Network Time Protocol (NTP) Client

In this example, you will use your WiFi Shield and your Arduino or Genuino board to query a Network Time Protocol (NTP) server. This way, your board can get the time from the Internet.

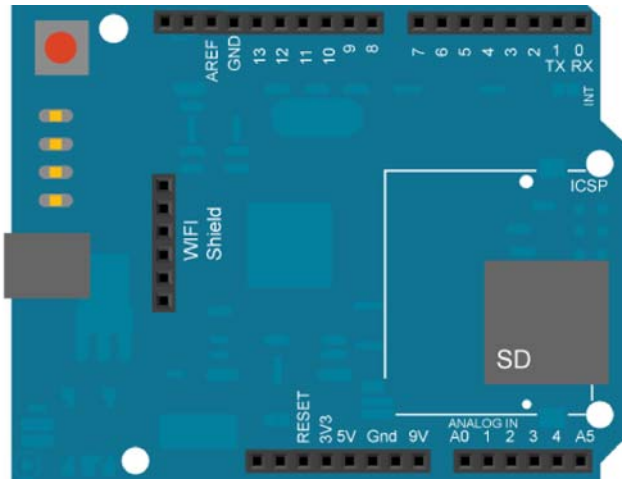
Hardware Required

- Arduino WiFi Shield
- Shield-compatible Arduino or Genuino 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.



(//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 or Genuino board would be stacked below the WiFi shield.

Code

```
/*

 Udp NTP Client

 Get the time from a Network Time Protocol (NTP) time server
 Demonstrates use of UDP sendPacket and ReceivePacket
```

```
 For more on NTP time servers and the messages needed to communicate with them,
 see http://en.wikipedia.org/wiki/Network_Time_Protocol

 created 4 Sep 2010
 by Michael Margolis
 modified 9 Apr 2012
 by Tom Igoe

 This code is in the public domain.

 */

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

int status = WL_IDLE_STATUS;
char ssid[] = "mynetwork";  //  your network SSID (name)
char pass[] = "mypassword";       // your network password
int keyIndex = 0;            // your network key Index number (needed only for WEP)

unsigned int localPort = 2390;      // local port to listen for UDP packets

IPAddress timeServer(129, 6, 15, 28); // time.nist.gov NTP server

const int NTP_PACKET_SIZE = 48; // NTP time stamp is in the first 48 bytes of the message

byte packetBuffer[ NTP_PACKET_SIZE]; //buffer to hold incoming and outgoing packets

// A UDP instance to let us send and receive packets over UDP
WiFiUDP Udp;

void setup() {
  // Open serial communications 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, pass);

    // wait 10 seconds for connection:
    delay(10000);
  }

  Serial.println("Connected to wifi");
  printWifiStatus();

  Serial.println("\nStarting connection to server...");
  Udp.begin(localPort);
}

void loop() {
  sendNTPpacket(timeServer); // send an NTP packet to a time server
  // wait to see if a reply is available
  delay(1000);
  if (Udp.parsePacket()) {
    Serial.println("packet received");
    // We've received a packet, read the data from it
```

```
    Udp.read(packetBuffer, NTP_PACKET_SIZE); // read the packet into the buffer

    //the timestamp starts at byte 40 of the received packet and is four bytes,
    // or two words, long. First, esxtract the two words:

    unsigned long highWord = word(packetBuffer[40], packetBuffer[41]);
    unsigned long lowWord = word(packetBuffer[42], packetBuffer[43]);
    // combine the four bytes (two words) into a long integer
    // this is NTP time (seconds since Jan 1 1900):
    unsigned long secsSince1900 = highWord << 16 | lowWord;
    Serial.print("Seconds since Jan 1 1900 = ");
    Serial.println(secsSince1900);

    // now convert NTP time into everyday time:
    Serial.print("Unix time = ");
    // Unix time starts on Jan 1 1970. In seconds, that's 2208988800:
    const unsigned long seventyYears = 2208988800UL;
    // subtract seventy years:
    unsigned long epoch = secsSince1900 - seventyYears;
    // print Unix time:
    Serial.println(epoch);


    // print the hour, minute and second:
    Serial.print("The UTC time is ");       // UTC is the time at Greenwich Meridian (GMT)
    Serial.print((epoch  % 86400L) / 3600); // print the hour (86400 equals secs per day)
    Serial.print(':');
    if (((epoch % 3600) / 60) < 10) {
      // In the first 10 minutes of each hour, we'll want a leading '0'
      Serial.print('0');
    }
    Serial.print((epoch  % 3600) / 60); // print the minute (3600 equals secs per minute)
    Serial.print(':');
    if ((epoch % 60) < 10) {
      // In the first 10 seconds of each minute, we'll want a leading '0'
      Serial.print('0');
    }
    Serial.println(epoch % 60); // print the second
  }
  // wait ten seconds before asking for the time again
  delay(10000);
}

// send an NTP request to the time server at the given address
unsigned long sendNTPpacket(IPAddress& address) {
  //Serial.println("1");
  // set all bytes in the buffer to 0
  memset(packetBuffer, 0, NTP_PACKET_SIZE);
  // Initialize values needed to form NTP request
  // (see URL above for details on the packets)
  //Serial.println("2");
  packetBuffer[0] = 0b11100011;   // LI, Version, Mode
  packetBuffer[1] = 0;     // Stratum, or type of clock
  packetBuffer[2] = 6;     // Polling Interval
  packetBuffer[3] = 0xEC;  // Peer Clock Precision
  // 8 bytes of zero for Root Delay & Root Dispersion
  packetBuffer[12]  = 49;
  packetBuffer[13]  = 0x4E;
  packetBuffer[14]  = 49;
  packetBuffer[15]  = 52;

  //Serial.println("3");

  // all NTP fields have been given values, now
  // you can send a packet requesting a timestamp:
  Udp.beginPacket(address, 123); //NTP requests are to port 123
  //Serial.println("4");
  Udp.write(packetBuffer, NTP_PACKET_SIZE);
  //Serial.println("5");
  Udp.endPacket();
  //Serial.println("6");
}


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] (//www.arduino.cc/en/Tutorial/UdpNTPClient?action=sourceblock&num=1)

See Also:

- WiFi library (//www.arduino.cc/en/Reference/WiFi) – Your reference for the WiFi Library.
- WiFi Shield (//www.arduino.cc/en/Main/ArduinoWiFiShield) – Product details for the retired WiFi Shield.
- Getting started (//www.arduino.cc/en/Guide/ArduinoWiFiShield) – Getting started with the retired WiFi Shield.
- Connect No Encryption (//www.arduino.cc/en/Tutorial/ConnectNoEncryption) - Demonstrates how to connect to an open network.
- Connect With WEP (//www.arduino.cc/en/Tutorial/ConnectWithWEP) - Demonstrates how to connect to a network that is encrypted with WEP.
- Connect With WPA (//www.arduino.cc/en/Tutorial/ConnectWithWPA) - Demonstrates how to connect to a network that is encrypted with WPA2 Personal.
- Scan Networks (//www.arduino.cc/en/Tutorial/ScanNetworks) - Displays all WiFi networks in range.
- Simple Web Server WiFi (//www.arduino.cc/en/Tutorial/SimpleWebServerWiFi) – Turn on and off an LED accessing this simple Web Server.
- WiFi Chat Server (//www.arduino.cc/en/Tutorial/WiFiChatServer) - Set up a simple chat server.
- WiFi Web Client (//www.arduino.cc/en/Tutorial/WiFiWebClient) - Connect to a remote webserver.
- WiFi Web Client Repeating (//www.arduino.cc/en/Tutorial/WiFiWebClientRepeating) - Repeatedly make HTTP calls to a server.
- WiFi Web Server (//www.arduino.cc/en/Tutorial/WiFiWebServer) - Serve a webpage from the WiFi shield with Analog Input values.
- WiFi Send Receive UDP String (//www.arduino.cc/en/Tutorial/WiFiUdpSendReceiveString) - Send and receive a UDP string.

Last revision 2015/09/17 by SM

Share

(https://twitter.com/arduino) (https://www.facebook.com/official.arduino) (https://plus.google.com/+Arduino)

(https://www.flickr.com/photos/arduino_cc) (https://youtube.com/arduinoteam)