

دستگاه پین بال کنترل شده توسط آردوینو

سجاد گل محمدی

—

مبانی رباتیک

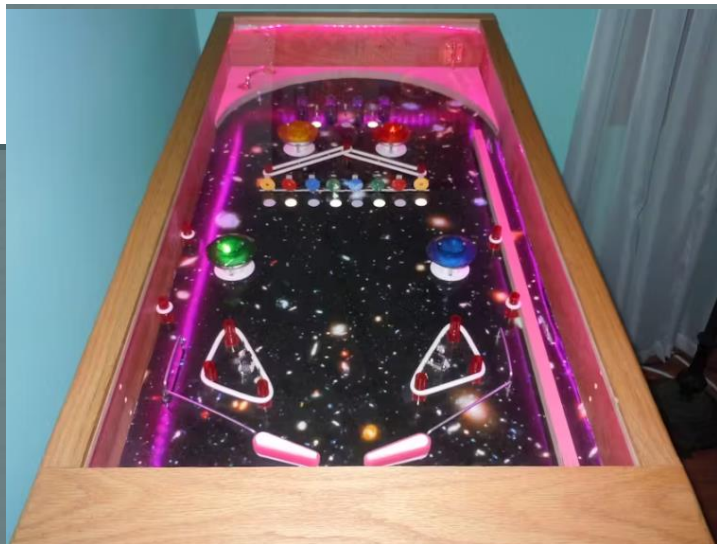
—

استاد: محمد زارع

تیر ۱۴۰۳

چکیده

پروژه “Arduino Controlled Pinball Machine” یک ماشین پین بال استاندارد است که با استفاده از قطعات موجود در فروشگاه‌های خانگی و فروشگاه‌های تامین‌کننده قطعات پین بال ساخته شده است. این ماشین با استفاده از یک برد Arduino Mega 2560 کنترل می‌شود و شامل قطعاتی مانند سنسورهای نیرو، سوئیچ‌های هدف، و لامپ‌های LED است. ساخت این پروژه نیازمند ترکیبی از مهارت‌های نجاری، مکانیکی، الکترونیکی و برنامه‌نویسی بوده و حدود شش ماه زمان برده است. این پروژه نشان‌دهنده توانایی ساخت یک سیستم پیچیده با استفاده از ابزارها و تکنولوژی‌های ساده و در دسترس است.



THE PROCESS

مقدمه

پروژه "ماشین پین‌بال کنترل شده با آردوینو" توسط باب بلومکویست، یک ماشین پین‌بال استاندارد است که با استفاده از قطعات موجود در فروشگاه‌های خانگی و فروشگاه‌های قطعات پین‌بال ساخته شده است. این پروژه ترکیبی از مهارت‌های نجاری، مکانیکی، الکترونیکی و برنامه‌نویسی را می‌طلبد و حدود شش ماه زمان برده است.

این ماشین با استفاده از یک برد آردوینو مگا ۲۵۶۰ کنترل می‌شود و شامل قطعاتی مانند نمایشگر LCD، منبع تغذیه ۲۴ ولت، سنسور نیرو، و مجموعه‌ای از قطعات پین‌بال مانند پاپ بامپر، سوئیچ‌های هدف و رول‌اور، و لامپ‌های LED است. ساخت این پروژه نیاز به آزمایش و پروتوتایپ کردن هر قطعه قبل از نصب در ماشین اصلی دارد تا از عملکرد صحیح آن اطمینان حاصل شود.

بلومکویست در این پروژه از تخته سه‌لایه روکش‌دار بلوط برای ساخت بدنه استفاده کرده است. ابعاد بدنه شامل پنل‌های جلو و عقب ۲۰ اینچ ارتفاع و ۲۳ اینچ عرض، و پنل‌های جانبی ۲۰ اینچ ارتفاع و ۴۷ اینچ طول است. قطعات با استفاده از اتصالات بیسکوییتی و چسب چوب به هم متصل شده‌اند. سطح بازی ۲۲ اینچ عرض و ۴۲ اینچ طول دارد و به راحتی قابل جدا شدن برای دسترسی به قطعات داخلی است.

هدف این پروژه ارائه راهنمایی‌های کافی برای ساخت یک ماشین پین‌بال مشابه است تا دیگران بتوانند با مشکلات کمتری این کار را انجام دهند. بلومکویست تأکید می‌کند که آزمایش و پروتوتایپ کردن هر قطعه قبل از نصب نهایی بسیار مهم است تا از عملکرد صحیح آن اطمینان حاصل شود.

۱. مواد و قطعات مورد نیاز:

- میکروکنترلر **Arduino Mega 2560**
- نمایشگر LCD با ۲ پین **Adafruit**
- **سنسور نیروی Adafruit**
- **موتناژ ضربه گیر پاپ**
- سوئیچ سپر پاپ
- سوئیچ هدف
- **سوئیچ Rollover**
- **موتناژ فلیپر**
- لامپ های ال ای دی

۲. مراحل اجرا:

****۱.** ساخت بدنه **: بدنه از تخته سه لای روکش دار بلوط ساخته شده است. پنل های جلو و عقب ۲۰ اینچ ارتفاع و ۲۳ اینچ عرض، و پنل های جانبی ۲۰ اینچ ارتفاع و ۴۷ اینچ طول دارند. قطعات با استفاده از اتصالات بیسکوییتی و چسب چوب به هم متصل شده اند.

****۲.** ساخت سطح بازی **: سطح بازی ۲۲ اینچ عرض و ۴۲ اینچ طول دارد و به راحتی قابل جدا شدن برای دسترسی به قطعات داخلی است. این سطح روی بلوک های چوبی کوچک نصب شده است.

****۳.** نصب قطعات الکترونیکی **: برد آردوینو مگا ۲۵۶۰ برای کنترل بازی استفاده شده است. قطعات الکترونیکی شامل نمایشگر LCD، منبع تغذیه ۲۴ ولت، سنسور نیرو، و مجموعه ای از قطعات پین بال مانند پاپ بامپر، سوئیچ های هدف و رول اور، و لامپ های LED است.

****۴.** آزمایش و پروتوتایپ کردن **: هر قطعه قبل از نصب در ماشین اصلی آزمایش و پروتوتایپ شده است تا از عملکرد صحیح آن اطمینان حاصل شود.

****۵.** نصب نهایی **: پس از اطمینان از عملکرد صحیح هر قطعه، قطعات در ماشین اصلی نصب شده اند و بازی آماده استفاده است^۱.

۳. 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**

این تابع برای تنظیمات اولیه استفاده می‌شود. در اینجا پین‌های ورودی و خروجی تعریف می‌شوند و ارتباطات سریال آغاز می‌شود.

- **():loop**

این تابع به صورت مداوم اجرا می‌شود و شامل منطق اصلی بازی است. اینجا جایی است که وضعیت سنسورها و سوئیچ‌ها بررسی می‌شود و بر اساس آن‌ها اقدامات لازم انجام می‌شود.

- **():checkSwitches**

این تابع وضعیت سوئیچ‌های مختلف (مانند سوئیچ‌های هدف و سوئیچ‌های رول‌اور) را بررسی می‌کند و در صورت فعال شدن آن‌ها، امتیازدهی و سایر اقدامات لازم را انجام می‌دهد.

- **():updateDisplay**

این تابع برای به‌روزرسانی نمایشگر LCD استفاده می‌شود و اطلاعات مربوط به امتیازات و وضعیت بازی را نمایش می‌دهد.

- **():controlFlippers**

این تابع برای کنترل فلیپرها (بازوهای بازی) استفاده می‌شود و بر اساس ورودی‌های کاربر، فلیپرها را حرکت می‌دهد.

نتیجه گیری

پروژه "Arduino Controlled Pinball Machine"، نمونه‌ای عالی از ترکیب خلاقیت، مهارت‌های فنی و کاربردهای آموزشی در زمینه رباتیک و آردوینو است. استفاده از آردوینو به عنوان کنترلر اصلی، هزینه‌های ساخت را به طور قابل توجهی کاهش می‌دهد. این امر باعث می‌شود که پروژه برای افراد بیشتری قابل دسترس باشد. این پروژه فرصتی برای یادگیری عملی مفاهیم الکترونیک، برنامه‌نویسی و مکانیک فراهم می‌کند. این پروژه امکان طراحی و ساخت یک ماشین پین‌بال با ویژگی‌ها و تنظیمات دلخواه را فراهم می‌کند. این امر به کاربران اجازه می‌دهد تا خلاقیت خود را در طراحی و ساخت به کار گیرند. همچنین استفاده از قطعات بازیافتی و موجود در خانه، نه تنها هزینه‌ها را کاهش می‌دهد بلکه به حفظ محیط زیست نیز کمک می‌کند. برنامه‌نویسی آردوینو برای کنترل اجزای مختلف ماشین پین‌بال، مهارت‌های برنامه‌نویسی را تقویت می‌کند و درک بهتری از منطق برنامه‌نویسی و کنترل دستگاه‌ها به دست می‌دهد. این پروژه شامل استفاده از سنسورها و محرک‌های مختلف است که به کاربران کمک می‌کند تا با نحوه کارکرد و کاربردهای آن‌ها آشنا شوند. یادگیری از این پروژه می‌تواند به عنوان پایه‌ای برای پروژه‌های پیچیده‌تر در زمینه رباتیک و آردوینو مورد استفاده قرار گیرد. در انتها این پروژه نه تنها یک سرگرمی جذاب است، بلکه یک ابزار آموزشی قدرتمند برای یادگیری و توسعه مهارت‌های فنی و خلاقیت است!

منبع

Arduino Controlled Pinball Machine

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