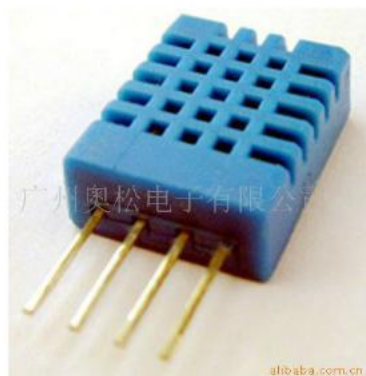


数字温湿度传感器

DHT11

- ▶ 相对湿度和温度测量
- ▶ 全部校准，数字输出
- ▶ 卓越的长期稳定性
- ▶ 无需额外部件
- ▶ 超长的信号传输距离
- ▶ 超低能耗
- ▶ 4 引脚安装
- ▶ 完全互换



DHT11产品概述

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。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。单线制串行接口，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。产品为 4 针单排引脚封装。连接方便，特殊封装形式可根据用户需求而提供。

应用领域

- ▶ 暖通空调
- ▶ 汽车
- ▶ 消费品
- ▶ 气象站
- ▶ 湿度调节器
- ▶ 除湿器
- ▶ 测试及检测设备
- ▶ 数据记录器
- ▶ 自动控制
- ▶ 家电
- ▶ 医疗

订货信息

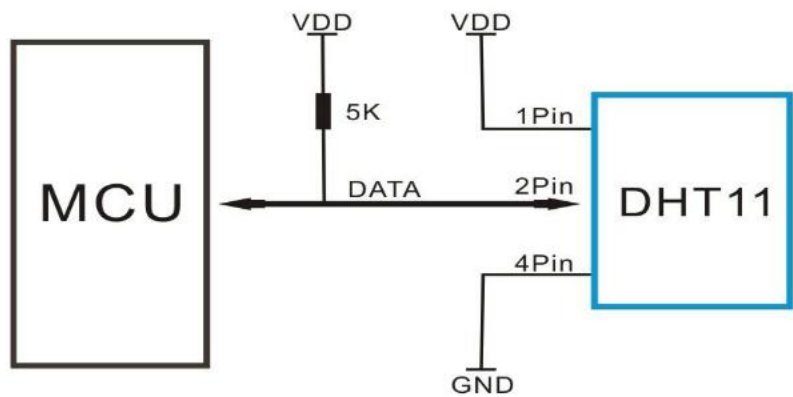
型号	测量范围	测湿精度	测温精度	分辨力	封装
DHT11	20—90%RH 0—50℃	±5%RH	±2℃	1	4 针单排直插

1、传感器性能说明

参数	条件	Min	Typ	Max	单位
湿度					
分辨率		1	1	1	%RH
			8		Bit
重复性			±1		%RH
精度	25℃		±4		%RH
	0—50℃			±5	%RH
互换性	可完全互换				
量程范围	0℃	30		90	%RH
	25℃	20		90	%RH
	50℃	20		80	%RH
响应时间	1/e (63%) 25℃, 1m/s 空气	6	10	15	S
迟滞			±1		%RH
长期稳定性	典型值		±1		%RH/yr
温度					
分辨率		1	1	1	℃
		8	8	8	Bit
重复性			±1		℃
精度		±1		±2	℃
量程范围		0		50	℃
响应时间	1/e (63%)	6		30	S

2、接口说明

建议连接线长度短于20米时用5K上拉电阻, 大于20米时根据实际情况使用合适的上拉电阻



典型应用电路

3、电源引脚

DHT11的供电电压为 3—5.5V。传感器上电后,要等待 1s 以越过不稳定状态在此期间无需发送任何指令。电源引脚 (VDD, GND) 之间可增加一个100nF 的电容,用以去耦滤波。

4、串行接口 (单线双向)

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
+8bit温度整数数据+8bit温度小数数据
+8bit校验和

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+8bit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. 通讯过程如图1所示

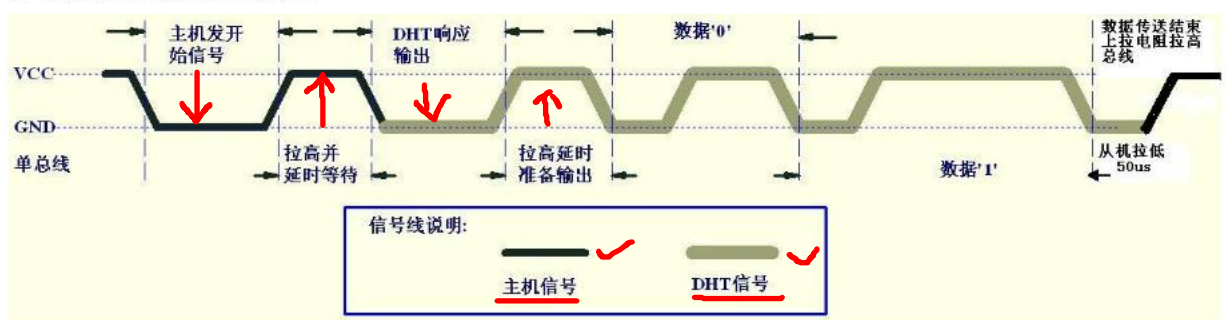


图1

总线空闲状态为高电平,主机把总线拉低等待DHT11响应,主机把总线拉低必须大于18毫秒,保证DHT11能检测到起始信号。DHT11接收到主机的开始信号后,等待主机开始信号结束,然后发送80us低电平响应信号。主机发送开始信号结束后,延时等待20~40us后,读取DHT11的响应信号,主机发送开始信号后,可以切换到输入模式,或者输出高电平均可,总线由上拉电阻拉高。

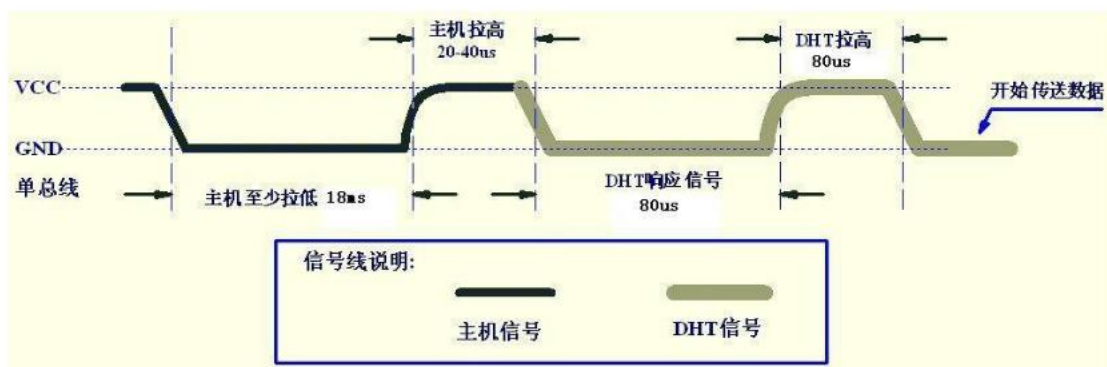


图2

总线为低电平，说明DHT11发送响应信号，DHT11发送响应信号后，再把总线拉高80us，准备发送数据，每一bit数据都以50us低电平时隙开始，高电平的长短定了数据位是0还是1。格式见下面图示。如果读取响应信号为高电平，则DHT11没有响应，请检查线路是否连接正常。当最后一bit数据传送完毕后，DHT11拉低总线50us，随后总线由上拉电阻拉高进入空闲状态。

数字0信号表示方法如图4所示

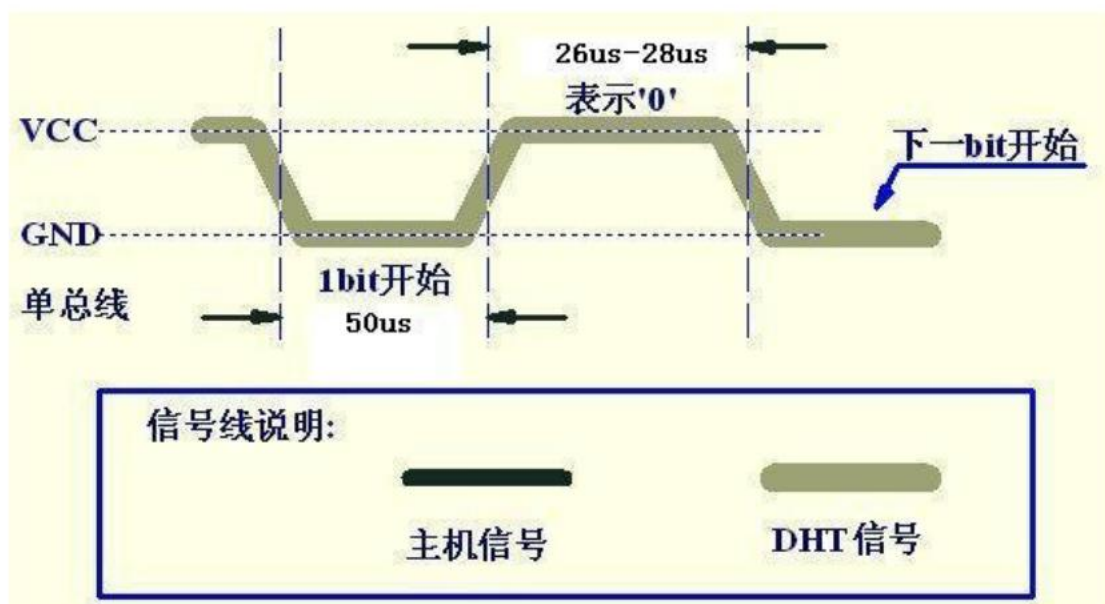


图4

数字1信号表示方法. 如图5所示

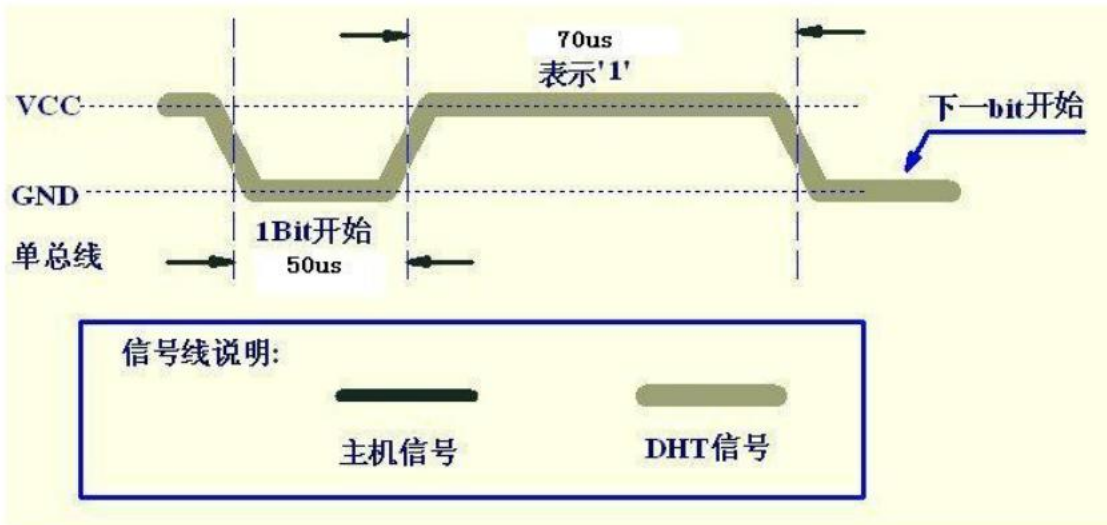


图5

5、 测量分辨率

测量分辨率分别为 8bit（温度）、8bit（湿度）。

6、电气特性

VDD=5V，T = 25℃，除非特殊标注

参数	条件	min	typ	max	单位
供电	DC	3	5	5.5	V
供电电流	测量	0.5		2.5	mA
	平均	0.2		1	mA
	待机	100		150	uA
采样周期	秒	1			次

注: 采样周期间隔不得低于1秒钟。

7、应用信息

7.1工作与贮存条件

超出建议的工作范围可能导致高达3%RH的临时性漂移信号。返回正常工作条后，传感器会缓慢地向校准状态恢复。要加速恢复进程/可参阅7.3小节的“恢复处理”。在非正常工作条件下长时间使用会加速产品的老化过程。

7.2暴露在化学物质中

电阻式湿度传感器的感应层会受到化学蒸汽的干扰，化学物质在感应层中的扩散可能导致测量值漂移和灵敏度下降。在一个纯净的环境中，污染物质会缓慢地释放出去。下文所述的恢复处理将加速实现这一过程。高浓度的化学污染会导致传感器感应层的彻底损坏。

7.3恢复处理

置于极限工作条件下或化学蒸汽中的传感器，通过如下处理程序，可使其恢复到校准时的状态。在50-60℃和< 10%RH的湿度条件下保持2 小时（烘

干)；随后在20-30℃和>70%RH的湿度条件下保持 5小时以上。

7.4温度影响

气体的相对湿度，在很大程度上依赖于温度。因此在测量湿度时，应尽可能保证湿度传感器在同一温度下工作。如果与释放热量的电子元件共用一个印刷线路板，在安装时应尽可能将DHT11远离电子元件，并安装在热源下方，同时保持外壳的良好通风。为降低热传导，DHT11与印刷电路板其它部分的铜镀层应尽可能最小，并在两者之间留出一道缝隙。

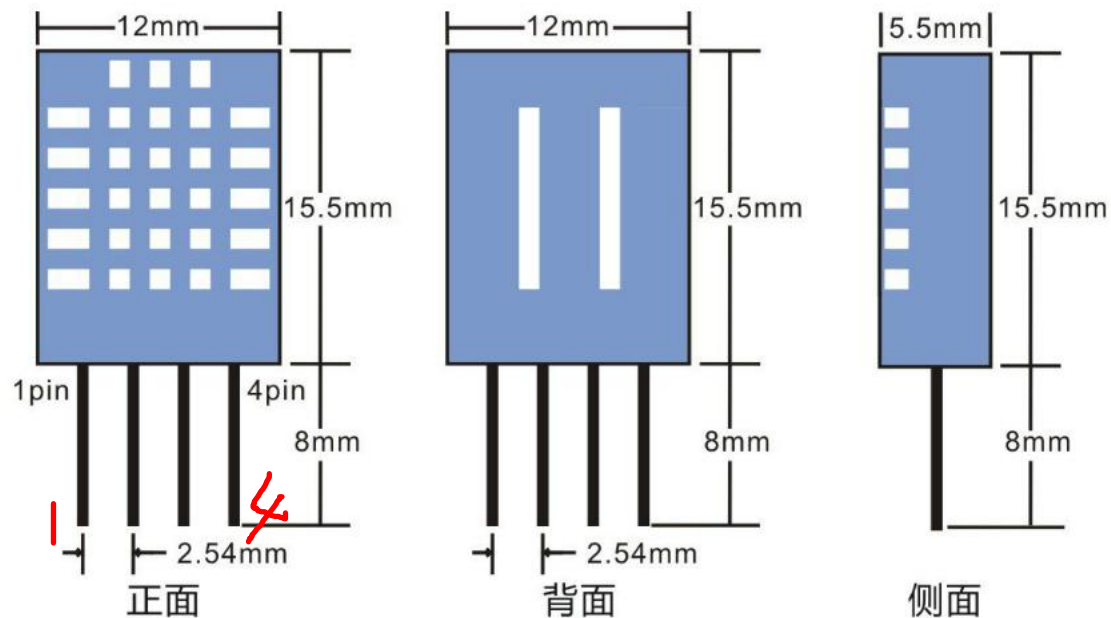
7.5光线

长时间暴露在太阳光下或强烈的紫外线辐射中，会使性能降低。

7.6配线注意事项

DATA信号线材质量会影响通讯距离和通讯质量, 推荐使用高质量屏蔽线。

8、封装信息



9、DHT11引脚说明

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

10、焊接信息

手动焊接，在最高260℃的温度条件下接触时间须少于10秒。

11、注意事项

- ## 12、例程

<http://wenku.baidu.com/view/1ab53fb465ce0508763213ca.html>

下面的只是DHT11模块的.h和.c文件,带详细注释

—

```

/*****
/*****
/*****
//文件名是 dht11.c
#include "dht11.h"
#include <reg52.h>

//请根据自己的 dht11 接的 IO 口来改动位定义
sbit dht11 = P2^0;

//防止在与硬件通信时发生死循环的计数范围

```

```

#define NUMBER 20

#define SIZE 5
static unsigned char status;
//存放五字节数据的数组
static unsigned char value_array[SIZE];
/*可在其他的文件引用温湿度值,实际是温度的整数的 10 倍
如 dht11 读回的温度是 26,则 temp_value = 260, 湿度同理*/
int temp_value, humi_value;

static unsigned char ReadValue(void);

extern void Delay_1ms(unsigned int ms)
{
    unsigned int x, y;
    for(x = ms; x > 0; x--)
    {
        for(y = 124; y > 0; y--);
    }
}

static void DHT11_Delay_10us(void)
{
    unsigned char i;
    i--;
    i--;
    i--;
    i--;
    i--;
    i--;
}

/*读一个字节的的数据*/
static unsigned char DHT11_ReadValue(void)
{
    unsigned char count, value = 0, i;
    status = OK;    //设定标志为正常状态
    for(i = 8; i > 0; i--)
    {
        //高位在先
        value <<= 1;
        count = 0;
        //每一位数据前会有一个 50us 的低电平时间.等待 50us 低电平结束

```



```

while(dht11 == 0 && count++ < NUMBER);
if(count >= NUMBER)
{
    status = ERROR;    //设定错误标志
    return 0;          //函数执行过程发生错误就退出函数
}
//26-28us 的高电平表示该位是 0,为 70us 高电平表该位 1
DHT11_Delay_10us();
DHT11_Delay_10us();
DHT11_Delay_10us();
//延时 30us 后检测数据线是否还是高电平
if(dht11 != 0)
{
    //进入这里表示该位是 1
    value++;
    //等待剩余(约 40us)的高电平结束
    while(dht11 != 0 && count++ < NUMBER)
    {
        dht11 = 1;
    }
    if(count >= NUMBER)
    {
        status = ERROR;    //设定错误标志
        return 0;
    }
}
}
return (value);
}

```

//读温度和湿度函数，读一次的数据,共五字节，读出成功函数返回 OK，错误返回 ERROR

```
extern unsigned char DHT11_ReadTempAndHumi(void)
```

```

{
    unsigned char i = 0, check_value = 0, count = 0;
    EA = 0;
    dht11 = 0;        //拉低数据线大于 18ms 发送开始信号
    Delay_1ms(20);    //需大于 18 毫秒
    dht11 = 1;        //释放数据线,用于检测低电平的应答信号
    //延时 20-40us,等待一段时间后检测应答信号,应答信号是从机拉低数据线 80us
    DHT11_Delay_10us();
    DHT11_Delay_10us();
    DHT11_Delay_10us();
    DHT11_Delay_10us();

```

```

if(dht11 != 0)    //检测应答信号,应答信号是低电平
{
    //没应答信号
    EA = 1;
    return ERROR;
}
else
{
    //有应答信号
    while(dht11 == 0 && count++ < NUMBER);    //等待应答信号结束
    if(count >= NUMBER)    //检测计数器是否超过了设定的范围
    {
        dht11 = 1;
        EA = 1;
        return ERROR;    //读数据出错,退出函数
    }
    count = 0;
    dht11 = 1; //释放数据线
    //应答信号后会有 一个 80us 的高电平, 等待高电平结束
    while(dht11 != 0 && count++ < NUMBER);
    if(count >= NUMBER)
    {
        dht11 = 1;
        EA = 1;
        return ERROR; //退出函数
    }

    //读出湿.温度值
    for(i = 0; i < SIZE; i++)
    {
        value_array[i] = DHT11_ReadValue();
        if(status == ERROR) //调用 ReadValue() 读数据出错会设定 status 为 ERROR
        {
            dht11 = 1;
            EA = 1;
            return ERROR;
        }
        //读出的最后一个值是校验值不需加上去
        if(i != SIZE - 1)
        {
            //读出的五字节数据中的前四字节数据和等于第五字节数据表示成功
            check_value += value_array[i];
        }
    }
} //end for

```

```

//在没用发生函数调用失败时进行校验
if(check_value == value_array[SIZE - 1])
{
    //将温湿度扩大 10 倍方便分离出每一位
    humi_value = value_array[0] * 10;
    temp_value = value_array[2] * 10;
    dht11 = 1;
    EA = 1;
    return OK;    //正确的读出 dht11 输出的数据
}
else
{
    //校验数据出错
    EA = 1;
    return ERROR;
}
}
}

/*****
/*****
/*****

```