

第三十六章 DHT11 数字温湿度传感器实验

上一章，我们介绍了数字温度传感器 DS18B20 的使用，本章我们将介绍数字温湿度传感器 DHT11 的使用，该传感器不但能测温度，还能测湿度。本章我们将向大家介绍如何使用 STM32 来读取 DHT11 数字温湿度传感器，从而得到环境温度和湿度等信息，并把从温湿度值显示在 TFTLCD 模块上。本章分为如下几个部分：

36.1 DHT11 简介

36.2 硬件设计

36.3 软件设计

36.4 下载验证

36.1 DHT11 简介

DHT11 是一款湿温度一体化的数字传感器。该传感器包括一个电阻式测湿元件和一个 NTC 测温元件，并与一个高性能 8 位单片机相连接。通过单片机等微处理器简单的电路连接就能够实时的采集本地湿度和温度。DHT11 与单片机之间能采用简单的单总线进行通信，仅仅需要一个 I/O 口。传感器内部湿度和温度数据 40Bit 的数据一次性传给单片机，数据采用校验和方式进行校验，有效的保证数据传输的准确性。DHT11 功耗很低，5V 电源电压下，工作平均最大电流 0.5mA。

DHT11 的技术参数如下：

- 工作电压范围：3.3V-5.5V
- 工作电流 ：平均 0.5mA
- 输出：单总线数字信号
- 测量范围：湿度 20~90%RH，温度 0~50℃
- 精度 ：湿度±5%，温度±2℃
- 分辨率 ：湿度 1%，温度 1℃

DHT11 的管脚排列如图 36.1.1 所示：

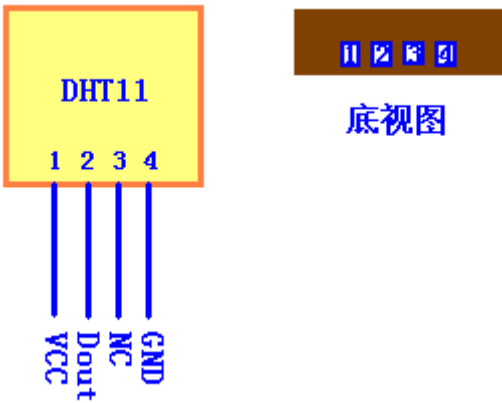


图 36.1.1 DHT11 管脚排列图

虽然 DHT11 与 DS18B20 类似，都是单总线访问，但是 DHT11 的访问，相对 DS18B20 来说要简单很多。下面我们先来看看 DHT11 的数据结构。

DHT11 数字湿温度传感器采用单总线数据格式。即，单个数据引脚端口完成输入输出双向传输。其数据包由 5Byte（40Bit）组成。数据分小数部分和整数部分，一次完整的数据传输为 40bit，高位先出。DHT11 的数据格式为：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。其中校验和数据为前四个字节相加。

传感器数据输出的是未编码的二进制数据。数据(湿度、温度、整数、小数)之间应该分开处理。例如，某次从 DHT11 读到的数据如图 36.1.2 所示：

byte4	byte3	byte2	byte1	byte0
00101101	00000000	00011100	00000000	01001001
整数	小数	整数	小数	校验和
湿度		温度		校验和

图 36.1.2 某次读取到 DHT11 的数据

由以上数据就可得到湿度和温度的值，计算方法：

湿度= byte4 . byte3=45.0 (%RH)
温度= byte2 . byte1=28.0 (℃)

校验= byte4+ byte3+ byte2+ byte1=73(=湿度+温度)(校验正确)

可以看出，DHT11 的数据格式是十分简单的，DHT11 和 MCU 的一次通信最大为 3ms 左右，建议主机连续读取时间间隔不要小于 100ms。

下面，我们介绍一下 DHT11 的传输时序。DHT11 的数据发送流程如图 36.1.3 所示：

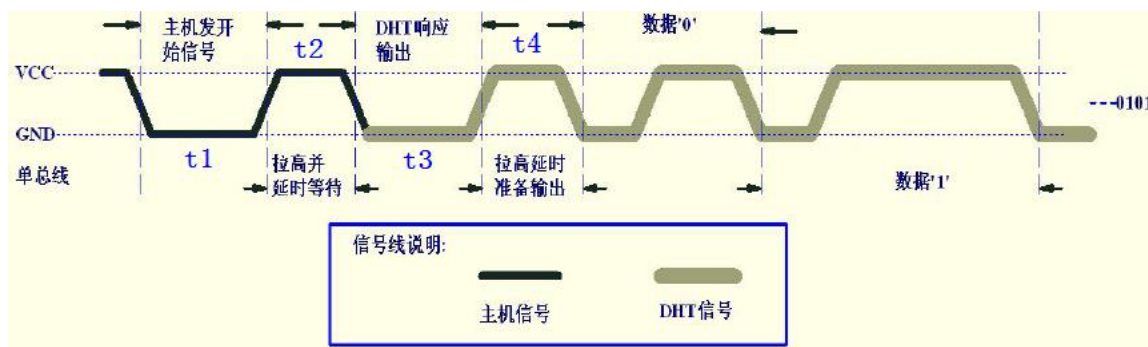


图 36.1.3 DHT11 数据发送流程

首先主机发送开始信号，即：拉低数据线，保持 t1（至少 18ms）时间，然后拉高数据线 t2（20~40us）时间，然后读取 DHT11 的响应，正常的话，DHT11 会拉低数据线，保持 t3（40~50us）时间，作为响应信号，然后 DHT11 拉高数据线，保持 t4（40~50us）时间后，开始输出数据。

DHT11 输出数字 ‘0’ 的时序如图 36.1.4 所示：

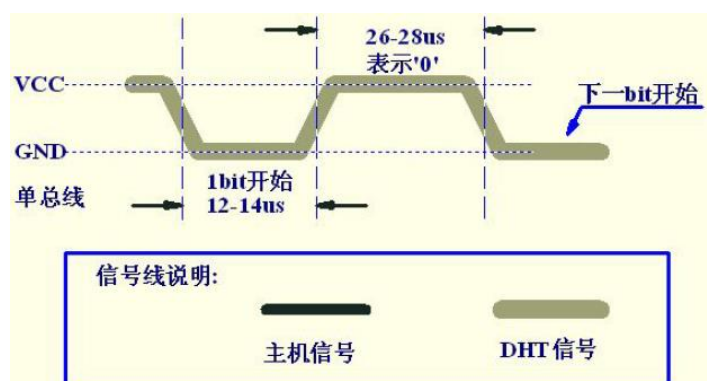


图 36.1.4 DHT11 数字 ‘0’ 时序

DHT11 输出数字 ‘1’ 的时序如图 36.1.5 所示：

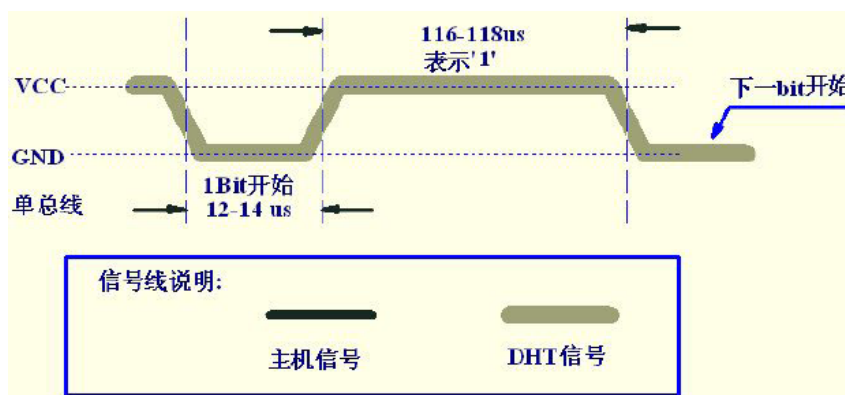


图 36.1.5 DHT11 数字 ‘1’ 时序

通过以上了解，我们就可以通过 STM32 来实现对 DHT11 的读取了。DHT11 的介绍就到这里，更详细的介绍，请参考 DHT11 的数据手册。

36.2 硬件设计

由于开发板上标准配置是没有 DHT11 这个传感器的,只有接口,所以要做本章的实验,大家必须找一个 DHT11 插在预留的 DHT11 接口上。

本章实验功能简介: 开机的时候先检测是否有 DHT11 存在,如果没有,则提示错误。只有在检测到 DHT11 之后才开始读取温湿度值,并显示在 LCD 上,如果发现了 DHT11,则程序每隔 100ms 左右读取一次数据,并把温湿度显示在 LCD 上。同样我们也是用 DS0 来指示程序正在运行。

所要用到的硬件资源如下:

- 1) 指示灯 DS0
- 2) TFTLCD 模块
- 3) DHT11 温湿度传感器

这些我们都已经介绍过了, DHT11 和 DS18B20 的接口是共用一个的,不过 DHT11 有 4 条腿,需要把 U6 的 4 个接口都用上,将 DHT11 传感器插入到这个上面就可以通过 STM32F1 来读取温湿度值了。连接示意图如图 36.2.1 所示:

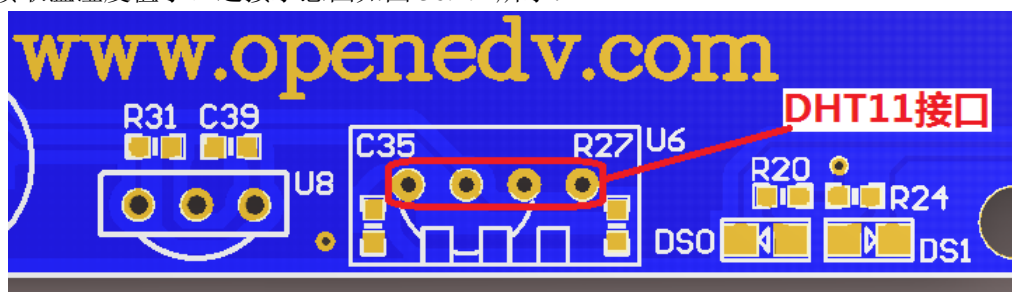


图 36.2.1 DHT11 连接示意图

这里要注意,将 DHT11 贴有字的一面朝内,而有很多孔的一面(网面)朝外,然后然后插入如图所示的四个孔内就可以了。

36.3 软件设计

打开 DHT11 数字温湿度传感器实验工程可以发现,我们在工程中添加了 dht11.c 文件和 dht11.h 文件,所有 DHT11 相关的驱动代码和定义都在这两个文件中。

打开 dht11.c 代码如下:

```
#include "dht11.h"
#include "delay.h"
//复位 DHT11
void DHT11_Rst(void)
{
    DHT11_IO_OUT();    //SET OUTPUT
    DHT11_DQ_OUT=0;    //拉低 DQ
    delay_ms(20);       //拉低至少 18ms
    DHT11_DQ_OUT=1;    //DQ=1
    delay_us(30);       //主机拉高 20~40us
}
//等待 DHT11 的回应
//返回 1:未检测到 DHT11 的存在
//返回 0:存在
```

```

u8 DHT11_Check(void)
{
    u8 retry=0;
    DHT11_IO_IN();//SET INPUT
    while (DHT11_DQ_IN&&retry<100) //DHT11 会拉低 40~80us
    {
        retry++;
        delay_us(1);
    };
    if(retry>=100)return 1;
    else retry=0;
    while (!DHT11_DQ_IN&&retry<100)//DHT11 拉低后会再次拉高 40~80us
    {
        retry++;
        delay_us(1);
    };
    if(retry>=100)return 1;
    return 0;
}
//从 DHT11 读取一个位
//返回值: 1/0
u8 DHT11_Read_Bit(void)
{
    u8 retry=0;
    while(DHT11_DQ_IN&&retry<100)//等待变为低电平
    {
        retry++;
        delay_us(1);
    }
    retry=0;
    while(!DHT11_DQ_IN&&retry<100)//等待变高电平
    {
        retry++;
        delay_us(1);
    }
    delay_us(40);//等待 40us
    if(DHT11_DQ_IN)return 1;
    else return 0;
}
//从 DHT11 读取一个字节
//返回值: 读到的数据
u8 DHT11_Read_Byte(void)
{
    u8 i,dat;

```

```

        dat<<=1;
        dat|=DHT11_Read_Bit();
    }
    return dat;
}
//从 DHT11 读取一次数据
//temp:温度值(范围:0~50° )
//humi:湿度值(范围:20%~90%)
//返回值: 0,正常;1,读取失败
u8 DHT11_Read_Data(u8 *temp,u8 *humi)
{
    u8 buf[5];
    u8 i;
    DHT11_Rst();
    if(DHT11_Check()==0)
    {
        for(i=0;i<5;i++)//读取 40 位数据
        {
            buf[i]=DHT11_Read_Byte();
        }
        if((buf[0]+buf[1]+buf[2]+buf[3])==buf[4])
        {
            *humi=buf[0];
            *temp=buf[2];
        }
    }else return 1;
    return 0;
}
//初始化 DHT11 的 IO 口 DQ 同时检测 DHT11 的存在
//返回 1:不存在
//返回 0:存在
u8 DHT11_Init(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOG, ENABLE); //使能 PG 端口
    时钟

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11; //PG11 端口配置
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP; //推挽输出
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOG, &GPIO_InitStructure); //初始化 IO 口

```

```

GPIO_SetBits(GPIOG,GPIO_Pin_11);           //PG11 输出高

DHT11_Rst();           //复位 DHT11
return DHT11_Check();   //等待 DHT11 的回应
}

```

该部分代码首先是通过函数 DHT11_Init 初始化传感器，然后根据我们前面介绍的单总线操作时序来读取 DHT11 的温湿度值的，DHT11 的温湿度值通过 DHT11_Read_Data 函数读取，如果返回 0，则说明读取成功，返回 1，则说明读取失败。同样我们打开 dht11.h 可以看到，头文件中主要是一些端口配置以及函数申明，代码比较简单。接下来我们打开 main.c，该文件代码如下：

```

int main(void)
{
    u8 t=0;
    u8 temperature;
    u8 humidity;

    delay_init();           //延时函数初始化
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2);//设置中断优先级分组为组 2
    uart_init(115200);       //串口初始化为 115200
    LED_Init();              //初始化与 LED 连接的硬件接口
    LCD_Init();              //初始化 LCD
    POINT_COLOR=RED;         //设置字体为红色
    LCD_ShowString(30,50,200,16,16,"WarShip STM32");
    LCD_ShowString(30,70,200,16,16,"DHT11 TEST");
    LCD_ShowString(30,90,200,16,16,"ATOM@ALIENTEK");
    LCD_ShowString(30,110,200,16,16,"2015/1/16");
    while(DHT11_Init())      //DHT11 初始化
    {
        LCD_ShowString(30,130,200,16,16,"DHT11 Error");
        delay_ms(200);
        LCD_Fill(30,130,239,130+16,WHITE);
        delay_ms(200);
    }
    LCD_ShowString(30,130,200,16,16,"DHT11 OK");
    POINT_COLOR=BLUE;//设置字体为蓝色
    LCD_ShowString(30,150,200,16,16,"Temp:  C");
    LCD_ShowString(30,170,200,16,16,"Humi:  %");
    while(1)
    {
        if(t%10==0)           //每 100ms 读取一次
        {
            DHT11_Read_Data(&temperature,&humidity); //读取温湿度值

            LCD_ShowNum(30+40,150,temperature,2,16); //显示温度

```

```

        LCD_ShowNum(30+40,170,humidity,2,16);    //显示湿度
    }
    delay_ms(10);
    t++;
    if(t==20)
    {
        t=0;
        LED0=!LED0;
    }
}
}

```

主函数比较简单，进行一系列初始化后，如果 DHT11 初始化成功，那么每隔 100ms 读取一次转换数据并显示在液晶上。至此，我们本章的软件设计就结束了。

36.4 下载验证

在代码编译成功之后，我们通过下载代码到 ALIENTEK 战舰 STM32 开发板上，可以看到 LCD 显示开始显示当前的温度值（假定 DHT11 已经接上去了），如图 36.4.1 所示：

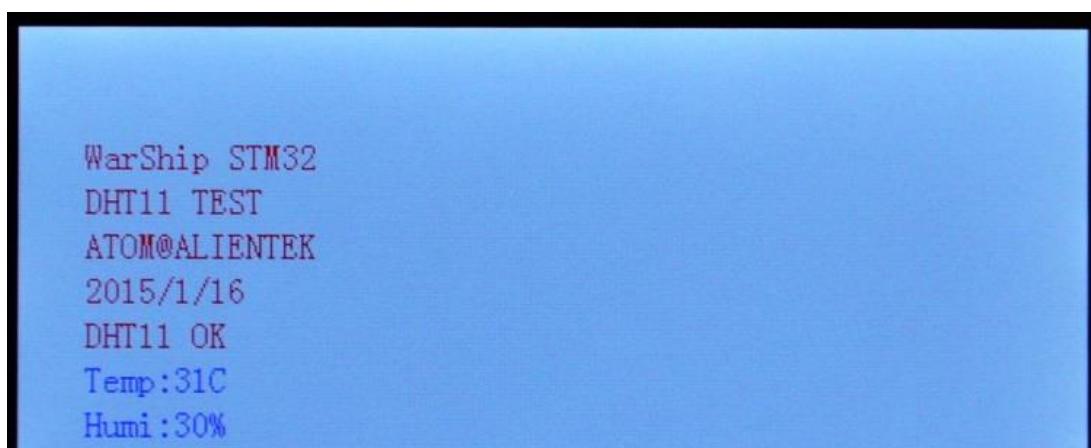


图 36.4.1 DHT11 读取到的温湿度值

至此，本章实验结束。大家可以将本章通过 DHT11 读取到的温度值，和前一章的通过 DS18B20 读取到的温度值对比一下，看看哪个更准确？

广州市星翼电子科技有限公司

淘宝店铺 1: <http://eboard.taobao.com>

淘宝店铺 2: <http://openedv.taobao.com>

技术支持论坛 (开源电子网) : www.openedv.com

官方网站: www.alientek.com

最新资料下载链接: <http://www.openedv.com/posts/list/13912.htm>

E-mail: 389063473@qq.com QQ: [389063473](https://www.qq.com/389063473)

咨询电话: [020-38271790](tel:020-38271790)

传真号码: [020-36773971](tel:020-36773971)

团队: [正点原子团队](#)

正点原子, 做最全面、最优秀的嵌入式开发平台软硬件供应商。

友 情 提 示

如果您想及时免费获取“正点原子”最新资讯, 敬请关注正点原子微信公众平台, 我们将及时给您发布最新消息和重要资料。



关注方法:

- (1) 微信“扫一扫”, 扫描右侧二维码, 添加关注
- (2) 微信→添加朋友→公众号→输入“正点原子”→关注
- (3) 微信→添加朋友→输入“alientek_stm32”→关注

