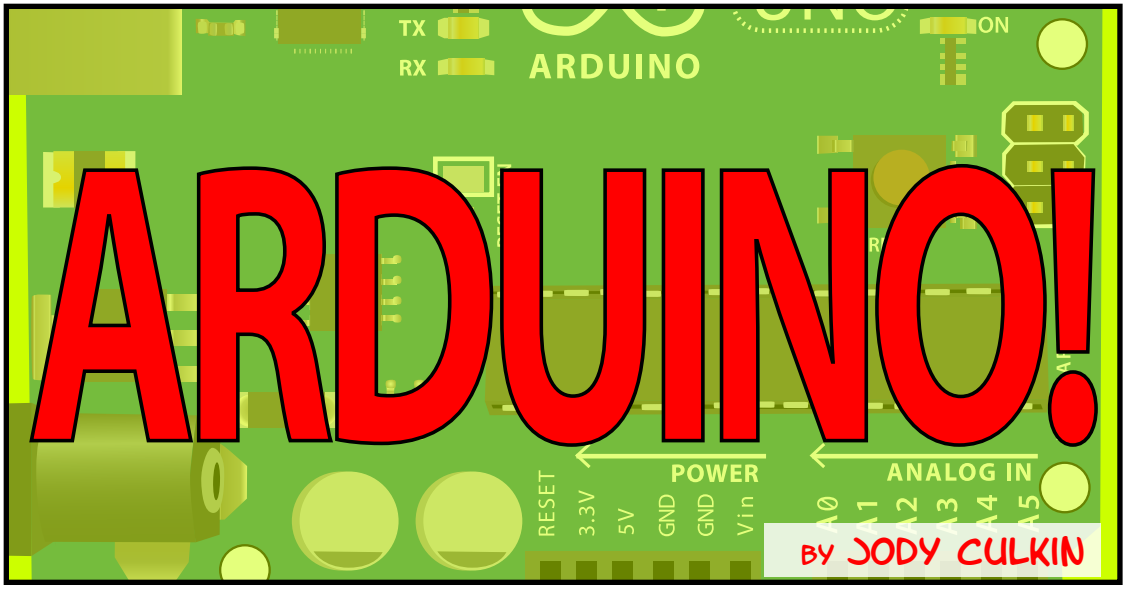


متن باز

منابعی که میتوانند به صورت رایگان استفاده ، بازنویسی و توزیع مجدد شوند را متن باز میگویند . معمولا برای سخت افزار یا نرم افزار به کار میرود .

الکترونیک

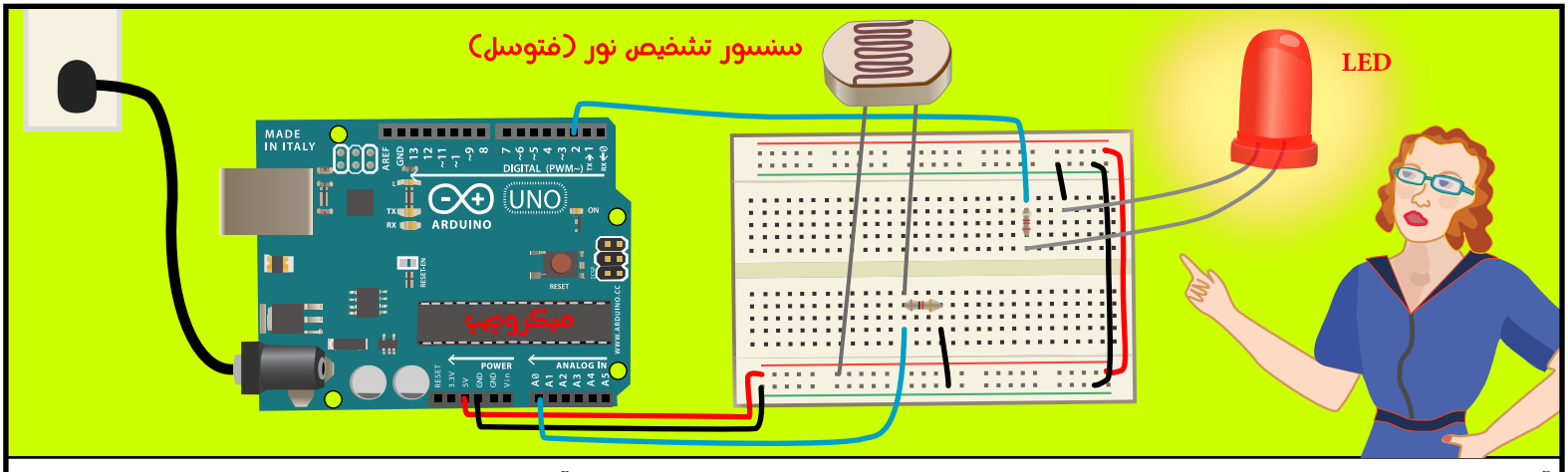
فناوری یا علمی که از حرکت کنترل شده الکترون ها در رسانه های مختلف استفاده میکند .

نمونه سازی

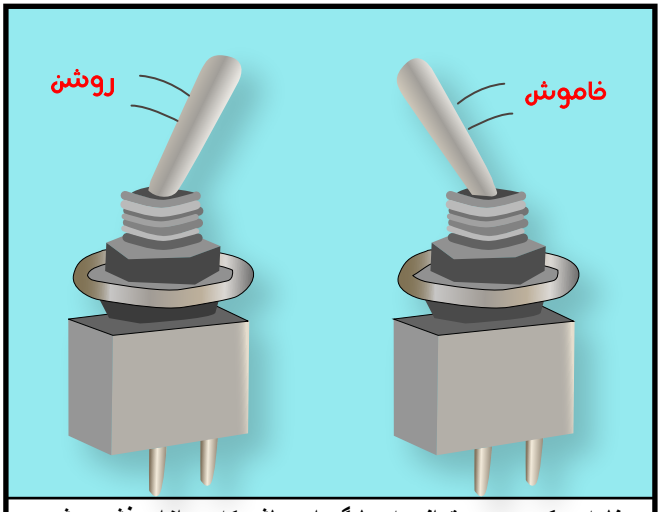
یک شکل مرجع و اولیه که میتواند به عنوان استاندارد و پایه برای چیز های دیگر استفاده شود را نمونه اولیه میگویند که از روی آن ، نمونه سازی انجام میشود .

پلتفرم

یک معماری سخت افزاری با نرم افزار (framework) روی آن که اجازه میدهد تا برنامه های دیگر نیز روی آن اجرا شوند .



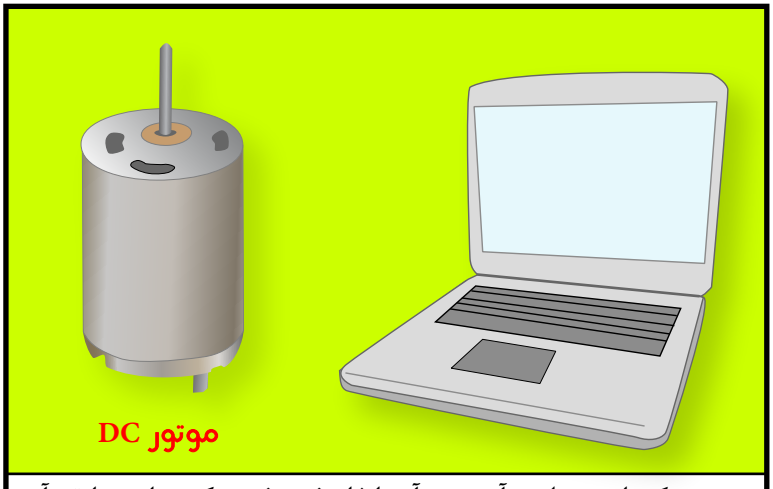
آردوینو شامل یک میکروچیپ است . میکروچیپ یا میکروکنترلر یک کامپیوتر بسیار کوچک است که میتوانیم آن را برنامه ریزی کنیم . همچنین میتوانیم سنسور های مختلف به آن وصل کنیم تا محرک هایی را اندازه گیری کنیم (مثلا سنسور نور که میزان نور محیط را اندازه گیری میکند) . به وسیله ی برنامه ریزی کردن میکروچیپ میتوانیم تعیین کنیم که چه واکنشی به این محرک ها و خروجی سنسور ها داده شود (مثلا یک موتور یاLED روشن شود)



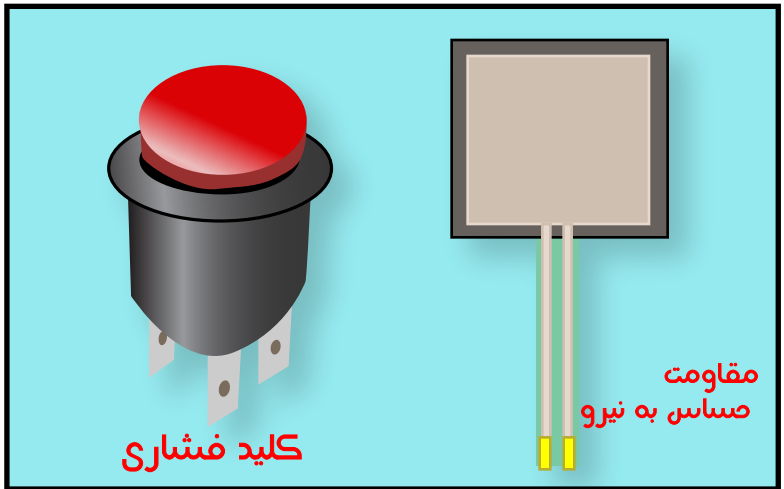
برنامه میکروچیپ میتواند به سادگی این باشد که در ازای فشرده شدن یک کلید کاری را انجام دهد

موس ، نمونه ای از یک دستگاه ورودی برای کامپیوتر است . در مقابل صفحه نمایش نمونه ای از یک دستگاه خروجی در کامپیوتر های خانگی است .

میکروکنترلر ها نیز مثل کامپیوتر ها ، ورودی هایی را میگیرند و خروجی هایی تولید میکنند . در ورودی ، داده هایی از کاربر و محیط دریافت میشود سپس روی این داده ها پردازش ها و محاسباتی انجام شده و سرانجام به عنوان خروجی نتیجه محاسبات و پردازش ها ارائه میشود .



هر چیزی که ما به وسیله ی آردوینو ، آن را خاموش روشن میکنیم یا به عبارتی آن را کنترل میکنیم ، یک خروجی است . میتواند یک موتور یا حتی یک کامپیوتر دیگر باشد .



یک ورودی برای آردوینو ، میتواند یک سنسور یا حتی یک کلید باشد . هر دو اطلاعاتی را در قالب ورودی به‌آردوینو ارسال میکنند . سنسور ممکن است میزان دمای محیط را و کلید هم حالت فشرده شدن یا نشدن خود را به‌آردوینو ارسال کند

یک کلید (سوییچ) یک ورودی دیجیتال است . زیرا فقط دو حالت قطع یا وصل بودن را دارد . یک سنسور میتواند ورودی دیجیتال محسوب شود . وقتی اطلاعات سنسور های آنالوگ را به دیجیتال تبدیل کنیم ، محدوده مقادیر آنان محدود میشود .

ورودی ها و خروجی ها میتوانند به دوصورت آنالوگ یا دیجیتال باشند . اطلاعات دیجیتال ، دودوئی یا به عبارتی در مبنای ۲ (صفر و یک) هستند . آن ها فقط دو حالت دارند : ۱ – درست ۲ – غلط . در مقابل اطلاعات آنالوگ شامل یک محدوده پیوسته از حالت ها هستند برخلاف دیجیتال ها که فقط دو حالت درست یا غلط بیشتر نداشتند .

مقاومت (R)

مقاومت (R) برای یک قطعه ، بیان کننده میزان مقاومت جسم در مقابل عبور جریان الکتریکی (الکترون ها) از آن است . در واحد اهم اندازه گیری میشود .

جریان (I)

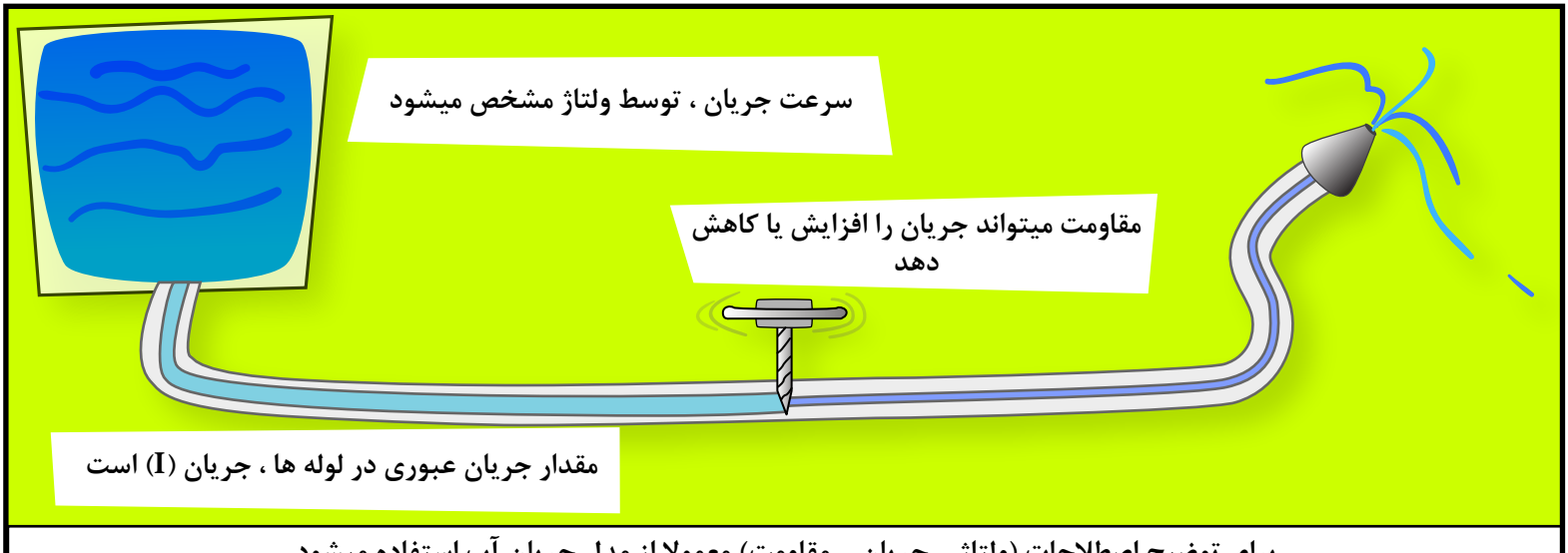
جریان (I) مقدار بار عبوری توسط الکترون ها در واحد زمان ، داخل یک رسانا است . در واحد آمپر (A) اندازه گیری میشود .

ولتاژ (V)

اختلاف پتانسیل (V) یا ولتاژ ، بیان کننده پتانسیل الکتریکی برای انتقال الکترون ها در مدار است . در واحد ولت (v) اندازه گیری میشود .

الکتریسیته ، عبور جریان الکترون (انرژی الکتریکی)داخل یک رسانا است .

قبل از اینکه وارد کار با آردوینو بشویم ، میبایست چند اصطلاح و اصل مربوط به الکتریسیته را دوره کنیم . دقت کنید اختلاف پتانیسل همان ولتاژ است . بنابراین در متن ، هر دو این واژه ها اشاره به یک چیز یکسان دارند .



برای توضیح اصطلاحات (ولتاژ - جریان - مقاومت) معمولا از مدل جریان آب استفاده میشود

قانون اُهم

مقاومت (R) ÷ ولتاژ (V) = جریان (I)

$(I = V / R)$

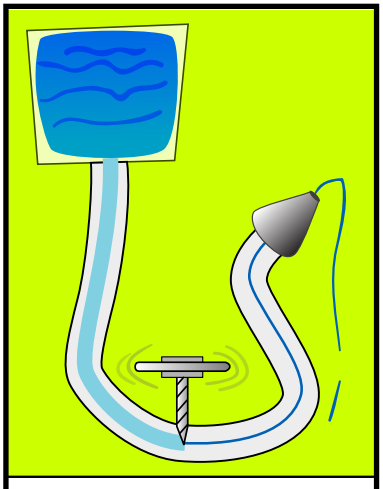
جریان (I) ÷ ولتاژ (V) = مقاومت (R)

$(R = V / I)$

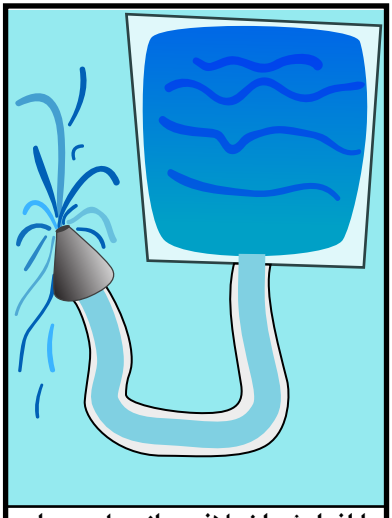
جریان (I) × مقاومت (R) = ولتاژ (V)

$(V = I * R)$

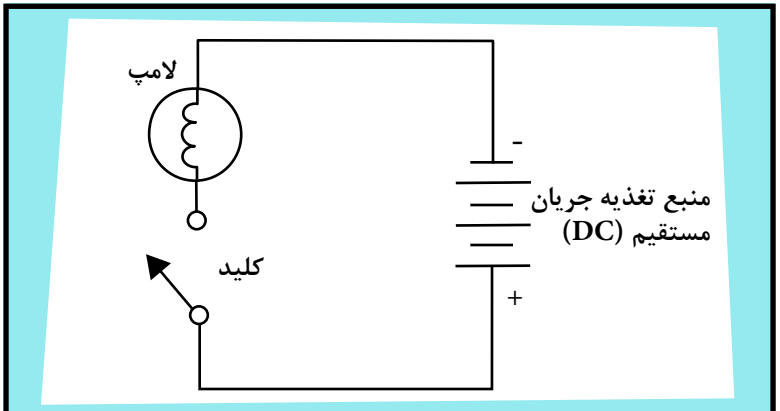
روابطی بین ولتاژ ، جریان و مقاومت وجود دارد که توسط جرج اهم فیزیکدان آلمانی کشف شده است .



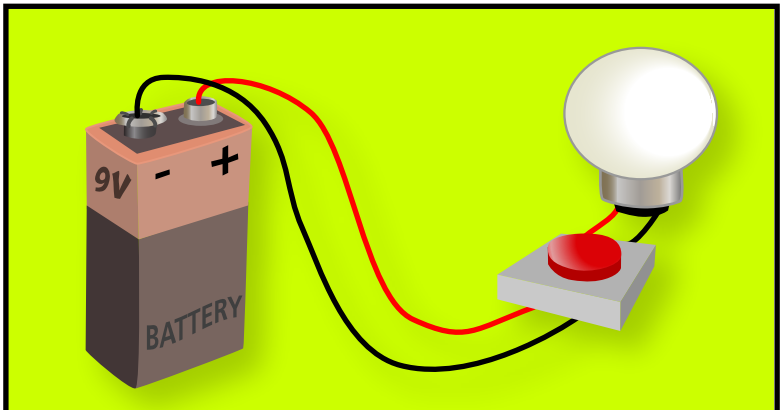
در مثال این روابط ، افزایش مقاومت جریان را کاهش میدهد .



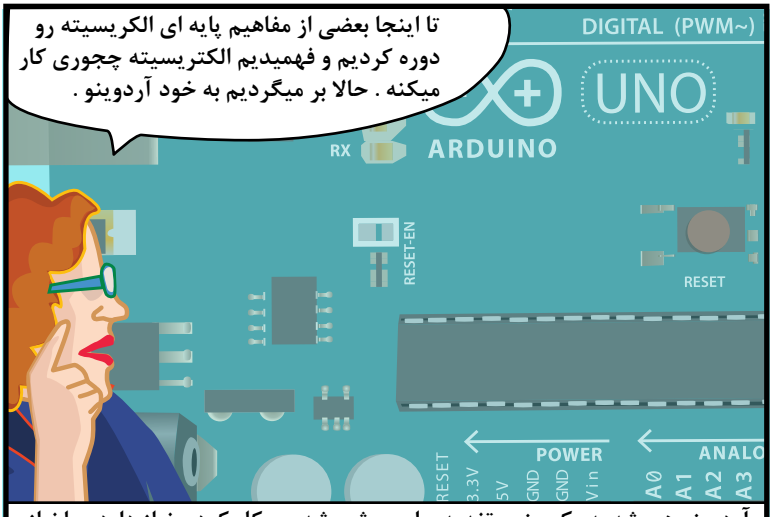
یا افزایش اختلاف پتانسیل جریان را افزایش میدهد .



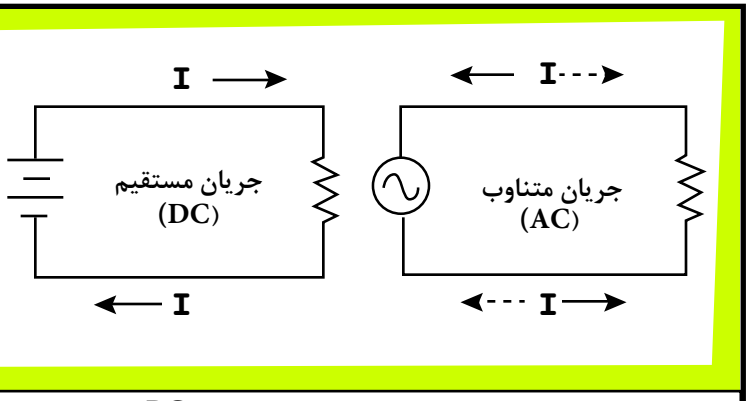
تصویری که میبینید شماتیک مدار قبل است .در مدل شماتیک هر قطعه الکترونیکی یک نماد خاص خود را دارد و در این مدل ، مدار مورد نظر توسط این نماد ها به نمایش در می آید . وقتی کلید بسته است (فشرده شده است) جریان از منبع تغذیه که همان باتری است در مدار حرکت میکند و به لامپ میرسد و باعث روشن شدن آن میشود .



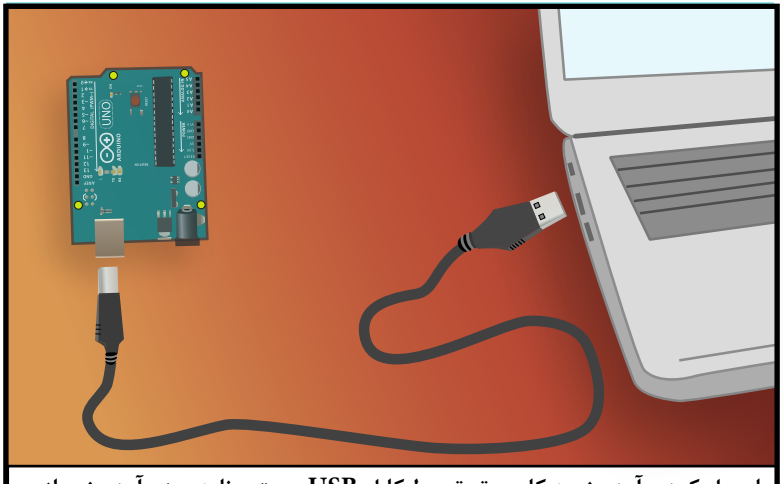
بیایید به بررسی یک مدار ساده بپردازیم . هر مدار یک حلقه بسته برای عبور جریان است که دارای یک یا چند منبع انرژی (باتری) و یک یا چند مصرف کننده (لامپ در این مثال) است . مصرف کننده انرژی الکتریکی دریافت شده از باتری را مصرف میکند و آن را به گونه های دیگر انرژی (نوری ، مکانیکی و ...) تبدیل میکند . در مدار این تصویر یک کلید هم وجود دارد .



آردوینو همیشه به یک منبع تغذیه برای روشن شدن و کار کردن نیاز دارد . ما نیاز خواهیم داشت تا آن را به کامپیوتر وصل کنیم تا برنامه ریزی اش کنیم .



خوب است بدانیم دو نوع جریان وجود دارد . جریان مستقیم (DC) و جریان متناوب (AC) . در مدار ، جریان مستقیم فقط در یک جهت حرکت میکند . در مقابل جریان متناوب دائما در یک جهت و جهت مخالف آن تغییر جهت میدهد . یعنی در چرخه های زمانی معین جهت حرکت خود را به جهت مخالف حرکت فعلی تغییر میدهد . در اینجا فقط با جریان مستقیم (DC) سر و کار داریم .



با وصل کردن آردوینو به کامپیوتر توسط کابل USB جهت برنامه ریزی آردوینو ، از طریق خود کامپیوتر ولتاژ مورد نیاز کارکرد آردوینو تامین میشود و در اینجا کامپیوتر نقش همان منبع تغذیه را ایفا میکند .

برای برنامه ریزی کردن آردوینو ، باید نرم افزار مورد نیاز برای اینکار را روی کامپیوترمان نصب کنیم . این برنامه که نرم افزارآردوینو است ، از لینک بالا به صورت رایگان قابل دانلود است . برنامه‌آردوینو روی Windows , Linux , Mac OS X قابل نصب است .

آموزش نصب آردوینو رویMac

http://www.arduino.cc/en/Guide/MacOSX

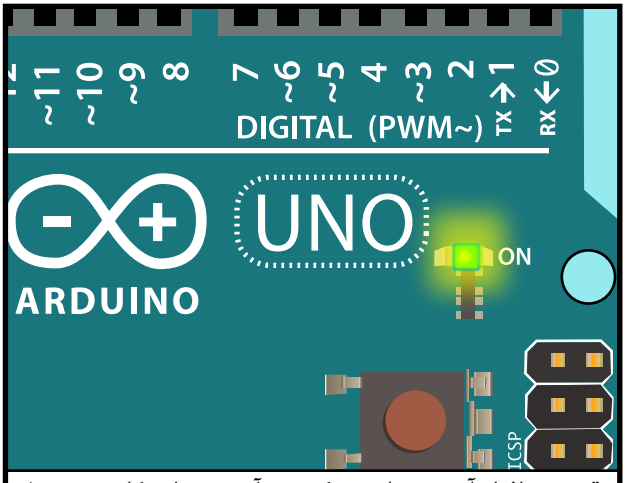
آموزش نصب روی Windows

http://www.arduino.cc/en/Guide/Windows

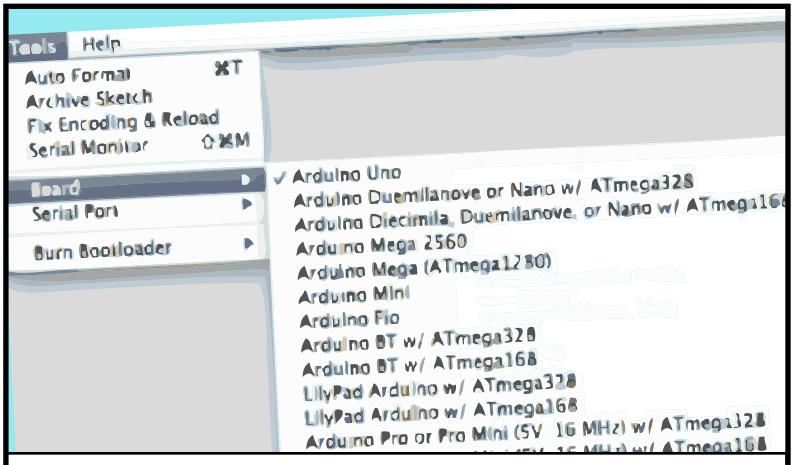
آموزش نصب روی Linux

http://www.arduino.cc/playground/Learning/Linux

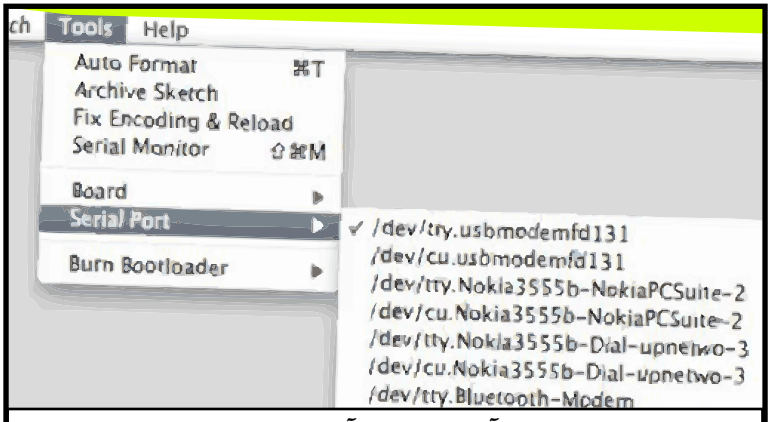
جهت دیدن آموزش نصب برنامه آردوینو با جزئیات بیشتر با توجه به سیستم عامل خود ، به لینک های بالا مراجعه کنید .



وقتی نرم افزار آردوینو را نصب کردیم ،آردوینو را به کامپیوتر وصل میکنیم .روی برد یک لامپ LED به محض اتصال روشن میشود که نشان دهنده روشن شدن آردوینو است .

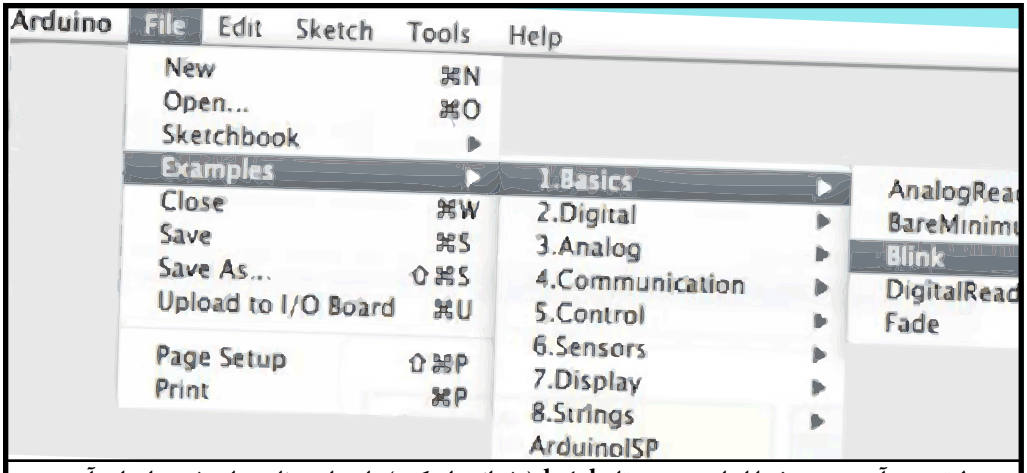


برنامه آردوینو را اجرا میکنیم و در منوی Tools قسمت Board، مدل برد آردوینو خود را انتخاب میکنیم . برای مثال Arduino Uno

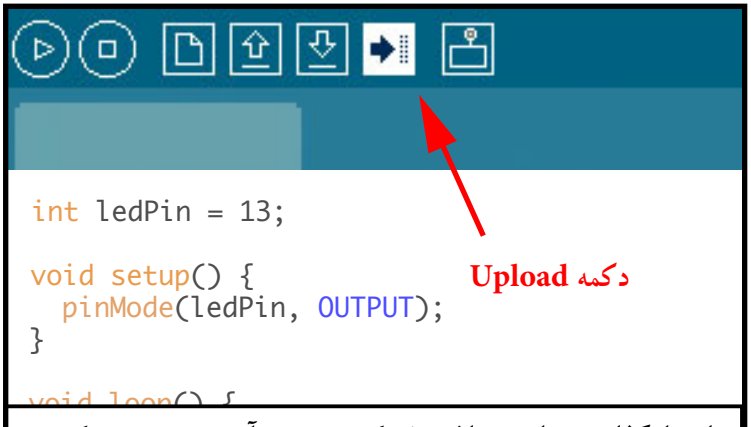


سپس پورت سریال که آردوینو روی آن به کامپیوتر وصل شده است را انتخاب میکنیم . برای اینکار به منوی Tools رفته و از قسمت Serial Port پورت مورد نظر خود را انتخاب میکنیم .پورت سریال داخل Windows چیزی شبیه COM۳ است .

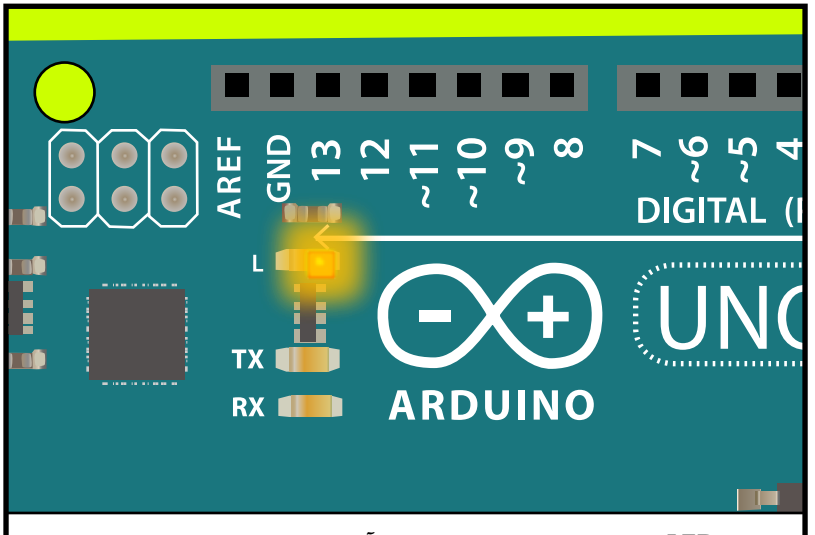
نرم افزار آردوینو که در مراحل قبل دانلود کردیم ، یک IDE یا محیط توسعه یکپارچه است.IDE آردوینو شامل یک ویرایش گر متن ، همراه با مترجم(Compiler) و قابلیت های دیگر است که به برنامه نویس کمک میکند تا برنامه های خودش را توسعه بدهد .



محیط توسعه آردوینو به شما اجازه میدهد تا sketch(بخوانید اِسکِچ) یا همان برنامه های خود را برای آردوینو بنویسید و آن را روی برد آردوینو خود بارگذاری (upload) کنید . مثال Blink که برنامه چراغ چشمک زن است را از منوی File به در مسیر (File > Examples > ۱.Basics > Blink) انتخاب کنید .



برای بارگذاری برنامه چراغ چشمک زن روی آردوینو ، روی دکمه Upload در قسمت بالای پنجره برنامه آردوینو کلیک کنید . پیغام هایی در پنجره پایینی نمایش داده میشوند و سپس بارگذاری با موفقیت انجام میشود .



چراغ LEDمتصل به پایه شماره ۱۳ روی برد آردوینو شروع به چشمک زدن میکند .

```
void setup() {                          پایه ۱۳ در اکثر آردوینو ها به LED داخلی روی برد متصل است.
  pinMode(13, OUTPUT);                  تنظیم پایه ۱۳ به عنوان خروجی دیجیتال
}
void loop() {
  digitalWrite(13, HIGH);               روشن کردن پایه ۱۳
  delay(1000);                          تاخیر به مدت ۱۰۰۰ میلی ثانیه
  digitalWrite(13, LOW);                خاموش کردن پایه ۱۳
  delay(1000);                          تاخیر به مدت ۱۰۰۰ میلی ثانیه

}
```

یک برنامه نوشته شده برای آردوینو ، مانند برنامه های نوشته شده در زبان های دیگر از مجموعه ای از دستورات برای کامپیوتر تشکیل شده است . اگر با دقت به برنامه چراغ چشمک زن نگاه کنیم ، میبینیم که دو بخش اصلی وجود دارد . بخش setupو loop

setup : دستورات داخل این بخش فقط یک بار به محض روشن شدن آردوینو اجرا میشوند.

loop : بعد از اجرای دستورات setup در ابتدای کار ، دستورات این بخش همیشه به صورت تکرار شونده اجرا خواهند شد .

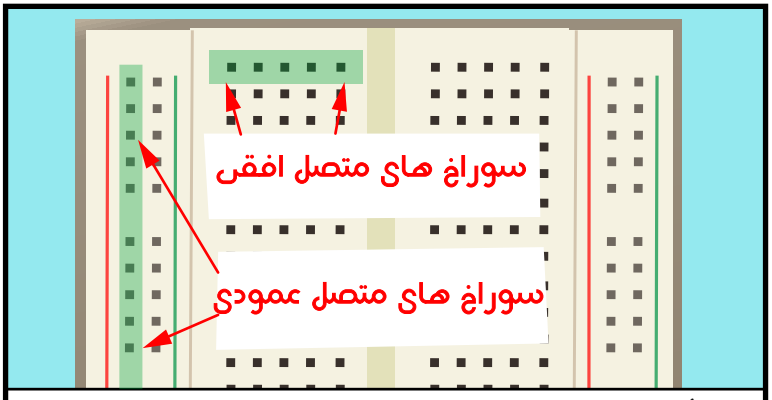
این دو بخش loop , setup هر دو مجموعه ای از دستورات هستند که به هر کدام از آن ها یک تابع میگویند . هر برنامه آردوینو حتما این دو تابع را دارد . آن ها بلوک هایی هستند که با آکولاد {} ابتدا و انتهای آن ها مشخص میشود.

http://arduino.cc/en/Reference/HomePage

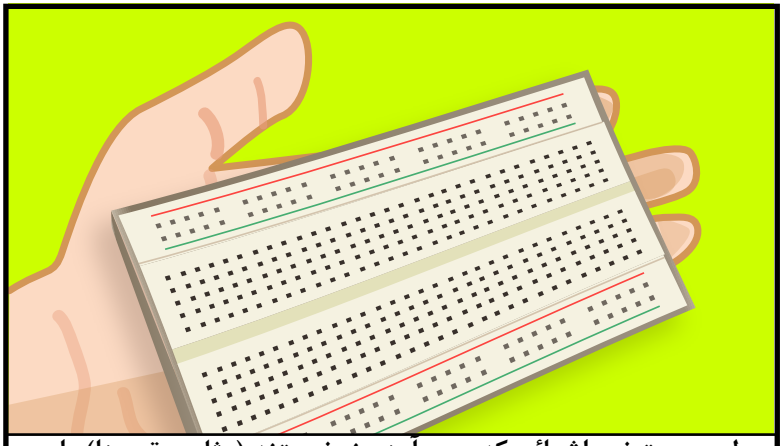
برای دیدن منابع آموزشی آردوینو و بسیاری منابع دیگر برای یادگیری زبان برنامه نویسیآردوینو به وبسایت آردوینو مراجعه کنید .

```
void setup() {                          تعریف بلوک setup
  pinMode(13, OUTPUT);                  تنظیم پایه ۱۳ به عنوان خروجی دیجیتال
}                                       اتمام بلوک setup


void loop() {                           شروع بلوک loop
  digitalWrite(13, HIGH);               روشن کردن پایه ۱۳
  delay(1000);                          تاخیر به مدت ۱۰۰۰ میلی ثانیه
  digitalWrite(13, LOW);                خاموش کردن پایه ۱۳
  delay(1000);                          تاخیر به مدت ۱۰۰۰ میلی ثانیه
}
```

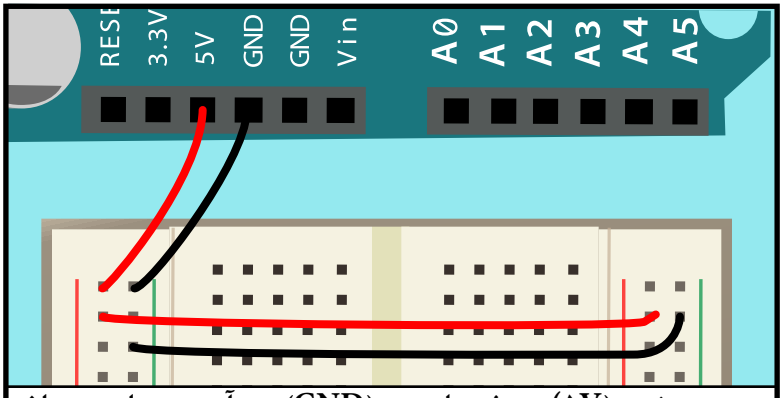
بیایید این اسکریپت ساده را خط به خط بررسی کنیم که هر خط چه کاری انجام میدهد (به دستورات نوشته شده برای آردوینو اسکریپت میگویند)



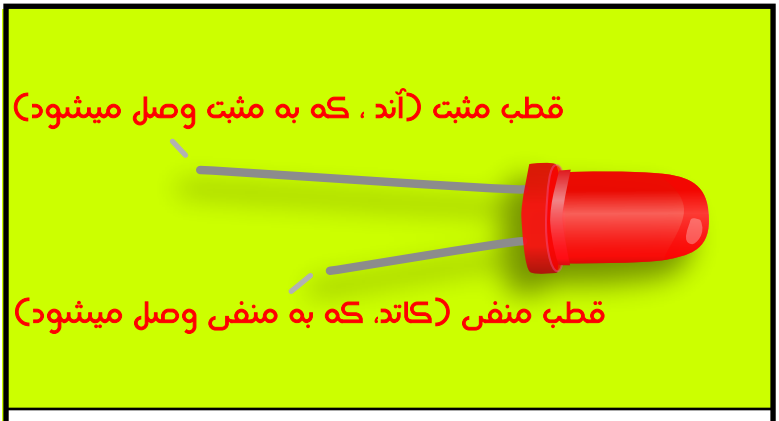
این بِردبُرد دو ردیف سوراخ از بالا تا پایین ، یکی در سمت چپ و یکی در سمت راستَ دارد . همچنین ۵ ردیف سوراخ در دو طرف تو رفتگی وسط دارد . طبق تصویر ، دو ردیف سوراخ های چپ و راست بِردبُرد به صورت عمودی بهم متصل اند (با رنگ سبز اتصالشان مشخص شده) . و ۵ رَدیف سوراخ های دو طرف تو رفتگی وسط ، به صورت افقی بهم متصل اند .



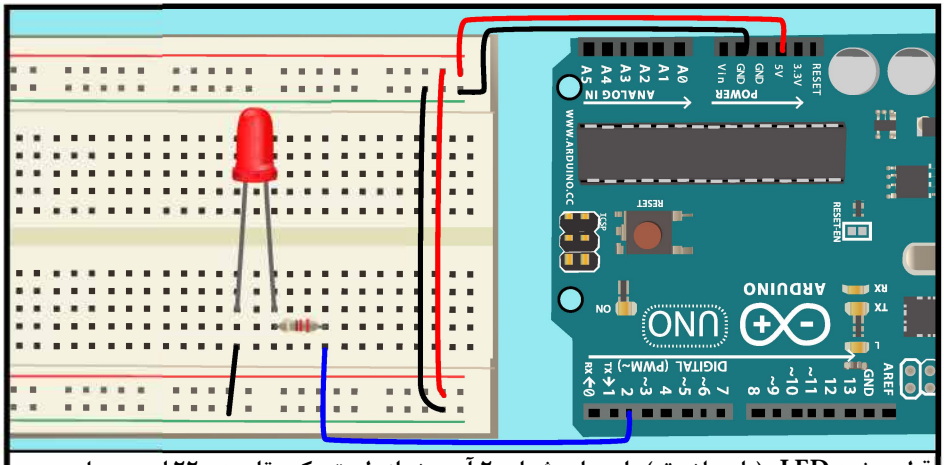
چطوری میتونیم اشیائی که روی آردوینو نیستند (مثل موتور ها) را کنترل کنیم ؟ برای اینکار آردوینو را به یک بِردبُرد وصل میکنیم . این کار به ما اجازه میدهد تا به سرعت مدار های مَختلف را ببندیم و تست کنیم .



دو سیم مثبت (۵V) و منفی یا زمین (GND) برد آردوینو را به سوراخ های عمودی بردبرد وصل میکنیم (طبق تصویر) . حالا هر قطعه ای که خواستیم به آردوینو وصل کنیم ، در سوراخ های وسط بردبرد سوار میشود و تغذیه آن از سوراخ های عمودی کنار بردبرد گرفته میشود که به مثبت و زمین آردوینو متصل است .



وقتی جریان ، در جهت صحیح از یک LED (دیود ساتع کننده نور) عبور میکند ، باعث روشن و نورانی شدن آن میشود . یک LED روی بردبرد سوار میکنیم و آن را به آردوینو وصل میکنیم تا بتوانیم از طریق کد و برنامه ریزی کردن آردوینو آن را کنترل کنیم .

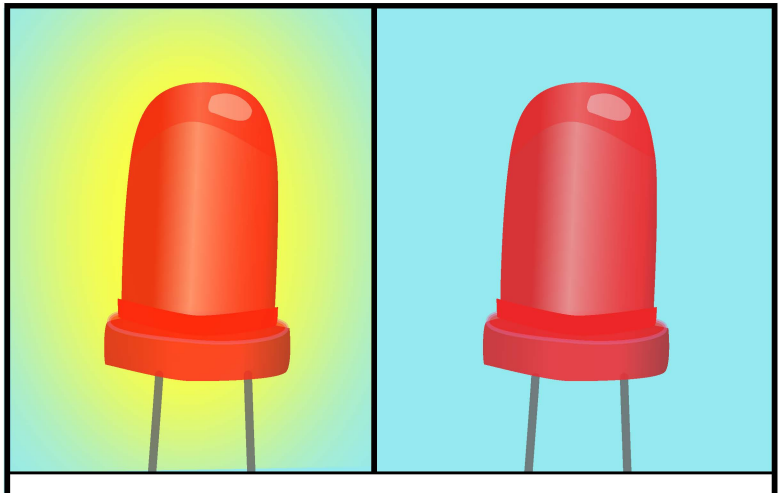


قطب مثبت LED (پایه بلند تر) را به پایه شماره ۲ آردوینو از طریق یک مقاومت ۲۲۰ اهم متصل میکنیم . قطب منفی LED (پایه کوچکتر) را نیز به زمین (GND) روی سوراخ های عمودی بردبرد وصل میکنیم . پایه های ۲ تا ۱۳ روی آردوینو میتوانند به عنوان خروجی یا ورودی تنظیم شوند . روی دکمه new در نرم افزار آردوینو کلیک کنید تا برنامه جدید را بنویسیم .

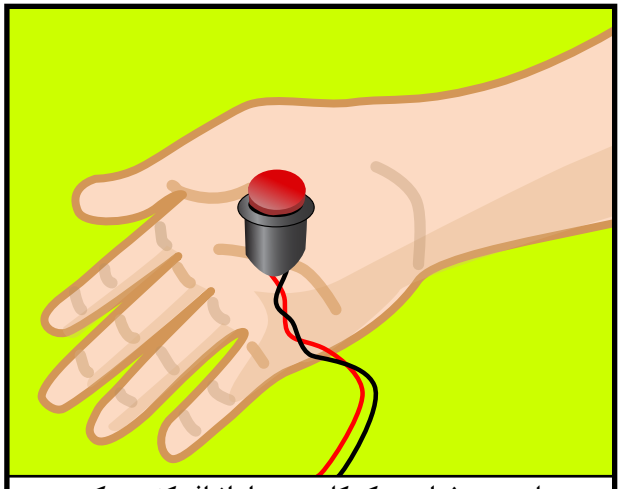
```
void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  delay(500);
}
```

در setup باید پایه ۲ (که LED به آن متصل است)را به عنوان خروجی تنظیم کنیم . در loop ابتدا پایه ۲ را روشن میکنیم که باعث روشن شدن LED میشود . سپس ۵۰۰ میلی ثانیه یا به عبارتی نیم ثانیه صبر میکنیم . سپس پایه ۲ را خاموش میکنیم که باعث خاموش شدن LED میشود . باز نیم ثانیه صبر میکنیم .

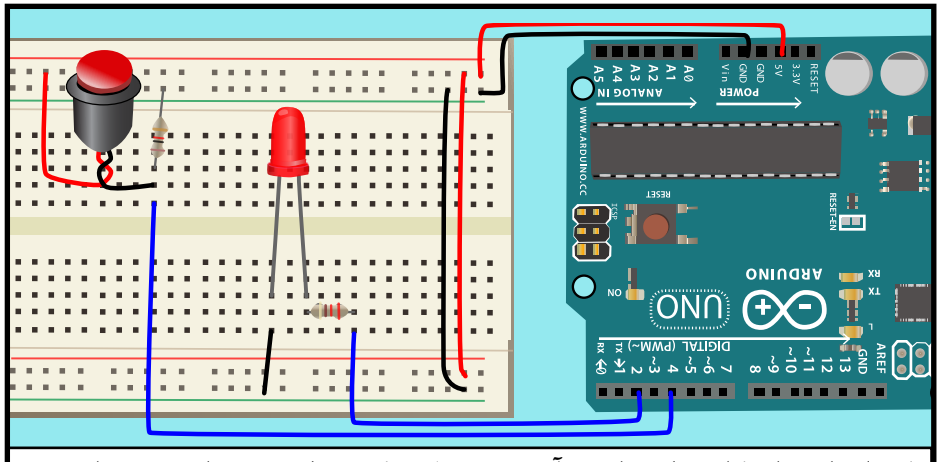
روی دکمه verify کلیک میکنیم تا کد نوشته شده مان بررسی شود تا اگر خطایی دارد آن را رفع کنیم . سپس روی upload کلیک میکنیم تا برنامه نوشته شده را روی برد آردوینو بارگذاری کنیم .



چراغ LED هر نیم ثانیه (۵۰۰ میلی ثانیه) یک بار چشمک میزند . به عبارتی نیم ثانیه روشن است و سپس نیم ثانیه خاموش میشود و این روند تکرار میشود



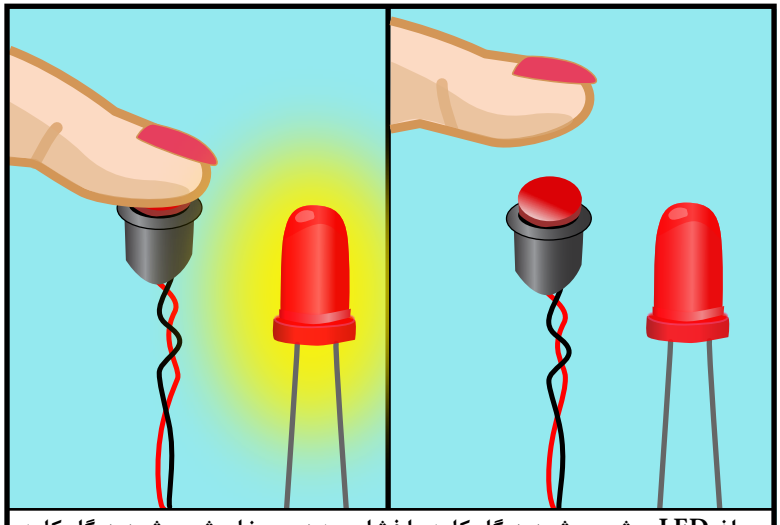
در مرحله بعد میخواهیم یک کلید به مدار اضافه کنیم . یک ورودی دیجیتال زیرا کلید فقط دو حالت باز و بسته دارد پس دیجیتال است . با این کلید میخواهیم LED را خاموش روشن کنیم .



یکی از پایه های کلید را به پایه ۴ آردوینو وصل میکنیم . از همین پایه به وسیله ی یک مقاومت ۱۰ کیلو اهم به زمین (GND) وصل میکنیم .سپس پایه دیگر کلید را به مثبت وصل میکنیمLED . نیز به همان شکلی که قبلا بود باشد .

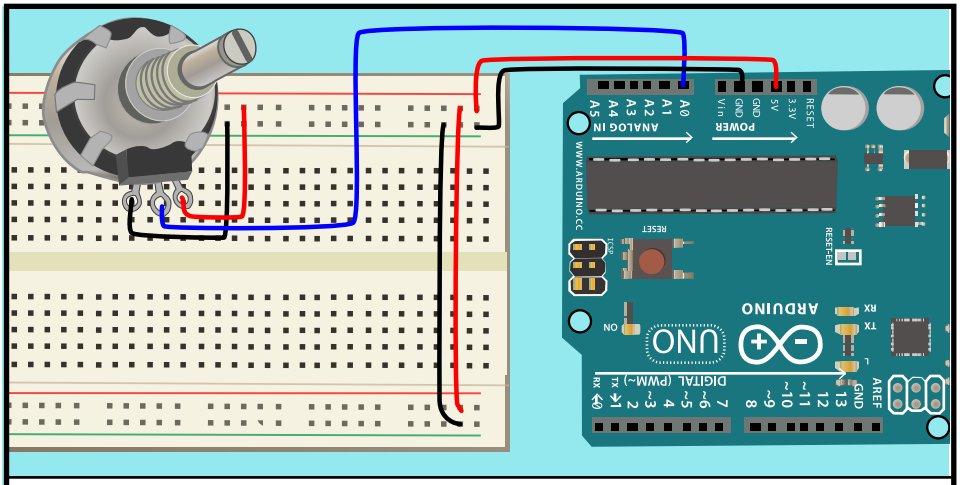
```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
}

void loop() {
  if(digitalRead(4)){
    digitalWrite(2, HIGH);
  }else{
  digitalWrite(2, LOW);
  }
}
```

در **setup** پایه ۲ را به عنوان خروجی (برای LED) و پایه ۴ را به عنوان ورودی (برای کلید) تنظیم میکنیم . در **loop** از ساختار شرطی **if** استفاده میکنیم . اگر سطح پایه ۴ برابر با ۱ یا HIGH شد (کلید فشرده شده) پایه مربوط به LED را نیز روی سطح ۱ (روشن) قرار میدهیم . در غیر این صورت پایه LED را روی سطح ۰ (خاموش) قرار میدهیم (نکته HIGH معادل ۱ و LOW معادل ۰ است).



چراغ LED روشن میشود هرگاه کلید را فشار میدهیم و خاموش میشود هرگاه کلید را رها میکنیم

حالا با توجه به اینکه از پتانسیومتر استفاده میکنیم یک ورودی آنالوگ خواهیم داشت .

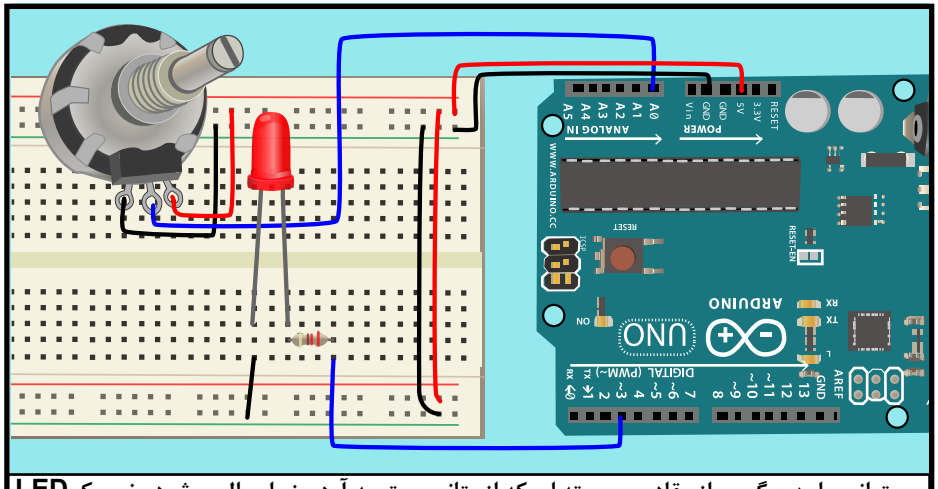


پایه وسط پتانسیومتر را به پایه آنالوگ **A0** روی آردوینو متصل میکنیم . سپس از دو پایه باقی مانده روی پتانسیومتر یکی را به زمین و دیگری را به تغذیه مثبت وصل میکنیم .

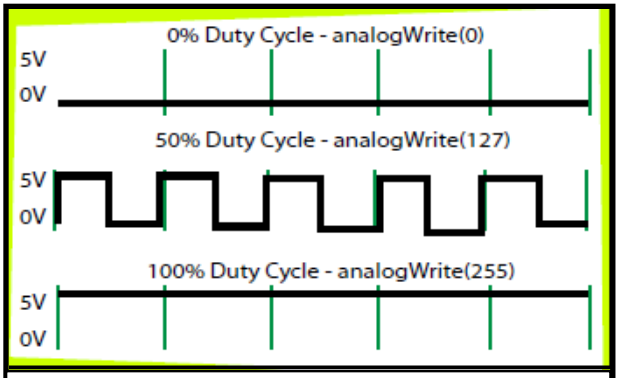
```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(A0));
}
```

در ابتدا میخواهیم بررسی کنیم چه محدوده ای از مقادیر توسط چرخاندن پتانسیومتر به آردوینو ارسال میشود . این بررسی از طریق پنجره **Serial Monitor** انجام میشود . در کد ، ابتدا شیء ارتباط سریال (**Serial**) را در تابع **setup** مقدار دهی اولیه میکنیم . آن را توسط **Baud rate** ۹۶۰۰ راه اندازی میکنیم . در تابع **loop** مقادیر آنالوگ را از پایه **A0** دریافت کرده و آن را در شیء **serial** توسط تابع **println** چاپ میکنیم که باعث پدیدار شدن مقادیر در پنجره **serial monitor** میشود .

بعد از اینکه کد را روی آردوینو بارگذاری کردیم ، با کلیک بر روی **serial monitor** پنجره ای نمایش داده میشود که مقادیری که پتانسیومتر به‌آردوینو میفرستد نمایش داده میشوند . مقادیری که خواهیم دید در محدوده ۰ تا ۱۰۲۳ خواهند بود . باچرخاندن پتانسیومتر این مقادیر کم و زیاد میشوند .



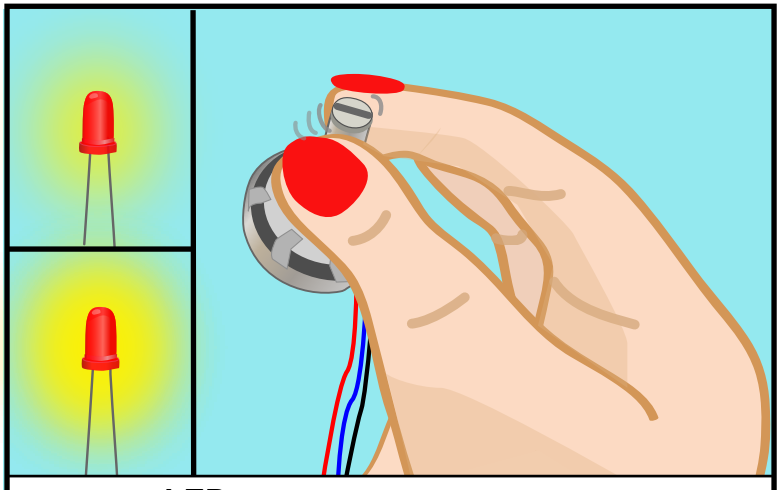
میتوانیم با بهره گیری از مقادیر پیوسته ای که از پتانسیومتر به آردوینو ارسال میشود ، نور یک **LED** را کنترل کنیم . برای اینکار قطب مثبت **LED** را توسط یک مقاومت به پایه ۴ آردوینو و قطب منفی آن را به زمین مدار وصل میکنیم



برای کنترل نور **LED** از مدولاسیون پهنای باند (**PWM**) استفاده میکنیم .
PWM یک روش برای شبیه سازی یک خروجی آنالوگ توسط قطع و وصل کردن ولتاژ در بازه های زمانی معین است . از **PWM** در پایه های ۳ ، ۵ ، ۶ ، ۹ ، ۱۰ و ۱۱ استفاده کرد .

```
int sensorValue = 0;

void setup() {
  pinMode(3,OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  analogWrite(3, sensorValue/4);
}
```

در ابتدا یک متغییر تعریف میکنیم (**sensorValue**) تا مقادیر دریافتی از پتانسیومتر را در آن ذخیره کنیم . در **setup** پایه ۳ را به صورت خروجی تنظیم میکنیم .
در **loop** مقدار دریافتی پتانسومتر در پایه **A0** را خوانده و در متغییر تعریف شده ذخیره میکنیم . سپس مقدار ذخیره شده را روی پایه ۳ برای **LED** قرار میدهیم . قبل از قرار دادن مقدار ذخیره شده روی **LED** باید آن را تقسیم بر ۴ کنیم . در این صورت بازه آن از ۰ تا ۲۵۵ خواهد بود .



با چرخاندن پتانسیومتر ، میزان روشنایی **LED** از خاموش تا پر نور ترین حالت تغییر میکند .

LINKS

SOFTWARE

SOFTWARE DOWNLOAD
HTTP://WWW.ARDUINO.CC/EN/MAIN/SOFTWARE
LANGUAGE REFERENCE
HTTP://ARDUINO.CC/EN/REFERENCE/HOMEPAGE

SUPPLIES

SPARKFUN ELECTRONICS
HTTP://WWW.SPARKFUN.COM/
ADAFRUIT INDUSTRIES
HTTP://ADAFRUIT.COM/
MAKER SHED
HTTP://WWW.MAKERSHED.COM/
SAMECO ELECTRONICS
HTTP://WWW.SAMECO.COM/

TUTORIALS

ARDUINO SITE TUTORIALS
HTTP://WWW.ARDUINO.CC/EN/TUTORIAL/HOMEPAGE
LADY ADA
HTTP://WWW.LADYADA.NET/LEARN/ARDUINO/
INSTRUCTABLES
HTTP://WWW.INSTRUCTABLES.COM/TAG/TYPE-ID/CATEGORY-TECHNOLOGY/CHANNEL-ARDUINO/

BOOKS

GETTING STARTED WITH ARDUINO BY MASSIMO BANZI
MAKING THINGS TALK: USING SENSORS, NETWORKS, AND ARDUINO TO SEE, HEAR, AND FEEL YOUR WORLD BY TOM IGOE
PHYSICAL COMPUTING: SENSING AND CONTROLLING THE PHYSICAL WORLD WITH COMPUTERS BY DAN O'SULLIVAN & TOM IGOE
ARDUINO COOKBOOK BY MICHAEL MARGOLIS

all text and drawings by Jody Culkin
for more, check out jodyculkin.com

Special Thanks to Tom Igoe, Marianne petit, Calvin Reid, The faculty and staff of the interactive telecommunications program at nyu, particularly Dan o'sullivan, Danny rozin and Red burns. thanks to Cindy karasek, chris Stein, sarah teitler, kathy goncharov & zannah marsh.
many, many thanks to the Arduino team for bringing us this robust and flexible open source platform.
and thanks to the lively, active and ever growing arduino community.

Introduction to Arduino by Jody Culkin is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

ترجمه : حسین احمدی
4hmadi.ho55ein@gmail.com