

Chapter Three

A Brief But Effective HTTP Communication between Arduino and Remote Server

3.1 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).

3.2 Hardware

The required materials in this experiment are shown in Table 3-1.

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 Figure 3-1. 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 \(http://www.smartarduino.com/arduino-development-board-cduino-base-power-adapter-omni-p94247.html\)](http://www.smartarduino.com/arduino-development-board-cduino-base-power-adapter-omni-p94247.html).

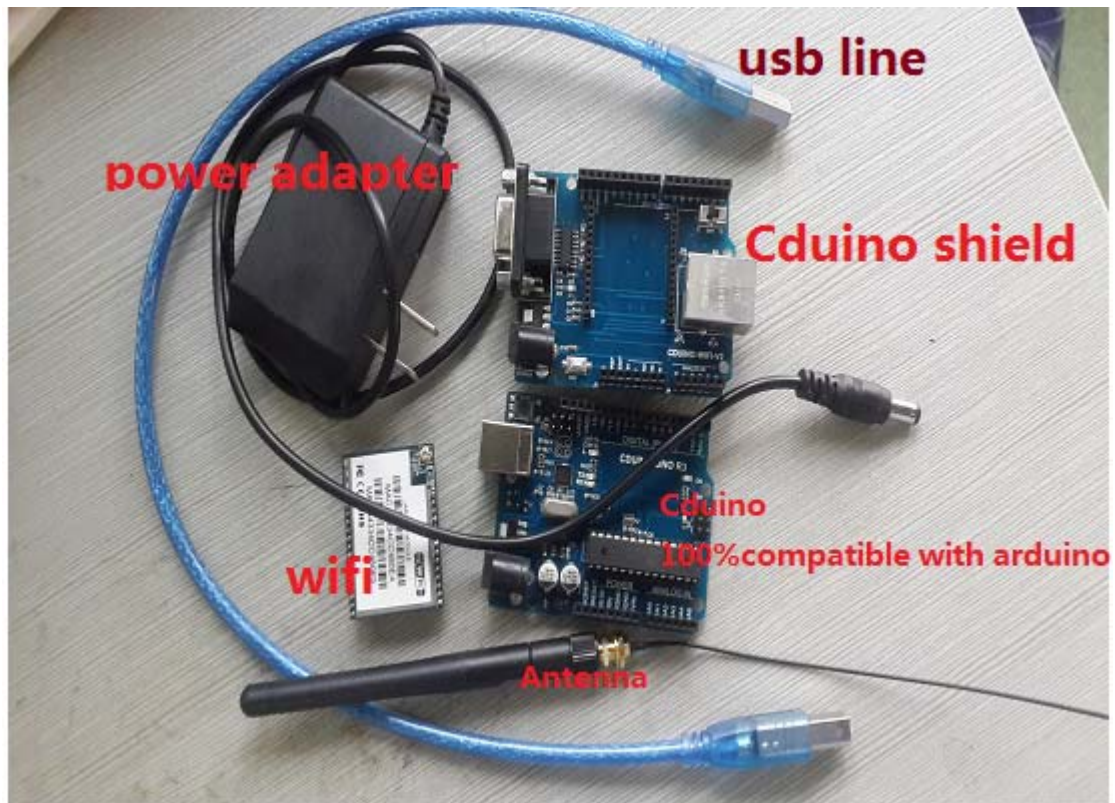


Figure 3-1 [Hardware](#)

Before using WiFi module, you should set the corresponding parameters, which can be seen in the blog <http://blog.smartarduino.com/> in Chapter Two.

Note that, the parameters setting of HTTP is different the one of TCP, which can be seen Figure 3-2.

WIRELESS-N ROUTER IEEE 802.11N

NetMode:

SSID:

Encrypt Type:

Password:

IP Type:

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

Figure 3-2: HTTP parameters setting.

From Fig. 3-2, we know that, the protocol port should be 8080, and other parameters are the same as TCP. So, we can refer to the parameters setting of TCP.

3.2 Experiments

The corresponding code can be downloaded from github:
<https://github.com/SmartArduino/IoT-System-on-OpenSource/tree/master/ArduinoHTTP>

About the experiments, it is the same as TCP. In other words, if we don't send the command "begin", the remote server cannot receive the sensed temperature data, as shown in Fig. 3-3.

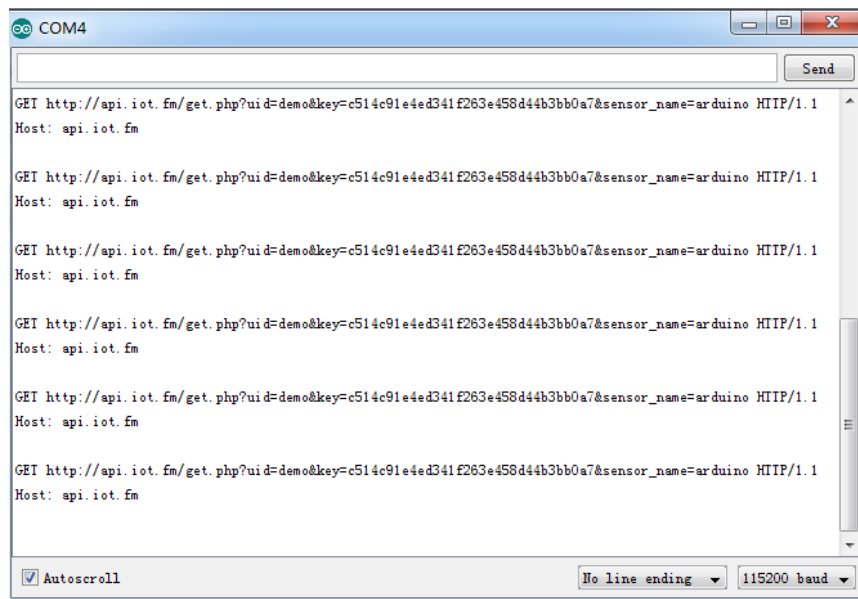


Figure 3-3: results without sending "begin"

But after send the "begin" (Fig. 3-4), the remote server can receive the sensed temperature data, which can be seen Fig. 3-5. The sensed data expressed by graph in Fig. 3-6 is also obtained, and also exhibited by table list in Fig. 3-7.

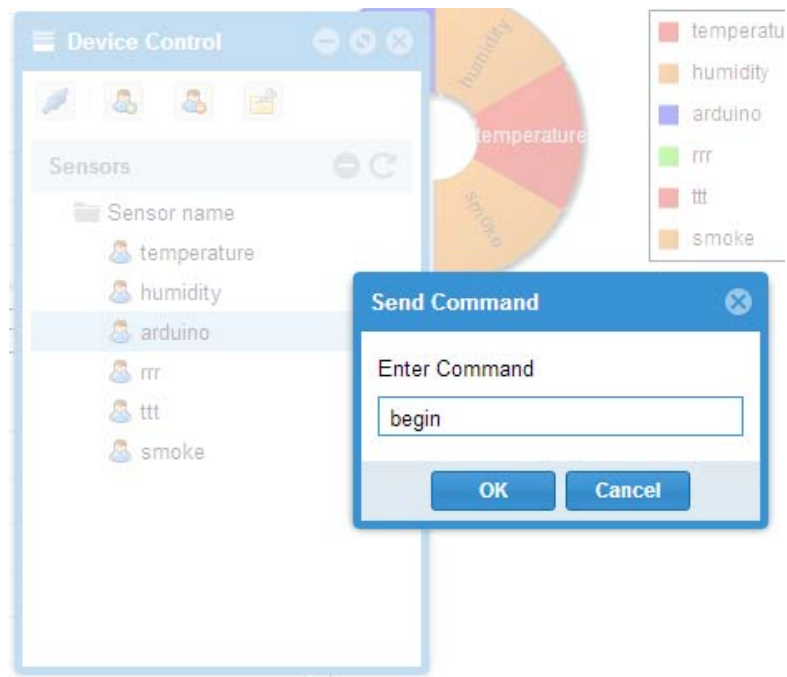


Figure 3-4: sending a command “begin”

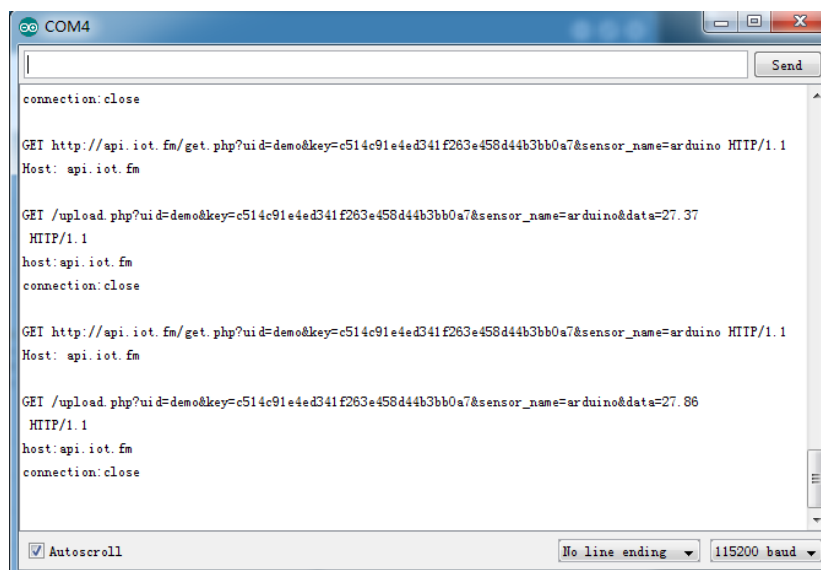


Figure 3-5: results with sending “begin”

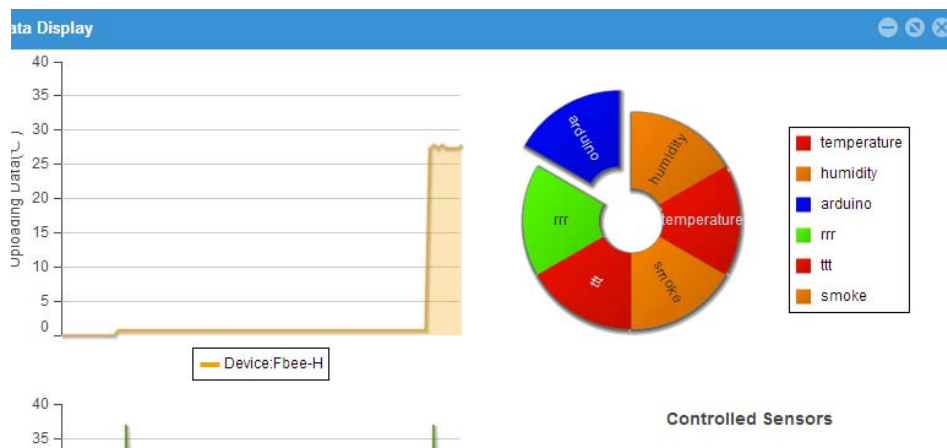


Figure 3-6: the sensed temperature data by graph

	sensor_name	data	upload_time
1	arduino	27.86	2014-05-22 07:54:52
2	arduino	27.86	2014-05-22 07:54:49
3	arduino	27.86	2014-05-22 07:54:46
4	arduino	27.86	2014-05-22 07:54:43
5	arduino	27.86	2014-05-22 07:54:39
6	arduino	27.86	2014-05-22 07:54:36
7	arduino	27.37	2014-05-22 07:54:33
8	arduino	27.86	2014-05-22 07:54:30
9	arduino	27.37	2014-05-22 07:54:26
10	arduino	27.86	2014-05-22 07:54:23
11	arduino	27.86	2014-05-22 07:54:20
12	arduino	27.86	2014-05-22 07:54:17
13	arduino	27.37	2014-05-22 07:54:14
14	arduino	27.37	2014-05-22 07:54:10
15	arduino	27.37	2014-05-22 07:54:07
16	arduino	27.86	2014-05-22 07:54:04

Figure 3-7: the sensed data exhibited by Table list

And, after sending a command “end” in Fig. 3-8, the remote server would not receive the sensed data, as shown in Fig. 3-9.

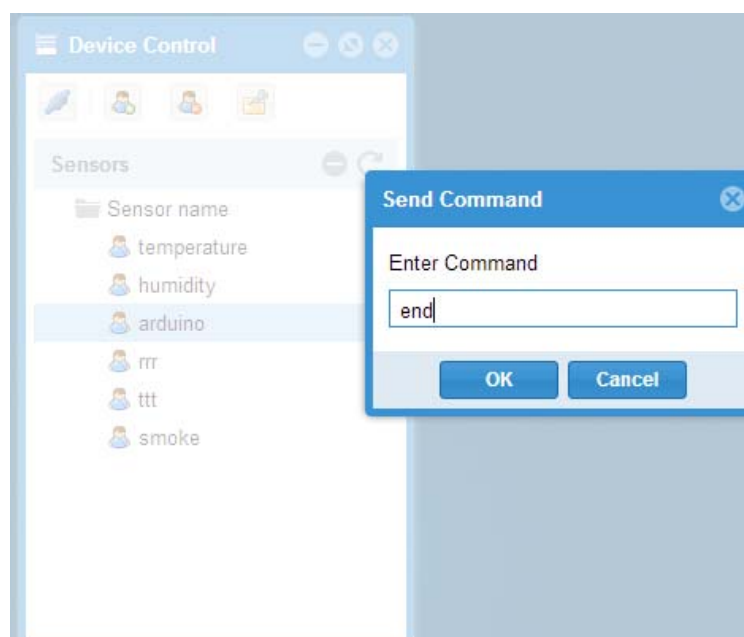


Figure 3-8: sending a command “end”

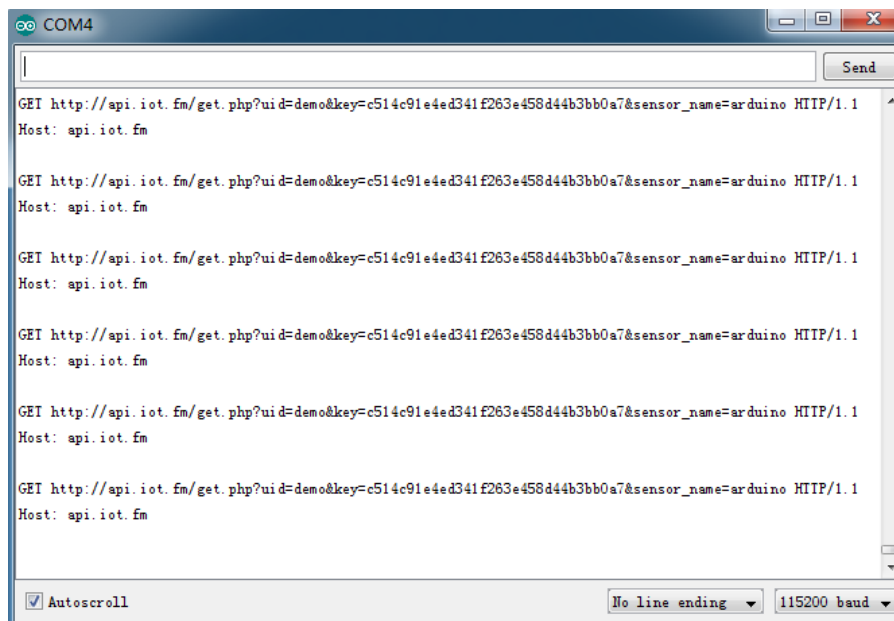


Figure 3-9: results after sending “end”