1 Temperature sent to Cloud Server by using Arduino and WiFi via HTTP Protocol

1.1 Problem presentation: how to use Arduino and wifi for sending temperature to the remote cloud server

In this section, we will give a presentation how can we send the sensed temperature data to the remote cloud server by using Arduino control board (Cduino) and WiFi module. Then, by login into the distributed clients, you can check the temperature around your lovers at any countries in the remote Atlantic side. Certainly, if using many types of sensors, you can check and read many other required information.

31.2 Hardware and software

The required materials in this experiment are shown in Table 1-1.

Table 1-1: the required materials

number	name	quantity	function	note
1	Arduino software	1	platform	
2	server	1	Collection data	
3	WiFi module	1	Wireless communication	
4	Arduino shield board	1	Connection wifi	All version
5	USB to serial RS232	1	conversion	
6	5V/1A voltage adapter	1	voltage	
7	Antenna	1	Transmit wifi signal	optional
8	HTC mobile phone	1	Wifi hot spot	
9	LM35	1	Collect Temperature	
10	Breadboard and line	several	connection	
11	Cduino	1	Development board	

The hardware materials can be seen in Figrue 1-1. All of these modules can be found at www.smartarduino.com, where, the wifi module is updated the new one. The Arduino kit can be found at SmartArduino.

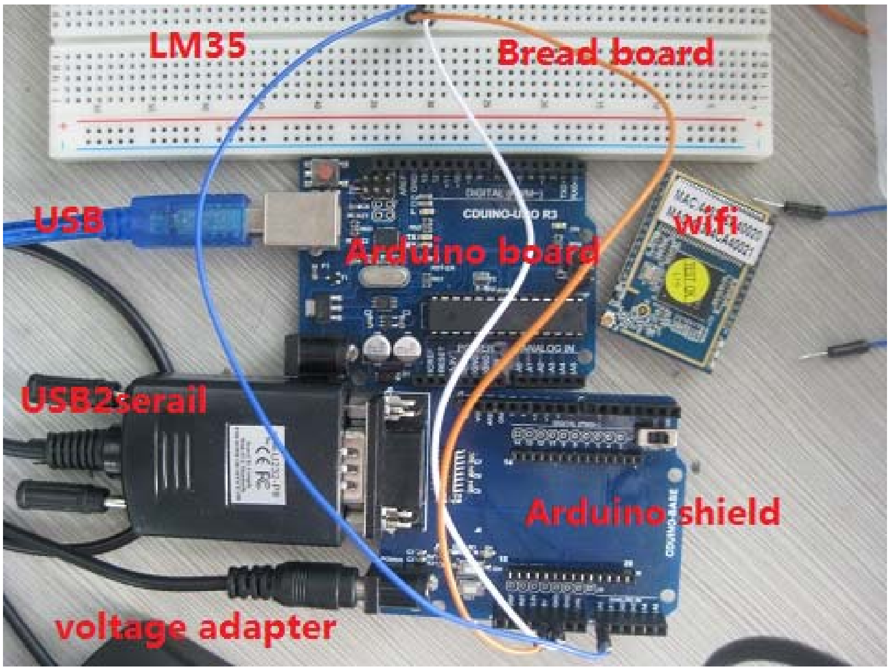


Figure 1-1 Hardware

1.3 Work principle

The work principle for this experiment can be seen in Figure 1-2.

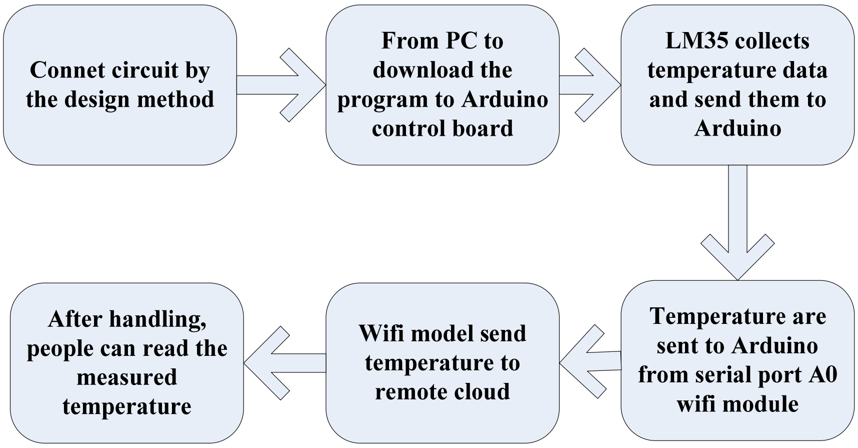

Connet circuit by the design method
From PC to download the program to Arduino control board
LM35 collects temperature data and send them to Arduino
Temperature are sent to Arduino from serial port A0 wifi module
Wifi model send temperature to remote cloud
After handling, people can read the measured temperature

Figure 1-2 working principle

1.4 Experimental analysis

(1) **draw the experimental schematics**: According to the design idea, you can draw the experimental schematics, which is shown in Figure 1-3, and this principle of LM35 is the same as Section 8 from the Basic Part. To readers' convenience, this figure is given here again. Compared with this experiment, it is short of Arduino shield board and wifi board. But the principle of collecting temperature by LM35 is the same as Section 8 in the Basic Part.

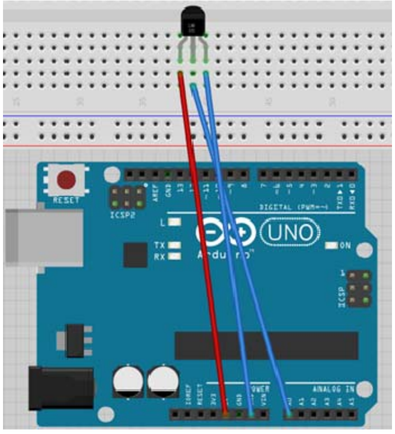


Figure 1-3 Experimental principle

(2) **circuit connection:** By the experimental principle, we firstly connect the circuit shown in Figure 1-4, where, the WiFi module has been updated into the new one, and a omni antenna is add for the better WiFi signal.

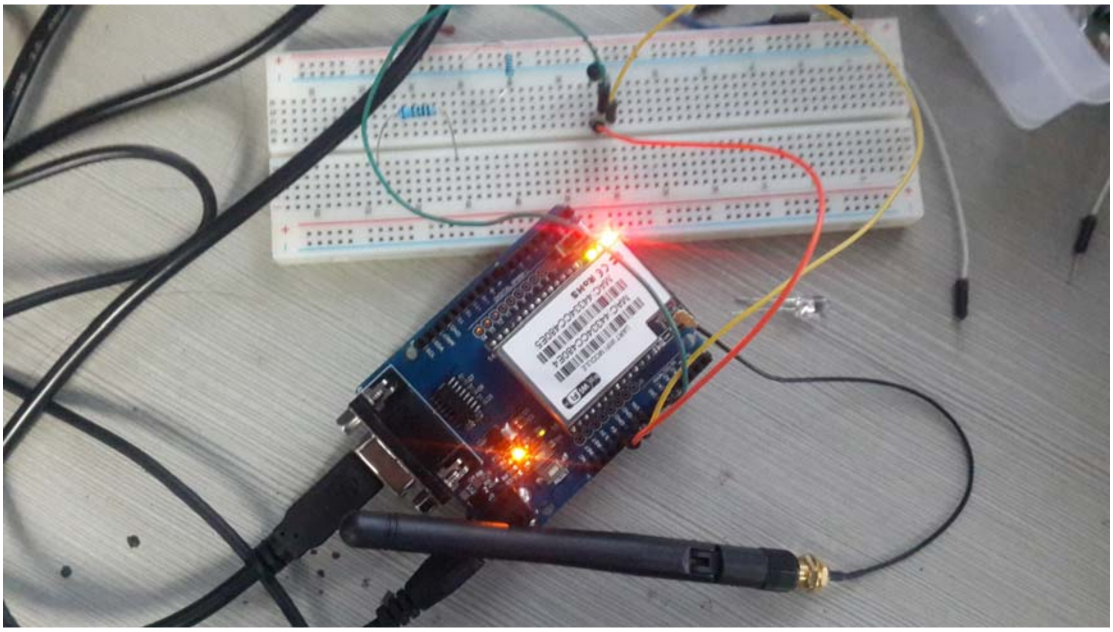
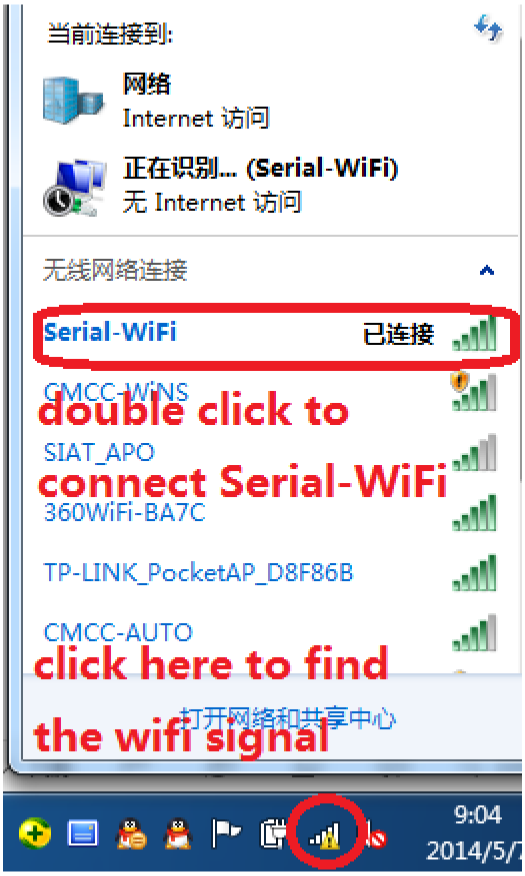


Figure 1-4 Experimental connection

(3) **wifi network connection:** when we power the wifi on the Arduino main board, we can get the wifi signal. But, for the stability of wifi signal, we should power the wifi shield board by the dedicated voltage adapter shown in Table 1-1. Note that, the wifi module could not be inserted in a wrong direction. If right, then the three led lights will be light (one red and other two is green); if wrong, just ONLY one red led is light. Or, you can connect it by Fig. 1-4.

If the inserted wifi module is right, after a short while, you can search a wifi signal in your compter named as **Serial-WiFi** transmitted by wifi module. Then double click the wifi network serial-wifi, and input the password (the default password is 12345678) shown in Figure 1-5. In general, it is not necessary to input the password. Then, the serial-wifi network is connected.

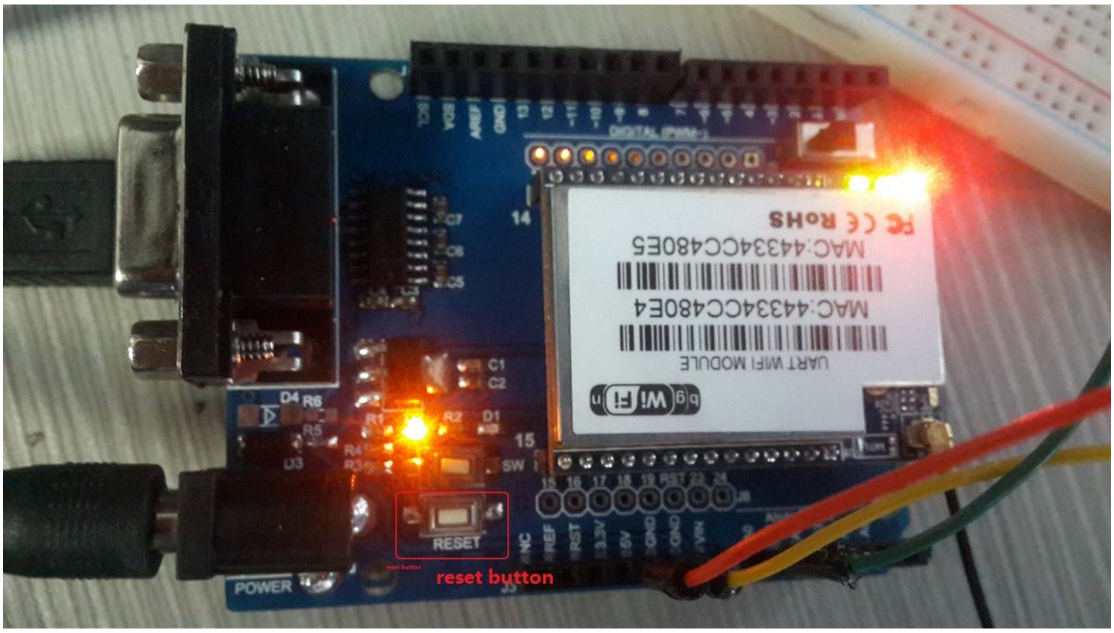


double click to connect Serial-WiFi

click here to find the wifi signal

Figure 1-5 Serial-WiFi connection

Note that, if you use the wifi module at the first time, to ensure the exactness of data configuration, wifi module needs to recover to the default factory settings (press the button RST on the Cduino shield board for 6 seconds, then power off and reboot for the wifi module, as shown in Figure 1-6. Also can see the method in Section 2).

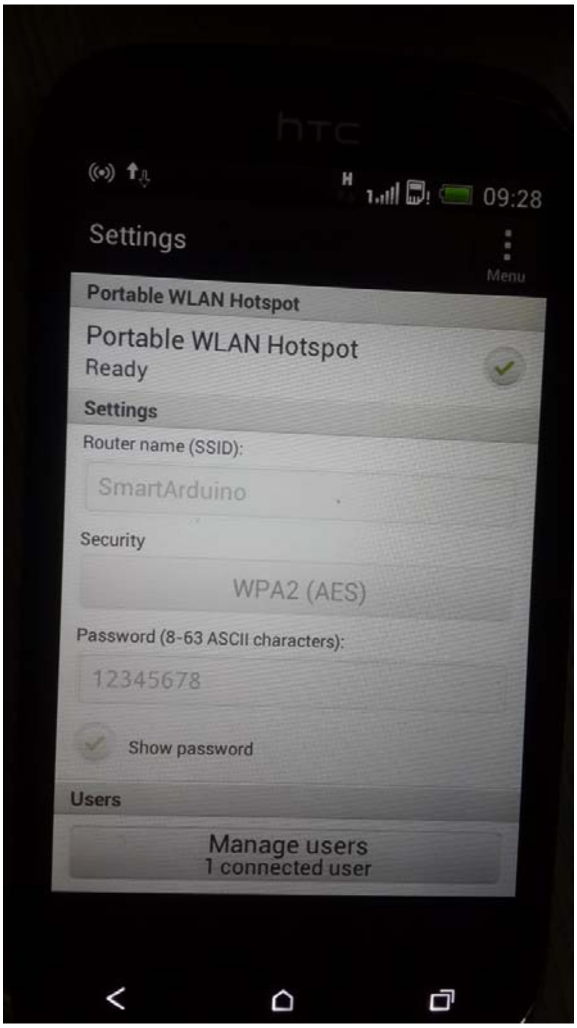


reset button

Figure 1-6 Reset wifi module to the factory default setting.

This step is similar to open the wifi in our phone to surf the internet. It shows that the wifi module can transmit the wifi signal. Therefore, if wifi module connects a wifi hot spot, then we can let data send by wifi communication.

(4) **set up wifi hot spot**: Since there is no other wifi hot spot in our such experimental environment, then we use OUR HTC mobile phone to set up a wifi hot spot (wifi router is suitable). After this, temperature data can be sent by wifi to the cloud server. At first, open the "WLAN hot spot" in the mobile phone. At the same time, remember to open 3G network signal (very important), which can make temperature data collected by LM35 send to server in the remote cloud. In this experiment, the WLAN hot spot name in my HTC phone is set as "SmartArduino" for the mobile SSID, and password is "12345678", as shown in Figure 1-7. You can also use the default name and password. However, for the input convenience, you had better to modify the default name and password.



Portable WLAN Hotspot

Portable WLAN Hotspot
Ready ✓

Settings

Router name (SSID):
SmartArduino

Security
WPA2 (AES)

Password (8-63 ASCII characters):
12345678

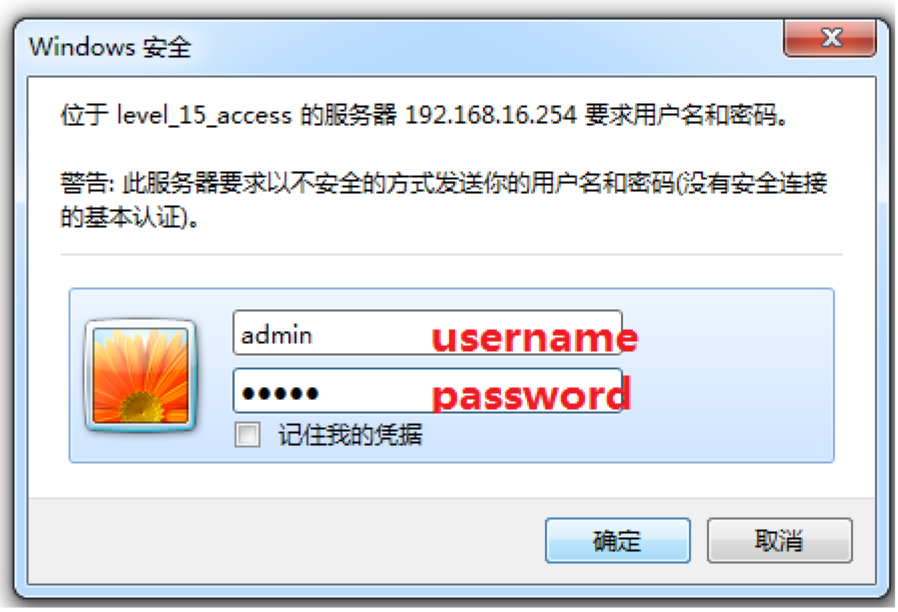
✓ Show password

Users

Manage users
1 connected user

Figure 1-7 wifi hot spot by HTC phone

(5) network configuration: This step is very important. If right, then you can do a successful experiment. When the computer finds and connects the wifi network Serial-WiFi from the wifi module successfully, you can input the website "192.168.16.254" in the browser, and input the user name and password (both are "admin") to configure the web data, which is shown in Figure 1-8.



Windows 安全

位于 level_15_access 的服务器 192.168.16.254 要求用户名和密码。

警告: 此服务器要求以不安全的方式发送你的用户名和密码(没有安全连接的基本认证)。

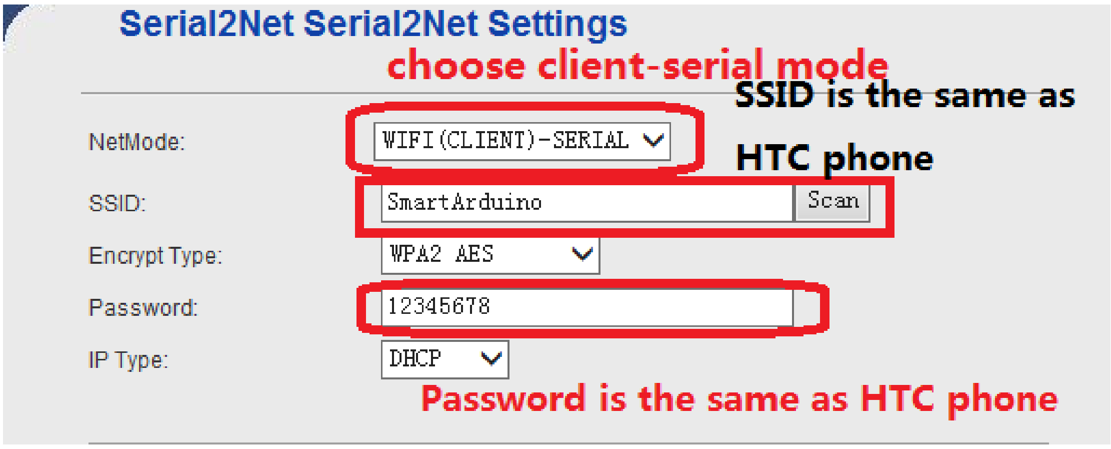
admin

●●●●●

□ 记住我的凭据

Figure 1-8 Login into the wifi configure system.

After login into the wifi configure system, we can set the wifi module some parameters. The detailed parameters and information is shown in Figure 1-9.



Serial2Net Serial2Net Settings

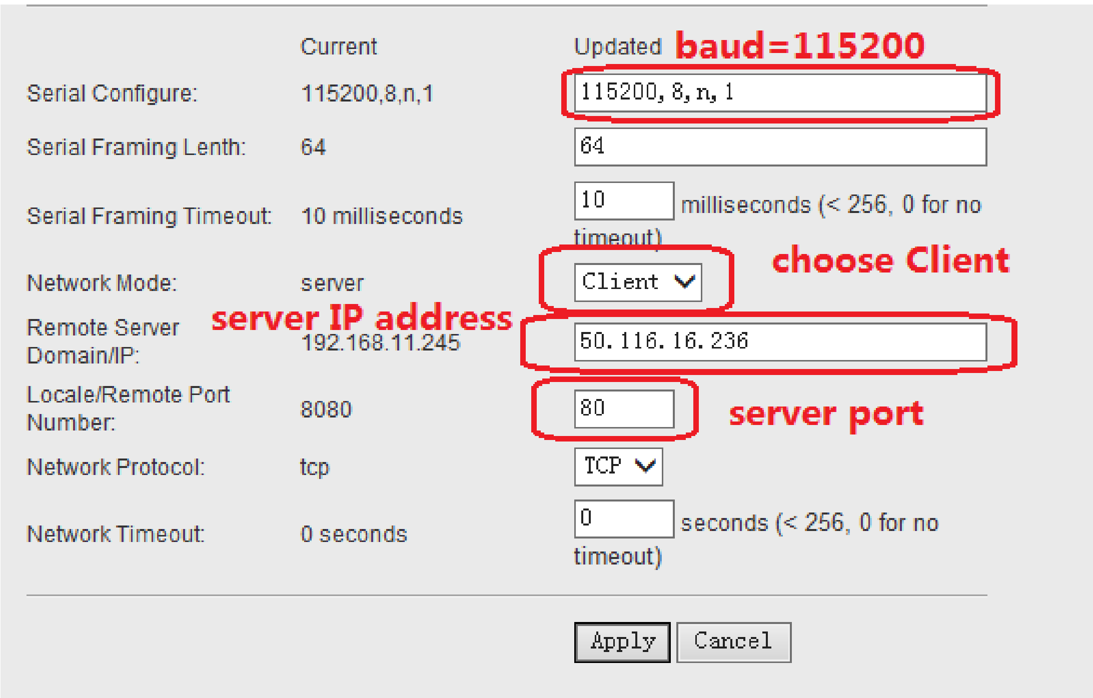
NetMode: WIFI(CLIENT)-SERIAL

SSID: SmartArduino

Encrypt Type: WPA2 AES

Password: 12345678

IP Type: DHCP



	Current	Updated
Serial Configure:	115200,8,n,1	115200,8,n,1
Serial Framing Lenth:	64	64
Serial Framing Timeout:	10 milliseconds	10 milliseconds (< 256, 0 for no timeout)
Network Mode:	server	Client
Remote Server Domain/IP:	192.168.11.245	50.116.16.236
Locale/Remote Port Number:	8080	80
Network Protocol:	tcp	TCP
Network Timeout:	0 seconds	0 seconds (< 256, 0 for no timeout)

baud=115200

choose Client

server IP address

server port

Figure 1-9 Serial-WiFi network configurations

After set, then click apply, then my HTC phone will find a connection, which is shown in Figure 1-7. When configure the network data parameters, please specially note that the marked by red rectangle, and others can be default settings. Some notes are given as follows.

1) At this time, wifi module can be viewed as a collection and transmission unite, which can send the collected temperature data to the server on the remote distributed cloud.
2) SSID and Password are the same as the SSID and Password from the mobile phone (or wifi router).
3) Remote ServerDomian/IP: It is pointed to the server IP address where the temperature data are stored at. In our such experiment, the IP address is 50.116.16.236, which is corresponding to the IoT system (www.IoT.fm). Anyone can use this server to do their own experiment.
4) Locale/Remote Port: the server port number, here is 80.

After configure, click "Apply". At this time, if you check your phone, there is a "1 user connects" in the bottom of the "WLAN hot spot", as shown in Figure 1-7. This implies that, the wifi module on the Arduino board has been connected the mobile hot spot. That is, the wifi module has been connected the internet. Then the sensed temperature data by LM35 can be sent to the cloud server.

(6) Arduino code

```
//the sensed temperature data by LM35 is sent to the remote server 50.116.16.236 (www.IoT.fm)
void setup()
{
  Serial.begin(115200);
}
void loop()
```

```
{
    int n=analogRead(A0);
    float vol=n*(5.0/1023*100);
    upload_sensor(vol);
    delay(5000);
}

void upload_sensor(float vol)
{
        // send the HTTP PUT request:
        char buf[200];
        memset(buf,0,200);
        int ret;
        ret=sprintf(buf,"GET
/upload.php?uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&da
ta=");
//here, uid=demo, and the key is encrypted by “Get API Key” on the IoT system
        Serial.print(buf);
        Serial.print(vol);
        Serial.println(" HTTP/1.1");
        Serial.println("Host: api.iot.fm"); //the remote server
        Serial.println("Connection: close");
        Serial.println();
 }
```

Then you can download the above code to the Arduino control board. Note that, there is a switch. You should let the switch on the outside (O) when downloading the Arduino code. This is because the serial connection may affect the data download, which is shown in Figure 1-10.

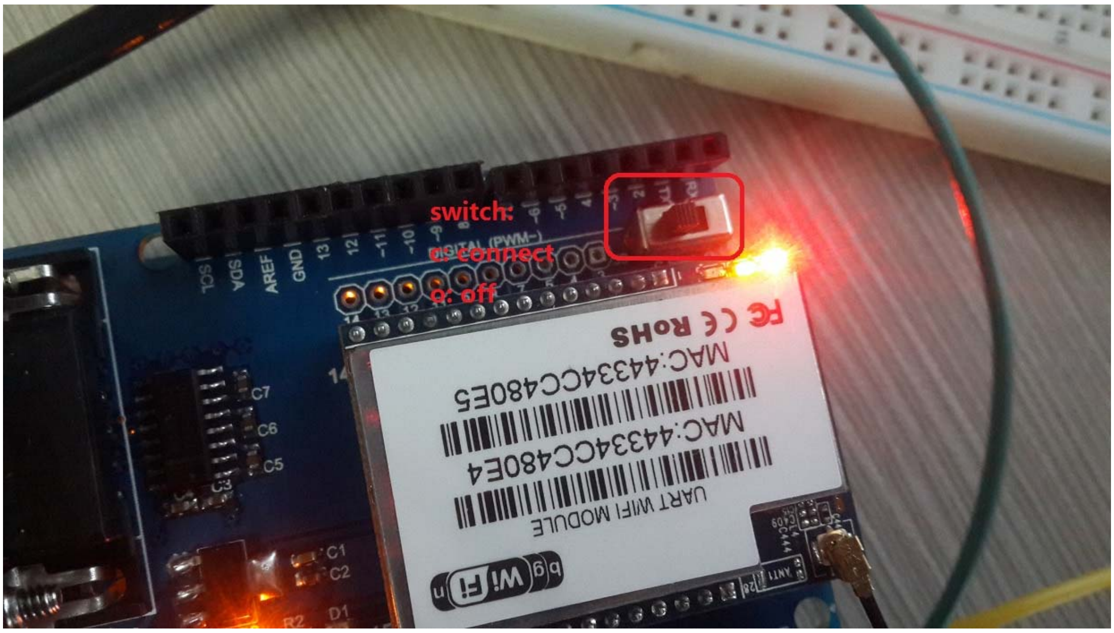
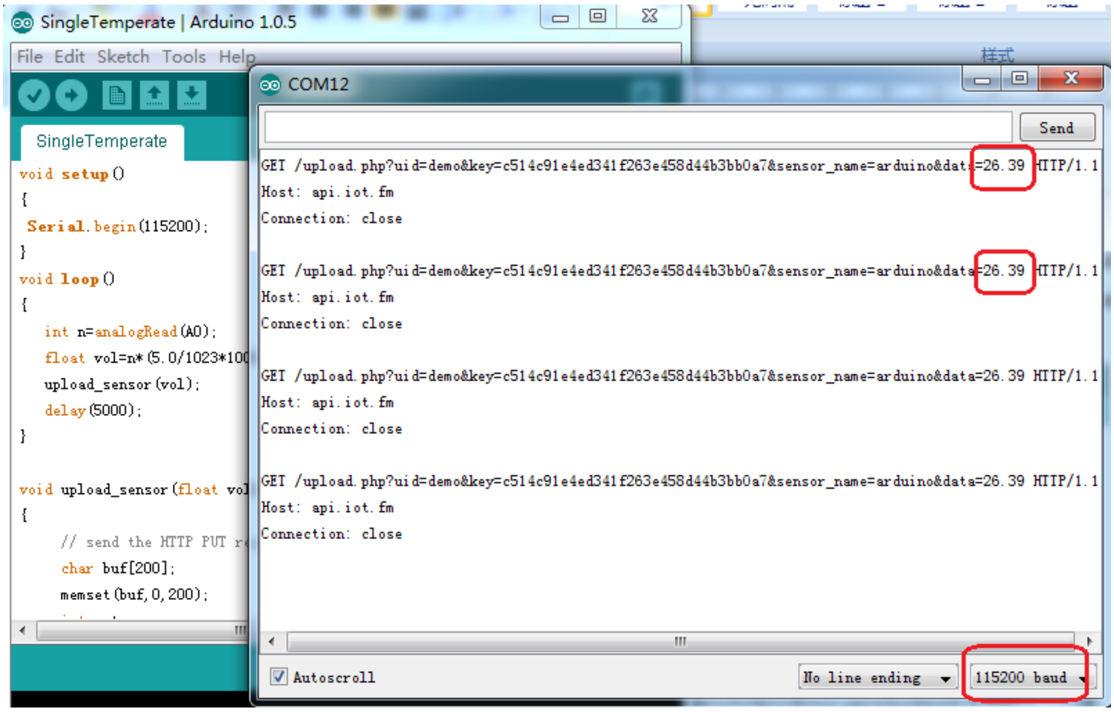


Figure 1-10 Switch on the Cduino base board

At the same time, you can track the data transmission on the monitor at the Arduino soft-platform, which is shown in Figure 1-11 marked by the red label. The temperature is 26.39, and this data has been transmitted to the remote server cloud by wifi networks.



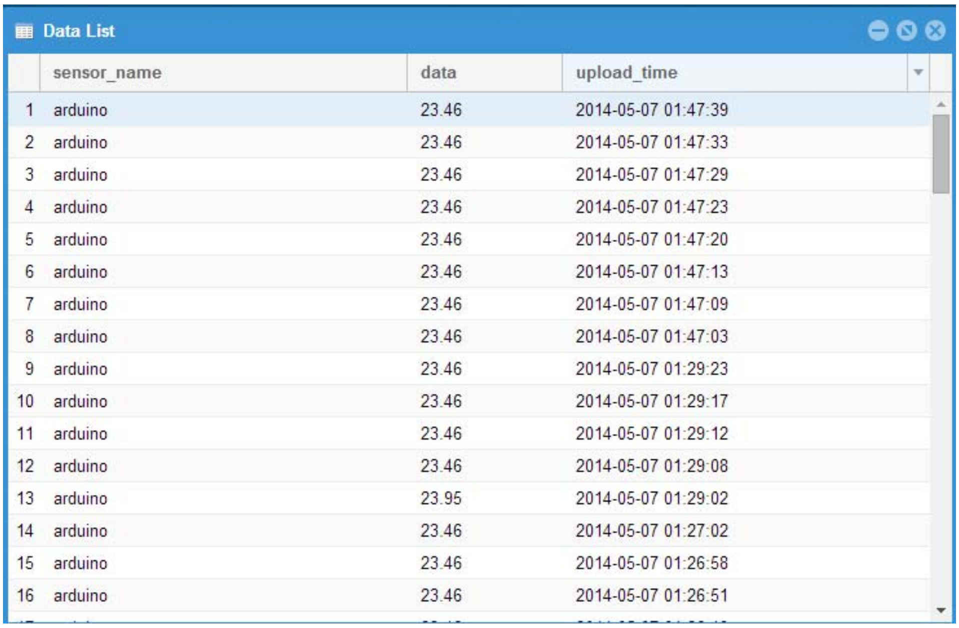
```
void setup()
{
 Serial.begin(115200);
}
void loop()
{
   int n=analogRead(A0);
   float vol=n*(5.0/1023*100
   upload_sensor(vol);
   delay(5000);
}

void upload_sensor(float vol
{
    // send the HTTP PUT r
    char buf[200];
    memset(buf,0,200);
```

```
GET /upload.php?uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=26.39 HTTP/1.1
Host: api.iot.fm
Connection: close

GET /upload.php?uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=26.39 HTTP/1.1
Host: api.iot.fm
Connection: close

GET /upload.php?uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=26.39 HTTP/1.1
Host: api.iot.fm
Connection: close

GET /upload.php?uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=26.39 HTTP/1.1
Host: api.iot.fm
Connection: close
```

☑ Autoscroll   No line ending   115200 baud

Figure 1-11 Data display on the Arduino serial monitor

(7) **data connection at the end:** if you finished the above six steps successfully, congratulations, you are successful. The temperature sensed by sensor LM35 has been successfully sent to the server on the internet by wifi module. The related temperature data can be exhibited in Figrue 1-12, which is obtained by click "Data List" in the IoT system. Note that, when you want to checkt the new sensed data from Arduino, you must refresh the IoT system to get the new sensed data.



	sensor_name	data	upload_time
1	arduino	23.46	2014-05-07 01:47:39
2	arduino	23.46	2014-05-07 01:47:33
3	arduino	23.46	2014-05-07 01:47:29
4	arduino	23.46	2014-05-07 01:47:23
5	arduino	23.46	2014-05-07 01:47:20
6	arduino	23.46	2014-05-07 01:47:13
7	arduino	23.46	2014-05-07 01:47:09
8	arduino	23.46	2014-05-07 01:47:03
9	arduino	23.46	2014-05-07 01:29:23
10	arduino	23.46	2014-05-07 01:29:17
11	arduino	23.46	2014-05-07 01:29:12
12	arduino	23.46	2014-05-07 01:29:08
13	arduino	23.95	2014-05-07 01:29:02
14	arduino	23.46	2014-05-07 01:27:02
15	arduino	23.46	2014-05-07 01:26:58
16	arduino	23.46	2014-05-07 01:26:51

Figure 1-12 Data List at server

1.5 Key notes and summaries

(1) Before configuring network parameters, to ensure the successful configuration, please

recovery the wifi module to the default factory settings.

(2) Please ensure the exactness of data configuration, especially the settings about mobile phone and server IP address.