

《嵌入式微处理器》课程

作业 1

实验题目： keil 或 segger 调试运行 c 代码并观察 print 输出

姓 名： 马利豪

学 号： 2001210370

日 期： 2020.10.18

目录

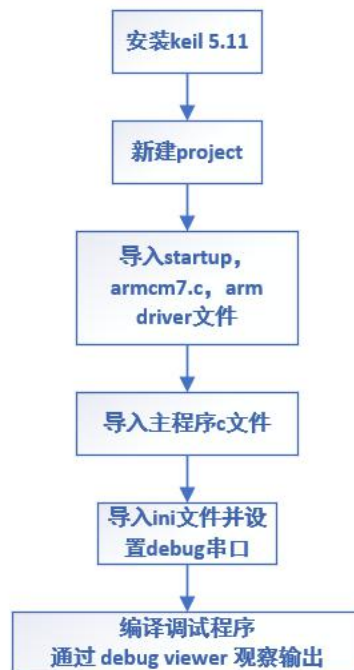
一、 实验过程与结果.....	3
1. 用 keil 调试 c 程序.....	3
(1) 实验流程.....	3
(2) 工程目录.....	3
(3) 编译.....	4
(4) debug 设置 ini 文件.....	4
(5) 设置串口（只打开 port0 即可）.....	4
(6) 打开 debug（print）viewer 窗口.....	5
(7) 单步调试，观察输出.....	5
2. 用 segger 模拟器调试 c 程序并观察输出.....	6
二、 实验中遇到的问题和解决方案.....	7
1.keil 中只有.c 文件无法编译成功。.....	7
2.keil 中无法打开 debug（print）viewer 窗口，只能观察变量。.....	7
3.Segger 新建工程时选择程序在 flash 中运行导致出现错误。.....	8
4.build 程序时，进度条一直不动。.....	9
三、 源代码.....	9

一、实验过程与结果

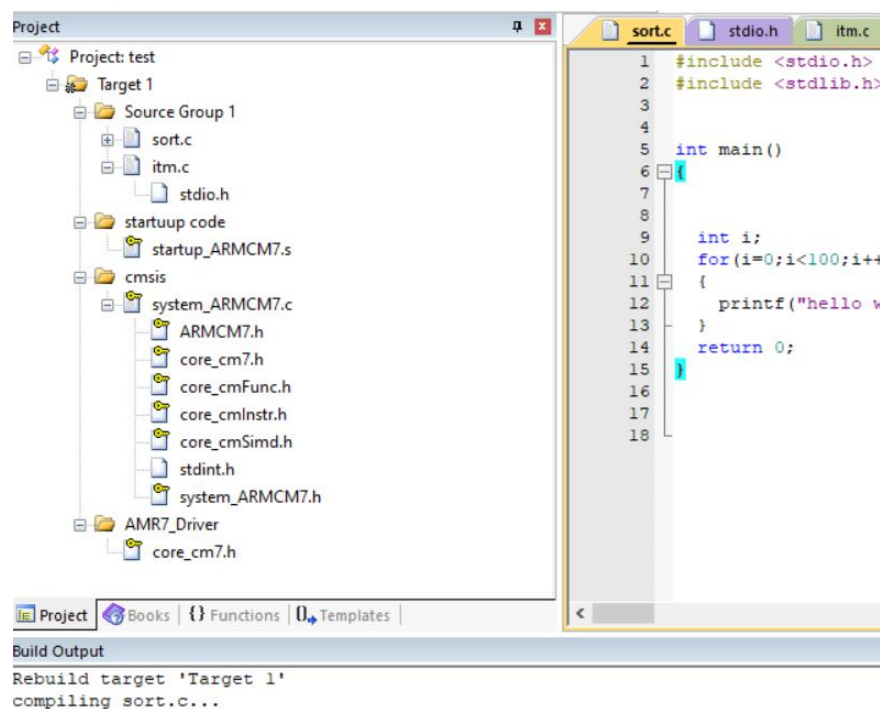
1. 用 keil 调试 c 程序

实验选用 ARM CM7 处理器，和 debug (print) viewer 窗口观察输出函数。

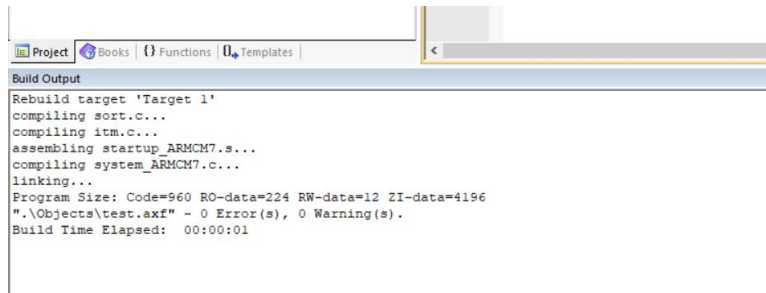
(1) 实验流程



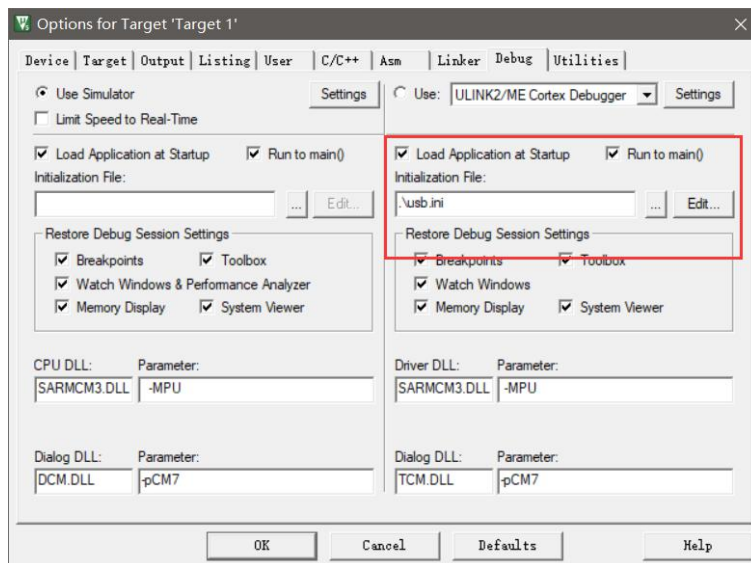
(2) 工程目录



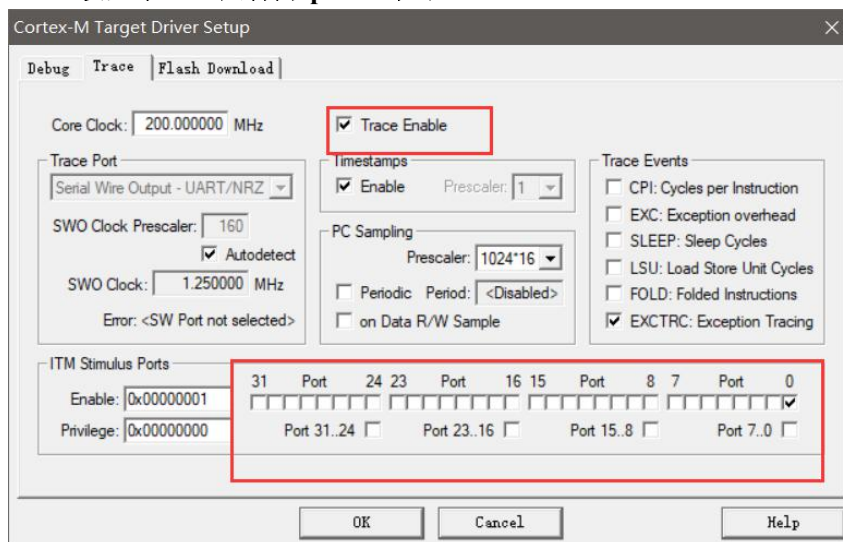
(3) 编译



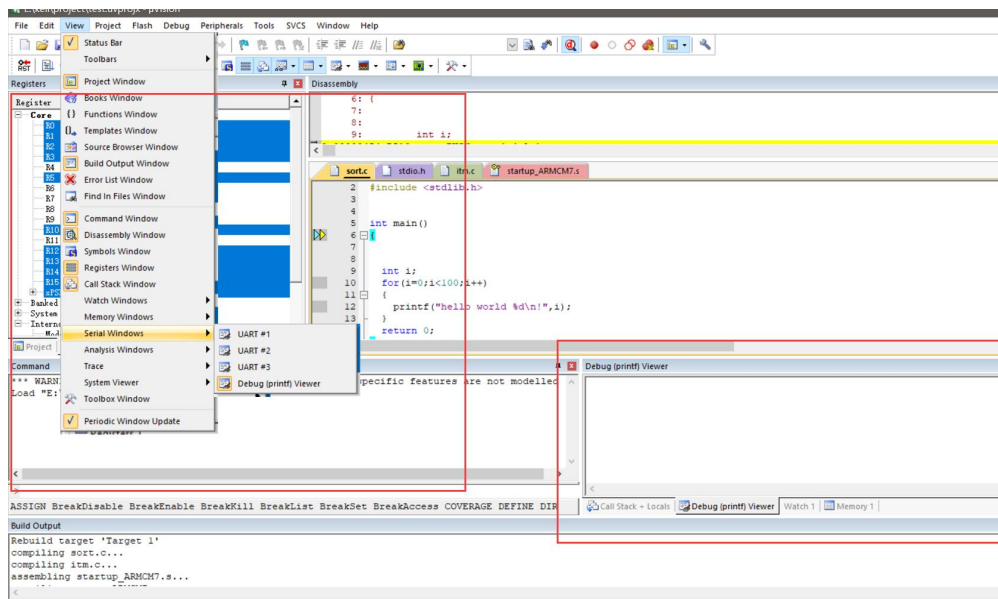
(4) debug 设置 ini 文件



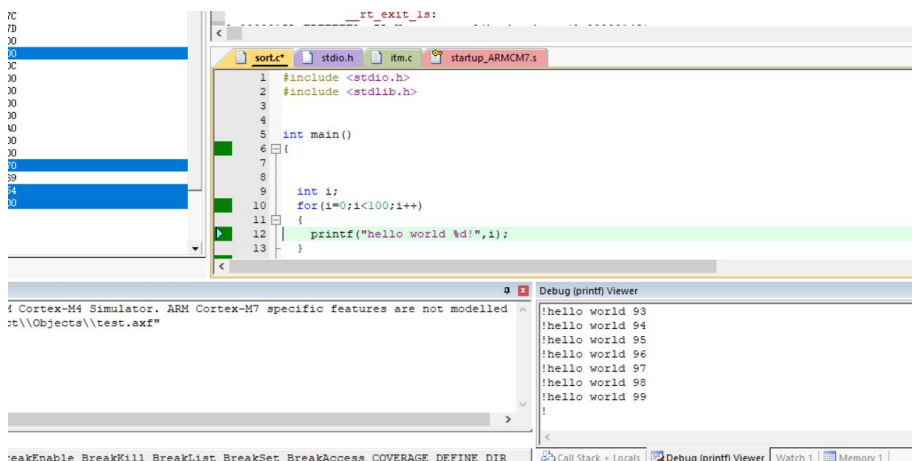
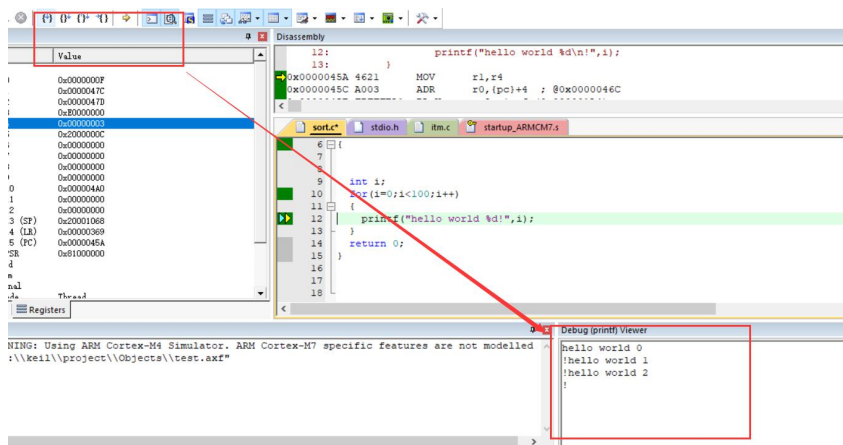
(5) 设置串口（只打开 port0 即可）



(6) 打开 debug (print) viewer 窗口



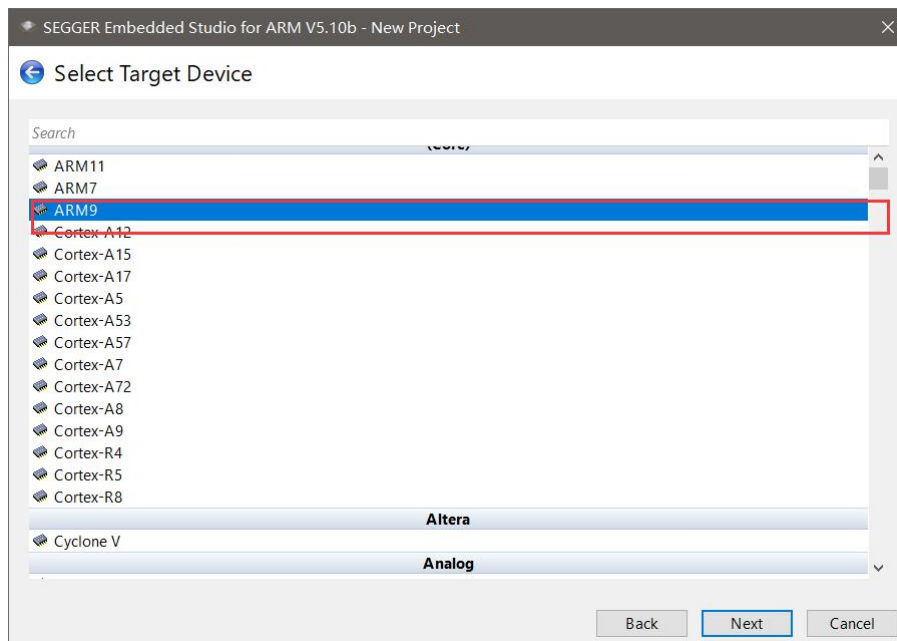
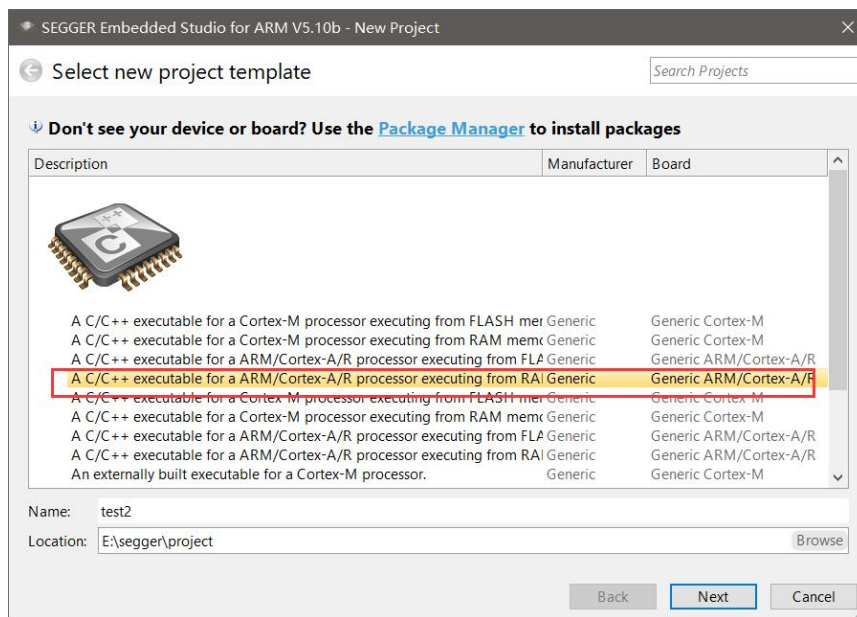
(7) 单步调试，观察输出

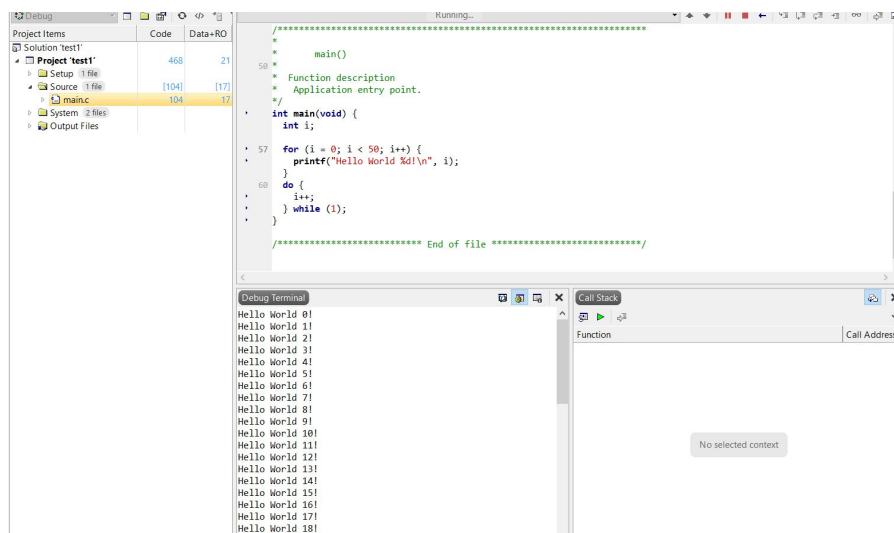


2. 用 segger 模拟器调试 c 程序并观察输出

实验选用 ARM9 处理器，和 output 终端窗口观察输出函数。

利用 segger 模拟器调试 c 程序过程比较简单，安装之后直接新建工程，在新建工程是选用程序在 RAM 中运行，同时选择 ARM9 处理器，建立完工程之后直接编译 debug 程序即可，通过 output 终端窗口观察输出。

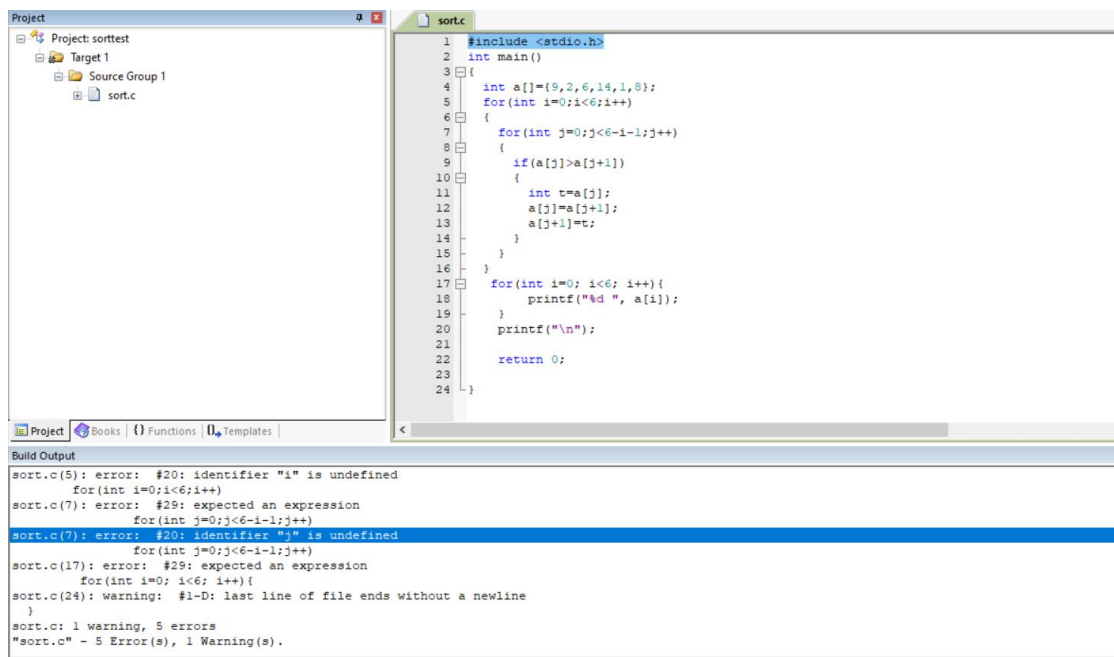




二、实验中遇到的问题 and 解决方案

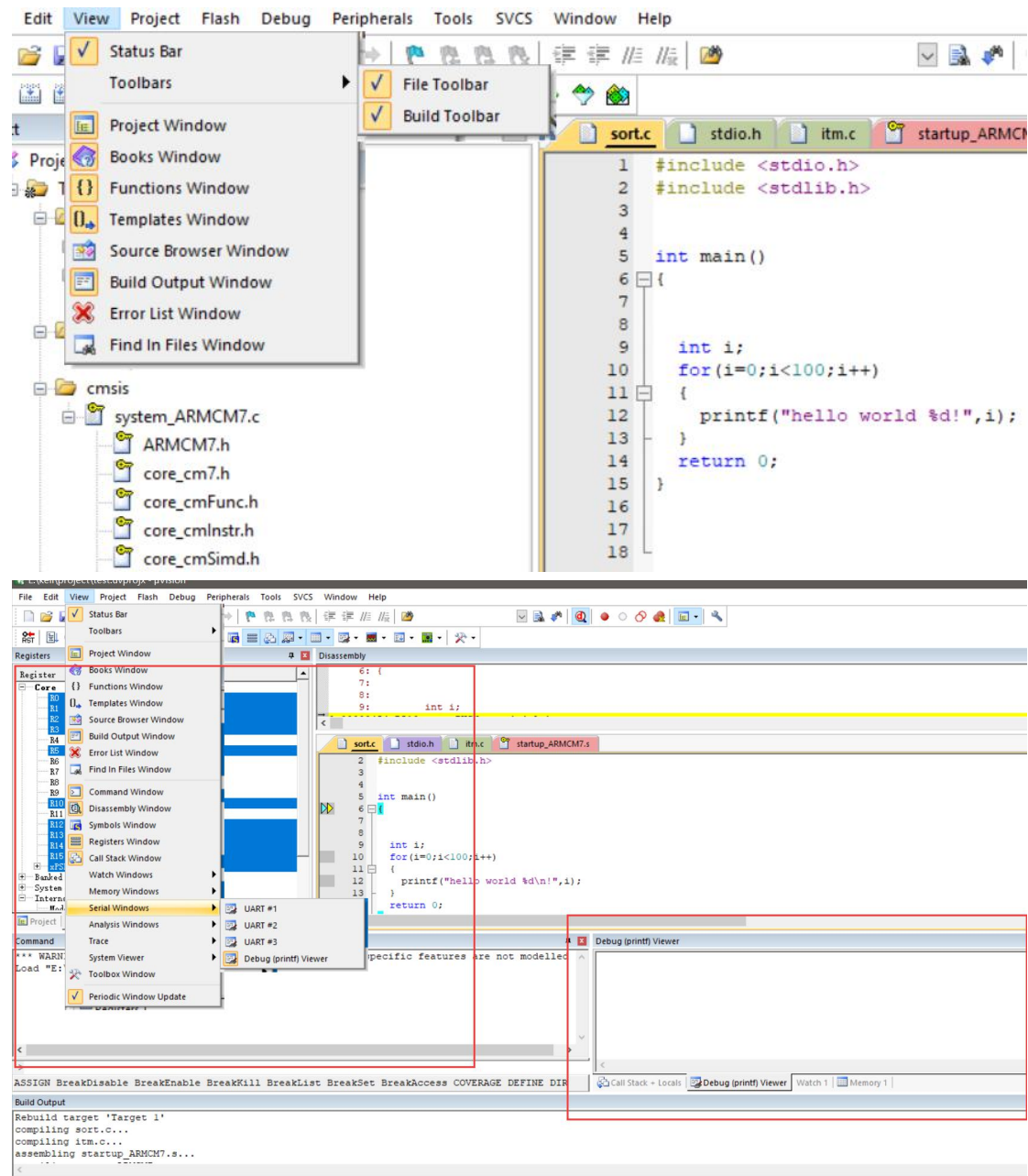
1. keil 中只有 .c 文件无法编译成功。

****解决方法：**发现并没有导入 startup 等一系列文件和驱动程序，根据网络论坛的教程，添加相关文件的路径并且设置环境变量之后即可解决问题。



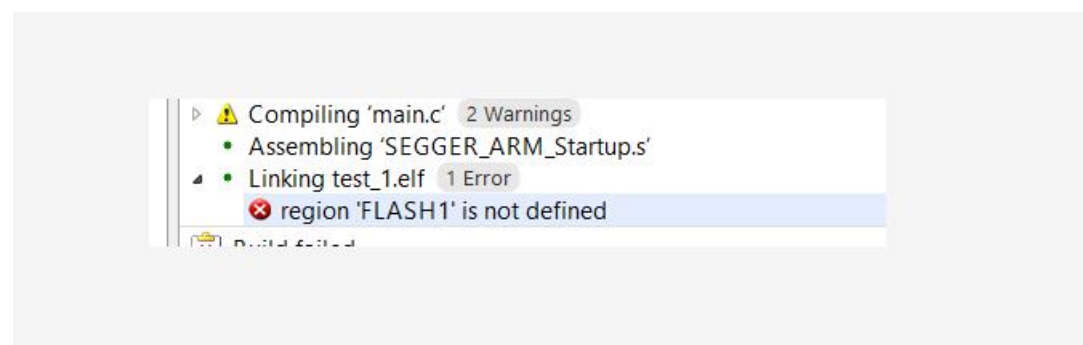
2. keil 中无法打开 debug (print) viewer 窗口，只能观察变量。

****解决方法：**使用 MDK 硬件调试_在 Debug printf Viewer 窗口显示打印信息_ITM，在工程中建立一个 .c 文件内容如下链接中显示，同时创建 .ini 文件导入 debug 设置中，设置好之后直接编译后，debug 打开窗口即可。



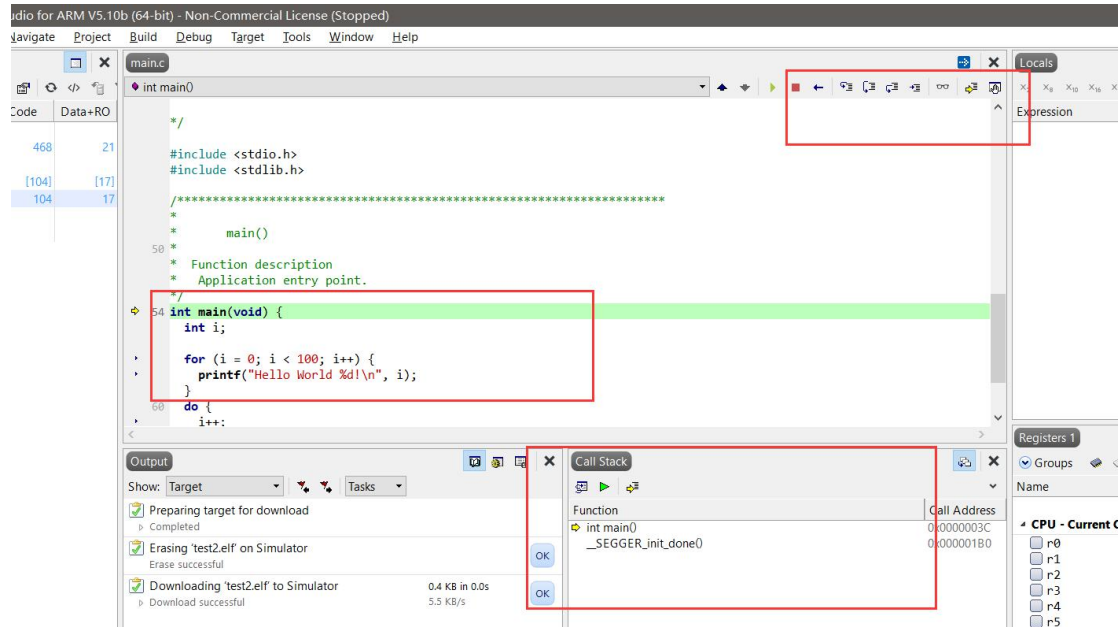
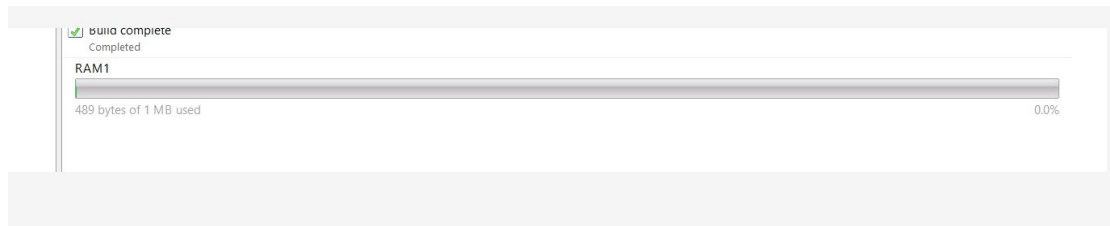
3.Segger 新建工程时选择程序在 flash 中运行导致出现错误。

**解决方法：重建程序在 RAM 中运行即可。



4.build 程序时，进度条一直不动。

**解决方法：直接选择 debug 中的 go 选项，单步调试程序。



三、源代码

//main.c

#include <stdio.h>

#include <stdlib.h>

int main(void) {

int i;

for (i = 0; i < 100; i++) {

printf("Hello World %d!\n", i);

}

do {

i++;

} while (1);

}

//itm.c

#include <stdio.h>

#define ITM_Port8(n) (*((volatile unsigned char *) (0xE0000000 + 4 * n)))

#define ITM_Port16(n) (*((volatile unsigned short *) (0xE0000000 + 4 * n)))

```

#define ITM_Port32(n)    (*((volatile unsigned long *)(0xE0000000+4*n)))
#define DEMCR            (*((volatile unsigned long *)(0xE000EDFC)))
#define TRCENA           0x01000000

struct __FILE { int handle; /* Add whatever you need here */ };
    FILE __stdout;
    FILE __stdin;

int fputc(int ch, FILE *f)
{
    if (DEMCR & TRCENA)
    {
        while (ITM_Port32(0) == 0);
        ITM_Port8(0) = ch;
    }
    return(ch);
}

//usb.ini
/*****
/
/* STM32DBG.INI: STM32 Debugger Initialization File */
/*****
/
// <<< Use Configuration Wizard in Context Menu >>> //
/*****
/
/* This file is part of the uVision/ARM development tools. */
/* Copyright (c) 2005-2007 Keil Software. All rights reserved. */
/* This software may only be used under the terms of a valid, current, */
/* end user licence from KEIL for a compatible version of KEIL software */
/* development tools. Nothing else gives you the right to use this software. */
/*****
/

FUNC void DebugSetup (void) {
// <h> Debug MCU Configuration
//  <01.0>   DBG_SLEEP      <i> Debug Sleep Mode
//  <01.1>   DBG_STOP       <i> Debug Stop Mode
//  <01.2>   DBG_STANDBY    <i> Debug Standby Mode
//  <01.5>   TRACE_IOEN     <i> Trace I/O Enable
//  <01.6..7> TRACE_MODE    <i> Trace Mode
//
//              <0=> Asynchronous
//              <1=> Synchronous: TRACEDATA Size 1

```

```

//          <2=> Synchronous: TRACEDATA Size 2
//          <3=> Synchronous: TRACEDATA Size 4
//  <o1.8>   DBG_IWDG_STOP <i> Independant Watchdog Stopped when Core is halted
//  <o1.9>   DBG_WWDG_STOP <i> Window Watchdog Stopped when Core is halted
//  <o1.10>  DBG_TIM1_STOP <i> Timer 1 Stopped when Core is halted
//  <o1.11>  DBG_TIM2_STOP <i> Timer 2 Stopped when Core is halted
//  <o1.12>  DBG_TIM3_STOP <i> Timer 3 Stopped when Core is halted
//  <o1.13>  DBG_TIM4_STOP <i> Timer 4 Stopped when Core is halted
//  <o1.14>  DBG_CAN_STOP  <i> CAN Stopped when Core is halted
// </h>
_WDWORD(0xE0042004, 0x00000027); // DBGMCU_CR
_WDWORD(0xE000ED08, 0x20000000); // Setup Vector Table Offset Register
}

DebugSetup(); // Debugger Setup

```