

# SimCoJLink 使用说明

齐天大圣的哥哥 (淘宝店铺/B 站/知乎 同名)

软件唯一仓库:<https://github.com/jiafan1949/SimCoJLink>

2022 年 2 月 24 日

本文介绍 SimCoJLink 工具的开发背景、使用方法以及注意事项。SimCoJLink 是在 Simulink 环境下的一组 JLink 驱动模块，利用 JLink 调试器盒子快速读写 ARM 系 MCU 的数据，并且不占用任何额外引脚与 CPU 时间，下发参数和上传信号只是将数据拷贝到指定内存位置即可。JLink Pro 传输速度可以高达 1.3MB/s，JLink V9 也最高支持 400KB/s，完全能够胜任电机控制的算法调试需求。配合 Simulink 强大的建模能力，可以方便快速的搭建上位机测控模型，绘制信号波形，下发控制命令。**该工具只提供编译文件，不提供源代码，也不会收费。**

①

**Info:** 按照载波频率 20kHz 上传变量，假设上传 4 个浮点数据，传输的带宽为

$$20e3 * 4 * 4Byte/s = 320e3Byte/s / 1024 = 312.5KB/s$$

## 1 SimCoJLink 介绍

SimCoJLink 库共有三个模块: SimCoJLink SET, SimCoJLink Transmit, SimCoJLink Receive。分别是对 SimCoJLink 进行相关设置，发送数据到 MCU，读取 MCU 上传数据的模块。

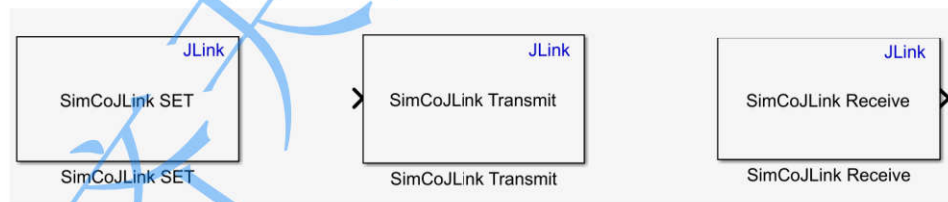


图 1: SimCoJLink 模块库

## 2 SimCoJLink 使用方法

### 2.1 SimCoJLink SET

SimCoJLink SET 总共有五个参数可以设置，JLink Interface, Start Search Address, RTT Search Length, JLink Speed, JLink Target Core。

- **JLink Interface** : JLink 调试接口选择, 根据电路的连接, 选择常用的 SWD 或者 JTAG
- JLink RTT Block Search(自动搜索 MCU 内负责上传和下发数据的指定内存地址)
  - **Start Search Address(HEX)**: 设置 RTT Block 开始搜索的起始地址, 可以直接设置 MCU 的 RAM 起始内存地址, 必须是 0x 开头的 16 进制。
  - **RTT Search Length(HEX)**: 内存搜索的最大长度, 必须是 0X 开头的 16 进制, 最大值是 0X20000, 超过这个最大值设置会报错。
- **JLink Speed(KHz)**: JLink 与 MCU 交互接口的频率, 默认是 50000KHz, 可以不用修改, 如果无法支持这个频率, JLink 驱动会自动降低。最大设置 80000, 必须是正整数。
- **JLink Target Core**: MCU 的核心类型, 根据 MCU 的数据手册选择, 例如 Cortex-M4, Cortex-M3, Cortex-M0 等等。

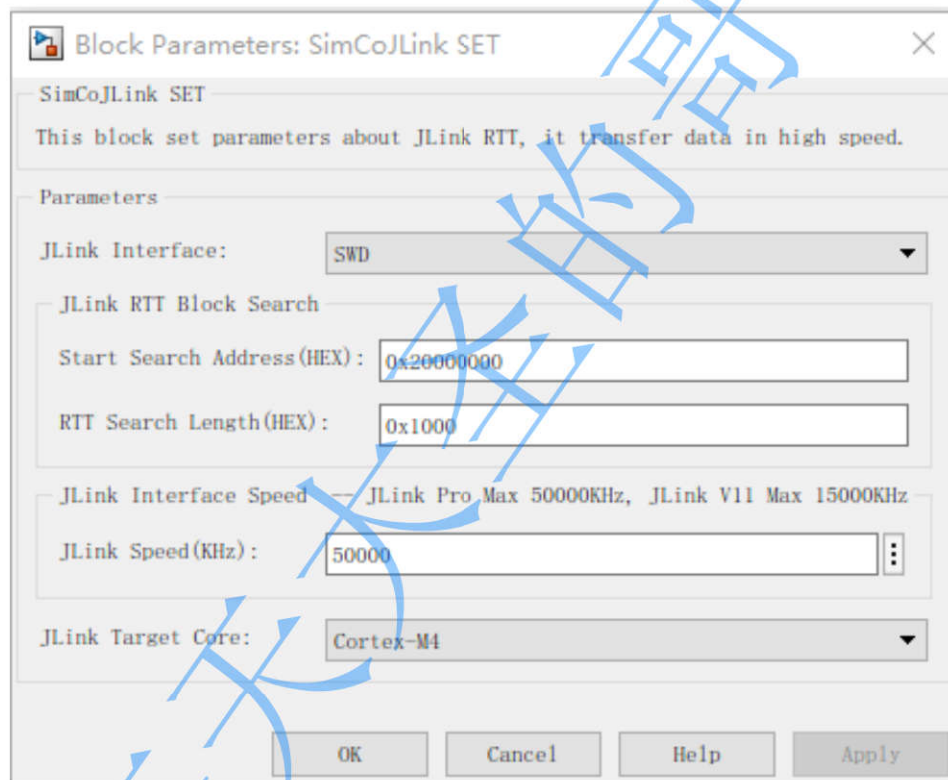


图 2: SimCoJLink SET 参数界面

## 2.2 SimCoJLink Receive

SimCoJLink Receive 是按照设置的采样时间去处理信号, 例如设置 0.1s, 那么每隔 0.1s 将处理刷新一次数据, 这种特殊的信号刷新方式需要更改 Simulink 示波器模块的设置为, **列作为通道 (基于帧)**。在 0.1s 时间接收到的所有数据中, 应该以 "TT:" 三个字符开始, 后面跟上的是所有的数据, 最后再加上帧尾的字符串 "ET!". 具体的 MCU 端发送逻辑可参考提供的示例。

SimCoJLink Receive 总共有五个参数可以设置, Frame Header, Frame Terminator, Data Type, Data Size, Block Sample Time(s)。

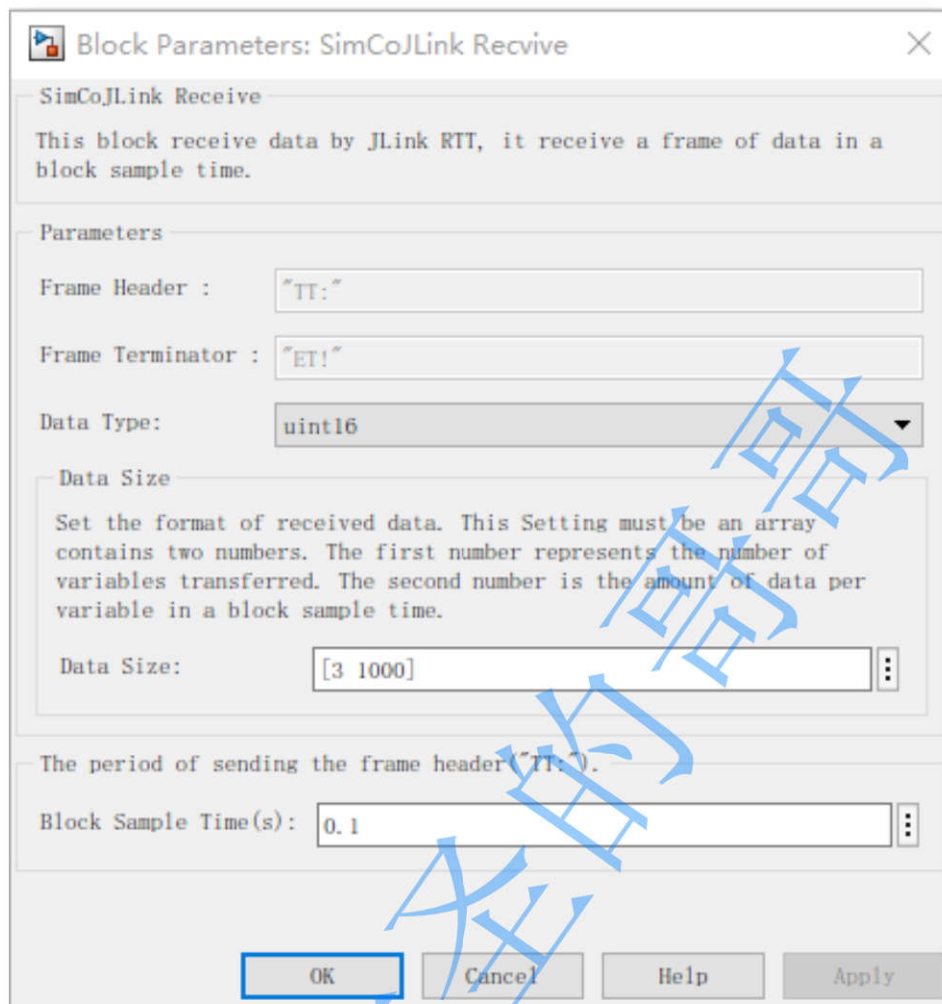


图 3: SimCoJLink Receive 参数界面

- **Frame Header:** 一个采样时间内所有数据称为一帧数据，每帧数据的帧头必须是 "TT:" 这三个字符。此参数无法更改。如果 MCU 发送的帧头不是此字符串，上位机模型将无法正确解析数据。
- **Frame Terminator:** 每帧数据的帧尾必须是 "ET!" 这三个字符，此参数无法更改。如果 MCU 发送的帧尾不是这三个字符，可能收到的数据不是完整的一帧，Simulink 通知栏会发出警告，但是不会报错终止模型运行。
- **Data Type:** MCU 上传变量的数据类型，数据类型一定要设置正确，否则无法正常解析数据。如果 MCU 上传的是 *float* 类型，选择 *single*，在 Simulink 里面，*single* 代表单精度浮点数，和 *float* 同样意义。
- **Data Size:** 一帧数据的数据尺寸，此参数是一个数组，前面的数字代表 MCU 每次发送的变量个数，后面的数字代表此模块采样周期内总共接收的数据次数。如 MCU 发送频率是 10KHz，每次发送 3 个 *uint16* 的数据，则在采样周期 0.1s 时间，收到 1000 次数据，每次数据有三个变量，所以设置为数组 [3 1000]  
前面的数字，变量的个数，最大值限制为 20。



后面的数字，接收变量的次数，最大值限制为 5000。

- **Block Sample Time(s)**: 模块的采样时间，也就是计算周期。此参数必须与 MCU 上传帧格式对应。MCU 发送帧头的周期必须与这里的计算周期相同。默认设置 0.1s，然后计算 MCU 发送的帧方式编码，并设置此模块的 Data Size。

### 2.2.1 MCU 端示例代码

```
1 // 10KHz
2 void UploadParameter(void)
3 {
4     uint16_t TransBuff[10];
5     static uint16_t count;
6     uint8_t TransHeader[] = { 'T', 'T', ':' };
7     uint8_t TransTerminator[] = { 'E', 'T', '!' };
8
9     TransBuff[0] = Phase;
10    TransBuff[1] = Phase1;
11    TransBuff[2] = Phase2;
12
13    if( count == 0)
14    {
15        count++;
16        SEGGER_RTT_WriteNoLock(0, TransHeader, 3);
17        SEGGER_RTT_WriteNoLock(0, (uint8_t *)TransBuff, 3*2);
18    }
19    else if( count == 999)
20    {
21        count = 0;
22        SEGGER_RTT_WriteNoLock(0, (uint8_t *)TransBuff, 3*2);
23        SEGGER_RTT_WriteNoLock(0, TransTerminator, 3);
24    }
25    else
26    {
27        count++;
28        SEGGER_RTT_WriteNoLock(0, (uint8_t *)TransBuff, 3*2);
29    }
30 }
```

## 2.3 SimCoJLink Transmit

SimCoJLink Transmit 是 PC 机下发参数到 MCU 端的模块，下发的参数要求准确，所以下发参数帧末尾会加上 CRC16 校验码，用来校验数据，校验出错可以放弃该帧数据。SimCoJLink Transmit

总共有两个参数，都是默认参数，不可修改。

- **Frame Header:** 此参数默认是 `TT`，每帧下发的参数开头三个字符都是这个帧头。
- **Block Sample Time(s):** 参数默认是 `-1`，该参数设置的模块的执行周期，`-1` 代表的是从前面输入的模块继承采样时间。前面模块的采样时间是多少，该模块采样时间和它相等。

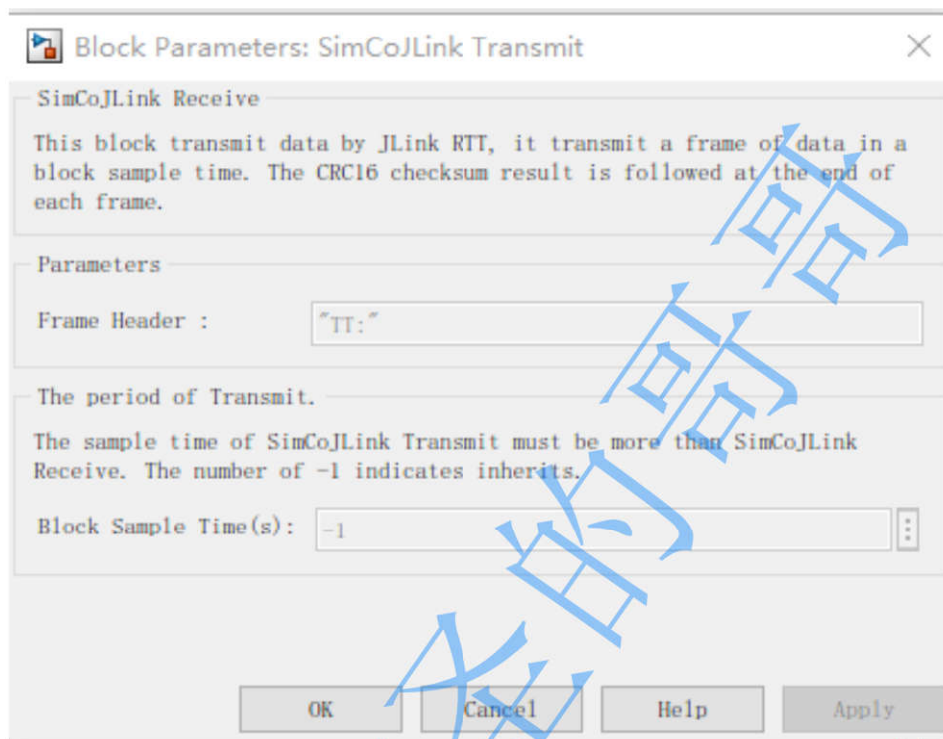


图 4: SimCoJLink Transmit 参数界面



#### Notice:

- 发送模块执行频率不可比接收模块执行频率高！可以相等。
- 发送帧末尾的 **2Byte** 数据是校验值，校验方法和 Modbus RTU 的 CRC16 计算方法相同，MCU 端示例代码有此校验函数代码。

## 3 SimCoJLink 安装与示例

### 3.1 SimCoJLink 安装步骤

1. 将所提供的安装包中的 `JLink_x64.dll` 文件拷贝到 MATLAB 安装目录下的 `\bin\win64` 下，方便运行的时候查找该 `dll` 文件。  
例如在作者电脑上的目录是 `C:\ProgramFiles\Polyspace\2020b\bin\win64`。
2. 将所提供安装包内的 `SimCoJLink_SimulinkLib` 文件夹添加到 MATLAB 的搜索目录。见 (图 5)。
3. 打开 Simulink 的 Library 查看是否成功添加 SimCoJLink 库。见 (图 6)

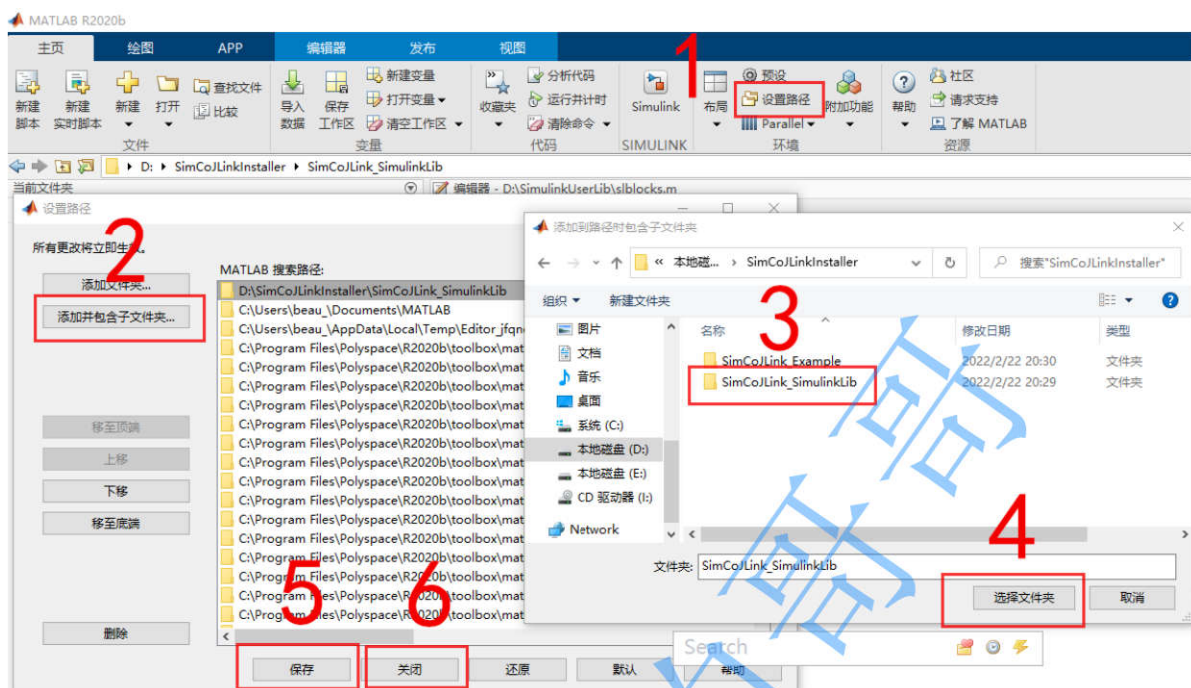


图 5: MATLAB 添加搜索目录步骤

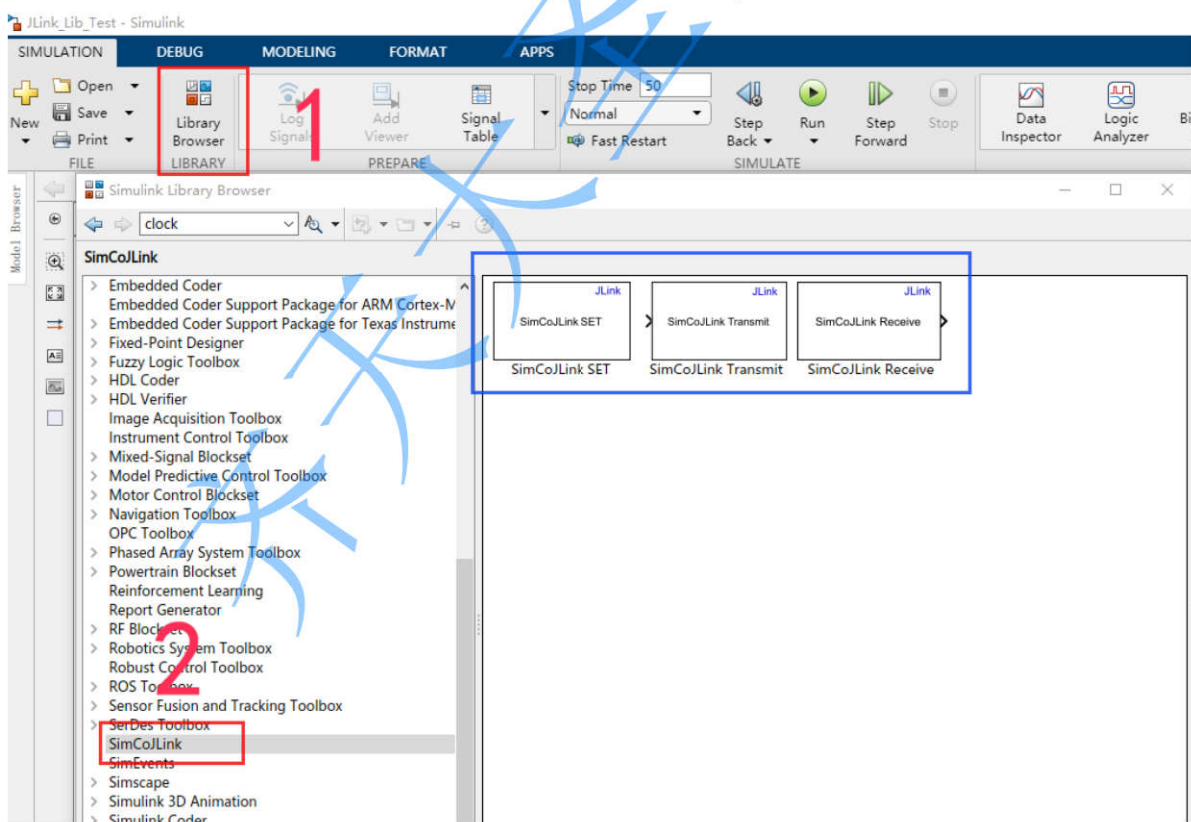


图 6: 查看 Simulink 库



## 3.2 示例说明

下载的安装包子目录 SimCoJLink\_Example 下，存在两个测试工程实例。一个主控芯片是 STM32G474RE 的 ST 官方开发板 NUCLEO-G474RE，内核为 Cortex-M4。另外一个主控芯片是 STM32G071RB 的官方开发板 NUCLEO-G071RB，内核为 Cortex-M0。

测试流程：打开 Keil 工程，下载代码，RUN 让程序跑起来。打开上位机 Simulink 模型，开始监控数据。



### Notice:

这两块开发板自带 STLINK，需要把 STLINK 的 SWD 线与主控芯片断开，或者将 STLINK 电源断开，保证 STLINK 不会干扰 SWD 的调试线与 JLINK 连接。具体查看 ST 官网型号原理图。

作者实际测试电路板硬件修改情况：

- NUCLEO-G474RE：去除 R9，R10，R13，R18。JP5 选 5V\_PWR，JLink 连接 CN4。USB 端口供电。
- NUCLEO-G071RB：去除 R24 电阻，JP2 选择 CHG，USB 插上供电电源，只负责供电。JLink 按照线序连接 CN11。

## 4 MCU 端编码概述

### 4.1 MCU 端编码流程

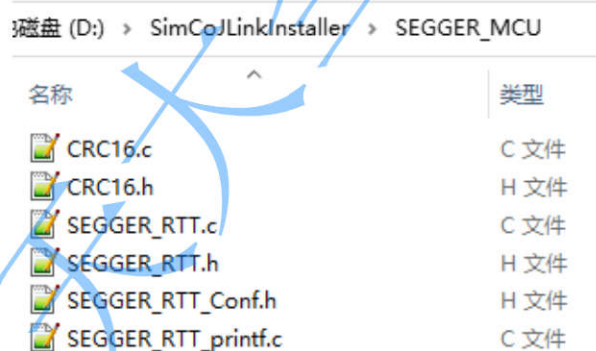


图 7: MCU 端 API 函数文件

1. 将作者提供包中 SGEEER\_MCU 目录的文件, 见图 (7), 拷入工程, 并添加头文件搜索目录,。

```
94 #ifndef BUFFER_SIZE_UP
95 #define BUFFER_SIZE_UP (1024)
96 #endif
97
98 #ifndef BUFFER_SIZE_DOWN
99 #define BUFFER_SIZE_DOWN (128)
100 #endif
```

图 8: RAM 大小设置

2. 设置用来传输信号的 RAM 大小。在 `SEGGER_RTT_Conf.h` 文件 95 行与 99 行。该内存大小是 MCU 用来暂存发送和接收数据的内存区域，理论上是越大越好。因 MCU 发送到 PC 速度较快，占用内存相对设置大点，默认可以设置 1024Byte。PC 发送到 MCU，速度较慢，可以设置相对小，设置三倍或者五倍实际每帧数据大小即可，默认设置 128Byte。见 (图 8)

```
SEGGER_RTT_Init();
SEGGER_RTT_SetTerminal(0);
```

图 9: 初始化函数

3. 初始化相关的结构体与内存等。在开始发送和接收数据之前初始化。调用图 (9) 中两个函数。

```
1 // 10KHz MCU 发送到 PC
2 void UploadParameter(void)
3 {
4     uint16_t TransBuff[10];
5     static uint16_t count;
6     uint8_t TransHeader[] = { 'T', 'T', ':' };
7     uint8_t TransTerminator[] = { 'E', 'T', '!' };
8
9     TransBuff[0] = Phase;
10    TransBuff[1] = Phase1;
11    TransBuff[2] = Phase2;
12
13    if( count == 0)
14    {
15        count++;
16        SEGGER_RTT_WriteNoLock(0, TransHeader, 3);
17        SEGGER_RTT_WriteNoLock(0, (uint8_t *)TransBuff, 3*2);
18    }
19    else if( count == 999)
20    {
21        count = 0;
22        SEGGER_RTT_WriteNoLock(0, (uint8_t *)TransBuff, 3*2);
23        SEGGER_RTT_WriteNoLock(0, TransTerminator, 3);
24    }
25    else
26    {
27        count++;
28        SEGGER_RTT_WriteNoLock(0, (uint8_t *)TransBuff, 3*2);
29    }
30 }
```



```

1 // while 循环中执行或其它低频任务中执行，MCU接收
2 // 读取已接收到数据字节数，大于一帧总字节数，开始处理
3 if(SEGGER_RTT_GetAvailReadSpace(0) > 5*2+5-1)
4 {
5     uint8_t RTT_RecvBuffer[15];
6
7     // 从RTT内存读取数据15字节数据到RTT_Buffer内
8     SEGGER_RTT_ReadNoLock(0, RTT_RecvBuffer, 15);
9
10    // 计算CRC16校验码，只对实际数据校验，不包含帧头
11    uint16_t CRC16_Result = usMBCRC16(RTT_RecvBuffer+3, 10);
12    uint16_t RecvResult = (RTT_RecvBuffer[13] << 8) +
13                        RTT_RecvBuffer[14];
14
15    if(CRC16_Result == RecvResult)
16    {
17        // 校验通过，将5个uint16数据拷贝到memTemp数组中
18        memcpy(memTemp, RTT_RecvBuffer+3, 10);
19    }
20 }

```

4. MCU 按照固定频率发送数据，并检查接收数据缓存大小接收数据。如示例 STM32G071 中，发送接收函数见上图代码块。MCU 发送到 PC 按照 10KHz 频率，每次发送 3 个 uint16 变量类型，每 1000 次为一帧。MCU 接收 PC 数据，每 0.1s 一帧，每帧数据共有 15 字节，帧头 "TT:" 3 个字节，5 个 uint16 是 10 个字节，帧尾是 CRC16 校验码 2 个字节。

## 4.2 MCU 端常用 API 函数

- SEGGER\_RTT\_Init

```

1 /*****
2 * 函数描述：
3 * 对RTT相关结构体与内存初始化。
4 *
5 *****/
6 void SEGGER_RTT_Init (void)

```

- SEGGER\_RTT\_SetTerminal

```

1 /*****
2 * 函数描述：
3 * 对RTT的终端进行设置。
4 *
5 * 输入参数

```

```

6  *   TerminalId   : 终端号，SimCoJLink 必须选择是0。
7  *
8  *   返回值
9  *   >=0，成功设置。
10 *   <0 ， 设置失败。
11 *****/
12 int SEGGER_RTT_SetTerminal (unsigned char TerminalId)

```

#### • SEGGER\_RTT\_WriteNoLock

```

1  /*****
2  *   函数描述：
3  *   将指定字节数目的内存数据尝试拷贝到RTT上传内存区域，
4  *   等待PC机读取到电脑端。如果拷贝不成功，不会死锁等待，
5  *   正常退出函数。
6  *
7  *   输入参数
8  *   BufferIndex   : 对于SimCoJLink工具，该参数是0。
9  *   pBuffer      : 待拷贝内存的指针。
10 *   NumBytes     : 待拷贝内存的字节数目。
11 *
12 *   返回值
13 *   成功写入的字节数目。
14 *
15 *   注意点
16 *   执行该函数之前，必须已调用过初始化函数，
17 *   在本说明书4.1节已说明。
18 *****/
19 unsigned SEGGER_RTT_WriteNoLock(unsigned BufferIndex,
20                                const void* pBuffer, unsigned NumBytes)

```

#### • SEGGER\_RTT\_ReadNoLock

```

1  /*****
2  *   函数描述：
3  *   从PC写入到MCU的内存区域读取指定字节数目的数据。
4  *   如果读取不成功，不会死锁等待，正常退出函数。
5  *
6  *   输入参数
7  *   BufferIndex   : 对于SimCoJLink工具，该参数是0。
8  *   pData        : 待写入的内存指针。
9  *   BufferSizde   : 读取的字节数目。
10 *
11 *   返回值

```

```

12 *    成功读取的字节数目。
13 *
14 *    注意点
15 *    执行该函数之前，必须已调用过初始化函数，
16 *    在本说明书4.1节已说明。
17 *****/
18 unsigned SEGGER_RTT_ReadNoLock(unsigned BufferIndex,
19                                void* pData, unsigned BufferSize)

```

#### • usMBCRC16

```

1  /*****
2  *    函数描述：
3  *    计算CRC16校验码
4  *
5  *    输入参数
6  *    pucFrame    : 待校验内存的指针。
7  *    usLen        : 待校验内存的字节数目。
8  *
9  *    返回值
10 *    CRC16校验码。
11 *****/
12 unsigned short usMBCRC16(unsigned char * pucFrame,
13                          unsigned short usLen)

```

#### • SEGGER\_RTT\_GetAvailReadSpace

```

1  /*****
2  *    函数描述：
3  *    获取PC到MCU端的RAM内存，已经写入的字节数目。
4  *
5  *    输入参数
6  *    BufferIndex   : 对于SimCoJLink工具，该参数是0。
7  *
8  *    返回值
9  *    已经写入的字节数目。
10 *****/
11 unsigned SEGGER_RTT_GetAvailReadSpace (unsigned BufferIndex)

```

#### • SEGGER\_RTT\_GetAvailWriteSpace

```

1  /*****
2  *    函数描述：
3  *    获取MCU到PC端的RAM内存，剩余可写入的字节数目。

```



```

4 *
5 * 输入参数
6 *   BufferIndex   : 对于 SimCoJLink 工具, 该参数是 0。
7 *
8 * 返回值
9 *   剩余可写入的字节数目。
10 *
11 * 注意点
12 *   返回值是 1 时, 说明可写入的字节为 1, 已经无法继续写入新数据,
13 *   解决方法   1. 更换速度更高的 JLink
14 *               2. 扩展 RAM 内存配置, 见本说明书 4.1
15 *               3. 降低发送频率和字节数目
16 *****/
17 unsigned SEGGER_RTT_GetAvailWriteSpace (unsigned BufferIndex)

```

## 5 广告

如果对《Simulink 代码生成与电机控制》感兴趣, 可以手机淘宝扫描下方二维码进入店铺查看,  
 店铺名称: 齐天大圣的哥哥。

