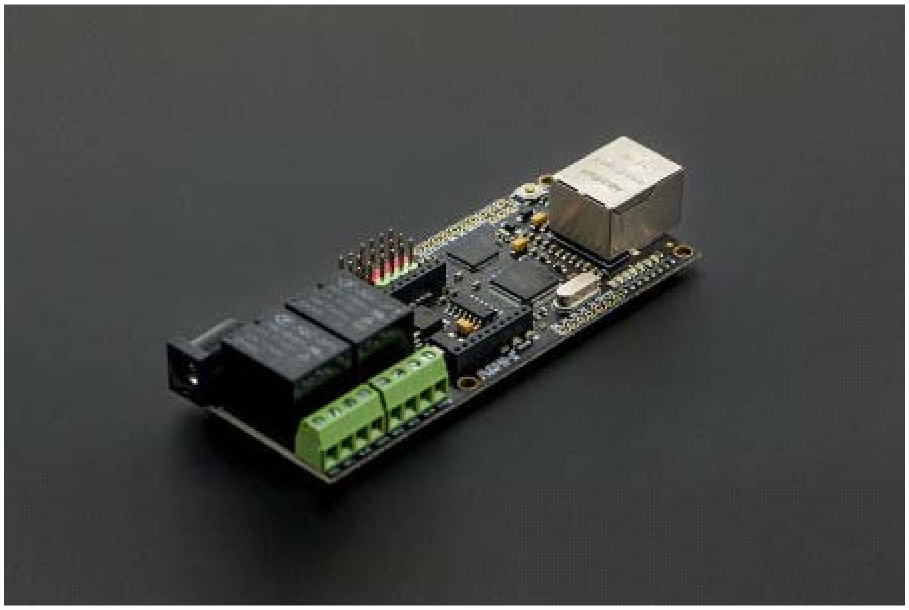
Xboard Relay - An Ethernet Controllered Relay

SKU:DFR0222



Contents

Introduction

Internet of thing is becoming so easy. With Xboard Relay, you can not only monitor data through internet, but also control it through internet. The Xboard Relay combines an Atmega 32u4 microprocessor and wiz5100 chip which is fully compatible with Arduino Lendardo and Ethernet library, it has build-in Xbee socket and 2 Relays which allow an easy play sensoring and controlling over internet.

Not like previous X-board, X-board relay has no programming adapter required. A micro usb cable is the only hardware needed to upload sketch.

Specification

- MCU:Atmega 32u4
- Clock Speed:16 MHz
- Flash Memory:32 KB (ATmega32u4) of which 4 KB used by bootloader
- SRAM:2.5 KB (ATmega32u4)
- EEPROM:1 KB (ATmega32u4)
- Ethernet Chip:Wiz5100
- Power Supply:7.2-12V
- USB Supply:Micro USB@5V
- Pin out:2 Analog/1 I2C/4 Digital Pin Out
- Relay Information:
  Rated through-current: 10A (NO) 5A (NC)
  Maximum switching voltage: 150VAC 24VDC
  Control signal: TTL level
  Contact Rating (Res. Load):10A 277VAC/24VDC
  Max. switching voltage 250VAC/30VDC 250VAC/30VDC
  Max. switching current 15A
  Max. switching power 2770VA 240W 2770VA 240W
  UL Rating: 10A 120VAC /10A 277VAC
  Operate tiem (at nomi. Vot.): 10ms
  Release time (at nomi. Vot.): 5ms

Pinout

Relay's connected to pins D7 and D8

Sample Code

Preparation

You will need:

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

Step 1: Connections

1.The X-Board Relay module access the Internet line into any router LAN port,
2.Micro USB cable plugged into the X-Board relay module, and the other end to the computer USB port.

NOTE: After uploading the sketch below successfully, you have to open Arduino Serial Monitor to get the card run. This is to ensure the initialization done, you could also delete or comments those three lines of code and add a delay(1000) instead.

Step 2:Download

Programming code :

```
//while (!Serial) {
//  ; // wait for serial port to connect. Needed for Leonardo only
//}
```

```
/*
DFRobot X-board V2 Sample Code

 A simple web server with DHPC capbabilty.
1)Get IP address from router automatically
2)Show the value of the analog input pins

 created 28 Sep 2012
 by Ricky
 */

#include <SPI.h>
#include <Ethernet.h>
EthernetServer server(80);
// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
```

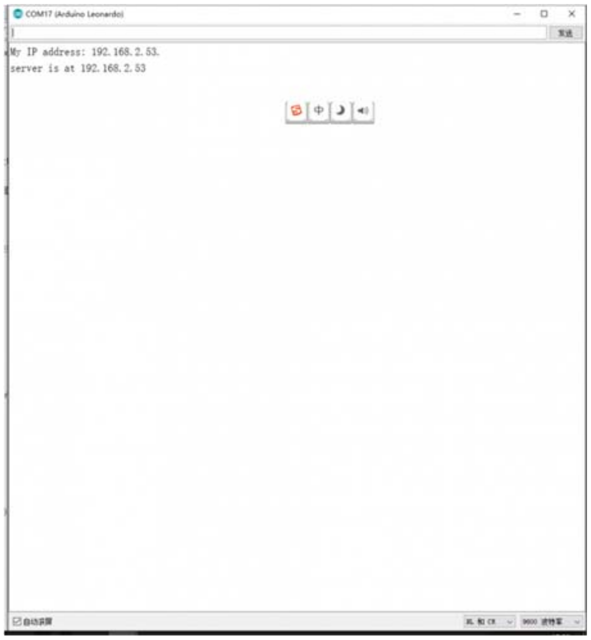
```
byte mac[] = {
  0xDE, 0xCD, 0xAE, 0x0F, 0xFE, 0xED };

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):


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:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    // no point in carrying on, so do nothing forevermore:
    for(;;)
      ;
  }
  // print your local IP address:
  Serial.print("My IP address: ");
  for (byte thisByte = 0; thisByte < 4; thisByte++) {
    // print the value of each byte of the IP address:
    Serial.print(Ethernet.localIP()[thisByte], DEC);
    Serial.print(".");
  }
  Serial.println();

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

```
  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("Connnection: close");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
                    // add a meta refresh tag, so the browser pulls again
every 5 seconds:
          client.println("<meta http-equiv=\"refresh\" content=\"5\">");
          client.println("<link rel=\"stylesheet\" type=\"text/css\" href=
\"http://www.dfrobot.com/ihome/stylesheet/stylesheet.css\" />");
```

```
            client.println("<center> <a href=\"http://www.dfrobot.com\"><i
mg src=\"http://alturl.com/qf6vz\"></a> </center> ");
           client.println("<br />");


           client.println("<div>");
          // 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 />");


          }


             // output the value of each digital input pin
          for (int digitalChannel = 2; digitalChannel < 10; digitalChannel
++) {
            int sensorReading = digitalRead(digitalChannel);
            if(digitalChannel!=7&&digitalChannel!=8)
            {
            client.print("Digital input ");
            client.print(digitalChannel);
            client.print(" is ");
            client.print(sensorReading);
             client.println("<br />");
            }

            else
            {
             client.print("Relay ");
            client.print(digitalChannel);
            client.print(" is ");
            client.print(sensorReading);
             client.println("<br />");

            }



          }




          client.println("</div>");
          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 disonnected");
  }
}
```

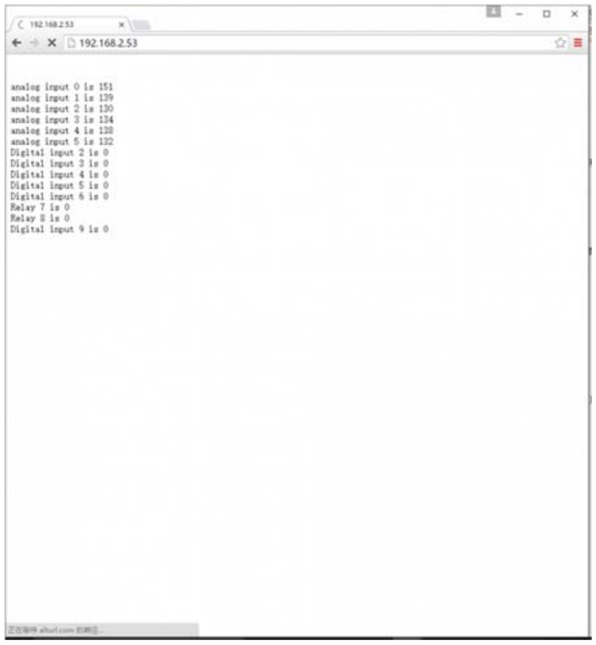
Check that it works

1. After burning is complete, open the serial monitor. Window displays your IP address.



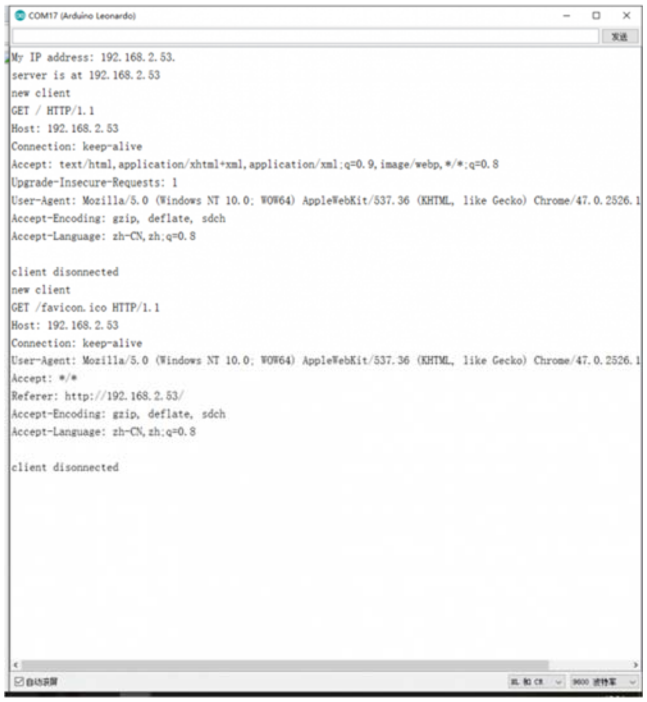
```
My IP address: 192.168.2.53.
server is at 192.168.2.53
```

2. Open a browser -> Enter the IP address -> Enter to open the page.



```
analog input 0 is 151
analog input 1 is 139
analog input 2 is 130
analog input 3 is 134
analog input 4 is 138
analog input 5 is 132
Digital input 2 is 0
Digital input 3 is 0
Digital input 4 is 0
Digital input 5 is 0
Digital input 6 is 0
Relay 7 is 0
Relay 8 is 0
Digital input 9 is 0
```

3. Serial Monitor has a corresponding feedback data



```
My IP address: 192.168.2.53.
server is at 192.168.2.53
new client
GET / HTTP/1.1
Host: 192.168.2.53
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.1
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8

client disonnected
new client
GET /favicon.ico HTTP/1.1
Host: 192.168.2.53
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.1
Accept: */*
Referer: http://192.168.2.53/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8

client disonnected
```

Relay module Application examples

Note that the actual parameters.

- Relay Information:
  Rated through-current: 10A (NO) 5A (NC)
  Maximum switching voltage: 150VAC 24VDC
  Control signal: TTL level
  Contact Rating (Res. Load):10A 277VAC/24VDC
  Max. switching voltage 250VAC/30VDC 250VAC/30VDC
  Max. switching current 15A
  Max. switching power 2770VA 240W 2770VA 240W
  UL Rating: 10A 120VAC /10A 277VAC
  Operate tiem (at nomi. Vot.): 10ms
  Release time (at nomi. Vot.): 5ms