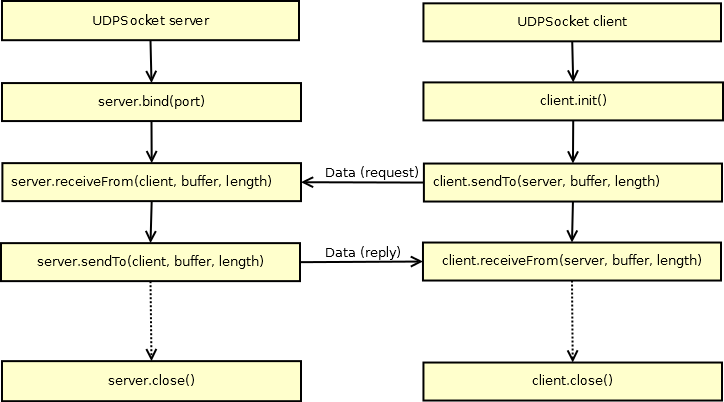
**2016年1月15日 星期五**

wifi NTP (Network Time Protocol) 校時

[Ameba 開發板系列介紹](http://njiot.blogspot.tw/2015/07/arduino-ameba.html)  
  
  
在 Arduino 官網, 有一個 NTP 校時 的 範例.  
  
<https://www.arduino.cc/en/Tutorial/UdpNTPClient>  
  
他走的是 UDP protocol.  
  
所以在個人實驗版 1.0.10, 從 ARM mbed 把 UDP porting 進來, 並且設定成 non-blocking mode.  
  
Ref: <https://developer.mbed.org/handbook/Socket>  
  
  
  
  
  
目前這隻校時程式, 是連接到 server 129.6.15.28; // time.nist.gov NTP server,

port 123 是 NTP protocol 定義的校時 port.

範例程式. :[WiFiUdpNtpClient](https://github.com/neojou/arduino-ameba/blob/master/hardware/libraries/WiFi/examples/WiFiUdpNtpClient/WiFiUdpNtpClient.ino)

     ( 雖然我是在 Ameba 上跑, 其他 Arduino 開發板也可使用 )  
  
     可以從 Wifi 範例拉進來. 記得要改 ssid , pass

             連 沒有密碼的 AP : WiFi.begin(ssid)

             連  有密碼的 AP : WiFi.begin(ssid, pass)

|  |
| --- |
|  |
| /\* |
|  |  |
|  | 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 <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); |
|  | //Serial.println(Udp.parsePacket()); |
|  | 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("sendNTPPacket"); |
|  | // 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"); |
|  | } |
|  |  |

程式說明.

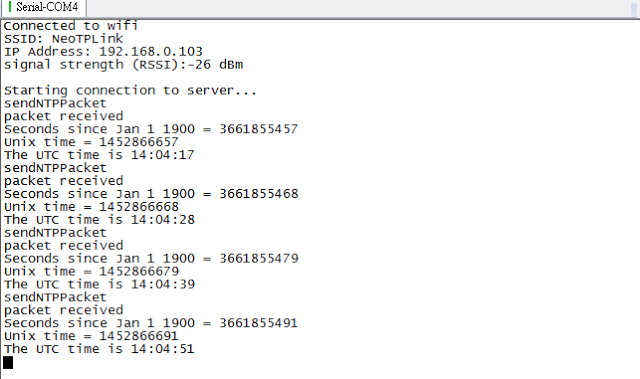
    wifi 連線部分請參考其他範例.  
  
    Arduino 有個  WiFiUDP 物件 Udp  
 WiFiUDP Udp;  
  
   begin localPort , 可以發現在 ARM mbed, 當 client 時, 並不需要  localPort, 可以隨便設一個.  
      Udp.begin(localPort);   
  
   sendNTPPacket() - 製作 NTP 格式封包, 並以 UDP 傳送出去  
  
      beginPacket(address, 123) : address 是 NTP server ip, 123 是 port number  
      write() : 寫入資料  
      endPacket() : 送出

|  |
| --- |
|  |
|  | // 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"); |

     parsePacket() 其實她只是看有沒有讀到資料, 並不會真的處理資料  
    
 Udp.parsePacket()  
  
      
     read() : 把資料讀到 packetBuffer

|  |
| --- |
|  |
| Udp.read(packetBuffer, NTP\_PACKET\_SIZE); // read the packet into the buffer |
|  |

執行結果 :

[](http://4.bp.blogspot.com/-DhTJjYXlxXw/Vpj8wyzShvI/AAAAAAAAEtg/aTDhCKTv8po/s1600/NTPtime.png)

時區不大對呀, 下次來接個 GPS 同時換算時區..

---  
另外有個 NTP pool project : http://www.pool.ntp.org/en/  
  
可以用 pool.ntp.org 來對時.