

# GD32 MCU 连接问题和调试方法总结

## V1.0

### 一、前言

使用 GD32 MCU 过程中大家常常有因为错误的应用导致无法连接 MCU, 还有只知道 IDE 硬件仿真调试 MCU。本文 GD32F303 为例子, 笔者总结下工作中所遇到连接问题和一些调试方法, 抛砖引玉引导大家自行检查和学习, 愉快的畅玩 GD32 MCU @^~@

### 二、调试方法

① 首先为大家介绍 jlink 的调试利器“Jlink cmd”, 它可以方便用户在非仿真的情况下, hold 内核、单步、全速、设置断点、查看内核和外设寄存器、读取 flash 代码等等, 方便大家拥有最高的权限查看在运行中的 MCU 情况, 查找非 IDE 仿真情况下, MCU 运行异常的原因。

#### 1. 如何打开 Jlink cmd :

在 jlink 安装版本目录下可以打开, 无安装版本可以在 MDK 的安装目录的“Segger”文件夹下打开 cmd。具体可如下图所示。

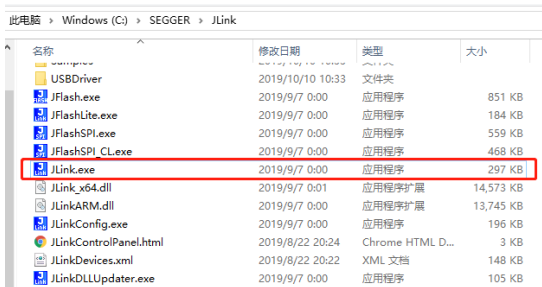


图 1 jlink 安装版本目录

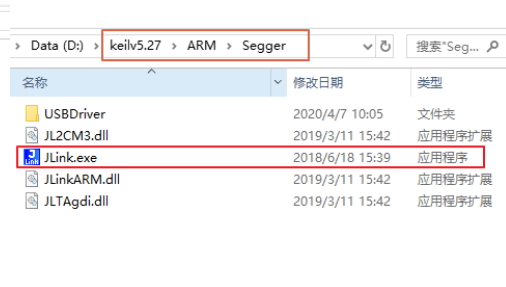


图 2 MDK 的安装

目录

#### 2. 如何使用 Jlink cmd :

首先使用 Jlink 连接好 MCU, 打开软件“Jlink.exe”

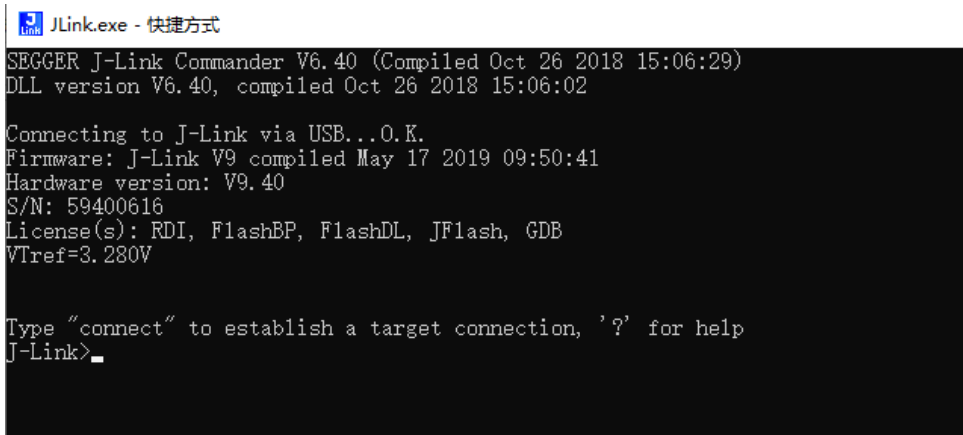


图 3

按照提示输入相应信息 connect->?->s->回车 (旧版只需要输入 usb 即可连接目标芯片)

```

Type "connect" to establish a target connection, '?' for help
J-Link>connect 1.输入connect连接目标芯片
Please specify device / core. <Default>: GD32F303RE
Type '?' for selection dialog
Device>? 2.输入? 选择对应的芯片型号 或者选择对应的内核
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>s 3.接口按照实际情况填写 swd/jtag
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed> 4.设置下载速度 可按实际情况输入
Device "GD32F303RG" selected.

```

图 4

正确的链接目标芯片后：

```

AP[1]: Stopped AP scan as end of AP map has been reached
AP[0]: AHB-AP (IDR: 0x24770011)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0xE00FF000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ E00FF000 5.识别到内核
ROMTbl[0][0]: E000E000, CID: B105E00D, PID: 000BB000 SCS
ROMTbl[0][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[0][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[0][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[0][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
ROMTbl[0][5]: E0041000, CID: 00000000, PID: 00000000 ???
Cortex-M4 identified.
J-Link>h 6.输入 h 可以hold内核, 查看内核寄存器
PC = 08000568, CycleCnt = B2C4B7D1
R0 = 00000001, R1 = 00000004, R2 = 40011000, R3 = 200000A5
R4 = 080007D0, R5 = 080007D0, R6 = 082FE024, R7 = CA21FD9E
R8 = 9836EDC3, R9 = 19B090EB, R10= AF7D007E, R11= CDD28E7B
R12= 00000010
SP(R13)= 200004C0, MSP= 200004C0, PSP= 484B3BAC, R14(LR) = 08000567
XPSR = 01000000: APSR = nzcvd, EPSR = 01000000, IPSR = 000 (NoException)
CFBP = 00000000, CONTROL = 00, FAULTMASK = 00, BASEPRI = 00, PRIMASK = 00

FPS0 = C1BA6C57, FPS1 = F087743F, FPS2 = D5879DC4, FPS3 = 850458A1
FPS4 = 9082EDA4, FPS5 = F75DB028, FPS6 = 3FB04589, FPS7 = 7E639BE7
FPS8 = 7A30C13C, FPS9 = D81E1224, FPS10= C2469636, FPS11= A72AF385
FPS12= 1CD6D3DF, FPS13= 0B803340, FPS14= 3E5AC915, FPS15= 77E705F1
FPS16= B4B2460D, FPS17= 7CA9963B, FPS18= C2E5B74C, FPS19= CCE0E73B
FPS20= 4C18EAFE, FPS21= 4B8B8D8D, FPS22= A122484C, FPS23= DE75DB18
FPS24= D1FBB6B5, FPS25= BAB33EAB, FPS26= 95FAA0DE, FPS27= DEB80CCA
FPS28= 7B07AE57, FPS29= 2E877004, FPS30= 5D9E3F7B, FPS31= 3BEB1DC4
FPSCR= 00000000
J-Link>mem 0x1fff800 10 7.输入mem 指令可以查看选项字节等信息
1FFF800 = A5 5A FF FF FF FF FF FF FF FF FF FF FF FF FF FF
J-Link>_

```

图 5

(如上图所示，选项字节里为“A5 5A”即为无保护状态)

**Jlink command 常用命令简介：**

mem	读内存
mem8	读 8 字节内存
mem16	读 16 字节内存
mem32	读 32 字节内存
w1	写 8 字节内存
w2	写 16 字节内存
w4	写 32 字节内存
h	停止 cpu 运行的程序
setbp	设置断点
g	跳到代码段地址执行
s	单步执行 (调试用)
r	复位

② 接下来向大家介绍一款用户程序里无需软件配置可以抓取对应的全局变量数据，并绘制波形的软件——“J-Scope”

1. 首先前往 segger 官网下载对应的软件安装包：  
<https://www.segger.com/downloads/jlink/#JScope>

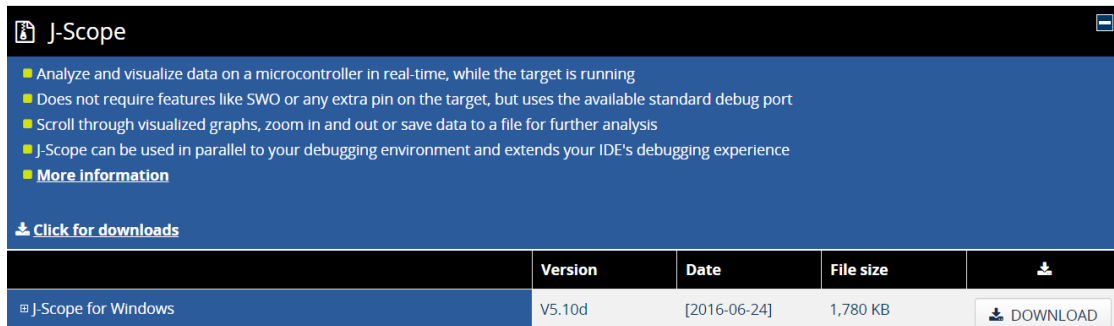


图 6

2. 打开正确安装的软件，选择创建一个新的工程，选择对应的调试接口和速度，在 ELF 文本框内添加，KEIL 工程下生成的.axf 文件或者 IAR 工程下生成的.out 文件。在下一个界面勾选添加想要查看的全局变量（数组可以先输入对应的序号）。最后点击红色开始按钮，选择对应的芯片型号或者内核，既可以绘制对应变量的波形。

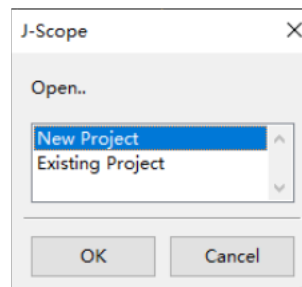


图 7

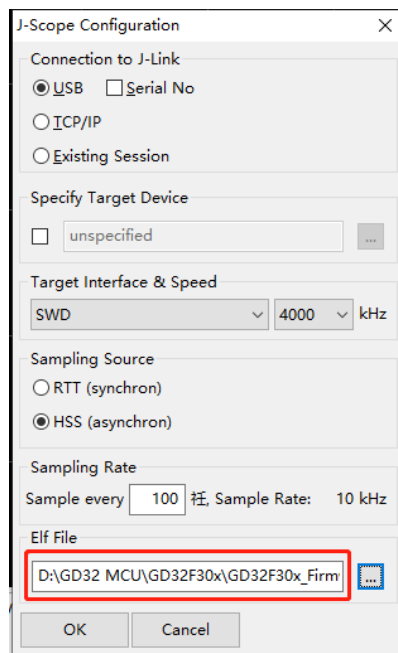


图 8

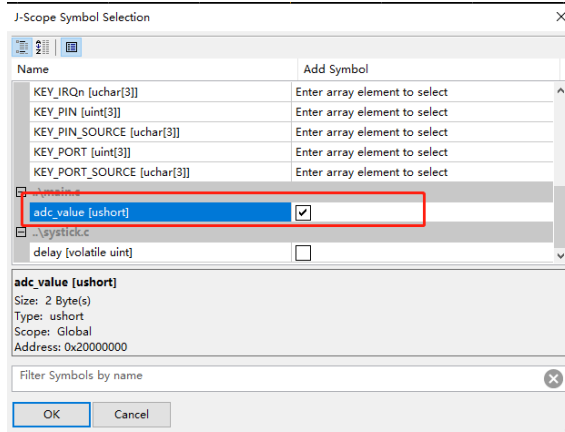


图 9

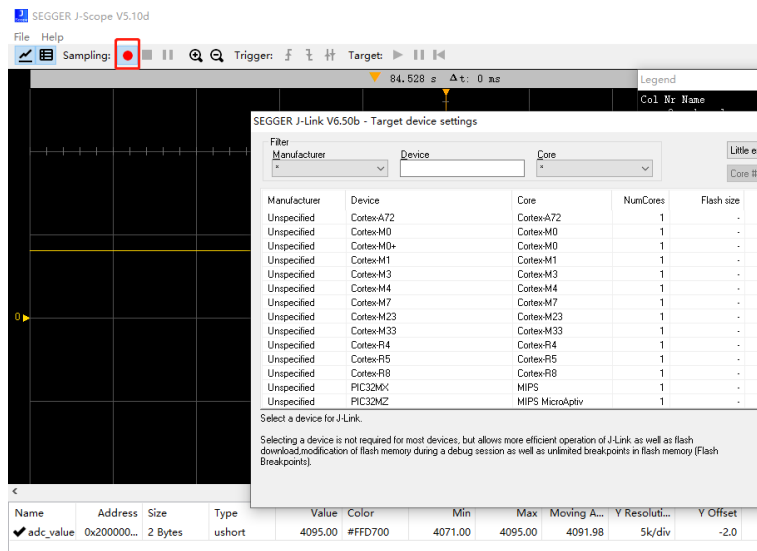


图 10

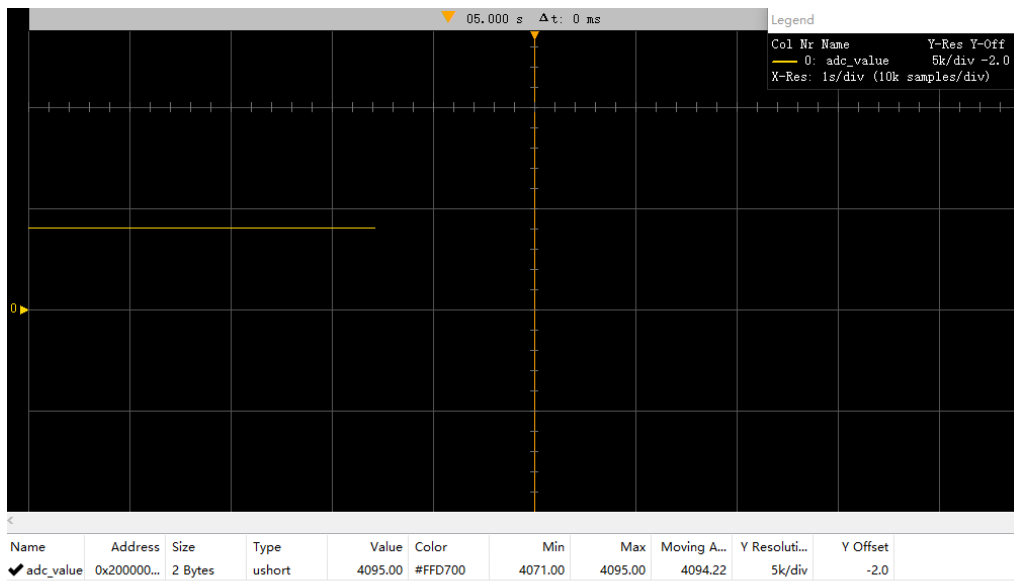


图 11

### 三、 连接烧录问题总结

#### 1. 在 MDK 内使用 jlink 调试器无法识别设备 ID

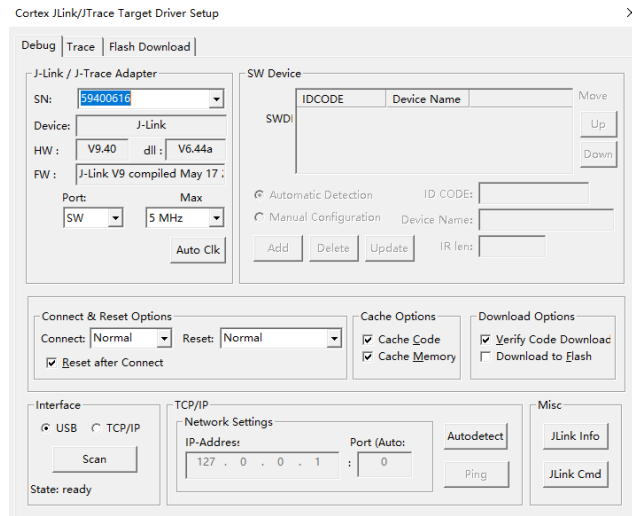


图 12

原因及解决办法：

1. 检查 jlink 是否正确连接目标芯片，芯片引脚 PA13 PA14 是否连接正确、虚焊，PA13 和 PA14 是否有接反现象。
2. 可能为 mcu 设置了高级读保护状态。仿真口已经熔断，MCU 内部程序依然正常运行。
3. 若连接的是 GD32E23x 系列，需要保证 jlink 硬件版本 v9 及以上，软件版本 v6.0 以上版本。如果硬件版本过低请换 v9 jlink，若软件版本过低，需要将图 1 安装版本内的对应的文件覆盖图 2 MDK Segger 文件夹内的文件。重新打开 MDK 既可以出现新版的驱动。
4. 若以上判定均不是，可以先使用前文所述，使用 jlink cmd 连接目标芯片，查看是否连接成功，若连接成功表示芯片是 OK 的，如图 13 所示需要将 MDK 工程目录下的 jlink 配置删除掉，从新打开软件或者从新在此配置的左下角点击扫描 device 设备。如图 10 所示在跳出的 unknown 设备提示框下，点击“ok”后选好对应的 mcu 型号或者对应的内核。若连接也不成功可查看 1 和 2 。

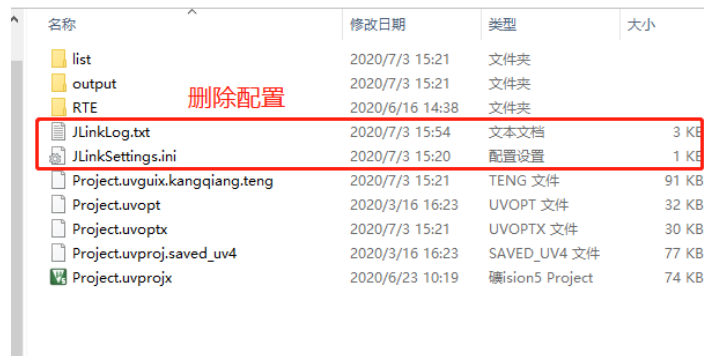


图 13

5. 程序里有配置 remp JTAG 和 SWD 接口，无法识别 id 使用 SWD 或者 JTAG 下载。可将 BOOT0 拉高后复位，即可以识别 id 下载。（支持大部分系列）  
使用外部晶振，可将晶振短接后下载。  
将 BOOT0 拉高复位，使用 isp 擦除后下载。

2. 仿真器配置里可以识别 id，但是下载却报错。无法实现下载。

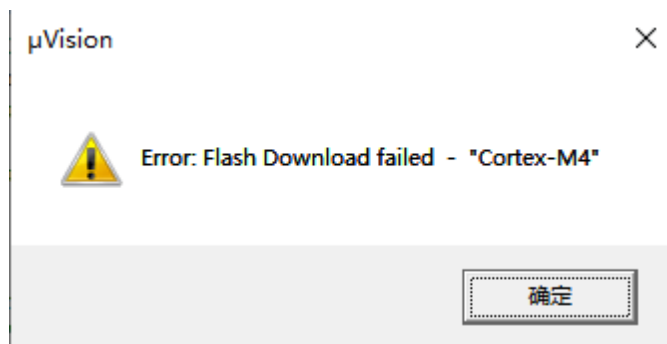


图 14

原因及解决办法：

1. 首先确保 Flash Download 里面的 Flash 算法是有正确的添加，有时候在转换工程或选择另外 MCU 型号 会导致 Flash 算法里面的 Device Size 和 Address Range 对应的大小不匹配，需要删除后重新添加。

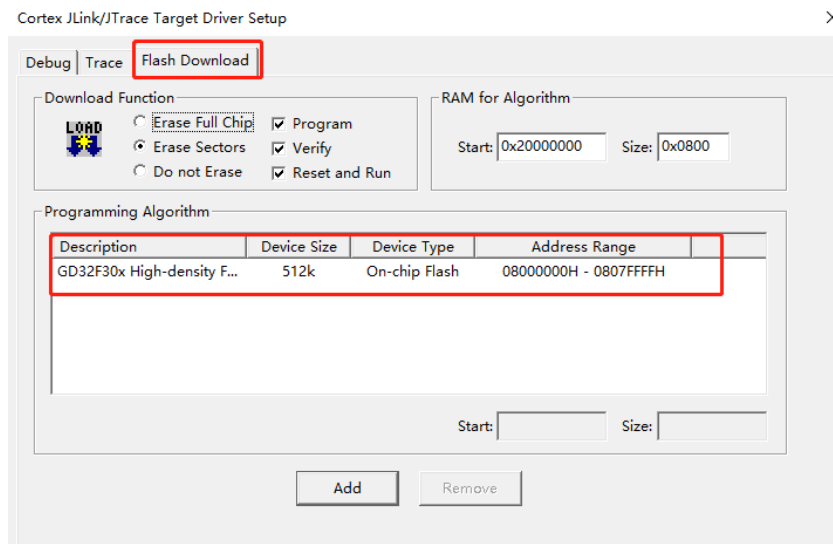


图 15

2. 芯片具有读保护或者写保护。可以使用调试方法①来看当前选项字节的信息状况  
mem 0x1ffff800 10 (F4xx 系列为 mem 0x1fffc001 10)  
正常无读保护状态的值为 A5 5A 无写保护的状态为 FF (如图 5 所示)  
若有保护可以使用 ISP gdlink 上位机 jlink 对 MCU 进行解锁，正常解锁后既可以下载。
3. MCU 因为晶振配置跑飞 (特别 E103 历程里面有 8M 和 25M 晶振的开发板)，无法正常下载。  
可将 BOOT0 拉高后复位下载。(支持大部分系列)  
使用外部晶振，可将晶振短接后下载。  
将 BOOT0 拉高复位，使用 isp 擦除后下载。
4. MCU 处在上电即进入低功耗模式，无法正常下载。解决方法可如上。

3. 能正常下载，但是调试有问题：地址非正确引导到 flash 的起始地址或者卡在 main 之前的地址。

原因及解决办法：

1. 注意 debug 配置里为硬件仿真，不然会卡在时钟初始化里面。

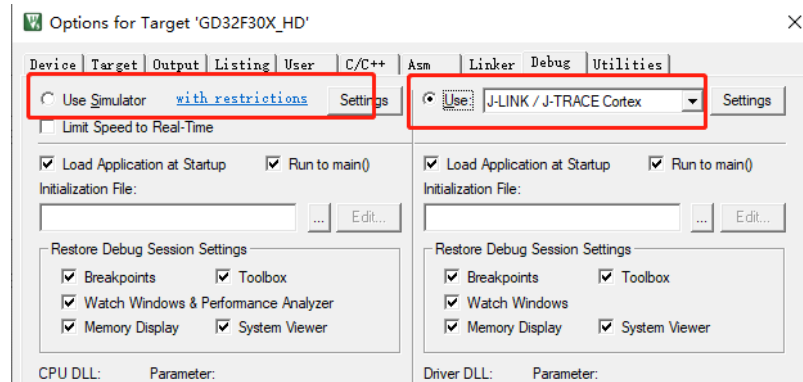


图 16

2. 检查 boot0 是否接地，不然 pc 指针跳转的地址不正确。
3. 在 IAP 应用中，检查对应的下载地址。

4. 仿真正常，但是下载后运行不正常。

1. 检查 boot0 连接方式，在 IAR 仿真环境下会自动跳转到下载程序的 main 函数。下载后就需要保证 BOOT0 接地。