

SEGGER SystemView

1. SEGGER SystemView简介及PC端程序安装

1.1什么是SEGGER SystemView?

SystemView 是一个用于分析嵌入式系统的工具。SystemView 可以完整的深入观察一个应用程序的运行时行为。

由两部分组成：

- SystemView 的PC端程序：收集目标板上传的信息
- SystemView 嵌入式端程序：分析嵌入式系统的行为。

类似BLE

首先上位机PC端有一个处理程序，下位机ARM上需增加部分代码，用于记录嵌入式系统的一些数据，通过连接的仿真器接口将数据传输到上位机，然后PC端的处理程序处理这些数据，由于数据传输使用的是SEGGER J-link的实时传输技术（RTT），所以SystemView可以实时分析和展示数据。分析的内容包括中断、任务、软件定时器执行的时间，切换的时间，以及发生的事件等，这些分析都是实时的，丝毫不影响下位机的运行。这样就可以验证我们设计的整个嵌入式系统是否按照我们的预期在工作，比如任务的切换逻辑，中断的触发等

1.2 SEGGER SystemViewPC端程序安装

安装包下载地址：

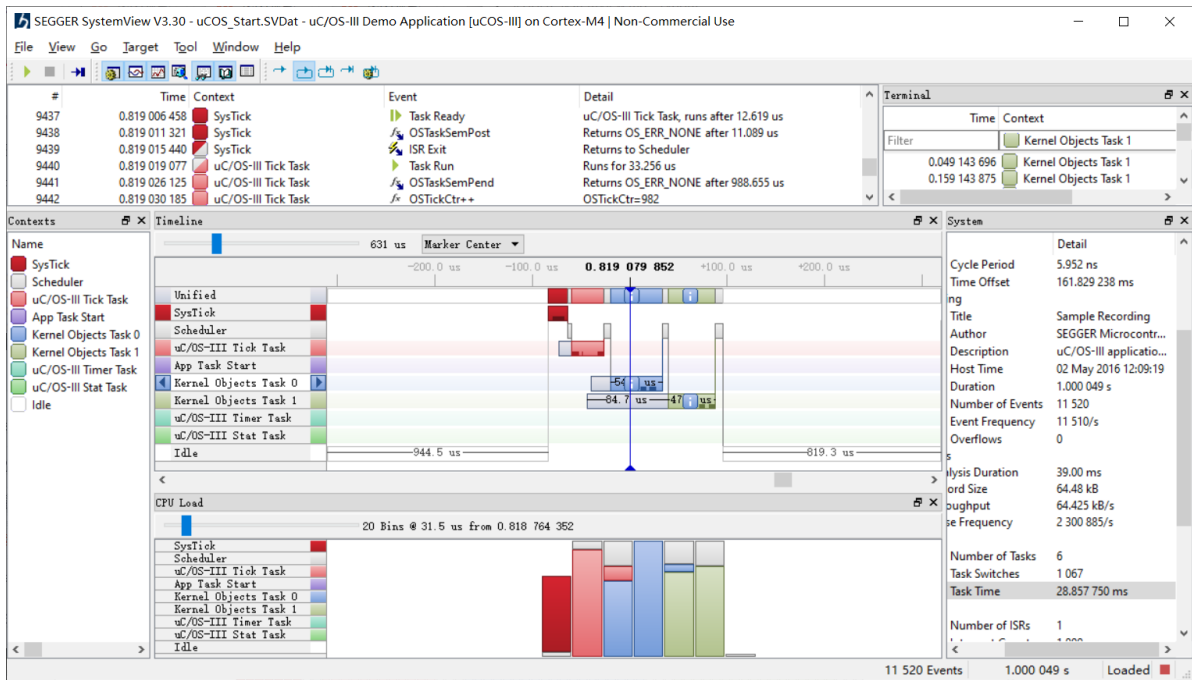
[SEGGER - The Embedded Experts - Downloads - SystemView](#)

文件	描述
./SystemView.exe	启动程序
./Doc/UM08027_SystemView.pdf	指导手册
./Description/SYSTEMVIEW_*.txt	SystemView API描述文件
./Sample	例程跟踪文件

1.3 SEGGER SystemViewPC端程序简介

File View Windows

- 主窗口
- System
- contexts
- timeline
- CPU load
- RunTime



2. SEGGER SystemView 嵌入式端程序安装

2.1将ST-LINK更新为J-Link

SEGGER 官网中有详细说明（

- Make sure that the ST-LINK USB drivers are installed: [Download](#)
 - Make sure that the J-Link software package V5.12b or later is installed: [Download](#)
 - Download the SEGGER STLinkReflash utility: [Download](#)
 - Start the STLinkReflash utility
 - Agree to the license terms
-)

主要步骤

(1) 安装ST-LINK和J-LINK的驱动 (2) 下载SEGGER官网提供的STLinkReflash程序 (3) 启动STLinkReflash程序，根据提示完成更新。

```
C:\Users\你安吧\Downloads\STLinkReflash_190812\STLinkReflash.exe

(c) 2016 SEGGER Microcontroller GmbH & Co. KG    www.segger.com
STLinkReflash compiled Aug 12 2019 10:30:05

The following terms come from SEGGER Microcontroller GmbH & Co. KG ("SEGGER")
You must agree to them in order to proceed.

=====
* IMPORTANT *
This utility enables You to replace the firmware of an existing ST-LINK
on-board with firmware from SEGGER that makes the ST-LINK on-board
behave J-Link compatible. You do this replacement at Your own risk.
Though extremely unlikely, You are aware that the replacement process may
result in an unusable ST-LINK on-board. The utility and firmware from SEGGER
are provided on an as-is basis and come without any warranty
and without support.
You further agree to only use the firmware provided by SEGGER via this utility,
within the bounds of the license stated on the download page:
https://www.segger.com/jlink-st-link.html
Except as expressly set forth in this agreement,
the Agreement remains unchanged and continues in full force and effect.

=====

I hereby accept the terms provided by SEGGER.
(A)cccept / (D)ecline
Selection>
```

(STLinkReflash) 程序界面

2.2 SEGGER SystemView目标实现模块

(1) 从[SEGGER - The Embedded Experts - Downloads - SystemView](#)下载最新的源码包。

源码包中的内容：Config，SEGGER和Sample

名称	修改日期	类型	大小
Config	2021/6/7 15:00	文件夹	
Sample	2021/6/7 15:00	文件夹	
SEGGER	2021/6/7 15:00	文件夹	

(2) 将源码添加到工程

我们在工程目录下建立三个新的文件夹，分别为Config，SEGGER和OS

Config文件夹和SEGGER文件夹

文件	描述
/Config/Global.h	SEGGER代码的全局类型定义
/Config/SEGGER_RTT_Conf.	SEGGER RTT配置文件
/Config/SEGGER_SYSVIEW_Conf.h	SEGGER SYSTEMVIEW配置文件

文件	描述
/SEGGER/SEGGER.h	定义了SEGGER全局类型的全局头文件和通用函数
/SEGGER/SEGGER_RTT.c	SEGGER RTT模块源码
/SEGGER/SEGGER_RTT.h	SEGGER RTT模块头文件
/SEGGER/SEGGER_SYSVIEW.c	SEGGER SYSTEMVIEW模块源码
/SEGGER/SEGGER_SYSVIEW.h	SEGGER SYSTEMVIEW模块头文件
/SEGGER/SEGGER_SYSVIEW_ConfDefault.h	SEGGER SYSTEMVIEW模块默认配置头文件
/SEGGER/SEGGER_SYSVIEW_Int.h	用于SEGGER SYSTEMVIEW模块内部调用的头文件

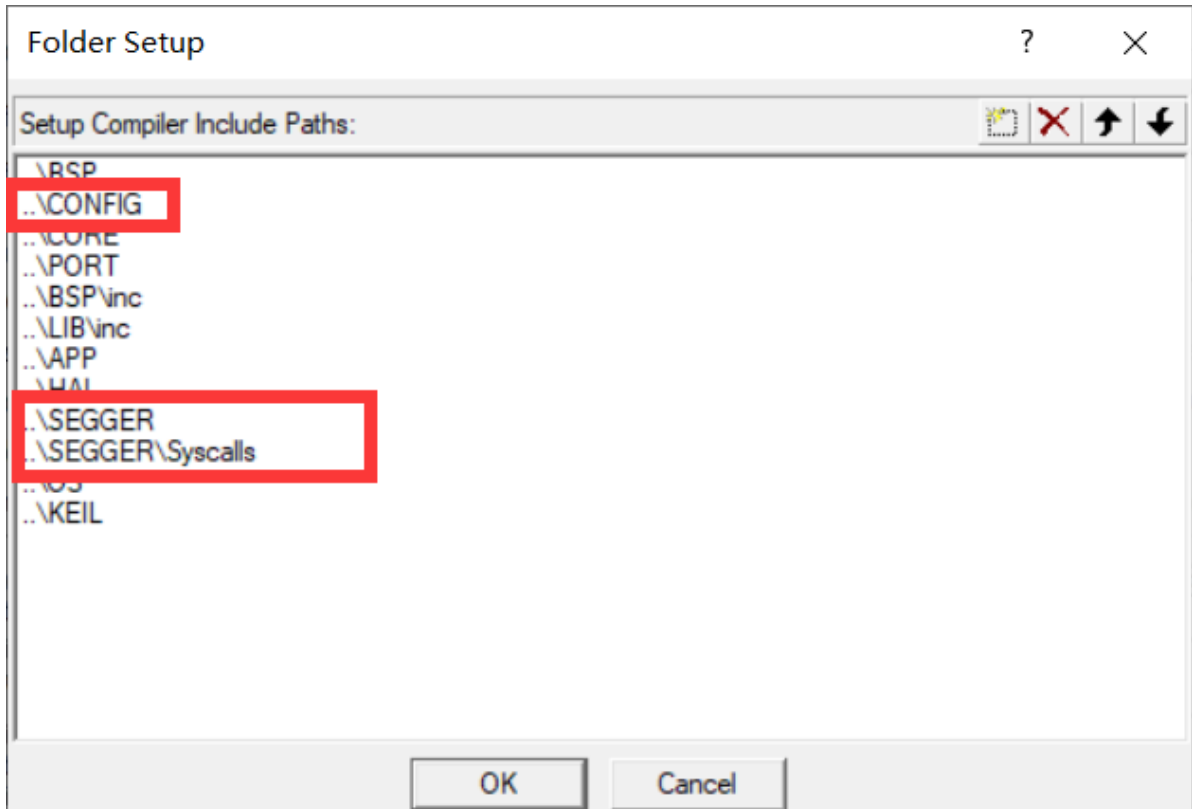
这两个文件夹可以直接复制到工程中，并且保持目录结构。

OS文件夹

文件	描述
Sample\uCOS-II\os_trace_events.h	API定义
Sample\uCOS-II\SEGGER_SYSVIEW_uCOSII.c	SystemView与uc的接口
Sample\uCOS-II\Config\os_cfg_trace.h	
Sample\uCOS-II\Config\Cortex-M\SEGGER_SYSVIEW_Config_uCOSII	ucosii的SystemView初始化配置

将源码包中这些文件添加到工程下OS文件夹中。

添加完毕后，在keil中添加文件，添加引用。



2.3 ST板子相关配置

用以下代码覆盖SEGGER_SYSVIEW_Conf.h

```
/*
*****
* (c) SEGGER Microcontroller GmbH & Co. KG *
*****
----- END-OF-HEADER -----
File : SEGGER_SYSVIEW_Conf.h
Purpose : SEGGER SysView configuration for Cortex-M3 / Cortex-M4.
*/
#ifndef SEGGER_SYSVIEW_CONF_H
#define SEGGER_SYSVIEW_CONF_H
/*
*****
*
* SysView timestamp configuration
*/
// 事件时间戳
#define SEGGER_SYSVIEW_GET_TIMESTAMP() ((* (U32 *) (0xE0001004)) >> 4)
#define SEGGER_SYSVIEW_TIMESTAMP_BITS 28
/*
*****
*
* SysView Id configuration
*/
// 中断ID
// Default value for the lowest Id reported by the application.
// Can be overridden by the application via SEGGER_SYSVIEW_SetRAMBase().
#define SEGGER_SYSVIEW_ID_BASE 0x20000000
// Number of bits to shift the Id to save bandwidth.
// (i.e. 2 when all reported Ids (pointers) are 4 byte aligned)
#define SEGGER_SYSVIEW_ID_SHIFT 0
*****
*/

```

```

*
* SysView interrupt configuration
*/
// Get the currently active interrupt Id. (read Cortex-M ICSR[8:0]= active
vector)
#define SEGGER_SYSVIEW_GET_INTERRUPT_ID() ((*(U32 *) (0xE00ED04)) & 0x1FF)
/*****
*
* SysView locking
*/
// SystemView锁定和解锁
// Lock SysView (nestable)
#define SEGGER_SYSVIEW_LOCK() SEGGER_RTT_LOCK()
// Unlock SysView (nestable)
#define SEGGER_SYSVIEW_UNLOCK() SEGGER_RTT_UNLOCK()
#endif
/***** End of file *****/

```

2.4 UCOSII相关适配

(1) 开启记录选项

注：打开API_ENTER和API_EXIT之后会增加触发的SystemView事件数量，这可能导致SystemView出现overflow的问题。如果这样的话disable这两个宏定义

os_cfg.h

```

#define OS_TRACE_EN 1
#define OS_TRACE_API_ENTER_EN 1
#define OS_TRACE_API_EXIT_EN 1

#define OS_TIME_DLY_HMSM_EN 1u /* Include code for OSTimeDlyHMSM()
*/
#define OS_TIME_DLY_RESUME_EN 1u /* Include code for OSTimeDlyResume()
*/
#define OS_TIME_GET_SET_EN 1u /* Include code for OSTimeGet() and
OSTimeSet() */
#define OS_TIME_TICK_HOOK_EN 1u /* Include code for OSTimeTickHook()
*/

```

(2) os_cfg_trace.h(一般不用修改)

```

#define OS_CFG_TRACE_MAX_TASK 32u
#define OS_CFG_TRACE_MAX_RESOURCES 128u

```

(3)SEGGER_RTT_Conf.h(缓冲区大小)

```

#ifndef BUFFER_SIZE_UP
#define BUFFER_SIZE_UP (4096)
#endif

```

(4)SEGGER_SYSVIEW_Conf.h

```

#define SEGGER_SYSVIEW_RTT_BUFFER_SIZE 4096u

```

(5)SEGGER_SYSVIEW_Config_uCOSII.c

注释掉

```
#include "os.h"
#include "cpu_core.h"
```

添加

```
#include "os_cfg.h"
```

```
#define SYSVIEW_TIMESTAMP_FREQ ((*(U32 *) (0xE0001004)) >> 4)
#define SYSVIEW_CPU_FREQ      ((*(U32 *) (0xE0001004)) >> 4)

/*~~~~~*/
#define SYSVIEW_RAM_BASE      0x20000000

/*~~~~~*/
修改_cbSendSystemDesc函数

/*~~~~~*/
修改CPU_ERR   local_err;
```

(6)main.c

添加

```
#include "SEGGER_SYSVIEW.h"
#include "os_trace_events.h"
```

在OS初始化后添加

```
OS_TRACE_INIT();
OS_TRACE_START();
```

3. 集成SEGGER SystemView

(1) API介绍

事件	描述	SystemView API
Task Create	创建了新任务	SEGGER_SYSVIEW_OnTaskCreate
Task Start Ready	任务被标记为准备好启动或者恢复执行	SEGGER_SYSVIEW_OnTaskStartReady
Task Start Exec	任务被激活（启动或者恢复执行）	SEGGER_SYSVIEW_OnTaskStart
Task Stop Ready	任务被阻塞或暂停	SEGGER_SYSVIEW_OnTaskStopReady
Task Stop Exec	任务终止	SEGGER_SYSVIEW_OnTaskStopExec
System Idle	没有任务执行，系统进入空闲状态	SEGGER_SYSVIEW_OnIdle

(2) OSTimeDly()

```
/*  
    if(OSTCBCur->OSTCBPrio!=OS_TASK_IDLE_PRIO){  
        SEGGER_SYSVIEW_OnTaskStopReady (OSTCBCur, ticks);  
    }  
*/
```

(3) OS_Sched()

```
if(OSPrioHighRdy==OS_TASK_IDLE_PRIO){  
    SEGGER_SYSVIEW_OnIdle();  
}  
else{  
    SEGGER_SYSVIEW_OnTaskStartExec(OSTCBHighRdy);  
}  
  
if(OSPrioHighRdy==OS_TASK_IDLE_PRIO){  
    SEGGER_SYSVIEW_OnIdle();  
}  
else{  
    SEGGER_SYSVIEW_OnTaskStartExec(OSTCBCur);  
}
```

(4) OSTaskCreate()

```
SEGGER_SYSVIEW_TASKINFO Info;  
    SEGGER_SYSVIEW_OnTaskCreate((unsigned)OSTCBList);  
    //memset(&Info, 0, sizeof(Info));  
    Info.TaskID = OSTCBList;  
    //Info.sName = " ";  
    Info.Prio = prio;  
    Info.StackBase = ptos;  
    SEGGER_SYSVIEW_SendTaskInfo(&Info);
```

(5)ISR

```
SEGGER_SYSVIEW_RecordEnterISR ();  
ISR;  
SEGGER_SYSVIEW_RecordExitISR ();
```

(6)命名

SEGGER_SYSVIEW_Config_uCOSII.c中命名中断

main函数中为任务命名