

第 3-2 讲：温湿度传感器 DHT11

1. 学习目的

1. 了解 DHT11 数字温湿度传感器的基本原理及其数据格式。
2. 掌握 STC8A8K64D4 与 DHT11 单总线通信的程序设计，通信步骤，数据校验等。

2. 硬件电路设计

2.1. 湿敏元件简介

湿敏元件是最简单的湿度传感器。湿敏元件比较常见的是电阻式湿敏元件和电容式湿敏元件。除此之外，还有电解质离子型湿敏元件、重量型湿敏元件(利用感湿膜重量的变化来改变振荡频率)、光强型湿敏元件、声表面波湿敏元件等。下面就电阻式湿敏元件和电容式湿敏元件进行介绍。

电阻式湿敏元件一般指的是湿敏电阻，湿敏电阻的特点是在基片上覆盖一层用感湿材料制成的膜，当空气中的水蒸气吸附在感湿膜上时，元件的电阻率和电阻值都发生变化，利用这一特性即可测量湿度。湿敏电阻的种类很多，例如金属氧化物湿敏电阻、硅湿敏电阻、陶瓷湿敏电阻等。

湿敏电阻的优点是灵敏度高，主要缺点是线性度和产品的互换性差。

电容式湿敏元件一般指的就是湿敏电容，湿敏电容一般是用高分子薄膜电容制成的，常用的高分子材料有聚苯乙烯、聚酰亚胺、醋酸醋酸纤维等。当环境湿度发生改变时，湿敏电容的介电常数发生变化，使其电容量也发生变化，其电容变化量与相对湿度成正比。湿敏电容的主要优点是灵敏度高、产品互换性好、响应速度快、湿度的滞后量小、便于制造、容易实现小型化和集成化，其精度一般比湿敏电阻要低一些。

湿敏元件的线性度及抗污染性差，在检测环境湿度时，湿敏元件要长期暴露在待测环境中，很容易被污染而影响其测量精度及长期稳定性。

湿度的概念大家比较了解，下面重点介绍下与湿度有关的知识点。湿度，一般在气象学中指的是空气湿度，他是空气中水蒸气的含量。空气中液态或固态的水不算在湿度中，不含水蒸气的空气被称为干空气。下面介绍与湿度有关的 3 个概念。

- 相对湿度 (Relative Humidity) 是指气体中 (通常为空气中) 所含水蒸气量 (水蒸气压) 与其空气相同情况下饱和水蒸气量 (饱和水蒸气压) 的百分比，一般用百分数表示。通常所说的湿度都是指相对湿度，比如说某个房间的湿度为 60%，即指该房间的相对湿度是 60%。
- 绝对湿度是指一定体积的空气中含有的水蒸气的质量，一般其单位是克/立方米。绝对湿度的最大限度是饱和状态下的最高湿度。绝对湿度只有与温度一起才有意义，因为空气中能够含有的湿度的量随温度而变化，在不同的温度中绝对湿度也不同，因为随着温度的变化空气的体积也要发生变化。但绝对湿度越靠近最高湿度，他随温度的变

化就越小。

- 饱和湿度是表示在一定温度下，单位容积空气中所能容纳的水蒸气量的最大限度。如果超过这个限度，多余的水蒸气就会凝结，变成水滴，此时的空气湿度便称为饱和湿度。

他们之间的关系是：相对湿度 = 绝对湿度 / 饱和湿度 × 100%。

2.2. DHT11 数字温湿度传感器

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，也是一款电容式湿度传感器。他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。

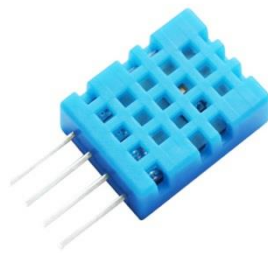


图 1：DHT11 数字温湿度传感器实物图

1. DHT11 数字温湿度传感器的规格参数

DHT11 数字温湿度传感器的规格参数如下表所示。

表 1：DHT11 数字温湿度传感器规格参数

DHT11 管脚号	功能描述
工作电压	3.3V~5.5V。
外形尺寸	23.2(L)mm×12.5(W)mm。
测量范围	温度：-20~+60℃ 湿度：5~95%RH。
精度	温度：±2℃。 湿度：±5%RH (25℃)。
分辨率	温度：0.1℃。 湿度：1%RH。
衰减 值	温度：<0.1℃/年。湿度：<1%RH/ 年。
输出信号	单总线数字信号。
外壳材料	ABS 塑料。
重量	1g。
引脚数	4 个。

2. DHT11 数字温湿度传感器的管脚定义

DHT11 数字温湿度传感器的引脚定义如下。

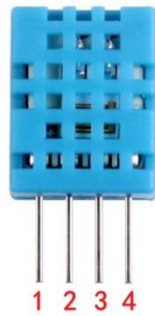


图 2: DHT11 数字温湿度传感器管脚号

4 个引脚的功能如下表所示。

表 2: DHT11 数字温湿度传感器引脚定义

DHT11 管脚号	管脚名	功能描述
1	VCC	供电，(3.3~5.5) V。
2	DATA	串行数据，单总线。
3	NC	空脚。
4	GND	接地，电源负极。

2.3. DHT11 接口电路

IK-64D4 开发板上设计了 DHT11 数字温湿度传感器接口，电路如下图所示，这里需要注意的是 DHT11 的 DATA 引脚需要上拉，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，开发板上连接 DHT11 的线长比较短，因此，开发板用的是 10K 上拉电阻。

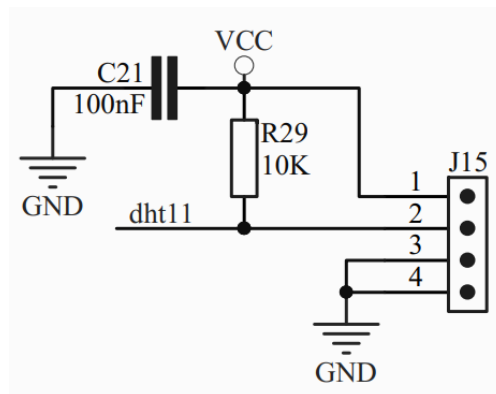


图 3: DHT11 接口电路

DHT11 占用的 STC8A8K64D4 的引脚如下表：

表 3: DHT11 引脚分配

DHT11	引脚	说明
DHT11 数据	P7.0	和 DAC 电路的输入共用

3. 软件设计

3.1. DHT11 应用步骤

DHT11 数字温湿度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。设备（主机或从机）通过一个漏极开路或三态端口口连至该数据线，以允许设备在不发送数据时能够释放总线，而让其他设备使用总线；单总线通常要求外接一个约 $4.7k\Omega$ 的上拉电阻，这样，当总线闲置时，其状态为高电平。由于他们是主从结构，只有主机呼叫从机时，从机才能应答，因此主机访问器件都必须严格遵循单总线序列，如果出现序列混乱，器件将不响应主机。

MCU 作为单总线通信的主机与 DHT11 从机之间通信的步骤如下图所示，共需 4 个步骤。

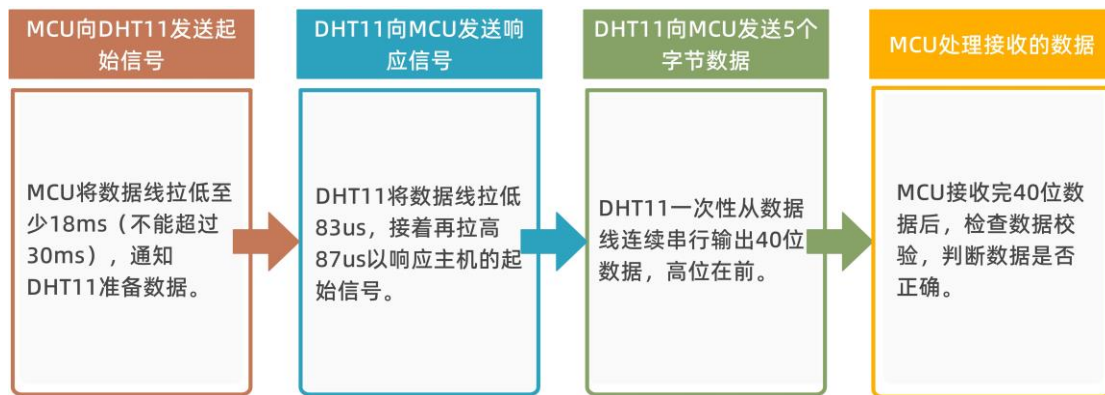


图 4：MCU 与 DHT11 单总线通信步骤

1. MCU 向 DHT11 发送起始信号

DHT11 上电后要等待 1S 以越过不稳定状态，在此期间不能发送任何指令。MCU 的 I/O 此时配置为输入，因为 DHT11 的 DATA 数据线接了上拉电阻，因此保持高电平。

MCU 从 DHT11 读取温湿度数据时，首先发送起始信号，即 MCU 配置 I/O 为输出，同时输出低电平，且低电平保持时间不能小于 18ms（最大不得超过 30ms），然后 MCU 的 I/O 设置为输入状态，由于上拉电阻，微处理器的 I/O 即 DHT11 的 DATA 数据线也随之变高，等待 DHT11 作出回答信号。

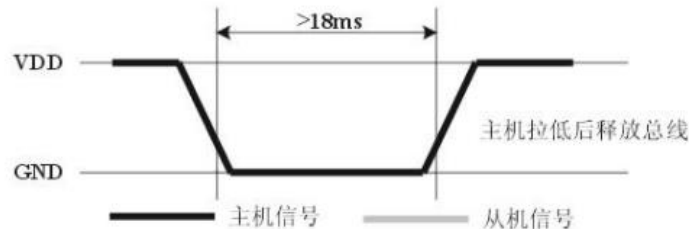


图 5：起始信号时序

2. DHT11 向 MCU 发送响应信号

DHT11 的 DATA 引脚检测到外部信号有低电平时，等待外部信号低电平结束，延迟后 DHT11 的 DATA 引脚处于输出状态，输出 83 微秒的低电平作为应答信号，紧接着输出 87

微秒的高电平通知外设准备接收数据，MCU 的 I/O 此时处于输入状态，检测到 I/O 有低电平（DHT11 回应信号）后，等待 87 微秒的高电平后的数据接收。

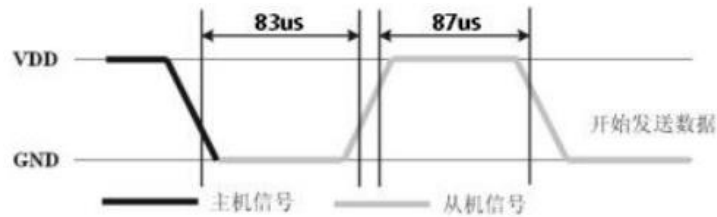


图 6: DHT11 响应信号时序

3. DHT11 向 MCU 发送 5 个字节数据

由 DHT11 的 DATA 引脚输出 40 位数据，MCU 根据 I/O 电平的变化接收 40 位数据，“0”和“1”的格式定义如下：

- 位数据“0”：54us 的低电平 + (23~27) us 的高电平；
- 位数据“1”：54us 的低电平 + (68~74) us 的高电平。

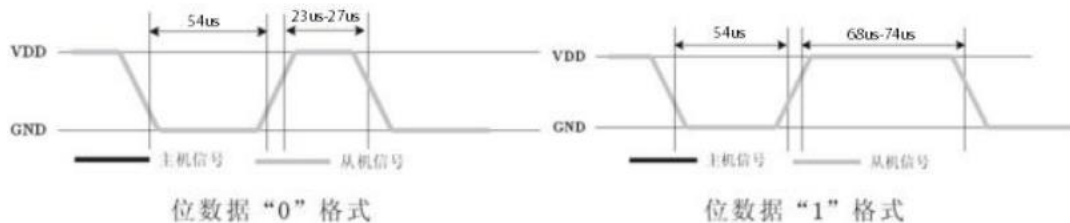


图 7: “0”和“1”的时序

40 位数据包含了温湿度和校验，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 校验位，其中湿度小数部分为 0。

结束信号：DHT11 的 DATA 引脚输出 40 位数据后，继续输出低电平 54 微秒后转为输入状态，由于上拉电阻随之变为高电平。但 DHT11 内部重测环境温度温湿度数据，并记录数据，等待外部信号的到来。

4. MCU 处理接收的数据

MCU 接收完数据之后，需要检查数据的校验，以判断接收的数据是否正确，检查方式为：8bit 校验位=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。

MCU 解析温度数据时，需要注意 DHT11 正/负温度的表示方式：当温度低于 0℃时温度数据的低 8 位的最高位置为 1，否则为 0。

■ 正温示例

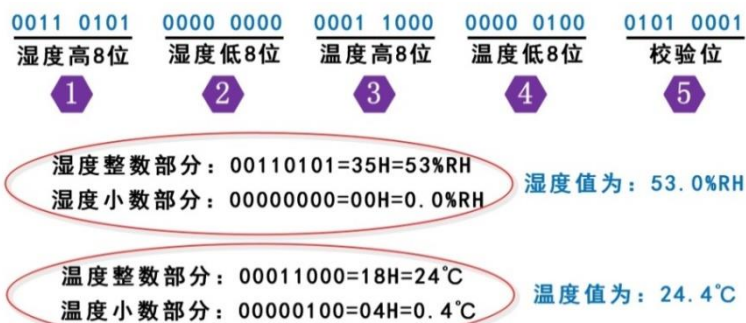


图 8：正温度数据

■ 负温示例



图 9：负温度数据

3.2. DHT11 温湿度读取实验

✧ 注：本节的实验是在“实验 2-6-1：串口 1 数据收发实验”的基础上修改，本节对应的实验源码是：“实验 2-21-1：DHT11 温湿度读取（串口发送温湿度）”。

3.2.1. 实验内容

每 2.5 秒从 DHT11 读取一次温湿度数据并通过串口 1 输出温湿度数据。

3.2.2. 代码编写

1. 新建一个名称为“dht11.c”的文件及其头文件“dht11.h”并保存到工程的“Source”文件夹，并将“dht11.c”加入到 Keil 工程中的“SOURCE”组。
2. 引用头文件

因为在“dht11.c”文件中使用了“dht11.c”文件中的函数，所以需要引用下面的头文件“dht11.h”。

代码清单：引用头文件

```
1. //引用 DHT11 的头文件
2. #include "dht11.h"
```

3. 读取温湿度数据

按照前文描述的步骤，编写读取 DHT11 温湿度数据的代码。MCU 首先向 DHT11 发送起始信号、之后等待 DHT11 向 MCU 发送响应信号，接收到响应信号后 MCU 继续接收

DHT11 向 MCU 发送的 40 位数据。注意数据读取完成后，需要检查数据的校验。

代码清单：读取温湿度数据

```
1.  /*****
2.  功能描述：一次完整的数据传输为 40bit，高位在前
3.  参    数：8bit 湿度整数 + 8bit 湿度小数 + 8bit 温度整数 + 8bit 温度小数 + 8bit 校验和
4.  返 回 值：DHT11_SUCCESS: 成功, DHT11_NACK: 无应答, DHT11_DATA_ERR: 数据校验错误
5.  *****/
6.  u8 Read_DHT11(DHT11_Data_t *DHT11_Data)
7.  {
8.      DATA_PIN = 0; //主机输出低电平
9.      delay_ms(20); //低电平时间不能小于 18ms，不能超过 30ms，这里延时 20ms
10.     DATA_PIN = 1; //主机输出高电平
11.     Delay10us(); //延时 30us
12.     Delay10us();
13.     Delay10us();
14.     if(!DATA_PIN) //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
15.     {
16.         //DHT11 发出 83us 低电平应答信号。主机等待低电平结束
17.         if(waitfor_state(1)!=DHT11_SUCCESS)return DHT11_NACK;
18.         //DHT11 发出 87us 高电平信号通知主机准备接收数据。主机等待高电平结束
19.         if(waitfor_state(0)!=DHT11_SUCCESS)return DHT11_NACK;
20.         //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 校验位
21.         //因此这里要连续读出 5 个字节
22.         DHT11_Data->h_int= Read_Byte();
23.         DHT11_Data->h_deci= Read_Byte();
24.         DHT11_Data->t_int= Read_Byte();
25.         DHT11_Data->t_deci= Read_Byte();
26.         DHT11_Data->check_sum= Read_Byte();
27.
28.         DATA_PIN = 1; //主机输出高电平
29.         //校验: 校验和=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 校验位
30.         if(DHT11_Data->check_sum == DHT11_Data->h_int + DHT11_Data->h_deci + DHT11_Data->t_int+
31.            DHT11_Data->t_deci)
32.             return DHT11_SUCCESS;
33.         else
34.             return DHT11_DATA_ERR; //校验错误，返回错误代码 DHT11_DATA_ERR
35.     }
36.     else //DHT11 无响应，返回错误代码 DHT11_NACK
37.     {
38.         return DHT11_NACK;
39.     }
40. }
```

上面代码中的 23~27 行，MCU 通过调用 Read_Byte()函数分 5 次接收 DHT11 的 40 位数据（40 位共 5 个字节），这样更便于程序的处理。Read_Byte()函数代码清单如下。

代码清单：接收一个字节数据

```

1.  *****
2.  功能描述：从 DHT11 读取一个字节，MSB 在前
3.  参    数：无
4.  返 回 值：读取的数据
5.  *****/
6.  static u8 Read_Byte(void)
7.  {
8.      u8 i, temp=0;
9.      //循环读取 8 个位
10.     for(i=0;i<8;i++)
11.     {
12.         //每 bit 以 50us 低电平标置开始。等待引脚状态变为高电平，100us 无变化认为超时，退出函数
13.         if(waitfor_state(1)!=DHT11_SUCCESS)return DHT11_NACK;
14.         //DHT11 以 23~27us 的高电平表示 0，以 68~74us 的高电平表示 1
15.         Delay10us(); //延时 40us
16.         Delay10us();
17.         Delay10us();
18.         Delay10us();
19.         //40us 后仍为高电平表示数据 1
20.         if(DATA_PIN==1)
21.         {
22.             if(waitfor_state(0)!=DHT11_SUCCESS)return DHT11_NACK; //等待数据 1 的高电平结束，低电平到来
23.             temp|=(u8)(0x01<<(7-i)); //保存数据位“1”，MSB 在前
24.         }
25.         else//40us 后为低电平表示数据 0
26.         {
27.             temp&=(u8)~(0x01<<(7-i)); //保存数据位“0”，MSB 在前
28.         }
29.     }
30.     return temp;
31. }

```

MCU 在等待 DHT11 信号时，为了防止堵塞，我们在等待中加入超时时间，如果达到超时时间后，MCU 等待的信号没有出现，则认为出现错误，程序退出等待，同时返回错误代码，这样就可以避免由于 DHT11 故障导致出现堵塞的问题，同时通过错误代码通知了应用程序故障原因。MCU 等待 DHT11 信号的代码清单如下。

代码清单：等待 DHT11 信号

```

1.  /*****
2.  功能描述：等待连接 DHT11 的引脚状态变化，为了防止 DHT11 损坏出现堵塞，设置超时时间为 100us
3.  参    数：[in]pin_state,0=低电平，1=高电平

```



```

4. 返回值: DHT11_SUCCESS: 成功, DHT11_NACK: 无应答
5. *****/
6. static u32 waitfor_state(u8 pin_state)
7. {
8.     //超时时间 100us, 100us 内引脚状态无变化, 认为超时
9.     u8 delay_us = 10;
10.    do
11.    {
12.        //读取引脚状态, 如引脚状态变化, 函数运行结束, 返回 DHT11_SUCCESS
13.        if(DATA_PIN == pin_state)
14.        {
15.            return DHT11_SUCCESS;
16.        }
17.        Delay10us();
18.        delay_us--;
19.    }while(delay_us);
20.
21.    return DHT11_NACK; //返回错误代码: DHT11_NACK
22. }

```

4. 主函数

主函数中配置连接 DHT11 数据引脚的 GPIO P7.0 为准双向后, 之后在主循环里面使用 Read_DHT11()函数每隔 2.5 秒读一次温湿度数据, 并通过串口 1 输出数据。

代码清单: 主函数

```

1.  /*****
2.  功能描述: 主函数
3.  参    数: 无
4.  返回值: int 类型
5.  *****/
6. int main(void)
7. {
8.     u8 err_code;
9.     P2M1 &= 0xBF;   P2M0 &= 0xBF;   //设置 P2.6 为准双向口 (指示灯 D1)
10.    P3M1 &= 0xFE;   P3M0 &= 0xFE;   //设置 P3.0 为准双向口 (串口 1 的 Rx/D)
11.    P3M1 &= 0xFD;   P3M0 |= 0x02;   //设置 P3.1 为推挽输出 (串口 1 的 Tx/D)
12.    P7M1 &= 0xFE;   P7M0 &= 0xFE;   //设置 P7.0 为准双向口 (DHT11 数据)
13.
14.    DATA_PIN=1;      //总线拉高
15.    uart1_init();      //串口 1 初始化
16.    EA = 1;           //使能总中断
17.
18.    while(1)
19.    {

```

```
20.     err_code = Read_DHT11(&DHT11_Data); //从 DHT11 读取温湿度数据
21.     //判断函数返回值，读取成功，串口输出温湿度数据
22.     if(err_code==DHT11_SUCCESS)
23.     {
24.         printf("Temperature: %d.%d °C   Humidity: %d.%d %RH\r\n",DHT11_Data.t_int,
25.             DHT11_Data.t_deci,DHT11_Data.h_int,DHT11_Data.h_deci);
26.     }
27.     else if(err_code==DHT11_DATA_ERR) //数据校验错误
28.     {
29.         printf("Data Checksum ERROR!\r\n");
30.     }
31.     else if(err_code==DHT11_NACK) //DHT11 无应答
32.     {
33.         printf("No ack ERROR!\r\n");
34.     }
35.     else
36.     {
37.     }
38.     delay_ms(2500); //每次读取温湿度数据之间的间隔不能小于 2 秒
39.     led_toggle(LED_1); //指示灯 D1 状态翻转，指示程序的运行
40. }
41. }
```

3.2.3. 硬件连接

本实验需要使用 DHT11 数字温湿度传感器，开发板需要将 P7.0 通过跳线连接到 DHT11 数字温湿度传感器，如下图所示。

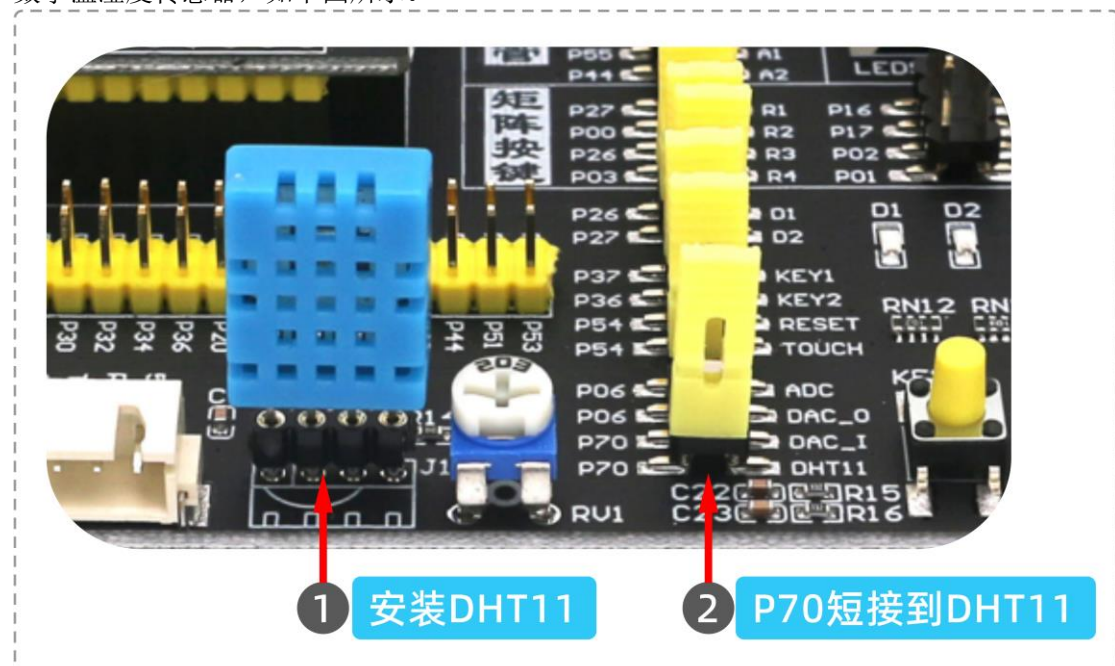


图 10：硬件连接

3.2.4. 实验步骤

- 1) 解压“…\第3部分：配套例程源码”目录下的压缩文件“实验 2-21-1：DHT11 温湿度读取（串口发送温湿度）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC8”（这样做的目的是为了防止中文路径或者工程存放的路径过深导致打开工程出现问题）。
- 2) 双击“…\dht11\project”目录下的工程文件“dht11.uvproj”。
- 3) 点击编译按钮编译工程，编译成功后生成的 HEX 文件“dht11.hex”位于工程的“…\dht11\Project\Object”目录下。
- 4) 打开 STC-ISP 软件下载程序，下载使用内部 IRC 时钟，IRC 频率选择：24MHz。
- 5) 电脑上打开串口调试助手，选择开发板对应的串口号，将波特率设置为 9600bps。
- 6) 程序运行后，开发板每 2.5 秒读取一次 DHT11 的温湿度数据，每次读取时取反一次 D1 指示灯的状态指示程序的运行。读取的数据通过串口输出，电脑上打开串口调试助手，即可观察到读取的温湿度数据，如下图所示。



图 11：串口接收的按键信息