

32 位微控制器

SWO Trace 使用方法

应用笔记

Rev1.0 2023 年 11 月

适用对象

产品系列	产品型号	产品系列	产品型号	产品系列	产品型号
HC32F460	HC32F460JCTA	HC32F4A0	HC32F4A0PGHB	HC32F448	HC32F448FAUI
	HC32F460JETA		HC32F4A0PGTB		HC32F448FCUI
	HC32F460JEUA		HC32F4A0PIHB		HC32F448JATI
	HC32F460KCTA		HC32F4A0PITB		HC32F448JAUJ
	HC32F460KETA		HC32F4A0RGTB		HC32F448JCTI
	HC32F460KETA		HC32F4A0RITB		HC32F448JCUI
	HC32F460KEUA		HC32F4A0SGHB		HC32F448KATI
	HC32F460PCTB		HC32F4A0SGTB		HC32F448KCTI
	HC32F460PEHB		HC32F4A0SIHB		HC32F448MATI
	HC32F460PETB		HC32F4A0SITB		HC32F448MCTI
HC32F451	HC32F451FEUB	HC32F452	HC32F452FEUB	-	-
	HC32F451JEUB		HC32F452JEUB		
	HC32F451KETB		HC32F452KETB		
	HC32F451PETB		HC32F452PETB		

声 明

- ★ 小华半导体有限公司（以下简称：“XHSC”）保留随时更改、更正、增强、修改小华半导体产品和/或本档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- ★ 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- ★ XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- ★ XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- ★ 任何带有“®”或“™”标识的图形或字样是 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- ★ 本通知中的信息取代并替换先前版本中的信息。

©2023 小华半导体有限公司 保留所有权利

目 录

适用对象	2
声 明	3
目 录	4
1 概述	5
2 SWO Trace 功能介绍	6
2.1 调试端口	7
3 SWO Trace 在 IAR EWARM 上的应用	8
3.1 IAR EWARM 基本配置	8
3.2 使用 Terminal I/O 打印调试信息	10
3.3 实时显示变量值变化	11
3.4 测量代码执行时间	13
4 SWO Trace 在 KEIL 上的应用	14
4.1 KEIL 基本配置	14
4.2 使用 Debug (printf) Viewer 打印调试信息	17
5 参考样例及驱动	18
5.1 参考代码介绍	18
5.1.1 重定向 fputs 函数	18
5.1.2 使能并配置 TPIU 时钟	18
5.1.3 Trace 引脚分配	18
6 总结	19
版本修订记录	20

1 概述

本文档主要介绍小华半导体 HC32F460、HC32F4A0、HC32F451、HC32F452、HC32F448 系列芯片的 SWO Trace 功能的使用方法。

2 SWO Trace 功能介绍

SWO（Serial Wire Output）串行线输出，是一个单引脚跟踪接口，它作为 Cortex-M 内核中 CoreSight 架构的一部分，充当片上跟踪数据与跟踪端口分析器（TPA）之间的桥梁。是 TPIU 输出模式中的一种。

SWO trace 支持对正在运行的 MCU 内存的访问，而不需要暂停 CPU 的运行。SWO trace 利用串行线调试（SWD）接口中的 SWO 引脚输出调试时产生的跟踪信息。SWD 调试接口是 ARM 公司提出的一种调试接口，相对于传统的 JTAG 接口，使用更少的信号引脚。除标准的 SWD 连接外，SWO trace 只需要一个额外的引脚。

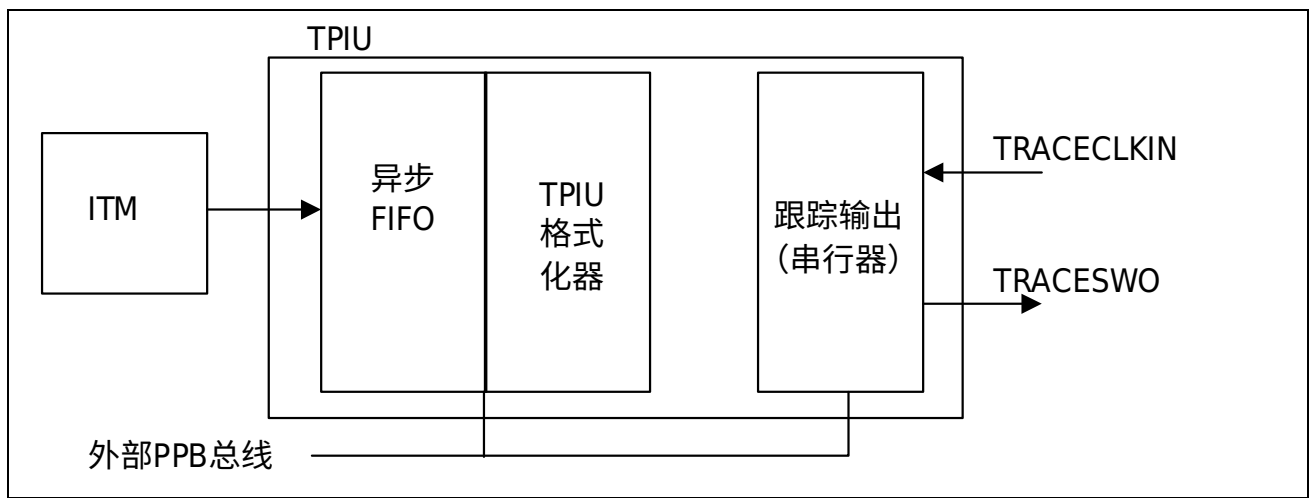


图 2-1 SWO 框图

SWO Trace 主要调试功能有：

- 以字符串的形式发送调试信息
- 监控中断进入/退出
- 监控函数进入/退出
- 变量值随时间的变化
- 事件通知

2.1 调试端口

调试端口集成了 SWD/JTAG 调试端口 (SWJ-DP)，在 SWJ-DP 中，SW-DP 的 2 个 JTAG 引脚与 JTAG-DP 的 5 个 JTAG 引脚中的部分引脚复用，JTDO 复用了 TRACESWO 与 TDO。因此 SWO Trace 功能只能在 SWD 调试端口下使用。

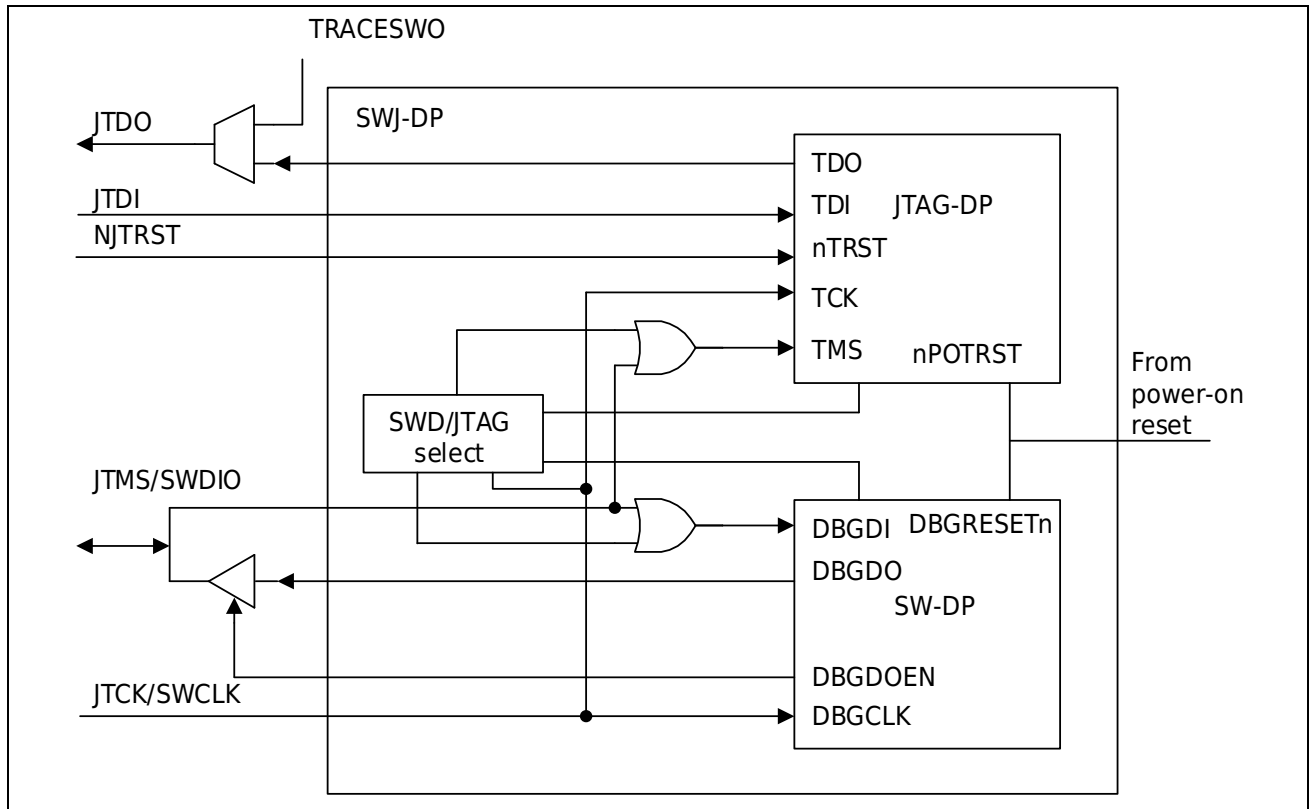


图 2-2 调试控制系统

SWO Trace 最基本和常用的应用是输出调试信息，本文将详细介绍在 IAR EWARM 和 KEIL 上使用 SWO Trace 该功能的相关配置。

3 SWO Trace 在 IAR EWARM 上的应用

使用 SWO Trace 功能，需要开启 TPIU 时钟和配置 Trace IO，参考章节【使能并配置 TPIU 时钟】和【Trace 引脚分配】的代码进行相关的配置。

3.1 IAR EWARM 基本配置

下面讲述 IAR EWARM 的配置，以 7.70.1 版本为基础做介绍。

- 在 Project -> Options -> General Options -> Library Configuration 中，在 Library 中选择 Full，在 stdout/stderr 中选择 Via SWO，如图 3-1 所示。

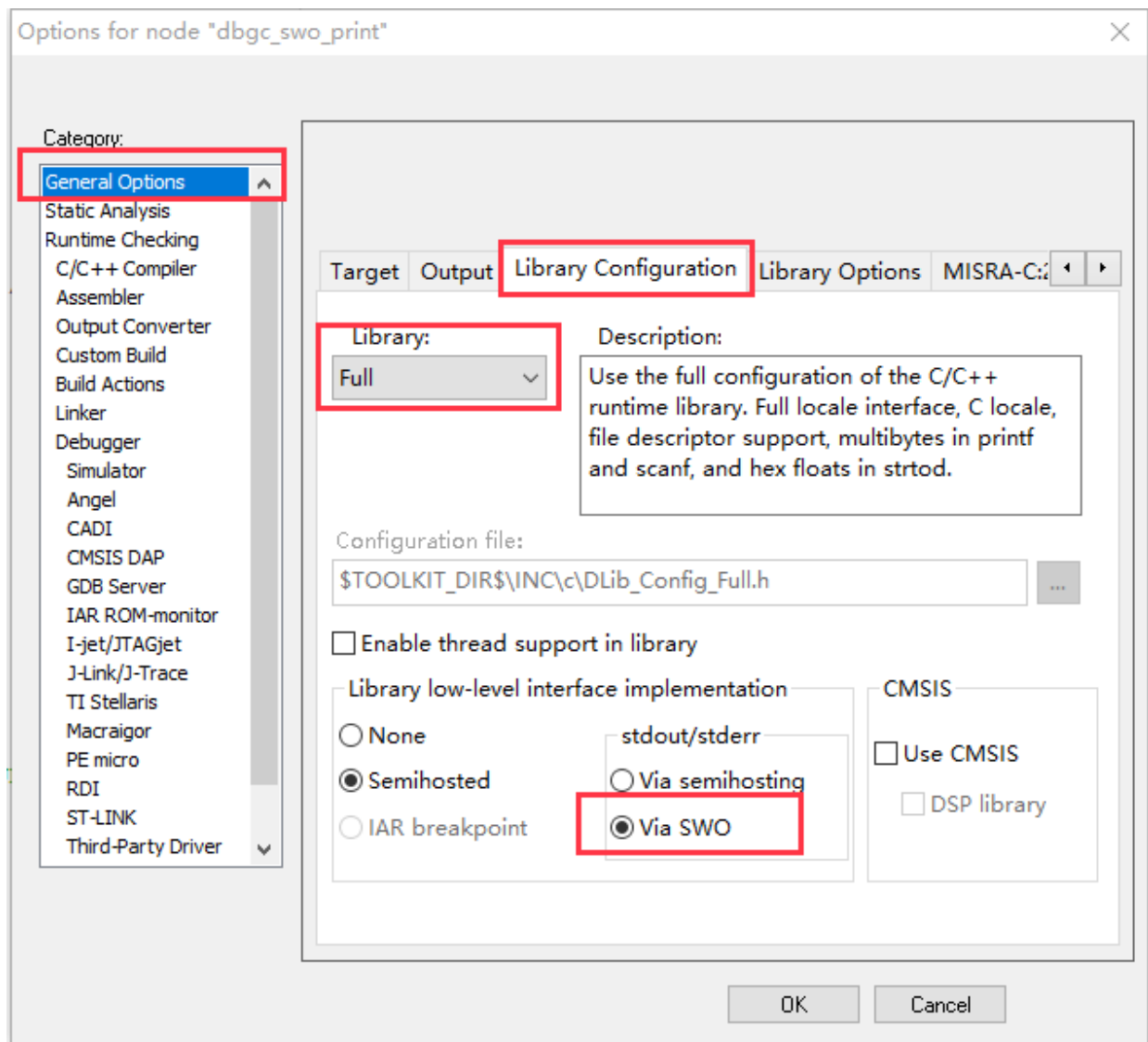


图 3-1 General Options 配置

- 在 Project -> Options -> Debugger -> Setup 选项中，在 Driver 中选择 J-Link/J-Trace，如图 3-2 所示。或其他支持 SWO 的调试接口，本文中以 J-Link/J-Trace 为例。选则其他调试接口时，确保连接方式是 SWD 模式。

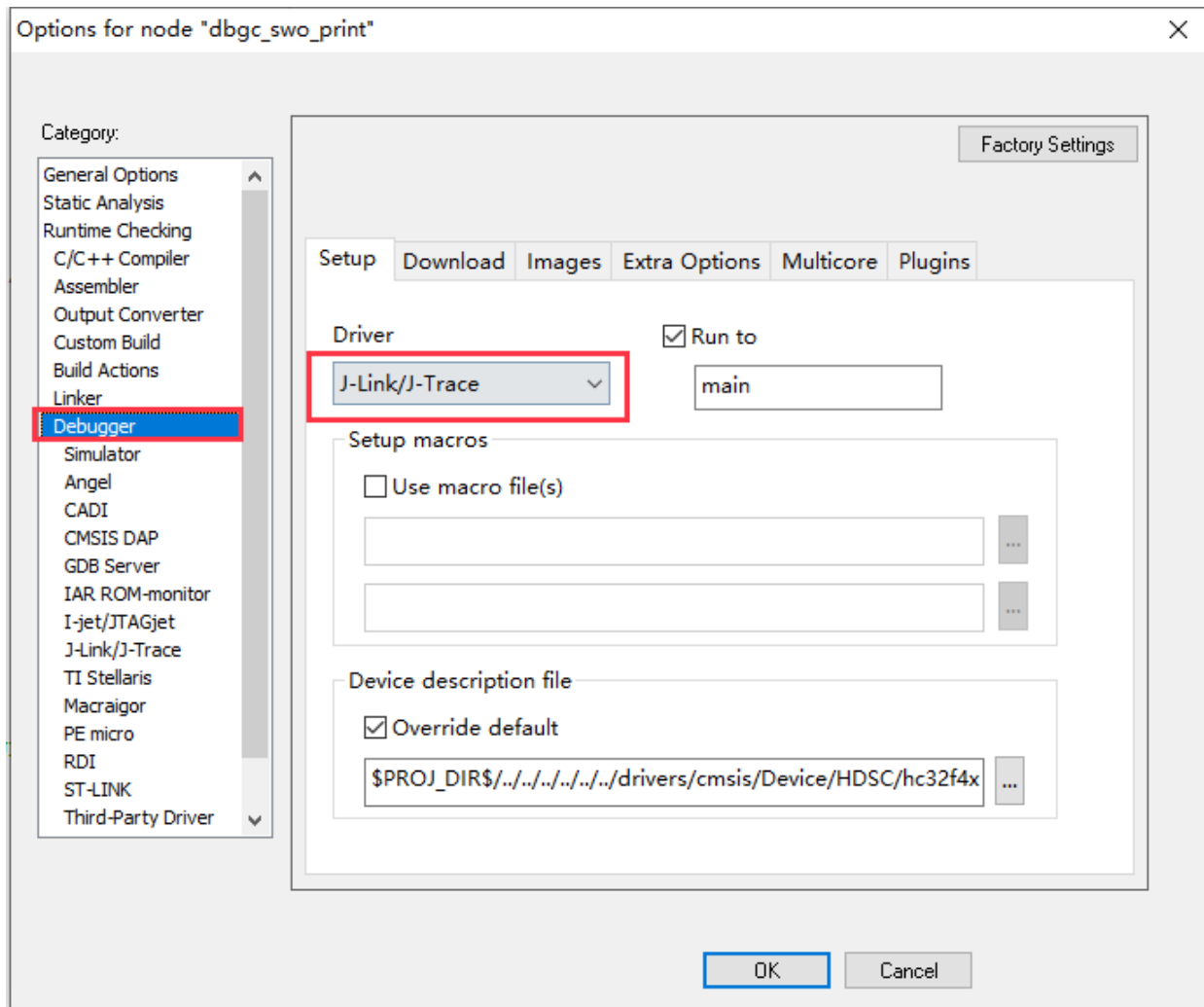


图 3-2 Debugger 配置

- 进入调试环境，从所使用的仿真器如 J-Link 调试菜单下，打开 SWO Trace Window Settings，根据需求选择跟踪方式，本文中选择 CPI，也可选择 PC Sample、Time Stamps 等，如图 3-3 所示。

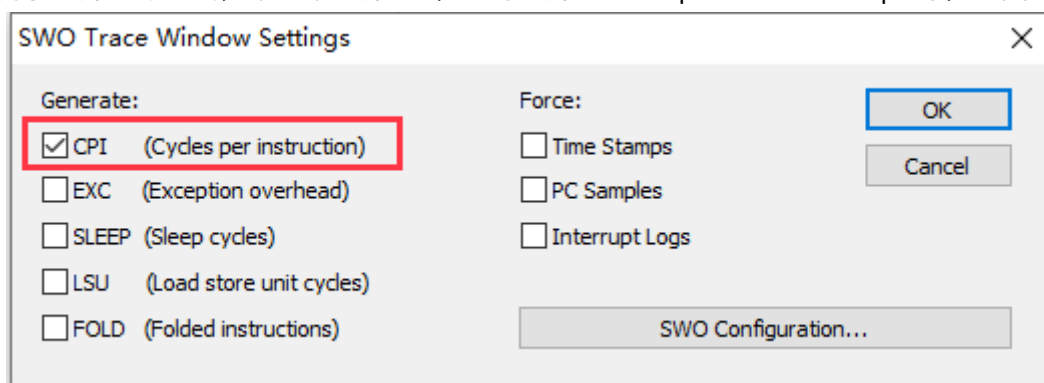


图 3-3 SWO Trace Window Settings

- 进入调试环境，从所使用的仿真器如 J-Link 调试菜单下，打开 SWO Configuration，配置时钟和 ITM Stimulus Ports。其中 CPU clock 为 TPIU 时钟频率，即 CPU 主频通过 CLK_SetTpiuClockDiv()分频后的时钟，参考样例中主频为 200 MHz，设置了 4 分频，因此 CPU clock 为 50 MHz。ITM 端口，

默认使用 port0，也可更改软件实现 0~31 任意端口输出，这里不做赘述。整体配置如图 3-4 所示。

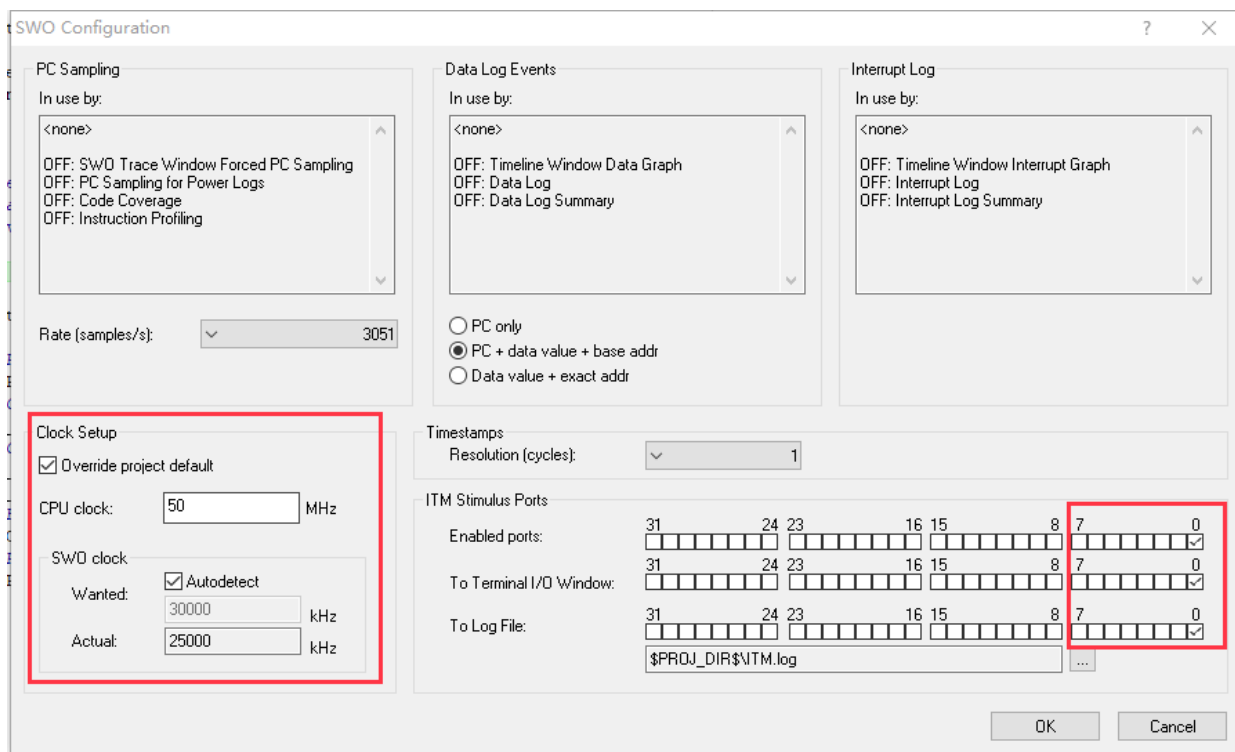


图 3-4 SWO Configuration

3.2 使用 Terminal I/O 打印调试信息

使用 terminal I/O 打印调试信息，需要重定向 fputs 函数，参考【重定向 fputs 函数】章节代码。

进入调试环境，通过 View -> Terminal I/O 打开 Terminal I/O 窗口，通过 printf 打印的调试信息会在该窗口中输出。

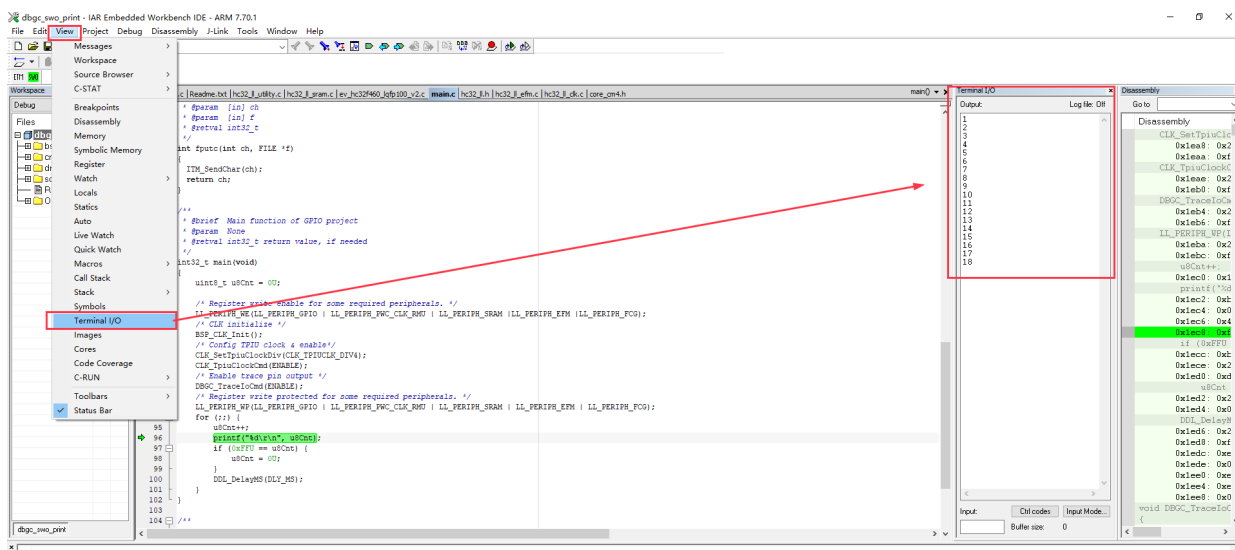


图 3-5 Terminal I/O 窗口

3.3 实时显示变量值变化

当变量为全局变量时，可通过 SWO 实时监测变量数值变化，并以图形化显示。

- 图 3-6 中 u8Cnt 为全局变量，在该变量处，右键单击，选择“Set Data Log Breakpoint for 'u8Cnt'”。

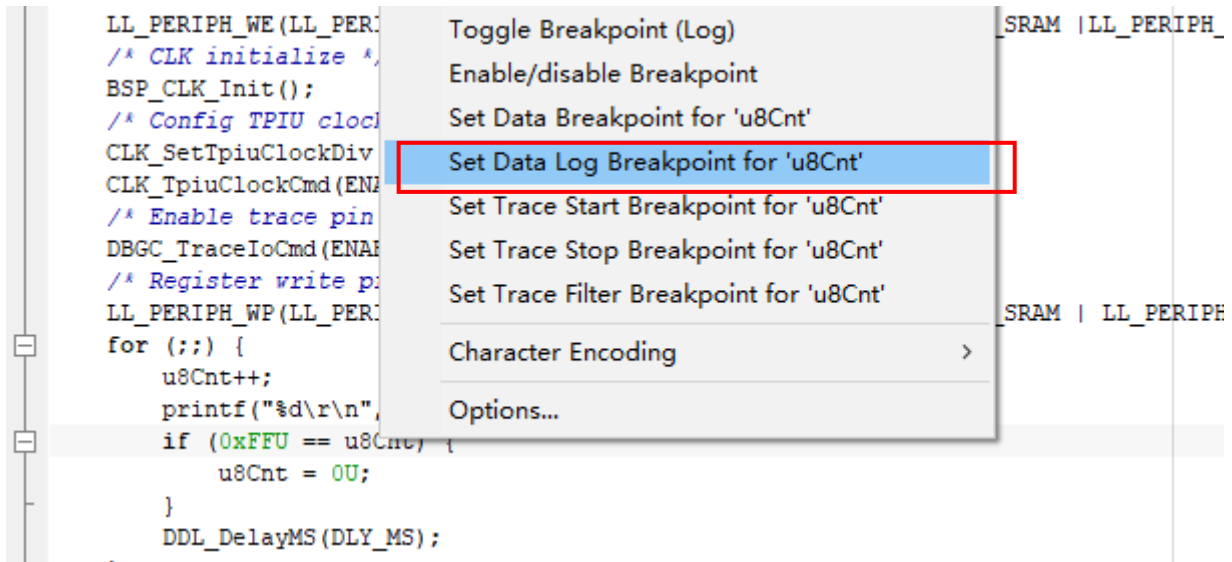


图 3-6 Data Log 设置

- 通过 View -> Breakpoints 打开 Breakpoint 窗口，在断点处右键单击 Edit 选项，进行编辑，选择该变量被 Write 改写时触发数据记录断点。如图 3-7 和图 3-8 所示。

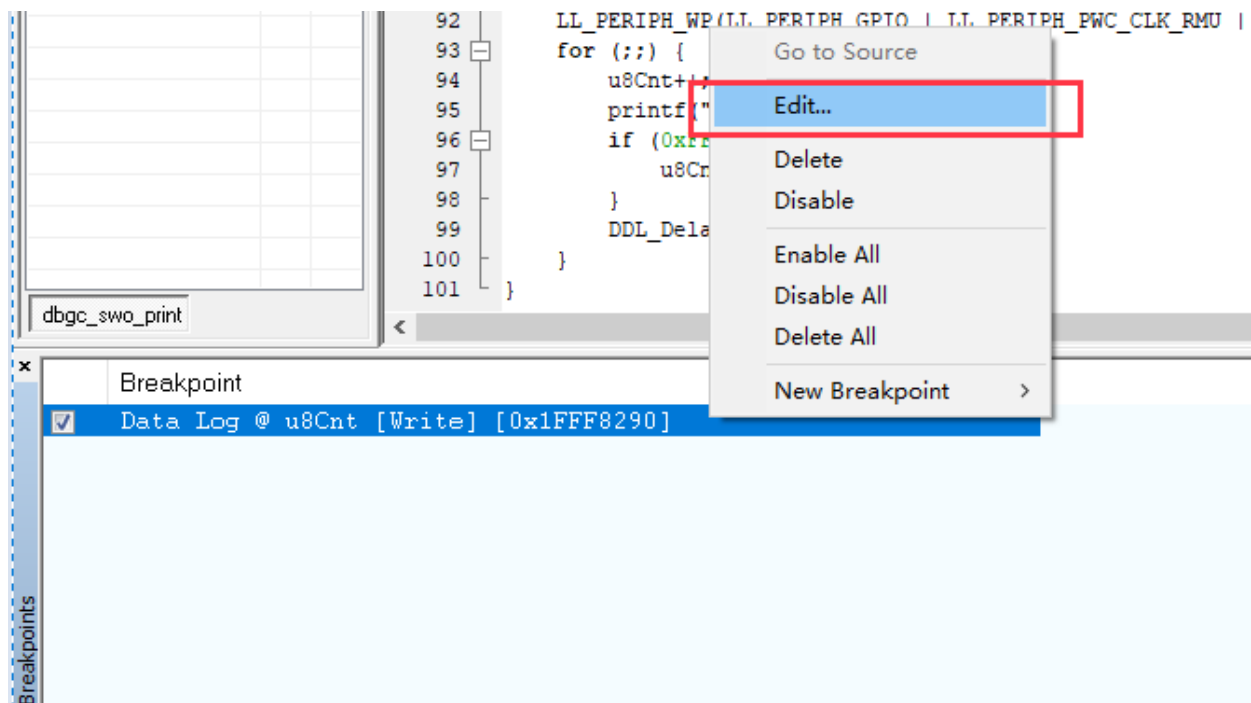


图 3-7 breakpoint 窗口

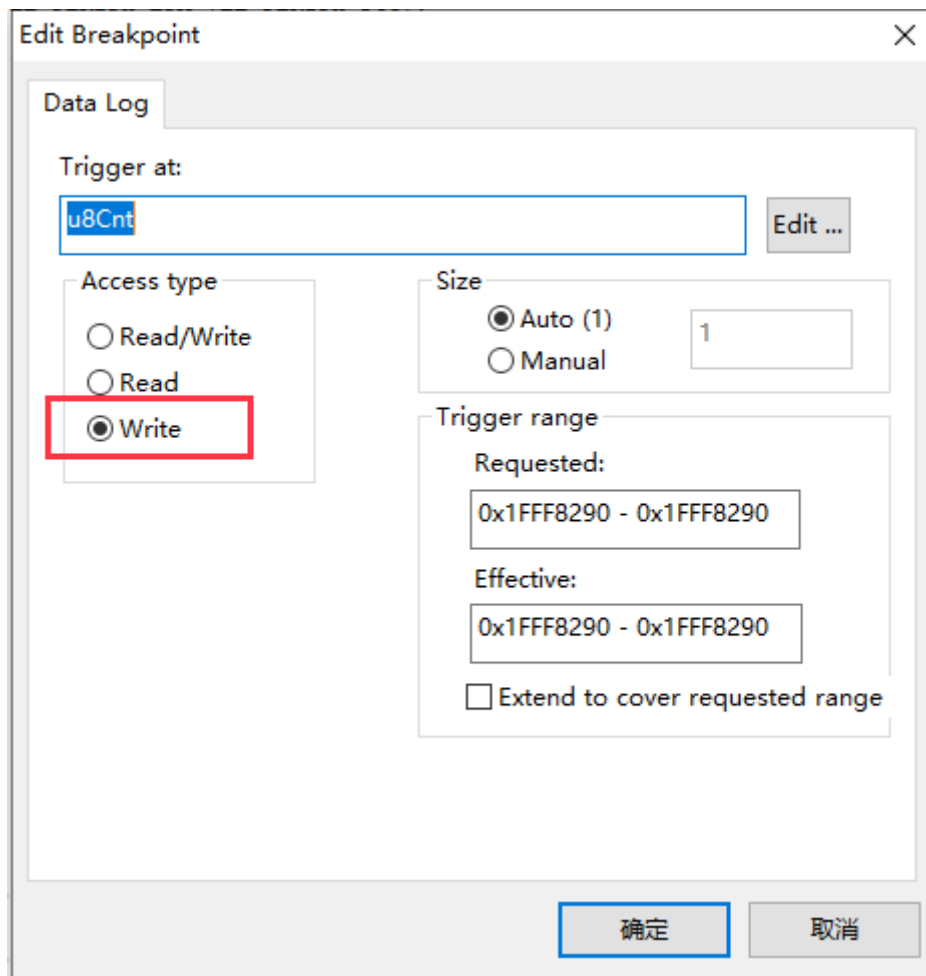


图 3-8 Edit Breakpoint

- 从所使用的仿真器如 J-Link 调试菜单下，打开 Data Log 和 Timeline 窗口，在 Data Log 窗口处右键选择“Enable”后，全速运行代码，在 Data Log 窗口中将会实时显示全局变量“u8Cnt”的跟踪记录，包括时间戳、变量值，变量地址等信息，如图 3-9 所示。

	Time	Program Counter	u8Cnt	Address
	600108.24 us	0x00001EC6	W 0x02	@ 0x1FFF8290+?
1s	200192.46 us	0x00001EC6	W 0x03	@ 0x1FFF8290+?
1s	800276.68 us	0x00001EC6	W 0x04	@ 0x1FFF8290+?
2s	400360.90 us	0x00001EC6	W 0x05	@ 0x1FFF8290+?
3s	445.12 us	0x00001EC6	W 0x06	@ 0x1FFF8290+?
3s	600529.34 us	0x00001EC6	W 0x07	@ 0x1FFF8290+?
4s	200613.56 us	0x00001EC6	W 0x08	@ 0x1FFF8290+?
4s	800697.82 us	0x00001EC6	W 0x09	@ 0x1FFF8290+?
5s	400782.08 us	0x00001EC6	W 0x0A	@ 0x1FFF8290+?
6s	870.24 us	0x00001EC6	W 0x0B	@ 0x1FFF8290+?

图 3-9 Data Log 窗口

- 在 Timeline 窗口 Data Log 一栏，再次点击右键选择“Enable”使能图像化显示，可获得变量值的实时变化图像，如图 3-10 所示。

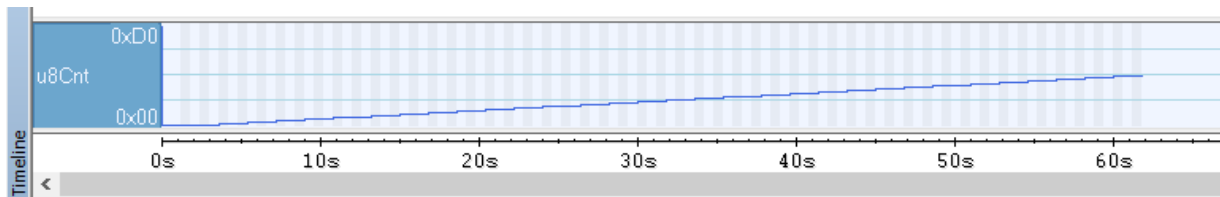


图 3-10 Timeline 窗口

3.4 测量代码执行时间

因为 Data Log 窗口中显示的信息包含了变量值变化时的时间戳，所以通过 Data Log 还可以测量代码执行的时间，在被测代码前后添加一个全局变量作为测量运行时间的监控变量。图 3-11 中通过变量 u8TimeStamp 测量函数 BSP_XTAL32_Init() 的执行时间。

```
u8TimeStamp = 1U;  
BSP_XTAL32_Init();  
u8TimeStamp = 2U;
```

图 3-11 测量代码运行时间

从图 3-12 中可以看出，u8TimeStamp 变化前后的时间为 $t = 9944.6 \mu s - 609.44 \mu s = 9335.16 \mu s$ 。由此可知 BSP_XTAL32_Init() 函数的执行时间为 $9335.16 \mu s$ 。

×	Time	Program Counter	u8TimeStamp	Address
	609.44 us	0x000023B4	W 0x01	@ 0x1FFF8291+?
	9944.60 us	0x000023BE	W 0x02	@ 0x1FFF8291+?

图 3-12 u8TimeStamp 的 Data Log 窗口

本文中对 SWO Trace 在 IAR EWARM 上的应用就介绍到这里。

4 SWO Trace 在 KEIL 上的应用

使用 SWO Trace 功能，需要开启 TPIU 时钟和配置 Trace IO，参考章节【使能并配置 TPIU 时钟】和【Trace 引脚分配】的代码进行相关的配置。

4.1 KEIL 基本配置

下面讲述 KEIL 的配置，以 5.36.0 版本为基础做介绍。

- Options -> Debug 选项中选择 J-Link/J-Trace，然后点击“Settings”，如图 4-1 所示，或其他支持 SWO 的调试接口，本文中以 J-Link/J-Trace 为例。

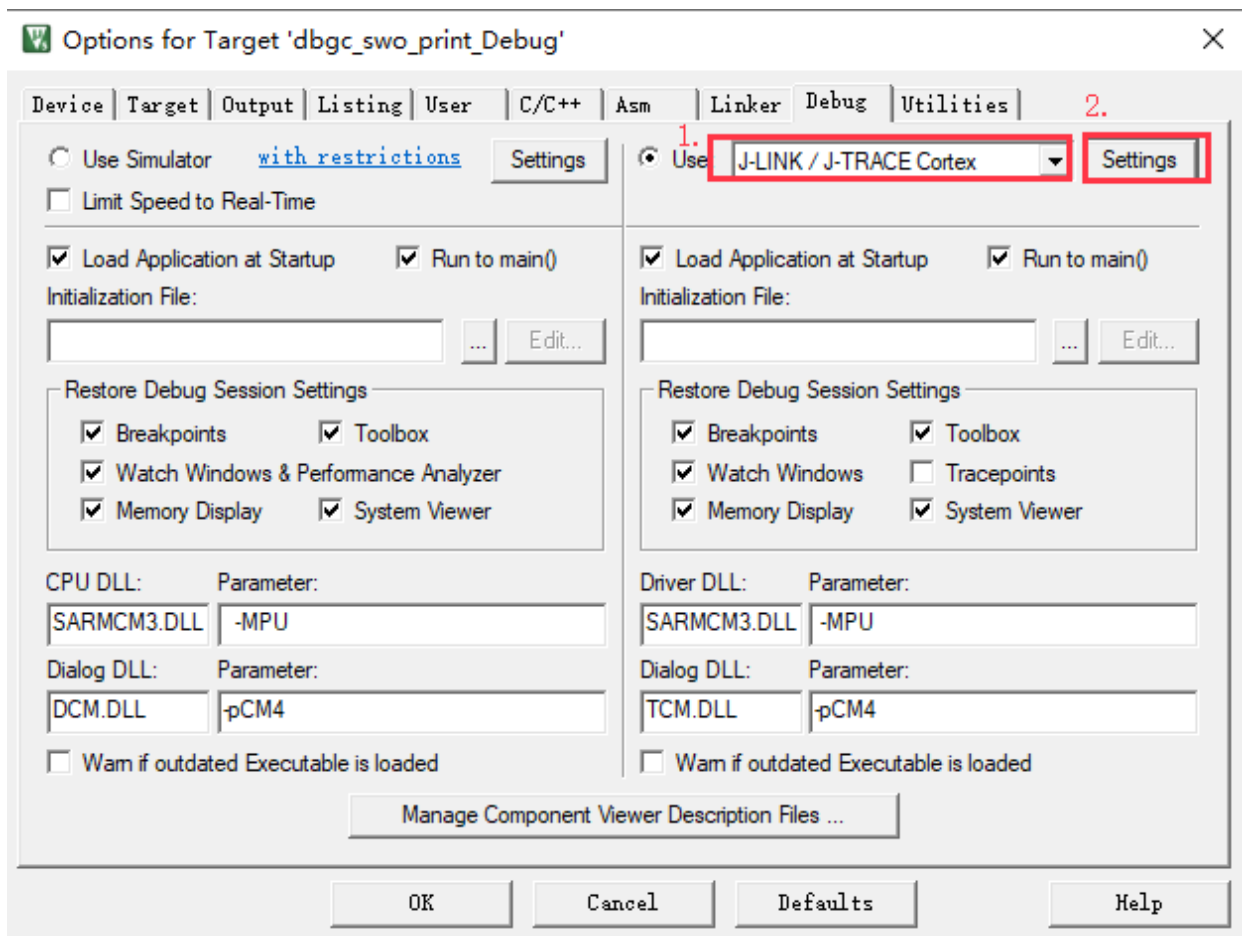


图 4-1 Options 窗口

- 点击 Settings 后，出现 “Cortex JLink/JTrace Target Driver Setup” 窗口，Debug 界面中 Port 选项，选择 SW 接口，如图 4-2 所示。

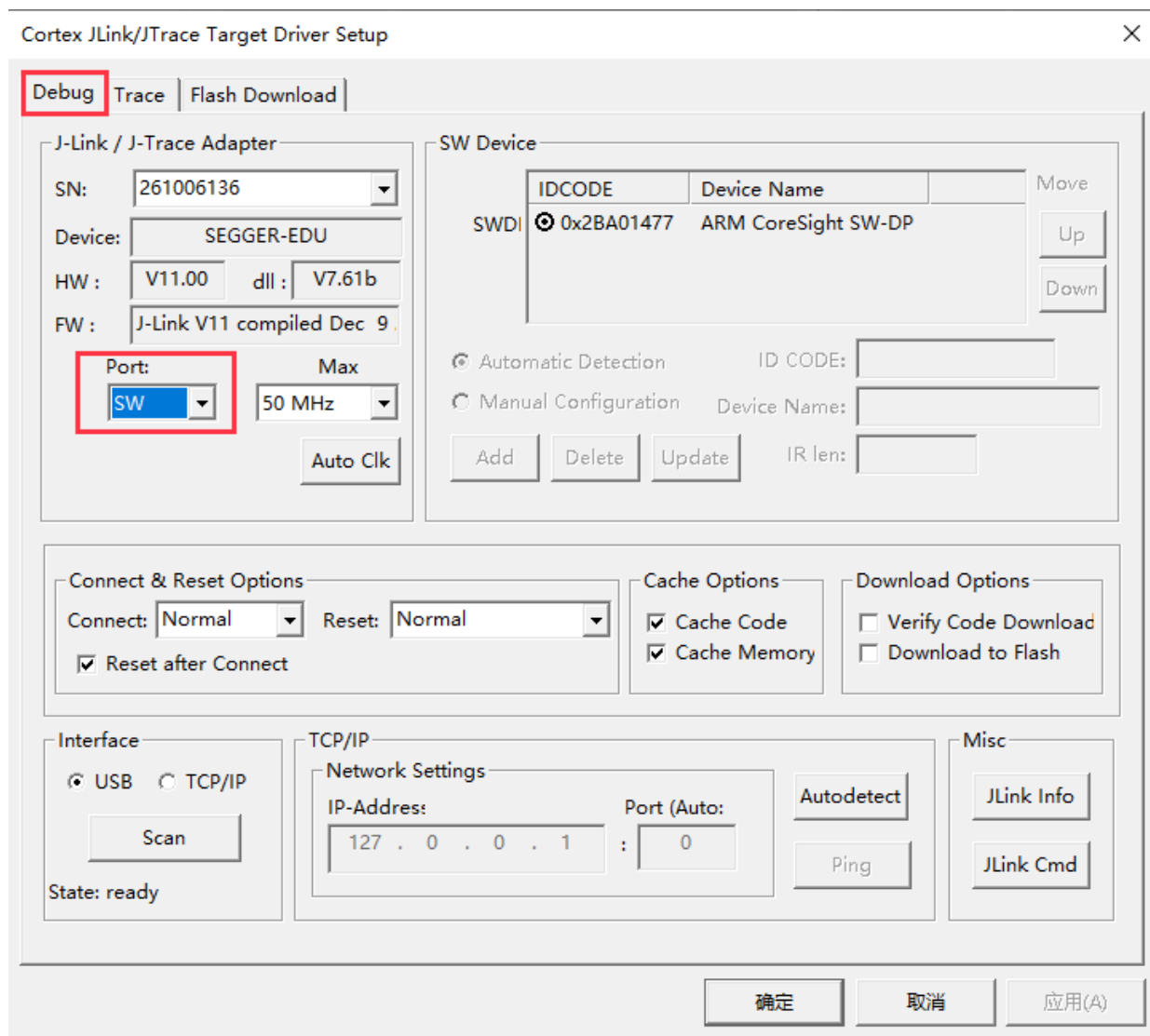


图 4-2 调试接口配置

- 在“Cortex JLink/JTrace Target Driver Setup”窗口的 Trace 页面，配置 Trace 相关信息，如图 4-3 所示。其中 Core Clock 为 TPIU 时钟频率，即 CPU 主频通过 CLK_SetTpiuClockDiv()分频后的时钟，参考样例中主频为 200 MHz，设置了 4 分频，因此 Core Clock 为 50 MHz。Trace Port 需选择 SWO 输出 (Serial Wire Output - UART/NRZ)；SWO Settings 框中，SWO 支持的最大时钟为 30MHz，因此，此处设置 2 分频；Trace Events 可根据需求自行配置。ITM Stimulus Ports 默认使用 port0，也可更改软件实现 0~31 任意端口输出，这里不做赘述。

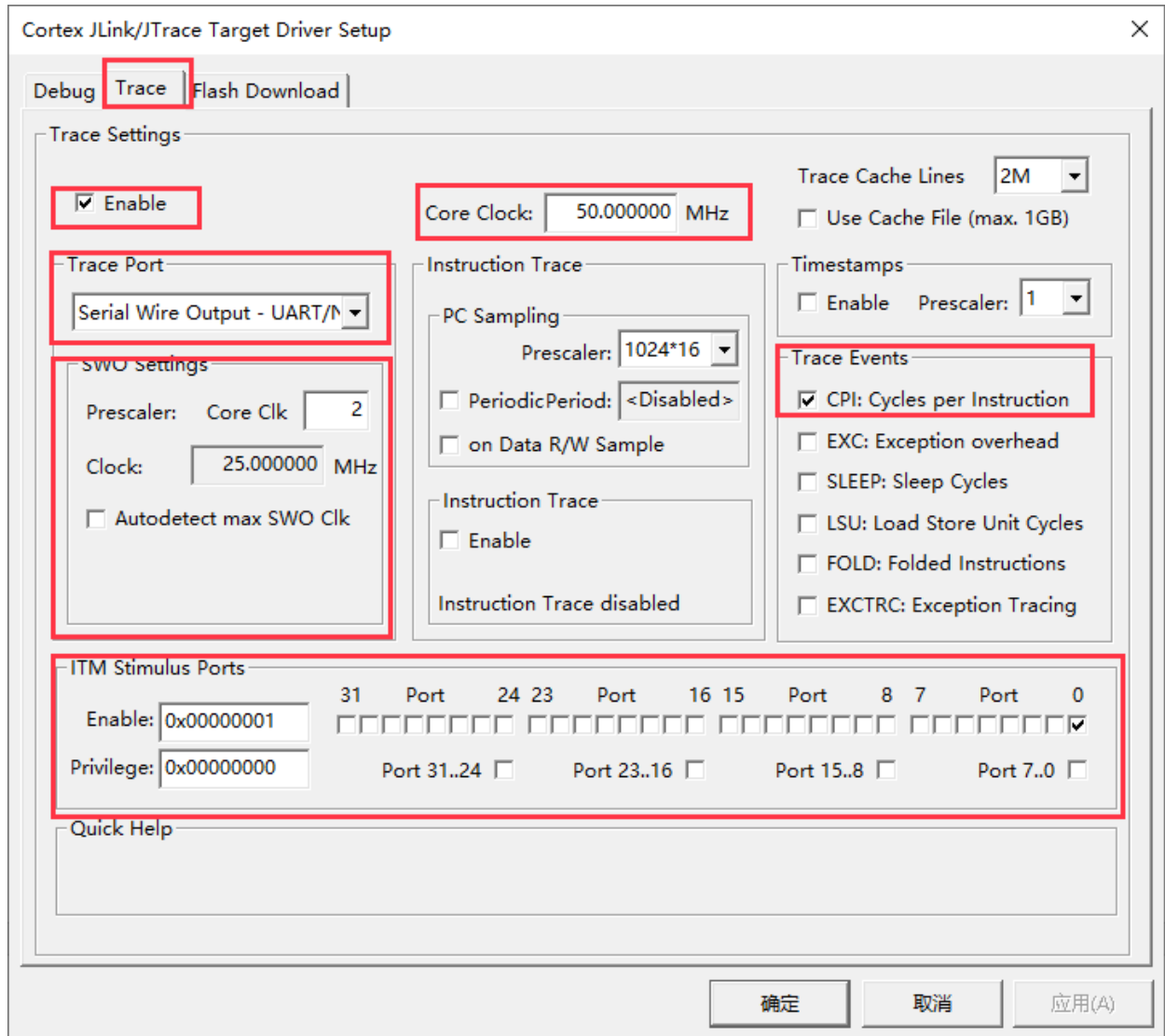


图 4-3 Trace 配置

4.2 使用 Debug (printf) Viewer 打印调试信息

使用 Debug (printf) Viewer 打印调试信息，需要重定向 fputc 函数，参考【重定向 fputc 函数】章节代码。

进入调试环境，通过 View -> Serial Windows -> Debug (printf) Viewer 打开 Debug (printf) Viewer 窗口，通过 printf 打印的调试信息会在该窗口中输出。如图 4-4 所示。

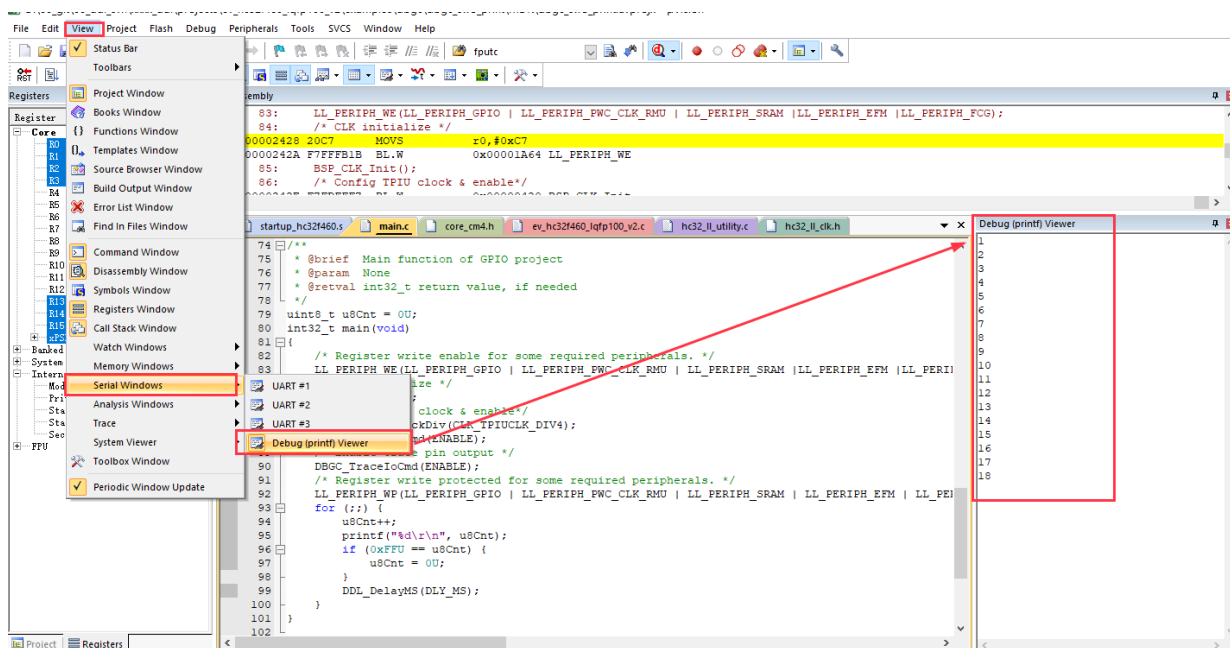


图 4-4 Debug (printf) Viewer 窗口

5 参考样例及驱动

5.1 参考代码介绍

5.1.1 重定向 fputc 函数

使用 SWO Trace 实现串口输出的功能，头文件中需要包含“stdio.h”，且重定向 fputc 函数，通过 ITM 函数实现输出功能。

```
#include "stdio.h"

int fputc(int ch, FILE *f)
{
    ITM_SendChar(ch);
    return ch;
}
```

5.1.2 使能并配置 TPIU 时钟

参考代码设置的是 4 分频，可根据实际使用情况进行配置。

```
/* Config TPIU clock & enable*/
CLK_SetTpiuClockDiv(CLK_TPIUCLK_DIV4);
CLK_TpiuClockCmd(ENABLE);
```

5.1.3 Trace 引脚分配

要分配 TRACE 引脚，调试主机必须对 MCU 调试配置寄存器（MCUTRACECTL）的位 TRACE_IOEN 和 TRACE_MODE[1:0]进行编程。默认情况下不分配 TRACE 引脚。TRACE_IOEN 位控制 SWO 引脚的输出，TRACE_MODE[1:0]位控制 traceD0~traceD3 的输出，本文中描述的使用，仅需使能 SWO 引脚的输出。

```
/* Enable trace pin output */
DBGCMC_TraceIoCmd(ENABLE);
```

6 总结

本文介绍了小华半导体 HC32F460、HC32F4A0、HC32F451、HC32F452、HC32F448 系列芯片的 SWO Trace 功能，及其在 IAR EWARM 和 KEIL 上的使用。给客户实际使用提供了参考。

版本修订记录

版本号	修订日期	修订内容
Rev1.0	2023/11/08	初版发布。