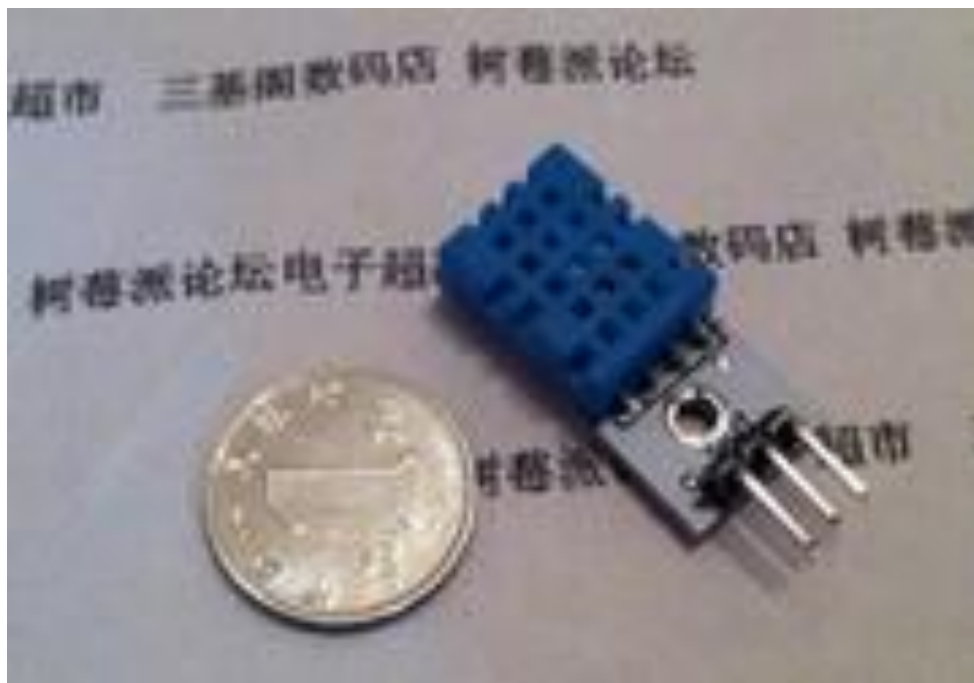


树莓派开发

17 用树莓派控制温度

1、DHT11温湿度传感器介绍

- DHT11:
- DHT11是一款比较便宜的温湿度传感器模块。
- 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

□ DHT11基本参数:

- 尺寸: 长28mmX宽12mmX高7.2mm
- 传感器型号: 奥松DHT11温湿度传感器
- 工作电压: 直流5V

□ 特点:

- 1、湿度测量范围: 20---90%RH
- 2、湿度测量精度: $\pm 5\%RH$
- 3、温度测量范围: 0---50℃
- 4、温度测量精度: $\pm 2^{\circ}C$
- 5、工作电压: DC5V/3.3V
- 6、数字信号输出
- 7、数据端口带上拉电阻
- 8、带3mm固定螺丝孔, 方便安装

1、DHT11温湿度传感器介绍

□ 接线方法：

VCC → 3.3V/5V电源正极

GND → 电源负极

DATA → 单片机IO口

□ 切勿将VCC与GND接反，接反必烧！



。 DHT11的数据手册可以看这里：

<http://wenku.baidu.com/view/1955cc70a417866>

2、pi4j介绍

- pi4j是基于wiringpi开发的，通过java来控制树莓派GPIO口的库文件。
- 在java程序中引入相关类就可以使用已经封装好的方法控制树莓派GPIO口。
- pi4j官网：<http://pi4j.com/>
- pi4j下载：<http://pi4j.com/download.html>
- pi4j安装：<http://pi4j.com/install.html>
- pi4j文档：<http://pi4j.com/apidocs/index.html>

3、编译和运行

- ❑ 使用了pi4j后，编译.java文件和运行.class文件时，需要使用root权限。
- ❑ 编译：
sudo javac -
classpath .:classes:/opt/pi4j/lib/"*" YourJavaFile.java
- ❑ 运行：
sudo java -
classpath .:classes:/opt/pi4j/lib/"*" YourClassFile.class
- ❑ 如果使用sudo编译和运行时，提示找不到javac和java，这是由于root用户没有加载环境变量，所以找不到java的执行命令。
- ❑ 可以使用如下命令切换到带有环境变量的root用户后，再执行编译和运行：
sudo su - root



4、wiringpi介绍

- ❑ wiringpi是通过C语言控制树莓派GPIO口的头文件。
- ❑ 在C语言中包含这个头文件之后，可以很简单的调用已经封装好的方法来控制树莓派GPIO口。
- ❑ 程序需要先安装。
- ❑ wiringpi官网：<http://wiringpi.com/>
- ❑ wiringpi下载和安装：<http://wiringpi.com/download-and-install/>
- ❑ wiringpi文档：<http://wiringpi.com/reference/>

4、wiringpi介绍

- 编译和运行：
- 写好C文件后，通过如下命令进行编译：
`gcc -Wall -o executefilename cfilename.c -lwiringPi`
- gcc是编译器，-Wall是在编译时显示警告信息，-o executefilename cfilename.c是将cfilename.c文件编译成文件名为executefilename的可执行文件，-lwiringPi是将wiringPi头文件包含在可执行文件中。
- 编译完之后会生成文件名为executefilename的文件，使用root权限执行如下命令即可运行：
`sudo ./executefilename`

5、硬件连接

- 我们使用的是B版树莓派，接口如下图。
- 其中的NAME一列是树莓派实际的IO口功能。
- PIN#一列是后面要介绍的wiringpi和pi4j库文件编程使用的接口编号。





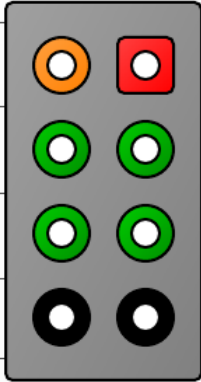


Raspberry Pi P1 Header					
PIN #	NAME		NAME	PIN #	
	3.3 VDC Power	1		2	5.0 VDC Power
8	SDA0 (I2C)	3		4	DNC
9	SCL0 (I2C)	5		6	0V (Ground)
7	GPIO 7	7		8	TxD 15
	DNC	9		10	RxD 16
0	GPIO 0	11		12	GPIO1 1
2	GPIO2	13		14	DNC
3	GPIO3	15		16	GPIO4 4
	DNC	17		18	GPIO5 5
12	MOSI	19		20	DNC
13	MISO	21		22	GPIO6 6
14	SCLK	23		24	CE0 10
	DNC	25		26	CE1 11

<http://www.pi4j.com>

5、硬件连接

- 我们使用的是B版树莓派，接口如下图。
- 其中的NAME一列是树莓派实际的IO口功能。
- PIN#一列是后面要介绍的wiringpi和pi4j库文件编程使用的接口编号。

Raspberry Pi P5 Header					
PIN #	NAME			NAME	PIN #
	3.3 VDC Power	2		1	5.0 VDC Power
18	GPIO18	4		3	GPIO17
20	GPIO20	6		5	GPIO19
	0V (Ground)	8		7	0V (Ground)
					
http://www.pi4j.com					

5、硬件连接

□ DHT11引脚有格子空的为正面，引脚朝下，从左数4个引脚分别为

引脚名称	引脚功能	连接到树莓派的GPIO
1. VCC	正极	连接3.3V或5V物理接口1，即3.3V
2. DATA	数据输入输出	物理接口7，即GPIO 7
3. NC	悬空	不连
4. GND	负极	物理接口 6，即GND

□ 此外按照DHT11数据手册要求，在DATA和VCC之间连接了一个4.7K欧姆的电阻，起到上拉作用。

6、软件编写

使用wiringpi编写C程序:

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#define MAX_TIME 85
#define DHT11PIN 7
#define ATTEMPTS 5           //retry 5 times when no response
int dht11_val[5]={0,0,0,0,0};
int dht11_read_val(){
    uint8_t lststate=HIGH;    //last state
    uint8_t counter=0;
    uint8_t j=0,i;
    for(i=0;i<5;i++)
        dht11_val[i]=0;
```



6、软件编写

```
//host send start signal
```

```
pinMode(DHT11PIN,OUTPUT);    //set pin to output
```

```
digitalWrite(DHT11PIN,LOW);  //set to low at least 18ms
```

```
delay(18);
```

```
digitalWrite(DHT11PIN,HIGH); //set to high 20-40us
```

```
delayMicroseconds(40);
```

```
//start recieve dht response
```

```
pinMode(DHT11PIN,INPUT);    //set pin to input
```

```
for(i=0;i<MAX_TIME;i++)
```

```
{
```

```
    counter=0;
```

```
    while(digitalRead(DHT11PIN)==1ststate){    //read pin state to see if dht responded. if dht always high for 255 + 1 times, break this while circle
```

```
        counter++;
```

```
        delayMicroseconds(1);
```

```
        if(counter==255)
```

```
            break;
```

```
}
```



6、软件编写

```
Iststate=digitalRead(DHT11PIN);           //read current state and store as last state.  
if(counter==255)                          //if dht always high for 255 + 1 times, break this for circle  
    break;  
// top 3 transistions are ignored, maybe aim to wait for dht finish response signal
```

if((i>=4)&&(i%2==0)){//前3次分别是：1低电平，2高电平（即响应信号），3低电平（即数据第一个低电平），i%2==0 是因为每次都是循环读取低电平和高电平，每次要循环2次才读出一个bit处理

dht11_val[j/8]<<=1; //读到后，j/8可以限制一个数的8个位，左移1位自动补0，相当于读出0

if(counter>16) //counter计数如果超过16，则高电平长，应读1

dht11_val[j/8]|=1; //故再将上面数与1位或，使最后一位变成1

j++; //j++8个换成下一个数据

}

}

// verify checksum and print the verified data

6、软件编写

- if((j>=40)&&(dht11_val[4]==((dht11_val[0]+dht11_val[1]+dht11_val[2]+dht11_val[3])&0xFF))){ ////这其中(dht11_val[0]+dht11_val[1]+dht11_val[2]+dht11_val[3])&0xFF)是将5个数相加，和1与。目的是防止读出数据都为0，和为0，0和1与后得0，所以if判断条件不成立，返回读取失败码。如果读出数据是不为0的正常数据，和1与后还得原数。

```
printf("RH:%d,TEMP:%d\n",dht11_val[0],dht11_val[2]);
```

```
return 1;
```

```
}
```

```
else
```

```
return 0;
```

```
}
```

```
int main(void){
```

```
int attempts=ATTEMPTS;
```

```
if(wiringPiSetup()==-1)
```

```
exit(1);
```

```
while(attempts){
```

```
//you have 5 times to retry
```

```
int success = dht11_read_val(); //get result including printing out
```

```
if (success) {
```

```
//if get result, quit program; if not, retry 5 times then quit
```

```
break;
```

```
}
```

专注IT教育二十年

6、软件编写

```
    attempts--;  
    delay(2500);  
}  
return 0;  
}
```

上述程序保存为.c文件后编译成可执行文件，运行后会在屏幕打印温度和湿度。

7、软件编写

- 使用pi4j编写java程序：
- 在使用pi4j编写DHT11要求的时序时，始终收不到DHT11回应信号，个人感觉是java达不到DHT11要求的微秒级时序，所以java程序没有成功。