

## 2.1.7: 调试器使用教程

### 2.1.7.1 支持的开发环境

- 支持MDK、IAR 免配置直接使用。
- 基于eclipse 框架的IDE

eclipse 框架的IDE 一般编译器配置为 gcc, 调试器配置为 pyocd 或者 openocd 的形式, 需要定制厂家对DAPLink 进行支持, 或者安装 DAPLINK 的支持插件, 典型的IDE 比如 RT-thread Studio 原生支持 DAPLink 的调试。

openocd 使用文档: [Documentation \(openocd.org\)](http://Documentation.openocd.org)

pyocd 开源地址: [pyocd/pyOCD: Open source Python library for programming and debugging Arm Cortex-M microcontrollers \(github.com\)](https://github.com/pyocd/pyOCD: Open source Python library for programming and debugging Arm Cortex-M microcontrollers)

⚠️::caution 提示

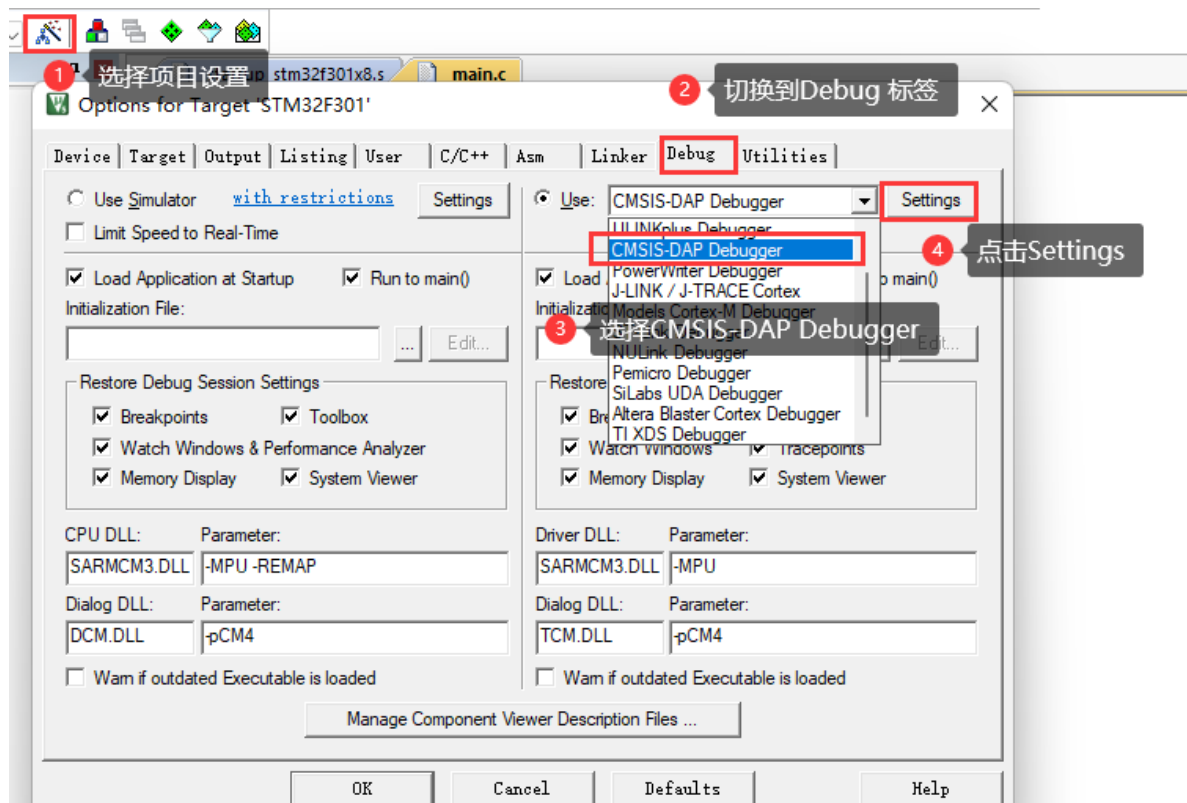
使用pyocd 或者 openocd 的用户, 以及使用eclipse 框架的IDE, 请尽可能的使用新版本的驱动和软件, 原因在于开源的发展速度相对较快, 过时的版本可能支持不完善, 或者已经弃用, 而避免产生问题。

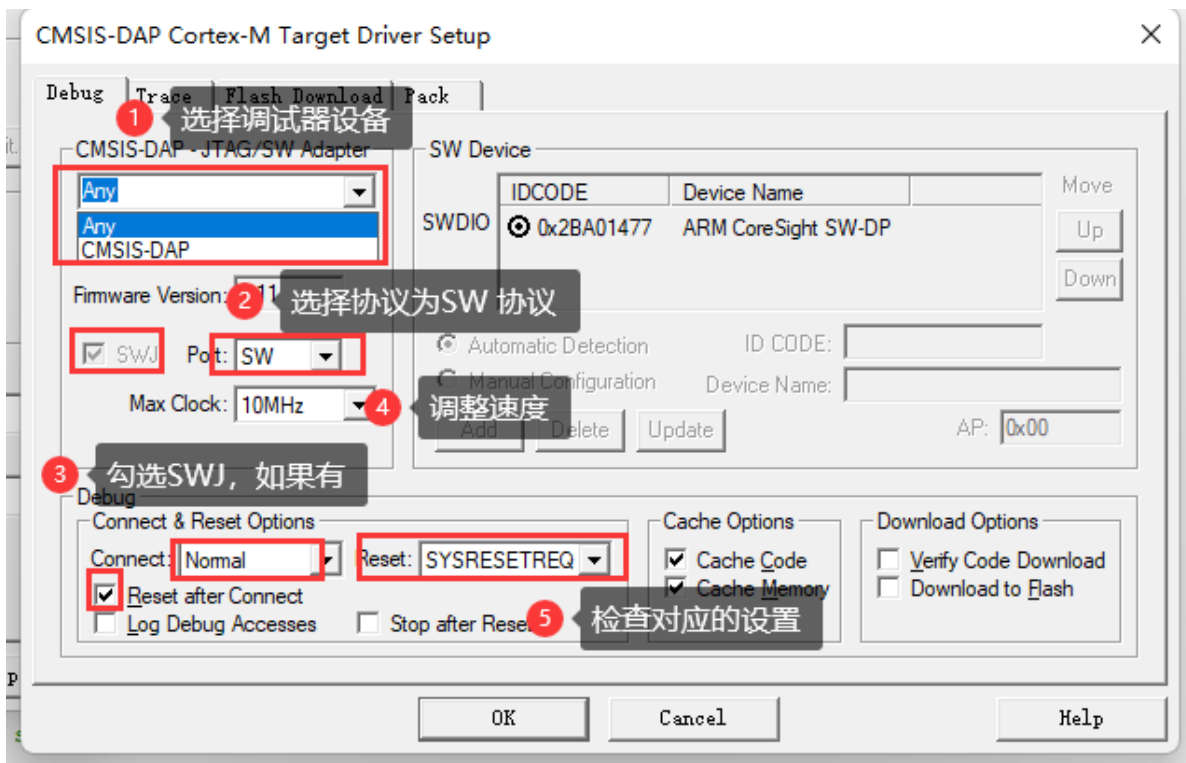
⋮

### 2.1.7.2 调试环境设置

#### 2.1.7.2.1 MDK 调试设置

按照如下的流程, 打开项目设置, 选择CMSIS-DAP 调试器, 如下图所示:



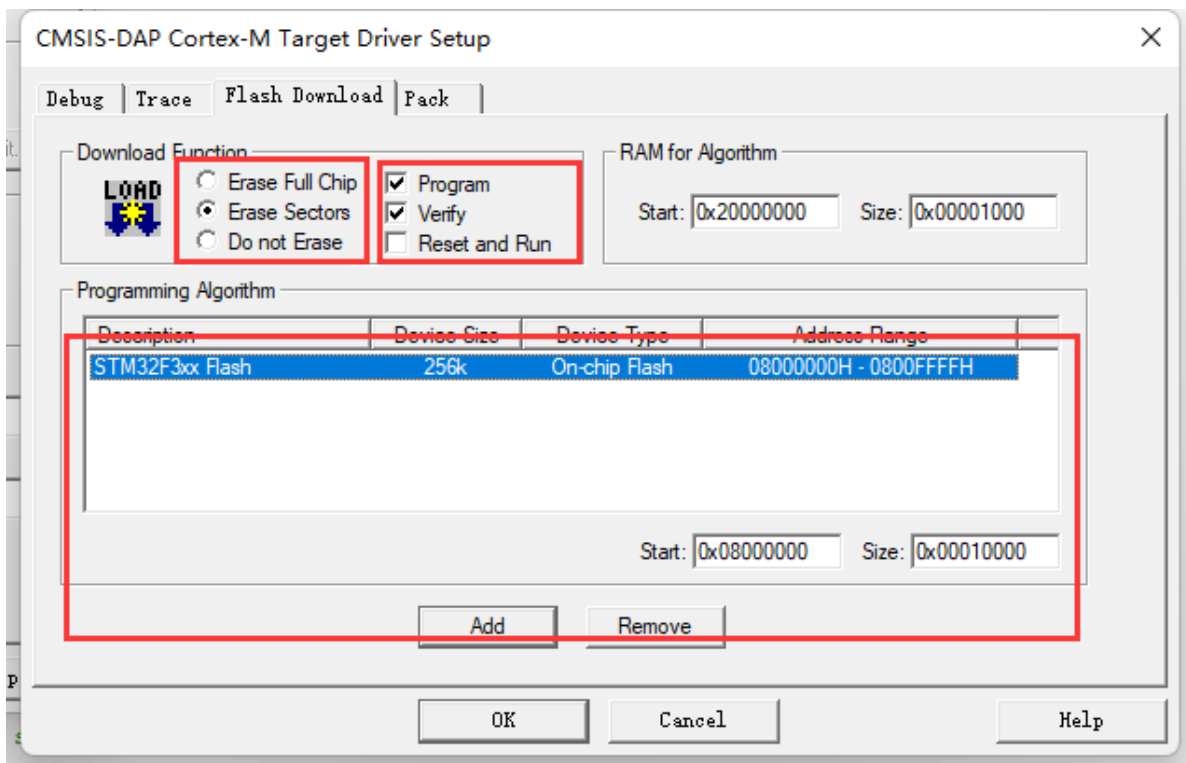


...tip

### SWD/JTAG Communication Failure/ RDDI Error 处理方法:

...

Flash Download 设置如下, 根据实际的项目要求, 设置合适的擦除方式, 以及Program、Verify、Reset and Run 选项, 并添加当前芯片的flash 算法, 如下图所示

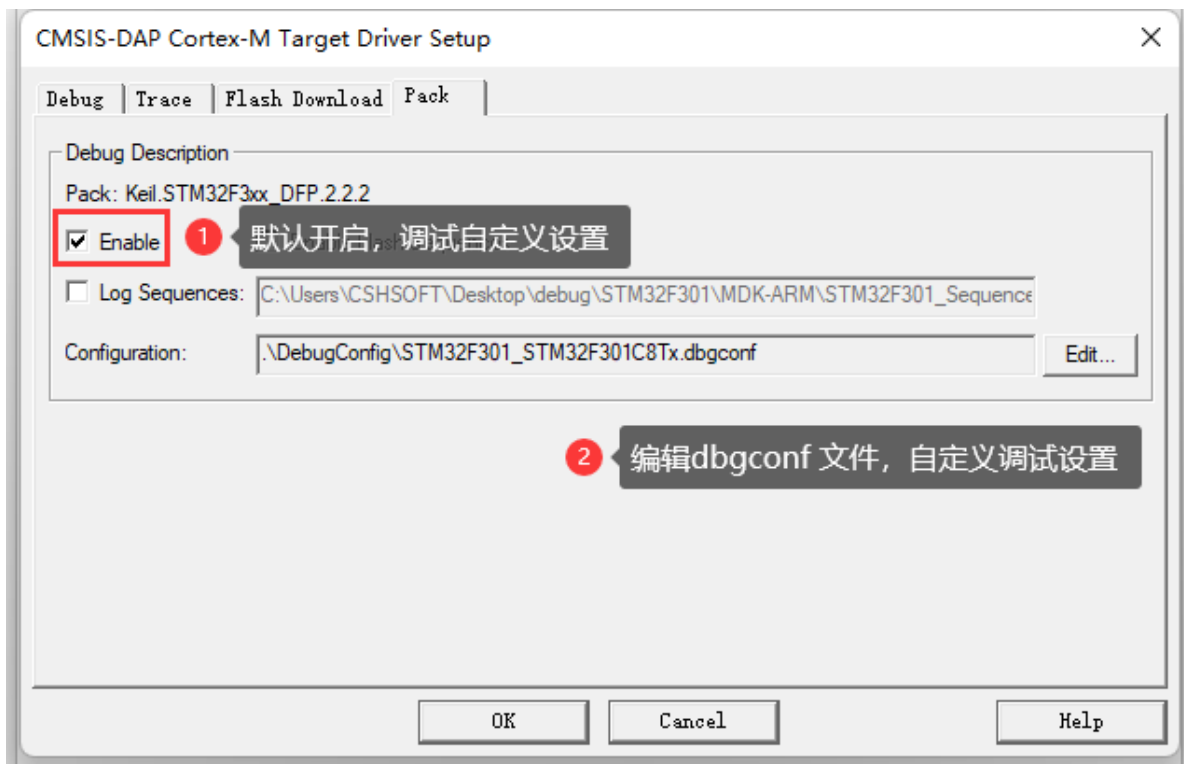


...tin

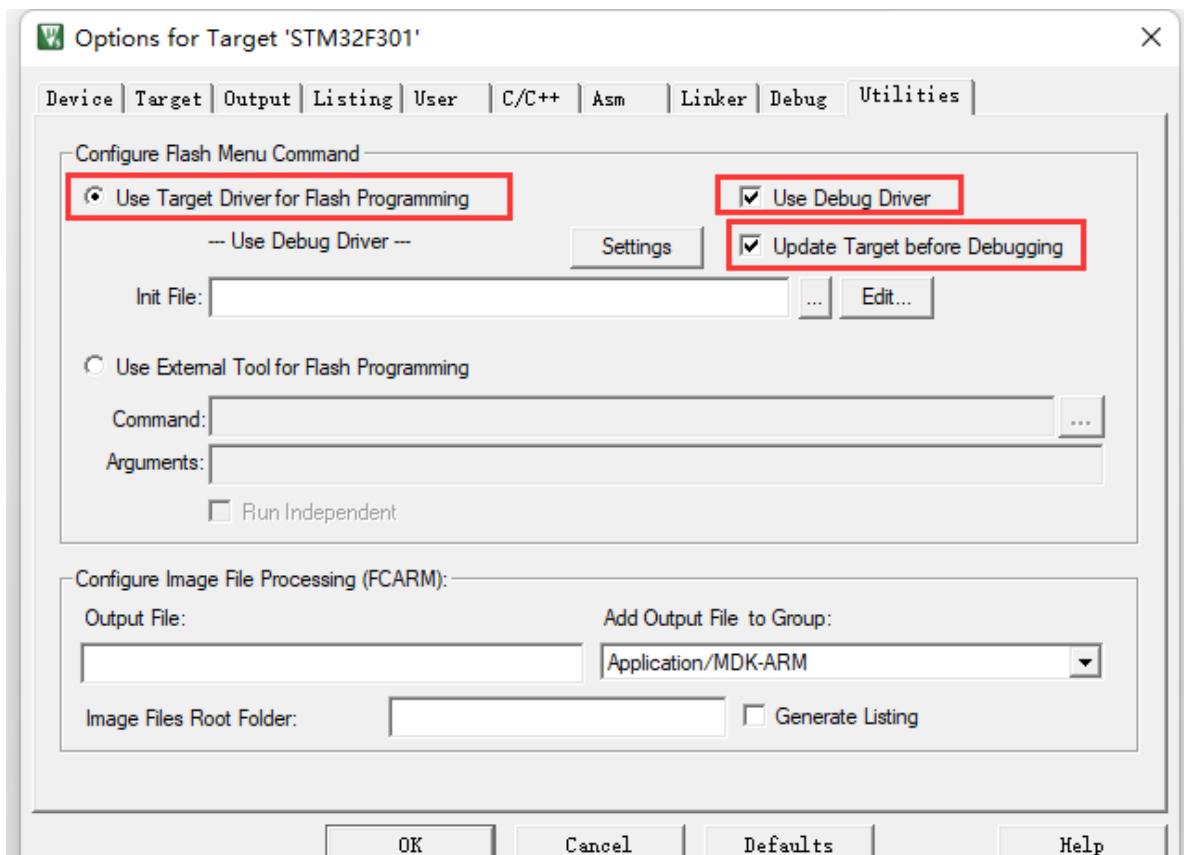
- **擦除方式选择**：如果需要保留芯片的某些数据，比如设置数据，签名信息，一般选择擦除方式为 Sector Sections
- **Reset and Run**：在现在固件后要执行复位并运行，需要勾选Reset and Run 选项

...

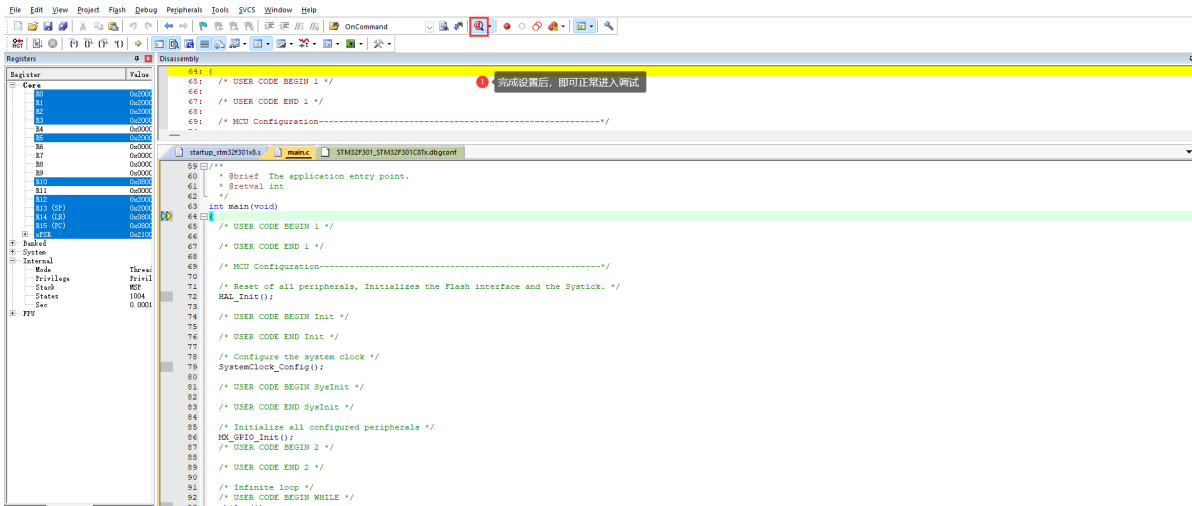
针对高级用户，可以根据需要调整自定义调试设置，如下所示：



检查Use Debug Driver选项，启用硬件调试器，此选项默认是开启的，如遇到问题，此选项也是重点需要检查的设置，如下所示

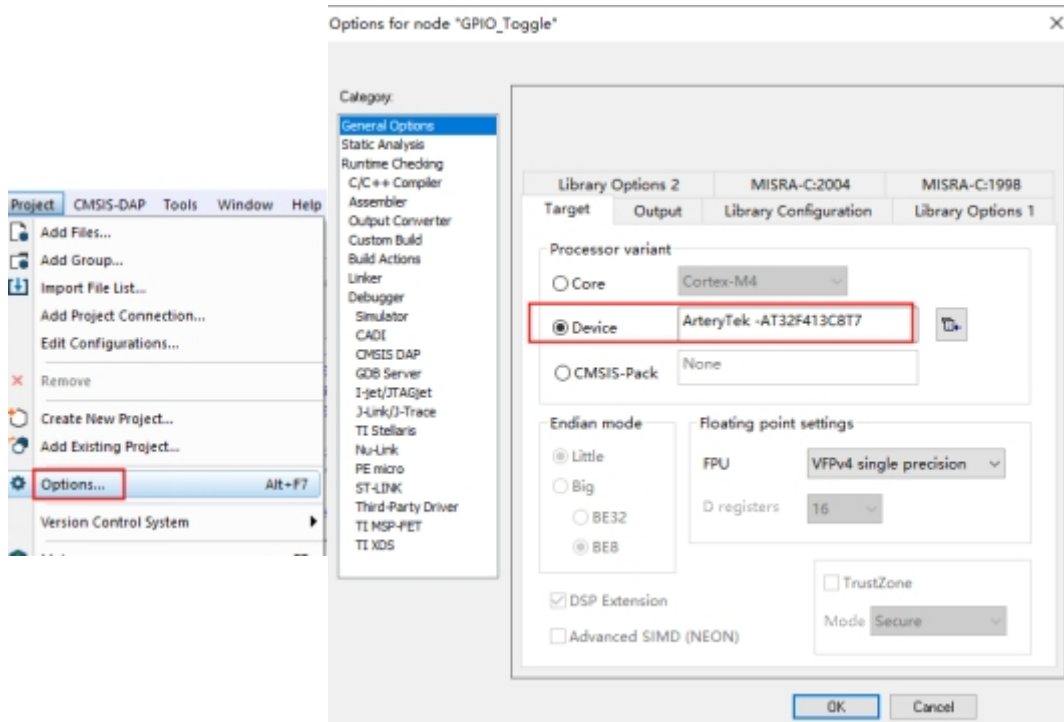


检查设置之后，即可正常进入调试

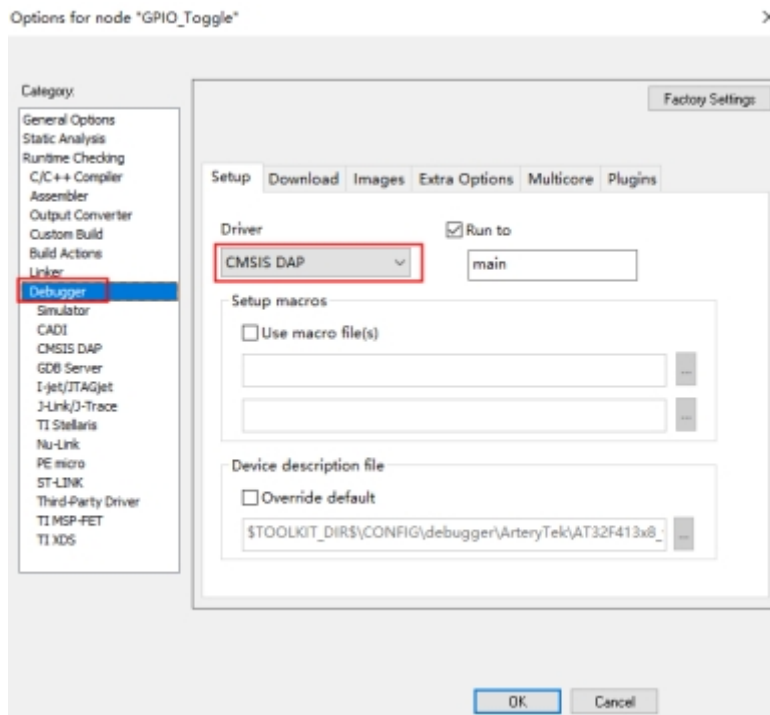


## 2.1.7.2.2 IAR 调试设置

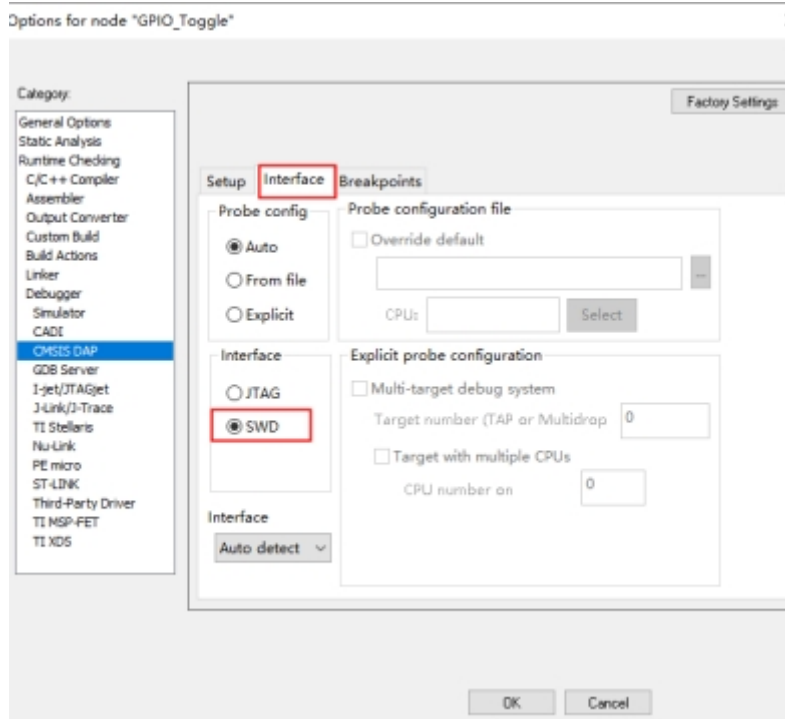
项目右键，进入 Options 设置，切换到 General Options 标准设置页面，检查 Target -> Device 是否选择正确，如下所示。



进入到 Debugger 菜单，Setup 标签页面，选择 Driver 为：CMSIS-DAP 设备，根据需要可以勾选 Run to main 选项，如下所示。



在Debugger -> CMSIS-DAP 的 Interface 标签页中，选择接口为 **SWD**，如下所示

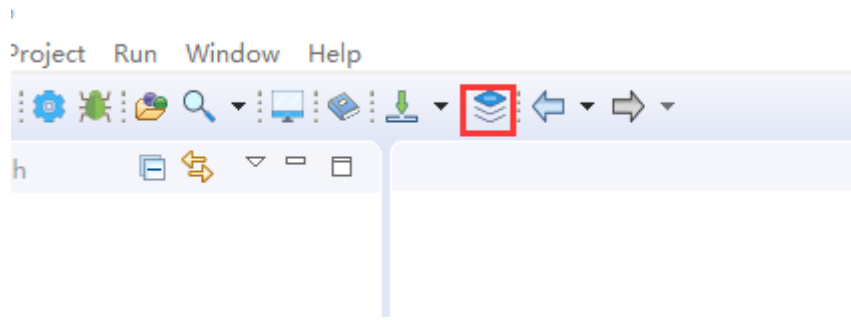


经过以上的设置就可以进行芯片的调试开发。

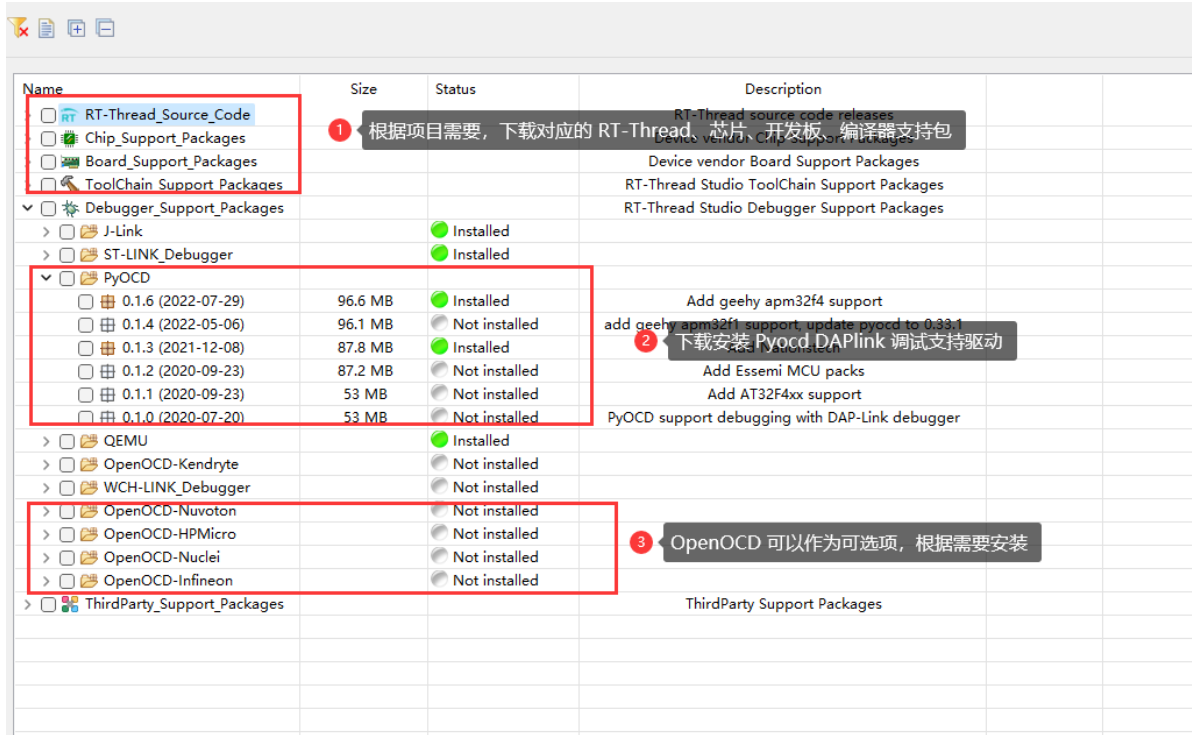
### 2.1.7.2.3 RT-thread Studio 调试设置

RT - thread Studio 下载地址: <https://www.rt-thread.org/studio.html>

进入SDK Manager，如下所示:



检查当前芯片支持包，RT-thread 源码，以及Pyocd 是否安装。

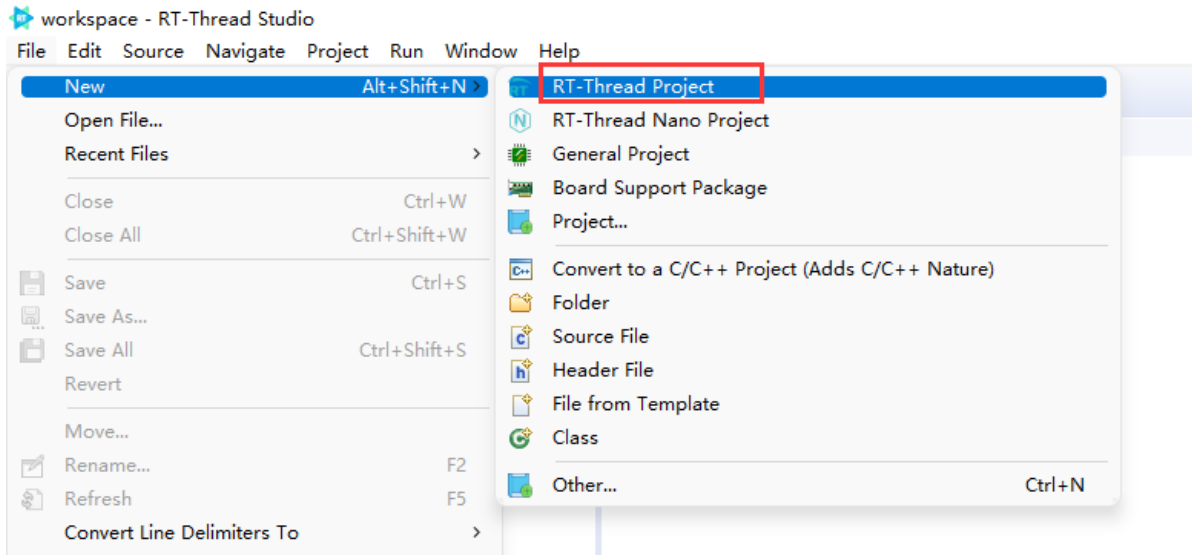


...tip

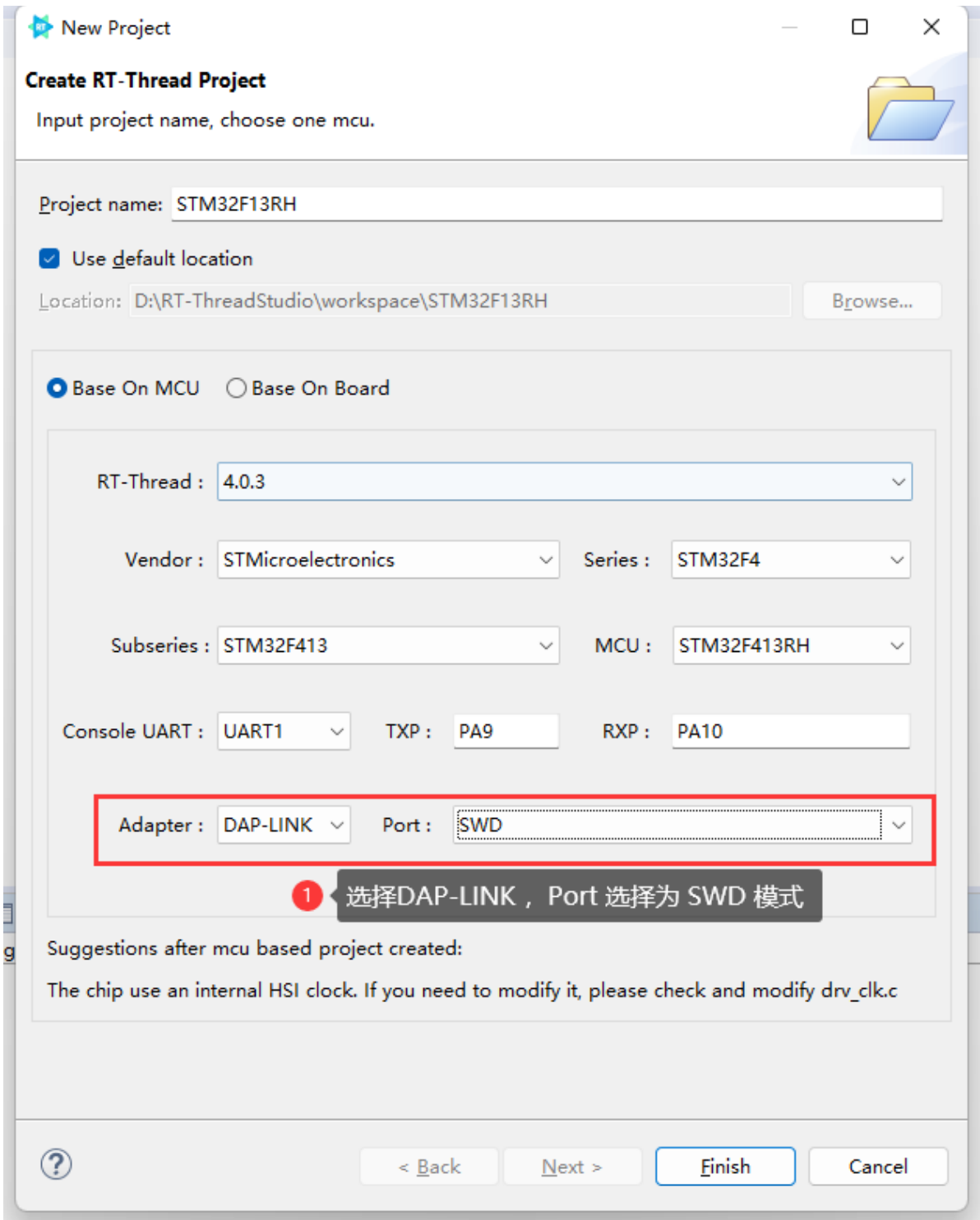
通过DAPLink进行项目调试，必须使用Pyocd 或者 OpenOCD 作为调试器的Adapter，最终通过 gdb 连接进行调试。

...

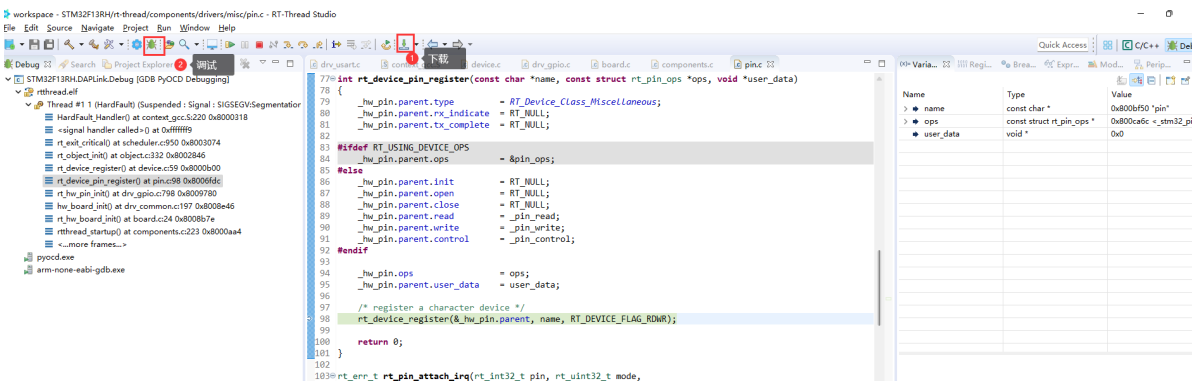
新建RT-Thread 项目，如下所示：



设置调试器，如下所示：



配置完成后，生成项目，即可进行下载和调试，如下：





在SDK-manager 中安装好 Pyocd 或者 Openocd 是必须的,

...

## 2.1.7.3 常见问题{#faq}

---

### 2.1.7.3.1 MDK自动启动无效

程序下载之后发现没运行, 需要注意以下几点:

- 检查Debugger设置中的复位模式是否正确;
- Reset and run 是否勾选
- Flash Algorithm 设置是否正确。

### 2.1.7.3.2 断点不生效

请检查编译的设置。如优化级别、debugger 设置、以及本身的代码功能, 如是否开启了看门狗等。

### 2.1.7.3.3 只能设置四个硬件断点

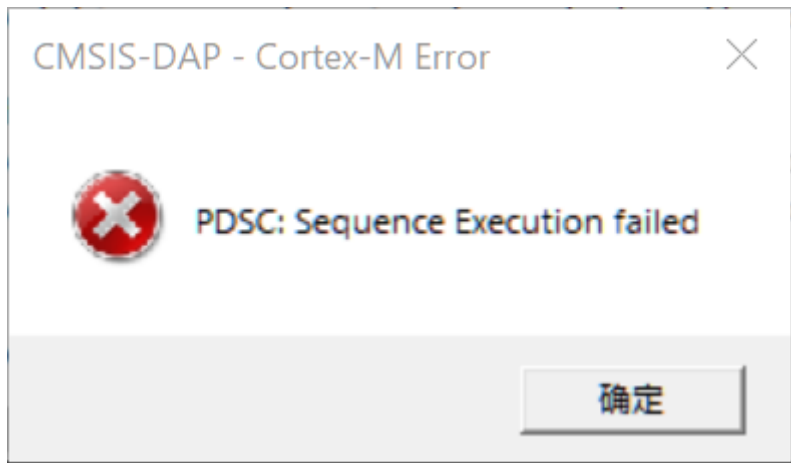
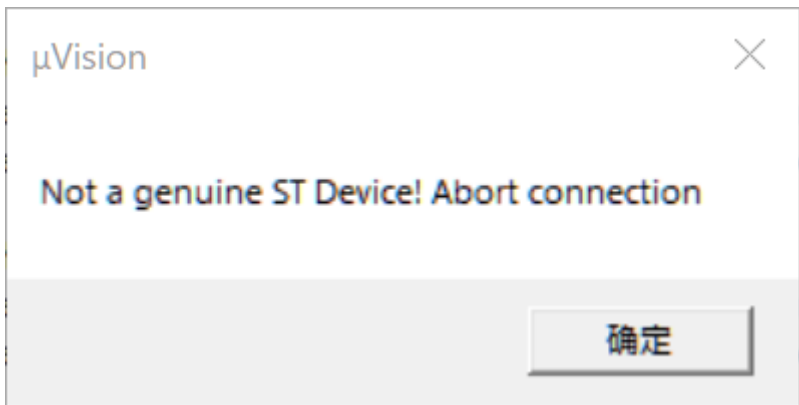
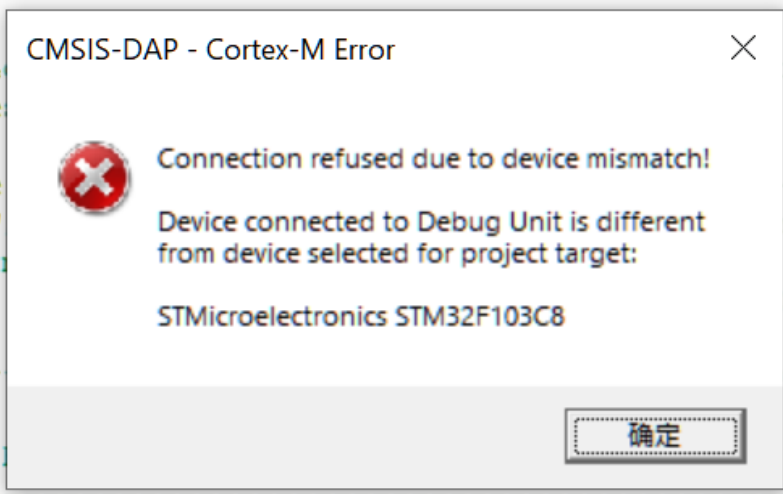
断点数量与芯片本身有关系, 取决于 MCU 内核版本单元, 手册上会列出当前芯片支持的断点数量, 请查看所使用的芯片手册关于调试章节内容

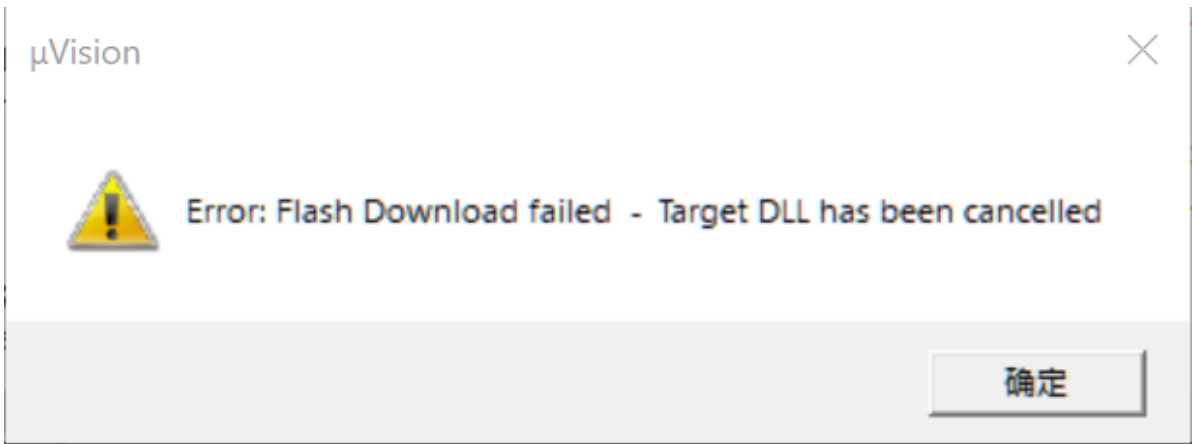
### 2.1.7.3.4 下载时删掉了Bootloader

- IROM 的设置, 首选要确保生成的 Image 地址是正确的
- 写入 Flash 的时候注意选择, 页面擦除, 在 Debugger 页面设置, 这样就可以避免整片擦除, 保留了芯片里面的其他数据

### 2.1.7.3.5 Connection refused due to device mismatch!(Not a genuine ST Device! Abort connection){#device\_mismatch}

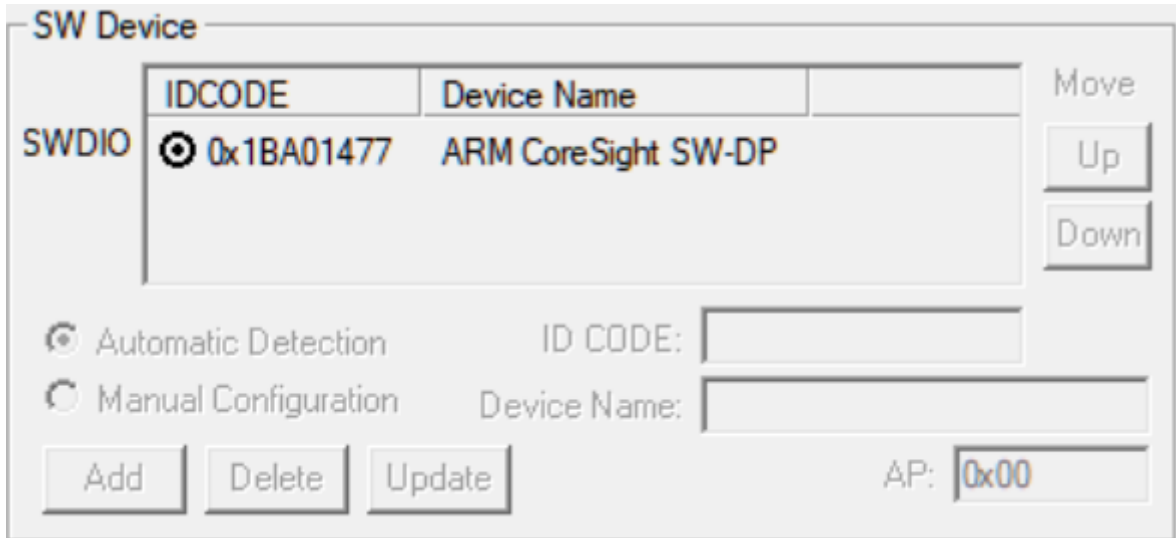
```
main program body
*****
tion
enter>&
ghts re
oftware
icense"
e. You
*****
DE END
s
main.h"
includes
DE BEGIN Includes */
```



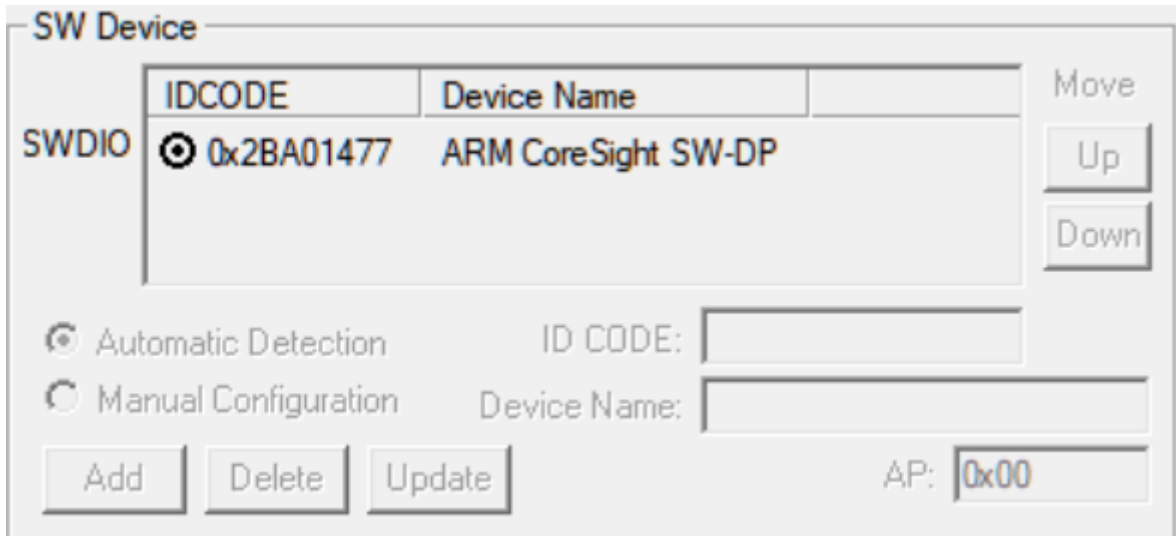


可能购买的芯片不是原装芯片，比如打磨过丝印，或者商家虚假宣传，由于部分热门芯片很多厂家都有类似的芯片，甚至PIN TO PIN 兼容，因此存在这些问题，低版本的MDK 以及 芯片支持包不会检查芯片的IDCODE，所以可能不报错，更换成高版本的MDK和 支持包，则有可能报错，具体可以**核对芯片手册的IDCODE**，来确定真正的芯片型号。

如STM32F1 芯片的IDCODE 为：**0x1B10417**。



但是样片却为：**0x2BA01477**，如下图所示：



经过查明，此芯片实为：**CS32F103C8T6(CKS32F103C8T6)**。

根据 **IDCODE** 查找芯片真正的厂商型号，下载对应厂商芯片的 设备包，安装之后，切换成实际芯片

[CKS32F103C8 资料下载 \(需要安装Keil.CS32F1xx DFP.pack、内有采用STM32F103 库、CKS32F103 库可正常运行Demo\)](#)

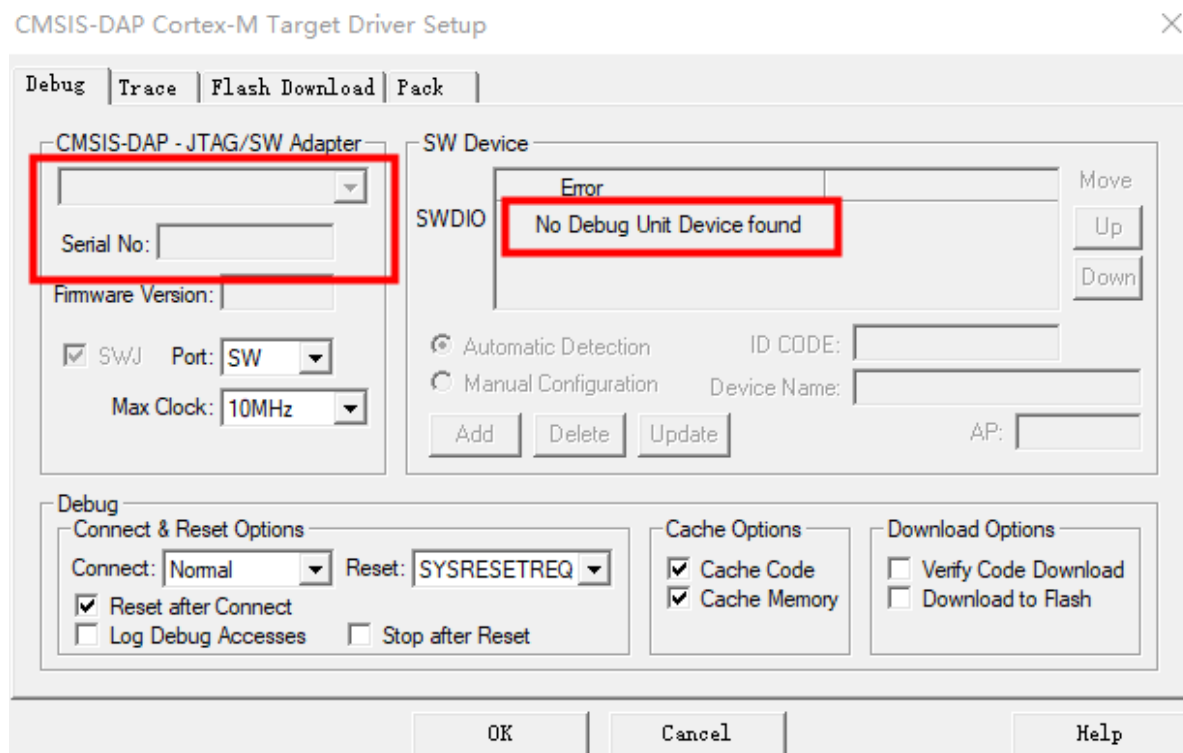
:::tip 提示

PowerWriter 团队不对真正的芯片型号进行溯源，以上的芯片IDCODE只是说明实际案例，与厂家无关，为销售渠道行为，特此说明。

:::

### 2.1.7.3.6 No Debug Unit Device found

原因：没有发现烧录器



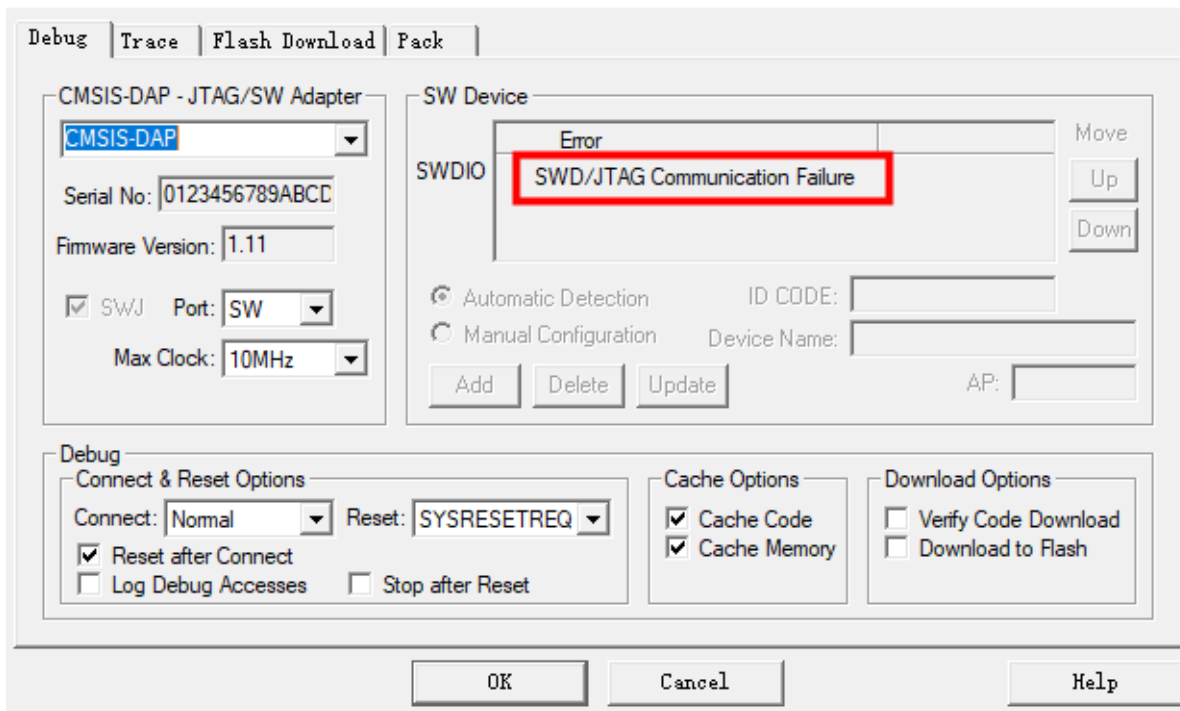
处理方法：重新连接烧录器，确保烧录器连接正常，驱动连接正常，详情请看：

[驱动异常如何处理](#)

### 2.1.7.3.7 SWD/JTAG Communication Failure(RDDI-DAP Error)

原因：识别不到芯片

### CMSIS-DAP Cortex-M Target Driver Setup



#### 解决方法:

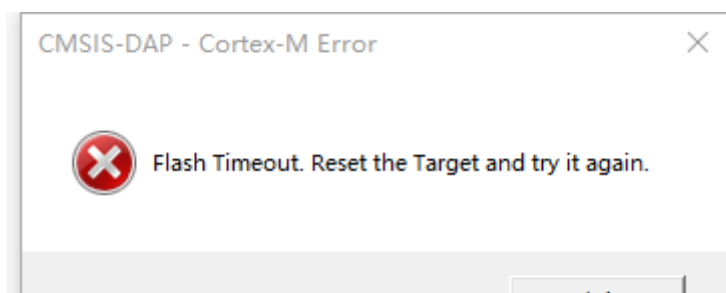
确保烧录器和芯片烧录口连接正常，芯片没有开启高级保护或者烧录口被复用了，具体处理方法请查看：

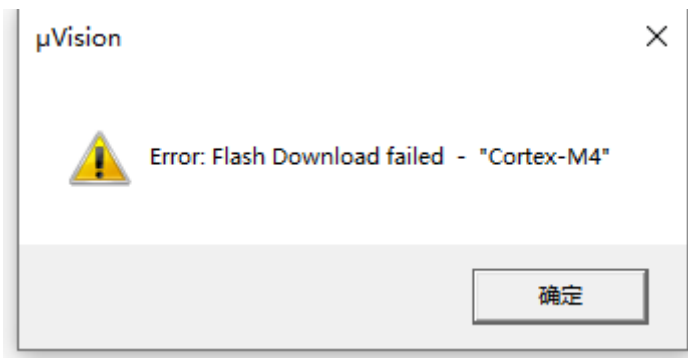
[芯片连接不上如何处理](#)

[https://docs.powerwriter.com/docs/faq/powerwriter/base/device\\_mismatch](https://docs.powerwriter.com/docs/faq/powerwriter/base/device_mismatch)

### 2.1.7.3.8: Flash Timeout. Reset the Target and try it again.

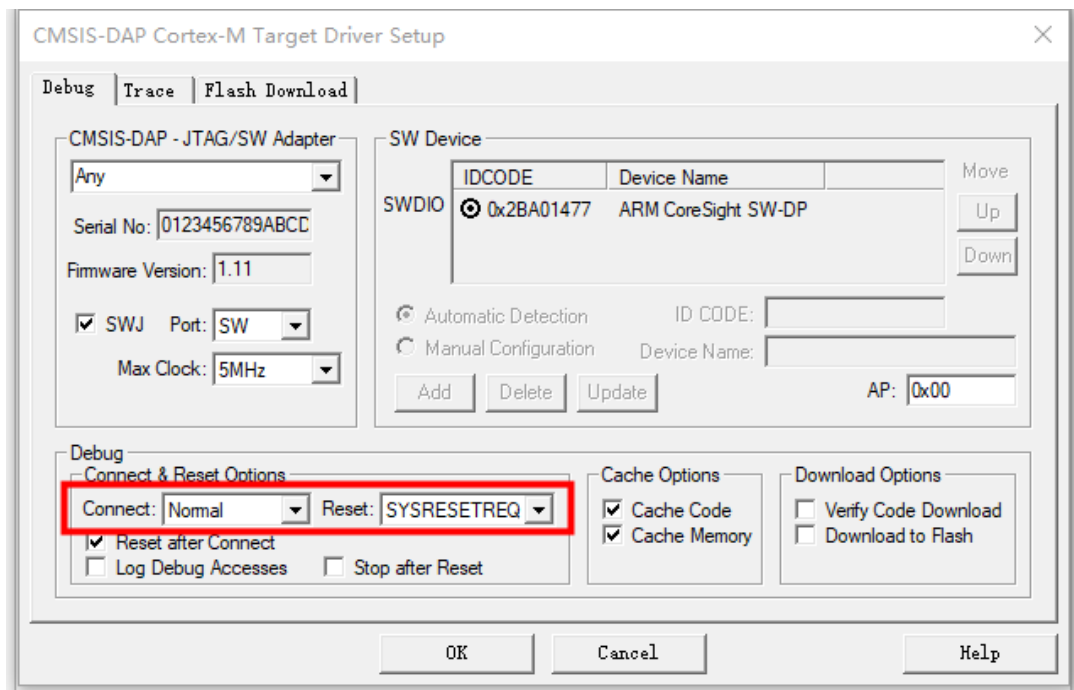
原因：烧录失败





### 解决方法:

- 确保芯片没有开启读保护和写保护，去除保护方法参考：[解除读保护](#)
- 程序跑飞时或者芯片复位不成功时会导致烧录不成功，请重新连接芯片并确保调试器设置选项如下图所示，并且可以尝试将boot0引脚拉高后再进行调试；



- 更改调试时钟大小
- 以上方法还是无法解决时请尝试使用[Power Writer上位机软件](#)进行擦除芯片后再使用keil进行仿真调试。

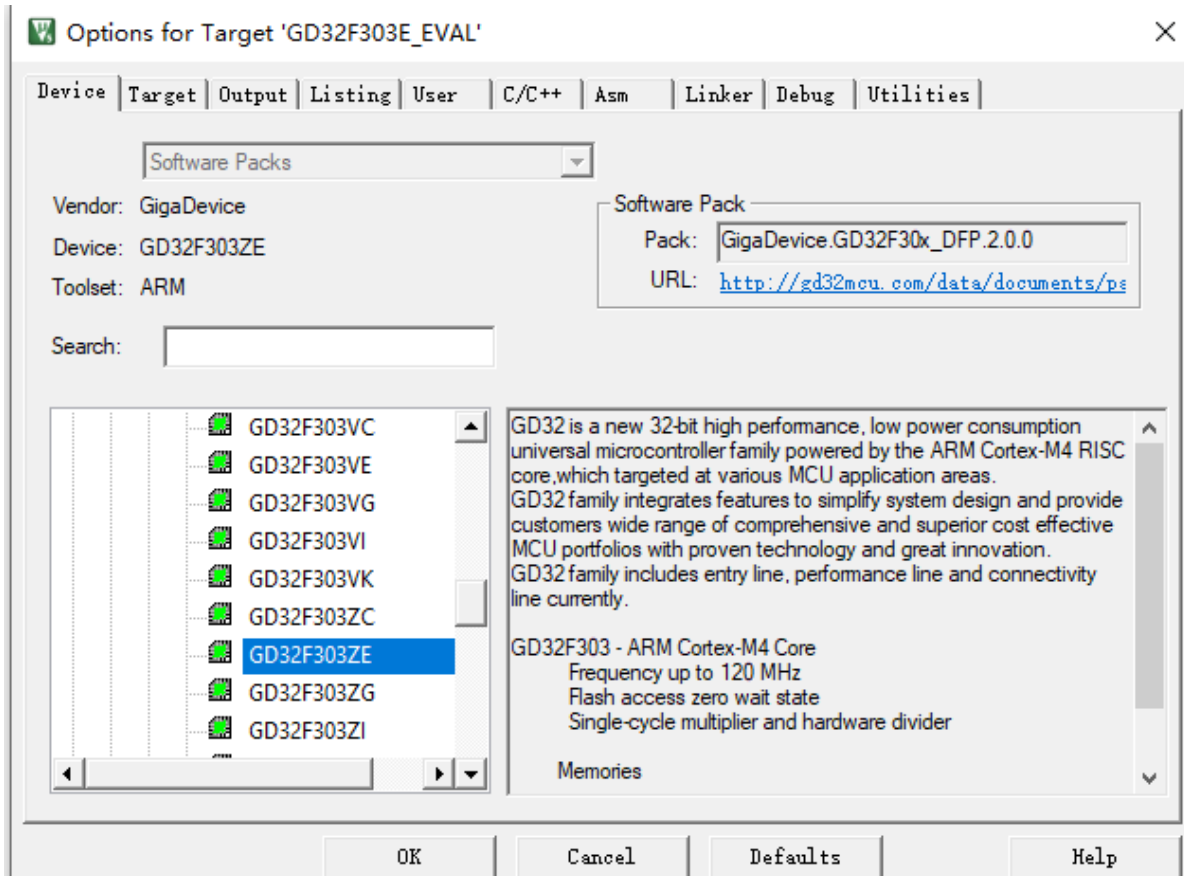
## 2.1.7.3.9 Contents mismatch

原因：烧录成功，校验失败

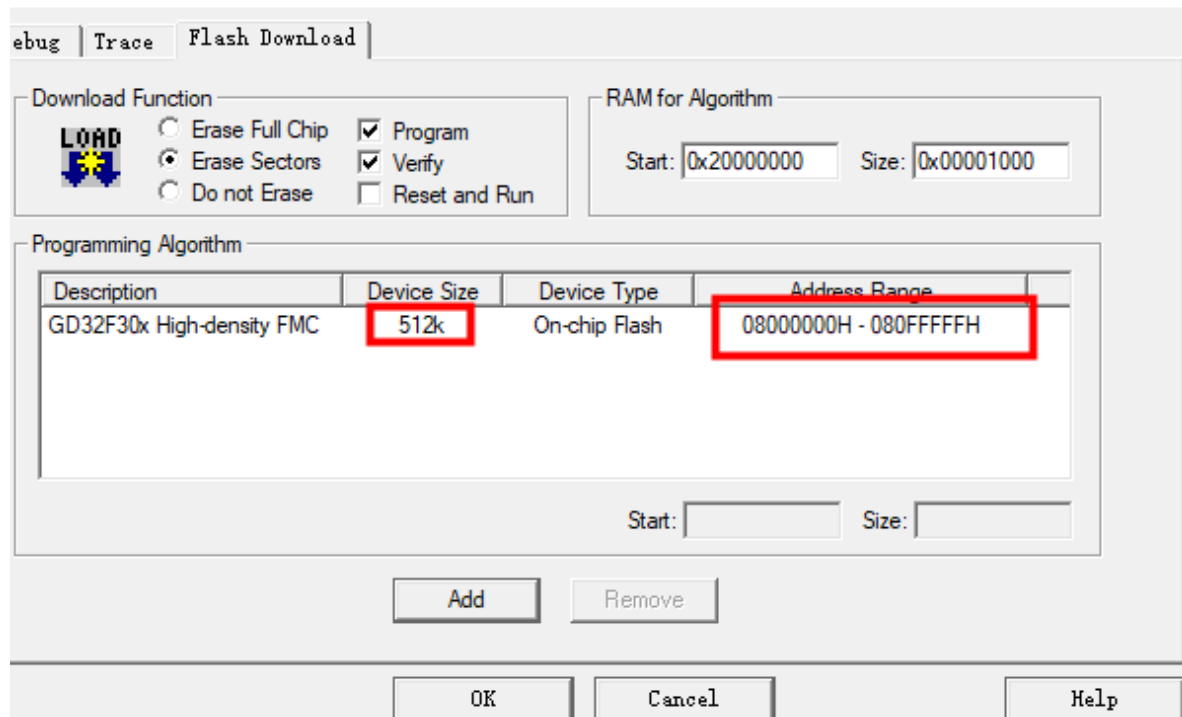
```
Contents mismatch at: 08000062H (Flash=FFH Required=0)
Contents mismatch at: 08000063H (Flash=FFH Required=0)
Too many errors to display !
Error: Flash Download failed - "Cortex-M4"
Flash Load finished at 16:21:51
```

排除问题：

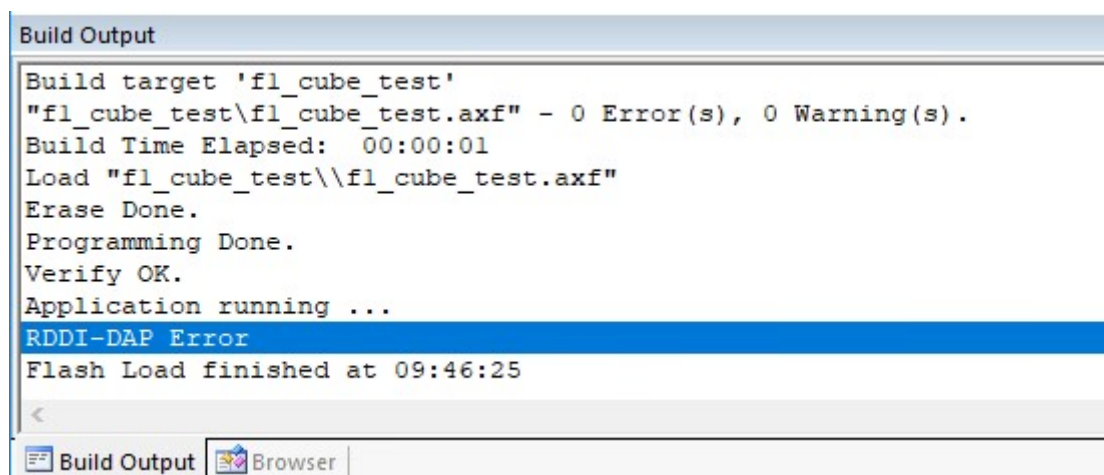
确保flash算法参数正常，或者可以重新选择芯片型号进行更新flash算法设置：



## vitis-DAP Cortex-M Target Driver Setup



## 2.1.7.3.10 调试过程断线

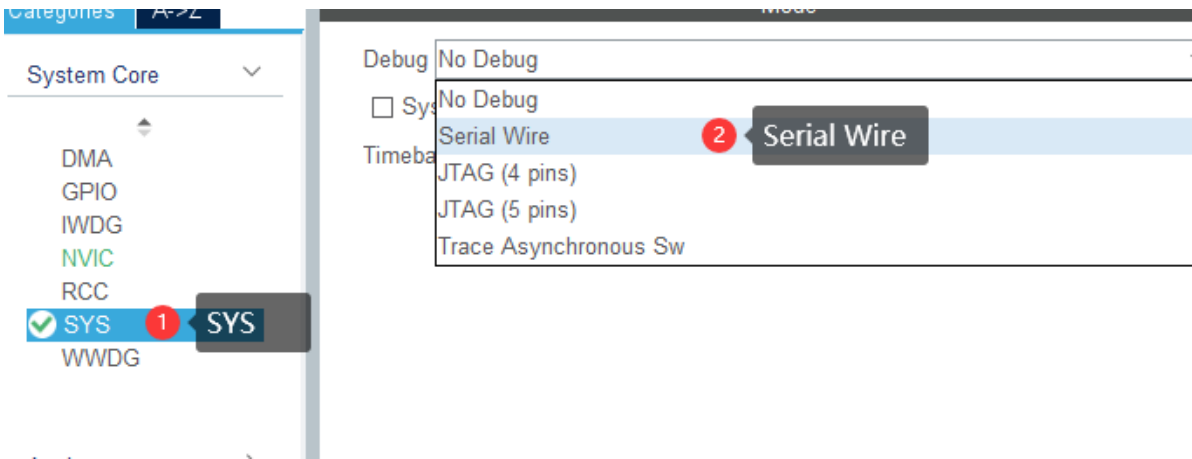


如上图所示，MDK **擦除、写入、校验都是通过的**。但是在 Application running ... 之后，突然出现 RDDI-DAP Error!

原因：程序中没有将 IO 的调试功能打开，比如 CubeMX 中默认调试功能是关闭的，如果直接生成代码去调试，就会出现此问题。

解决方法：在 CubeMX 中，左侧，SYS->Debug 中打开 Serial Wire，如下图所示，然后更新代码。





如果直接在代码中，调整 调试模式。

```

61  * Initializes the Global MSP.
62  */
63  void HAL_MspInit(void)
64  {
65      /* USER CODE BEGIN MspInit 0 */
66
67      /* USER CODE END MspInit 0 */
68
69      __HAL_RCC_AFIO_CLK_ENABLE();
70      __HAL_RCC_PWR_CLK_ENABLE();
71
72      /* System interrupt init*/
73
74      /** NOJTAG: JTAG-DP Disabled and SW-DP Enabled
75      */
76      __HAL_AFIO_REMAP_SWJ_NOJTAG();
77
78      /* USER CODE BEGIN MspInit 1 */
79
80      /* USER CODE END MspInit 1 */
81  }
82
83  /* USER CODE BEGIN 1 */
84

```

:::tip 提示

其他的品牌和工具，也是类似处理，此类问题，都是因为 调试引脚没有配置为调试模式导致。

:::

### 2.1.7.3.11 PowerWriter 正常识别，MDK识别不到目标芯片

此现象原因跟上一个现象一致，由于芯片中有代码运行，禁用了调试口，PowerWriter 响应的效率比调试器要快一些，存在PowerWriter 能识别，但是MDK 失败的情况。

#### 处理方法：

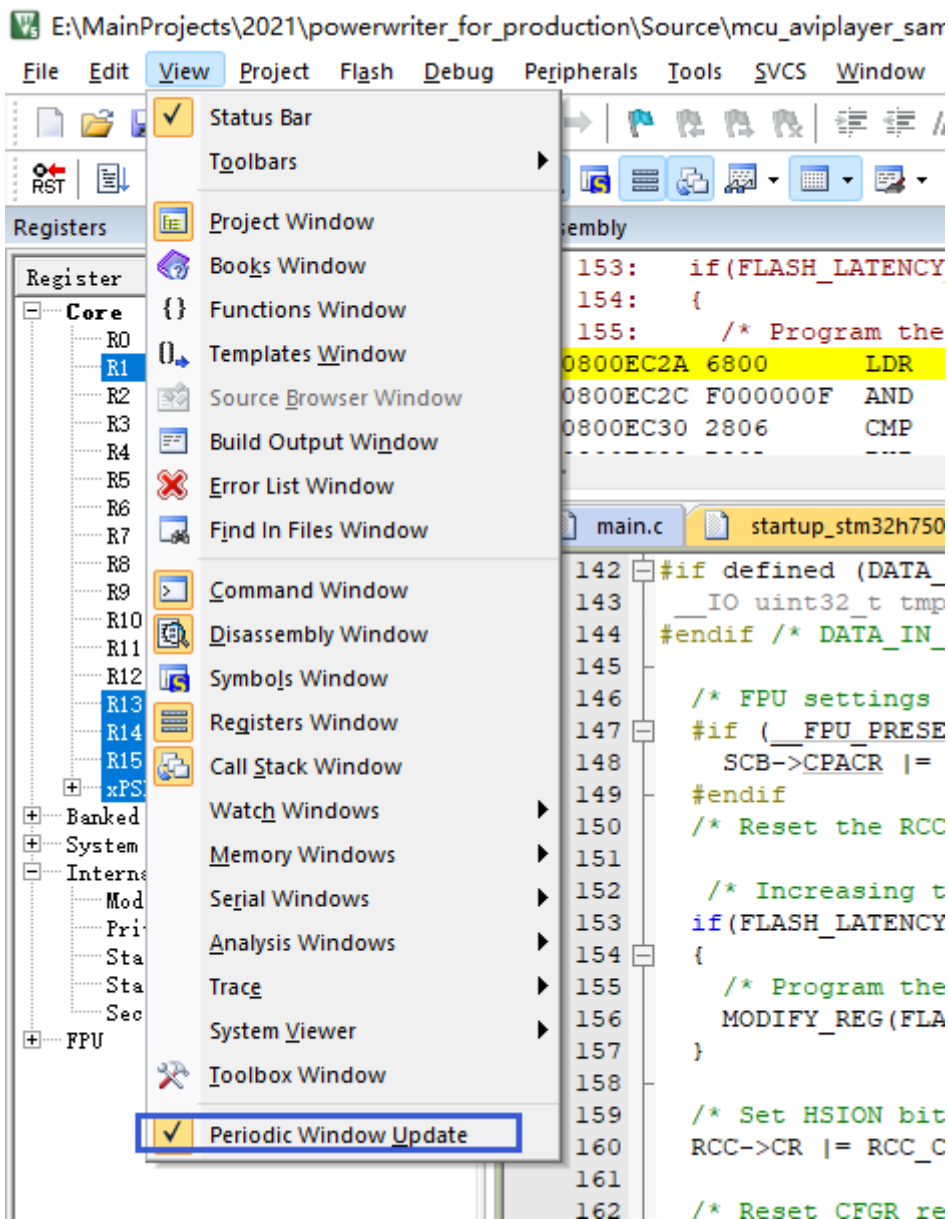
使用PowerWriter 连接上目标芯片后，执行以下操作擦除所有数据：

- 选项字节，恢复默认，然后写入。
- 擦除Program Memory 程序空间数据。

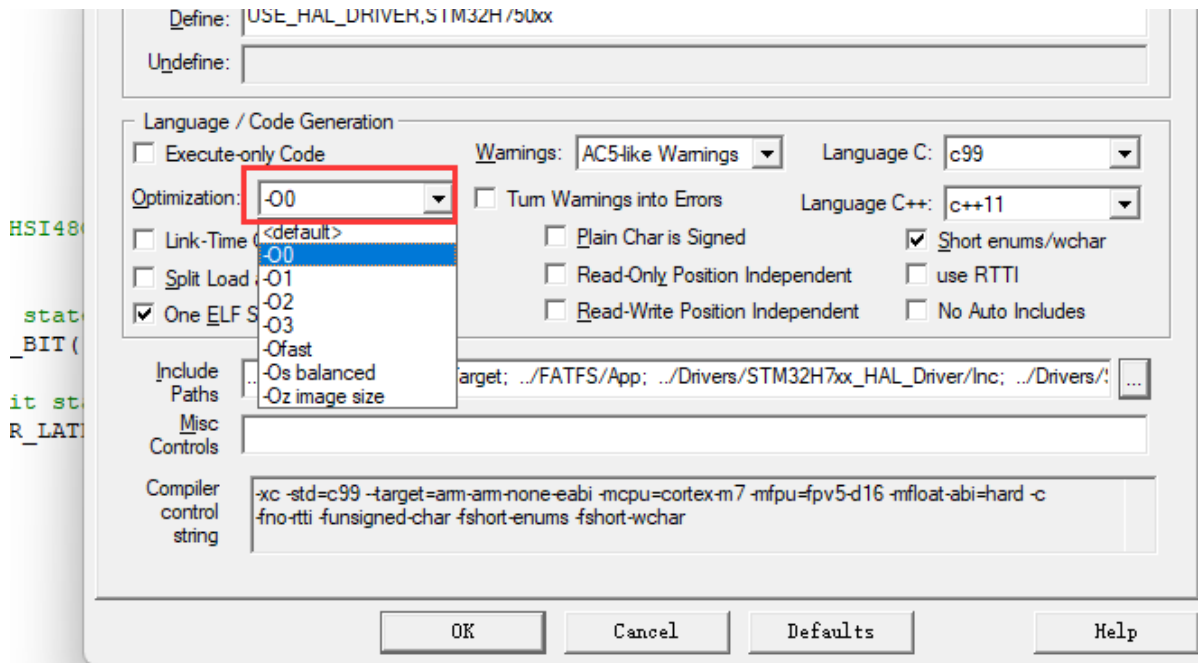
执行以上操作后，再去MDK 中调试。

### 2.1.7.3.12 Watch 窗口变量不刷新或显示灰色

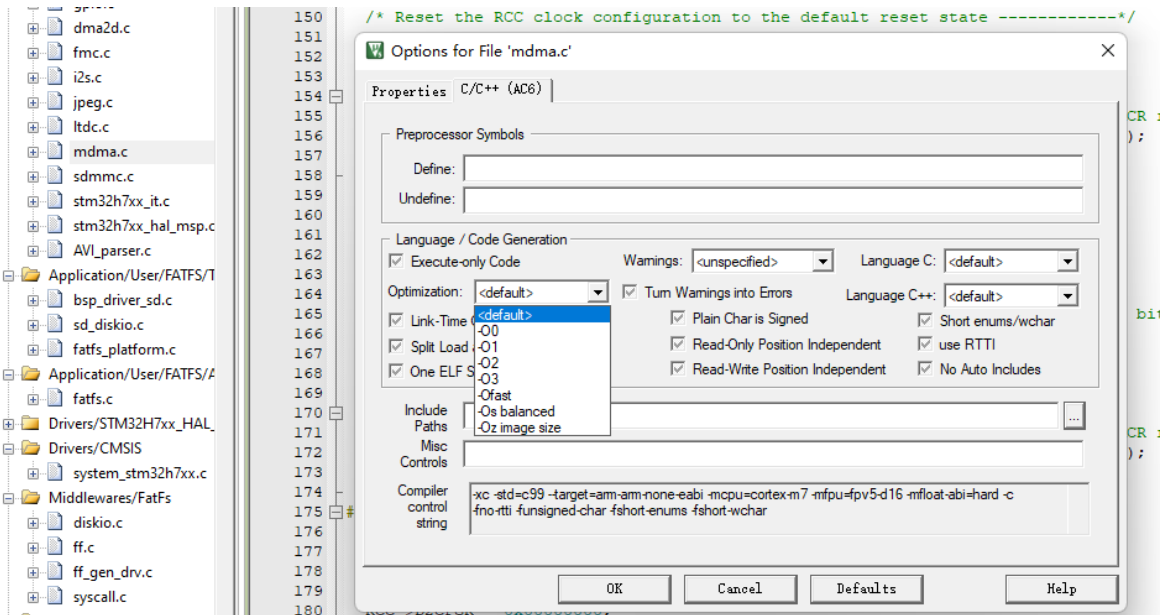
数据没刷新情况，是由于没有勾选 **Periodic Window Update**，进入调试状态后，从菜单的 View -> Periodic Window Update 勾选此功能，调试器，Watch 窗口的数据将会实时刷新，如下图所示：



变灰色的原因，在于编译器优化级别过高，导致实际生成的调试信息和源码断点位置不完全一致而出现同步的问题，此时可以通过调整整个项目的优化级别，如改为0级无优化后，源码断点位置和实际变量的访问完全一致，如下图所示：



此外，可以单独调整部分源码的优化级别，在MDK 中通过源码文件(.c、.cpp) 右键打开文件属性，单独设置文件的优化级别，如下图所示：



:::tip 提示

[下载本页PDF文件](#)

:::