**云南大学软件学院**

**实 验 报 告**

课程： 物联网技术 任课教师： 陈清毅 实验指导教师（签名）：

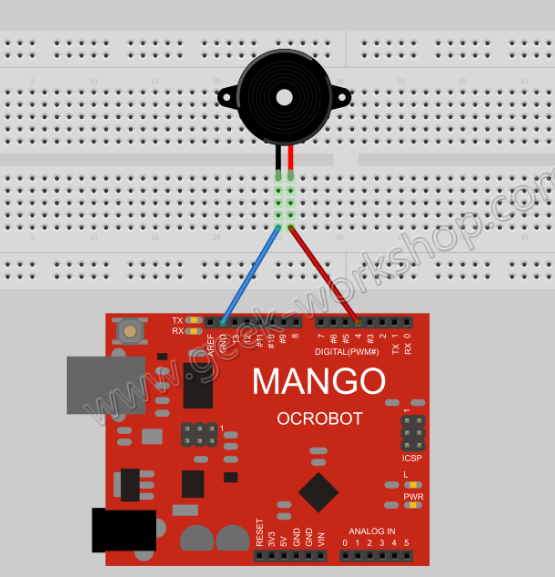
**实验五 Arduino 综合传感器实验**

1. **实验目的**
2. 熟练掌握Arduino程序结构
3. 熟悉通过Arduino 控制不同类型传77感器；

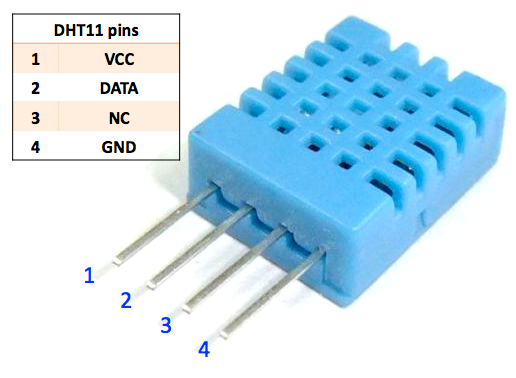
。

**二、实验内容**

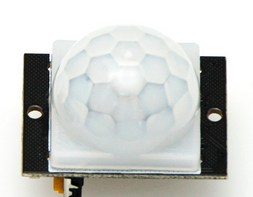
1. **根据持有的传感器类型，下列5个实验任选3个完成即可。**
2. **蜂鸣器实验**
   1. 要使用蜂鸣器。
   2. 通过输入控制蜂鸣器发出不同的声音。



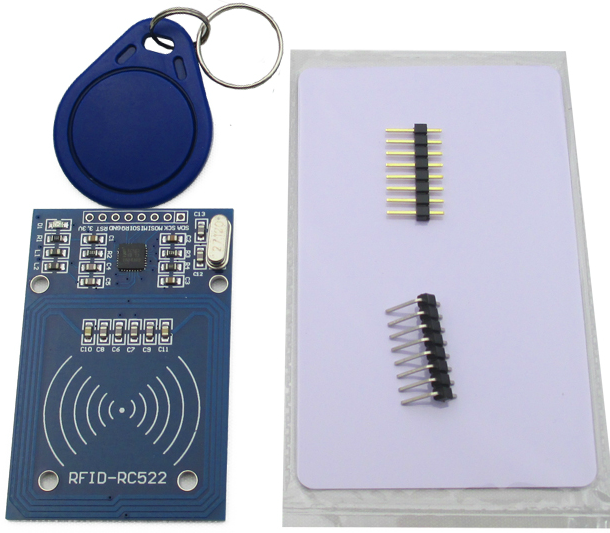
1. **温湿度传感器**
   1. 连接温湿度传感器。
   2. 读取温湿度，并通过串口打印（或LCD）显示结果。
   3. 没有数字温湿度传感器的同学，可以分别使用独立的温度、湿度传感器。



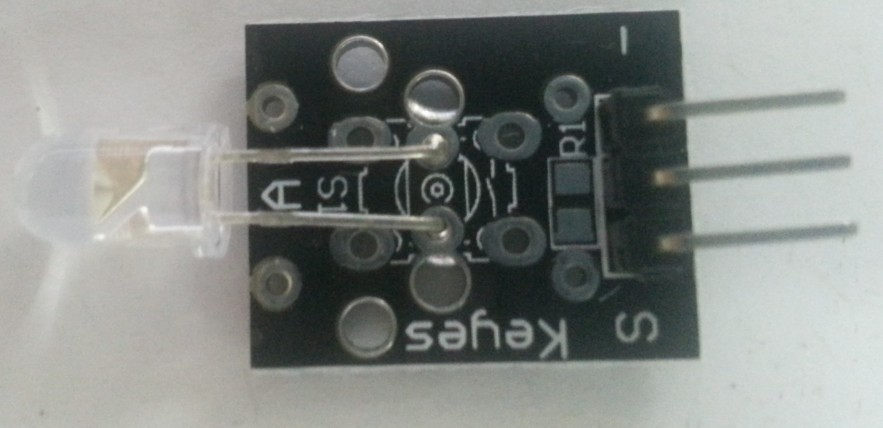
1. **热红外人体感应器**
   1. 连接人体红外传感器。
   2. 连接一个LED灯
   3. 检测到人体时，LED闪烁报警。

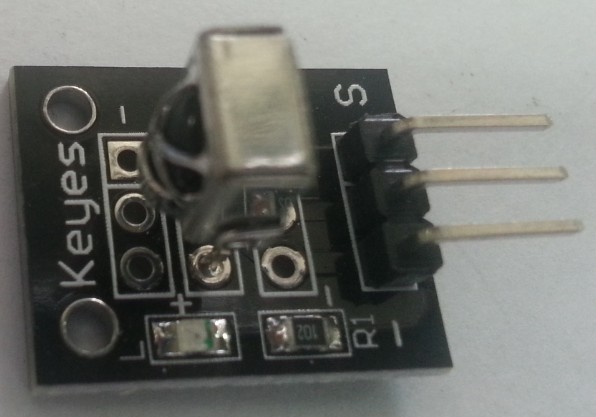


1. **RFID读卡器**
   1. 连接RFID读卡器，可以写入并读取信息。



1. **红外发射+接收传感器**
   1. **使用一组发射，接收传感器**
   2. **通过标准库实现发射和接收数据3**





**三、****实验要求**

（1）完成实验内容，源码作为实验报告附件一起打为一个压缩包提供。该压缩包要包含实验报告、代码文件。

（2）关键部分要求有注释，注释量不低于20%

（3）页面要求独立完成，不得抄袭代码。

（4）提交实验报告：报告电子版纸质报告（代码）下次上课前提交，打包发送到邮箱hexen@163.com。

**四、关键实验步骤（请粘贴关键步骤、代码、实验结果）**

**2.代码1：**

int speakerPin = 7;//I/O接7号数字接口，GND接地，VCC接5V

int length = 15; // the number of notes

char notes[] = "ccggaagffeeddc "; // a space represents a rest

int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };

int tempo = 300;

void playTone(int tone, int duration) {

for (long i = 0; i < duration \* 1000L; i += tone \* 2) {

digitalWrite(speakerPin, HIGH);

delayMicroseconds(tone);

digitalWrite(speakerPin, LOW);

delayMicroseconds(tone);

}

}

void playNote(char note, int duration) {

char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };

int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };

// play the tone corresponding to the note name

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

if (names[i] == note) {

playTone(tones[i], duration);

}

}

}

void setup() {

pinMode(speakerPin, OUTPUT);

}

void loop() {

for (int i = 0; i < length; i++) {

if (notes[i] == ' ') {

delay(beats[i] \* tempo); // rest

} else {

playNote(notes[i], beats[i] \* tempo);

}

// pause between notes

delay(tempo / 2);

}

}

**实验结果：arduino控制蜂鸣器播放《小星星》**

**代码2：**

int tonepin = 6;//蜂鸣器I/O接6号接口

int ledp = 10;//led接10号针脚

#define NTD0 -1

#define NTD1 294

#define NTD2 330

#define NTD3 350

#define NTD4 393

#define NTD5 441

#define NTD6 495

#define NTD7 556

#define NTDL1 147

#define NTDL2 165

#define NTDL3 175

#define NTDL4 196

#define NTDL5 221

#define NTDL6 248

#define NTDL7 278

#define NTDH1 589

#define NTDH2 661

#define NTDH3 700

#define NTDH4 786

#define NTDH5 882

#define NTDH6 990

#define NTDH7 112

//c pinlv

#define WHOLE 1

#define HALF 0.5

#define QUARTER 0.25

#define EIGHTH 0.25

#define SIXTEENTH 0.625

int tune[] = //根据简谱列出各频率

{

NTD3, NTD3, NTD4, NTD5,

NTD5, NTD4, NTD3, NTD2,

NTD1, NTD1, NTD2, NTD3,

NTD3, NTD2, NTD2,

NTD3, NTD3, NTD4, NTD5,

NTD5, NTD4, NTD3, NTD2,

NTD1, NTD1, NTD2, NTD3,

NTD2, NTD1, NTD1,

NTD2, NTD2, NTD3, NTD1,

NTD2, NTD3, NTD4, NTD3, NTD1,

NTD2, NTD3, NTD4, NTD3, NTD2,

NTD1, NTD2, NTDL5, NTD0,

NTD3, NTD3, NTD4, NTD5,

NTD5, NTD4, NTD3, NTD4, NTD2,

NTD1, NTD1, NTD2, NTD3,

NTD2, NTD1, NTD1

};

//根据简谱列出各节拍

float durt[] =

{

1, 1, 1, 1,

1, 1, 1, 1,

1, 1, 1, 1,

1 + 0.5, 0.5, 1 + 1,

1, 1, 1, 1,

1, 1, 1, 1,

1, 1, 1, 1,

1 + 0.5, 0.5, 1 + 1,

1, 1, 1, 1,

1, 0.5, 0.5, 1, 1,

1, 0.5, 0.5, 1, 1,

1, 1, 1, 1,

1, 1, 1, 1,

1, 1, 1, 0.5, 0.5,

1, 1, 1, 1,

1 + 0.5, 0.5, 1 + 1,

};

int length;

void setup()

{

pinMode(tonepin, OUTPUT);//设置数字IO脚模式，OUTPUT为输出

pinMode(ledp, OUTPUT);

length = sizeof(tune) / sizeof(tune[0]);//计算长度

}

void loop()

{

for (int x = 0; x < length; x++)//输出一个频率的声音

{

tone(tonepin, tune[x]);

digitalWrite(ledp, HIGH);//发声音

delay(400 \* durt[x]); ////这里用来根据节拍调节延时，这里将500分为了400和100，分别控制led的开与关，对于蜂鸣器来说依然是500.

digitalWrite(ledp, LOW);//不发声音

delay(100 \* durt[x]);

noTone(tonepin);

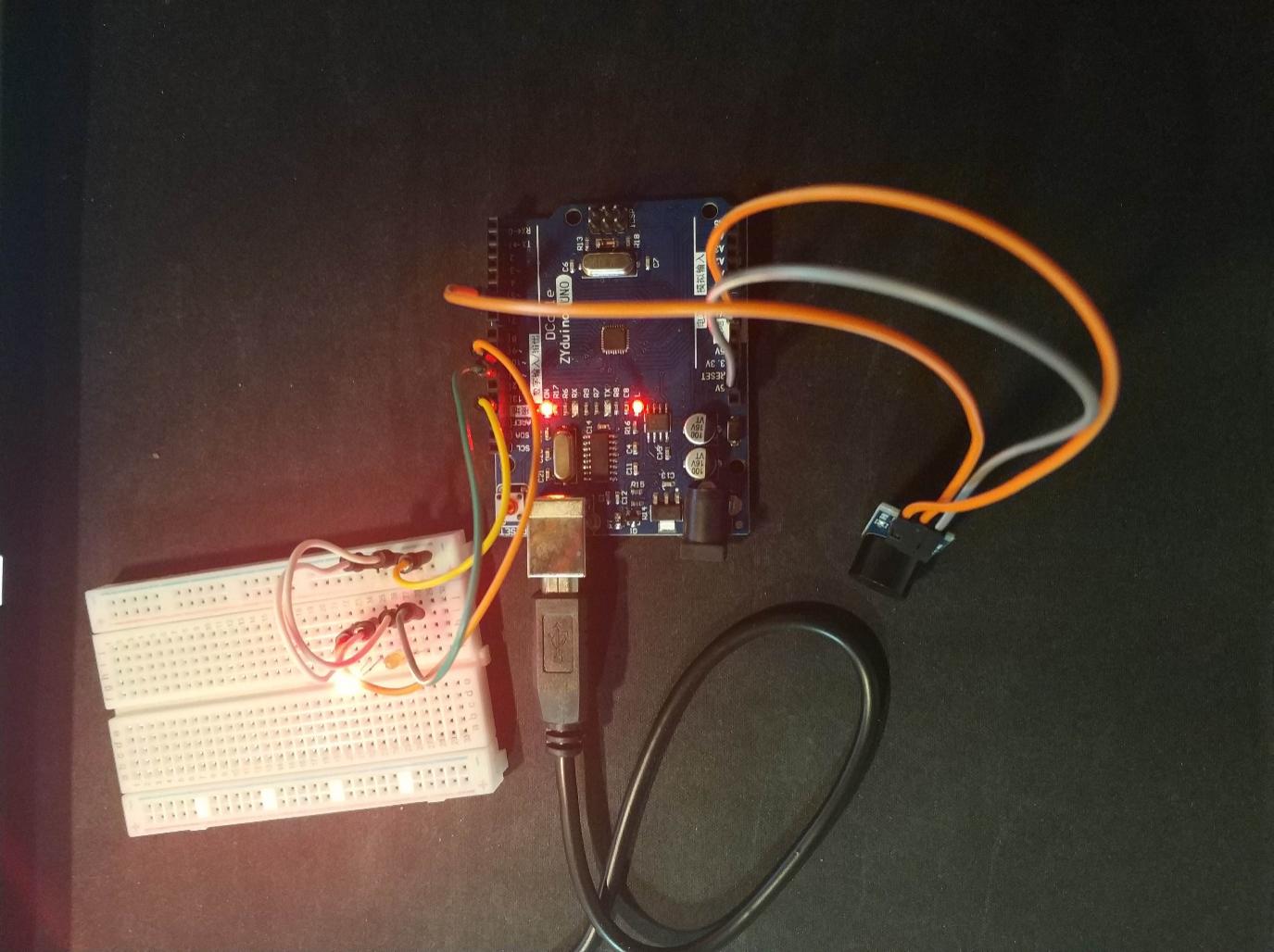
}

delay(2000);

}

**实验结果：arduino控制蜂鸣器播放《欢乐颂》，**LED**灯伴随音乐进行闪烁。**

**电路图：**

****

**3.代码：**

#include<OneWire.h>

#include<DallasTemperature.h>

#include<Wire.h>

#include<LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);

#define ONE\_WIRE\_BUS 6//温度传感器接PIN6接口

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

sensors.begin();

lcd.init();

lcd.backlight();

}

void loop(void) {

// put your main code here, to run repeatedly:

sensors.requestTemperatures();//发送命令获取温度

lcd.setCursor(0,0);

lcd.print("TemC:");

lcd.print(sensors.getTempCByIndex(0));

Serial.print(sensors.getTempCByIndex(0));//打印温度

Serial.println("C");

delay(1000);

lcd.print(char(223));//在LCD屏幕上打印°C

lcd.print("C");

lcd.setCursor(0,1);

lcd.print("TemF:");

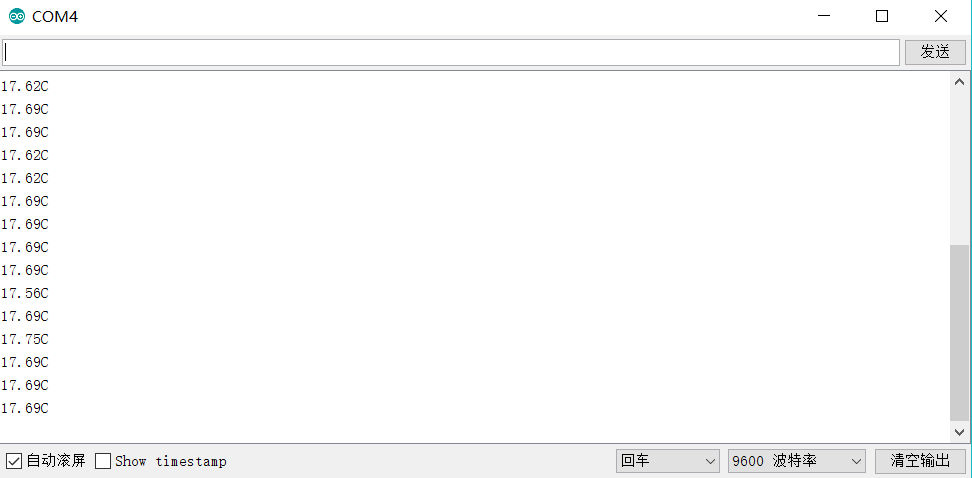
lcd.print(1.8\*sensors.getTempCByIndex(0) + 32.0);//计算华氏温度

lcd.print(char(223));

lcd.print(" F");

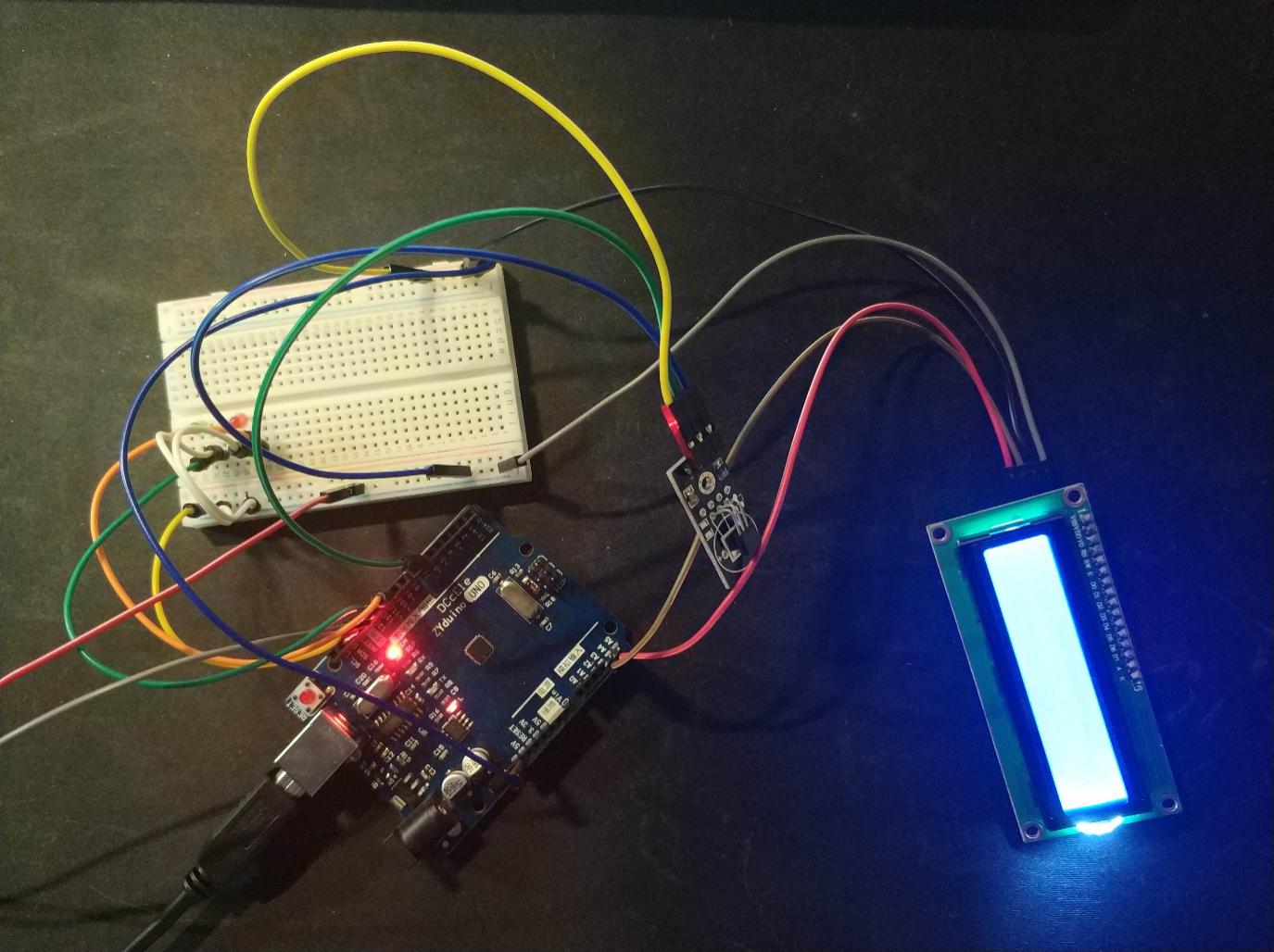
}

**实验结果：**



**在串口打印出当前的温度（没有湿度传感器），在LCD显示屏上显示温度。**

**电路图：**

****

**6. 代码：**

#include <IRremote.h>// 使用IRRemote函数库

const int irReceiverPin = 7;// 红外接收器的 DO 引脚接在 PIN7 接口 定义irReceiverPin变量为PIN7接口

const int ledPin = 13; //RGB灯接在PIN13接口

IRrecv irrecv(irReceiverPin);// 设置irReceiverPin定义的端口为红外信号接收端口

decode\_results results;// 定义results变量为红外结果存放位置

void setup() {

// put your setup code here, to run once:

pinMode(ledPin, OUTPUT);

Serial.begin(9600);

irrecv.enableIRIn();// 启动红外解码

}

void loop() {

// put your main code here, to run repeatedly:

if (irrecv.decode(&results)) { // 解码成功，把数据放入results变量中

Serial.print("irCode:");

Serial.print(results.value, HEX); // 显示红外编码

Serial.print(",bits:");

Serial.println(results.bits);// 显示红外编码位数

irrecv.resume();// 继续等待接收下一组信号

}

delay(600);//延时600毫秒，做一个简单的消抖

if (results.value == 0xFFA25D) {

digitalWrite(ledPin, HIGH);

} else {

digitalWrite(ledPin, LOW);

}

}

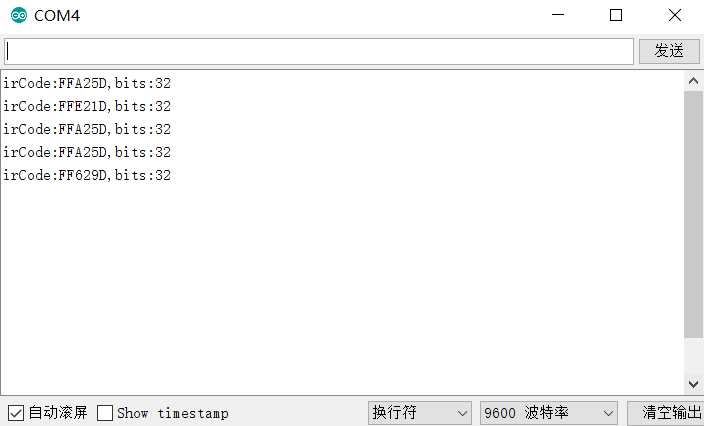
**实验步骤：**

**第1步：建立电路；**

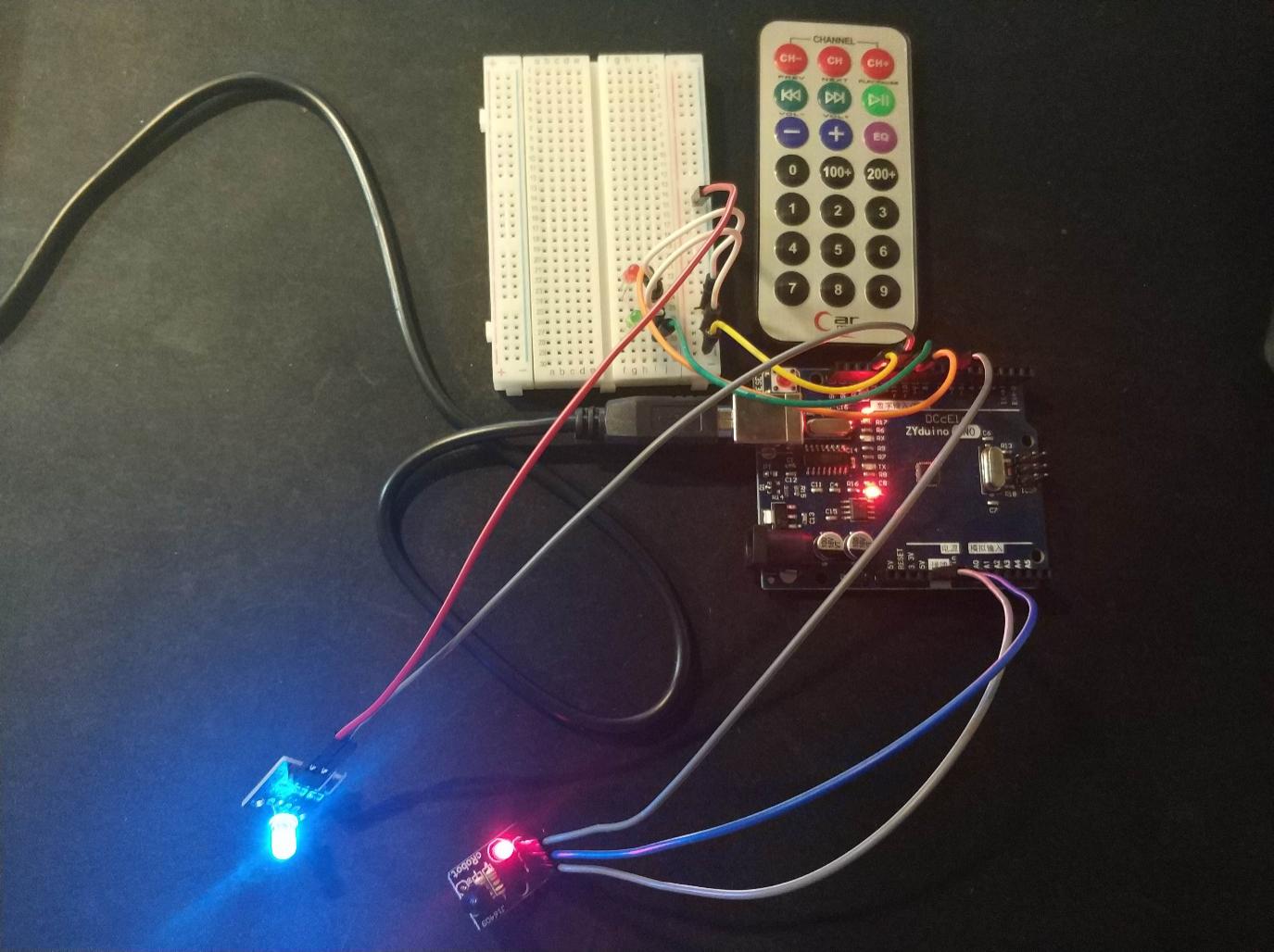
**第二步：编写程序，导入IRRemote函数库并烧录至开发板；**

**第三步：按下遥控器的“CH-”键，连接到第13号针脚的LED灯亮起，按其他任何键LED灯熄灭。**

**实验结果：**



**电路图：**

****