AN0015
应用笔记

AT32 Printf Debug Demo

前言

应用代码调试过程中，经常会需要确认调试过程信息，常规情况下可使用串口助手进行输出查看，但当硬件环境不支持串口助手打印信息时，测试过程信息的观察就成为难点。

本应用笔记介绍了在AT32微控制器上的各种调试过程信息输出方法，可满足不具备串口助手条件下的调试过程信息输出。

注：本应用笔记对应的代码是基于雅特力提供的V2.x.x 板级支持包（BSP）而开发，对于其他版本BSP，需要注意使用上的区别。

支持型号列表：

支持型号	AT32 全系列

目录

表目录

图目录

1 内容概述

本文档主要介绍AT芯片在Keil和IAR两种工程环境下的printf功能使用方法。其共包含如下表统计的6种方法,各方法的具体用法在具体内容中描述。

表 1. AT 芯片的 printf 函数使用方法汇总

AT 芯片的 printf 函数使用方法汇总			
序号	**工程环境**	**用法简介**	**详细说明**
用法 1	IAR	经平台自身的 Terminal I/O 虚拟终端输出	具体内容 3.1
用法 2		重定向为串口输出	具体内容 3.2
用法 3	Keil	经平台自身的 Debug(printf) Viewer 虚拟终端输出	具体内容 3.3
用法 4		使用 MicroLIB 的情况下，重定向为串口输出	具体内容 3.4
用法 5		不使用 MicroLIB 的情况下，重定向为串口输出	具体内容 3.5
用法 6	IAR/Keil	经 JLinkRTTClient 窗口输出	具体内容 3.6

2 具体内容

2.1 IAR 环境下经 Terminal I/O 虚拟终端输出

2.1.1 简介

IAR提供的链接到其Terminal的驱动内就包含有常用的scanf和printf等标准输入输出驱动函数，所以工程文件可直接经IAR自带的Terminal I/O窗口实现信息交互。

2.1.2 例程路径

001_Printf_Test_IAR_Terminal\project\iar_v8.2

2.1.3 环境及硬件设计

2.1.3.1 环境

本方法需在IAR环境下使用，例程支持的编译环境为IAR_V8，硬件电路板为AT-START-F403A_V1.2。

2.1.3.2 硬件连接

J-Link/AT-Link&... connection

表 2. 硬件连接关系表--(无 JTDO)

硬件连接关系表--(无 JTDO)			
序号	**AT-START-F403A_V1.2**	**J-Link/AT-Link&...**	**Attention**
1	3.3V	3.3V	None
2	PA13	SWDIO	Must Pull up external
3	PA14	SWCLK	Must Pull down external
4	NRST	RSTn	None
5	GND	GND	None

2.1.4 软件设计

2.1.4.1 头文件

代码工程文件内添加 “stdio.h”。

2.1.4.2 重定向设定

Printf解除重定向（屏蔽与实际串口的重定向）。

2.1.5 仿真与下载

代码经编译后下载到MCU内，然后进入Debug调试环境中，经View->Terminal I/O（下图1）调出虚

拟终端，然后全速运行代码即可看到程序主循环内的“Hello World”等内容被打印到了终端交互窗口Output栏（下图1）内，且在该窗口的Input栏内输入的数据也同样会被打印到Output窗口内。

图 1. 虚拟终端窗口路径

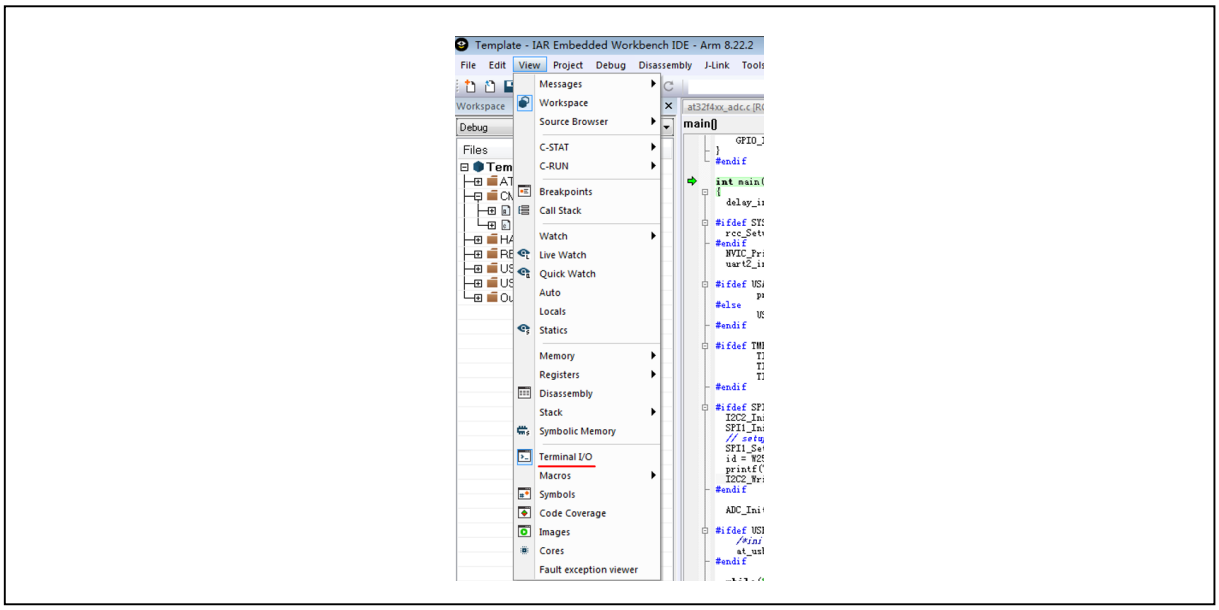
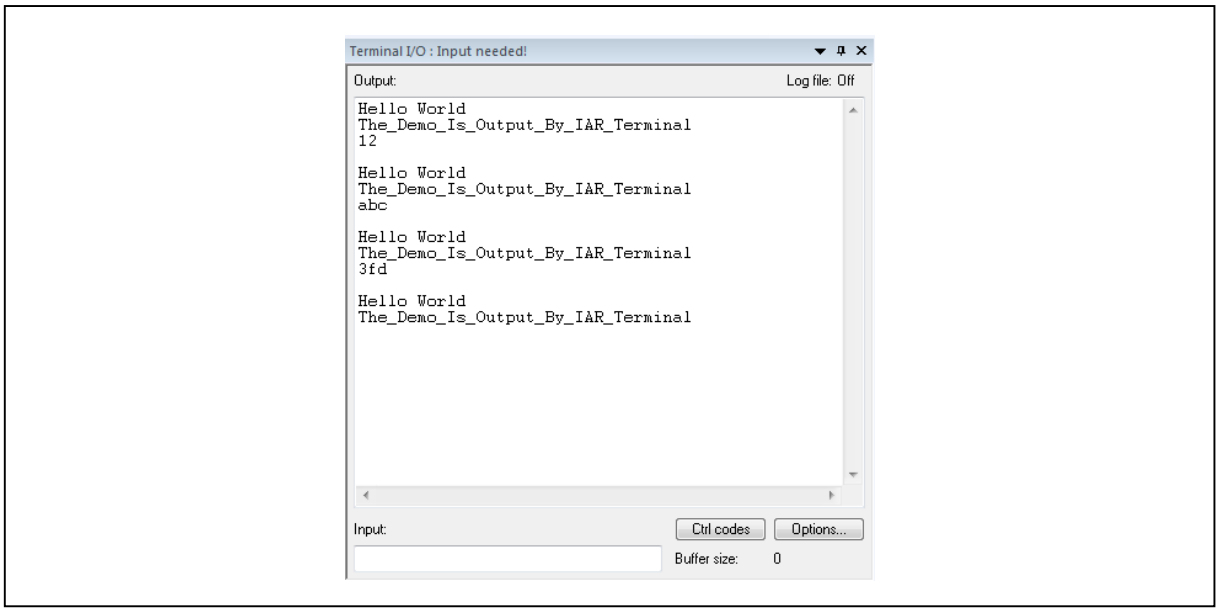


图 2. 虚拟终端交互窗口



2.2 IAR 环境下重定向为串口输出

2.2.1 简介

将printf函数重定向到芯片内的一组实际串口，经串口TX脚输出，最后由串口助手工具进行信息交互。

2.2.2 例程路径

002_Printf_Test_IAR_USART2\project\iar_v8.2

2.2.3 环境及硬件设计

2.2.3.1 环境

本方法需在IAR环境下使用，例程支持的编译环境为IAR_V8，硬件电路板为AT-START-F403A_V1.2。

2.2.3.2 硬件连接

2.2.3.2.1 J-Link/AT-Link&... connection

J-Link/AT-Link&... connection

表 3. 硬件连接关系表--(无 JTDO)

硬件连接关系表--(无 JTDO)			
序号	**AT-START-F403A_V1.2**	**J-Link/AT-Link&...**	**Attention**
1	3.3V	3.3V	None
2	PA13	SWDIO	Must Pull up external
3	PA14	SWCLK	Must Pull down external
4	NRST	RSTn	None
5	GND	GND	None

2.2.3.2.2 USART2 connection

表 4. 硬件连接关系表--(USART)

硬件连接关系表--(USART)			
序号	**AT-START-F403_V1.2**	**USB_To_TTL(CH340)**	**Attention**
1	GND	GMD	None
2	PA2	RXD	None
3	PA3	TXD	None

2.2.4 软件设计

2.2.4.1 头文件

代码工程文件内添加“stdio.h”；

2.2.4.2 重定向设定

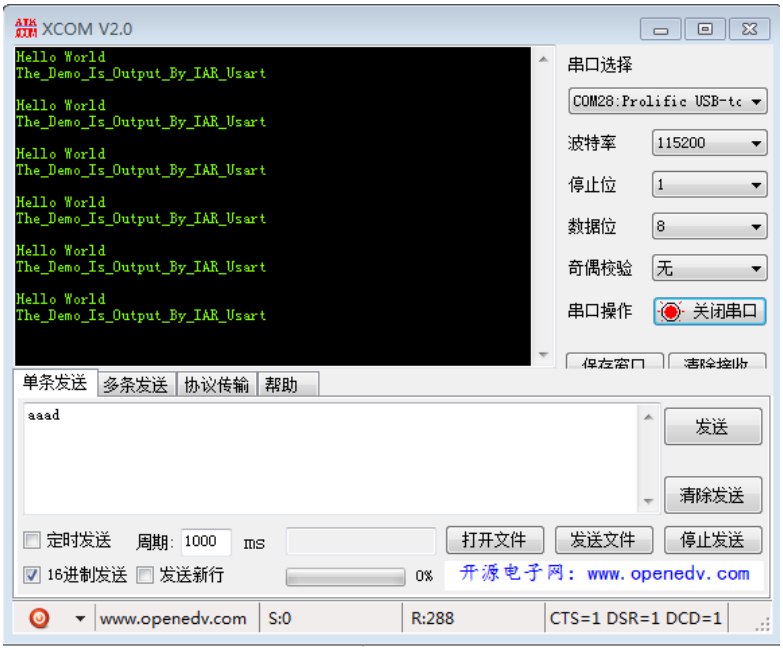
串口初始化并将Printf重定向到实际的串口，重定向函数如下

```
PUTCHAR_PROTOTYPE
{
  USART_SendData( USART2, ch);
  while ( USART_GetFlagStatus( USART2, USART_FLAG_TRAC) == RESET );
  return ch;
}
```

2.2.5 仿真与下载

代码经编译后下载到MCU内，然后全速运行代码即可看到程序主循环内的“Hello World”等内容被打印到了终端交互窗口（下图3）内。

图 3. 串口助手交互窗口



2.3 经 Keil 平台的 Debug(printf) Viewer 虚拟终端输出

2.3.1 简介

Keil平台自带有Debug(printf) Viewer接口，在ARM内核集成有常用的scanf和printf等标准输入输出驱动函数的前提下，该接口可用于标准的Printf交互。

2.3.2 例程路径

003_Printf_Test_Keil_JTDO\project\mdk_v 5

2.3.3 环境及硬件设计

2.3.3.1 环境

本方法需在Keil环境下使用，例程支持的编译环境为Keil_V5，硬件电路板为AT-START-F403A_V1.2

2.3.3.2 硬件连接

J-Link/AT-Link&... connection

表 5. 硬件连接关系表--(含 JTDO)

硬件连接关系表--(含 JTDO)			
序号	**AT-START-F403A_V1.2**	**J-Link/AT-Link&…**	**Attention**
1	3.3V	3.3V	None
2	PA13	SWDIO	Must Pull up external
3	PA14	SWCLK	Must Pull down external
4	NRST	RSTn	None
5	PB3	JTDO	Must Pull up external
6	GND	GND	None

2.3.4 软件设计

2.3.4.1 头文件

代码工程文件内添加 “stdio.h”；

2.3.4.2 跟踪引脚分配

```
DEBUG->ctrl_bit.trace_ioen = FALSE;
DEBUG->ctrl_bit.trace_ioen = TRUE;
```

2.3.4.3 Printf 映射

```
int fputc(int c, FILE *f)
{
if (c == '\n')
{
SER_PutChar('\r');
}
return (SER_PutChar(c));
}
int SER_PutChar (int c)
{
ITM_SendChar(c);
return (c);
}
```

2.3.5 仿真与下载

勾选如下图4中的Enable，并设定Core值，Core值需与系统时钟相等。

设定串口时钟，通常通过勾选如下图4中的Autodetect max SWO C1来实现。当出现打印乱码时，此时可尝试不勾选Autodetect max SWO C1，并手动修改Prescale Core Clk保证打印信息正常。

随后即可编译代码下载到MCU内，然后进入Debug调试环境中，经View->Serial Windows->Debug (printf) Viewer（下图5）调出虚拟终端窗口，然后运行代码即可看到Hello World被实际打印到了终端交互窗口（下图6）内。

图 4. Trace 相关设定

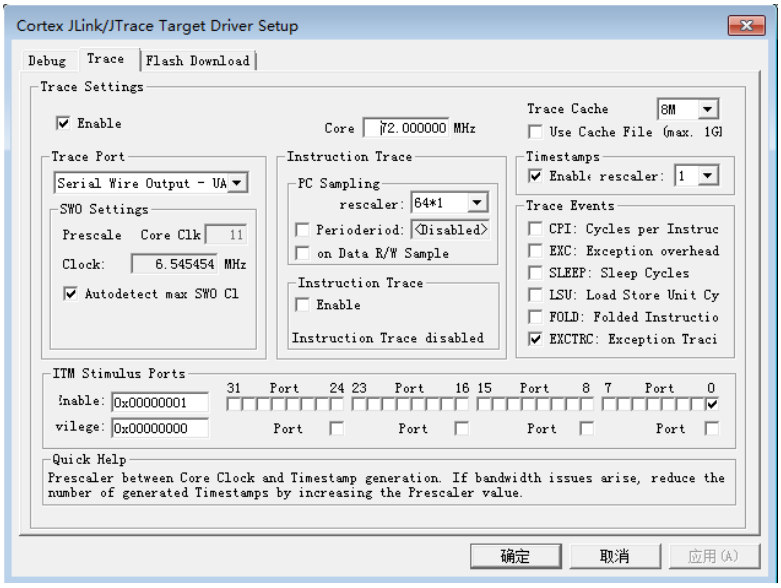


图 5. Keil 虚拟终端窗口路径

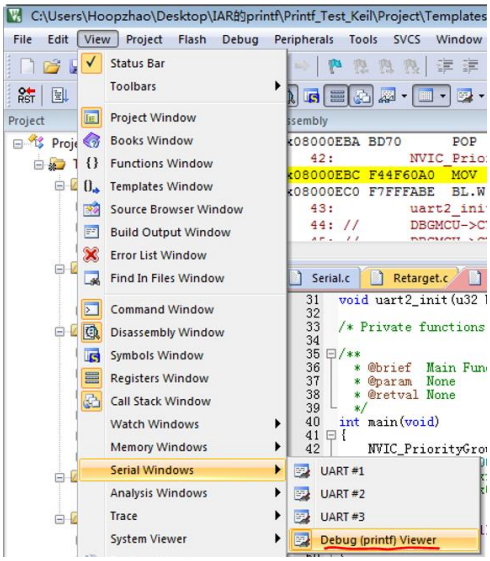
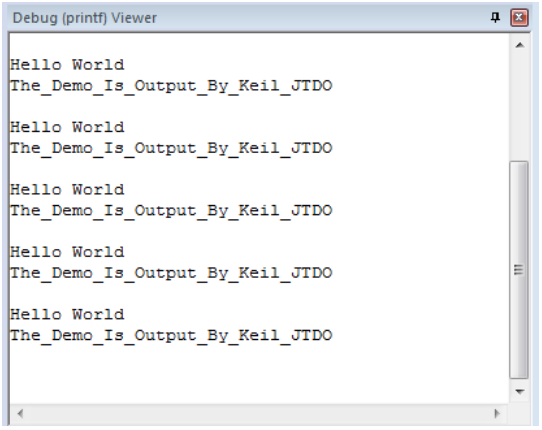


图 6. Keil 虚拟终端交互窗口



2.4 Keil 环境下重定向为串口输出（使用 MicroLIB）

2.4.1 简介

Keil环境有自带一个MicroLIB库，其内包含一些支持Printf函数的代码，在将Printf重定向到串口输出时，勾选使用MicroLIB后可由串口助手工具进行信息交互。

2.4.2 例程路径

004_Printf_Test_Keil_USART2_MicroLIB\project\mdk_v5

2.4.3 环境及硬件设计

2.4.3.1 环境

本方法需在Keil环境下使用，例程支持的编译环境为Keil_V5，硬件电路板为AT-START-F403A_V1.2

2.4.3.2 硬件连接

2.4.3.2.1 J-Link/AT-Link&... connection

表 6. 硬件连接关系表--(无 JTDO)

硬件连接关系表--(无 JTDO)			
序号	**AT-START-F403A_V1.2**	**J-Link/AT-Link&...**	**Attention**
1	3.3V	3.3V	None
2	PA13	SWDIO	Must Pull up external
3	PA14	SWCLK	Must Pull down external
4	NRST	RSTn	None
5	GND	GND	None

2.4.3.2.2 USART2 connection

表 7. 硬件连接关系表--(USART)

硬件连接关系表--(USART)			
序号	**AT-START-F403_V1.2**	**USB_To_TTL(CH340)**	**Attention**
1	GND	GMD	None
2	PA2	RXD	None
3	PA3	TXD	None

2.4.4 软件设计

2.4.4.1 头文件

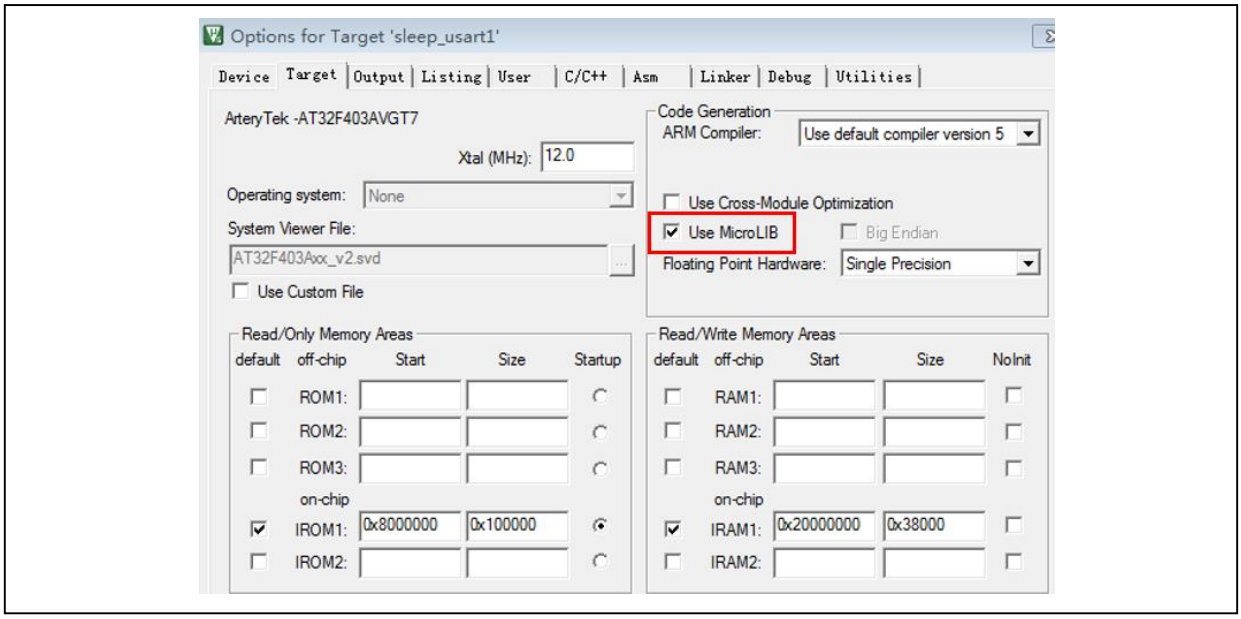
代码工程文件内添加 “stdio.h”；

2.4.4.2 Printf 重定向

```
PUTCHAR_PROTOTYPE
{
  USART_SendData( USART2, ch);
  while ( USART_GetFlagStatus( USART2, USART_FLAG_TRAC) == RESET );
  return ch;
}
```

2.4.4.3 MicroLIB 设定

图 7. MicroLIB 设定



2.4.5 仿真与下载

代码经编译后下载到MCU内，然后全速运行代码即可看到程序主循环内的“Hello World”等内容被打印到了终端交互窗口（下图8）内。

图 8. 串口助手交互窗口


XCOM V2.0
Hello World
The_Demo_Is_Output_By_Keil_USART2_MicroLIB
Hello World
The_Demo_Is_Output_By_Keil_USART2_MicroLIB
Hello World
The_Demo_Is_Output_By_Keil_USART2_MicroLIB
Hello World
The_Demo_Is_Output_By_Keil_USART2_MicroLIB
串口选择
COM28:Prolific USB-tc
波特率
115200
停止位
1
数据位
8
奇偶校验
无
串口操作
关闭串口
单条发送
多条发送
协议传输
帮助
aaad
发送
清除发送
定时发送
周期:
1000
ms
打开文件
发送文件
停止发送
16进制发送
发送新行
0%
开源电子网：www.openedv.com
www.openedv.com
S:0
R:236
CTS=1 DSR=1 DCD=1

2.5 Keil 环境下重定向为串口输出（不使用 MicroLIB）

2.5.1 简介

Keil环境有自带一个MicroLIB库，其内包含一些支持Printf函数的代码。在将Printf重定向到串口输出时，如果不勾选使用MicroLIB，自行在工程文件内添加支持Printf函数的代码段，也同样可由串口助手工具进行信息交互。

2.5.2 例程路径

005_Printf_Test_Keil_USART2__Without_MicroLIB\project\mdk_v5

2.5.3 环境及硬件设计

2.5.3.1 环境

本方法需在Keil环境下使用，例程支持的编译环境为Keil_V5，硬件电路板为AT-START-F403A_V1.2

2.5.3.2 硬件连接

2.5.3.2.1 J-Link/AT-Link&... connection

表 8. 硬件连接关系表--(无 JTDO)

硬件连接关系表--(无 JTDO)			
序号	**AT-START-F403A_V1.2**	**J-Link/AT-Link&...**	**Attention**
1	3.3V	3.3V	None
2	PA13	SWDIO	Must Pull up external
3	PA14	SWCLK	Must Pull down external
4	NRST	RSTn	None
5	GND	GND	None

2.5.3.2.2 USART2 connection

表 9. 硬件连接关系表--(USART)

硬件连接关系表--(USART)			
序号	**AT-START-F403A_V1.2**	**USB_To_TTL(CH340)**	**Attention**
1	GND	GMD	None
2	PA2	RXD	None
3	PA3	TXD	None

2.5.4 软件设计

2.5.4.1 头文件

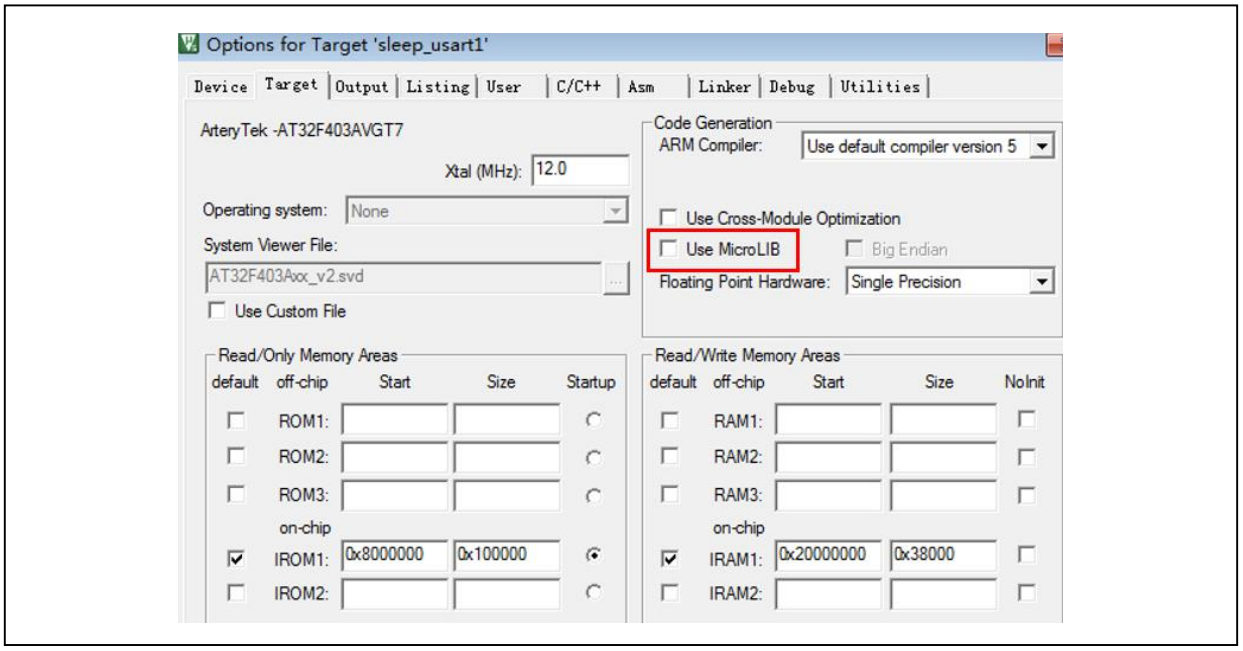
代码工程文件内添加 “stdio.h”；

2.5.4.2 Printf 重定向

```
PUTCHAR_PROTOTYPE
{
  USART_SendData( USART2, ch);
  while ( USART_GetFlagStatus( USART2, USART_FLAG_TRAC) == RESET );
  return ch;
}
```

2.5.4.3 MicroLIB 设定

图 9. MicroLIB 设定



2.5.4.4 Printf 函数支持代码添加

```
#if (__ARMCC_VERSION > 6000000)
  __asm (".global __use_no_semihosting\n\t");
  void _sys_exit(int x)
  {
    x = x;
  }
  /* __use_no_semihosting was requested, but _ttywrch was */
  void _ttywrch(int ch)
  {
    ch = ch;
```

```
  }
  FILE __stdout;
#else
 #ifdef __CC_ARM
  #pragma import(__use_no_semihosting)
  struct __FILE
  {
    int handle;
  };
  FILE __stdout;
  void _sys_exit(int x)
  {
    x = x;
  }
 #endif
#endif#if (__ARMCC_VERSION > 6000000)
  __asm (".global __use_no_semihosting\n\t");
  void _sys_exit(int x)
  {
    x = x;
  }
  /* __use_no_semihosting was requested, but _ttywrch was */
  void _ttywrch(int ch)
  {
    ch = ch;
  }
  FILE __stdout;
#else
 #ifdef __CC_ARM
  #pragma import(__use_no_semihosting)
  struct __FILE
  {
    int handle;
  };
  FILE __stdout;
  void _sys_exit(int x)
  {
    x = x;
  }
 #endif
#endif
```

2.5.5 仿真与下载

代码经编译后下载到MCU内，然后全速运行代码即可看到程序主循环内的“Hello World”等内容被打印到了终端交互窗口（下图10）内。

图 10. 串口助手交互窗口

2.6 经 JLinkRTT 窗口输出

2.6.1 简介

JLink自带有调试输出功能，在添加JLink RTT库代码后，可根据指定的指令实现代码调试输出到对应的窗口。

2.6.2 例程路径

006_Printf_Test_Jlink_RTT\project\mdk_v5

2.6.3 环境及硬件设计

2.6.3.1 环境

本方法在IAR及Keil环境下均可使用，例程支持的编译环境为IAR_V8、Keil_V5，硬件电路板为AT-START-F403A_V1.2。

2.6.3.2 硬件连接

J-Link connection

表 10. 硬件连接关系表--(无 JTDO)

硬件连接关系表--(无 JTDO)			
序号	**AT-START-F403A_V1.2**	**J-Link**	**Attention**
1	3.3V	3.3V	None
2	PA13	SWDIO	Must Pull up external
3	PA14	SWCLK	Must Pull down external
4	NRST	RSTn	None
5	GND	GND	None

2.6.4 软件设计

2.6.4.1 头文件

代码工程文件内添加 “stdio.h”；

2.6.4.2 添加 JLink RTT 库代码

- 分别将JLink RTT库代码中的SEGGER_RTT.c和SEGGER_RTT_printf.c添加到工程文件内；
- 根据编译环境选择添加SEGGER_RTT_Syscalls_IAR.c或SEGGER_RTT_Syscalls_KEIL.c到工程文件内；

2.6.4.3 输出到 PC

此时，代码内调用如下SEGGER_RTT_WriteString或SEGGER_RTT_printf命令即可输出到PC端

```
SEGGER_RTT_WriteString(0, "SEGGER Real-Time-Terminal Sample\r\n\r\n");
SEGGER_RTT_printf(0, "printf Test: %%c, 'S' : %c.\r\n", 'S');
```

2.6.5 仿真与下载

2.6.5.1 通过 JLinkRTTClient 窗口输出

代码经编译后下载到MCU内，然后进入Debug调试环境中，打开JLink安装路径下名称为JLinkRTTClient的应用程序。此时单步执行代码时即可看到打印信息被依次输出到JLinkRTTClient窗口，如下图11和图12。

图 11. 代码工程 Debug

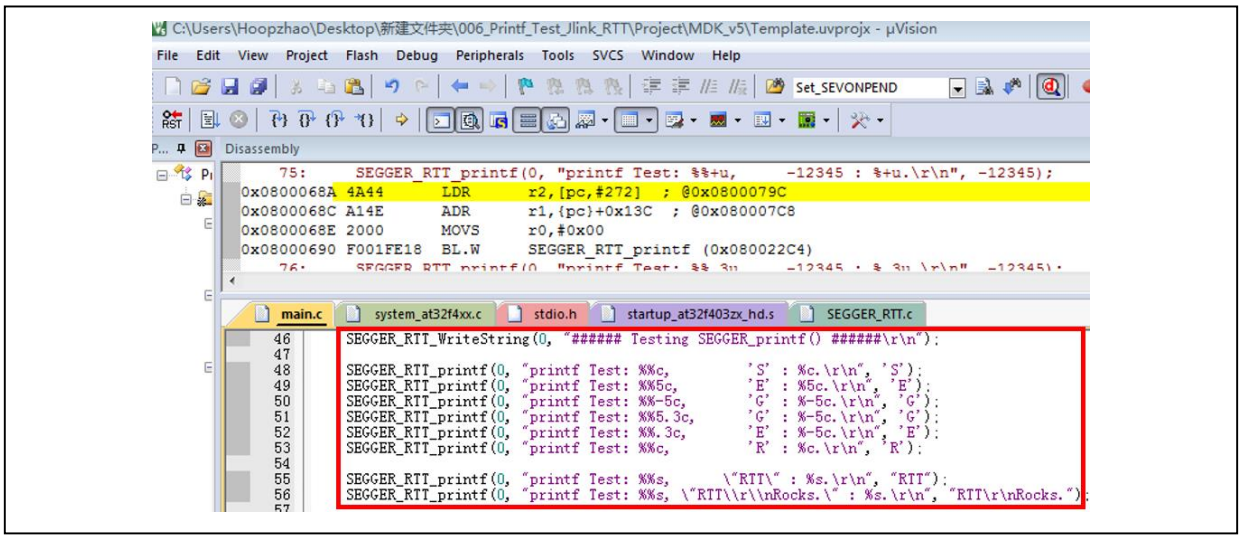
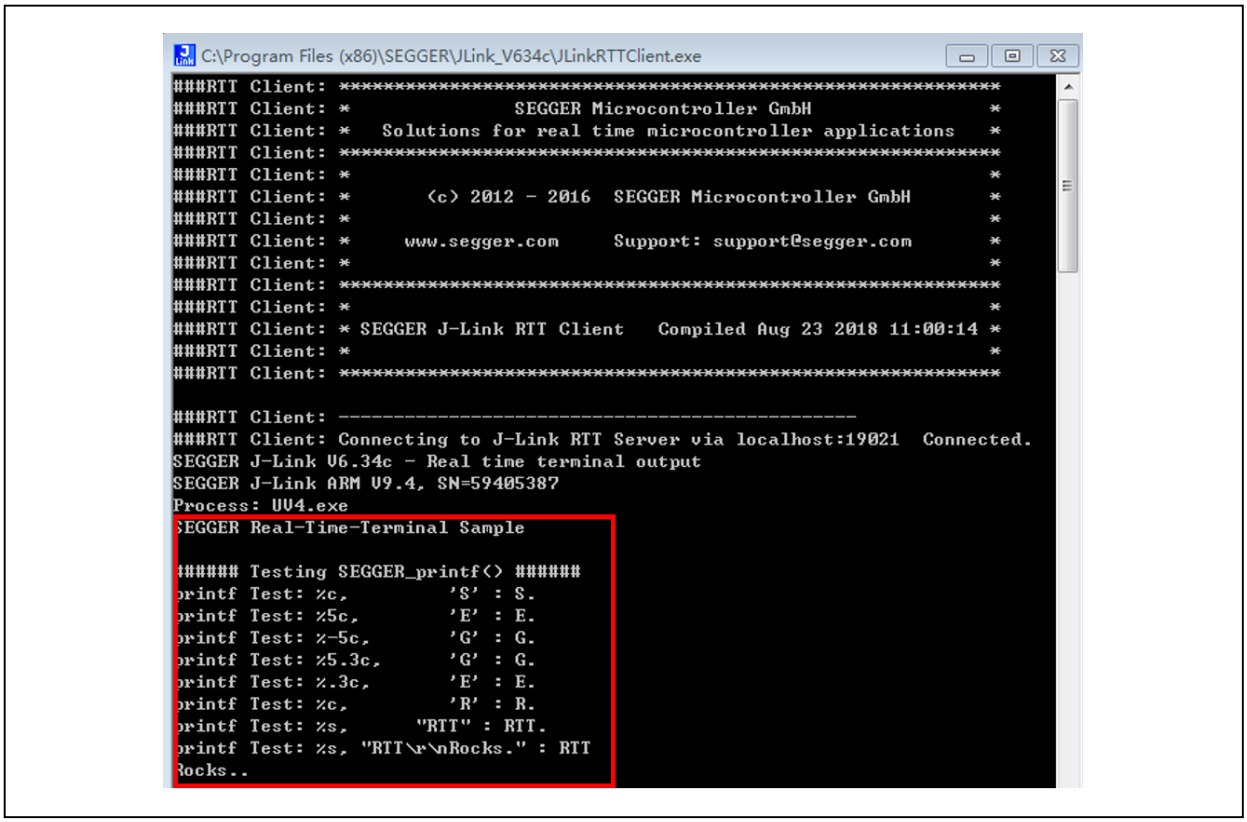


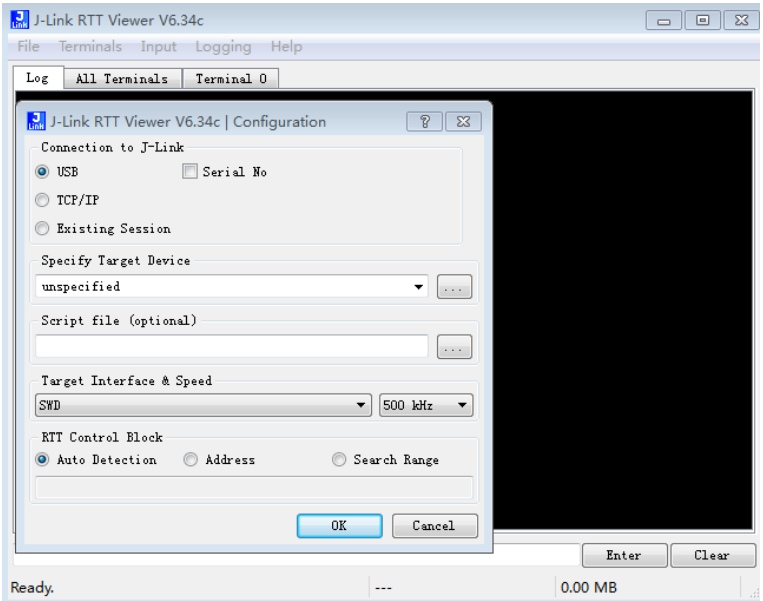
图 12. JLinkRTTClient 窗口输出信息



2.6.5.2 通过 JLinkRTTViewer 窗口输出

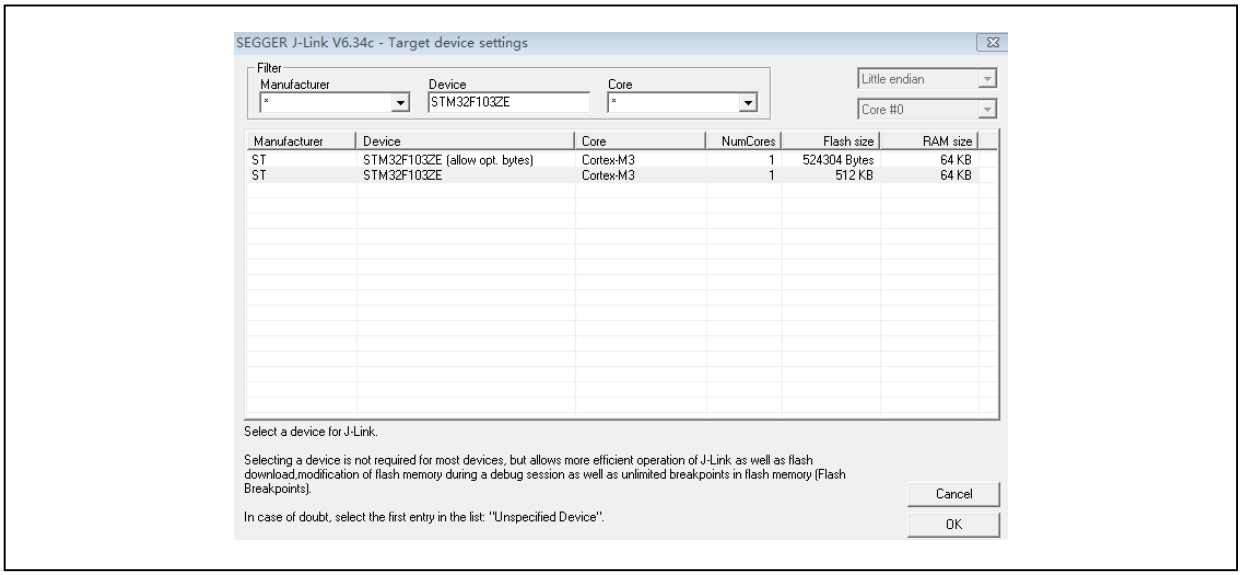
1. 代码经编译后下载到MCU内，然后打开JLinkRTTViewer窗口，如下图13。

图 13. 打开 JLinkRTTViewer 窗口



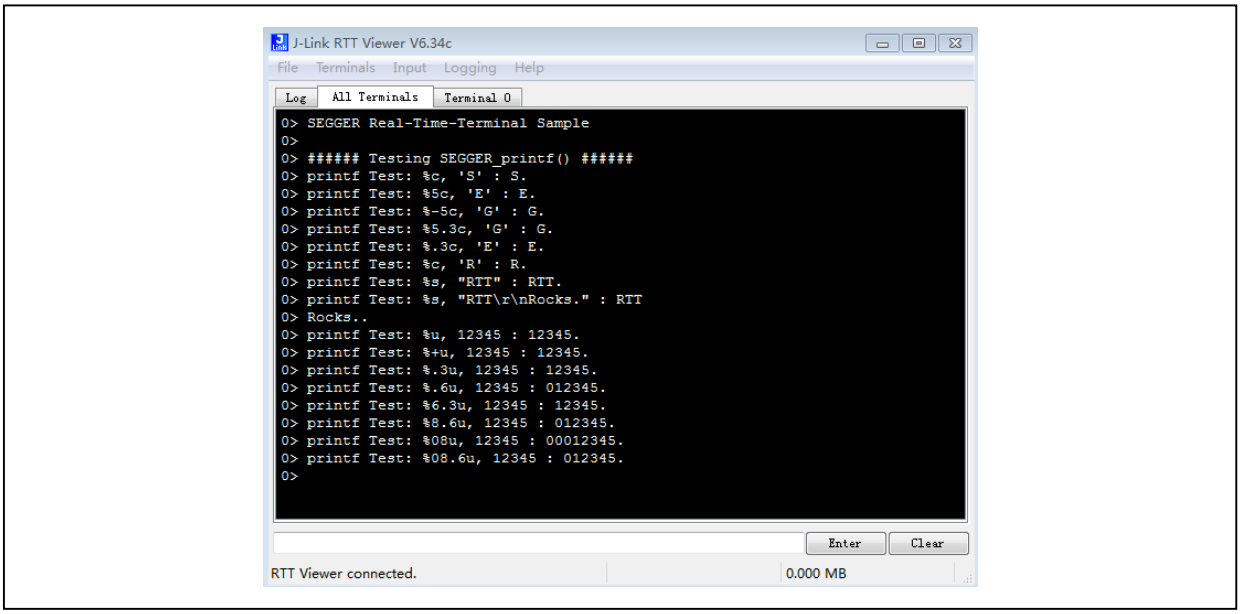
2. 点击OK，并在弹出的窗口再点击OK，然后在弹出的窗口输入并选择如下信息（此处以ZE系列为例），并点击OK。如下图14

图 14. device 选择窗口



3. 拿前述代码进入Debug调试环境中，此时单步执行代码时即可看到打印信息被依次输出到JLinkRTTViewer窗口。如下图15

图 15. JLinkRTTViewer 窗口输出信息



3 注意事项

- 前述具体内容2.3和2.6的测试时，只能用J-Link，AT-Link暂不支持；
- 前述具体内容2.1和具体内容2.2的测试时，如果使用AT-Link的话，工程内Options→CMSIS DAP→Reset选项必须选择为Hardware或者System，不然无法连接及下载代码；
- 前述具体内容2.2的测试时，工程内Options→General Options→Library Configuration→Library选项必须选择为Full，不然无法正常输出。因为只有选择为Full时，支持Printf函数的代码才会被包含进来。
- 前述具体内容2.6.5.1测试时，因输出窗口应用程序暂时无法指定芯片型号，为保证代码与窗口型号匹配且输出正常，目前工程内Device须选择ST的型号。且工程代码内必须要勾选“Options”内的“Use MicroLIB”，不然代码编译可能会出现异常。

4 版本历史

表 11. 文档版本历史

日期	版本	变更
2021.12.07	2.0.0	最初版本

重要通知 – 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独力负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。