FinO便携气候监测仪

——智能百叶窗

**队伍名 焊锡长隆**

**队长 周镇峰**

**队员 刘宇帆**

**队员 杨联标**

目录

[一、功能说明和选题背景 **错误!未定义书签。**](#_Toc5629596)

[二、整机电路 3](#_Toc5629597)

[2.1 原理方框图 3](#_Toc5629598)

[2.2 整机电路图 4](#_Toc5629599)

[三、单元电路设计 7](#_Toc5629600)

[3.1 温湿度传感器模块——SHT3x-DIS 7](#_Toc5629601)

[3.1.1 模块单元电路图 7](#_Toc5629602)

[3.1.2 核心元件简介 7](#_Toc5629603)

[7](#_Toc5629604)

[3.1.3 模块功能及工作原理 7](#_Toc5629605)

[3.2 紫外线传感器模块 8](#_Toc5629606)

[3.2.1 模块单元电路图 8](#_Toc5629607)

[3.2.2 核心元件简介 8](#_Toc5629608)

[3.2.3 模块功能及工作原理 8](#_Toc5629609)

[3.3 光照强度传感器模块 9](#_Toc5629610)

[3.2.1 模块单元电路图 9](#_Toc5629611)

[3.2.2 核心元件简介 9](#_Toc5629612)

[3.2.3 模块功能及工作原理 10](#_Toc5629613)

[3.4 OLED显示屏模块 10](#_Toc5629614)

[3.2.1 模块单元电路图 10](#_Toc5629615)

[3.2.2 核心元件简介 10](#_Toc5629616)

[3.2.3 模块功能及工作原理 11](#_Toc5629617)

[3.5 步进电机模块 11](#_Toc5629618)

[3.2.1 模块单元电路图 11](#_Toc5629619)

[3.2.2 核心元件简介 12](#_Toc5629620)

[3.2.3 模块功能及工作原理 12](#_Toc5629621)

[四、程序代码 12](#_Toc5629622)

[主机程序： 12](#_Toc5629623)

[从机程序： 19](#_Toc5629624)

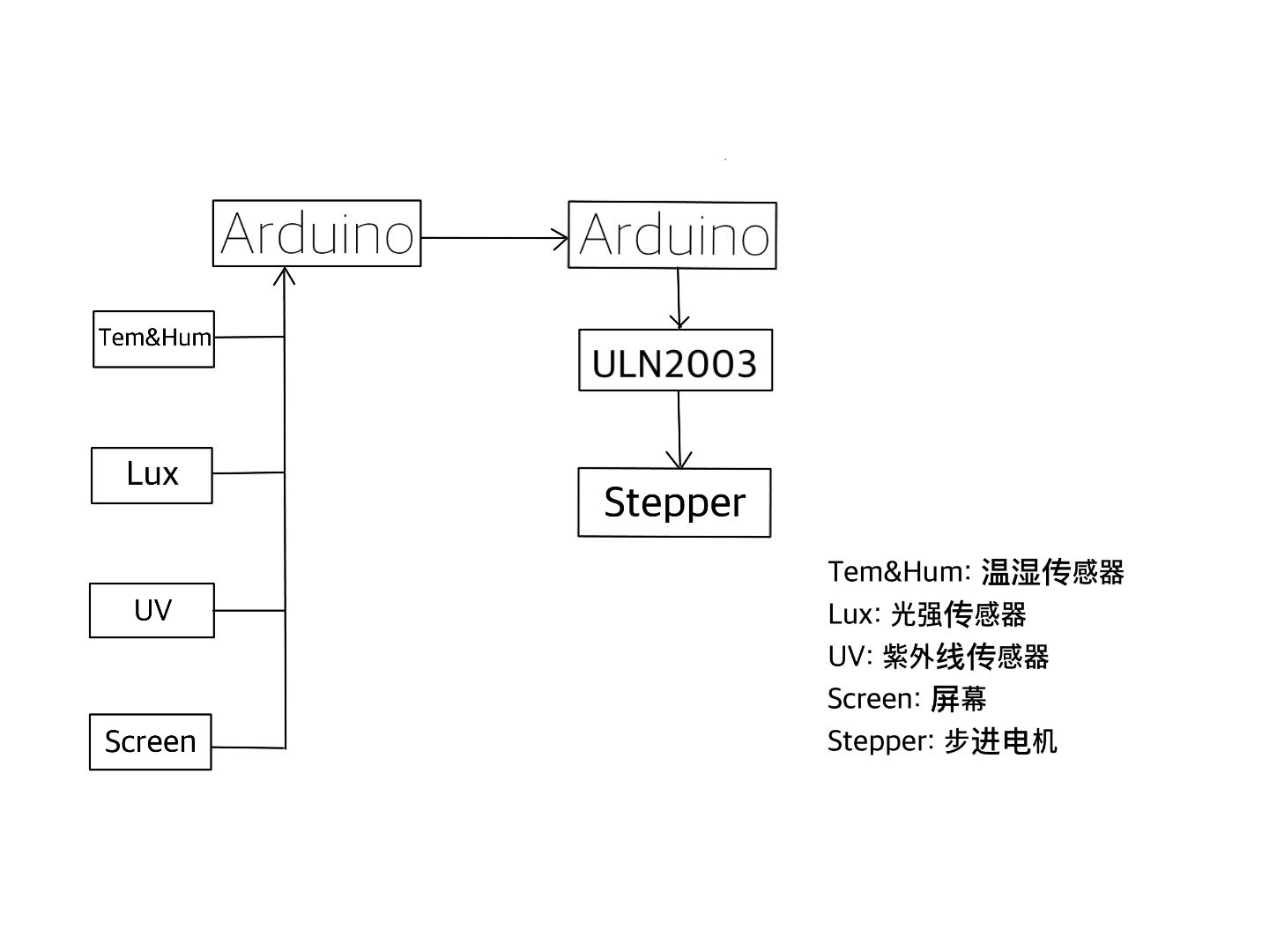
# 功能说明和选题背景

FinO便携气候监测仪（以下简称FinO）可以监测紫外线强度，光照强度、温度和湿度，并通过屏幕显示出来。其体积小，待机长，可随身携带，随时随地得到当前位置的气候信息，弥补天气预报泛而不精的缺点，从而帮助您决定是否要增添衣服、涂抹防晒霜。

另外，FinO还可通过FinO特殊的USB口将数据输出。你只需将FinO插入FinO生态设备，即可享受智能家居。比如在这个项目中，我们打造了一个智能百叶窗，它能根据光照调智能整开合角度，让您的室内舒适明亮。未来，将有更多设备加入FinO生态。

# 二、整机电路

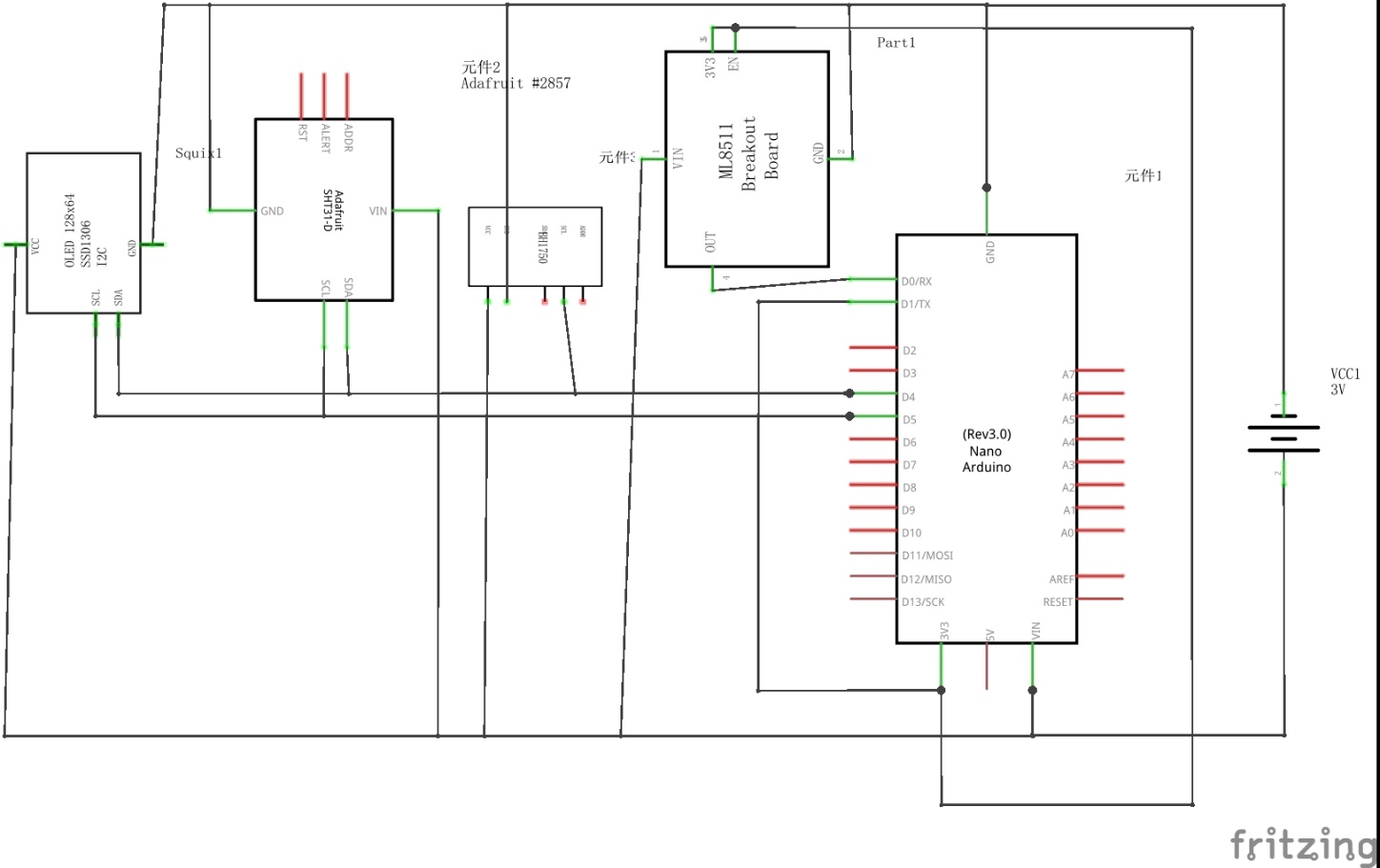
## 2.1 原理方框图



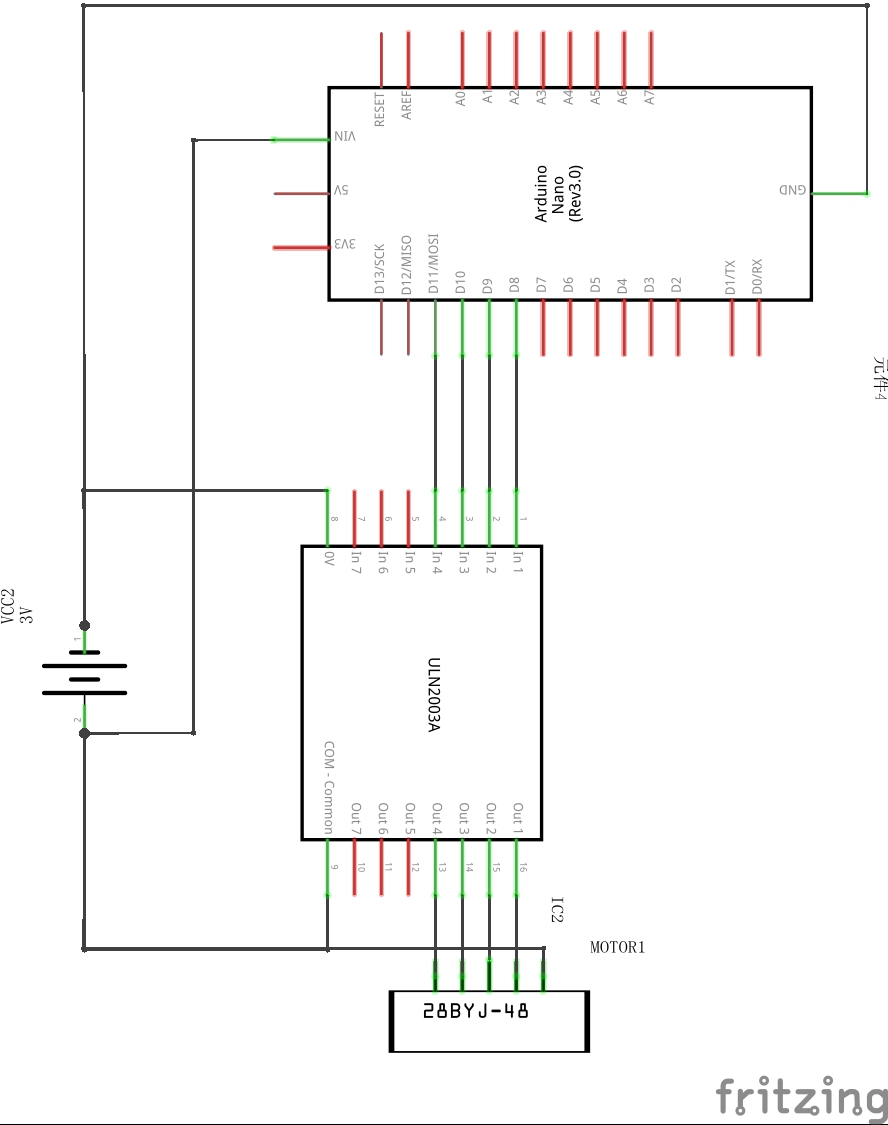
**（图2.1）**

## 2.2 整机电路图

## （图2.2-1）



**(图2.2-2)**

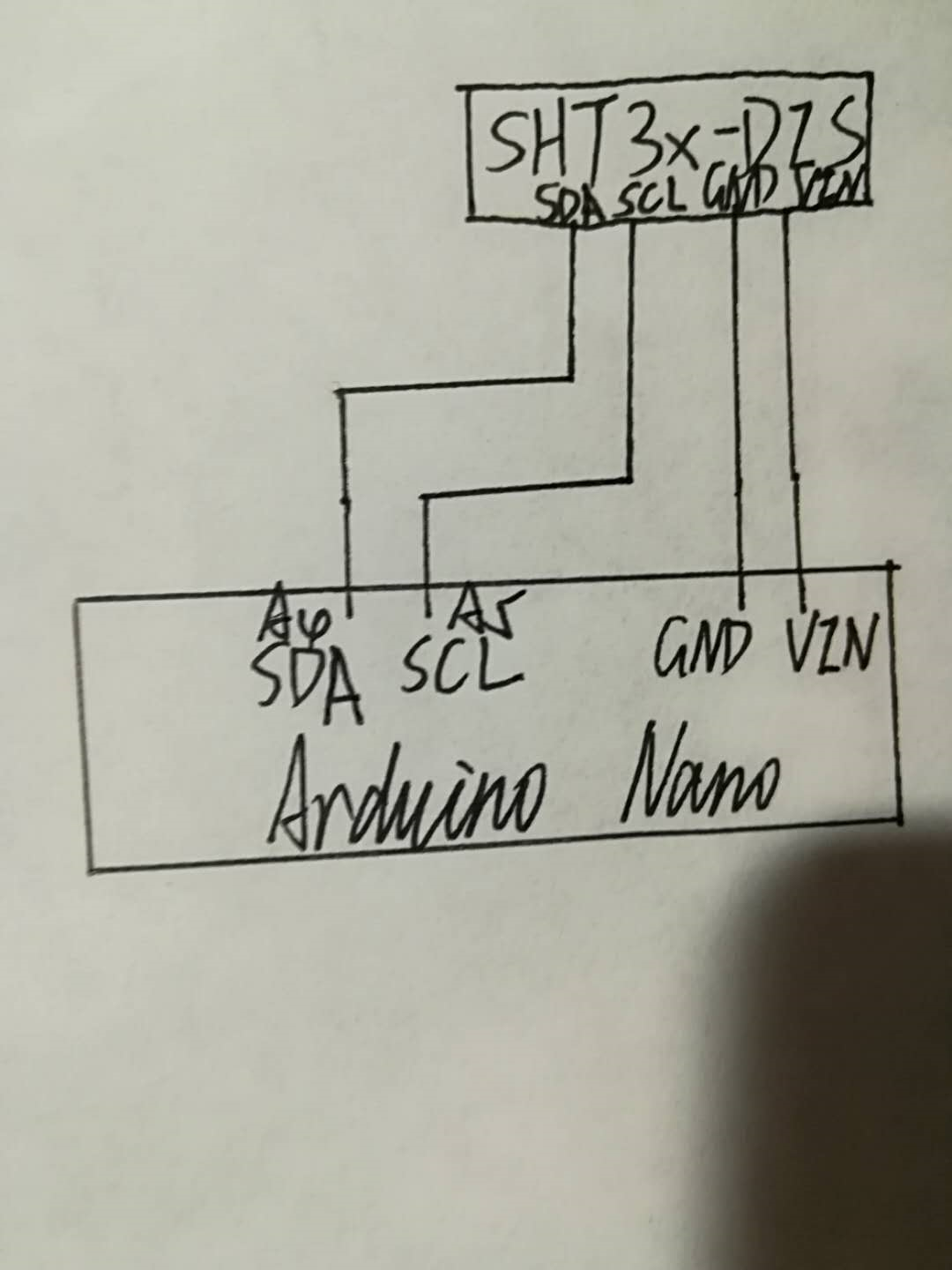


**（图2.2-3）**

# 三、单元电路设计

## 3.1 温湿度传感器模块——SHT3x-DIS

### 3.1.1 模块单元电路图



**（图3.1.1）**

### 3.1.2 核心元件简介

### 

**（图3.1.2）**

SHT3x-DIS是校准刻度线性化的温湿度传感器。它的工作电压从2.4V到5.5V，使用I2C通信，拥有两个供用户选择的地址，精度范围为正负2%RH和正负0.3%摄氏度

### 3.1.3 模块功能及工作原理

温度测量的原理是：采用热电偶的方法，热电偶由两种不同材料的金属丝组成，两种丝材的一端焊接在一起，形成工作端，置于被测温度处；另一端称为自由端，与测量仪表相连，形成一个闭合回路。当工作端与自由端的温度不同时，回路中就会出现热电动势，经过电路的转换将这个电压的变化送到单片机，转化成机器能够识别的信号。

湿度测量的原理是：使用沉积在两个导电电极上的聚胺盐或醋酸纤维聚合物薄膜。当薄膜吸水或失水后，会改变两个电极间的介电常数，进而引起电容器容量的变化，利用外部测量电路可将电容器的容量变化进行捕捉、转化处理，最终在输出端显示成易识别的信号。

## 3.2 紫外线传感器模块

### 3.2.1 模块单元电路图（图3.2.1）

### 3.2.2 核心元件简介

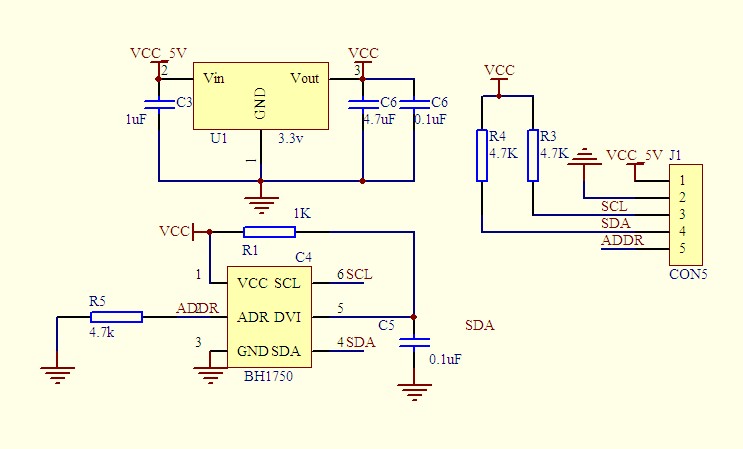
ML8511紫外线传感器可读取室内或户外的紫外线强度，并转化为电压输出。

### 3.2.3 模块功能及工作原理

ML8511内部的光电二极管对UV-A和UV-B两种紫外线敏感（UV-C已被臭氧层吸收，无需测量），光电二极管产生的光电流通过运算放大器转换为电压。在-25℃到75℃的范围内，0~15级紫外线强度线性地输出为1~3伏电压。

## 3.3 光照强度传感器模块

### 3.3.1 模块单元电路图



**（图3.3.1）**

### 3.3.2 核心元件简介

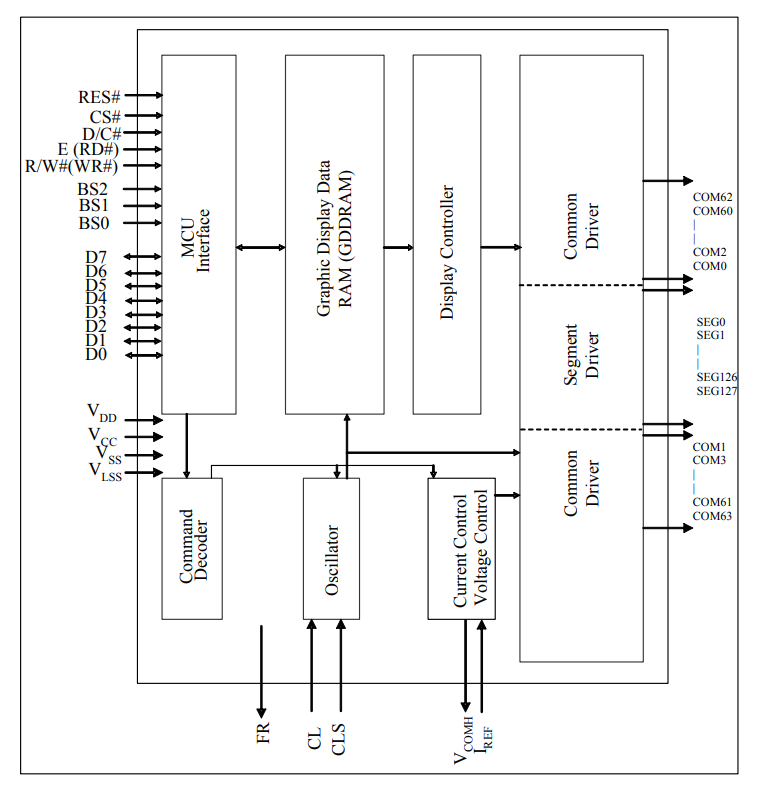
BH1750FVI是一种使用I2C协议的数字环境光传感集成电路。该集成电路最适合于获取手机LCD和键盘背光电源调节所需的环境光数据。拥有极高的感光范围（1-65536流明）。

### 3.2.3 模块功能及工作原理

### BH1750通过光敏二极管测量光强，光强信号经过集成运算放大器处理后输出

## 3.4 OLED显示屏模块

### 3.4.1 模块单元电路图



**（图3.4.1）**

### 3.4.2 核心元件简介

OLED显示屏模块分辨率为128x64，驱动芯片为SSD1306，采用I2C通信。

### 3.4.3 模块功能及工作原理

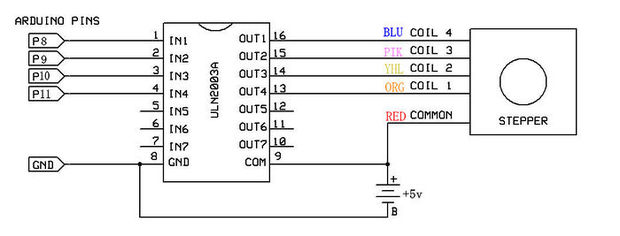
OLED（Organic Light-Emitting Diode）称为有机发光二极管，其原理为：在外界电压的驱动下，由电极注入的电子和空穴在发光层中复合形成处于束缚能级的电子空穴对即激子，激子辐射退激发发出光子，产生可见光。

利用U8g2lib库向SSD1306的缓冲区写入像素，再由SSD1306输出到OLED屏幕上

## 3.5 步进电机模块

### 3.5.1 模块单元电路图

**（图3.5.1-1）**

**(图3.5.1-2)**

### 3.5.2 核心元件简介

28BYJ48为常见的5V五线四相步进电机，利用ULN2003作为驱动芯片。

### 3.5.3 模块功能及工作原理

步进电机用于开关百叶窗，其原理为：按一定时序给定子通电，使其产生一个旋转的磁场，从而带动永磁式转子转动。

由于Arduino不能直接驱动功率较大的步进电机，采用ULN2003驱动。ULN2003的原理为：输入通过达林顿三极管控制反相输出。

# 四、程序代码

## 主机程序：

**#include<Arduino.h>**

**#include <U8g2lib.h>**

**#include <Wire.h>**

**#include "SHTSensor.h"**

**SHTSensor sht;//温湿度传感器的对象**

**int BH1750address=0x23;//光强传感器地址**

**byte buff[2];**

**int UVOUT = A0; //Output from the sensor**

**int REF\_3V3 = A1; //3.3V power on the Arduino board**

**unsigned int time=0;**

**float Lux,Tem,Hum,Uv,UVMAX=0;**

**U8G2\_SSD1306\_128X64\_NONAME\_F\_HW\_I2C u8g2(U8G2\_R0, /\* reset=\*/ U8X8\_PIN\_NONE);**

**void setup() {**

**Serial.begin(9600);**

**u8g2.setI2CAddress(0x78);//屏幕地址**

**u8g2.begin();**

**Wire.begin();**

**}**

**void loop() {**

**firstDisplay();**

**u8g2.setFont(u8g2\_font\_victoriabold8\_8r);**

**while(1){**

**u8g2.clearBuffer();**

**u8g2.drawFrame(4,4,120,56);**

**u8g2.drawHLine(4,32,120);**

**u8g2.drawVLine(64,4,56); //以上是屏幕初始化**

**float X,Y,Z;**

**int L=0;**

**tem\_hum(X,Y); //读取温湿度X是温度，Y是湿度**

**uv(Z); //读取紫外线指数**

**L=light(); //读取光强**

**Trans(); //传输光强**

**if(UVMAX<Z) //以下是屏幕显示**

**{**

**UVMAX=Z;**

**}**

**u8g2.setCursor(6,13);**

**u8g2.print("Temp:");**

**u8g2.setCursor(6,26);**

**u8g2.print(X);**

**u8g2.setCursor(66,13);**

**u8g2.print("Hum:");**

**u8g2.setCursor(66,26);**

**u8g2.print(Y);**

**u8g2.setCursor(6,41);**

**u8g2.print("Lux:");**

**u8g2.setCursor(6,54);**

**u8g2.print(L);**

**u8g2.setCursor(66,41);**

**u8g2.print("UV:");**

**u8g2.setCursor(66,54);**

**u8g2.print(UVMAX);**

**u8g2.sendBuffer();**

**delay(2000);**

**}**

**}**

**//HXCL logo**

**void firstDisplay(){**

**u8g2.clear();**

**u8g2.setFont(u8g2\_font\_maniac\_tr);**

**u8g2.drawBox(4,4,120,56);**

**u8g2.setDrawColor(0);**

**u8g2.setFontMode(1);**

**u8g2.drawStr(25,40,"HXCL");**

**u8g2.sendBuffer();**

**u8g2.setDrawColor(1);**

**delay(1500);**

**}**

**void Trans() //传输函数**

**{**

**Wire.beginTransmission(0x99);//括号内补充从机地址**

**Wire.write(buff [0]);**

**Wire.write(buff [1]);**

**Wire.endTransmission();**

**}**

**void tem\_hum(float &x,float &y) //获取温湿度函数**

**{**

**if (sht.init());**

**sht.setAccuracy(SHTSensor::SHT\_ACCURACY\_MEDIUM);**

**if (sht.readSample())**

**{**

**y=sht.getHumidity();x=sht.getTemperature();**

**}**

**}**

**void uv(float &uvIntensity) //获取紫外线函数**

**{**

**int uvLevel = averageAnalogRead(UVOUT);**

**int refLevel = averageAnalogRead(REF\_3V3);**

**//Use the 3.3V power pin as a reference to get a very accurate output value from sensor**

**float outputVoltage = 3.3 / refLevel \* uvLevel;**

**//uvIntensity = mapfloat(outputVoltage, 0.99, 2.9, 0.0, 15.0);**

**uvIntensity = (outputVoltage - 0.99) \* 15.0 / 1.91;**

**}**

**//Takes an average of readings on a given pin**

**//Returns the average**

**int averageAnalogRead(int pinToRead)**

**{**

**byte numberOfReadings = 16;**

**unsigned int runningValue = 0;**

**for(int x = 0 ; x < numberOfReadings ; x++)**

**runningValue += analogRead(pinToRead);**

**runningValue /= numberOfReadings;**

**return(runningValue);**

**}**

**int light(){ //获取光强函数**

**int i;**

**double lux=0;**

**BH1750\_Init(BH1750address);**

**if(2==BH1750\_Read(BH1750address)){**

**lux=((buff[0]<<8)|buff[1])/1.2; //数据处理**

**}**

**return lux;**

**}**

**int BH1750\_Read(int address){**

**int i=0;**

**Wire.beginTransmission(address);**

**Wire.requestFrom(address, 2);**

**while(Wire.available()){**

**buff[i] = Wire.read(); // receive one byte**

**i++;**

**}**

**Wire.endTransmission();**

**return i;**

**}**

**void BH1750\_Init(int address){**

**Wire.beginTransmission(address);**

**Wire.write(0x10);//1lx reolution 120ms**

**Wire.endTransmission();**

**}**

## 从机程序：

**#include<Arduino.h>**

**#include <Wire.h>**

**#define IN1 8**

**#define IN2 9**

**#define IN3 10**

**#define IN4 11**

**int x[14];**

**int clockwise=0,states;**

**int a=200,b=500;//光强的上下界，初始值为起居室的适宜光强**

**unsigned int mode=1;//光照模式，0，1，2三档，初始值为第一档；**

**void rotate(int,int);**

**unsigned long triggertime=0;**

**union temp**

**{**

**float Float;**

**byte Byte[4];**

**}temp1;**

**void setup()**

**{**

**pinMode(IN1,OUTPUT);**

**pinMode(IN2,OUTPUT);**

**pinMode(IN3,OUTPUT);**

**pinMode(IN4,OUTPUT);**

**Serial.begin(9600);**

**rotate(102,0); //百叶窗启动调零**

**delay(1000);**

**rotate(102,0);**

**delay(1000);**

**rotate(102,0);**

**delay(1000);**

**rotate(102,0);**

**delay(1000);**

**rotate(102,1);**

**Wire.begin(0x99); //括号内补充本机地址，作为从机**

**Wire.onReceive(receiveEvent); //接收数据的函数**

**attachInterrupt(0,changeMode,FALLING); //添加按键中断**

**states=1;**

**}**

**void changeMode() //按键中断修改模式函数**

**{**

**if(millis()-triggertime>100) //消除抖动**

**{**

**triggertime=millis();**

**mode++;**

**mode=mode%3;**

**switch (mode)**

**{**

**case 0:a=0;b=200;break;**

**case 1:a=200;b=500;break;**

**case 2:a=500;b=800;break;**

**}**

**}**

**}**

**void receiveEvent()**

**{**

**int L;**

**while(Wire.available()>0)**

**{**

**for(int i=0;i<14;i++)**

**{**

**x[i]=Wire.read();**

**}**

**L=((x[0]<<8) | x[1])/1.2;**

**}**

**for(int i=2;i<6;i++)**

**{**

**temp1.Byte[i-2]=x[i];**

**}**

**Serial.println(temp1.Float);**

**for(int i=6;i<10;i++)**

**{**

**temp1.Byte[i-6]=x[i];**

**}**

**Serial.println(temp1.Float);**

**for(int i=10;i<14;i++)**

**{**

**temp1.Byte[i-10]=x[i];**

**}**

**Serial.println(temp1.Float);**

**Serial.println(L);**

**clockwise=judge(a,b,L);**

**if(clockwise==1&&states!=0) //states记录电机的当前位置**

**{**

**rotate(50,clockwise);**

**states--;**

**}**

**if(clockwise==0&&states!=10) //states记录电机的当前位置**

**{**

**rotate(50,clockwise);**

**states++;**

**}**

**}**

**int judge(int a,int b,float L) //判断当前光强是否是**

**{ //预定范围**

**if(L>=a&&L<=b) return -1;**

**else if(L<a) return 0;**

**else return 1;**

**}**

**void STEP(int i) //驱动步进电机的函数**

**{**

**switch (i)**

**{**

**case 0:**

**{**

**digitalWrite(IN1, 0);**

**digitalWrite(IN2, 1);**

**digitalWrite(IN3, 1);**

**digitalWrite(IN4, 1);**

**break;**

**}**

**case 1:**

**{**

**digitalWrite(IN1, 0);**

**digitalWrite(IN2, 0);**

**digitalWrite(IN3, 1);**

**digitalWrite(IN4, 1);**

**break;**

**}**

**case 2:**

**{**

**digitalWrite(IN1, 1);**

**digitalWrite(IN2, 0);**

**digitalWrite(IN3, 1);**

**digitalWrite(IN4, 1);**

**break;**

**}**

**case 3:**

**{**

**digitalWrite(IN1, 1);**

**digitalWrite(IN2, 0);**

**digitalWrite(IN3, 0);**

**digitalWrite(IN4, 1);**

**break;**

**}**

**case 4:**

**{**

**digitalWrite(IN1, 1);**

**digitalWrite(IN2, 1);**

**digitalWrite(IN3, 0);**

**digitalWrite(IN4, 1);**

**break;**

**}**

**case 5:**

**{**

**digitalWrite(IN1, 1);**

**digitalWrite(IN2, 1);**

**digitalWrite(IN3, 0);**

**digitalWrite(IN4, 0);**

**}**

**case 6:**

**{**

**digitalWrite(IN1, 1);**

**digitalWrite(IN2, 1);**

**digitalWrite(IN3, 1);**

**digitalWrite(IN4, 0);**

**break;**

**}**

**case 7:**

**{**

**digitalWrite(IN1, 0);**

**digitalWrite(IN2, 1);**

**digitalWrite(IN3, 1);**

**digitalWrite(IN4, 0);**

**break;**

**}**

**case 8:**

**{**

**digitalWrite(IN1, 0);**

**digitalWrite(IN2, 0);**

**digitalWrite(IN3, 0);**

**digitalWrite(IN4, 0);**

**break;**

**}**

**}**

**}**

**void rotate(int times, int clockwise=1)**

**{**

**//times为多少次8相，times=512对应一周。**

**//clockwise是标记顺逆时针，0为顺时针转动，1为逆时针转动**

**int j=times;**

**while(j--)**

**{**

**if(clockwise==0)**

**{**

**for(int i=0;i<8;++i)**

**{**

**STEP(i);**

**delay(2);**

**}**

**STEP(8);**

**delay(1);**

**}**

**if(clockwise==1)**

**{**

**for(int i=7;i>=0;--i)**

**{**

**STEP(i);**

**delay(2);**

**}**

**STEP(8);**

**delay(1);**

**}**

**}**

**}**

**void loop()**

**{**

**STEP(8);**

**delay(1000);**

**}**