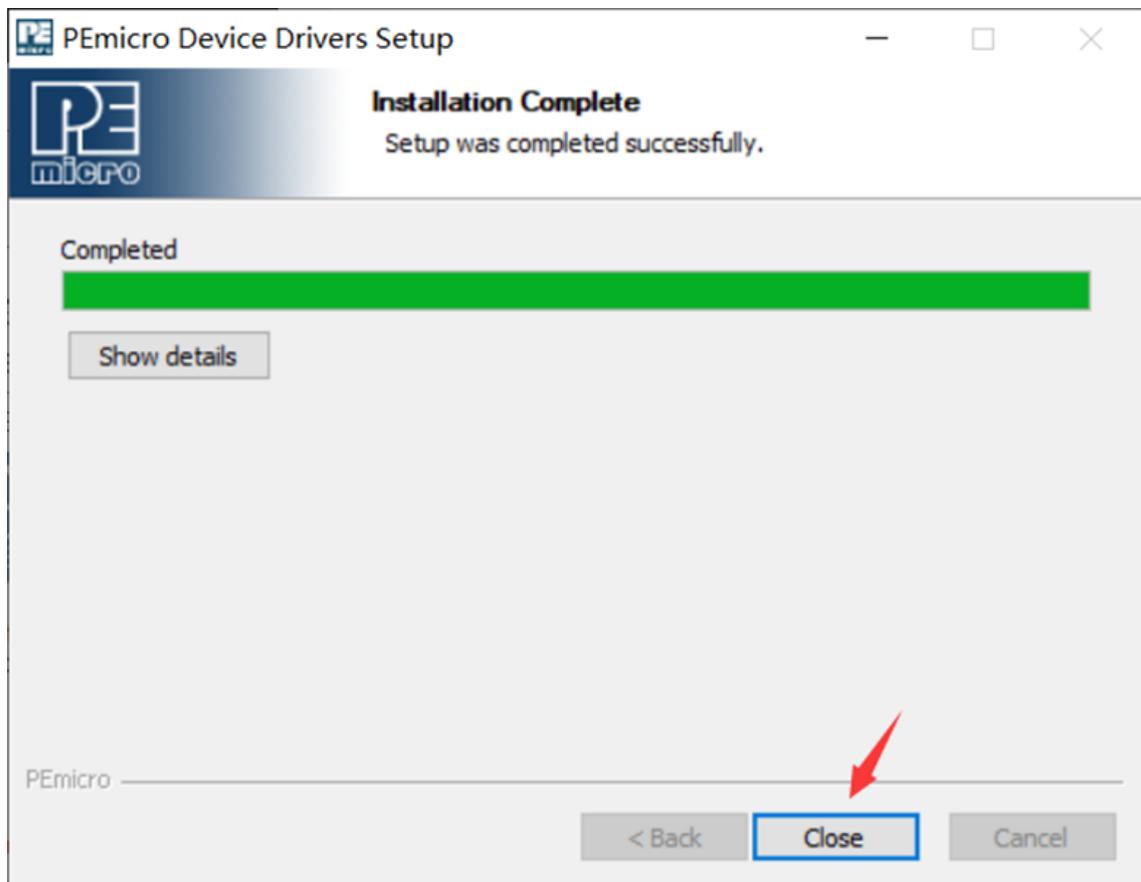
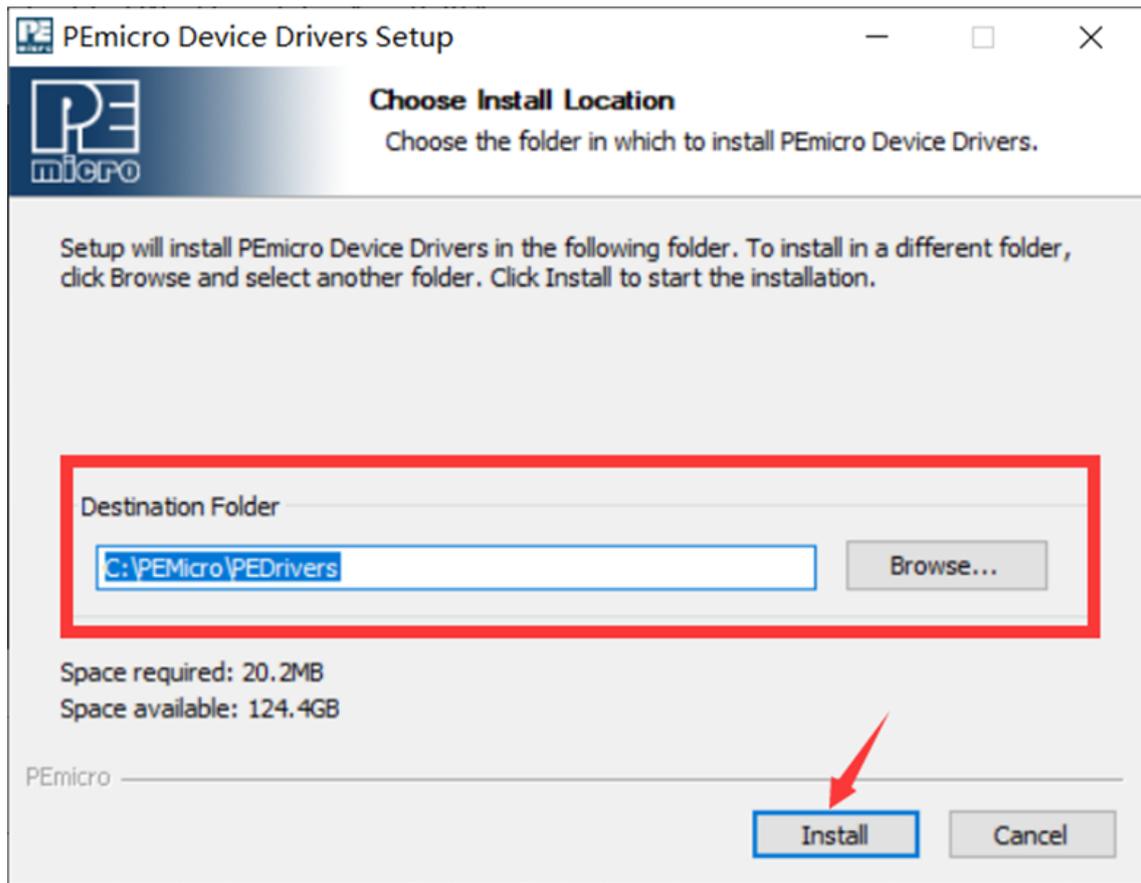
七、点击"install"



八、等待安装完成， 点击"finish"完成安装

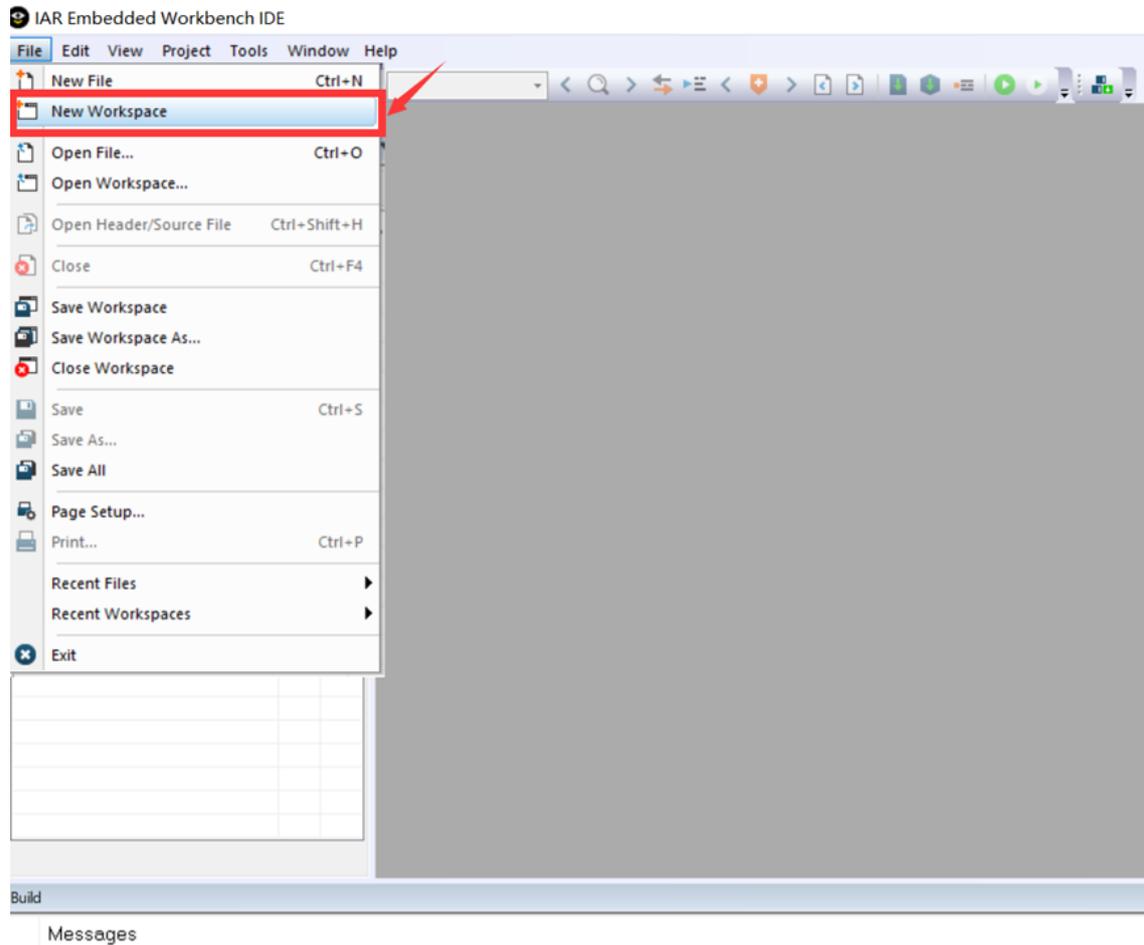


九、安装完会出现弹窗，是第 7 条选择的驱动的安装，每个弹窗直接点击 "Install","close"即可，所有驱动按照默认安装即可。



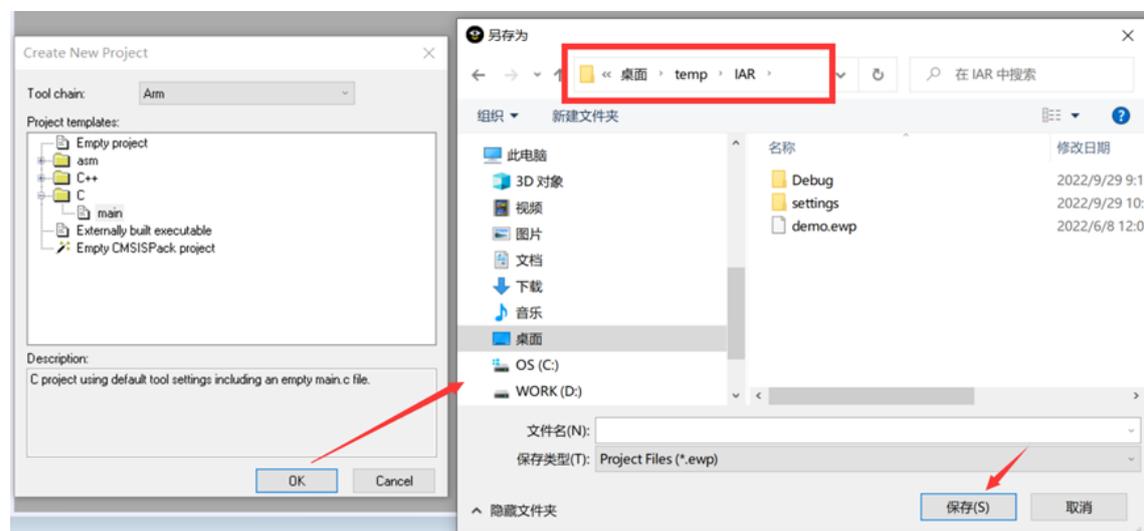
1.2 IAR 使用流程（以 ACM32G1XX 为例）

一、建立新工作区（File->>New Workspace）



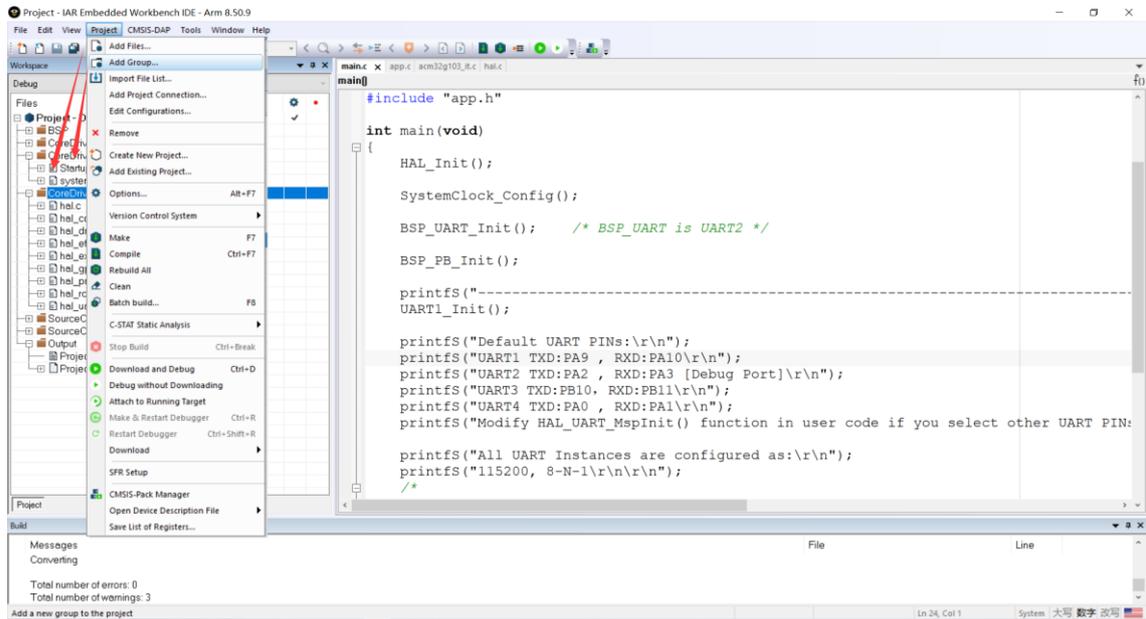
二、建立新项目（Project->>Create New Project...）

选择对应的工程模板后点击“OK”，之后将工程保存在对应的文件夹。



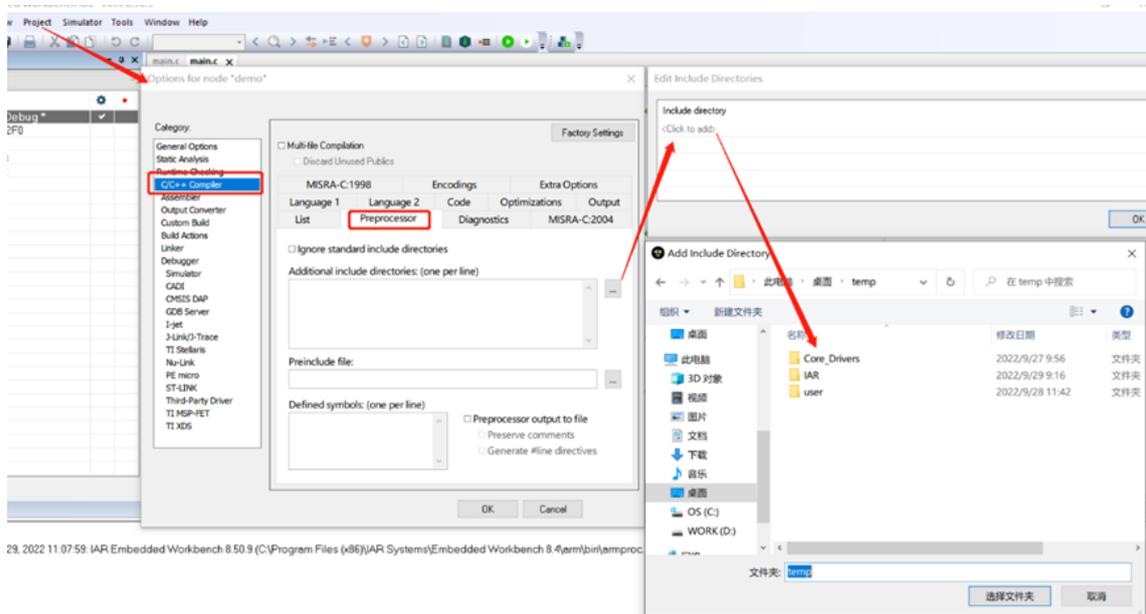
三、添加文件组和源文件

通过 Add Files 可以添加自己编写好的源文件 (.C) 或者别的 demo 文件中的源文件, 通过 Add Group 可以添加文件组, 来给每个文件分组。



四、添加头文件

点击 Project->>Options, 选择 C/C++ Compiler->>Preprocessor 添加头文件的路径。



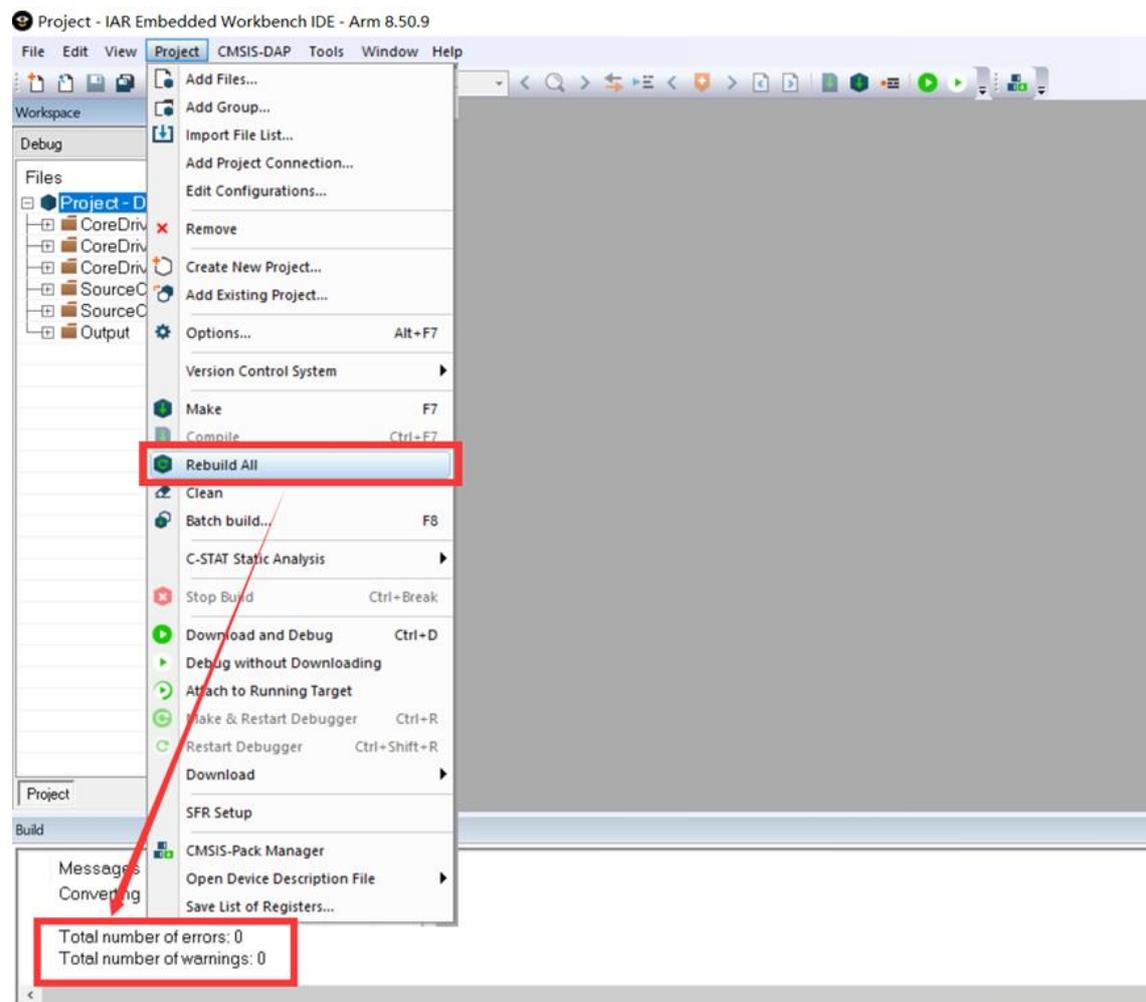
1.3 IAR 编译、下载、运行

一、编译

Compile: 只对目前打开的 c 文件进行语法检查, 不对它进行 link; Make: 对工程所有打开的 c 文件进行语法检查和 link (只编译有改动的文件或者设置变动的文件);

Rebuild All: 编译链接当前工程 (不管文件或者设置是否有变动);

Clean: 清除当前工程的编译状态; Batch build: 批量编译, 打开 Batch build 批量编译后, 需要给批量编译起个名字, 然后将本工程中的三个分类都添加到 “Configurations to build” 中, 然后单击 “Make” 就可以对这工程中的三个分类全部进行编译;



二、下载

下载程序需要先安装调试文件

名称	修改日期	类型	大小
ACM32G1XX.ddf	2022/12/12 15:41	DDF 文件	2 KB
Flash_ACM32G1XX.mac	2022/12/12 15:41	MAC 文件	1 KB
FlashACM32G1XX.board	2022/12/12 15:41	BOARD 文件	1 KB
FlashACM32G1XX.flash	2022/12/12 15:41	FLASH 文件	1 KB
FlashACM32G1XX.out	2022/12/12 15:41	OUT 文件	20 KB

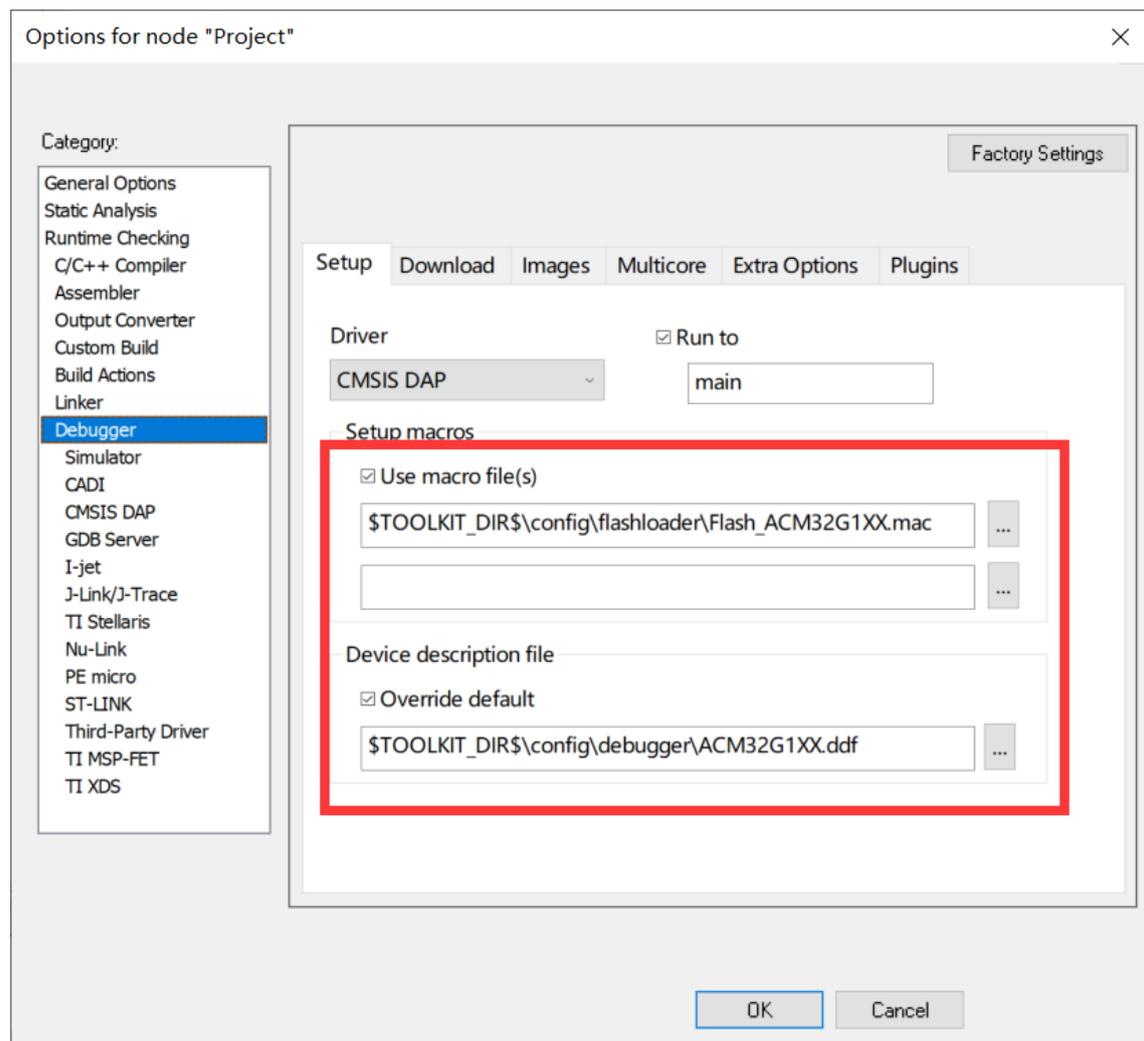
- 将 ACM32G1XX.ddf 放置于 IAR 的安装目录的\arm\config\debugger 下

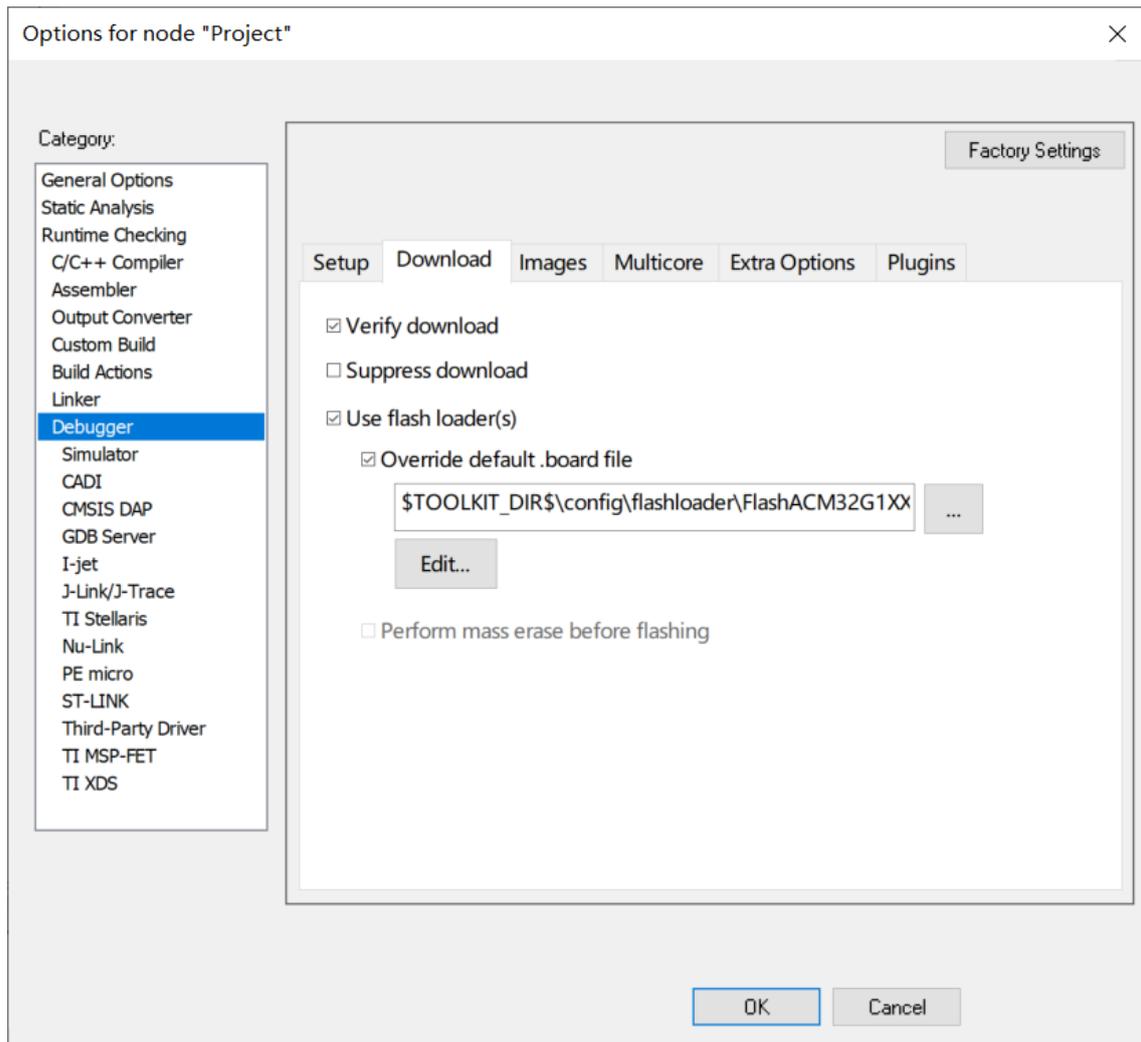
名称	修改日期	类型	大小
NetSilicon	2022/9/29 10:43	文件夹	
NordicSemi	2022/9/29 10:43	文件夹	
Nuvoton	2022/9/29 10:43	文件夹	
NXP	2022/9/29 10:43	文件夹	
ON Semiconductor	2022/9/29 10:43	文件夹	
Qorvo	2022/9/29 10:43	文件夹	
Redpine	2022/9/29 10:43	文件夹	
Renesas	2022/9/29 10:43	文件夹	
Samsung	2022/9/29 10:43	文件夹	
SiliconLaboratories	2022/9/29 10:43	文件夹	
Socle	2022/9/29 10:43	文件夹	
Sonix	2022/9/29 10:43	文件夹	
ST	2022/9/29 10:41	文件夹	
TexasInstruments	2022/9/29 10:43	文件夹	
Toshiba	2022/9/29 10:44	文件夹	
Vorago	2022/9/29 10:44	文件夹	
Xilinx	2022/9/29 10:44	文件夹	
ACM32F0XX.ddf	2022/2/16 10:04	DDF 文件	2 KB
ACM32E4XX.ddf	2021/5/20 11:15	DDF 文件	2 KB
ACM32G1XX.ddf	2022/12/12 15:41	DDF 文件	2 KB

- 将其余 4 个文件放置于 IAR 的安装目录的\arm\config\flashloader 下

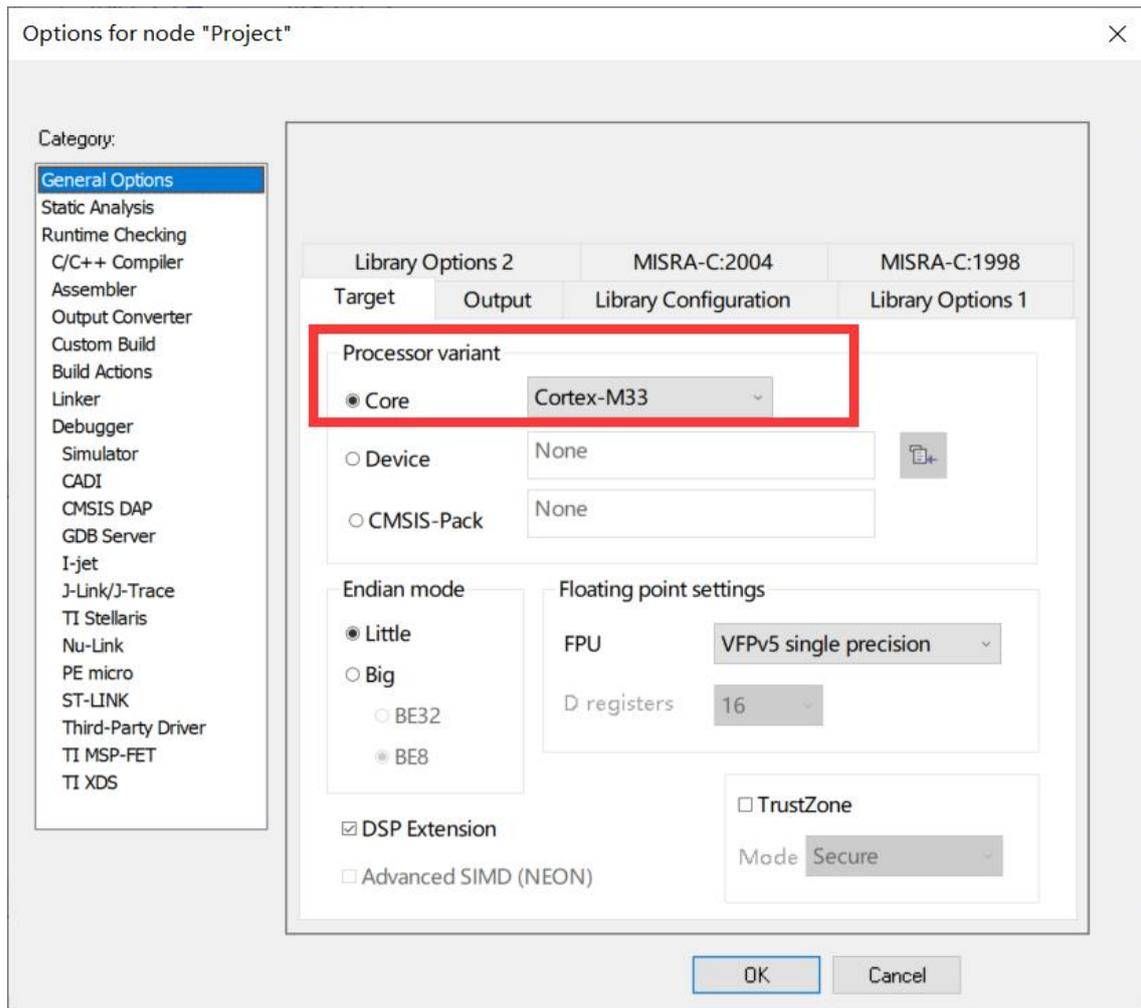
名称	修改日期	类型	大小
NXP	2022/9/29 10:43	文件夹	
ON Semiconductor	2022/9/29 10:43	文件夹	
Qorvo	2022/9/29 10:43	文件夹	
Redpine	2022/9/29 10:43	文件夹	
Renesas	2022/9/29 10:43	文件夹	
Samsung	2022/9/29 10:43	文件夹	
Silicon Laboratories	2022/9/29 10:43	文件夹	
Sonix	2022/9/29 10:43	文件夹	
ST	2022/9/29 10:42	文件夹	
Texas Instruments	2022/9/29 10:44	文件夹	
Toshiba	2022/9/29 10:44	文件夹	
Vorago	2022/9/29 10:44	文件夹	
Flash_ACM32F0XX.mac	2022/2/7 17:39	MAC 文件	1 KB
Flash_ACM32F4XX.mac	2021/5/21 9:20	MAC 文件	1 KB
Flash_ACM32G1XX.mac	2022/12/12 15:41	MAC 文件	1 KB
FlashACM32F0XX.board	2022/2/16 10:02	BOARD 文件	1 KB
FlashACM32F0XX.flash	2022/2/16 10:05	FLASH 文件	1 KB
FlashACM32F0XX.out	2022/2/16 13:35	OUT 文件	21 KB
FlashACM32F4XX.board	2021/5/20 9:05	BOARD 文件	1 KB
FlashACM32F4XX.flash	2021/5/20 11:06	FLASH 文件	1 KB

在目录中放置好后需要在 IAR 中配置调试文件

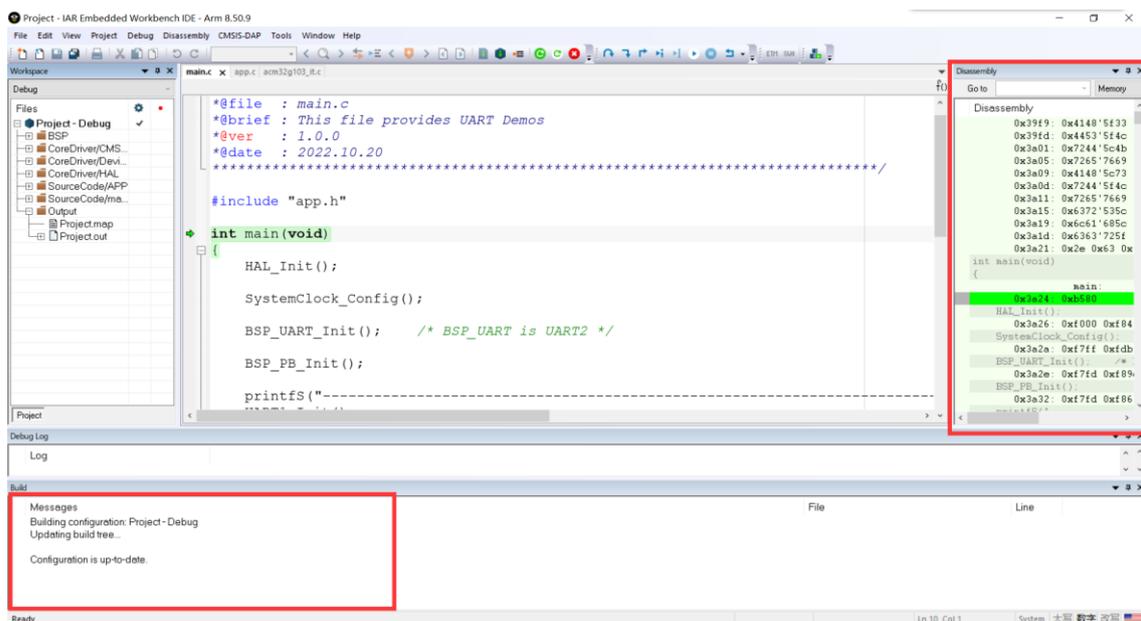




内核选择 Cortex-M33，使用 CMSIS-DAP 驱动烧录。ACM32G103 同样支持 JLINK 烧录，详细步骤参考 ACM32F4 系列开发环境 IAR 篇。

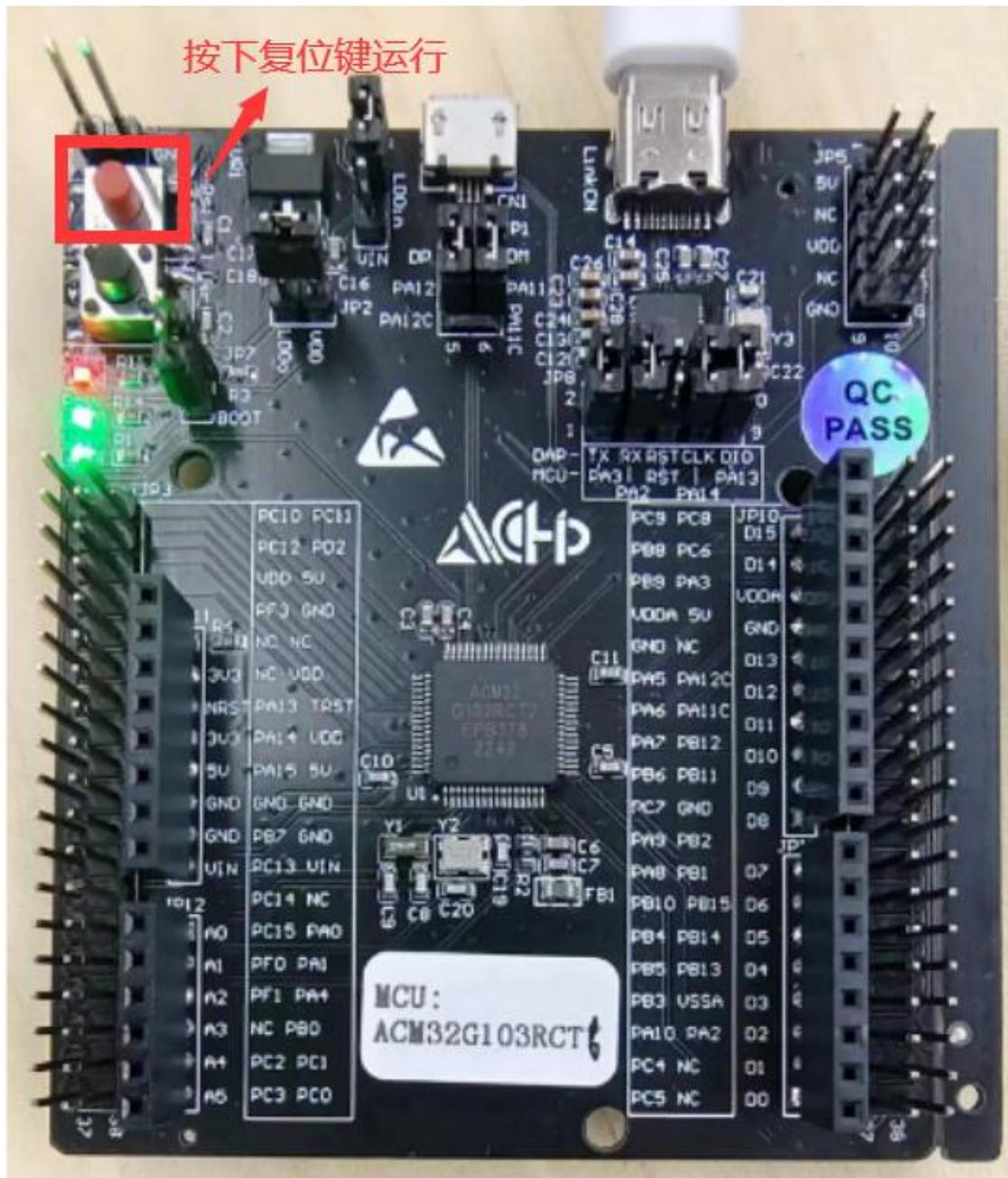


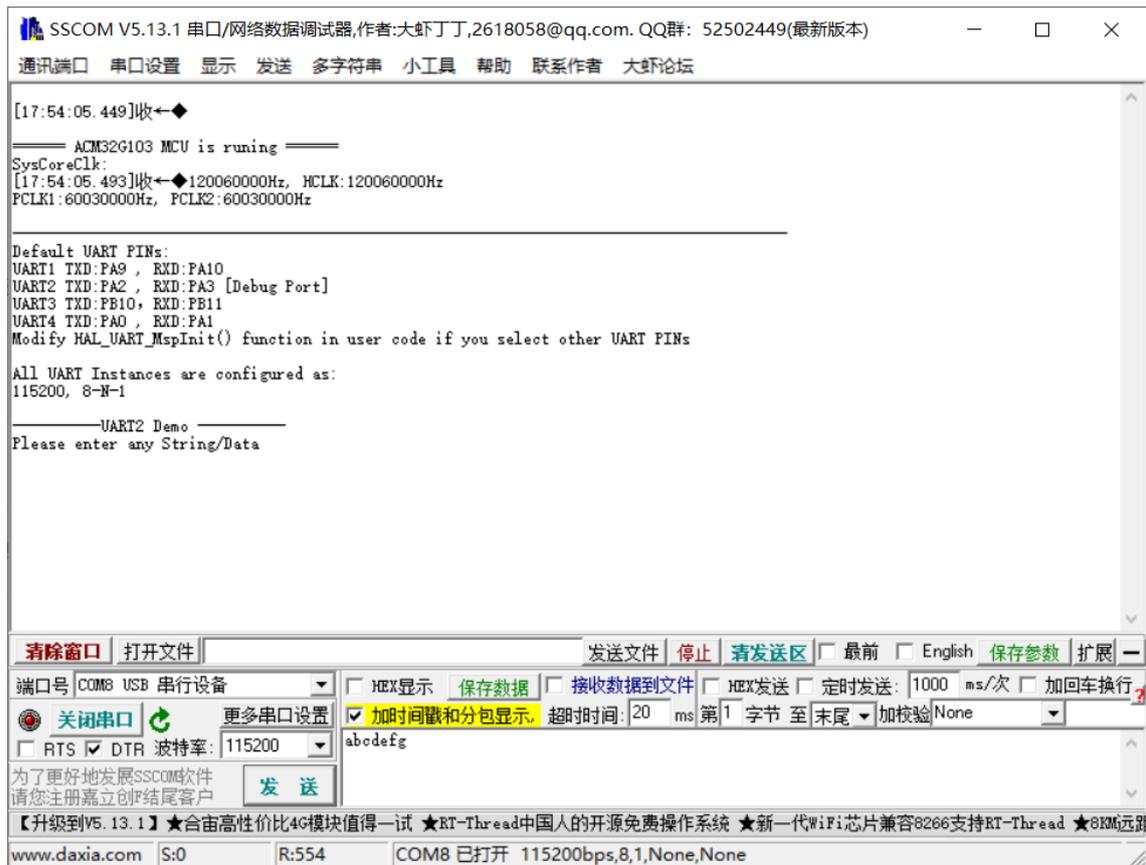
上述设置完成后点击 Download and DeBUG,烧录成功



三、运行（目前芯片不支持下载后自启动，需要按下 RESET 键后，程序才能运行），以 UART_Printf_Demo 为例，部分示例代码（完整代码见 ACM32G103 的 SDK 库）

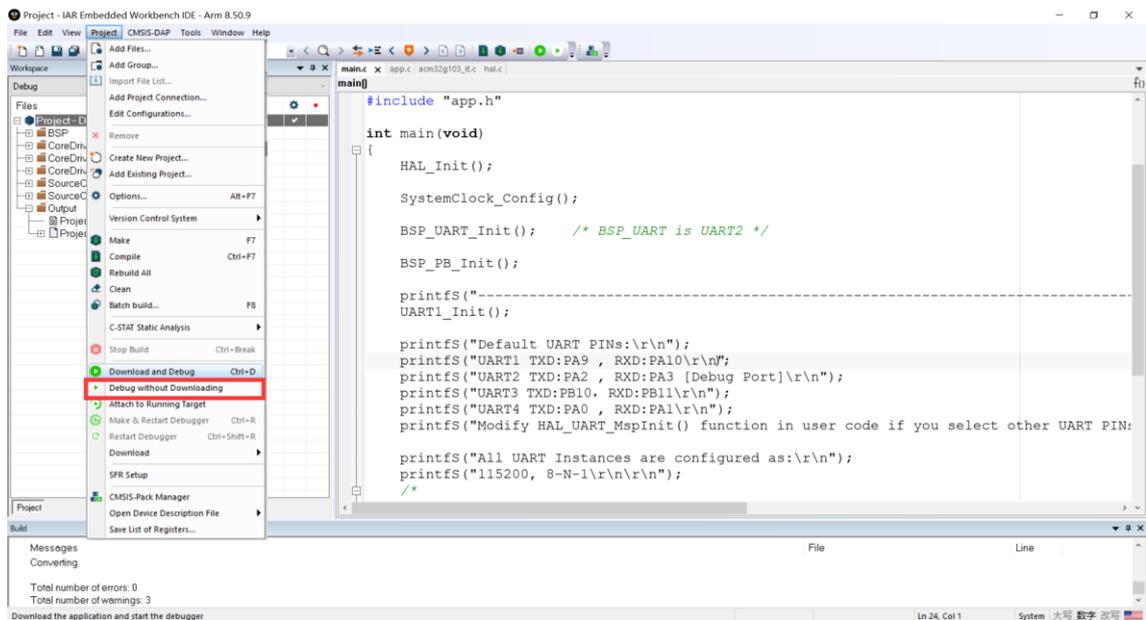
和现象如下：



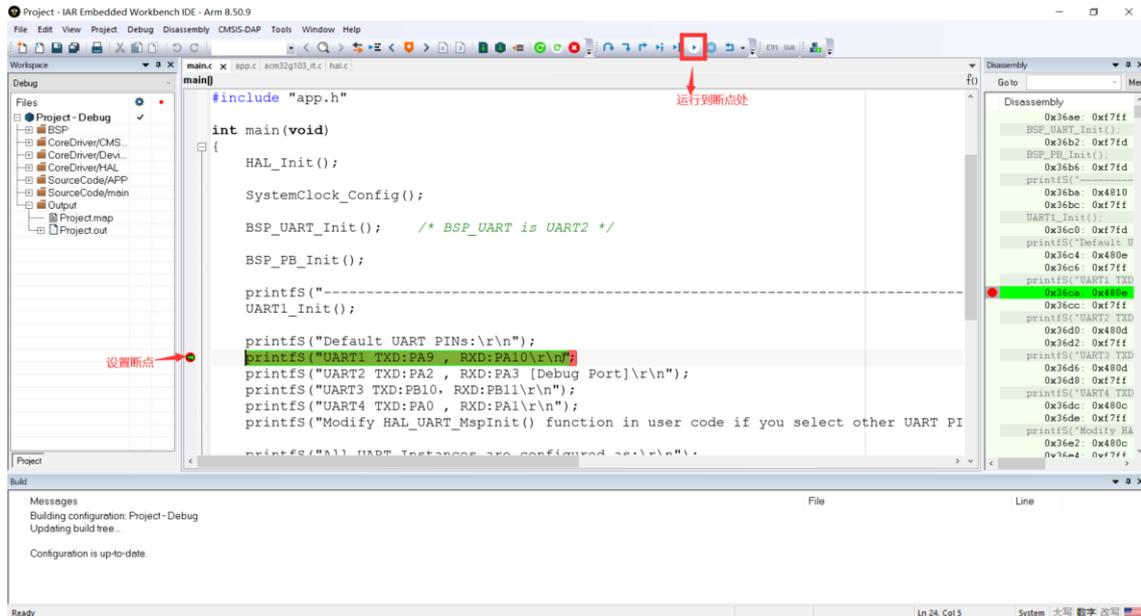


1.4 IAR Debug 使用说明

一、点击白色按键或者通过 Project->>Debug without Download 进入仿真调试界面



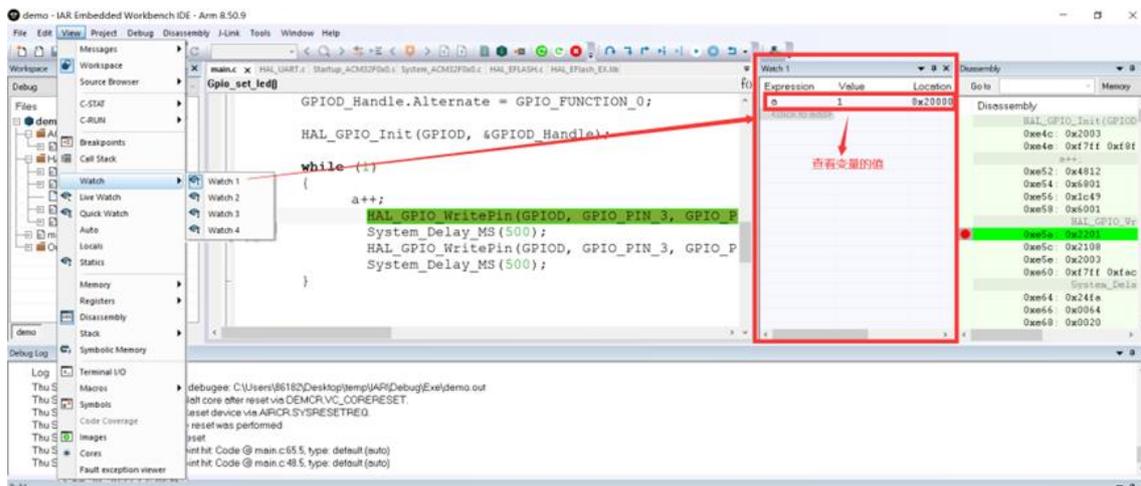
二、设置断点



按 F5 键或工具条上的 Go 按钮都可以让程序执行到断点。Debug Log 窗口将显示关于断点的信息

三、查看变量 (Watch 窗口),

打开 Live Watch 窗口的方法是选择主菜单 View → Watch 命令。Watch 窗口用于观察静止位置上的变量，如全局变量。点击鼠标右键选择要查看的变量，点击 Add to Watch;变量的值在执行时会变化并显示出来



四、监视寄存器

选择主菜单 View > Register 打开寄存器窗口，显示的是 CPU 寄存器。可以从寄存器窗口左上方的下拉菜单中选择需要查看的任何寄存器组。

The screenshot displays the IAR Embedded Workbench IDE interface. The main window shows a code editor with the following C code snippet:

```
* @date  
* @author  
* @brief  
*****  
*/  
#include "AP  
int a=0;  
UART_HandleT  
GPIO_InitTyp  
/*****  
* function  
* Descripti
```

The 'Registers 1' window is open, showing a list of registers and their values. A red box highlights this window, and a red arrow points to the 'Registers 1' entry in the 'Registers' menu. The registers listed are:

Name	Value
R7	0xffff
R8	0xffff
R9	0xffff
R10	0xffff
R11	0xffff
R12	0x0000
APSR	0x0000
IPSR	0x0000
EPSR	0x0100
PC	0x0000
SP	0x2000
LR	0x0000
PRIMASK	0x0000
CONTROL	0x0000
CYCLECOUNTER	0
CCTIMER1	0
CCTIMER2	0
CCSTEP	0

The 'Debug Log' window at the bottom shows the following messages:

```
reset  
int hit: Code @ main.c:65.5, type: default (auto)  
int hit: Code @ main.c:48.5, type: default (auto)  
must have a name  
must have a name  
must have a name  
must have a name
```