

# 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

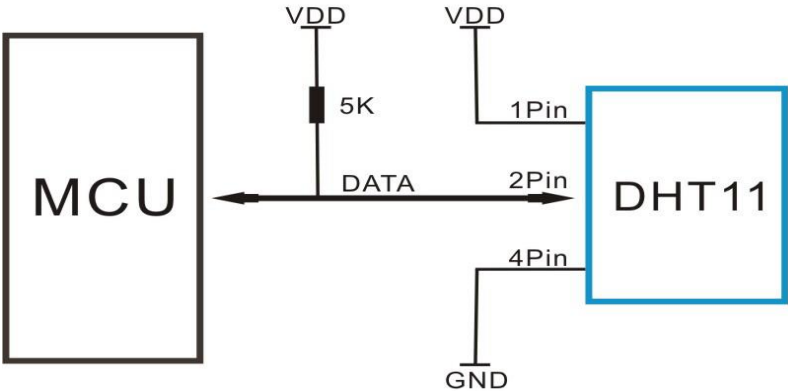
型号	测量范围	测湿精度	测温精度	分辨力	封装
DHT11	20—90%RH 0—50℃	±5%RH	±2℃	1	4 针单排直插

1、传感器性能说明

参数	条件	Min	Typ	Max	单位
<b>湿度</b>					
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
<b>温度</b>					
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

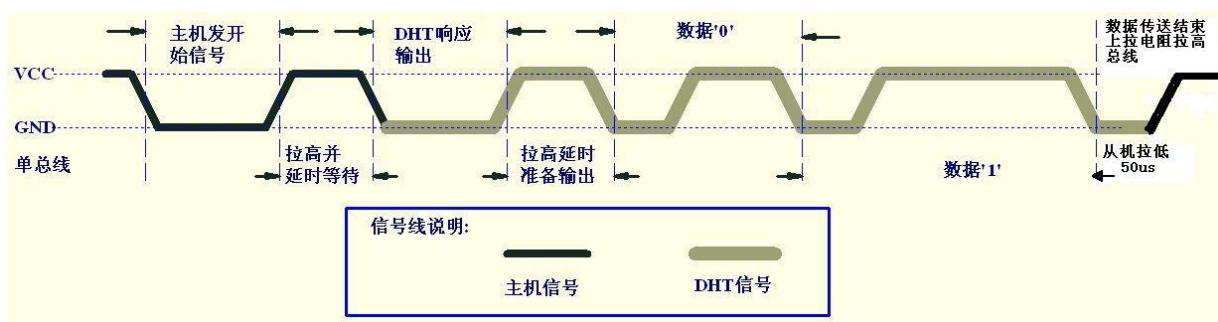


图1

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。

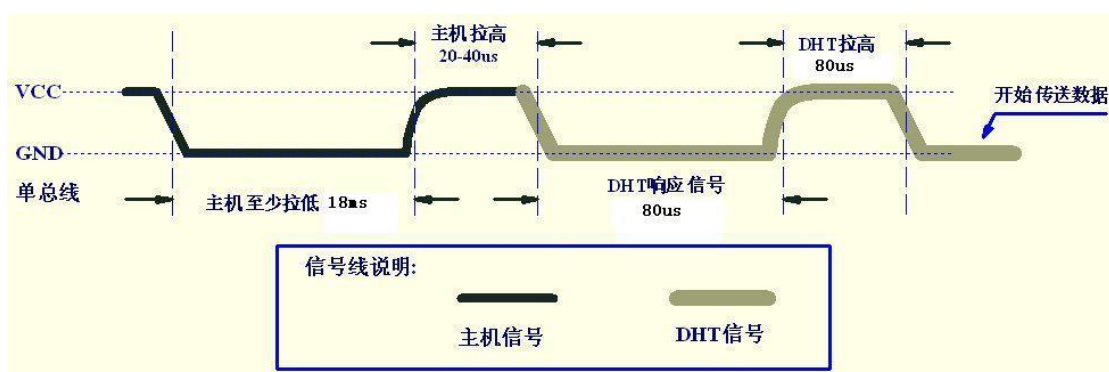


图2

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

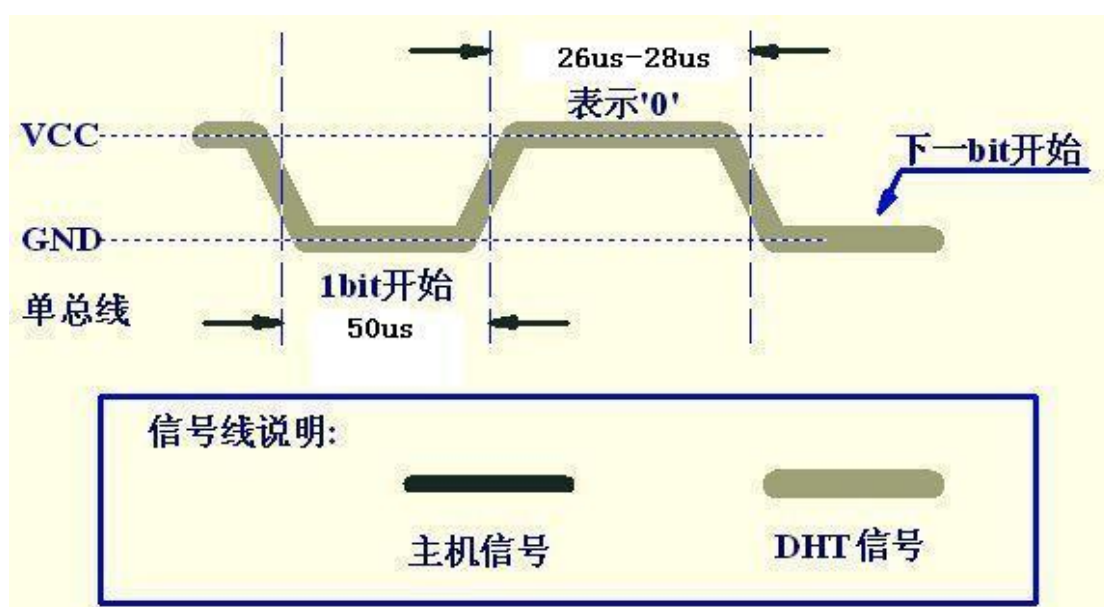


图4

数字1信号表示方法. 如图5所示

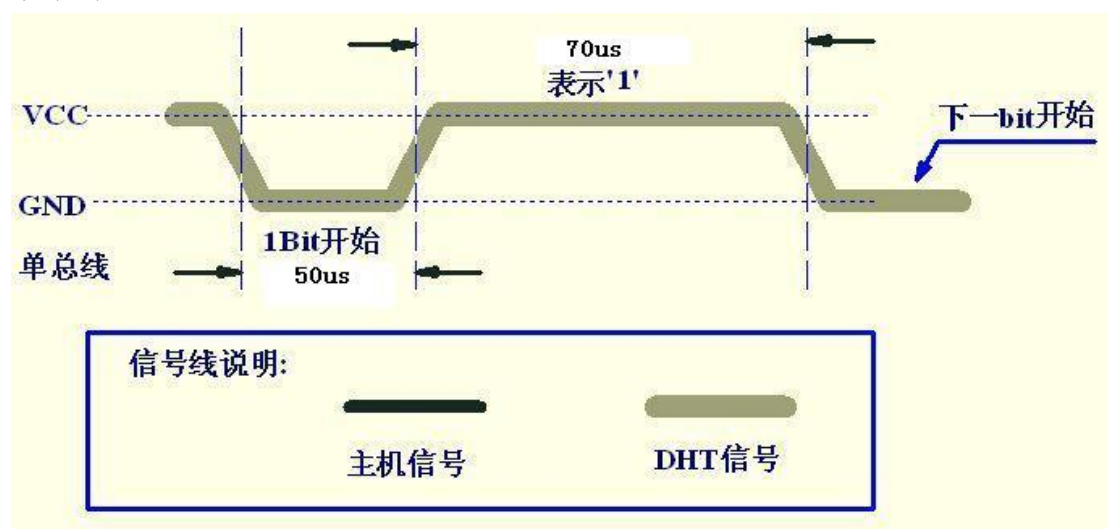


图5

## 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。

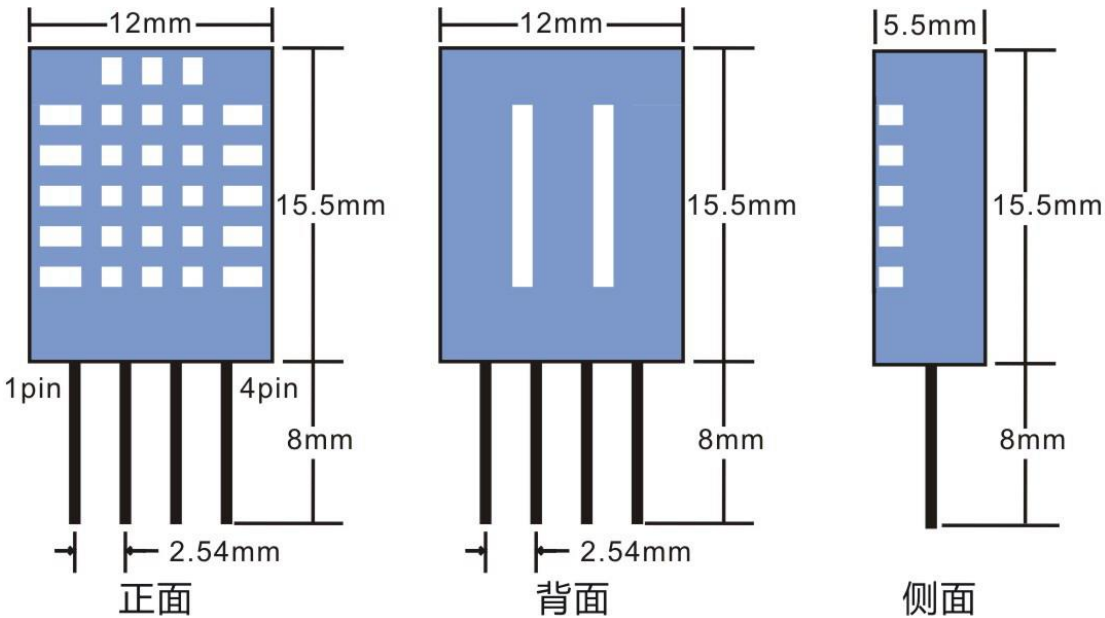
### 7.5 光线

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 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

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

- (1) 避免结露情况下使用。
- (2) 长期保存条件： 温度10—40℃， 湿度60%以下。

//硬件连接： P2.0 口为通讯口连接 DHT1

#include <reg51.h>

#include <intrins.h>

```
typedef unsigned char  U8;          /* defined for unsigned 8-bits integer
variable    无符号 8 位整型变量 */
typedef signed   char  S8;          /* defined for signed 8-bits integer variable
有符号 8 位整型变量 */
typedef unsigned int   U16;         /* defined for unsigned 16-bits integer
variable    无符号 16 位整型变量 */
```

#define uchar unsigned char

#define uint unsigned int

sbit P2\_0 = P2^0 ;

sbit P2\_1 = P2^1 ;

//-----//

//-----定义区-----//

//-----//

U8 U8FLAG;

U8 U8count,U8temp;

U8 U8T\_data\_H,U8T\_data\_L,U8RH\_data\_H,U8RH\_data\_L,U8checkdata;

U8

U8T\_data\_H\_temp,U8T\_data\_L\_temp,U8RH\_data\_H\_temp,U8RH\_data\_L\_te  
mp,U8checkdata\_temp;

U8 U8comdata;

U8 indata[5];

U8 count, count\_r=0;

U16 U16temp1,U16temp2;

sbit d1=P2^4;

sbit d2=P2^5;

sbit d3=P2^6;

sbit d4=P2^7;

uchar code LEDData[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,

```

                                0x80,0x90,0xff};
uchar data  display[5] =  {0x00,0x00,0x00,0x00,0x00};

```

```

void Delay(U16 j)

```

```

{
    U8 i;
    for(;j>0;j--)
    {
        for(i=0;i<27;i++);
    }
}

```

```

void delay(unsigned int num)//延时函数

```

```

{
    while( --num );
}

```

```

void Delay_10us(void)

```

```

{
    U8 i;
    i--;
    i--;
    i--;
    i--;
    i--;
    i--;
}

```

```

//串行总线

```

```

void COM(void)

```

```

{

    U8 i;
    for(i=0;i<8;i++)
    {

```

```

        U8FLAG=2;
        //-----
        P2_1=0 ; //T
        P2_1=1 ; //T
        //-----

```

```

        while((!P2_0)&&U8FLAG++);
        Delay_10us();
        Delay_10us();

```



```

// Delay_10us();
U8temp=0;
if(P2_0)U8temp=1;
U8FLAG=2;
while((P2_0)&&U8FLAG++);

//-----
P2_1=0 ; //T
P2_1=1 ; //T
//-----
//超时则跳出 for 循环
if(U8FLAG==1)break;
//判断数据位是 0 还是 1

// 如果高电平高过预定 0 高电平值则数据位为 1

U8comdata<=1;
U8comdata|=U8temp;          //0
} //rof
}

//-----
//-----湿度读取子程序 -----
//-----
//----以下变量均为全局变量-----
//----温度高 8 位== U8T_data_H-----
//----温度低 8 位== U8T_data_L-----
//----湿度高 8 位== U8RH_data_H-----
//----湿度低 8 位== U8RH_data_L-----
//----校验 8 位 == U8checkdata-----
//----调用相关子程序如下-----
//---- Delay();, Delay_10us();,COM();
//-----

void RH(void)
{
    //主机拉低 18ms
    P2_0=0;
    Delay(180);
    P2_0=1;
    //总线由上拉电阻拉高 主机延时 20us

```

```

    Delay_10us();
    Delay_10us();
    Delay_10us();
    Delay_10us();
//主机设为输入 判断从机响应信号
    P2_0=1;
//判断从机是否有低电平响应信号 如不响应则跳出，响应则向下运行
    if(!P2_0)    //T !
    {
        U8FLAG=2;////////////////////////////////////////
//判断从机是否发出 80us 的低电平响应信号是否结束
        while(!P2_0)&&U8FLAG++);
        U8FLAG=2;
//判断从机是否发出 80us 的高电平，如发出则进入数据接收状态
        while(P2_0)&&U8FLAG++);
//数据接收状态
        COM();
        U8RH_data_H_temp=U8comdata;
        COM();
        U8RH_data_L_temp=U8comdata;
        COM();
        U8T_data_H_temp=U8comdata;
        COM();
        U8T_data_L_temp=U8comdata;
        COM();
        U8checkdata_temp=U8comdata;
        P2_0=1;
//数据校验

U8temp=(U8T_data_H_temp+U8T_data_L_temp+U8RH_data_H_temp+U8R
H_data_L_temp);
    if(U8temp==U8checkdata_temp)
    {
        U8RH_data_H=U8RH_data_H_temp;
        U8RH_data_L=U8RH_data_L_temp;
        U8T_data_H=U8T_data_H_temp;
        U8T_data_L=U8T_data_L_temp;
        U8checkdata=U8checkdata_temp;
    }//fi
} //fi

```

```

}

Disp_Temperature()//显示温度
{
    unsigned char  n=0;

    // display[4]=temp_data[0]&0x0f;
    // display[0]=ditab[display[4]];      //查表得小数位的值

    // display[4]=((temp_data[0]&0xf0)>>4)|((temp_data[1]&0x0f)<<4);

    display[4]=U8RH_data_H;
    display[4]=display[4]%100;
    display[3]=display[4]/10;
    display[2]=display[1]%10;

    display[4]=U8T_data_H;
    display[4]=display[4]%100;
    display[1]=display[4]/10;
    display[0]=display[4]%10;

    if(!display[3])      //高位为 0，不显示
    {
        display[3]=0x0a;
        if(!display[1])  //次高位为 0，不显示
            display[1]=0x0a;
    }

    d1=1;
    P0=LEDData[display[0]];
    delay(5);d1=0;

    d2=1;
    P0=LEDData[display[1]];
    delay(5);d2=0;

    d3=1;
    P0=LEDData[display[2]];
    delay(5);d3=0;

    d4=1;
    P0=LEDData[display[3]];

```

```
    delay(5);d4=0;
}

void main(void)
{
    while(1)
    {
        RH();
        Disp_Temperature();
        //读取模块数据周期不易小于 2S
        Delay(20000);
    }
}
```