



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر  
پروژه درس ساختار و زبان کامپیوتر

عنوان:

## بازی دایناسور خودکار

نگارش

مهیار افشین مهر - محمد پیام تائبی - علی آقایاری

استاد

دکتر حسین اسدی

بهمن ۱۴۰۱

## فهرست مطالب

۲	۱ شرح کلی پروژه
۲	۱-۱ وسایل مورد نیاز
۳	۲ اتصالات و مراحل ساخت
۳	۱-۲ مدار عادی و بدون LCD
۳	۱-۱-۲ تصاویر مربوط به اتصالات LDR و موتور servo
۴	۲-۲ اتصالات مربوط به LCD
۵	۱-۲-۲ تصاویر مربوط به اتصالات LCD و مدار کلی
۶	۳ شرح کد
۶	۱-۳ متغیرها و کتابخانه‌ها
۷	۲-۳ توابع مربوط به مقادیر نورهای سفید و سیاه
۹	۳-۳ تابع setup
۱۰	۴-۳ تابع loop ، حلقه‌ی اصلی
۱۱	۴ نتیجه گیری
۱۲	آ منابع و مراجع
۱۲	ب مطالب تکمیلی

# ۱ شرح کلی پروژه

در این پروژه بازی دایناسور کروم را به صورت خودکار توسط یک برد arduino و یک LDR و یک موتور servo پیاده سازی کرده ایم. در این پروژه از یک LCD استفاده کرده ایم تا به کمک آن بتوان به کاربر نشان داد که با استفاده از سنسور LDR خود، رنگ های سفید و سیاه را تشخیص دهد و به نوعی سنسور را کالیبره کند. کارکرد کلی به این صورت است که ابتدا LCD اعلام می کند که ۱۰ ثانیه سنسور LDR را روی نور سفید قرار دهیم. سپس ۱۰ ثانیه دیگر سنسور باید روی نور سیاه قرار بگیرد و پس از آن LCD با نوشتن کلمه ی play شروع بازی را اعلام می کند. در ادامه با قرار دادن LDR روی صفحه ی مانیتور هرگاه درخت ها توسط سنسور تشخیص داده شوند، به موتور servo دستور داده می شود تا کلید space فشرده شود و بازی انجام شود.

## ۱-۱ وسایل مورد نیاز

- برد Arduino Uno : این برد یک برد open source است که دارای تعدادی پین ورودی و خروجی دیجیتال و آنالوگ ( ۱۴ پین ) است. همچنین می توان آن را از طریق کابل USB یا باتری ۹ ولتی تغذیه کرد.
- یک عدد موتور servo : یک موتور الکتریکی است که تعدادی مدارات الکتریکی کنار آن قرار می گیرد. موتور الکتریکی وظیفه ی گردش و یا حرکت خطی را دارند و مدارهای الکتریکی دیگر به افزایش دقت آن کمک می کنند.
- یک عدد سنسور LDR : این سنسور نوعی مقاومت وابسته به نور است که از نیمه رساناهایی با مقاومت بالا ساخته شده است. به طوری که مقاومت آن بر اثر شدت نور تابیده شده بر آن کاهش می یابد.
- مقاومت ۱۰ K اهمی
- یک عدد bread board : یک برد مستطیل شکل که از آن برای برقراری ارتباط میان سایر قطعات الکتریکی استفاده می شود.
- یک عدد LCD
- تعدادی سیم

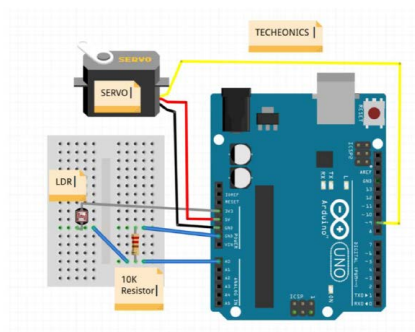
## ۲ اتصالات و مراحل ساخت

در این بخش نحوه اتصالات و مراحل ساخت مدار مورد نیاز را مرور می‌کنیم.

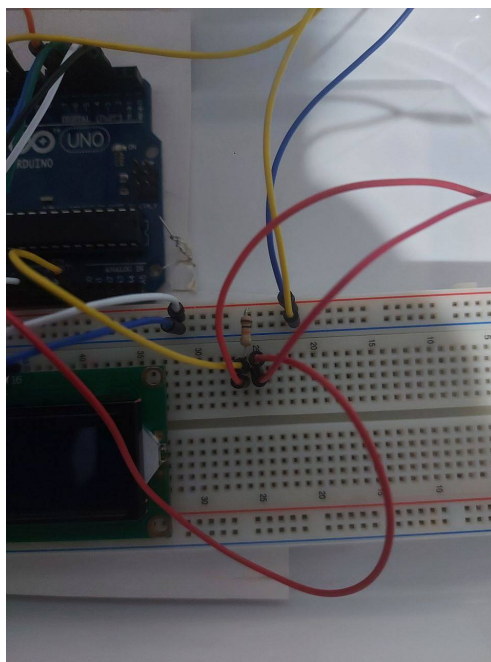
### ۱-۲ مدار عادی و بدون LCD

ابتدا پین‌های موتور servo را به برد arduino وصل می‌کنیم. موتور servo ۳ پین VCC ، GND و PWM دارد که به ترتیب به پین‌های VCC ، GND و D۹ وصل می‌کنیم تا فرکانس‌های مورد نیاز برای موتور servo از طریق پین ۹ منتقل شوند. (توضیحات بیشتر در بخش کد) در ادامه یک پایه‌ی سنسور LDR را به برد مورد و پایه‌ی دیگر آن را به پین ۳V۳ در بخش power برد arduino وصل می‌کنیم. حال نوبت به وصل کردن مقاومت می‌رسد. مقاومت را به برد مورد وصل کرده و یک پایه‌ی آن را با سیم به GND و پایه‌ی دیگر را به همان پایه‌ی LDR که روی برد است وصل می‌کنیم. مدار مورد نیاز کامل خواهد بود.

#### ۱-۱-۲ تصاویر مربوط به اتصالات LDR و موتور servo



شکل ۱: شماتیک کلی مدار LDR و موتور servo



شکل ۲: اتصالات مربوط به LDR و موتور servo

## ۲-۲ اتصالات مربوط به LCD

LCD یک parallel interface است. به این معنا که میکروکنترلر باید تعداد زیادی پین را به طور همزمان کنترل کند تا بتواند خروجی دهد. این interface شامل این پین‌ها می‌شود:

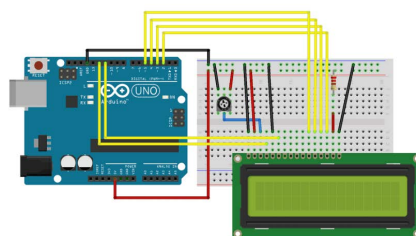
- پین RS ( register select ) : این پین کنترل می‌کند که در کدام خانه‌ی حافظه‌ی LCD می‌نویسیم. می‌توانیم data register را انتخاب کنیم که تعیین کند چه چیزی روی صفحه‌ی LCD می‌رود ، یا اینکه instruction register را انتخاب کنیم که تعیین کند چه دستوری اجرا شود.
- پین R/W ( Read/Write ) : این پین تعیین می‌کند که در مود نوشتن هستیم یا خواندن
- پین Enable
- ۸ پین دیتا ( D<sub>7</sub> - D<sub>0</sub> ) : استیت این پین‌ها (بالا یا پایین بودن ولتاژ) تعیین می‌کند که چه چیزی خوانده یا نوشته می‌شود.

اتصال این پین‌ها در کد تعیین شده است که به آن می‌پردازیم.

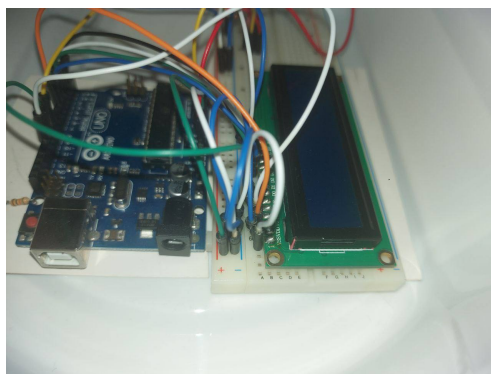
ابتدا LCD را روی برد برد قرار دادیم و سپس اتصالات آن را برد arduino برقرار کردیم. اتصالات آن به این شکل است:

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- LCD R/W pin to GND
- LCD VSS pin to GND
- LCD VCC pin to 5V
- LCD K to GND
- LCD A to VCC
- LCD V0 to pin 6 arduino (PWM)

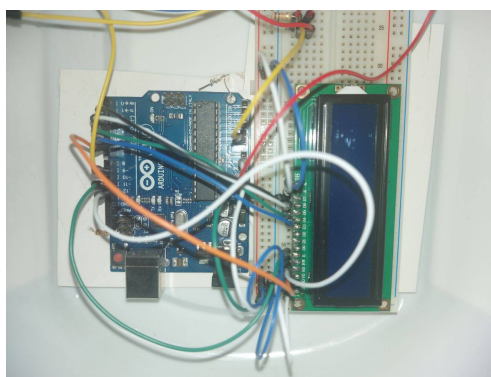
## ۱-۲-۲ تصاویر مربوط به اتصالات LCD و مدار کلی



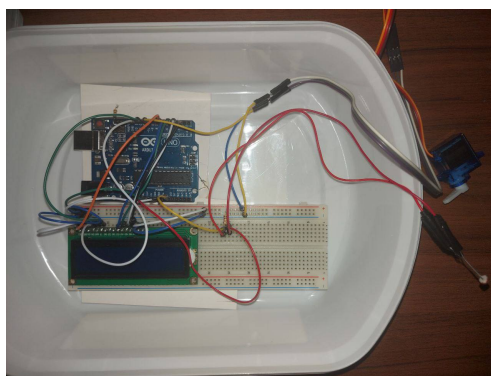
شکل ۳: شماتیک کلی مدار LCD



شکل ۴: اتصالات مربوط به VCC و GND برای LCD و بردبورد



شکل ۵: مدار کامل LCD



شکل ۶: مدار کلی

### ۳ شرح کد

#### ۱-۳ متغیرها و کتابخانه‌ها

در ابتدا کتابخانه‌های مربوط به LCD و موتور Servo را include می‌کنیم. در ادامه متغیرهایی برای زاویه موتور servo تعریف می‌کنیم. `unpressAngle` مربوط به زاویه در زمانی است که کلید

space فشرده نشده و pressAngle مربوط به زمانی است که کلید space فشرده شده است. متغیر timer زمانی را تعیین می‌کند که LCD برای تشخیص نورهای سفید و سیاه جهت کالیبره کردن نیاز دارد. در ادامه یک object از LCD ساخته‌ایم و پین‌های مربوط به آن را مشخص کرده ایم. در کامنت روبروی خط کد مربوط به ساخت LCD می‌توانید نحوه تعیین کردن پین‌ها را ببینید. متغیر myservo یک شیء از جنس Servo است که برای موتور servo استفاده می‌شود. همچنین مقادیر مربوط به نور سفید و سیاه در متغیرهای whiteInt و blackInt ذخیره شده و میانگین آن‌ها در threshold قرار می‌گیرد. به طوری که اگر مقدار نور تشخیص داده شده توسط LDR از threshold کمتر باشد، به عنوان نور سیاه و در غیر این صورت به عنوان نور سفید تشخیص داده می‌شود.

---

```
#include <LiquidCrystal.h>

#include <Servo.h>

#define unpressAngle 90

#define pressAngle 82

#define timer 9

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Creates an LCD object. Parameters:
    (rs, enable, d4, d5, d6, d7)

Servo myservo; // create servo object to control a servo

bool trig=true;

int whiteInt;

int blackInt;

int threshold;
```

---

## ۲-۳ توابع مربوط به مقادیر نورهای سفید و سیاه

دو تابع setWhite و setBlack به منظور تشخیص نورهای سفید و سیاه و کالیبره کردن خودکار استفاده شده‌اند. برای شرح نحوه کارکرد توابع تنها تابع setWhite را توضیح می‌دهیم چرا که کارکرد هر دو مشابه است. در تابع setWhite ابتدا محل کرسر در LCD ست شده و با کمک تابع print عبارت مشخص شده روی LCD قرار می‌گیرد. سپس در حلقه‌ی for ، LCD برای ۱۰ ثانیه صبر کرده و شمارش معکوس می‌کند. پس از صفر شدن تایمر روی آن با تابع analogRead مقدار whiteInt خوانده شده و روی LCD با عبارتی مناسب چاپ می‌شود. در ادامه با تابع delay اندکی



صبر کرده و با تابع clear متن LCD پاک می‌شود و کرسر روی محل اولیه خود قرار می‌گیرد. به این صورت مقدار متغیر whiteInt تعیین می‌شود. نحوه‌ی کارکرد تابع setBlack نیز به همین شکل بوده که مقدار متغیر blackInt را تعیین خواهد کرد.

---

```
void setWhite() {  
    lcd.setCursor(0, 0);  
    lcd.print("set white in:");  
    delay(1000);  
    lcd.setCursor(0, 1);  
    for (int i = timer; i >= 0; i--) {  
        lcd.print(i);  
        delay(1000);  
        lcd.setCursor(0, 1);  
    }  
    lcd.clear();  
    whiteInt = analogRead(A0);  
    lcd.setCursor(0, 0);  
    lcd.print("white number is:");  
    lcd.setCursor(0, 1);  
    lcd.print(String(whiteInt));  
    delay(3000);  
    lcd.clear();  
    lcd.setCursor(0, 0);  
}
```

---

```
void setBlack() {  
    lcd.setCursor(0, 0);  
    lcd.print("set black in:");  
    delay(1000);  
    lcd.setCursor(0, 1);  
    for (int i = timer; i >= 0; i--) {  
        lcd.print(i);
```

```

    delay(1000);
    lcd.setCursor(0, 1);
}
lcd.clear();
blackInt = analogRead(A0);
lcd.setCursor(0, 0);
lcd.print("black number is:");
lcd.setCursor(0, 1);
lcd.print(String(blackInt));
delay(3000);
lcd.clear();
lcd.setCursor(0, 0);
}

```

---

### ۳-۳ تابع setup

این تابع در ابتدای اجرای برنامه فقط یک بار اجرا شده و پس از آن تابع loop به صورت مداوم انجام می‌شود. در این تابع ابتدا پین myservo را به پین ۹ متصل می‌کنیم. در ادامه با تابع analogWrite با ورودی ۶ و ۱۲۰ نور LCD را تنظیم می‌کنیم. لازم به ذکر است که پین ۶ مربوط به LCD است. در ادامه تابع lcd.begin به نوعی LCD را initialize کرده و ابعاد آن را مشخص می‌کند. سپس توابع setBlack و setWhite فراخوانی می‌شوند تا با استفاده از سنسور LDR بتوانیم نور سفید و سیاه را کالیبره کنیم. سپس threshold مورد نیاز برای تشخیص نور سفید و سیاه را از میانگین متغیرهای whiteInt و blackInt بدست می‌آوریم. در نهایت نیز LCD با نوشتن کلمه‌ی play دستور شروع بازی را می‌دهد.

---

```

void setup() {
    Serial.begin(9600);
    myservo.attach(9); // attaches the servo on pin 9 to the servo object
    myservo.write(unpressAngle);
    analogWrite(6, 120); // Generate PWM signal at pin D11, value of 100 (out of
                          255)
}

```

```

lcd.begin(16, 2); // Initializes the interface to the LCD screen, and
                  specifies the dimensions (width and height) of the display }

setWhite();

setBlack();

threshold = blackInt + whiteInt;

threshold /= 2;

lcd.print("play");
}

```

---

### ۴-۳ تابع loop ، حلقه‌ی اصلی

همانطور که گفته شد، این تابع پس از اجرا شدن تابع setup به طور مداوم اجرا می‌شود. در هر بار اجرای آن ابتدا با تابع myservo.write وضعیت موتور servo را در زاویه ۹۰ درجه که برابر فشار ندادن کلید space است قرار می‌دهیم. سپس با خواندن سیگنالی که توسط سنسور LDR ارسال می‌شود ( تکه کد analogRead ) مقدار نور مشاهده شده را تشخیص می‌دهیم و سپس اگر از threshold کمتر باشد به عنوان دیدن نور سیاه بوده و زاویه‌ی موتور servo روی ۸۲ درجه که مربوط به فشردن کلید space است قرار می‌گیرد. در نتیجه دایناسور پرش می‌کند. در ضمن در هر بار پرش دایناسور با استفاده از تابع delay به مدت ۵۰۰ms صبر می‌کنیم.

```

void loop() {

myservo.write(unpressAngle); // unpress the button

int value = analogRead(A0);

Serial.print(String(value) + "\n");

if (value < threshold)

{

myservo.write(pressAngle); // press the button

delay(500); // waits 500ms for the servo to reach the position

}

}

```

---

## ۴ نتیجه گیری

همانطور که مشاهده کردیم می‌توانیم برد arduino را طوری program کنیم که با استفاده از قطعات الکتریکی و الکترونیکی پردازش‌های انجام دهد و آن‌ها را فعال یا غیر فعال کند و حتی به کاربر نتیجه‌ی این پردازش را نشان دهد. هرچند که این برد از قدرت پردازشی کم و حافظه‌ی کمی برخوردار است ولی می‌تواند برای استفاده‌های کوچک حتی به عنوان یک کامپیوتر خانگی نیز استفاده شود.

## آ منابع و مراجع

- [arduino website for lcd](#)
- [youtube](#)

## ب مطالب تکمیلی

برای دیدن فیلم [روی لینک](#) کلیک کنید.