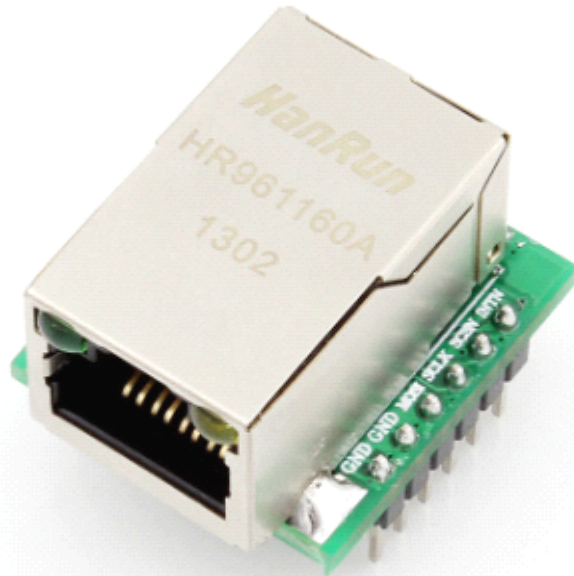


SPI to Ethernet Module

(USR-ES1)

File: Ver1.1



Jinan USR IOT Technology Co., Ltd. works on LAN and WAN and wireless for MCU to Ethernet Solutions, Ethernet, WIFI, GPRS, Zigbee and Wireless modules, we can supply custom design for those usage, looking forward to cooperate with you.

SPI to Ethernet Module.....	1
1. Quick use.....	3
2. Product Introduction.....	5
2.1. Function characteristics.....	6
2.2. The characteristics.....	6
2.3. product model.....	6
2.4. Goods packaging.....	7
3. Hardware description.....	7
3.1. Hardware.....	7
3.2. Pin diagram.....	8
3.3. Diameters.....	9
3.4. Reset Timing.....	10
4. Develop Tools.....	10
4.1. IAR Embedded.....	10
5. Application of structure diagram.....	11
6. Common problem.....	11
6.1. No communication.....	11
6.2. IAR routine compilation errors.....	11
6.3. No communication, cable connection does not recognize.....	11
7. Contact us.....	11
8. Modified history.....	12

1. Quick use

Hardware Requirement:

1 MCU development board(we use one kind of STM32F103RBT6 core board), +3.3V power from core board, Network cable, USB to TTL converter.

The hardware connection diagram as follows:

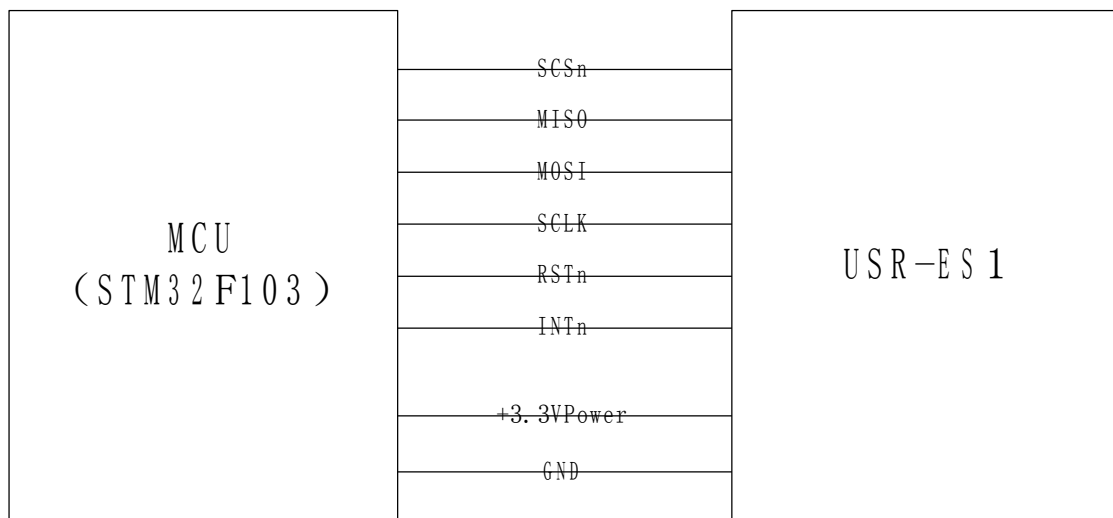


Diagram 1-1 Connection Diagram

Because here the use of the 3.3V microcontroller system, so the middle without adding a level conversion.

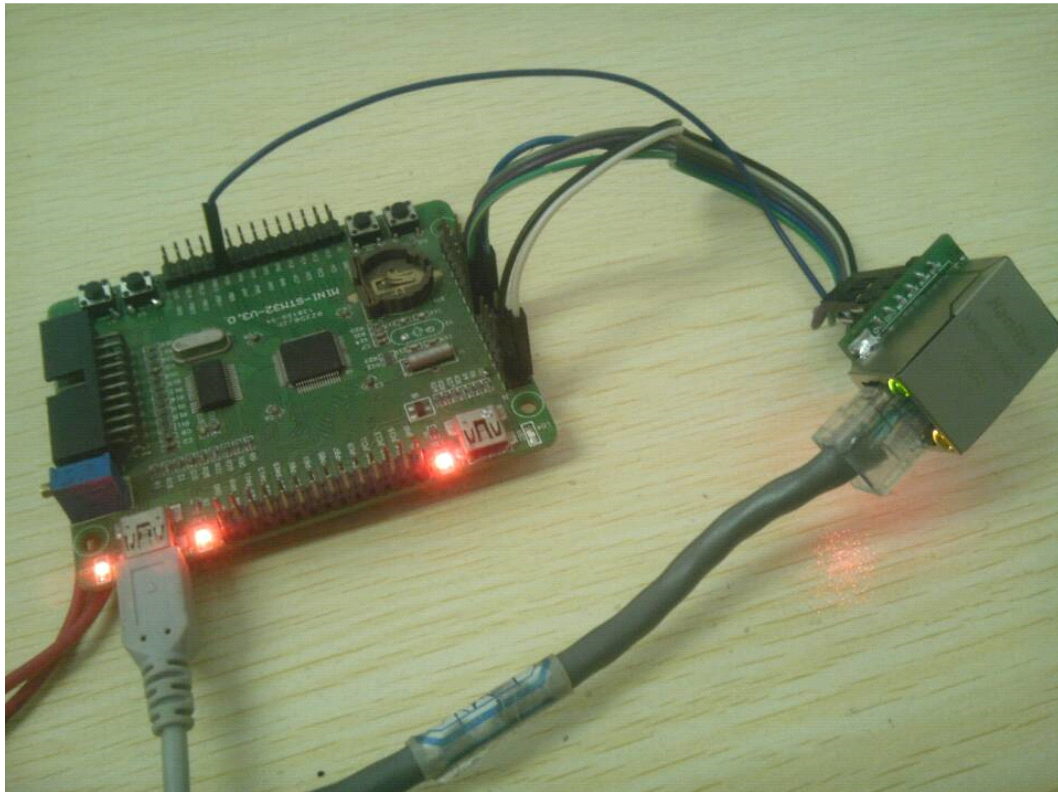


Diagram 1-2 Connection with an Core Board

Routine parameters module, the default for the:

IP Address: 192.168.1.101

The subnet mask: 255.255.255.0

GateWay Address: 192.168.1.1

```

app - IAR Embedded Workbench IDE
File Edit View Project Tools Window Help

Workspace
Release
Files
app - Release
  ewarmv5.4
  w5500
    md5.c
    socket.c
    SPI2.c
    w5500.c
  at24c16.c
  device.c
  httpd.c
  httputil.c
  main.c
  mcu_init.c
  stm32f10x_it.c
  system_stm32f10x.c
  util.c
  Output
    app.out

config.h | main.c | device.c | w5500.c | SPI2.c | mcu_init.c | httputil.c

for (i = 0, j = 0; i < (uint8)(sizeof(ConfigMsg)-4); i++)
{
    data = *(uint8 *) (ConfigMsg.mac+j);
    at24c16_write(dAddr, data);
    dAddr += 1;
    j +=1;
}

static void set_default(void)
{
    uint8 mac[6]={0x00,0x08,0xdc,0x11,0x11,0x11};
    uint8 lip[4]={192,168,1,101};
    uint8 sub[4]={255,255,255,0};
    uint8 gw[4]={192,168,1,1};
    uint8 dns[4]={8,8,8,8};
    memcpy(ConfigMsg.lip, lip, 4);
    memcpy(ConfigMsg.sub, sub, 4);
    memcpy(ConfigMsg.gw, gw, 4);
    memcpy(ConfigMsg.mac, mac, 6);
    memcpy(ConfigMsg.dns, dns, 4);
    /*
    uint8 dhcp;
    uint8 debug;
  
```

Diagram 1-3 The Sample Project

Ping module's IP address.



Diagram 1-4 Ping

Open the Webpage in browser.

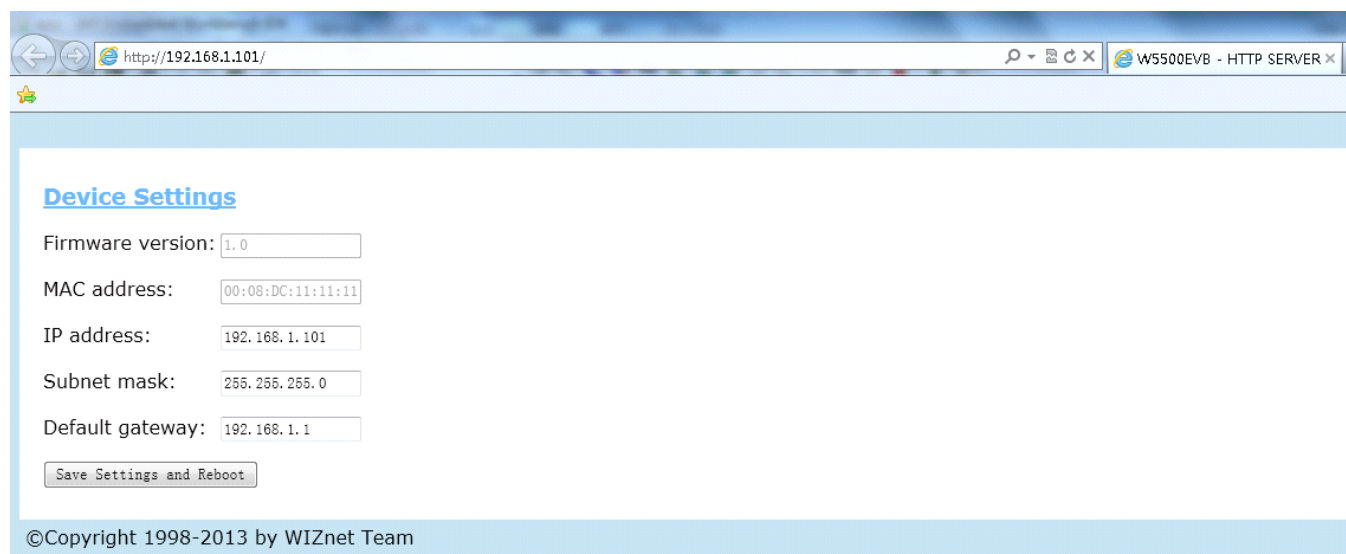


Diagram 1-5 Module' webpage

2. Product Introduction

USR-ES1 is the Ethernet module of a SPI interface, interface is TTL level of 3.3V, power supply voltage of +3.3V, please ensure that the current is not less than 200mA, voltage is continuous and stable +3.3V.

2.1. Function characteristics

- High speed SPI interface 80MHz
- Hardware-TCP/IP-protocol stack user built in, almost without understanding the complex network protocol knowledge
- Supports up to 8 Socket
- Support TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE protocol
- Integration of the data link layer, physical layer
- Support the wakeup
- Support high speed serial peripheral interface (SPI model 0, 3)
- Internal 32K bytes receive buffer
- The embedded 10BaseT/100BaseTX Ethernet physical layer (PHY)
- Support auto negotiation (10/100-Based full duplex or half duplex)
- Does not support the IP patch
- The working voltage of 3.3V, I/O and 5V voltage signal
- LED display (full duplex or half duplex, network connection, network speed, active)
- Pin type package ultra small, convenient for embedded applications
- Application of C sample project

2.2. The characteristics

Name	Description
Power supply mode	3.3V external power supply, current should be more than 200mA
Control interface	The TTL level, 3.3V SPI interface; 2* single chip
PCB size	23 * 25 mm
Mechanical Dimensions	28.5 * 23 * 24

Diagram 2-1 Characteristic

2.3. product model

Type	characteristic	remarks
USR-ES1	The SPI communication interface, pin package, TTL level 3.3V	

Diagram 2-2 Product model

2.4. Goods packaging

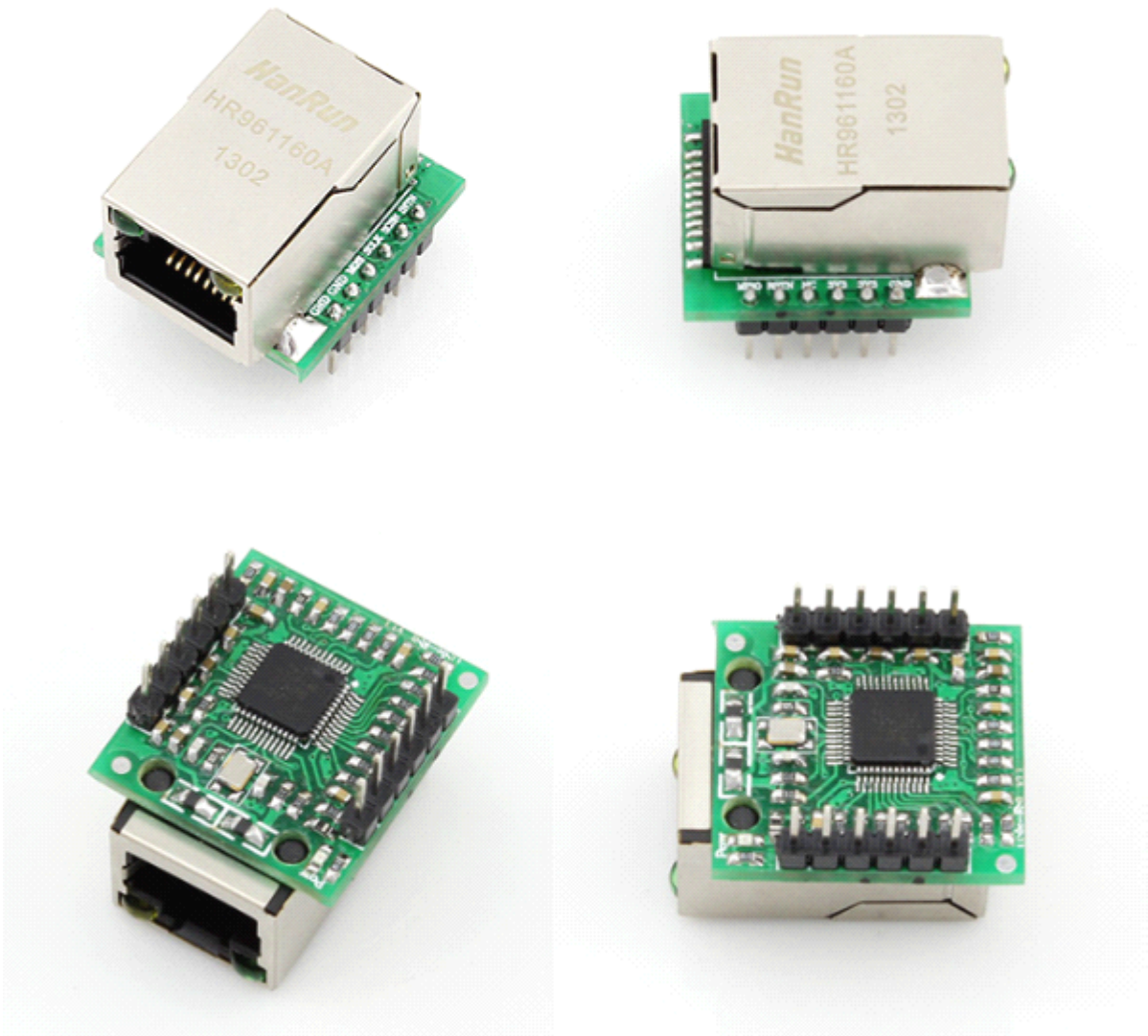
Delivery list:

One USR-ES1 Ethernet module

One CD

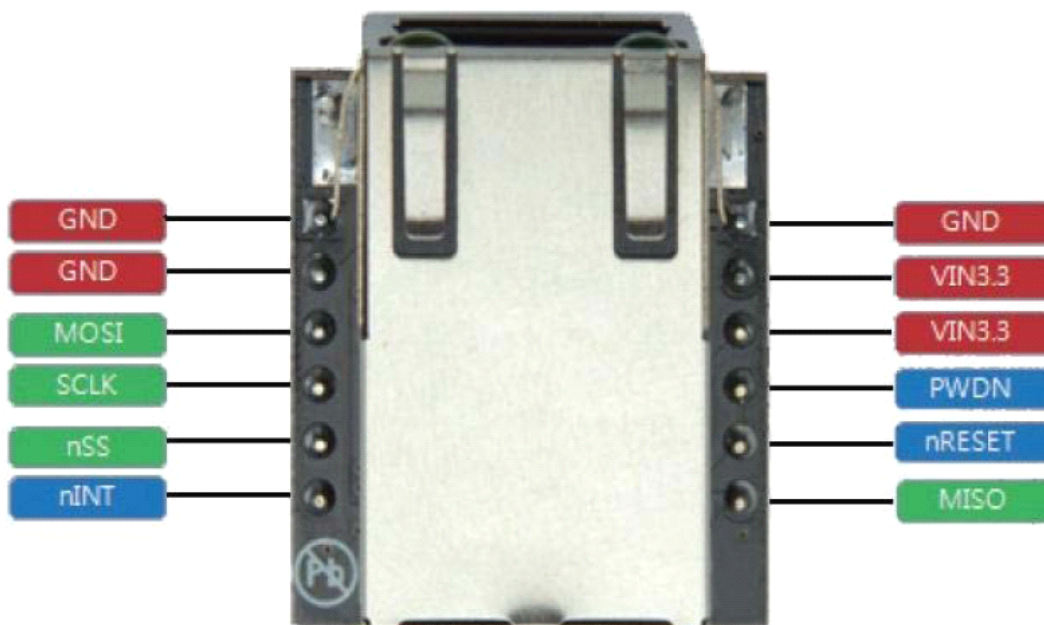
3. Hardware description

3.1. Hardware



A total of two sets of pins, the first group and the second group.

3.2. Pin diagram



< TOP side view >

Diagram 3-1 Pin Diagram

Pin	name	function
1 - 1	GND	Ground
1 - 2	GND	Ground
1 - 3	MOSI	SPI Master Out Slave In This pin is used to SPI MOSI signal pin.
1 - 4	SCLK	SPI Clock This pin is used to SPI Clock Signal pin.
1 - 5	SCSn	SPI Slave Select : Active Low This pin is used to SPI Slave Select signal Pin when using SPI interface.
1 - 6	INTn	Interrupt : Active low This pin indicates that W5200 requires MCU attention after socket connecting, disconnecting, data receiving timeout, and WOL (Wake on LAN). The interrupt is cleared by writing IR Register or Sn_IR (Socket n-th Interrupt Register). All interrupts are maskable.
2 - 1	GND	Ground
2 - 2	+3.3V	Power : 3.3 V power supply

2 - 3	+3.3V	Power : 3.3 V power supply
2 - 4	NC	Not connected
2 - 5	RSTn	Reset : This pin is active low input to initialize or re-initialize W5200. It should be held at least 2us after low assert, and wait for at least 150ms after high de-assert in order for PLL logic to be stable.
2 - 6	MISO	SPI Master In Slave Out This pin is used to SPI MISO signal pin.

Diagram 3-2 Pin Description

3.3. Diameters

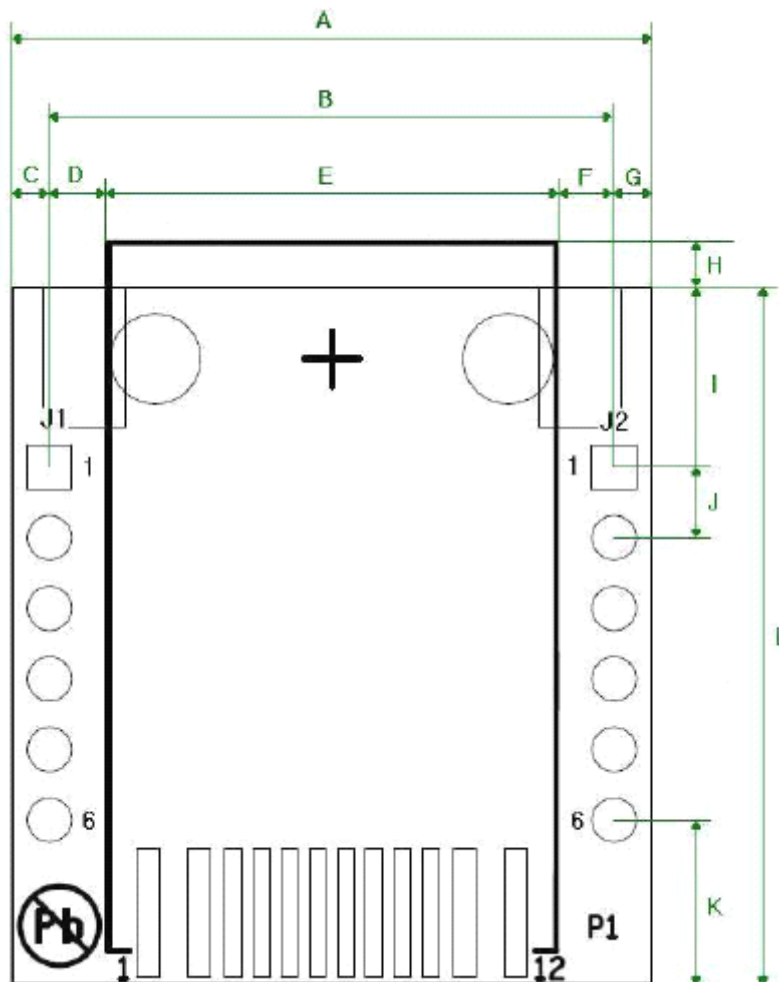


Diagram 3-3 Diameters

Symbol	Dimension(mm)	Symbol	Dimension(mm)
A	23.00	G	1.34
B	20.32 (2.54 x 8)	H	3.50 (+/- 0.50)

C	1.34	I	6.40
D	2.11	J	2.54
E	16.10	K	5.80
F	2.11	L	25.00

Diagram 3-4 diameters in millimeter

3.4. Reset Timing

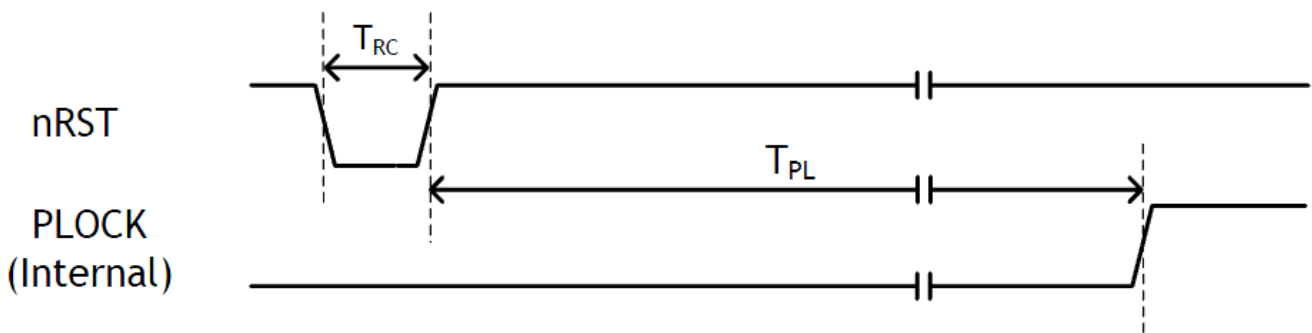


Diagram 3-5 Reset Timing

Symbol	Description	Min	Max
TRC	Reset Cycle Time	2 us	-
TPL	nRST internal PLOCK	-	150 ms

Diagram 3-6 Reset Timing

4. Develop Tools

4.1. IAR Embedded

IAR embedded workbench. currently support ARM IDE. (other IDE tools also support ARM IDE, for example, such as the Keil). software version is for ARM 5.41 embedded workbench. On how to use IAR, see the IAR manual.

5. Application of structure diagram

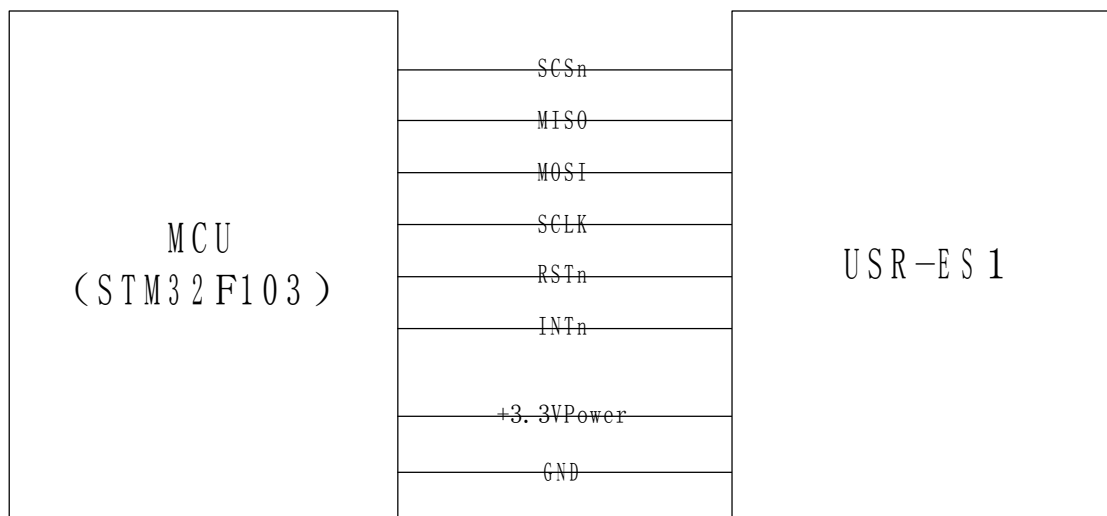


Diagram 5-1 Connection Diagram

6. Common problem

6.1. No communication

Power use 3.3V power supply after, to ensure that current above 200mA.

6.2. IAR routine compilation errors

IAR compiler environment, use IAR5.3, 5.4 version, version of IAR6.5 due to the difference in the official library, will be an error situation, you need to reinstall IAR5.4 or manual modification to eliminate these errors.

6.3. No communication, cable connection does not recognize

W5500 does not support Auto-MDI-X, which does not support the cross connect switch automatically, for some older equipment, may need to cross line specific or direct connection. Such as modules connected module, require the use of cross line, module. The router or switch requires the use of straight line.

8. Modified history

- 1) V1.0 file established