

# Arduino模拟半自动晾晒系统研究记录

马曜崢行 17307110206

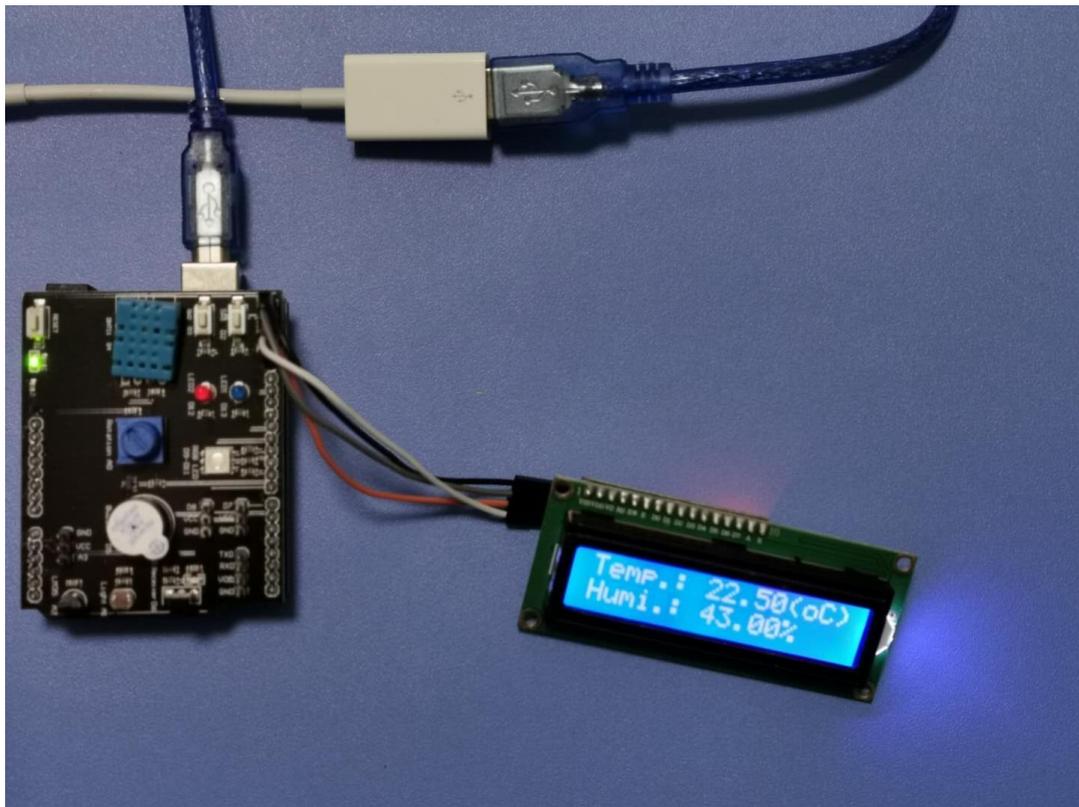
## 一、课题设计：

半自动晾晒系统应有三部分组成。旋转部分，及时旋转晾晒物品，使得晾晒物品正对光源，以达到最大晾晒效果。降雨预警部分，当检测到雨水时，及时报警，提醒收回晾晒物品。环境监测部分，监测周围环境的温度、湿度、气压等信息，并预测降雨概率，以提供信息作为参考是否进行晾晒。

## 二、实验过程：

第一周：

购买得到Arduino组件，进行温度计项目。Arduino连线连接如下：



使用9合1拓展板当中的DHT11温度湿度传感器，LCD1602显示器。分别引用DHT库与LiquidCrystal\_I2C库后，使DHT11与LCD1602工作。代码如下：

## Thermometer\_

```
#include <DHT.h> //引用DHT库
#include <LiquidCrystal_I2C.h> //引用I2C库

#define DHTPIN 4 //DHT11引脚D4
#define DHTTYPE DHT11 //DHT类型

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,16,2); //端口号0x27, 显示模式 16x2

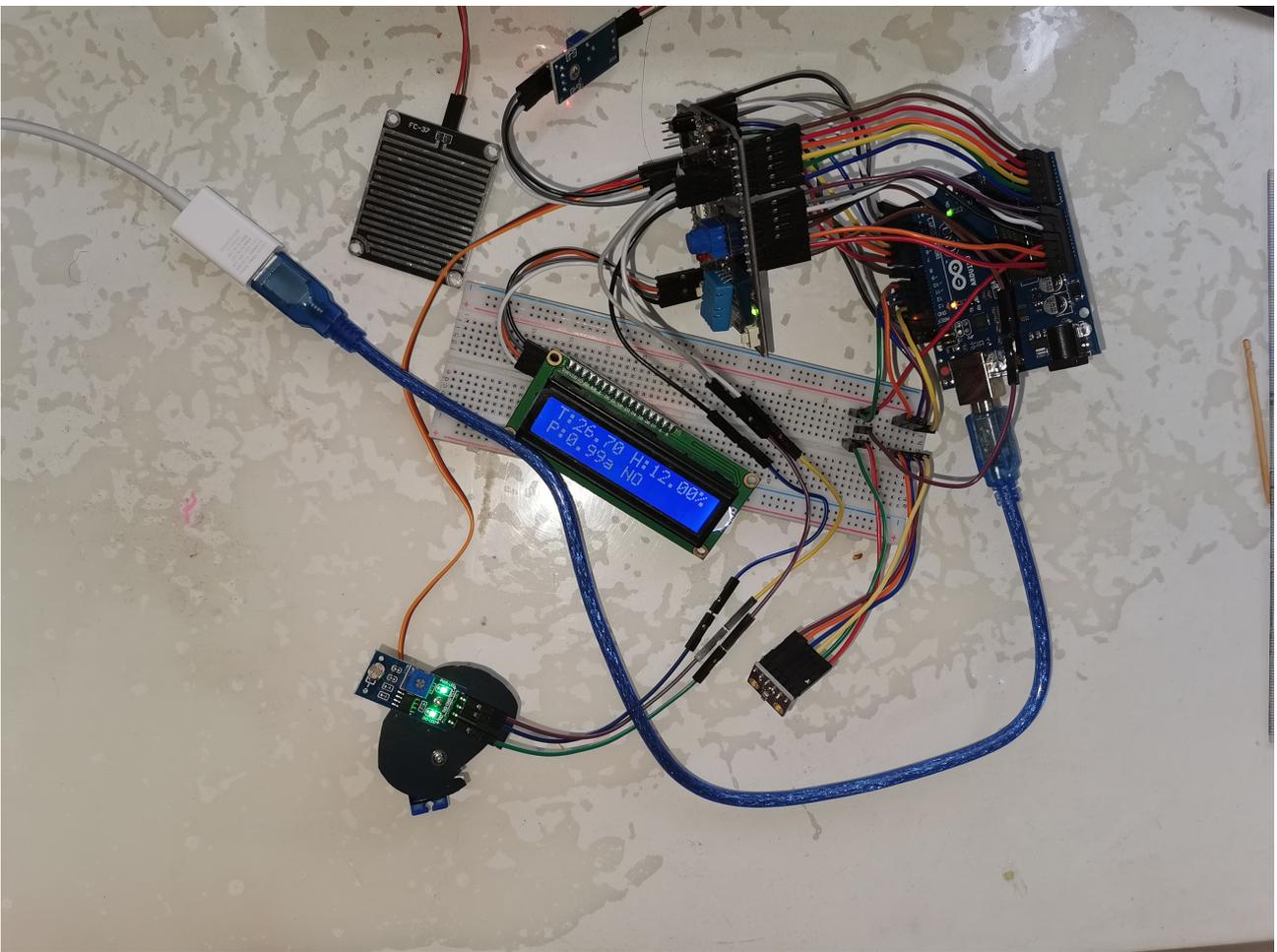
void setup() {
  lcd.init(); // 初始化LCD
  lcd.backlight(); //设置LCD背景等亮
  dht.begin();
}

void loop() {
  float humidity, temperature;
  humidity = dht.readHumidity(); //读取湿度
  temperature = dht.readTemperature(); //读取温度
  lcd.setCursor(0,0); //设置显示指针在第一行
  lcd.print("Temp.: "); //输出温度
  lcd.print(temperature);
  lcd.print("oC");
  lcd.setCursor(0,1); //设置指针在第二行
  lcd.print("Humi.: "); //输出湿度
  lcd.print(humidity);
  lcd.print("%");
  delay(1000); //间隔1s
}
```

第二周:

进行Arduino模块的组装。旋转晾晒部分，使用硬纸板制作旋转用的圆盘，将转盘固定在SG90舵机上，利用舵机悬臂的旋转带动圆盘的转动，同时将亮度传感器固定在圆盘上，利用圆盘旋转测量不同指向的亮度。降雨预警部分，连接FC37雨滴监测器、蜂鸣器在主板上，利用水滴在雨滴监测器上产生的高低电平来控制蜂鸣器的开关。环境监测部分，连接DHT11温度湿度传感器、GY-BMP280大气压强传感器、LCD1602显示器到主板上，通过DHT11温度湿度传感器进行温度湿度测量，GY-BMP280大气压强传感器进行高度、气压测量，LCD1602显示器进行数据综合显示。

线路连接如下:



第三周:

编写Arduino控制代码, 如图:

```
#include <DHT.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#define BMP_SCK 13
#define BMP_MISO 12
#define BMP_MOSI 11
#define BMP_CS 10
#define DHTPIN 4
#define DHTTYPE DHT11
#define RAIN 8
#define LIGHT 1
#define LIGHT2 3
#define BUZZER 5

Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK);
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,16,2); //端口号0x27, 显示模式 16x2
Servo servo;

int angle = 0;
int light;
```

```

void setup() {
  pinMode(RAIN, INPUT);
  pinMode(BUZZER, OUTPUT);
  servo.attach(7);
  lcd.init();           // 初始化LCD
  lcd.backlight();     // 设置LCD背景等亮
  dht.begin();
  Serial.begin(9600);
  Serial.println(F("BMP280 测试"));
  if (!bmp.begin()) {
    Serial.println(F("找不到对应的传感器"));
    while (1);
  }
  servo.write(0);
  light = analogRead(LIGHT);
}

void loop() {
  int lightTmp;

  //舵机控制部分
  printData();
  angle += 1;
  servo.write(angle);
  lightTmp = analogRead(LIGHT);
  if(lightTmp <= light) {
    light = lightTmp;
  } else {
    angle -- 1;
    servo.write(angle);
  }

  delay(1000);
}

void printData() {
  float humidity, temperature, pressure, altitude;
  int rain;
  long tmp;
  String pressureString;
  humidity = dht.readHumidity();    //读取湿度
  temperature = dht.readTemperature(); //读取温度
  rain = digitalRead(RAIN);        //读取是否降雨
  pressure = bmp.readPressure();    //读取气压
  altitude = bmp.readAltitude(1012.25); //读取海拔数据

  //输出全部数据, 电脑显示器
  Serial.print(F("温度: "));
  Serial.print(temperature);
  Serial.println(" *C");
  Serial.print(F("气压 = "));
  Serial.print(pressure);
  Serial.println(" Pa");
  Serial.print(F("海拔 = "));
  Serial.print(altitude);
  Serial.println(" m");
  Serial.print(F("当前是否降雨: "));
  if(rain==LOW) {
    Serial.println(" YES");
  } else {
    Serial.println(" NO ");
  }
  Serial.println();

  //格式化气压输出, 显示小数点后两位
  pressureString = "P:";
  tmp = pressure/100000;
  pressureString += tmp;
  pressureString += ".";
  pressure = pressure-tmp*100000;
  tmp = pressure/10000;
  pressureString += tmp;
}

```

```

pressureString += ".";
pressure = pressure-tmp*100000;
tmp = pressure/10000;
pressureString += tmp;
tmp = (pressure-tmp*10000)/1000;
pressureString += tmp;
pressureString += "a";

//主要数据输出部分, LCD显示器
lcd.setCursor(0,0);
lcd.print("T:");           //输出温度
lcd.print(temperature);
lcd.print(" H:");         //输出湿度
lcd.print(humidity);
lcd.print("%");
lcd.setCursor(0,1);
lcd.print(pressureString); //输出气压

//降雨警报部分
if(rain==LOW) {
  lcd.print(" YES");
  digitalWrite(BUZZER, HIGH);
} else {
  lcd.print(" NO ");
  digitalWrite(BUZZER, LOW);
}
}

```

### 三、结果：

完成以上设置之后，系统能够每隔1s监测一次环境信息（温度、湿度、气压、光照强度、是否降雨），并使舵机旋转1度，如果得到的光照强度大于等于上个角度光照强度，则保存这个角度转变，如果光照强度小于上个光照强度，则旋转回到上个角度，这样能够使舵机逐渐旋转到光照强度极大值位置。

### 四、分析与改进：

模拟系统与最初预想相差降雨概率预测部分，该部分由TensorFlow Lite完成。可以通过在电脑端使用TensorFlow训练模型后，使用xxd程序将模型转化为C数组文件，最后使用EloquentTinyML库在Arduino使用模型。在进行该过程的过程中，发现EloquentTinyML库所使用的<cstdarg>、<cstdlib>、<utility>等头文件不被Clang所支持，进一步修改EloquentTinyML库中的src/tensorflow/lite/core/api/error\_reporter.cpp与src/flatbuffers/bast.h中的头文件引用，可修改<cstdarg>为<stdarg.h>、<cstdlib>为<stdlib.h>，但<utility>未能找

到替代头文件。由于多方求证后发现无法更改Arduino IDE的默认C++编译器，最终无法完成EloquentTinyML库的编译工作。

建议：如果要使用EloquentTinyML等库进行TensorFlow等应用，尽量避免使用MacOS操作系统，而使用Linux操作系统。