

به نام خدا

گزارش کار آزمایش ششم ریزپردازنده

گروه 7

آریان بوکانی 9731012

وسایل استفاده شده در آزمایش:

- Arduino Mega 2560
- Keypad
- LCD 16*2
- Servo motor
- Button
- LED
- Buzzer

قطعه‌ی جدید استفاده شده در این آزمایش buzzer می‌باشد. این قطعه دو پایه دارد و در صورت دادن پتانسیل به این دو سر، شروع به صدا دادن می‌کند.

برای فعال کردن آن در کد باید از تابع tone استفاده کرد که سه ورودی می‌گیرد. اولی پایه‌ای که بازر به آن وصل است، دومی فرکانس صدای تولید شده و سومی مدت زمانی است که می‌خواهیم بازر صدا تولید کند.

tone(int pinNum, int frequency, int duration);

پی‌نوشت: در گزارش آزمایش قبلی توضیحی در مورد ترانزیستور داده نشد. بنابراین توضیحات مربوط به آن را در این آزمایش آورده‌ام.

در آزمایش قبل از ترانزیستور NPN استفاده شد. این ترانزیستور با قراردادن ماده‌ی نیمه‌هادی نوع P بین دو ماده‌ی نیمه‌هادی نوع N ساخته شده است. این ترانزیستور از سه بخش Base, Emitter و Collector ساخته شده است. دستگاه در حالتی که ولتاژ base از emitter کمتر باشد، خاموش است. و وقتی ولتاژ base بیشتر از emitter می‌شود، روشن می‌شود. در این حالت یک جریان از collector به سمت emitter به حرکت می‌افتد و در حالت کلی داریم:

$$I_{emitter} = I_{collector} + I_{base}$$

در حالت اشباع ترانزیستور می‌توان از مقاومت ترانزیستور از collector به emitter صرف نظر کرد.

```

#include <Servo.h>

#include <LiquidCrystal.h>

#include <Keypad.h>


const byte ROWS = 4;
const byte COLS = 3;

const int closeButton = 13, changePassButton = 10, changeTimerButton = 8;
const int rs = 4, en = 5, d4 = 3, d5 = 2, d6 = 1, d7 = 0;

const int buzzerPin = 19, redBulb = 21, greenBulb = 20, servoPin = 6;
String input = "", password = "12345";
char key;

bool opened = false, shouldBeOpened = false, shouldBeClosed = false;
unsigned int openedFrom;

int mode = 0, maxTime = 20000;


char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte rowPins[ROWS] = { A7, A3, A2, A1 };
byte colPins[COLS] = { A6, A5, A4};

```

```
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  
Servo servo;
```

```
void setup() {  
  pinMode(closeButton, INPUT);  
  pinMode(changePassButton, INPUT);  
  pinMode(changeTimerButton, INPUT);  
  pinMode(redBulb, OUTPUT);  
  pinMode(greenBulb, OUTPUT);  
  lcd.begin(16, 2);  
  lcd.print(input);  
  servo.attach(servoPin, 1000, 2000);  
  servo.write(0);  
}
```

```
void loop() {  
  key = kpd.getKey();  
  if (key != NO_KEY)  
    DetectButton();  
  if(shouldBeOpened){  
    lcd.clear();  
    lcd.print("OPENED...");  
    opened = true;  
    shouldBeOpened = false;
```

```

    openedFrom = millis();
    tone(buzzerPin, 500, 100);
    servo.write(180);
    digitalWrite(greenBulb, HIGH);
    digitalWrite(redBulb, LOW);
} else if (opened && (shouldBeClosed || (millis() - openedFrom >= maxTime))) {
    lcd.clear();
    lcd.print("CLOSED...");
    lcd.setCursor(0, 1);
    lcd.print("Time:" + String(millis() - openedFrom));
    lcd.setCursor(0, 0);
    opened = false;
    shouldBeClosed = false;
    tone(buzzerPin, 500, 100);
    servo.write(0);
    digitalWrite(greenBulb, LOW);
    digitalWrite(redBulb, HIGH);
}

if (opened && (digitalRead(closeButton) == HIGH))
    shouldBeClosed = true;

if (opened && (digitalRead(changePassButton) == HIGH)) {
    mode = 1;
    input = "";
}

if (opened && (digitalRead(changeTimerButton) == HIGH)) {

```

```

    mode = 2;
    input = "";
}
}

bool CheckPassword(){
    String temp = input;
    input = "";
    return temp == password;
}

void DetectButton(){
    lcd.clear();
    if ( key >= '0' && key <= '9' ){
        input += key;
        lcd.print(input);
    }
    else if ( key == '*' ){
        if(mode == 0){
            shouldBeOpened = CheckPassword();
            lcd.clear();
            if(shouldBeOpened)
                lcd.print("Correct");
            else
                lcd.print("Incorrect");
            delay(500);
            lcd.setCursor(0, 0);

```

```

}else if(mode == 1){
    lcd.setCursor(0, 1);
    password = input;
    lcd.print("Set new pass");
    lcd.setCursor(0, 0);
    input = "";
    mode = 0;
}else if(mode == 2){
    lcd.setCursor(0, 1);
    maxTime = input.toInt();
    lcd.print("Set new timer");
    lcd.setCursor(0, 0);
    input = "";
    mode = 0;
}
}
}

```

توضیحات:

در ابتدا کتابخانه‌های مورد نیاز را اضافه کرده و پایه‌های و قطعات را مشخص می‌کنیم.

maxTime حداکثر زمان داده شده برای باز ماندن گاو صندوق است. برای مشخص کردن زمان باز ماندن از تابع millis() استفاده می‌شود. خروجی این تابع برابر تعداد ms های سپری شده نسبت به ابتدای اجرای مدار است. به صورتی که لحظه‌ی باز شدن گاو صندوق این تابع صدا زده می‌شود و مقدار آن در openedFrom ریخته می‌شود. در هر loop چک می‌شود که اگر صندوق باز باشد و اختلاف مقدار (millis() الان و openedFrom بیشتر از maxTime بود، در آن را ببندد.

در این آزمایش از یک عدد mode برای چک کردن حالت فعلی استفاده می‌کنیم. به صورتی mode صفر بیانگر حالت اولیه، mode 1 بیانگر حالت تغییر رمز در صورتی که صندوق باز باشد و mode 2 حالتی است که صندوق باز است و می‌خواهیم حداکثر زمان باز ماندن را تغییر دهیم. هر کدام از این حالت‌ها با سه کلید مشخص می‌شوند.

همچنین باید گفت که برای submit کردن ورودی خود باید کاراکتر * را فشار دهیم.

چند نمونه از اجرا:



