

Examples > WiFi Library

Send and Receive UDP String

This sketch waits for a UDP packet on a local port. When a valid packet is received, an acknowledge packet is sent back to the client on a specified outgoing port.

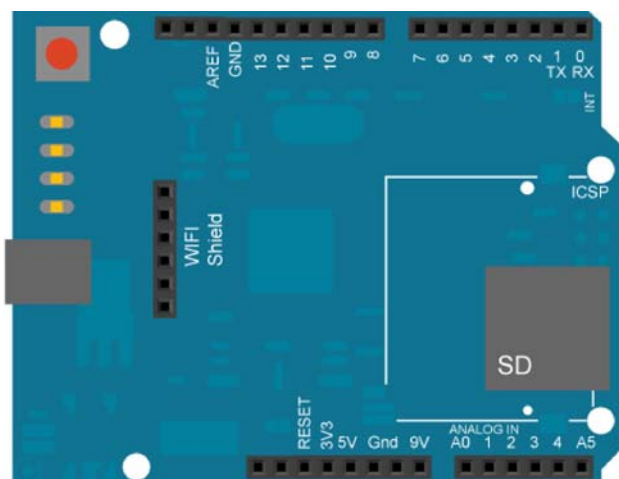
Hardware Required

- Arduino WiFi Shield
- Shield-compatible Arduino board

Circuit

The WiFi shield uses pins 10, 11, 12, and 13 for the SPI connection to the HDG104 module. Digital pin 4 is used to control the slave select pin on the SD card.

You should have access to a 802.11b/g wireless network that connects to the internet for this example. You will need to change the network settings in the sketch to correspond to your particular networks SSID.



(http://www.arduino.cc/en/uploads/Tutorial/WiFiShield_bb.png)

image developed using Fritzing (<http://www.fritzing.org>). For more circuit examples, see the Fritzing project page (<http://fritzing.org/projects/>)

In the above image, the Arduino would be stacked below the WiFi shield.

Code:

```
/*  
  WiFi UDP Send and Receive String  
  
  This sketch wait an UDP packet on localPort using a WiFi shield.  
  When a packet is received an Acknowledge packet is sent to the client on port remotePort  
  
  Circuit:
```

```

* WiFi shield attached

created 30 December 2012
by dlf (Metodo2 srl)

*/

#include <SPI.h>
#include <WiFi.h>
#include <WiFiUdp.h>

int status = WL_IDLE_STATUS;
char ssid[] = "yourNetwork"; // your network SSID (name)
char pass[] = "secretPassword"; // your network password (use for WPA, or use as key for WEP)
int keyIndex = 0; // your network key Index number (needed only for WEP)

unsigned int localPort = 2390; // local port to listen on

char packetBuffer[255]; //buffer to hold incoming packet
char ReplyBuffer[] = "acknowledged"; // a string to send back

WiFiUDP Udp;

void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // check for the presence of the shield:
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
    // don't continue:
    while (true);
  }

  String fv = WiFi.firmwareVersion();
  if (fv != "1.1.0") {
    Serial.println("Please upgrade the firmware");
  }

  // attempt to connect to Wifi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid);

    // wait 10 seconds for connection:
    delay(10000);
  }
  Serial.println("Connected to wifi");
  printWifiStatus();

  Serial.println("\nStarting connection to server...");
  // if you get a connection, report back via serial:
  Udp.begin(localPort);
}

void loop() {

  // if there's data available, read a packet
  int packetSize = Udp.parsePacket();
  if (packetSize) {
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remoteIp = Udp.remoteIP();
    Serial.print(remoteIp);
    Serial.print(", port ");
    Serial.println(Udp.remotePort());

    // read the packet into packetBuffer
    int len = Udp.read(packetBuffer, 255);

```

```

    if (len > 0) {
        packetBuffer[len] = 0;
    }
    Serial.println("Contents:");
    Serial.println(packetBuffer);

    // send a reply, to the IP address and port that sent us the packet we received
    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
    Udp.write(ReplyBuffer);
    Udp.endPacket();
}
}

void printWifiStatus() {
    // print the SSID of the network you're attached to:
    Serial.print("SSID: ");
    Serial.println(WiFi.SSID());

    // print your WiFi shield's IP address:
    IPAddress ip = WiFi.localIP();
    Serial.print("IP Address: ");
    Serial.println(ip);

    // print the received signal strength:
    long rssi = WiFi.RSSI();
    Serial.print("signal strength (RSSI):");
    Serial.print(rssi);
    Serial.println(" dBm");
}

[Get Code] (http://www.arduino.cc/en/Tutorial/WiFiSendReceiveUDPString?action=sourceblock&num=1)

```

See Also:

- [ConnectNoEncryption](http://www.arduino.cc/en/Tutorial/ConnectNoEncryption) (<http://www.arduino.cc/en/Tutorial/ConnectNoEncryption>) : Demonstrates how to connect to an open network
- [ConnectWithWEP](http://www.arduino.cc/en/Tutorial/ConnectWithWEP) (<http://www.arduino.cc/en/Tutorial/ConnectWithWEP>) : Demonstrates how to connect to a network that is encrypted with WEP
- [ConnectWithWPA](http://www.arduino.cc/en/Tutorial/ConnectWithWPA) (<http://www.arduino.cc/en/Tutorial/ConnectWithWPA>) : Demonstrates how to connect to a network that is encrypted with WPA2 Personal
- [ScanNetworks](http://www.arduino.cc/en/Tutorial/ScanNetworks) (<http://www.arduino.cc/en/Tutorial/ScanNetworks>) : Displays all WiFi networks in range
- [WiFiChatServer](http://www.arduino.cc/en/Tutorial/WiFiChatServer) (<http://www.arduino.cc/en/Tutorial/WiFiChatServer>) : Set up a simple chat server
- [WiFiWebClient](http://www.arduino.cc/en/Tutorial/WiFiWebClient) (<http://www.arduino.cc/en/Tutorial/WiFiWebClient>) : Connect to a remote webserver
- [WiFiWebClientRepeating](http://www.arduino.cc/en/Tutorial/WiFiWebClientRepeating) (<http://www.arduino.cc/en/Tutorial/WiFiWebClientRepeating>): Repeatedly make HTTP calls to a server
- [WiFiWebServer](http://www.arduino.cc/en/Tutorial/WiFiWebServer) (<http://www.arduino.cc/en/Tutorial/WiFiWebServer>) : Serve a webpage from the WiFi shield

Share



NEWSLETTER



[\(https://twitter.com/arduino\)](https://twitter.com/arduino)

[\(https://www.facebook.com/official.arduino\)](https://www.facebook.com/official.arduino)

[\(https://plus.google.com/+Arduino\)](https://plus.google.com/+Arduino)

[\(https://www.flickr.com/photos/arduino_cc\)](https://www.flickr.com/photos/arduino_cc)

[\(https://youtube.com/arduinoteam\)](https://youtube.com/arduinoteam)