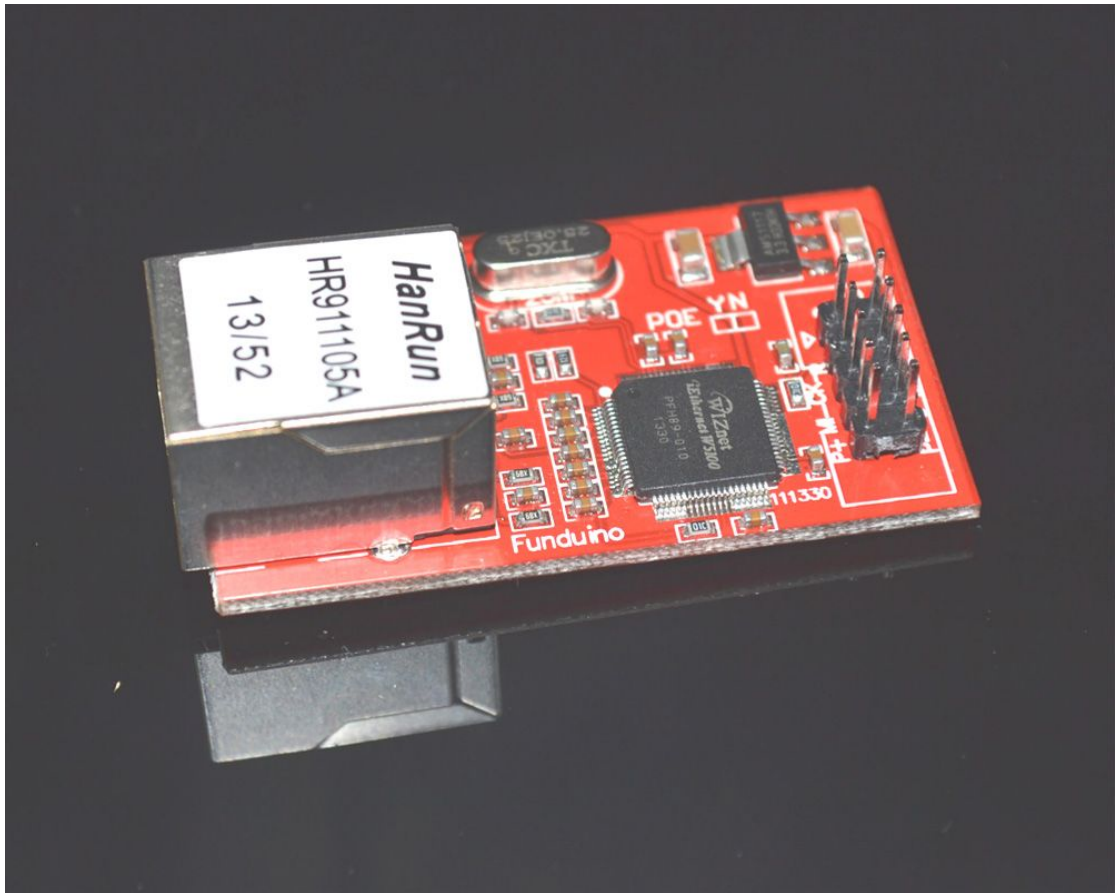


W5100 以太网模块 Ethernet 网络模块



简介:

W5100 是一款多功能的单片网络接口芯片，内部集成有 10/100 以太网控制器，主要应用于高集成、高稳定、高性能和低成本的嵌入式系统中。使用 W5100 可以实现没有操作系统的 Internet 连接。W5100 与 IEEE802.3 10BASE-T 和 802.3u 100BASE-TX 兼容。W5100 内部集成了全硬件的、且经过多年市场验证的 TCP/IP 协议栈、以太网介质传输层 (MAC) 和物理层 (PHY)。使用 W5100 不需要考虑以太网的控制，只需要进行简单的端口 (Socket) 编程。

产品特性

- 多种接口可以选择：直接总线、间接总线和 SPI 总线；
- 支持硬件 TCP/IP 协议栈，支持 TCP,UDP,ICMP,IGMP,IPv4, ARP,PPPoE.
- 多达4个独立端口；
- 内部集成16KBYTE 收发缓存；
- 多种信息指示输出，包括 RX,TX,Full/Duplex,Colision,Link,Speed；
- 支持自动极性转换 (DMI/MDIX)；
- 3.3V 供电，IO 信号兼容5V 电压；
- 自带网口，双排2*14 2.0mm 插针；

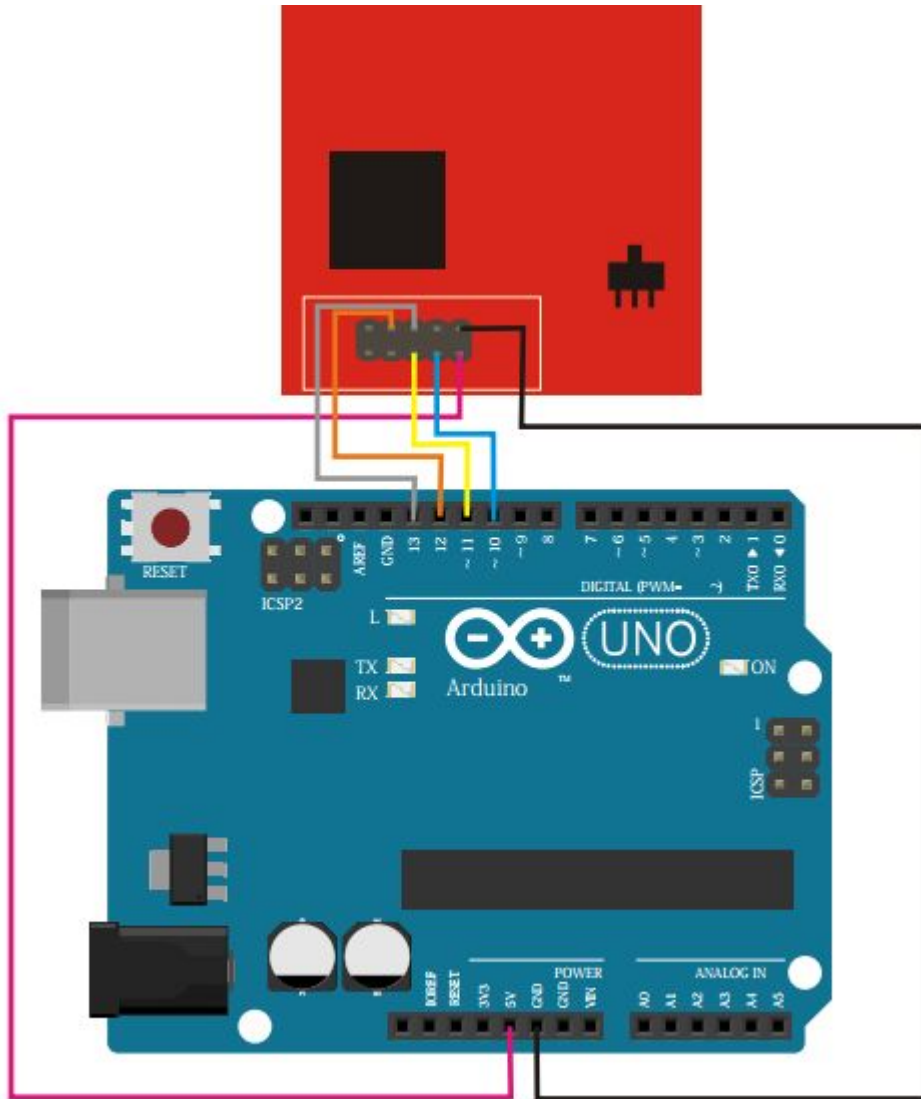
兼容 Arduino 的官方以太网库。

三、模块测试

在测试这个模块之前，首先来看下它与 Arduino 的连接，参照下表用杜邦线将它们连接起来：

W5100 pin	Arduino pin
GND	GND
VCC	VCC
CK	D13
MI	D12
MO	D11
SS	D10

连接示意图：



OK 连接好硬件之后就可以开始下载代码了

程序代码：

/*

Web Server

A simple web server that shows the value of the analog input pins.
using an Arduino Wiznet Ethernet shield.

Circuit:

- * Ethernet shield attached to pins 10, 11, 12, 13
- * Analog inputs attached to pins A0 through A5 (optional)

created 18 Dec 2009

by David A. Mellis
modified 9 Apr 2012
by Tom Igoe

```
*/
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
// Enter a MAC address and IP address for your controller below.
```

```
// The IP address will be dependent on your local network:
```

```
byte mac[] = {
```

```
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
IPAddress ip(192,168,1,177);//根据自己的 IP 修改
```

```
// Initialize the Ethernet server library
```

```
// with the IP address and port you want to use
```

```
// (port 80 is default for HTTP):
```

```
EthernetServer server(80);
```

```
void setup() {
```

```
  // Open serial communications and wait for port to open:
```

```
  Serial.begin(9600);
```

```
  while (!Serial) {
```

```
    ; // wait for serial port to connect. Needed for Leonardo only
```

```
  }
```

```
// start the Ethernet connection and the server:
```

```
Ethernet.begin(mac, ip);
```

```
server.begin();
```

```
Serial.print("server is at ");
```

```
Serial.println(Ethernet.localIP());
```

```
}
```

```
void loop() {
```

```
  // listen for incoming clients
```

```
  EthernetClient client = server.available();
```

```
  if (client) {
```

```
    Serial.println("new client");
```

```
    // an http request ends with a blank line
```

```
    boolean currentLineIsBlank = true;
```

```
    while (client.connected()) {
```

```

if (client.available()) {
  char c = client.read();
  Serial.write(c);
  // if you've gotten to the end of the line (received a newline
  // character) and the line is blank, the http request has ended,
  // so you can send a reply
  if (c == '\n' && currentLineIsBlank) {
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close"); // the connection will be closed after completion
of the response
    client.println("Refresh: 5"); // refresh the page automatically every 5 sec
    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");
    // output the value of each analog input pin
    for (int analogChannel = 0; analogChannel < 6; analogChannel++) {
      int sensorReading = analogRead(analogChannel);
      client.print("analog input ");
      client.print(analogChannel);
      client.print(" is ");
      client.print(sensorReading);
      client.println("<br />");
    }
    client.println("</html>");
    break;
  }
  if (c == '\n') {
    // you're starting a new line
    currentLineIsBlank = true;
  }
  else if (c != '\r') {
    // you've gotten a character on the current line
    currentLineIsBlank = false;
  }
}
}
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
Serial.println("client disconnected");
}

```

}

程序下载进去之后，就可以在浏览器输入自己的 IP(需要跟程序代码匹配)，然后回车 OK 结果出来了（有时候需要重启一下控制板）