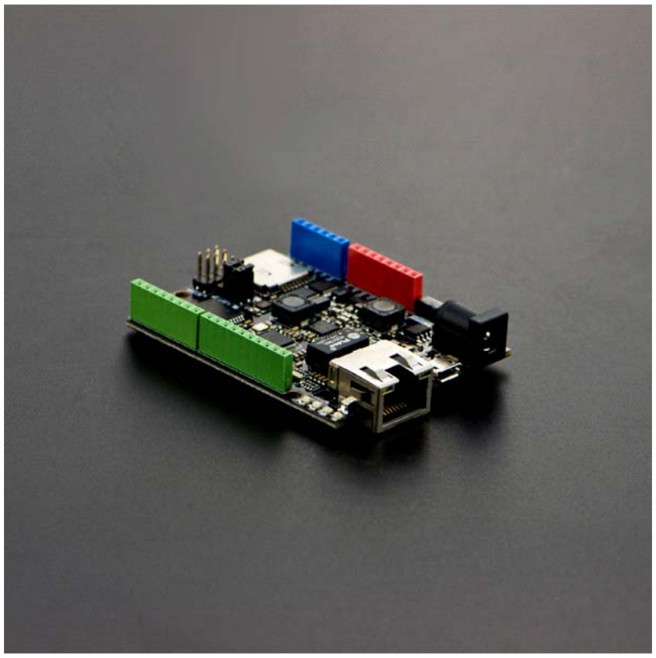
Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

W5500 Ethernet with POE Mainboard SKU: DFR0342

From Robot Wiki



Contents

Introduction

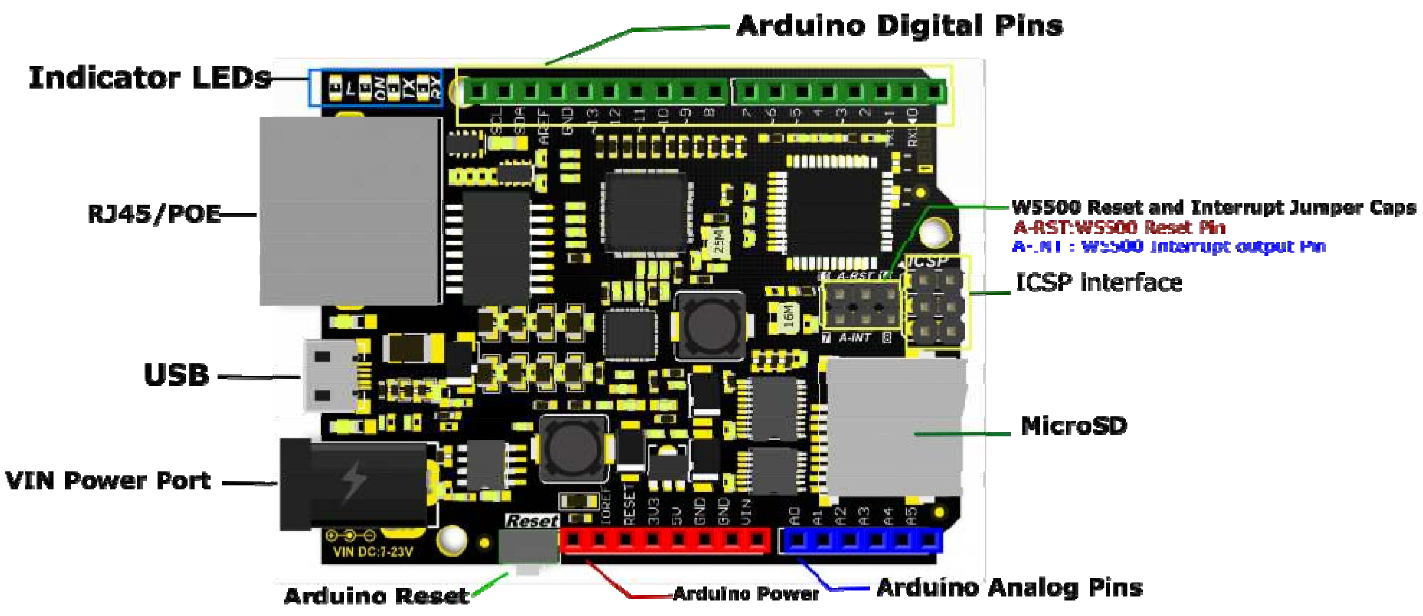
The W5500 Ethernet mainboard is the newest member of the DFRobot Ethernet family. It Is a microcontroller based on the ATmega32u4 and W5500 Ethernet chip with the same footprint as an Arduino Leonardo board, as well as being compatible with most Arduino shields and sensors, making it suitable for many kinds of IOT applications. The W5500 chip is a hardwired TCP/IP embedded Ethernet chip that provides easy internet connection for embedded systems. The board has the TCP/IP stack, 10/1000 ethernet MAC and PHY embedded, allowing internet connectivity in the users application using just the board and nothing else. The W5500 Ethernet mainboard uses a high-efficiency SPI protocol which supports a speed of 80MHZ for high speed network communication. In order to reduce power consumption it also includes WOL (wake on LAN) and power down modes. The board can be powered using a regular VIN, or POE as the power supply. It also integrates a power regulation chip that allows it to work under a complex environment.

Specification

- Microcontroller: Atmel Atmega32u4 (Arduino Leonardo)
- External Input Voltage Range (recommended): 7V~20V DC
- External Input Voltage Range (limit): 6-23V
- POE Input Voltage: 48V AC/DC (802.3af standard PD device)
- Digital I/O Pins: 20
- Analog I/O Pins: 6
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (ATmega32u4) (4KB used by bootloader)
- SRAM: 2 KB (ATmega32u4)
- EEPROM: 1 KB (ATmega32u4)
- Clock Speed: 16 MHz
- PHY: WIZnet W5500
- PHY Clock Speed: 25MHz
- Dimension: 73.5 x 53.5x15mm

Pinout Diagram

We recommend that you remove the jumper caps if you don't need to use the W5500 reset and interrupt functions, or pins 7, 8, 11 or 12 may not function properly.



Tutorial

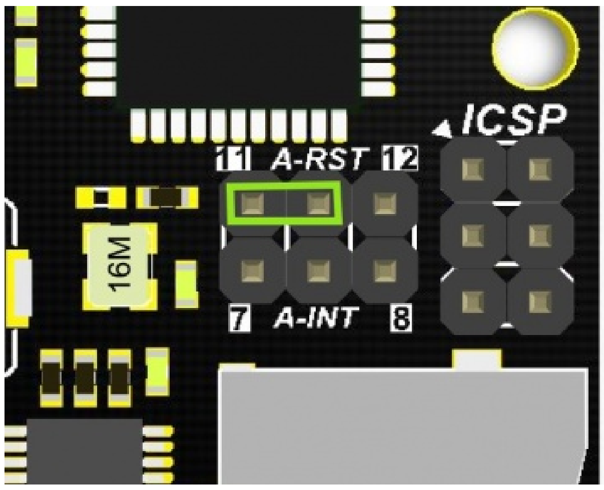
Preparation

You will need:

1. W5500 Ethernet with POE Mainboard
2. Micro USB cable
3. PC
4. RJ45 network cable

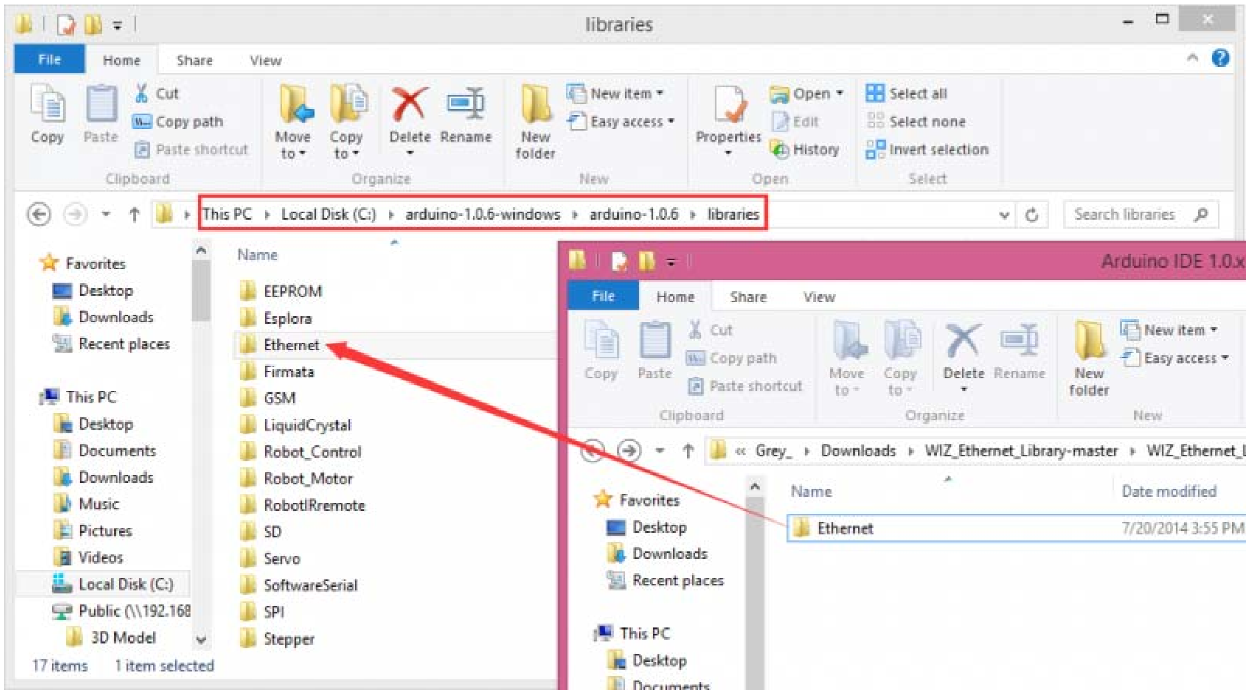
Step 1: Connections

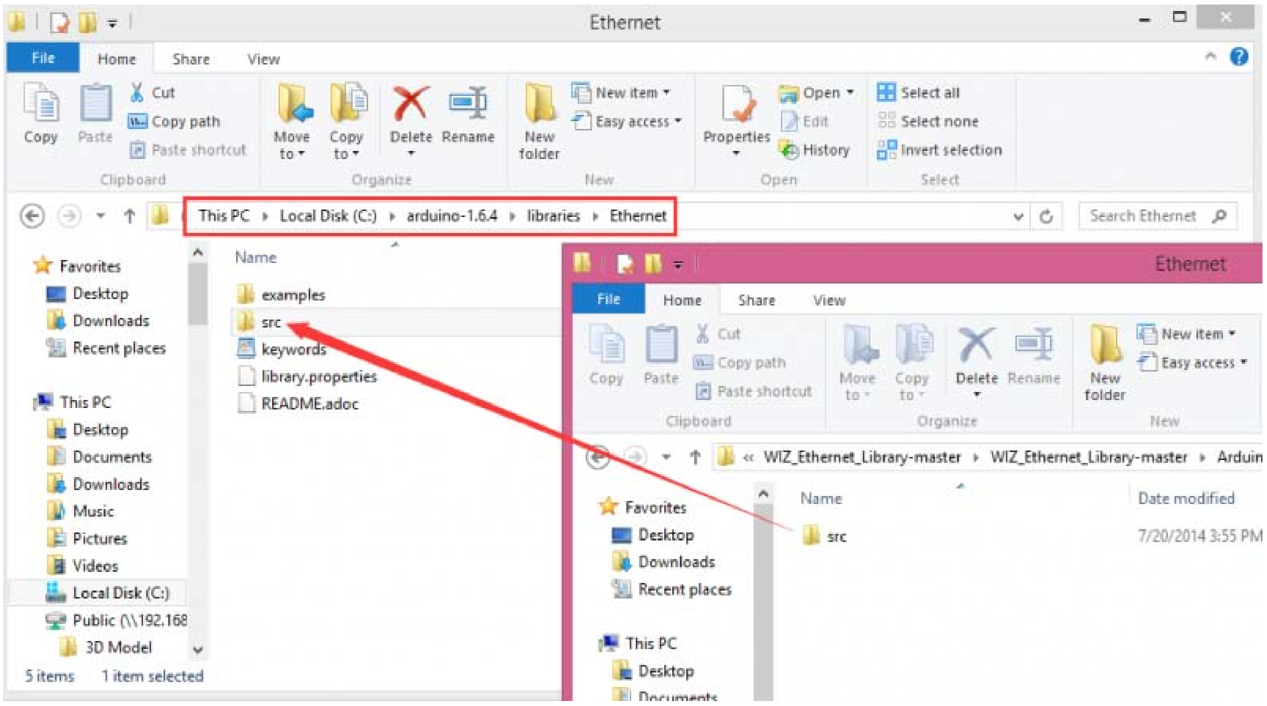
Connect W5500 Reset pins: Connect A-RST pin to D11. (Default: Connected)



Step 2: Library

1. Download the W5500 Ethernet library here https://github.com/Wiznet/WIZ_Ethernet_Library
2. Replace the existing Ethernet library according to your Arduino IDE version:


libraries
This PC ▸ Local Disk (C:) ▸ arduino-1.0.6-windows ▸ arduino-1.0.6 ▸ libraries
EEPROM
Esplora
Ethernet
Firmata
GSM
LiquidCrystal
Robot_Control
Robot_Motor
RobotIRremote
SD
Servo
SoftwareSerial
SPI
Stepper
17 items 1 item selected
Arduino IDE 1.0.x
« Grey_ ▸ Downloads ▸ WIZ_Ethernet_Library-master ▸ WIZ_Ethernet_L
Ethernet 7/20/2014 3:55 PM


Ethernet
This PC ▸ Local Disk (C:) ▸ arduino-1.6.4 ▸ libraries ▸ Ethernet
examples
src
keywords
library.properties
README.adoc
5 items 1 item selected
Ethernet
« WIZ_Ethernet_Library-master ▸ WIZ_Ethernet_Library-master ▸ Arduin
src 7/20/2014 3:55 PM

NOTE: Good news! You can use arduino IDE v1.7 from Arduino.org which has "Ethernet 2" library, which works with this board. It means you don't have to download any modded libraries. Thank for the Vladimir Akopyan comments. Welcome share your projects and comments on the Disqus.

http://www.arduino.org/downloads

Step 3: Pin Definitions

Note: These parts should be declared at the beginning of the sketch.

1. Define W5500 SPI "SS" pin and "Reset" Pin:

```
#define SS     10U    //D10----- SS
#define RST    11U    //D11----- Reset
```

2. Setup function

```
void setup() {
pinMode(SS, OUTPUT);
pinMode(RST, OUTPUT);
digitalWrite(SS, LOW);
digitalWrite(RST,HIGH);  //Reset
delay(200);
digitalWrite(RST,LOW);
delay(200);
digitalWrite(RST,HIGH);
delay(200);              //Wait W5500
```

Example

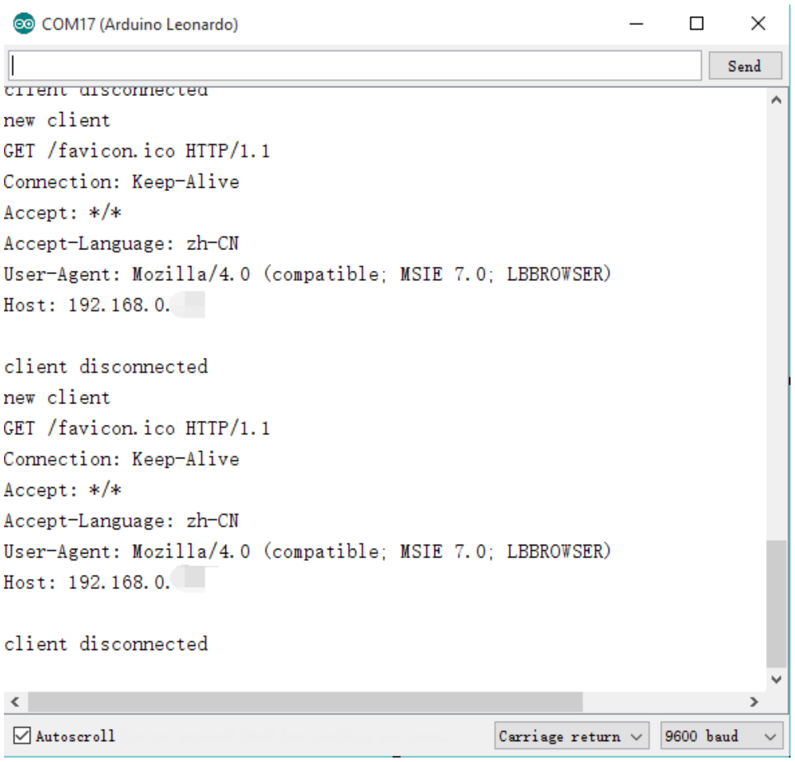
Open Arduino IDE, Select Boards -->Arduino Leonardo and COM port

```
/*  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>
#define SS     10U    //D10 ---- SS
#define RST    11U    //D11 -----Reset

// 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);

// 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() {
  pinMode(SS, OUTPUT);
  pinMode(RST, OUTPUT);
  digitalWrite(SS, LOW);
  digitalWrite(RST, HIGH); //Reset
  delay(200);
  digitalWrite(RST, LOW);
  delay(200);
  digitalWrite(RST, HIGH);
  delay(200);              //Wait W5500
  //Open serial communications and wait for port to open :
  Serial.begin(9600);
  //while (!Serial) {
  //  ; // wait for serial port to connect. Needed for Leonardo only
  //}
  delay(1000);
  // 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");

          // the connection will be closed after completion of the response
          client.println("Connection: close");
          client.println("Refresh: 5");  // refresh the page automatically ev
ery 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");
    }
  }
```

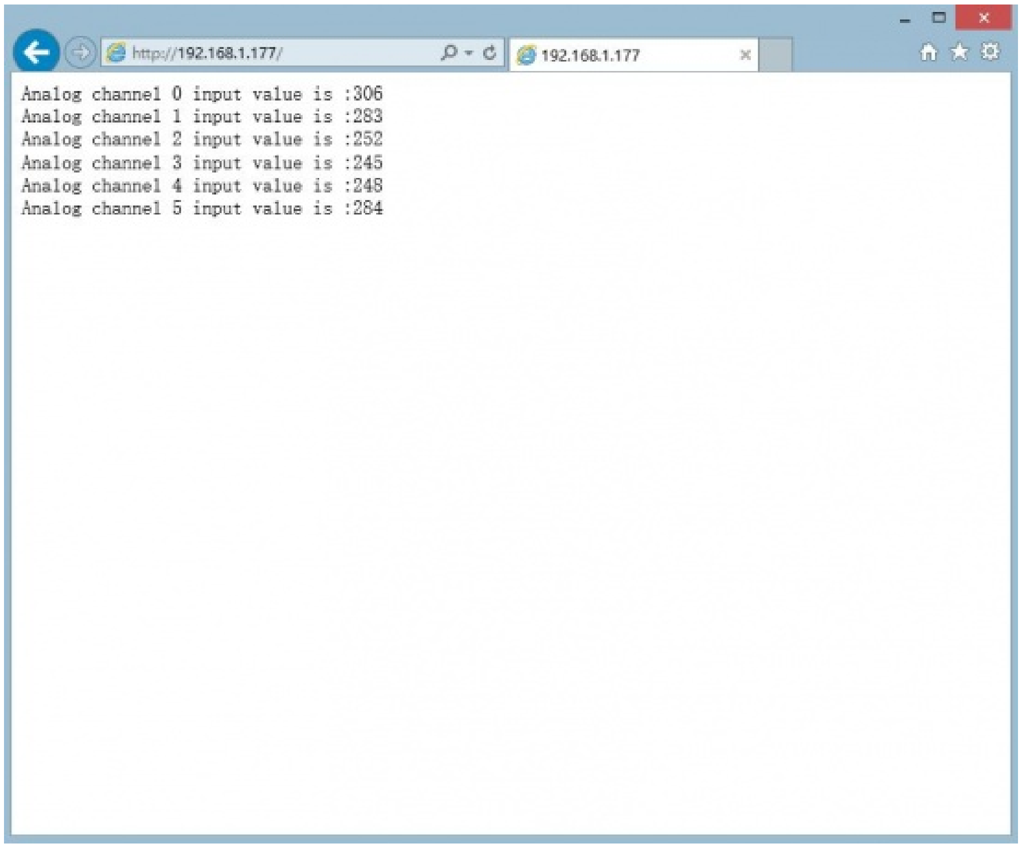
Check that it works

Open the Arduino Serial Monitor to initialize the setup and you will see some information about the connection processing. Open your browser, input the IP address and you should get see data of virtual analog pins:



```
client disconnected
new client
GET /favicon.ico HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; LBBROWSER)
Host: 192.168.0.

client disconnected
new client
GET /favicon.ico HTTP/1.1
Connection: Keep-Alive
Accept: */*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; LBBROWSER)
Host: 192.168.0.

client disconnected
```

Open the Serial Monitor first


http://192.168.1.177/

192.168.1.177

```
Analog channel 0 input value is :306
Analog channel 1 input value is :283
Analog channel 2 input value is :252
Analog channel 3 input value is :245
Analog channel 4 input value is :248
Analog channel 5 input value is :284
```

W5500 Ethernet webserver

NOTE: If you want to skip the serial monitor step, you can comment out the lines waiting for **!serial**, and add a line: delay(2000); to ensure that initialization has finished.

```
Serial.begin(9600);
delay(2000);
//  while (!Serial) {
//    ; // wait for serial port to connect. Needed for Leonardo only
```

// }

POE Power Supply

This board supports POE power supply, so it doesn't need any other power source if you use an active ethernet connection. For more information please check: POE (Power Over Ethernet)Wikipedia   https://en.wikipedia.org/wiki/Power_over_Ethernet

For any question/advice/cool idea to share, please visit **DFRobot Forum**.