

概述

本文描述了当代码完全运行在 RAM 的情况下，如何使用 IDE 进行仿真调试。

适用范围	
SPC1125 系列	SPC1125, SPC1128, SPD1121
SPC1168 系列	SPC1155, SPC1156, SPC1158, SPC1168, SPD1148, SPD1178, SPD1188, SPD1163, SPM1173
SPC2168 系列	SPC2168, SPC2165, SPC2166, SPC1198
SPC1169 系列	SPC1169, SPD1179, SPD1176, SPD1177, SPD1179B
SPC2188 系列	SPC1185, SPC2188
SPC1198B 系列	SPC1198B

- 注意：
1. SPC2168 的 CAU 核，默认运行在 RAM 中，不在此文档适用范围内，详细内容请参考《SPC2168 CAU 双核调试指南》。
 2. SPC2188 的 CPU1 核，默认运行在 RAM 中，不在此文档适用范围内，详细内容请参考《SPC2188 双核调试指南》。

目录

1	安装 IDE	6
2	J-LINK 与目标板连接	6
3	检查 SYSTEM 中 Debug 模式寄存器	9
4	检查 Debug 控制位	10
5	KEIL IDE 调试 RAM 中的程序	13
5.1	配置代码地址	13
5.1.1	Keil UI 配置代码地址	13
5.1.2	Sct 文件配置代码地址	14
5.2	配置 ini 文件	14
5.3	取消 Flash 配置	17
5.4	调试	19
6	IAR IDE 调试 RAM 中的程序	20
6.1	配置代码地址	20
6.2	配置 mac 文件	20
6.3	取消 Flash 配置	21
6.4	调试	22

图片列表

图 2-1: J-LINK 接口	6
图 5-1: 采用 UI 配置代码地址.....	13
图 5-2: 采用 sct 文件配置代码地址.....	14
图 5-3: J-LINK	15
图 5-4: .ini 文件与 Keil 配置代码地址关系	16
图 5-5: .ini 文件与 Keil 配置代码地址关系	16
图 5-6: DEBUG 工具读取目标 ID	17
图 5-7: Flash Download 配置	18
图 5-8: Debug warning.....	19
图 5-9: Debug 仿真.....	19
图 6-1: 采用 icf 文件配置代码运行地址	20
图 6-2: J-LINK	21
图 6-3: Flash Download 配置	21
图 6-4: Debug warning.....	22
图 6-5: 选中 Cortex-M4.....	22
图 6-6: Debug 仿真.....	22

表格列表

表 2-1: SWD 接口信号定义	6
表 2-2: J-LINK 与各芯片引脚连接	7
表 2-3: 芯片与 BOOT 电平	7
表 2-4: 芯片调试接口电平	8
表 4-1: 配置字的构成及其描述	10
表 4-2: 配置字的构成及其描述	11
表 4-3: 配置字的构成及其描述	11
表 4-4: 配置字的构成及其描述	11
表 4-5: 配置字的构成及其描述	12
表 4-6: 配置字的构成及其描述	12

版本历史

版本	日期	作者	状态	变更
C/0	2024-06-04	HangSu	已过期	1. 首次发布。
C/1	2024-08-26	LemengZhou	已过期	1. 修改为全系列通用文档。
C/2	2025-03-31	Xq.he	已发布	1. 添加 SPC1198B 系列平台支持。 2. 适用范围增加。

1 安装 IDE

不同的 KEIL 及 IAR 版本，会有不同的配置方式，在本文中 KEIL 版本为 MDK523，IAR 版本为 EWARM-CD-8324-20889。

2 J-LINK 与目标板连接

J-LINK 适配器支持 2 种接口，如图 2-1 所示，推荐使用 SWD 接口，因为更省引脚。

图 2-1: J-LINK 接口

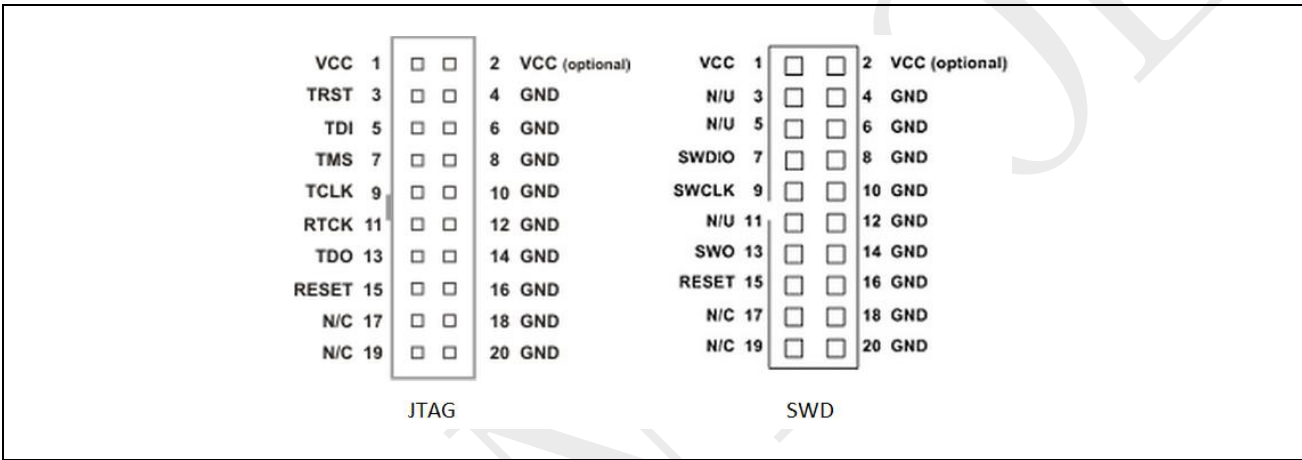


表 2-1: SWD 接口信号定义

信号	连接
SWDIO	数据 I/O 引脚
SWCLK	时钟引脚
VCC	电源电压正端，引脚可选
GND	数字地
RESET	RSTIN 引脚，引脚可选
SWO	串行数据输出，引脚可选

J-LINK 与各型号芯片的硬件连接如表 2-2 所示。

表 2-2: J-LINK 与各芯片引脚连接

芯片型号	SWD 引脚	
	SWDIO	SWCLK
SPC1169 系列	GPIO17	GPIO18
SPC1168 系列	GPIO38	GPIO39
SPC2168_CPU, SPC1198, SPC2166_CPU, SPC2165_CPU,	GPIO49	GPIO48
SPC2188_CPU0	GPIO80	GPIO81
SPC1185	GPIO80	GPIO81
SPC1125 系列	GPIO38	GPIO39
SPC1198B	SWD (芯片固定 Pad)	SWCK (芯片固定 Pad)

- 注意:
1. J-LINK 调试时, TRSTN 和 BOOT 电平必须与表 2-3 一致;
 2. J-LINK 下载器端口电压需要与芯片端口电压
 3. 表 2-4 一致。

表 2-3: 芯片与 BOOT 电平

芯片型号	BOOT	TRSTN
SPC1169 系列	X	高
SPC1168 系列	高	高
SPC2168, SPC1198	高	高
SPC2188	X	高
SPC1185	X	高
SPC1125 系列	X	高
SPC1198B 系列	X	

[1] 图中 X 代表高低电平都可以。

[2] SPC1198B 系列没有 TRSTN 引脚。

注意: 芯片对应的 BOOT 和 TRSTN 管脚号, 可通过查询对应的数据手册进行确认。

表 2-4：芯片调试接口电平

芯片	调试接口电平
SPC1168 系列，SPC1125 系列，SPC2168 系列，SPC2188 系列， SPC1198B 系列	3.3V
SPC1169 系列	5V

- 注意：
1. 给 J-Link 引脚 1（VCC）接入
 2. 表 2-4 参考电压前，需要确保其上无电压（部分 J-Link 调试器默认给引脚 1（VCC）接入 3.3V 电平），否则会烧芯片；
 3. 具有默认电压（3.3V）的 J-Link 引脚 1（VCC），可以和调试接口电平为 3.3V 的芯片正常通信。

3 检查 SYSTEM 中 Debug 模式寄存器

如果发现通过 SWD 接口无法 Debug 连接芯片，可通过 ISP Tool 工具下载串口打印代码，打印 Debug 模式寄存器的值，用以进行以下检查。

如果是 SPC1169 系列，需要检查 SYSTEM->DBGIFCTL 是否为 0（SWD）；

如果是 SPC1185，需要检查 SYSTEM->DBGIFCTL 是否为 4（SWD for CPU0）；

如果是 SPC2188，需要检查 SYSTEM->DBGIFCTL 是否为 4（SWD for CPU0）或 6（SWD for CPU0 and CPU1）；

SPC1168 系列，SPC2168 系列，SPC1125 系列，SPC1198B 系列不需要做以上检查。

4 检查 Debug 控制位

如果发现通过 SWD 接口无法 Debug 连接芯片，可通过 ISP Tool 工具下载串口打印代码，打印 Debug 控制位的值，用以进行以下检查。

如果是 SPC1168 系列，SPC2168 系列如表 4-1 所示。如果任意分区被保护，Debug 功能就不会开启，因此必须均为 0xFFFFFFFF。

表 4-1：配置字的构成及其描述

地址	配置字名称	配置字描述
0x1100060C	ZONE0_FLASH_PROT	Flash 分区 0 保护字段 0xFFFFFFFF：关闭 Flash 分区 0 保护 其他值：使能 Flash 分区 0 保护
0x11000614	ZONE0_RAM_PROT	RAM 分区 0 保护字段 0xFFFFFFFF：关闭 RAM 分区 0 保护 其他值：使能 RAM 分区 0 保护
0x1100064C	ZONE1_FLASH_PROT	Flash 分区 1 保护字段 0xFFFFFFFF：关闭 Flash 分区 1 保护 其他值：使能 Flash 分区 1 保护
0x11000654	ZONE1_RAM_PROT	RAM 分区 1 保护字段 0xFFFFFFFF：关闭 RAM 分区 1 保护 其他值：使能 RAM 分区 1 保护
0x1100068C	ZONE2_FLASH_PROT	Flash 分区 2 保护字段 0xFFFFFFFF：关闭 Flash 分区 2 保护 其他值：使能 Flash 分区 2 保护
0x11000694	ZONE2_RAM_PROT	RAM 分区 2 保护字段 0xFFFFFFFF：关闭 RAM 分区 2 保护 其他值：使能 RAM 分区 2 保护
0x110006CC	ZONE3_FLASH_PROT	Flash 分区 3 保护字段 0xFFFFFFFF：关闭 Flash 分区 3 保护 其他值：使能 Flash 分区 3 保护
0x110006D4	ZONE3_RAM_PROT	RAM 分区 3 保护字段 0xFFFFFFFF：关闭 RAM 分区 3 保护 其他值：使能 RAM 分区 3 保护
0x11000700	WDT_ENABLE	看门狗使能字 0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗

如果是 SPC1125 系列，如表 4-2 所示。

表 4-2：配置字的构成及其描述

地址	配置字名称	配置字描述
0x11000700	WDT_ENABLE	看门狗使能字 0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗
0x1100060C 0x11000614	CHIP_SECURITY	芯片调试接口锁定字 均为 0xFFFFFFFF：芯片调试接口将不被锁定 其他数值：芯片调试接口将被锁定

如果是 SPC1169 系列的产品，如表 4-3 所示。

表 4-3：配置字的构成及其描述

地址	配置字名称	配置字描述
0x1001FFF8	WDT_ENABLE	看门狗使能字 0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗
0x1001FFFC	CHIP_SECURITY	芯片调试接口锁定字 0xFFFFFFFF：芯片调试接口将不被锁定 其他数值：芯片调试接口将被锁定

如果是 SPC2188，如表 4-4 所示。

表 4-4：配置字的构成及其描述

地址	配置字名称	配置字描述
0x100FFFF8	WDT_ENABLE	看门狗使能字 0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗
0x100FFFC	CHIP_SECURITY	芯片调试接口锁定字 0xFFFFFFFF：芯片调试接口将不被锁定 其他数值：芯片调试接口将被锁定

如果是 SPC1185，如表 4-5 所示。

表 4-5：配置字的构成及其描述

ECC	地址	配置字名称	配置字描述
开启	0x1003FFF8	WDT_ENABLE	看门狗使能字
关闭	0x1007FFF8		0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗
开启	0x1003FFFC	CHIP_SECURITY	芯片调试接口锁定字
关闭	0x1007FFFC		0xFFFFFFFF：芯片调试接口将不被锁定 其他数值：芯片调试接口将被锁定

如果是 SPC1198B 系列，如表 4-6 所示。

表 4-6：配置字的构成及其描述

地址	配置字名称	配置字描述
0x000C4C04	CHIP_SECURITY	芯片调试接口锁定字 0xFFFFFFFF：芯片调试接口将不被锁定 其他数值：芯片调试接口将被锁定
0x000C4C08	WDT_ENABLE	看门狗使能字 0xFFFFFFFF：芯片启动时禁用看门狗 其他数值：芯片启动时启用看门狗

5 KEIL IDE 调试 RAM 中的程序

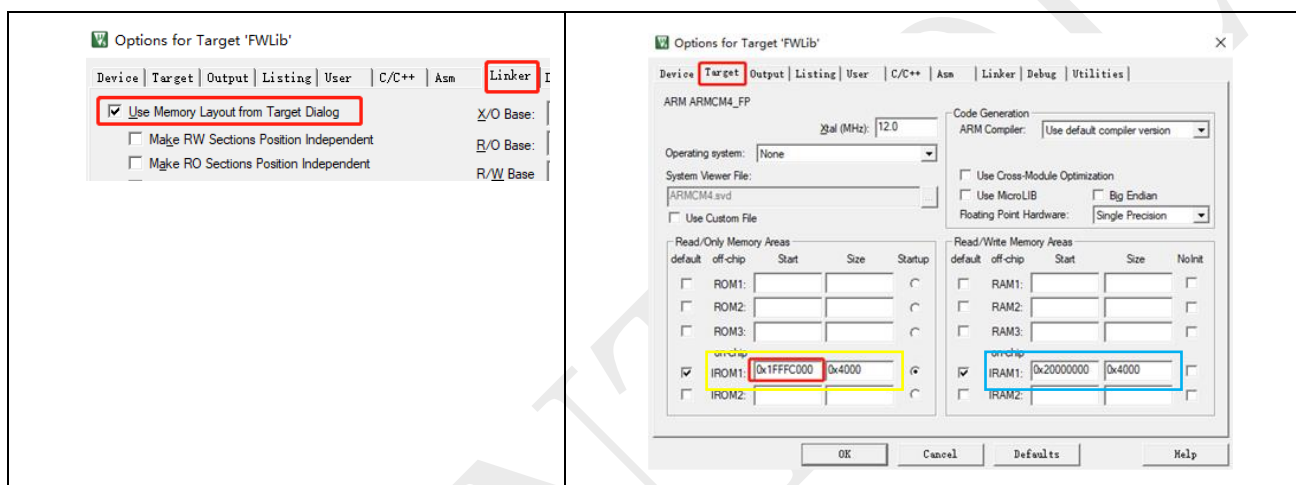
5.1 配置代码地址

5.1.1 Keil UI 配置代码地址

将 Keil 配置文件中代码的存放位置调整到 RAM 中。

如果“Linker”页 “Use Memory Layout from Target Dialog” 被勾选，则采用如图 5-1 所示 UI 改法，否则采用如图 5-2 所示 sct 文件改法。

图 5-1: 采用 UI 配置代码地址



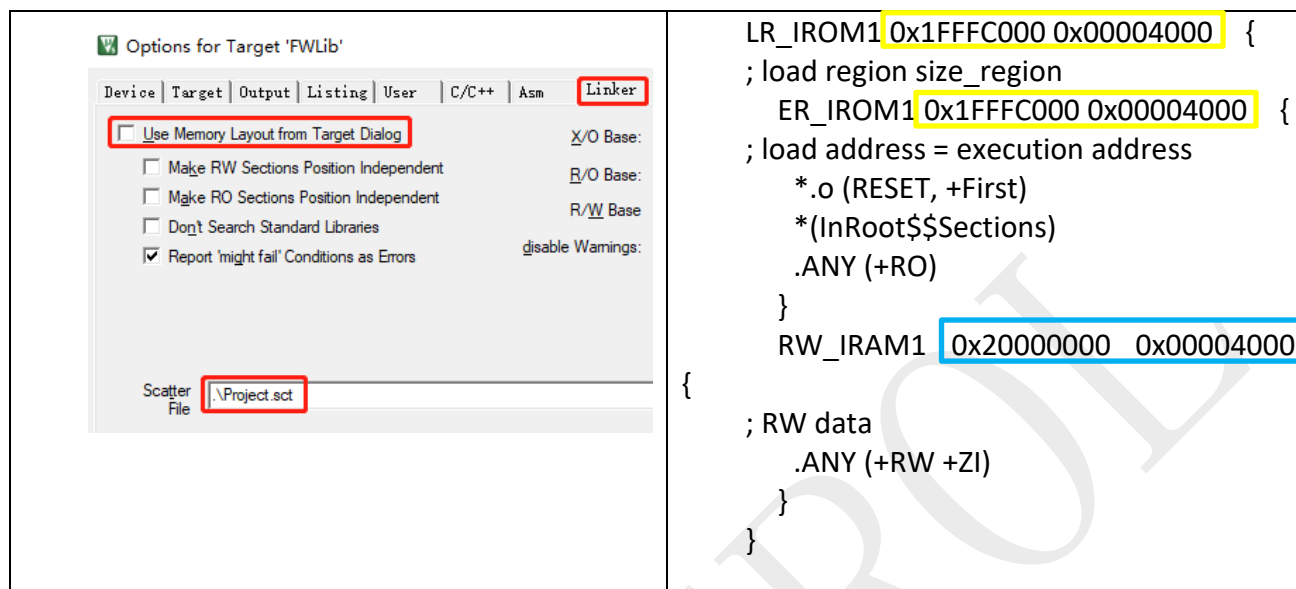
[1] 上图以 32K RAM SPC1169 芯片为例

[2] 黄框内为 SPC1169 的代码段存储范围，蓝框内为数据段存储范围。

- 注意：
1. 本文所述为如何在 RAM 运行程序，因此代码段存储范围也在应在 RAM 区。
 2. 不同型号芯片需要根据 TRM 手册，查询其 RAM 区在存储空间上的地址映射范围，并在 RAM 区中进行代码段与数据段的划分。
 3. 代码段与数据段的地址范围不可有重合部分。
 4. 需要特别注意图中黄框中的起始地址，在 ini 文件中将会继续使用到这个地址信息。

5.1.2 Sct 文件配置代码地址

图 5-2: 采用 sct 文件配置代码地址



- [1] 上方 Project.sct, 是以 32K RAM SPC1169 芯片为例。
- [2] 黄框内为 SPC1169 的代码段存储范围, 蓝框内为数据段存储范围。

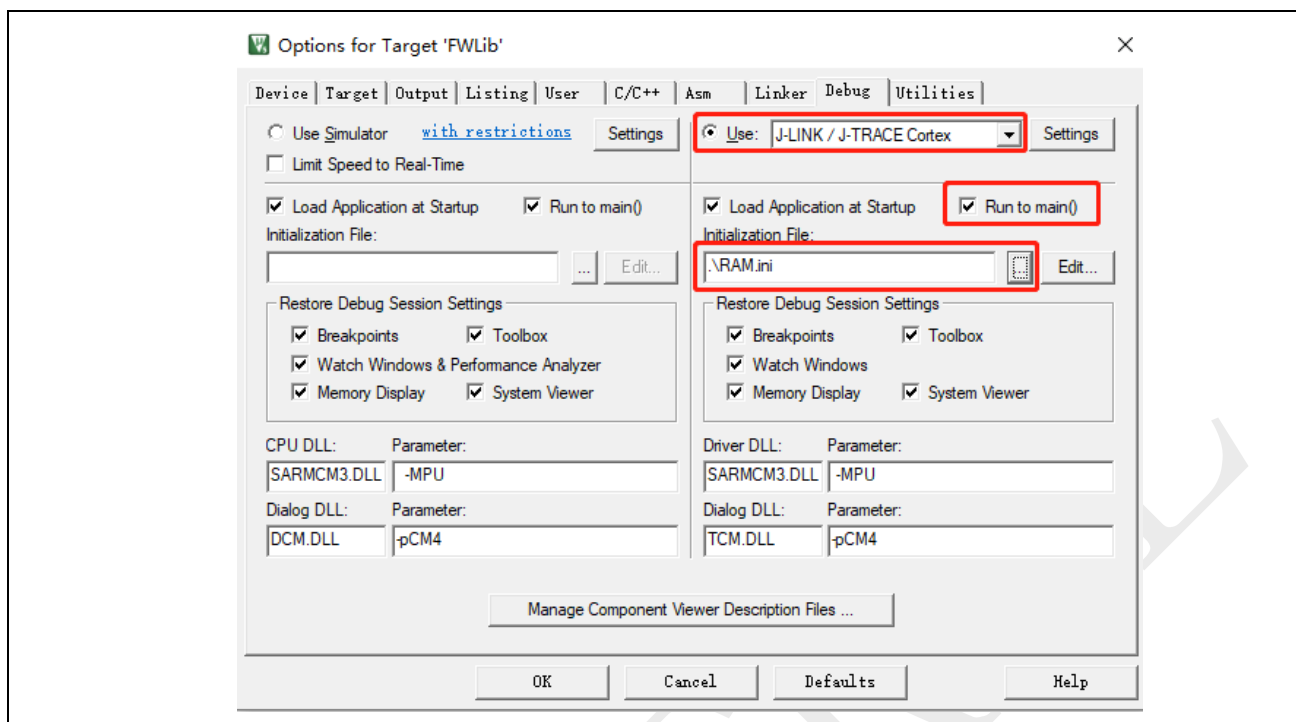
- 注意:
1. 本文所述为如何在 RAM 运行程序, 因此代码段存储范围也在应在 RAM 区。
 2. 不同型号芯片需要根据 TRM 手册, 查询其 RAM 区在存储空间上的地址映射范围, 并在 RAM 区中进行代码段与数据段的划分。
 3. 代码段与数据段的地址范围不可有重合部分。
 4. 需要特别注意图中黄框中的起始地址, 在 ini 文件中将会继续使用到这个地址信息。

5.2 配置 ini 文件

与 DEBUG Flash 中代码不同, DEBUG RAM 中代码时, 需要告知跳转 PC, 以及堆栈信息等, 这需要 ini 的文件的帮助。

选中 Keil IDE “Debug” 页, 如图 5-3 所示。选择 J-LINK, 勾选 “Run to main”, 加载 ini 文件。

图 5-3: J-LINK



ini 文件内容

```

FUNC void Setup (void)
{
    SP = _RDWORD(0x1FFFC000);           // Setup Stack Pointer
    PC = _RDWORD(0x1FFFC004);           // Set Program Counter
    _WDWORD(0xE000ED08, 0x1FFFC000);    // Set Vector Table Offset Register
}

_WDWORD(0x40001024, 0x1ACCE551);       // Enable WDT0 operation
_WDWORD(0x40001124, 0x1ACCE551);       // Enable WDT1 operation
_WDWORD(0x40001008, 0x0);              // Close WDT0
_WDWORD(0x40001108, 0x0);              // Close WDT1

LOAD %L INCREMENTAL                    // Download to RAM
Setup();
//g, main

```

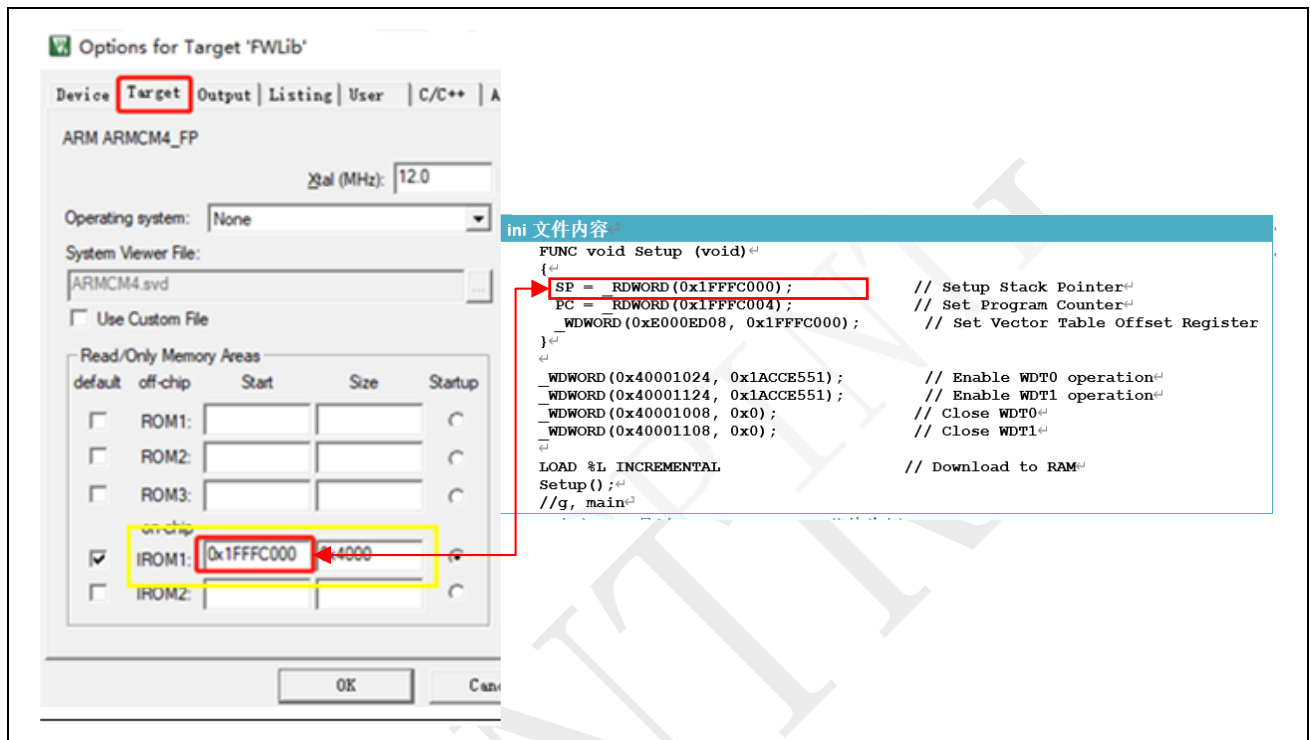
[1] 上方.ini, 是以 32K RAM SPC1169 芯片为例。

注意： 由于在图 5-3 中勾选了“Run to main()”选项，所以在 ini 文件中就不需要再写“g, main”语句，如果写了，将会触发两次进入 main 的动作。

对方文件中重要的几个设置做如下相关说明：

1. .ini 文件与 Keil 配置代码地址的关系，如图 5-4，红色箭头连接的框图内填写的地址需一致。

图 5-4: .ini 文件与 Keil 配置代码地址关系



2. .ini 文件与 sct 文件配置代码地址的关系，如图 5-5，红色箭头连接的框图内填写的地址需一致。

图 5-5: .ini 文件与 Keil 配置代码地址关系



3. SP = _RDWORD(0x1FFFC000) : 调试开始时，将堆栈寄存器初始值设置为 0x1FFFC000 地址存储的内容。

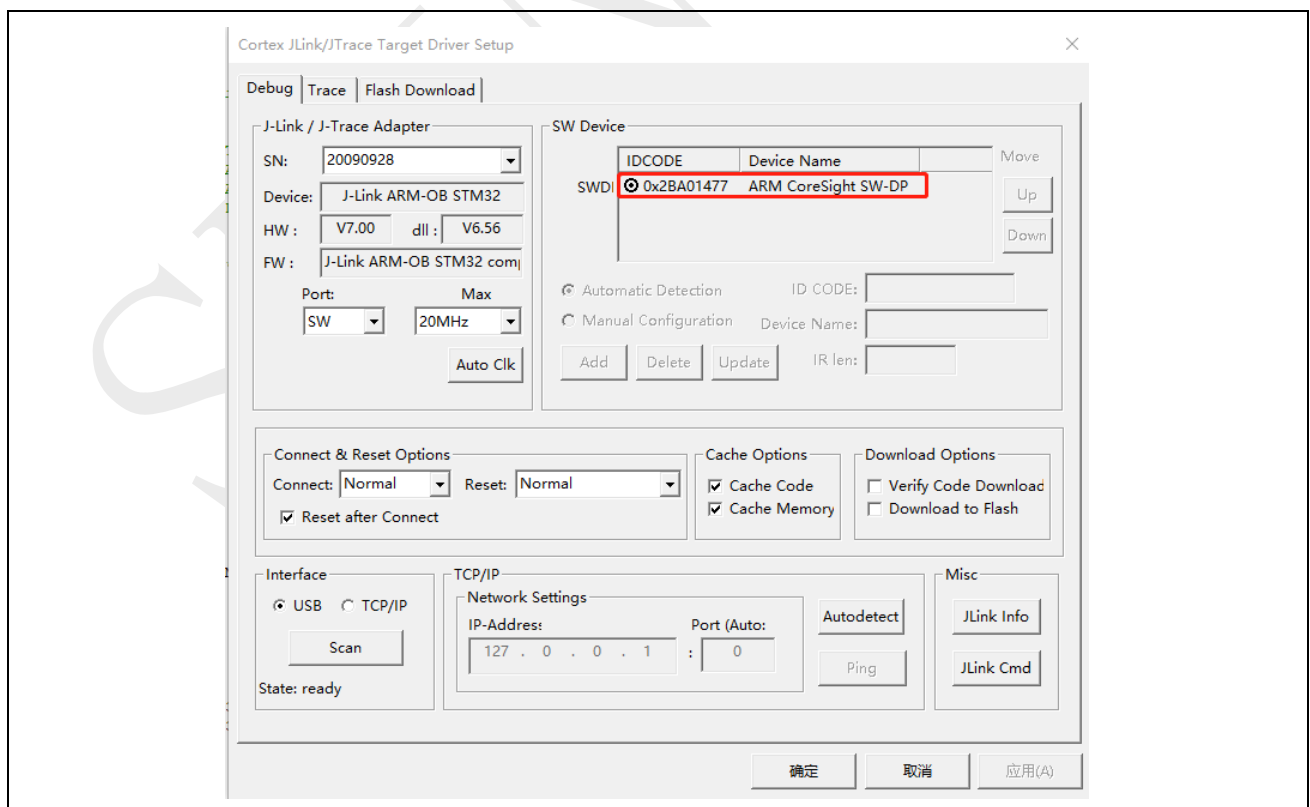
4. PC = _RDWORD(0x1FFFC004): 调试开始时, 将 PC 设置为 “0x1FFFC000+4”, 即 0x1FFFC004 地址存储的内容。
5. _WDWORD(0xE000ED08, 0x1FFFC000): 调试开始时, 将中断向量表偏移寄存器 (地址为 0xE000ED08) 设置为 0x1FFFC000。
6. 虽然默认情况下 WDT 是关闭的, 但为了确保调试一切顺利, 还需要主动关闭 WDT。

- 注意:
1. 因为 UI 界面已勾选 “Run to main”, ini 文件中不能重复配置 g, main。
 2. 请根据当前工程图 5-1 或图 5-2 配置 SP, PC, Vector Table Offset Register 值, “0x1FFFC000” 与图中所配置代码段起始地址一致。
 3. 在添加设备信息时, 请根据具体产品填入 WDT 寄存器地址, 具体信息请仔细阅读相关产品的 TRM。
 4. SP = _RDWORD(“RAM 内存地址”)需与 5.1 章节中为代码段配置的 RAM 区首地址一致。

5.3 取消 Flash 配置

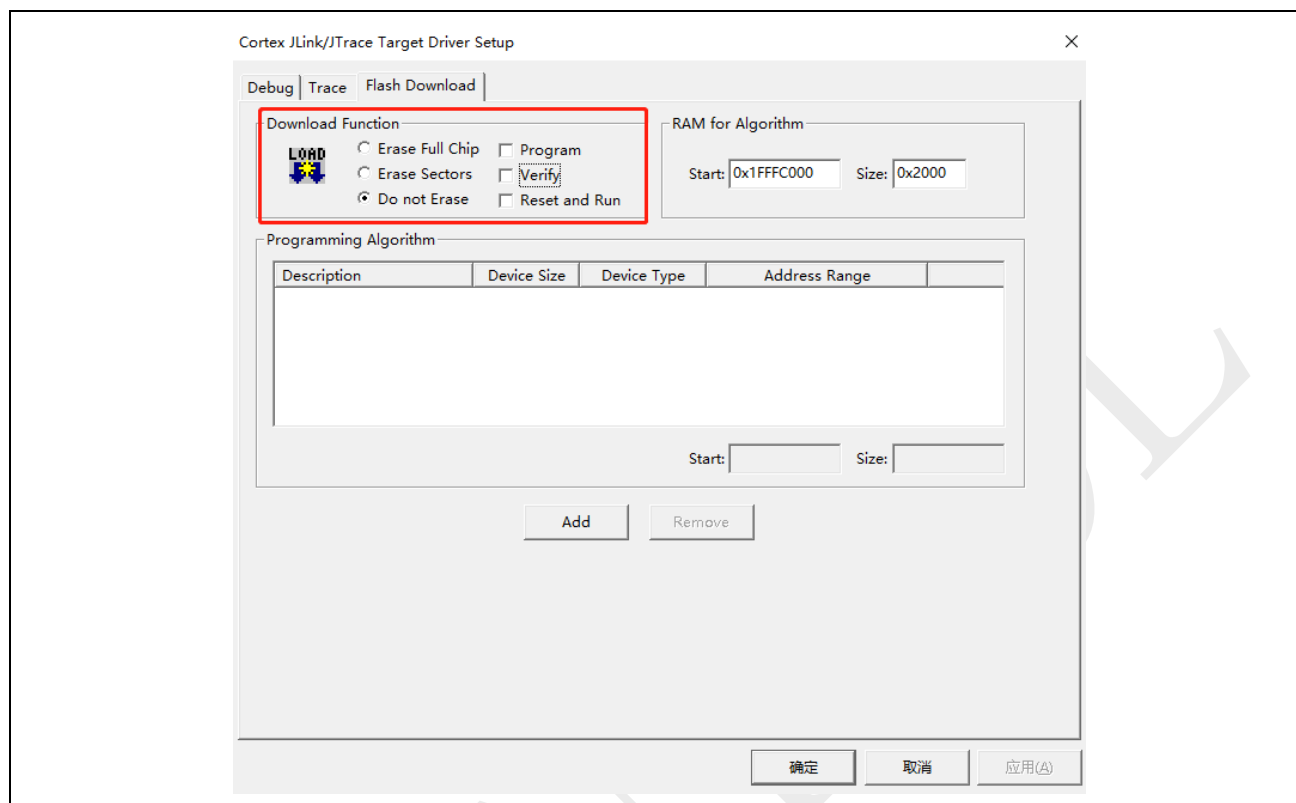
进入 J-LINK “Settings” 页, 识别目标板 ID, 如图 5-6 所示, 如无法识别目标板的 ID, 需回到章节 2, 检查 J-Link 连线以及电平是否配置正确。

图 5-6: DEBUG 工具读取目标 ID



如图 5-7 所示，取消“Flash Download”页中所有 Flash 配置。

图 5-7：Flash Download 配置



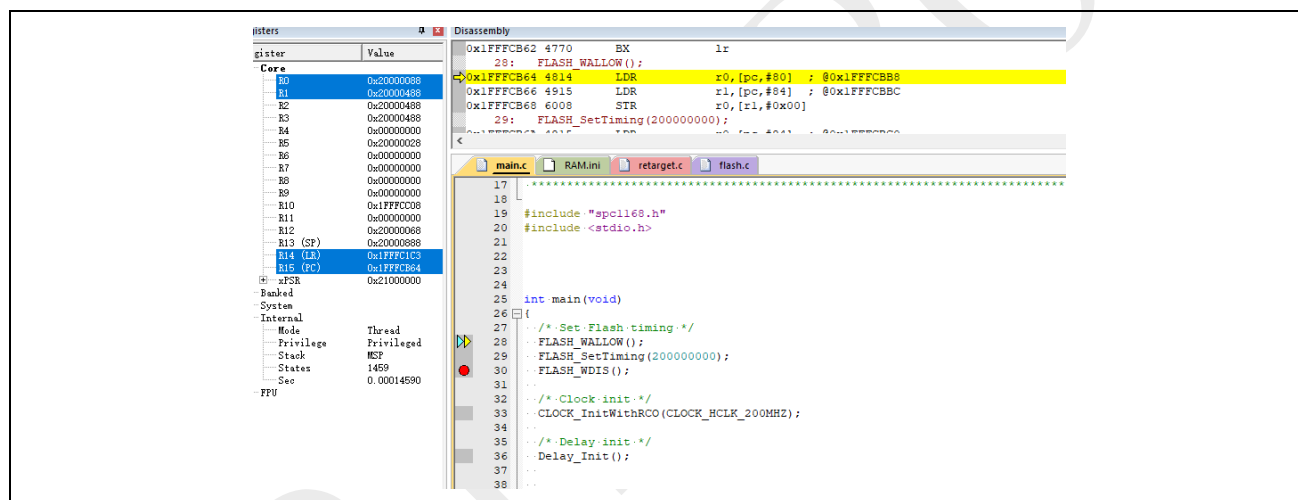
5.4 调试

点击 Keil IDE 中的 Debug 按钮，就可以进行调试，如果产生 Warning，如图 5-8 所示，直接点确定忽略。进入 Debug 仿真界面如图 5-9。

图 5-8: Debug warning



图 5-9: Debug 仿真



如无法进入 Debug 仿真，需回到[章节 5.1](#)，检查所有配置。

6 IAR IDE 调试 RAM 中的程序

6.1 配置代码地址

将 IAR 配置文件 icf 中代码的存放位置改到 RAM 中，如图 6-1 所示。

图 6-1：采用 icf 文件配置代码运行地址

```
/*-Specials-*/
define symbol __ICFEDIT_intvec_start__ = 0x1FFFC000;
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x1FFFC000;
define symbol __ICFEDIT_region_ROM_end__ = 0x1FFFFFFF;
define symbol __ICFEDIT_region_RAM_start__ = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__ = 0x20003FFF;
```

- [1] 上方.icf，是以 32K RAM SPC1169 芯片为例。
- [2] 黄框内为设置中断向量表的首地址，红框内为设置程序代码段的首地址和结束地址，蓝框内为设置程序数据段的首地址和结束地址。

- 注意：
1. 本文所述为如何在 RAM 运行程序，因此代码段存储范围也在应在 RAM 区。
 2. 不同型号芯片需要根据 TRM 手册，查询其 RAM 区在存储空间上的地址映射范围，并在 RAM 区中进行代码段与数据段的划分。
 3. 代码段与数据段的地址范围不可有重合部分。

6.2 配置 mac 文件

mac 文件用来关闭 WDT。虽然默认情况下 WDT 是关闭的，但为了调试一切顺利，还是需要主动关闭 WDT。

在工程下新建 RAM.mac。

mac 文件内容

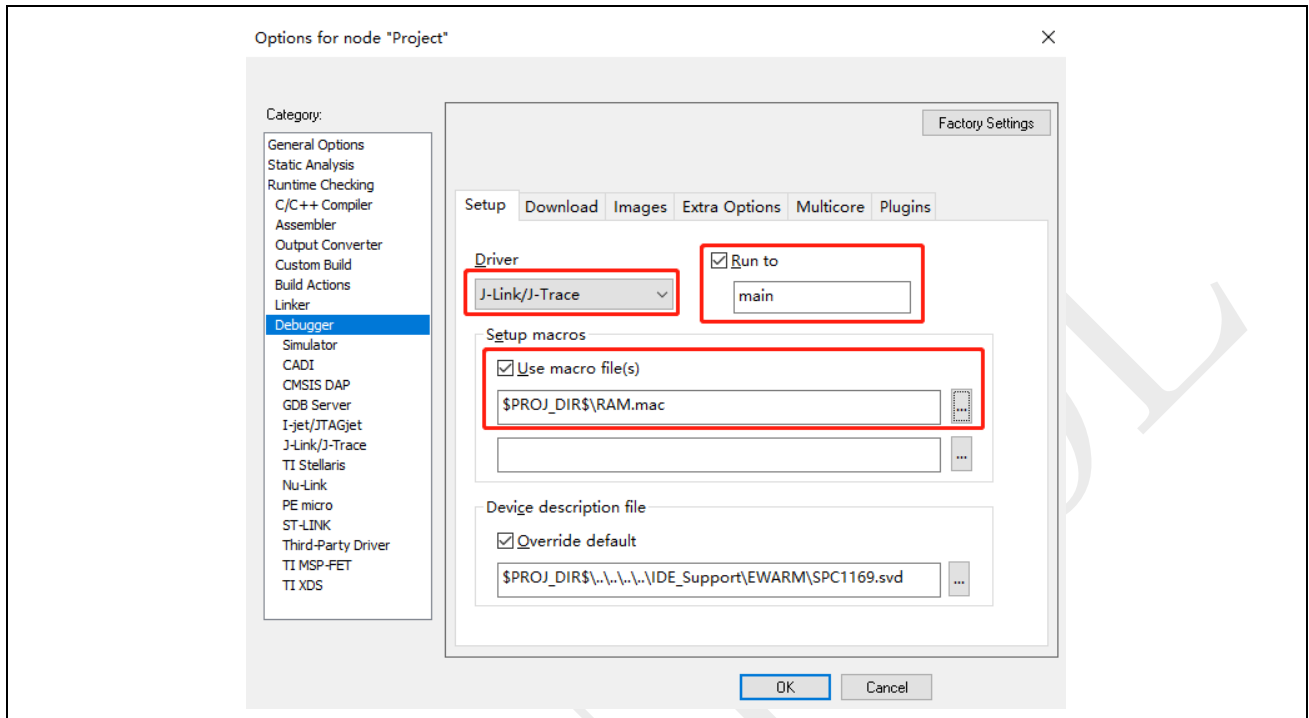
```
execUserSetup()
{
    message"----- execUserSetup -----";
    // Disable WTD
    __writeMemory32(0x1ACCE551, 0x40001024, "Memory");
    __writeMemory32(0x1ACCE551, 0x40001124, "Memory");
    __writeMemory32(0x0, 0x40001008, "Memory");
    __writeMemory32(0x0, 0x40001108, "Memory");
    message"----- execUserSetupEnd -----";
}
```

- [1] 上方 mac 文件，是以 32K RAM SPC1169 芯片为例。

- 注意： 在添加设备信息时，请根据具体产品填入 WDT 寄存器地址，具体信息请仔细阅读相关产品的 TRM。

选中 IAR IDE “Debugger” 中 “Setup” 页，如图 6-2 所示。选择 J-LINK，勾选 “Run to main”，加载 mac 文件。

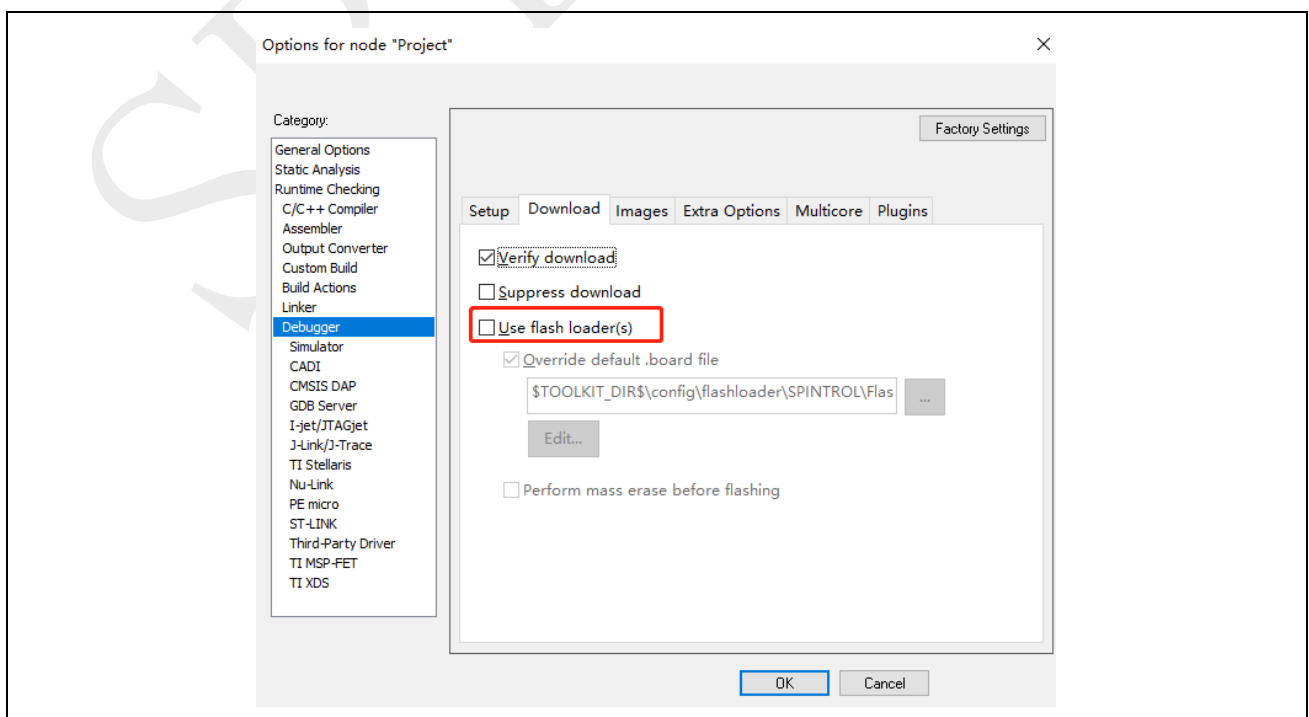
图 6-2: J-LINK



6.3 取消 Flash 配置

如图 6-3 所示，取消 “Debugger” “Download” 页中 Flash 配置。

图 6-3: Flash Download 配置



6.4 调试

点击 IAR IDE 中的 Debug 按钮, 就可以进行调试, 如果产生 Warning, 如图 6-4 所示, 则需要选择内核 Cortex-M4。

图 6-4: Debug warning

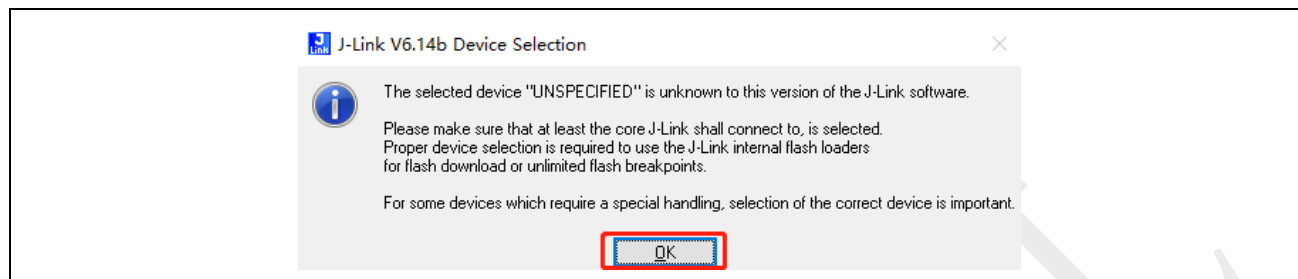


图 6-5: 选中 Cortex-M4

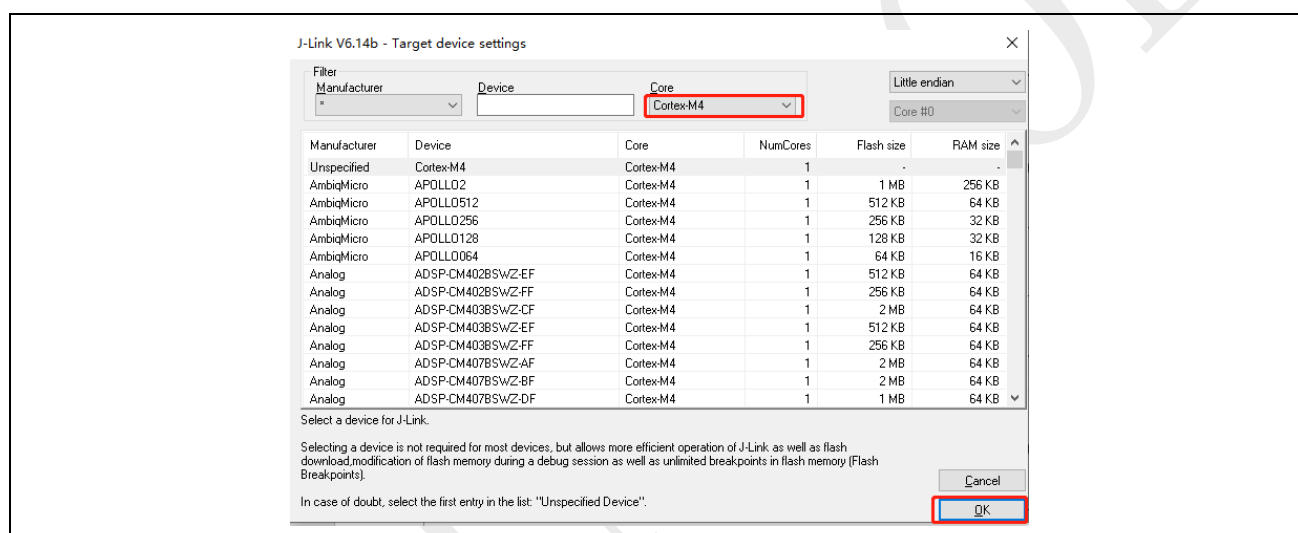
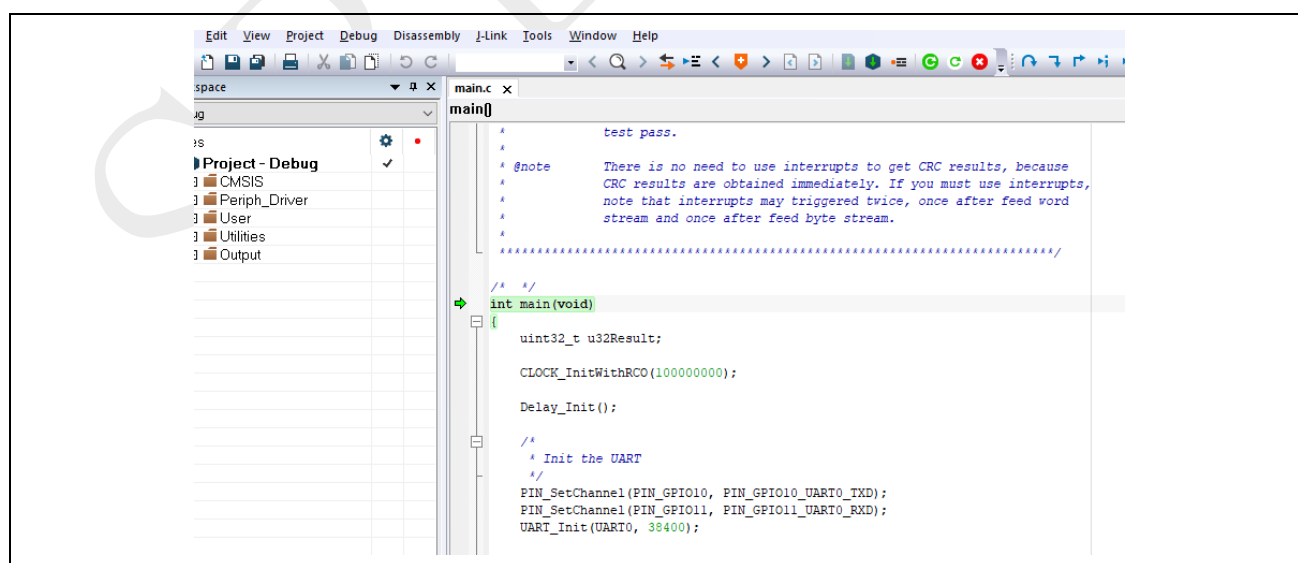


图 6-6: Debug 仿真



如无法进入 Debug 仿真, 需回到章节 2, 检查 J-Link 连线以及电平是否配置正确; 以及回到章节 6.1, 检查所有软件配置。