

RI 指令操作指南

V1.0.0

北京联盛德微电子有限责任公司 (winner micro)

地址：北京市海淀区上园村 3 号交大知行大厦七层

电话：+86-10-62161900

公司网址：www.winnermicro.com

文档历史

版本	完成日期	修订记录	作者	审核	批准
V1.0.0	2015-7-15	创建	章鹏飞		

目录

1	概况	4
2	SPI 方案特性	4
3	RI 指令加网	4
4	RI 指令分析	6
5	通过 PC 工具对 RI 指令进行测试	10
5.1	硬件环境	10
5.2	软件环境	11
5.3	固件修改	11
5.4	参数修改	11
5.5	测试工具	12
5.5.1	查询中断状态	12
5.5.2	获取 SPI 有效数据长度	13
5.5.3	获取 SPI 上传数据	13
6	基于 PC 工具完成 RI 指令联网	14
6.1	设置无线网卡的网络类型	14
6.2	设置无线网卡加入的网络名称	14
6.3	设置无线网卡加入的网络密钥	14
6.4	启用 DHCP	15
6.5	保存参数到 SPI Flash	15
6.6	无线网卡加网	16
7	基于 RI 指令的 Socket 通讯	16
7.1	创建 Socket 连接	16
7.2	无线网卡发送数据	17
7.3	无线网卡接收数据	17
7.4	关闭 Socket 连接	18
8	STM32 的 Keil 工程实现 SPI Master 通信	18

1 概况

北京联盛德微电子有限责任公司(winnermicro)的 WIFI 模块 TLN13SP01 支持 UART, SPI, SDIO 三种接口。对于 UART 接口, winnermicro 的 SDK 开发 AT 指令。但受限于 UART 的速度。而用户如需要使用更快的接口速度, 譬如传音频、视频的数据时候, 则需要更换接口 SPI 或 SDIO 接口。对于 SPI、SDIO 接口, winnermicro 则用另外一套 RI 指令(又称精简指令)来控制模块, 完成加网, 通信等基本功能。

2 SPI 方案特性

要求: 传音频、视频大数据

接口: SPI 或 SDIO 接口

硬件设置: MCU (SPI_Master) + WIFI 模块 (SPI_Slave)

3 RI 指令加网

步骤如下:

1、WPRT(0x40) //设置工作模式为 STA:

MasterSendData (==SlaveReceiveData) :

aa ,01, 00, 05, 02, 00, 00, 08, 01, 40, 00, 01, 00, 00, 00

//黄色背景的 00, 则是对 40 设置的数值为 0, 即 WPRT 的值为 0, 类似 AT 指令: AT+WPRT=0

MasterReceiveData:

[0,aa] [1,1][2,0][3,4][4,1][5,0][6,0][7,0][8,2][9,40][10,0][11,0][12,2][13,40][14,0][15,0]

//返回数据中, 其中 40 前的 2, 表示返回正确, 即 AT 指令返回 OK 类似。

2、SSID(0x41) //设置需要加入的 AP 名称如 winnermicro:

MasterSendData :

aa, 01, 00, 10, 03, 00, 00, 14, 01, 41, 00, 01, 0b, 77, 69, 6e, 6e, 65, 72, 6d, 69, 63, 72, 6f,

//41 的数值为 77, 69, 6e, 6e, 65, 72, 6d, 69, 63, 72, 6f。这 11 个 HEX 字符为 winnermicro 的 ASCII 码, 即设置 SSID 为 winnermicro, 类似 AT 指令 AT+SSID="winnermicro"(大小写区分)

MasterReceiveData:

[0,aa][1,1][2,0][3,4][4,2][5,0][6,0][7,0][8,2][9,41][10,0][11,0][12,2][13,41][14,0][15,0]

3、KEY(0x42) //设置需要加入的 AP 的无线密钥 winnermicro.com

MasterSendData:

aa, 01, 00, 16, 04, 00, 00, 1b, 01, 42, 00, 01, 01, 00, 0f, 77, 69, 6e, 6e, 65, 72, 6d, 69, 63, 72, 6f, 2e, 63, 6f, 6d, 00, 00,

[0,aa][1,1][2,0][3,4][4,3][5,0][6,0][7,0][8,2][9,42][10,0][11,0][12,2][13,42][14,0][15,0]

[illegible]

[0,aa][1,1][2,0][3,4][4,4][5,0][6,0][7,0][8,2][9,60][10,0][11,0][12,2][13,60][14,0][15,0]

aa, 01, 00, 04, 06, 00, 00, 0b, 01, 04, 00, 00,

[0,aa][1,1][2,0][3,4][4,5][5,0][6,0][7,0][8,2][9,4][10,0][11,0][12,2][13,4][14,0][15,0]

aa, 01, 00, 04, 07, 00, 00, 0c, 01, 01, 00, 00,

[0,aa][1,1][2,0][3,4][4,6][5,0][6,0][7,0][8,2][9,1][10,0][11,0][12,2][13,1][14,0][15,0]

ASCII	HEX	功能
INITED	0xE0	启动完成，满足下列条件之一时发出： ♡ 软件复位，初始化完成后

[0,aa][1,1][2,0][3,4][4,0][5,0][6,0][7,0][8,0][9,e0][10,0][11,0][12,0][13,e0][14,0][15,0]

aa, 01, 00, 04, 08, 00, 00, 0d, 01, 20, 00, 00,

[0,aa][1,1][2,0][3,4][4,1][5,0][6,0][7,0][8,2][9,20][10,0][11,0][12,2][13,20][14,0][15,0]

WJNRES	0xE3	无线网络联网结果，满足下列条件之一时发出： Infra 网络，本端为 STA，加入 AP 成功 Infra 网络，本端为 STA，加入 AP 失败
--------	------	---

MasterReceiveData

[0,aa][1,1][2,0][3,1b][4,2][5,0][6,0][7,0][8,0][9,e3][10,0][11,1][12,1][13,14][14,75][15,90][16,2c][17,9e][18,a6][19,0][20,1][21,4][22,b][23,77][24,69][25,6e][26,6e][27,65][28,72][29,6d][30,69][31,63][32,72][33,6f][34,19][35,c5]

LINKUP	0xE6	网络层已就绪，满足下列条件之一时发出： ✧ 设置完成静态 IP 地址 ✧ Infra 网络，本端为 STA，DHCP 成功获取并设置 IP 地址
--------	------	--

MasterReceiveData

[0,aa][1,1][2,0][3,4][4,3][5,0][6,0][7,0][8,0][9,e6][10,0][11,0][12,0][13,e6][14,0][15,0]

4 RI 指令分析

本文档主要请参考《TLN13SP01 用户使用手册.pdf》。主要通过 STA 加网具体分析 RI 指令的结构。

实例分析：

WPRT (0x40) //设置工作模式为 STA:

MasterSendData (==SlaveReceiveData) :

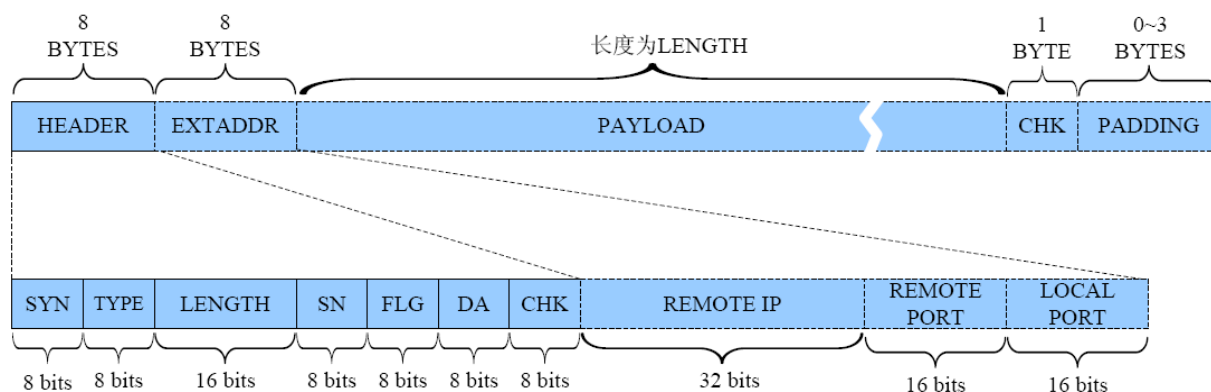
aa ,01, 00, 05, 02, 00, 00, 08, 01, 40, 00, 01, 00, 00, 00, 00

MasterReceiveData:

[0,aa] [1,1][2,0][3,4][4,1][5,0][6,0][7,0][8,2][9,40][10,0][11,0][12,2][13,40][14,0][15,0]

从上述指令中可以找到，WPRT 对应精简数据为 0x40。

现在参照 winnermicro 的 SP01 模块的 RI 指令的数据格式



第一个数据是 SYN=0xAA

第二个数据是 TYPE=0x00 或 0x01

备注：因为本教程是 SPI 接口，使用 RI 指令控制 WIFI，所以只能是 0x00 或者 0x01。

值	含义
0x00	用户数据（注：在 H-SPI 模式下，需要使用 SPI 接口的数据端口传输）
0x01	RI 指令数据（注：在 H-SPI 模式下，需要使用 SPI 接口的命令端口传输）
0x02	AT+指令数据（注：在 H-SPI 模式下，需要使用 SPI 接口的命令端口传输）

第三个数据是 LENGTH（高 8 位）

第四个数据是 LENGTH（低 8 位）

备注：LENGTH 负载数据长度（PAYLOAD 域，不含 PADDING），字节序为大端（注：后续所有长度在 16bit 以上字段的字节序均为大端）

第五个数据是 SN（帧序号）

第六个数据是 FLG（标识域）（一般情况下，默认为 0）

	值	含义
FLG_PAYLOAD_CHK	Bit0	PAYLOAD 域是否包含校验字段
	Bit1-7	保留

第七个数据是 DA：

DA：用户数据帧地址，（注：仅适用于类型为用户数据帧，其它类型帧此字段填0）

	值	含义
DA_TCPSOCKET	Bit0~5	TCP 连接 Socket 号（仅当 DA_TYPE=0 时）
DA_TYPE	Bit6~7	0x0: TCP 连接 0x1: UDP 连接 0x2: WEB 服务器（XML-RPC） 0x3: 保留

第八个数据：CHK

CHK：校验和，帧头校验和的计算范围包括 TYPE~DA 之间所有字段，帧体校验和的计算范围为 PAYLOAD 域。校验和为 Byte 累加和。

备注：以上 8 个数据的位置和意义都是固定的，如不存在，则填充 0，占据此位置。

第 9 个数据（可选）：EXTADDR（负载数据地址信息）

说明:负载数据地址信息,为可选字段,仅当 UDP 用户数据帧(TYPE=0,DA_TYPE=1) 时有效,其中:

值	含义
REMOTE IP	对端 IP 地址
REMOTE PORT	对端端口号
LOCAL PORT	本端端口号

第 10 个数据: PAYLOAD:

负载数据,根据 TYPE 字段设置,可以为用户数据、RI 指令协议数据或 AT+指令协议数据。

说明:此字节和第 2 个数据进行对应,如果是 TYPE=0X00,则此处数据是用户数据,用户数据可以是多字节;如果 TYPE=0X01,此处为 RI 指令命令,一般为单字节(可对照 RI 指令列表,暂无双字节指令);如果是 TYPE=0X02 则是 AT 指令(16 进制的表示)。根据《TLN13SP01 用户使用手册》描述 WPRT 指令,见下截图:

格式 (HEX):



值	含义
0	infra 网络 (模块作为 STA)
1	adhoc 网络
2	infra 网络 (模块作为 AP)

由于模块此时作为 STA,则 type==00,从而得到 PAYLOAD 数据为: 01 40 00 XX 00。

第 11 个数据是 CHK: 帧体校验和的计算范围为 PAYLOAD 域。校验和为 Byte 累加和。

第 4*X 个数据 PADDING: 填充字段,保证整个帧的长度数据长度为 4 的整数倍。即如果是 15 个数据,则添加 1 个为 16;如果除掉 PADDING 之后的数据 30 个,则添加 2 个,为 32。所以数据段,至少是 12 个,以及以上。

至此我们可以组合 WPRTRI 指令的数据格式:

MasterSendData (==SlaveReceiveData)

1	2	3(H8)	4(L8)	5	6	7	8	9	10	11	12	13	14	15	16
SYN	TYPE	LENGTH	SN	FLG	DA	CHK	EXTADDR	PAYLOAD	CHK	PADDING					
AA	01 02	01~0200		0		Add(2~7)		01	40	00	xx	00	xx	xx	

于是 MasterSendData (==SlaveReceiveData):

aa,01,00,05,02,00,00,08,01,40,00,01,00,00,00,00

MasterReceiveData

aa,01,00,04,01,00,00,00,02,40,00,00,02,40,00,00

说明：从上述说明可知，主机虽然按照标准给从机发送了 RI 指令，但是从机并未完全按照上述规则给主机返回字符。特别是 CHK，则没有进行校验。

SSID 指令解析：

SSID (0x41) //设置需要加入的 AP 名称如 winnermicro

MasterSendData :

aa,01,00,10,03,00,00,14,01,41,00,01,0b,77,69,6e,6e,65,72,6d,69,63,72,6f,

SYN=0xAA //同步字符

TYPE=0x01 //代表命令

LENGTH = 0x0010 //数据长度

SN = 0x03

FLG = 0X00

DA = 0x00

CHK = 0x01+0x10+0x03=0x14

TYPE =0x01

SSID =0x41 //此时就要参考《TLN13SP01 用户使用手册》对 SSID 这条指令的描述了。

4.1.4.23 SSID(0x41)

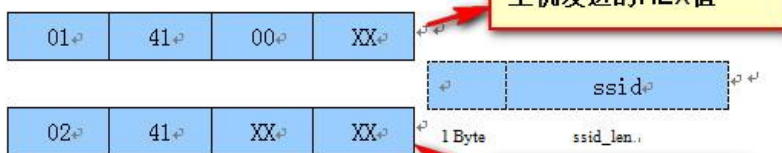
功能：

设置/查询无线网络名称，即 ssid。

格式 (ASCII)：

```
AT+SSID=[!?][ssid]<CR>
+OK[=ssid]<CR><LF><CR><LF>
```

格式 (HEX)：



参数：

ssid_len: 网络名称长度（仅 HEX 格式命令）

ssid: 无线网络名称，1~32 个字符

SSID_后面第一个 BYTE =0X00

SSID_后面第二个 BYTE =0xXX

SSID_后面第三个 BYTE =0X0b // 代表了 SSID 之后的数据长度，即

strlen(“wintermicro”)

SSID_后面第四个 BYTE 开始 = “wintermicro”

最后两个 0x00 则是将命令补充为 4 的倍数。

此条指令概括一下便是：41 的数值为 77, 69, 6e, 6e, 65, 72, 6d, 69, 63, 72, 6f。

这 11 个 HEX 字符为 wintermicro 的 ASCII 码，即设置 SSID 为 wintermicro，类似 AT 指令 AT+SSID=“wintermicro”。注意：大小写区分，同时复制时主要中英文标点号，测试时，最好在英文输入法下手打字符。

至于从机，即模块返回的数据，前面的数据类似，即从 PAYLOAD 开始，再次参考《TLN13SP01 用户使用手册》对 SSID 这条指令的描述：

4.1.4.23 SSID(0x41)

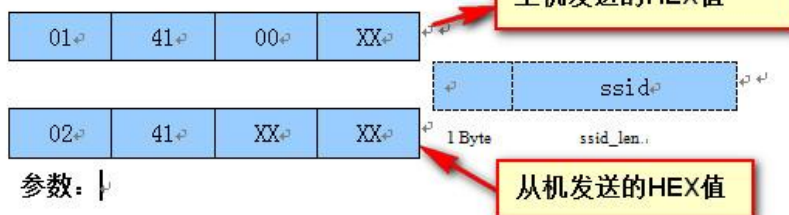
功能：

设置/查询无线网络名称，即 ssid。

格式 (ASCII)：

```
AT+SSID=[!~?][ssid]<CR>
+OK[=ssid]<CR><LF><CR><LF>
```

格式 (HEX)：



参数：

ssid_len: 网络名称长度（仅 HEX 格式命令）

ssid: 无线网络名称，1~32 个字符

MasterReceiveData:

[0,aa][1,1][2,0][3,4][4,2][5,0][6,0][7,0][8,2][9,41][10,0][11,0][12,2][13,41][14,0][15,0]

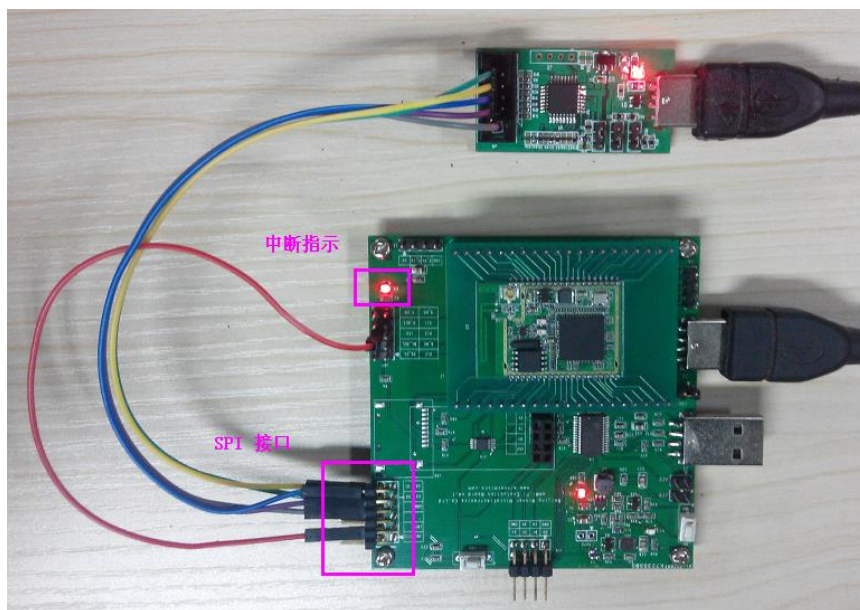
5 通过 PC 工具对 RI 指令进行测试

5.1 硬件环境

Wintermicro 评估板一套。

USB 转 SPI 接口板一块，可参考下面链接地址 demo 板。

http://item.taobao.com/item.htm?spm=a1z09.2.9.161.UUABpc&id=17559606098&_u=9lu_hjf7598c



如图所示连接评估板和 USB 转 SPI 接口板 SPI 接口对应的数据线。

备注：

1、USB 转 SPI 接口板没有中断脚，评估板中断脚接评估板 LED 指示灯，用于指示 SPI 的中断状态（即红色线所接的线）。

2、评估板和 USB 转 SPI 接口板 3.3V 电源不需要连接，共地就可以。

5.2 软件环境

联盛德 WIFI 模组默认为 UART 接口，使用 SPI 接口需做对应的固件修改。

WMSDK 开发包、EM-Wifi 工具可从我司官网获取：www.winnermicro.com

5.3 固件修改

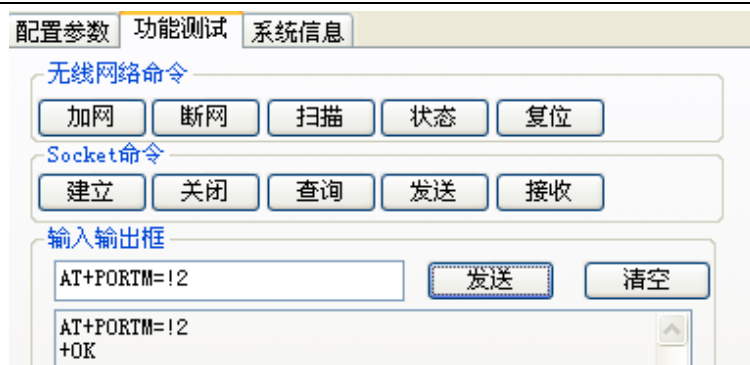
修改 WMSDK 开发包中 `wm_config.h` 中如下的宏定义：

```
#define TLS_CONFIG_HS_SPI      CFG_ON /*High Speed SPI*/  
#define TLS_CONFIG_RI_CMD     (CFG_ON && TLS_CONFIG_HOSTIF)
```

编译固件后更新到待测试的 TLN13SP01 模块。

5.4 参数修改

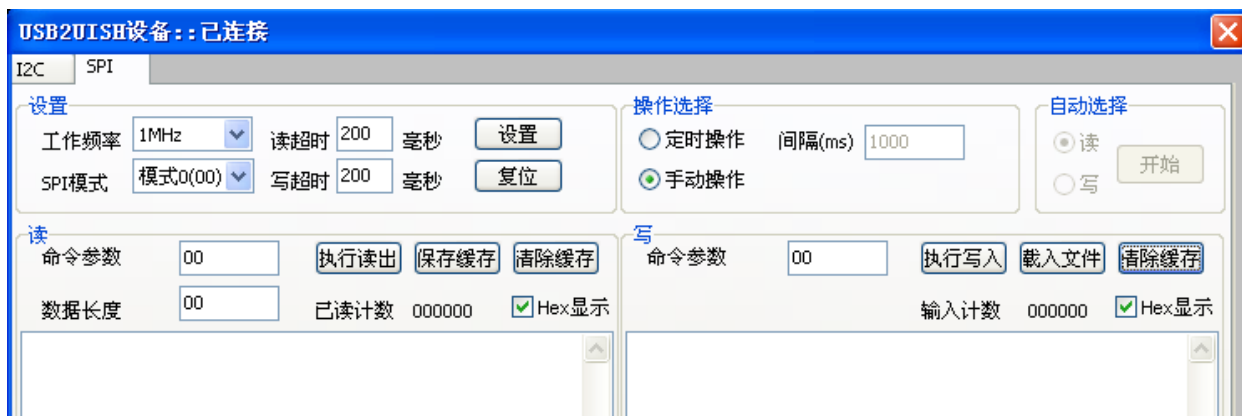
使用 EM-WIFI 工具通过 Uart0 搜索模块，通过如图 7-2 命令，配置 TLN13SP01 模块为 SPI 接口方式后，重启模块后，模块工作在 SPI 接口方式。



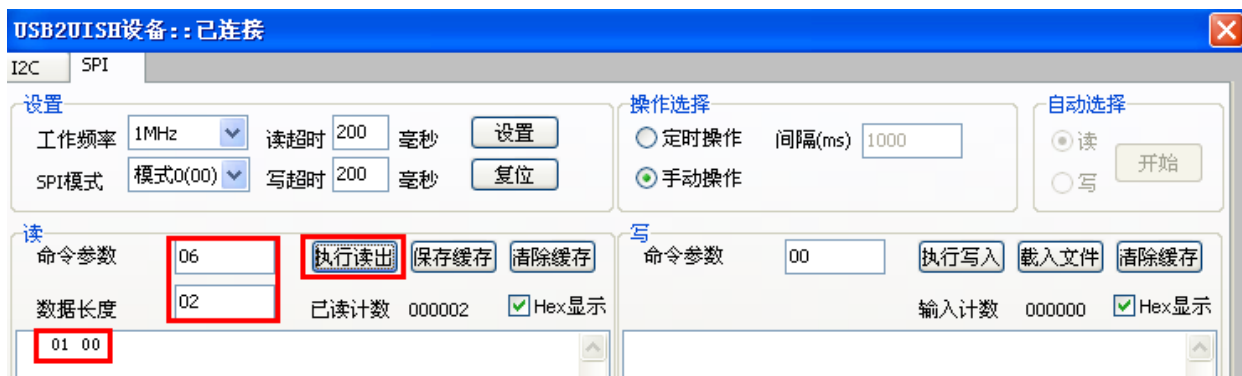
图设置为 SPI 接口方式

5.5 测试工具

本测试使用 USB 转 SPI 接口板自带的 PC 端 USB2UI5H 工具测试，如图所示。



5.5.1 查询中断状态



查询中断状态

如图所示，从 06 寄存器读取 2Byte 数据，查询当前是否有 SPI 中断触发，返回值 01，说明当前有上行数据。

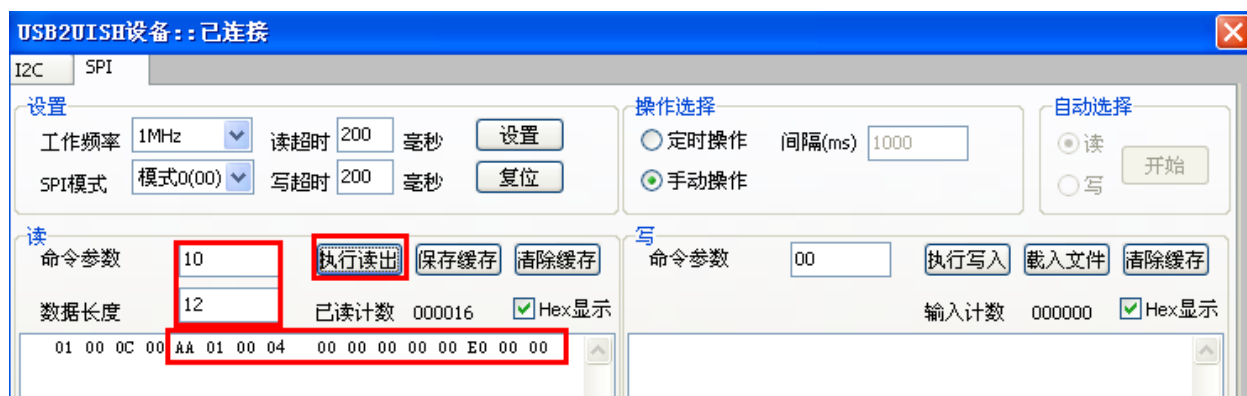
5.5.2 获取 SPI 有效数据长度



图查询有效数据长度

如图所示，从 02 寄存器读取 2byte 数据，获得当前下行到 SPI host 设备的数据长度，返回值为 0C，说明有 12Byte 的上行数据。

5.5.3 获取 SPI 上传数据



获取 SPI 上传数据

如图所示，通过 10 命令一次性从下位机读取 12Byte 数据到 SPI host 设备，解析上行数据，发现网卡返回了一个 E0 给 SPI host 设备，查询 TLN13SP01 用户手册，E0 表示系统启动完成。

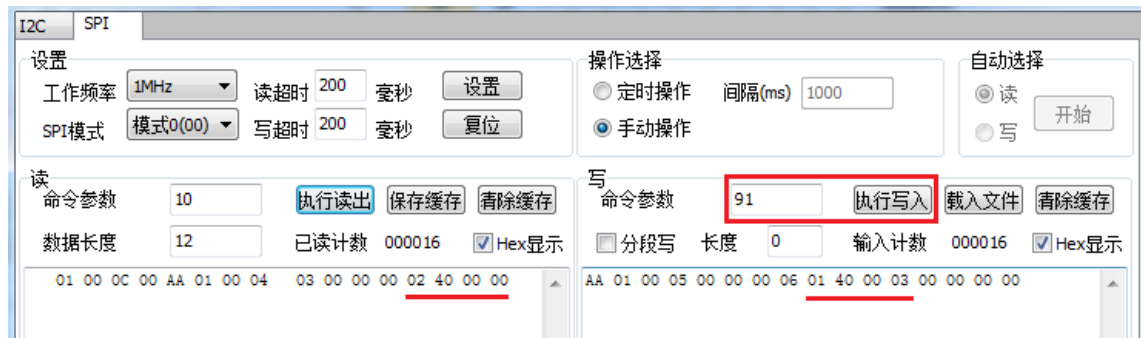
事件代码：

本系统支持的事件定义如下（注：事件仅适用于 RI 指令协议）：

ASCII	HEX	功能
INITED	0xE0	启动完成，满足下列条件之一时发出： ✧ 软件复位，初始化完成后

6 基于 PC 工具完成 RI 指令联网

6.1 设置无线网卡的网络类型



设置网卡的网络类型

如图所示，根据 SPI 指令协议将网卡设置为 STA 模式，依次读取 06、02、10 寄存器获取返回结果。

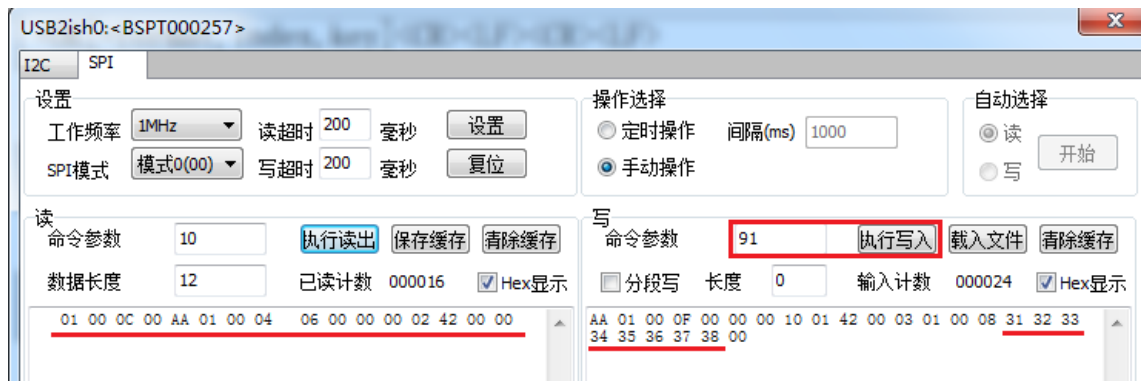
6.2 设置无线网卡加入的网络名称



设置网络名称

如图所示，根据 SPI 指令协议设置无线网卡加入的网络名称，依次读取 06、02、10 寄存器获取返回结果。

6.3 设置无线网卡加入的网络密钥



设置网络密钥

如图所示，根据 SPI 指令协议设置无线网卡加入的网络密钥，并依次读取 06、02、10 寄存器获取返回结果。

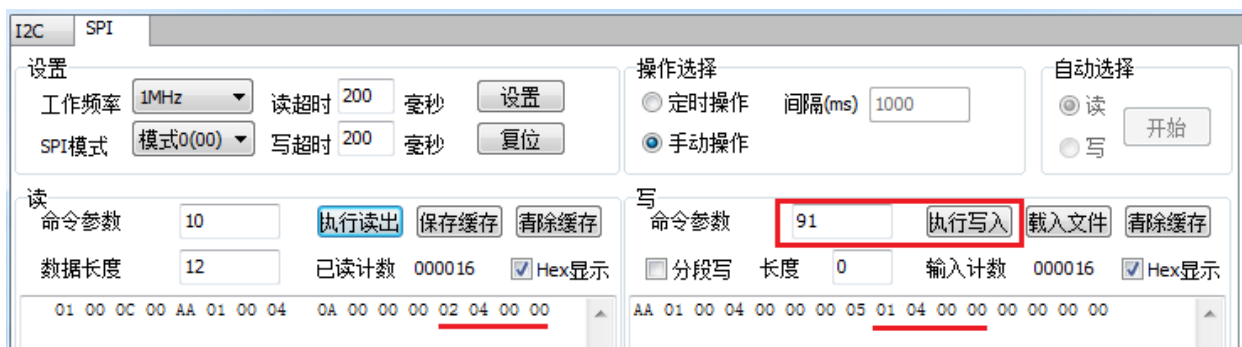
6.4 启用 DHCP



启用 DHCP

如图所示，根据 SPI 指令协议启用 DHCP，并依次读取 06、02、10 寄存器获取返回结果。

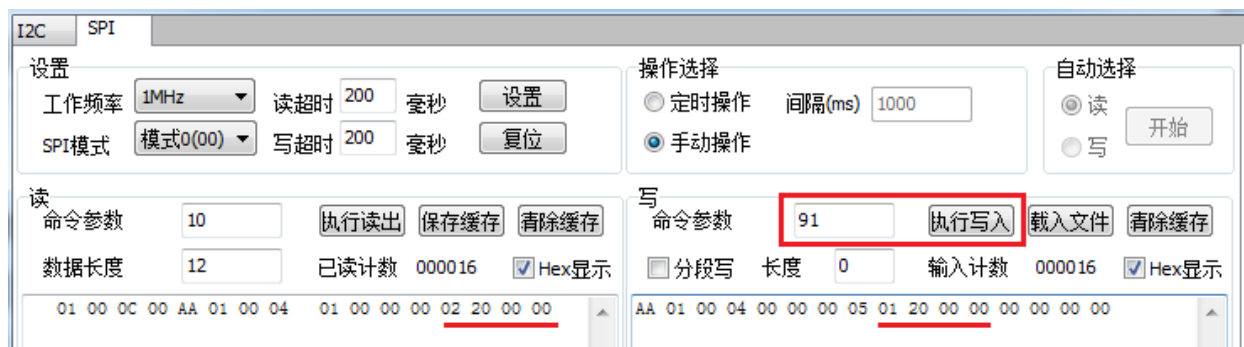
6.5 保存参数到 SPI Flash



更新参数到 Flash

如图所示，根据 SPI 指令协议将参数更新到 SPI Flash 中，并依次读取 06、02、10 寄存器获取返回结果。

6.6 无线网卡加网



无线网卡加网

如图所示，根据 SPI 指令协议将无线网卡加网，并依次读取 06、02、10 寄存器获取返回结果。

7 基于 RI 指令的 Socket 通讯

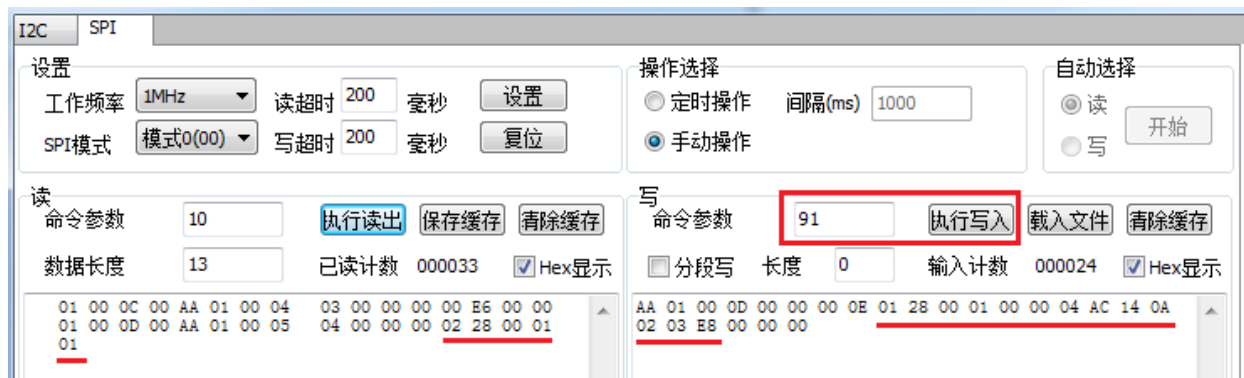
7.1 创建 Socket 连接

1、用 TCP 助手创建一个 PC 端的 TCP Server，TCP Server 的地址为 172.20.10.2，监听端口为 1000，如图 7-14：



创建 TCP Server

2、如图所示，根据 SPI 指令协议创建 Socket，依次读取 06、02、10 寄存器获取返回结果，返回创建成功的 Socket 为 1。



创建 Socket

7.2 无线网卡发送数据

如图所示，通过创建的 Socket 号发送数据 31 32 33 34，TCP Server 端收到对应的数据 1234。

备注：TCP Server 显示的是字符，无线网卡端为 Hex 格式。

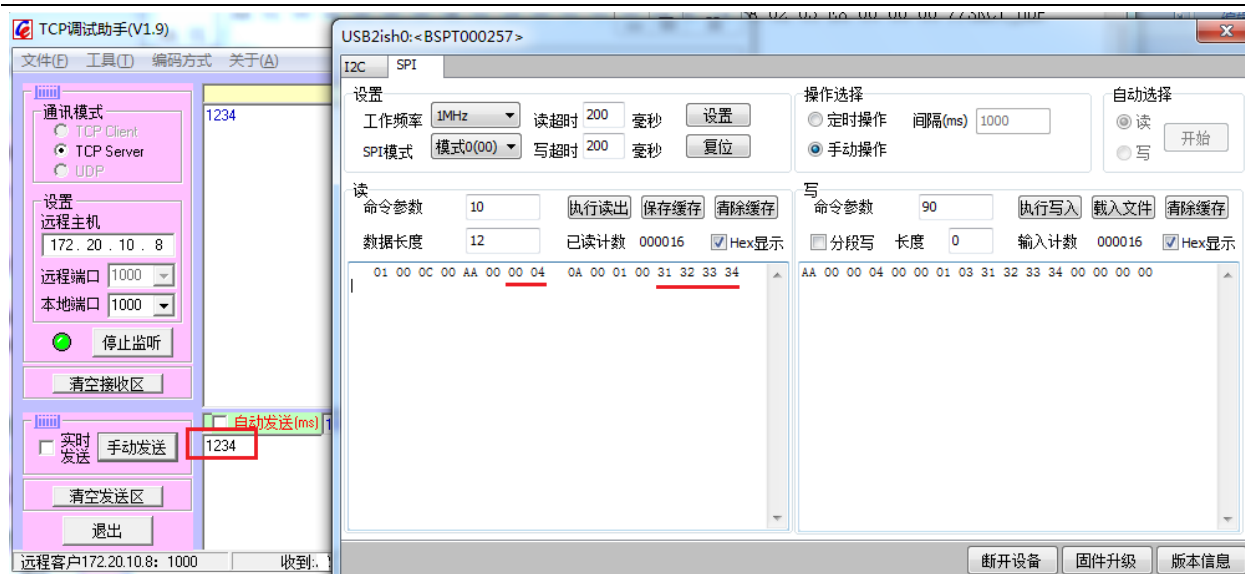


TCP Server 接收

7.3 无线网卡接收数据

如图所示，TCP Server 端发送字符 1234，网卡模块接收到 31 32 33 34。

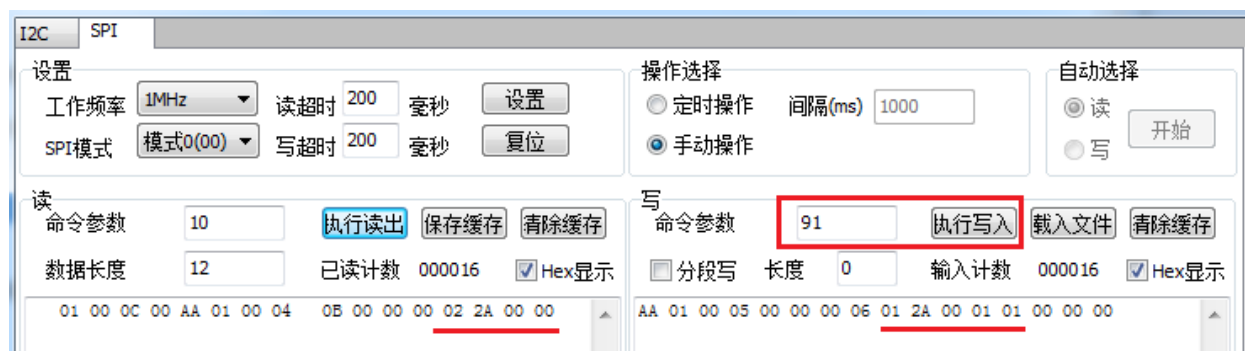
备注：TCP Server 显示的是字符，无线网卡端为 Hex 格式。



TCP Server 发送

7.4 关闭 Socket 连接

如图所示，根据 SPI 指令协议关闭 Socket 连接，依次读取 06、02、10 寄存器获取返回结果。



关闭 Socket

8 STM32 的 Keil 工程实现 SPI Master 通信

Wintermicro 官网的软件资料有 STM32 的 DEMO 工程，实现了 SPI 的 Master 功能，建议用户买一套开发板和下载器 JLINK，通过此工程熟悉 SPI 接口的主机如何实现和我司 RI 指令通信。

用户打开北京联盛德微电子有限责任公司（wintermicro）的官网：www.wintermicro.com。直接下载 STM32 DEMO 工程，通过 KEIL 打开工程，自行查看分析代码，此 DEMO 工程是如何实现 SPI MASTER。其他芯片请自行参考此工程，实现 SPI MASTER，完成和我司模块 SPI 接口的通信。