

# بِنَاءِ مَرْكَبَةِ حَسَنَةِ دُولَةِ اِنْدُونِيْسِيَا

ترجمه پروژه درسایت **hackster**

پروژه رباتیکز

دانشگاه اپادانا

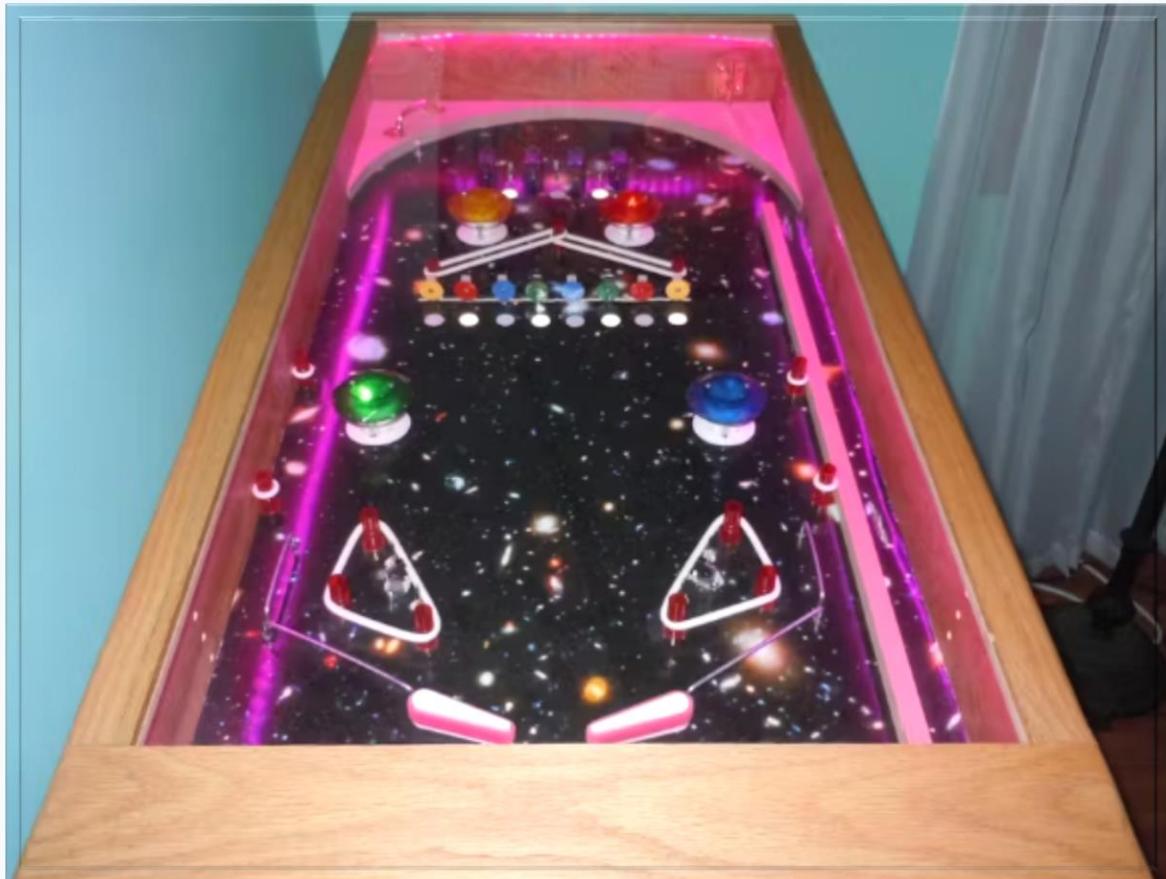
سیده غزل شجاعی

**9922028064**

## • Arduino Controlled Pinball Machine

### • دستگاه پین بال کنترل شده آردوینو

دستگاه پین بال با اندازه استاندارد که با استفاده از قطعات فروشگاه خانگی و خانه های عرضه پین بال ساخته شده است. بازی توسط آردوینو کنترل می شود.



## موارد استفاده شده در این پروژه

قطعات سخت افزاری:

1. آردوینو مگا 2560:

آردوینو:



آردوینو یک پلتفرم نمونه سازی الکترونیکی منبع باز است که بر اساس سخت افزار و نرم افزار انعطاف پذیر و استفاده از آسان است.

آردوینو مگا 2560:

آردوینو مگا طراحی شده برای جاه طلبانه ترین پروژه ها که به پین های اضافی و حافظه اضافی نیاز دارد



**2. نمایشگر LCD 2 پین Adafruit** *Adafruit 2-pin LCD Display*

**24V Power Supply**

**3. منبع تغذیه 24 ولت**

**Adafruit Force Sensor**

**4. سنسور نیروی Adafruit**

**Pop Bumper Assembly**

**5. مونتاژ ضربه گیر پاپ**

**Pop Bumper Switch**

**6. سوئیچ سپر پاپ**

**Target Switch**

**7. سوئیچ هدف**

**Rollover Switch**

**8. سوئیچ رولوور**

**Flipper Assembly**

**9. مونتاژ فلیپر**

**LED Lamps**

**10. لامپ های ال ای دی**

برچسب مرتبط

## Adafruit



لوازم الکترونیکی و کیت های DIY منحصر به فرد و سرگرم کننده -

سخت افزار منبع باز و بهترین آموزش ها در وب

## Gaming



بازی

بازی در بازکن های گاراژ چندان سرگرم کننده نیست.

بهترین پروژه های بازی DIY از سراسر جامعه

از بازی های رومیزی گرفته تا کنسول های بازی ویدیویی دستی،

بهترین هاکستر

## دستگاه پین بال کنترل شده آردوینو

ساخت یک ماشین پین بال با کنترل آردوینو به ترکیبی از مهارت های نجاری، مکانیکی، الکترونیکی و برنامه نویسی نیاز داشت. حدود شش ماه تلاش پاره وقت طول کشید تا ایجاد شود و پیچیده ترین چیزی بود که تا به حال ساخته شد.



همانطور که در تصویر مشاهده می کنید، این دستگاه دارای یک کابینت چوبی است و از قطعات تعویض ماشین پین بال که در بازار موجود است استفاده می کند. این دستگاه توسط یک برد آردوینو مگا 2560 کنترل می شود.

از تخته سه لا با روکش بلوط ساخته شده است. تمام ابعاد در این سند بر حسب اینچ است. پنل جلو و عقب  $23W \times 20H$  هستند.

پانل های جانبی  $20H \times 47L$  هستند.

قطعات با لبه های میتر بریده شده و با استفاده از چسب چوب به هم متصل شدند. پیچ های جیبی و چسب نیز می توانستند کار کند.

دستگاه مورد سوء استفاده زیادی قرار می گیرد بنابراین باید قوی باشد. قبل از اینکه آن را به هم وصل کنم، یک  $\frac{3}{4}$  اینچی را نیز به اندازه 1 اینچ از پایین برش دادم قفسه را چسب نزنید. برای انساط و انقباض باید شناور شود قفسه برای نگه داشتن منبع تغذیه اصلی و منبع تغذیه چراغ LED تزئینی نصب شده در زیر درب استفاده شد. سطح بازی  $42L \times 22W$  است کوتاه تر از کابینت است.

حدود 3 اینچ در جلو سیم هارا تا دکمه های بالن و به چراغ های LED منتقل شود و فضایی را برای مکانیزم پیستون توپ فراهم می کند سطح بازی به سادگی روی چند بلوك چوبی کوچک قرار دارد که به دو طرف حدود 4 اینچ زیر بالای کیس پیچ شده است. برای دستیابی به دستگاه به سادگی درب آن را باز می کنیم زمین بازی را بر می داریم و آن را بر می گردانیم



کیس و سطح بازی همگی در زوایای قائمه ساخته شده اند. زمین بازی همانطور که در این نمای جانبی نشان داده شده توسط پاها فراهم می شود. دستگاه های پین بال می توانند زاویه هایی از ۱ تا ۷ درجه داشته باشند هر چه زاویه بالاتر باشد سرعت بازی بیشتر می شود پاها طوری طراحی شده که قابل تنظیم باشند. خود پاها چوبی هستند و در اکثر فروشگاه های خانگی موجود است. سپس با استفاده از یک مارپیچ سوراخی به عمق ۱۲ اینچ در انتهای هر پایه ایجاد شده. در انتهای پایین از یک اپوکسی دو جزئی برای قرار دادن مهره سه راهی  $\frac{8}{3}$  اینچی استفاده شده. مراقب باشید که هیچ چسبی روی نخ ها یا جایی که سوراخی که دریل کرده اید را مسدود نکند. سپس یک میله نخ  $\frac{8}{3}$  اینچی ۱۲ اینچی را داخل ساق پا فرو شده و یک مهره را روی میله رزوه شد.

صفحات نصب پایه میز به سادگی به قفسه پایینی پیچ می شوند و پایه ها به داخل پیچ می شوند برای تنظیم ارتفاع دستگاه به سادگی مهره را شل کنید و سپس میله رزوه دار را به داخل یا خارج کنید. سپس مهره را دوباره محکم کنید تا مطمئن شوید که میز محکم می ماند.

در تصویر متوجه سیم برق و کابل USB خواهید شد. سیم برق از یک نوار برق می آید که به پشت دستگاه وصل شده. همه چیز در آنجا وصل می شود، به طوری که یک سوئیچ همه چیز را روشن یا خاموش می کند. کابل USB آنجاست تا بتوان برد آردوینو را بدون نیاز به برداشتن زمین بازی دوباره برنامه ریزی کرد.

### The Lid

قاب درب از چوب بلوط ساخته شده است و همان ابعاد بیرونی کابینت است. عرض بالا و کناره ها 1.5 اینچ است در حالی که عرض پایینی آن 5 اینچ است عرض اضافی در پایین برای پنهان کردن شکاف بین زمین بازی و کابینت استفاده می شود. از لبه داخلی قطعات قاب پایین کشیده شد تا پوشش پلکسی گلاس را نگه دارد. به جای شیشه از پلکسی گلاس استفاده شده زیرا سبک تر، ایمن تر و برش راحت تر است. درب با استفاده از پیچ های جیبی کنار هم قرار داده شد این کار را باعث

میشود تا در صورتی که پلکسی گلاس نیاز به تعویض داشت بتوان آن را جدا کرد. همچنین یک نوار از چراغ های LED رنگارنگ را برای اهداف تزئینی زیر درب نصب شده. در نهایت با استفاده از یک لولای پیانو که هم در درب و هم در کیس فرو رفته درب روی کیس نصب شده است



## Arduino

آردوینو مگا 2560 اساساً چهار کار را انجام می دهد تعیین می کند که چه زمانی سوئیچ ها فعال می شوند، چراغها را بر اساس آن روشن یا خاموش می کند، امتیاز را پیگیری می کند، و امتیاز و شماره توپ را

روی یک LCD کوچک نمایش می‌دهد. اکثر سیم کشی‌ها بسیار ساده هستند.

وقتی که تشخیص داد که ورودی از زیاد به پایین تغییر کرده است، متوجه می‌شود که هدف مورد اصابت قرار گرفته است و سپس چراغ پین بال LED مناسب را روشن می‌کند.

تنها عارضه در این مورد مربوط به پاپ بامپر‌ها است. سوئیچ‌های رول اور و اهداف مورد استفاده در این بازی همگی قدرت خود را از آردوبینو می‌گیرند. پاپ بامپر‌ها از یک منبع تغذیه 25 ولت جداگانه تامین می‌شوند زیرا به آب بیشتری نسبت به آردوبینو نیاز دارند. ضربه‌گیرهای پاپ دارای یک سوئیچ هستند که آنها را فعال می‌کند اما به دلیل ولتاژ بالا نمی‌توانم مستقیماً آن را خواند. در نتیجه، اول خروجی را از کلید پاپ بامپر به یک تقسیم کننده ولتاژ میفرستیم.

پس از عبور از تقسیم کننده ولتاژ، ولتاژ به 5 ولت کاهش می‌یابد و آردوبینو می‌تواند آن را مدیریت کند. اگر از ورودی‌های آنالوگ استفاده شود خوانش‌های نادرست کمتری با اینها دریافت می‌شود چون به نظر میرسد تقسیم کننده‌های ولتاژ همیشه مقداری جریان دارند. با ورودی

های آنالوگ، میتوان بررسی کرد که از آستانه فراتر رفته است تا بتوان به راحتی سیگنال را از نویز جدا کرد

در حین بازی اگر همه نوع خاصی از دستگاه (هدف، ضربه گیرهای پاپ یا رول اورها) فعال شده بود آردوینو را برنامه ریزی شد تا تمام چراغ های آن دستگاه ها را فلش کند و سپس امتیاز آنها را افزایش دهد همچنین از یک سنسور نیروی کوچک برای تشخیص زمان گم شدن توپ استفاده کرد.

تعداد توپ های بازی شده را پیگیری کند و تعیین کند که بازی چه زمانی تمام شده است. وقتی بازی تمام می شود همه چراغ ها به ترتیب چشمک میزنند

در نهایت از آردوینو برای نمایش امتیاز و شماره توپ استفاده میشود

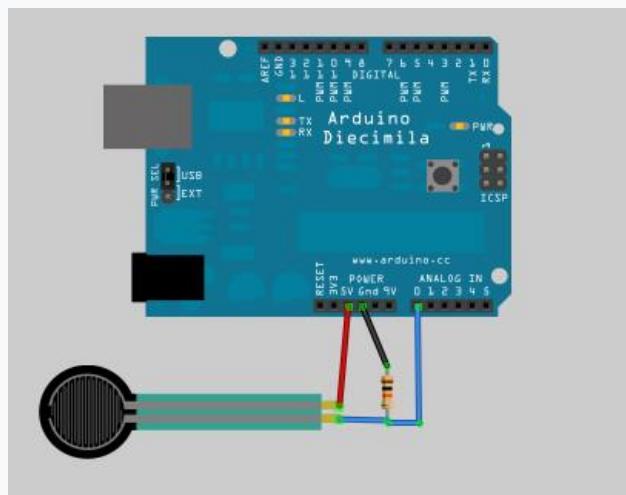
نمودار سیم کشی پیوست شده نحوه سیم کشی یکی از اجزای اصلی را نشان میدهد. اگر همه آنها نمایش داده شوند نمودار بسیار گیج کننده خواهد بود.

## سنسور نیرو

### استفاده از FSR

### روش خواندن ولتاژ آنالوگ

ساده ترین راه برای اندازه گیری یک سنسور مقاومتی این است که یک سر آن را به Power و دیگری را به یک مقاومت کششی به زمین متصل کنید. بعد نقطه بین مقاومت pull-down ثابت و مقاومت متغیر FSR به ورودی آنالوگ یک میکروکنترلر مانند آردوینو متصل میشود



## اندازه ها

اصلی (427×338)

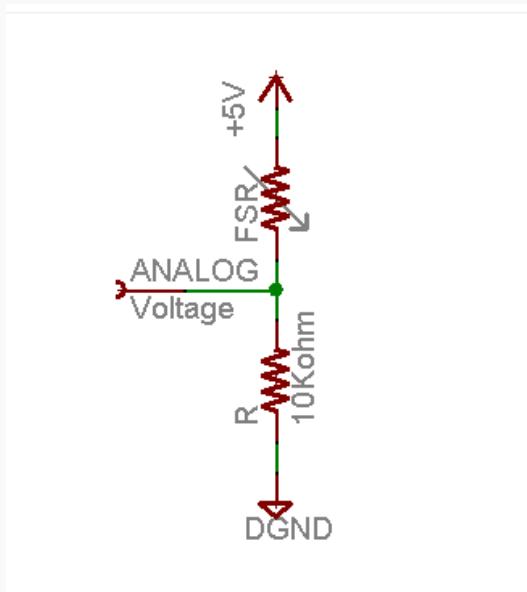
کوچک 360 (360×285)

کوچک 240 (240×190)

160 Thumb (160×127)

100 Thumb (100×79)

100<sup>2</sup> Thumb (100×100)



اصلی (184×334)

کوچک 240 (240×436)

160 Thumb (160×290)

100 Thumb (100×182)

100<sup>2</sup> Thumb (100×100)

برای این مثال آن را با منبع 5 ولت نشان شده اما توجه داشته باشد که می توان از آن با منبع 3.3 ولت به همین راحتی استفاده کرد. در این پیکربندی ولتاژ آنالوگ از 0 ولت (زمین) تا حدود 5 ولت (یا تقریباً مشابه ولتاژ منبع تغذیه) است.

روش کار به این صورت است که با کاهش مقاومت FSR، مقاومت کل FSR و مقاومت pulldown از حدود 100 Kohm به کاهش میباید. این به این معناست که جریان عبوری از هر دو مقاومت افزایش میباید که باعث افزایش ولتاژ در مقاومت K10 ثابت می شود. این کاملاً یک ترفند است

| Force<br>(lb) | Force<br>(N) | FSR<br>Resistance | (FSR + R)<br>ohm | Current<br>thru<br>FSR+R | Voltage<br>across R |
|---------------|--------------|-------------------|------------------|--------------------------|---------------------|
| None          | None         | Infinite          | Infinite!        | 0 mA                     | 0V                  |
| 0.04<br>lb    | 0.2 N        | 30 Kohm           | 40 Kohm          | 0.13 mA                  | 1.3 V               |
| 0.22<br>lb    | 1 N          | 6 Kohm            | 16 Kohm          | 0.31 mA                  | 3.1 V               |
| 2.2 lb        | 10 N         | 1 Kohm            | 11 Kohm          | 0.45 mA                  | 4.5 V               |
| 22 lb         | 100 N        | 250 ohm           | 10.25<br>Kohm    | 0.49 mA                  | 4.9 V               |

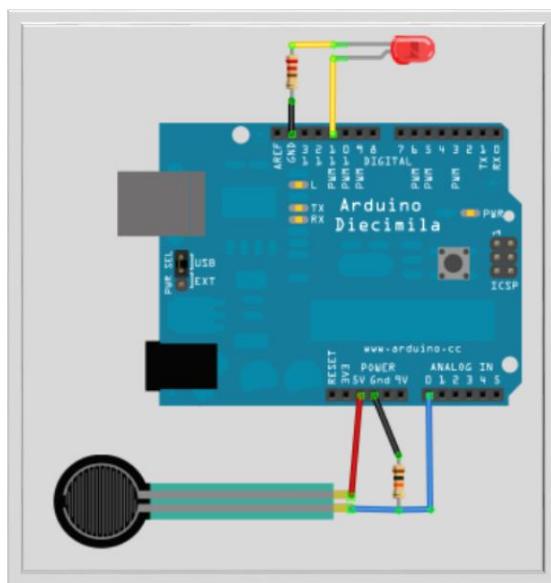
این جدول ولتاژ تقریبی آنالوگ را بر اساس نیرو/ مقاومت سنسور با منبع تغذیه 5 ولت و مقاومت pulldown 10K نشان میدهد.

این روش تا حدودی مقاومت خطی را می گیرد اما ولتاژ خطی ارائه نمیدهد به این دلیل که معادله ولتاژ به صورت زیر است:

$$V_O = V_{CC} \left( R / (R + FSR) \right)$$

یعنی ولتاژ متناسب با معکوس مقاومت FSR است.

FSR را مانند مثال بالا سیم کشی کنید اما این بار یک LED به پین 11 اضافه کنید.



اندازه ها

اصلی (384×390)

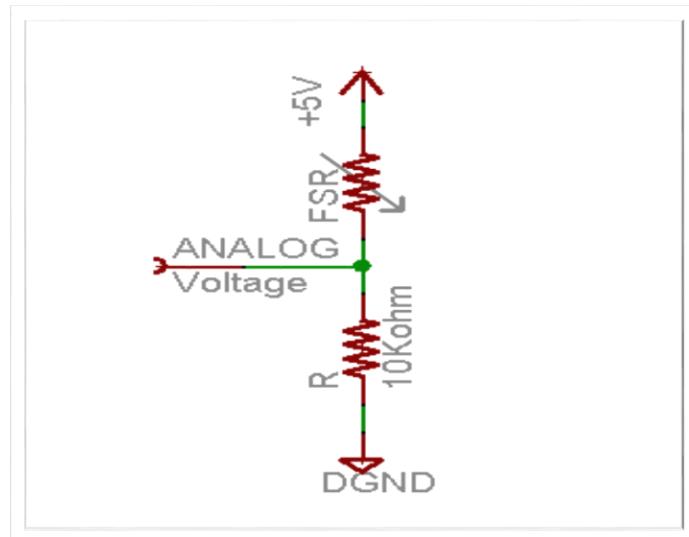
کوچک (360×366) 360

کوچک (240×244) 240

(160×163) 160 Thumb

(120×100) 100 Thumb

(100×100) 100<sup>2</sup> Thumb



اندازه ها

اصلی (311×387)

کوچک (240×299) 240

(160×199) 160 Thumb

(100×124) 100 Thumb

(100×100) 100<sup>2</sup> Thumb

این طرح ولتاژ آنالوگ را می خواند و از آن برای تعیین میزان روشنایی LED قرمز استفاده می کند. هر چه بیشتر روی FSR فشار دهد LED روشن تر می شود LED باید به یک پین PWM متصل شود تا کار کند در این مثال از پایه 11 استفاده شده.

`/* FSR testing sketch.`

Connect one end of FSR to 5V, the other end to Analog 0.

یک سر FSR را به 5 ولت و سر دیگر را به آنالوگ 0 وصل کنید.

Then connect one end of a 10K resistor from Analog 0 to ground

یک سر مقاومت K10 را از آنالوگ 0 به زمین وصل کنید

Connect LED from pin 11 through a resistor to ground

LED را از پایه 11 از طریق یک مقاومت به زمین وصل کنید

`int fsrAnalogPin = 0; // FSR is connected to analog 0`

آنالوگ 0 متصل است

`int LEDpin = 11; // connect Red LED to pin 11 (PWM pin)`

قرمز را به پایه 11 وصل کنید

`int fsrReading; // the analog reading from the FSR resistor divider`

`int LEDbrightness;`

خواندن آنالوگ از تقسیم کننده مقاومت FSR روشنایی LED

`void setup(void) {`

```
Serial.begin(9600); // We'll send debugging information via the Serial monitor
```

اطلاعات اشکال زدایی را از طریق مانیتور سریال ارسال خواهیم کرد

```
pinMode(LEDpin, OUTPUT);  
}  
  
void loop(void) {  
  
    fsrReading = analogRead(fsrAnalogPin);  
  
    Serial.print("Analog reading = ");  
  
    Serial.println(fsrReading);
```

// we'll need to change the range from the analog reading (0-1023) down to the range

ما باید محدوده را از خواندن آنالوگ (0-1023) به محدوده تغییر دهیم

```
// used by analogWrite (0-255) with map!
```

استفاده شده توسط نقشه

```
LEDbrightness = map(fsrReading, 0, 1023, 0, 255);
```

// LED gets brighter the harder you press

```
analogWrite(LEDpin, LEDbrightness);
```

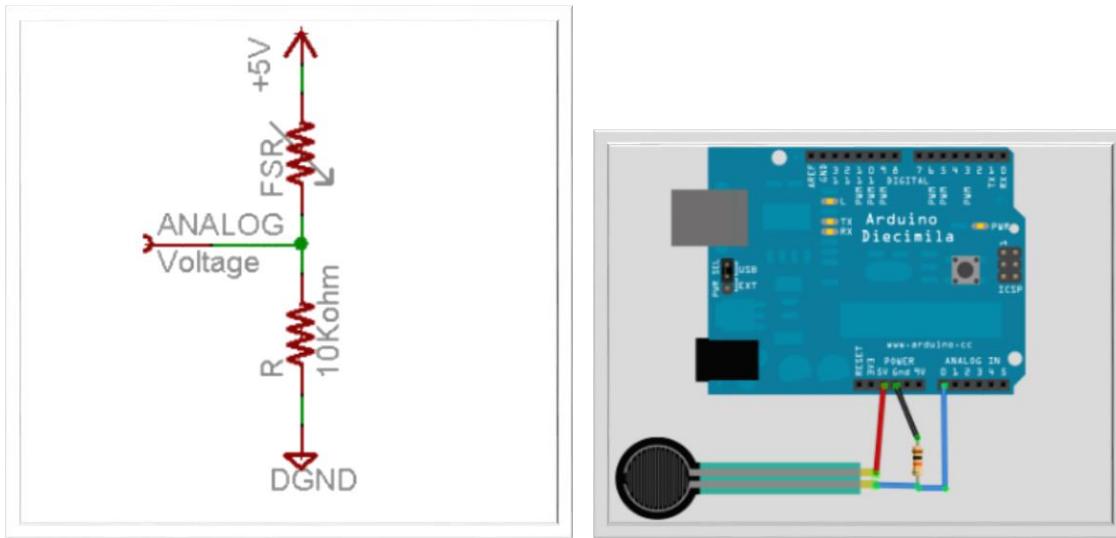
هر چه بیشتر فشار دهید LED روشن تر می شود

```
delay (100);
```

```
}
```

کد ساده برای اندازه گیری FSR آنالوگ

یک نمونه کد برای اندازه گیری FSR در یک پین آنالوگ :



```

Analog reading = 0 - No pressure
Analog reading = 124 - Light touch
Analog reading = 303 - Light squeeze
Analog reading = 655 - Medium squeeze
Analog reading = 736 - Medium squeeze
Analog reading = 901 - Big squeeze
Analog reading = 968 - Big squeeze
Analog reading = 0 - No pressure
Analog reading = 0 - No pressure

```

23

این کد هیچ محاسبه ای انجام نمی دهد فقط آنچه را که به عنوان مقدار فشار تفسیر می کند را به صورت کیفی چاپ می کند برای اکثر پروژه ها این تقریباً تمام چیزی است که مورد نیاز است!

```
/* FSR simple testing sketch.
```

Connect one end of FSR to power, the other end to Analog 0. Then connect one end of a 10K resistor from Analog 0 to ground

یک سر FSR را به برق و سر دیگر را به آنالوگ 0 وصل کنید. سپس یک سر مقاومت K10 را از آنالوگ 0 به زمین وصل کنید

```
int fsrPin = 0; // the FSR and 10K pulldown are connected to a0
```

```
int fsrReading; // the analog reading from the FSR resistor divider
```

خواندن آنالوگ از تقسیم کننده مقاومت

```
void setup(void) {
```

```
// We'll send debugging information via the Serial moni
```

اطلاعات اشکال زدایی را از طریق مانیتور سریال ارسال خواهیم کرد

```
tor Serial.begin(9600); }
```

```
void loop(void) {
```

```
fsrReading = analogRead(fsrPin);
```

خواندن آنالوگ

```
Serial.print("Analog reading = ");
```

خواندن انالوگ خام

```
Serial.print(fsrReading); // the raw analog reading
```

اطلاعات اشکال زدایی

```
// We'll have a few thresholds, qualitatively determined
```

```
if (fsrReading < 10) {
```

```
Serial.println(" - No pressure");
```

بدون فشار

```
} else if (fsrReading < 200) {
```

```
Serial.println(" - Light touch");
```

لمس سبک

```
} else if (fsrReading < 500) {
```

```
Serial.println(" - Light squeeze");
```

فشار سبک

```
} else if (fsrReading < 800) {
```

```
Serial.println(" - Medium squeeze");
```

فشار متوسط

```
} else {
```

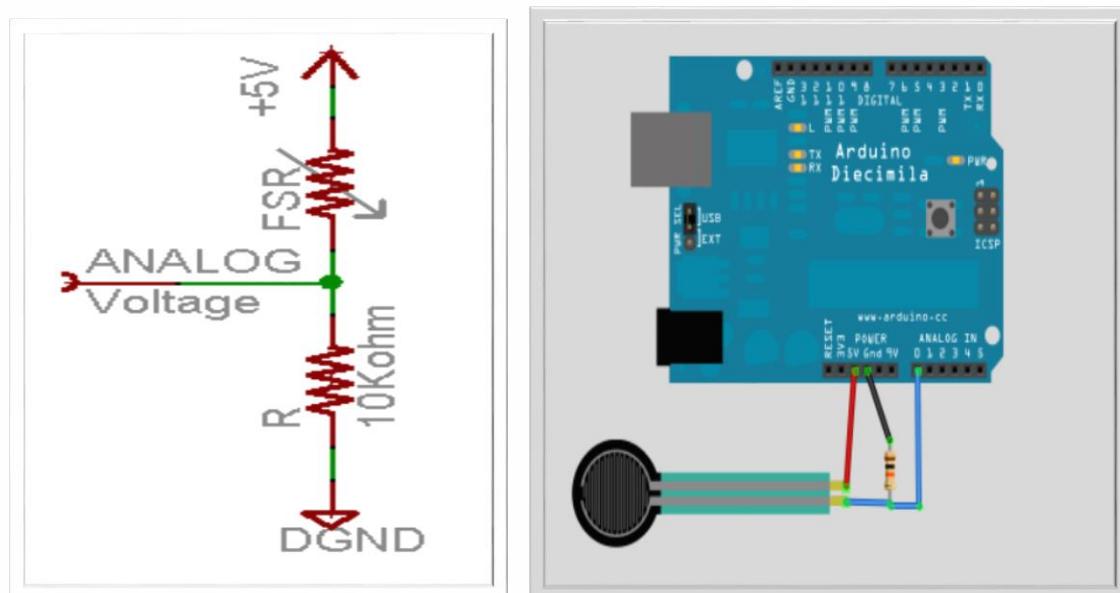
```
Serial.println(" - Big squeeze");
```

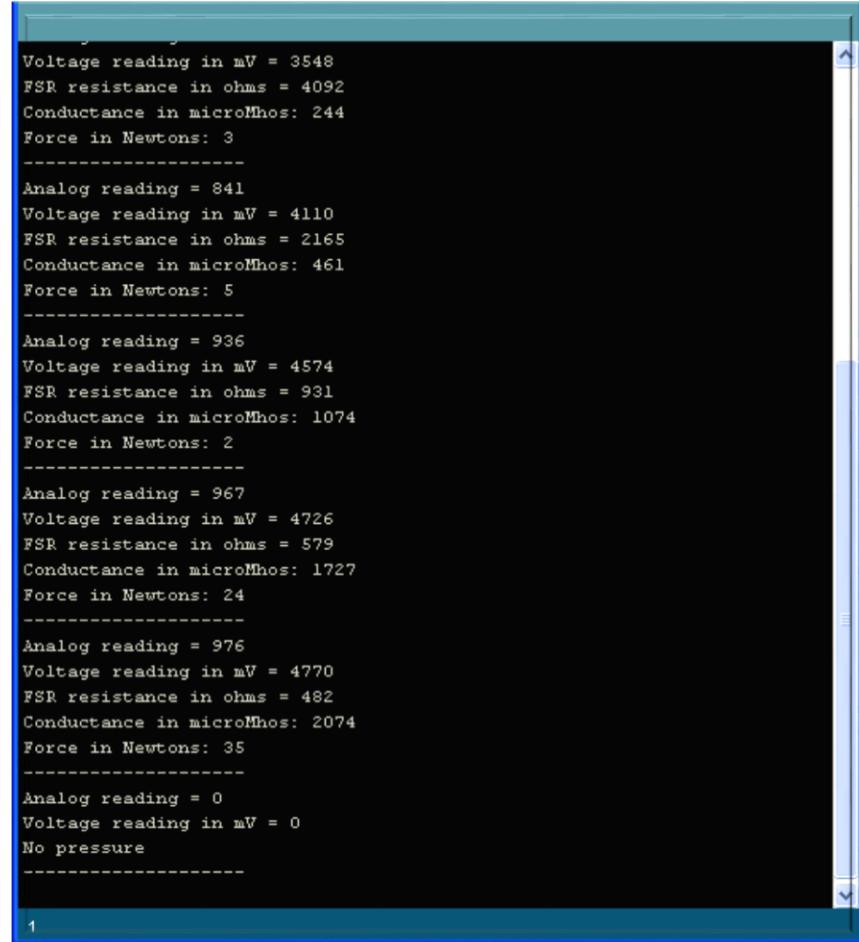
فشار بزرگ

```
} delay (1000);}
```

## کد عمیق برای اندازه گیری های FSR آنالوگ

این طرح آردوینو که فرض می کند FSR را مانند بالا با K10 سیم کشی کرده اید؟ مقاومت را پایین بیاورید و سنسور روی پین آنالوگ ۰ خوانده می شود. این بسیار پیشرفته است و نیروی تقریبی نیوتون اندازه گیری شده توسط FSR را اندازه گیری می کند. این می تواند برای کالیبره کردن نیروهایی که فکر می کنید FSR تجربه خواهد کرد بسیار مفید باشد.





Voltage reading in mV = 3548  
FSR resistance in ohms = 4092  
Conductance in microMhos: 244  
Force in Newtons: 3  
-----  
Analog reading = 841  
Voltage reading in mV = 4110  
FSR resistance in ohms = 2165  
Conductance in microMhos: 461  
Force in Newtons: 5  
-----  
Analog reading = 936  
Voltage reading in mV = 4574  
FSR resistance in ohms = 931  
Conductance in microMhos: 1074  
Force in Newtons: 2  
-----  
Analog reading = 967  
Voltage reading in mV = 4726  
FSR resistance in ohms = 579  
Conductance in microMhos: 1727  
Force in Newtons: 24  
-----  
Analog reading = 976  
Voltage reading in mV = 4770  
FSR resistance in ohms = 482  
Conductance in microMhos: 2074  
Force in Newtons: 35  
-----  
Analog reading = 0  
Voltage reading in mV = 0  
No pressure  
-----

```
/* FSR testing sketch.
```

Connect one end of FSR to power, the other end to Analog 0.

یک سر FSR را به برق و سر دیگر را به Analog 0 وصل کنید.

Then connect one end of a 10K resistor from Analog 0 to ground

یک سر مقاومت K10 را از آنالوگ 0 به زمین وصل کنید

```
int fsrPin = 0;      // the FSR and 10K pulldown are connected to a0
```

```
int fsrReading;     // the analog reading from the FSR resistor divider
```

خواندن آنالوگ از تقسیم کننده مقاومت FSR

```
int fsrVoltage;    // the analog reading converted to voltage
```

خواندن آنالوگ به ولتاژ تبدیل شد

```
unsigned long fsrResistance; // The voltage converted to resistance, can be
```

```
very big so make "long"
```

ولتاژ تبدیل شده به مقاومت، می تواند بسیار بزرگ باشد، بنابراین "طول" را ایجاد کنید

```
unsigned long fsrConductance;
```

```
long fsrForce;       // Finally, the resistance converted to force
```

سرانجام مقاومت به نیرو تبدیل شد

```
void setup(void) {
```

```
Serial.begin(9600); // We'll send debugging information via the Serial
```

اطلاعات اشکال زدایی را از طریق مانیتور سریال ارسال خواهیم کرد  
monitor

}

void loop(void) {

    fsrReading = analogRead(fsrPin);

    Serial.print("Analog reading = ");

خواندن آنالوگ

    Serial.println(fsrReading);

    // analog voltage reading ranges from about 0 to 1023 which maps to 0V to 5V

    (= 5000mV)

حدوده خواندن ولتاژ آنالوگ از حدود 0 تا 1023 است که به 0 ولت تا 5 ولت (= 5000 میلی ولت)  
نشان داده می شود.

    fsrVoltage = map(fsrReading, 0, 1023, 0, 5000);

    Serial.print("Voltage reading in mV = ");

خواندن ولتاژ

    Serial.println(fsrVoltage);

    if (fsrVoltage == 0) {

```

Serial.println("No pressure");                                بدون فشار

} else {

// The voltage = Vcc * R / (R + FSR) where R = 10K and Vcc = 5V

// so FSR = ((Vcc - V) * R) / V          yay math!

fsrResistance = 5000 - fsrVoltage;      // fsrVoltage is in millivolts so

5V = 5000mV

fsrResistance *= 10000;                  // 10K resistor

fsrResistance /= fsrVoltage;

Serial.print("FSR resistance in ohms = ");           مقاومت در اهم

Serial.println(fsrResistance);

fsrConductance = 1000000;                  // we measure in micromhos so

fsrConductance /= fsrResistance;

Serial.print("Conductance in microMhos: ");

Serial.println(fsrConductance);

```

```
// Use the two FSR guide graphs to approximate the force
```

```
if (fsrConductance <= 1000) {
```

از دو نمودار راهنمای FSR برای تقریب نیرو استفاده کنید

```
    fsrForce = fsrConductance / 80;
```

نیرو در نیوتن

```
    Serial.println("Force in Newtons: ");
```

```
} else {
```

```
    fsrForce = fsrConductance - 1000;
```

```
    fsrForce /= 30;
```

```
    Serial.print("Force in Newtons: ");
```

نیرو در نیوتن

```
    Serial.println(fsrForce);
```

```
}
```

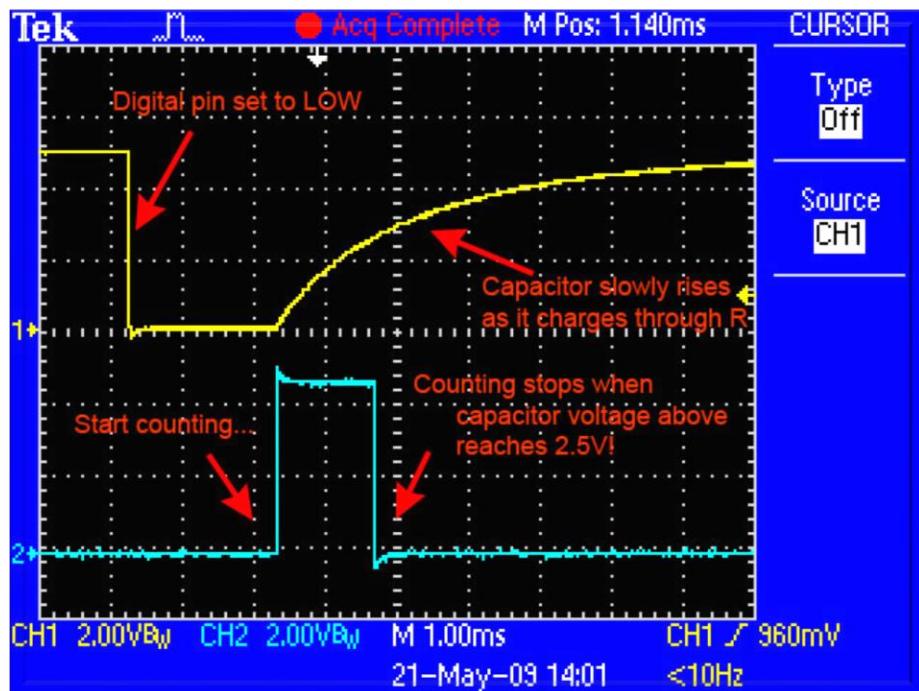
```
}
```

```
Serial.println("-----");
```

```
delay(1000);}
```

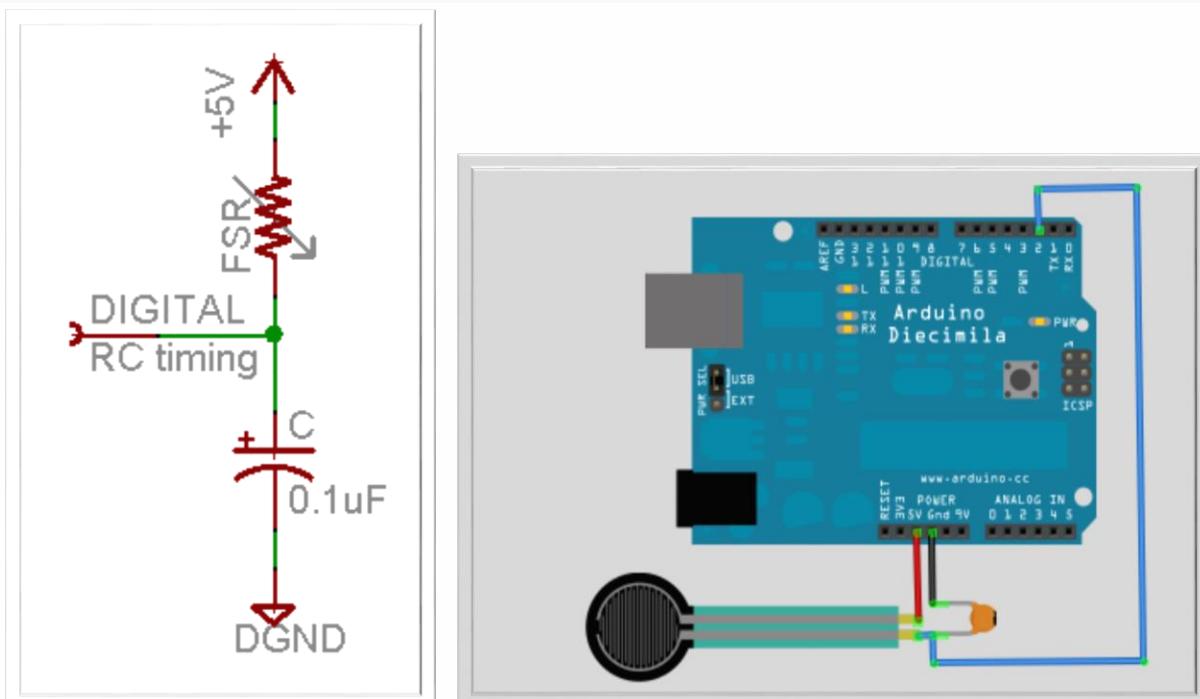
## خواندن اندازه گیری های FSR بدون پین های آنالوگ

از آنجایی که FSR ها اساساً مقاومت هستند میتوان از آنها استفاده کرد حتی اگر هیچ پایه آنالوگ روی میکروکنترلر خود ندارد ( اگر می خواهید بیشتر از پین های ورودی آنالوگ وصل شوید. روشی که ما این کار را انجام می دهیم با استفاده از مزایای یک ویژگی الکترونیکی اولیه از مقاومت ها و خازن ها به نظر می رسد که اگر خازن را در ابتدا بدون ولتاژ ذخیره کنید و سپس آن را از طریق یک مقاومت به برق وصل کنید به آرامی تا ولتاژ برق شارژ می شود. کندر است



این عکس برداری از یک اسیلوسکوپ نشان می دهد که روی پین دیجیتال (زرد) چه اتفاقی می افتد. خط آبی نشان می دهد که طرح شروع به شمارش می کند و زمانی که کوتینگ کامل می شود حدود 1.2 میلی ثانیه بعد.

خازن مانند یک سطل عمل می کند و مقاومت مانند یک لوله نازک است. پر کردن یک سطل با یک لوله بسیار نازک به زمان کافی نیاز دارد که می توانید با زمان بندی مدت زمانی که طول می کشد تا سطل تا نیمه پر شود عرض لوله را بفهمید.



"سطل" ما یک خازن سرامیکی  $0.1\text{ uF}$  است. شما می‌توانید خازن را تقریباً به هر شکلی که می‌خواهید تغییر دهید اما مقادیر زمان بندی نیز تغییر می‌کند. به نظر می‌رسد  $0.1\text{ uF}$  مکان خوبی برای شروع این ها باشد FSR

```
/* FSR simple testing sketch.
```

Connect one end of FSR to power, the other end to pin 2.

Then connect one end of a  $0.1\mu\text{F}$  capacitor from pin 2 to ground

For more information see [www.ladyada.net/learn/sensors/fsr.html](http://www.ladyada.net/learn/sensors/fsr.html) \*/

```
int fsrPin = 2;      // the FSR and cap are connected to pin2

int fsrReading;     // the digital reading

int ledPin = 13;    // you can just use the 'built in' LED

void setup(void) {

  // We'll send debugging information via the Serial monitor

  Serial.begin(9600);

  pinMode(ledPin, OUTPUT); // have an LED for output }
```

```
void loop(void) {  
  
    // read the resistor using the RCtime technique  
  
    fsrReading = RCtime(fsrPin);  
  
    if (fsrReading == 30000) {  
  
        // if we got 30000 that means we 'timed out'  
  
        Serial.println("Nothing connected!");  
  
    } else {  
  
        Serial.print("RCtime reading = ");  
  
        Serial.println(fsrReading);      // the raw analog reading  
  
        // Do a little processing to keep the LED blinking  
  
        fsrReading /= 10;  
  
        // The more you press, the faster it blinks!  
  
        digitalWrite(ledPin, HIGH);  
  
        delay(fsrReading);  
  
        digitalWrite(ledPin, LOW);
```

```

    delay(fsrReading); }

delay(100);}

// Uses a digital pin to measure a resistor (like an FSR or photocell!)

// We do this by having the resistor feed current into a capacitor and

// counting how long it takes to get to Vcc/2 (for most arduinos, thats 2.5V)

int RCtime(int RCpin) {

    int reading = 0; // start with 0

    // set the pin to an output and pull to LOW (ground)

    pinMode(RCpin, OUTPUT);

    digitalWrite(RCpin, LOW);

    // Now set the pin to an input and...

    pinMode(RCpin, INPUT);

    while (digitalRead(RCpin) == LOW) { // count how long it takes to rise up to

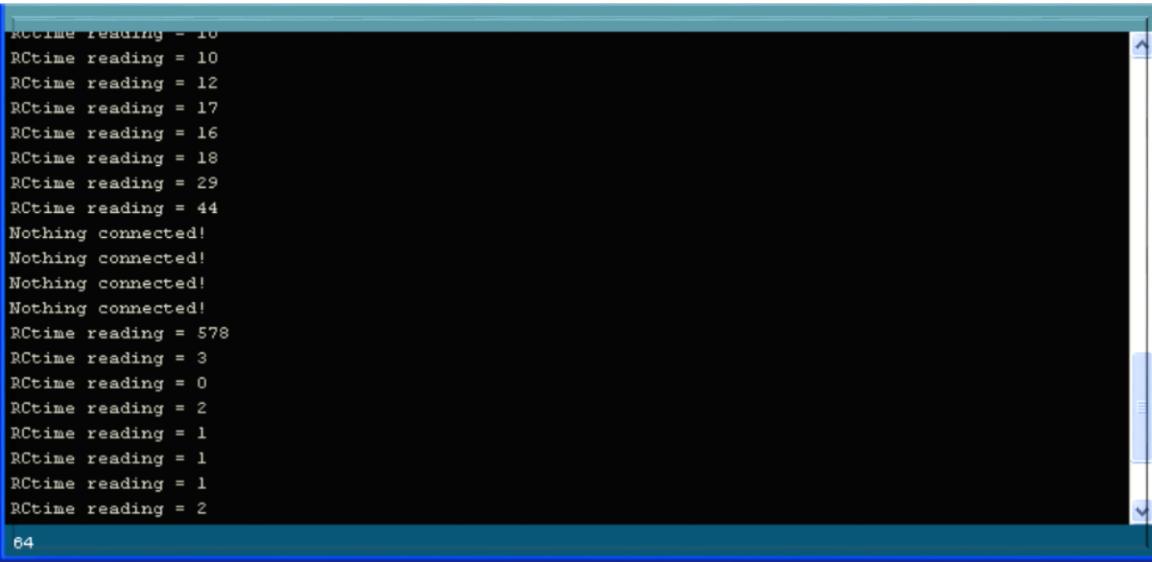
HIGH

    reading++; // increment to keep track of time
}

```

```
if (reading == 30000) {  
  
    // if we got this far, the resistance is so high  
  
    // its likely that nothing is connected!  
  
    break;          // leave the loop }  
  
}  
  
// OK either we maxed out at 30000 or hopefully got a reading, return the  
count  
  
return reading;}
```

---



```
RCtime reading = 10
RCtime reading = 10
RCtime reading = 12
RCtime reading = 17
RCtime reading = 16
RCtime reading = 18
RCtime reading = 29
RCtime reading = 44
Nothing connected!
Nothing connected!
Nothing connected!
Nothing connected!
RCtime reading = 578
RCtime reading = 3
RCtime reading = 0
RCtime reading = 2
RCtime reading = 1
RCtime reading = 1
RCtime reading = 2
RCtime reading = 2
64
```

می توان مقاومت واقعی را از روی خواندن محاسبه کرد اما متأسفانه تغییرات در IDE و برد آردوینو آن را ناسازگار می کند. توجه داشته باشید که اگر نسخه های IDE سیستم عامل را تغییر دهید یا از آردوینو 3.3 ولتی به جای 5 ولت استفاده کنید یا از آردوینو 16 مگاهرتزی به 8 مگاهرتز (مانند لیلی پد) تغییر دهید ممکن است به دلیل مدت زمانی که خواندن آن طول می کشد تفاوت هایی وجود داشته باشد. ارزش یک پین معمولاً این مسئله مهمی نیست اما اگر انتظارش را ندارید می تواند اشکال زدایی پروژه شما را سخت کند

## منبع تغذیه

برای تغذیه اجزای پین بال با یک منبع تغذیه 24 ولت استفاده شده. این واحد قدرت زیادی برای باله‌ها داشت. برای ضربه گیرهای پاپ کمی ضعیف است ممکن است از منبع تغذیه قوی تری برای این موارد استفاده کنید باید از  تقسیم کننده های ولتاژ را بسازید زیرا فقط توانستم آنهایی را پیدا کنم که حداقل 25 ولت داشته باشند.

به جای اینکه همه چیز را مستقیماً به منبع تغذیه وصل کنیم سیم‌ها را از آن به یک جفت اتوبوس متصل کرده تا اینکه در قسمت زیرین زمین بازی نصب شده این امر اتصال قطعات مختلف را بسیار آسان تر کرد

## منبع تغذیه 24 ولت - تک خروجی 6.5 آمپر



این منبع تغذیه 150 واتی 24 ولتی/ 6.5 آمپری با قابلیت اطمینان بالا جدید است. آنها دارای تاییدیه UL و CE هستند و 100٪ بار کامل تست شده و با اضافه بار / بیش از حد و ولتاژ / اتصال کوتاه محافظت می‌شوند...

## تقسیم کننده های ولتاژ

تقسیم کننده ولتاژ یک مدار ساده است که یک ولتاژ بزرگ را به یک ولتاژ کوچکتر تبدیل می‌کند. فقط با استفاده از دو مقاومت سری و یک ولتاژ ورودی می‌توان ولتاژ خروجی ایجاد کر که کسری از ورودی است تقسیم کننده های ولتاژ یکی از اساسی‌ترین مدارها در الکترونیک هستند اگر یادگیری قانون اهم مانند آشنا شدن با ABC بود یادگیری در مورد تقسیم کننده های ولتاژ مانند یادگیری املای گربه است.

اینها نمونه‌هایی از پتانسیومترها هستند - مقاومت‌های متغیری که می‌توانند برای ایجاد یک تقسیم کننده ولتاژ قابل تنظیم استفاده شوند



Thumb Joystick

Trimpot 10K Ohm with Knob

Rotary Potentiometer - Linear (10k ohm)

Slide Pot - Small (10k Linear Taper)

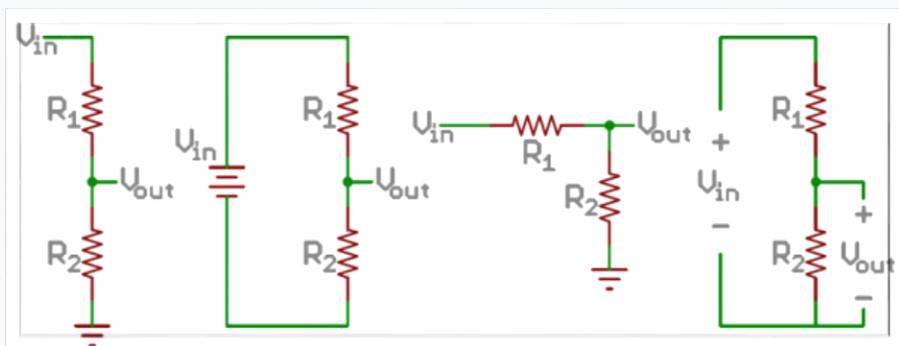
## تقسیم کننده ولتاژ ایده آل

دو بخش مهم در تقسیم کننده ولتاژ وجود دارد:

مدار و معادله

مدار

یک تقسیم کننده ولتاژ شامل اعمال یک منبع ولتاژ در یک سری از دو مقاومت است ممکن است آن را به روش‌های مختلف ترسیم کنید  
اما آنها باید اساساً یک مدار باشند.



نمونه هایی از شماتیک های تقسیم کننده ولتاژ کوتاه، دست بلند، مقاومت در زوایای یکسان/متفاوت و غیره.

ما نزدیکترین مقاومت به ولتاژ ورودی  $V_{in}$  را  $R_1$  و نزدیکترین مقاومت به زمین را  $R_2$  می نامیم. افت ولتاژ در  $V_{out}$  در  $R_2$  نامیده می شود این ولتاژ تقسیم شده ای است که مدار ما برای ایجاد آن وجود دارد.

### معادله

معادله تقسیم کننده ولتاژ فرض می کند که شما سه مقدار مدار فوق را می دانید: ولتاژ ورودی  $V_{in}$  و هر دو مقدار مقاومت  $R_1$  و  $R_2$  با توجه به این مقادیر می توانیم از این معادله برای یافتن ولتاژ خروجی استفاده کنیم:

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

این معادله بیان می کند که ولتاژ خروجی با ولتاژ ورودی و نسبت  $R_1$  و  $R_2$  نسبت مستقیم دارد. اگر می خواهید بدانید که این از کجا می آید این بخش را که در آن معادله مشتق شده است بررسی کنید.

## ساده سازی ها

چند کلیت وجود دارد که خوب است هنگام استفاده از تقسیم کننده ولتاژ به خاطر داشته باشید اینها ساده سازی هایی هستند که ارزیابی مدار تقسیم ولتاژ را کمی آسان تر می کنند

$$if R_1 = R_2 : V_{out} = V_{in} \cdot \frac{R}{2R} = \frac{V_{in}}{2}$$

ابتدا اگر  $R_2$  و  $R_1$  برابر باشند ولتاژ خروجی نصف ولتاژ ورودی است این بدون توجه به مقادیر مقاومت ها درست است

$$if R_2 \gg R_1 : V_{out} \approx V_{in} \cdot \frac{R_2}{R_2} = V_{in}$$

اگر  $R_2$  بسیار بزرگتر از  $R_1$  باشد ولتاژ خروجی بسیار نزدیک به ورودی خواهد بود ولتاژ بسیار کمی در  $R_1$  وجود خواهد داشت

$$if R_2 \ll R_1 : V_{out} \approx V_{in} \cdot \frac{0}{R_1} = 0$$

ولی اگر  $R_2$  بسیار کوچکتر از  $R_1$  باشد ولتاژ خروجی در مقایسه با ورودی ناچیز خواهد بود بیشتر ولتاژ ورودی روی  $R_1$  خواهد بود

## برنامه های کاربردی

تقسیم‌کننده‌های ولتاژ کاربردهای زیادی دارند آنها از رایج‌ترین مدارهایی هستند که مهندسان برق استفاده می‌کنند در اینجا فقط تعدادی از مکانهای متعددی که تقسیم‌کننده‌های ولتاژ را پیدا خواهید کرد آورده شده است

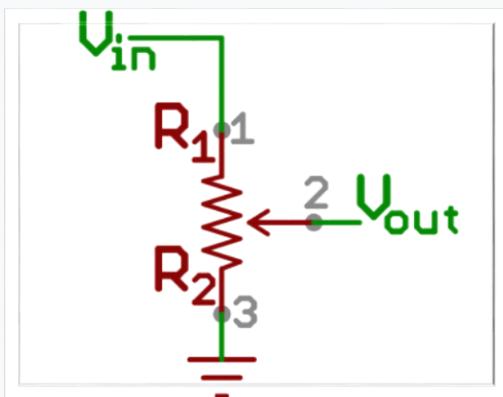
## پتانسیومترها

پتانسیومتر یک مقاومت متغیر است که می‌تواند برای ایجاد یک تقسیم‌کننده ولتاژ قابل تنظیم استفاده شود



تعداد زیادی پتانسیومتر از بالا-چپ در جهت عقربه‌های ساعت داخل یک مقاومت تک و یک برف پاک کن وجود دارد که مقاومت را به دو نیم می‌کند و برای تنظیم نسبت بین هر دو نیمه حرکت می‌کند.

در خارج معمولاً سه پایه وجود دارد دو پایه به هر انتهای مقاومت وصل می شود در حالی که سومین پایه به برف پاک کن دیگ متصل می شود.



نماد شماتیک پتانسیومتر پایه های 1 و 3 انتهای مقاومت هستند. پایه 2 به برف پاک کن متصل می شود

اگر پایه های بیرونی به یک منبع ولتاژ یکی به زمین دیگری به  $V_{in}$  متصل شوند خروجی  $V_{out}$  در پایه میانی یک تقسیم کننده ولتاژ را تقلید می کند. دیگ را تا انتها در یک جهت بچرخانید ممکن است ولتاژ صفر شود. چرخش به سمت دیگر ولتاژ خروجی به ورودی نزدیک می شود.

پتانسیومترها در بسته بندی های مختلف عرضه می شوند و کاربردهای زیادی برای خود دارند. آنها ممکن است برای ایجاد یک ولتاژ مرجع

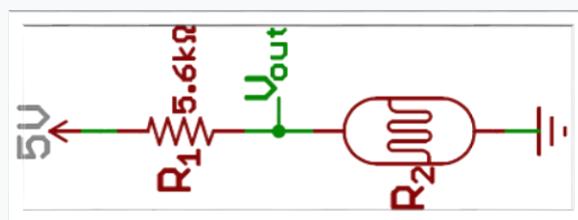
تنظیم ایستگاه های رادیویی، اندازه گیری موقعیت روی یک جوی استیک، یا در هزاران برنامه دیگر که به ولتاژ ورودی متغیر نیاز دارند استفاده شوند.

## خواندن سنسورهای مقاومتی

بسیاری از حسگرها در دنیای واقعی دستگاه های مقاومتی ساده ای هستند. فتوسل یک مقاومت متغیر است که مقاومتی متناسب با میزان نوری که حس می کند ایجاد می کند دستگاه های دیگری مانند سنسورهای انعطاف پذیر، مقاومت های حساس به نیرو و ترمیستورها نیز مقاومت های متغیر هستند.

به نظر می رسد اندازه گیری ولتاژ برای میکروکنترلرهای (آنها یی که مبدل های آنالوگ به دیجیتال - حداقل ADC دارند) بسیار آسان است. با افزودن یک مقاومت دیگر به سنسورهای مقاومتی می توان یک تقسیم کننده ولتاژ ایجاد کنیم هنگامی که خروجی تقسیم کننده ولتاژ مشخص شد، می توان به عقب برگردیم و مقاومت سنسور را محاسبه کنیم.

به عنوان مثال، مقاومت فتوسل بین  $1\text{k}\Omega$  در نور و حدود  $10\text{k}\Omega$  در تاریکی متغیر است. اگر آن را با مقاومت ایستا در جایی در وسط ترکیب کنیم - مثلاً  $5.6\text{k}\Omega$  اهم می‌توانیم طیف وسیعی از تقسیم‌کننده ولتاژ ایجاد شده را دریافت کنیم



فتول نیمی از این تقسیم‌کننده ولتاژ را تشکیل می‌دهد ولتاژ برای یافتن مقاومت سنسور نور اندازه گیری می‌شود

| Light Level | $R_2$ (Sensor)     | $R_1$ (Fixed)       | Ratio $R_2/(R_1+R_2)$ | $V_{out}$ |
|-------------|--------------------|---------------------|-----------------------|-----------|
| Light       | $1\text{k}\Omega$  | $5.6\text{k}\Omega$ | 0.15                  | 0.76 V    |
| Dim         | $7\text{k}\Omega$  | $5.6\text{k}\Omega$ | 0.56                  | 2.78 V    |
| Dark        | $10\text{k}\Omega$ | $5.6\text{k}\Omega$ | 0.67                  | 3.21 V    |

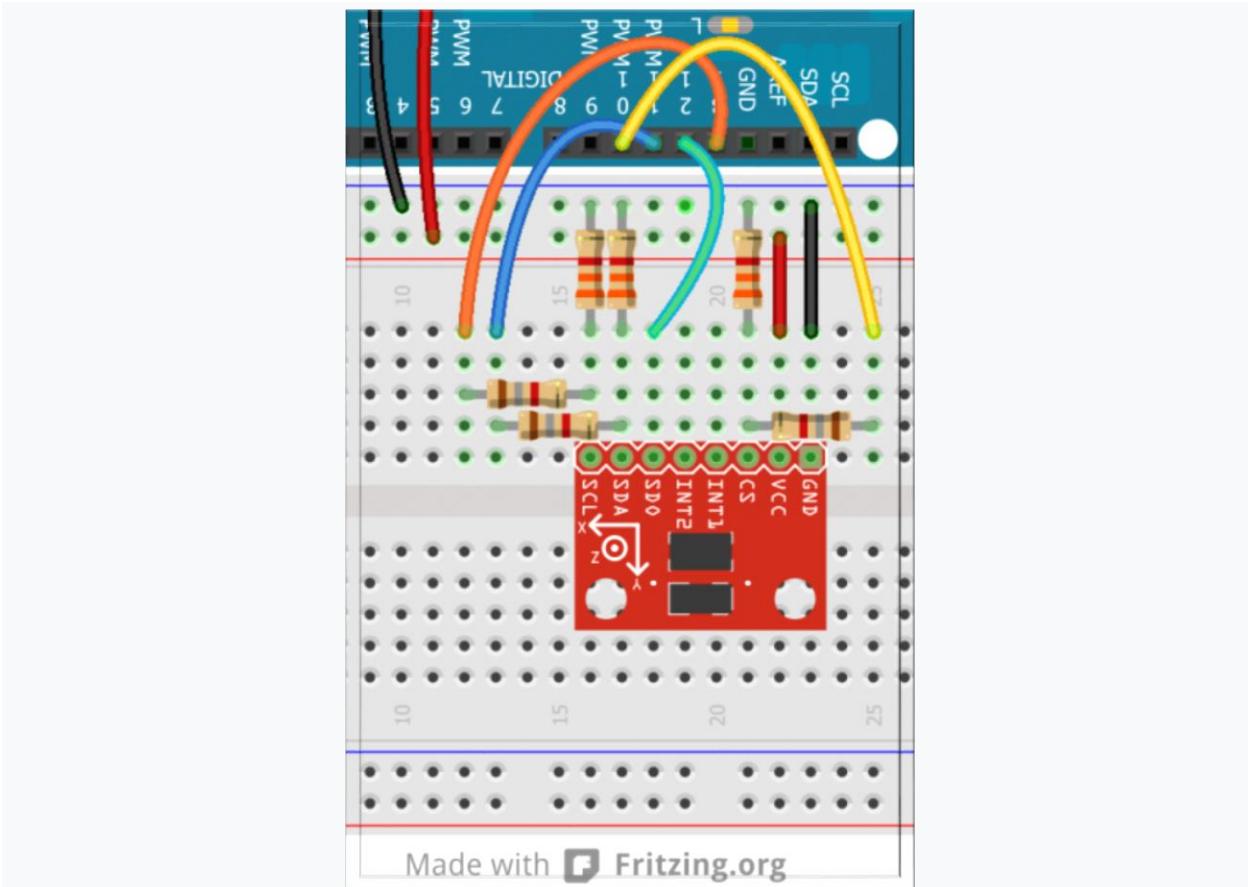
یک نوسان در حدود 2.45 ولت از روشن به تاریکی وضوح زیاد برای اکثر ADC ها

## تغییر سطح

سنسورهای پیچیده‌تر ممکن است خوانش‌های خود را با استفاده از رابطهای سریال سنگین‌تر مانند UART، SPI یا I2C منتقل کنند. بسیاری از این سنسورها به منظور صرفه جویی در مصرف برق با ولتاژ نسبتاً کم کار می‌کنند. متأسفانه غیرعادی نیست که آن سنسورهای ولتاژ پایین در نهایت با یک میکروکنترلر که با ولتاژ سیستم بالاتر کار می‌کند ارتباط برقرار می‌کنند. این منجر به مشکل تغییر سطح می‌شود که راه حل‌های مختلفی از جمله تقسیم ولتاژ دارد.

به عنوان مثال

شتاپ‌سنج ADXL345 حداقل ولتاژ ورودی 3.3 ولت را امکان‌پذیر می‌کند بنابراین اگر سعی کنید آن را با آردوینو وصل کنید (فرض کنید در ولتاژ 5 ولت کار می‌کند) باید کاری انجام دهید تا سیگنال 5 ولت را به 3.3 ولت کاهش دهد تقسیم کننده ولتاژ تنها چیزی که نیاز است دو مقاومت است که نسبت آنها سیگنال 5 ولت را به حدود 3.3 ولت تقسیم کند مقاومت‌هایی در محدوده  $1\text{k}\Omega - 10\text{k}\Omega$  معمولاً برای چنین کاربردهایی بهترین هستند.



مقادیر مقاومت های  $R_1 = 1.8 \text{ k}\Omega$  و  $R_2 = 3.3 \text{ k}\Omega$  نارنجی، قرمز هستند. نمونه ای از تقسیم کننده های ولتاژ در تخته نان، سیگنال های 5 ولتی را به 3.24 ولت تغییر می دهد. این راه حل فقط در یک جهت کار می کند. یک تقسیم کننده ولتاژ به تنها یی هرگز نمی تواند ولتاژ کمتر را به ولتاژ بالاتر برساند.

## Dont's برنامه

استفاده از تقسیم کننده ولتاژ برای پایین آوردن مثلاً منبع تغذیه 12 ولت به 5 ولت ممکن است و سوشه انگیز باشد، اما نباید از تقسیم کننده های ولتاژ برای تامین برق یک بار استفاده کرد

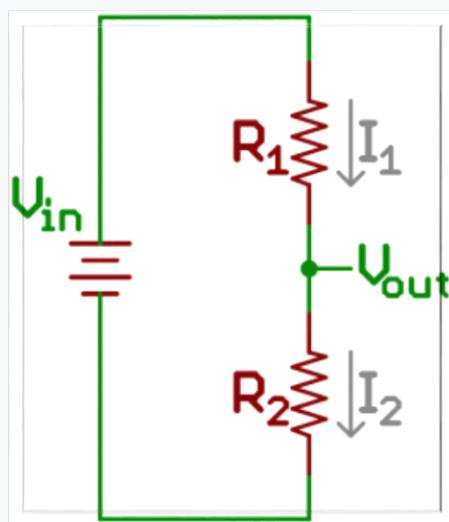
هر جریانی که بار نیاز دارد باید از R1 عبور کند جریان و ولتاژ روی R1 توان تولید می کند که به صورت گرما تلف می شود اگر این توان از درجه بندی مقاومت بیشتر شود (معمولاً بین  $\frac{1}{8}$  وات و 1 وات) گرما شروع به تبدیل شدن به یک مشکل بزرگ می کند و به طور بالقوه مقاومت ضعیف را ذوب می کند

این حتی به ناکارآمدی یک منبع تغذیه - تقسیم کننده ولتاژ - اشاره نمی کند اساساً از تقسیم کننده ولتاژ به عنوان منبع ولتاژ برای هر چیزی که حتی به مقدار متوسطی برق نیاز دارد استفاده نکنید اگر برای استفاده از آن به عنوان منبع تغذیه نیاز به کاهش ولتاژ دارید به تنظیم کننده های ولتاژ یا منابع سوئیچینگ نگاه نکنید

## ارزیابی مدار

اگر بخواهید ولتاژ را در  $V_{out}$  اندازه گیری کنید چه؟  
چگونه می توان از قانون اهم برای ایجاد فرمولی برای محاسبه ولتاژ آنجا استفاده کرد؟

فرض کنید که مقادیر  $V_{in}$ ,  $R_1$  و  $R_2$  را می دانیم بنابراین اجازه دهید معادله  $V_{out}$  خود را بر حسب این مقادیر بدست آوریم  
با کشیدن جریان های مدار--1 و --2 اکه جریان های مقاومت های مربوطه را می نامیم شروع کنیم

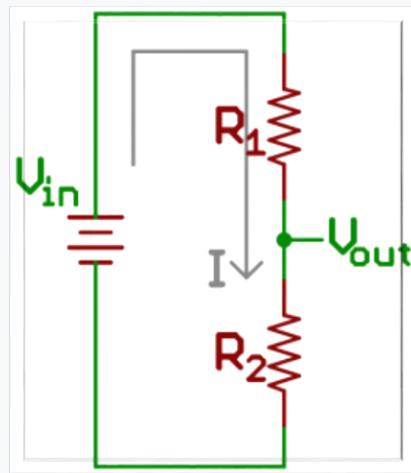


هدف ما محاسبه  $V_{out}$  است اگر قانون اهم را برای آن ولتاژ اعمال کنیم چه؟ فقط یک مقاومت و یک جریان درگیر است

$$V_{out} = I_2 \cdot R_2$$

ارزش  $R_2$  را می‌دانیم اما در مورد ۱۲) چطور؟

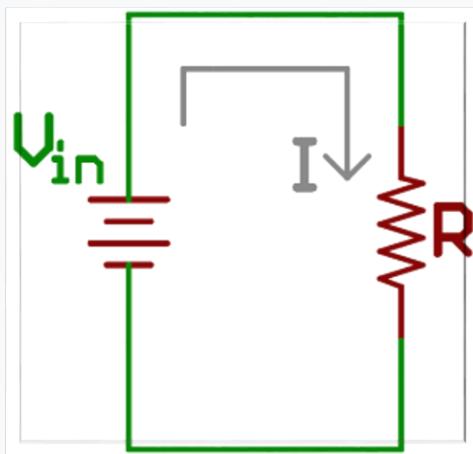
این یک مقدار ناشناخته است می‌توانیم فرض کنیم که ۱۱) معادل ۱۲) است. اما آیا این به ما کمک می‌کند؟ این فکر را نگه دارید مدار ما اکنون به این شکل است جایی که من با ۱۱) و ۱۲) برابر است



ولتاژ دو مقاومت  $R_1$  و  $R_2$  است. این مقاومت‌ها سری هستند. مقاومت‌های سری تا یک مقدار جمع می‌شوند بنابراین می‌توانیم بگوییم

$$R = R_1 + R_2$$

و برای لحظه‌ای می‌توانیم مدار را به صورت زیر ساده کنیم



قانون اهم در ابتدایی ترین حالتش که اگر آن  $R = V_{in} / I$  را به  $R_1 + R_2$  برگردانیم می‌تواند به صورت زیر نوشته شود

$$I = \frac{V_{in}}{R_1 + R_2}$$

و از آنجایی که  $I$  معادل  $|I|$  است آن را به معادله  $V_{out}$  وصل کنید تا به دست آورید

$$V_{out} = R_2 \cdot \frac{V_{in}}{R_1 + R_2}$$

ولتاژ خروجی کسری از ولتاژ ورودی است و آن کسری  $R_2$  تقسیم بر مجموع  $R_1$  و  $R_2$  است.

## توب شوتر یا پیستون

برای عبور پیستون باید از قسمت جلوی کیس سوراخ ایجاد شود سپس به داخل کیس پیچ می شود دقت زیادی باید انجام شود تا مطمئن شوید که پیستون به درستی با زمین بازی در یک راستا قرار دارد یک لبه مستقیم روی زمین بازی قرار دهید و نقطه ای را که به داخل کیس برخورد می کند علامت بزنید سپس به اندازه ای بالا بروید که وسط پیستون از نظر ارتفاع با نصف ارتفاع پین بال برابر شود.  
(سایز پین بال ها 1 و 1/16 اینچ است و در پین بال موجود است)

## مجموعه تیرانداز توب - کروم



## توب پین بال

فولاد کربنی سخت شده استاندارد صنعت با قطر ۱۶/۱-۱ اینچ (۲۷ میلی‌متر) ۸۰.۶ گرم پین بال دارد در تمام ماشین‌های پین بال دارد تجاری معاصر که پس از سال ۱۹۵۰ ساخته شده‌اند استفاده می‌شود از مشخصات کارخانه برای سختی، گرد بودن و وزن فراتر می‌رود یک توب زنگ زده خراشیده یا سوراخ شده به سرعت زمین بازی شما را خراب می‌کند تعویض توب قدیمی و تمیز کردن منظم زمین بازی بهترین دفاع در برابر سایش زمین بازی است

توب پین بال استاندارد ۱۶/۱-۱ اینچ

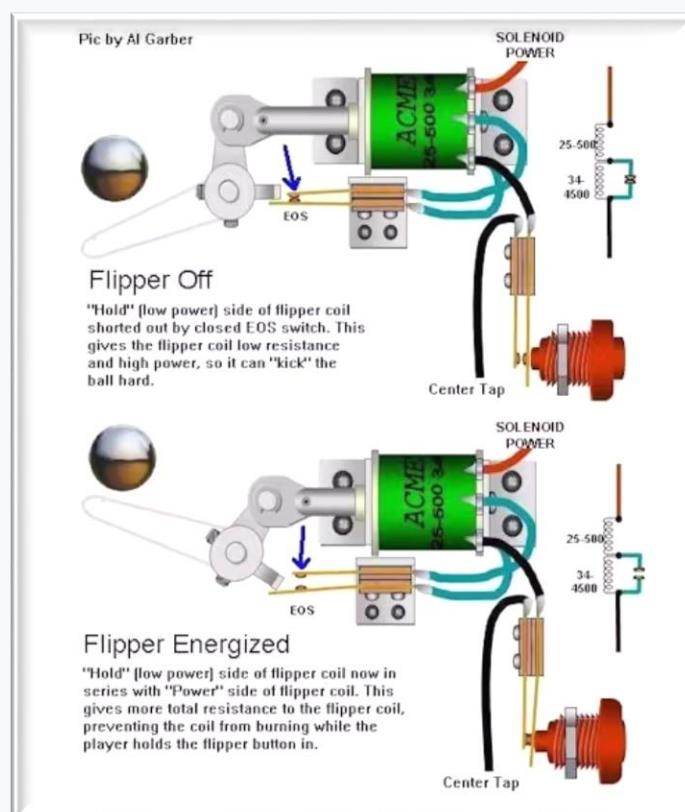


## زمین بازی

زمین بازی از تخته سه لا با روکش توس  $\frac{1}{4}$  اینچی ساخته شده است  
ابتدا تخته سه لا را با چسب تماسی اسپری کرده و سپس پوستر را روی  
آن لمینت شده سپس آن را برگردانده و به اندازه آن کوتاه شده

## فلایپر

وقتی اکثر اجزای پین بال را می خرید معمولاً دستورالعمل های مفیدی  
ندارند نحوه سیم کشی باله ها در تصویر زیر نشان داده شده است



## فلیپر خاموش

طرف سیم پیچ فلیپر را نگهدارید (با قدرت کم) با کلید EOS بسته کوتاه شده است این به سیم پیچ فلیپر مقاومت کم و قدرت بالایی میدهد میتواند توپ را محکم لگد بزند

## فلیپر با انرژی

سمت "HOLD" (کم برق) سیم پیچ فلیپر با سمت "POWER" سیم پیچ فلیپر ترکیب شده است این مقاومت کلی به سیم پیچ فلیپر میدهد و از سوختن سیم پیچ در حالی که بازیکن دکمه فلیپر را نگه میدارد و جلو گیری میکند

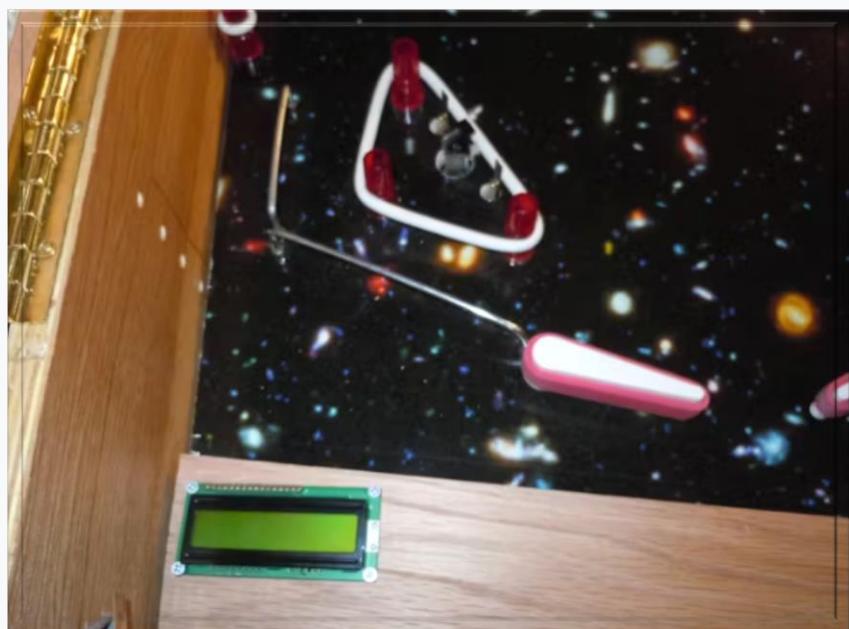
علاوه بر مکانیزم باله به یک چوب باله و سوئیچ باله و یک دکمه نیاز خواهید داشت

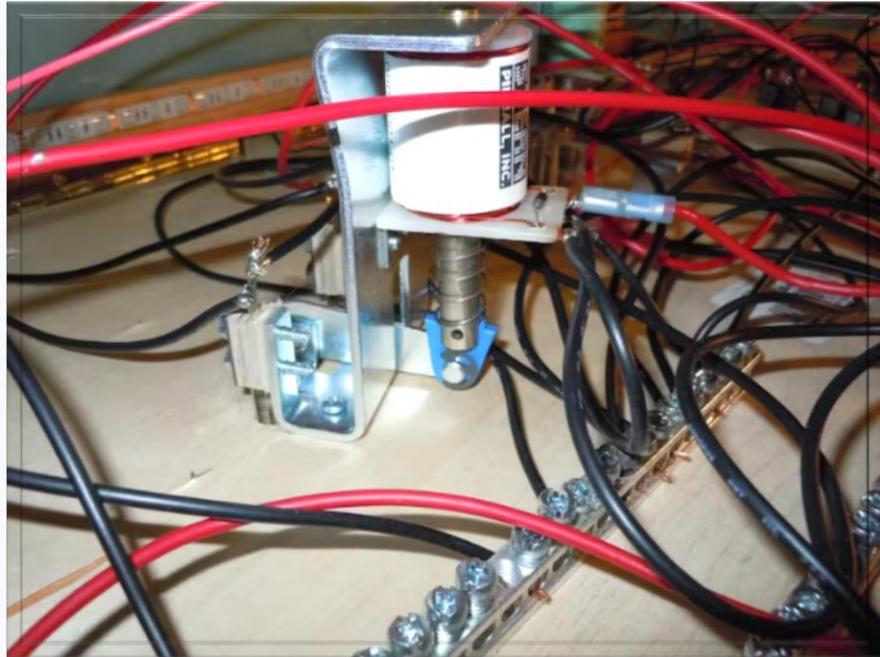
## تیرکمان

یک مجموعه کامل تیرکمان بچه گانه را می توانید در اینجا پیدا کنید همچنین به پین ها و نوارهای لاستیکی نیاز خواهید داشت می توانید

آنها را در همان مکان پیدا کنید همچنین تصویری از تیرکمان در هنگام نصب در زیر نشان داده شده است. لازم به ذکر است که برای تمامی قطعات وصل شده به منبع تغذیه ۲۴ ولت از سیم پیج ۱۶ و برای اتصالات برد آردوینو از سیم پیج ۲۲ استفاده شده است

تصویر اول نیز چند ریل فلزی در کنار خفاش های باله ای را نشان می دهد زمانی که چمن ها با حشره کش پاشیده می شوند از آن برای علامت گذاری استفاده می شد سیم را ببرید و خم کرده در زمین بازی سوراخ کرده و ریل ها را در جای خود اپوکسی شد





## ضربه گیرهای پاپ

به سوئیچ و کلاهک نیاز خواهید داشت ضربه گیرهای پاپ با سوراخ کردن سه سوراخ، دو سوراخ برای میله های نگهدارنده فلزی و یکی برای محوری که سوئیچ را فعال می کند نصب می شوند. هنگامی که یک توپ به دیسک سفیدی که در نمای بالا دیده می شود می غلتد شیفت را به سمت پایین هل می دهد سپس یک حلقه فلزی را به سمت پایین می کشد که توپ را در جهت مخالف می فرستد.

## مونتاژ ضربه گیر پاپ



مونتاژ ضربه گیر پاپ مطابق تصویر برای برخی از بازی های پین  
بال الکترومکانیکی

شامل قسمت های زیر است:

دامن سپر پاپ (سفید)

لینک یوغ فلزی

حلقه/میله سپر پاپ

بدنه سپر پاپ

فر دامن سپر پاپ

پایه سپر پاپ ه

سوکت لامپ با سرب

براکت سیم پیچ و مونتاژ توقف

براکت توقف

پیستون کویل

فرن پلانجر کویل

پیوند فیر یوغ

واشر تخت

پیچ برآکت

مهره نایلوک

کویل

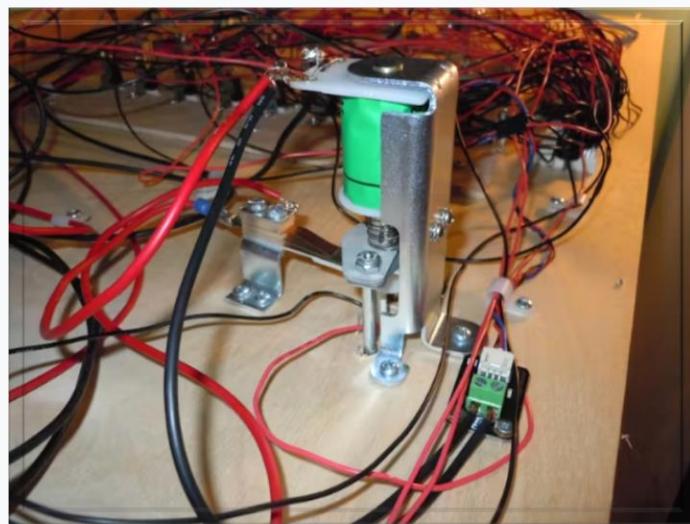
## مونتاژ پشته سوئیچ سپر پاپ



مونتاژ کامل سوئیچ با برآکت و محرک برای استفاده در ضربه گیرهای پاپ در بسیاری از بازی های حالت جامد که از یک مجموعه نقطه تماس استفاده می کنند برآکت نصب در سمت راست نصب می شود

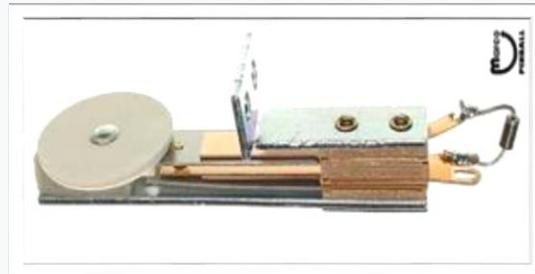
پین بال دزد:

## قطعات سپر پاپ (پاپ / جت / قطعه ضربه گیر)



اهداف، کلیدهای واژگون و چراغ ها

سوئیچ هدف - پایه جلویی سفید گرد



هدف ثابت، برآکت نصب جلو، گرد سفید

سوئیچ ها و پایه های چرخشی :

سوئیچ و مونتاژ برآکت



## لامپ ها

### LED-#555 4-Element



تنوع رنگ

LED 4 عنصری (روشنایی بالا)

استاندارد و غیر ارواح

6 ولت DC یا AC

قطر 7/16 اینچ (11.1 میلی متر)

بهترین برای درج لامپ میدان بازی و روشنایی روشن و مرکز  
رنگها:

سفید سرد

سفید گرم

قرمز

سبز

آبی

رنگ زرد

نارنجی

رنگ بنفش

## ghosting چیست؟

Ghosting علامتی است که برخی از LED ها ممکن است گهگاه هنگام استفاده در مکان های خاص در برخی بازی های الکترونیکی پین بال نشان دهند این روشنایی جزئی یک لامپ LED است زمانی که قرار است کاملاً خاموش باشد به عنوان مثال اگر LED پاداش 1000 بازی روشن شود LED دیگری در جای دیگری در زمین بازی که قرار است خاموش باشد ممکن است همزمان کم نور شود. این به دلیل ویژگی های الکترونیکی برخی از انواع لامپ های LED و سیم کشی/سوئیچینگ منحصر به فرد به سبک ماتریسی است که در برخی بازی های پین بال دارد استفاده می شود این یک علامت صرفاً زیبایی است و آسیبی به بازی شما وارد نمی کند

گاهی قابل توجه است و گاهی اوقات نه - همه اینها به نوع LED، رنگ، دید و مکان قرارگیری آن در بازی بستگی دارد.

اگر LED دارید که شبحدار است و ترجیح می‌دهید جلوه را از بین ببرید می‌توانید به جای LED مشکل از نوع LED غیر شبحدار استفاده کنید

پایه های لامپ  
سوکت لامپ - پایه گوه - پایه عمودی



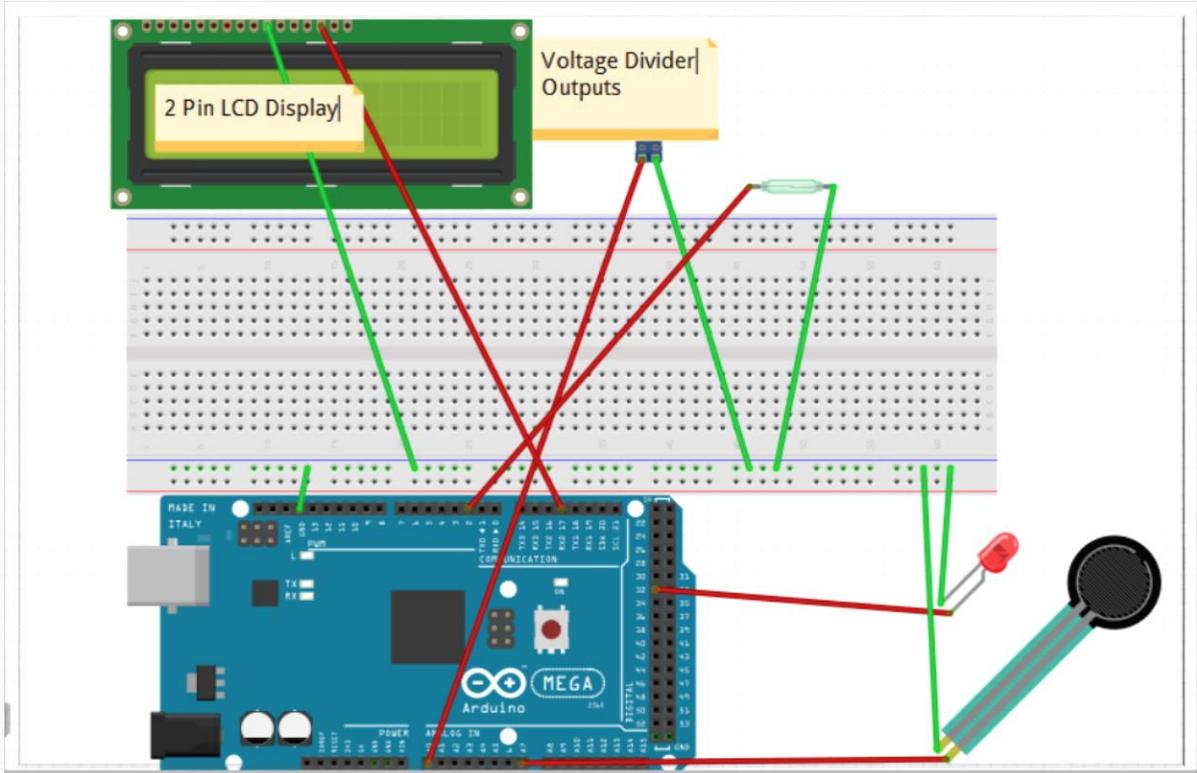
سیم کشی همه اینها اساساً یکسان است و در قسمت آردوینو ذکر شد

## شماتیک ها

شماتیک اجزای اصلی دستگاه پین بال دزد آردوینو

دستگاه پین بال آردوینو کد آردوینو

به آردوینو اجازه می دهد تا بازی یک ماشین پین بال خانگی  
را کنترل کند



# Code

```
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;

}

}
```



