



成 绩	
批阅教师	
日 期	

# 桂林电子科技大学 实训报告

2021-2022 学年第 2 学期

学 院 电子信息学院

课 程 智能控制实训

姓 名 文海霞 陆树森 杨丰铭

学 号 201202020211 201202020225 201202020233

指导老师 江德业

# 实训题目：基于 DS18B20 的水温预警系统设计

## 1 概述

### 1.1 设计要求

#### 1.1.1 设计任务

基础部分：（1）用单片机实现。（2）用 C 语言编程。（3）电路板布局合理（4）用数码管显示器显示。（5）温度测量范围是：25 度到 90 度。（6）误差小于 8%。

扩展部分：基本要求为实现精确到小数点后四位的温度测量与显示。在此基础上额外增加了湿度检测与光照强度检测，时间显示等功能。

在温度传感器的选型上采用 DS18B20，原因是它具有 12 位精度，最小分辨温度为  $0.0625^{\circ}\text{C}$ ，误差范围为  $\pm 0.5^{\circ}\text{C}$ ，可实现高精度测量。

在湿度传感器的选型上采用 DTH11，原因是它价格较低，传感器稳定性可靠性高，携带方便，较为合适。

在光照度传感器的选型上采用 GY-302，原因是它稳定性好，可以直接数字输出数据，省略复杂的计算过程，且不区分环境光源，接近于视觉灵敏度的分光特性。

在数据显示器件的选型上采用淘晶驰 4024T032\_011 串口屏，因为该屏幕可通过串口通信连接至电脑上位机，实现上位机自由编程，可塑性强，开发方便，显示效果多样。

#### 1.1.2 性能指标要求

能读取 DS18B20 温度，并将其显示出来，其中温度误差不大于  $\pm 8\%$ ，扩展功能湿度光照度无误差要求。

#### 1.1.3 设计背景简介

在生活中，无论是冬天还是夏天我们都离不开热水。把水烧开很容易，但是要把水温控制在一定的温度而且进行自动控制就不是那么随便。所以为了更高效和方便人们的需求。在如今，温度的测量无处不在，水温的测量，气温的测量，工作环境的温度测量等等。本设计就专门利用单片机和 DS18B20 温度传感器来实现了一个水温自动控制系统，希望通过本设计能够对未来的趋势有所把握，从而适应社会需要。本次设计使用单片机，液晶显示器件和温度传感器为主要器件，由温度采集电路，报警电路和单片机最小系统电路构成，温度传感器采集温度信号经过放大电路的处理送进单片机，然后通过单片机输出信号到液晶显示器显示温度。

### 1.2.1 基本原理

在通讯方式上 DS18B20,DTH11 均采用单总线通信方式,即可以传输时钟又传输数据,而且数据是双向的,资源结构简单、成本低廉。由于数字式集成温度传感器 DS18B20 的高度集成化,大大的降低了外接放大转化等电路的误差因数,温度误差变得很小,并且由于其检测温度的原理与热敏电阻检测的原理有着本质的不同,使得其温度分辨率极高。温度值在器件内部转化成数字量直接输出,简化了系统程序设计,又由于该温度传感器采用先进的单总线技术,与单片机接口变得非常简洁,抗干扰能力强。在测温精度、转换时间、传输距离、分辨率等方面较 DS1820 有了很大的改进,给用户带来了更方便的使用和更令人满意的效果。同时照度传感器 GY-302 则采用 I2C 通信,并将读取的数据传给串口屏显示。

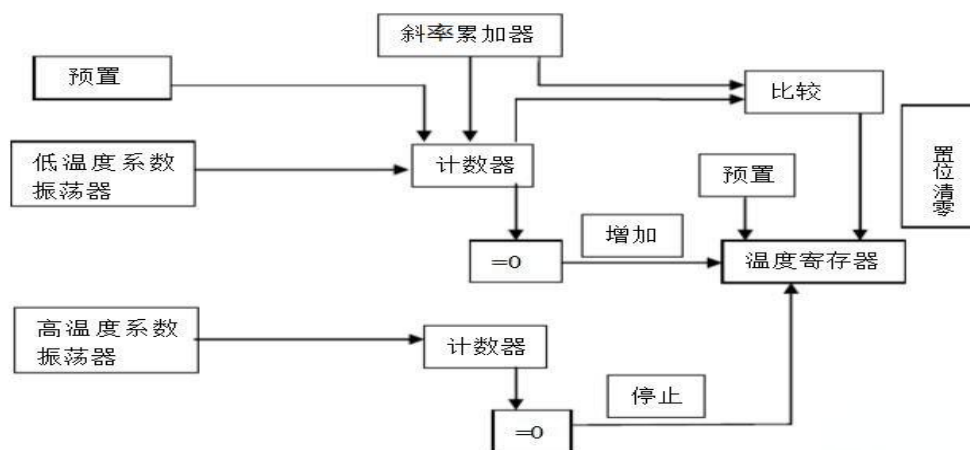


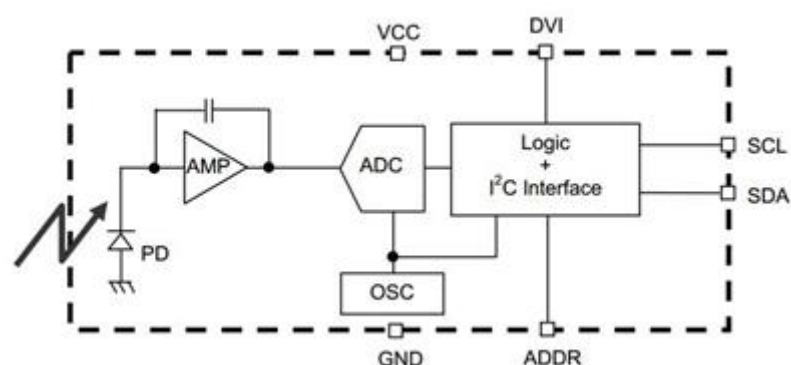
图 1

DS18B20 的测温原理如图 1 所示，图中低温度系数晶振的振荡频率受温度的影响很小，用于产生固定频率的脉冲信号送给减法计数器 1，高温温度系数晶振随温度变化其震荡频率明显改变，所产生的信号作为减法计数器 2 的脉冲输入，图中还隐含着计数门，当计数门打开时，DS18B20 就对低温度系数振荡器产生的时钟脉冲后进行计数，进而完成温度测量。

计数门的开启时间由高温系数振荡器来决定，每次测量前，首先将 0-55℃ 所对应的基数分别置入减法计数器 1 和温度寄存器中，减法计数器 1 和温度寄存器被预置在 -55℃ 所对应的一个基数值。

减法计数器 1 对低温系数晶振产生的脉冲信号进行减法计数，当减法计数器 1 的预置值减到 0 时温度寄存器的值将加 1，减法计数器 1 的预置将重新被装入，减法计数器 1 重新开始对低温系数晶振产生的脉冲信号进行计数，如此循环直到减法计数器 2 计数到 0 时，停止温度寄存器值的累加，此时温度寄存器中的数值即为所测温度。

图 1 中的斜率累加器用于补偿和修正测温过程中的非线性，其输出用于修正减法计数器的预置值，只要计数门仍未关闭就重复上述过程，直至温度寄存器值达到被测温度值。



GY-302 原理:

光照传感器是一种传感器，用于检测光照强度，简称照度，工作原理是将光照强度值转为电压值，主要用于农业林业温室大棚培育等。

设面元  $dS$  上的光通量为  $d\Phi$ ，则此面元上的照度  $E$  为： $E=d\Phi/dS$ 。照度的单位为 lx（勒克斯），也有用 lux 的， $1lx=1lm/m^2$ 。照度表示物体表面积被照明程度的量。不同角度光线透过余弦修正器汇聚到感光区域、汇聚到感光区域的太阳光通过蓝色和黄色进口滤光片过滤掉可见光以外的光线;透过滤光片的可见光照射到进口光敏二极管，光敏二极管根据可见光照度大小转换成电信号，电信号进入单片机系统，单片机系统根据温度感应电路，将采集到的光电信号进行温度补偿，以输出精准的线性电信号。为了提高变送器的精度和分辨能力，我们采用了独特地专利技术，双通道信号输出，分别是 KLUX 级信号和 LUX 级信号。

GY302 如图 2:

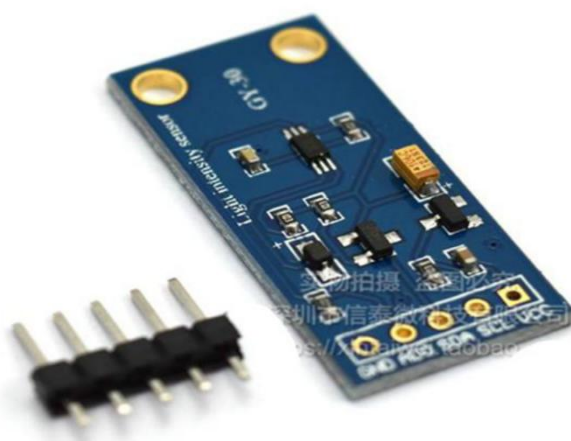
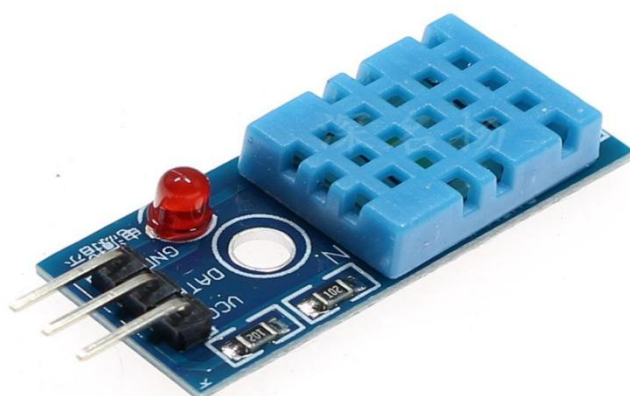


图 2

BH1750 的内部由光敏二极管、运算放大器、ADC 采集、晶振等组成。PD 二极管通过光生伏特效应将输入光信号转换成电信号，经运算放大电路放大后，由 ADC 采集电压，然后通过逻辑电路转换成 16 位二进制数存储在内部的寄存器中。进入光窗的光越强，光电流越大，电压就越大，所以通过电压的大小就可以判断光照大小。BH1750 引出了时钟线和数据线，单片机通过 I2C 协议可以与 BH1750 模块通讯，可以选择 BH1750 的工作方式，也可以将 BH1750 寄存器的光照度数据提取出来。

DTH11 如图 3:



### 图 3

DHT11 原理:

DHT11 如图 3，包含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。

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。设备（主机或从机）通过一个漏极开路或三态端口连至该数据线，以允许设备在不发送数据时能够释放总线，而让其它设备使用总线。

蜂鸣器原理:

蜂鸣器是一种一体化结构的电子讯响器，采用直流电压供电，广泛应用于计算机、打印机、复印机、报警器、电子玩具、汽车电子设备、电话机、定时器等电子产品中作发声器件。蜂鸣器主要分为压电式蜂鸣器和电磁式蜂鸣器两种类型。蜂鸣器在电路中用字母“H”或“HA”（旧标准用“FM”、“LB”、“JD”等）表示。本次采用的是压电式有源蜂鸣器主要由多谐振荡器、压电式蜂鸣片、阻抗匹配器及共鸣箱、外壳等组成。



蜂鸣器如图 4:

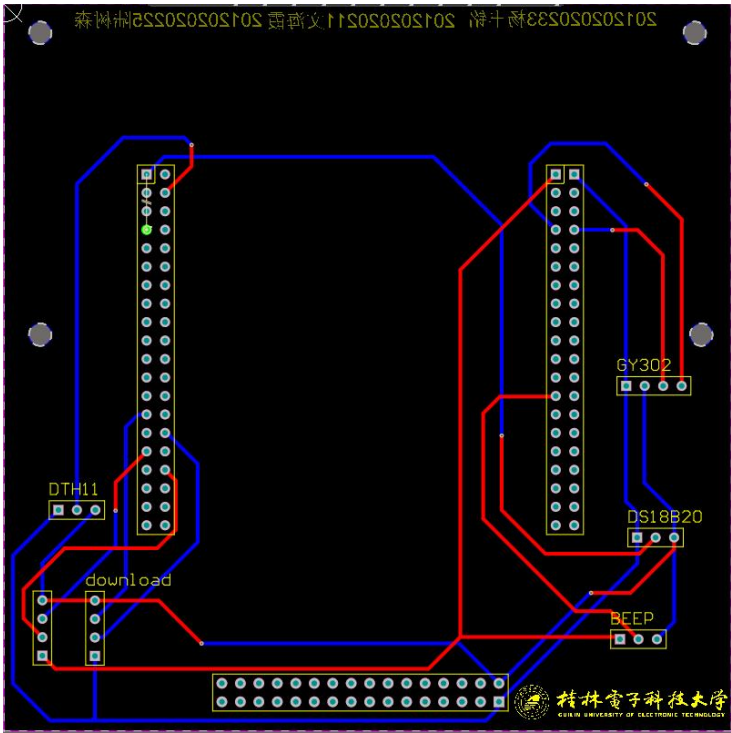
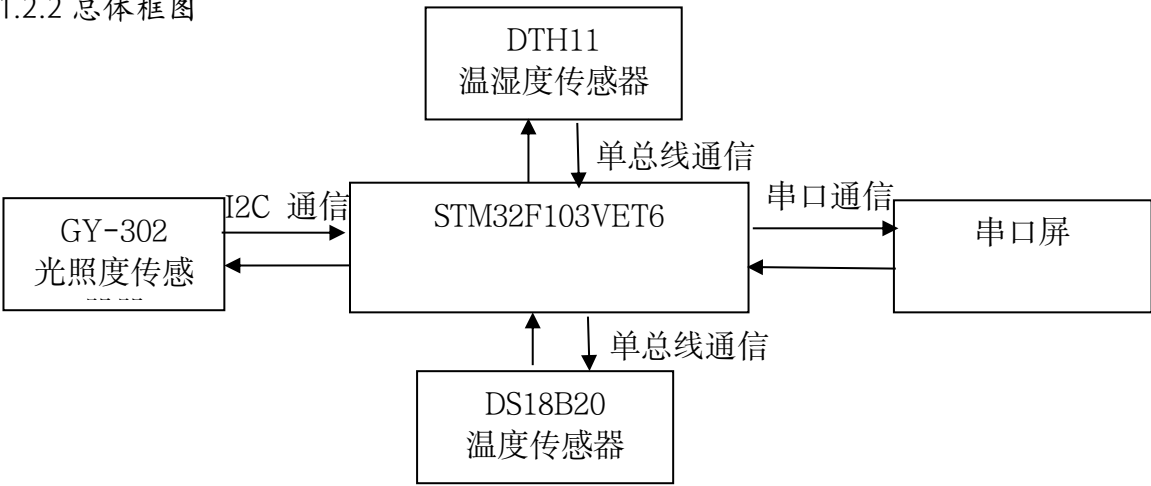
### 图 4

串口屏原理:

通过接收 STM32F103VET6 串口 1 发送来的带有\xff\xff\xff 结束标志位的数据来实现显示各项数据的功能。其中时间加减是先由 page3 的虚拟按键将局

部变量 n0.val、n1.val、n2.val 加减然后传递给全局变量 sys0、sys1、sys2 实现，时钟运行依赖串口屏自带 48M 时钟定时器实现。定时器代码详见附录 5。

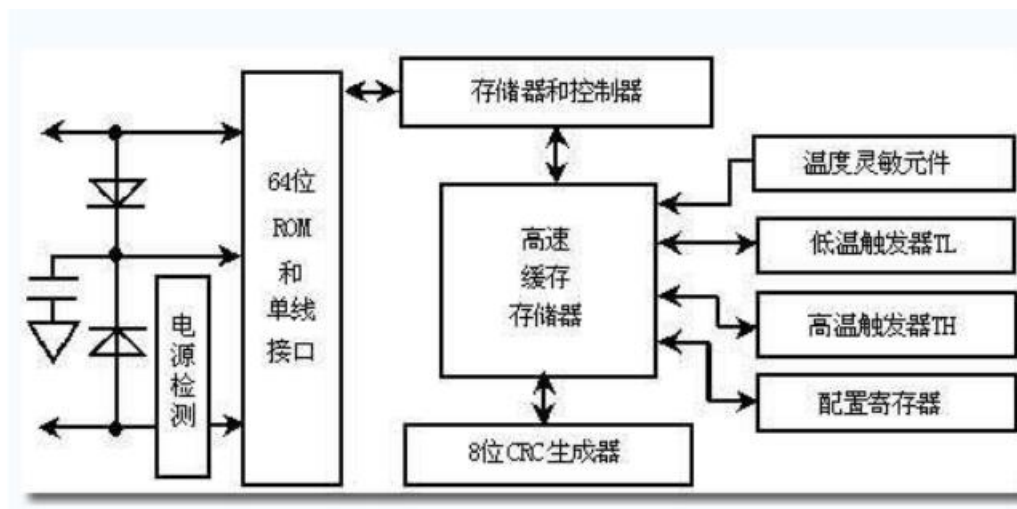
1.2.2 总体框图



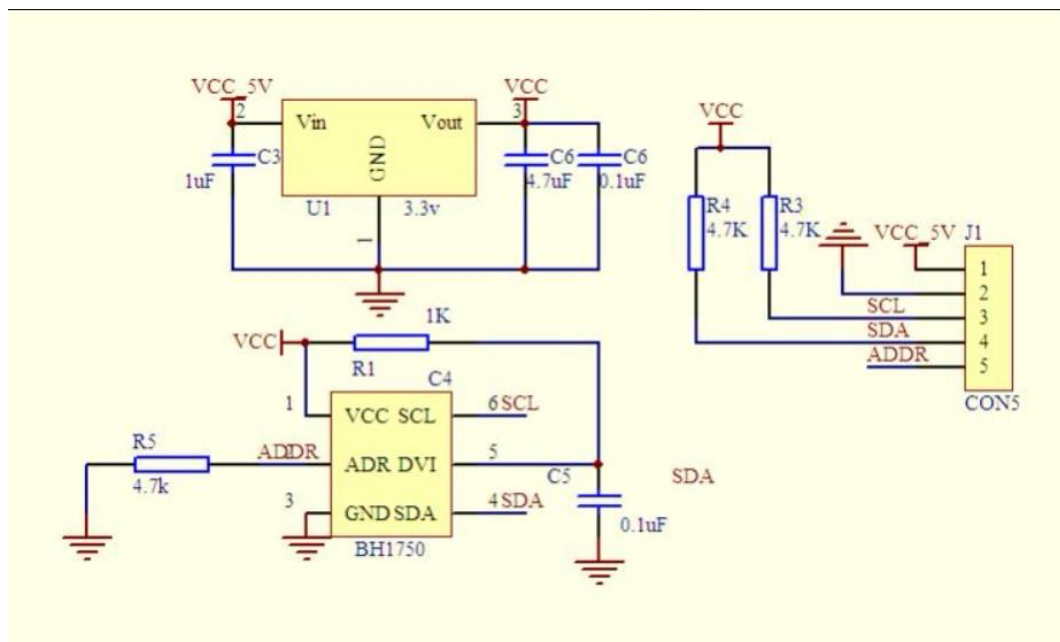
2 系统硬件电路设计

硬件 pcb 图



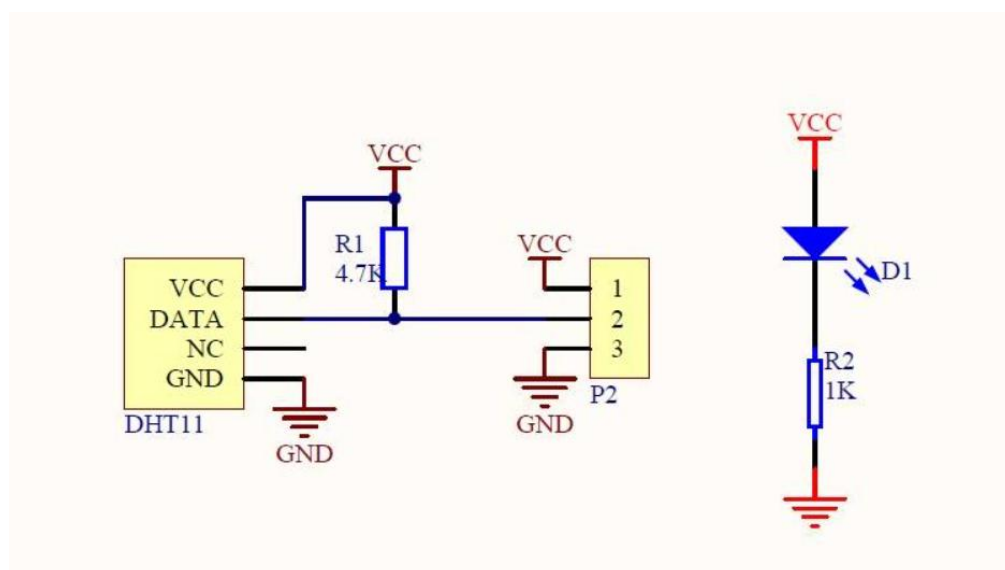


DS18B20 底层原理图

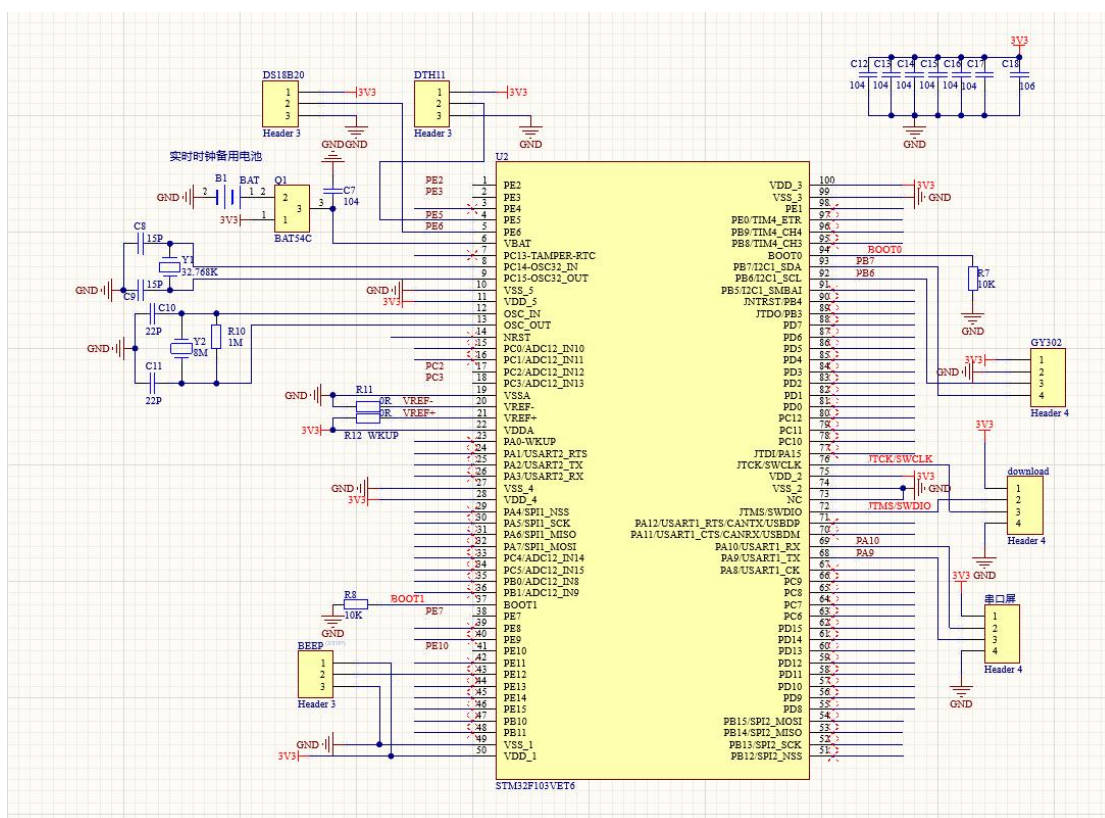


GY302 原理图





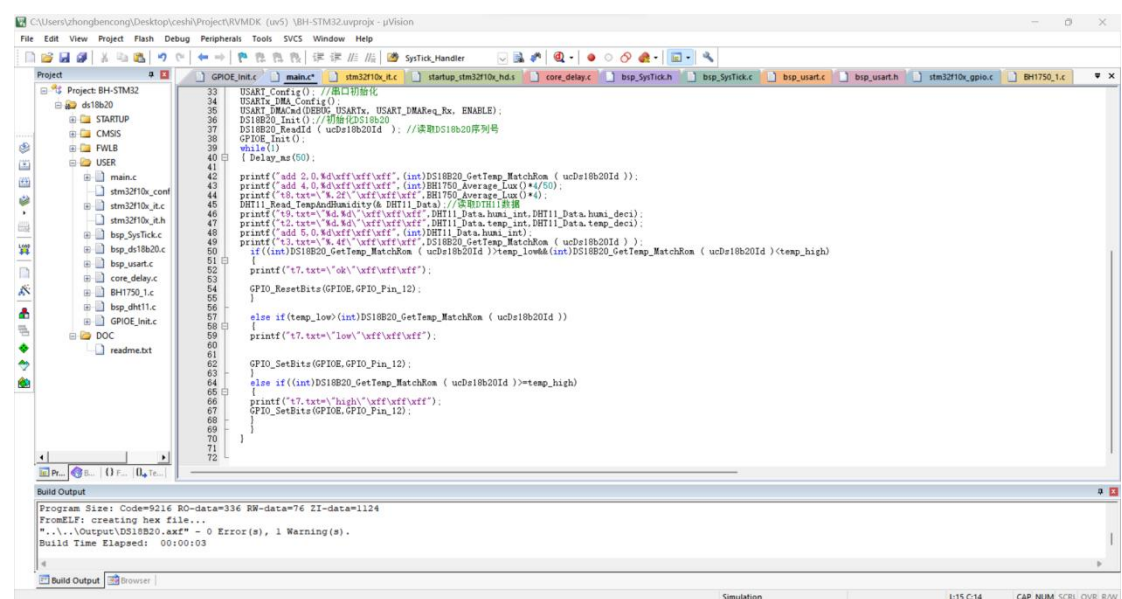
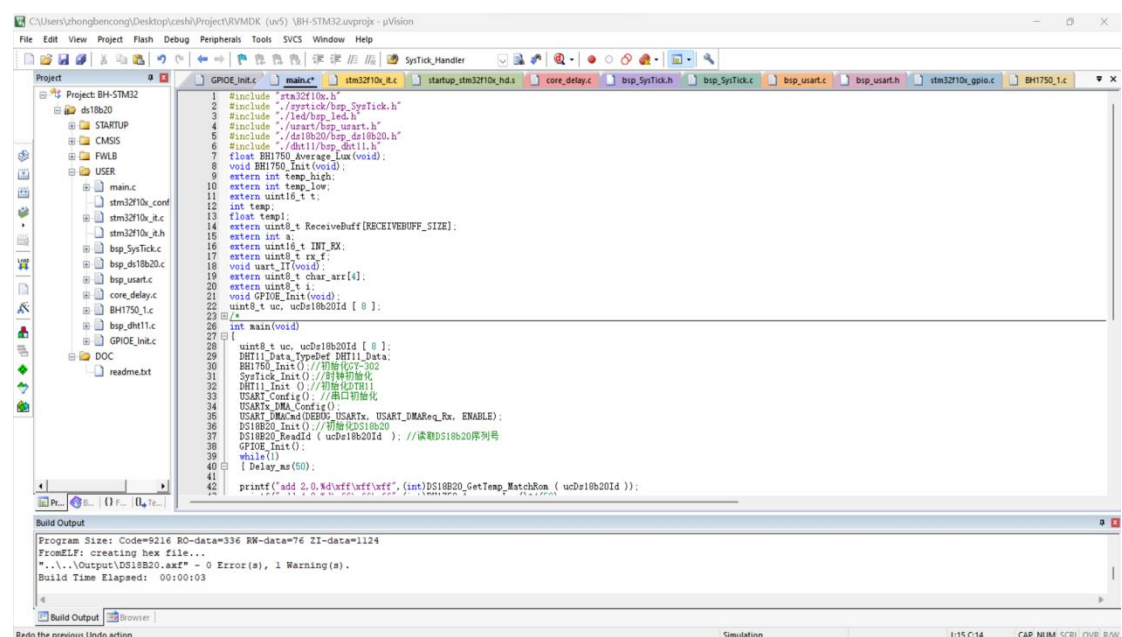
## DTH11 原理图



## 硬件原理图

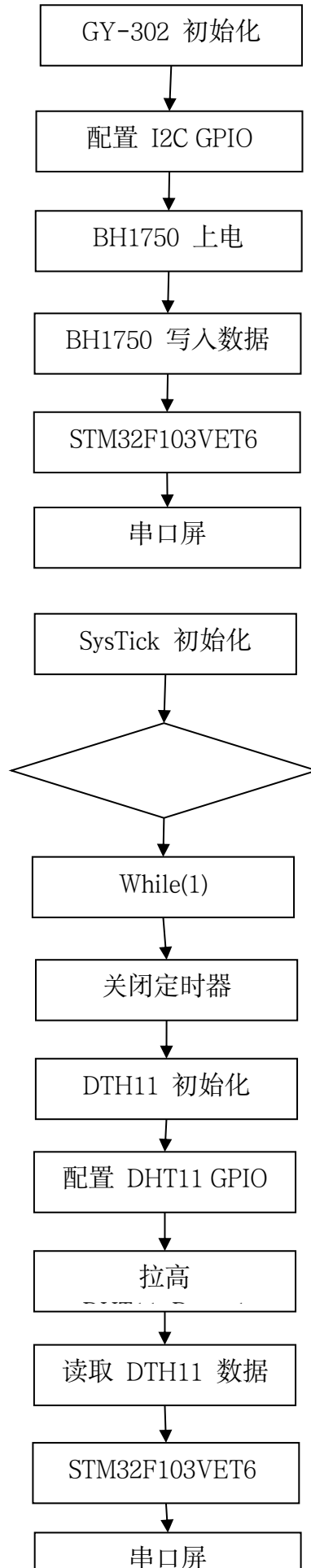
## 3 系统程序设计

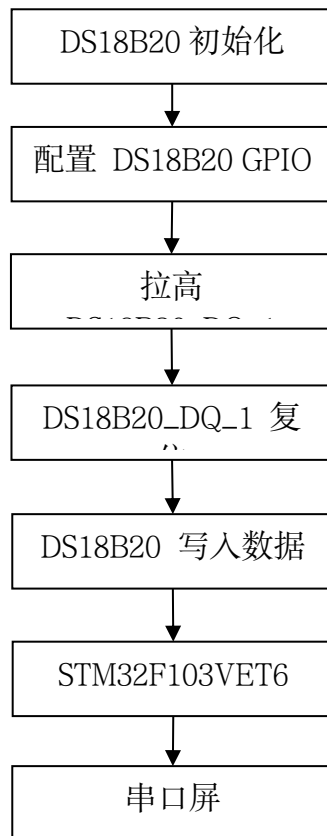
### 3.1 系统主程序流程设计



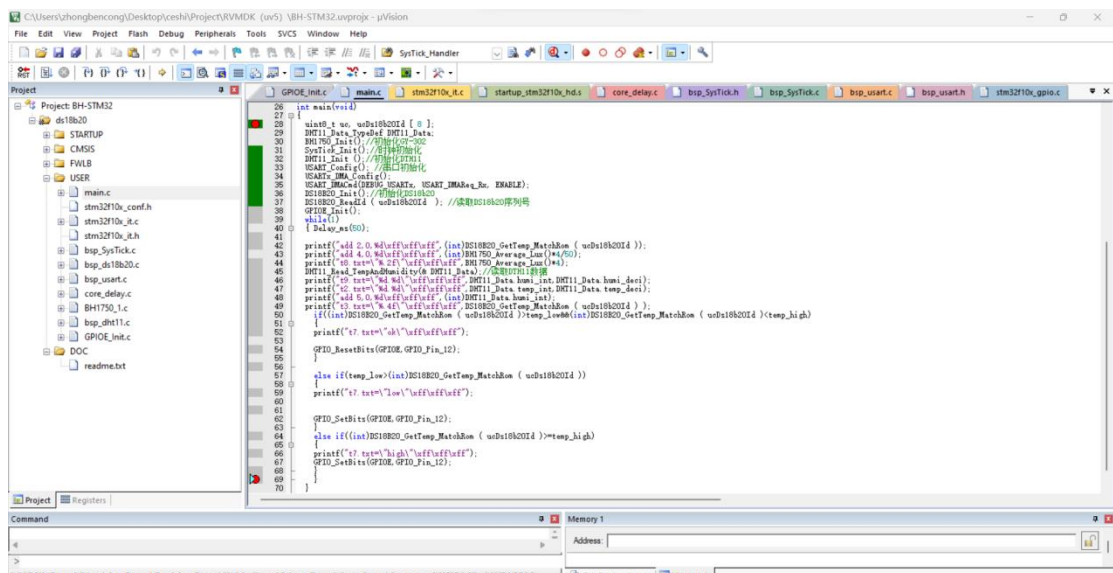
## 系统主程序设计图

### 3.2 系统各功能模块子程序流程设计





## 4 系统测试(调试)



### 4.1 硬件调试

经过硬件仿真程序可以正常运行，无卡死报错现象

### 4.2 系统联调

经过上电测试，各模块运行状态稳定，系统工作正常

主页



温度监控



湿度及光照检测



#### 4.3 测试效果及结果分析

测量温度	21.9375	24.2500	28.4375	32.6875	34.8750	37.0625
实际温度	21.56	24.67	28.96	32.22	34.43	36.51
误差	-1.72%	+1.73%	+1.83%	-1.45%	-1.27%	-1.49%

测量光照	974	1204	3204	4320	6421	8435
实际光照	1034	1345	3478	4024	6962	9035
误差	+6.16%	+11.71%	+8.55%	-6.85%	+8.42%	+7.11%

测试结果误差都在合理范围内，证明实训作品成功实现设计目标

## 5 实训总结

经过此次实训加深了对 STM32 时钟和串口的了解，同时复习了 I2C 通信和单总线通信的原理，提高了绘制 PCB 的能力，了解了上位机的使用，STM32 的调试。完成此次实训是比较系统化的一个课题，需要我们共同学习交流，锻炼我们的学习能力和动手能力，将理论与实践结合起来，才能达到我们想要的效果。此次实训基础部分实现了对温度的监测与报警，拓展部分实现了光照与湿度检测，作品整体模块化程度高，工作状态稳定，不足在于未实现对温度的控

制，后续可以通过 PID 算法配合相关加热设备实现对温度的同步测控。

## 6 参考文献

- [1] 刘火良. 杨森. 基于 STM32+STM32 库开发实战指南: 基于 STM32F103 (第 2 版). 北京:机械工业出版社,2019
- [2] 叶俊明.苏鹏鉴. 单片机 C 语言程序设计. 西安:西安电子科技大学出版社,2017
- [3]宋雪臣. 传感器与检测技术 (第 2 版). 北京:人民邮电出版社,2015

## 附 录

## 附录 1 main 程序

```
#include "stm32f10x.h"
#include "../systick/bsp_SysTick.h"
#include "../led/bsp_led.h"
#include "../usart/bsp_usart.h"
#include "../ds18b20/bsp_ds18b20.h"
#include "../dht11/bsp_dht11.h"
float BH1750_Average_Lux(void);
void BH1750_Init(void);
extern int temp_high;
extern int temp_low;
extern uint16_t t;
int temp;
float temp1;
extern uint8_t ReceiveBuff[RECEIVEBUFF_SIZE];
extern int a;
extern uint16_t INT_RX;
extern uint8_t rx_f;
void uart_IT(void);
extern uint8_t char_arr[4];
extern uint8_t i;
void GPIOE_Init(void);
uint8_t uc, ucDs18b20Id [ 8 ];
/*
\xff\xff\xff 是串口屏接受单片机指令结束标志
*/
int main(void)
{
    uint8_t uc, ucDs18b20Id [ 8 ];
    DHT11_Data_TypeDef DHT11_Data;
    BH1750_Init();//初始化 GY-302
    SysTick_Init();//时钟初始化
    DHT11_Init ();//初始化 DTH11
    USART_Config();    //串口初始化
```



```

USARTx_DMA_Config();
    USART_DMAMCmd(DEBUG_USARTx, USART_DMAREq_Rx, ENABLE);
    DS18B20_Init();//初始化 DS18b20
DS18B20_ReadId ( ucDs18b20Id );//读取 DS18b20 序列号
    GPIOE_Init();
    while(1)
    {
        Delay_ms(50);

        printf("add          2,0,%d\\xff\\xff\\xff",(int)DS18B20_GetTemp_MatchRom
( ucDs18b20Id ));
        printf("add 4,0,%d\\xff\\xff\\xff",(int)BH1750_Average_Lux()*4/50);
        printf("t8.txt=\\\"%.2f\\\"\\xff\\xff\\xff",BH1750_Average_Lux()*4);
        DHT11_Read_TempAndHumidity(& DHT11_Data);//读取 DTH11 数据
        printf("t9.txt=\\\"%.d.%d\\\"\\xff\\xff\\xff",DHT11_Data.humi_int,DHT11_Data.humi_dec
i);
        printf("t2.txt=\\\"%.d.%d\\\"\\xff\\xff\\xff",DHT11_Data.temp_int,DHT11_Data.temp_deci
;
        printf("add 5,0,%d\\xff\\xff\\xff",(int)DHT11_Data.humi_int);
        printf("t3.txt=\\\"%.4f\\\"\\xff\\xff\\xff",DS18B20_GetTemp_MatchRom ( ucDs18b20Id ) );
        if((int)DS18B20_GetTemp_MatchRom
(          ucDs18b20Id          )>temp_low&&(int)DS18B20_GetTemp_MatchRom
( ucDs18b20Id )<temp_high)
        {
            printf("t7.txt=\\\"ok\\\"\\xff\\xff\\xff");

            GPIO_ResetBits(GPIOE,GPIO_Pin_12);
        }

        else if(temp_low>(int)DS18B20_GetTemp_MatchRom ( ucDs18b20Id ))
        {
            printf("t7.txt=\\\"low\\\"\\xff\\xff\\xff");

            GPIO_SetBits(GPIOE,GPIO_Pin_12);
        }
    }
}

```

```

    }
    else if((int)DS18B20_GetTemp_MatchRom ( ucDs18b20Id )>=temp_high)
    {
        printf("t7.txt=\\high\\xff\xff\xff");
        GPIO_SetBits(GPIOE,GPIO_Pin_12);
    }
}
}
}

```

## 附录 2 串口屏定时器代码

```

sys0++
n2.val++
if(n2.val>=60)
{
    n2.val=0
    sys0=0
    n4.val++
}
if(n4.val>=60)
{
    n4.val=0
    n3.val++
}
if(n3.val>=24)
{
    n3.val=0
}

```