

# 以太網路

## 一、86Duino EduCake 網路介紹

86Duino EduCake 開發板 是一款 x86 架構的開源(Open Source)微電腦學習機，內部採用高性能 32 位 x86 相容的處理器 Vortex86EX，可以相容並執行 Arduino 的程式，特點是內建麵包板，使用者不需經由焊接過程，即可快速將許多電子元件、感測器及週邊配件加以連接或置換並進行電子實驗。其內建的特殊電路保護設計，能防止錯誤操作而導致燒毀 I/O 接腳。除此之外，EduCake 開發板 外殼是由堅固的金屬和麵包板組合而成，重要的電子零件皆被包覆其內，周圍留下常用的 I/O 接口，使得 EduCake 開發板 不容易受到外力破壞，適合讓使用 Arduino 、微電腦及嵌入式系統的初學者、設計師、業餘愛好者、任何有興趣的人，打造自己專屬的電子互動裝置。

86Duino EduCake 開發板不需要像 Arduino 開發板，需要外加如 W5100 網路擴充板(曹永忠, 許智誠, & 蔡英德, 2015a, 2015c, 2015d, 2015e)才能運行在網際網路，只要用內建的以太網路模組，一樣可以將 TCP/IP Protocols (TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet) 整合在 86Duino EduCake 硬體電路上，減輕了外加擴充板的麻煩與成本。

86Duino EduCake 開發板也只要使用 Ethernet Library<sup>1</sup> 便可以輕易完成連至網際網路的動作。

<sup>1</sup> 可到 Arduino.cc 的官網：<http://www.arduino.cc/en/reference/ethernet>，下載函式庫與相關範例。



圖 1 86Duino EduCake 開發板

如上圖所示，86Duino EduCake 開發板的網路在主機側邊的以太網路 RJ45 接口，如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



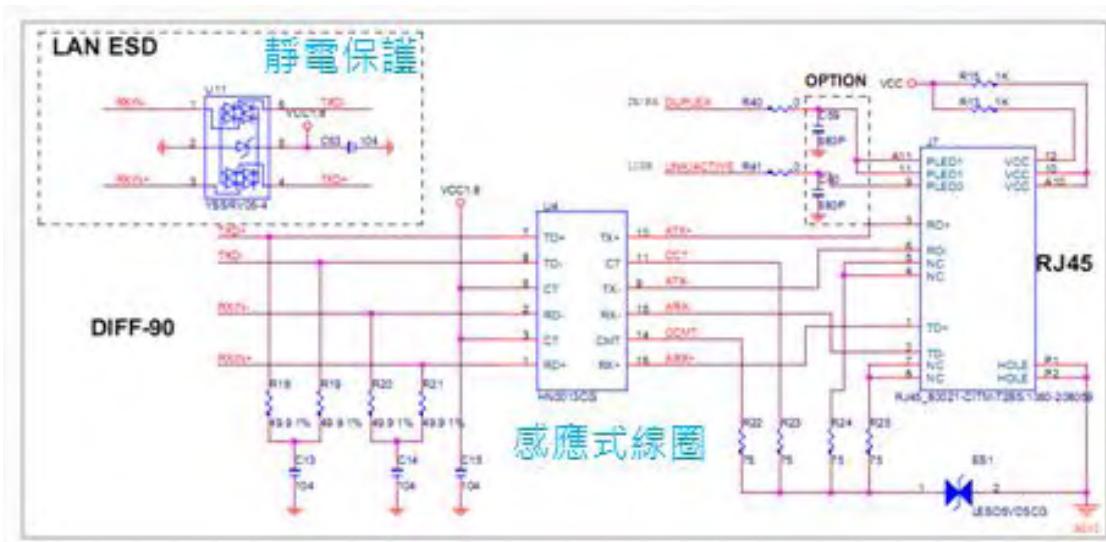
(a). EduCake 網路接法



(b). 集線器網路接法

圖 2 86Duino EduCake 開發板之網路接法示意圖

E86Duino EduCake 開發板側面提供一組網路接口(Local Area Network:LAN)，支援 10/100Mbps 傳輸速度。在網路接口(Local Area Network:LAN)上，如下圖所示，86Duino EduCake 開發板更使用了靜電保護晶片以及感應式線圈來保護內部重要的電子零件：



將 86Duino EduCake 開發板開機後，如下圖所示，網路線連接至 LAN 接口，若網路信號正常，不久後即可看到 LAN 接口的指示燈亮起，接口右邊的綠燈恆亮，左邊的橘燈閃爍：



圖 3 86Duino EduCake 開發板網路燈號示意圖

## 二、簡單 Web Server

如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 4 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之 WebServer 測試程式，我們就可以讓 86Duino EduCake 開發板變成一台簡易的網頁伺服器運作，並可以將 Analog Port0 ~Port5 的狀態顯示在流覽器畫面。

表 1 WebServer 測試程式(WebServer)

```
#include <SPI.h>
#include <Ethernet.h>      // 使用網路必需要用的

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
    0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
}; //使用這個 MAC ADDRESS 當做這台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200); //預設的網路 IP 位址，讀者可以自行更改為家中可用的位址
IPAddress dnServer(168, 95, 1, 1); //預設的 DNS 伺服器，本書為 Hinet 的 DNS 伺服器，讀者可以自行更改為家中可用的 DNS 伺服器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //預設的閘道器的位址(就是 Router 或 AP 的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家中可用的閘道器的位址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子網路遮罩，本書為 Class C

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80); //宣告伺服器並使用 Port 80 為通訊埠

void setup() {
    // Open serial communications and wait for port to open:
    Serial.begin(9600); //宣告監控畫面的通訊速率
    while (!Serial) {
        ; // wait for serial port to connect. Needed for Leonardo only
    }
}
```

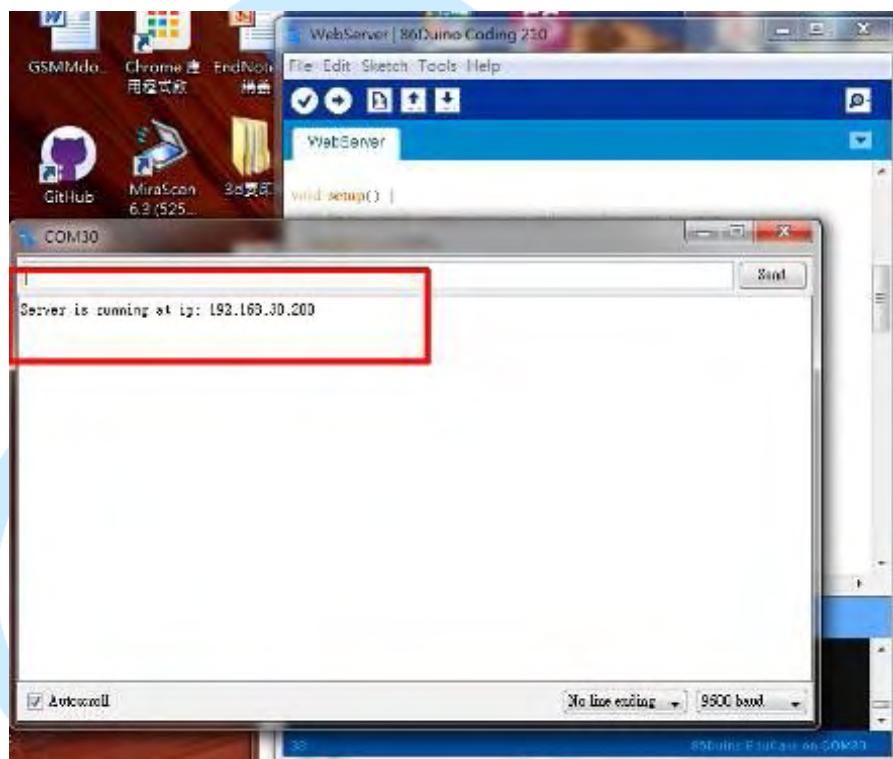
```
// start the Ethernet connection and the server:  
Ethernet.begin(mac, ip, dnServer, gateway, subnet);      //使用上述宣告  
的網路設定來啟動 WEB 伺服器  
server.begin();          //開啟 Web 伺服器運作  
Serial.print("Server is running at ip: ");           //印出伺服器資訊  
Serial.println(Ethernet.localIP());        //印出伺服器 IP 位址  
}  
  
void loop() {  
    // listen for incoming clients  
    EthernetClient client = server.available(); //如果有人連入 WEB 伺服器，啟  
動一個 Client 連線端  
    if (client) { //如果啟動一個 Client 連線端成功  
        Serial.println("New Client");           //印出"新連線"  
        // an http request ends with a blank line  
        boolean currentLineIsBlank = true;       //  
        while (client.connected()) {           //如果連線成功  
            if (client.available()) {           //如果連線的 Client 連線端有送資料過來  
                char c = client.read();         //讀入連線的 Client 連線端資料到變數  
                c  
                Serial.write(c);             //印出變數 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) { //如果變數 c 是換行資料  
                   而且目前資料為空  
                    // send a standard http response header  
                    client.println("HTTP/1.1 200 OK"); //送出 Http 標頭資料(固  
定要送)  
                    client.println("Content-Type: text/html"); //送出 Http 標頭資  
料(固定要送)  
                    client.println("Connection: close"); // the connection will be
```

```
closed after completion of the response      //送出 Http 標頭資料(固定要送)
client.println("Refresh: 5");    // refresh the page automatically
every 5 sec      //送出 Http 標頭資料(固定要送)
client.println();      //送出 Http 標頭資料(固定要送)
client.println("<!DOCTYPE HTML>");      //送出 Http 標頭資料(固定要送)
client.println("<html>");      //送出 Http 標頭資料(固定要送)
// output the value of each analog input pin
for (int analogChannel = 0; analogChannel < 6; analogChannel++) { //將類比腳位 0 到 5 得資料讀出來
    int sensorReading = analogRead(analogChannel);
    client.print("Analog Input ");      //送出畫面資料
    client.print(analogChannel);      //送出第幾個類比通訊埠
    client.print(" is ");      //送出畫面資料
    client.print(sensorReading);      //送出讀出該類比通訊埠 的
    client.println("<br />");      //送出網頁換行鍵
}
client.println("</html>");      //送出網頁 Tag 資料(固定要送)
break;
}
if (c == '\n') {      //新的一行
    // you're starting a new line
    currentLineIsBlank = true;
}
else if (c != '\r') {      //如果不是換行鍵
    // you've gotten a character on the current line
    currentLineIsBlank = false;
}
}
// give the web browser time to receive the data
```

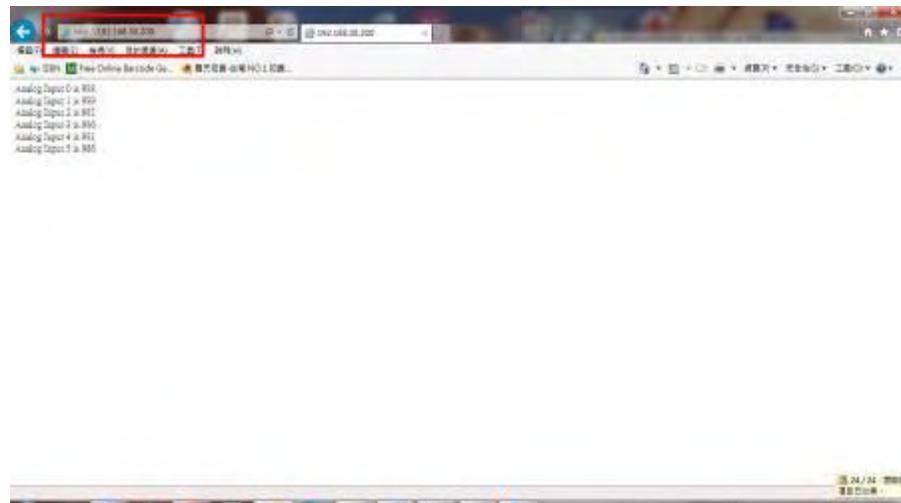
資料

```
delay(1); //延遲一毫秒  
// close the connection:  
client.stop(); //關閉連線的 Client 連線端(必需要關閉，不然會掛掉)  
Serial.println("Client Disconnected"); //送出連線斷線  
}  
}
```

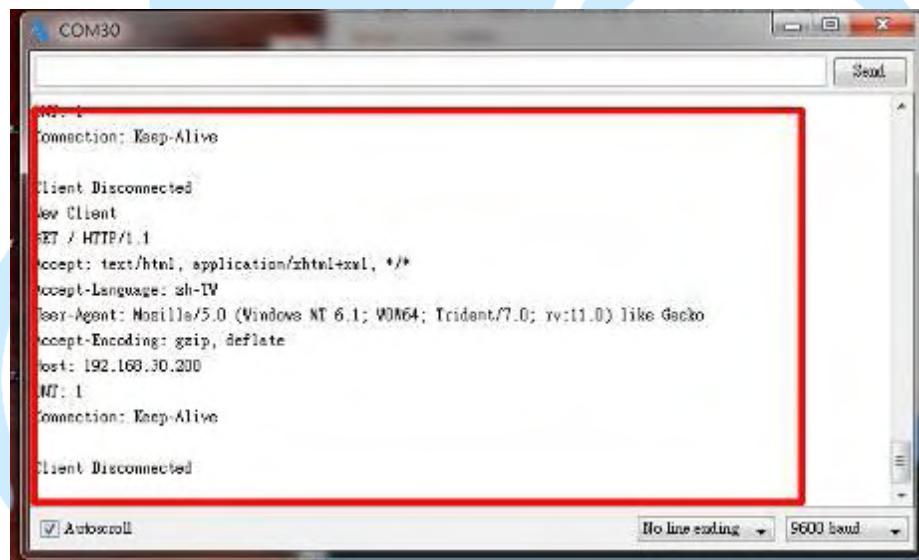
如下圖所示，讀者可以看到本次實驗- WebServer 測試程式結果畫面。



(a). WebServer 測試程式開始畫面



(b). 啟動瀏覽器連接



(c). WebServer 連線後結果畫面

圖 5 WebServer 測試程式結果畫面

### 三、使用 DHCP 架設 Web Server

如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 6 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之 WebServer 測試程式一，我們就可以讓 86Duino EduCake 開發板變成一台簡易的網頁伺服器運作，所不同的是 86Duino EduCake 開發板所成為的 WebServer 所取得的 IP 位址不是自訂，而是從 DHCP 伺服器取得的 IP 位址，並可以將 Analog Port0 ~Port5 的狀態顯示在流覽器畫面。

表 2 WebServer 測試程式一(WebServer\_Dhcp)

```
#include <SPI.h>
#include <Ethernet.h>      // 使用網路必需要用的

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
```

```
0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
}; //使用這個 MAC ADDRESS 當做這台 eduke 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200); //預設的網路 IP 位址，讀者可以自行更改為
家中可用的位址

IPAddress dnServer(168, 95, 1, 1); //預設的 DNS 伺服器，本書為 Hinet
的 DNS 伺服器，讀者可以自行更改為家中可用的 DNS 伺服器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //預設的閘道器的位址(就是 Router
或 AP 的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家
中可用的閘道器的位址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子網路遮罩，本書為 Class C

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80); //宣告伺服器並使用 Port 80 為通訊埠

void setup() {
    // Open serial communications and wait for port to open:
    Serial.begin(9600); //宣告監控畫面的通訊速率
    Serial.println("Now Program Start");

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

    // 啟用 Ethernet 連線，預設會以 DHCP 取得 IP 位址
    if (Ethernet.begin(mac) == 0) {
        Serial.println("I can't get any IP address from DHCP Server");
    }
}
```

```
// 無法取得 IP 位址，不做任何事情
for(;;)
;
}

// 輸出 IP 位址
Serial.print("Default IP Address is : "); //印出伺服器資訊
Serial.println(ip); //印出伺服器 IP 位址

// Ethernet.begin(mac, ip, dnServer, gateway, subnet);
Ethernet.begin(mac); // 如果要用 DHCP 伺服器取得 ip, 必需要這樣寫

server.begin(); //開啟 Web 伺服器運作
Serial.print("Server is running at ip: "); //印出伺服器資訊
Serial.print(Ethernet.localIP()); //印出伺服器資訊
Serial.print("from DHCP Server \n"); //印出伺服器資訊

void loop() {
// listen for incoming clients
EthernetClient client = server.available(); //如果有人連入 WEB 伺服
器，啟動一個 Client 連線端

if (client) { //如果啟動一個 Client 連線端成功
Serial.println("New Client"); //印出"新連線"
// an http request ends with a blank line
boolean currentLineIsBlank = true;
while (client.connected()) { //如果連線成功
if (client.available()) { //如果連線的 Client 連線端有送資料過來
char c = client.read(); //如果連線的 Client 連線端有送資料過來
Serial.write(c); //印出變數 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,
}
```

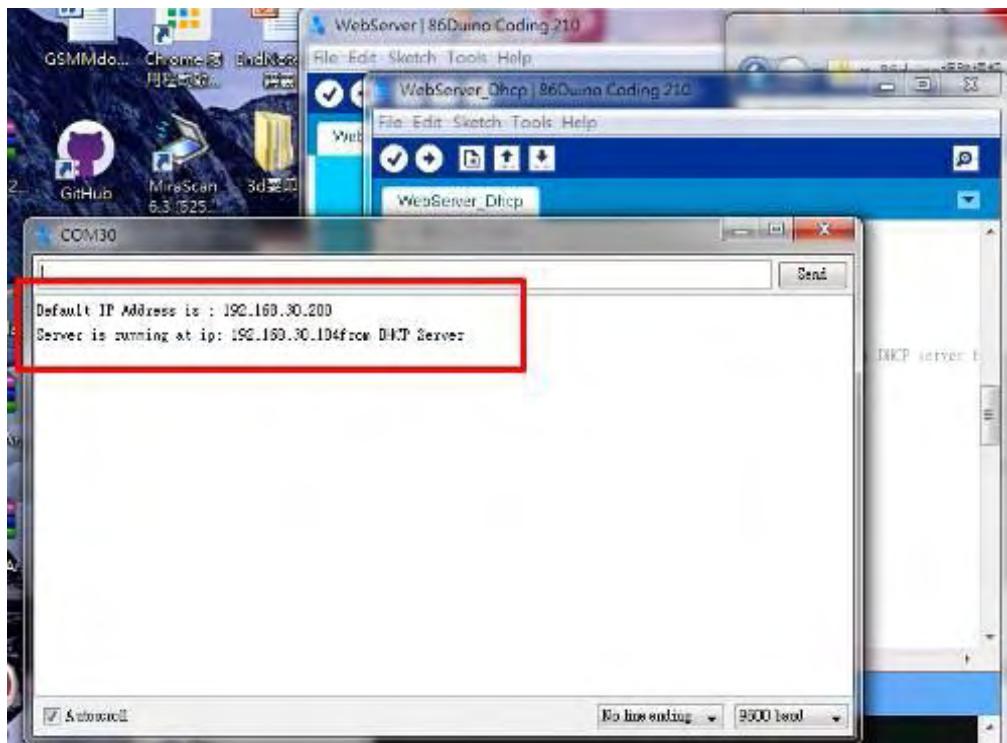
```
// so you can send a reply
if (c == '\n' && currentLineIsBlank) {      //如果變數 c 是換行資料而且
目前資料為空
    // send a standard http response header
    client.println("HTTP/1.1 200 OK");      //送出 Http 標頭資料(固定要
送)
    client.println("Content-Type: text/html");      //送出 Http 標頭資
料(固定要送)
    client.println("Connection: close"); // the connection will be closed
after completion of the response //送出 Http 標頭資料(固定要送)
    client.println("Refresh: 5"); // refresh the page automatically every
5 sec //送出 Http 標頭資料(固定要送)
    client.println();      //送出 Http 標頭資料(固定要送)
    client.println("<!DOCTYPE HTML>");      //送出 Http 標頭資料(固
定要送)
    client.println("<html>");      //送出 Http 標頭資料(固定要送)
    // output the value of each analog input pin
    for (int analogChannel = 0; analogChannel < 6; analogChannel++)
{   //將類比腳位 0 到 5 得資料讀出來
    int sensorReading = analogRead(analogChannel);
    client.print("Analog Input ");      //送出畫面資料
    client.print(analogChannel);      //送出第幾個類比通訊埠
    client.print(" is ");      //送出畫面資料
    client.print(sensorReading);      //送出讀出該類比通訊埠 的
資料
    client.println("<br />");      //送出網頁換行鍵
}
client.println("</html>");      //送出網頁 Tag 資料(固定要送)
break;
}
```

```
if (c == '\n') {      //新的一行
    // you're starting a new line
    currentLineIsBlank = true;
}
else if (c != '\r') { //如果不是換行鍵
    // you've gotten a character on the current line
    currentLineIsBlank = false;
}
}
}

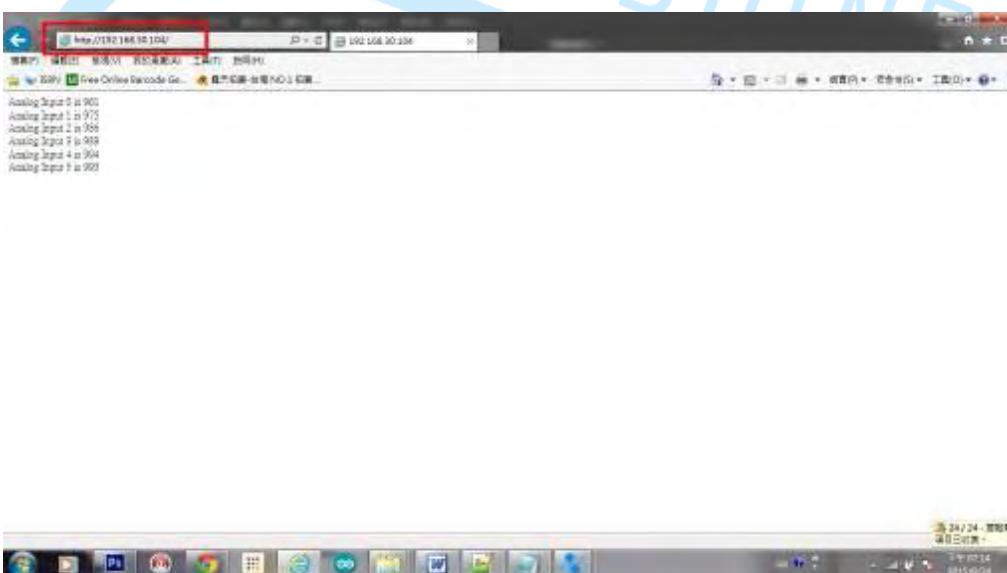
// give the web browser time to receive the data
delay(1); //延遲一毫秒

// close the connection:
client.stop(); //關閉連線的 Client 連線端(必需要關閉，不然會掛掉)
Serial.println("Client Disconnected"); //送出連線斷線
}
```

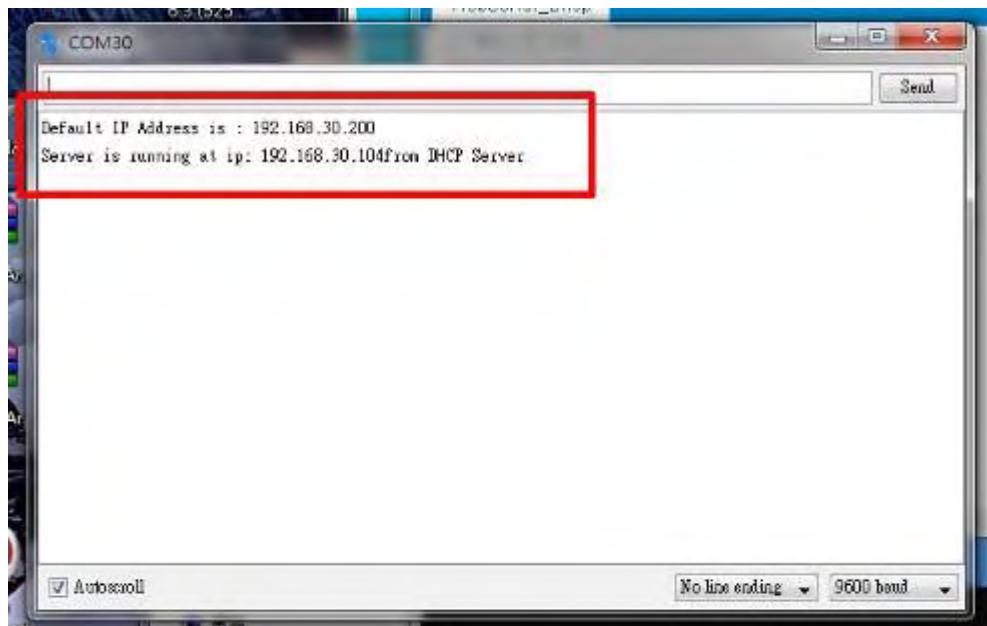
如下圖所示，讀者可以看到本次實驗- WebServer 測試程式一結果畫面。



(a). WebServer 測試程式開始畫面



(b). 啟動瀏覽器連接



(c). WebServer 連線後結果畫面

圖 7 WebServer 測試程式一結果畫面結果畫面

## 四、Telnet 用戶端程式

首先，如下圖所示，我們使用一般 Cate 5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 8 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之 Telnet 用戶端程式測試程式，我們就可以讓 86Duino EduCake 開發板變成一台簡易的 Telnet 用戶端工作站。

表 3 Telnet 用戶端程式測試程式(TelnetClient)

```
#include <SPI.h>
#include <Ethernet.h>          // 使用網路必需要用的

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
    0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
};      //使用這個 MAC ADDRESS 當做這台 educake 的 MAC ADDRESS

IPAddress ip(192, 168, 30, 200);      //預設的網路 IP 位址，讀者可以自行更改為家中
可用的位址

IPAddress dnServer(168, 95, 1, 1);      //預設的 DNS 伺服器，本書為 Hinet 的 DNS 伺
服器，讀者可以自行更改為家中可用的 DNS 伺服器

// the router's gateway address:
IPAddress gateway(192, 168, 30, 254);      //預設的閘道器的位址(就是 Router 或 AP
的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家中可用的閘
道器的位址

// the subnet:
IPAddress subnet(255, 255, 255, 0);      //子網路遮罩，本書為 Class C

// Enter the IP address of the server you're connecting to:
IPAddress server(140, 112, 172, 11);      //台大 PTT 網址

// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 23 is default for telnet;
// if you're using Processing's ChatServer, use port 10002):
EthernetClient client;      //宣告一個 TCP/IP 的 Client 物件

void setup() {
    // start the Ethernet connection:
```

```
Ethernet.begin(mac, ip, dnServer, gateway, subnet); // 啟用 Ethernet 連線
```

```
// Open serial communications and wait for port to open:
```

```
Serial.begin(9600); //宣告監控畫面的通訊速率
```

```
while (!Serial) {
```

```
; // wait for serial port to connect. Needed for Leonardo only
```

```
}
```

```
// give the Ethernet shield a second to initialize:
```

```
delay(1000); //延持 1000 毫秒
```

```
Serial.println("connecting..."); //印出連線中字句....
```

```
// if you get a connection, report back via serial:
```

```
if (client.connect(server, 23)) { //如果 Client 物件連到 server(140, 112, 172, 11);
```

```
台大 PTT 網址 成功的話
```

```
Serial.println("connected"); //印出連線成功字句....
```

```
}
```

```
else {
```

```
// if you didn't get a connection to the server:
```

```
Serial.println("connection failed"); //印出連線失敗字句....
```

```
}
```

```
}
```

```
void loop()
```

```
{
```

```
// if there are incoming bytes available
```

```
// from the server, read them and print them:
```

```
if (client.available()) { //如果 Client 物件有東西回傳
```

```
char c = client.read(); //讀出 Client 物件，並存到變數 c
```

```
Serial.print(c); //印出變數 c //
```

```
}
```

```

// as long as there are bytes in the serial queue,
// read them and send them out the socket if it's open:
while (Serial.available() > 0) {      //如果我們從監控畫面輸入文字，
    char inChar = Serial.read();      //讀出監控畫面輸入文字，並存到變數 c
    if (client.connected()) {        //如果連線中
        client.print(inChar);        //將輸入文字傳到 Client 物件
    }
}

// if the server's disconnected, stop the client:
if (!client.connected()) {          //如果 Client 物件已斷線
    Serial.println();
    Serial.println("disconnecting."); //印出已斷線
    client.stop();                 //Client 物件關閉
    // do nothing:
    while (true);                //永久迴圈，不再做事
}
}

```

如下圖所示，讀者可以看到本次實驗結果畫面。

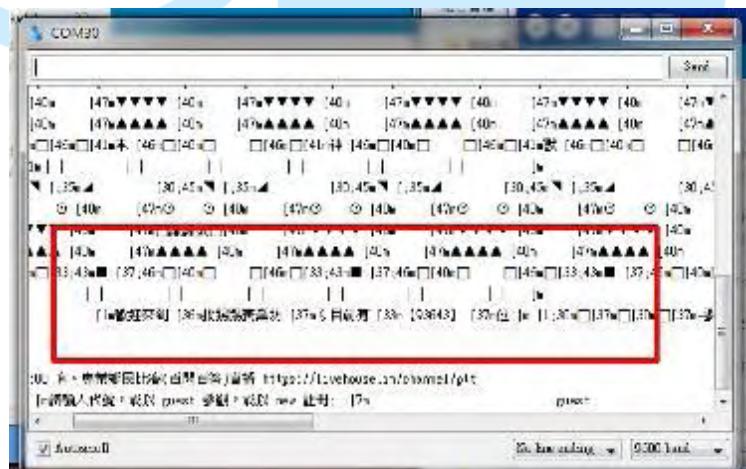
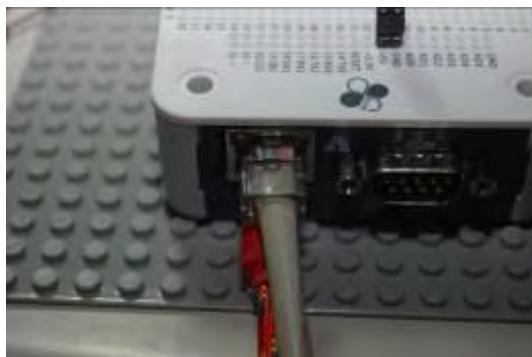


圖 9 Telnet 用戶端程式測試程式結果畫面

## 五、文字型 Browser 用戶端程式

如下圖所示，我們使用一般 Cate 5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 10 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之 Telnet 用戶端程式測試程式，我們就可以讓 86Duino EduCake 開發板變成一台簡易的網瀏覽器文字介面處理機運作了，未來我們還可以透過文字過濾的方式，來搜尋想要的資訊了。

表 4 文字型 Browser 用戶端程式(WebClient)

```
#include <SPI.h>
#include <Ethernet.h>      // 使用網路必需要用的

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = {
    0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
};      //使用這個 MAC ADDRESS 當做這台 educake 的 MAC ADDRESS
```

```
IPAddress ip(192, 168, 30, 200);      //預設的網路 IP 位址，讀者可以自行更改  
為家中可用的位址  
  
IPAddress dnServer(168, 95, 1, 1);    //預設的 DNS 伺服器，本書為 Hinet 的 DNS  
伺服器，讀者可以自行更改為家中可用的 DNS 伺服器  
  
// the router's gateway address:  
  
IPAddress gateway(192, 168, 30, 254);   //預設的閘道器的位址(就是 Router  
或 AP 的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家  
中可用的閘道器的位址  
  
// the subnet:  
  
IPAddress subnet(255, 255, 255, 0);     //子網路遮罩，本書為 Class C  
  
  
// if you don't want to use DNS (and reduce your sketch size)  
// use the numeric IP instead of the name for the server:  
  
//IPAddress server(74,125,232,128); // numeric IP for Google (no DNS)  
//www.google.com 的網址  
  
char server[] = "www.google.com";       // name address for Google (using DNS)  
  
  
// Initialize the Ethernet client library  
// with the IP address and port of the server  
// that you want to connect to (port 80 is default for HTTP):  
EthernetClient client;  
  
  
void setup() {  
    // Open serial communications and wait for port to open:  
    Serial.begin(9600);      //宣告監控畫面的通訊速率  
    while (!Serial) {  
        ; // wait for serial port to connect. Needed for Leonardo only  
    }  
  
    // start the Ethernet connection:  
    if (Ethernet.begin(mac) == 0) { //要求 DHCP 伺服器發放 ip 位址等資訊
```

```
Serial.println("Failed to configure Ethernet using DHCP"); //要求 DHCP  
伺服器發放 ip 位址失敗，發出警告訊息  
// no point in carrying on, so do nothing forevermore:  
// try to conigure using IP address instead of DHCP:  
Ethernet.begin(mac, ip, dnServer, gateway, subnet); //使用系統預設的  
ip 位址  
  
}  
// give the Ethernet shield a second to initialize:  
delay(1000); //延持 1000 毫秒  
Serial.println("connecting..."); //印出連線中字句....  
  
// if you get a connection, report back via serial:  
if(client.connect(server, 80)) { //如果 Client 物件連到 server(74,125,232,128) www.google.com 的網址 成功的話  
    Serial.println("connected"); //印出連線成功字句....  
    // Make a HTTP request:  
    //以下是固定 http 通訊 protocol  
    client.println("GET /search?q=arduino HTTP/1.1");  
    client.println("Host: www.google.com"); //以下是告知連到  
www.google.com  
    client.println("Connection: close"); //連完後關閉  
    client.println();  
}  
else {  
    // if you didn't get a connection to the server:  
    Serial.println("connection failed"); // 連接 www.google.com 失敗，告訴  
使用者連接失敗  
}
```

```

void loop()
{
    // if there are incoming bytes available
    // from the server, read them and print them:
    if (client.available()) {          //如果 Client 物件有東西回傳
        char c = client.read();        //讀出 Client 物件，並存到變數 c
        Serial.print(c);               //印出變數 c
    }

    // if the server's disconnected, stop the client:
    if (!client.connected()) {         //如果 Client 已經關閉連線
        Serial.println();
        Serial.println("disconnecting."); //印出中斷連線成功字句....
        client.stop();                //連現中止
    }

    // do nothing forevermore:
    while (true);                   //永久迴圈中斷程式
}
}

```

如下圖所示，讀者可以看到本次實驗結果畫面。

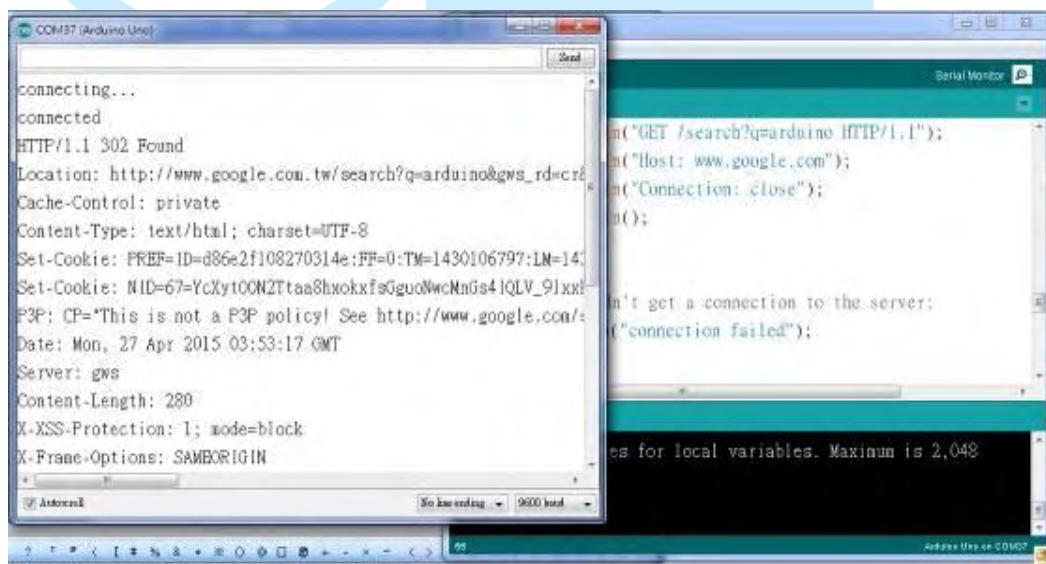


圖 11 文字型 Browser 用戶端程式結果畫面

## 六、取得網路校時時間資料

如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 12 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之網路校時測試程式，我們就可以讓 86Duino EduCake 開發板變成一台簡易的網路校時機器，取得網路校時時間。

表 5 網路校時測試程式(UdpNtpClient)

```
#include <SPI.h>
#include <Ethernet.h>      // 使用網路必需要用的
#include <EthernetUdp.h>    // 使用網路 UDP 通訊協定必需要用的

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = {
  0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
};      //使用這個 MAC ADDRESS 當做這台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200);      //預設的網路 IP 位址，讀者可以自行更改為
```

## 家中可用的位址

```
IPAddress dnServer(168, 95, 1, 1); //預設的 DNS 伺服器, 本書為 Hinet 的 DNS  
伺服器, 讀者可以自行更改為家中可用的 DNS 伺服器
```

```
// the router's gateway address:
```

```
IPAddress gateway(192, 168, 30, 254); //預設的閘道器的位址(就是 Router 或  
AP 的網址), 本書為作者寫書環境的閘道器的位址, 讀者可以自行更改為家中可  
用的閘道器的位址
```

```
// the subnet:
```

```
IPAddress subnet(255, 255, 255, 0); //子網路遮罩, 本書為 Class C
```

```
unsigned int localPort = 8888; //要校時必須設定一個 local port 來給 UDP 使  
用
```

```
// local port to listen for UDP packets
```

```
char timeServer[] = "time.nist.gov"; // time.nist.gov NTP server 校時伺服器
```

```
const int NTP_PACKET_SIZE = 48; // NTP time stamp is in the first 48 bytes of the  
message 封包大小 48 位元組
```

```
byte packetBuffer[ NTP_PACKET_SIZE]; //buffer to hold incoming and outgoing  
packets 收入封包資料變數
```

```
// A UDP instance to let us send and receive packets over UDP
```

```
EthernetUDP Udp; //宣告一個 UDP 網路物件
```

```
void setup()
```

```
{
```

```
    // Open serial communications and wait for port to open:
```

```
    Serial.begin(9600); //宣告監控畫面的通訊速率
```

```
    while (!Serial) {
```

```
// wait for serial port to connect. Needed for Leonardo only
}

// start Ethernet and UDP
if (Ethernet.begin(mac) == 0) {      //要求 DHCP 啟用器發放 ip 位址等資訊
    Serial.println("Failed to configure Ethernet using DHCP"); //要求 DHCP 啟用器發放 ip 位址失敗，發出警告訊息
    // no point in carrying on, so do nothing forevermore:
    Ethernet.begin(mac, ip, dnServer, gateway, subnet); //使用系統預設的 ip 位址
}

}

Udp.begin(localPort); //啟動 UDP
}

void loop()
{
    sendNTPpacket(timeServer); // send an NTP packet to a time server 傳送封包到 time.nist.gov NTP server 校時伺服器

    // wait to see if a reply is available
    delay(1000); //延遲 1000 毫秒
    if ( Udp.parsePacket() ) { //如果收到 time.nist.gov NTP server 校時伺服器 傳送封包
        // We've received a packet, read the data from it
        Udp.read(packetBuffer, NTP_PACKET_SIZE); // read the packet into the buffer //讀取封包

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

```
unsigned long highWord = word(packetBuffer[40], packetBuffer[41]); //  
將高位元組轉到 highWord 變數  
  
unsigned long lowWord = word(packetBuffer[42], packetBuffer[43]); //將低  
位元組轉到 lowWord 變數  
  
// 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; //將高  
低位元組合為一個變數  
  
Serial.print("Seconds since Jan 1 1900 = "); //印出時間  
Serial.println(secsSince1900);  
  
// now convert NTP time into everyday time:  
Serial.print("Unix time = "); //印出 Unix time =  
// Unix time starts on Jan 1 1970. In seconds, that's 2208988800:  
const unsigned long seventyYears = 2208988800UL; //設定 unix 系統初始  
時間  
// subtract seventy years:  
unsigned long epoch = secsSince1900 - seventyYears; //計算取得時間差  
// print Unix time:  
Serial.println(epoch); //印出時間差  
  
// print the hour, minute and second:  
Serial.print("The UTC time is "); // UTC is the time at Greenwich Me-  
ridian (GMT)  
Serial.print((epoch % 86400L) / 3600); // print the hour (86400 equals secs per  
day) 印出共有多少天  
Serial.print(':');  
if ( ((epoch % 3600) / 60) < 10 ) {  
    // In the first 10 minutes of each hour, we'll want a leading '0' 印出多少小時  
    Serial.print('0');
```

```
}

Serial.print((epoch % 3600) / 60); // print the minute (3600 equals secs per
minute) 印出多少分鐘

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 印出多少秒

}

// wait ten seconds before asking for the time again
delay(10000); //延持 10000 毫秒

}

//送出 NTP 封包函數

// send an NTP request to the time server at the given address
unsigned long sendNTPpacket(char* address) //送出 NTP 封包函數

{

    //送出 NTP 封包函數

    // set all bytes in the buffer to 0
    memset(packetBuffer, 0, NTP_PACKET_SIZE); //設定 packetBuffer

    // Initialize values needed to form NTP request
    // (see URL above for details on the packets)
    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;
```

```
// 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      利用 Port  
123 來傳送到 address 變數中的 ip 位址  
Udp.write(packetBuffer, NTP_PACKET_SIZE); //傳送 packetBuffer 變數  
資料  
Udp.endPacket(); //傳送結束  
}
```

如下圖所示，讀者可以看到本次實驗-網路校時測試程式結果畫面。

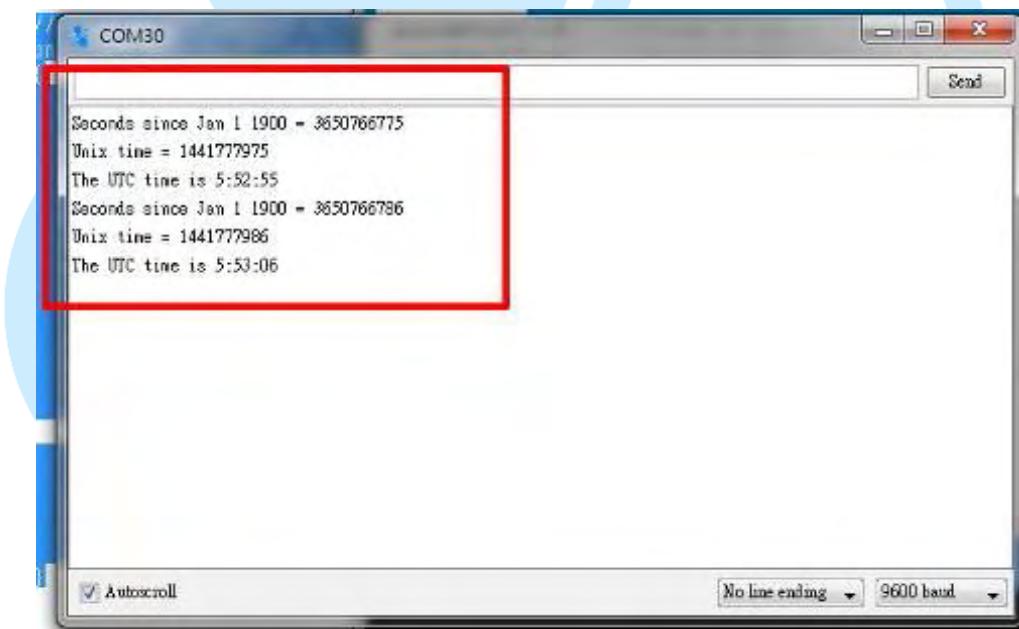
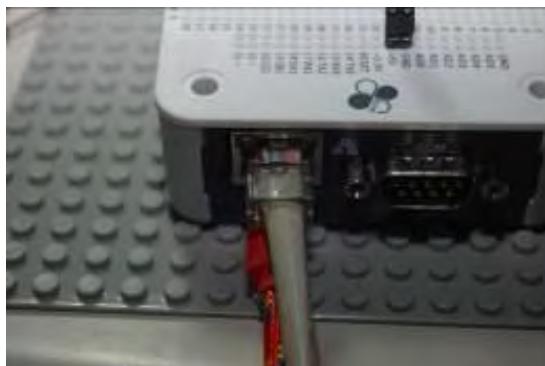


圖 13 網路校時測試程式結果畫面

## 七、Telnet 簡單聊天室

如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 14 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之 Telnet 簡單聊天室測試程式，我們就可以讓 86Duino EduCake 開發板變成一台簡易的 Telnet 簡單聊天室運作了。

表 6 Telnet 簡單聊天室測試程式(ChatServer)

```
#include <SPI.h>
#include <Ethernet.h>      // 使用網路必需要用的

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = {
    0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
};      //使用這個 MAC ADDRESS 當做這台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200);      //預設的網路 IP 位址，讀者可以自行更改為家中可用的位址
```

```
IPAddress dnServer(168, 95, 1, 1);      //預設的 DNS 伺服器，本書為 Hinet 的 DNS  
伺服器，讀者可以自行更改為家中可用的 DNS 伺服器  
  
// the router's gateway address:  
  
IPAddress gateway(192, 168, 30, 254);    //預設的閘道器的位址(就是 Router 或  
AP 的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家中可用  
的閘道器的位址  
  
// the subnet:  
  
IPAddress subnet(255, 255, 255, 0);       //子網路遮罩，本書為 Class C  
  
  
// telnet defaults to port 23  
EthernetServer server(23);                 //宣告一個以太網路伺服器物件，並使用 Port23 為  
listening 通訊埠  
  
boolean alreadyConnected = false; // whether or not the client was connected previously  
連線旗標  
  
  
void setup() {  
    // initialize the ethernet device  
    Ethernet.begin(mac, ip, gateway, subnet); //啟動以太網路  
  
    // start listening for clients  
    server.begin();           //啟動 Port23 的伺服器物件  
  
    // Open serial communications and wait for port to open:  
    Serial.begin(9600);        //宣告監控畫面的通訊速率  
  
    while (!Serial) {  
        ; // wait for serial port to connect. Needed for Leonardo only  
    }  
  
  
    Serial.print("Chat server address:");     //印出 Chat server address:  
    Serial.println(Ethernet.localIP());         //印出網址  
}
```

```
void loop() {
    // wait for a new client:
    EthernetClient client = server.available(); //產生一個以太網路用戶端網路物件

    // when the client sends the first byte, say hello:
    if (client) {      //當以太網路用戶端網路物件有人連入
        if (!alreadyConnected) { //如果是新連線
            // clear out the input buffer:
            client.flush();      //以太網路用戶端網路物件 更新緩衝區資料
            Serial.println("We have a new client"); //印出有新連線
            client.println("Hello, client!"); //送出 Hello 歡迎詞
            alreadyConnected = true; //設定已連線
        }

        if (client.available() > 0) { //如果以太網路用戶端網路物件有資料送出
            // read the bytes incoming from the client:
            char thisChar = client.read(); //讀取以太網路用戶端網路物件送出的資料到 thisChar 變數
            // echo the bytes back to the client:
            server.write(thisChar); //送出 thisChar 變數到 server 物件
            // echo the bytes to the server as well:
            Serial.write(thisChar); //印出 thisChar 變數
        }
    }
}
```

如下圖所示，讀者可以看到本次實驗- Telnet 簡單聊天室，Telnet 簡單聊天室待機畫面。

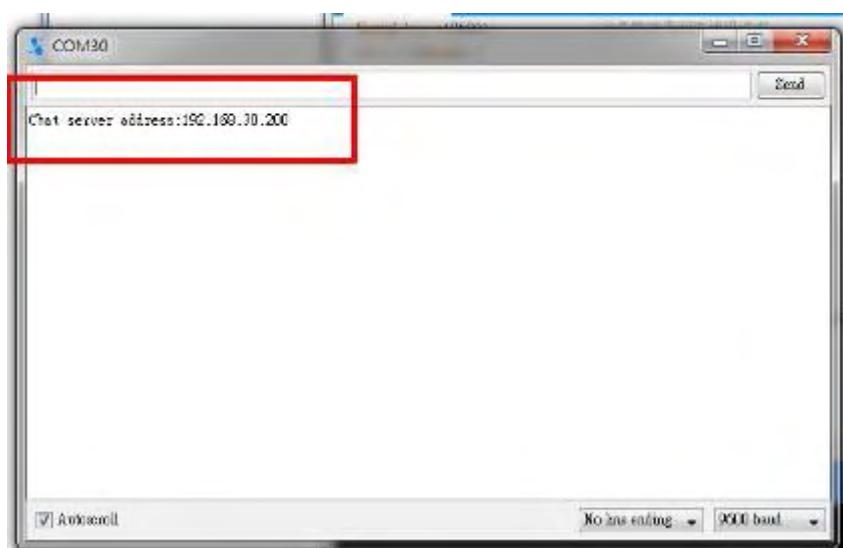


圖 15 Telnet 簡單聊天室待機畫面

如下圖所示，讀者可以看到本次實驗- Telnet 簡單聊天室，我們使用 Putty 通訊軟體，預備進行連線的畫面。

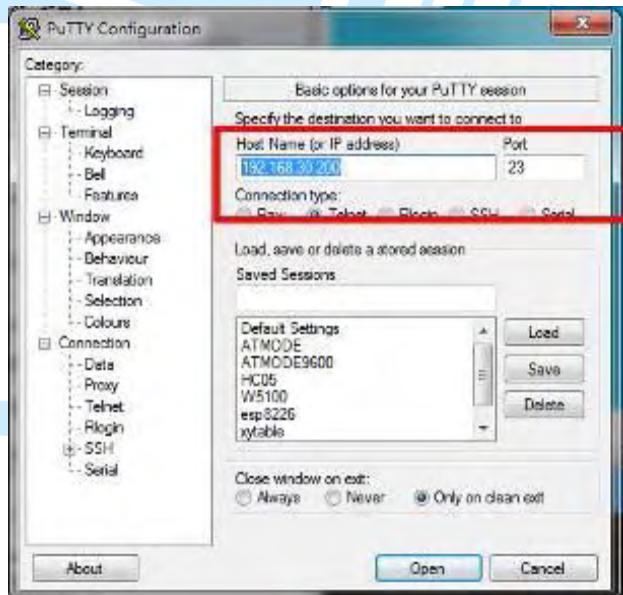


圖 16 Telnet 簡單聊天室預備進行連線的畫面

我們可以使用 putty 通訊軟體進行測試聊天室的功能，如下圖所示，為連線到簡單聊天室用戶端畫面。

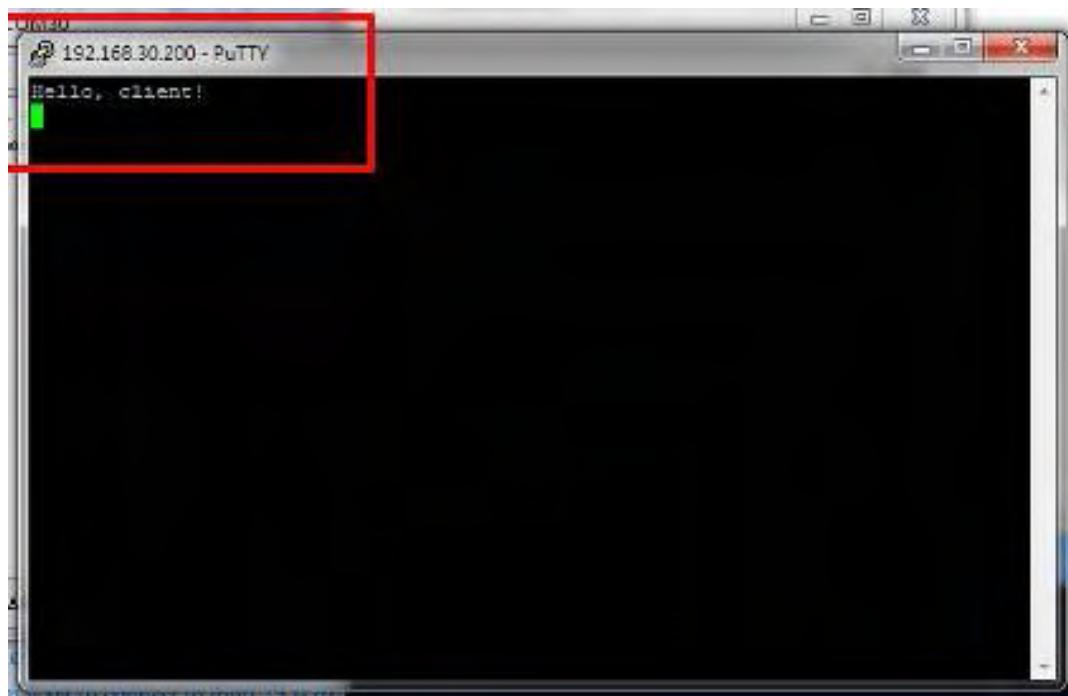


圖 17 連線到簡單聊天室用戶端畫面

如下圖所示，我們使用 Putty 通訊軟體，進行連線後，讀者可以看到本次實驗- Telnet 簡單聊天室之 Telnet 簡單聊天室用戶端連線的畫面。

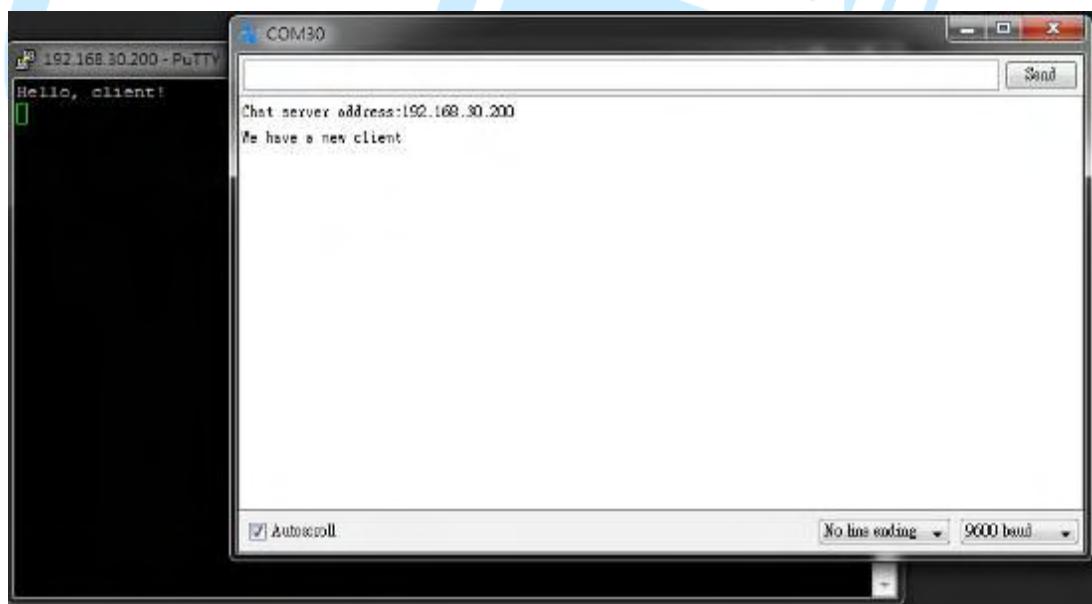


圖 18 Telnet 簡單聊天室用戶端連線的畫面

## 八、Telnet 多人版聊天室

上節中，我們已經建立一個簡單而完整的『Telnet 簡單聊天室』，但是，這個聊天室太過於簡陋，只能供一位用戶端連入聊天室，所以我們要改進這樣的問題。

首先，如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



圖 19 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之 Telnet 簡單聊天室測試程式，我們就可以讓 86Duino EduCake 開發板變成一台專業版的多人聊天室運作了。

表 7 Telnet 簡單多人版聊天室測試程式(AdvancedChatServer2)

```
#include <SPI.h>
#include <Ethernet.h>      // 使用網路必需要用的

// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the shield
byte mac[] = {
```

```
0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
};      //使用這個 MAC ADDRESS 當做這台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200);      //預設的網路 IP 位址，讀者可以自行更改為
家中可用的位址
IPAddress dnServer(168, 95, 1, 1);      //預設的 DNS 伺服器，本書為 Hinet 的 DNS
伺服器，讀者可以自行更改為家中可用的 DNS 伺服器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254);    //預設的閘道器的位址(就是 Router 或
AP 的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家中可
用的閘道器的位址
// the subnet:
IPAddress subnet(255, 255, 255, 0);      //子網路遮罩，本書為 Class C

// telnet defaults to port 23
EthernetServer server(23);      //宣告一個以太網路伺服器物件，並使用 Port23
為 listening 通訊埠
boolean alreadyConnected; // whether or not the client was connected previously 連
線旗標
boolean ConnectedFlag[10]; // whether or not the client was connected previously 連
線旗標
int connectNo = 0;      //連線號碼
EthernetClient client;      //產生一個傾聽的以太網路用戶端網路物件
EthernetClient Connectclient[10];    //產生十個以太網路用戶端網路物件

void setup() {

    // initialize the ethernet device
    Ethernet.begin(mac, ip, gateway, subnet); //啟動乙太網路連線
    // start listening for clients
    server.begin();      //啟動 Port23 乙太網路連線
```

```
// Open serial communications and wait for port to open:  
Serial.begin(9600);          //宣告監控畫面的通訊速率  
initConnectingFlag(); // 初始化連線旗標  
  
while (!Serial) {  
    ; // wait for serial port to connect. Needed for Leonardo only  
}  
  
Serial.print("Chat server address:"); //印出 Chat server address:  
Serial.println(Ethernet.localIP()); //印出網址  
DisplayConnectingStatus(); //顯示連線狀況  
}  
  
void loop() {  
    // wait for a new client:  
    client = server.available(); //產生傾聽的以太網路用戶端網路物件  
  
    // when the client sends the first byte, say hello:  
    if (client) { //當以太網路用戶端網路物件有人連入  
        connectNo = 0;  
        while (connectNo < 10) //迴圈十組連線  
        {  
            if (!ConnectedFlag[connectNo]) { //如果是新連線  
                // clear out the input buffer:  
                Connectclient[connectNo] = client;  
                Connectclient[connectNo].flush(); //以太網路用戶端網路物件  
                更新緩衝區資料  
                Serial.println("We have a new client"); //印出有新連線  
                client.println("Hello, client!"); //送出 Hello 歡迎詞  
                ConnectedFlag[connectNo] = true; //設定已連線  
                break;  
            }  
        }  
    }  
}
```

```
        }

        connectNo ++;

    }

}

connectNo = 0;

while (connectNo <10)

{

    if (!Connectclient[connectNo].connected())      //檢查是否斷線

        ConnectedFlag[connectNo] = false;           //設定連線旗標參數

=false(未連線)

    if (Connectclient[connectNo].available() > 0)

    {   //如果以太網路用戶端網路物件有資料送出

        // clear out the input buffer:

        char thisChar = Connectclient[connectNo].read();           //讀取以太網路

        用戶端網路物件送出的資料到 thisChar 變數

        // echo the bytes back to the client:

        Serial.print("Connect ");          //印出連線中

        Serial.print(connectNo);          //印出第幾號連線

        Serial.print(":");               //印出":"

        server.write(thisChar);           //送出 thisChar 變數到 server 物件

        // echo the bytes to the server as well:

        Serial.write(thisChar);           //印出 thisChar 變數

    }

    connectNo ++;

}

}

void initConnectingFlag()

{   //初始化連線旗標參數
```

```
for(connectNo=0 ; connectNo < 10 ; connectNo++)  
{  
    ConnectedFlag[connectNo] = false ;           //設定連線旗標參數=false(未連  
線)  
  
}  
}  
  
void DisplayConnectingStatus()  
{  //顯示連線狀況  
    for(connectNo=0 ; connectNo < 10 ; connectNo++)  
    {  
        if (ConnectedFlag[connectNo])           //如果連線中  
        {  
            Serial.print("Connection ") ;      //該區顯示連線中的資訊  
            Serial.print(connectNo) ;  
            Serial.print(": Connected \n") ;  
        }  
        else          //如果未連線中  
        {  
            Serial.print("Connection ") ;      //該區顯示等待連線中的資訊  
            Serial.print(connectNo) ;  
            Serial.print(": Waiting Connecting \n") ;  
        }  
  
        // Connectclient[connectNo] = server.available(); //產生一個以太網路用戶  
端網路物件  
    }  
}
```

如下圖所示，讀者可以看到本次實驗- Telnet 多人版聊天室，主機開始的畫面。

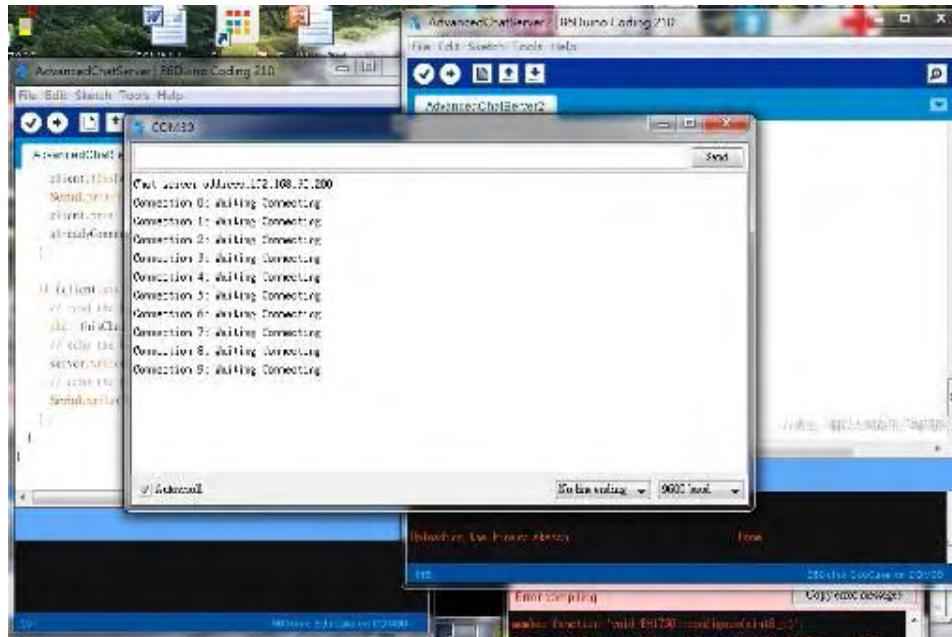


圖 20 Telnet 聊天室多人版主機畫面

如下圖所示，讀者可以看到本次實驗- Telnet 簡單多人版聊天室，我們使用 Putty 通訊軟體，預備進行連線的畫面。

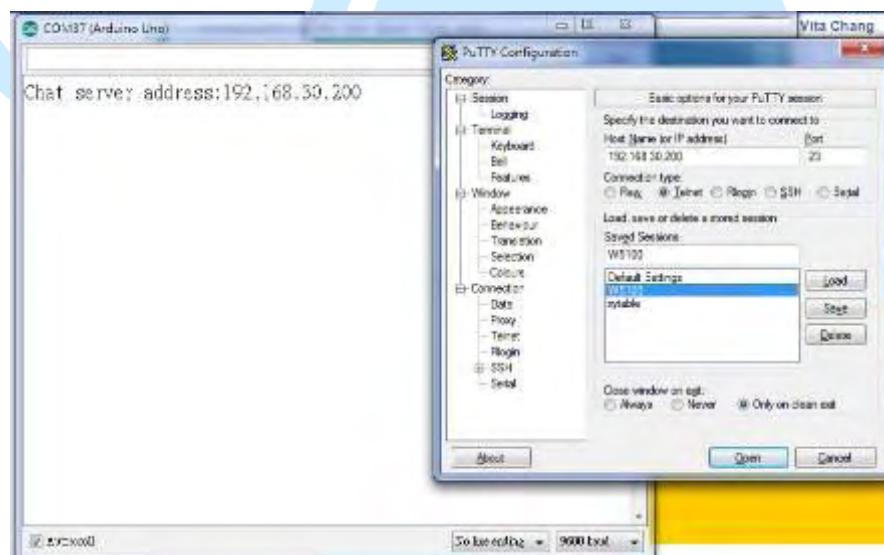


圖 21 Telnet 簡單聊天室多人版預備進行連線的畫面

如下圖所示，讀者可以看到本次實驗- Telnet 簡單聊天室，我們使用 Putty 通訊軟體，第一位 Telnet 簡單聊天室多人版連線中的畫面。

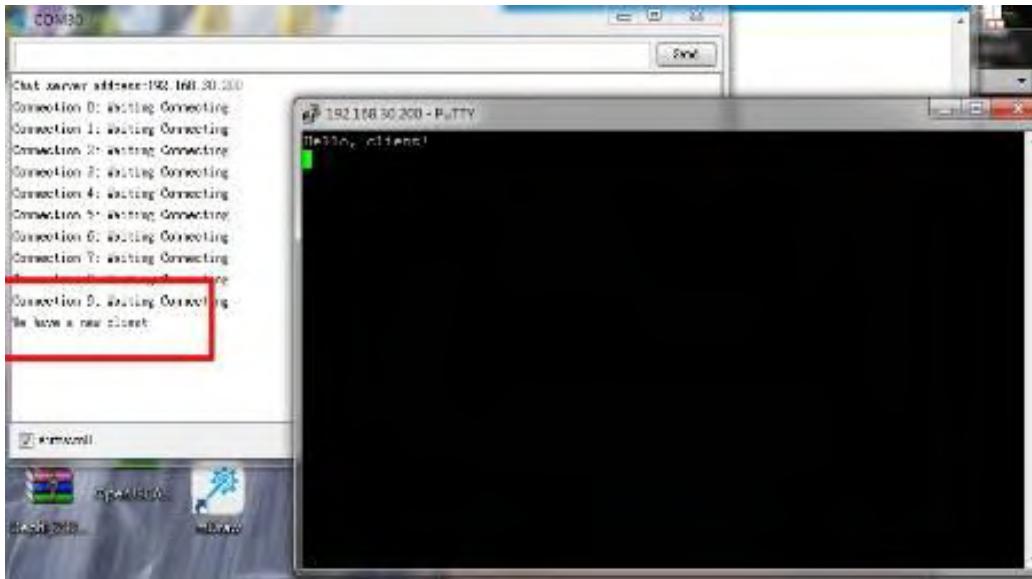


圖 22 第一位 Telnet 簡單聊天室多人版連線中的畫面

如下圖所示，讀者可以看到本次實驗- Telnet 簡單聊天室多人版，我們使用 Putty 通訊軟體，第二位 Telnet 簡單聊天室多人版進行的畫面。

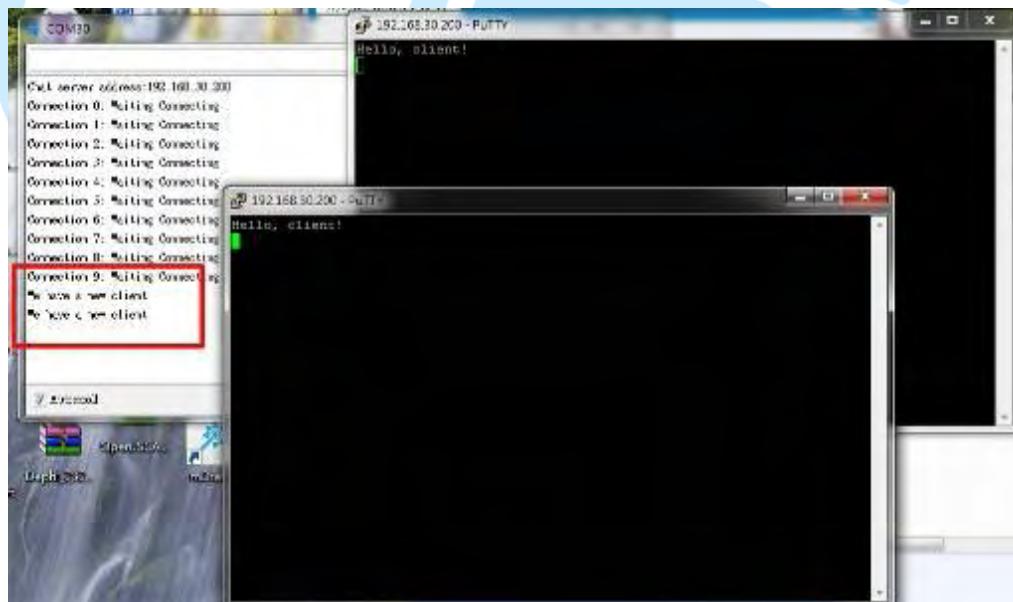


圖 23 第二位 Telnet 簡單聊天室多人版進行的畫面

## 九、溫度監控

如果我們要量測溫度，我們可以使用溫度感測器，如果我們又要量測濕度，我們也可以使用量測感測器，這樣我們會需要很多的感測器，所以本文介紹溫濕度感測模組(DHT11)，如下圖所示，它主要是使用 DHT-11 作成溫濕度感測模組。

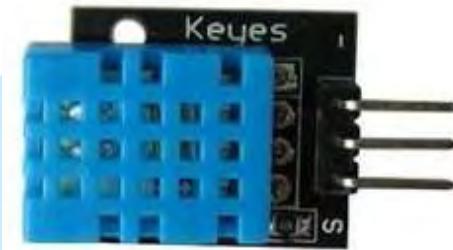


圖 24 DHT11 溫濕度感測模組

本實驗是採用 DHT11 溫濕度感測模組，如下圖所示，由於 DHT-11 溫濕度感測器需要搭配基本量測電路，所以我們使用 DHT11 溫濕度感測模組來當實驗主體，並不另外組立基本量測電路。

如下圖所示，先參考 DHT11 溫濕度感測模組腳位接法，在遵照下表之 DHT11 溫濕度感測模組接腳表進行電路組裝。

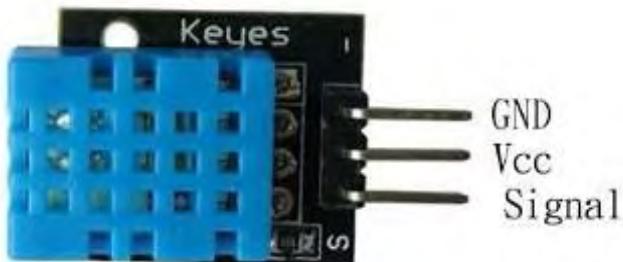
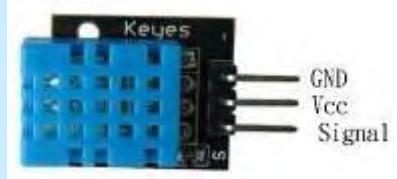


圖 25DHT11 溫濕度感測模組腳位圖

表 8DHT11 溫濕度感測模組接腳表

接腳	接腳說明	86Duino EduCake 開發板接腳
S	Vcc	電源 (+5V)
2	GND	EduCake GND
3	Signal	EduCake digital pin 7



資料來源：Arduino 程式教學(常用模組篇):Arduino Programming (37 Sensor Modules)(曹永忠, 許智誠, & 蔡英德, 2015b, 2015f)

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，編寫一段程式，如下表所示之 DHT11 溫濕度感測模組測試程式，我們就可以透過 DHT11 溫濕度感測模組來偵測任何溫度與濕度。

表 9 DHT11 溫濕度感測模組測試程式(DHT11)

```

int DHpin=7;
byte dat[5];

byte read_data()
{
    byte data;
    for(int i=0; i<8;i++)
    {
        if(digitalRead(DHpin)==LOW)
        {

```

```
while(digitalRead(DHpin)==LOW); //等待  
50us  
delayMicroseconds(30);  
//判斷高電位的持續時間，以判定數據是‘0’還是‘1’  
  
if(digitalRead(DHpin)==HIGH)  
    data |=(1<<(7-i));  
//高位在前，低位在後  
  
while(digitalRead(DHpin) == HIGH); //數據  
‘1’，等待下一位的接收  
}  
}  
return data;  
}  
  
void start_test()  
{  
    digitalWrite(DHpin,LOW); //拉低總線，發開始信號  
    delay(30); //延遲  
    時間要大於 18ms，以便檢測器能檢測到開始訊號；  
    digitalWrite(DHpin,HIGH);  
    delayMicroseconds(40); //等待感測器響應；  
    pinMode(DHpin,INPUT);  
    while(digitalRead(DHpin) == HIGH);  
        delayMicroseconds(80); //發出響應，拉低总线  
80us：  
    if(digitalRead(DHpin) == LOW);
```

```
delayMicroseconds(80); //線路 80us 後開始  
發送數據；  
  
for(int i=0;i<4;i++) //接收溫溼度數  
據，校驗位不考慮；  
dat[i] = read_data();  
  
pinMode(DHpin,OUTPUT);  
digitalWrite(DHpin,HIGH); //發送完數據後  
釋放線路，等待下一次的開始訊號；  
}  
  
void setup()  
{  
Serial.begin(9600);  
pinMode(DHpin,OUTPUT);  
}  
  
void loop()  
{  
start_test();  
Serial.print("Current humidity = ");  
Serial.print(dat[0], DEC); //顯示濕度的整  
數位；  
Serial.print('.');  
Serial.print(dat[1],DEC); //顯示濕度的小  
數位；  
Serial.println('%');  
Serial.print("Current temperature = ");
```

```
Serial.print(dat[2], DEC); //顯示溫度的整數  
位：  
Serial.print('.');  
Serial.print(dat[3],DEC); //顯示溫度的小數  
位：  
Serial.println('C');  
delay(700);  
}
```

參考資料：DMP 商店(<http://shop.dmp.com.tw/INT/products/67>)

如下圖所示，我們可以看到溫度感測模組測試程式結果畫面。

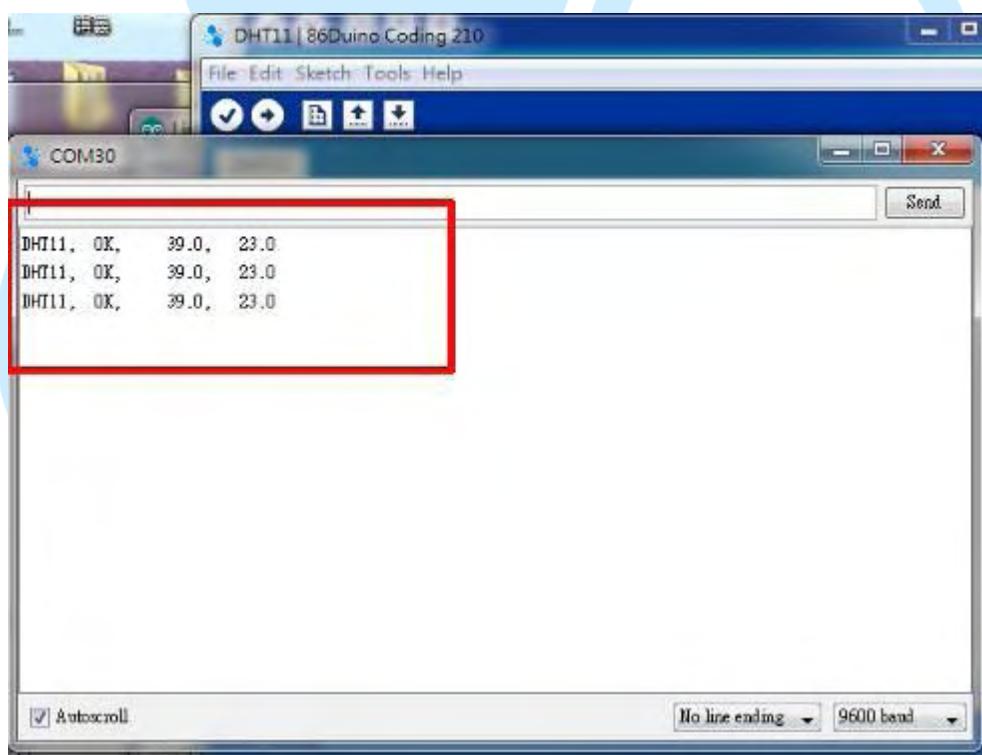


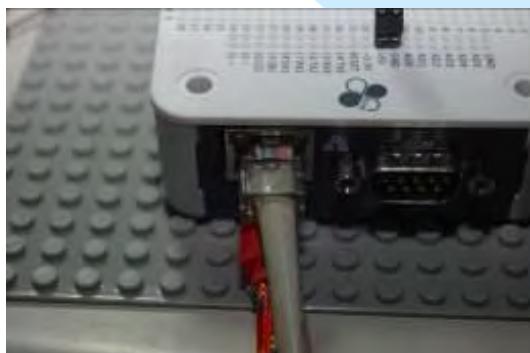
圖 26 溫度感測模組測試程式結果畫面

上面的的程式碼因為使用低階的控制訊號，對於一般人太吃力，所以作者將會將這些程式碼轉化成函式庫(Library)的方式來擴寫程式碼。

## 十、實做溫濕度監控網站

本章前面所敘述之內容，主要是讓讀者可以了解 86Duino EduCake 開發板如何使用網路資源，如何使用 Telnet、Http 的通訊協定，甚至用這些網路資源來建立一個簡單的網頁伺服器、聊天伺服器等，這些無異是為本書最後的專案打基礎，在這之前，我們先來做一個簡單的物聯網的小專案，如果我們建立一個溫濕度讀取的裝置，使其定時讀取溫度並已經建立一個簡單而完整的『溫濕度監控網站』，如此一來，所有的使用者就可以使用瀏覽器，透過網際網路來取得溫度的資訊了。

首先，如下圖所示，我們使用一般 Cat5 的網路線，配合 RJ45 接頭，一端接入 86Duino EduCake 開發板側邊網路接口(如下圖.(a))，一端接入網路集線器之網路接口(如下圖.(b))，完成 86Duino EduCake 開發板網路實體線路連接。



(a). EduCake 網路接法



(b). 集線器網路接法

圖 27 86Duino EduCake 開發板之網路接法示意圖

我們遵照前幾章所述，將 86Duino EduCake 開發板的驅動程式安裝好之後，我們打開 86Duino EduCake 開發板的開發工具：Sketch IDE 整合開發軟體，擷寫一段程式，如下表所示之實做溫濕度監控網站測試程式，我們就可以讓 86Duino EduCake 開發板變成一台的溫度監控網站伺服器運作了。

表 10 實做溫濕度監控網站測試程式(TempMonitorServer)

```
#include <SPI.h>
#include <Ethernet.h> // 使用網路必需要用的
#include <Wire.h> // 使用網路必需要用的
#include "dht.h" //使用 DHT11 溫濕度感測器所需要的函式庫

#define DHT11_PIN 7 //宣告 DHT11 溫濕度感測器通訊腳位
dht DHT; //宣告 DHT11 溫濕度感測器物件
// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
    0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
}; //使用這個 MAC ADDRESS 當做這台 educate 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200); //預設的網路 IP 位址，讀者可以自行更改為家中可用的位址
IPAddress dnServer(168, 95, 1, 1); //預設的 DNS 伺服器，本書為 Hinet 的 DNS 伺服器，讀者可以自行更改為家中可用的 DNS 伺服器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //預設的閘道器的位址(就是 Router 或 AP 的網址)，本書為作者寫書環境的閘道器的位址，讀者可以自行更改為家中可用的閘道器的位址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子網路遮罩，本書為 Class C

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80); //宣告伺服器並使用 Port 80 為通訊埠
```

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);          //宣告監控畫面的通訊速率  
    while (!Serial) {  
        ; // wait for serial port to connect. Needed for Leonardo only  
    }  
  
    int chk = DHT.read11(DHT11_PIN);      //讀取 DHT11 溫濕度感測器 狀態資  
料，並檢查是否可以使用  
    if (chkDHT(chk) == 0)    // 使用 chkDHT 函式檢查是否 DHT11 溫濕度感  
測器 狀態資料是否正常，不正常傳回 0，正常傳回 1  
    {  
        Serial.println("ERROR on init DHT Sensor");      //告知使用者  
        DHT11 溫濕度感測器不正常  
        while (true);          //程式終止  
    }  
  
    // start the Ethernet connection and the server:  
    Ethernet.begin(mac, ip, dnServer, gateway, subnet);    //使用上述宣告的網路設  
定來啟動 WEB 伺服器  
    server.begin();          //開啟 Web 伺服器運作  
    Serial.print("server is at ");    //印出伺服器資訊  
    Serial.println(Ethernet.localIP());      //印出伺服器 IP 位址  
}  
  
void loop() {  
    EthernetClient client = server.available();      //如果有人連入 WEB 伺服  
器，啟動一個 Client 連線端  
    if (client) {      //如果啟動一個 Client 連線端成功
```

```
Serial.println("new client");      //印出"新連線"  
// an http request ends with a blank line  
boolean currentLineIsBlank = true;  
while (client.connected()) {      //如果連線成功  
    if (client.available()) {      //如果連線的 Client 連線端有送資料過來  
        char c = client.read();    //讀入連線的 Client 連線端資料到變數 c  
        Serial.write(c);          //印出變數 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) {  
            //如果變數 c 是換行資料而且  
            //目前資料為空  
            // send a standard http response header  
            client.println("HTTP/1.1 200 OK");  
            //送出 Http 標頭資料(固定要  
            //送)  
            client.println("Content-Type: text/html");  
            //送出 Http 標頭資料(固  
            //定要送)  
            client.println("Connection: close"); // the connection will be closed af-  
            ter completion of the response //送出 Http 標頭資料(固定要送)  
            client.println("Refresh: 5"); // refresh the page automatically every 5  
            sec //送出 Http 標頭資料(固定要送)  
            client.println();  
            //送出 Http 標頭資料(固定要送)  
            client.println("<!DOCTYPE HTML>");  
            //送出 Http 標頭資料(固  
            //定要送)  
            client.println("<html>");  
            //送出 Http 標頭資料(固定要送)  
            // output the value of each analog input pin  
  
            client.print("Humidity: ");  
            //送出目前濕度提示字句  
            client.print(DHT.humidity, 1);  
            //送出目前濕度  
            client.println("<br>");  
            //送出網頁換行鍵
```

```
client.print("Temperature: ");      //送出目前溫度提示字句
client.print(DHT.temperature, 1);    //送出目前溫度
client.println("<br>");            //送出網頁換行鍵

client.println("</html>");          //送出網頁 Tag 資料(固定要送)
break;
}

if (c == '\n') {                  //新的一行
// you're starting a new line
currentLineIsBlank = true;
}
else if (c != '\r') {             //如果不是換行鍵
// you've gotten a character on the current line
currentLineIsBlank = false;
}
}

// give the web browser time to receive the data
delay(1);                      //延遲一毫秒

// close the connection:
client.stop();                  //關閉連線的 Client 連線端(必需要關閉，不然會掛掉)
Serial.println("client disconnected"); //送出連線斷線
}

delay(2000);                    //延遲 2000 毫秒
}

unsigned int chkDHT( int chk )    //檢核 DHT11 溫濕度感測器 狀態資料是否
正確
{
```

```
switch (chk)          //檢查 chk 變數;多重檢查
{
    case DHTLIB_OK:      //如果是 ok 狀態
        Serial.println("DHT init is OK,\t");
        return 1;
    case DHTLIB_ERROR_CHECKSUM: //如果是數值加總不正確狀態
        Serial.println("DHT Checksum error,\t");
        //印出數值加總
        不正確狀態
        return 0;
    case DHTLIB_ERROR_TIMEOUT: //如果是在固定時間讀不到正確資料
        Serial.println("DHT Time out error,\t");
        //印出在固定時間讀
        不到正確資料
        return 0;
    default:              //不知名錯誤
        Serial.println("DHT Unknown error,\t");
        //不知名錯誤
        return 0;
}
```

如下圖所示，讀者可以看到本次實驗- 實做溫濕度監控網站監控主畫面。

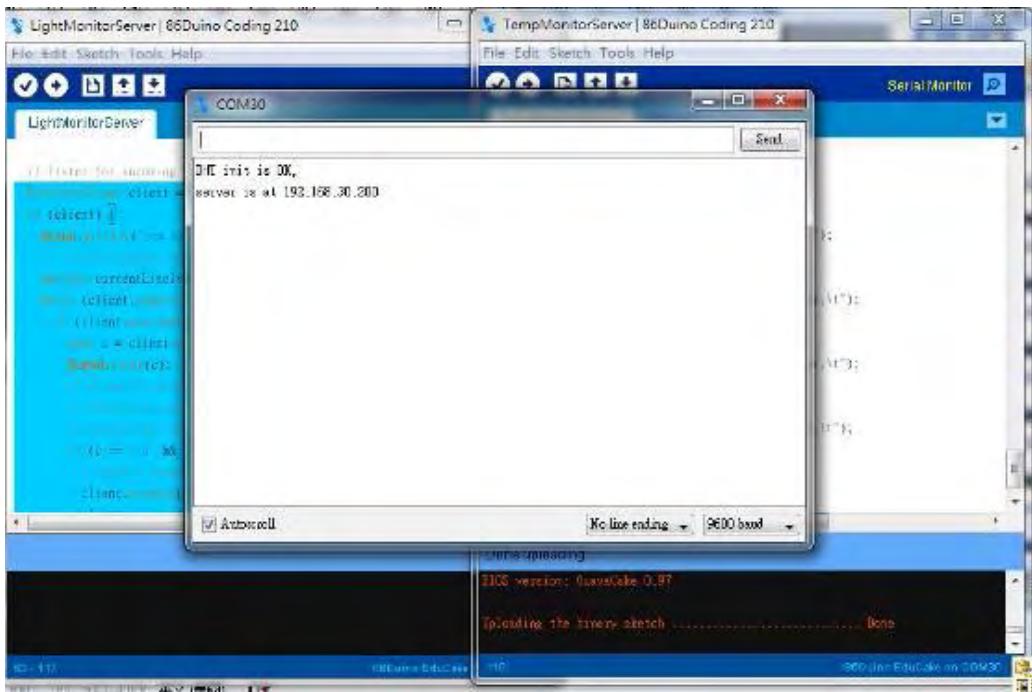


圖 28 溫濕度監控主機畫面

如下圖所示，我們使用瀏覽器連入溫濕度監控主機，讀者看到瀏覽器可以讀出溫濕度監控裝置的溫濕度資訊的畫面。

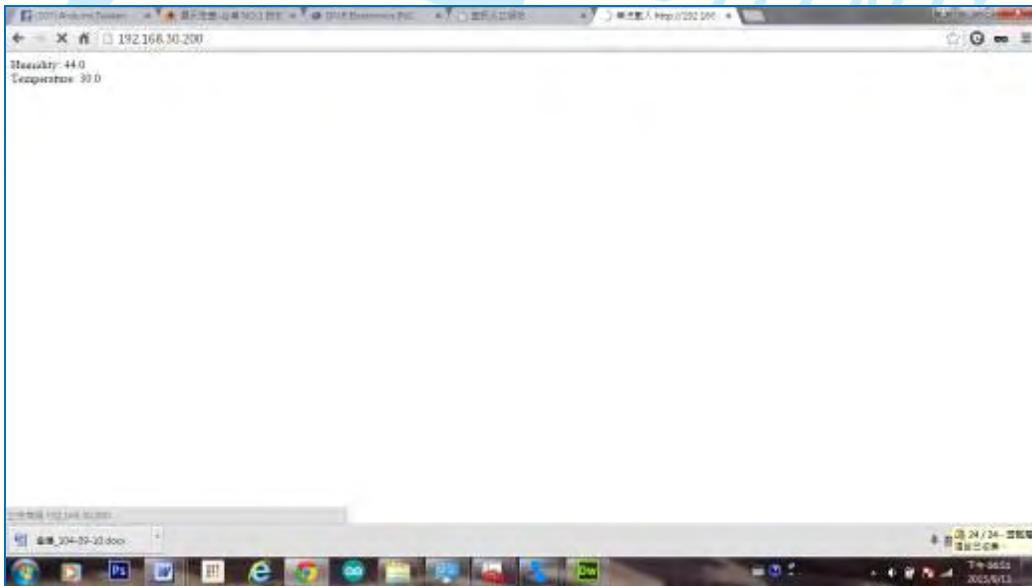


圖 29 用瀏覽器連入溫濕度監控主機

如下圖所示，我們可以看到溫濕度監控主機回應瀏覽器端，所回應的 HTML 內容，可以讓使用者端的瀏覽器讀出溫濕度監控裝置的溫濕度資訊。

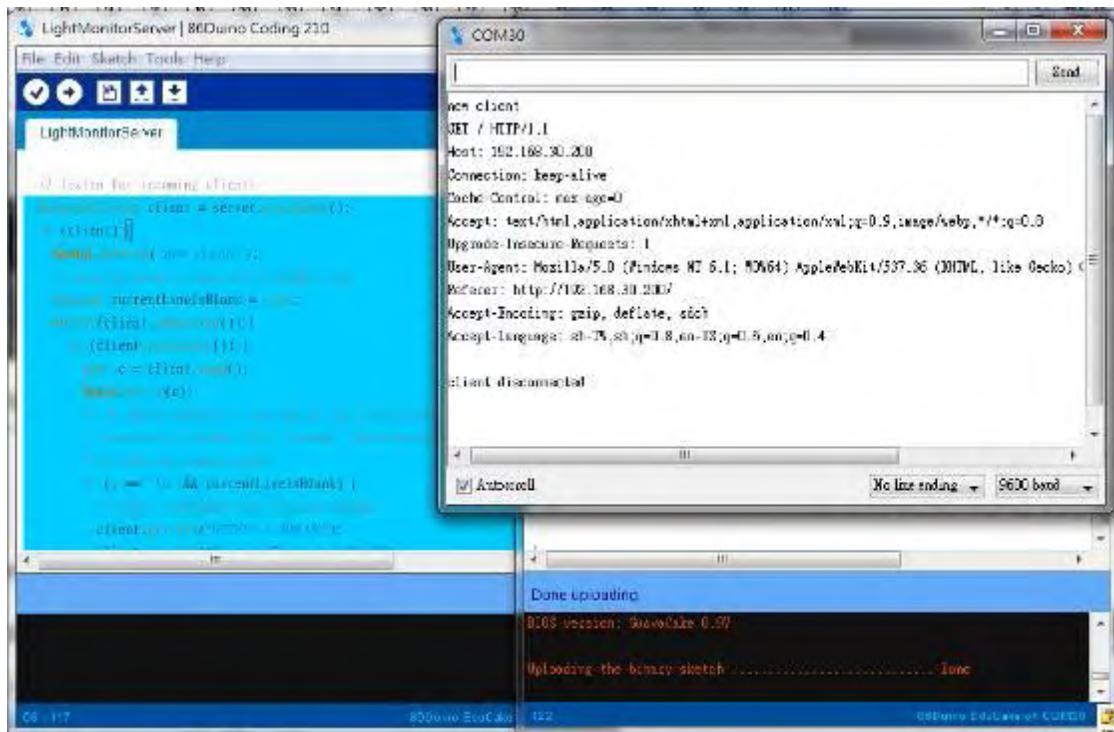


圖 30 溫濕度監控主機回饋瀏覽器畫面

## 十一、章節小結

本章主要介紹 86Duino EduCake 開發板，透過內建的乙太網路來傳送、接收溫濕度資料並透過瀏覽器顯示資訊。