**MDK硬件调试\_在Debug printf Viewer窗口显示打印信息\_ITM**

2017年08月11日 17:09:48 [BurgessKzg](https://me.csdn.net/burgessKZG) 阅读数：4391 标签： [stm32](http://so.csdn.net/so/search/s.do?q=stm32&t=blog)[printf](http://so.csdn.net/so/search/s.do?q=printf&t=blog)[仿真](http://so.csdn.net/so/search/s.do?q=%E4%BB%BF%E7%9C%9F&t=blog)

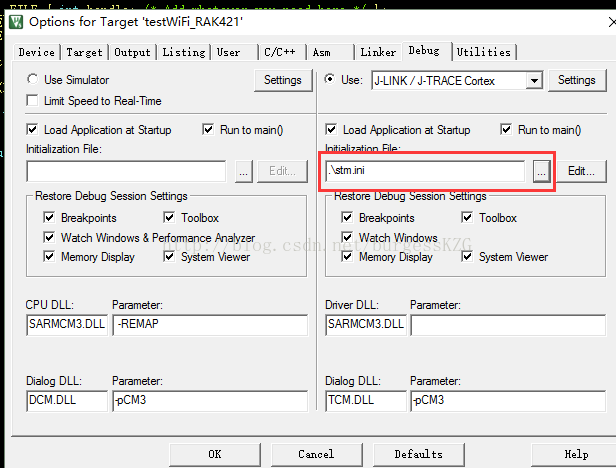
平时调试代码的时候大家都喜欢用printf函数来输出一些打印信息，来提示自己代码的执行情况。在刚开始接触这个调试方法的时候，我采用的方式就是浪费一个芯片的串口，然后利用TTL转USB模块，在PC端的串口调试助手中打印提示信息。另一种方式是硬件仿真的时候可以利用J-Link等工具实现的一种打印提示信息的方式。由于需要重定向fputc，所以两者最终只能选择其一，如果在资源不紧张的情况下，浪费一个串口其实也没啥，但是有些项目中会有很多通信接口需要使用这个时候就不可能浪费一个串口给我们做调试用了，这个时候通过MDK输出一些打印信息对于我们调试代码是很有帮助的。能实现MDK硬件调试打印调试信息的方式有两种：ITM机制和semihosting(半主机)机制。

这里总结一下ITM机制在MDK调试串口打印出信息的步骤：

1、建立一个.c文件，复制下列代码进去，并添加到工程目录中

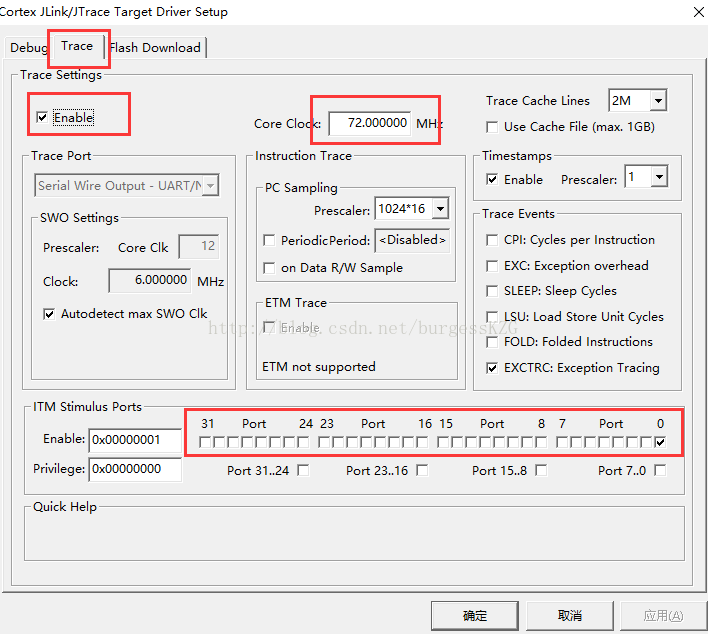
#include <stdio.h>  
  
#define ITM\_Port8(n)    (\*((volatile unsigned char \*)(0xE0000000+4\*n)))  
#define ITM\_Port16(n)   (\*((volatile unsigned short\*)(0xE0000000+4\*n)))  
#define ITM\_Port32(n)   (\*((volatile unsigned long \*)(0xE0000000+4\*n)))  
#define DEMCR           (\*((volatile unsigned 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);  
}  
  
2、建立一个.ini的文件，这个文件不要添加到工程目录中，复制以下内容进去  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
/\* 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  
//   <o1.0>    DBG\_SLEEP     <i> Debug Sleep Mode  
//   <o1.1>    DBG\_STOP      <i> Debug Stop Mode  
//   <o1.2>    DBG\_STANDBY   <i> Debug Standby Mode  
//   <o1.5>    TRACE\_IOEN    <i> Trace I/O Enable   
//   <o1.6..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  
  
3、ITM机制需要使用SWD模式调试，但同时必须连接SWO引脚，所以在仿真的时候最好用JTAG线与开发板连接；

4、点击魔术棒，在Debug下添加刚才的.ini文件（图1）；

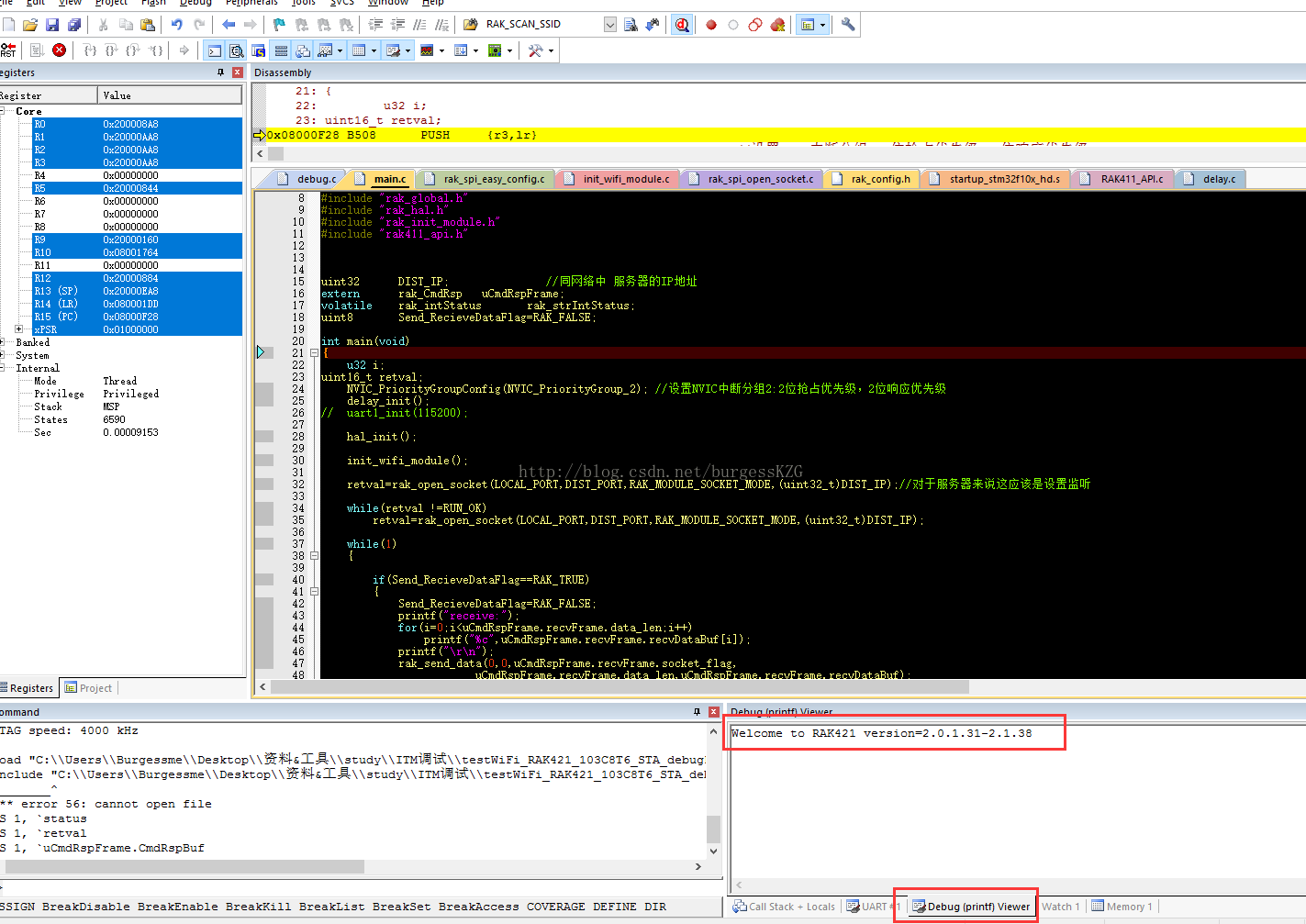


5、选择仿真器Port口为SW模式，在Trace下按照下图设置，使能Trace，由于我的芯片系统时钟是72M,所以设置

为72M,ITM Stimulus Ports口只保留0（图2）；



6、编译程序之后进入调试模式，调用Debug printf Viewer窗口，运行程序，这时候就可以发现有数据输出了（图3）；



备注：本方法是百度从豆瓣某页中找到的，总结一下备自己和各位朋友参考。