树莓派开发

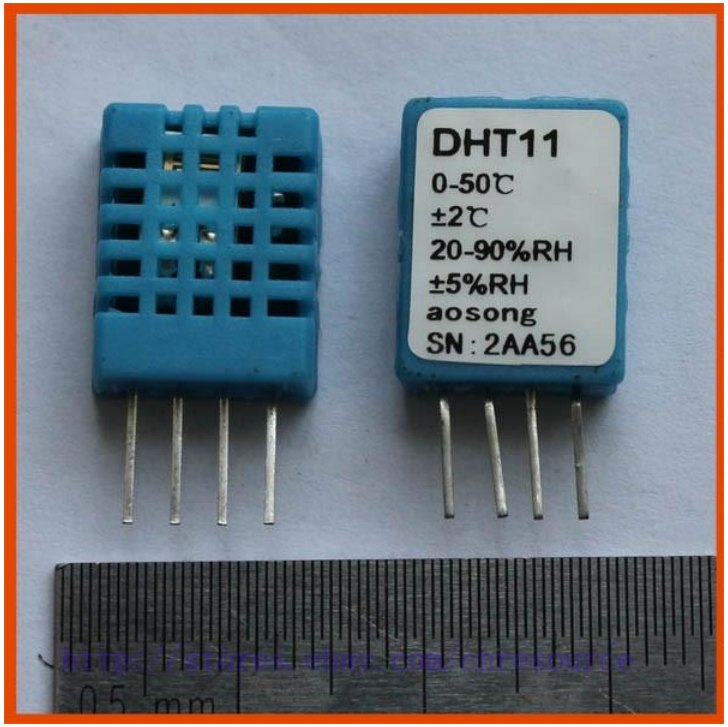
21 用树莓派监控温度湿度

（arduino版）

1、DHT11温湿度传感器介绍

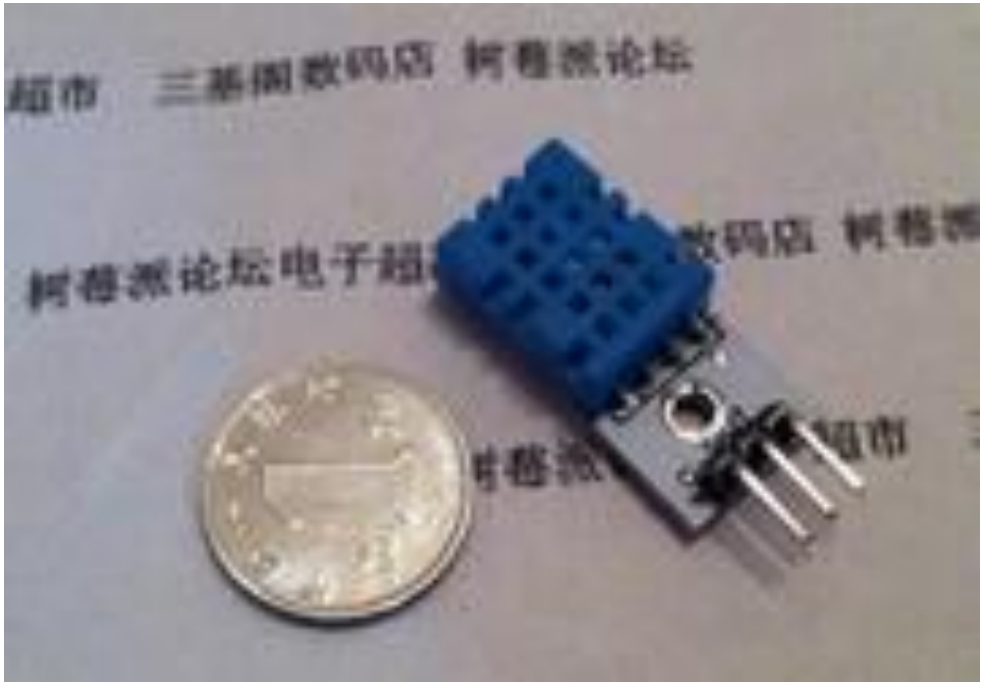
- DHT11数字温湿度传感器是一款含有已校准数字信号输出的温湿度复合传感器。
- 它应用专用的数字模块采集技术和温湿度传感技术，确保产品具有极高的可靠性与卓越的长期稳定性。

- 传感器包括一个电阻式感湿元件和一个NTC测温元件，并与一个高性能8 位单片机相连接。
- 因此该产品具有品质卓越、超快响应、抗干扰能力强、性价比极高等优点。



1、DHT11温湿度传感器介绍

- 每个DHT11传感器都在极为精确的湿度校验室中进行校准。校准系数以程序的形式储存在OTP 内存中，传感器内部在检测信号的处理过程中要调用这些校准系数。
- 单线串行接口（单线双向），使系统集成变得简易快捷。超小的体积、极低的功耗，信号传输距离可达20 米以上，使其成为各类应用甚至最为苛刻的应用场合的最佳选则。
- DHT11 数字温湿度传感器模块为3针PH2.0 封装，连接方便。
- 读取数据只需要占用一个IO口。能够同时测量温度和相对湿度。



1、DHT11温湿度传感器介绍

性能描述：

- 1. 供电电压：3-5.5V
- 2. 供电电流：最大2.5Ma
- 3. 温度范围：0-50℃ 误差±2℃
- 4. 湿度范围：20-90%RH 误差±5%RH
- 5. 响应时间: 1/e(63%) 6-30s
- 6. 测量分辨率分别为 8bit（温度）、8bit（湿度）
- 7. 采样周期间隔不得低于1 秒钟
- 8. 模块尺寸：30x20mm

1、DHT11温湿度传感器介绍

传感器的时序

- 由于DHT11传感器是采用单线制串行通讯的方法，进行采样数据的，要配合时序，一位位从单条通讯线传过来，再合成8位字节，然后还要进行校验和，判断数据传送是否正确
- DATA 用于微处理器与 DHT11之间的通讯和同步,采用单总线数据格式,一次通讯时间4ms左右,数据分小数部分和整数部分,具体格式在下面说明,当前小数部分用于以后扩展,现读出为零.操作流程如下:

  一次完整的数据传输为40bit,高位先出。
- 数据格式:

  8bit湿度整数数据+8bit湿度小数数据

  +8bi温度整数数据+8bit温度小数数据

  +8bit校验和

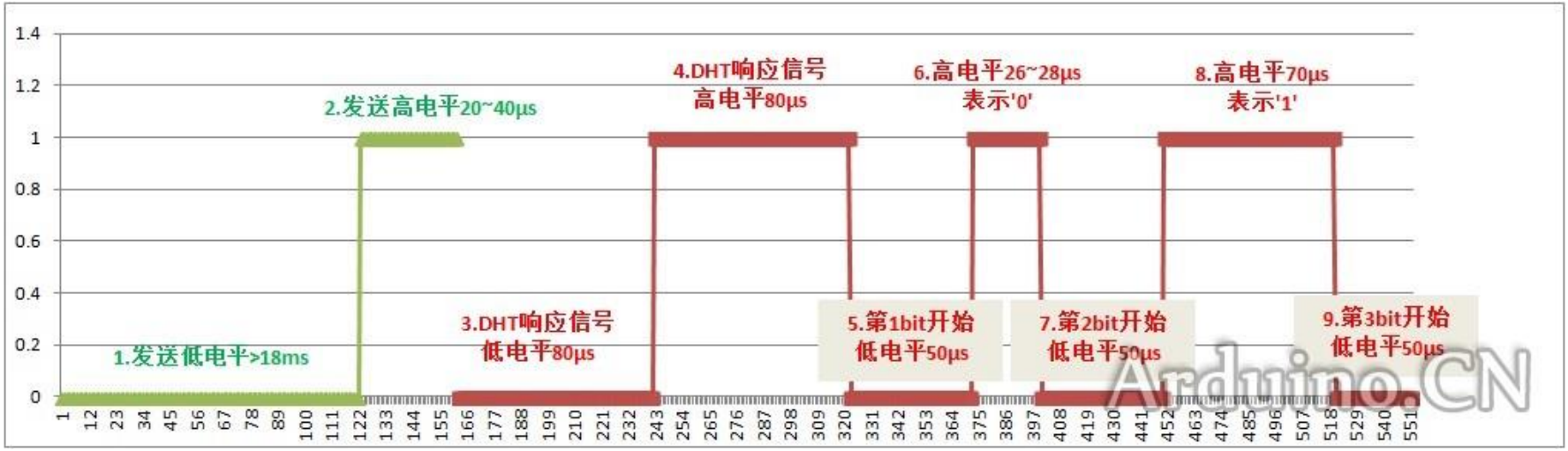
1、DHT11温湿度传感器介绍

传感器的时序

- 数据传送正确时校验和数据等于“**8bit湿度整数数据+8bit湿度小数数据+8bi温度整数数据+8bit温度小数数据**”所得结果的末8位。
- 用户MCU发送一次开始信号后,DHT11从低功耗模式转换到高速模式,等待主机开始信号结束后,DHT11发送响应信号,
- 送出40bit的数据,并触发一次信号采集,用户可选择读取部分数据.从模式下,DHT11接收到开始信号触发一次温湿度采集,如果没有接收到主机发送开始信号,DHT11不会主动进行温湿度采集.采集数据后转换到低速模式。

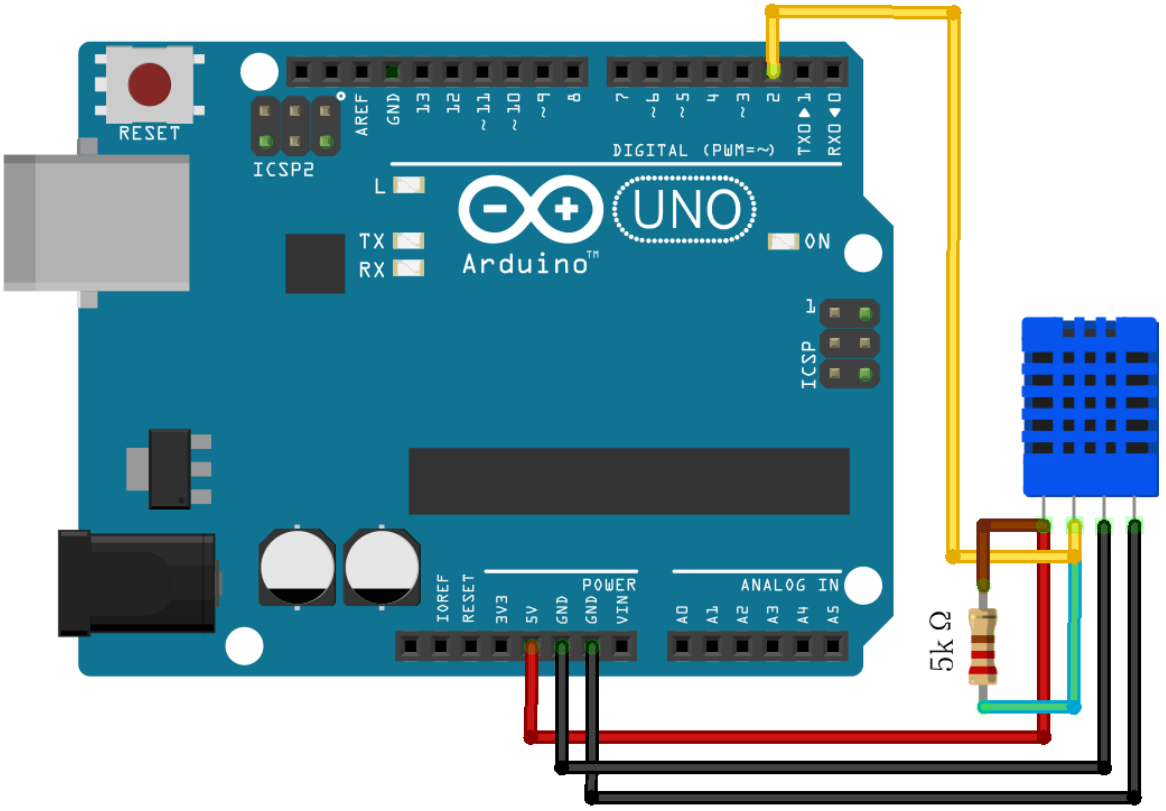
1、DHT11温湿度传感器介绍

传感器的时序



2、DHT11传感器硬件连接

- 将DHT11温湿度传感器的VCC、GND分别连接至Arduino Uno控制器的+5V、GND，以给DHT11提供电源
- DHT11模块的DOUT引脚接至ArduinoUno控制器数字引脚D2，且并联5kΩ的上拉电阻，DHT11模块的NC引脚也连接至GND
- 上拉电阻不用也可以，主要起保护作用。



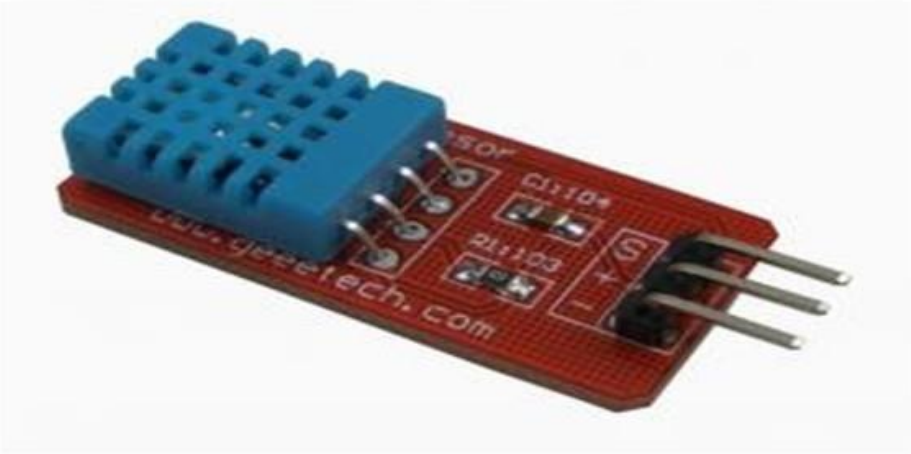
2、DHT11传感器硬件连接

- 上拉电阻的作用：
- 总体而言：上拉就是将不确定的信号通过一个电阻嵌位在高电平,电阻同时起限流作用，具体情况有：
- 1、TTL驱动CMOS时,如果TTL输出最低高电平低于CMOS最低高电平时,提高输出高电平值
- 2、OC门必须加上拉,提高电平值
- 3、加大输出的驱动能力(单片机较常用)
- 4、CMOS芯片中(特别是门的芯片),为防静电干扰,不用的引脚也不悬空,一般上拉,降低阻抗,提供泄荷通路
- 5、提高输出电平,提高芯片输入信号的噪声容限,增强抗干扰
- 6、提高总线抗电磁能力,空脚易受电磁干扰
- 7、长线传输中加上拉,是阻抗匹配抑制反射干扰

2、DHT11传感器硬件连接

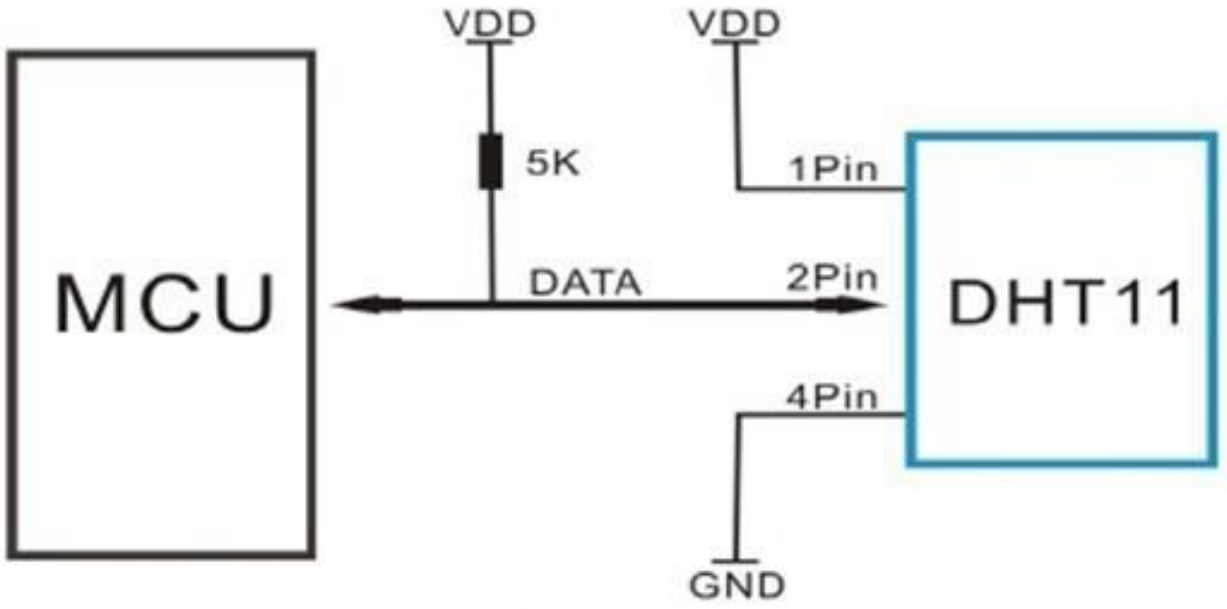
- 引脚定义

Pin	名称	注释
1	VDD	供电 3－5.5VDC
2	DATA	串行数据，单总线
3	NC	空脚，请悬空
4	GND	接地，电源负极



2、DHT11温湿度传感器硬件

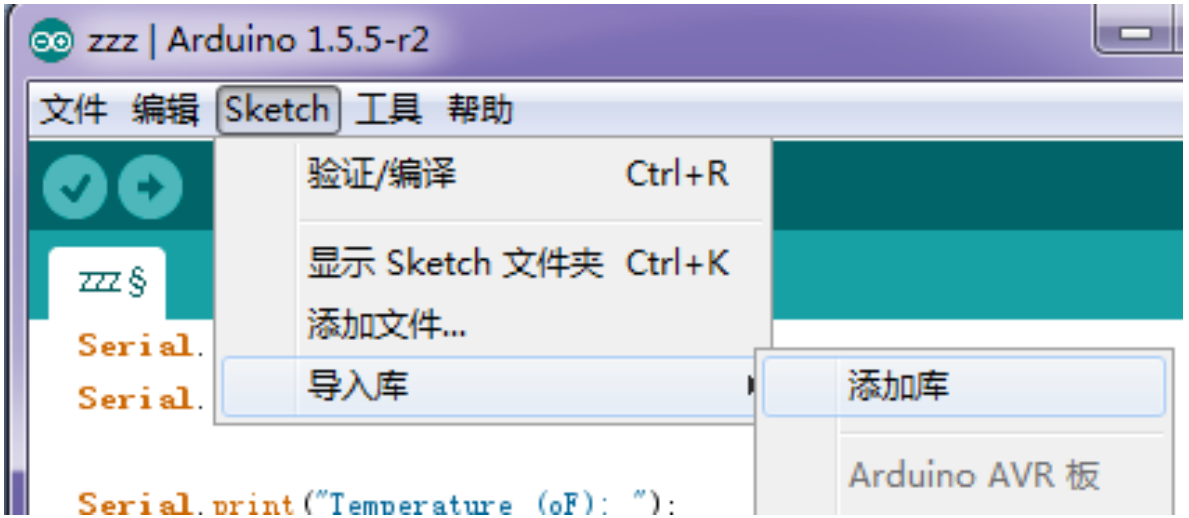
- **接线方法（蓝色正面向上）：**
  VCC （左） → 3.3V/5V电源正极
  GND （右） →电源负极
  DATA （中） →单片机IO口
- 切勿将VCC与GND接反，接反必烧！



典型应用电路

3、DHT11温湿度传感器软件

- 下载库文件，解压在arduino的IDE下libraries文件下
- http://www.arduino.cn/forum.php?mod=attachment&aid=ODEwfGFhMjQwNzJifDE0MTg3MTI5NzF8MzE3MTd8MTQyOQ%3D%3D
- 用arduino的IDE上的Sketch的导入库，直接导入该压缩文件



3、DHT11温湿度传感器软件

- 在IDE上，创建以下程序

```
double Fahrenheit(double celsius)
{    return 1.8 * celsius + 32;
}    //摄氏温度度转化为华氏温度
 double Kelvin(double celsius)
{     return celsius + 273.15;
}     //摄氏温度转化为开氏温度
// 露点计算（点在此温度时，空气饱和并产生露珠）
double dewPoint(double celsius, double humidity)
{       double A0= 373.15/(273.15 + celsius);
        double SUM = -7.90298 * (A0-1);
        SUM += 5.02808 * log10(A0);
        SUM += -1.3816e-7 * (pow(10, (11.344*(1-1/A0)))-1) ;
        SUM += 8.1328e-3 * (pow(10,(-3.49149*(A0-1)))-1) ;
        SUM += log10(1013.246);
        double VP = pow(10, SUM-3) * humidity;
        double T = log(VP/0.61078);   // temp var
        return (241.88 * T) / (17.558-T);
}
```

3、DHT11温湿度传感器软件


```
// 快速计算露点，速度是5倍dewPoint()
double dewPointFast(double celsius, double humidity)
{       double a = 17.271;
        double b = 237.7;
        double temp = (a * celsius) / (b + celsius) + log(humidity/100);
        double Td = (b * temp) / (a - temp);
        return Td;
}
#include <dht11.h> //导入的库文件
dht11 DHT11;
#define DHT11PIN 2 //使用arduino uno D2脚
void setup()
{ Serial.begin(9600); //串口9600/s
  Serial.println("DHT11 TEST PROGRAM ");
  Serial.print("LIBRARY VERSION: ");
  Serial.println(DHT11LIB_VERSION);
  Serial.println();
}
```

3、DHT11温湿度传感器软件

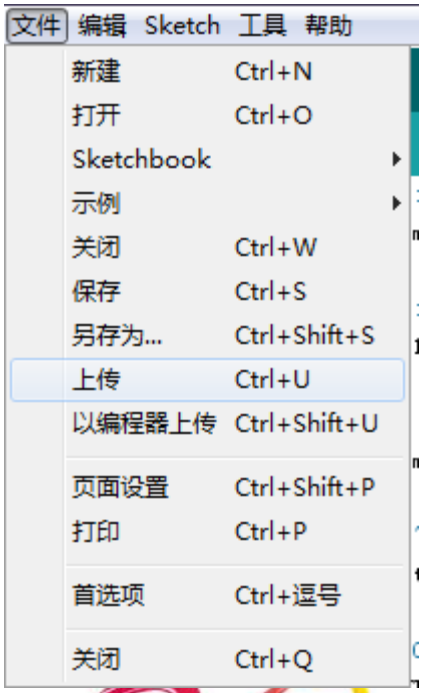

```
void loop()
{  Serial.println("\n");
   int chk = DHT11.read(DHT11PIN); //读DHT11，DHT11.read是dht11.cpp提供的例程
   Serial.print("Read sensor: ");
  switch (chk)
  {
    case DHTLIB_OK:
                Serial.println("OK");
                break;
    case DHTLIB_ERROR_CHECKSUM:
                Serial.println("Checksum error");
                break;
    case DHTLIB_ERROR_TIMEOUT:
                Serial.println("Time out error");
                break;
    default:
                Serial.println("Unknown error");
                break;
  }
```

3、DHT11温湿度传感器软件

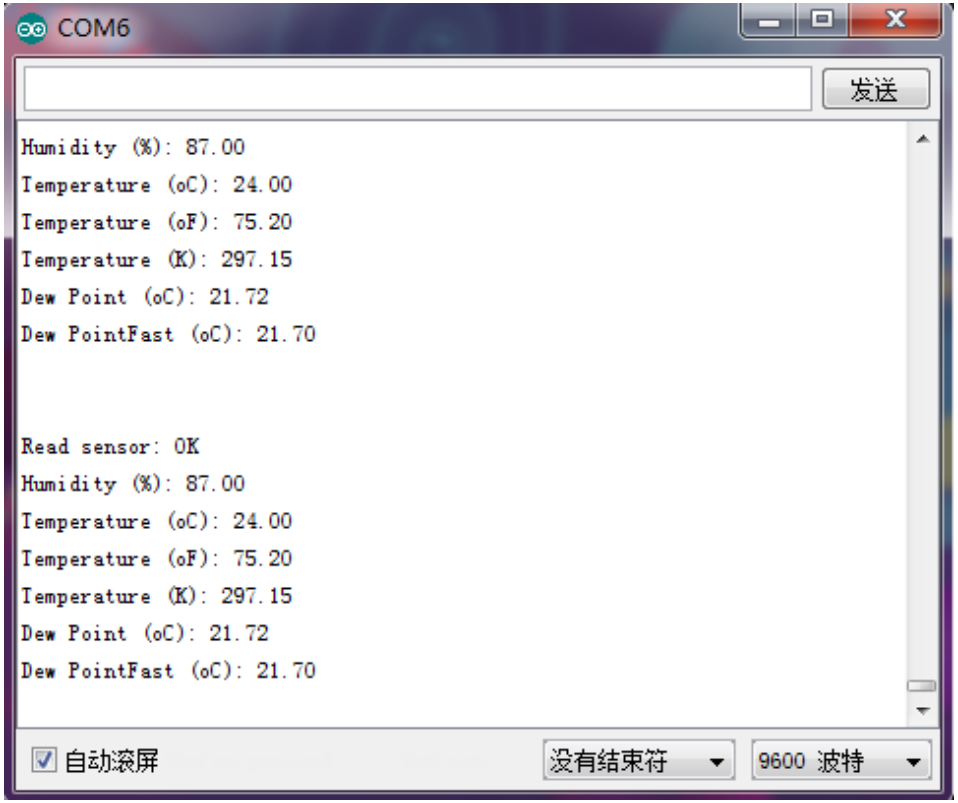

```
Serial.print(“Humidity (%): ”); //湿度
Serial.println((float)DHT11.humidity, 2);

Serial.print(“Temperature (oC): ”); //摄氏温度
Serial.println((float)DHT11.temperature, 2);

Serial.print(“Temperature (oF): ”); //华氏温度
Serial.println(Fahrenheit(DHT11.temperature), 2);

Serial.print(“Temperature (K): ”); //K氏温度
Serial.println(Kelvin(DHT11.temperature), 2);

Serial.print(“Dew Point (oC): ”);//露点
Serial.println(dewPoint(DHT11.temperature, DHT11.humidity));

Serial.print(“Dew PointFast (oC): ”); //快速计算的露点
Serial.println(dewPointFast(DHT11.temperature, DHT11.humidity));
delay(2000);
}
```

3、DHT11温湿度传感器软件

- 在arduino IDE菜单上，选择验证/编译
- 编译通过后，将代码上传到arduino uno
- 将arduino uno 重启一下。



- 在arduino IDE的串口监视器上，看看运行效果：



- 把探头放到冒着热气的茶杯上面，看看有什么变化？

3、DHT11温湿度传感器软件

- 作为一般应用，由于使用了标准例库，因此，在读取DHT11温度和湿度时，我们并不关心“时序”
- 与第18节，用wiringpi（C语言）编写的相同功能的代码相比，实现更为简单。

4、在Android手机上监控你家的温度

- 从YeeLink下载iOS/Android手机APP，实时监控家中的温度

- 也可以利用YeeLink API开发自己的APP
- 下学期的课程是否有：《移动互联网的应用开发》
- 开发重点在移动端（客户端），主要有基于IE和基于iOS/Android两种
- 服务器端就是YeeLink

移动客户端

iOS和Android移动客户端下载。

移动客户端iOS版

By Yeelink, 2013.5.30

移动客户端Android版

By Yeelink, 2013.5.2

http://www.yeelink.net/developer
或者用老师提供的YeelinkV1.0.4.apk
记得要先注册哟！
上去可以看见很多人的“实时温度”

好好学习，增长姿势！

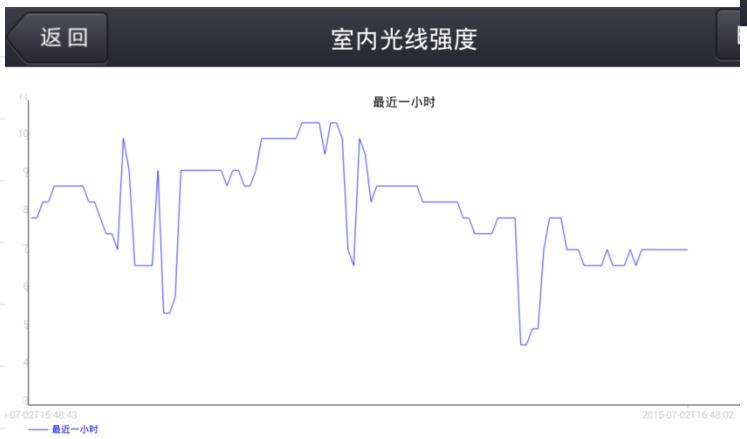
4、在Android手机上监控你家的温度

- 从YeeLink下载iOS/Android手机APP，实时监控家中的温度

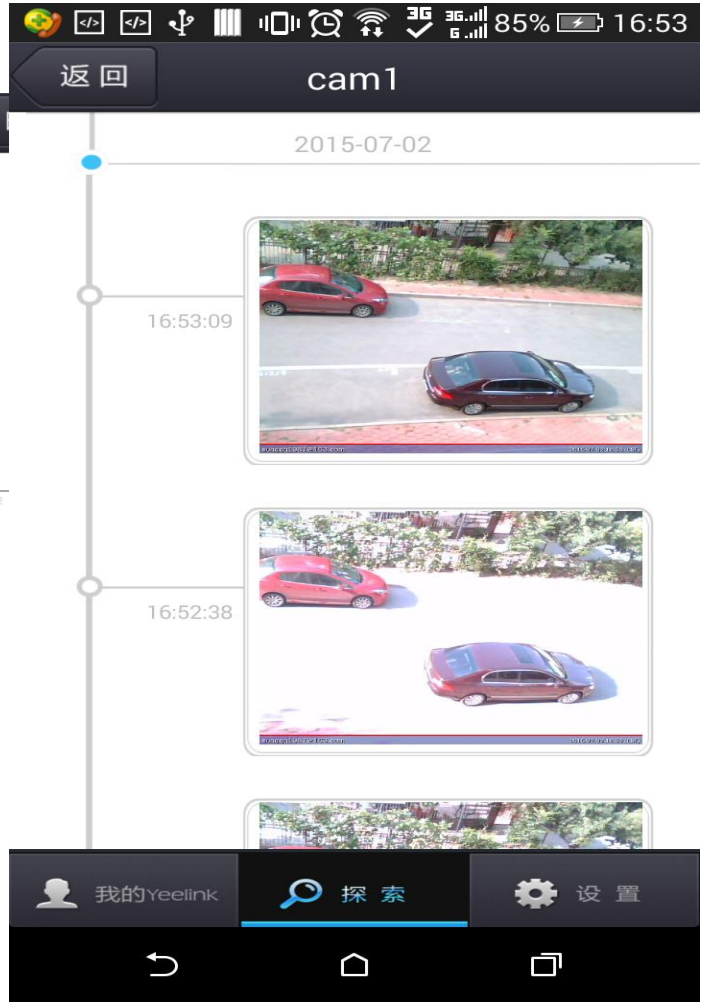
4、在Android手机上监控你家的温度

- 在Android手机上运行YeelinkV1.0.4.apk，可以看到很多人上传的数据（不保密，所以，传视频要当心哟！）

- 选室内光线强度：



- 选一个视频图像：
- 就是这个样子的。



5、与YeeLink结合

- 参考这些案例，开发自己更“炫”的应用

- YeeLink的定位：
  - 实时数据保存和发布服务器
- 树莓派的定位：
  - 实时数据的采集（可以有很多）
- iOS/Android（当然也可以是IE）的定位：
  - 用户界面+数据应用（用这些数据做什么）
  - 真正就是“物联网”——把“物”联起来干什么？
- YeeLink与云服务器的不同：暂时还没有数据处理能力

DS18B20温度传感器

DS18B20温度传感器DS18B20是美国DALLAS半导体公司的数字化单总线智能温度传感器，与传统的热敏电阻相比，它能够直接读出被测温度，并且可根据实际要求通过简单的编程实现9～12位的数字值读数方式。

从DS18B20读出信息或写入信息仅需要一根线(单总线)读写，总线本身也可以向所挂接的设备供电，而无需额外电源。

DS18B20的性能特点如下:

(1) 单线接口方式实现双向通讯;

(2) 供电电压范围：+3.0V～+5.5V，可用数据线供电;

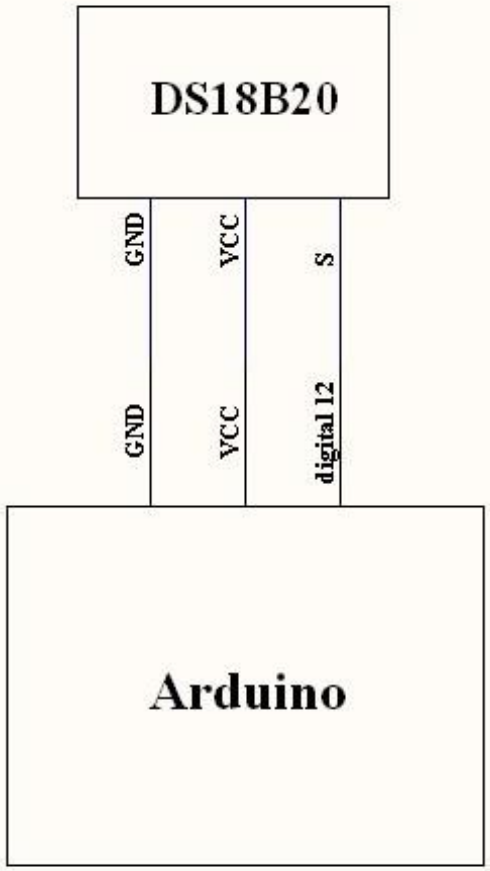
(3) 测温范围：-55～+125℃，固有测温分辨率为0.5℃;

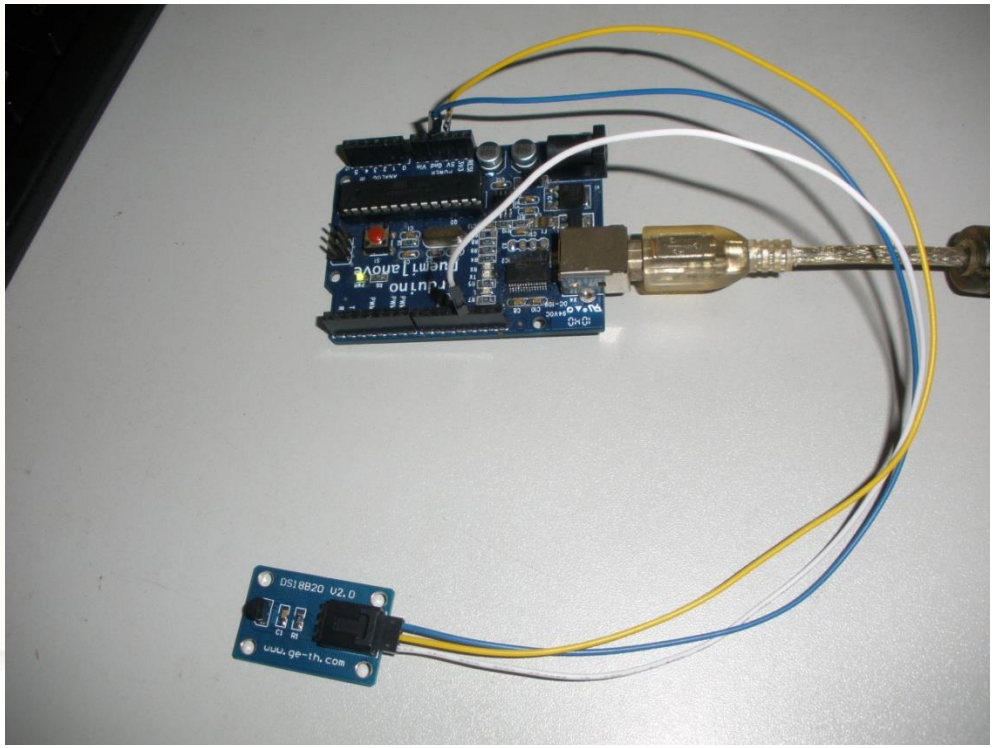
(4) 通过编程可实现9～12位的数字读数方式;

(5) 支持多点的组网功能，多个DS18B20可以并联在唯一的单总线上，实现多点测温。

与arduino

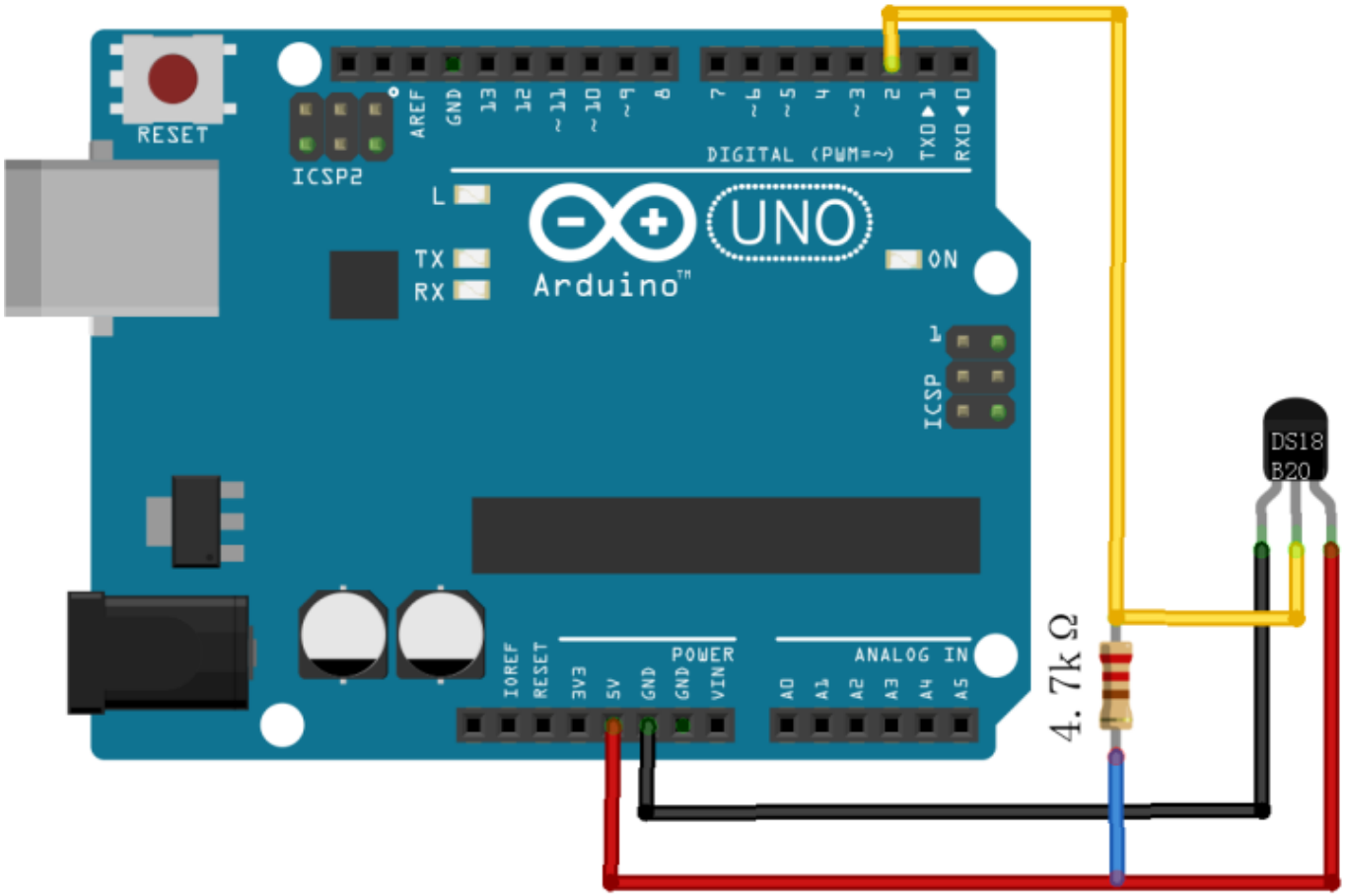
DS18B20与arduino接线图





DS18B20与arduino接线图

将DS18B20温度传感器的VCC和GND分别连接至Arduino Uno控制器的+5V和GND，以给DS18B20提供电源，DS18B20的DQ引脚接至ArduinoUno控制器数字引脚D2，且并联4.7kΩ的上拉电阻，如图所示。（我们不用



□DS18B20编程：

Arduino要实现对DS18B20的操作，需要OneWire和Dallas Temperature Control两个库文件，下载地址分别为：http://playground.arduino.cc/Learning/OneWire和https://github.com/milesburton/Arduino-Temperature-Control-Library。

一定要记住：下载这两个库文件（××××．h）文件后，在Arduino的libraries目录下，创建一个ds18b20目录（不能用中文名啊），把.h文件拷到这个目录下，重启Arduino后，在库目录中，可以看见这个库。

Dallas Temperature Control函数库是基于OneWire函数库进行开发的，更便于使用，下面讲解一下主要函数的功能和用法。

```
#include <OneWire.h>
#include <DS18B20.h>
// Low/high alarm in degrees Celcius.
#define LOW_ALARM 20
#define HIGH_ALARM 25
// 1-Wire devices connected to digital pin 2 on the Arduino.
DS18B20 ds(2);
// Address of the device.
uint8_t address[] = {40, 250, 31, 218, 4, 0, 0, 52};
// Indicates if the device was successfully selected.
uint8_t selected;

void setup()
{
  Serial.begin(9600);

  // Select device.
  selected = ds.select(address);
    if(selected)
  {
    // Set alarms.
    ds.setAlarms(LOW_ALARM, HIGH_ALARM);
  }
  else
  {
    Serial.println("Device not found!");
  }
}
```

```
void loop()
{
  // Check if the device has an active alarm condition.
  if(selected)
  {
    if(ds.hasAlarm())
    {
      Serial.print("Warning!  Temperature is ");
      Serial.print(ds.getTempC());
      Serial.println("  C");
    }
  }
  else
  {
    Serial.println("Device  not found!");
  }

  // Wait 10 seconds.
  delay(10000);
}
```

代码：

(1) void begin(void)：初始化，无输入参数，无返回参数。

(2) getDeviceCount(void)：获取单总线上所连接器件的总数，无输入参数，返回参数为器件数目。

(3) validAddress(uint8_t*)：验证指定地址的器件是否存在，输入参数为器件地址，返回参数为布尔型。

(4) getAddress(uint8_t*， const uint8_t)：验证的器件的地址与索引值是否匹配，输入参数为器件地址和索引值，返回参数为布尔型。

(5) getResolution(uint8_t*)：获取指定器件的精度，输入参数为器件地址，返回参数为精度位数。

(6) setResolution(uint8_t*，uint8_t)：设置器件的精度，输入参数为器件地址和精度位数，无返回参数。精度位数有9，10，11和12可供选择。

代码：

(7) requestTemperatures(void)：向单总线上所有器件发送温度转换的请求，无输入参数，无返回参数。

(8) requestTemperaturesByAddress(uint8_t*)：向单总线上指定地址的器件发送温度转换的请求，输入参数为器件地址，无返回参数。

(9) requestTemperaturesByIndex(uint8_t) ：向单总线上指定索引值的器件发送温度转换的请求，输入参数为器件索引值，无返回参数。

(10) getTempC(uint8_t*)：通过器件地址获取摄氏温度，输入参数为器件地址，返回参数为摄氏温度。

(11) getTempF(uint8_t*)：通过器件地址获取华氏温度，输入参数为器件地址，返回参数为华氏温度。

(12) getTempCByIndex(uint8_t)：通过索引值来获取摄氏温度，输入参数为器件索引值，返回参数为摄氏温度。

(13) getTempFByIndex(uint8_t)：通过器件索引值来获取华氏温度，输入参数为器件索引值，返回参数为华氏温度。

温度测试效果