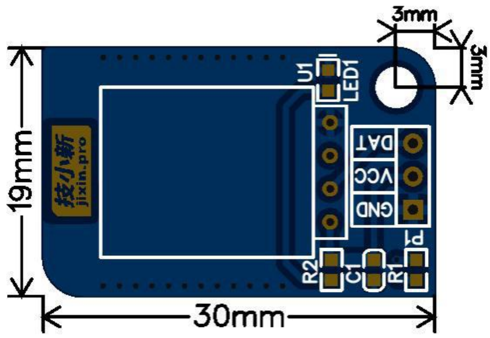
DHT11 温湿度传感器模块学习手册（51 单片机版）

1、模块介绍

1.1 模块特点

DHT11 温湿度传感器模块是以 DHT11 温湿度传感器为核心，已含有校准数字信号，可测量环境温度，相对湿度。具有极高的可靠性与卓越的长期稳定性。模块带有电源指示灯。主要特点如下：

- 测量相对湿度范围：5~95%RH
- 测量温度范围：-20~60°C
- 支持 3.3V~5.5V 工作电压
- 带有电源指示灯



DHT11 温湿度传感器模块

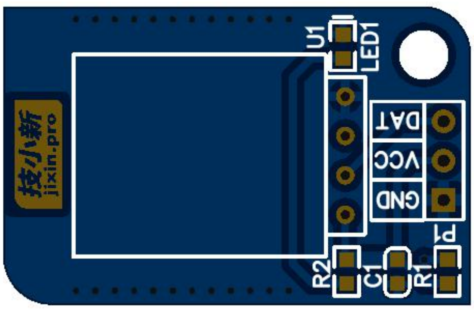
1.2 模块接口引脚

Symbol （符号）	Type （类型）	Description （描述）
GND	电源	接地引脚
VCC	电源	电源输入引脚
DAT	IO	数据输出引脚

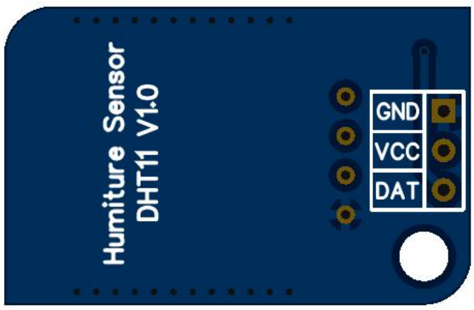
模块引脚接口功能表

1.3 模块丝印

模块上的每一个器件都有一个相对应的丝印在上面，可以通过 BOM 表中的器件编号在板子找到器件的位置。



模块正面丝印图



模块背面丝印图

2、模块用途

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，确保产品具有枀高的可靠性与卓越的长期稳定性。传感器包括一个电容式感湿元件和一个 NTC 测温元件，并与一个高性能 8 位单片机相连接。

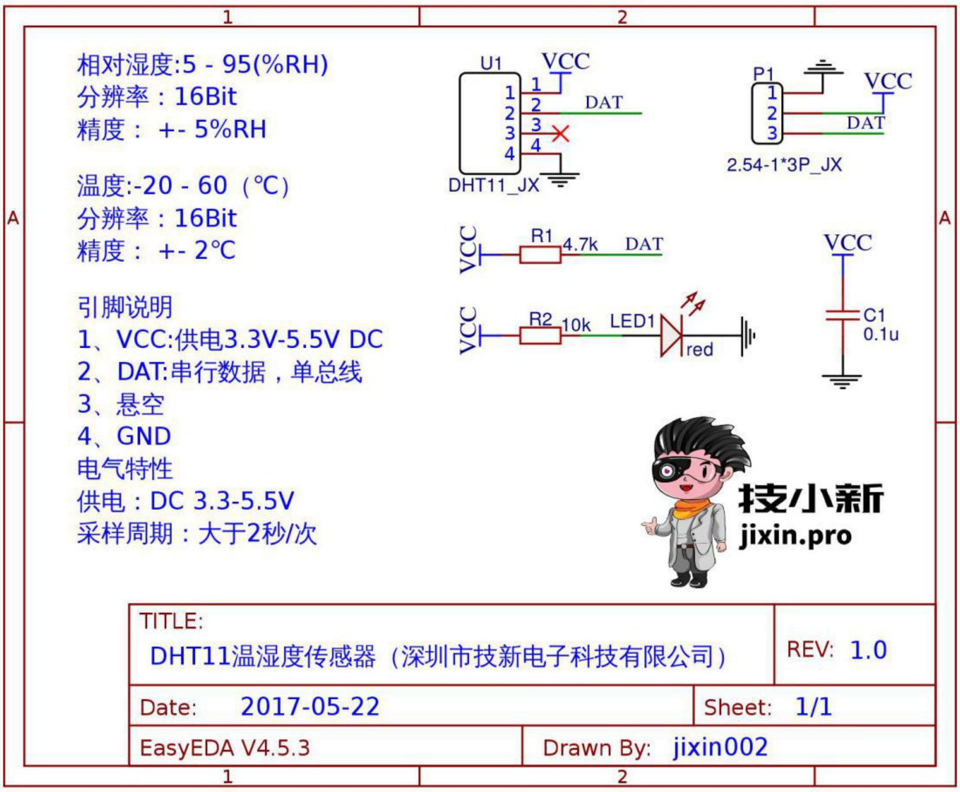
根据 DHT11 的特点，DHT11 模块可以应用在暖通空调、除湿器、农业、冷链仓储、测试及检测设备、消费品、汽车、自动控制、数据记录器、气象站、家电、湿度调节器、医疗、其他相关湿度检测控制 等方面。

3、硬件设计

硬件设计内容比较少，主要按照 DHT11 的官方手册的典型应用部分进行设计，DHT11 的官方手册可到技新网 https://www.jixin.pro/product/1589.html 下的教程与资料中下载。在设计中有以下 4 点需要注意的地方：

1) 典型应用电路中建议连接线长度短于 5m 时用 4.7K 上拉电阻，大于 5m 时根据实际情况降低上拉电阻的阻值。
2) 使用 3.3V 电压供电时连接线尽量短，接线过长会导致传感器供电不足，造成测量偏差。
3) 每次读出的温湿度数值是上一次测量的结果，欲获取实时数据,需连续读取 2 次，但不建议连续多次读取传感器，每次读取传感器间隔大于 2 秒即可获得准确的数据 。
4) 电源部分如有波动，会影响到温度。如使用开关电源纹波过大，温度会出现跳动。

3.1 模块原理图



模块原理图

4、软件设计

DHT11 传感器采用简化的单总线通信。单总线即只有一根数据线，系统中的数据交换、控制均由单总线完成。

软件设计采用 IAP15W4K58S4 单片机与 DHT11 温湿度传感器模块进行通讯，单片机与模块的引脚连接：GND --> GND、5V --> VCC、P11 --> DAT。

4.1DHT11 的通讯与数据说明

DHT11 采用单总线通讯协议，与单片机（主机）的通讯的过程：主机发送起始信号--->DHT11 检测到并发送响应信号--->DHT11 发送 40 位数据（高位先出）--->DHT11 发送结束信号。

名称	单总线线格式定义
起始信号	主机把数据总线拉低 T(18ms < T < 30ms)，通知传感器准备数据
响应信号	DHT11 把数据总线拉低 83us 以响应主机的起始信号，再拉高 87us 通知主机准备接收数据
数据	DHT11 在拉高总线 87us 后紧接着发送 40Bit 数据（高位先出，后边会介绍数据格式与时序）
结束信号	DHT11 在发送完 40Bit 数据后接着输出 54us 低电平作为结束信号，然后转为输入状态

接受到的 40 位数据为：湿度高 8 位，湿度低 8 位，温度高 8 位，温度低 8 位，8 位校验位。湿度的高 8 位是测量湿度的整数，湿度的低 8 位是测量湿度的小数（湿度的小数部分为 0）；温度的高 8 位是测量温度的整数，温度的低 8 位是测量温度的小数；8 位校验和 = 湿度高 8 位 + 湿度低 8 位 + 温度高 8 位 + 温度低 8 位，可参考下图：

示例一：接收到的 40 位数据为：

0011 0101	0000 0000	0001 1000	0000 0100	0101 0001
湿度高 8 位	湿度低 8 位	温度高 8 位	温度低 8 位	校验位

计算：

0011 0101+0000 0000+0001 1000+0000 0100= 0101 0001

接收数据正确：

湿度：0011 0101(整数)=35H=53%RH 0000 0000(小数)=00H=0.0%RH =>53%RH + 0.0%RH = 53.0%RH

温度：0001 1000(整数)=18H=24℃ 0000 0100(小数)=04H=0.4℃ =>24℃ + 0.4℃ = 24.4℃

◎**特殊说明：**

当温度低于 0 ℃ 时温度数据的低 8 位的最高位置为 1。

示例： -10.1 ℃ 表示为 0000 1010 1000 0001

温度： 0000 1010(整数)=0AH=10℃，0000 0001(小数)=01H=0.1℃ =>-(10℃+0.1℃)= -10.1℃

示例二：接收到的 40 位数据为：

0011 0101	0000 0000	0001 1000	0000 0100	0100 1001
湿度高 8 位	湿度低 8 位	温度高 8 位	温度低 8 位	校验位

计算：

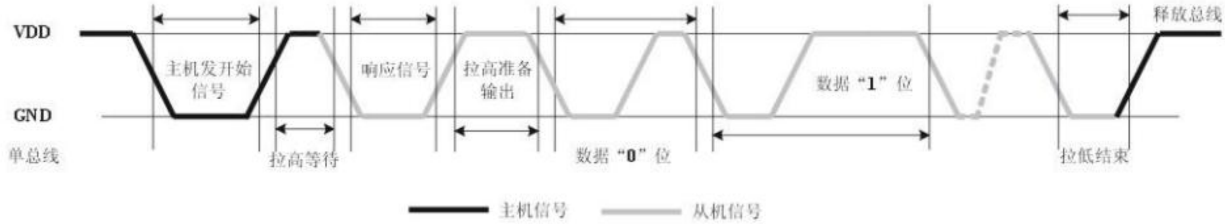
0011 0101+0000 0000+0001 1000+0000 0100= 0101 0001

0101 0001 不等于 0100 1001

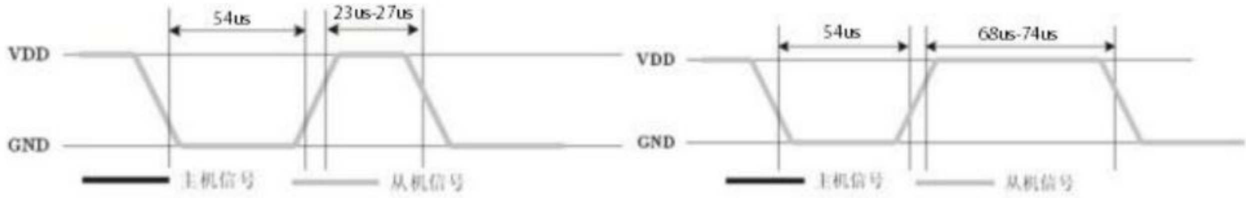
本次接收的数据不正确，放弃，重新接收数据。

4.2 DHT11 温湿度传感器模块的通讯解析（51 单片机）

（一）看一下单片机和 DHT11 的通信协议发现单片机只需要发送一个起始信号即可，剩下的事情就是接收信号，然后处理信号。单片机发送信号后 DHT11 将会返回一个 83uS 的低电平和一个 87uS 的高电平。



（二）单片机接收的信号是一连串的脉冲，不同的脉宽组合表示 0 或者 1。一共 40Bit 的有效数据，那么 DHT11 将会发送 80 个高电平和低电平。



位数据“0”格式　　位数据“1”格式

（三）DHT11 在发送完所有数据后，还会发送一个结束信号，告诉单片机已经发送完了，这是一个 54uS 的低电平。

（四）分析上述通信协议，发现使用单片机的“脉宽捕获”功能非常适合这里。“脉宽捕获”功能，可以自动记录单片机指定引脚上的高电平持续时间和低电平持续时间。如果我们能够记录下来这一连串的时间，然后就可以分析这一串时间，解析出 DHT11 的数据。

4.3 DHT11 温湿度传感器模块的通讯实现（51 单片机）

1) 单片机发送一个 25mS 的低电平，发送完毕后再拉高，表示起始信号。
2) 单片机打开“脉宽捕获”功能，开始记录高低电平的数据。每来一个脉冲，不管是高电平还是低电平，都会把它们的脉宽数据存储到数组里面。
3) 单片机软件延时 7mS，等待 DHT11 发送完毕所有数据。（DHT11 协议告诉我们，它发送数据的时间最长的可能性是：5344uS。
4) 单片机开始解析接收到的数据。

这里主要说明一下解析数据的方法：

A. 首先要找到 DHT11 发送的起始信号，其实是一个 83uS 的低电平和 87uS 的高电平，进入到我们的程序里，就是两个 80uS 以上的脉宽，处理的时候可以判断只要有两个连续大于 80uS 的脉宽，有就算一个正确的起始信号。（因为测量都存在误差，而且通信协议中，只有起始信号是两个连续大于 78uS 的，所以可以判断大于 78uS 也可以）
B. 如果连起始信号都没匹配的，那么通信可能失败了，所以就放弃后面的工作，再来一次吧。如果匹配到了起始信号，那么开始进行解码。每两个脉宽时间为一组，依次进行判断，如果第一个时间大于第二个时间，那么接收的是数据 0，否则就是数据 1，这样连续判断 40 组数据，就算处理完 40Bit 数据了。
C. 接收的数据一共是 5 个字节数据，最后一个字节是校验码，如果校验码也是对的，那么认为这组数据可靠的，恭喜你，完成了一次可靠的解析。

下图代码是查找数组中的起始信号：

```
  //根据DHT11协议，检测一个83uS的应答和一个87uS的开始信号，下一个信号就是数据
  for(i=0;i<90;i++)
  {
    if((DHT11_Buffer[i]>78)&&(DHT11_Buffer[i+1]>78))
    {
      j=i+2;
      i = 100;//用于退出这个循环
    }
  }
  if(i < 80)//表示没有检测到起始信号，通信失败
    return 1;
  i = j;
  j = 0;
  Data->Humi_H = 0;Data->Humi_L = 0;
  Data->Temp_H = 0;Data->Temp_L = 0;
  Data->Check_Sum = 0;
```

下图代码是解析查找到的有效信息：

```
  for(i;i<90;i+=2)//从有效数据开始解码
  {
    if(j < 8)
    {
      Data->Humi_H <<= 1;
      if(DHT11_Buffer[i]<DHT11_Buffer[i+1])//根据DHT11协议，这表示第1个数据
        Data->Humi_H++;
    }
    if((j > 7) && (j < 16))
    {
      Data->Humi_L <<= 1;
      if(DHT11_Buffer[i]<DHT11_Buffer[i+1])//根据DHT11协议，这表示第2个数据
        Data->Humi_L++;
    }
    if((j > 15) && (j < 24))
    {
      Data->Temp_H <<= 1;
      if(DHT11_Buffer[i]<DHT11_Buffer[i+1])//根据DHT11协议，这表示第3个数据
        Data->Temp_H++;
    }
    if((j > 23) && (j < 32))
    {
      Data->Temp_L <<= 1;
      if(DHT11_Buffer[i]<DHT11_Buffer[i+1])//根据DHT11协议，这表示第4个数据
        Data->Temp_L++;
    }
    if((j > 31) && (j < 40))
    {
      Data->Check_Sum <<= 1;
      if(DHT11_Buffer[i]<DHT11_Buffer[i+1])//根据DHT11协议，这表示第5个数据
        Data->Check_Sum++;
    }
    j++;
  }
  //进行校验，如果出错返回2
  if((Data->Humi_H + Data->Humi_L + Data->Temp_H + Data->Temp_L) == Data->Check_Sum)
    return 0;//成功读取数据
  else
    return 2;
}
```