

# 模块说明

W5500的厂商是韩国WIZnet, 特性如下

- 全硬件TCP/IP协议栈: TCP,UDP,ICMP,IPv4,ARP,IGMP,PPPoE -- 注意只有IPv4
- 支持SPI模式0,3, 最高80MHz SPI时钟
- 8个独立的硬件Socket, 各路通信互不影响
- 32K字节收发缓存, 可以灵活定义各个Socket的大小
- 集成802.3以太网MAC
- 集成10Base-T / 100Base-T以太网PHY
- 支持嵌入式操作系统: Linux & RTOS
- 支持掉电模式 & UDP网络唤醒
- 工作电压3.3V, I/O 5V耐压
- 支持自动协商 (全/半双工, 10M/100M)

用起来比ENC28J60简单多了, 不用费劲巴拉的套uIP, LWIP了.

# 在stm32f103c8t6上运行W5500

## 需要的组件

- STM32F103C8T6开发板
- W5500模块
- USB2TTL: PL2303 or CH340 or FT232 or CP2102
- ST-Link or J-Link
- 带网口的路由器或交换机, 方便联网测试

## 接线图

这个示例中只需要用到SPI接口, RST和INT都是置空. PA9和PA10用来观察log输出

W5500	STM32	USB2TTL
GND	GND	GND
VCC	3.3V	
RST		
INT		
SCS	PA0	
CLK	PA5	
MISO	PA6	
MOSI	PA7	
	PA9	RX

W5500	STM32	USB2TTL
	PA10	TX

## 运行和代码说明

### 代码地址

<https://github.com/IOsetting/stm32f103-w5500>

### 代码说明

基于官方的代码库: [https://github.com/Wiznet/ioLibrary\\_Driver](https://github.com/Wiznet/ioLibrary_Driver)

现在淘宝商家给的代码示例, 以及网络上搜到的W5500项目, 代码大都是非官方的实现(或者是从官方早期某个版本修改而来), 运行在F103Vxx或者F103Rxx的板子上是正常的, 但是换到F103C8xx板子上就会有问题, 而且不好修改.

建议基于官方Github上的库做开发.

- WIKI: <https://wizwiki.net/wiki/doku.php/products:w5500:driver>
- 关于读取是使用中断还是寄存器的讨论: <https://forum.wiznet.io/t/topic/4407/4>

### 接收数据

Use `getSn_RX_RSR(sn)` : Sn\_RX\_RSR indicates the data size received and saved in Socket n RX Buffer.

Some interesting reading:

```
uint16_t getSn_RX_RSR(uint8_t sn)
{
    uint16_t val=0, val1=0;
    do {
        val1 = WIZCHIP_READ(Sn_RX_RSR(sn));
        val1 = (val1 << 8) + WIZCHIP_READ(WIZCHIP_OFFSET_INC(Sn_RX_RSR(sn), 1));
        if (val1 != 0) {
            val = WIZCHIP_READ(Sn_RX_RSR(sn));
            val = (val << 8) + WIZCHIP_READ(WIZCHIP_OFFSET_INC(Sn_RX_RSR(sn), 1));
        }
    } while (val != val1);
    return val;
}
```

In this method, RX\_RSR was read twice in each iteration till the values are the same. This is to avoid the problem described in UDP header seems to contain the expected size, but the RX\_RSR value is much smaller

Read all data if the length exceeds your buffer size:

```
if((size = getSn_RX_RSR(sn)) > 0) { // Sn_RX_RSR: Socket n Received Size Register,
Receiving data length
    if(size > DATA_BUF_SIZE) size = DATA_BUF_SIZE; // DATA_BUF_SIZE means user
defined buffer size (array)
    ret = recv(sn, buf, size); // Data Receive process (H/W Rx socket buffer -
> User's buffer)
```

```
    if(ret <= 0) return ret; // If the received data length <= 0, receive
failed and process end
    size = (uint16_t) ret;
    // Add code here to handle the data
}
```

## 发送数据

```
uint16_t sentsize = 0;
while(size != sentsize) {
    ret = send(sn, buf+sentsize, size-sentsize); // Data send process (User's buffer
-> Destination through H/W Tx socket buffer)
    if(ret < 0) { // Send Error occurred (sent data length < 0)
        close(sn); // socket close
        return ret;
    }
    sentsize += ret; // Don't care SOCKERR_BUSY, because it is zero.
}
```

## 配置 MAC, IP, Mask, Gateway and Remote Server IP

编辑 main.c, 在下面的代码中进行修改

```
void Load_Net_Parameters(void)
{
    gWIZNETINFO.gw[0] = 192; //Gateway
    gWIZNETINFO.gw[1] = 168;
    gWIZNETINFO.gw[2] = 1;
    gWIZNETINFO.gw[3] = 1;

    gWIZNETINFO.sn[0]=255; //Mask
    gWIZNETINFO.sn[1]=255;
    gWIZNETINFO.sn[2]=255;
    gWIZNETINFO.sn[3]=0;

    gWIZNETINFO.mac[0]=0x0c; //MAC
    gWIZNETINFO.mac[1]=0x29;
    gWIZNETINFO.mac[2]=0xab;
    gWIZNETINFO.mac[3]=0x7c;
    gWIZNETINFO.mac[4]=0x00;
    gWIZNETINFO.mac[5]=0x01;

    gWIZNETINFO.ip[0]=192; //IP
    gWIZNETINFO.ip[1]=168;
    gWIZNETINFO.ip[2]=1;
    gWIZNETINFO.ip[3]=204;

    gWIZNETINFO.dhcp = NETINFO_STATIC;
}
uint8_t destip[4] = {192, 168, 1, 210};
uint16_t destport = 3333;
```

## 输出日志

在编译参数中增加 `_DHCP_DEBUG_` and `_LOOPBACK_DEBUG_` 将在UART1中输出日志.

## 编译

---

包含路径

```
..\libraries\CMSIS\CM3\CoreSupport;..\libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x;..\libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm;..\libraries\STM32F10x_StdPeriph_Driver\inc;..\hardware;..\user
```

产生的编译控制字符串

```
--c99 --gnu -c --cpu Cortex-M3 -D__MICROLIB -g -O0 --apcs=interwork --split_sections -I ../libraries\CMSIS\CM3\CoreSupport -I ../libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x -I ../libraries\CMSIS\CM3\DeviceSupport\ST\STM32F10x\startup\arm -I ../libraries\STM32F10x_StdPeriph_Driver\inc -I ../hardware -I ../user -I./RTE/_stm32f103c8w5500 -IC:/Keil_v5/ARM/PACK/Keil/STM32F1xx_DFP/2.3.0/Device/Include -IC:/Keil_v5/ARM/CMSIS/Include -D__UVISION_VERSION="525" -DSTM32F10X_MD -DUSE_STDPERIPH_DRIVER -D_DHCP_DEBUG_ -D_LOOPBACK_DEBUG_ -o .\Objects\*.o --omf_browse .\Objects\*.crf --depend .\Objects\*.d
```

## 烧录

---

使用 ST-Link 将代码写入开发板

ST-Link 和 STM32F103C8T6 开发板的连接方式

```
G    -- GND
CLK  -- SWCLK
IO   -- SWDIO
V3   -- 3.3V
```

## 测试

---

- Ping test: Ping the IP address you defined in main.c
- TCP client test: Run `nc -l 3333` to listen port 3333 on server IP, and restart stm32 board