Chapter Two

A Brief TCP Communication between Arduino and Remote Server

2.1 Problem presentation: How to transmit the sensed temperature by LM35 to the remote server by TCP protocol based on Arduino platform.

In this chapter, based on the Arduino UNO R3 development board, we will realize the function that the sensed temperature data can be sent to the remote server by using wifi module. Furthermore, we can control the data to send or not. Similarly, by using this TCP communication protocol, you can control the network-based electronic devices at anywhere through your phone or other network communication ways (e.g., wifi).

2.2 Hardware

The required materials in this experiment are shown in Table 2-1.

Table 1-1: the required materials

number	name	quantity	function	note
1	Arduino software	1	platform	
2	server	1	manager 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	Samsung 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 2-1. 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 (http://www.smartarduino.com/arduino-development-board-cduino-base-power-adapter-omni_p94247.html).

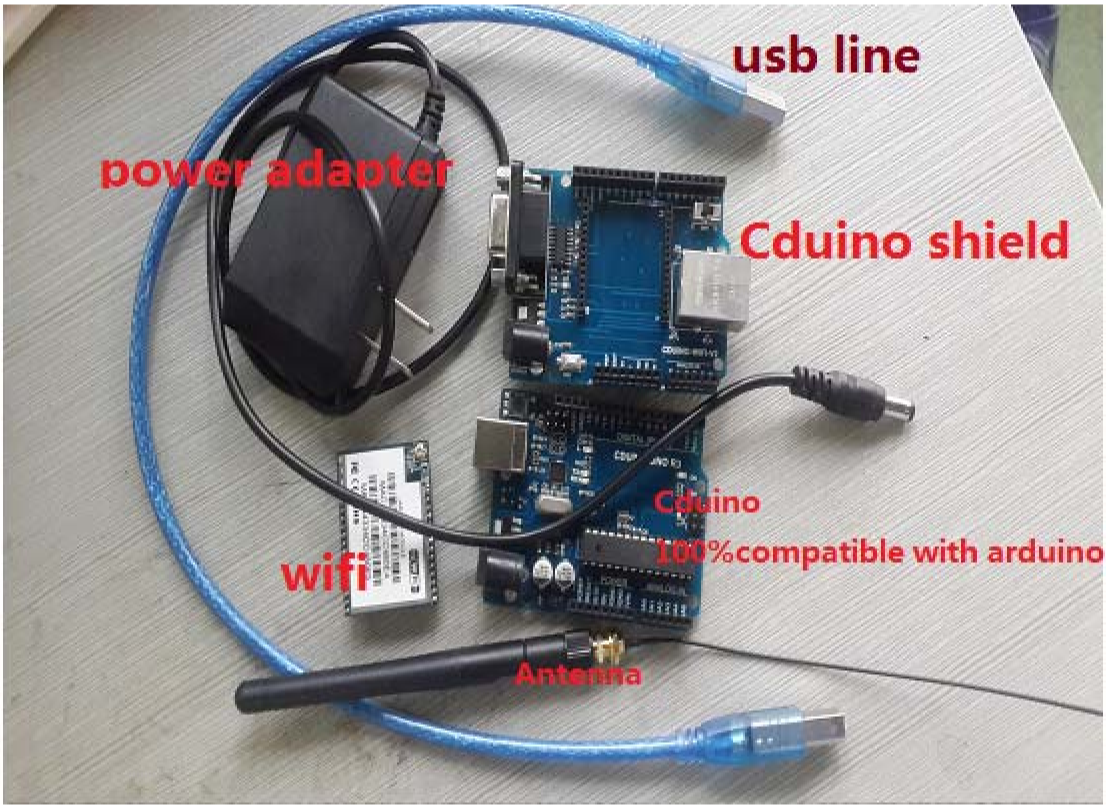
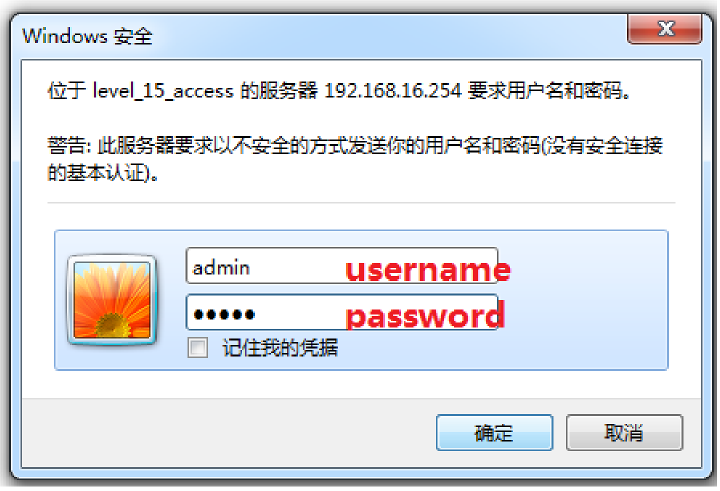


Figure 2-1 Hardware

2.3 WiFi module parameters setting

For readers’ convenience, we will repeat the wifi module parameters setting by the following steps:

1) Connect wifi module: find the signal transmitted by the wifi module in your PC, and double it to connect this wifi network;
2) Login into the wifi local server: input the URL: http://192.168.16.254/, and input the default name (admin) and password (admin).



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)。

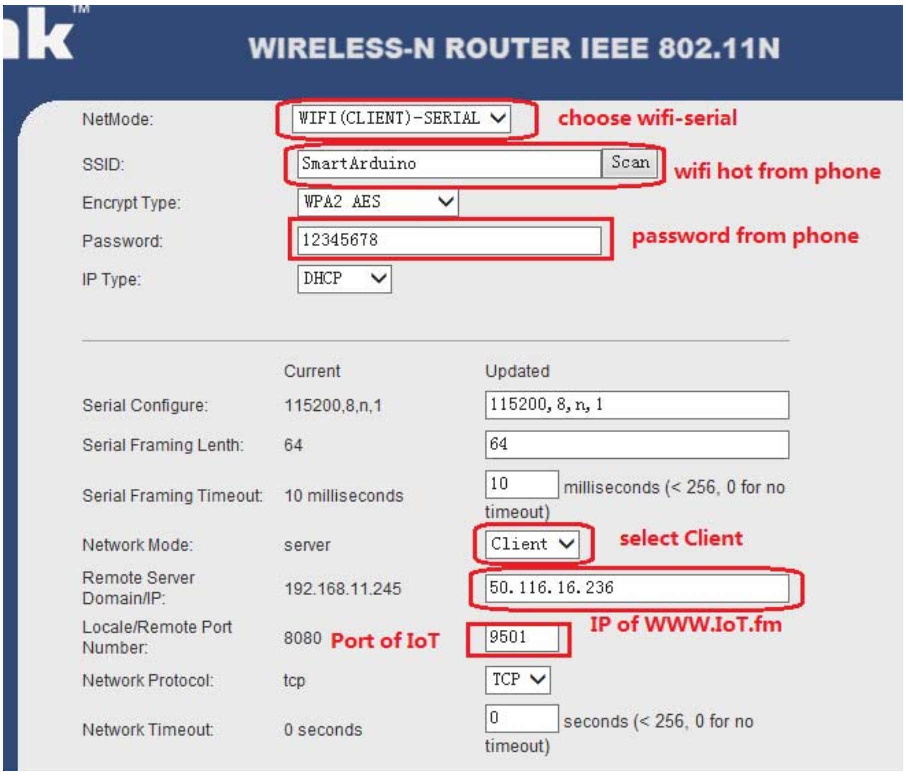
admin

●●●●●

☐ 记住我的凭据

Figure 2-2 login into wifi local server

3) Parameters setting: You can set the corresponding parameters shown in Fig. 2-3. Note that, in our such experiment, we use phone as a wifi router (i.e., hot spot). Its name and password is SmartArduino and 12345678, respectively. The remote server is our IoT system (http://www.iot.fm/ext/examples/desktop/). Its IP is 50.116.16.236, and the TCP port is 9501. Certainly, you will change the parameters if you choose other communication ways and the remote server.



WIRELESS-N ROUTER IEEE 802.11N

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	9501
Network Protocol:	tcp	TCP
Network Timeout:	0 seconds	0 seconds (< 256, 0 for no timeout)

choose wifi-serial

wifi hot from phone

password from phone

select Client

IP of WWW.IoT.fm

Port of IoT

Figure 2-3 wifi module parameters setting

Warm prompt: to make sure the parameters setting right, you had better reset the wifi module to recover the default factory setting, as shown in Fig. 2-4.

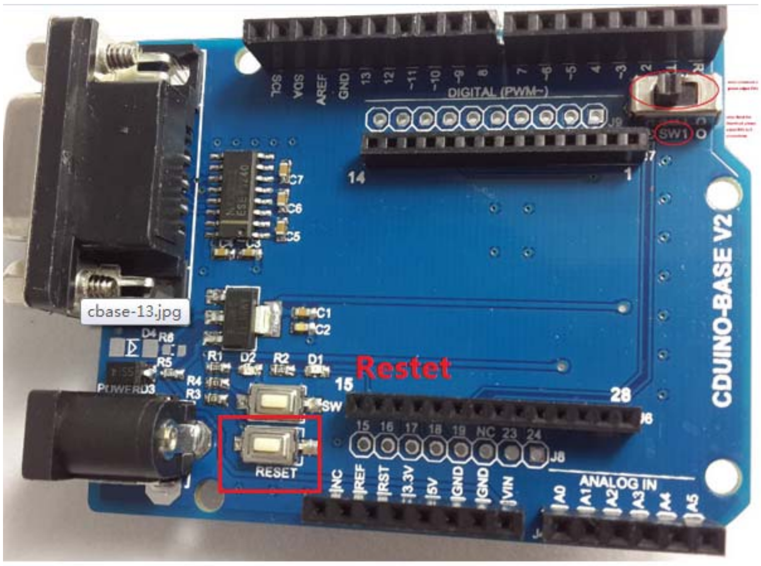
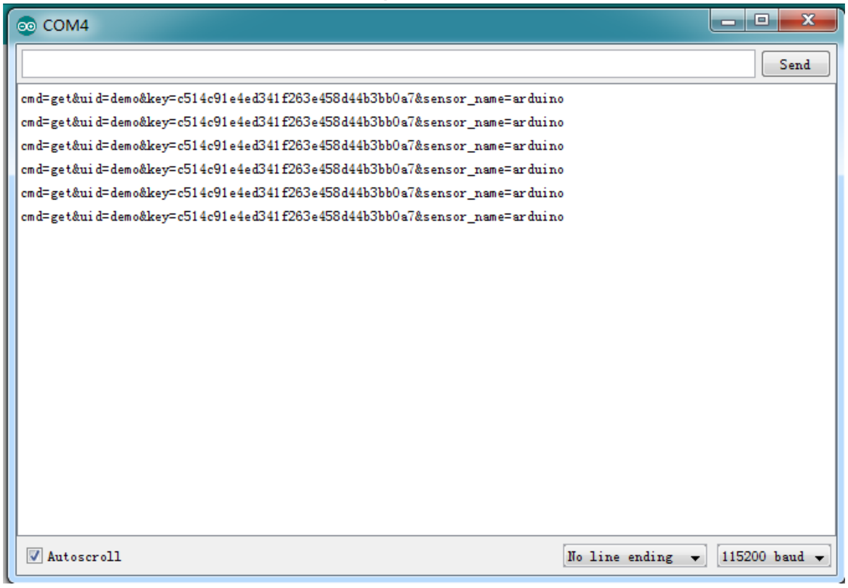


Figure 2-4: reset wifi module

2.4 Experiment

After the above setting, you can download the Arduino code to the Cduino development board. The Arduino code can be download from the github ().

Then, open the serial monitor, we have the following result, as seen in Fig. 2-5. It is shown that, the sensor arduino is waiting for the command from the server.

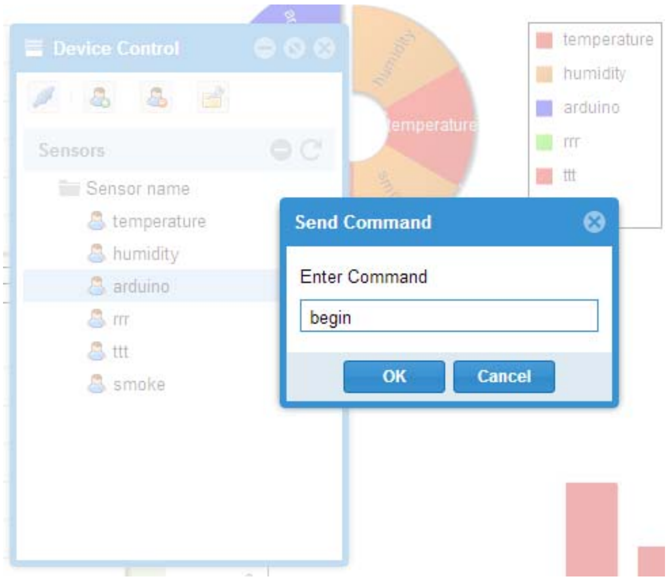


```
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
```

☑ Autoscroll
No line ending
115200 baud

Figure 2-5: experimental result by serial monitor

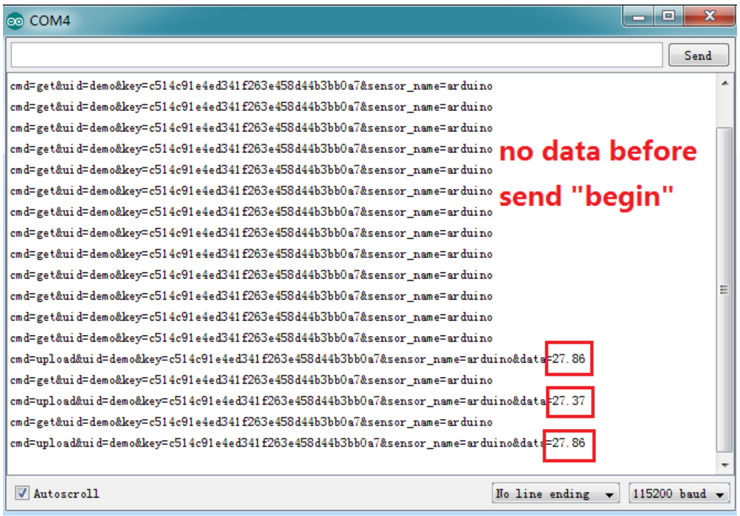
So, if we send a command "begin" to the sensor arduino, seen in Fig. 2-6.



Enter Command
begin

Figure 2-6: send a command "begin" from the remote server.

and Arduino board also receives the command, then we will see that, the sensed data has been tested by serial monior, shown in Fig. 2-7.



```
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.86
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.37
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.86
```

Figure 2-7: the sensed data by using serial monitor.

At the same time, we can also see the sensed data at the remote server. Open the “Data Display”, and select the sensor “arduino” at the pie graph. We see the sensed temperature data by LM35, which is shown by the pulse in Fig. 2-8.

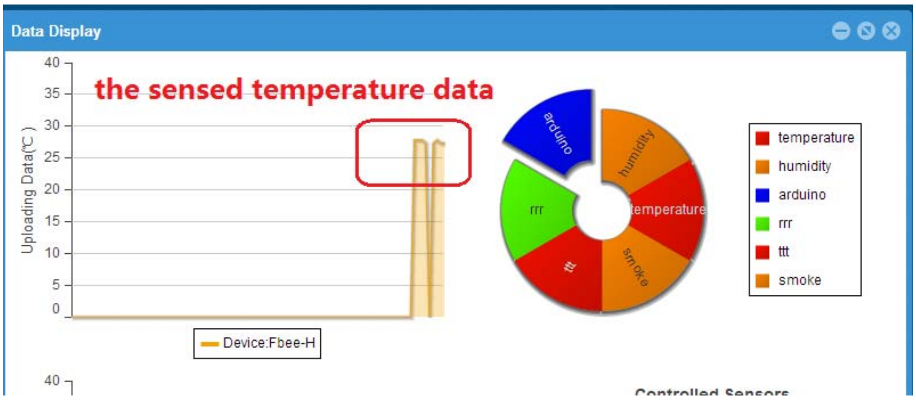
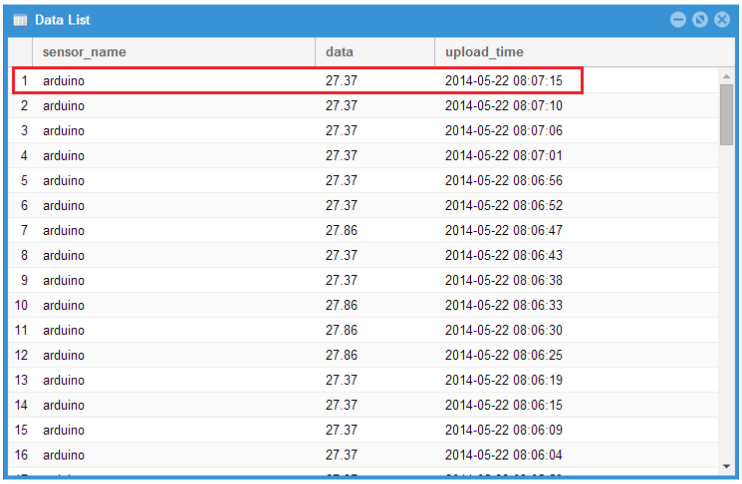


Figure 2-8: the sensed temperature data from Data Display.

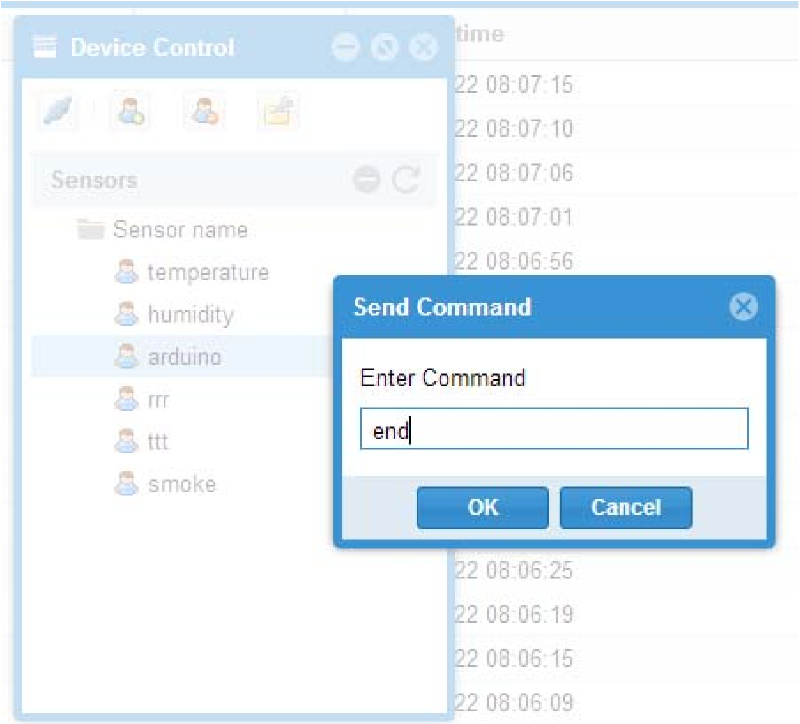
In addition, we can use the “Data List” to see the concrete data value, as shown in Fig. 2-9. For example, at the time: 2014-05-22 08:07:15, the sensor “arduino” receives the temperature data “27.37”.



	sensor_name	data	upload_time
1	arduino	27.37	2014-05-22 08:07:15
2	arduino	27.37	2014-05-22 08:07:10
3	arduino	27.37	2014-05-22 08:07:06
4	arduino	27.37	2014-05-22 08:07:01
5	arduino	27.37	2014-05-22 08:06:56
6	arduino	27.37	2014-05-22 08:06:52
7	arduino	27.86	2014-05-22 08:06:47
8	arduino	27.37	2014-05-22 08:06:43
9	arduino	27.37	2014-05-22 08:06:38
10	arduino	27.86	2014-05-22 08:06:33
11	arduino	27.86	2014-05-22 08:06:30
12	arduino	27.86	2014-05-22 08:06:25
13	arduino	27.37	2014-05-22 08:06:19
14	arduino	27.37	2014-05-22 08:06:15
15	arduino	27.37	2014-05-22 08:06:09
16	arduino	27.37	2014-05-22 08:06:04

Figure 2-9: the concrete temperature data from “Data List”

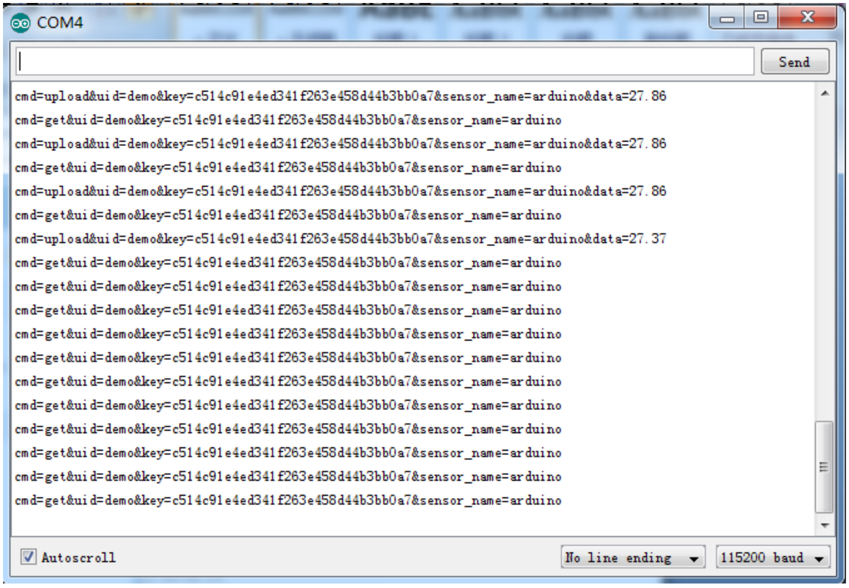
If we send a command “end”, the remote server doesn’t receive the sensed temperature data, which can be seen Figs. 2-10, 11.



Enter Command

end

Figure 2-10: send a command “end” at the remote server.



```
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.86
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.86
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.86
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=upload&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino&data=27.37
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
cmd=get&uid=demo&key=c514c91e4ed341f263e458d44b3bb0a7&sensor_name=arduino
```

☑ Autoscroll

No line ending

115200 baud

Figure 2-11: the sensed data disappears after send an “end” command