开发环境学习手册

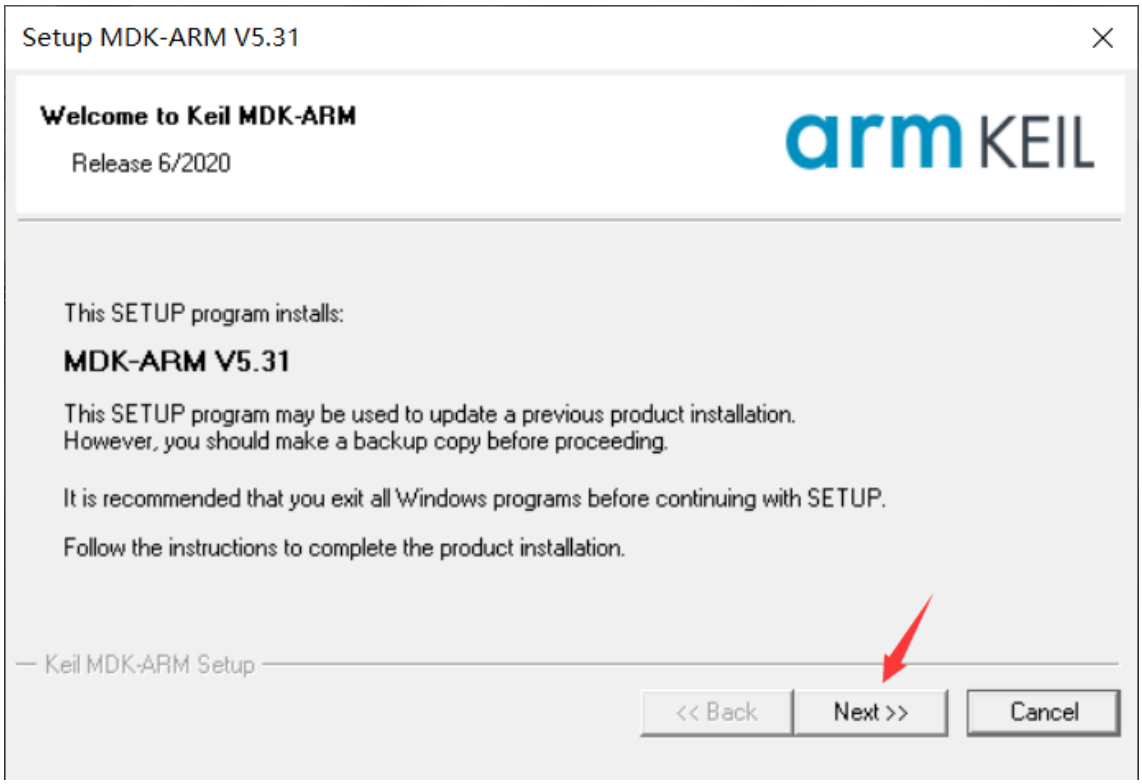
Keil 篇

1.1 Keil 安装步骤

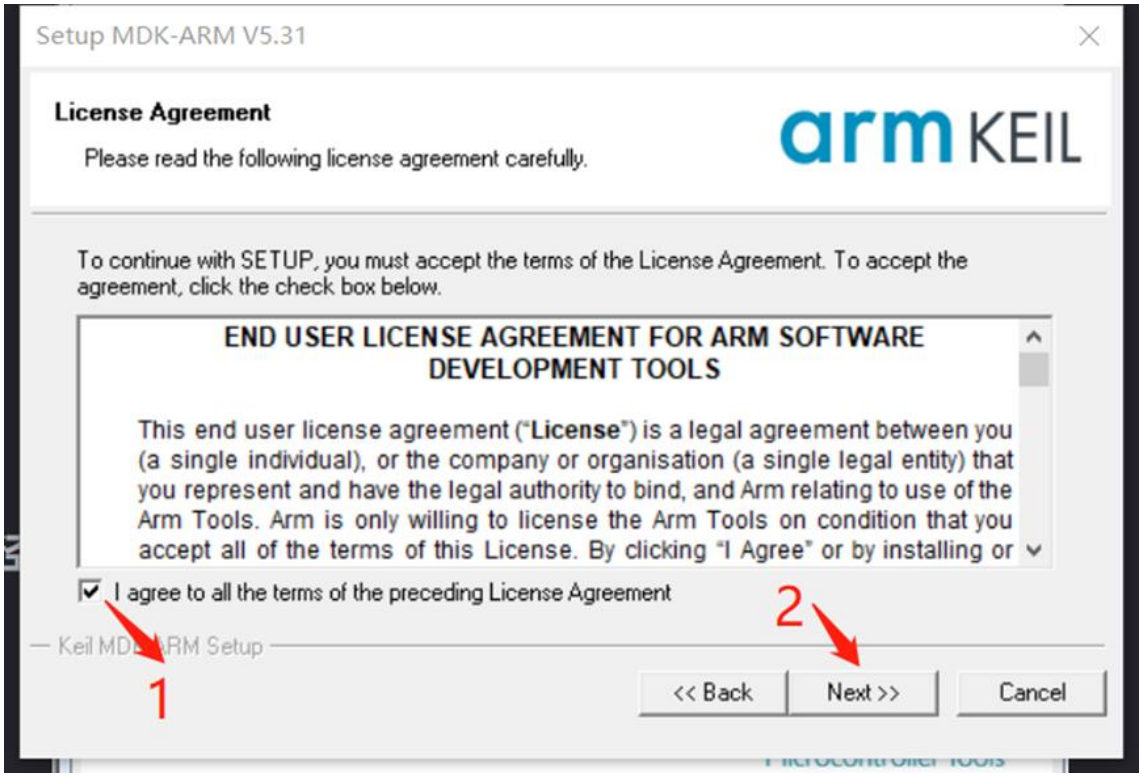
安装包链接：https://cloud.tencent.com/developer/article/1605195 （keil 推荐在 MDK5.31（含）以上）。

一、下载并解压安装包，并按步骤完成安装

二、运行安装程序，点击 next



三、勾选 accept，点击 next

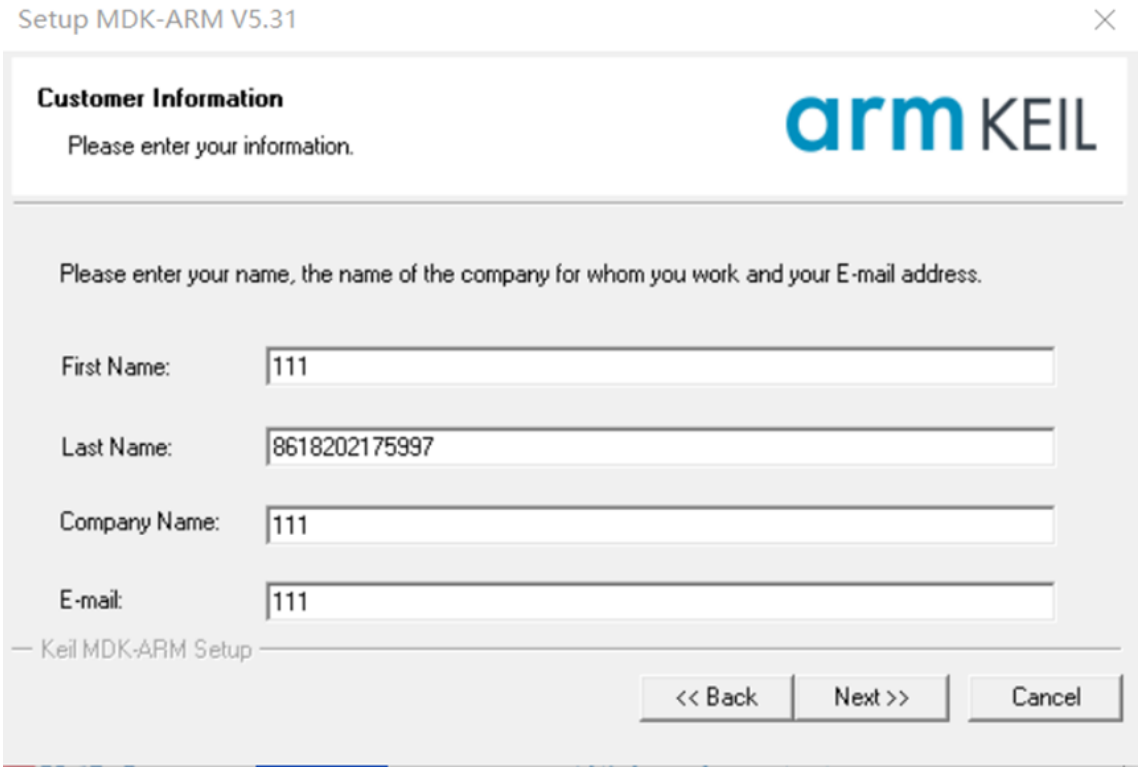


四、选择安装路径，点击 next（最好选择默认路径，如果自行修改需要注意不要添加中文）

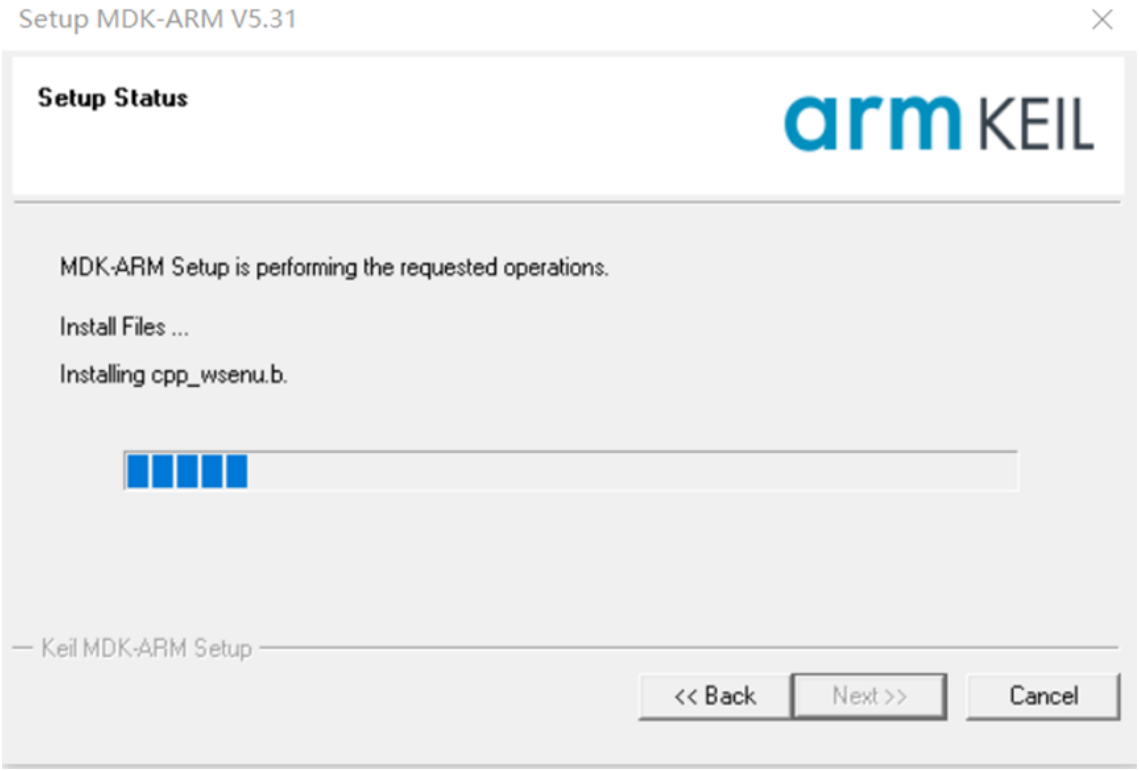
注意事项：

- 安装路径不能有中文
- 安装目合

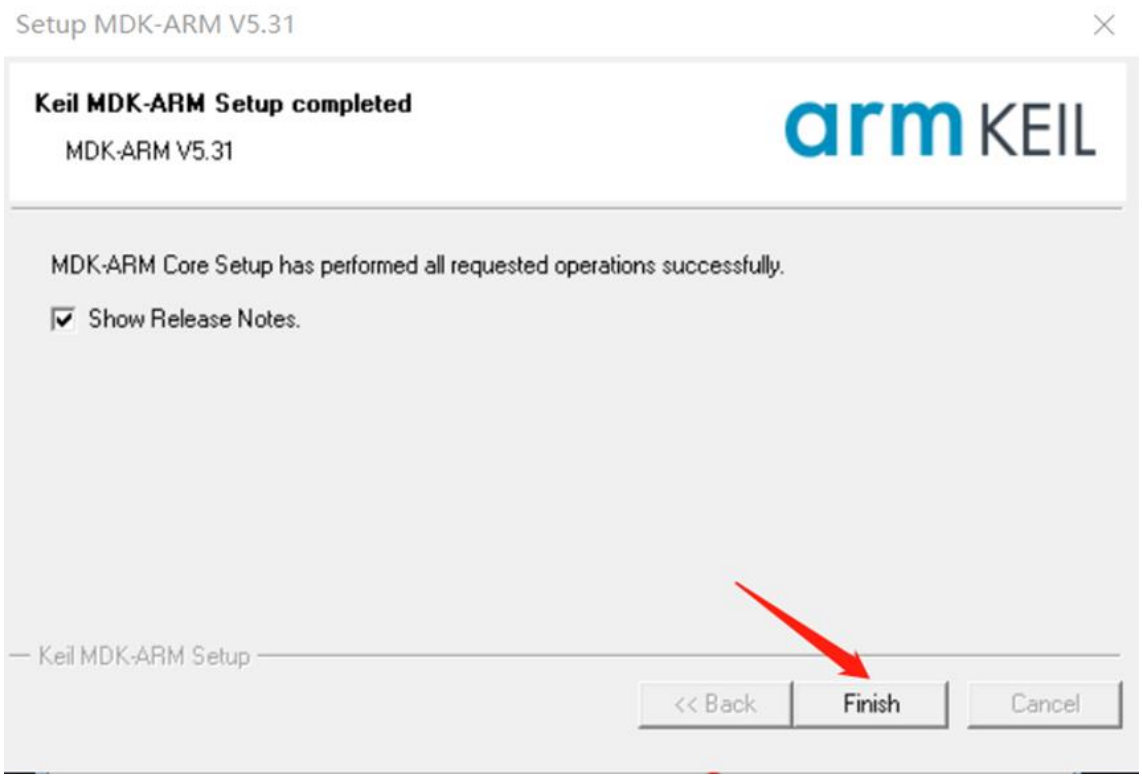
五、信息随意填写，点击 next



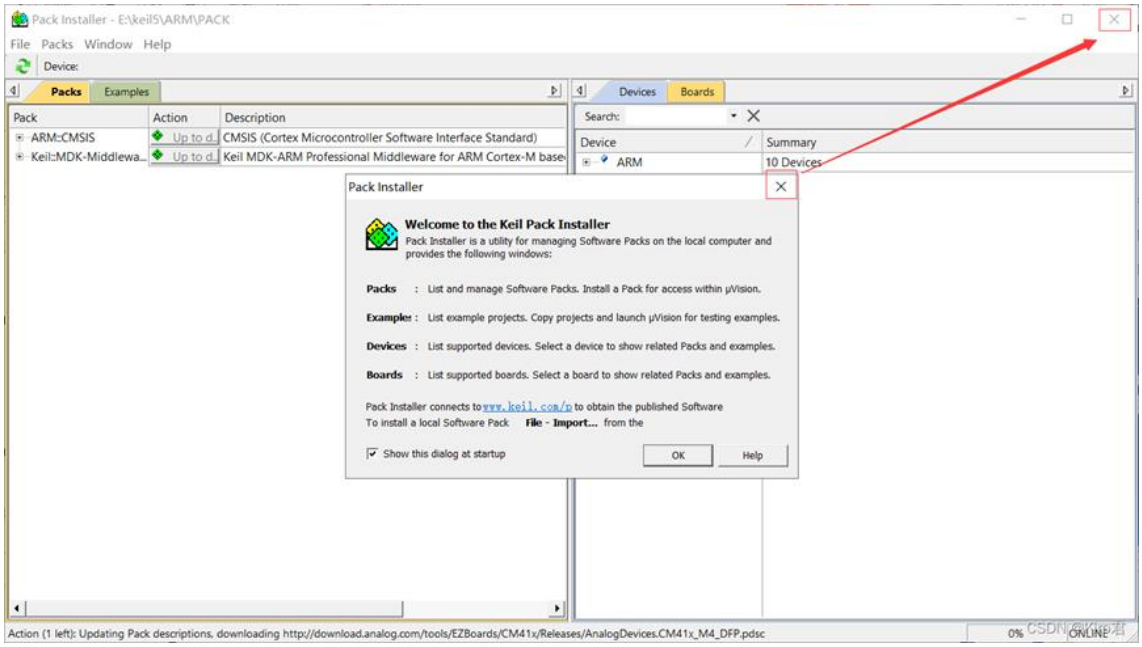
六、等待安装



七、点击 finish，完成安装



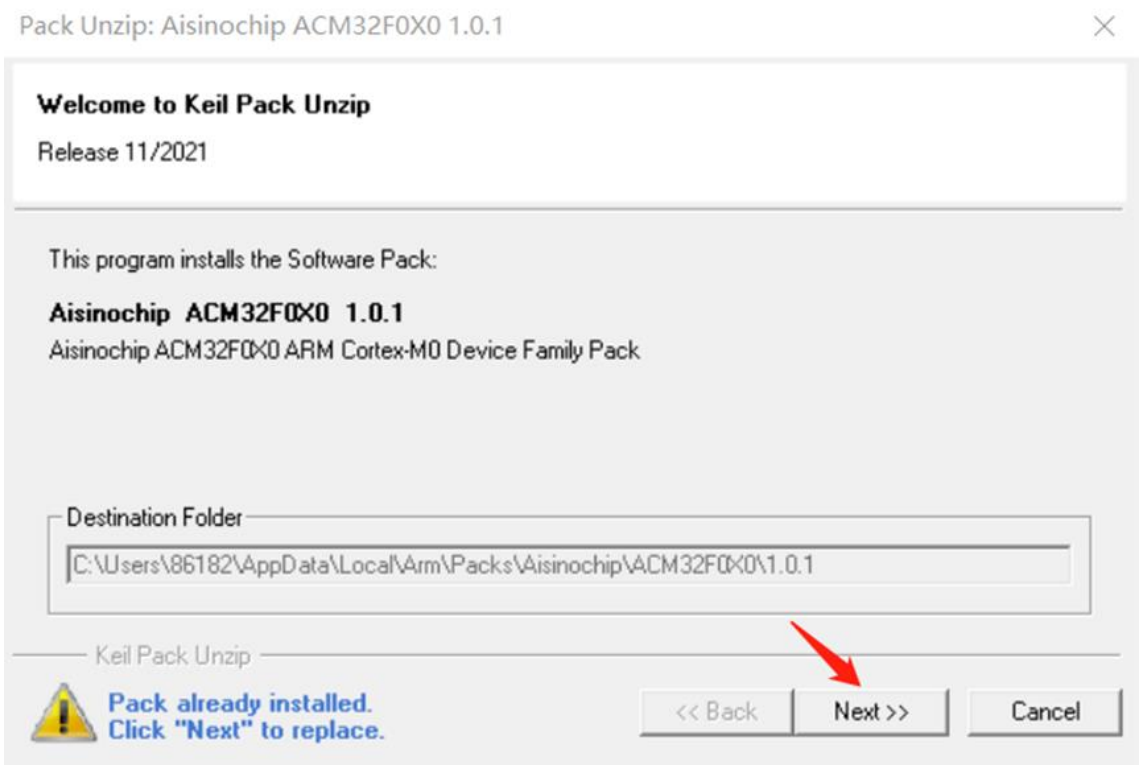
八、突然会弹出“Pack installer”页面，先后关闭两个窗口

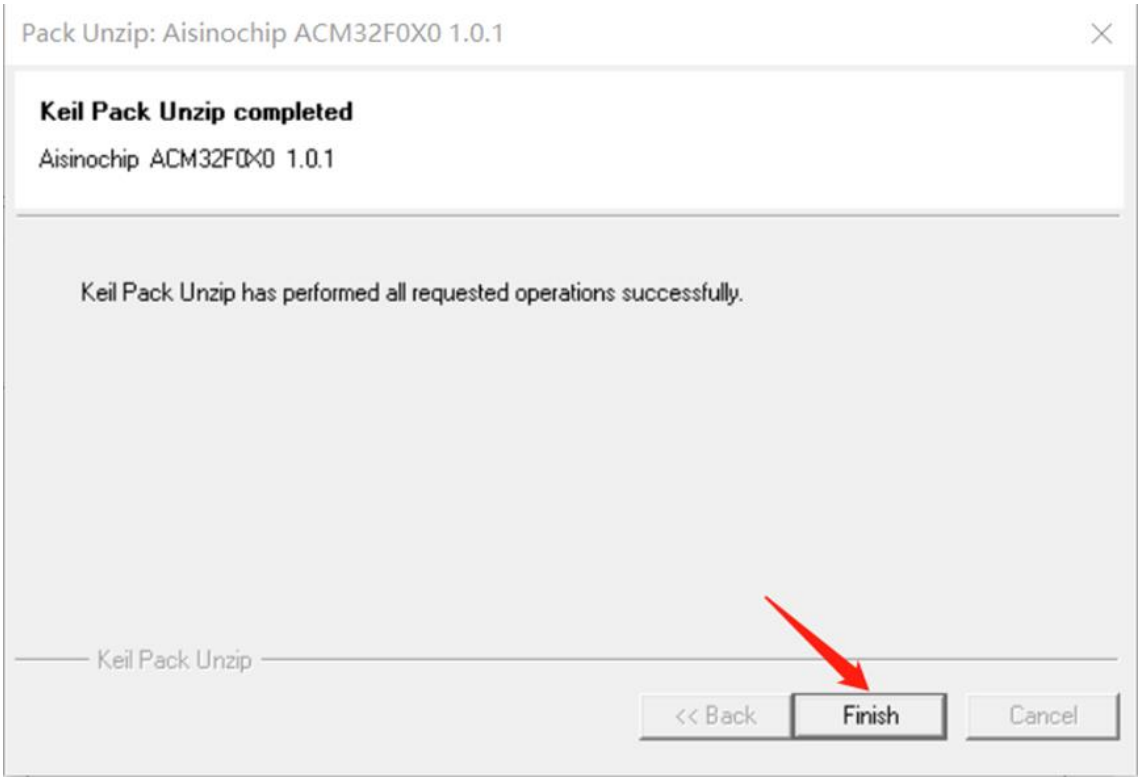


1.2Keil 使用流程（以 ACM32F0X0 系列为例）

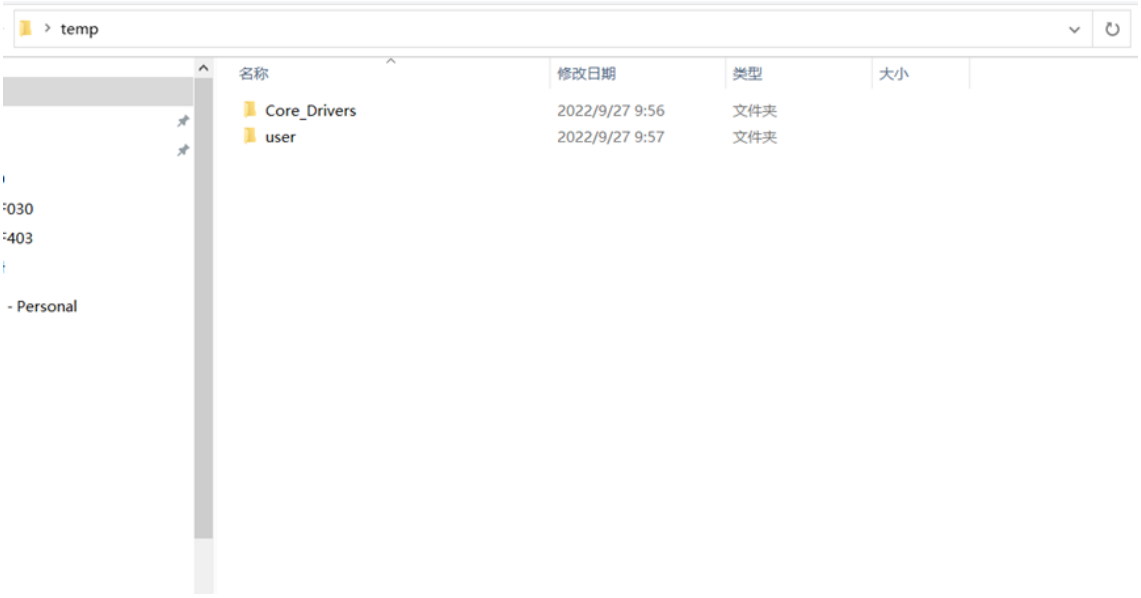
一、 安装航芯的芯片包


此电脑 › WORK (D:) › F0code › ACM32F0x0_SDK_v2.0.2 › 固件库 › 固件库 ›
名称
修改日期
类型
大小
ModulesDemo_Rev2.0.2
2022/9/13 16:11
文件夹
Aisinochip.ACM32F0X0.1.0.1.pack
2021/11/11 10:39
uVision Softwar...
15 KB
ModulesDemo_Rev2.0.2.rar
2022/8/18 17:23
WinRAR 压缩文件
2,504 KB
rsonal

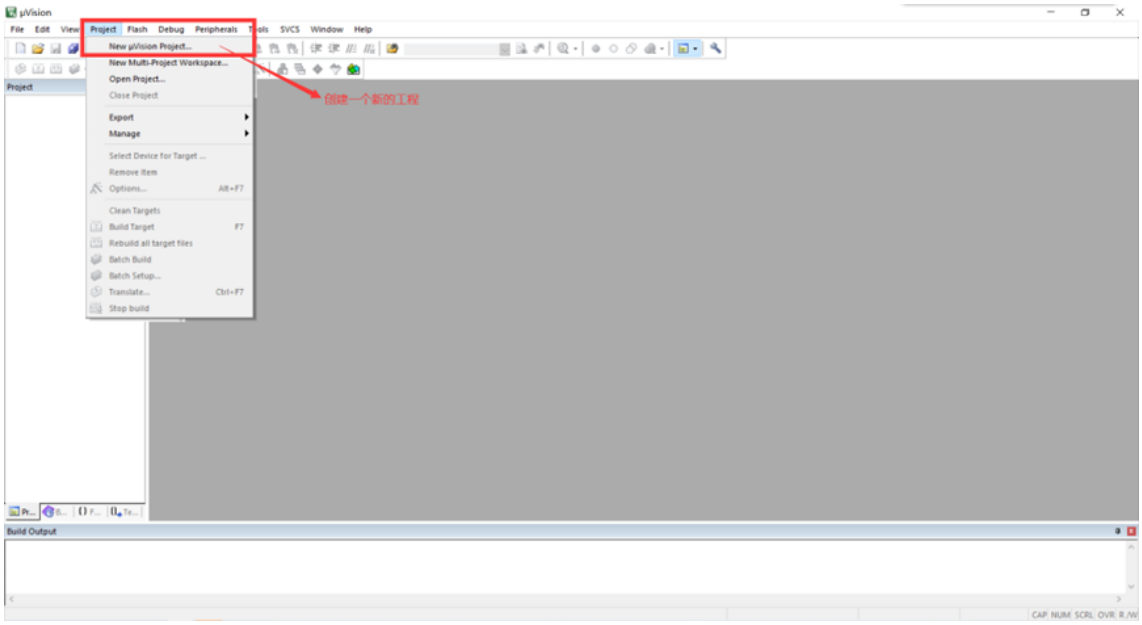

Pack Unzip: Aisinochip ACM32F0X0 1.0.1
Welcome to Keil Pack Unzip
Release 11/2021
This program installs the Software Pack:
Aisinochip ACM32F0X0 1.0.1
Aisinochip ACM32F0X0 ARM Cortex-M0 Device Family Pack
Destination Folder
C:\Users\86182\AppData\Local\Arm\Packs\Aisinochip\ACM32F0X0\1.0.1
Keil Pack Unzip
Pack already installed.
Click "Next" to replace.
<< Back
Next >>
Cancel

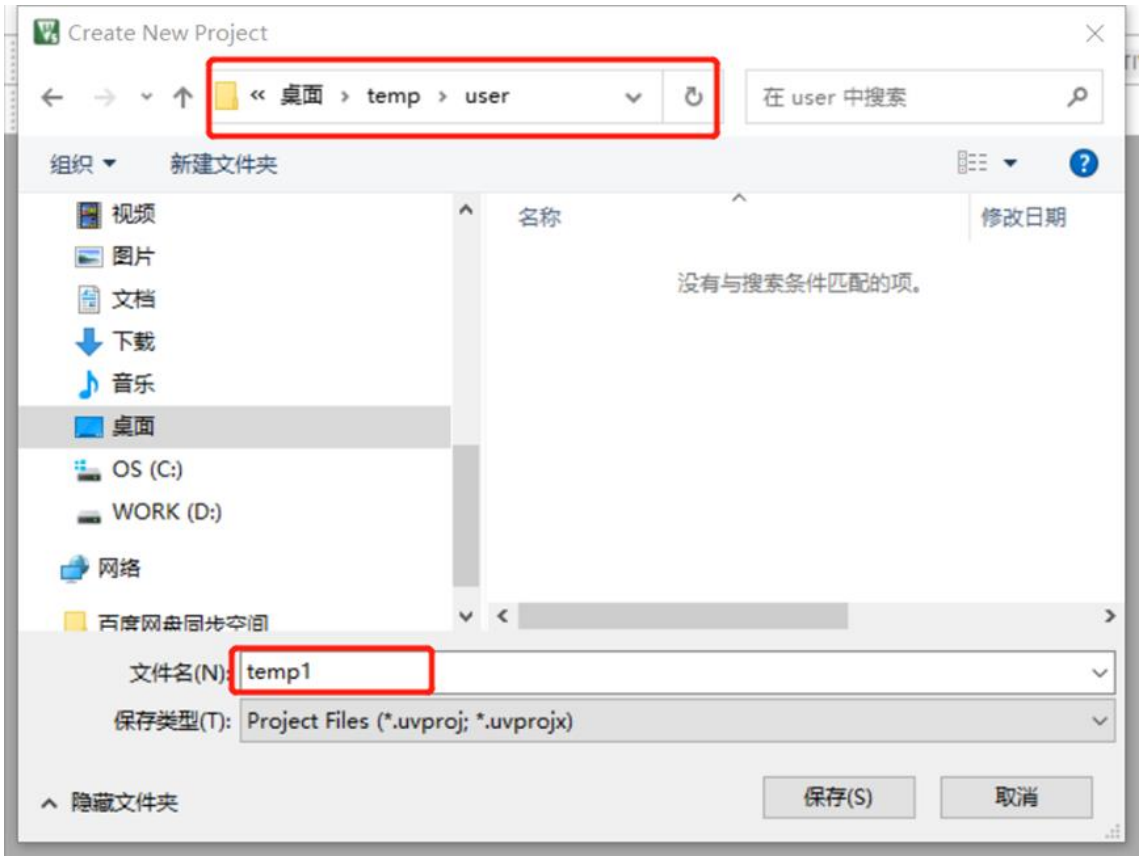


二、创建一个新的文件夹 temp，放入航芯提供的 Core_Drivers 文件，同时在里面创建一个新的文件夹 user。

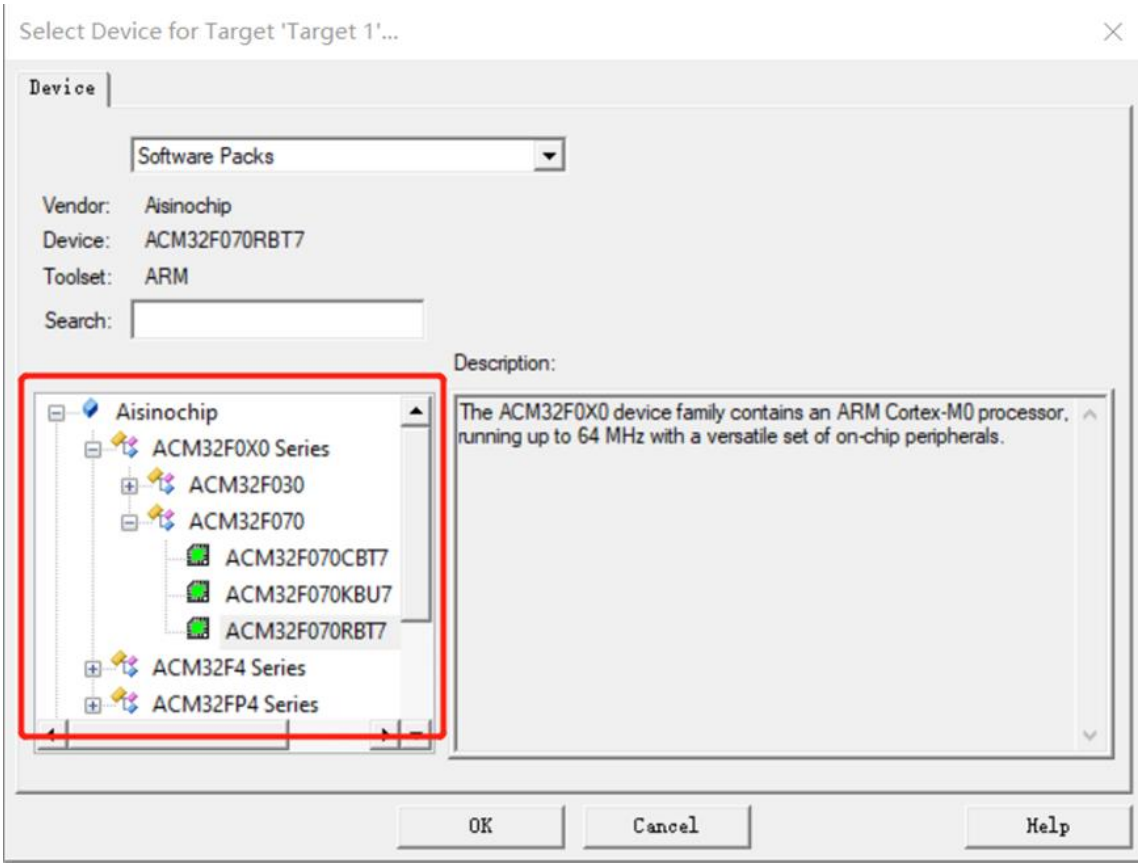


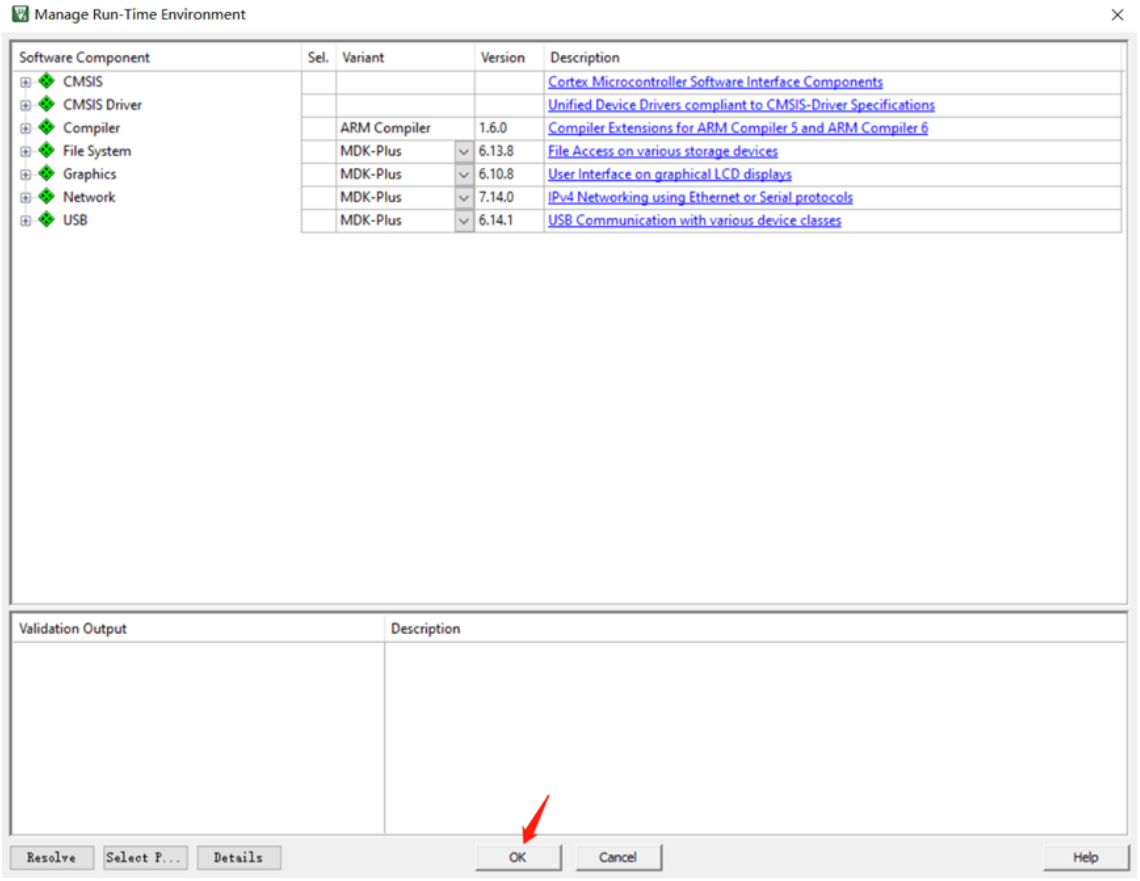
三、打开 keil，建立一个新的工程



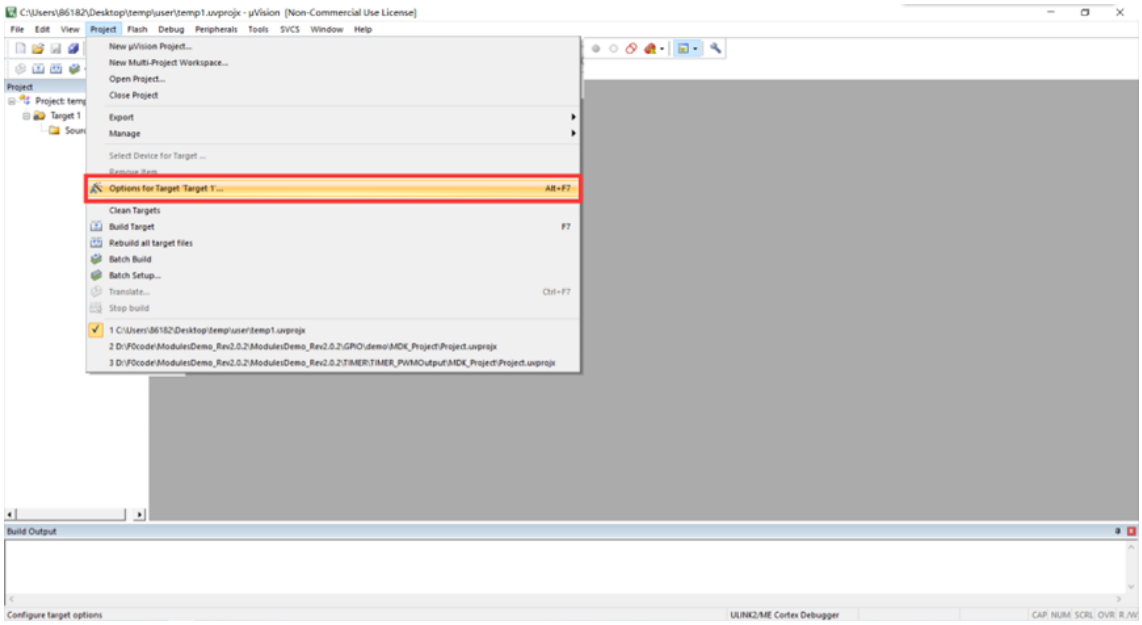


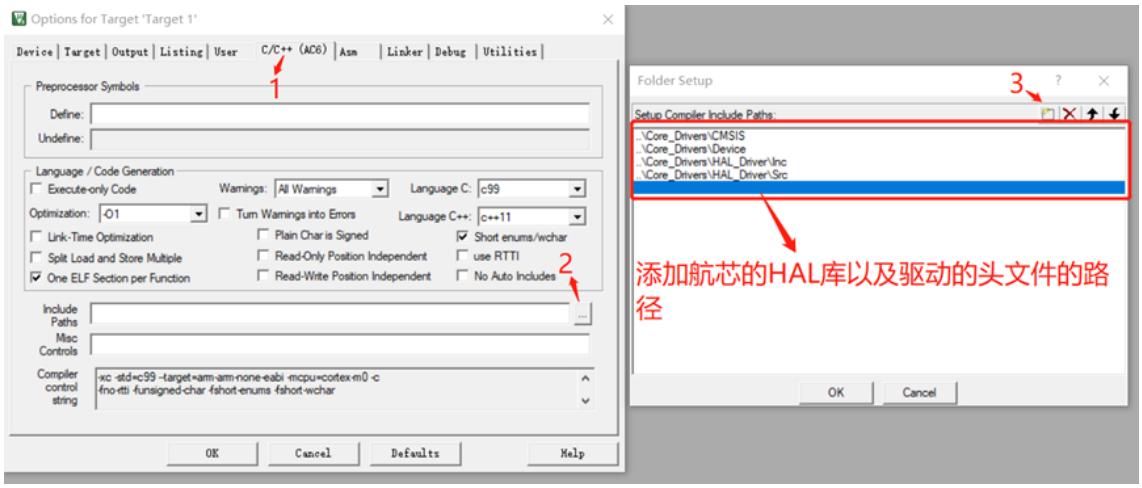
四、安装好了航芯的芯片包后，可以选择对应的设备号（没有找到对应的设备号可能是芯片包未安装好）


Select Device for Target 'Target 1'...
Device
Software Packs
Vendor: Aisinochip
Device: ACM32F070RBT7
Toolset: ARM
Search:
Description:
Aisinochip
ACM32F0X0 Series
ACM32F030
ACM32F070
ACM32F070CBT7
ACM32F070KBU7
ACM32F070RBT7
ACM32F4 Series
ACM32FP4 Series
The ACM32F0X0 device family contains an ARM Cortex-M0 processor, running up to 64 MHz with a versatile set of on-chip peripherals.
OK
Cancel
Help


Manage Run-Time Environment
Software Component
Sel.
Variant
Version
Description
CMSIS
Cortex Microcontroller Software Interface Components
CMSIS Driver
Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler
ARM Compiler
1.6.0
Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
File System
MDK-Plus
6.13.8
File Access on various storage devices
Graphics
MDK-Plus
6.10.8
User Interface on graphical LCD displays
Network
MDK-Plus
7.14.0
IPv4 Networking using Ethernet or Serial protocols
USB
MDK-Plus
6.14.1
USB Communication with various device classes
Validation Output
Description
Resolve
Select P...
Details
OK
Cancel
Help

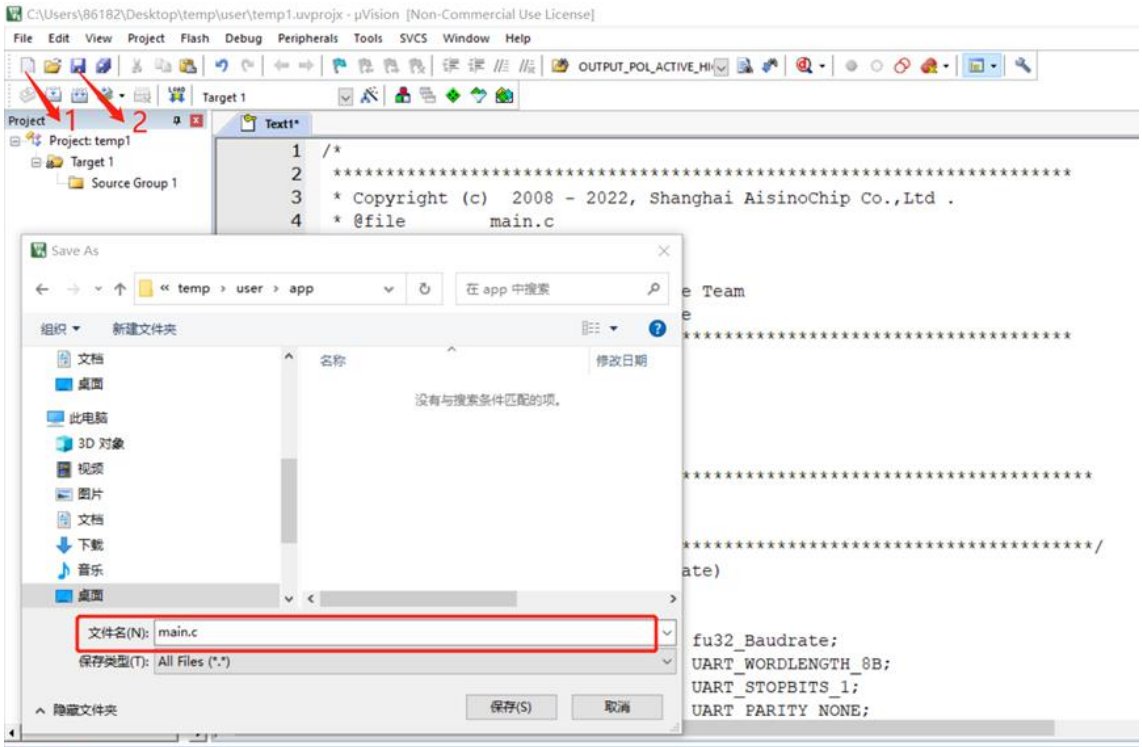
五、添加头文件（.h 文件）

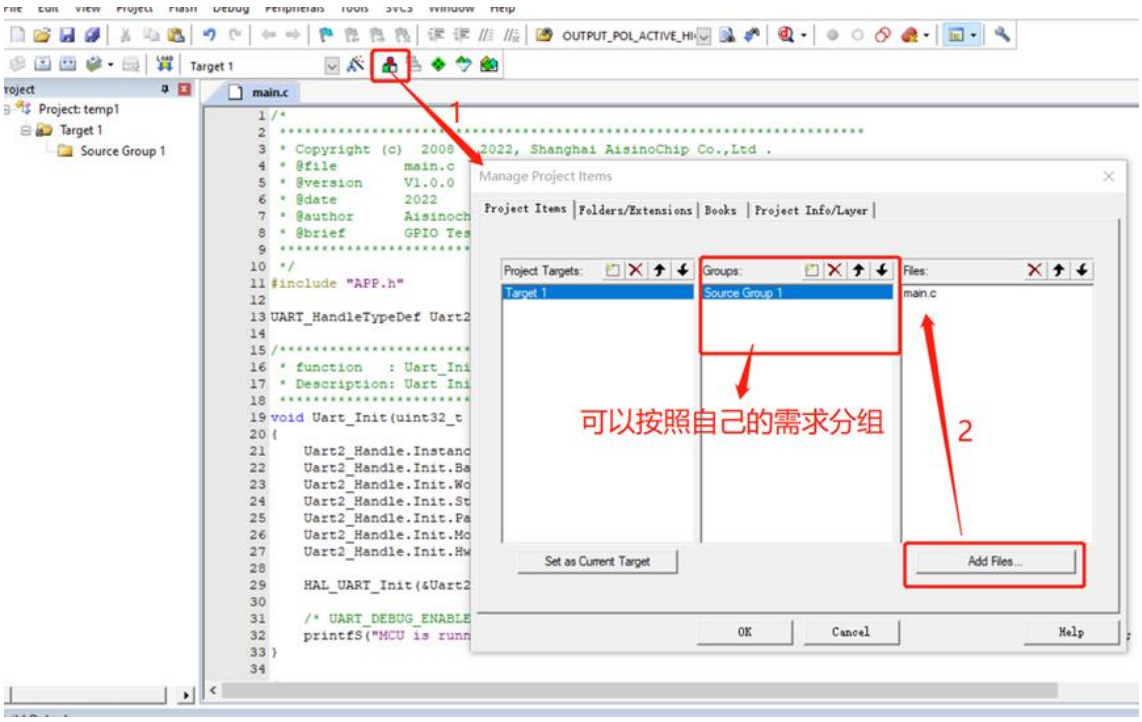


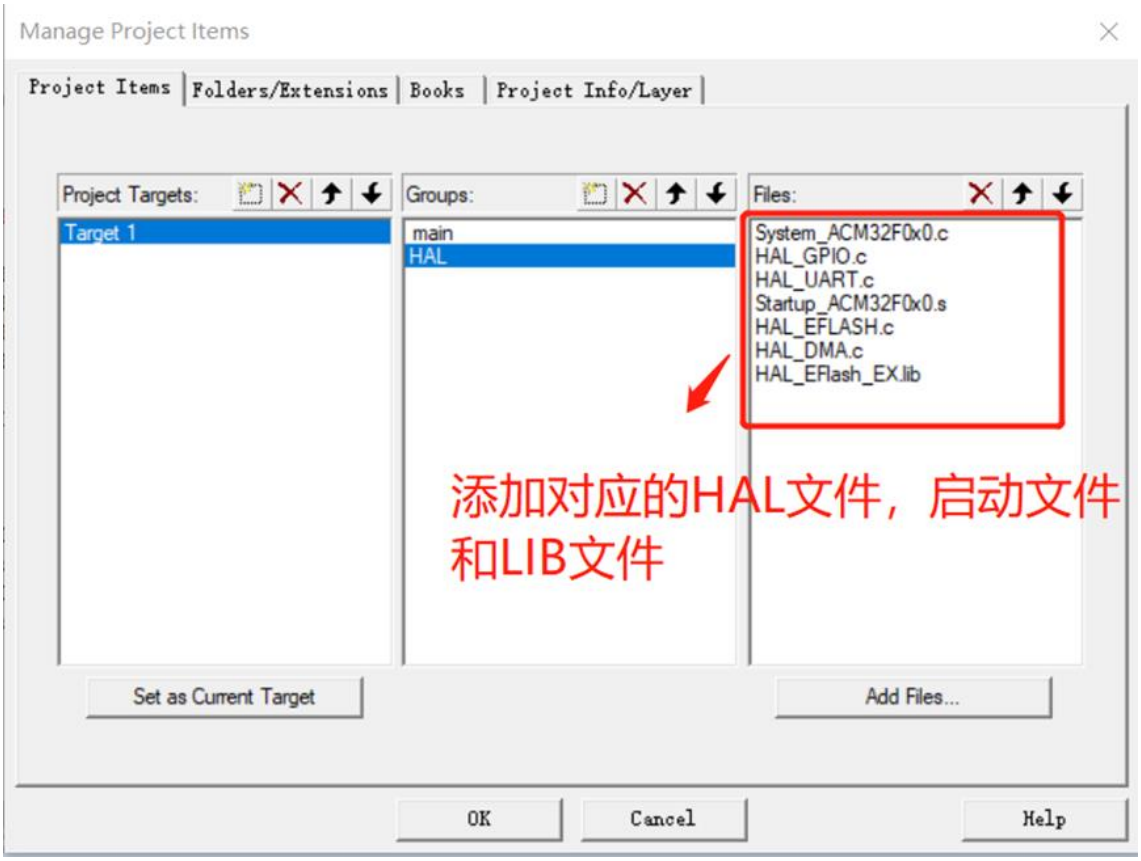


六、添加源文件（.C 文件）

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。

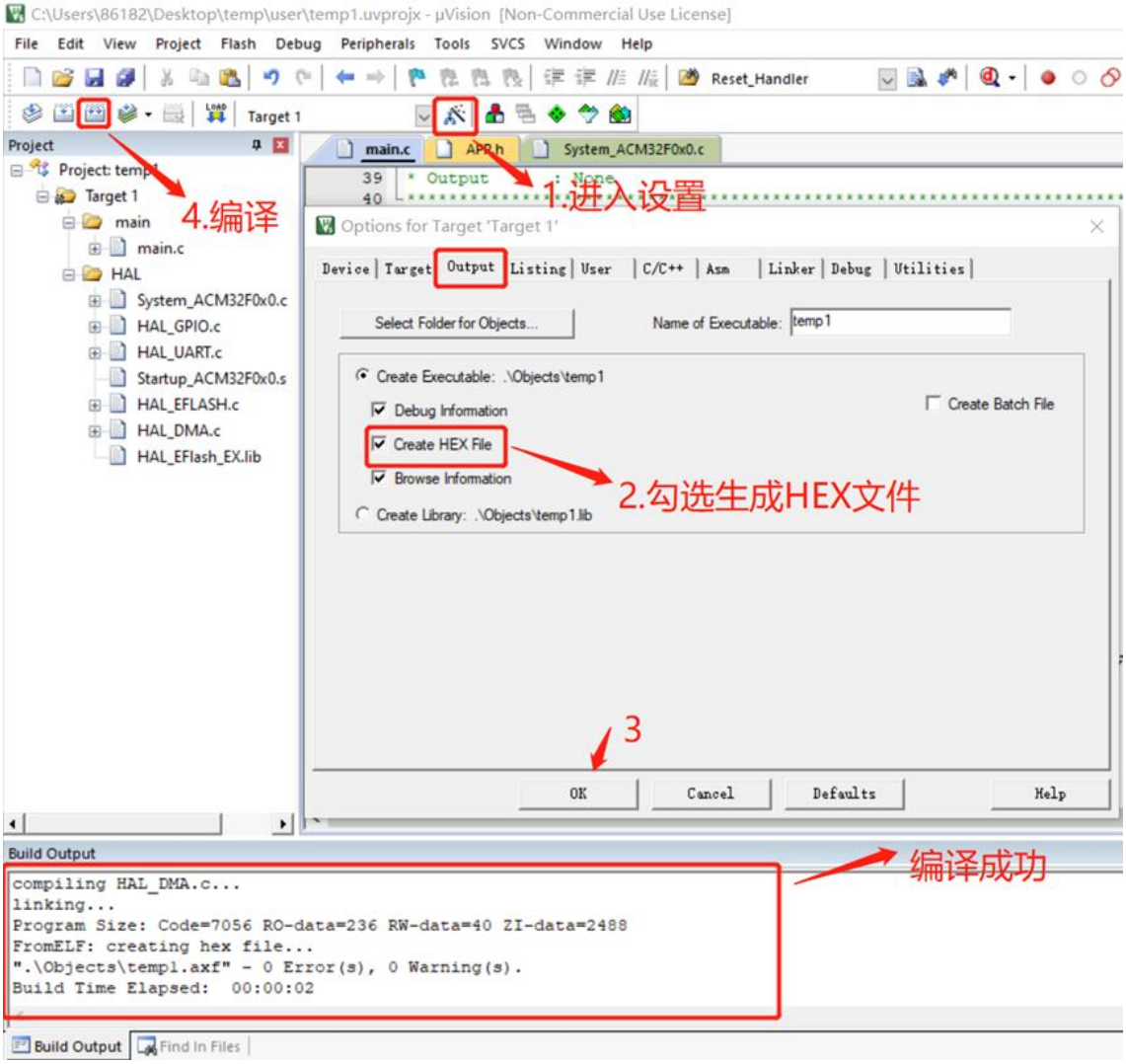

C:\Users\86182\Desktop\temp\user\temp1.uvprojx - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: temp1
Target 1
Source Group 1
1
2
Text1*
/*
* Copyright (c) 2008 - 2022, Shanghai AisinoChip Co.,Ltd .
* @file main.c
Save As
temp › user › app
在 app 中搜索
组织
新建文件夹
文档
桌面
此电脑
3D 对象
视频
图片
文档
下载
音乐
桌面
名称
修改日期
没有与搜索条件匹配的项。
文件名(N): main.c
保存类型(T): All Files (*.*)
隐藏文件夹
保存(S)
取消
Build Output
Rebuild started: Project: temp1


Target 1
Project
Project: temp1
Target 1
Source Group 1
main.c
1
Manage Project Items
Project Items
Folders/Extensions
Books
Project Info/Layer
Project Targets:
Groups:
Files:
Target 1
Source Group 1
main.c
可以按照自己的需求分组
2
Set as Current Target
Add Files...
OK
Cancel
Help

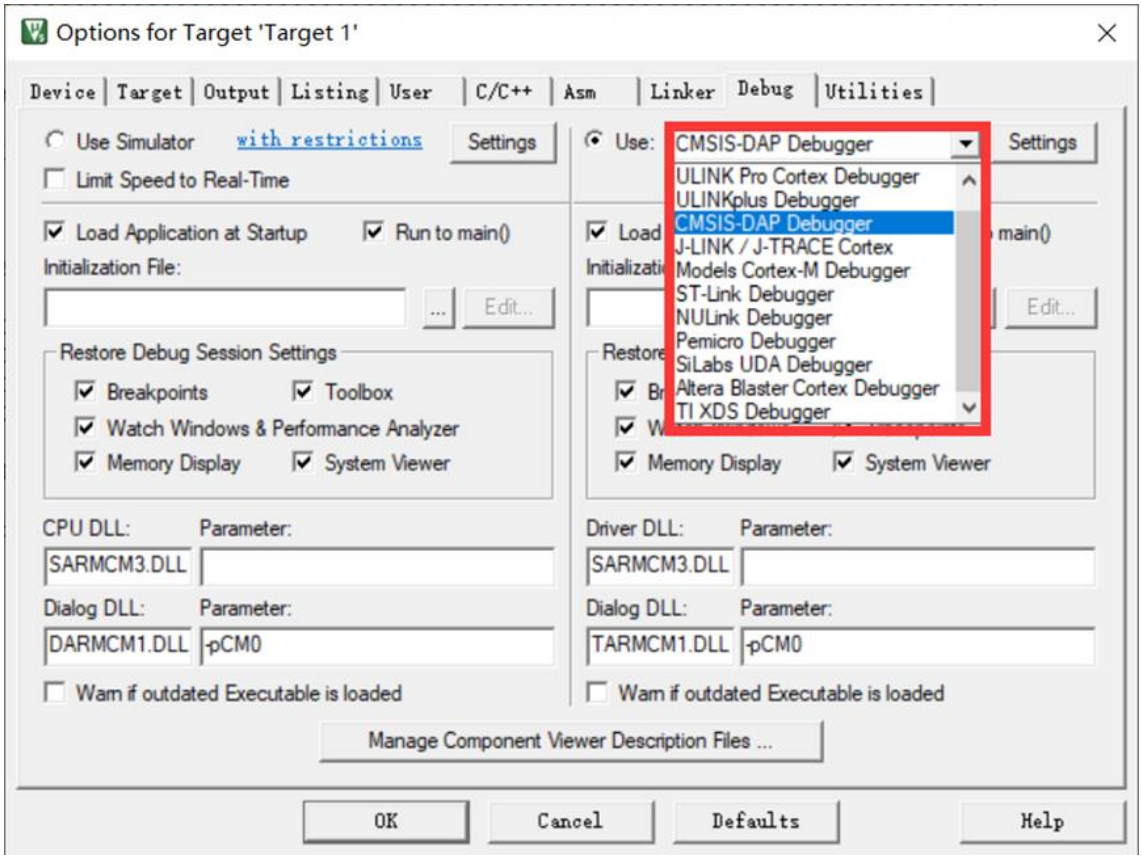


1.3 keil 编译、下载、运行

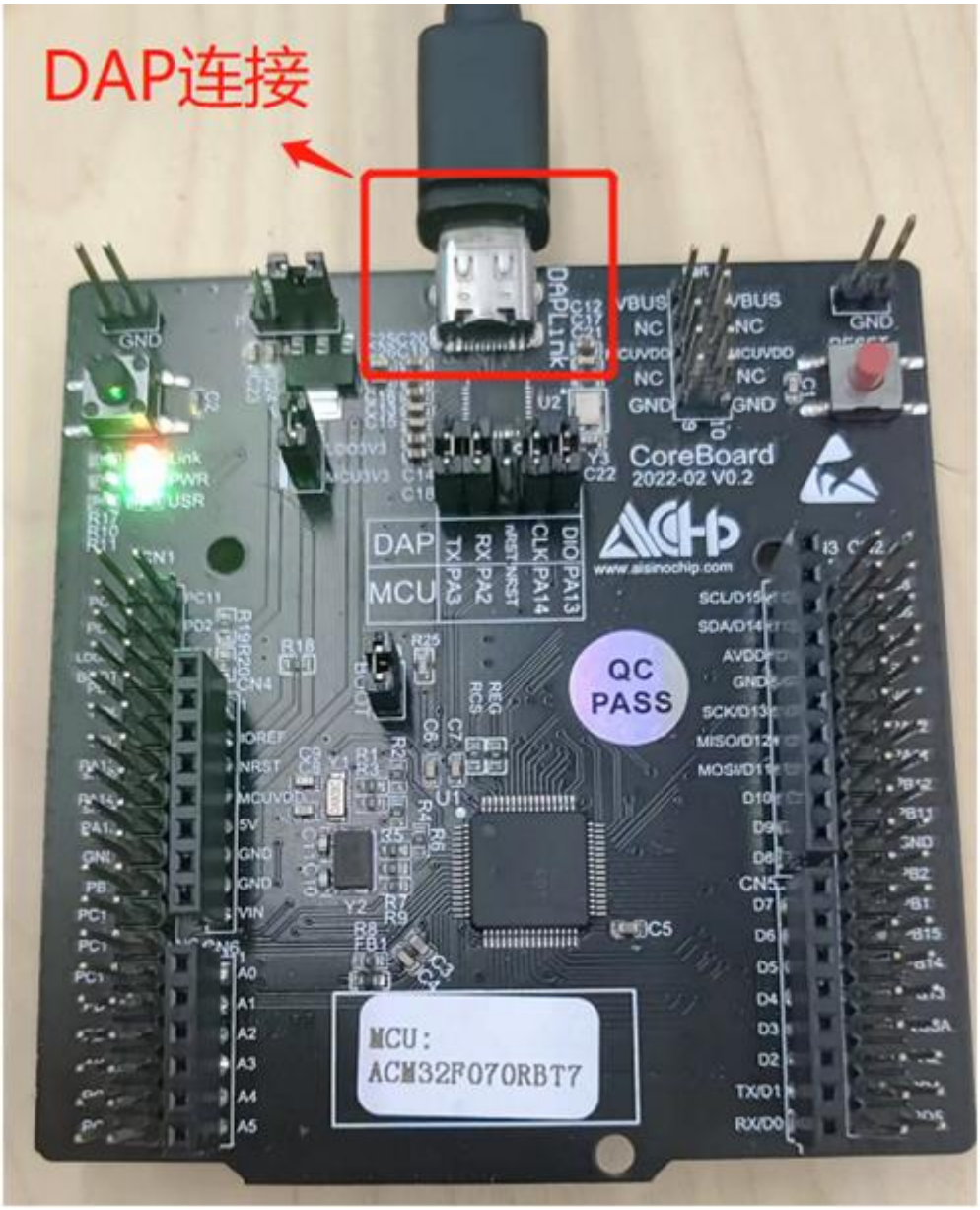
一、工程编译

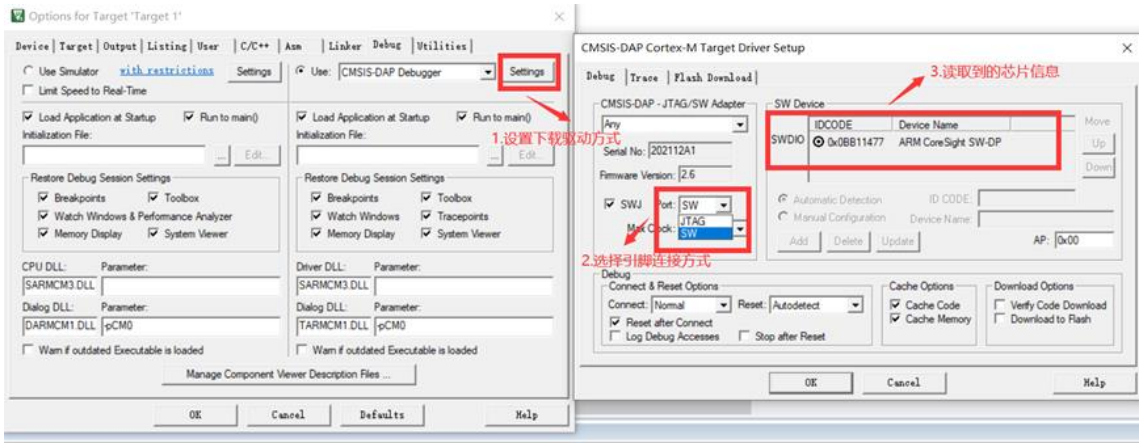


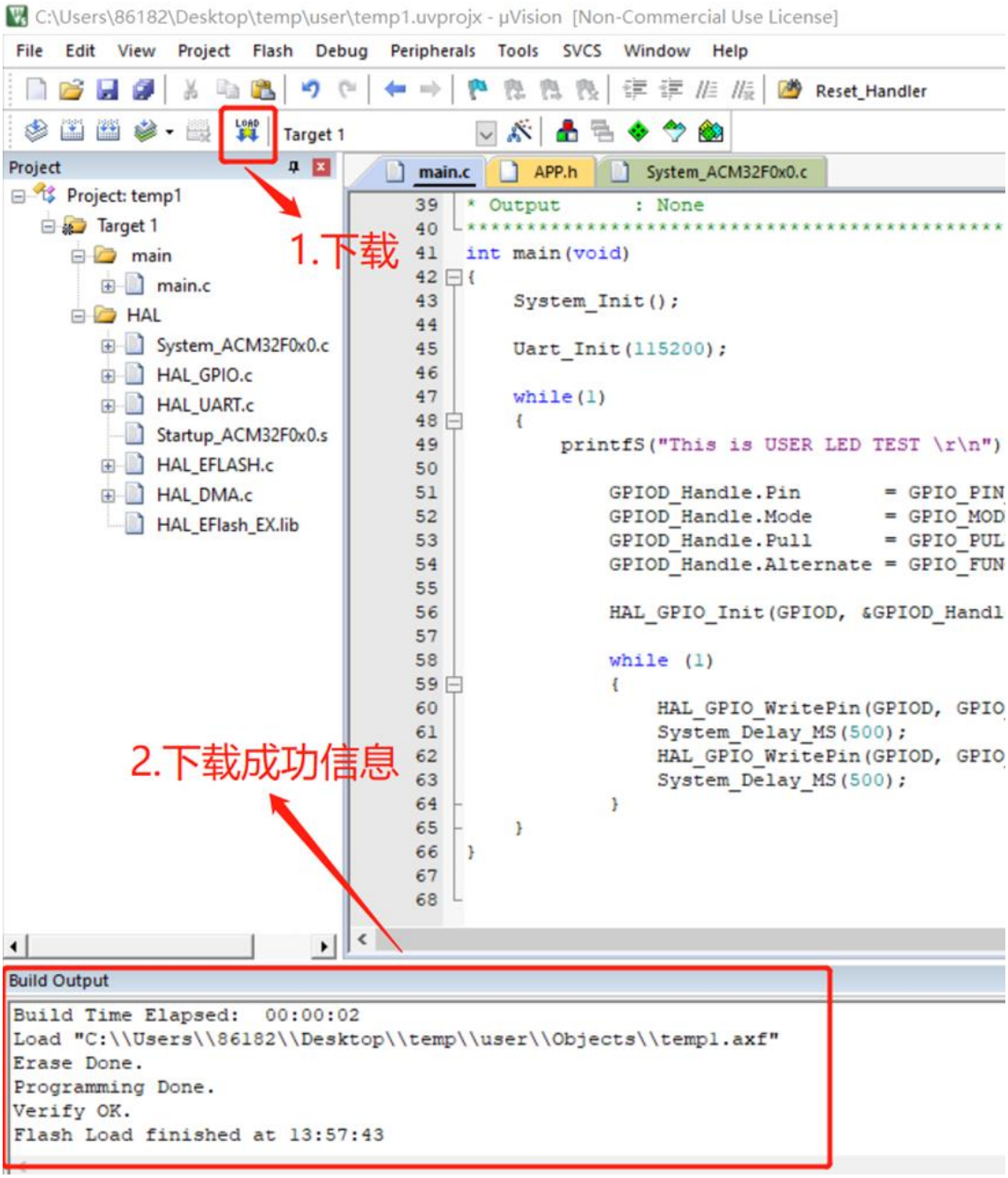
二、下载程序（ACM32F0X0 系列支持的在线仿真器包括： J-Link、 U-Link2、 ST-Link、 CMSIS-DAP，ACM32F4XX 系列支持的在线仿真器包括： J-Link-V9（含）以上、 U-Link2、 CMSIS-DAP 等，使用 J-Link 在线调试时， Keil 推荐在 MDK5.31（含）以上， J-Link 驱动建议在 V6.70e（含） 以上）



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》。


DAP连接
CoreBoard
2022-02 V0.2
www.aisinochip.com
QC
PASS
MCU:
ACM32F070RBT7

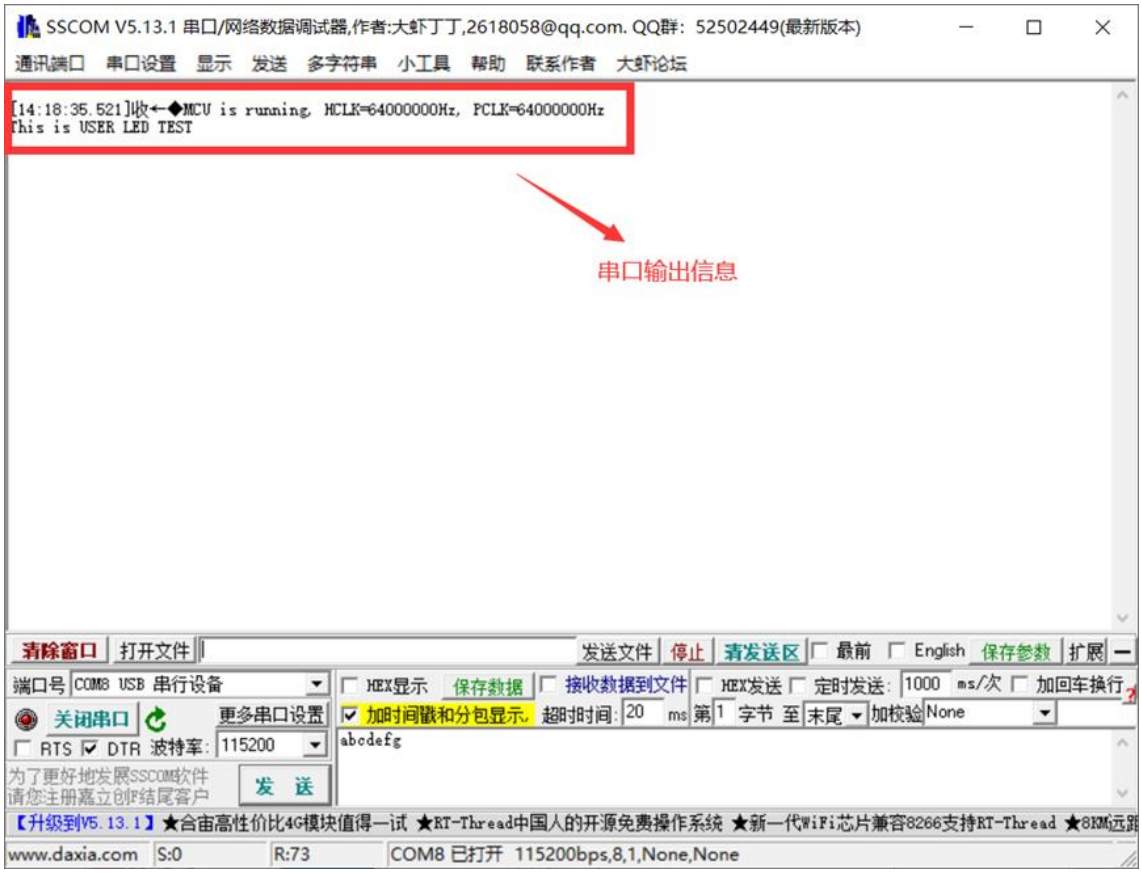

Options for Target 'Target 1'
Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities
Use Simulator with restrictions
Settings
Use: CMSIS-DAP Debugger
Settings
Limit Speed to Real-Time
Load Application at Startup
Run to main()
Initialization File:
Restore Debug Session Settings
Breakpoints
Toolbox
Watch Windows & Performance Analyzer
Memory Display
System Viewer
CPU DLL:
Parameter:
SARMCM3.DLL
Dialog DLL:
DARMCM1.DLL
pCM0
Warn if outdated Executable is loaded
Manage Component Viewer Description Files ...
OK
Cancel
Defaults
Help
1.设置下载驱动方式
2.选择引脚连接方式
CMSIS-DAP Cortex-M Target Driver Setup
3.读取到的芯片信息
Debug
Trace
Flash Download
CMSIS-DAP - JTAG/SW Adapter
SW Device
IDCODE
Device Name
SWDIO
0x0BB11477
ARM CoreSight SW-DP
Serial No: 202112A1
Firmware Version: 2.6
SWJ
Port: SW
JTAG
SW
Max Clock:
Automatic Detection
Manual Configuration
ID CODE:
Device Name:
Add
Delete
Update
AP: 0x00
Debug
Connect & Reset Options
Connect: Normal
Reset: Autodetect
Reset after Connect
Log Debug Accesses
Stop after Reset
Cache Options
Cache Code
Cache Memory
Download Options
Verify Code Download
Download to Flash
OK
Cancel
Help



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 GPIO 口翻转控制 LED 灯闪烁为例，部分示例代码（完整代码见附录）和现象如下：

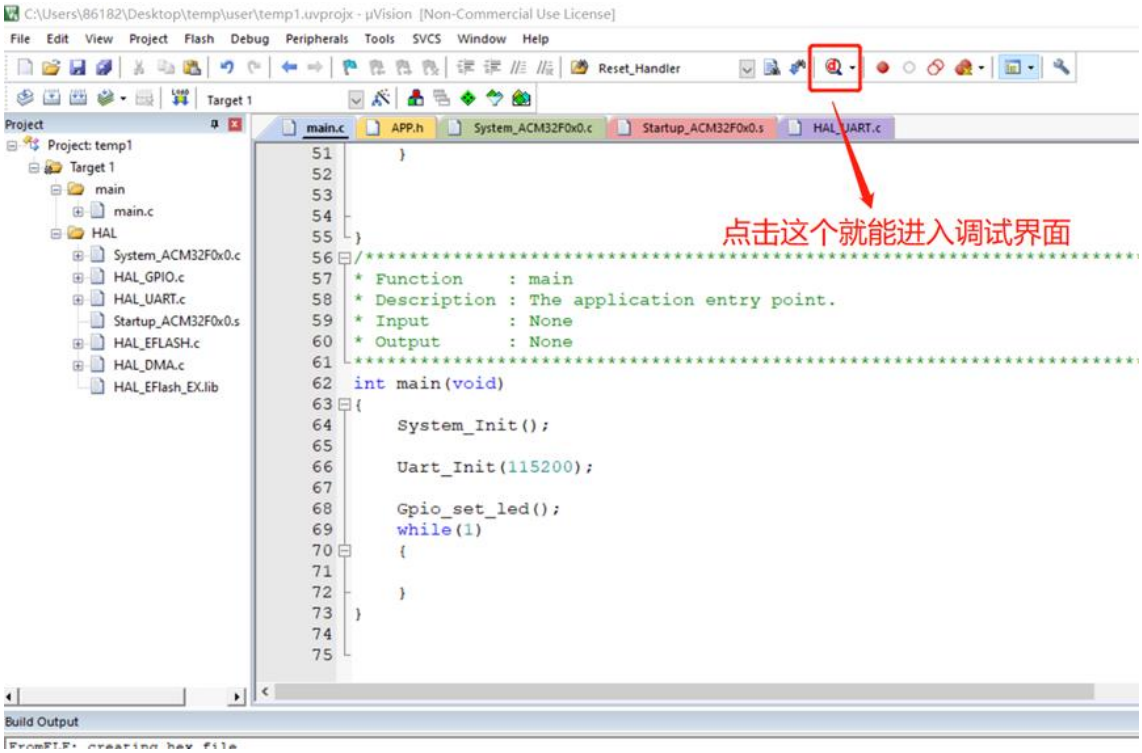
```c
void Uart_Init(uint32_t fu32_Baudrate)
{
    Uart2_Handle.Instance = UART2;
    Uart2_Handle.Init.BaudRate   = fu32_Baudrate;
    Uart2_Handle.Init.WordLength = UART_WORDLENGTH_8B;
    Uart2_Handle.Init.StopBits   = UART_STOPBITS_1;
    Uart2_Handle.Init.Parity     = UART_PARITY_NONE;
    Uart2_Handle.Init.Mode       = UART_MODE_TX_RX_DEBUG;
    Uart2_Handle.Init.HwFlowCtl  = UART_HWCONTROL_NONE;

    HAL_UART_Init(&Uart2_Handle);

    /* UART_DEBUG_ENABLE control printfS */
    printfS("MCU is running, HCLK=%dHz, PCLK=%dHz\n", System_Get_SystemClock(), System_Get_APBClock());
}
void Gpio_set_led()
{
    printfS("This is USER LED TEST \r\n");

    GPIOD_Handle.Pin       = GPIO_PIN_3;
    GPIOD_Handle.Mode      = GPIO_MODE_OUTPUT_PP;
    GPIOD_Handle.Pull      = GPIO_PULLUP;
    GPIOD_Handle.Alternate = GPIO_FUNCTION_0;

    HAL_GPIO_Init(GPIOD, &GPIOD_Handle);

    while (1)
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_3, GPIO_PIN_SET);
        System_Delay_MS(500);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_3, GPIO_PIN_CLEAR);
        System_Delay_MS(500);
    }
```


按下按键后运行
CoreBoard
2022-02 V0.2
www.aisinochip.com
QC
PASS
MCU:
ACM32F070RBT7



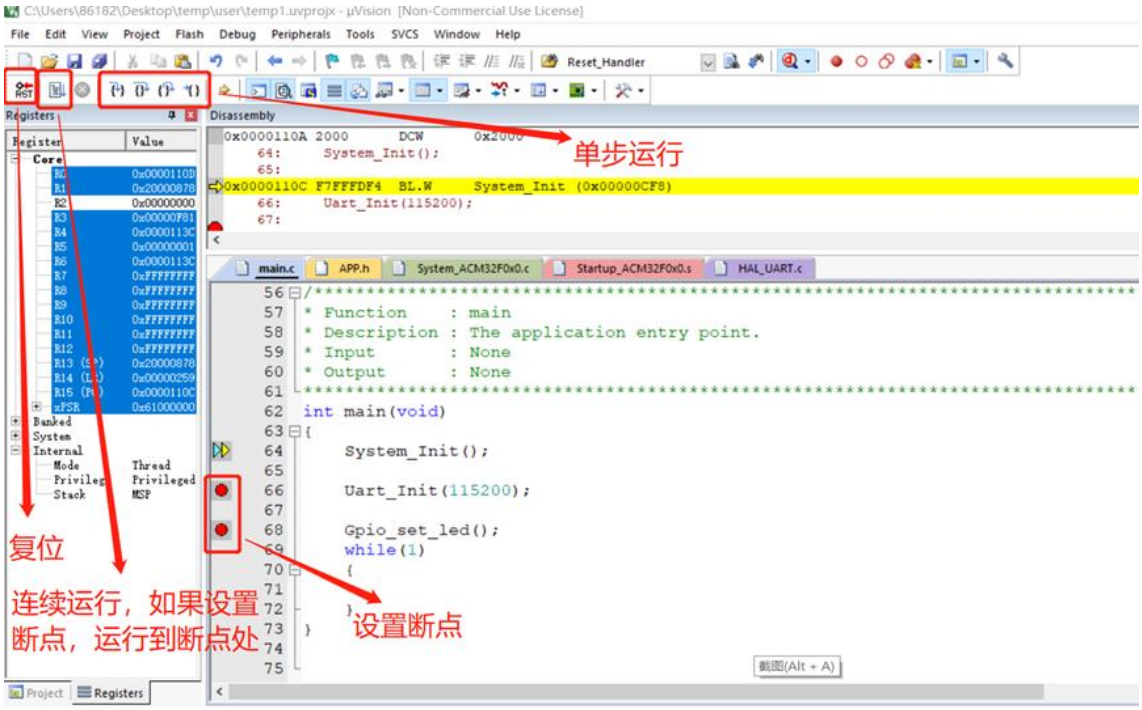
1.4 Keil Debug 使用说明

一、进入调试界面

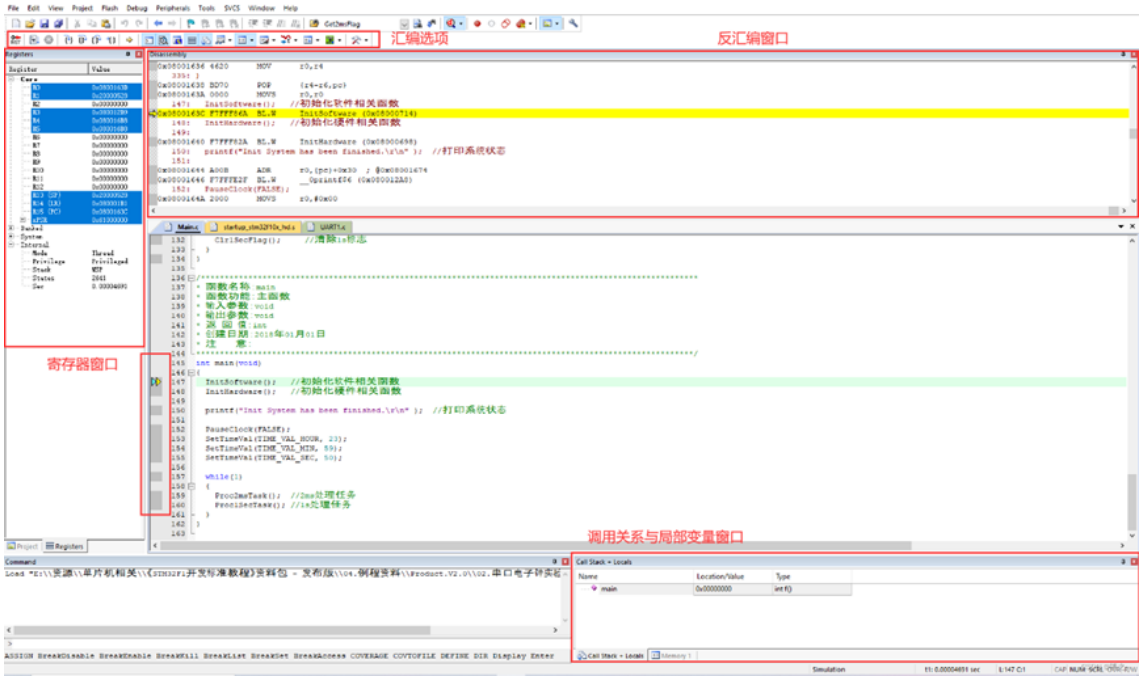


二、调试工具栏介绍

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

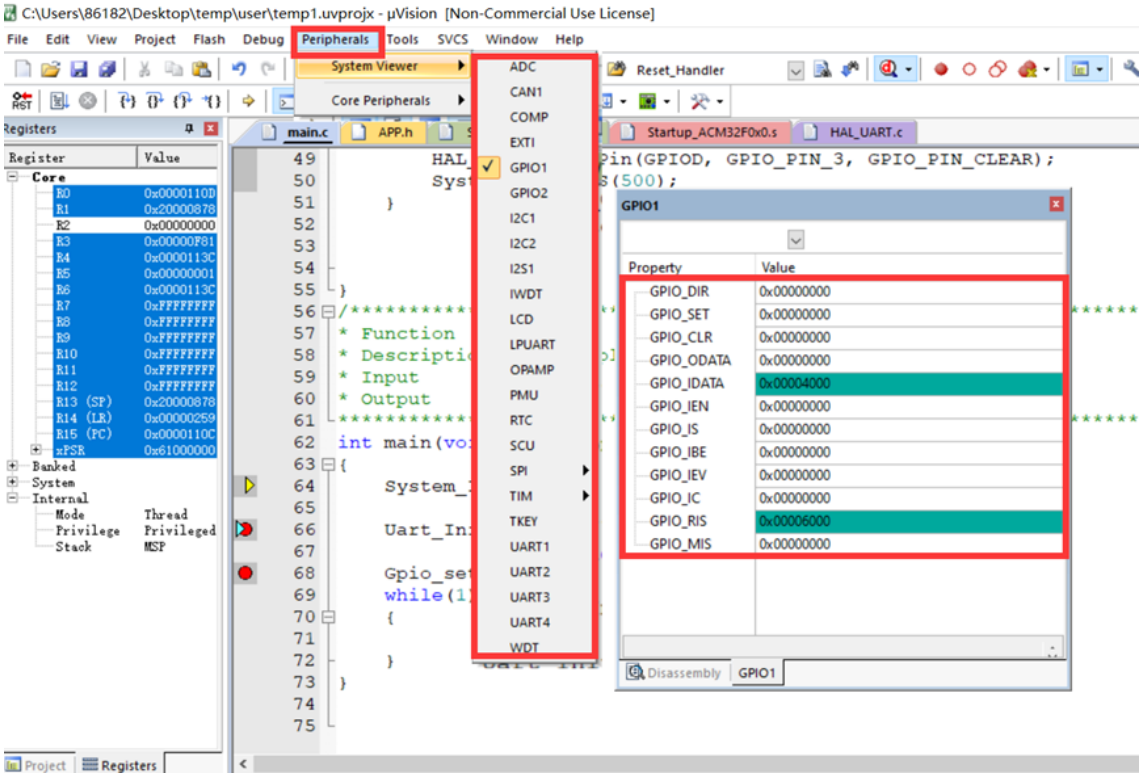


调试选项	名称	介绍
	Reset	实现程序复位，以重新运行
	Run	使程序开始运行
	Stop	使程序停止运行（程序运行状态时才有效）
	Step	运行当前行代码（若为函数则进入函数）
	Step Over	运行至下一行代码（若为函数会将函数执行完毕）
	Step Out	跳出当前函数
	Run to Cursor line	运行至蓝色箭头处
	Show Next Statement	跳转至当前代码（方便在打开多个文件时找到代码运行处）

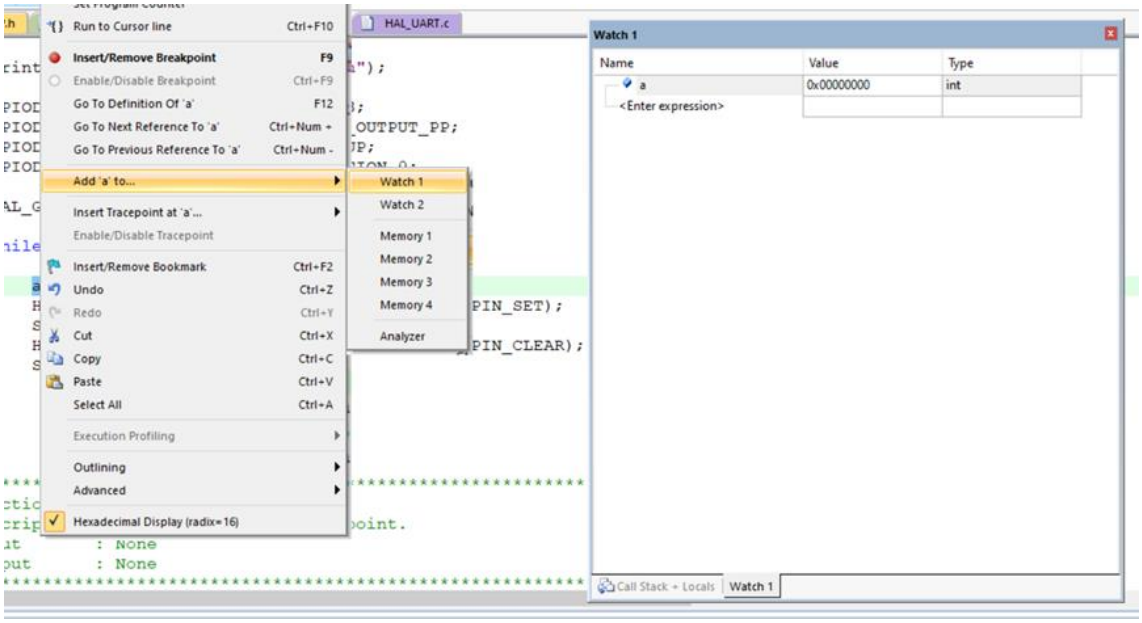


三、调试窗口介绍

1 查看外设寄存器的值，参考《航芯 ACM32F0X0_FP0X_用户手册_V1.6.pdf》查看各个外设寄存器的功能。



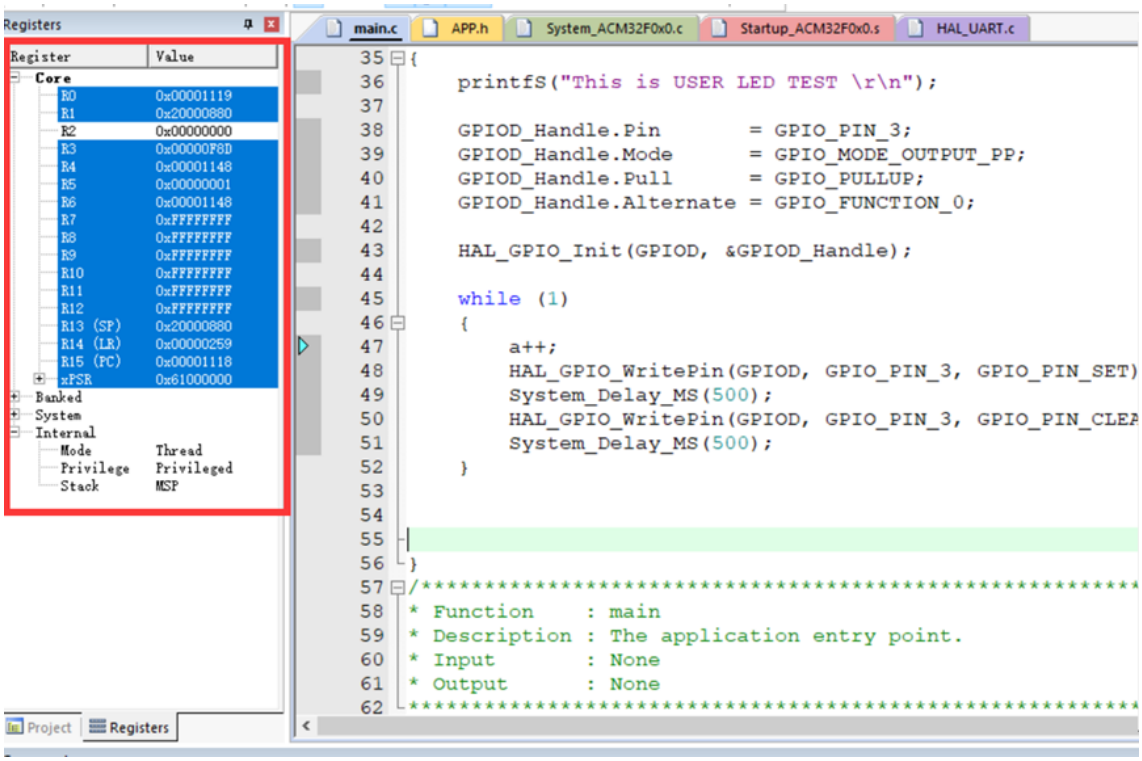
2 选中需要查看的参数，单击鼠标右键，可以将参数放入观察窗口，实时查看参数的变化



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

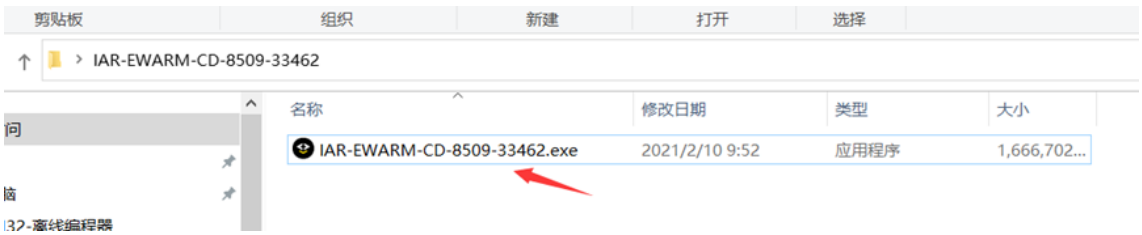
外设寄存器），关于内核寄存器的介绍可以查看 ARM 架构基本寄存器。



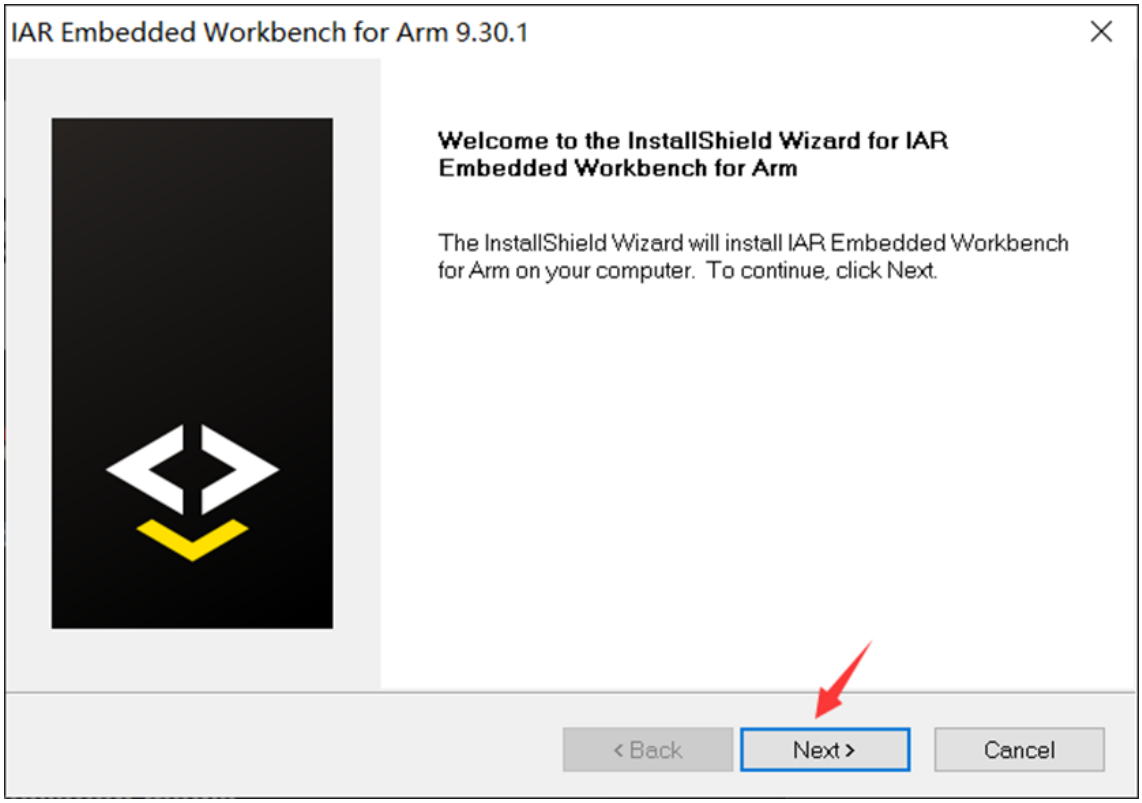
IAR 篇

1.1 IAR 安装步骤

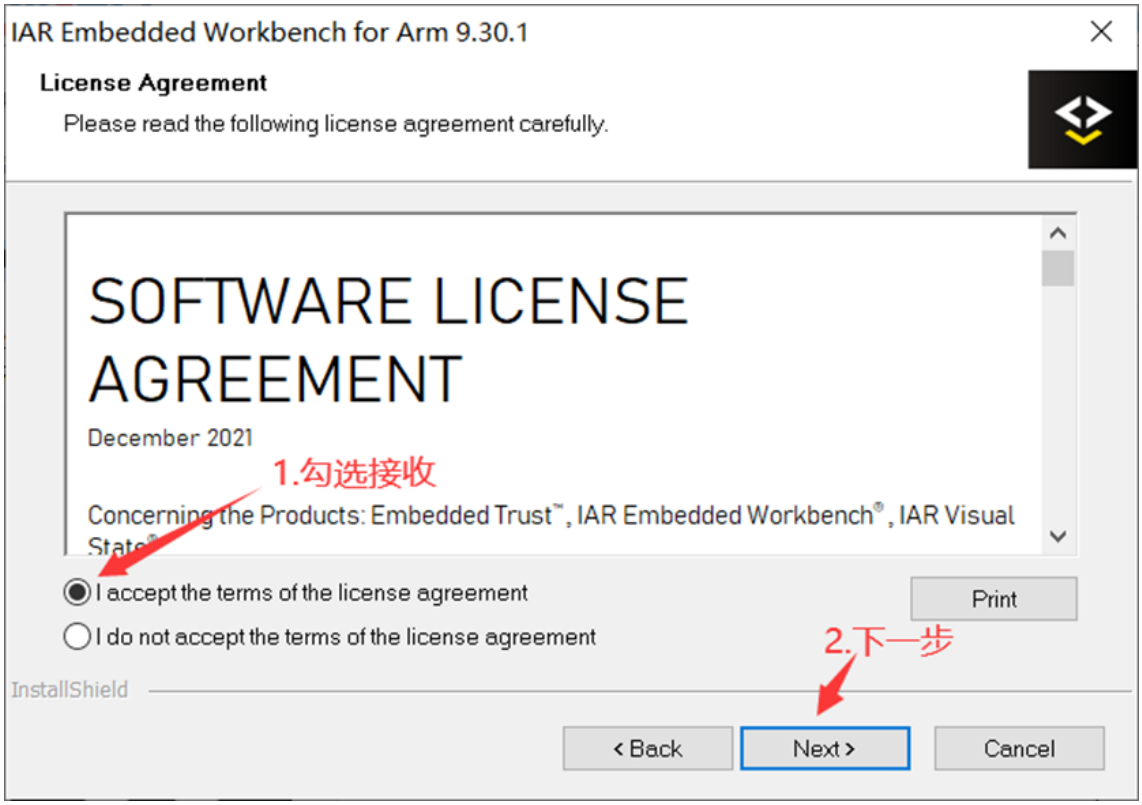
一、下载并解压安装包，并按步骤完成安装



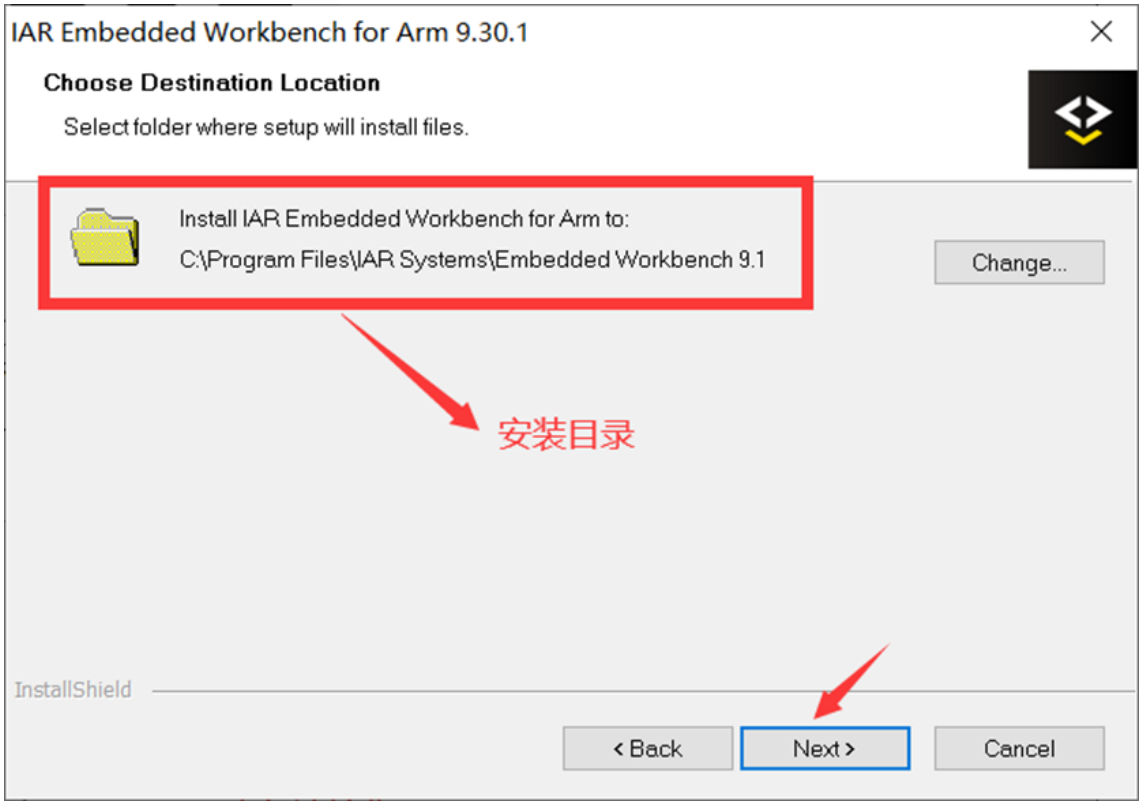
二、运行安装程序，点击 next



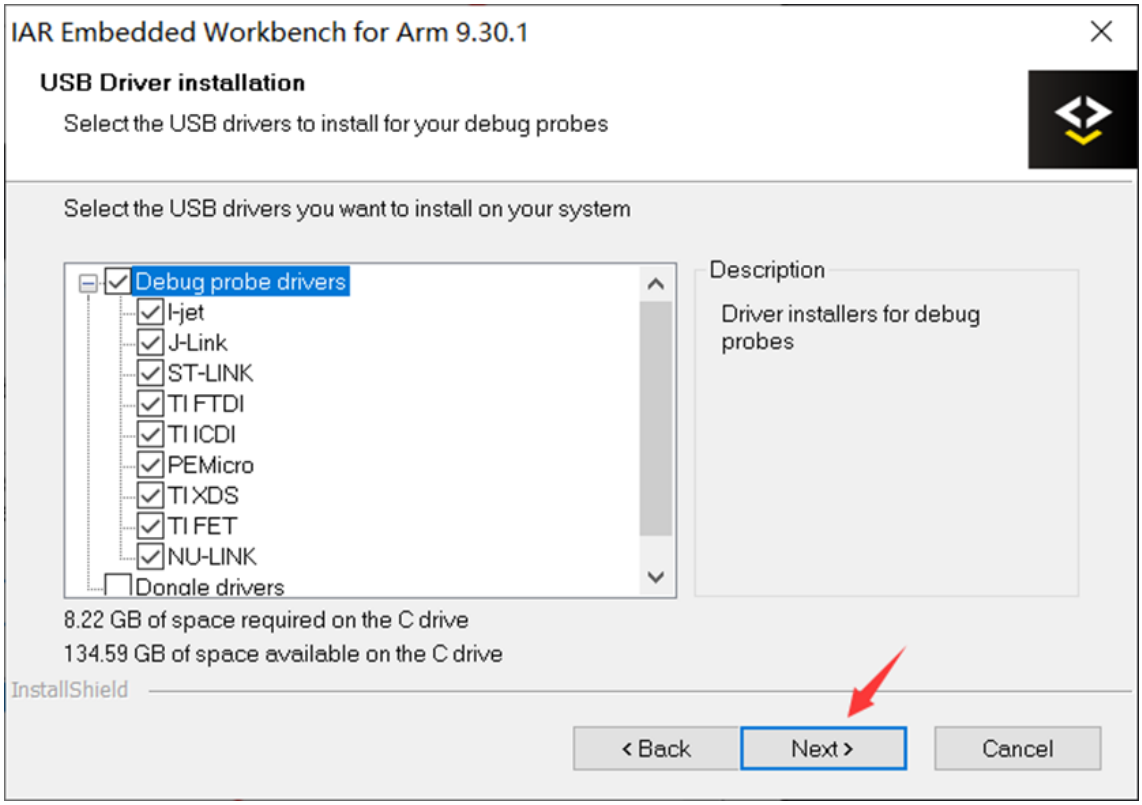
三、勾选 accept，点击 next



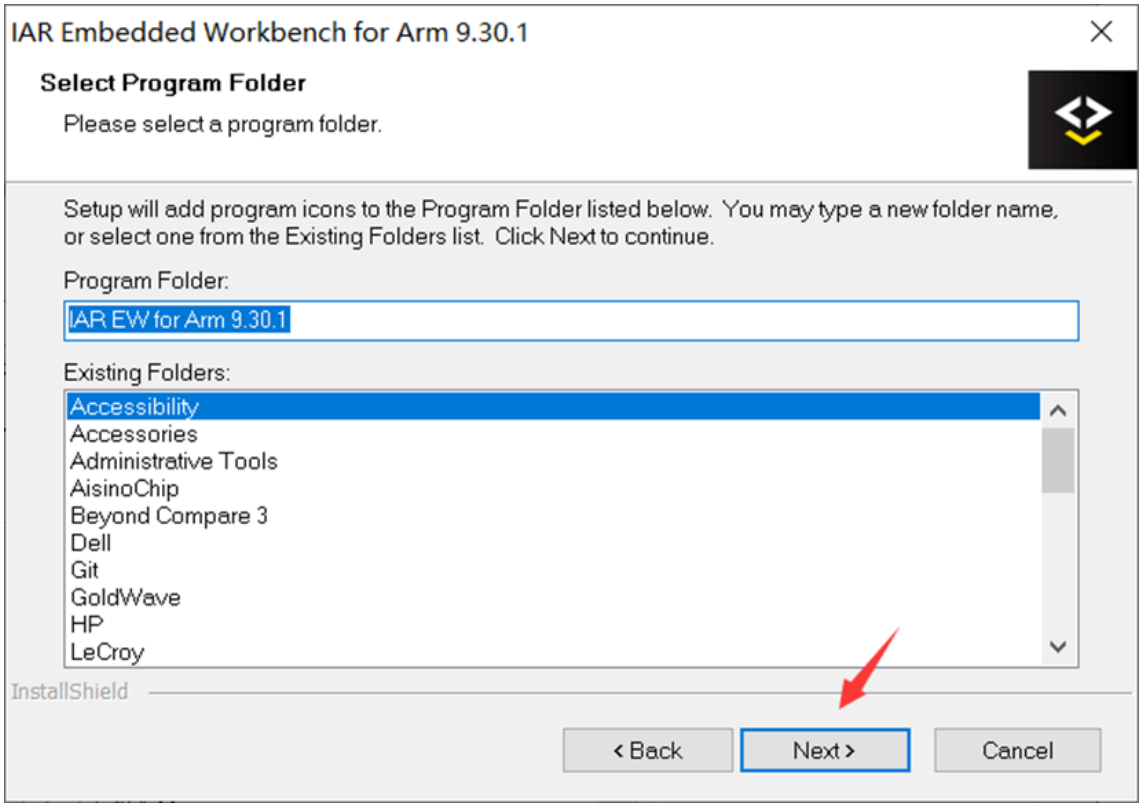
四、选择安装路径，点击 next



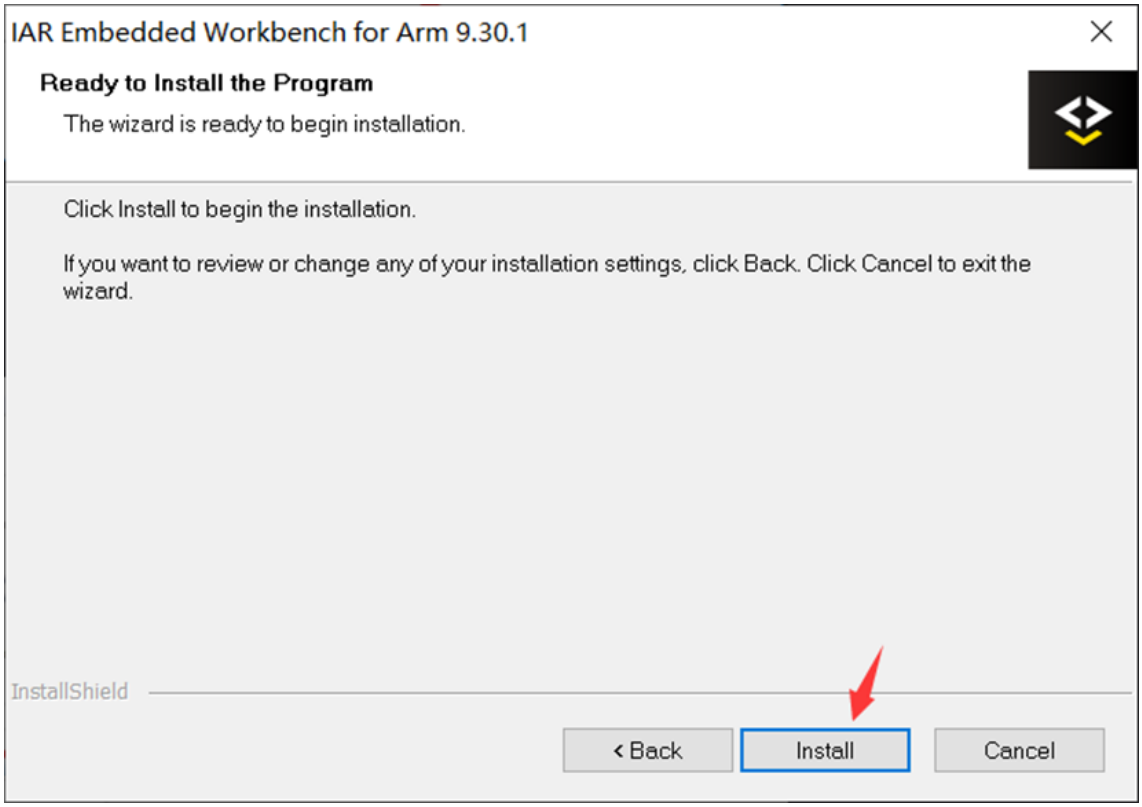
五、安装驱动，为避免以后可能使用到其他下载器，默认即可，会自动安装一些驱动



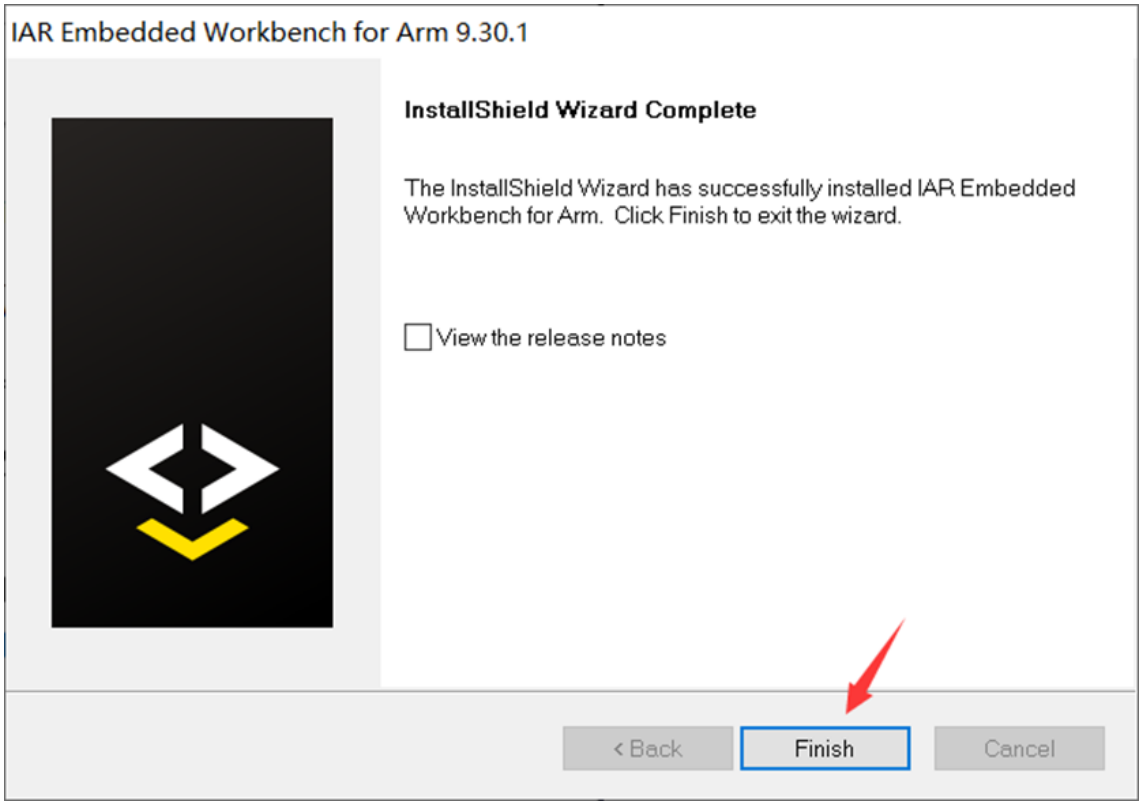
六、直接选择"NEXT"



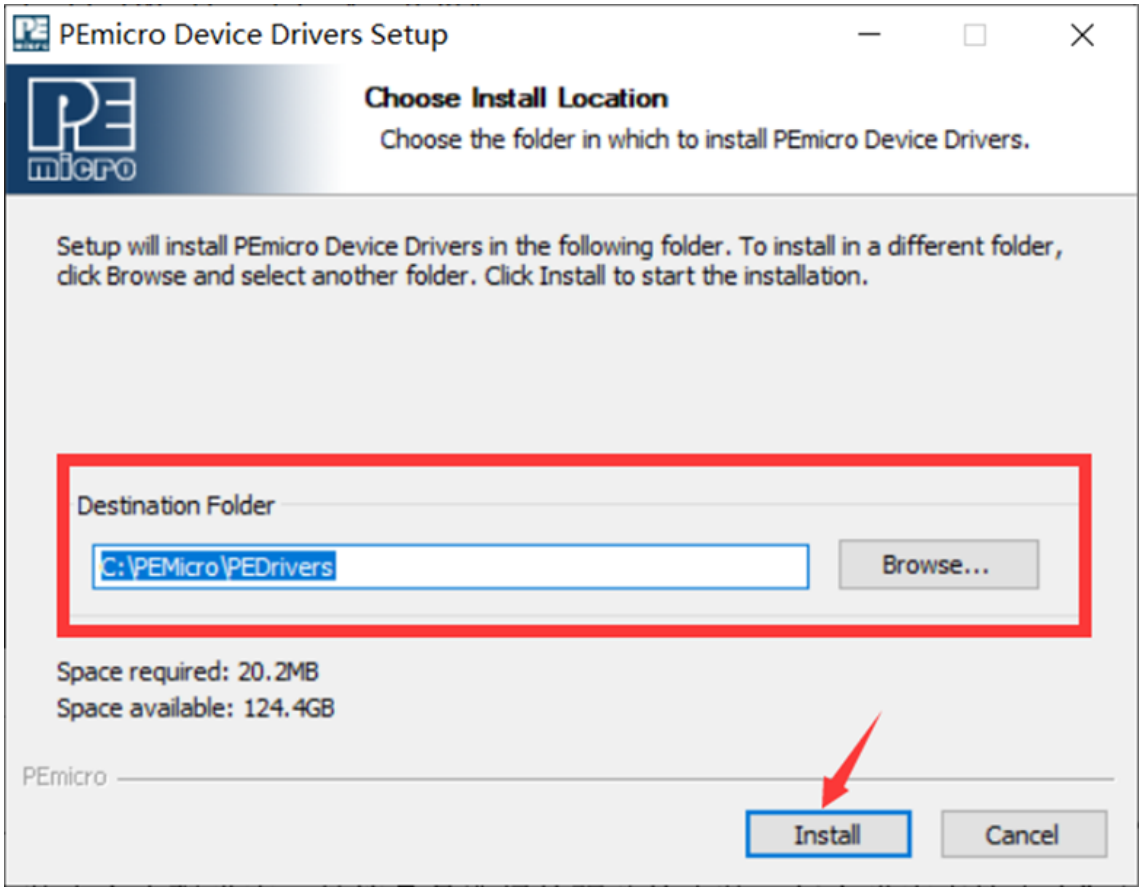
七、点击"install"

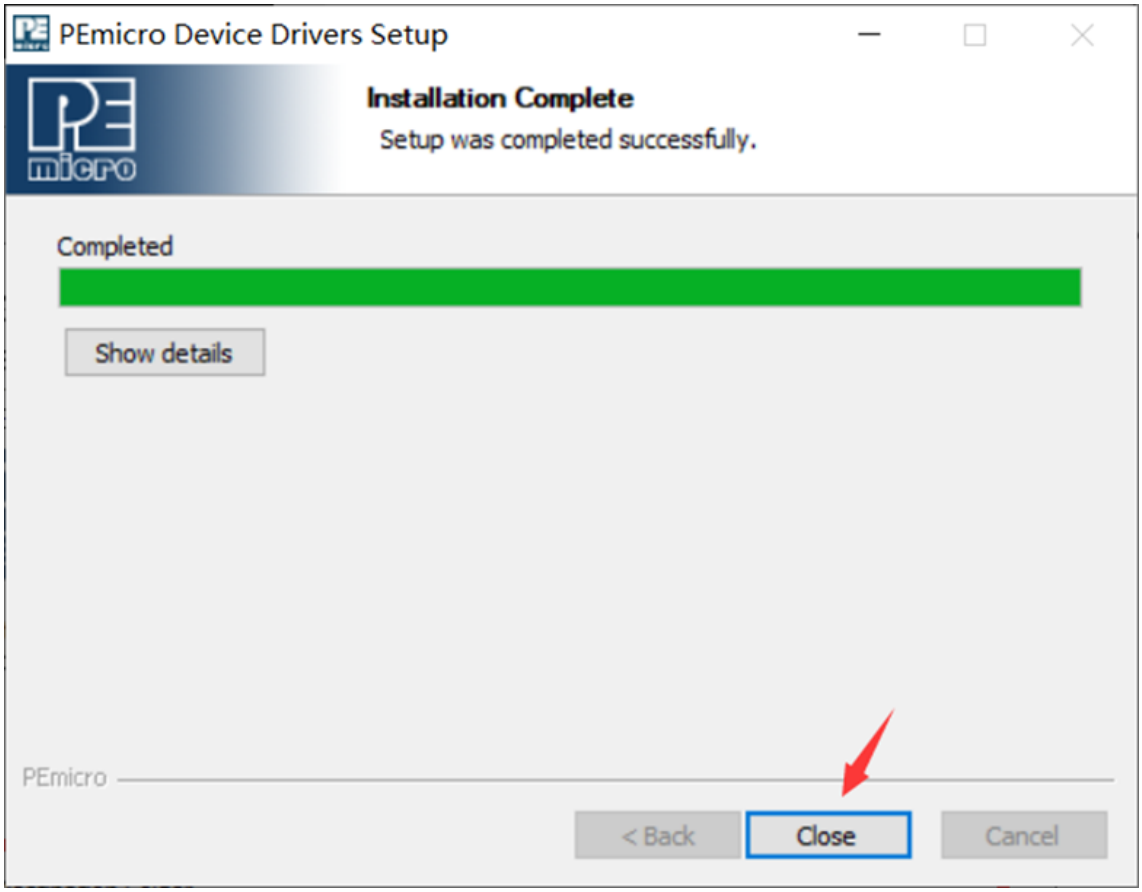


八、等待安装完成， 点击"finish"完成安装



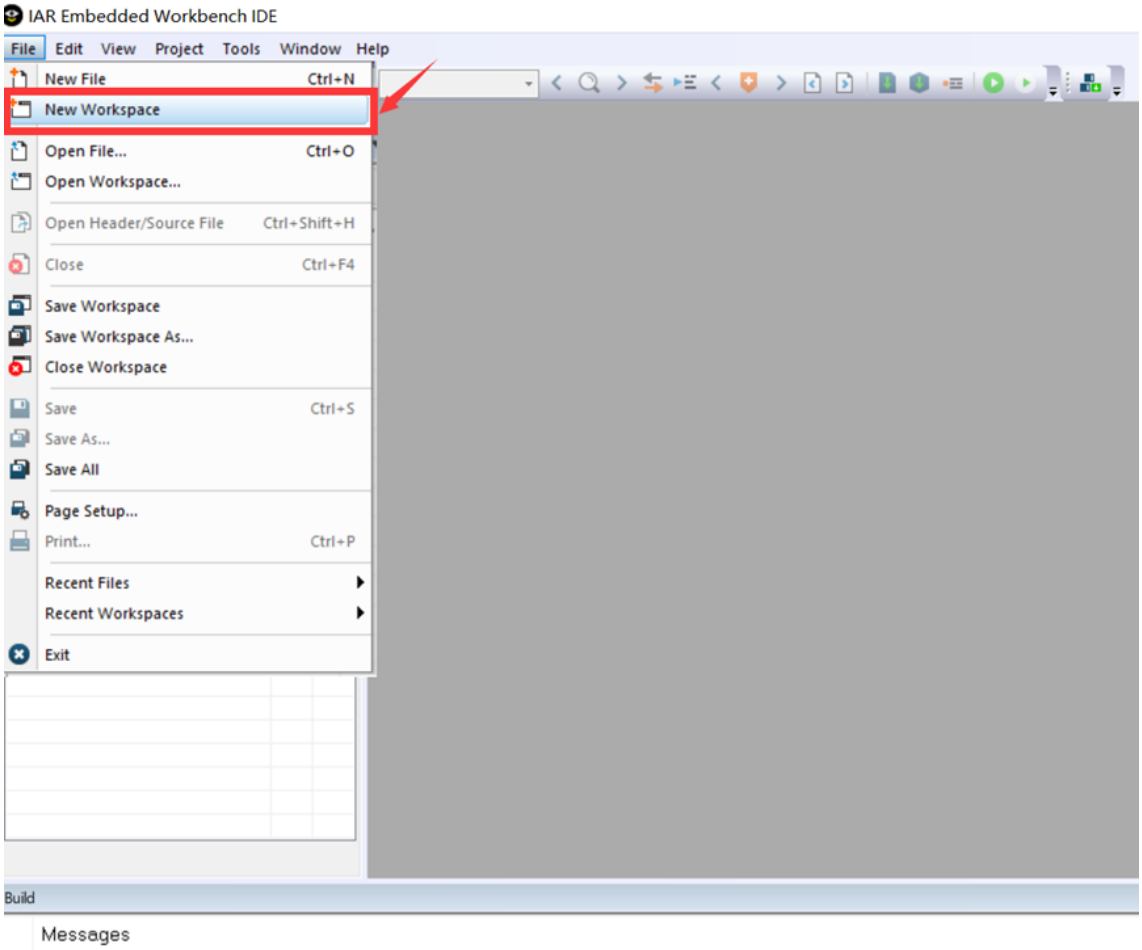
九、安装完会出现弹窗，是第 7 条选择的驱动的安装，每个弹窗直接点击"Install","close"即可，所有驱动按照默认安装即可。


PEmicro Device Drivers Setup
Choose Install Location
Choose the folder in which to install PEmicro Device Drivers.
Setup will install PEmicro Device Drivers in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\PEMicro\PEDrivers
Browse...
Space required: 20.2MB
Space available: 124.4GB
PEmicro
Install
Cancel


PEmicro Device Drivers Setup
Installation Complete
Setup was completed successfully.
Completed
Show details
PEmicro
< Back
Close
Cancel

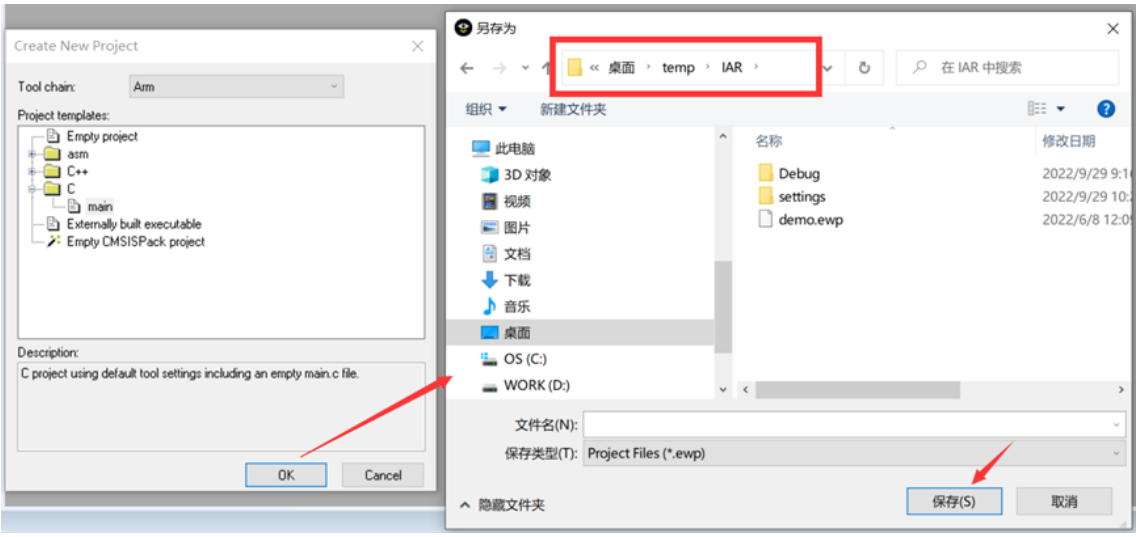
1.2 IAR 使用流程（以 ACM32F0X0 为例）

一、建立新工作区（File–>>New Workspace）



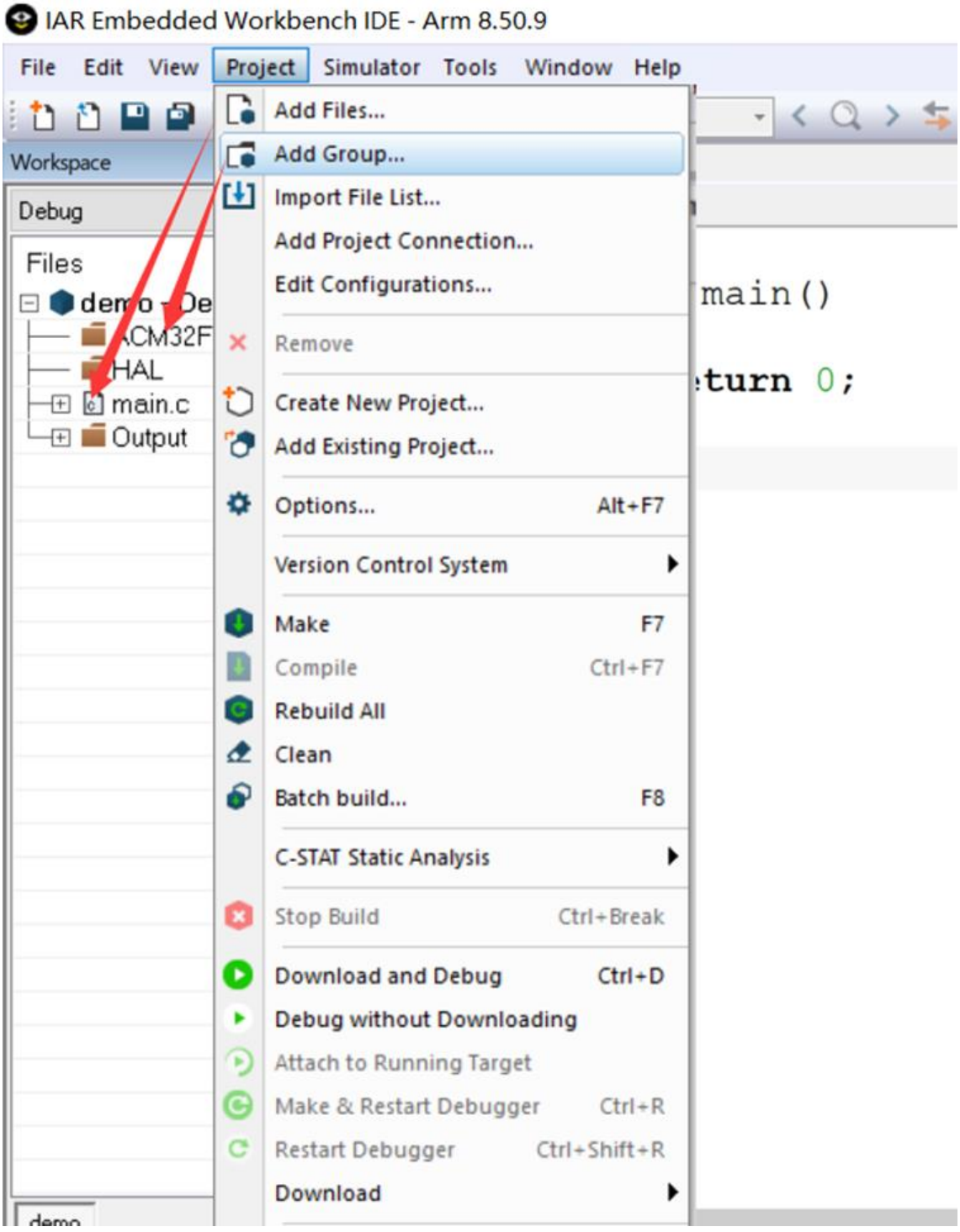
二、建立新项目（Project–>>Create New Project…）

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。



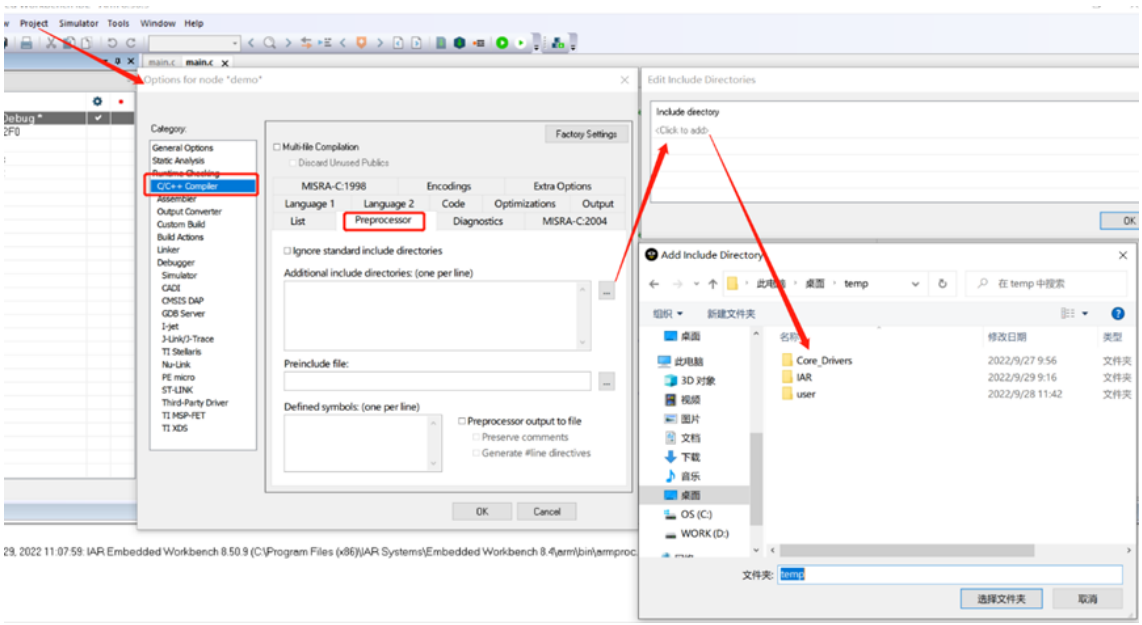
三、添加文件组和源文件

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++ Complier->>Preprocessor 添加头文件的路径。

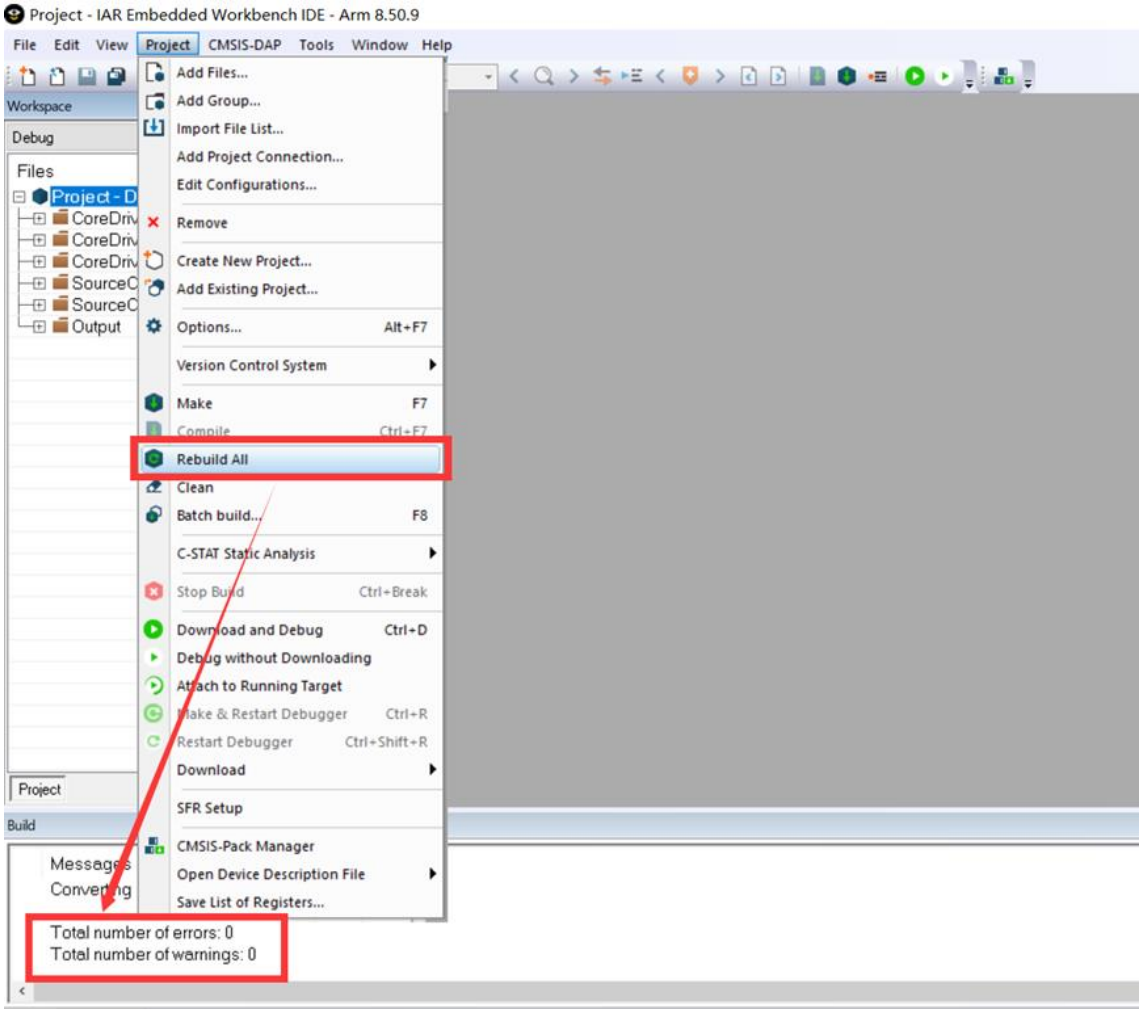


1.3 IAR 编译、下载、运行

一、编译

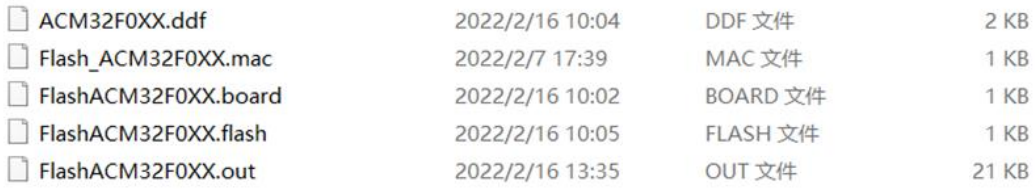
Compile：只对目前打开的 c 文件进行语法检查，不对它进行 link； Make：对工程所有打开的 c 文件进行语法检查和 link（只编译有改动的文件或者设置变动的文件）；Rebuild All：编译链接当前工程（不管文件或者设置是否有变动）；

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；

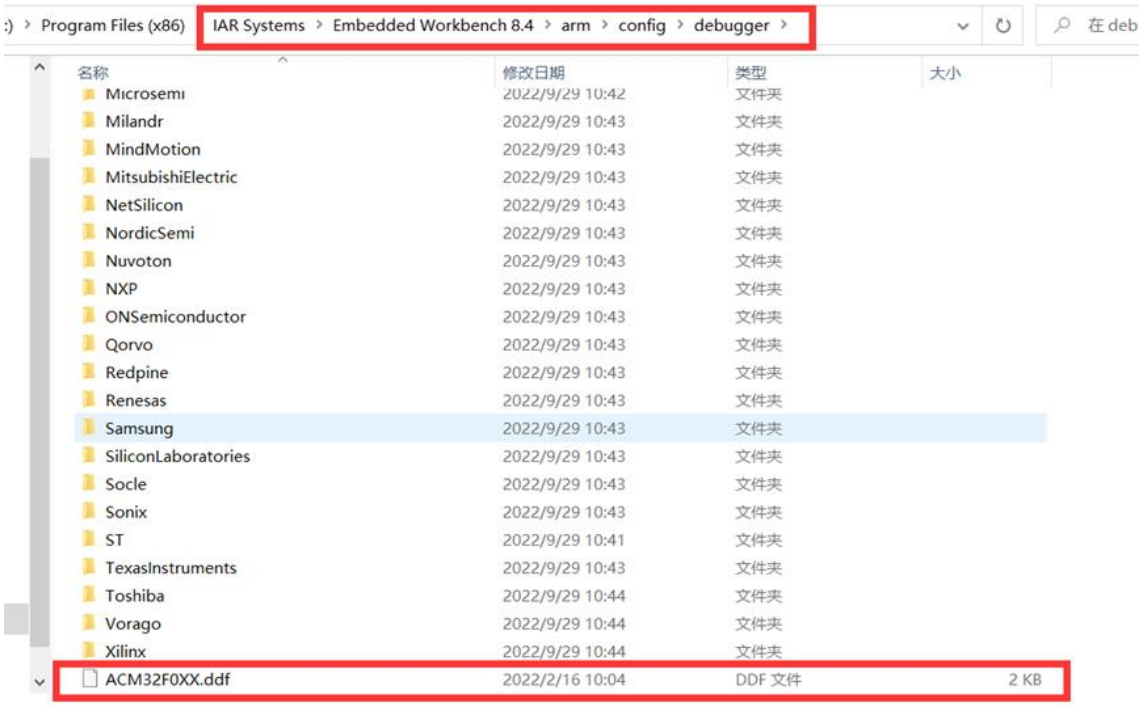


二、下载

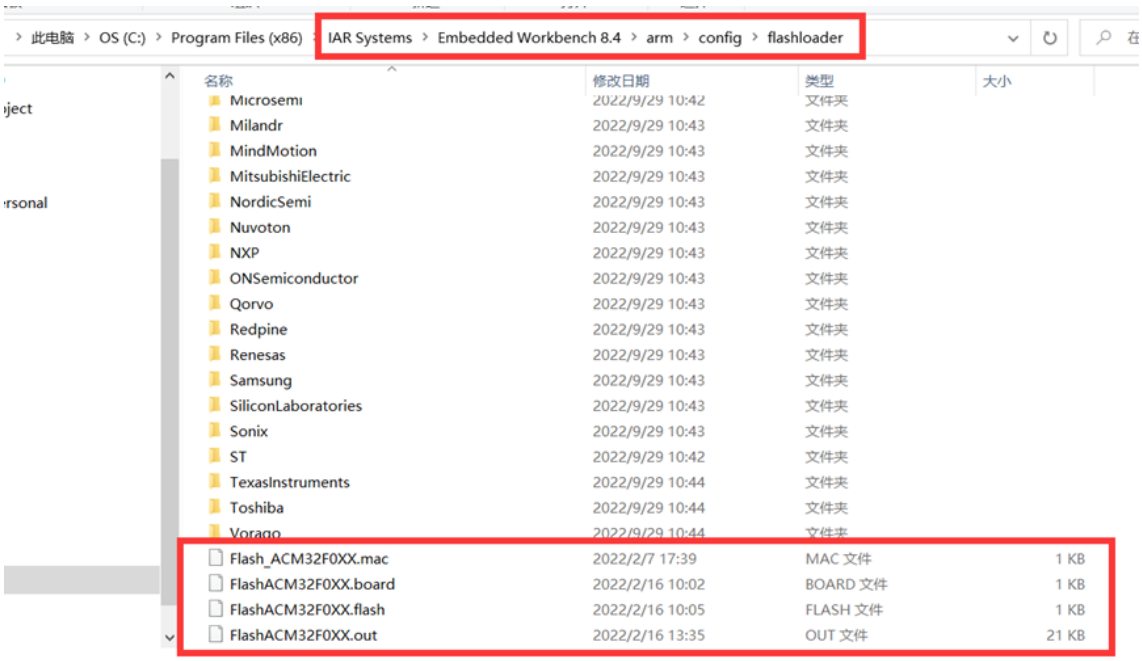
下载程序需要先安装调试文件（ACM32F4 系列也需要安装相应的调试文件）



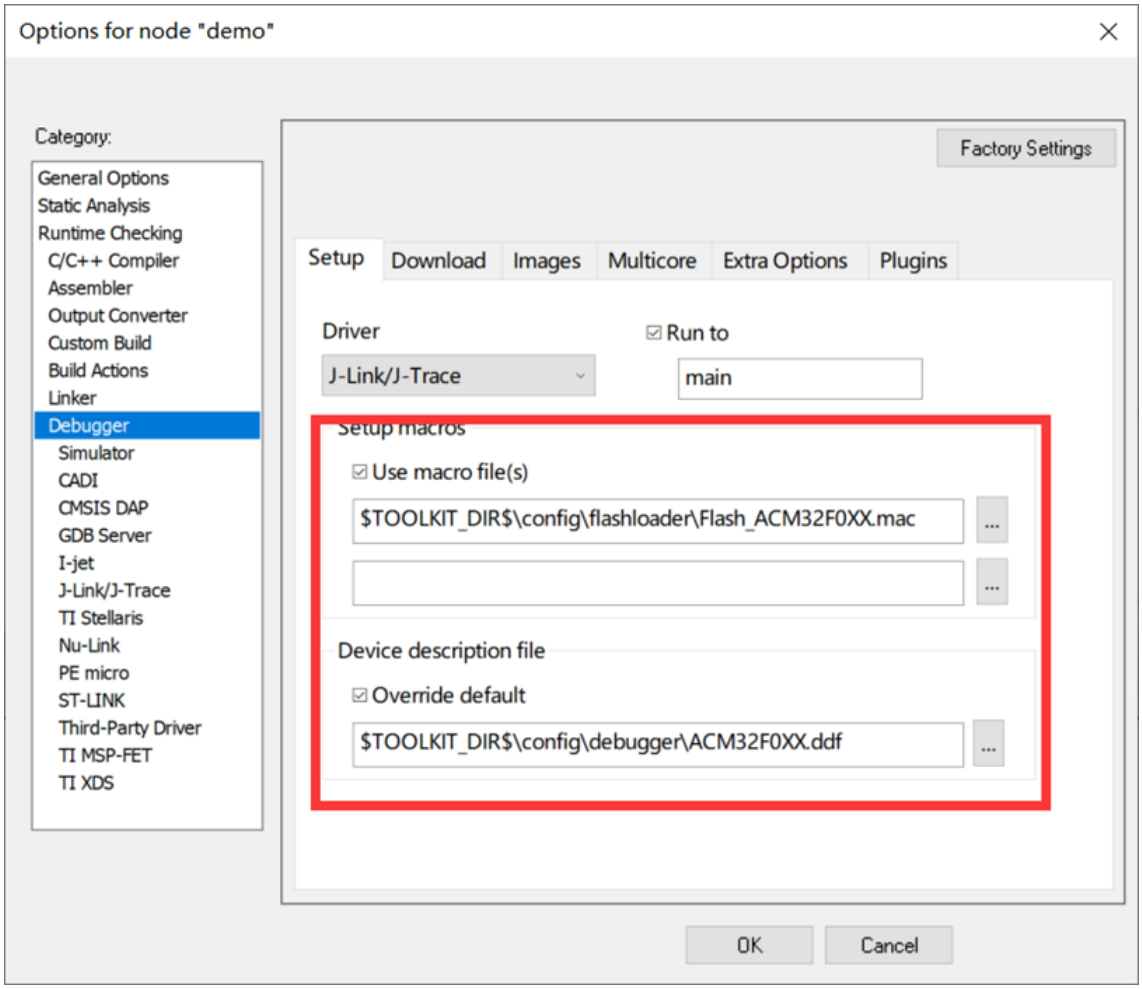
- 将 ACM32F0XX.ddf 放置于 IAR 的安装目录的\arm\config\debugger 下

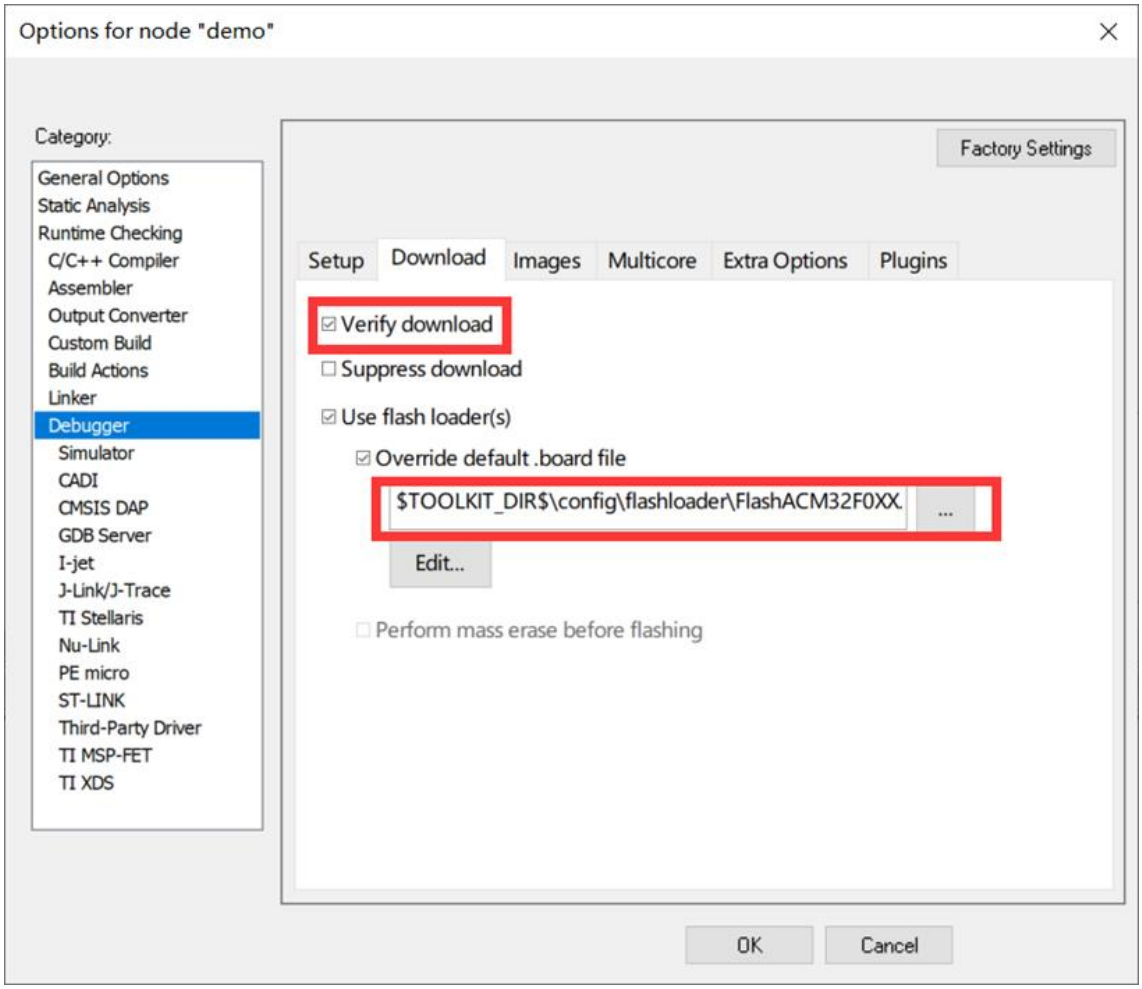


- 将其余 4 个文件放置于 IAR 的安装目录的\arm\config\flashloader 下

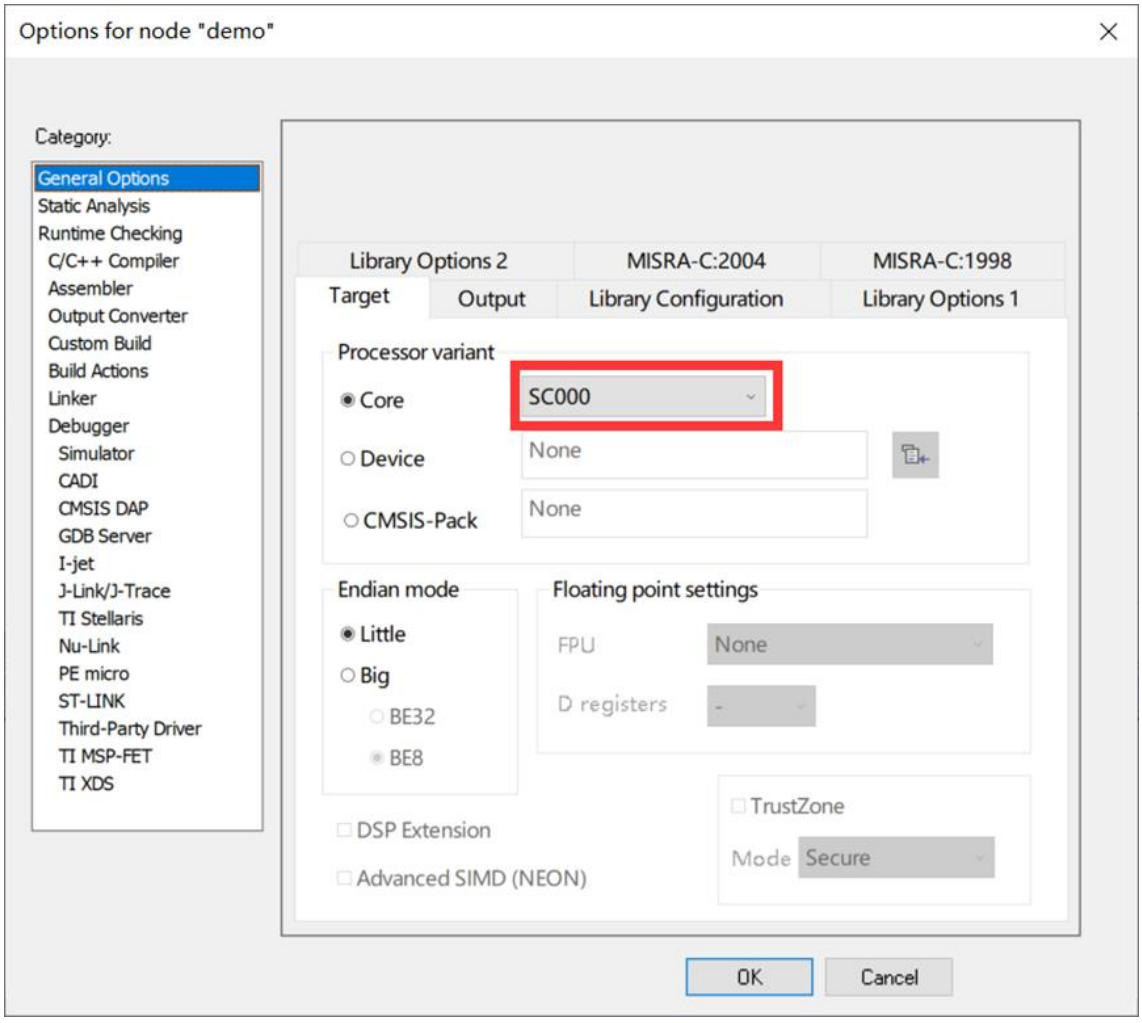


在目录中放置好后需要在 IAR 中配置调试文件

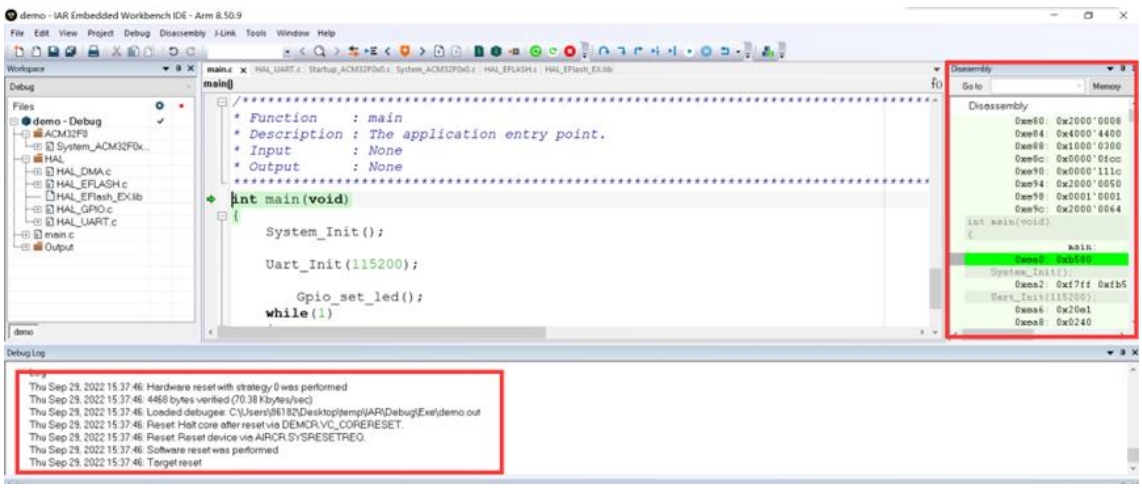

Options for node "demo"
Category:
General Options
Static Analysis
Runtime Checking
C/C++ Compiler
Assembler
Output Converter
Custom Build
Build Actions
Linker
Debugger
Simulator
CADI
CMSIS DAP
GDB Server
I-jet
J-Link/J-Trace
TI Stellaris
Nu-Link
PE micro
ST-LINK
Third-Party Driver
TI MSP-FET
TI XDS
Factory Settings
Setup
Download
Images
Multicore
Extra Options
Plugins
Driver
J-Link/J-Trace
Run to
main
Setup macros
Use macro file(s)
$TOOLKIT_DIR$\config\flashloader\Flash_ACM32F0XX.mac
Device description file
Override default
$TOOLKIT_DIR$\config\debugger\ACM32F0XX.ddf
OK
Cancel



内核选择 SC000，使用 JLINK 驱动烧录



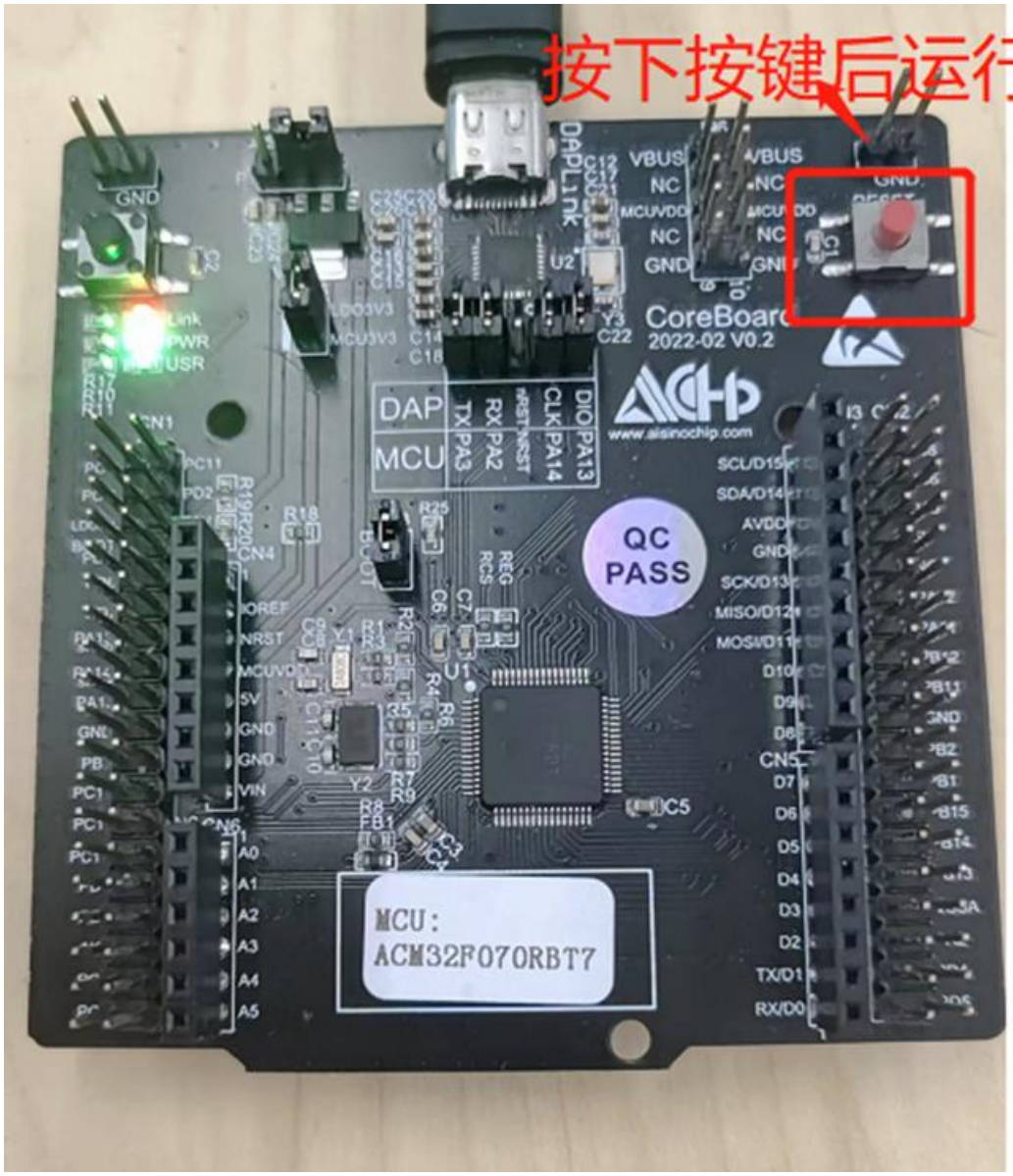
上述设置完成好后点击 Download and DeBUG,烧录成功

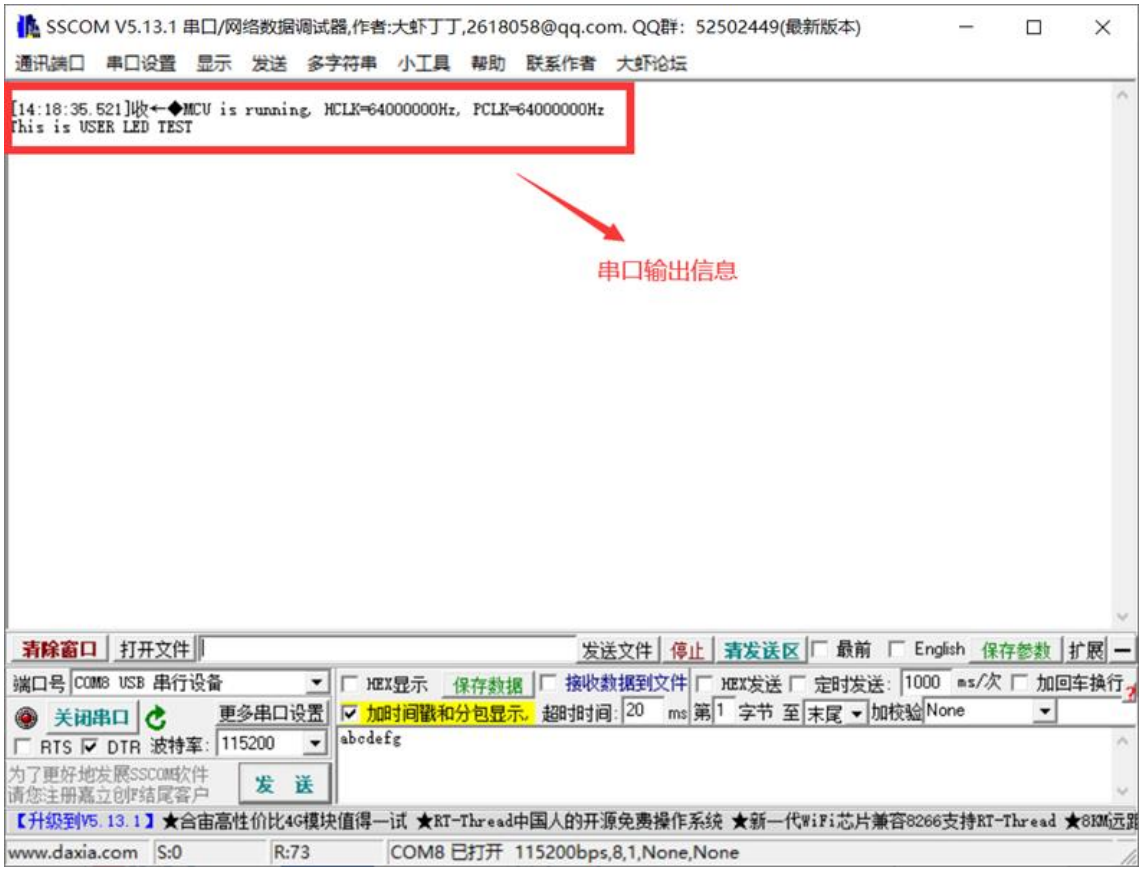


三、运行

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 GPIO 口翻转控制 LED 灯闪烁为例，部分示例代码（完整代码见附录）和现象如下：

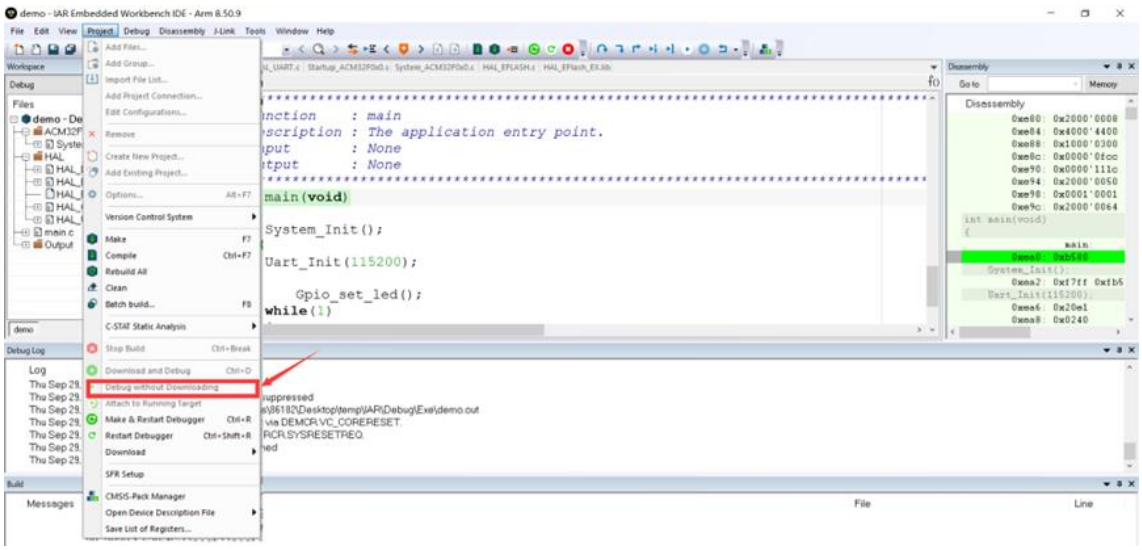
```c
void Uart_Init(uint32_t fu32_Baudrate)
{
    Uart2_Handle.Instance = UART2;
    Uart2_Handle.Init.BaudRate   = fu32_Baudrate;
    Uart2_Handle.Init.WordLength = UART_WORDLENGTH_8B;
    Uart2_Handle.Init.StopBits   = UART_STOPBITS_1;
    Uart2_Handle.Init.Parity     = UART_PARITY_NONE;
    Uart2_Handle.Init.Mode       = UART_MODE_TX_RX_DEBUG;
    Uart2_Handle.Init.HwFlowCtl  = UART_HWCONTROL_NONE;

    HAL_UART_Init(&Uart2_Handle);

    /* UART_DEBUG_ENABLE control printfS */
    printfS("MCU is running, HCLK=%dHz, PCLK=%dHz\n", System_Get_SystemClock(), System_Get_APBClock());
}
void Gpio_set_led()
{
    printfS("This is USER LED TEST \r\n");

    GPIOD_Handle.Pin       = GPIO_PIN_3;
    GPIOD_Handle.Mode      = GPIO_MODE_OUTPUT_PP;
    GPIOD_Handle.Pull      = GPIO_PULLUP;
    GPIOD_Handle.Alternate = GPIO_FUNCTION_0;

    HAL_GPIO_Init(GPIOD, &GPIOD_Handle);

    while (1)
    {
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_3, GPIO_PIN_SET);
        System_Delay_MS(500);
        HAL_GPIO_WritePin(GPIOD, GPIO_PIN_3, GPIO_PIN_CLEAR);
        System_Delay_MS(500);
    }
```


按下按键后运行
QC
PASS
MCU:
ACM32F070RBT7
CoreBoard
2022-02 V0.2
www.aisinochip.com
DAPLink
DAP
MCU
VBUS
NC
MCUVDD
GND
PWR
USR
IOREF
NRST
5V
VIN
A0
A1
A2
A3
A4
A5
D2
D3
D4
D5
D6
D7
SCK/D13
MISO/D12
MOSI/D11
SCL/D15
SDA/D14
AVDD
TX/D1
RX/D0

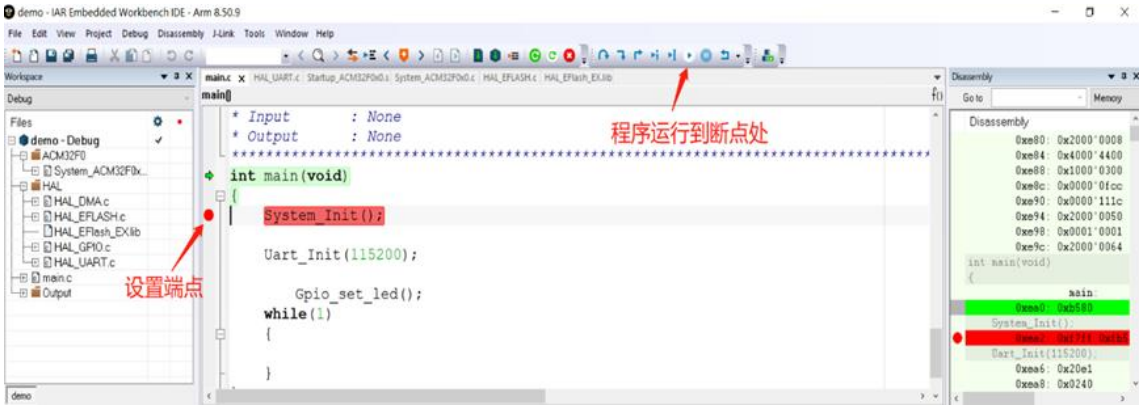


1.4 IAR Debug 使用说明

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



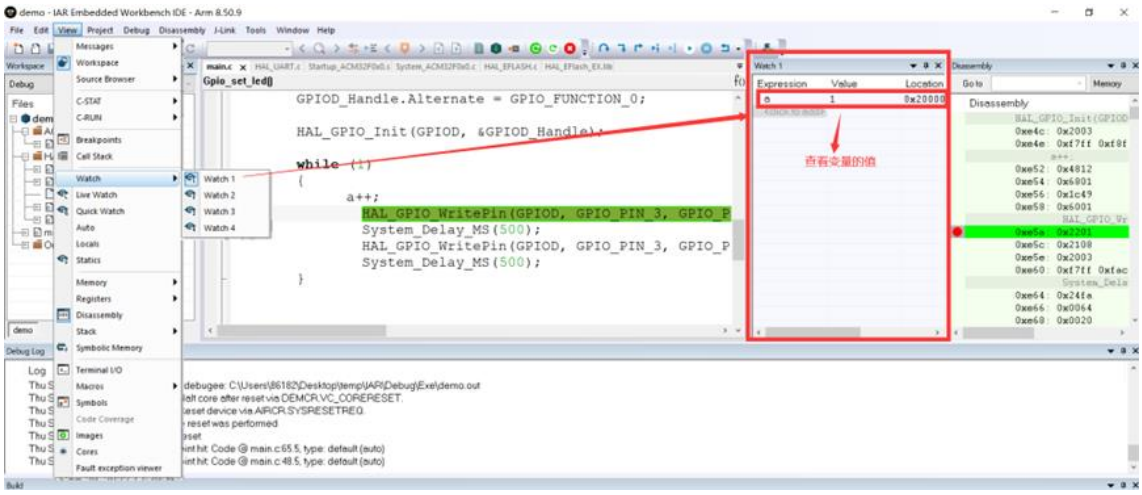
二、设置断点



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

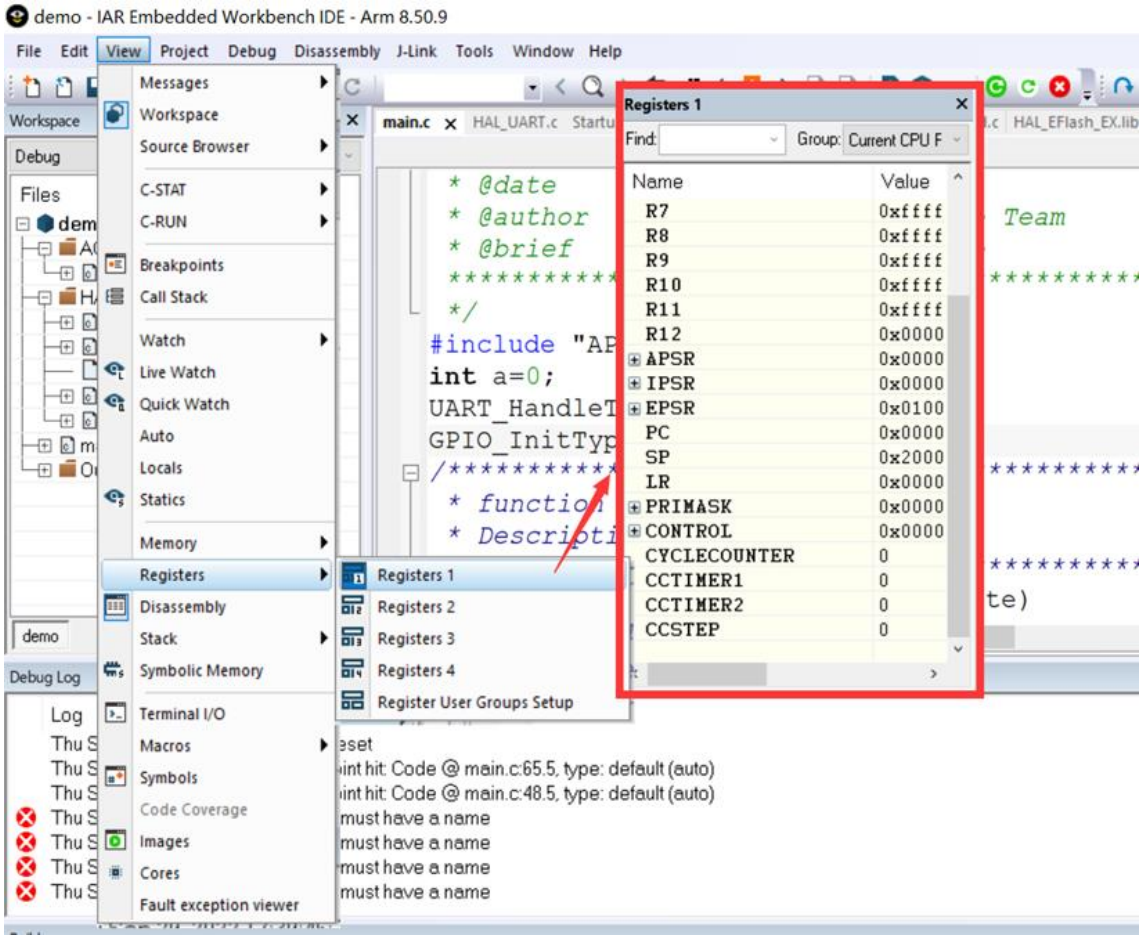
三、查看变量（Watch 窗口），

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。



Keil+GCC

1.1 Keil 安装及使用

具体安装及使用相关信息参照 `Keil` 篇，本篇不做具体赘述。

1.2 GNU 编译器集合 (GCC) 安装步骤

ARM GCC 编译器下载地址 https://developer.arm.com/downloads/-/gnu-rm

Downloads: 10.3-2021.10 October 21, 2021

What's new in 10.3-2021.10

This release provides mitigation for the VLLDM instruction security vulnerability.

In this release:

1. gcc-arm-none-eabi-10.3-2021.10-win32.exe
   Windows 32-bit Installer (Signed for Windows 10 and later) (Formerly SHA2 signed binary)
   MD5: 8d0f75f33f9e3d5f9600197626297212

2. gcc-arm-none-eabi-10.3-2021.10-win32.zip
   Windows 32-bit ZIP package
   MD5:2bc8f0c4c4659f8259c8176223eeafc1

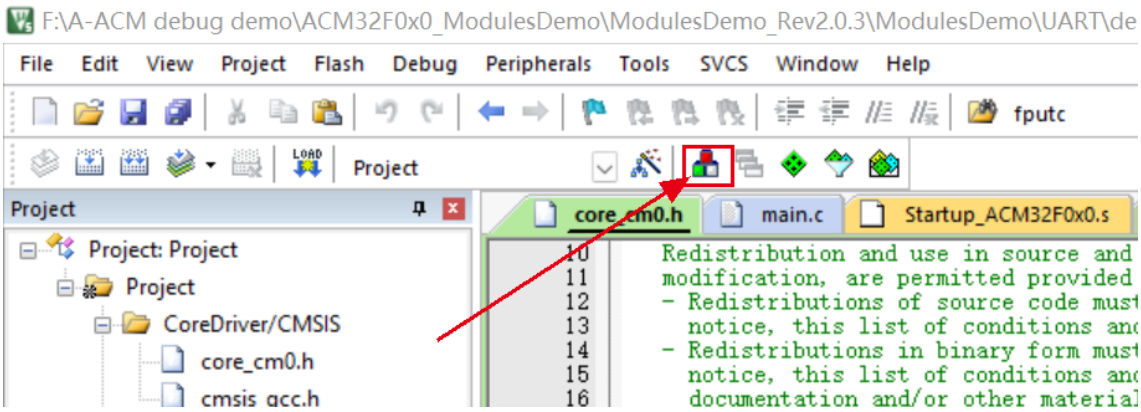
①使用安装文件 exe 安装时，默认安装目录为 C:\Program Files (x86)\GNU Arm Embedded Toolchain\10 2021.10;

②使用压缩包 zip 文件时，将文件放入 C:\Keil_v5\ARM 文件夹内解压，安装目录为 C:\Keil_v5\ARM\gcc-arm-none-eabi-10.3-2021.10

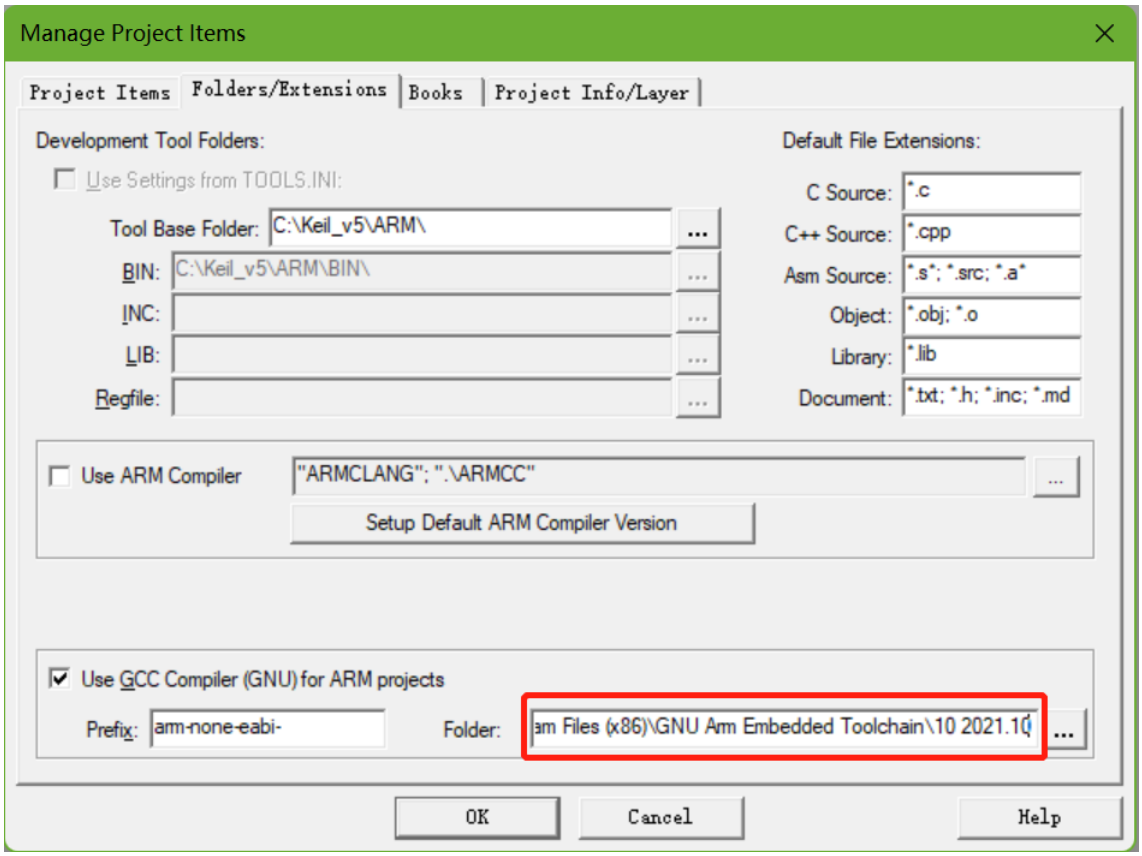
安装目录后续配置 Keil 时使用

1.3 Keil + GCC 相关配置

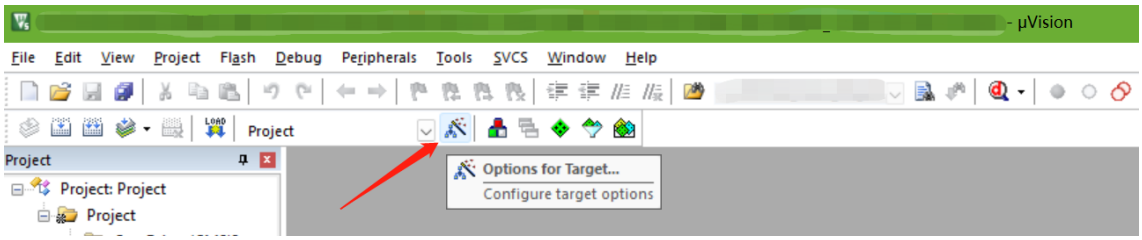
①打开 Keil 后，点击图示按钮 Manager Project Items



②在弹出界面中，Folder/Extensions 选项中选择 Use GCC Compiler (GNU) for ARM projects，Folder 填入 GCC 安装目录

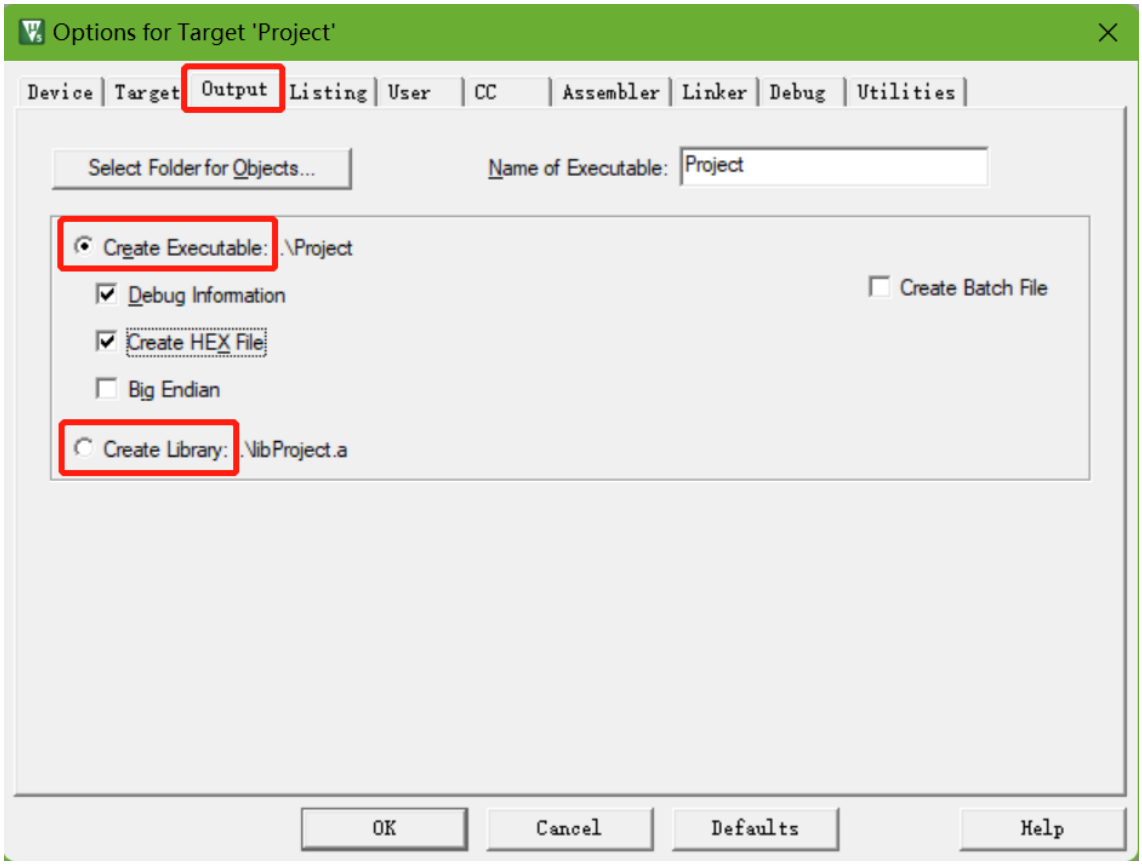


③点击图示按钮 Configure target options



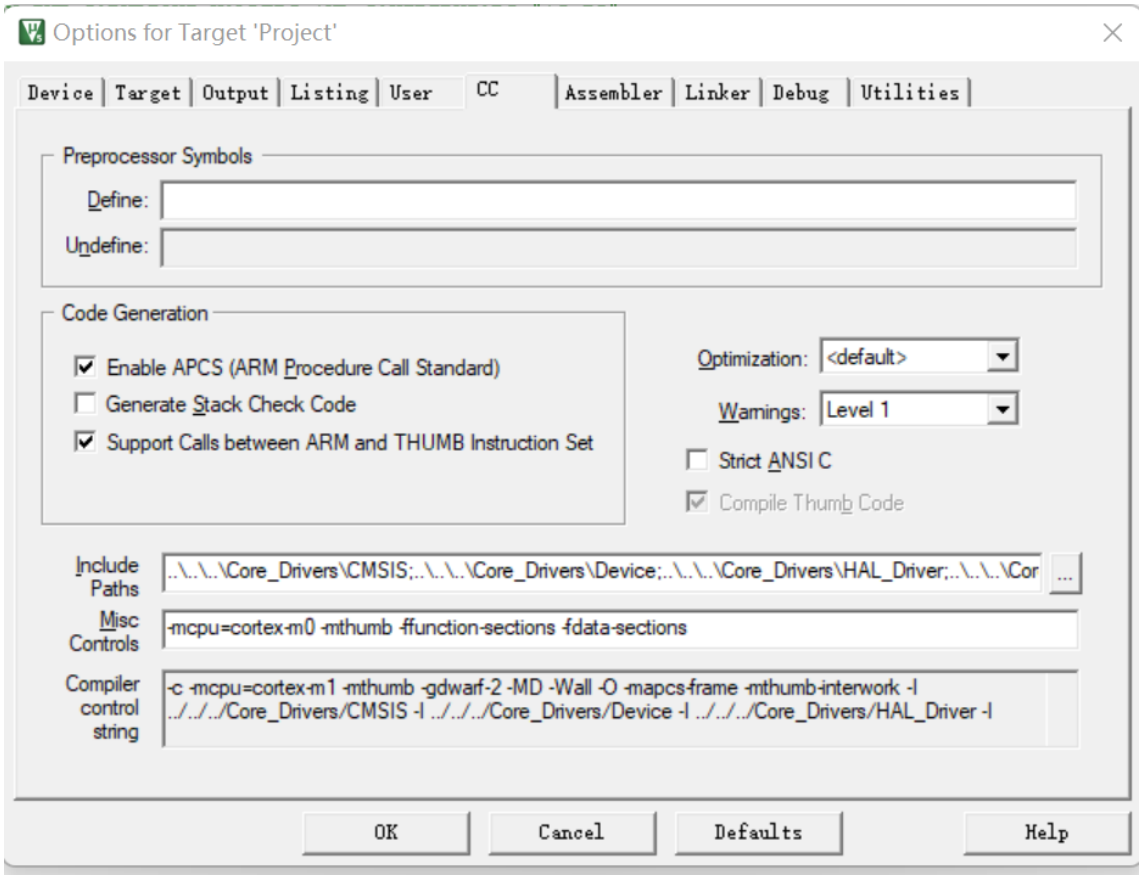
④在弹出界面中，Output 选项中，如果选择 Creat Executable，则需要在 Linker

中配置脚本文件，如果选择 Creat Library 则不需要在 Linker 中配置脚本文件。

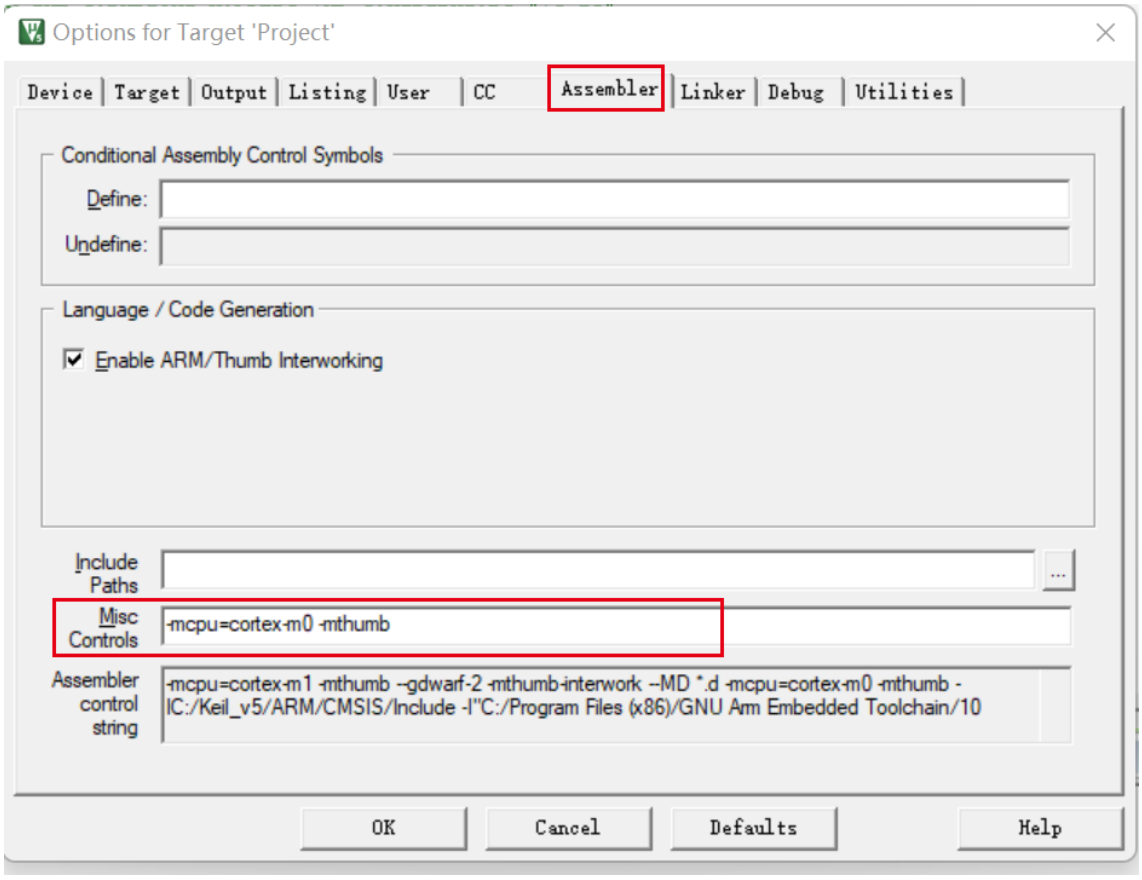


⑤配置 CC 选项，Include Paths 添加 CMSIS、Device、HAL 库等相关路径

Misc Controls 中 F0 系列添加 -mcpu=cortex-m0 -mthumb -ffunction-sections -fdata-sections。

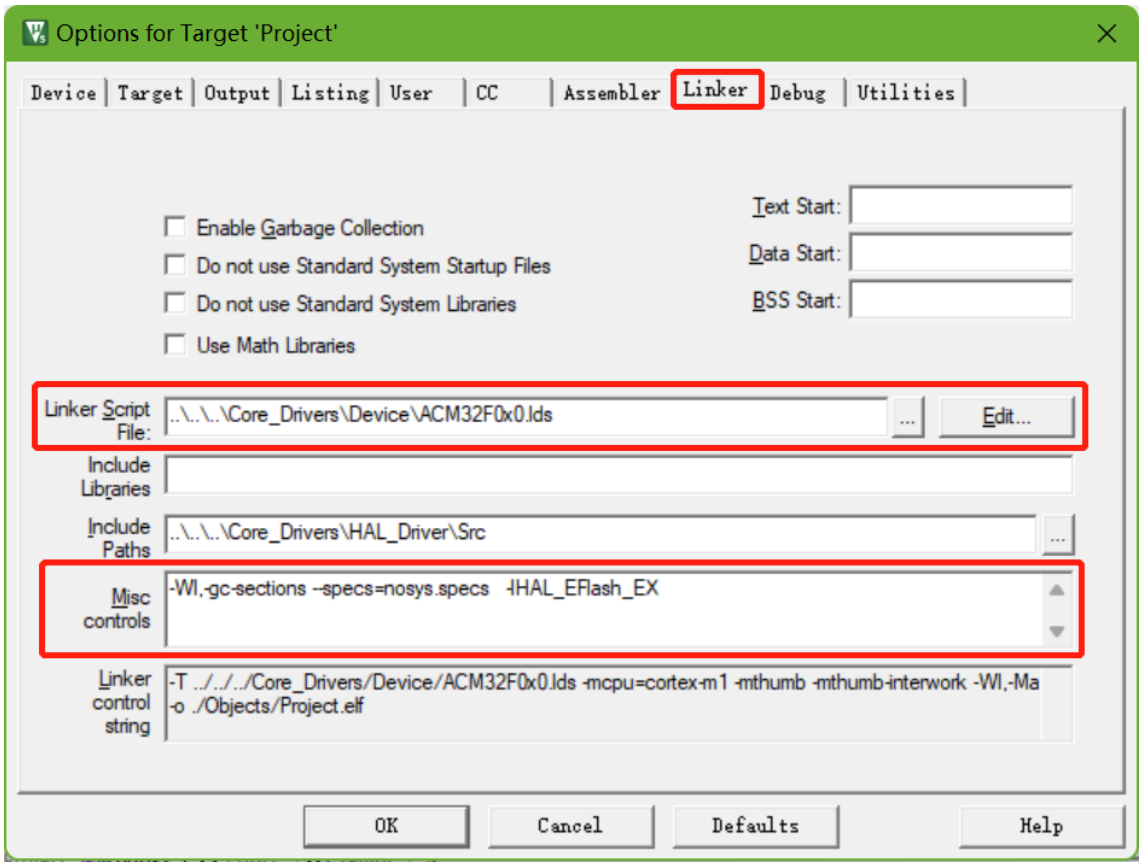


⑥配置 Assmbler 选项，Misc Controls F0 系列添加-mcpu=cortex-m0 -mthumb。

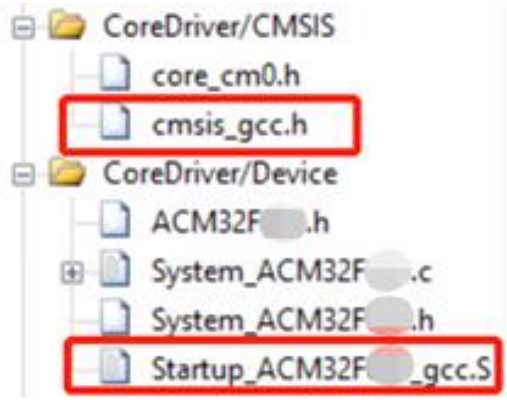


⑦配置 Linker 选项，如步骤④中所述，仅在选择 Creat Executable 时需要在 Linker Script File 中配置脚本文件，建议脚本文件包含在项目工程根目录中。

Misc Controls 添加-Wl,–gc-sections ，如需使用封装库增加<-l 库名>来链接封装库



⑧项目创建或移植时，CMSIS 添加 cmsis_gcc.h 文件，Device 更新 GCC 专用的.S 启动文件(Startup_ACM32F0x0_gcc.S)



1.4 Keil + GCC 调试使用说明

调试过程参照 `Keil` 篇，需要注意的是使用 GCC 编译器时，在 Keil 环境下无法进行跳转（快捷键 F12）功能。

常见错误和注意事项

一、在 GCC 环境下，printf 重定向问题

在 Keil 中的 C 库中，printf、scanf 等输入输出流函数是通过 fputc、fgetc 来实现最底层操作的，所以我们只需要在我们的工程中重定义这两个函数的功能就可以实现 printf、scanf 等流函数的重映射。与 keil C 库类似 GNU C 库下的流函数底层是通过 _read、_write 函数实现的，我们只要在工程中将他们重新定义就可以实现重映射的功能了。

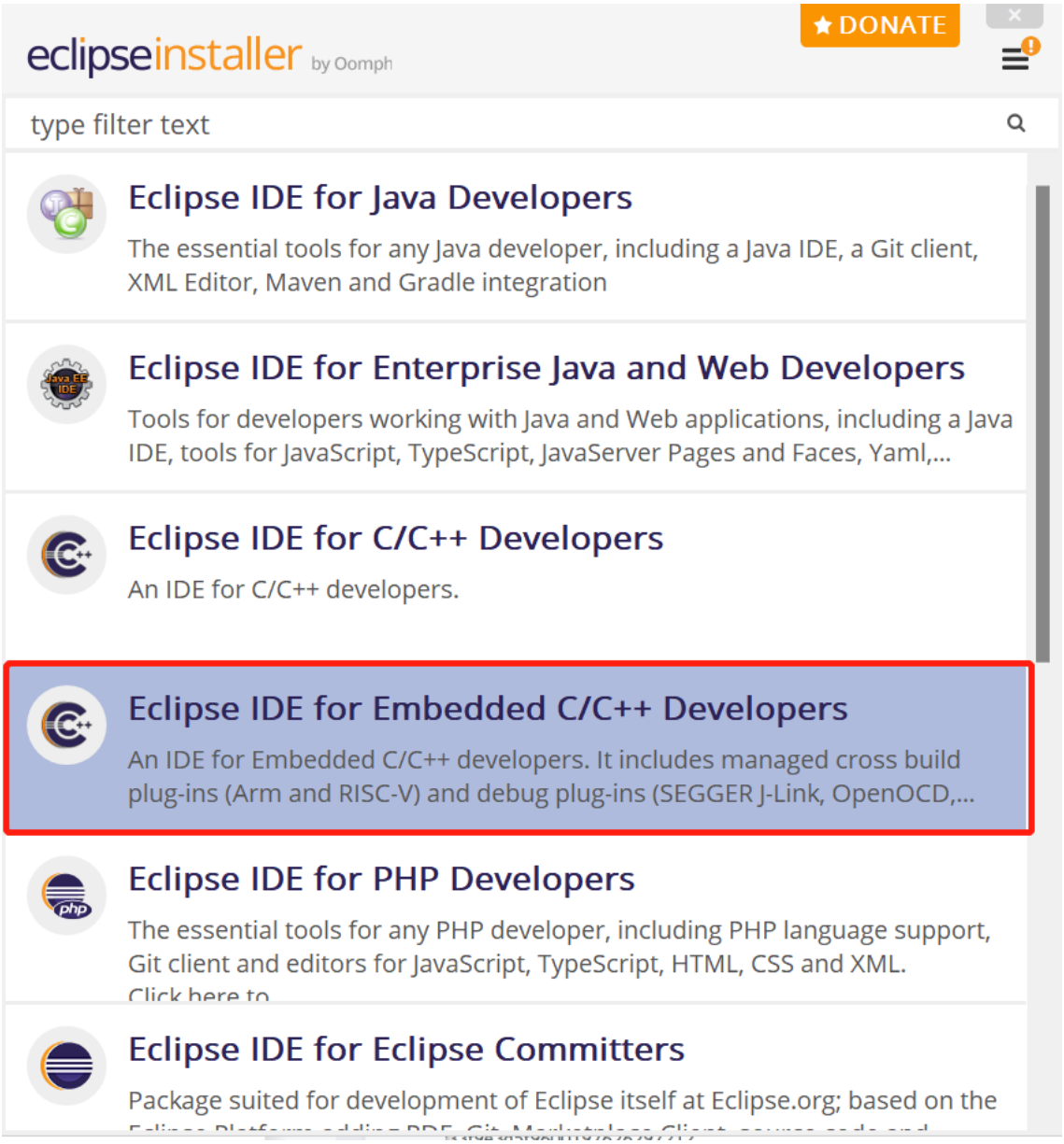
```C++
#ifdef __GNUC__
int __io_putchar(int ch)
{
    Uart_Debug->DR = ch;
    while ((Uart_Debug->FR & UART_FR_BUSY));
    return ch;
}
int _write(int fd, char* buffer, int size)
{
    for(int i = 0;i<size;i++)
    {
        __io_putchar(*buffer++);
    }
    return size;
}
#else
int fputc(int ch, FILE *f)
{
    if (Uart_Debug == NULL)
    {
        return 0;
    }
    Uart_Debug->DR = ch;
    while ((Uart_Debug->FR & UART_FR_BUSY));
    return ch;
}
#endif
```

Eclipse+GCC

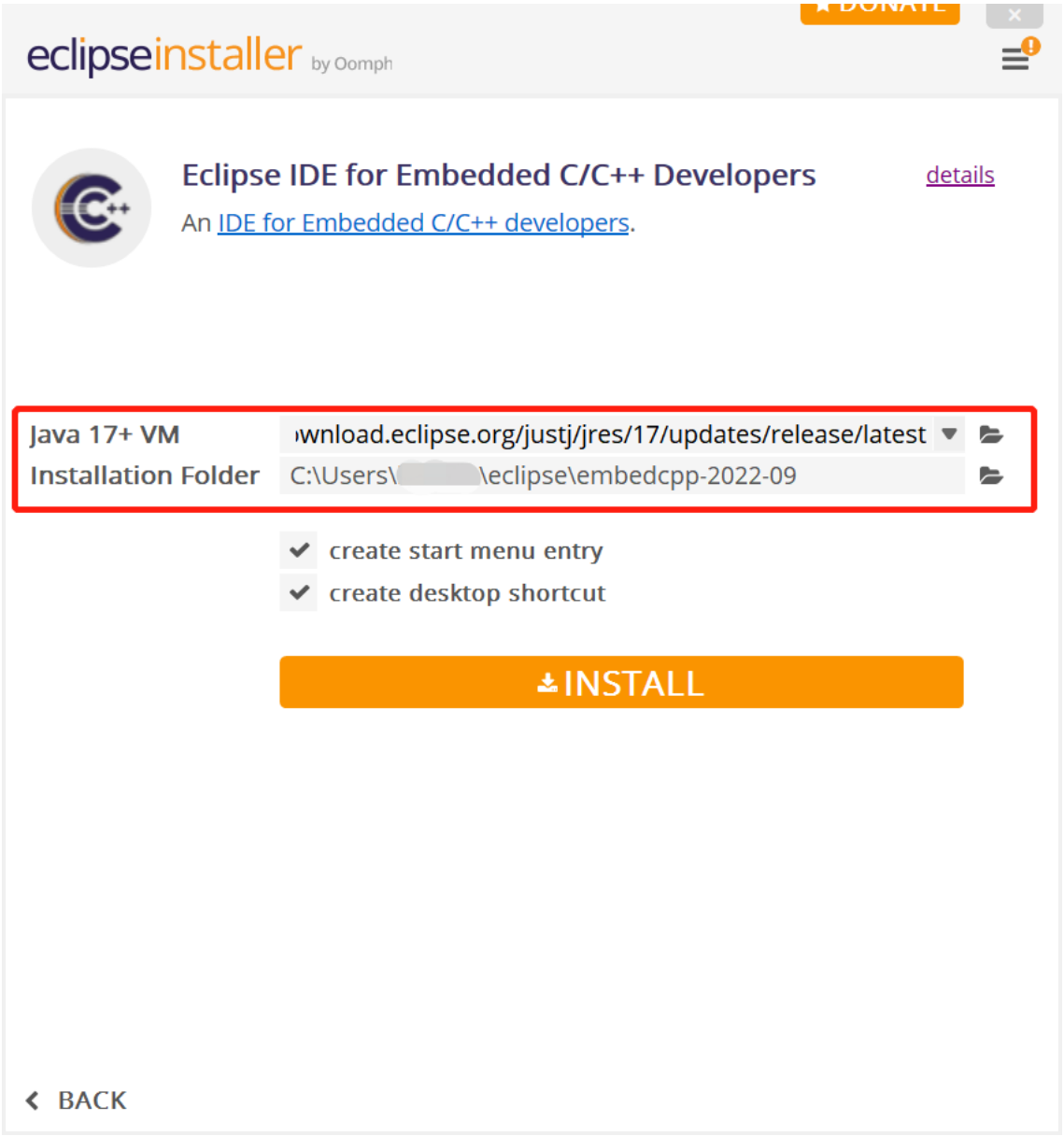
1.1 Eclipse 安装

①Eclipse 下载地址 https://www.eclipse.org/downloads/ 整体安装过程需联网进行

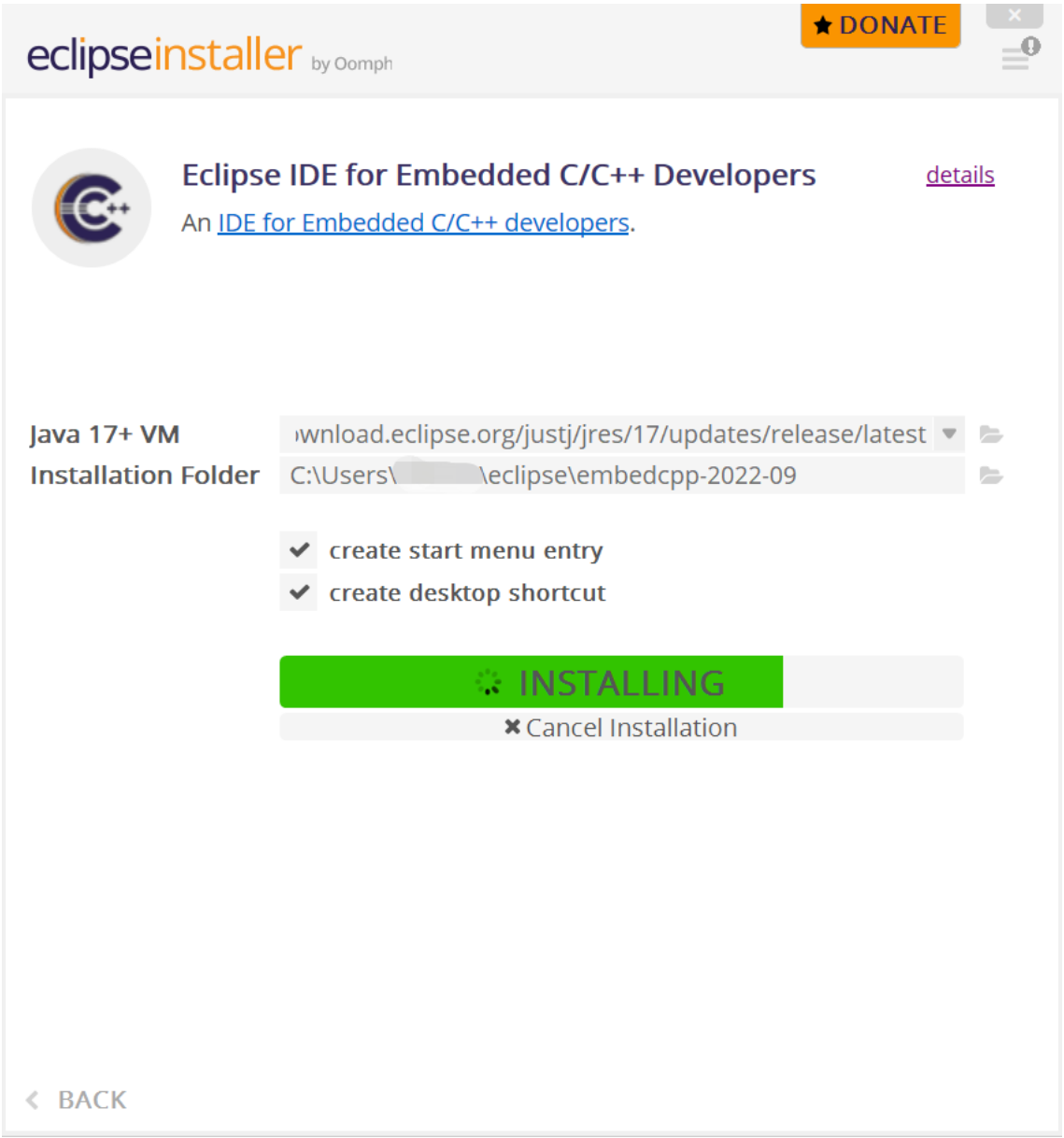
②下载完成后双击打开，界面如下图所示，点击红色框标记的选项



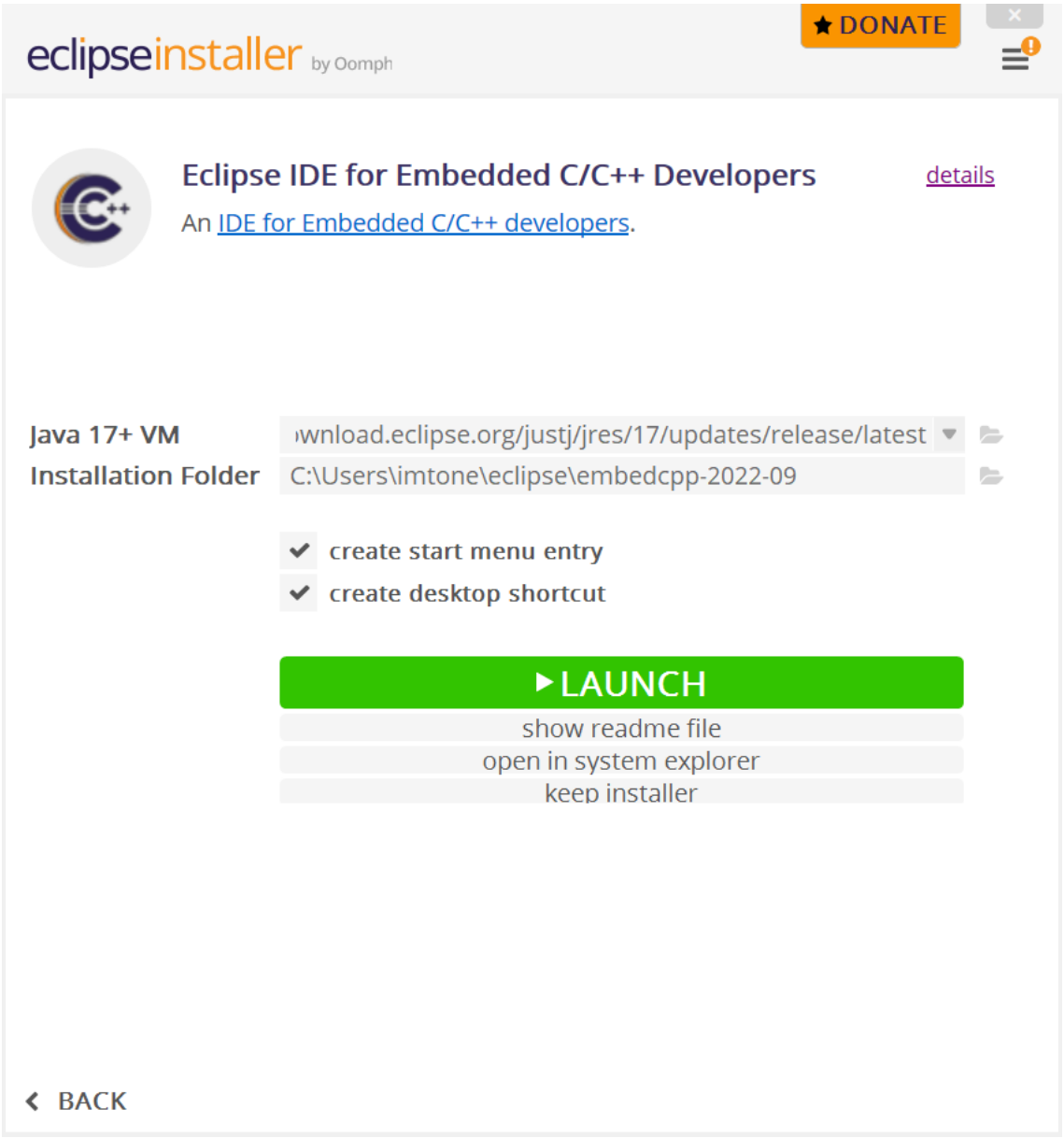
③点击后出现下图设定界面，两个设定项目分别是 JRE 的版本与安装的位置，可自行选定



④点击安装后，开始进入到下载流程，此时需全程联网进行

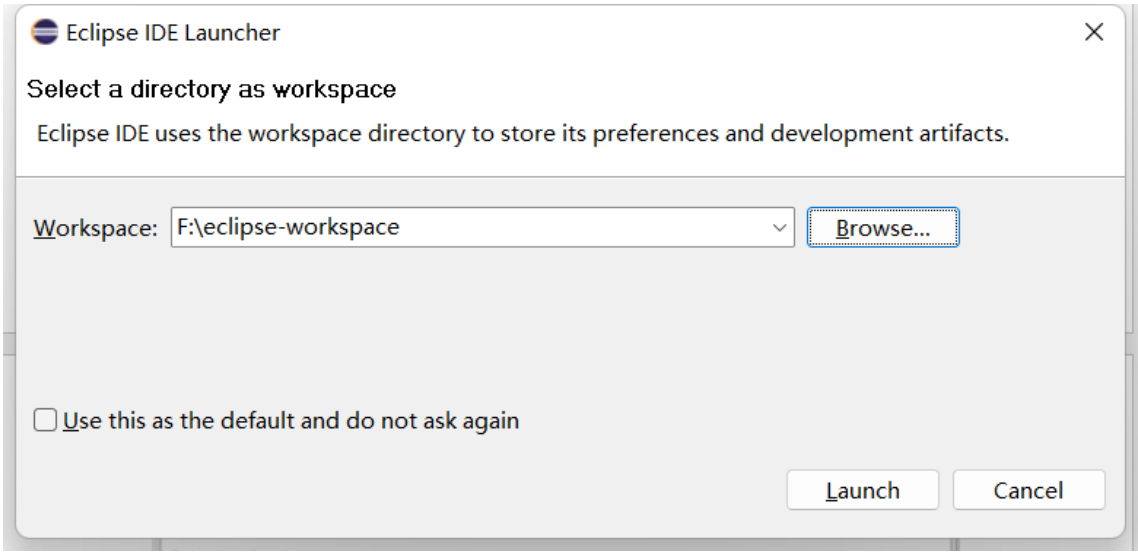


⑤安装完成后，点击 LAUNCH 启动 eclipse，开始插件安装



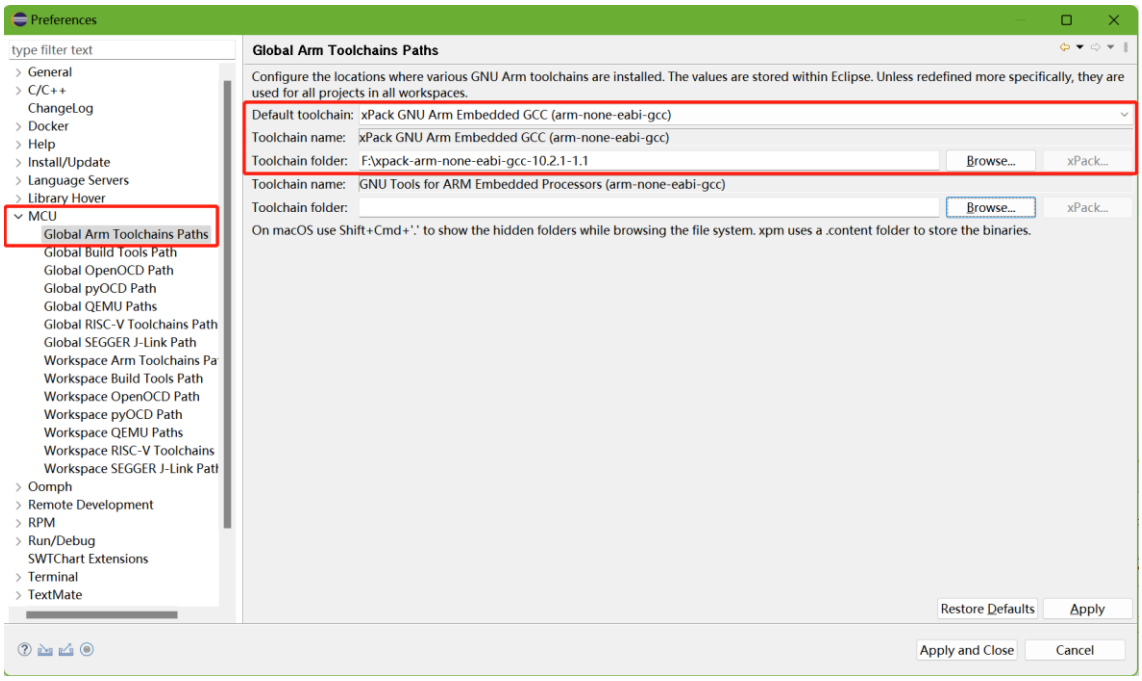
1.2 Eclipse 使用

打开 eclipse 软件，会弹出对话框让选择 workspace，界面如下图所示：

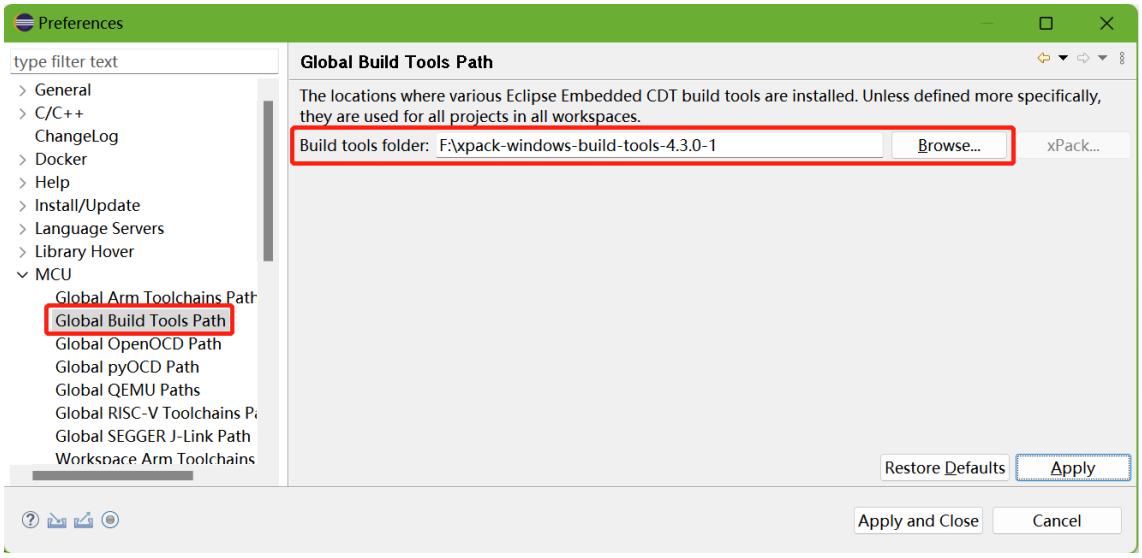


Workspace 由用户自己定义，用来存储相关文件。

ACM32 系列 MCU 是 ARM 内核，所以需要下载一个支持编译 ARM 内核的编译器，示例选择了 xpack 模式的编译器，下载地址 xpack-arm-none-eabi-gcc-11.2.1-1.2-win32-x64



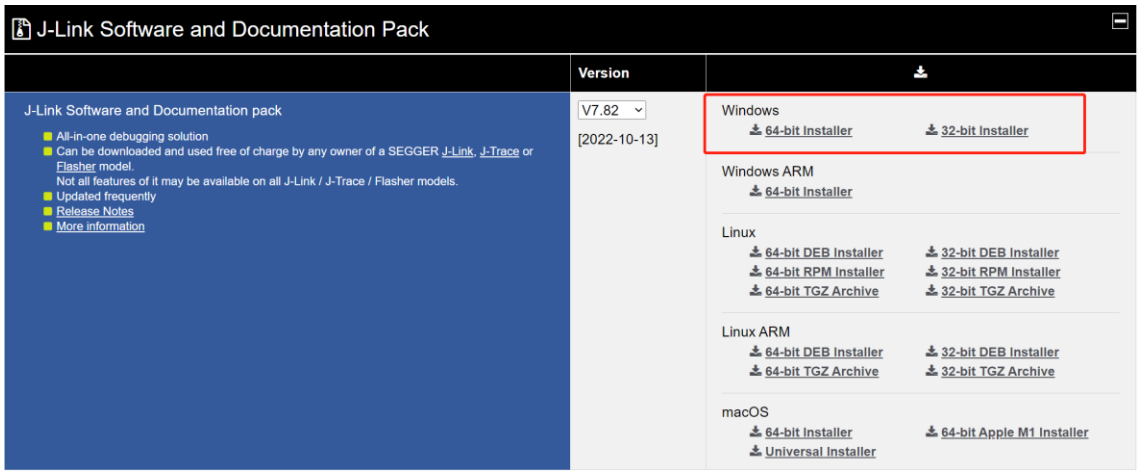
编译时有清理或拷贝的操作，需要使用到 xPack Windows Build Tools 中的一些工具，同编译工具链设定相似，软件包下载完成后解压到指定的目录下，依次点击 Windows->Preferences->MCU->Global Build Tool Path 进入到设定中，指定构建工具的目录



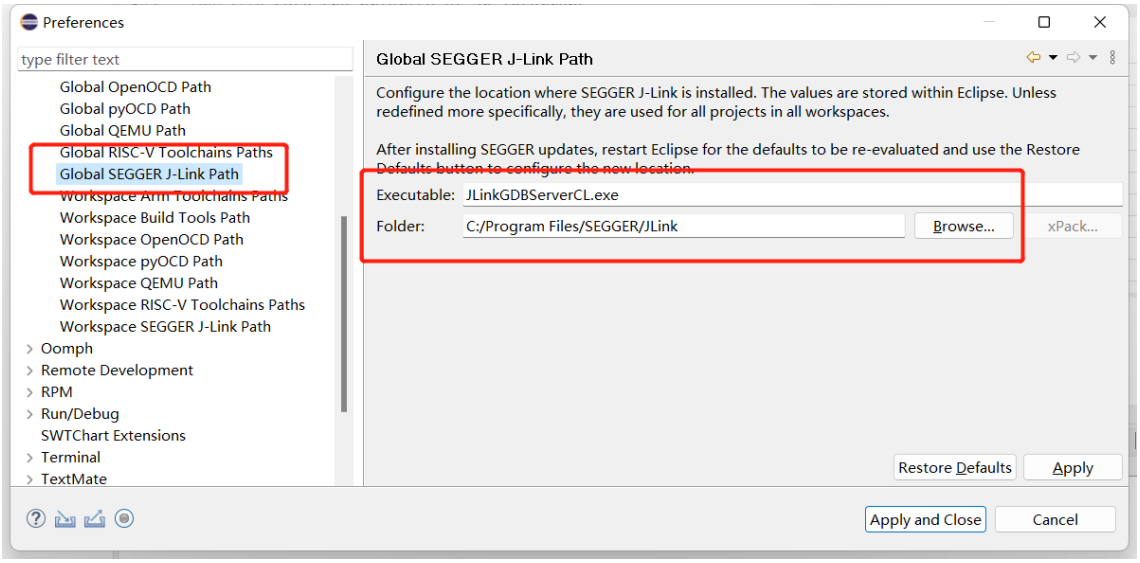
1.3 调试工具驱动安装

1.3.1 J-Link

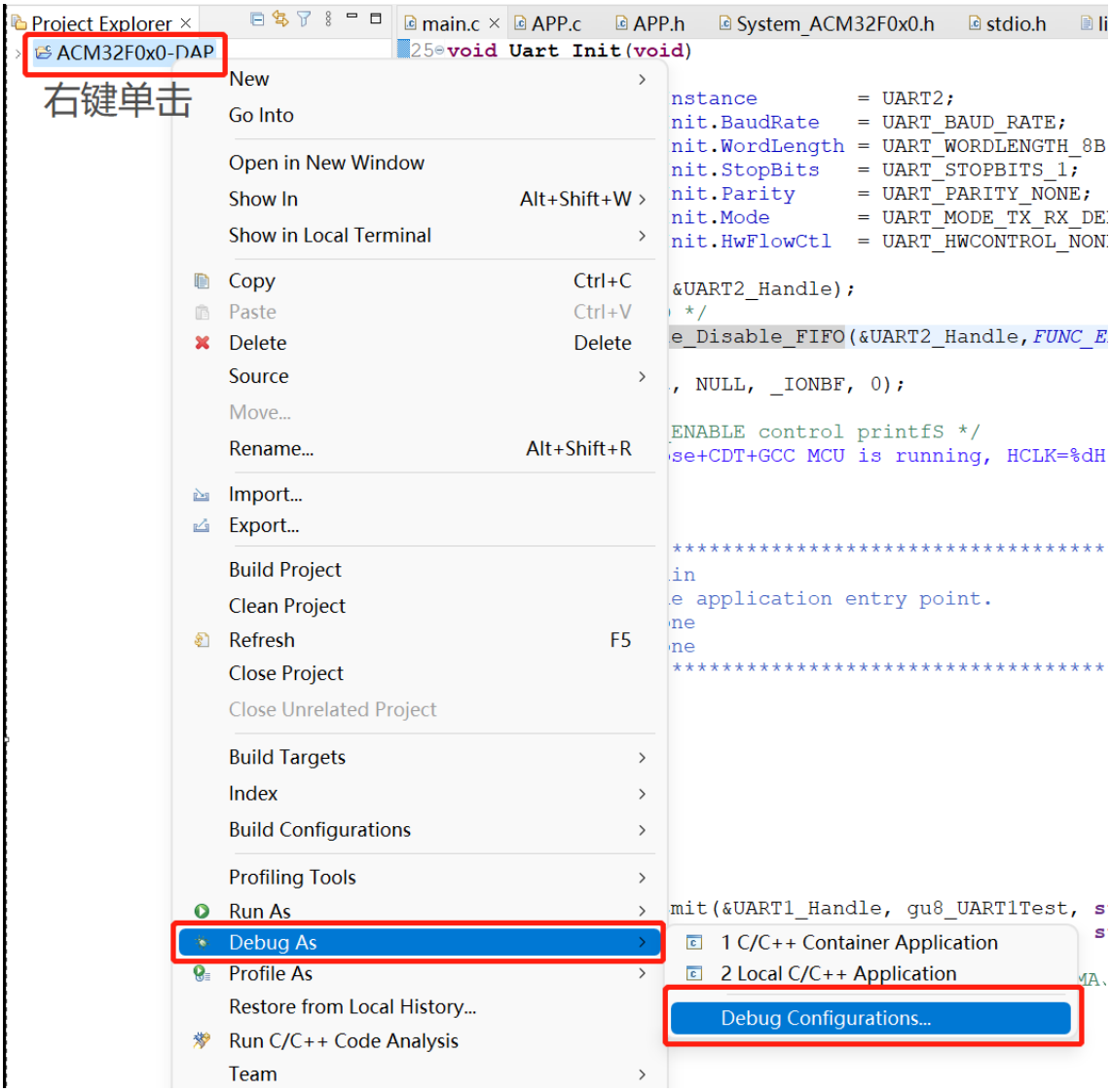
①下载 J-Link 软件，访问 https://www.segger.com/downloads/jlink#可选择不同版本 J-Link 驱动下载，本文中选取下载了 J-Link V7.82 版本



②安装完成后，依次点击 eclipse 的 Windows->Preferences->MCU->Global SEGGER J-Link Path 进入到设定中，指定 JLink 工具的目录，图示如下:

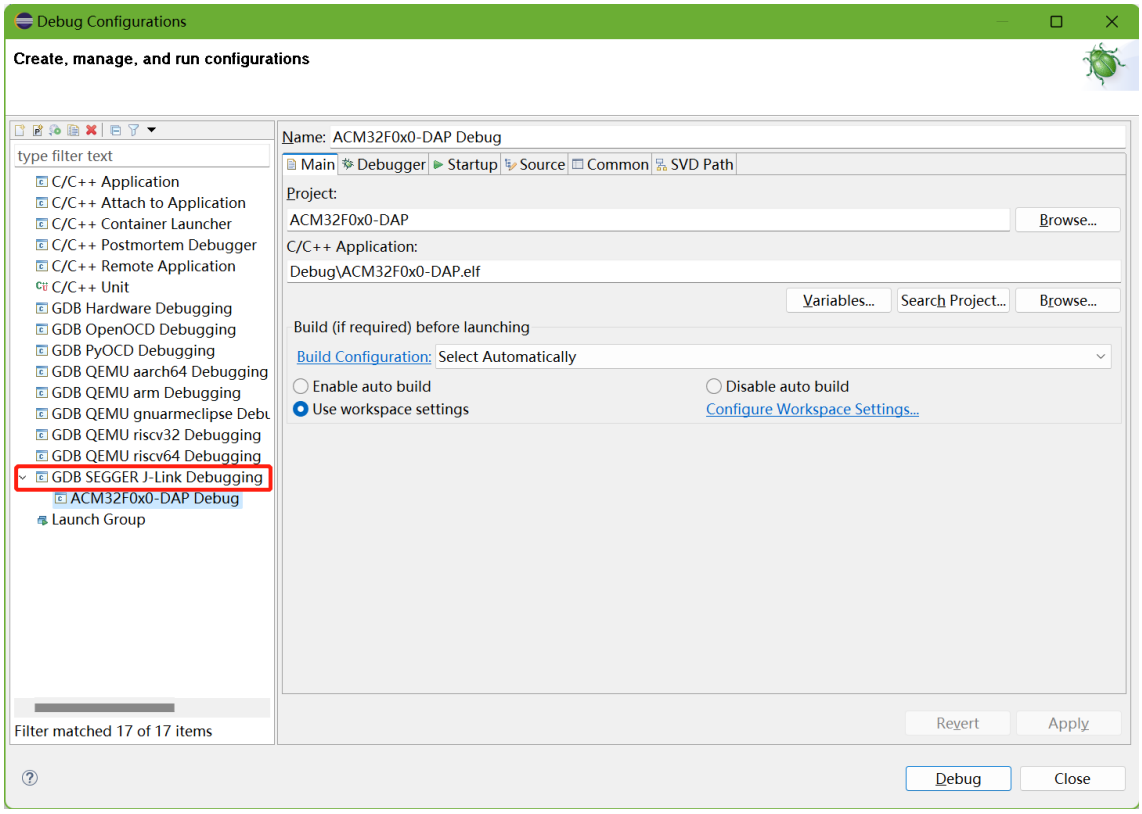


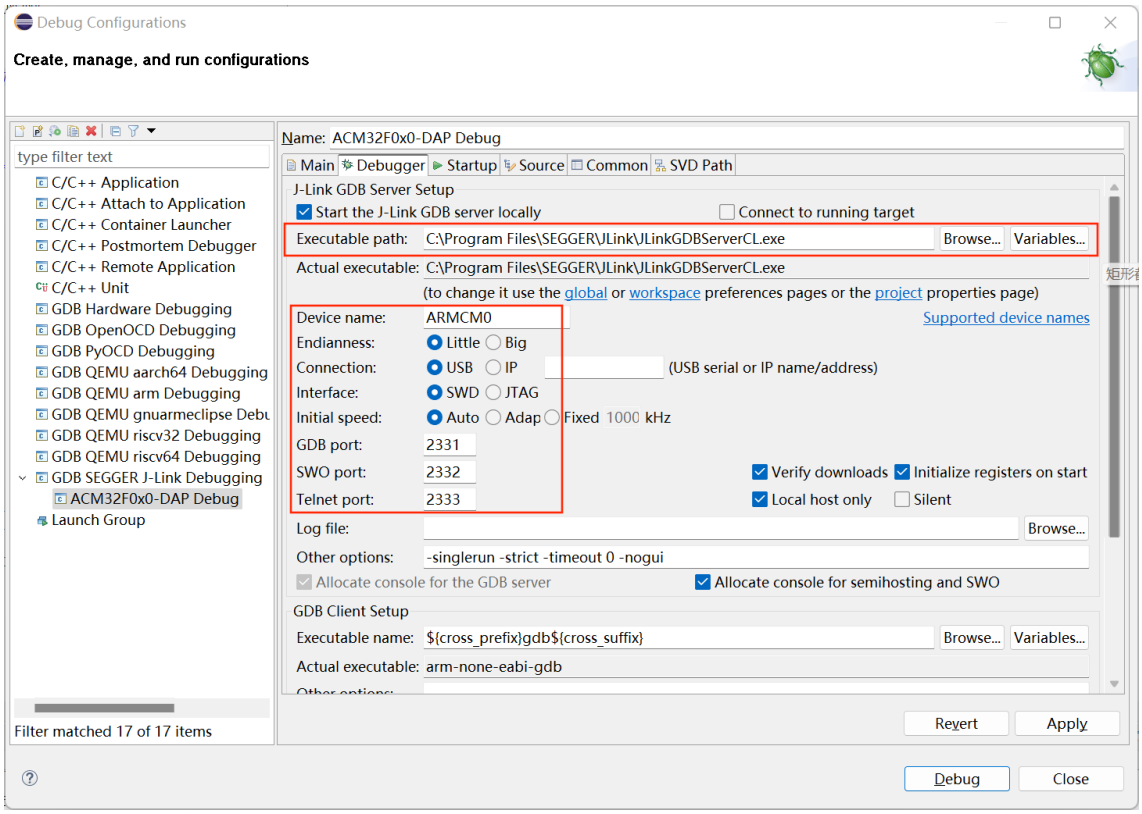
③进入到 Debug Configurations 配置页面

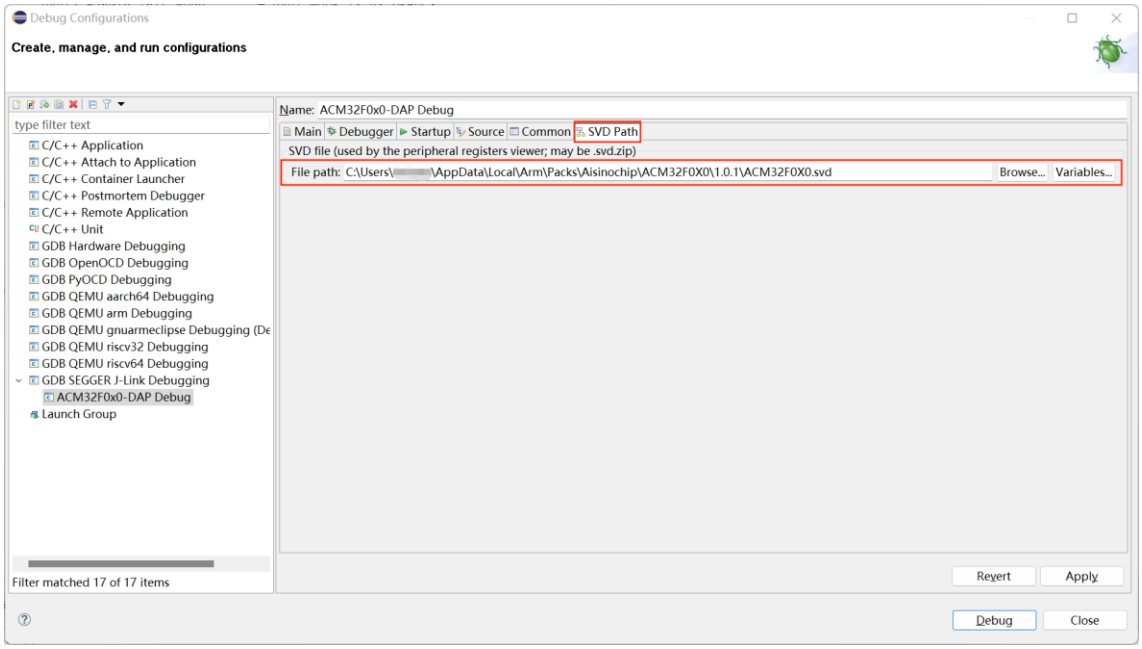


④双击 GDB SEGGER J-Link Debugging 创建 J-Link 调试的配置，修改相应的配置后

即可使用 J-Link 进行调试







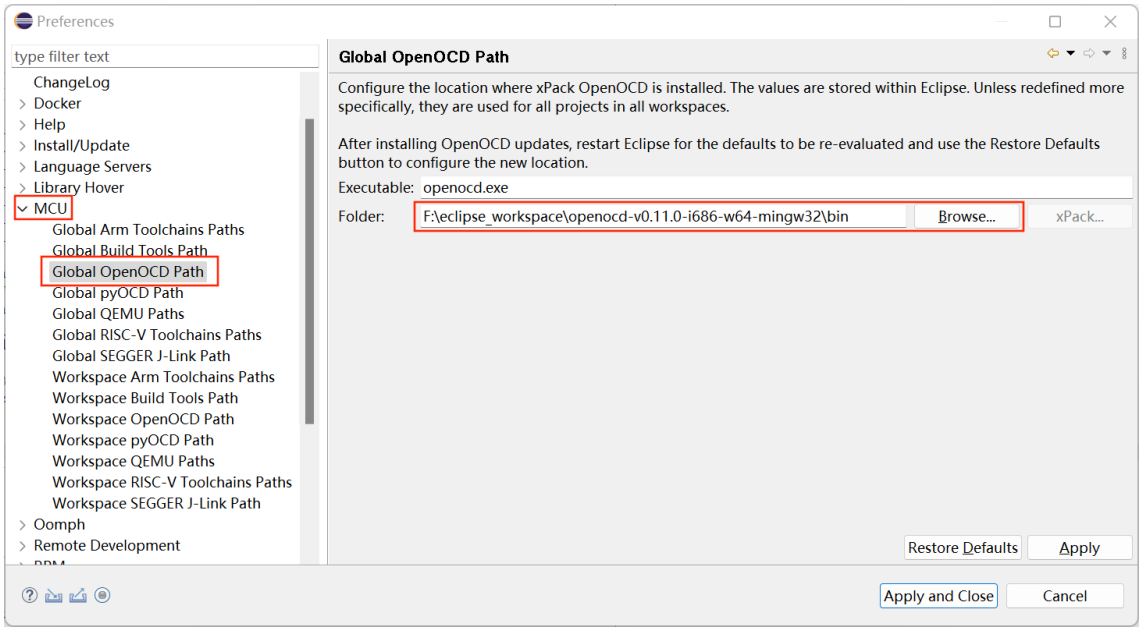
ACM32 相关 PACKS 安装后，SVD 文件可在相关文件夹内索引，索引地址为

C:\Users\用户名\AppData\Local\Arm\Packs\Aisinochip

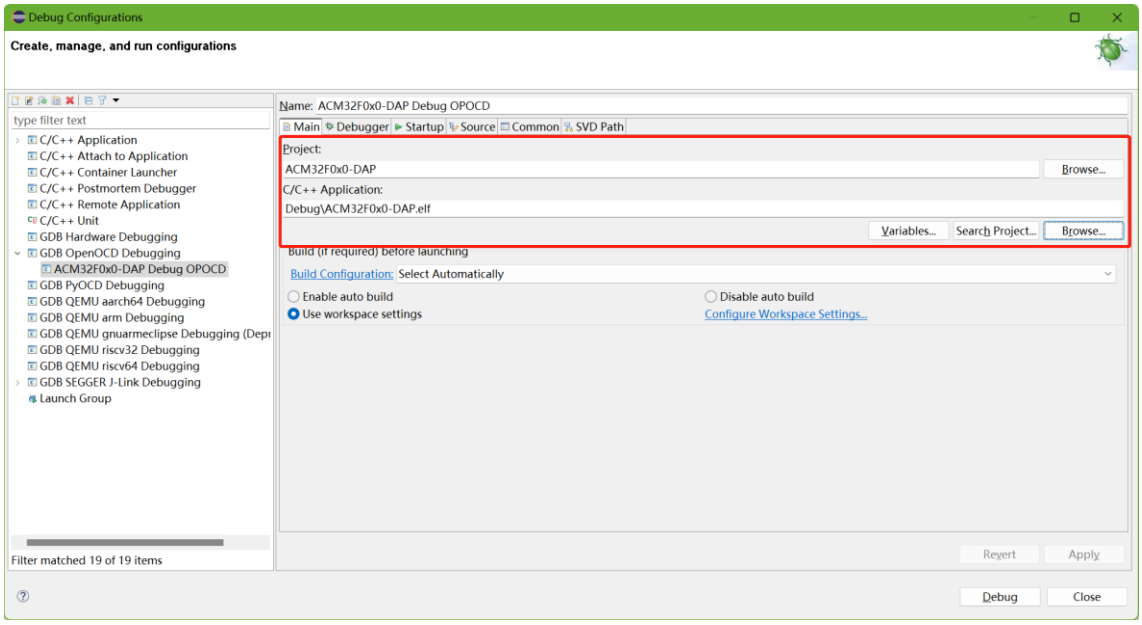
配置完成后即可使用 J-Link 进行调试

1.3.2 CMSIS-DAP

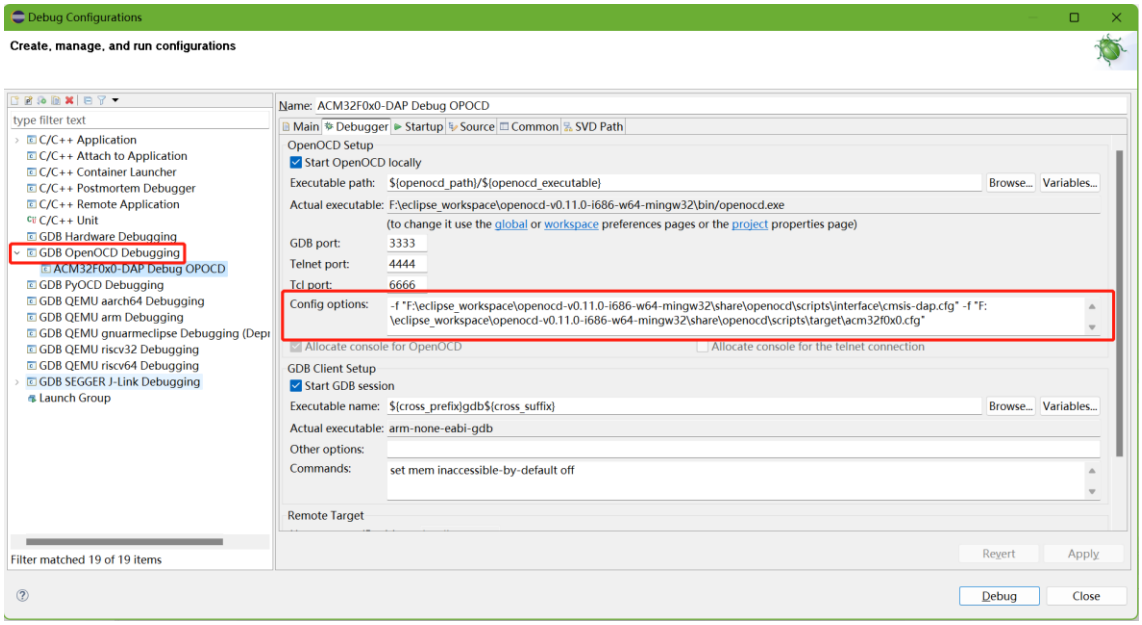
DAP 调试需要使用 OpenOCD 插件，目前网络上的版本是由于没有针对 ACM32 系列的 eFlash 操作做支持，是无法直接使用的，需要使用航芯提供的版本，相应的软件包名称为 openocd-v0.11.0-i686-w64-mingw32。直接把该软件放入到指定的目录下，依次点击 eclipse 的 Windows->Preferences->MCU->Global OpenOCD Path 进入到设定中，指定 OpenOCD 工具的目录,图示如下：

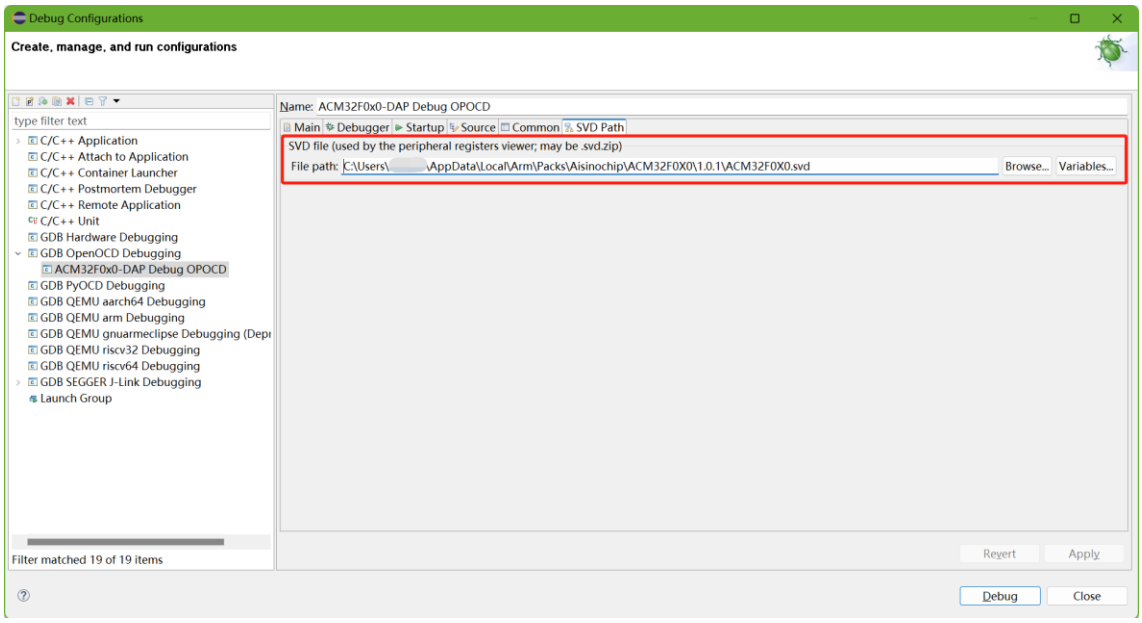


然后进入到 Debug Configurations 配置页面中，双击 GDB OpenOCD Debugging 创建 DAP 调试的配置，修改相应的配置后即可使用 DAP 进行调试，相应的配置项如图所示：



按照格式索引 cmsis-dap.cfg 与 acm32f0x0.cfg 两个文件


Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
C/C++ Attach to Application
C/C++ Container Launcher
C/C++ Postmortem Debugger
C/C++ Remote Application
C/C++ Unit
GDB Hardware Debugging
GDB OpenOCD Debugging
ACM32F0x0-DAP Debug OPOCD
GDB PyOCD Debugging
GDB QEMU aarch64 Debugging
GDB QEMU arm Debugging
GDB QEMU gnuarmeclipse Debugging (Depr
GDB QEMU riscv32 Debugging
GDB QEMU riscv64 Debugging
GDB SEGGER J-Link Debugging
Launch Group
Filter matched 19 of 19 items
Name: ACM32F0x0-DAP Debug OPOCD
Main
Debugger
Startup
Source
Common
SVD Path
OpenOCD Setup
Start OpenOCD locally
Executable path:
${openocd_path}/${openocd_executable}
Browse...
Variables...
Actual executable:
F:\eclipse_workspace\openocd-v0.11.0-i686-w64-mingw32\bin/openocd.exe
(to change it use the global or workspace preferences pages or the project properties page)
GDB port:
3333
Telnet port:
4444
Tcl port:
6666
Config options:
-f "F:\eclipse_workspace\openocd-v0.11.0-i686-w64-mingw32\share\openocd\scripts\interface\cmsis-dap.cfg" -f "F:
\eclipse_workspace\openocd-v0.11.0-i686-w64-mingw32\share\openocd\scripts\target\acm32f0x0.cfg"
Allocate console for OpenOCD
Allocate console for the telnet connection
GDB Client Setup
Start GDB session
Executable name:
${cross_prefix}gdb${cross_suffix}
Browse...
Variables...
Actual executable:
arm-none-eabi-gdb
Other options:
Commands:
set mem inaccessible-by-default off
Remote Target
Revert
Apply
Debug
Close


Debug Configurations
Create, manage, and run configurations
type filter text
C/C++ Application
C/C++ Attach to Application
C/C++ Container Launcher
C/C++ Postmortem Debugger
C/C++ Remote Application
C/C++ Unit
GDB Hardware Debugging
GDB OpenOCD Debugging
ACM32F0x0-DAP Debug OPOCD
GDB PyOCD Debugging
GDB QEMU aarch64 Debugging
GDB QEMU arm Debugging
GDB QEMU gnuarmeclipse Debugging (Depr
GDB QEMU riscv32 Debugging
GDB QEMU riscv64 Debugging
GDB SEGGER J-Link Debugging
Launch Group
Filter matched 19 of 19 items
Name: ACM32F0x0-DAP Debug OPOCD
Main
Debugger
Startup
Source
Common
SVD Path
SVD file (used by the peripheral registers viewer; may be .svd.zip)
File path:
C:\Users\
\AppData\Local\Arm\Packs\Aisinochip\ACM32F0X0\1.0.1\ACM32F0X0.svd
Browse...
Variables...
Revert
Apply
Debug
Close