



深圳四博智联科技有限公司

---

## Arduino 与 doit 物联网系统



二〇一五年十二月

#### 4.1 问题描述：如何在云端（服务器）将 Arduino 采集的数据处理和展现

在前面第 3 讲中，咱们已经将 Arduino 控制板采集到的数据成功上传到远端的云服务器上。但是，这些上传到服务器的数据还需要经过处理和加工，这样便能以更加美观的方式展现给自己或别人查看。在这部分内容中，咱们将介绍一个云服务器端的管理系统---无线传感网管理信息系统（WSN MIS），通过这个无线传感网系统，我们可以很方便地管理、展现和控制 Arduino 采集的数据。

#### 4.2 硬件与软件

本次实验所需要的器材和第 3 讲一样，如表 4-1 所示。

表 4-1：所需材料

序号	名称	数量	作用	备注
1	Arduino 软件平台	1	提供平台	
2	服务器端	1	收集数据	
3	Wifi 模块	1 块	无线通信	
4	Arduino 扩展板	1 块	连接 wifi	各种版本均可
5	USB 转串口 RS232	1 条	转换连接	
6	5V1A 电压适配器	1 个	提供电压	
7	天线	1 个	发送 wifi 信号	可选
8	HTC 手机	1 部	提供 wifi 热点	
9	客服端	1	查看数据	

本次实验所需的硬件材料如后图 4-1 所示。所有开发板均可在深圳四博智联淘宝商城购的：

<https://szdoit.taobao.com/>

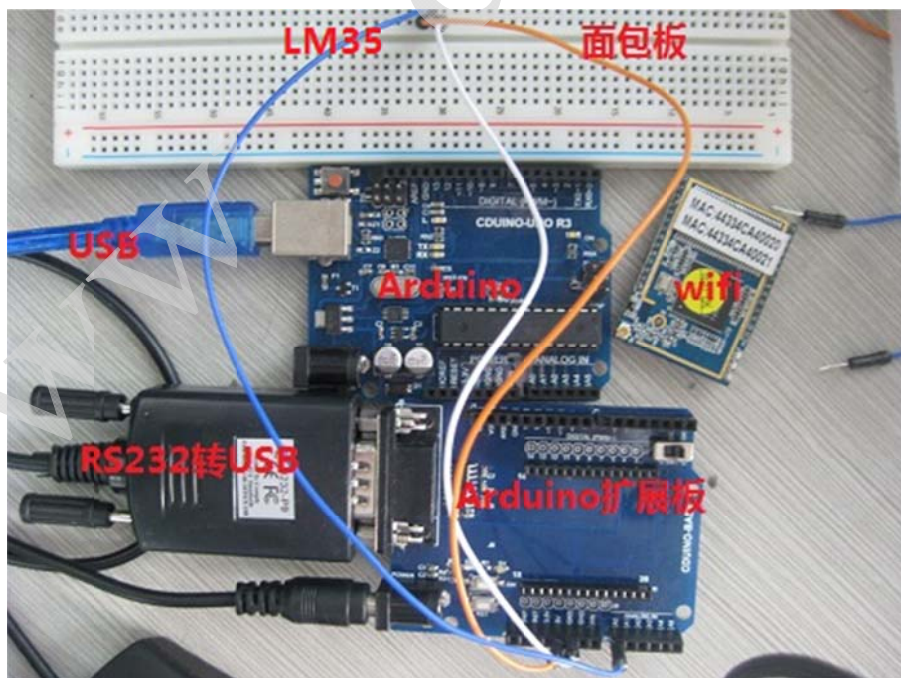


图 4-1 硬件实物

#### 4.3 无线传感网系统介绍

咱们将以第 3 讲的实例为例来讲述无线传感网系统的使用。通过该无线传感网系统，可以将 wifi 上传上来的温度数据很好地在系统上实时动态地显示。

### (1) 登陆与注册

打开浏览器，输入处在云端的无线传感网系统：<http://demol.cduino.com/>，进入如图 4-2 所示的登入界面。



图 4-2 无线传感网登入界面

其中用户名和密码为 10008，这是一个测试账户。为了便于管理日后的数据管理，咱们新建一个账户，如图 4-3 所示。



图 4-3 用户注册界面

在本次实验中，咱们注册的用户名和密码均为 ycf。然后在登入界面输入刚才注册的用户名和密码 ycf，登入无线管理系统，如图 4-4 所示。

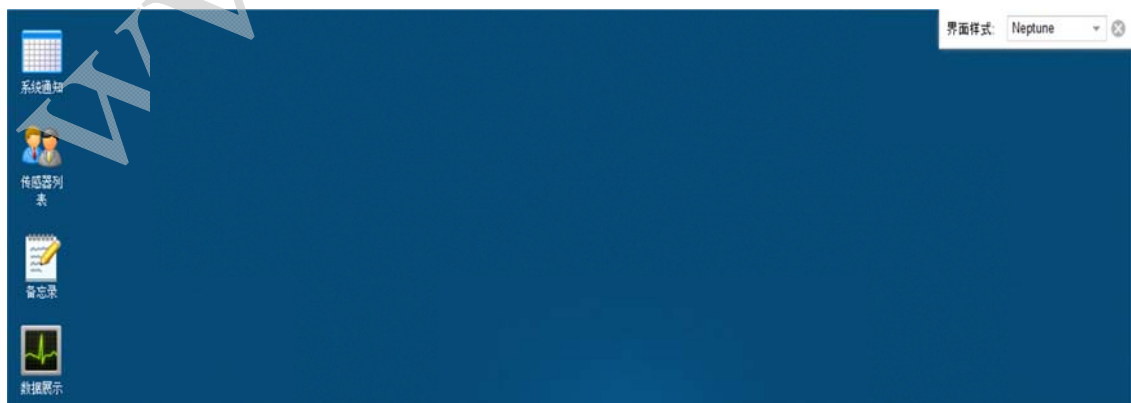


图 4-4 无线管理系统

双击系统中的“传感器列表”模块，咱们可以添加、删除一个传感器，并可以控制传感器，如图 4-5 所示。这个模块还可以在界面上任意拖动。



图 4-5 双击“传感器列表”

需要注意的是，新增传感器节点的名称需要和 Arduino 代码中的名称一致，例如，在本次实验中，我们新增的温度传感器节点的名称是 arduino，这个名称应和我们在 arduino 程序中定义的传感器节点名称一致，比较图 4-5 和 4-6 中红色标记所示。

```
void upload_sensor(float vol)
{
    // send the HTTP PUT request: 核心代码
    char buf[200];
    memset(buf, 0, 200);
    int ret;
    ret=sprintf(buf, "GET /upload.php?uid=yctf@ps=yctf@sensor_name=arduino&data=");
    Serial.print(buf);
    Serial.print(vol);
    Serial.println(" HTTP/1.1");
    Serial.println("Host: api.eduino.com");
    Serial.println("Connection: close");
    Serial.println();
}
```

系统用户名

传感节点名称

图 4-6 Arduino 程序和无线传感网系统的对应关系

这样，咱们在云端的无线传感网系统就可以接受，删除和控制 Arduino 控制的传感节点了，具体 Arduino 代码可参考第 3 讲中的代码。

## (2) 客户端的上传与控制协议





## 上传数据协议

- 1) 上传数据的传输协议为 HTTP, 使用 GET 方法;
  - 2) 请求地址: <http://api.cduino.com/upload.php>, 参数:  
uid=yfc&ps=yfc&sensor\_name=arduino; 有多个传感节点的数据用 “&” 符号联接,  
uid=yfc&ps=yfc&sensor\_name=arduino&sensor\_name=pluse&data=120。
  - 3) 应答: res=1&desc=ok, res=0&desc=error。
- 例如: [http://api.cduino.com/upload.php?uid=yfc&ps=yfc&sensor\\_name=arduino](http://api.cduino.com/upload.php?uid=yfc&ps=yfc&sensor_name=arduino), 此时的返回值: res=1&desc=ok。

## 控制协议

- 1) 客户端向设备发送命令:
  - 上传数据的传输协议为 HTTP, 使用方法为 GET
  - 请求地址: <http://api.cduino.com/httpsqs/control.php>
  - 参数: sensor\_name=传感器名称&cmd=命令
  - 应答: 1
  - 例如: 客户端发送 [http://api.cduino.com/httpsqs/control.php?sensor\\_name=arduino&cmd=begin](http://api.cduino.com/httpsqs/control.php?sensor_name=arduino&cmd=begin) 收到应答 1
- 2) 设备接收命令:
  - 获取命令的网络协议为 HTTP, 使用方法为 GET
  - 请求地址: <http://api.cduino.com/httpsqs/get.php>
  - 参数: uid=用户 id&sensor\_name=传感器名称
  - 应答: 客户端发送的命令
  - 例如: 客户端发送命令 begin, 发送请求 [http://api.cduino.com/httpsqs/get.php?uid=10008&sensor\\_name=temperature](http://api.cduino.com/httpsqs/get.php?uid=10008&sensor_name=temperature), 收到应答 begin
- (3) 控制传感器的数据传输  
首先选中无线传感网系统的传感器列表栏, 然后选中需要上传数据的传感器名称, 再单击传感器列表中的图标, 输入指定的发送指令, 点击 “OK” 按钮发送指令。传感器节点数据传输客户端在接收到指令后, 将远程控制客户端发送来的数据上传至数据库。同时也可以通过在传感器列表中发送停止指令来控制数据停止上传至数据库。

### 4.4 无线传感器系统的数据上传与控制

假设无线传感器系统的用户名和密码均为 ycf, 新增的传感器节点名称为 arduino,

#### (1) 数据上传

将下列代码下载至 Arduino 控制板上, 则可实现由温度传感器 LM35 采集的温度数据上传到无线传感系统中。

```
//Arduino 将温度数据上传至无线传感器系统的完整代码
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: 核心代码
    char buf[200];
    memset(buf,0,200);
    int ret;
    ret=sprintf(buf,"GET /upload.php?uid=yfc&ps=yfc&sensor_name=arduino&data=");
    //输入的用户名和密码(均为 ycf)及传感器名称(arduino)
    Serial.print(buf);
    Serial.print(vol);
    Serial.println(" HTTP/1.1");
    Serial.println("Host: api.cduino.com");
    Serial.println("Connection: close");
    Serial.println();
}
```

在云端的服务器中，点击如图 4-7 所示右上饼图中的传感器节点 **arduino**，即可在左上①中实时显示传感器节点 **arduino** 采集的数据；左下图②为传感节点 **arduino** 的历史数据；右下图为传感节点 **arduino** 数据的分布图。

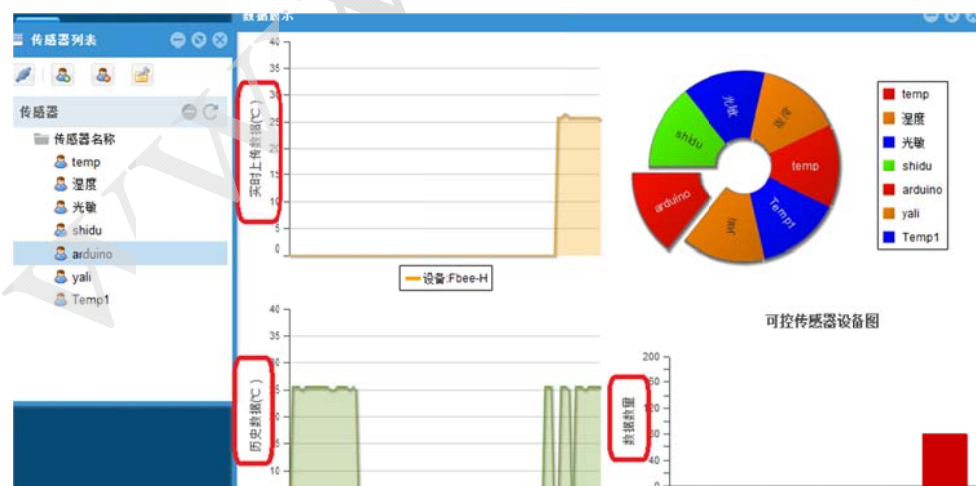



图 4-7 传感节点“arduino”的数据显示图

## (2) 节点控制

将下列代码下载至 Arduino 主板上，当我们选择传感节点“arduino”，点击“向选中的传感器发送命令”图标 ，即可控制 Arduino 主板上传感节点。

```
//#include "parsehttp_ino.h";
```



```
#include "Scheduler.h";

boolean com=false;

void setup()
{
  Serial.begin(115200);
  Sch.init();
  Sch.addTask(upload_data,0,900,1);
  Sch.addTask(check_end,10,1000,1);
  Sch.start();
}

void loop()
{
  String command="";
  if(!com){
    command=get_command();
    if(command=="begin"){
      com=true;
      //command="";
    }
    else{
      delay(4000);
    }
  }
  else{
    Sch.dispatchTasks();
  }
}

String get_http_header()
{
  String header="";
  boolean get=false;
  char res;
  //Serial.print("header:");
  int limit=5000;
  while(1){
    if(Serial.available() > 0){
      res=char(Serial.read());
      header+=res;
      if(header.endsWith("\r\n\r\n") || limit<=0){
```



```
        break;
    }

    limit--;

}

}

return header;
}

int get_content_length(String header)
{
    int length=0;
    //boolean avail=false;
    String tmp;
    String source=header;
    int pos=source.indexOf('\r\n');
    //int pos_que;
    //int i=0;
    while(pos!=-1){
        tmp=source.substring(0,pos);
        source=source.substring(pos+1);
        pos=source.indexOf('\r\n');
        if(tmp.startsWith("Content-Length:")){
            int pos=tmp.indexOf(':');
            String len=tmp.substring(pos+1);
            len.trim();
            length=len.toInt();

        }

        //i++;
    }
    return length;
}

String get_http_body(int length)
{
    String body="";
    char res;
    int i=0;
    while(1){
        if(Serial.available() > 0){
            res=char(Serial.read());
            body+=res;
            i++;
        }
        if(i==length)
```





```
        break;
    }
    return body;
}

String get_command()
{
    // send the HTTP PUT request: 核心代码
    char buf[200];
    memset(buf,0,200);
    int ret;
    ret=sprintf(buf,"GET
http://api.cduino.com/httpsqs/get.php?uid=ycf&sensor_name=arduino");
    //Serial.clear();
    Serial.flush();
    Serial.print(buf);
    Serial.println(" HTTP/1.1");
    Serial.println("Host: api.cduino.com");
    Serial.println("Connection: close");
    Serial.println();
    delay(160);
    Serial.flush();
    String header=get_http_header();
    int length=get_content_length(header);
    String body=get_http_body(length);
    return body;
    //Serial.println(body);
}

void upload_data()
{
    //Serial.println("upload");

    int n=analogRead(A0);
    float vol=n*(5.0/1023*100);
    char buf[200];
    memset(buf,0,200);
    int ret;
    ret=sprintf(buf,"GET /upload.php?uid=ycf&ps=ycf&sensor_name=arduino&data=");
    Serial.print(buf);
    Serial.print(vol);
    Serial.println(" HTTP/1.1");
    Serial.println("Host: api.cduino.com");
    Serial.println("Connection: close");
    Serial.println();
}
```

```
delay(160);
Serial.flush();
String header=get_http_header();
int length=get_content_length(header);
String body=get_http_body(length);
}

void check_end()
{
    String command=get_command();
    if(command=="end"){
        com=false;
        //command="";
    }
}
```

如图 4-8，在下面代码中，当咱们发送“begin”命令时，将告诉传感器节点可以上云端服务器发送数据。其结果如图 4-9 所示。当向节点“arduino”发送“end”命令时，Arduino 主板上的传感器节点“arduino”将不会向云端服务器发送数据，如图 4-10 所示。

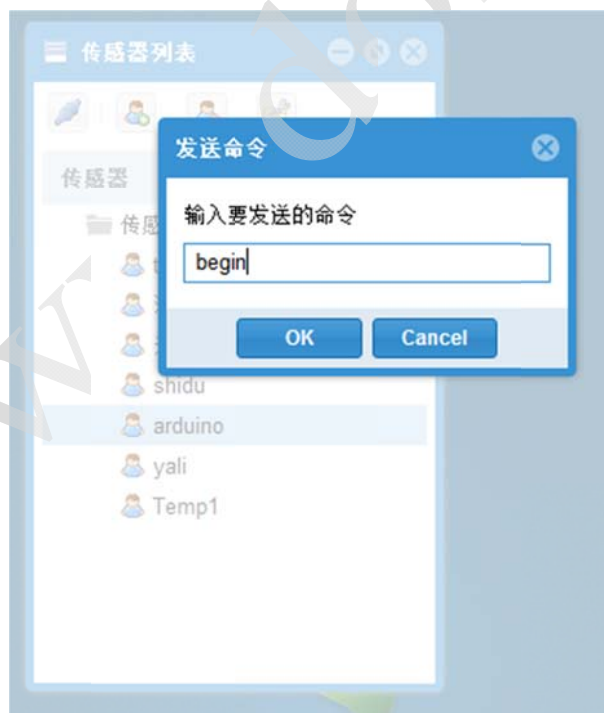


图 4-8 向传感节点“arduino”发送“begin”命令

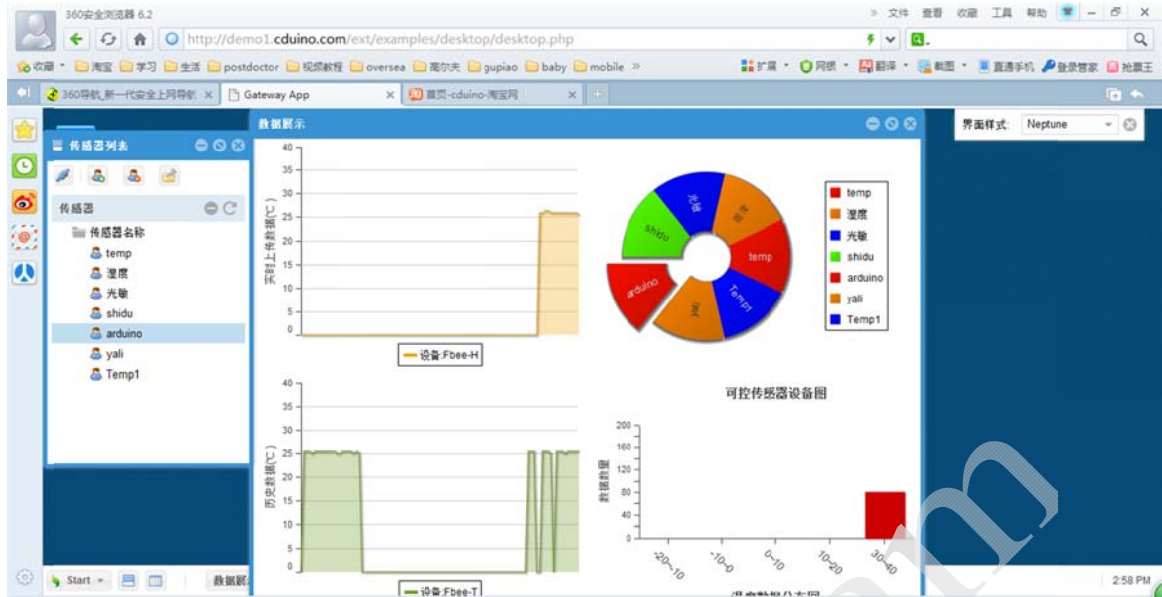


图 4-9 云端服务器向节点“arduino”发送“begin”命令时效果

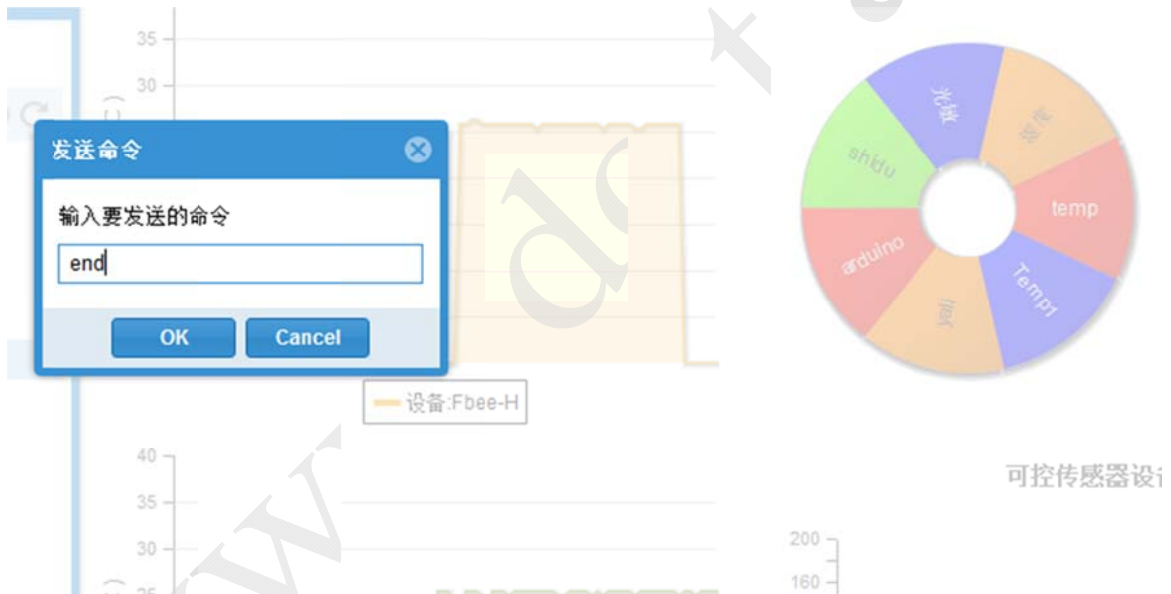


图 4-10 云端服务器向传感器节点“arduino”发送“end”命令效果

