
MM32 MCU Debug Kit 使用说明

MM32 MCU Debug Kit 主要是协助 MM32 用户解决、分析和追溯芯片的定位工具，方便 MM32 MCU 用户快速定位程序进入 Hardfault 的原因以及脱离 IDE 对芯片进行出厂初始化操作等。

1.目的

一、快速追溯芯片版本和出货时间等芯片信息；

1、基本信息：

内核版本

内核型号

UID

2、选项字节：

RDP

nBOOT1

WRP

3、代码区块保护：

区块 1、2、3、4

4、配置：

Register0

Register1

Register2

5、Triming:

HSI_3.3V

HSI_5V

Temp

Vref

6、ISP:

ISP_DATA0

ISP_DATA1

二、快速定位 MCU 进 hardfault 的原因，跟踪显示内核寄存器状态，方便根据内核信息获取程序运行状态，目前仅支持用户编程文件定位问题，针对用户操作 RCC 等设置问题暂时还没想到方法支持；

三、针对芯片进行出厂格式化操作，目前支持对所有的芯片进行出厂格式化操作功能；

2.支持芯片

支持全系列 MM32 MCU 芯片，包含 m/n/o/p/q/s 版。

3.支持工具

支持 J-link，后续需 MM32-LINK 驱动开放可以支持。

4.操作说明



图 1 读取芯片信息

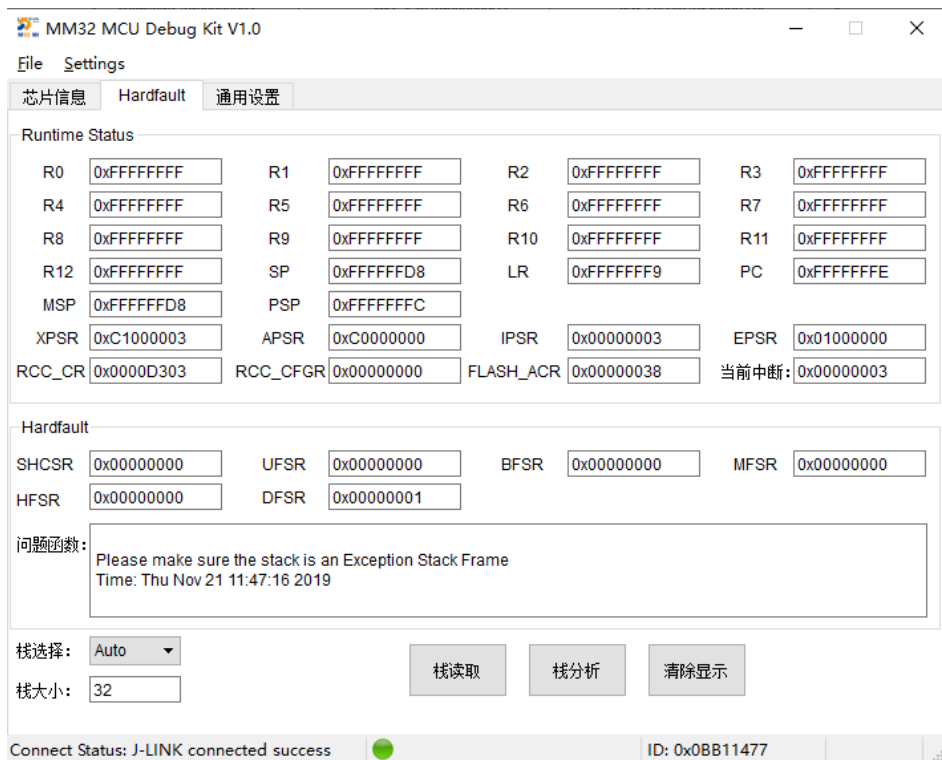


图 2 定位 Hardfault 原因

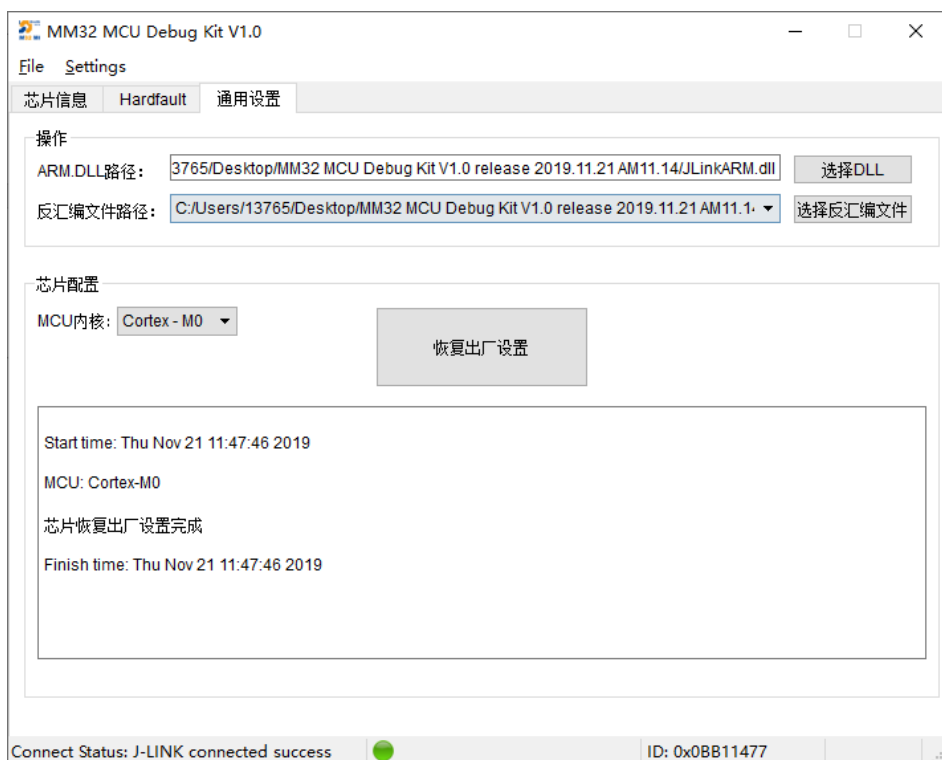


图 3 恢复出厂设置和读 Flash

第一步：添加驱动和反汇编文件，进入通用设置界面添加驱动文件和反汇编文件，目前先支持 J-Link，后面继续支持其它调试器。

JLinkARM.dll 是加载 Jlink 的驱动文件，默认驱动文件下面已经工具的文件夹中，用户也可以选择自己电脑安装 J-link 驱动路径下面的 JLinkARM.dll 文件。

反汇编文件是用来自动定位 Hardfault 故障函数，如果用户不需要，可以不设置。反汇编文件支持 MDK 和 IAR 工程，设置方式如下：

MDK 反汇编指令：fromelf --text -a -c -o "\$L@L.asm" "#L"

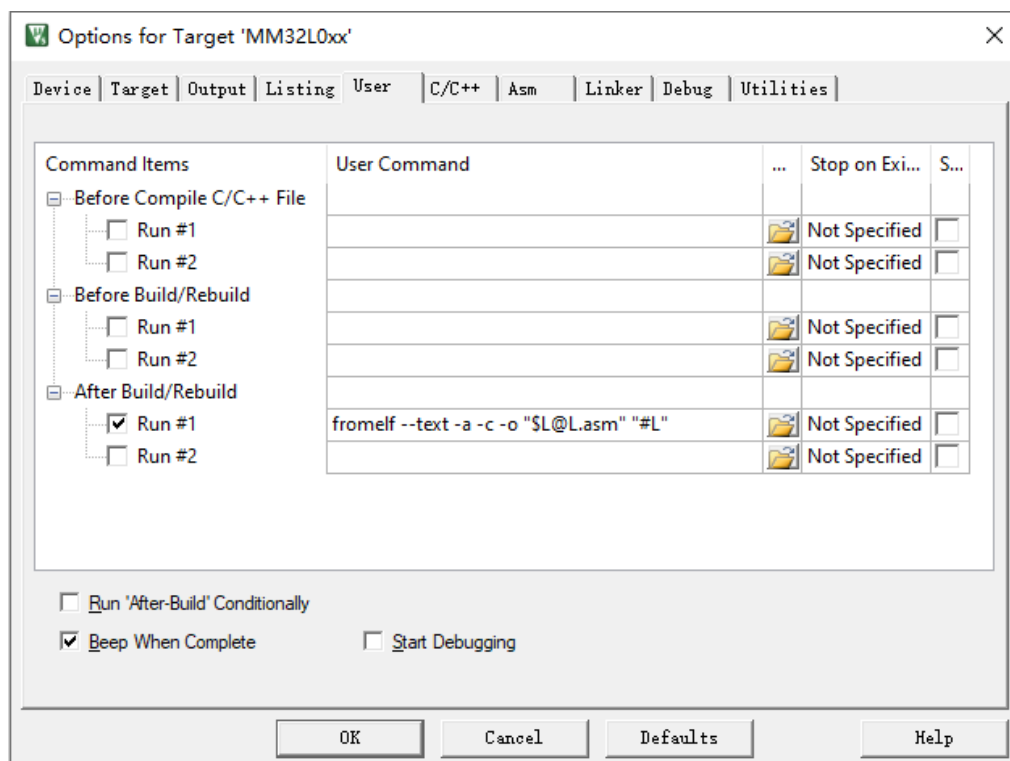


图 4 MDK 生成反汇编

IAR 反汇编指令：ielfdumparm --code --source \$TARGET_PATH\$ -o \$TARGET_PATH\$.dis ， IAR 里面设置如图五：

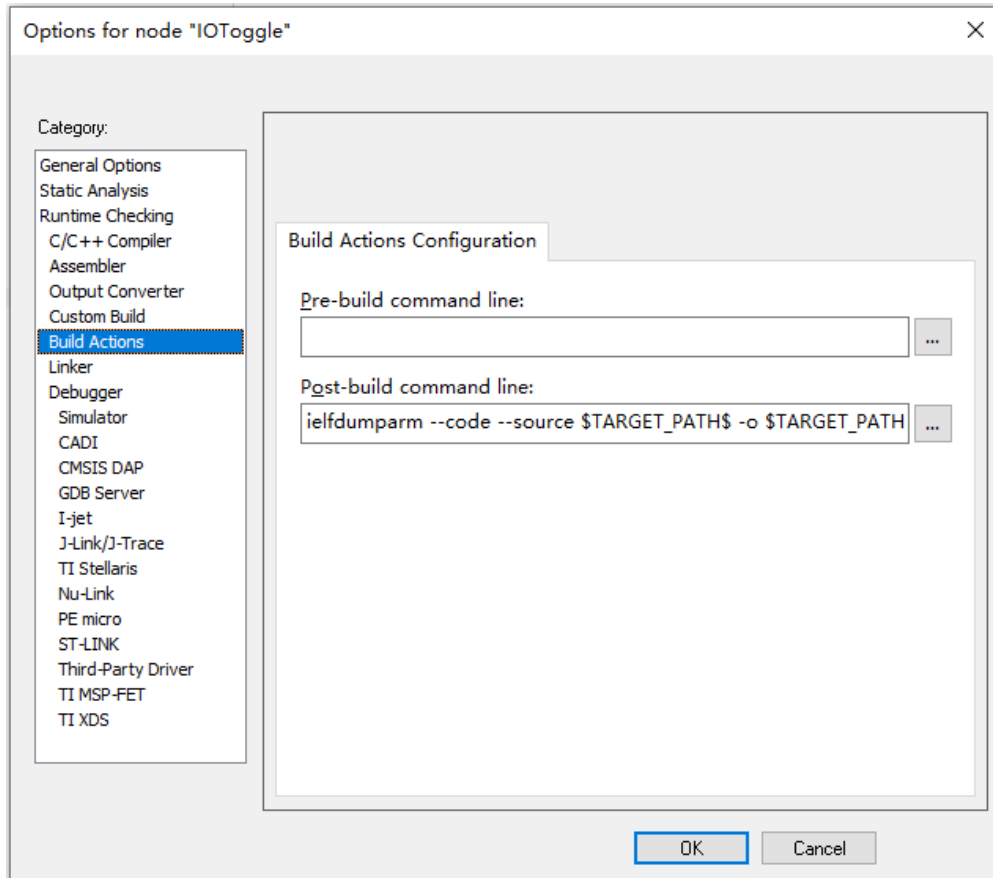


图 5 IAR 生成反汇编

第二步连接 J-LINK，需要先接上 J-LINK，点击菜单栏的 **Setting** 子菜单 **Connect**，连接成功会显示图六，下面的状态栏会显示 **J-LINK connected success**、MCU 的连接状态、内核 ID。下面的状态栏会显示绿灯，连接失败会显示红灯。



图 6 J-LINK 连接成功状态栏

如果没有连接上 MCU 会显示如图七，会显示连接 JLINK 功能，但是没有连接上 MCU。



图 7 MCU 连接失败状态栏

第三步读取芯片信息。



图 8 读取芯片信息

第四步解析 Hardfault，点击栈读取和栈解析，窗口会显示 ARM 内核寄存器信息以及 Hardfault 出现问题的函数。

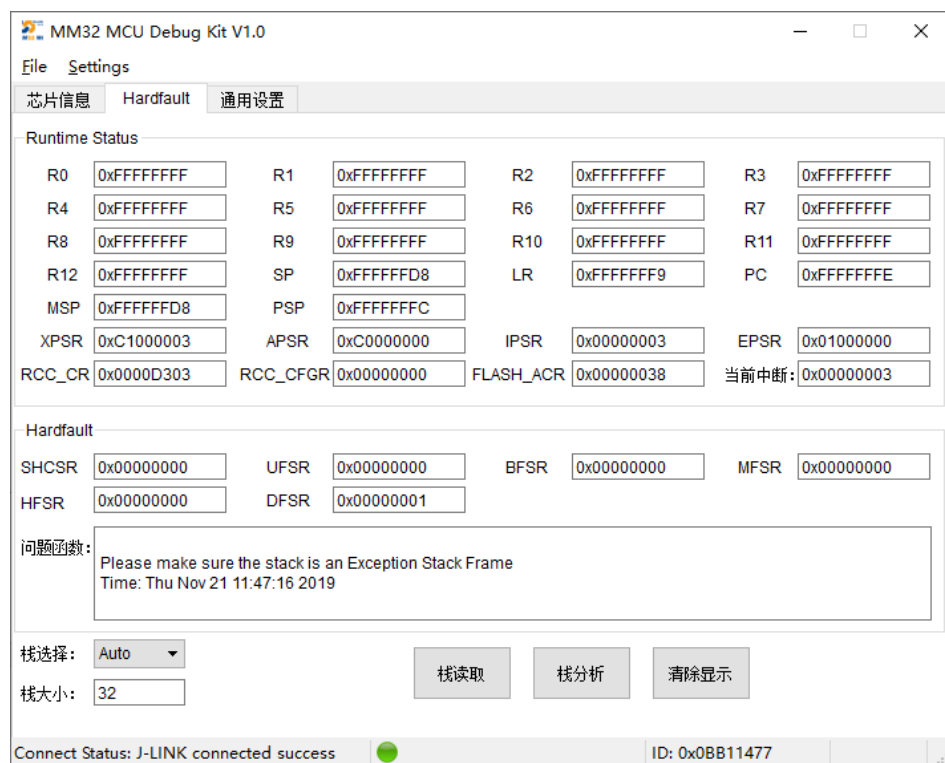


图 9 定位 Hardfault 原因

第四步恢复出厂设置，需要选择对应的内核版本才能进行恢复出厂设置。



图 10 恢复出厂设置

5.信息另存

用户读取信息后直接 **Ctrl + s** 另存即可将读取的芯片信息另存在工具的文件夹内，只需提供 MM32_MCU_INFO.ini 文件即可。

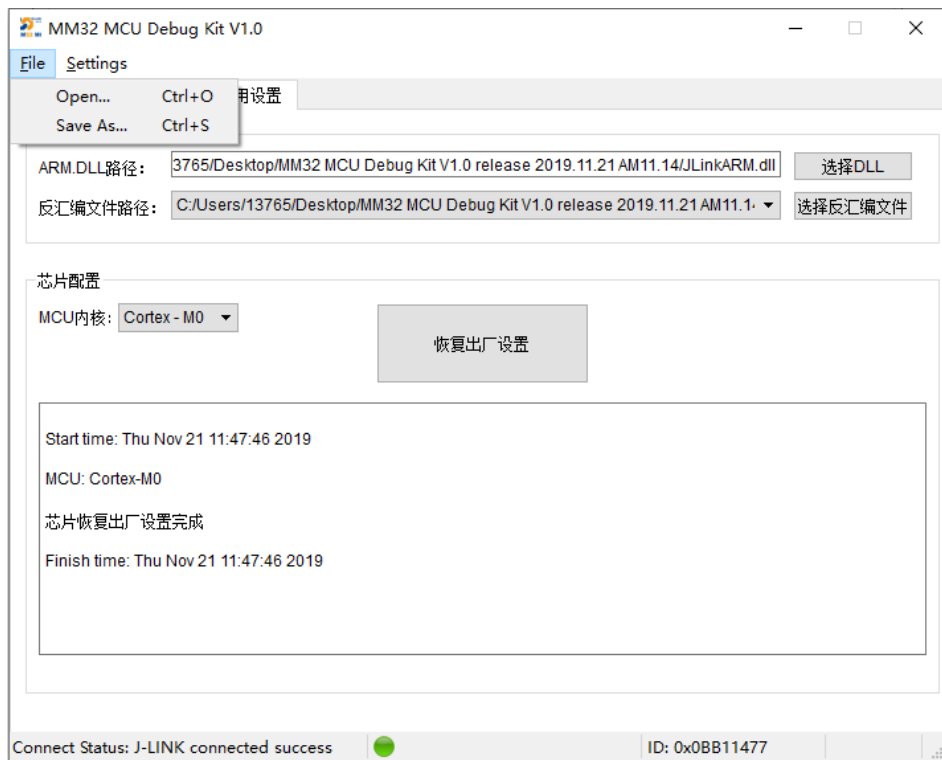


图 11 信息另存