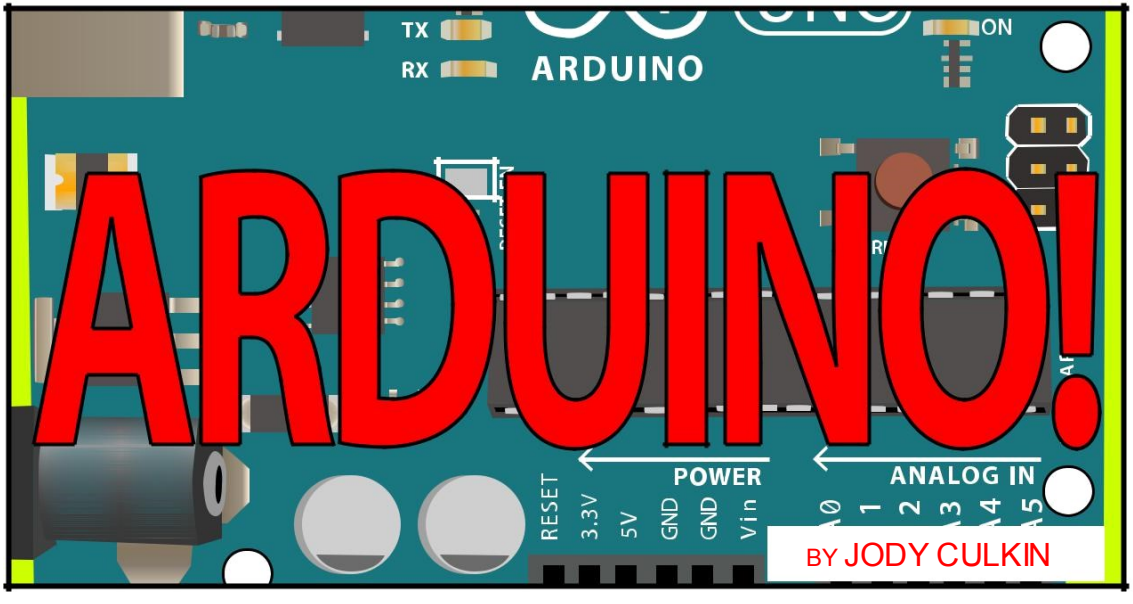
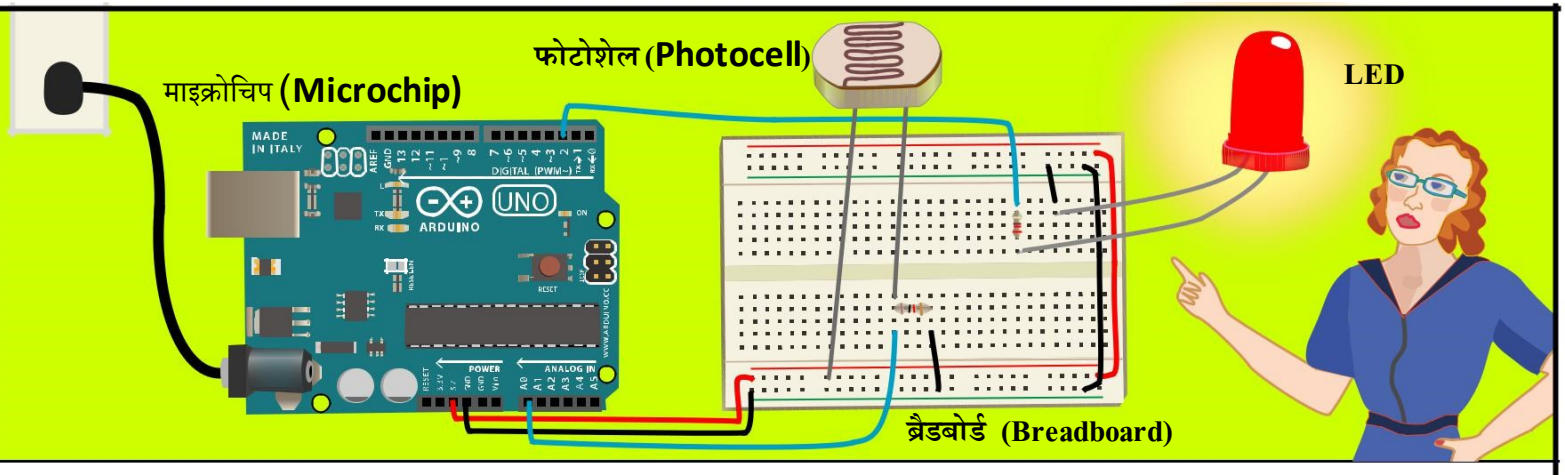


ओपन सोर्स - "संसाधनों जिनका फ्री में उपयोग या वितरित या पुन: लिखित किया जा सकता है जैसे सॉफ्टवेयर या हार्डवेयर।"

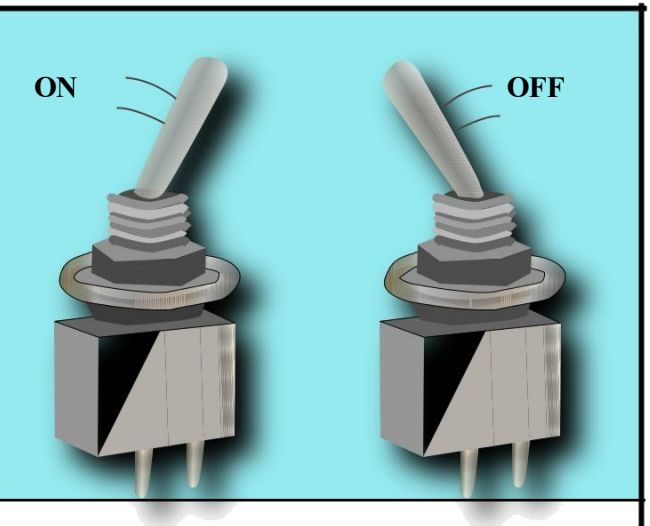
इलेक्ट्रॉनिक्स - "टेक्नोलॉजी जो विभिन्न मीडिया के द्वारा इलेक्ट्रॉनों के नियंत्रित गति का उपयोग करती है ।"

प्रोटोटाइप - "एक मूल रूप जो अन्य चीजों के लिए एक आधार, मानक और मॉडल देता है ।"

प्लेटफार्म - "सॉफ़्टवेयर फ्रेमवर्क के साथ हार्डवेयर आर्किटेक्चर जिस पर अन्य सॉफ्टवेयर चल सकते हैं ।"

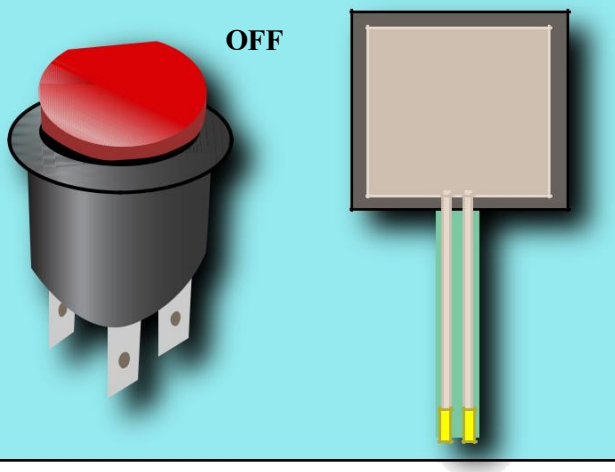


एक Arduino में **माइक्रोचिप** होता है, जो एक बहुत छोटा कंप्यूटर होता है जिसे आप प्रोग्राम कर सकते हैं। आप उस पर सेंसर लगा सकते हैं जिससे वातावरण की कंडीशंस को माप सकते हैं (जैसे कमरे में कितना प्रकाश है)। दूसरी चीज़ो को इन कंडीशंस पर क्या करना है को कण्ट्रोल कर सकता है | (जब कमरे में अंधेरा होता है, अपने आप जल जाती है|)

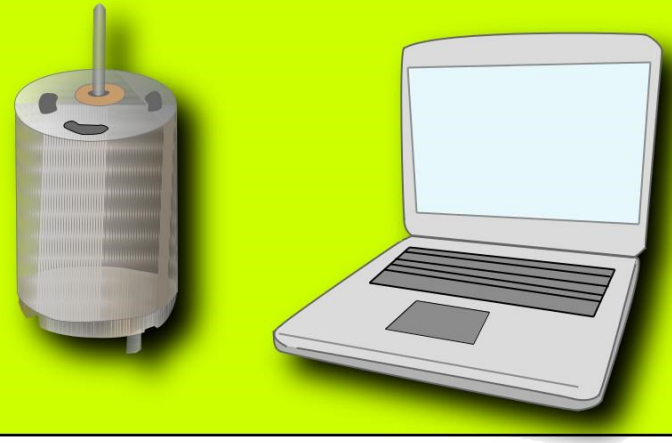


जब एक स्विच दबाया जाता है, तब यह कुछ एक्शन ले सकता है|

माइक्रोकंट्रोलर, एक कंप्यूटर की तरह **इनपुट** और **आउटपुट** का उपयोग करते हैं | इनपुट, यूजर या पर्यावरण से इनफार्मेशन को कैप्चर करते हैं, जबकि प्राप्त इनफार्मेशन पर प्रोसेस के बाद, आउटपुट उसको स्क्रीन पर दर्शाता है |



Arduino में इनपुट, एक स्विच या सेंसर हो सकता है |



किसी वस्तु को एक आउटपुट के द्वारा ऑन, ऑफ और नियंत्रित किया जा सकता है। यह एक मोटर या एक कंप्यूटर भी हो सकता

इनपुट और आउटपुट, **डिजिटल** या **एनालॉग** हो सकते हैं | डिजिटल इनफार्मेशन बाइनरी होती है- यह या तो TRUE या FALSE होता है| एनालॉग इनफार्मेशन निरंतर होती है, यह कई वैल्यूज ले सकती है|

स्विच एक डिजिटल इनपुट और सेंसर एक एनालॉग इनपुट है| एनालॉग सेंसर की रेंज, इसको डिजिटल डेटा में बदलने से निर्धारित होती है |

Arduino को प्लग करने से पहले, हमें कुछ शब्दों और सिद्धांतों की समीक्षा करनी चाहिए कि इलेक्ट्रिसिटी (और इसलिए इलेक्ट्रॉनिक्स) कैसे काम करती है ?

वोल्टेज (V)

एक सर्किट में विद्युत क्षमता को मापता है । यह वोल्ट (**volts**) में मापा जाता है |

करंट (I)

एक सुचालक मैटेरियल में विद्युत प्रवाह की मात्रा है यह एम्पीयर (**amperes**) या एम्पप्स (**amps**) में मापा जाता है|

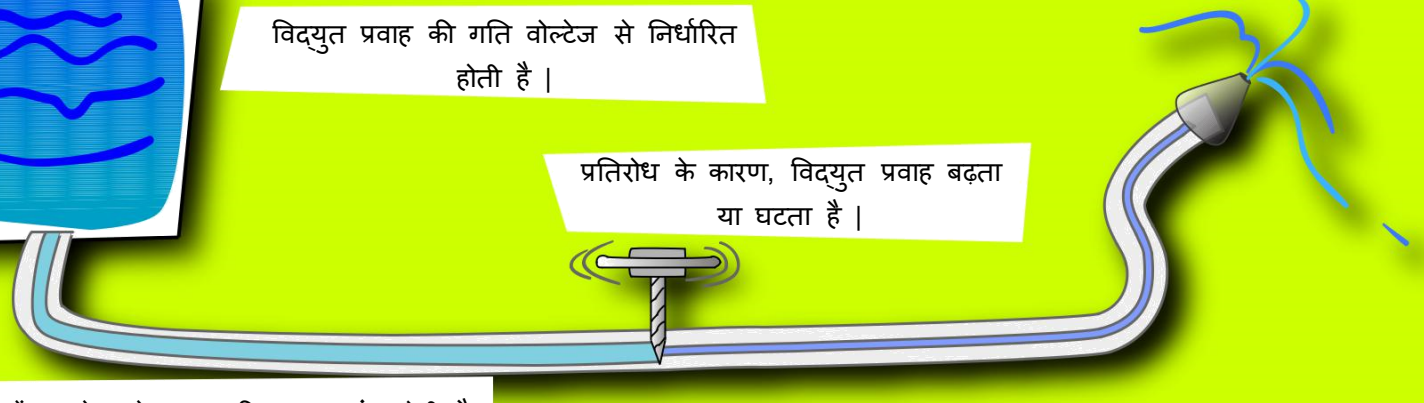
प्रतिरोध (R)

विद्युत धारा के प्रवाह का अवरोधक है। यह ओम (**ohms**) में मापा जाता है |

विद्युत एक सुचालक मैटेरियल में विद्युत ऊर्जा का प्रवाह है।

विद्युत प्रवाह की गति वोल्टेज से निर्धारित होती है |

प्रतिरोध के कारण, विद्युत प्रवाह बढ़ता या घटता है |



पाइपों में चलने वाले प्रवाह की मात्रा **करंट** होती है|

सामान्यतः इन शब्दों को समझाने के लिए पानी के समानता का उपयोग किया जाता है| यहाँ एक मॉडल है |

Ohm's law ओम का नियम

Current = Voltage/Resistance

(I= V/R) **करंट = वोल्टेज / प्रतिरोध**

OR

Resistance = Voltage/Current

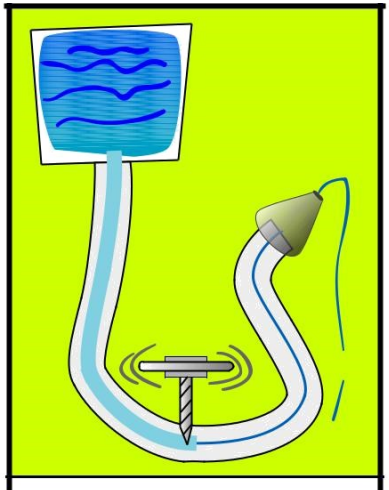
(R = V/I) प्रतिरोध = वोल्टेज / **करंट**

OR

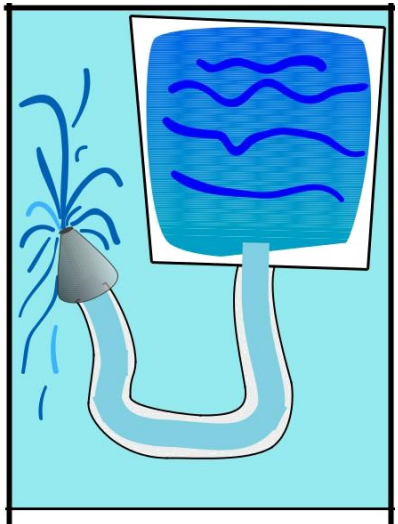
Voltage = Resistance * Current

(V = R*I) वोल्टेज = प्रतिरोध * **करंट**

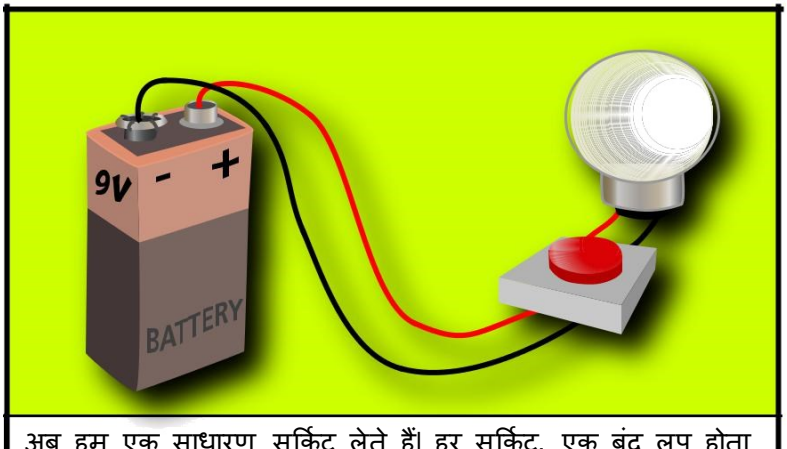
एक जर्मन भौतिक विज्ञानी जॉर्ज ओम द्वारा की गई वोल्टेज, वर्तमान और प्रतिरोध के बीच संबंध है |



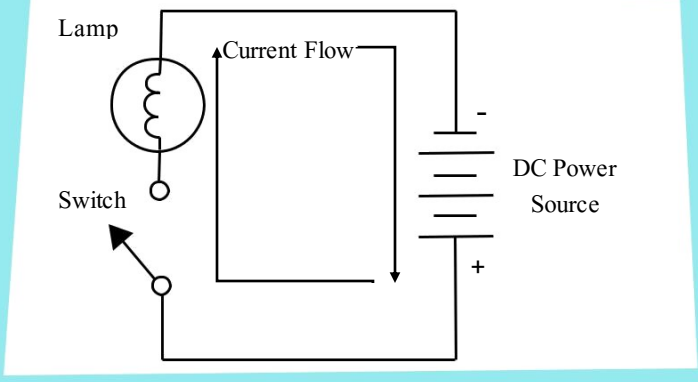
उदाहरण के लिए:
प्रतिरोध में वृद्धि => कम प्रवाह



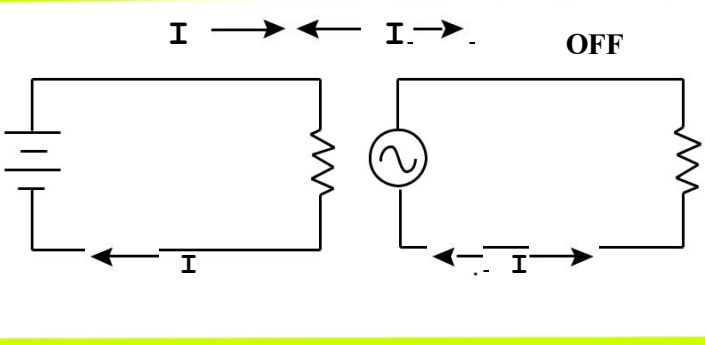
वोल्टेज में वृद्धि => अधिक प्रवाह



अब हम एक साधारण सर्किट लेते हैं| हर सर्किट, एक बंद लूप होता है, जिसमें एक ऊर्जा स्रोत (बैटरी) और load (लैंप) होता है। Load, बैटरी की विद्युत ऊर्जा को परिवर्तित करके इसका उपयोग करती है। इस सर्किट में एक स्विच भी है |

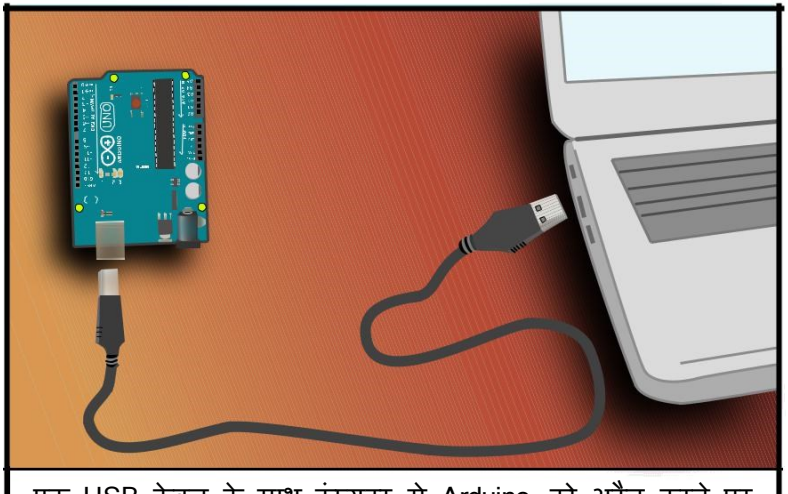


यह एक ही सर्किट का एक स्कीमैटिक (**schematic**) है (यह इलेक्ट्रॉनिक कंपोनेंट्स के लिए सिम्बल्स का उपयोग कर सर्किट को दर्शाता है)। जब स्विच ऑन होता है, तो बिजली के स्रोत से करंट प्रवाह होता है और लैंप ऑन हो जाता है ।



सर्किट के दो सामान्य प्रकार हैं, **direct current** और **alternating current** । डी.सी. सर्किट में, करंट हमेशा एक दिशा में बहती है। ए.सी. सर्किट में, नियमित cycles में विपरीत दिशाओं में करंट प्रवाह होता है। हम यहाँ केवल डीसी सर्किट्स के बारे में ही बात करेंगे ।

Arduino को चलाने के लिए पावर की जरूरत होती है, प्रोग्राम करने के लिए इसे कंप्यूटर से अटैच करना होगा।



एक USB केबल के साथ कंप्यूटर से Arduino को अटैच करने पर, Arduino ऑन हो जाता है और अब हम प्रोग्रामिंग शुरू कर सकते हैं|

आपको Arduino को प्रोग्राम करने के लिए सॉफ़्टवेयर डाउनलोड और इंस्टॉल करना होगा। यह ऊपर दिए URL पर फ्री में उपलब्ध है | Arduino सॉफ्टवेयर Mac OS X, विंडोज (Windows) और लिनक्स (Linux) प्लेटफार्मों पर चलता है।

Mac पर Arduino सॉफ़्टवेयर को इनस्टॉल करने के निर्देशों के लिए:

HTTP://WWW.ARDUINO.CC/EN/GUIDE/MACOSX

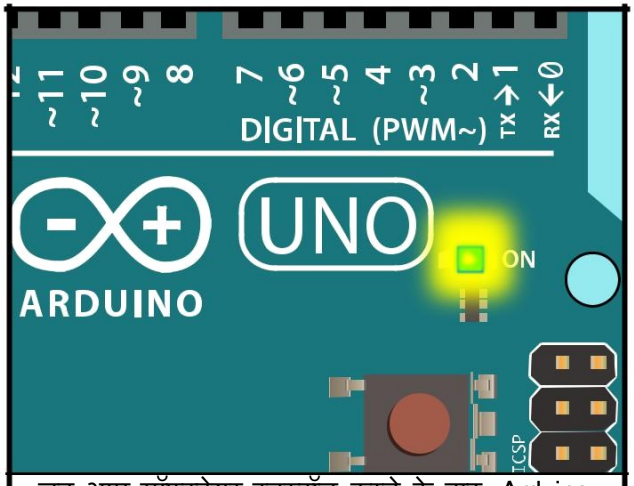
Windows पर इंस्टॉल करने के निर्देशों के लिए:

HTTP://WWW.ARDUINO.CC/EN/GUIDE/WINDOWS

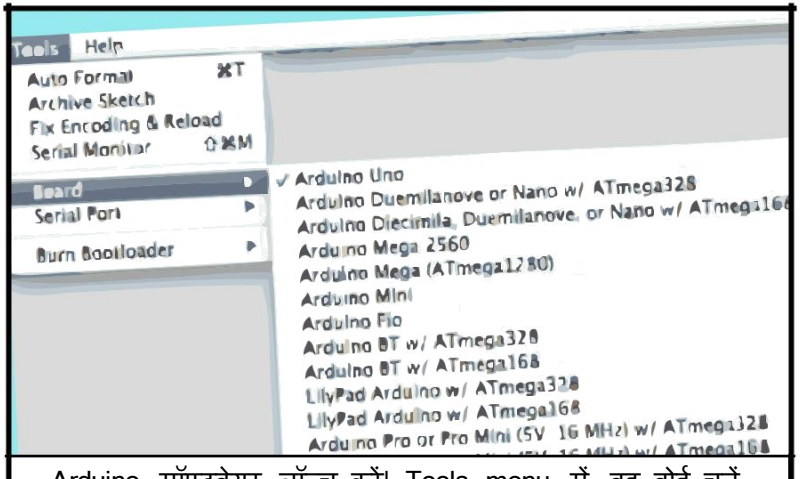
Linux पर इंस्टॉल करने के निर्देशों के लिए:

HTTP://WWW.ARDUINO.CC/PLAYGROUND/LEARNING/LINUX

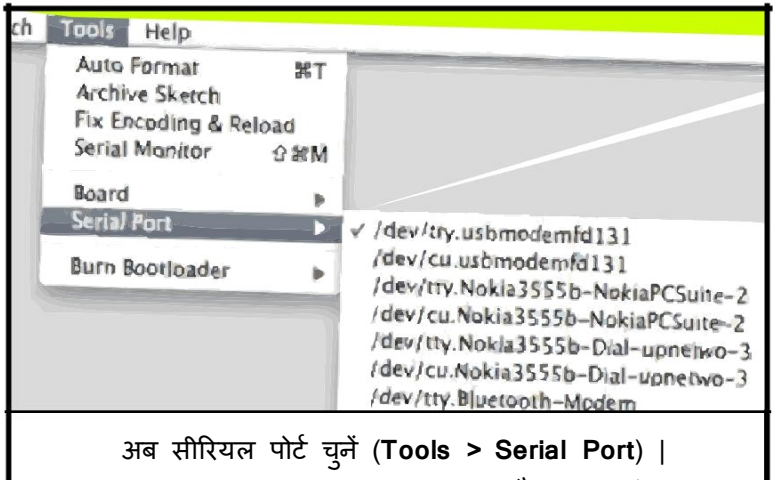
इन प्लेटफार्मों पर सॉफ़्टवेयर इनस्टॉल करने के बारे में विस्तृत निर्देशों के लिए ऊपर दिए URLs पर जाएं |



जब आप सॉफ्टवेयर इनस्टॉल करने के बाद, Arduino कनेक्ट करते हैं तो बोर्ड पर स्थित एक LED ऑन हो जानी चाहिए |

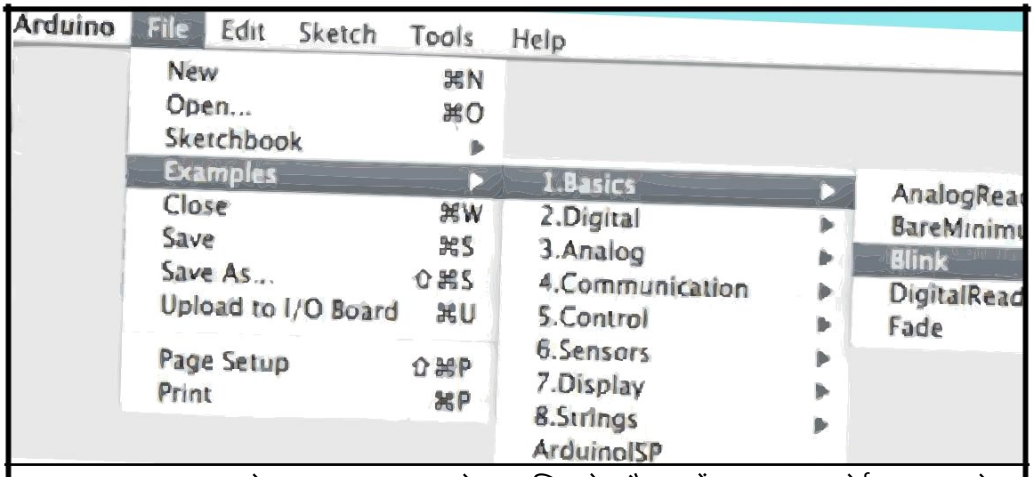


Arduino सॉफ्टवेयर लॉन्च करें| Tools menu में, वह बोर्ड चुनें जिसका उपयोग आप कर रहे हैं **(Tools > Board)**, जैसे **Arduino Uno** |



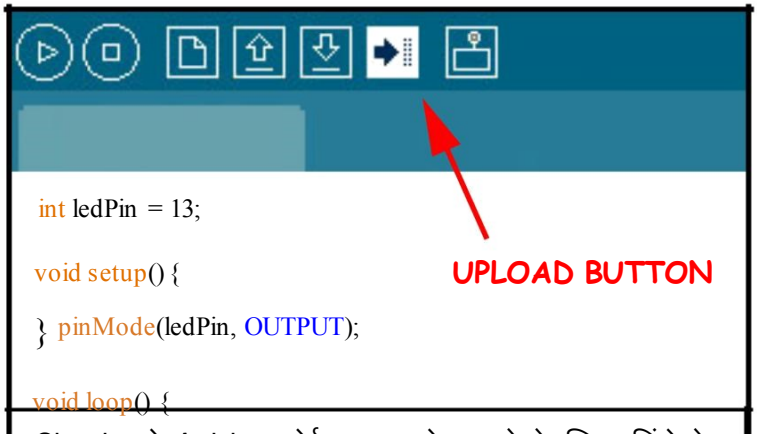
अब सीरियल पोर्ट चुनें (**Tools > Serial Port**) |
Mac पर यह /dev/tty.usbmodem जैसा कुछ होगा,
एक विंडो (Windows) मशीन पर, यह **COM3** या ऐसा कुछ होगा |

जब आप Arduino सॉफ़्टवेयर डाउनलोड करते हैं, तो आप एक **IDE** डाउनलोड करते हैं, जो प्रोग्रामर को सॉफ्टवेयर विकसित करने के लिए, यह एक Text Editor को कंपाइलर और अन्य फीचर्स के साथ जोड़ती है|

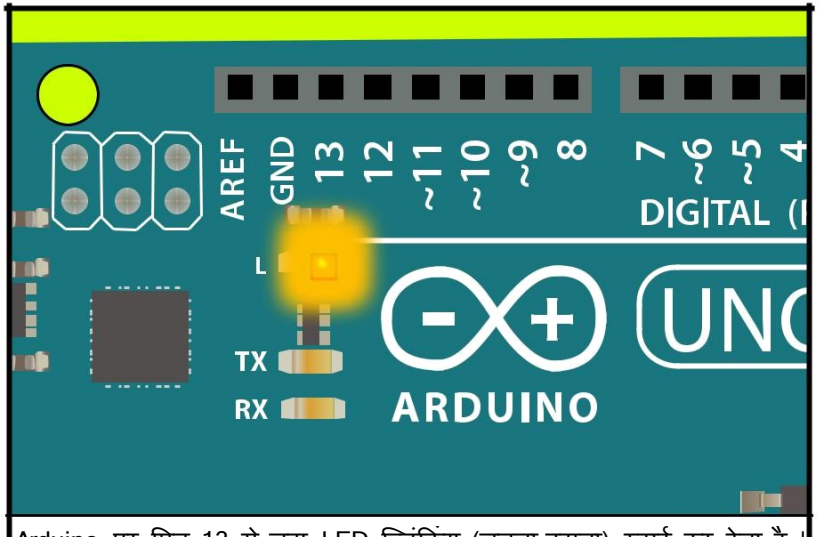


Arduino IDE आपको **sketches**, या प्रोग्राम लिखने और उन्हें Arduino बोर्ड पर अपलोड करने की में सहायता करता है। फ़ाइल मेनू में **Blink** उदाहरण खोलें।

File > Examples > 1.Basics > Blink



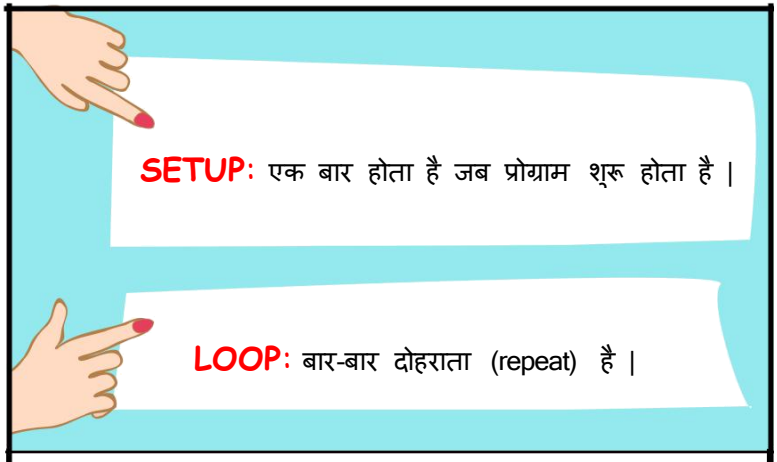
Sketch को Arduino बोर्ड पर अपलोड करने के लिए, विंडो के top पर स्थित बटनों की पट्टी पर **UPLOAD BUTTON** पर क्लिक करें। क्लिक के बाद, विंडो के निचले भाग में कुछ मैसेज आने लगेंगे, अंत में **Done Uploading** आएगा |



Arduino पर पिन 13 से जुड़ा LED ब्लिंकिंग (जलना-बुझना) स्टार्ट कर देता है |

```
void setup() {
  // initialize the digital pin as an output.
  // Pin13 has LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);            // wait for a second
}
```

किसी स्केच, किसी भी भाषा में लिखे गए कार्यक्रम की तरह, कंप्यूटर के लिए निर्देशों का एक सेट है| अगर हम Blink sketch को ध्यान से देखें, तो हम देखते हैं कि इसके दो बड़े पार्ट्स, **setup** और **loop** हैं |



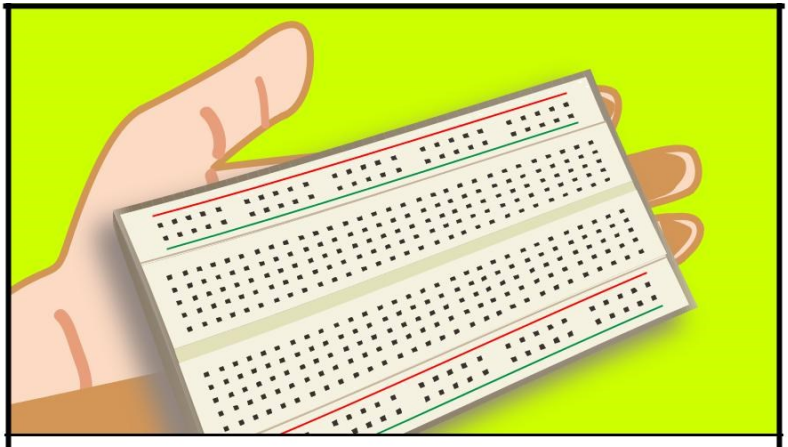
ये कोड के दोनों पार्ट्स, **functions** कहलाते हैं जो कि प्रत्येक स्केच में होते हैं और ये curly braces { } के अंदर ही रहते हैं |

HTTP://ARDUINO.CC/EN/REFERENCE/HOMEPAGE

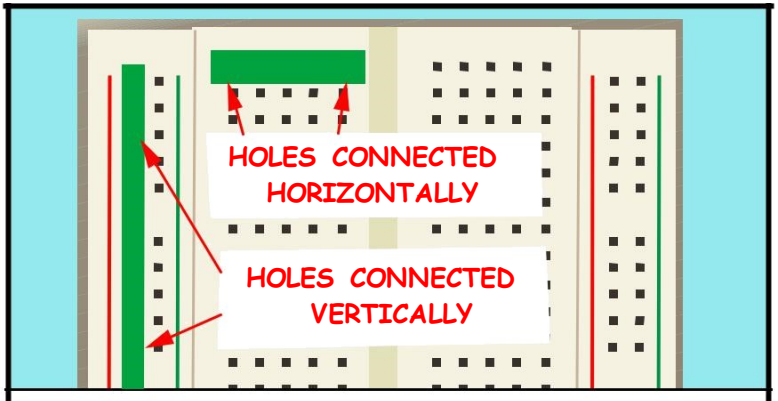
Arduino प्रोग्रामिंग भाषा सीखने के लिए जरूरी Arduino संदर्भ गाइड और कई अन्य संसाधनों के लिए ऊपर दिए Arduino वेबसाइट को देखें |

```
void setup() {           //DECLARES BLOCK OF CODE
  pinMode(13, OUTPUT);   //SETS PIN 13 TO OUTPUT
}                        //END BLOCK OF CODE


void loop() {              //DECLARES BLOCK OF CODE
  digitalWrite(13, HIGH);  //SETS PIN 13 HIGH
  delay(1000);             //PAUSE 1 SECOND
  digitalWrite(13, LOW);   //SETS PIN 13 LOW
  delay(1000);             //PAUSE 1 SECOND
}                          //END BLOCK OF CODE
```

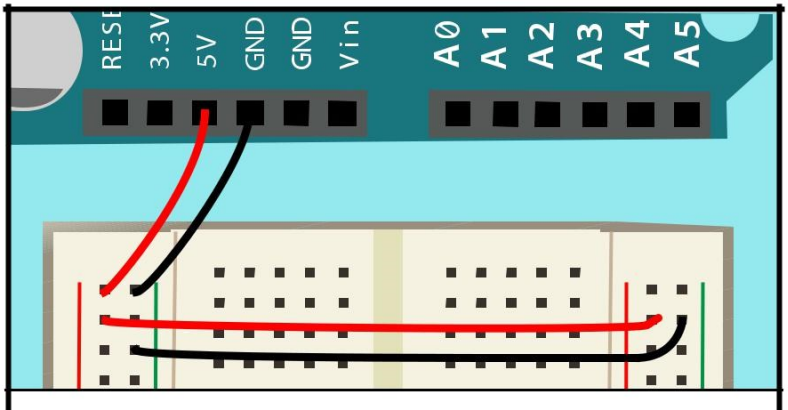
अब इस सरल प्रोग्राम (sketch) की हर लाइन को देखेंगे, कि हर लाइन में हो क्या रहा है ?



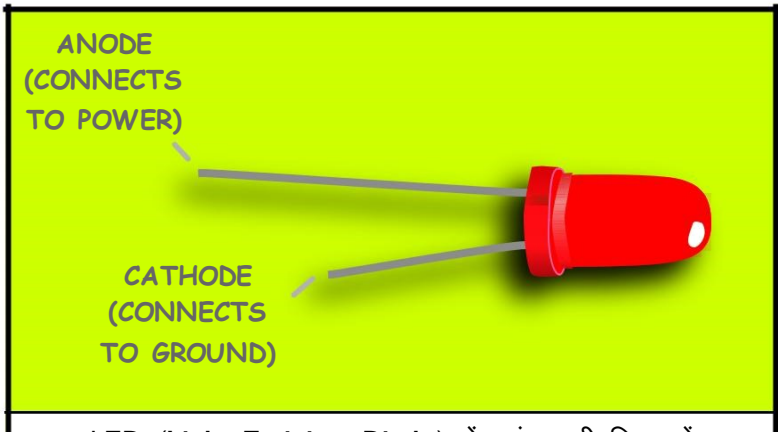
हम उन ऑब्जेक्टों को कैसे नियंत्रित करते हैं जो Arduino बोर्ड पर नहीं हैं? हम Arduino को एक शोल्डरलेस ब्रेडबोर्ड से जोड़ेंगे। यह हमें कम समय में सर्किट बनाने और उसको टेस्ट करने में मदद करेगा |



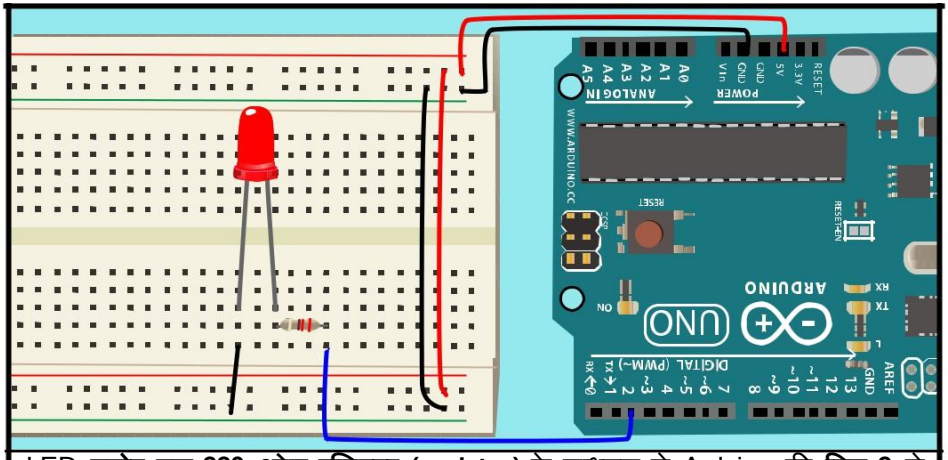
इस ब्रेडबोर्ड में बायीं और दाहिनी साइड पर छेदों की 2 पंक्तियाँ हैं, और एक मध्य के दोनों तरफ छेद की 5 पंक्तियाँ हैं। साइड वाली पंक्तियाँ वर्टिकली (**vertically**) कनेक्ट रहती हैं | मध्य में हर पंक्ति में 5 होल्स हॉरिज़ॉन्टली / क्षैतिज (**horizontally**) कनेक्ट रहते हैं |



हम Arduino बोर्ड से पावर और ग्राउंड को vertically जुड़े स्ट्रिप्स में बाएं और दाईं ओर, 22 गेज तार के साथ जोड़ेंगे । अन्य कंपोनेंट्स मध्य के होल्स, पावर और ग्राउंड के साथ आवश्यकतानुसार जोड़े जा सकते हैं |



जब LED (**Light Emitting Diode**) में करंट सही दिशा में प्रवाह होता है, तो यह रोशनी देने लगता है | हम breadboard पर एक LED को कनेक्ट करेंगे, और फिर Arduino से, ताकि हम इसे अपने कोड के साथ नियंत्रित कर सकें।

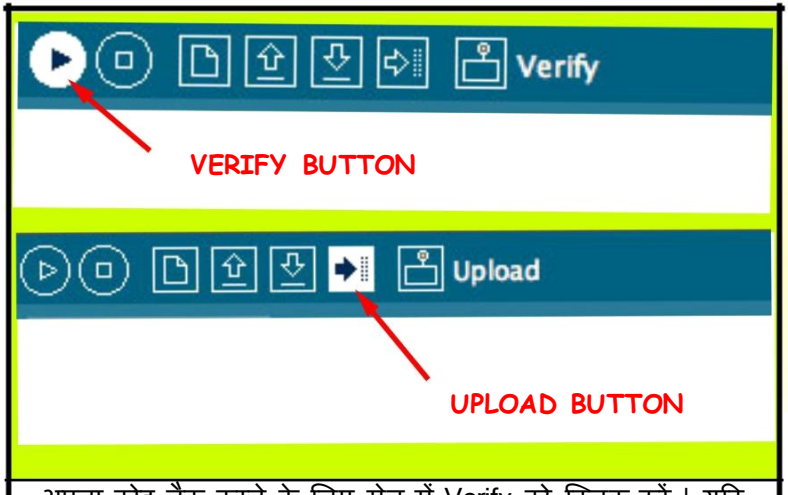


LED **एनोड** एक **220 ओम रसिस्टर (resistor)** के माध्यम से Arduino की **पिन 2** से जुड़ा हुआ है। **कैथोड ग्राउंड** से जुड़ा हुआ है। पिन 2 से 13 तक सब पिनों को डिजिटल इनपुट या आउटपुट के रूप में कॉन्फ़िगर किया जा सकता है।

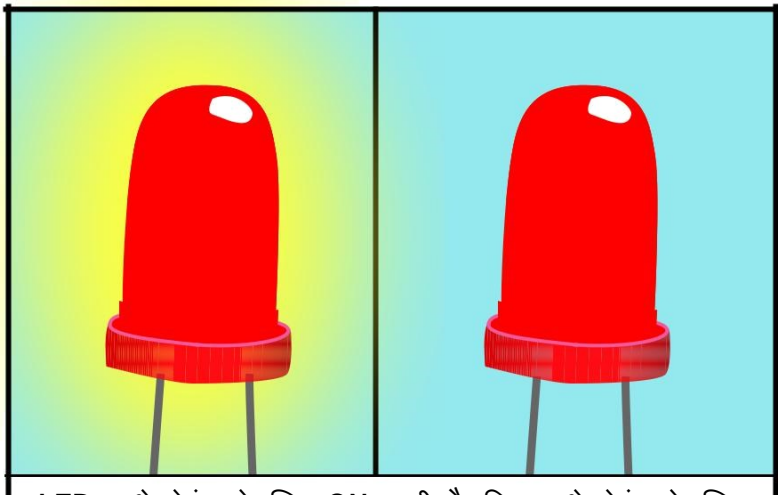
स्केच शुरू करने के लिए **new** बटन क्लिक करें।

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

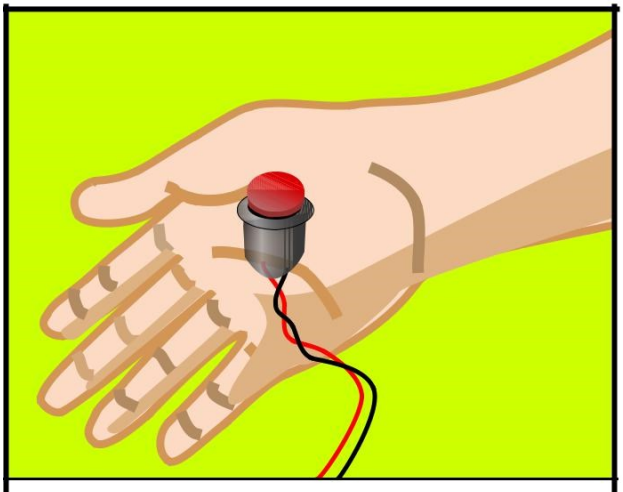
SETUP में, हम पिन-2 पर आउटपुट सेट करते हैं। **LOOP** में, पहले हम पिन-2 को हाई सेट करते हैं, जिससे वो जलने लगती है | डिले (Delay) 500 मिलीसेकंड या आधे सेकंड का रखते है। जब पिन-2 low सेट होता है, तो एलईडी बंद हो जाता है, उसके बाद दूसरे आधे सेकंड का डिले देते हैं।



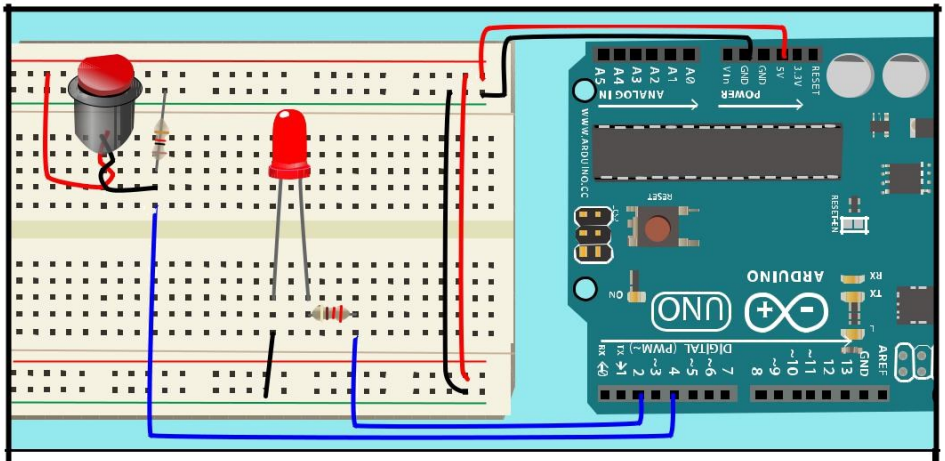
अपना कोड चैक करने के लिए मेनू में Verify को क्लिक करें | यदि कोई भी त्रुटियां नहीं हैं, तो अपने कार्यक्रम को Arduino पर चलाने के लिए Upload को क्लिक करें |



LED आधे सेकंड के लिए **ON** रहती है, फिर आधे सेकंड के लिए **OFF** रहती है और ये बार-बार रिपीट होता है |



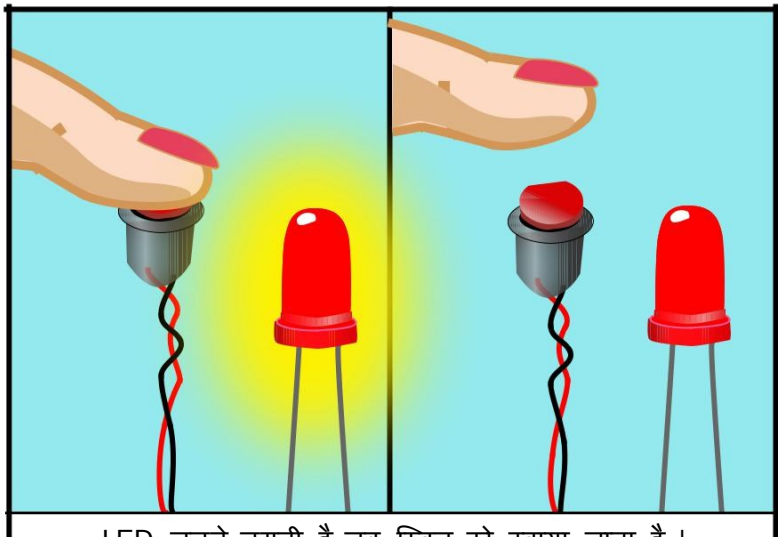
अब हम एक स्विच और एक डिजिटल इनपुट को जोड़ेंगे, ताकि हम एलईडी को ON और OFF कर सकें |



Arduino पर पिन 4 से एक क्षणिक स्विच के एक छोर को कनेक्ट करें और एक 10k ओम के रेसिस्टर का एक छोर ग्राउंड से जुड़ा हो | दूसरे छोर को पावर से कनेक्ट करें | हम LED तो उसी पिन से कनेक्ट कर देंगे |

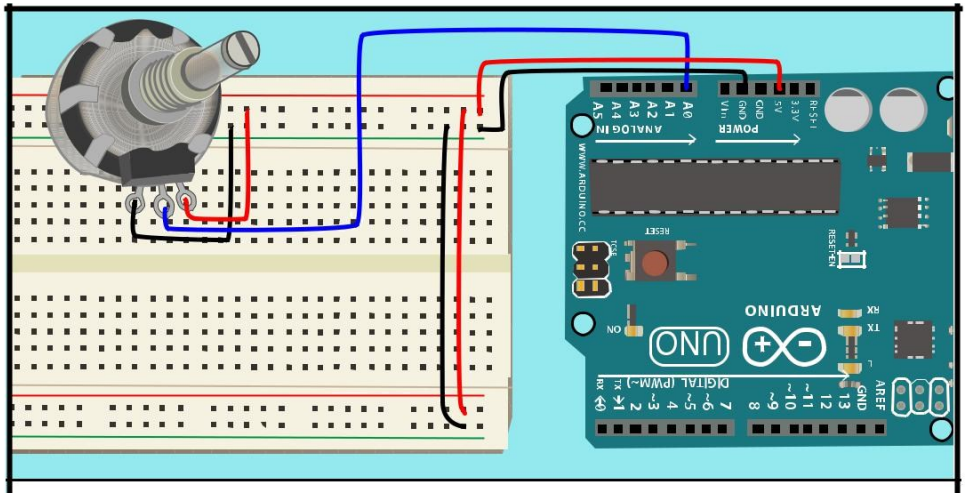
```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
}
void loop() {
  if(digitalRead(4)){
    digitalWrite(2, HIGH);
  }else{
  digitalWrite(2, LOW);
  }
}
```

अब हम कोड लिखेंगे| **SETUP** में, हम पिन-2 को आउटपुट और पिन-4 को इनपुट बनाते हैं। **LOOP** में, हम एक **if** स्टेटमेंट का उपयोग करते हैं, अगर हम पिन-4 को **HIGH** पढ़ते हैं, तो हम LED पिन को High सेट करते हैं, अन्यथा हम एलईडी पिन को LOW सेट करते हैं, जिससे ये बंद हो जाती है |



LED जलने लगती है जब स्विच को दबाया जाता है |

अब हम एक एनालॉग इनपुट का सेट अप करेंगे और एक potentiometer का उपयोग करेंगे।

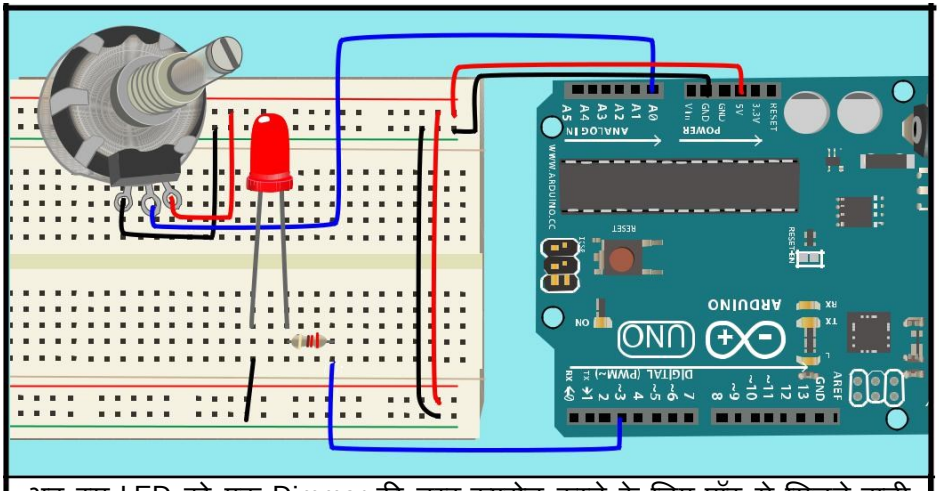


एनालॉग पिन A0 को potentiometer की मध्य पिन से कनेक्ट करें। पॉट का एक छोर पावर से और दूसरा ग्राउंड से कनेक्ट करें |

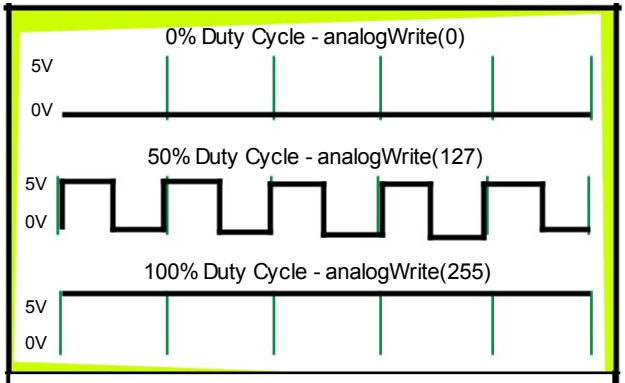
```
void setup() {
  Serial.begin(9600);
}

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

हम सीरियल मॉनिटर (**serial monitor**) का इस्तेमाल करते हुए पॉट के नॉब को घूमने से मिलती हुयी वैल्यूज़ को देखेंगे। अपने कोड में, हम **SETUP** कोड में सीरियल ऑब्जेक्ट को इनिशियलाइज़ करते हैं, 9600 की बॉड रेट (Baudrate) सेट करते हैं। **LOOP** में, हम एनालॉग पिन **A0** से वैल्यूज़ पढ़ते हैं और **printLn** फ़ंक्शन का उपयोग करके सीरियल ऑब्जेक्ट पर प्रिंट करते हैं ।

Sketch को Arduino में अपलोड करने के बाद, सीरियल मॉनिटर (**serial monitor**) बटन पर क्लिक करें, ताकि pot की वैल्यूज़ सीरियल मॉनिटर पर देख सकें | एक नयी विंडो ओपन होती है और 0 से 1023 तक वैल्यूज़ प्रिंट होती हैं |



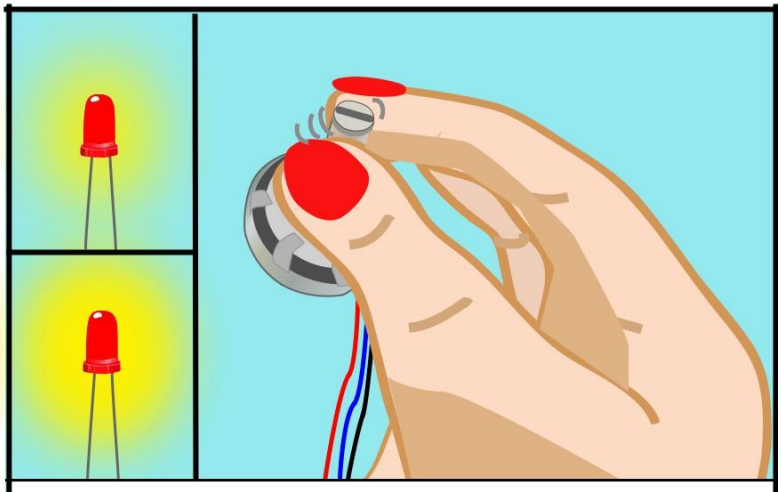
अब हम LED को एक Dimmer की तरह कण्ट्रोल करने के लिए पॉट से मिलने वाली वैल्यूज को इस्तेमाल करेंगे, LED एनोड को रेसिस्टर (resistor) के द्वारा बोर्ड की पिन-3 से कनेक्ट करें और कैथोड को ग्राउंड से |



हम **Pulse Width Modulation** (PWM) का प्रयोग करेंगे। वोल्टेज को विभिन्न रेट या या DUTY CYCLE से बदल कर, यह एनालॉग सिग्नल को सिमुलेट करने का एक तरीका है | आप 3, 5, 6, 9, 10 और 11 पिनों पर पीडब्ल्यूएम का उपयोग कर सकते हैं |

```
int sensorValue = 0;

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

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

पहले हम pot की वैल्यू को स्टोर करने के लिए एक वैरिएबल बनाते हैं। Setup में हम पिन 3 को एक आउटपुट बनाते हैं। Loop में, हम हमारे वैरिएबल में पिन A0 से पढ़े गए वैल्यू को स्टोर करते हैं। फिर हम पिन 3 पर वैल्यू लिखते हैं | अब हमें वेरिएबल तो 4 से डिवाइड करना होगा, इसलिए हमारे पास 0 से 255 तक की वैल्यूज होंगी |



जैसे ही पॉट को पावर देते हैं, की चमक ऑफ से पूरी चमक तक बदलती है ।

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/

Jameco Electronics
http://www.jameco.com/

TUTORIALS

Arduino site Tutorials
http://www.arduino.cc/en/Tutorial/HomePage

The IoT Academy
https://www.theiotacademy.co/

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

UniConverge Technologies Pvt Ltd (www.uniconvergetech.in) की IoT Team को देवनागरी (हिंदी) स्क्रिप्ट में रूपांतरण की समीक्षा करने के लिए विशेष धन्यवाद |

विशेष धन्यवाद to Tom Igoe, Marianne Petit, Calvin Reid, the faculty and staff of the Interactive Telecommunications.

हमें यह Robust और Flexible ओपन सोर्स प्लॅटफॉम देने के लिए, Arduino टीम को बहुत बहुत धन्यवाद |

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