登录 | 注册

哎呦不错哦！

也就笔记啊心得啊什么的啦~~~

目录视图 摘要视图 RSS 订阅

个人资料

Maple_Leaf_15

访问： 1857次
积分： 96
等级： BLOG 1
排名： 千里之外

原创： 8篇 转载： 4篇
译文： 0篇 评论： 1条

文章搜索

移动信息安全的漏洞和逆向原理 程序员11月书讯，评论得书啦 Get IT技能知识库，50个领域一键直达

单片机调试——《浅谈工程师的调试法宝（四）——RTT的应用》

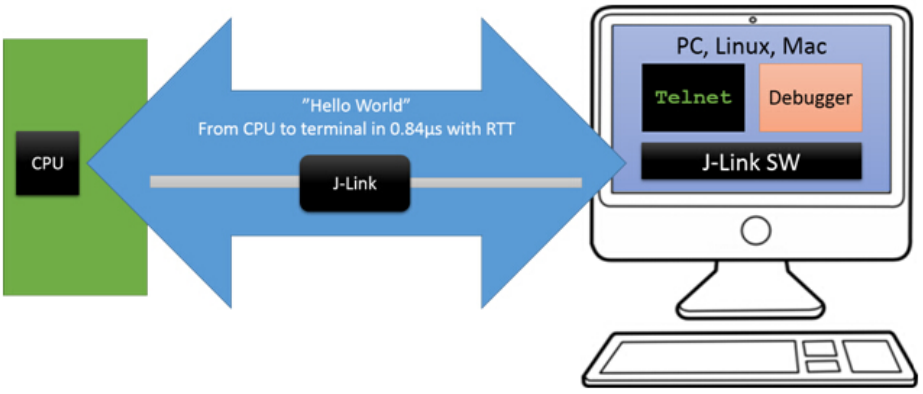
2016-03-30 22:19 371人阅读 评论(0) 收藏 举报

目录(?) [+]

摘要: 我们前三篇的文档中介绍的调试方法，都因为各种原因而不能在所有的MCU上做到通用，而今天这一切将发生改变。现在就一起来看一下这个神奇的调试工具-RTT。

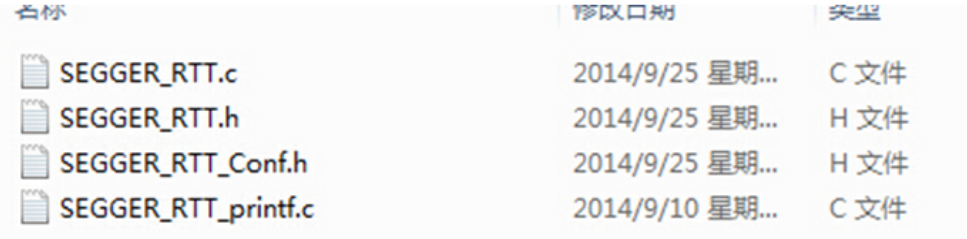
RTT（ Real Time Terminal ）是SEGGER公司新出的可以在嵌入式应用中与用户进行交互的实时终端。J-Link驱动4.90之后的版本都有这个软件哦。

用RTT可以从目标MCU上输出信息的同时也可以非常高速的向应用程序发送信息，并且不影响MCU的实时性。其实现原理就是J-link与MCU共享内存，具体实现细节感兴趣的读者可以自己去查阅下资料，本文以应用为主。RTT的工作框图如图 1所示。MCU通过J-link与电脑连接并将打印信息输出到电脑上，电脑同时可以通过键盘等向MCU发送数据。



如果你想使用它，操作也非常简单，首先从官网下载RTT代码，然后把如图 2所示的4个文件添加到你的工程中。并且在主函数文件的起始处添加SEGGER_RTT.h文件。如下所示。

```
#include "SEGGER_RTT.h"
```



然后我们就可以直接在主函数中调用SEGGER_RTT_printf函数来打印调试信息了，该函数用法和printf函数类似，只是多了一个参数用来指定RTT通道。其中通道0，就是我们在调试时使用的通道。在主函数中添加如下代码。

```
SEGGER_RTT_printf(0,"Times %d\r\n",++u32Counter);
```

u32Counter 这个变量每次打印完之后都会递增。我们把程序编译，然后进入调试模式，在开始菜单下打开J-link RTT
Client，可以看到如图 3所示的信息。

static全局变量与普通的 (0)
从舵机程序到栈 (0)
"*** error 65: access viol (0)
单片机调试——《浅谈工 (0)
Keil for ARM-MDK的使用 (0)
笔记-一些MOS管和栅驱 (0)
连接器 (0)

推荐文章

* 程序员10月书讯，评论得书
* Android中Xposed框架篇---修改系统位置信息实现自身隐藏功能
* Chromium插件（Plugin）模块（Module）加载过程分析
* Android TV开发总结--构建一个TV app的直播节目实例
* 架构设计：系统存储--MySQL简单主从方案及暴露的问题

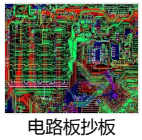
最新评论

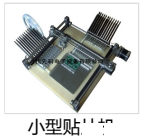
基于TCP的STM32 IAP bootload Maple_Leaf_15: 博主你好！你这篇文章讲的很好，虽然后半部分我没有弄懂。。我读懂的部分想问几个问题：1.能不能不用外部...

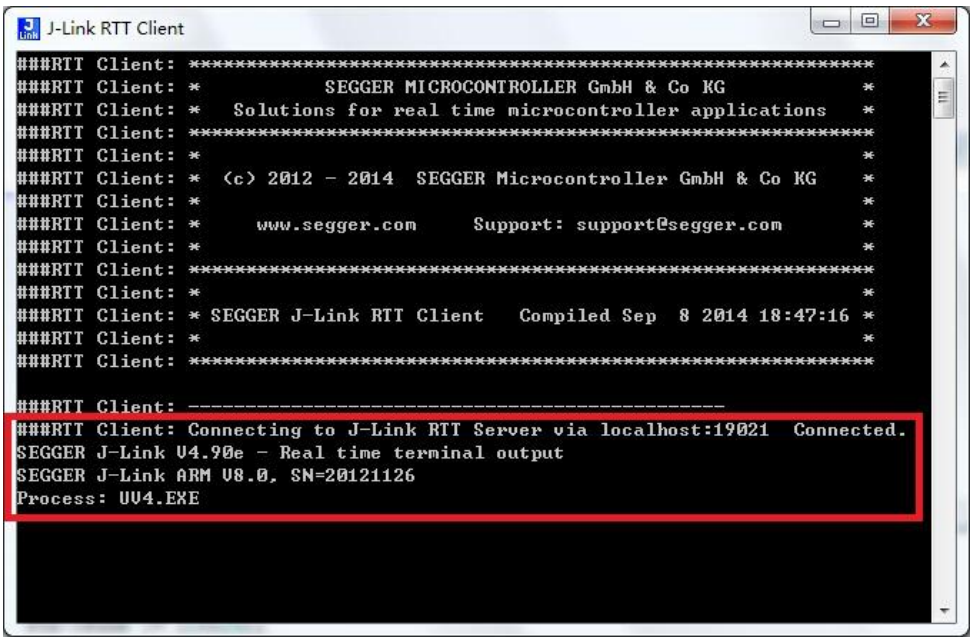
基于TCP的STM32 IAP bootload Maple_Leaf_15: 博主你好！你这篇文章讲的很好，虽然后半部分我没有弄懂。。就我读懂的部分想问几个问题：1.能不能不用外...


埋线瘦脸

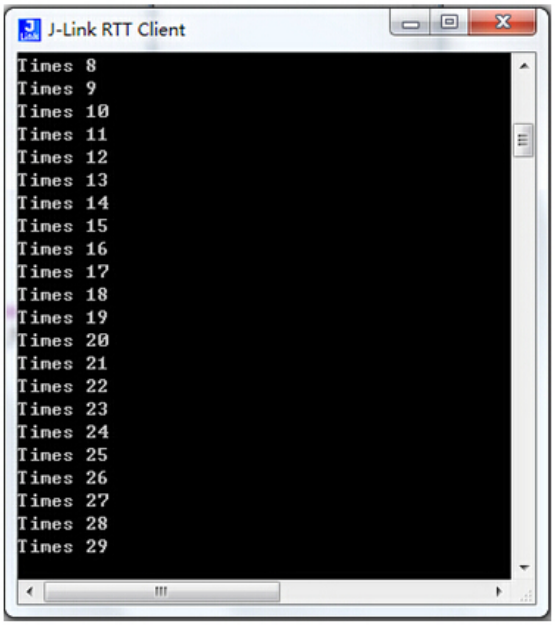

软件工程师待遇


电路板抄板


小型贴

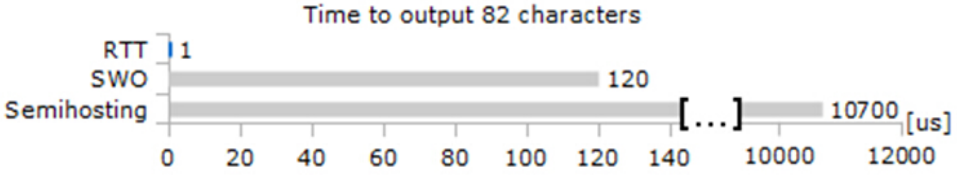


如果已经正常连接，就会打印出入红色框内的信息，标明RTT版本号和J-link固件版本号等，如果没有这些信息则表示连接异常。点击全速运行之后，打印出的信息如图 3所示。



是不是感觉很酷？只需要简单的添加几个文件，就可以直接调用SEGGER_RTT_printf函数打印输出了。没有复杂的工程配置，没有MCU内核的限制，并且打印字符还非常的流畅，好像回到了用VC6.0编写Win32控制台程序的时代一样。

与前面介绍的SWO、Semihosting，相比，RTT字符输出更快。输出82个字符所需要的时间如图 4所示。



通过与前面的3篇文档进行对比，我们发现每个调试工具都各有长短，关于它们的具体比较请参考表 1。

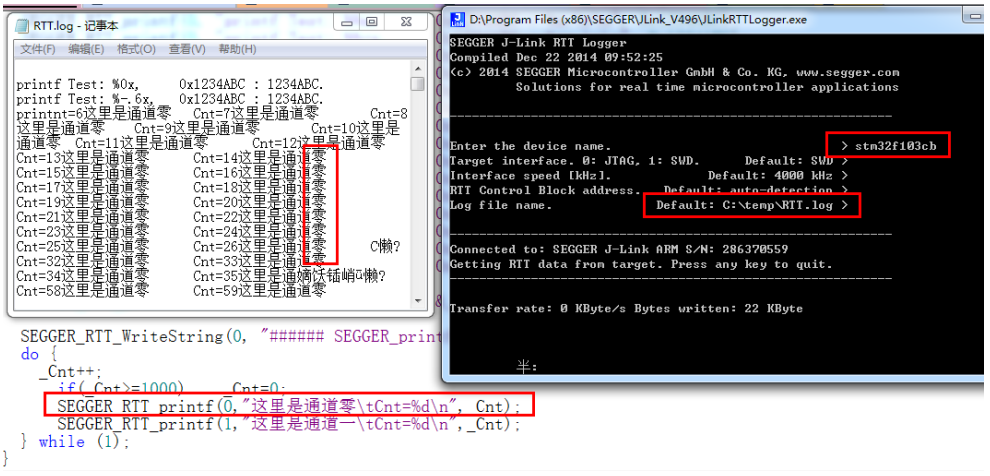
名称	MCU 内核	字符输出速率	编译环境	备注
UART	M0\M0+\M3\M4	依赖于串口配置	所有	使用 MCU 的外设
Semihosting	M0\M0+\M3\M4	慢	IAR	打印速率慢
SWO	M3\M4	较快	KEIL\ IAR	内核有限制
RTT	M0\M0+\M3\M4	快	所有	使用 J-link 调试

通过对上面表格的对比，我们可以根据不同的情况选择合适的工具。加快我们的开发进度。

> 但是有的时候，我们可能还需要将一个变量以曲线的形式形象的表现出来。比如说传感器的变化趋势、电机转速等，这个时候，如果再去开发一个上位机，又加大了开发任务。那么有没有一个工具，可以不用占用MCU外设资源，又可以形象的看到波形呢？敬请大家关注下一篇文档《浅谈工程师的调试法宝（5）之JScope的使用》。

以上文章转载自http://maker.zlgmcu.com/portal.php?mod=view&aid=1880。以下是实验中的几个要点：

1. Segger官网关于RTT功能的说明以及库文件的下载可以从这里找到；
2. 最简单易用的两个函数是
   - `SEGGER_RTT_ConfigUpBuffer(0,NULL,NULL,0,SEGGER_RTT_MODE_BLOCK_IF_FIFO_FULL)` ：非必要的初始化（不知有什么用？）。它是官方例程 `RTT_Implementation_141217-Examples-Main_RTT_PrintfTest.c` 的第一句
   - `SEGGER_RTT_WriteString(0,"字符串")` ：直接输出字符串
   - `SEGGER_RTT_printf（0，"字符串"，输出格式）` ：相当于printf，可以输出各种数据类型
3. 查看RTT输出的工具有三个：
   - RTTViewer：不支持中文。至少要进入一次Debugger才能正常显示输出。建议进入Debugger之后再打开，否则经常不能正常显示输出
   - RTTLogger：支持中文，并且可以保存为log文件。使用具体的正确使用方法不清楚。根据手册说明，log只接收RTT通道1的输出，即 `SEGGER_RTT_printf（1，"字符串"，输出格式）` 。但是实测，只能输出RTT通道0的信息，并且要求代码中要有使用到通道1的语句。否则收不到数据。
   - RTTClient：必须配合RTTLogger或者keil的Debugger来使用，而RTTLogger也必须配合Debugger使用。Client、Logger和Debugger三个窗口都打开的时候，Client和Logger只有其中一个能正常显示，另外一个会严重丢失数据。
4. 无论哪个查看RTT的工具，都会丢失数据。可能与输出函数的使用频率过高有关（？）。加大 `SEGGER_RTT_Conf.h` 中的 `#define BUFFER_SIZE_UP` 的值可以有效降低丢失率。
5. 从开始菜单栏找到 `J-Link User Manual (UM08001)` 文档，里面有JLink各个工具的说明。



顶 0　　踩 0

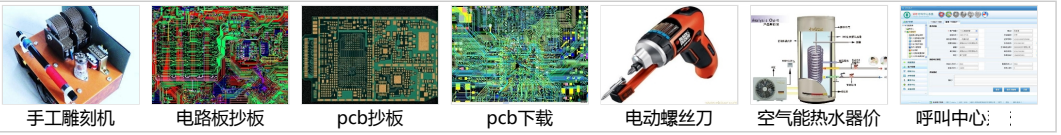
上一篇 Keil for ARM-MDK的使用

下一篇 “*** error 65: access violation at 0x0000000C : no 'read' permission”错误的解决

猜你在找

iOS开发—真机调试 | 应用发布
iOS进阶开发-调试程序
幕后英雄的用武之地浅谈Java内部类的四个应用场景转
android应用程序fps meter帧数显示的分析 浅谈root的

iOS8开发视频教程Swift语言版-Part 1:第一个iOS应用
C语言在嵌入式开发中的应用
大数据应用开发之数据清洗开胃、生产、实操
android应用程序fps meter帧数显示的分析 浅谈root的
幕后英雄的用武之地浅谈Java内部类的四个应用场景
幕后英雄的用武之地浅谈Java内部类的四个应用场景



查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack FTC
coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved