

# 智能云仓库

**摘要：**基于开源硬件 `arduino` 和开源云服务平台 `yeelink`，设计可以远程监测、控制的智能云仓库。实现远程监测仓库环境温度、湿度等变化；火灾报警、远程控制灯光、电器等功能。

**关键词：**`arduino` `yeelink` 云仓库 远程检测 远程控制

## 引言

随着社会的不断进步，物联网技术得到了迅猛的发展，现已广泛运用在了各个领域，如智能家居、地质环境监测、远程监控等。仓库是人们日常生活中是必不可少的，特别是对于公司、生产单位等，同时仓库环境的实时监测、安全防范对是相当重要的。现阶段仓库的监控管理主要是靠人力，然而人的精力是有限的，往往不能有效的对仓库进行高效监控管理。全国每年都有许多仓库发生失火、失窃等不安全事故，造成人员、财产的重大损失。本文针对物联网技术在仓库中的应用展开分析，基于开放的硬件和软件平台，完成了智能云仓库的设计和制作。可用于实时监测仓库环境，如温度、湿度、二氧化碳气体等变化；也能远程控制仓库灯光、电器等，这对于推进智能云仓库建设有着重大意义。

## 1. 方案设计

### 1.1 开发环境

选取合适的开发工具将有利于开发工作的顺利开展，开源平台有着强大的开发优势，在开源平台上硬件和软件的更新频率较高，许多开发者都会开源自己的硬件、软件设计，并且出现的 `bug` 也会及时的得到开发者们的纠正。所以在此选取开源的开发平台，将很有利于开发工作的进展。`arduino` 是开源的硬件设计平台，有着丰富的开发例程，开发者在大多数情况下可以实现代码的轻松移植；`yeelink` 是开源的云服务平台，开发者可以将自己的传感器数据接入到 `yeelink` 中，其提供的友好的显示界面，并且可以实现 `Web` 控制功能、`Email` 提醒功能等，可实时实现在线检测、控制。具体开发环境见表 1.1。

表 1.1 智能云仓库开发环境

开发环境			说明
硬件	传感器采集板	arduino uno	① arduino 是一个基于开放原始码的硬件设计平台； ② 使用 C 语言开发，简单快速； ③ 开发板接口丰富； ④ 价格便宜，适合低成本开发。
	以太网接入板	W5100	① W5100 内部集成 10/100Mbps 以太网控制器；全硬件的、且经过多年市场验证的 TCP/IP 协议栈、以太网介质传输层（MAC）和物理层（PHY）； ② 价格便宜，适合低成本开发。
	路由器	TP-LINK 150M 迷你无线路由器	① 有两个有线接入口，分别是 LAN/WAN、LAN； ② 带无线功能。
	传感器	LM35 温度传感器 DHT11 湿度传感器 HC-SR 人体红外传感器 MQ2 烟雾传感器 雨滴传感器 火焰传感器等	① 各类传感器采用集成的模块，这样便于开发使用，并且可以提高传感器的稳定性。 ② 传感器多采用模拟采集方式，这样便于实时数据数据采集。
软件	PC 端	Yeelink 平台/自己开发相应平台	① yeelink 是开源的云服务平台，提供友好用户界面； ② yeelink 数据接入简单； ③ yeelink 有设置邮件、微博提醒功能。 ④ 自己开发流程较长，并且难度较大，但是一次不错的学习过程。
	移动端	Yeelink 移动平台/自己开发相应平台	① yeelink 移动端，提供友好用户界面； ② yeelink 有设置邮件、微博提醒功能。 ③ 自己开发流程较长，难度一般，开发效果不一定好，但是一次不错的学习过程。

1.2 方案设计

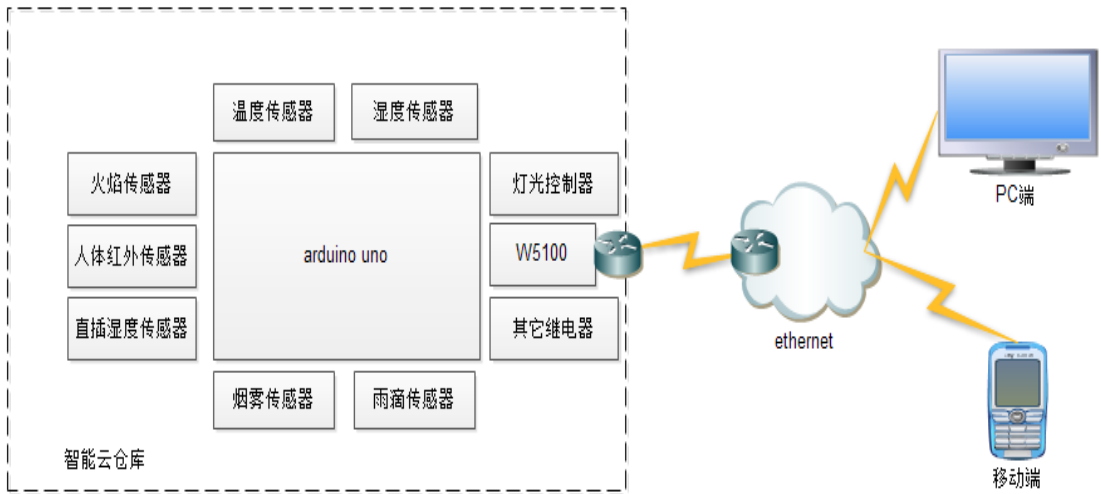


图 1.1 方案设计图

方案设计如上图 1.1 所示，各种传感器通过模拟或数字的方式挂载在 arduino uno 开发板上，采集到的数据经开发板处理后，将数据通过以太网控制板 W5100、无线路由器接入以太网中，这样就可以实现远程通过 yeelink、移动终端实时监测、控制仓库的环境情况。以下图 1.2、1.3 分别给出了 PC 端、移动端数据监控效果图。

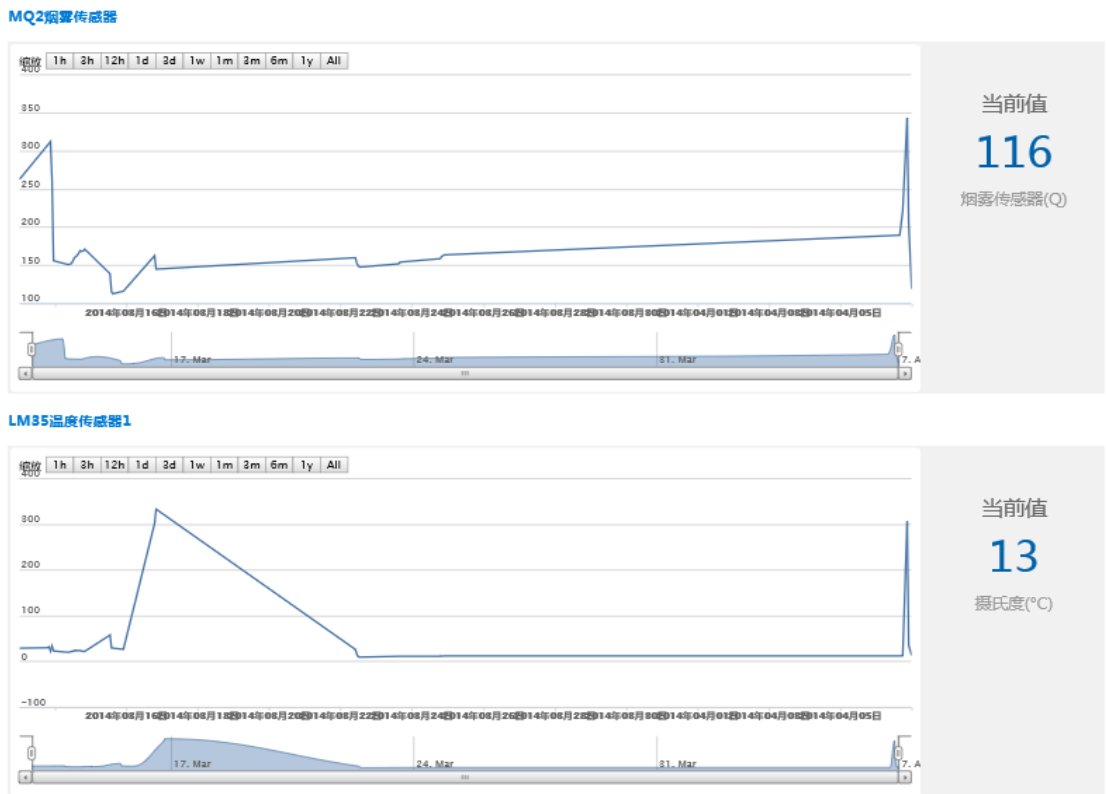


图 1.2 PC 端数据显示效果图

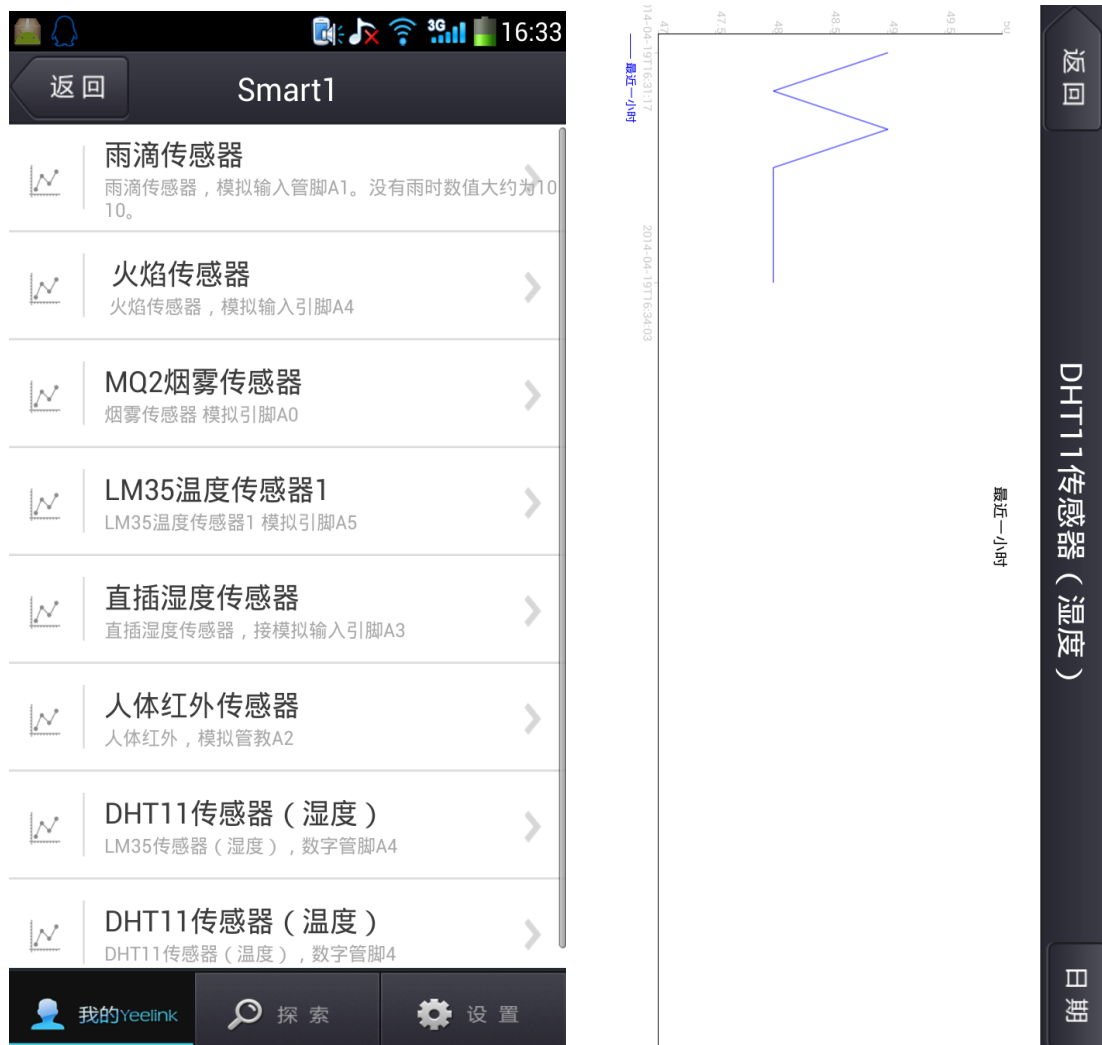


图 1.3 移动端数据显示效果图

## 1.3 方案实现

### （1）温度监控

温度监控用于监控仓库中温度，对于某些用于特殊用途的仓库，如冷冻室等，温度监控就显得尤为重要。温度监控相关说明如下表 2 所示。

表 2 温度监控相关说明

温度监控	实现方式	连接方式
	LM35 温度芯片	与 arduino 模拟 A5 管脚相接
说明	工作电压：5V 调用函数：analogRead (pin)、float lm35_convertor(int analog_num)、therm_5.single_post(messenger, dp5)等	

### （2）空气湿度监控

湿度监控用于监控仓库中的环境湿度，对于需要保持干燥的仓库、湿度控制是很重要的。湿度监控相关说明如下表 3 所示。

表 3 湿度监控相关说明

空气湿度	实现方式	连接方式
监控	DHT11 湿度监测模块	与 arduino 数字 4 管脚相接
说明	工作电压：5V 调用函数：dht.h、therm_6.single_post(messenger, dp11)等	

(3) 人体红外传感器

人体红外传感器通过捕捉人体释放出的红外热能来感知人的存在，通过这一特性，我们可以在仓库的重要位置部署人体红外传感器来监测是否有人到过那里，通过与蜂鸣器相连接，起到了一定的报警、防盗作用。人体红外检测相关说明如下表 4 所示。

表 4 人体红外检测相关说明

人体红外	实现方式	连接方式
监控	HC-SR 人体红外检测模块	与 arduino 模拟 A2 管脚相接
说明	工作电压：5V 调用函数：analogRead (pin)、therm_2.single_post(messenger, dp2)等	

(4) 火焰传感器

火焰传感器可以检测到明火，通过在仓库的关键地方部署适量的火焰传感器，可以起到预防火灾、及时报警的作用。这对于仓库的防火建设有着重大的意义。火焰检测相关说明如下表 5 所示。

表 5 火焰检测相关说明

火焰监控	实现方式	连接方式
	火焰传感器监测模块	与 arduino 模拟 A3 管脚相接
说明	工作电压：5V 调用函数：analogRead (pin)、therm_3.single_post(messenger, dp3)等	

(5) 烟雾传感器

配合火焰传感器可以提高火灾报警的能力，也可用作特殊用途，比如该仓库是用于熏制腊肉的，这时就可以通过部署较多的烟雾传感器来控制烟雾的量。烟雾检测相关说明如下表 6 所示。

表 6 烟雾检测相关说明

烟雾监控	实现方式	连接方式
	MQ2 烟雾传感器监测模块	与 arduino 模拟 A0 管脚相接
说明	工作电压：5V 调用函数：analogRead (pin)、therm.single_post(messenger, dp)等	

#### (6) 雨滴传感器

有的仓库需要经常打开窗户通风，但是在忘记关窗时，有可能会導致下雨时雨点飘进仓库里去，造成损失。这时如果在仓库楼顶搭一个雨滴传感器，就可以自动感应到下雨，然后控制继电器去打开或者关闭仓库的窗户，实现仓库的窗户自动控制。雨滴传感器相关说明如下表 7 所示。

表 7 雨点传感器相关说明

下雨监控	实现方式	连接方式
	雨滴监测模块	与 arduino 模拟 A1 管脚相接
说明	工作电压：5V 调用函数：analogRead (pin)、therm_1.single_post(messenger, dp1)等	

#### (7) 直插湿度传感器

对于在仓库放置比如粮食等这类货物时，不仅仅需要了解空气中的湿度，还必须了解粮食堆里面的湿度，这样才能保证粮食不会变坏，这时就需要直插湿度传感器，相关说明见表 8。

表 8 直插湿度检测模块相关说明

直插湿度检测	实现方式	连接方式
	直插湿度检测模块	与 arduino 模拟 A4 管脚相接
说明	工作电压：5V 调用函数：analogRead (pin)、therm_4.single_post(messenger, dp4)等	

#### (7) 电器设备控制

在设计智能仓库时，考虑到实际情况中可能会远程控制电器设备操作，如控制灯光、风扇等，所以需要设计继电器控制组，来控制仓库中的电器设备。

表 9 电器控制相关说明

电器控制	实现方式	连接方式
	继电器模块	与 arduino 数字 7 等管脚相接
说明	工作电压：5V 调用函数：???????	

1.4 作品制作



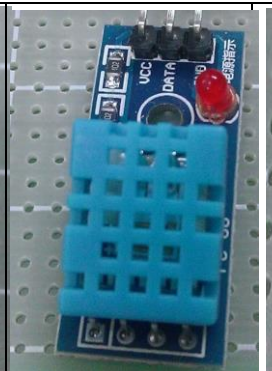


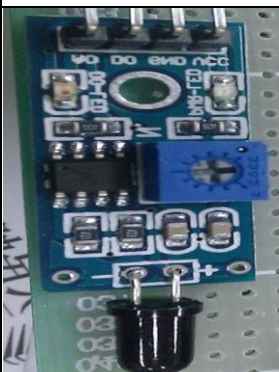
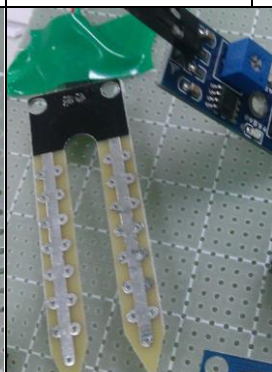
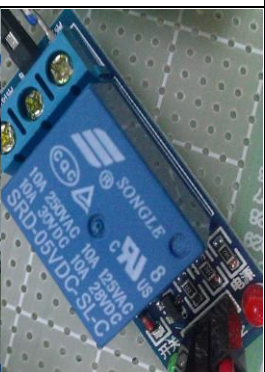
			
人体红外传感器	烟雾传感器	空气湿度传感器	温度传感器
			
雨滴传感器	火焰传感器	直插湿度传感器	继电器

图 1.4 部分传感器实物图

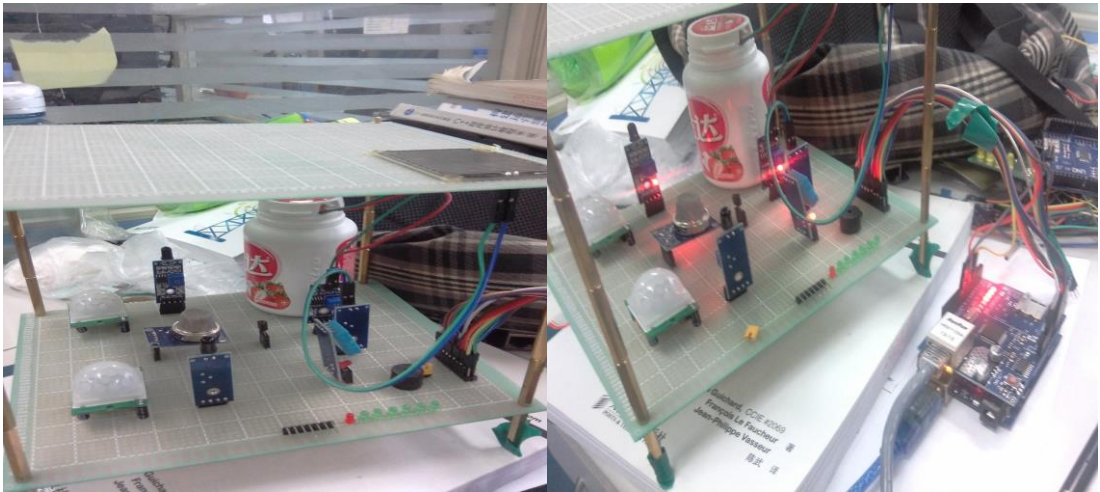




图 1.5 整体实物图

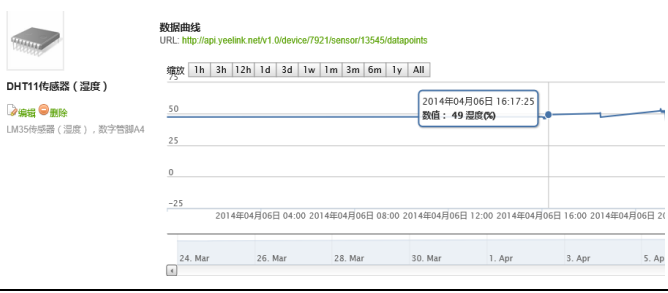
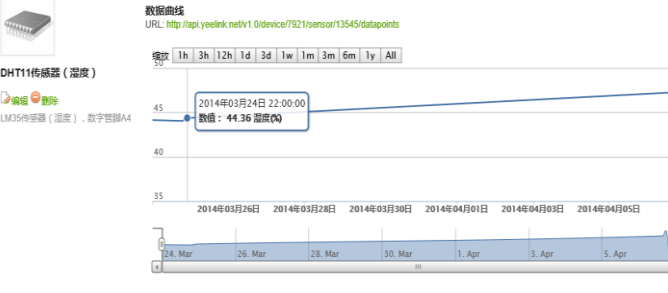
## 2. 效果测试

### 2.1 温度传感器测试

时间	环境实际温度	截图	误差
2014 年 4 月 6 日 22:26:08	12℃		0℃
2014 年 3 月 24 日 19:00:00	11℃		0.05℃

说明：实际温度以周围测温计测得。





### 2.2 湿度传感器测试

时间	环境实际湿度	截图	误差
2014 年 4 月 6 日 16:17:25	46%		49%
2014 年 3 月 24 日 22:00:00	39%		5.36%


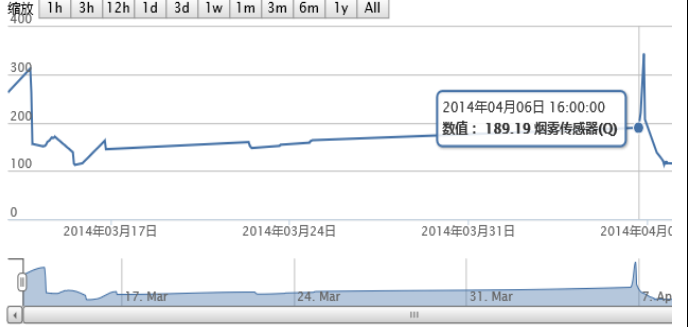
说明：实际湿度以互联网发布的及时数据为准。




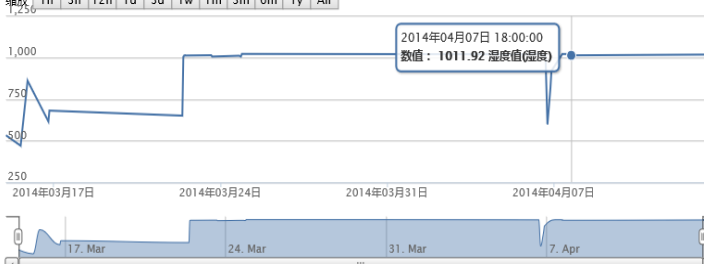


2.3 人体红外传感器测试

时间	截图
2014 年 4 月 6 日 21:16:53	<div><p>人体红外传感器</p><p>编辑 删除</p><p>人体红外，模拟管脚A2</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/13416/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/13416/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年04月06日 21:16:53 数值: 740 人体红外(红外)</p></div>
2014 年 4 月 13 日 20:04:20	<div><p>人体红外传感器</p><p>编辑 删除</p><p>人体红外，模拟管脚A2</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/13416/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/13416/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年04月13日 20:04:20 数值: 740 人体红外(红外)</p></div>
<p>说明：人体红外传感器在没有感应到人存在时，传回的数值为 0；当感应到有人在附近时，传感器传回的值为 740 左右。通过这个数值的变化，即可判断传感器附件是否有人。测试时，人不定时的在传感器周围走到，测得的数据如上所示。</p>	


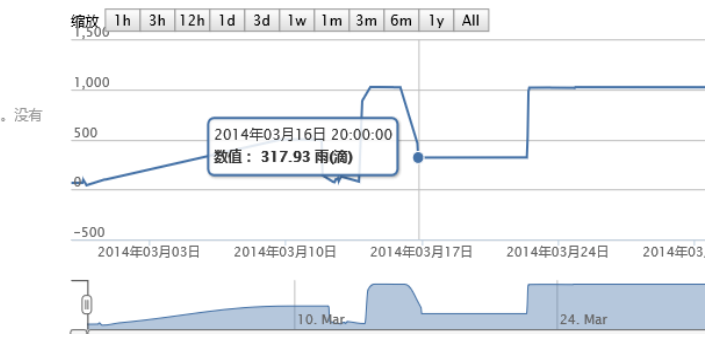
2.4 烟雾传感器测试

时间	截图
2014 年 4 月 6 日 14:00:00	<div><p>MQ2烟雾传感器</p><p>编辑 删除</p><p>烟雾传感器 模拟引脚A0</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/13050/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/13050/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年04月06日 16:00:00 数值: 189.19 烟雾传感器(CO)</p></div>
<p>说明：烟雾传感器在正常情况传回的值为 150 左右（这要根据实际情况而定，不同环境，不同传感器模块都有较大的差别），测试时点燃一张纸，然后将其熄灭用其产生的烟雾去靠近传感器，得到的数据如上图所示。</p>	




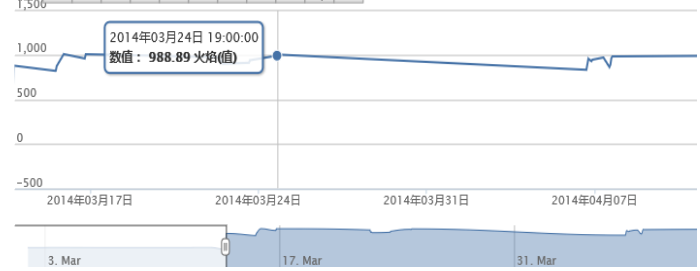
2.5 直插湿度传感器测试

时间	截图
2014 年 4 月 7 日 18:00:00	<div><p>直插湿度传感器</p><p>编辑 删除</p><p>直插湿度传感器，接模拟输入引脚A3</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/13170/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/13170/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年03月17日 2014年03月24日 2014年03月31日 2014年04月07日</p></div>
2014 年 3 月 16 日 20:00:00	<div><p>直插湿度传感器</p><p>编辑 删除</p><p>直插湿度传感器，接模拟输入引脚A3</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/13170/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/13170/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年03月17日 2014年03月24日 2014年03月31日</p></div>
说明：直插湿度传感器完全干燥（暴露在空气中）的情况下传回的数字大约为 1020 左右，不同湿度下传回的值是不同的，例如将其插入到较湿的土壤中，传回的值大约为 680 左右，传回的数值参见上图。	

2.6 雨滴传感器测试

时间	截图
2014 年 3 月 16 日 20:00:00	<div><p>雨滴传感器</p><p>编辑 删除</p><p>雨滴传感器，模拟输入引脚A1。没有雨时数值大约为1010。</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/12562/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/12562/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年03月03日 2014年03月10日 2014年03月17日 2014年03月24日 2014年03月31日</p></div>
说明：雨滴传感器在正常没雨（暴露在空气中）情况下，传回的值大约为 1000 左右；将它放在雨中，得到的数值会因雨的大小不同而不同。测试时将不同数目的水滴滴在传感器上，得到的数值图见上所示。	

2.7 火焰传感器测试

时间	截图
2014 年 4 月 13 日 20:42:19	<div><p>火焰传感器</p><p>编辑 删除</p><p>火焰传感器，模拟输入引脚A4</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/12763/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/12763/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年04月13日 20:42:19 数值: 29 火焰(值)</p></div>
2014 年 4 月 7 日 18:38:08	<div><p>火焰传感器</p><p>编辑 删除</p><p>火焰传感器，模拟输入引脚A4</p></div> <div><p>数据曲线</p><p>URL: <a href="http://api.yeelink.net/v1.0/device/7921/sensor/12763/datapoints">http://api.yeelink.net/v1.0/device/7921/sensor/12763/datapoints</a></p><p>缩放 1h 3h 12h 1d 3d 1w 1m 3m 6m 1y All</p><p>2014年03月24日 19:00:00 数值: 988.89 火焰(值)</p></div>
说明：火焰传感器在没有检测到火时，传回的数值为 1000 左右（不同的传感器会不同）；当检测到火焰时会根据不同的强度传回不同的数值。测试时将打火机打燃，靠近火焰传感器得到的数值为 29，具体参见上图。	

2.8 电器控制测试

开关	截图	说明
Switch1	<div><p>Switch1 灯光1</p><p>编辑 删除</p><p>Switch1 灯光1</p></div> <div><p>状态URL</p><p><a href="http://api.yeelink.net/v1.0/device/8562/sensor/13552/datapoints">http://api.yeelink.net/v1.0/device/8562/sensor/13552/datapoints</a></p><p>控制操作</p><p>按动图标以改变开关的状态</p></div>	灯光 1 控制开关，此状态为开。
Switch2	<div><p>Switch2 电机1</p><p>编辑 删除</p><p>Switch2 电机2</p></div> <div><p>状态URL</p><p><a href="http://api.yeelink.net/v1.0/device/8562/sensor/13553/datapoints">http://api.yeelink.net/v1.0/device/8562/sensor/13553/datapoints</a></p><p>控制操作</p><p>按动图标以改变开关的状态</p></div>	电机 1 控制开关，此状态为关。

## 2.8 整体测试

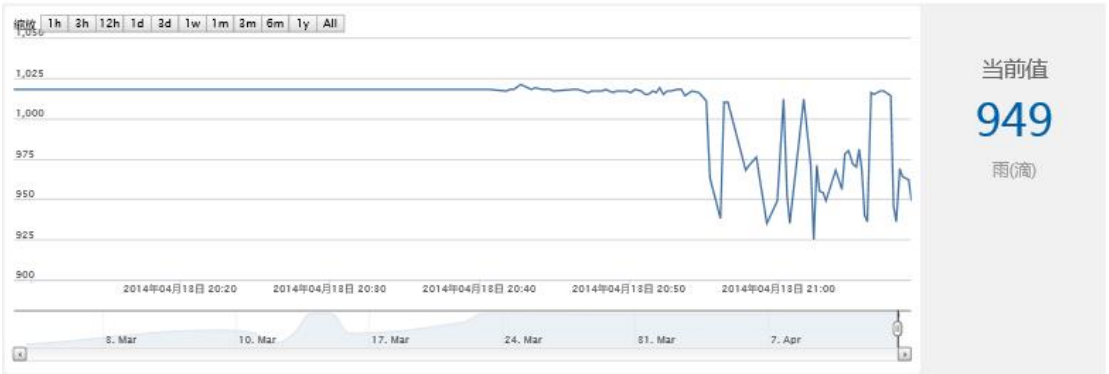
传感器数值读取网站：<http://www.yeelink.net/devices/7921>，以下截取了部分数据图。



Smart1

传感器数据 设备详情 用户评论(0)

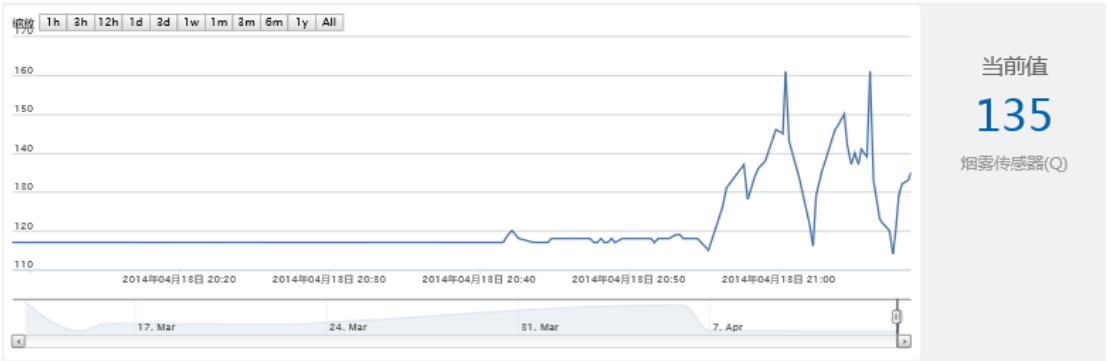
### 雨滴传感器



### 火焰传感器



### MQ2烟雾传感器



### LM35温度传感器1



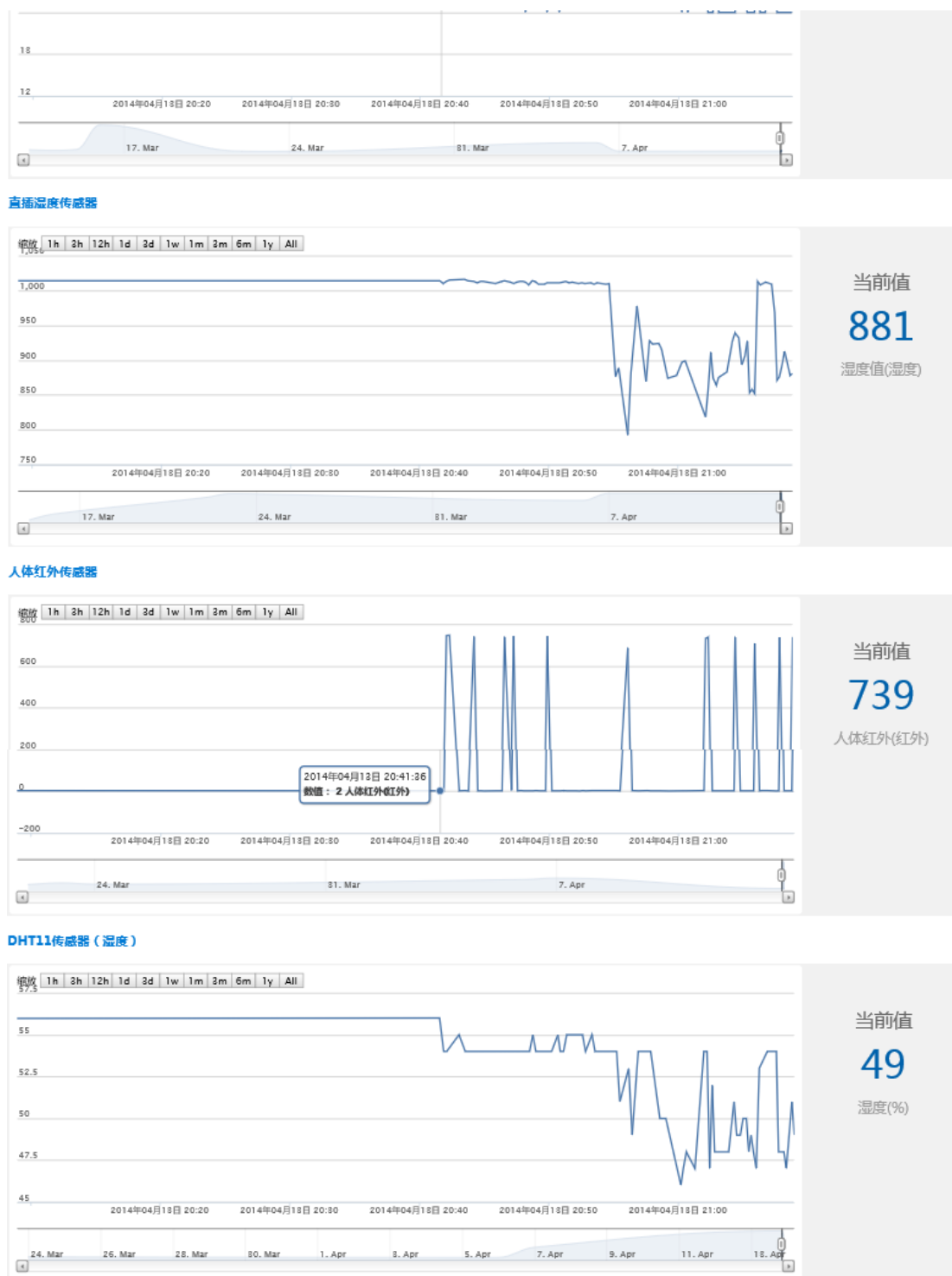


图 2.1 整体测试数据图

## 附录 A：部分源程序

```
#include <Ethernet.h>
#include <WiFi.h>
#include <SPI.h>
#include <yl_data_point.h>
#include <yl_device.h>
#include <yl_w5100_client.h>
#include <yl_wifi_client.h>
#include <yl_messenger.h>
#include <yl_sensor.h>
#include <yl_value_data_point.h>
#include <yl_sensor.h>
#include <Wire.h>
#include <dht.h>

dht DHT;

//this example reads data from a lm35dz sensor, convert value to degree Celsius
//and then post it to yeelink.net
yl_device ardu(7921);
yl_sensor therm(13050, &ardu);
yl_sensor therm_1(12562, &ardu);
yl_sensor therm_2(13416, &ardu);
yl_sensor therm_3(13170, &ardu);
yl_sensor therm_4(12763, &ardu);
yl_sensor therm_5(13123, &ardu);
yl_sensor therm_6(13545, &ardu);
yl_sensor therm_7(13546, &ardu);
//replace first param value with ur u-apikey
yl_w5100_client client;
yl_messenger messenger(&client, "a6a75584ce63ad5aa7a504a33b726d0a", "api.yeelink.net");
int LED1 = 7;
int SW1 = 6;
int buzzer=8;
#define DHT11_PIN 4
const int THERM_PIN = A0;
const int THERM_PIN_1 = A1;
const int THERM_PIN_2 = A2;
const int THERM_PIN_3 = A3;
const int THERM_PIN_4 = A4;
const int THERM_PIN_5 = A5;
float lm35_convertor(int analog_num)
{
    return analog_num * (5.0 / 1024.0 * 100);
}
```

```

void setup()
{
    pinMode(LED1,OUTPUT);
    pinMode(SW1,OUTPUT);
    pinMode(buzzer,OUTPUT);
    digitalWrite(buzzer,LOW);

    Serial.begin(9600);
    byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xAA};
    Ethernet.begin(mac);
}

void loop()
{

    int val_mq2 = analogRead(THERM_PIN);
    Serial.println(val_mq2);
    yl_value_data_point dp(val_mq2);
    therm.single_post(messenger, dp);
    if(val_mq2>300)
    {
        blinkLED(LED1,10);
        digitalWrite(SW1,LOW);
    }
    else if(val_mq2>300&&val_mq2<400)
        blinkLED(LED1,100);
    else
        digitalWrite(LED1,LOW);
    int v5 = analogRead(THERM_PIN_5);
    int ST = lm35_convertor(v5)-14;
    Serial.println(ST);
    yl_value_data_point dp5(ST);
    therm_5.single_post(messenger, dp5);
    if(ST>=15)
    //    digitalWrite(SW1,LOW);
        buzzerbuzzer(1);
    else
        ;
    //    digitalWrite(SW1,HIGH);
    int v1 = analogRead(THERM_PIN_1);
    Serial.println(v1);
    yl_value_data_point dp1(v1);
    therm_1.single_post(messenger, dp1);
    if(v1<600)

```

```

        digitalWrite(SW1,LOW);
    else
        digitalWrite(SW1,HIGH);
    int v4 = analogRead(THERM_PIN_4);
    Serial.println(v4);
    yl_value_data_point dp4(v4);
    therm_4.single_post(messenger, dp4);
    int v3 = analogRead(THERM_PIN_3);
    Serial.println(v3);
    yl_value_data_point dp3(v3);
    therm_3.single_post(messenger, dp3);
    int v2 = analogRead(THERM_PIN_2);
    Serial.println(v2);
    yl_value_data_point dp2(v2);
    therm_2.single_post(messenger, dp2);
    int chk = DHT.read11(DHT11_PIN);
    // switch (chk)
    //{
        //case 0: Serial.print("OK,\t"); break;
        //case -1: Serial.print("Checksum error,\t"); break;
        //case -2: Serial.print("Time out error,\t"); break;
        //default: Serial.print("Unknown error,\t"); break;
    //}
    Serial.println(DHT.humidity);
    yl_value_data_point dp11(DHT.humidity);
    therm_6.single_post(messenger, dp11);
    therm_7.single_post(messenger, dp11);
}

unsigned int blinkLED(int x,int period)
{
    unsigned long time = millis();
    unsigned int signal;
    if((time / period) % 2 == 0)
    {
        digitalWrite(x, HIGH);
    }
    else
    {
        digitalWrite(x, LOW);
    }
}

unsigned int buzzerbuzzer(int x)
{
    int i,j;

```



```
if(x==1)
for(j=0;j<10;j++)
{
for(i=0;i<150;i++)
{
digitalWrite(buzzer,HIGH);
delay(1);
digitalWrite(buzzer,LOW);
delay(1);
}
}
else if(x==2)
for(i=0;i<100;i++)
{
digitalWrite(buzzer,HIGH);
delay(2);
digitalWrite(buzzer,LOW);
delay(2);
}
}
```