

# Arduino课题研究记录

16302010009 张皓通

## 一、课题设计

Arduino是一款便捷灵活、方便上手的开源电子原型平台，是初学者上手开发板的最佳选择。本实验主要通过各类传感器将外界环境参数传入Arduino，再根据我们设置的程序作出反馈如将示数显示在液晶屏、控制舵机转动等。

本次实验主要使用到了温度传感器与光学传感器，制作出了一个简易温度计、光度计和基于光敏的自动感应开关。实验装置为Arduino UNO、电脑以及IDE开发环境、9合一多功能扩展版、LCD1602A配I2C接口、SG90舵机、面包板以及导线等。

## 二、实验过程与分析

首先安装Arduino的驱动和开发环境，在LED闪烁例程运行成功后，就可以开始愉快的玩耍Arduino了。

### 1.LM35温度计

首先最重要的，是链接好各个元件的接线。9合一多功能拓展板是集成了多个传感器的强大拓展板，可以直接安装在Arduino上（当然也可以接线），扩展板上也有直接连接LCD1602A的借口，不过我还是通过导线连接这三个模块的，如下：

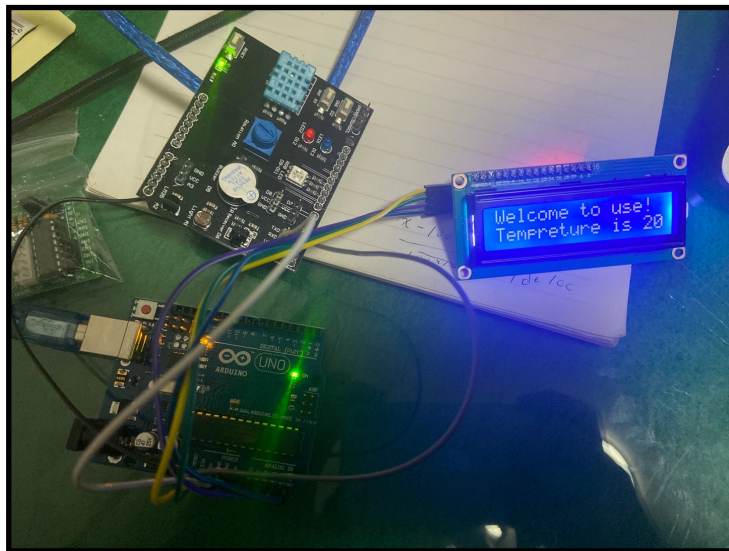


图1-简易温度计的连线

引用Wire库中的analogRead函数即可以直接得到温度计的示数（LM35使用2号引脚），而将示数显示到液晶屏则需要调用<LiquidCrystal\_I2C.h>库的lcd相关函数，每一秒刷新一次。

LM35精度：Arduino UNO 为 5V供电电压，而传感器只能输出0~1023的正整数值，因此根据电压转换公式，精度大约为0.5摄氏度。

## 2.光度计

与简易温度计原理类似，但这次感知的是光的强度。与前面中不同的是，这次我们直接将9合一拓展板安装在Arduino之上，依旧通过LCD1602A来显示实时数据，如下图：

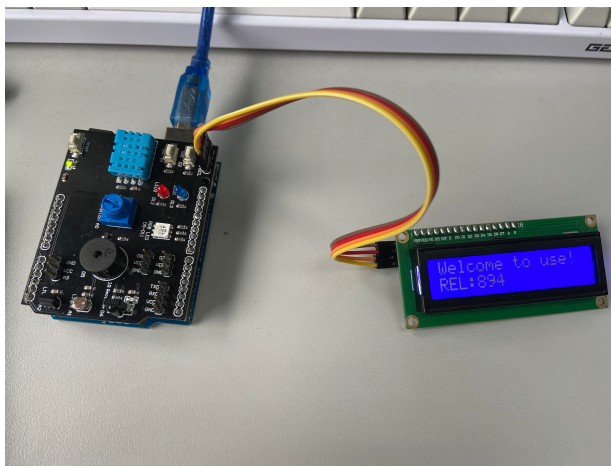


图2-简易光度计

该光度计较为灵敏，与光强呈一次函数线性关系，校准后可作为光学实验如光的衍射实验的光度计。

## 3.光敏自动感应开关

基于以上简单的光度计，我进一步设计出了基于光敏驱动舵机的感应开关。其算法逻辑如下：

- ① 连接舵机，舵机初始化到特定角度。
- ② 初始化LCD1602A液晶屏并显示当前光强，待稳定后记录。
- ③ 循环：当光强下降到稳定值的40%时，舵机转动，使得盖子打开，持续10秒。
- ④ 恢复到原始状态，继续进入循环，监督光强数值。

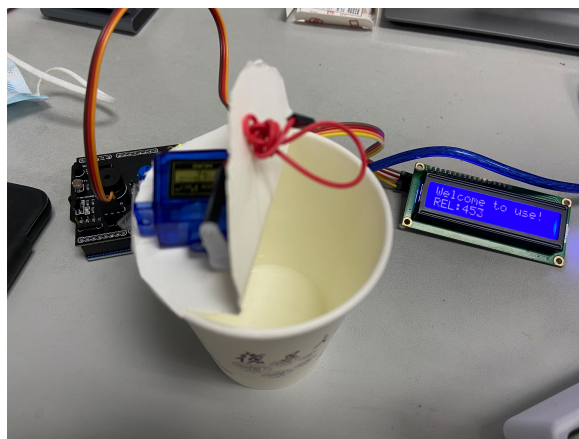


图3-光感开关的制作

---

### 三、实验结果

通过本次实验，我学习了Arduino及其配套传感器、液晶屏等组件的使用方式，运行了数个程序，并大致估算了温度传感器精度。

---

### 附录

#### 1.LM35温度计代码

```
#INCLUDE <WIRE.H>
#include <LIQUIDCRYSTAL_I2C.H> //引用I2C库
LIQUIDCRYSTAL_I2C LCD(0X27,16,2);

VOID SETUP()
{
  SERIAL.BEGIN(9600);
  LCD.INIT();          // 初始化LCD
  LCD.BACKLIGHT();    //设置LCD背景等亮
  LCD.PRINT("WELCOME TO USE!");
  DELAY(1000);
  LCD.CLEAR();
}

VOID LOOP()
{
  INT VAL;
  INT DAT;
  VAL=ANALOGREAD(2);
  DAT=(125*VAL)>>8;
  LCD.SETCURSOR(0,0);    //设置显示指针
  LCD.PRINT("WELCOME TO USE!");//输出字符到LCD1602上
  LCD.SETCURSOR(0,1);
  LCD.PRINT("TEMPRETURE IS ");
  LCD.PRINT(DAT);
  DELAY(1000);
}
```

#### 2.光度计代码

```
#INCLUDE <WIRE.H>
#include <LIQUIDCRYSTAL_I2C.H> //引用I2C库
LIQUIDCRYSTAL_I2C LCD(0X27,16,2);

VOID SETUP(){
  SERIAL.BEGIN(9600);
  LCD.INIT();          // 初始化LCD
  LCD.BACKLIGHT();    //设置LCD背景等亮
  LCD.PRINT("WELCOME TO USE!");
  DELAY(1000);
  LCD.CLEAR();
}

VOID LOOP() {
  INT VAL;
  VAL=ANALOGREAD(1);
```

```

SERIAL.PRINTLN(VAL,DEC);
  LCD.SETCURSOR(0,0);          //设置显示指针
  LCD.PRINT("WELCOME TO USE!");//输出字符到LCD1602上
  LCD.SETCURSOR(0,1);
  LCD.PRINT("REL:");
  LCD.PRINT(VAL);
  DELAY(100);
}

```

### 3.舵机代码

```

#include <SERVO.H>
#include <WIRE.H>
#include <LIQUIDCRYSTAL_I2C.H> //引用I2C库
LIQUIDCRYSTAL_I2C LCD(0X27,16,2);
SERVO MYSERVO; // 建立SERVO物件，控制舵机

```

```

VOID SETUP()
{
  MYSERVO.ATTACH(7); // 连接7号引线
  MYSERVO.WRITE(180);
  SERIAL.BEGIN(9600);
  LCD.INIT();          // 初始化LCD
  LCD.BACKLIGHT();    //设置LCD背景等亮
  LCD.PRINT("WELCOME TO USE!");
  DELAY(1000);
  LCD.CLEAR();
}

```

```

VOID LOOP()
{
  INT VAL;
  VAL=ANALOGREAD(1);
  SERIAL.PRINTLN(VAL,DEC);
  LCD.SETCURSOR(0,0);          //设置显示指针
  LCD.PRINT("WELCOME TO USE!");//输出字符到LCD1602上
  LCD.SETCURSOR(0,1);
  LCD.PRINT("REL:");
  LCD.PRINT(VAL);
  DELAY(100);
  IF(VAL < 200){
    FOR(INT I = 180; I >= 0; I-=1){
      MYSERVO.WRITE(I);// 使用WRITE，传入角度
      DELAY(5);
    }
    DELAY(5000);
    FOR(INT I = 0; I <= 180; I+=1){
      MYSERVO.WRITE(I); // 使用WRITE，传出角度
      DELAY(5);
    }
  }
}

```