

به نام خدا

ماشین پین بال کنترل شده با آردوینو

نگین السادات حسینی

—

اصول رباتیکز

—

استاد: دکتر محمد زارع

بهمن 1402

## چکیده

در این پروژه، یک ماشین پین بال کنترل شده با آردوینو طراحی و ساخته شده است. این ماشین از یک کابینت چوبی، سطح بازی تزئین شده، و مجموعه‌ای از قطعات استاندارد پین بال مانند فلیپرها، پاپ بامپرها، و سوئیچ‌ها تشکیل شده است. کنترل بازی توسط آردوینو Mega 2560 انجام می‌شود که وظایف مختلفی از جمله تشخیص فعال شدن سوئیچ‌ها، روشن و خاموش کردن چراغ‌ها، نگهداری امتیازات و نمایش آن‌ها روی یک نمایشگر LCD را بر عهده دارد. فرآیند ساخت شامل نجاری دقیق، سیم‌کشی الکترونیکی، و برنامه‌نویسی آردوینو بود که در مجموع حدود شش ماه به طول انجامید. این پروژه به عنوان یک نمونه آموزشی، منابع و دستورالعمل‌های لازم برای علاقه‌مندان به ساخت ماشین پین بال را فراهم می‌کند و به آن‌ها کمک می‌کند تا با موفقیت ماشین خود را بسازند.





## فرایند

---

### مقدمه

این پروژه پین بال، یک بازی محبوب و سرگرم کننده است که در سال های اخیر با استفاده از تکنولوژی های جدید، به یک فعالیت جذاب تبدیل شده است. در این پروژه، ما قصد داریم یک ماشین پین بال کنترل شده با اردوینو را معرفی کنیم. این پروژه، ترکیبی از مهندسی مکانیک، الکترونیک و برنامه نویسی است که به شما اجازه می دهد تا یک بازی پین بال کاملاً سفارشی و کنترل شده را ایجاد کنید. اردوینو، یک برد میکروکنترلر است که برای ساخت پروژه های الکترونیکی استفاده می شود. این برد قابلیت های فراوانی دارد و می تواند با سایر قطعات الکترونیکی ارتباط برقرار کند. در این پروژه، ما از اردوینو برای کنترل عملکرد ماشین پین بال استفاده می کنیم. به یاد داشته باشید که این پروژه، نیاز به برخی مهارت های برنامه نویسی و الکترونیک دارد. اما نگران نباشید، ما در هر مرحله، راهنمایی های مفصلی را برای شما ارائه خواهیم داد. پس بیایید شروع کنیم!

---

## 1. مواد و قطعات مورد نیاز:

- آردوینو Mega 2560
- نمایشگر LCD دو پین Adafruit
- منبع تغذیه 24 ولتی
- حسگر نیرو Adafruit
- مجموعه پاپ بامپر
- سوئیچ پاپ بامپر
- سوئیچ هدف
- سوئیچ رول اور
- مجموعه فلیپر
- لامپ‌های LED

## 2. مراحل اجرا:

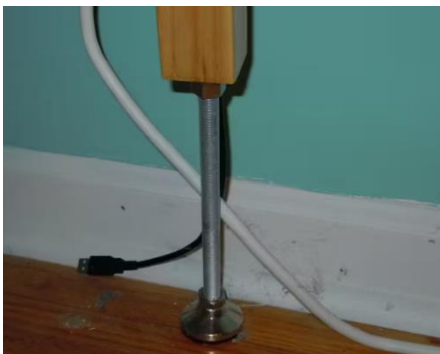
همانطور که در تصویر می‌بینید، ماشین دارای کابینت چوبی است و از قطعات جایگزین ماشین پین‌بال تجاری استفاده می‌کند. ماشین توسط یک برد آردوینو Mega 2560 کنترل می‌شود



### ○ کیس

کیس از تخته چندلای با روکش بلوط که از فروشگاه خانگی خریداری شده، ساخته شده است. تمام ابعاد در این سند به اینچ هستند. پانل‌های جلو و عقب ۲۰ ارتفاع در ۲۳ عرض هستند. پانل‌های جانبی ۲۰ ارتفاع در ۴۷ طول دارند. قطعات با لبه‌های مورب بریده شده و با استفاده از بیسکویت‌زن و چسب چوب به هم متصل شده‌اند. پیچ‌های جیب‌دار و چسب نیز کار می‌کردند. ماشین قرار است تحت فشار زیادی قرار گیرد بنابراین باید محکم باشد. قبل از اتصال قطعات به هم، همچنین یک شیار  $\frac{3}{4}$  اینچی ۱ اینچ بالا از کف بریده‌ام. قفسه نباید چسبانده شود. نیاز دارد تا برای انبساط و انقباض شناور باشد. قفسه برای نگه داشتن منبع تغذیه اصلی و منبع تغذیه برای چراغ LED تزئینی که زیر درب نصب شده، استفاده شد. سطح بازی ۲۲ عرض در ۴۲ طول است. توجه داشته باشید که این سطح به طور قابل توجهی

کوتاه‌تر از کابینت است. این به من حدود ۳ اینچ در جلو می‌دهد تا سیم‌ها را به دکمه‌های فلیپر و به چراغ‌های LED عبور دهم. همچنین فضایی برای مکانیزم پرتاب توپ فراهم می‌کند. سطح بازی به سادگی روی چند بلوک چوبی کوچک که حدود ۴ اینچ زیر بالای کیس به طرفین پیچ شده‌اند، قرار می‌گیرد. برای دسترسی به اجزای داخلی ماشین، به سادگی درب را باز می‌کنم، سطح بازی را برداشته و آن را برمی‌گردانم.



کیس و سطح بازی همگی در زوایای راست ساخته شده‌اند. شیب بازی توسط پایه‌ها ایجاد می‌شود، همانطور که در نمای جانبی نشان داده شده است. ماشین‌های پین‌بال می‌توانند زاویه‌ای از ۱ تا ۷ درجه داشته باشند. هر چه زاویه بیشتر باشد، بازی سریع‌تر خواهد بود. چون مطمئن نبودم به چه زاویه‌ای نیاز خواهم داشت، پایه‌ها را به صورت قابل تنظیم طراحی کردم. پایه‌ها خودشان چوبی هستند و در اکثر فروشگاه‌های خانگی قابل تهیه‌اند. سپس با استفاده از یک مته‌ی مارپیچ، یک سوراخ ۱۲ اینچی در انتهای هر پایه حفاری کردم. در انتهای پایین، از یک اپوکسی دو جزئی برای قرار دادن یک مهره T ۸/۳ اینچی استفاده کردم. مراقب باشید که هیچ چسبی به رزوه‌ها نرسد یا جایی که سوراخی که حفاری کرده‌اید را مسدود کند. سپس یک میله‌ی رزوه‌ای ۱۲ اینچی ۸/۳ اینچی را داخل پایه قرار دادم و یک مهره را به میله رزوه کردم. سپس یک پایه تراز را به انتهای میله وصل کردم.

#### ○ نصب پایه‌های میز

صفحات نصب پایه‌های میز به سادگی به قفسه‌ی پایین پیچ شده و پایه‌ها به آن‌ها پیچ شده‌اند. برای تنظیم ارتفاع ماشین، به سادگی مهره را شل کرده و سپس میله‌ی رزوه‌ای را به داخل یا خارج پیچ کنید. سپس مهره را دوباره سفت کنید تا مطمئن شوید که میز محکم باقی می‌ماند.

در تصویر، یک کابل برق و یک کابل USB مشاهده می‌کنید. کابل برق از یک نوار برق می‌آید که به پشت ماشین متصل شده است. همه چیز به آنجا وصل می‌شود، بنابراین

یک کلید همه چیز را روشن یا خاموش می‌کند. کابل USB نیز برای این است که بتوانم برد آردوینو را بدون نیاز به برداشتن سطح بازی دوباره برنامه‌ریزی کنم.

#### ○ درب

قاب درب از چوب بلوط ساخته شده و ابعاد بیرونی آن همانند کابینت است. قسمت‌های بالا و کناری ۱.۵ اینچ عرض دارند، در حالی که قسمت پایین ۵ اینچ عرض دارد. عرض اضافی در پایین برای پنهان کردن فاصله بین سطح بازی و کابینت استفاده می‌شود. یک شیار در لبه داخلی قطعات قاب برای نگه داشتن پوشش پلکسی‌گلاس ایجاد شده است. من به جای شیشه از پلکسی‌گلاس استفاده کردم زیرا سبک‌تر، ایمن‌تر و برش آن آسان‌تر است. درب با استفاده از پیچ‌های جیبی موتتاژ شد. این کار را کردم تا بتوانم در صورت نیاز به تعویض پلکسی‌گلاس، آن را جدا کنم. همچنین یک نوار از چراغ‌های LED رنگارنگ زیر درب برای اهداف تزئینی نصب کردم. در نهایت، درب را با استفاده از یک لولا پیانویی که به هر دو درب و کیس تعبیه شده بود، به کیس متصل کردم.



#### ○ آردوینو

برد آردوینو Mega 2560 اساساً چهار کار انجام می‌دهد: تعیین زمان فعال شدن سوئیچ‌ها، روشن یا خاموش کردن چراغ‌ها به صورت متناسب، پیگیری امتیازها، و نمایش امتیاز و شماره توپ روی یک LCD کوچک. بیشتر سیم‌کشی‌ها نسبتاً ساده هستند. برای مثال، اهداف به یکی از ورودی‌های دیجیتال آردوینو و به زمین متصل می‌شوند. ورودی با استفاده از دستوری مشابه زیر تنظیم می‌شود

```
pinMode(Pin_Number, INPUT_PULLUP);
```

زمانی که کد تشخیص می‌دهد که ورودی از حالت بالا به پایین تغییر کرده است، می‌داند که هدف مورد اصابت قرار گرفته و سپس چراغ پین‌بال مناسب را روشن می‌کند. تنها پیچیدگی در این میان از سمت پاپ بامپرها است. سوئیچ‌های رول اور و اهداف استفاده شده در این بازی همه از آردوینو تغذیه می‌شوند. پاپ بامپرهای تغذیه خود را از



یک منبع تغذیه جداگانه ۲۵ ولتی دریافت می‌کنند زیرا به توان بیشتری نسبت به آنچه آردوینو می‌تواند تأمین کند، نیاز دارند. پاپ بامپرهای دارای یک سوئیچ هستند که آن‌ها را فعال می‌کند، اما نمی‌توانم آن را مستقیماً به دلیل ولتاژ بالا بخوانم. بنابراین، ابتدا خروجی سوئیچ پاپ بامپر را به یک تقسیم‌کننده ولتاژ می‌فرستم.

#### ○ تقسیم‌کننده ولتاژ

پس از عبور از تقسیم‌کننده ولتاژ، ولتاژ به 5 ولت کاهش می‌یابد و آردوینو می‌تواند آن را مدیریت کند. متوجه شدم که با استفاده از ورودی‌های آنالوگ، خوانش‌های نادرست کمتری دریافت می‌کنم، زیرا تقسیم‌کننده‌های ولتاژ به نظر می‌رسد همیشه مقداری جریان نشت می‌دهند. با ورودی‌های آنالوگ، می‌توانستم بررسی کنم که آیا یک آستانه فراتر رفته است، به طوری که می‌توانستم به راحتی یک سیگنال را از نویز جدا کنم. در حین بازی، اگر همه دستگاه‌های نوع خاصی (هدف‌ها، پاپ بامپرهای رول اورها) فعال شده بودند، آردوینو را برنامه‌ریزی کردم تا تمام چراغ‌های آن دستگاه‌ها را فلش کند و سپس امتیاز آن‌ها را افزایش دهد.

همچنین از یک حسگر نیرو کوچک برای تشخیص زمان از دست رفتن توپ استفاده کردم. این به من امکان می‌دهد تا تعداد توپ‌های بازی شده را پیگیری کرده و زمان پایان بازی را تعیین کنم. زمانی که بازی به پایان می‌رسد، تمام چراغ‌ها به ترتیب فلش می‌زنند. در نهایت، از آردوینو برای نمایش امتیاز و شماره توپ استفاده کردم.

#### ○ دیاگرام سیم‌کشی

دیاگرام سیم‌کشی پیوست نشان می‌دهد که چگونه یکی از هر یک از اجزای اصلی سیم‌کشی شده است. اگر همه آن‌ها نمایش داده می‌شدند، دیاگرام خیلی پیچیده می‌شد.

#### ○ منبع تغذیه

برای تأمین برق اجزای ماشین پین‌بال، از یک منبع تغذیه 24 ولتی استفاده کردم. این واحد توان کافی برای فلیپرها و سایدکیک‌ها را داشت. کمی ضعیف برای پاپ بامپرهای است. ممکن است بخواهید برای این قطعات از منبع تغذیه قوی‌تری استفاده کنید. اگر چنین کنید، باید تقسیم‌کننده‌های ولتاژ خود را بسازید، زیرا فقط توانستم نمونه‌هایی با حداکثر ولتاژ 25 ولت پیدا کنم.

به جای اتصال مستقیم همه چیز به منبع تغذیه، سیم‌هایی از آن به یک جفت شینه (bus bars) که زیر سطح بازی نصب کرده بودم، کشیدم. این کار اتصال اجزای مختلف را بسیار آسان‌تر کرد. می‌توانید یکی از شینه‌ها را در تصویر دوم بخش Slingshot ببینید.

## ○ تیرانداز توپ یا پلاژر

برای اجازه گذر تیرانداز توپ یا پلاژر از جلوی قاب، باید یک سوراخ در جلوی قاب برای گذر تیرانداز توپ یا پلاژر ایجاد شود. سپس آن را در داخل قاب می‌پیچانند. باید با دقت بسیار از اطمینان حاصل شود که تیرانداز توپ یا پلاژر به درستی با زمینه بازی هم‌تراز است. یک لبه مستقیم روی زمینه بازی قرار داده و نقطه‌ای را که با داخل قاب برخورد می‌کند علامت‌گذاری کنید. سپس به اندازه کافی بالا بروید تا وسط تیرانداز توپ یا پلاژر به ارتفاع نصف ارتفاع توپ‌پینبال برسد توپ‌های پینبال اندازه 1 و 16/1 اینچ دارند و در Pinball قابل دسترس هستند .

## ○ زمینه بازی

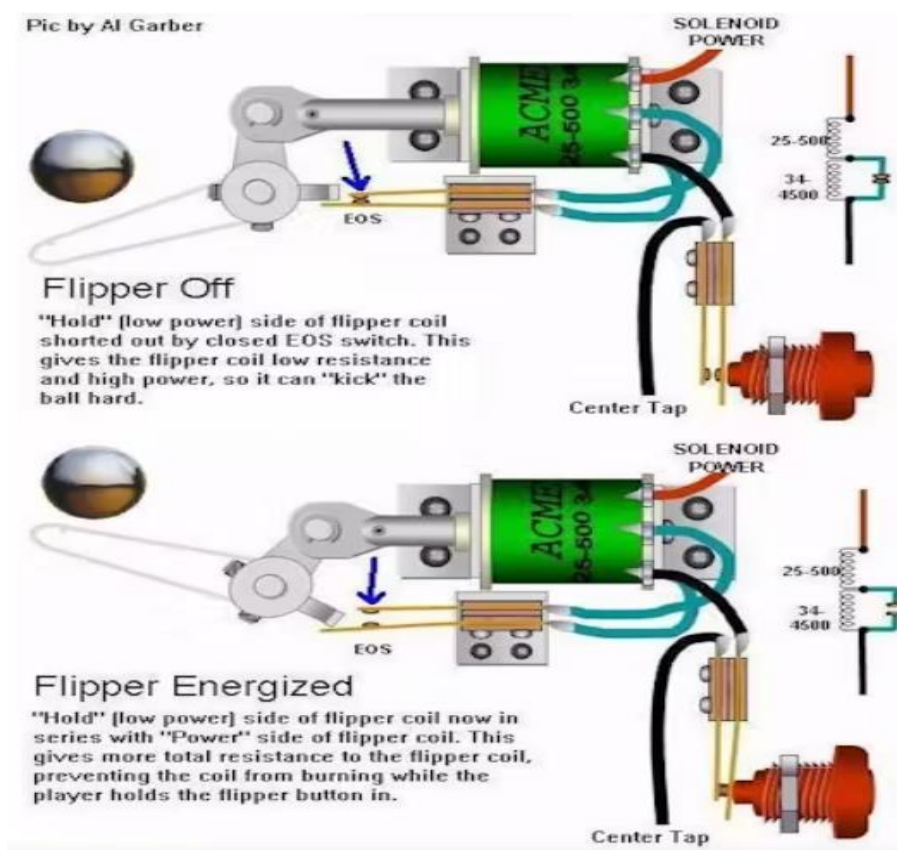
زمینه بازی از پوسته خشک چوب درخت بیرچ ضخیم 1/4 اینچ ساخته شده است.

## ○ لمینت کردن تصویر بر روی زمینه

ابتدا پوسته چوب را با چسب تماسی پاشیدم و سپس تصویر را بر روی آن لمینت کردم. سپس آن را برگرداندم و به اندازه مورد نیاز برش دادم.

## ○ فلیپرها

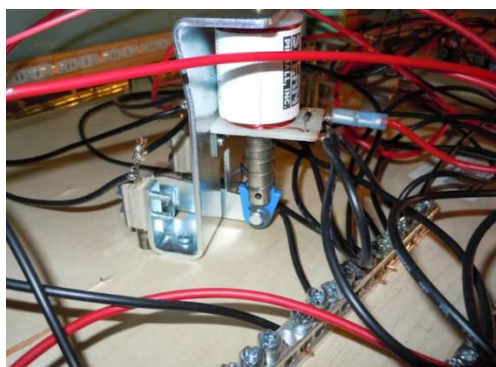
برای نصب مکانیزم فلیپر، شما نیاز به یک تیرانداز توپ، سوئیچ فلیپر و یک دکمه دارید.





### ○ اسلینگ‌شات‌ها

همچنین به پین‌ها و بندهای لاستیکی نیاز خواهید داشت که می‌توانید آن‌ها را در همان مکان پیدا کنید. همچنین تصویر اسلینگ‌شات نصب شده زیرین نیز نشان داده شده است. باید اشاره کنم که برای همه اجزای متصل به منبع تغذیه 24 ولت از سیم گیج 16 استفاده شده است و برای اتصالات به برد آردوینو از سیم گیج 22 استفاده شده است. در تصویر اول همچنین چندین ریل فلزی در کنار تیراندازهای توپ نیز نمایش داده شده است. من نتوانستم مکانی برای خرید ریل‌ها پیدا کنم، بنابراین آن‌ها را خود ساختم. از یک سیم فولاد کروم استفاده کردم که در فروشگاه خانگی پیدا کردم. این سیم برای نشان دادن مرزها زمانی استفاده می‌شود که با افتراق حشرات پاشیده شود. سیم را برش دادم و خم کردم. سپس در زمینه بازی چندین سوراخ حفر کردم و ریل‌ها را در محل مورد نظر چسباندم.

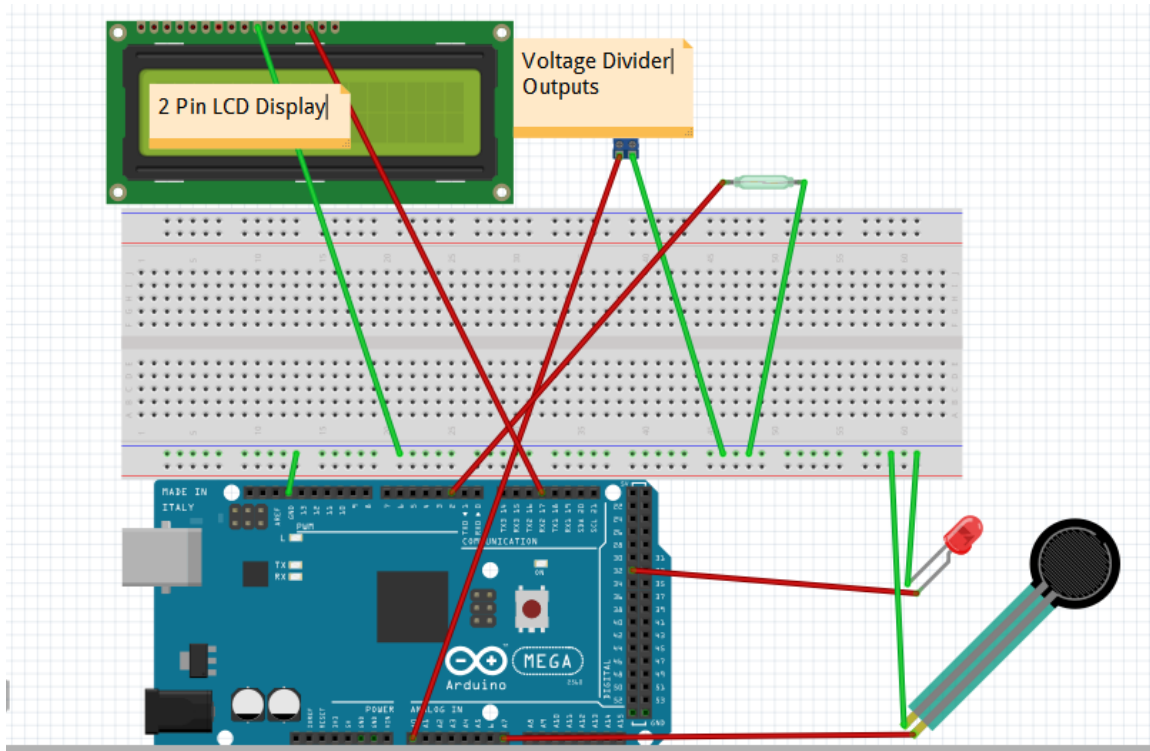


### ○ پاپ بامپر‌ها

همچنین به سوئیچ‌ها و کاپ‌ها نیاز خواهید داشت. پاپ بامپر‌ها با حفر سه سوراخ نصب می‌شوند، دو سوراخ برای تیغه‌های فلزی حمایتی و یک سوراخ برای شافتی که سوئیچ را فعال می‌کند. وقتی یک توپ وارد دیسک سفید در دید بالا می‌شود، شافت را به پایین فشار می‌دهد. این عمل باعث فعال شدن سوئیچ برگ می‌شود که سپس سولنوئید را فعال می‌کند. سپس سولنوئید حلقه فلزی را به پایین می‌کشد که توپ را به جهت مخالف ارسال می‌کند.



### 3. نمای کلی اولیه از قطعات اصلی ماشین پین بال با برد آردوینو



### 4. کد آردوینو:

```
const int TxPin = 17;
long Score = 0;
long OldScore = 0;
long Target = 1;
long Pop = 1;
long Roll = 10;
int Targets[8];
int Rolls[3];
int Pops[4];
int Milli = 10;
int Sum = 0;
int Flash = 100;
int Ball = 0;
int i=0;
int Shot = 0;
int Lost = 0;
```

```

int Pressure = 1024;

#include <SoftwareSerial.h>;
SoftwareSerial mySerial = SoftwareSerial(255, TxPin);

void setup() {
  /* Words without an s are the value achieved by interacting with a device.
  * Works with an s keep track of which individual ones were interacted with.
  * The latter is needed to determine when all have been hit and the value needs
  upgrading
  * and the lights need turning off.
  */
  pinMode(TxPin, OUTPUT);
  digitalWrite(TxPin, HIGH);
  mySerial.begin(9600);
  mySerial.write(12);      // Clear
  mySerial.write(17);      // Turn backlight on

  //target inputs
  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
  pinMode(4,INPUT_PULLUP);
  pinMode(5,INPUT_PULLUP);
  pinMode(6,INPUT_PULLUP);
  pinMode(7,INPUT_PULLUP);
  pinMode(8,INPUT_PULLUP);
  pinMode(9,INPUT_PULLUP);
  //rollover inputs
  pinMode(10,INPUT_PULLUP);
  pinMode(11,INPUT_PULLUP);
  pinMode(12,INPUT_PULLUP);
  //lower ball shot switch
  pinMode(15,INPUT_PULLUP);
  //upper ball shot switch
  pinMode(16,INPUT_PULLUP);
  //lcd output

```

```

pinMode(17,OUTPUT);
//target lights, respective
pinMode(32,OUTPUT);
pinMode(33,OUTPUT);
pinMode(34,OUTPUT);
pinMode(35,OUTPUT);
pinMode(36,OUTPUT);
pinMode(37,OUTPUT);
pinMode(38,OUTPUT);
pinMode(39,OUTPUT);
//rollover lights, respective
pinMode(40,OUTPUT);
pinMode(41,OUTPUT);
pinMode(42,OUTPUT);
//pop bumper lights
pinMode(50,OUTPUT);
pinMode(51,OUTPUT);
pinMode(52,OUTPUT);
pinMode(53,OUTPUT);
}

void loop() {
// put your main code here, to run repeatedly:
//If a pull-down resistor is used, the input pin will be LOW when the switch is open and
HIGH when the switch is closed.
//check if a target was hit

//***** Targets *****

for (int i=0; i<8; i++){
if (digitalRead(i+2) == LOW){
//Target activated
Targets[i]=1;
Score = Score + Target;
//turn on Target light
digitalWrite(i+32,HIGH);
//delay so as not get multiple points for one hit

```

```

delay(Milli);
break;
}
}
Sum = 0;
for (int i=0; i<8; i++){
Sum = Sum + Targets[i];
}
if (Sum == 8){
//all Targets lit, so flash and then turn off.
for (int j=0; j<3; j++){
for (int i=0; i<8; i++){
digitalWrite(i+32, LOW);
}
delay(Flash);
for (int i=0; i<8; i++){
digitalWrite(i+32, HIGH);
}
delay(Flash);
}
for (int i=0; i<8; i++){
digitalWrite(i+32, LOW);
Targets[i]=0;
}
delay(Flash);
//Multiply target value by 10
Target = Target * 5;
//goto Skip;
}

// ***** Rollovers *****

for (int i=0; i<3; i++){
if (digitalRead(i+10) == LOW){
//rollover activated

```

```

Rolls[i]=1;
Score = Score + Roll;
//turn on rollover light
digitalWrite(i+40,HIGH);
//delay so as not get multiple points for one hit
delay(Milli);
break;
}
}
Sum = 0;
for (int i=0; i<3; i++){
Sum = Sum + Rolls[i];
}
if (Sum == 3){
//all rollovers lit, so flash and then turn off.
for (int j=0; j<3; j++){
for (int i=0; i<3; i++){
digitalWrite(i+40, LOW);
}
delay(Flash);
for (int i=0; i<3; i++){
digitalWrite(i+40, HIGH);
}
delay(Flash);
}
for (int i=0; i<3; i++){
digitalWrite(i+40, LOW);
Rolls[i]=0;
}
delay(Flash);
//Multiply score by 2
Score = Score * 2;
Roll = Roll * 10;
//goto Skip;
}

//***** Pop Bumpers *****

```



```
for (int i=0; i<4; i++){
if (analogRead(i) > 500){
//pop activated
Pops[i]=1;
Score = Score + Pop;
//turn on pop bumper light
digitalWrite(i+50,HIGH);
//delay so as not get multiple points for one hit
//mySerial.print(analogRead(i));
//mySerial.print(" ");
delay(Milli);
break;
}
}

Sum = 0;
for (int i=0; i<4; i++){
Sum = Sum + Pops[i];
}

if (Sum == 4){
//all pop bumpers lit, so flash and then turn off.
for (int j=0; j<3; j++){
for (int i=0; i<4; i++){
digitalWrite(i+50, LOW);
}

delay(Flash);
for (int i=0; i<4; i++){
digitalWrite(i+50, HIGH);
}

delay(Flash);
}

for (int i=0; i<4; i++){
digitalWrite(i+50, LOW);
Pops[i]=0;
}

delay(Flash);
//Multiply target value by 10
```

```
Pop = Pop * 2;
//goto Skip;
}
Skip:

//Determine ball number
if (digitalRead(15) == LOW){
//ball hit lower alley switch
//if not already done so, increase Ball
if (Shot == 0){
//Set Lost = 0 since not on pressure pad
Lost = 0;
Pressure = analogRead(7) + 20;
//set OldScore so as to reprint ball value on LCD
OldScore = -1;
Ball = Ball + 1;
if (Ball == 6){
Ball = 1;
Score = 0;
Target = 1;
Roll = 1;
Pop = 1;
}
Shot = 1;
}
}
if (digitalRead(16) == LOW){
//ball hit lower alley switch
//if not already done so, increase Ball
if (Shot == 0){
//Set Lost = 0 since not on pressure pad
Lost = 0;
Pressure = analogRead(7) + 15;
//set OldScore so as to reprint ball value on LCD
OldScore = -1;
Ball = Ball + 1;
if (Ball == 6){
```

```

Ball = 1;
Score = 0;
Target = 1;
Roll = 1;
Pop = 1;
}
Shot = 1;
}
}

if (analogRead(7) > Pressure){
//ball on pressure pad
Shot = 0;
if (Lost == 0){
//mySerial.print(analogRead(7));
//Score = Score + 100;
Lost = 1;
if (Ball == 5){
//Game Over
//flash rollovers and then turn off.
for (int j=0; j<3; j++){
for (int i=0; i<3; i++){
digitalWrite(i+40, LOW);
}
delay(Flash);
for (int i=0; i<3; i++){
digitalWrite(i+40, HIGH);
}
delay(Flash);
}
for (int i=0; i<3; i++){
digitalWrite(i+40, LOW);
Rolls[i]=0;
}
// flash pop bumpers and then turn off
for (int j=0; j<3; j++){
for (int i=0; i<4; i++){

```

```
digitalWrite(i+50, LOW);
}
delay(Flash);
for (int i=0; i<4; i++){
digitalWrite(i+50, HIGH);
}
delay(Flash);
}
for (int i=0; i<4; i++){
digitalWrite(i+50, LOW);
Pops[i]=0;
}
//Flash Targets and then turn off.
for (int j=0; j<3; j++){
for (int i=0; i<8; i++){
digitalWrite(i+32, LOW);
}
delay(Flash);
for (int i=0; i<8; i++){
digitalWrite(i+32, HIGH);
}
delay(Flash);
}
for (int i=0; i<8; i++){
digitalWrite(i+32, LOW);
Targets[i]=0;
}
mySerial.write(12);          // Clear
delay(5);
// Required delay
mySerial.print(Score); // First line
mySerial.write(13);          // Form feed
mySerial.print("Game Over!!!"); // Second line
}
}
}
//print to LCD
```

```

if (Score != OldScore){
mySerial.write(12);          // Clear
delay(5);                    // Required delay
//mySerial.print(analogRead(7));
mySerial.print(Score); // First line
mySerial.write(13);          // Form feed
mySerial.print("Ball = "); // Second line
mySerial.print(Ball);
OldScore = Score;
}
}

```

توضیحات توابع:

- **setup ()** : تنظیمات اولیه برای آماده‌سازی محیط اجرا شامل تعریف پایه‌ها، شروع ارتباط با سریال، روشن کردن پس‌زمینه LCD و تعریف وضعیت پایه‌ها و نورهای مختلف.
- **loop ()** : اجرای اصلی برنامه که تکرار شود و وظایف زیر را انجام می‌دهد:
- **Targets ()** : بررسی وضعیت هدف‌ها، افزودن امتیاز به اسکور، روشن کردن نور مربوطه و تنظیم مقدار هدف بر اساس تعداد هدف‌های فعال.
- **Rollovers ()** : بررسی وضعیت رول‌اورها، افزودن امتیاز به اسکور، روشن کردن نور مربوطه و تنظیم مقدار رول بر اساس تعداد رول‌اورهای فعال.
- **Pop Bumpers ()** : بررسی وضعیت پاپ‌بامپرهای، افزودن امتیاز به اسکور، روشن کردن نور مربوطه و تنظیم مقدار پاپ بر اساس تعداد پاپ‌بامپرهای فعال.
- **Determine Ball Number ()** : بررسی وضعیت سوئیچ‌های پایینی و بالایی برای تعیین زمانی که توپ جدیدی وارد بازی می‌شود و افزایش شماره توپ.
- **Game Over ()** : بررسی وضعیت توپ در بالا و اعلام پایان بازی اگر توپ آخرین رول را سفته باشد، همراه با نورپردازی فعالیت‌های مختلف و نمایش اسکور و پیام بازی تمام شده بر روی LCD.
- **Print to LCD ()** : نمایش اسکور جدید و شماره توپ بر روی LCD هر زمان که اسکور تغییر کند.

## نتیجه گیری

در این پروژه، ما به بررسی و ساخت یک ماشین پین بال کنترل شده با استفاده از اردوینو پرداختیم. با توجه به انعطاف پذیری و قابلیت برنامه ریزی اردوینو، توانستیم یک سیستم کنترل دقیق برای مدیریت عملکردهای مختلف ماشین پین بال ایجاد کنیم. از جمله این عملکردها می توان به کنترل پرتاب کننده توپ، فعال سازی بامپرها و مدیریت نمایشگر امتیاز اشاره کرد.

این پروژه نشان داد که چگونه می توان با استفاده از تکنولوژی های موجود و برنامه نویسی خلاقانه، یک بازی کلاسیک را بهبود بخشید و تجربه ای تعاملی و مدرن ارائه داد. همچنین، این پروژه فرصتی برای یادگیری مهارت های مهندسی الکترونیک و برنامه نویسی فراهم کرد و امکان سفارشی سازی بیشتر را برای کاربران نهایی مهیا ساخت.

در نهایت، ماشین پین بال کنترل شده با اردوینو نمونه ای از ترکیب هنر و فناوری است که می تواند به عنوان الهام بخش برای پروژه های آموزشی و سرگرمی در آینده عمل کند. این پروژه تأکید می کند که با اشتیاق و تلاش، می توان ایده های خلاقانه را به واقعیت تبدیل کرد.

## منبع

<https://www.hackster.io/BobB/arduino-controlled-pinball-machine-525863#story>

---