以太网络

一、86Duino EduCake 网络介绍

86Duino EduCake 开发板 是一款 x86 架构的开源(Open Source)微电脑学 习机,内部采用高性能 32 位 x86 兼容的处理器 Vortex86EX,可以兼容并执行 Arduino 的程序,特点是内建面包板,使用者不需经由焊接过程,即可快速将许 多电子组件、传感器及外围配件加以连接或置换并进行电子实验。其内建的特殊 电路保护设计,能防止错误操作而导致烧毁 I/O 接脚。除此之外,EduCake 开 发板 外壳是由坚固的金属和面包板组合而成,重要的电子零件皆被包覆其内, 周围留下常用的 I/O 接口,使得 EduCake 开发板 不容易受到外力破坏,适合 让使用 Arduino 、微电脑及嵌入式系统的初学者、设计师、业余爱好者、任何 有兴趣的人,打造自己专属的电子互动装置。

86Duino EduCake 开发板不需要像 Arduino 开发板,需要外加如 W5100 网络扩充板(曹永忠,許智誠, & 蔡英德, 2015a, 2015c, 2015d, 2015e)才能运行在因特网,只要用内建的以太网络模块,一样可以将 TCP/IP Protocols (TCP, UDP, ICMP, IPv4 ARP, IGMP, PPPoE, Ethernet) 整合在 86Duino EduCake 硬件电路上,减轻了外加扩充板的麻烦与成本。

86Duino EduCake 开发板也只要使用 Ethernet Library¹ 便可以轻易完成连 至因特网的动作。

¹ 可到 Arduino.cc 的官网: http://www.arduino.cc/en/reference/ethernet, 下载函式库与相关范例。

www.86duino.com



图 1 86Duino EduCake 开发板

如上图所示,86Duino EduCake 开发板的网络在主机侧边的以太网络 RJ45 接口,如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器之网 络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

(b). 集线器网络接法

图 2 86Duino EduCake 开发板之网络接法示意图

E86Duino EduCake 开发板侧面提供一组网络接口(Local Area Network:LAN),支持 10/100Mbps 传输速度。在网络接口(Local Area Network:LAN) 上,如下图所示,86Duino EduCake 开发板更使用了静电保护芯片以及感应式 线圈来保护内部重要的电子零件:

www.86duino.com



将 86Duino EduCake 开发板开机后,如下图所示,网络线连接至 LAN 接口,若网络信号正常,不久后即可看到 LAN 接口的指示灯亮起,接口右边的绿灯恒亮,左边的橘灯闪烁:



图 3 86Duino EduCake 开发板网路灯号示意图

二、简单 Web Server

如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器之网 络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

- (b). 集线器网络接法
- 图 4 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将86Duino EduCake 开发板的驱动程序安装好之后, 我们打开86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥 写一段程序,如下表所示之 WebServer 测试程序,我们就可以让86Duino EduCake 开发板变成一台简易的网页服务器运作,并可以将 Analog Port0 ~Port5 的状态显示在浏览器画面。

表 1 WebServer 测试程序(WebServer)



- 5 -

86DUIND

// start the Ethernet connection and the server:
Ethernet.begin(mac, ip, dnServer, gateway, subnet); //使用上述宣告
的网络设定来启动 WEB 服务器
server.begin(); //开启 Web 服务器运作
Serial.print("Server is running at ip: "); //印出服务器信息
Serial.println(Ethernet.localIP()); //印出服务器 IP 地址
}
void loop() {
// listen for incoming clients
EthernetClient client = server.available(); //如果有人连入 WEB 服务器,启
动一个 Client 联机端
if (client) { //如果启动一个 Client 联机端成功
Serial.println("New Client"); //印出"新联机"
// an http request ends with a blank line
boolean currentLineIsBlank = true; //
while (client.connected()) { //如果联机成功
if (client.available()) { //如果联机的 Client 联机端有送数据过来
char c = client.read(); //读入联机的 Client 联机端数据到变量
с
Serial.write(c); //印出变数 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) { //如果变量 c 是换行数据
而且目前数据为空
// send a standard http response header
client.println("HTTP/1.1 200 OK"); //送出 Http 标头数据(固
定要送)
client.println("Content-Type: text/html"); //送出 Http 标头数
据(固定要送)
client.println("Connection: close"); // the connection will be

```
closed after completion of the response
                                         //送出 Http 标头数据(固定要
送)
         client.println("Refresh: 5"); // refresh the page automatically
every 5 sec
              //送出 Http 标头数据(固定要送)
         client.println();
                         //送出 Http 标头数据(固定要送)
         client.println("<!DOCTYPE HTML>"); //送出 Http 标头数据(固
定要送)
         client.println("<html>");
                                    //送出 Http 标头数据(固定要送)
         // output the value of each analog input pin
         for (int analogChannel = 0; analogChannel < 6; analogChan-
           //将模拟脚位0到5得资料读出来
nel++)
           int sensorReading = analogRead(analogChannel);
           client.print("Analog Input ");
                                             //送出画面资料
           client.print(analogChannel);
                                         //送出第几个模拟通讯端口
       client.print(" is ");
                             //送出画面资料
           client.print(sensorReading);
                                         //送出读出该模拟通讯端口
的数据
           client.println("<br />");
                                     //送出网页换行键
         }
         client.println("</html>");
                                     //送出网页 Tag 数据(固定要送)
         break;
       if (c == '\n') \{
                          //新的一行
         // you're starting a new line
         currentLineIsBlank = true;
       }
                         //如果不是换行键
       else if (c != '\r') {
         // you've gotten a character on the current line
         currentLineIsBlank = false;
       }
     }
   }
   // give the web browser time to receive the data
```

www.86duino.com



如下图所示,读者可以看到本次实验-WebServer测试程序结果画面。



(a). WebServer 测试程序开始画面





		BECOR.
(b) 启动浏览器连接		
COM30		. • ×
		Send
Mire to		
Connection: Keep-Alive		
13 (11) (11)		
Lient Disconnected		
ccept: text/html, application/rhtml+rwl, */*		
ccept-Language: zh-TV		
Ser-Agent: Nosille/5.0 (Vindows NT 6.1; VON64; Trident/7.0; rv:11.0) lil	ke Gecko	
ccept-Encoding: gzip, deflate		
tost: 192.168.30.200		
.MJ: 1		
Connection: Meep-Allye		
Client Disconnected		
		1
llenzeter A	No line ending 🖕 🦻	600 band 🗸
T MARKAN A		

(c). WebServer 联机后结果画面

图 5 WebServer 测试程序结果画面

三、使用 DHCP 架设 Web Server

如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器之网 络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

(b). 集线器网络接法

图 6 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将 86Duino EduCake 开发板的驱动程序安装好之后, 我们打开 86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥写 一段程序,如下表所示之 WebServer 测试程序一,我们就可以让 86Duino EduCake 开发板变成一台简易的网页服务器运作,所不同的是 86Duino Edu-Cake 开发板所成为的 WebServer 所取得的 IP 地址不是自定义,而是从 DHCP 服务器取得的 IP 地址,并可以将 Analog Port0~Port5 的状态显示在浏览器画面。

表 2 WebServer 测试程序一(WebServer_Dhcp)



```
0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF
};
       //使用这个 MAC ADRESS 当做这台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200); //默认的网络 IP 地址,读者可以自行更改为
家中可用的地址
IPAddress dnServer(168, 95, 1, 1); //预设的 DNS 服务器,本书为 Hinet
的 DNS 服务器,读者可以自行更改为家中可用的 DNS 服务器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或
AP 的网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可用
的网关的地址
// the subnet:
IPAddress subnet(255, 255, 255, 0);
                                     //子网掩码,本书为 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); //宣告服务器并使用 Port 80 为通讯端口
void setup() {
 // Open serial communications and wait for port to open:
                    //宣告监控画面的通讯速率
 Serial.begin(9600);
   Serial.println("Now Program Start");
 while (!Serial) {
   ; // wait for serial port to connect. Needed for Leonardo only
 }
 // 启用 Ethernet 联机,预设会以 DHCP 取得 IP 地址
 if (Ethernet.begin(mac) == 0) {
   Serial.println("I can't get any IP address from DHCP Server");
```

86Duino

// 无法取得 IP 地址,不做任何事情
for(;;)
. ,
}
// 输出 IP 地址
Serial.print("Default IP Address is : "); //印出服务器信息
Serial.println(ip); //印出服务器 IP 地址
// Ethernet.begin(mac, ip, dnServer, gateway, subnet);
Ethernet.begin(mac); // 如果要用 DHCP 服务器取得 ip,必需要这样写
server.begin(); //开启 Web 服务器运作
Serial.print("Server is running at ip: "); //印出服务器信息
Serial.print(Ethernet.localIP()); //印出服务器信息
Serial.print("from DHCP Server \n");} //印出服务器信息
Void ioop() {
// listen for incoming clients EthernetClient client = server available(): //加里有人连入WEB 服务界
ethemetchent Chent 平和提
后初 个 Client 联机端
II (Clefit) { //如来后幼 [—] /「Clefit 联机场成功
Senal.printin(New Client); //印出 新联机
// an nttp request ends with a blank line
boolean currentLineisblank – true,
wille (Client.collinected()) { //如果联机放功
II (Clefit.available()) { //如采联机的 Clefit 联机端有达数据过术
cnar c = client.read(); // 如果联机的 Client 联机端有达数据过米
Serial.write(c); //印出受数 c
<pre>// it you've gotten to the end of the line (received a newline // above star) and the line is black the line but</pre>
// character) and the line is blank, the http request has ended,

// so you can send a reply
if (c == '\n' && currentLineIsBlank) { //如果变量 c 是换行数据而且
目前数据为空
// send a standard http response header
client.println("HTTP/1.1 200 OK"); //送出 Http 标头数据(固定要
送)
client.println("Content-Type: text/html"); //送出 Http 标头数
据(固定要送)
client.println("Connection: close"); // the connection will be closed
after completion of the response //送出 Http 标头数据(固定要送)
client.println("Refresh: 5"); // refresh the page automatically every
5 sec //送出 Http 标头数据(固定要送)
client.println(); //送出 Http 标头数据(固定要送)
client.println(" HTML "); //送出 Http 标头数据(固
定要送)
client.println(" <html>"); //送出 Http 标头数据(固定要送)</html>
// output the value of each analog input pin
for (int analogChannel = 0; analogChannel < 6; analogChannel++)
{ //将模拟脚位 0 到 5 得资料读出来
int sensorReading = analogRead(analogChannel);
client.print("Analog Input "); //送出画面资料
client.print(analogChannel); //送出第几个模拟通讯端口
client.print(" is "); //送出画面资料
client.print(sensorReading); //送出读出该模拟通讯端口
的数据
client.println(" "); //送出网页换行键
}
client.println(""); //送出网页 Tag 数据(固定要送)
break;
}

www.86duino.com



如下图所示,读者可以看到本次实验-WebServer测试程序一结果画面。

www.86duino.com





(b). 启动浏览器连接



四、Telnet 客户端程序

首先,如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器 之网络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

(b). 集线器网络接法

图 8 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将 86Duino EduCake 开发板的驱动程序安装好之后, 我们打开 86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥 写一段程序,如下表所示之 Telnet 客户端程序测试程序,我们就可以让 86Duino EduCake 开发板变成一台简易的 Telnet 客户端工作站。

表 3 Telnet 客户端程序测试程序(TelnetClient)

#include <spi.h></spi.h>
#include <ethernet.h> // 使用网络必需要用的</ethernet.h>
// 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
}; //使用这个 MAC ADRESS 当做这台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 200); //默认的网络 IP 地址,读者可以自行更改为家中
可用的地址
IPAddress dnServer(168, 95, 1, 1); //预设的 DNS 服务器,本书为 Hinet 的 DNS 服
务器,读者可以自行更改为家中可用的 DNS 服务器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或 AP 的
网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可用的网关的
地址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子网掩码,本书为 Class C
// Enter the IP address of the server you're connecting to:
IPAddress server(140, 112, 172, 11); //台大 PTT 网址
// 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; //宣告一个 TCP/IP 的 Client 对象
void setup() {
// start the Ethernet connection:
// start the Ethernet connection.

```
Ethernet.begin(mac, ip, dnServer, gateway, subnet); // 启用 Ethernet 联机
  // Open serial communications and wait for port to open:
                         //宣告监控画面的通讯速率
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  // give the Ethernet shield a second to initialize:
                     //延持 1000 毫秒
  delay(1000);
  Serial.println("connecting..."); //印出联机中字句....
  // if you get a connection, report back via serial:
  if (client.connect(server, 23)) { //如果 Client 对象连到 server(140, 112, 172, 11);
                                  //印出联机成功字句....
台大 PTT 网址 成功的话
    Serial.println("connected");
  }
  else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed"); //印出联机失败字句....
  }
void loop()
  // if there are incoming bytes available
  // from the server, read them and print them:
  if (client.available()) { //如果 Client 对象有东西回传
    char c = client.read(); //读出 Client 对象,并存到变量 c
                    //印出变数 c //
    Serial.print(c);
  }
```

www.86duino.com

// 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) { //如果我们从监控画面输入文字,
char inChar = Serial.read(); //读出监控画面输入文字,并存到变量 c
if (client.connected()) { //如果联机中
client.print(inChar); //将输入文字传到 Client 对象
}
}
// if the server's disconnected, stop the client:
if (!client.connected()) { //如果 Client 对象已断线
Serial.println();
Serial.println("disconnecting."); ///印出已断线
client.stop(); //Client 对象关闭
// do nothing:
while (true); //永久循环,不再做事
}
}

如下图所示,读者可以看到本次实验结果画面。



图 9 Telnet 客户端程序测试程序结果画面

五、文字型 Browser 客户端程序

如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器之网 络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

(b). 集线器网络接法

图 10 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将86Duino EduCake 开发板的驱动程序安装好之后, 我们打开86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥 写一段程序,如下表所示之 Telnet 客户端程序测试程序,我们就可以让86Duino EduCake 开发板变成一台简易的网浏览器文字接口处理机运作了,未来我们还 可以透过文字过滤的方式,来搜询想要的信息了。

表 4 文字型 Browser 客户端程序(WebClient)

#include <SPI.h> #include <Ethernet.h> // 使用网络必需要用的 // 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 //使用这个 MAC ADRESS 当做这台 educake 的 MAC ADDRESS };

```
//默认的网络 IP 地址,读者可以自行更改
IPAddress ip(192, 168, 30, 200);
为家中可用的地址
IPAddress dnServer(168, 95, 1, 1); //预设的DNS服务器,本书为Hinet的DNS
服务器,读者可以自行更改为家中可用的 DNS 服务器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或
AP 的网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可
用的网关的地址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子网掩码,本书为 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 的网址
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);
                        //宣告监控画面的通讯速率
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
                                //要求 DHCP 服务器发放 ip 地址等信息
```

86DUIND

```
Serial.println("Failed to configure Ethernet using DHCP");
                                                            //要求 DHCP
服务器发放 ip 地址失败,发出警告讯息
    // no point in carrying on, so do nothing forevermore:
    // try to congifure using IP address instead of DHCP:
      Ethernet.begin(mac, ip, dnServer, gateway, subnet); //使用系统默认的
ip 地址
  }
  // give the Ethernet shield a second to initialize:
  delay(1000);
                         //延持 1000 毫秒
  Serial.println("connecting...");
                                 //印出联机中字句....
  // if you get a connection, report back via serial:
                                 //如果 Client 对象连到 serv-
  if (client.connect(server, 80)) {
er(74,125,232,128) www.google.com 的网址 成功的话
    Serial.println("connected");
                                  //印出联机成功字句.
    // Make a HTTP request:
    //以下是固定 http 通讯 protocol
    client.println("GET /search?q=arduino HTTP/1.1");
    client.println("Host: www.google.com");
                                               //以下是告知连到
www.google.com
    client.println("Connection: close");
                                         //连完后关闭
    client.println();
  }
  else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed"); // 连接 www.google.com 失败,告诉
使用者连接失败
  }
}
```

www.86duino.com

```
void loop()
 // if there are incoming bytes available
 // from the server, read them and print them:
                            //如果 Client 对象有东西回传
  if (client.available()) {
                            //读出 Client 对象,并存到变量 c
    char c = client.read();
                        //印出变数 c
    Serial.print(c);
  }
 // if the server's disconnected, stop the client:
  if (!client.connected()) { //如果 Client 已经关闭联机
    Serial.println();
    Serial.println("disconnecting.");
                                          //印出中断联机成功字句...
                                           DUINT
                    // 连现中止
    client.stop();
    // do nothing forevermore:
    while (true);
                        //永久循环中断程序
  }
```

如下图所示,读者可以看到本次实验结果画面。

COM37 (Arduine Une)	analiti assa		23
	Zend		
connecting connected HTTP/1.1 302 Found Location: http://www.google.co	m.tw/search?q=arduino&gws_rd=cr8	sensiverer n("GET /search?q=arduino HTTP/1.1"); n("Host: www.google.com"); n("Connection: close");	
Cache-Control: private Content-Type: text/html; chard Set-Cookie: PREF=ID=d86e2f1082 Set-Cookie: NID=67=YCXytCON2Tt P3P: CP="This is not a P3P pol Date: Mon, 27 Apr 2015 03:53:1	set=UTF-8 270314e:FF=0:TM=1430106797:LM=14 aa8hxokxfsGguoNwcMnGs41QLV_91xx icy! See http://www.google.com/: 7 GMT	<pre>n't get a connection to the server: ("connection failed");</pre>	4
Server: gws Content Length: 290			, "
X-XSS-Protection: 1; mode=bloc X-Frame-Options: SAMEORIGIN	.k	es for local variables. Maximum is 2,048	2
M Automal	No has ending + 9600 hotel +		
· · · · [* % & · * 0 0 0	8 + - x - < x 20	Andrew Man de CO	1467

图 11 文字型 Browser 客户端程序结果画面

六、取得网络校时时间数据

如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器之网 络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法



(b). 集线器网络接法

图 12 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将 86Duino EduCake 开发板的驱动程序安装好之后, 我们打开 86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥 写一段程序,如下表所示之网络校时测试程序,我们就可以让 86Duino EduCake 开发板变成一台简易的网络校时机器,取得网络校时时间。

表 5 网络校时测试程序(UdpNtpClient)

#include <spi.h></spi.h>	
<pre>#include <ethernet.h></ethernet.h></pre>	// 使用网络必需要用的
<pre>#include <ethernetudp.h></ethernetudp.h></pre>	// 使用网络 UDP 通讯协议必需要用的
// Enter a MAC address for y	our controller below.
// Newer Ethernet shields have	e a MAC address printed on a sticker on the shield
byte mac[] = {	
0xAA, 0xBB, 0xCC, 0xDI	D, 0xEE, 0xFF
}; //使用这个 MAC	ADRESS 当做这台 educake 的 MAC ADDRESS
IPAddress ip(192, 168, 30, 24	00); //默认的网络 IP 地址,读者可以自行更改为

86DUIND

家中可用的地址
IPAddress dnServer(168, 95, 1, 1); //预设的 DNS 服务器,本书为 Hinet 的 DNS
服务器,读者可以自行更改为家中可用的 DNS 服务器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或
AP 的网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可用
的网关的地址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子网掩码,本书为 Class C
unsigned int localPort = 8888; //要校时必须设定一个 local port 来给 UDP 使
用
// local port to listen for UDP packets
char timeServer[] = "time.nist.gov"; // time.nist.gov NTP server 校时服务器
const int NTD DACKET SIZE = $49 \cdot 1/$ NTD time stamp is in the first 48 bytes of the
message 封句大小 48 字节
byte packetBuffer[NTP PACKET SIZE]; //buffer to hold incoming and outgoing
packets 收入封包数据变量
// A UDP instance to let us send and receive packets over UDP
EthernetUDP Udp; //宣告一个 UDP 网络对象
void setup()
{
// Open serial communications and wait for port to open:
Serial.begin(9600); //宣告监控画面的通讯速率
while (!Serial) {

```
; // wait for serial port to connect. Needed for Leonardo only
  }
  // start Ethernet and UDP
  if (Ethernet.begin(mac) == 0) { //要求 DHCP 服务器发放 ip 地址等信息
    Serial.println("Failed to configure Ethernet using DHCP"); //要求 DHCP
服务器发放 ip 地址失败,发出警告讯息
    // no point in carrying on, so do nothing forevermore:
                                                 //使用系统默认的ip地址
  Ethernet.begin(mac, ip, dnServer, gateway, subnet);
  }
  Udp.begin(localPort);
                        //启动 UDP
                                             JIIN
void loop()
  sendNTPpacket(timeServer); // send an NTP packet to a time server
                                                               传送封包
到 time.nist.gov NTP server 校时服务器
  // wait to see if a reply is available
                    //延持 1000 毫秒
  delay(1000);
  if (Udp.parsePacket()) { //如果收到 time.nist.gov NTP server 校时服务
器 传送封包
    // We've received a packet, read the data from it
    Udp.read(packetBuffer, NTP PACKET SIZE); // read the packet into the buffer
        //读取封包
    //the timestamp starts at byte 40 of the received packet and is four bytes,
    // or two words, long. First, esxtract the two words:
```

86DUIND

unsigned long highWord = word(packetBuffer[40], packetBuffer[41]); //
将高字节转到 highWord 变量
unsigned long lowWord = word(packetBuffer[42], packetBuffer[43]); //将低
字节转到 lowWord 变量
// 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; //将高
低位组合成一个变量
Serial.print("Seconds since Jan 1 1900 = "); //印出时间
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; //设定 unix 系统初始
时间
// subtract seventy years:
unsigned long epoch = secsSince1900 - seventyYears; //计算取得时间差
// print Unix time:
Serial.println(epoch); //印出时间差
// 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
<pre>Serial.print(':');</pre>
if (((epoch % 3600) / 60) < 10) {
// In the first 10 minutes of each hour, we'll want a leading '0' 印出多少小时
Serial.print('0');

```
Serial.print((epoch % 3600) / 60); // print the minute (3600 equals secs per
minute) 印出多少分钟
    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
                           //延持 10000 毫秒
  delay(10000);
}
//送出 NTP 封包函数
// send an NTP request to the time server at the given address
unsigned long sendNTPpacket(char* address)
                                              //送出 NTP 封包函
Ş
         //送出 NTP 封包函数
  // set all bytes in the buffer to 0
  memset(packetBuffer, 0, NTP_PACKET_SIZE); //设定 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;
```

www.86duino.com

// all NTP fields have been given values, now		
// you can send a packet requesting a timestamp:		
Udp.beginPacket(address, 123); //NTP requests are to po	ort 123	利用 Port
123 来传送到 address 变量中的 ip 地址		
Udp.write(packetBuffer, NTP_PACKET_SIZE);	//传送 packet	Buffer 变量
数据		
Udp.endPacket(); //传送结束		
}		

如下图所示,读者可以看到本次实验-网络校时测试程序结果画面。

Seconds since Jan 1 1900 - 3650766775 Unix time = 1441777975	
The UTC time is 5:52:55	
Jaconus since Jan 1 1900 = 3030706786	
The UTC time is 5:53:06	

图 13 网络校时测试程序结果画面

七、Telnet 简单聊天室

如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)), 一端接入网络集线器之网 络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

(b). 集线器网络接法

图 14 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将86Duino EduCake 开发板的驱动程序安装好之后, 我们打开 86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件, 攥 写一段程序,如下表所示之 Telnet 简单聊天室测试程序,我们就可以让 86Duino EduCake 开发板变成一台简易的 Telnet 简单聊天室运作了。

表 6 Telnet 简单聊天室测试程序(ChatServer)

#include <SPI.h>

#include <Ethernet.h> // 使用网络必需要用的

// 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

//使用这个 MAC ADRESS 当做这台 educake 的 MAC ADDRESS }; IPAddress ip(192, 168, 30, 200); //默认的网络 IP 地址,读者可以自行更改为家

86DUIND

www.86duino.com

```
中可用的地址
IPAddress dnServer(168, 95, 1, 1); //预设的 DNS 服务器,本书为 Hinet 的 DNS
服务器,读者可以自行更改为家中可用的 DNS 服务器
// the router's gateway address:
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或 AP
的网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可用的网
关的地址
// the subnet:
IPAddress subnet(255, 255, 255, 0); //子网掩码,本书为 Class C
// telnet defaults to port 23
EthernetServer server(23); //宣告一个以太网络服务器对象,并使用 Port23 为
listening 通讯端口
boolean alreadyConnected = false; // whether or not the client was connected previously
                                            UINI
联机旗标
void setup() {
 // initialize the ethernet device
  Ethernet.begin(mac, ip, gateway, subnet); //启动以太网络
 // start listening for clients
                  //启动 Port23 的服务器对象
  server.begin();
 // Open serial communications and wait for port to open:
                      //宣告监控画面的通讯速率
  Serial.begin(9600);
  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());
                                  //印出网址
```

- 32 -

```
void loop() {
  // wait for a new client:
  EthernetClient client = server.available(); //产生一个以太网络客户网络对象
  // when the client sends the first byte, say hello:
               //当以太网络客户网络对象有人连入
  if (client) {
    if (!alreadyConnected) { //如果是新联机
      // clead out the input buffer:
      client.flush();
                    //以太网络客户网络对象 更新缓冲区数据
      Serial.println("We have a new client");
                                             //印出有新联机
                                        //送出 Hello 欢迎词
      client.println("Hello, client!");
      alreadyConnected = true;
                               //设定已联机
    }
    if (client.available() > 0) {
                                //如果以太网络客户网络对象有数据送出
      // read the bytes incoming from the client:
                                    //读取以太网络客户网络对象送出的数据
      char thisChar = client.read();
到 thisChar 变量
      // echo the bytes back to the client:
                                //送出 thisChar 变量到 server 对象
      server.write(thisChar);
      // echo the bytes to the server as well:
                           //印出 thisChar 变数
      Serial.write(thisChar);
    }
  }
}
```

如下图所示,读者可以看到本次实验-Telnet 简单聊天室,Telnet 简单聊天室 待机画面。

www.86duino.com

\$ COM30	II. care and		- C X
(Chat server address:192.180.70.2	00		2 Ernd
🐨 Autoscoil		No has enting 🝝	9900 bant 👻

图 15 Telnet 简单聊天室待机画面

如下图所示,读者可以看到本次实验-Telnet 简单聊天室,我们使用 Putty 通 讯软件,预备进行联机的画面。

Category:		64N
E Seesion	Basic options for your Pul	TTY eession
- Logging	Specify the destination you want to	connect to
Keyboard	Host Name (or IP address)	Port
Bel	192 168 30 200	23
Features	Connection type:	0 CCU
- Appearance - Behaviour - Translation	Load, save or delete a stored assau Saved Sessions	on
- Colours - Colours Connection - Deta Proxy - Tehet - Riogin	Default Settings ATMODE ATMODE9600 HC05 W5100 ess8226 xytable	* Loed Save Delete
B- Serial	Close window on est: Always © Never @ On	ly on clean exit

图 16 Telnet 简单聊天室预备进行联机的画面

我们可以使用 putty 通讯软件进行测试聊天室的功能,如下图所示,为联机 到简单聊天室客户端画面。

www.86duino.com

QM30	
P 192.168.30.200 - PuTTY	
Hello, client!	-
	-

图 17 联机到简单聊天室客户端画面

如下图所示,我们使用 Putty 通讯软件,进行联机后,读者可以看到本次实验-Telnet 简单聊天室之 Telnet 简单聊天室客户端联机的画面。

	COM30	
🚰 192 168 30 200 - PuTTY	1	Seed
Hello, client!	Chat server address:192.168.30.200 We have a new client	
	Iloussoul 💟	No line ending 🖌 9600 baud 🔹
		-

图 18 Telnet 简单聊天室客户端联机的画面

八、Telnet 多人版聊天室

上节中,我们已经建立一个简单而完整的『Telnet 简单聊天室』,但是,这 个聊天室太过于简陋,只能供一位客户端连入聊天室,所以我们要改进这样的问题。

首先,如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器 之网络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法



(b). 集线器网络接法

图 19 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将 86Duino EduCake 开发板的驱动程序安装好之后, 我们打开 86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥 写一段程序,如下表所示之 Telnet 简单聊天室测试程序,我们就可以让 86Duino EduCake 开发板变成一台专业版的多人聊天室运作了。

表 7 Telnet 简单多人版聊天室测试程序(AdvancedChatServer2)



86DUIND

0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF				
}; //使用这个 MAC ADRESS 当做这台 educake 的 MAC ADDRESS				
IPAddress ip(192, 168, 30, 200); //默认的网络 IP 地址, 读者可以自行更改为				
家中可用的地址				
IPAddress dnServer(168, 95, 1, 1); //预设的 DNS 服务器,本书为 Hinet 的 DNS				
服务器,读者可以自行更改为家中可用的 DNS 服务器				
// the router's gateway address:				
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或				
AP 的网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可用				
的网关的地址				
// the subnet:				
IPAddress subnet(255, 255, 255, 0); //子网掩码,本书为 Class C				
// telnet defaults to port 23				
EthernetServer server(23); //宣告一个以太网络服务器对象,并使用 Port23				
为 listening 通讯端口				
boolean alreadyConnected ; // whether or not the client was connected previously $ \ensuremath{\mathbb{K}}$				
机旗标				
boolean ConnectedFlag[10]; // whether or not the client was connected previously 联				
机旗标				
int connectNo = 0; //联机号码				
EthernetClient client; //产生一个倾听的以太网络客户网络对象				
EthernetClient Connectclient[10]; //产生十个以太网络客户网络对象				
<pre>void setup() {</pre>				
// initialize the ethernet device				
Ethernet.begin(mac, ip, gateway, subnet); //启动以太网络联机				
// start listening for clients				
server.begin(); //启动 Port23 以太网络联机				

```
// Open serial communications and wait for port to open:
                       //宣告监控画面的通讯速率
  Serial.begin(9600);
    initConnectingFlag(); // 初始化联机旗标
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
  Serial.print("Chat server address:");
                                    //印出 Chat server address:
                                //印出网址
  Serial.println(Ethernet.localIP());
  DisplayConnectingStatus() ;
                                        //显示联机状况
}
void loop() {
 // wait for a new client:
                               //产生倾听的以太网络客户网络对象
    client = server.available();
 // when the client sends the first byte, say hello:
               //当以太网络客户网络对象有人连入
  if (client) {
    connectNo = 0;
    while (connectNo <10) //循环十组联机
    {
        if (!ConnectedFlag[connectNo]) { //如果是新联机
          // clead out the input buffer:
              Connectclient[connectNo] =client;
            Connectclient[connectNo].flush(); //以太网络客户网络对象 更
新缓冲区数据
          Serial.println("We have a new client"); //印出有新联机
          client.println("Hello, client!"); //送出 Hello 欢迎词
          ConnectedFlag[connectNo] = true; //设定已联机
          break;
```

```
connectNo ++;
    }
  }
      connectNo = 0;
        while (connectNo <10)
    {
        if (!Connectclient[connectNo].connected()) //检查是否断线
               ConnectedFlag[connectNo] = false;
                                                //设定联机旗标参数
=false(未联机)
        if (Connectclient[connectNo].available() > 0)
            //如果以太网络客户网络对象有数据送出
        {
          // clead out the input buffer:
          char thisChar = Connectclient[connectNo].read();
                                                          //读取以太网络
客户网络对象送出的数据到 thisChar 变量
          // echo the bytes back to the client:
                                     //印出联机中
          Serial.print("Connect ");
                                     //印出第几号联机
          Serial.print(connectNo);
                            //印出":"
          Serial.print(":");
          server.write(thisChar);
                                    //送出 thisChar 变量到 server 对象
          // echo the bytes to the server as well:
          Serial.write(thisChar);
                                 //印出 thisChar 变数
        }
       connectNo ++ ;
}
void initConnectingFlag()
        //初始化联机旗标参数
```

```
for(connectNo=0 ; connectNo < 10 ; connectNo++)</pre>
  {
    ConnectedFlag[connectNo] = false; //设定联机旗标参数=false(未联
机)
  }
}
void DisplayConnectingStatus()
 //显示联机状况
{
  for(connectNo=0 ; connectNo < 10 ; connectNo++)</pre>
  {
    if (ConnectedFlag[connectNo])
                                    //如果联机中
    {
        Serial.print("Connection ");
                                     //该区显示联机中的信息
        Serial.print(connectNo) ;
        Serial.print(": Connected \n");
            //如果未联机中
    else
        Serial.print("Connection"); //该区显示等待联机中的信息
        Serial.print(connectNo) ;
        Serial.print(": Waiting Connecting \n");
    }
 // Connectclient[connectNo] = server.available(); //产生一个以太网络客户
网络对象
  }
```

如下图所示,读者可以看到本次实验-Telnet 多人版聊天室,主机开始的画面。



图 20 Telnet 聊天室多人版主机画面

如下图所示,读者可以看到本次实验-Telnet 简单多人版聊天室,我们使用 Putty 通讯软件,预备进行联机的画面。



图 21 Telnet 简单聊天室多人版预备进行联机的画面

www.86duino.com

如下图所示,读者可以看到本次实验-Telnet 简单聊天室,我们使用 Putty 通 讯软件,第一位 Telnet 简单聊天室多人版联机中的画面。



图 22 第一位 Telnet 简单聊天室多人版联机中的画面

如下图所示,读者可以看到本次实验-Telnet简单聊天室多人版,我们使用 Putty通讯软件,第二位Telnet简单聊天室多人版进行的画面。



图 23 第二位 Telnet 简单聊天室多人版进行的画面

九、温度监控

如果我们要量测温度,我们可以使用温度传感器,如果我们又要量测湿度, 我们也可以使用量测传感器,这样我们会需要很多的传感器,所以本文介绍温湿 度感测模块(DHT11),如下图所示,它主要是使用 DHT-11 作成温湿度感测模块。



图 24 DHT11 温湿度感测模块

本实验是采用 DHT11 温湿度感测模块,如下图所示,由于 DHT-11 温湿度 传感器需要搭配基本量测电路,所以我们使用 DHT11 温湿度感测模块来当实验 主体,并不另外组立基本量测电路。

如下图所示,先参考 DHT11 温湿度感测模块脚位接法,在遵照下表之 DHT11 温湿度感测模块接脚表进行电路组装。



图 25DHT11 温湿度感测模块脚位图

www.86duino.com



表 8DHT11 温湿度感测模块接脚表

数据源: Arduino 程序教学(常用模块篇):Arduino Programming (37 Sensor Modules)(曹永忠, 許智誠, & 蔡英德, 2015b, 2015f)

我们遵照前几章所述,将86Duino EduCake 开发板的驱动程序安装好之后, 我们打开86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,编 写一段程序,如下表所示之 DHT11 温湿度感测模块测试程序,我们就可以透过 DHT11 温湿度感测模块来侦测任何温度与湿度。

表 9 DHT11 温湿度感测模块测试程序(DHT11)

```
int DHpin=7;
byte dat[5];
byte read_data()
{
    byte data;
    for(int i=0; i<8;i++)
    {
        if(digitalRead(DHpin)==LOW)
        {
```





www.86duino.com

	Serial.print(dat[2], DEC);	//显示温度的整数
字;		
	Serial.print('.');	
	Serial.print(dat[3],DEC);	//显示温度的小数
字;		
	Serial.println('C');	
	delay(700);	
}		

参考数据: DMP 商店(<u>http://shop.dmp.com.tw/INT/products/67</u>)

613	4	HT11 86Duine Coding 210		-
No.	File	Edit Sketch Tools Help		
0				
COM30				x
				Send
MII, OK,	39.0,	3.0		
111, OK,	39.0,	3.0		
П11, OX,	39.0,	3.0		
			production and the production	

如下图所示,我们可以看到温度感测模块测试程序结果画面。

图 26 温度感测模块测试程序结果画面

上面的的程序代码因为使用低阶的控制讯号,对于一般人太吃力,所以作者 将会将这些程序代码转化成函式库(Library)的方式来攥写程序代码。

十、实做温湿度监控网站

本章前面所叙述之内容,主要是让读者可以了解 86Duino EduCake 开发板 如何使用网络资源,如何使用 Telnet、Http 的通讯协议,甚至用这些网络资源来 建立一个简单的网页服务器、聊天服务器等,这些无异是为本书最后的项目打基 础,在这之前,我们先来做一个简单的物联网的小项目,如果我们建立一个温湿 度读取的装置,使其定时读取温度并已经建立一个简单而完整的『温湿度监控网 站』,如此一来,所有的使用者就可以使用浏览器,透过因特网来取得温度的信 息了。

首先,如下图所示,我们使用一般 Cate 5 的网络线,配合 RJ45 接头,一端接入 86Duino EduCake 开发板侧边网络接口(如下图.(a)),一端接入网络集线器 之网络接口(如下图.(b)),完成 86Duino EduCake 开发板网络实体线路连接。



(a). EduCake 网络接法

(b). 集线器网络接法

图 27 86Duino EduCake 开发板之网络接法示意图

我们遵照前几章所述,将86Duino EduCake 开发板的驱动程序安装好之后, 我们打开86Duino EduCake 开发板的开发工具: Sketch IDE 整合开发软件,攥 写一段程序,如下表所示之实做温湿度监控网站测试程序,我们就可以让 86Duino EduCake 开发板变成一台的温度监控网站服务器运作了。

表 10 实做温湿度监控网站测试程序(TempMonitorServer)

#include <spi.h></spi.h>				
#include <ethernet.h> // 使用网络必需要用的</ethernet.h>				
#include <wire.h> // 使用网络必需要用的</wire.h>				
#include "dht.h" //使用 DHT11 温湿度传感器所需要的函式库				
#define DHT11_PIN 7 //宣告 DHT11 温湿度传感器通讯脚位				
dht DHT; //宣告 DHT11 温湿度传感器对象				
// 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				
}; //使用这个 MAC ADRESS 当做这台 educake 的 MAC ADDRESS				
IPAddress ip(192, 168, 30, 200); //默认的网络 IP 地址,读者可以自行更改为				
家中可用的地址				
IPAddress dnServer(168, 95, 1, 1); //预设的 DNS 服务器,本书为 Hinet 的 DNS				
服务器,读者可以自行更改为家中可用的 DNS 服务器				
// the router's gateway address:				
IPAddress gateway(192, 168, 30, 254); //默认的网关的地址(就是 Rounter 或				
AP 的网址),本书为作者写书环境的网关的地址,读者可以自行更改为家中可用				
的网关的地址				
// the subnet:				
IPAddress subnet(255, 255, 255, 0); //子网掩码,本书为 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); //宣告服务器并使用 Port 80 为通讯端口				

86Duino

```
void setup() {
 // put your setup code here, to run once:
 Serial.begin(9600);
                        //宣告监控画面的通讯速率
 while (!Serial) {
   ; // wait for serial port to connect. Needed for Leonardo only
 }
 int chk = DHT.read11(DHT11 PIN); //读取 DHT11 温湿度传感器 状态资
料,并检查是否可以使用
  if (chkDHT(chk) == 0) // 使用 chkDHT 函式检查是否 DHT11 温湿度传
感器 状态数据是否正常,不正常传回0,正常传回1
      {
                                                //告知使用者
         Serial.println("ERROR on init DHT Sensor");
                                    DUING
DHT11 温湿度传感器不正常
                         //程序终止
         while (true);
 // start the Ethernet connection and the server:
 Ethernet.begin(mac, ip, dnServer, gateway, subnet); //使用上述宣告的网络设
定来启动 WEB 服务器
 server.begin(); //开启 Web 服务器运作
 Serial.print("server is at "); //印出服务器信息
                              //印出服务器 IP 地址
 Serial.println(Ethernet.localIP());
}
void loop() {
  EthernetClient client = server.available(); //如果有人连入WEB服务器,
启动一个 Client 联机端
 if (client) { //如果启动一个 Client 联机端成功
```

	Construction of the Annual Architecture				
Serial.println("new client");	//印出"新联机"				
// an http request ends with a blank line					
boolean currentLineIsBlank = true	·				
while (client.connected()) {	//如果联机成功				
<pre>if (client.available()) {</pre>	//如果联机的 Client 联机端有送数据过来				
<pre>char c = client.read();</pre>	//读入联机的 Client 联机端数据到变量 c				
Serial.write(c);	//印出变数 c				
// if you've gotten to the end of	of the line (received a newline				
// character) and the line is bl	ank, the http request has ended,				
// so you can send a reply					
if (c == '\n' && currentLineIs	Blank) { //如果变量 c 是换行数据而且				
目前数据为空					
// send a standard http resp	onse header				
client.println("HTTP/1.1 2	00 OK"); //送出 Http 标头数据(固定要				
送)					
client.println("Content-Typ	pe: text/html"); //送出 Http 标头数据(固				
定要送)					
client.println("Connection:	close"); // the connection will be closed af-				
ter completion of the response //送	出Http 标头数据(固定要送)				
client.println("Refresh: 5")	; // refresh the page automatically every 5				
sec //送出 Http 标头数据(固定要送)					
client.println(); //送	出Http 标头数据(固定要送)				
client.println(" DOCTYP</td <td>EHTML>"); //送出Http 标头数据(固</td>	EHTML>"); //送出Http 标头数据(固				
定要送)					
<pre>client.println("<html>");</html></pre>	//送出 Http 标头数据(固定要送)				
// output the value of each	analog input pin				
client.print("Humidity: '	'); //送出目前湿度提示字句				
client.print(DHT.humid	ity, 1); //送出目前湿度				
client println(" "):	//送出网页换行键				

```
client.print("Temperature: "); //送出目前温度提示字句
          client.print(DHT.temperature, 1);
                                       //送出目前温度
           client.println("<br>");
                                    //送出网页换行键
         client.println("</html>"); //送出网页 Tag 数据(固定要送)
         break;
       }
       if (c == '\n') { //新的一行
         // you're starting a new line
         currentLineIsBlank = true;
       }
       // you've gotten a character on the current line
         currentLineIsBlank = false;
                                      DUIND
   // give the web browser time to receive the data
   delay(1);
              //延迟一毫秒
   // close the connection:
                  //关闭联机的 Client 联机端(必需要关闭,不然会挂掉)
   client.stop();
                                        //送出联机断线
   Serial.println("client disconnected");
 }
 delay(2000); //延迟 2000 毫秒
}
unsigned int chkDHT( int chk ) //检核 DHT11 温湿度传感器 状态数据是否
正确
```

www.86duino.com

switch (chk)	//检查 chk 变数;多重检查
{	
case DHTLIB	_OK: //如果是 ok 状态
	Serial.println("DHT init is OK,\t"); //印出 ok 状态
	return 1;
case DHTLIB	_ERROR_CHECKSUM: //如果是数值加总不正确状态
	Serial.println("DHT Checksum error,\t"); //印出数值加总
不正确状态	
	return 0;
case DHTLIB	_ERROR_TIMEOUT : //如果是在固定时间读不到正确数
据	
	Serial.println("DHT Time out error,\t"); //印出在固定时间读
不到正确数据	
	return 0;
default:	//不知名错误
	Serial.println("DHT Unknown error,\t"); //不知名错误
	return 0;
}	
}	

如下图所示,读者可以看到本次实验-实做温湿度监控网站监控主画面。

www.86duino.com

🖇 LightMonitorServer 86	5Duino Coding 210 🗖	TempMonitorServer 86Duino Coding 210	
Flo Edit Sketch Tools H	lalp.	File Edit Sketch Tools Help	
	(OO NOO	Serial Monitor 🔎
LichtMonitorServer	COM30		
	1	Sent	
d lister for incomme	DHT init is DK,		
historial and client	005.06.361.192.168		
(client) j			
Sector Contractory		194	
currentLizels			
while (client) where		18.5	("):
Citizent anno 101			
c = citert.			in the second
second and the second			5.85
		1.1.	16
10 = 0' hk			1
classic			1
	Manzarall	No line ending + 9600 hand +	, *
	1	The subleacing	
		1105 version: QuaveCabe 0.97	
		Tplouding the tivery sketch	Done
Gr-112		-10	SCOulor EduCate on COM30
a	Zimmedt that		

图 28 温湿度监控主机画面

如下图所示,我们使用浏览器连入温湿度监控主机,读者看到浏览器可以读出温湿度监控装置的温湿度信息的画面。



图 29 用浏览器连入温湿度监控主机

如下图所示,我们可以看到温湿度监控主机响应浏览器端,所响应的HTML 内容,可以让使用者端的浏览器读出温湿度监控装置的温湿度信息。

www.86duino.com

S LightMonitorServer 86Duino Coding 210	COM3	10	COLUMN STREET, SAME	a design and the second se	- 🗆 🗙	
File Edit Sketch Tools Heip					Stad	
	ace clies				- December of	
LightMontlorBerver	GET / HEI	P/J.I			1	
<pre>// listm for recoming elicits // listm for recoming elicits (client) # // listm for recommendation (client,</pre>	Hest: 192 Connection Dache Con Kocept: 4 Upgrade 1 User-Agen Midacar: 1 Kocept-Im Kocept-Im	Heat: 192.192.30.200 Connection: keep-alive Dachd Control: nor ege-0 Accept: text/html.application/xhtml+xml.application/xml;g=0.5,image/webg,*/*:g=0.8 Upgmade-Instance-Requests: 1 User-Agent: Hosilla/5.0 (finices NI 5.1; ND%64) Apple/ebKi4/537.36 (DHDML, like Gecko) (Michaer: http://Uke.108.30.200/ Accept-Incoing: grip, deflate, sich Accept-Incoing: grip, deflate, sich Accept-Incoing: grip, deflate, sich Accept-Incoing: grip, deflate, sich Accept-Incoing: grip, deflate, sich				
The data wards applied to call and all call library					-	
The second second second	4		81		+	
(c — 'm Ak currenfluce)(Black) (W Anton	1000		No line and ing 🚽	👻 based 00090	
clientiger client (http://www.icensectory.	-	1				
4		4	111	13	•	
		Done upload	ina			
		BIOS Version	SucceSte 0.97			
24/14/2	152	Bilburo Editara or CURIO				

图 30 温湿度监控主机回馈浏览器画面

INT

十一、章节小结

本章主要介绍 86Duino EduCake 开发板,透过内建的以太网络来传送、接收温湿度数据并透过浏览器显示信息。