

Arduinon perusteet

Kuinka aloittaa?

Lataa Arduino IDE netistä

Hyvä lähdekirja aloittamiseen on [Getting Started With Arduino](#) (löytyy myös suomennettuna).

Lisäksi tarvitset Arduino mikrokontrollerin, usb-kaapelin, peruskomponentteja sekä koekytkentälevyn.



Ohjelmointi Arduino IDE-ohjelmointiympäristössä tapahtuu tekstipohjaisesti. Jos kuitenkin haluat ohjelmoida graafisella, Scratchiin pohjautuvalla ohjelmointiympäristöllä, tutustu esimerkiksi S4A-ohjelmointiympäristöön. www.s4a.cat Huomaa, että tällöin ohjelmoitu laite on pidettävä kiinni tietokoneen usb-portissa. Netistä löytyy myös ratkaisuja graafiseen ohjelmointiin ja koodin lataamiseen Arduinoon, jolloin usb-kaapelia ei tarvitse pitää laitteessa kiinni. Tässä opastuksessa emme kuitenkaan käsittele tätä aihetta, vaan keskitymme ohjelmointiin.



```
sketch_feb10a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino/Genuino Uno on /dev/cu.usbmodem1421

Arduino-koodin rakenne

Arduinossa on aina kaksi koodiosaa, setup ja loop. Setup on kuin käynnistys tai alustus, ja loop on toistettava koodi joka kulkee nimensä mukaisesti alusta loppuun ja takaisin alkuun.

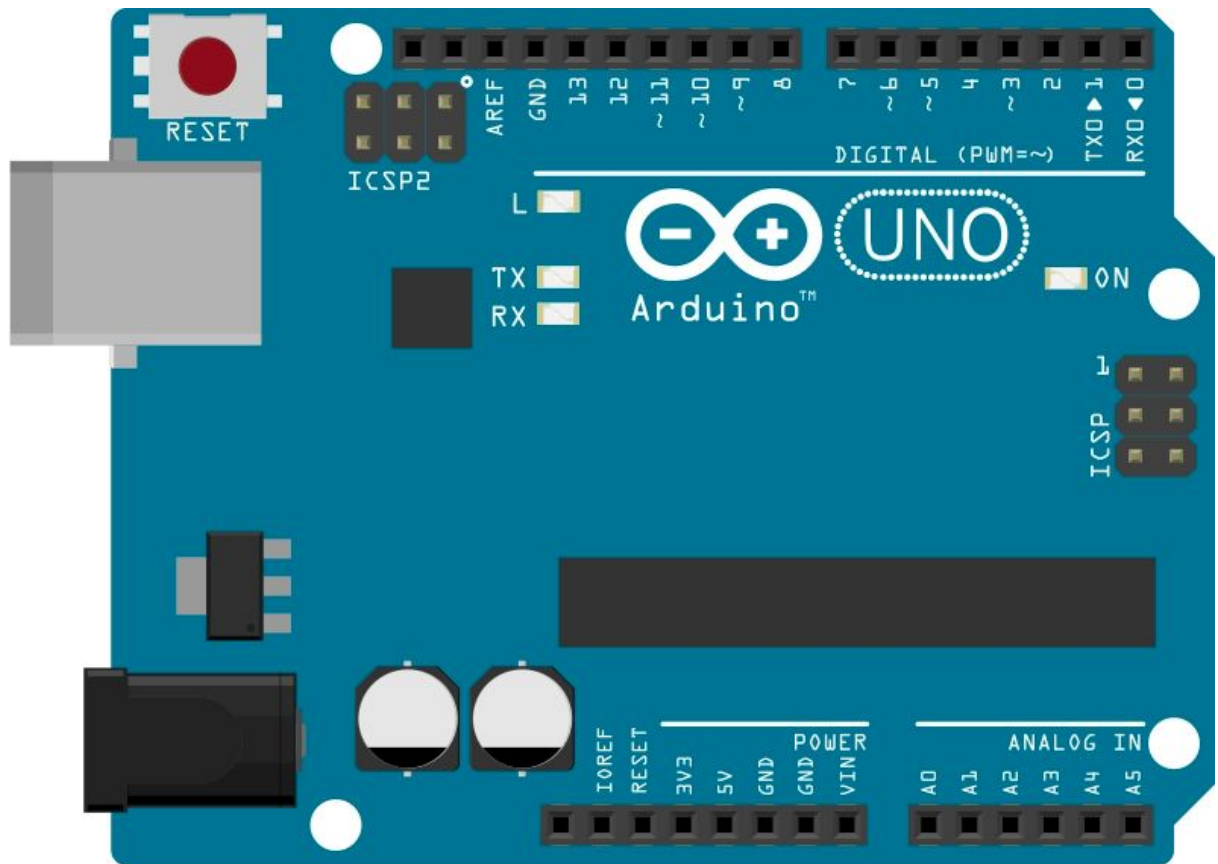
- [setup\(\)](#)

Pyörähtää kerran kun Arduino käynnistetään

Käytä setup() funktiota asioihin jotka tulee tehdä vain kerran, kuten serial portin avaaminen, pinnien määrittäminen jne.

- [loop\(\)](#)

Pyörii niin kauan kuin Arduinossa on virtaa

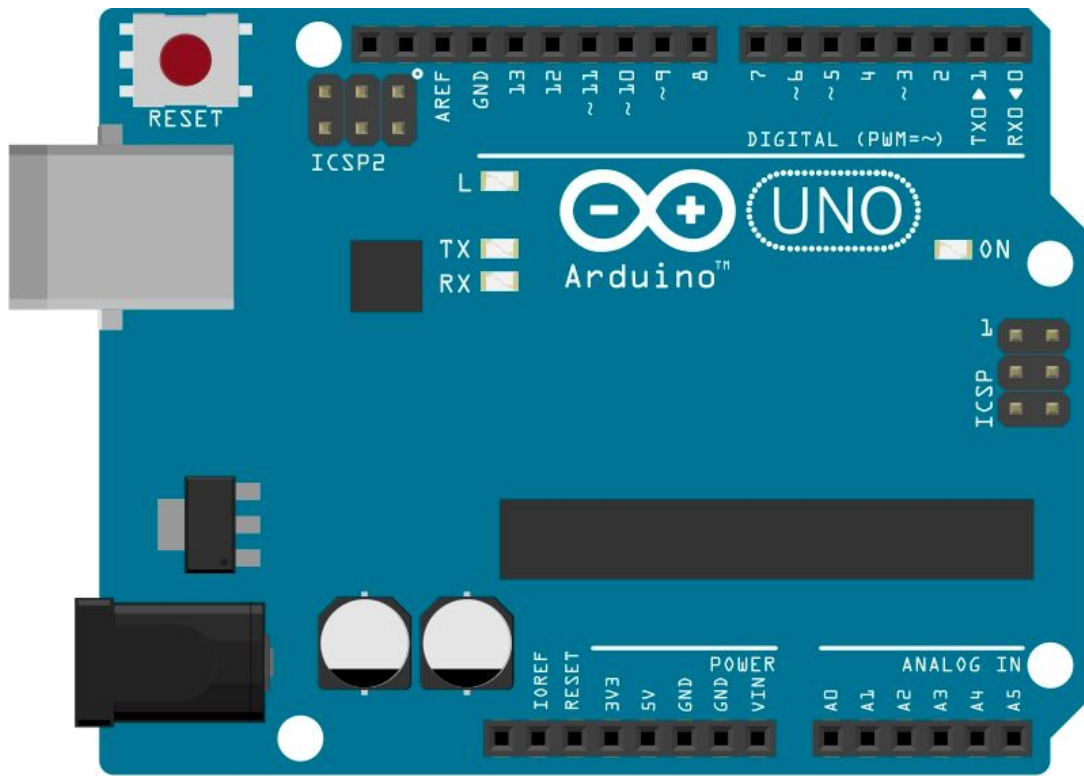


fritzing

Arduino

Arduino is a open source electronics prototyping platform.

- 5V voltage pin
- Ground (GND)
- Digital I/O pins (including PWM)
- Analog pins

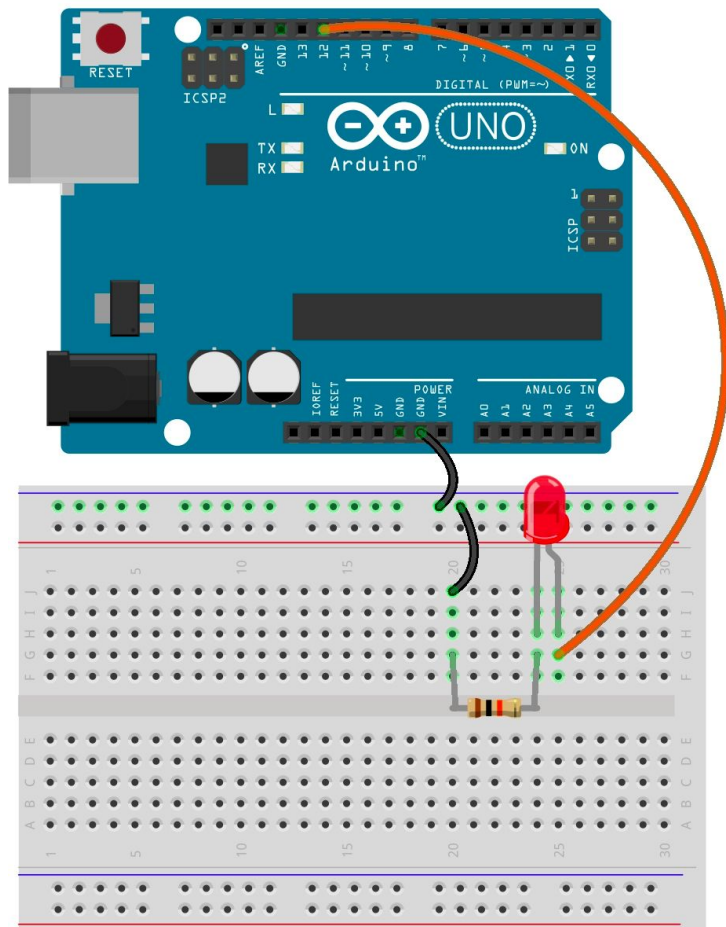


fritzing

Breadboard

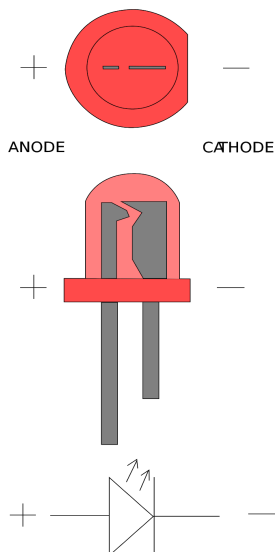
Easy prototyping without soldering.

Led päälle ja pois, eli *blink*



fritzing

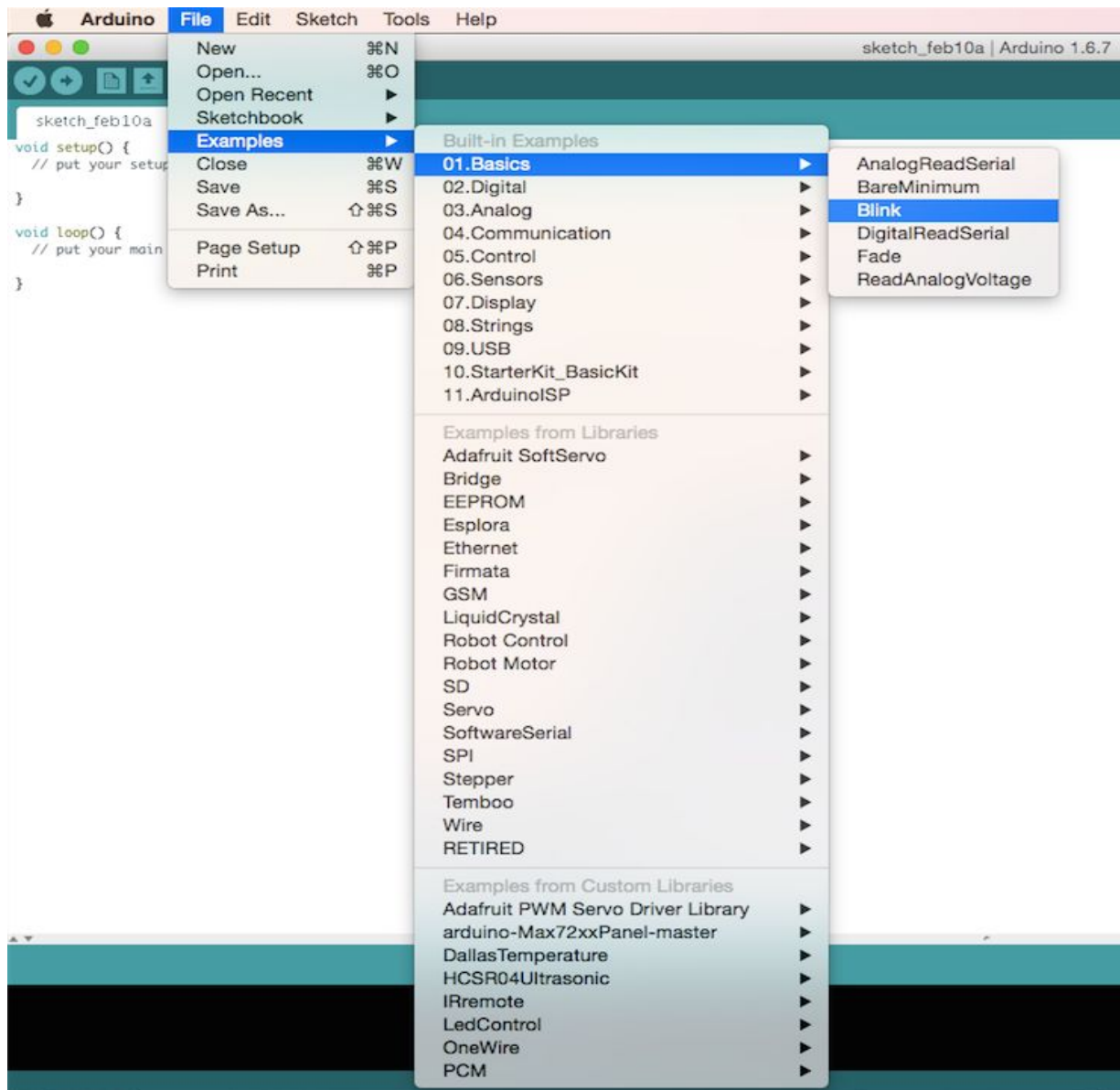
Led (light emitting diode)



- + and - legs, anode and cathode
- Needs a resistor (ie. ledcalc.com) → opettele Ohmin laki!

Sample code:

```
void setup() {  
  
    pinMode(12, OUTPUT);  
  
}  
  
void loop() {  
  
    digitalWrite(12, HIGH);  
    delay(1000);  
    digitalWrite(12, LOW);  
    delay(1000);  
}
```

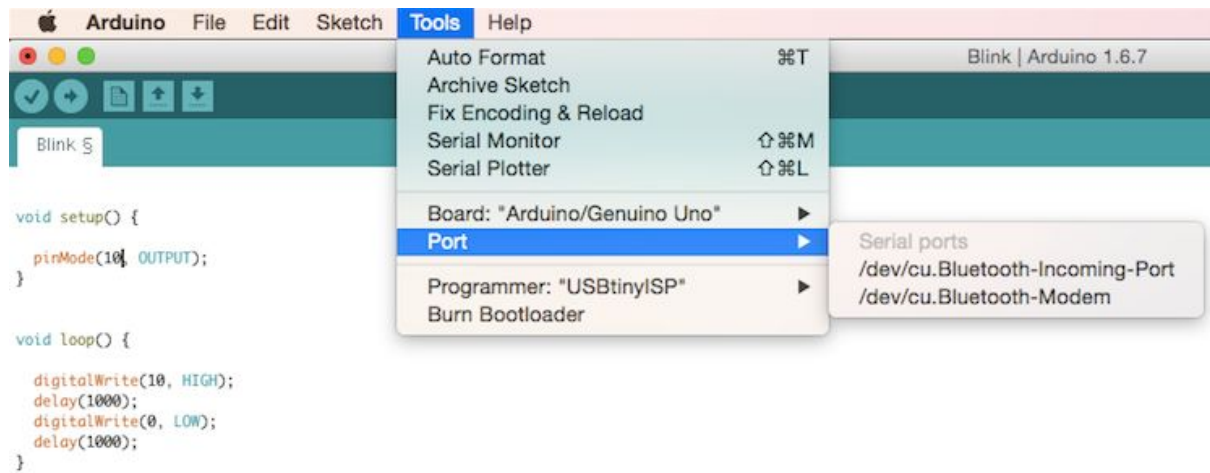


Kerro koneelle ensimmäistä kertaa koodia ladatessasi, mistä ja minne koodin tulee kulkea usb-kaapelia pitkin

Upload your code to the board. Check that you have the correct **Board** and **Serial Port** selected from the **Tools** menu. Go through the getting started part again, if you have problems.

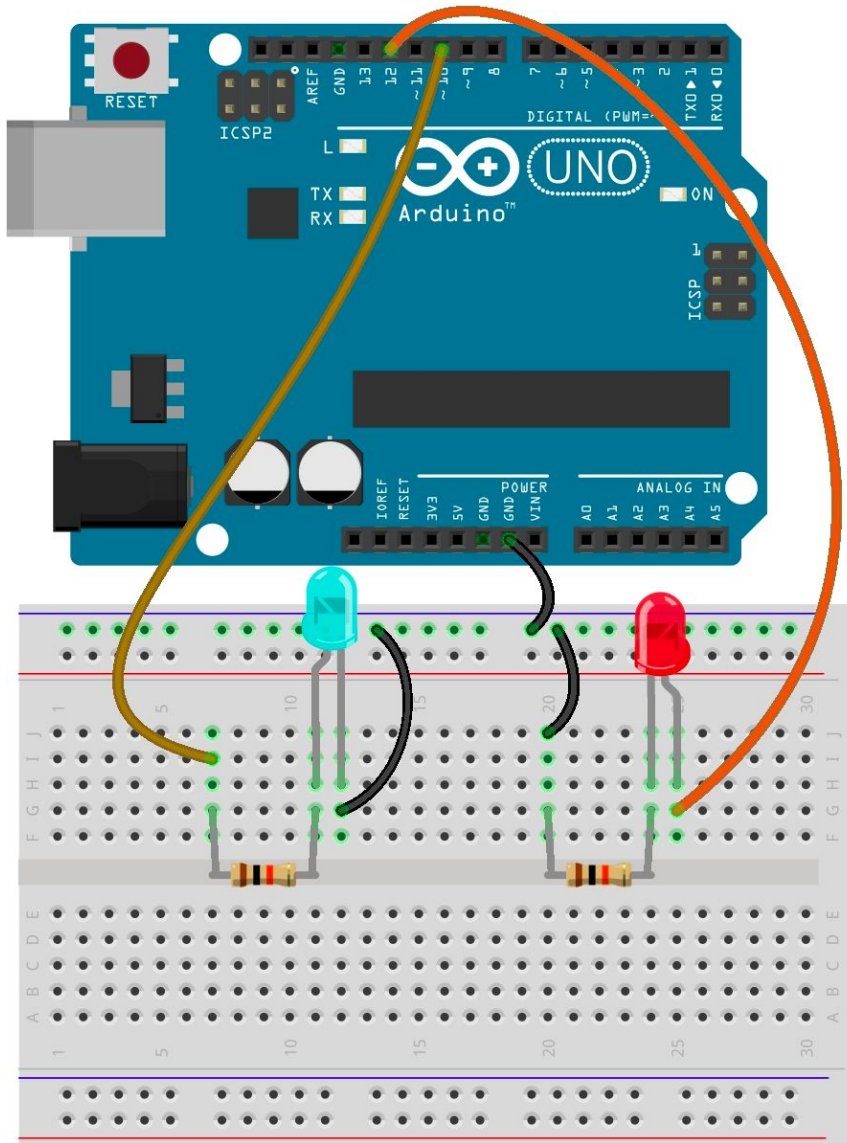
Windowsissa Port-valikossa tulee lukea COM3 tai muu usb-porttia määrittävä luku.

Macissa lukee /dev/tty.usbserial....



Näkymä kun usb-kaapeli on jäänyt kytkemättä

Kaksi lediä ja koodin muokkausta



fritzing

Sample code:

Rakenna koodi muokkaamalla aiempaa koodia "leikkaamalla ja liimaamalla".

Huomioi mihin pinneihin ledit kiinnitetään ja kirjoita vastaavat määrittelykset koodiin.

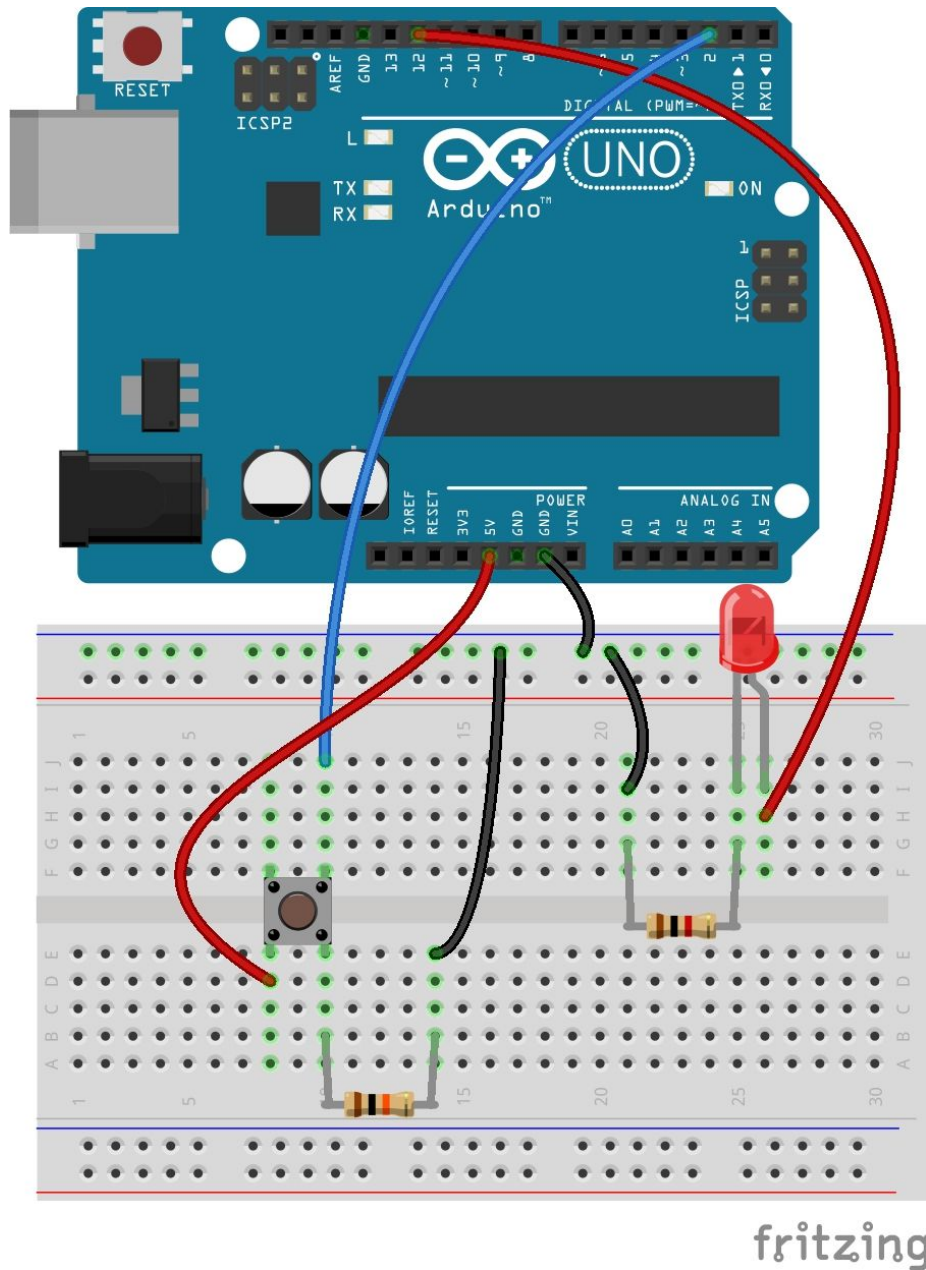
```
void setup() {  
  
  pinMode(12, OUTPUT);  
  pinMode(10, OUTPUT);  
  
}  
  
void loop() {  
  
  digitalWrite(12, HIGH);  
  delay(1000);  
  digitalWrite(12, LOW);  
  
  digitalWrite(10, HIGH);  
  delay(1000);  
  digitalWrite(10, LOW);  
  
}
```

Muokkaa koodia nyt eri tavoin. Tallenna eri versiot eri nimillä.

PWM

Huomaa, että pinnin 10 edessä on merkintä PWM. Tämä tarkoittaa Pulse width modulation, pinniä. Pinnit joissa on tämä merkintä, voivat antaa sähkövirtaa päälle/pois todella nopeasti, jolloin vaikutelma on kuin portti olisi analoginen signaalilähde. Kokeile muokata ledi palamaan himmeämmin tekemällä pinnin taajuudesta todella nopea.

Button eli painokytin (digital input)



TÄSSÄ MENOSSA. Korjaa seuraava button koodi. Lisää servo-ohjaus. Lisää buttonilla ohjattava servo. "Taita" vihkoksi. Pakkaa mukaan buttonit, vastukset, ledit, servot, trankut, dc-moottorit (kolvaa johdot).

Salmpe code:

```
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 12; // the number of the LED pin

int buttonState = 0; // variable for reading the pushbutton status, (changes)

void setup() {

  pinMode(ledPin, OUTPUT); // initialize the LED pin as an output:
  pinMode(buttonPin, INPUT); // initialize the pushbutton pin as an input:
}

void loop() {

  buttonState = digitalRead(buttonPin); // read the state of the pushbutton value:

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {

    digitalWrite(ledPin, HIGH); // turn LED on:

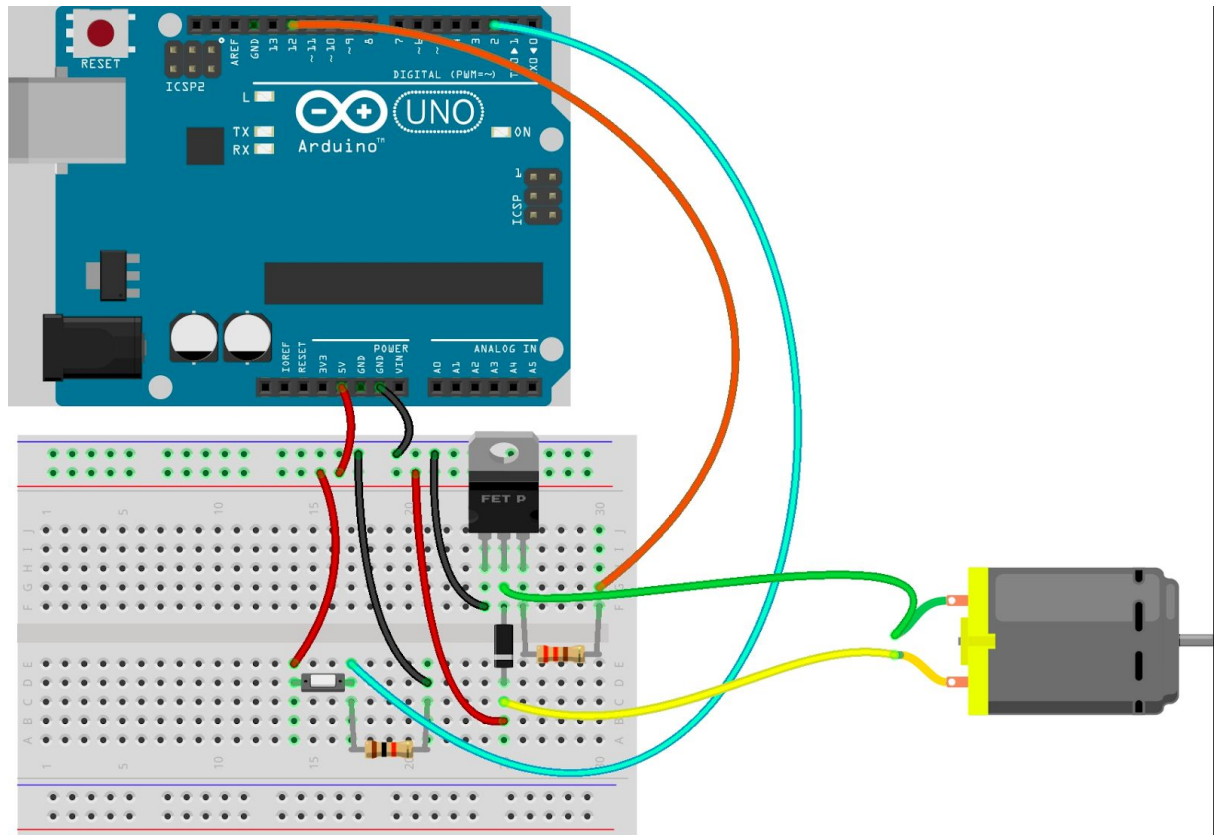
  } else {

    digitalWrite(ledPin, LOW); // turn LED off:

  }
}
```

Kokeile saatko ledin vilkkumaan napin painalluksesta?

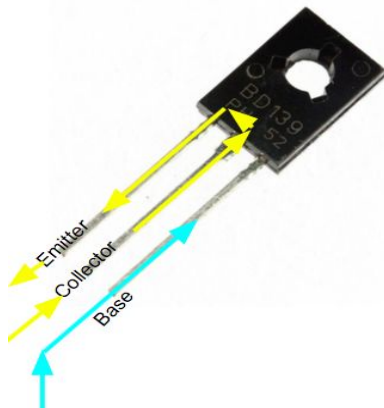
DC-sähkömoottori



fritzing

DC motor

- Takes up to 40mA (millampere), so you need to take power from 5V pin. Make shure the actuator can handle 5V current max!
- Transistor works as a gate/door/electronic switch that can be driven applying a voltage to one of its three pins (Base).
- With motor you need 1N4001 diode to protect Arduino and make shure the electricity flows on right direction.

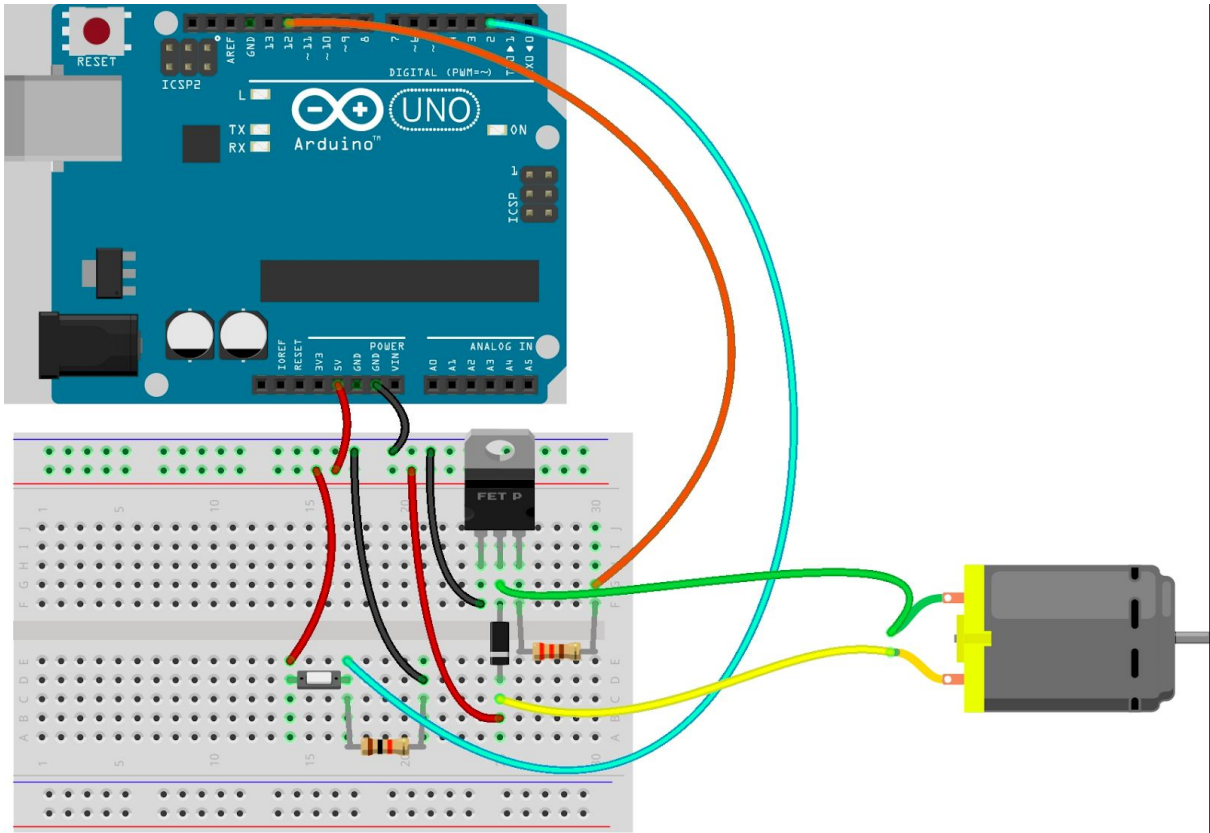


Sample codes:

Motor on/off

```
void setup() {  
  
  pinMode(12, OUTPUT);  
  
}  
  
void loop() {  
  
  digitalWrite(12, HIGH);  
  delay(1000);  
  digitalWrite(12, LOW);  
  delay(1000);  
  
}
```

Kuten huomaat, koodi on sama kuin ledin kanssa. Erona kytkennöissä on se, että nyt ohjaat led-pinnillä ohjausvirtaa transistorille. Transistorin läpi kulkee suurempi virta ja Arduinon pinnit säilyvät ehjinä.



fritzing

Motor on/off with button

```
const int buttonPin = 2; // the number of the pushbutton pin
const int motorPin = 12; // the number of the Motor pin

int buttonState = 0; // variable for reading the pushbutton status

void setup() {

  pinMode(motorPin, OUTPUT); // initialize the Motor pin as an output:
  pinMode(buttonPin, INPUT); // initialize the pushbutton pin as an input:
}

void loop() {

  buttonState = digitalRead(buttonPin); // read the state of the pushbutton value:

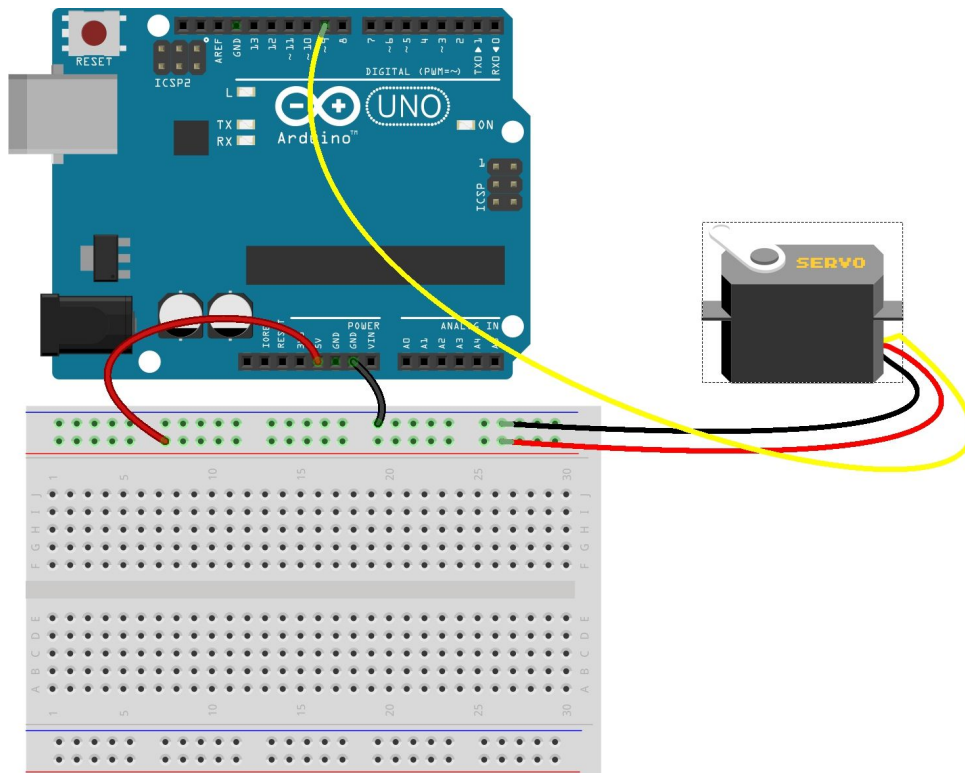
  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {

    digitalWrite(motorPin, HIGH); // turn Motor on:
    delay(1000);
    digitalWrite(motorPin, LOW); //turn Motor off after 1 sec

  } else {

    digitalWrite(motorPin, LOW); // keep Motor off
  }
}
```

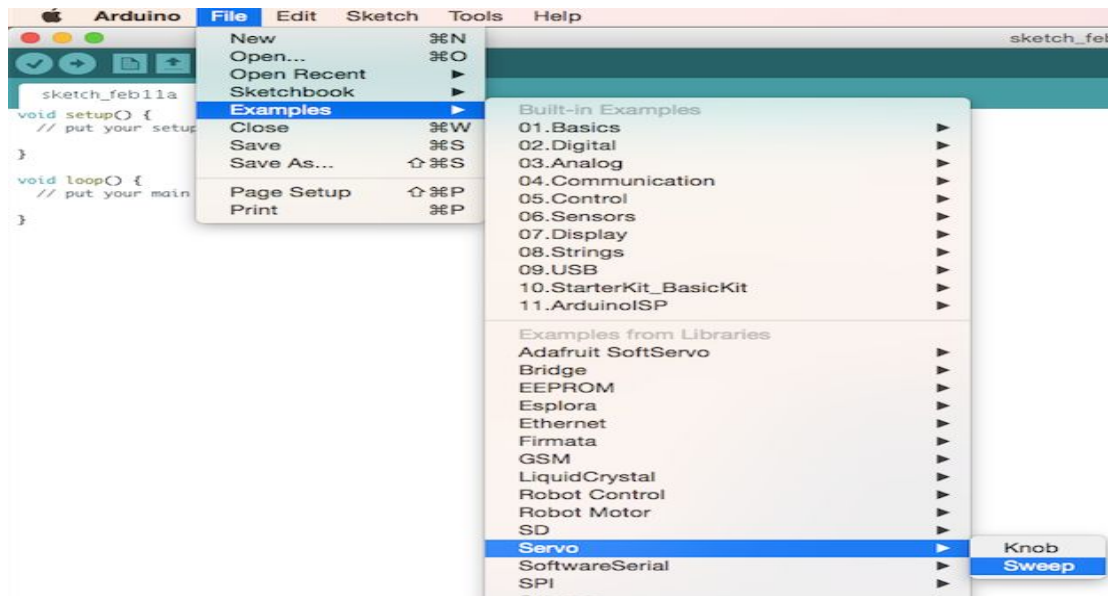
Servomoottori



fritzing

Servomoottori on valmis sähkömoottorityyppi, jonka asennon pystyy asettamaan pysymään missä tahansa 0-180 asteen välillä. On olemassa myös 360-asteen servomoottoreita, mutta tässä ohjeessa keskitymme 180-asteiseen servoon.

Saat servon liikkumaan helpommin suoraan Arduinin IDE:n sisältö löytyvästä koodista Sweep.



```
#include <Servo.h>
```

```
Servo myservo; // create servo object to control a servo  
// twelve servo objects can be created on most boards
```

```
int pos = 0; // variable to store the servo position
```

```
void setup() {
```

```
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
```

```
}
```

```
void loop() {
```

```
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees  
    // in steps of 1 degree
```

```
    myservo.write(pos); // tell servo to go to position in variable 'pos'
```

```
    delay(15); // waits 15ms for the servo to reach the position
```

```
  }
```

```
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
```

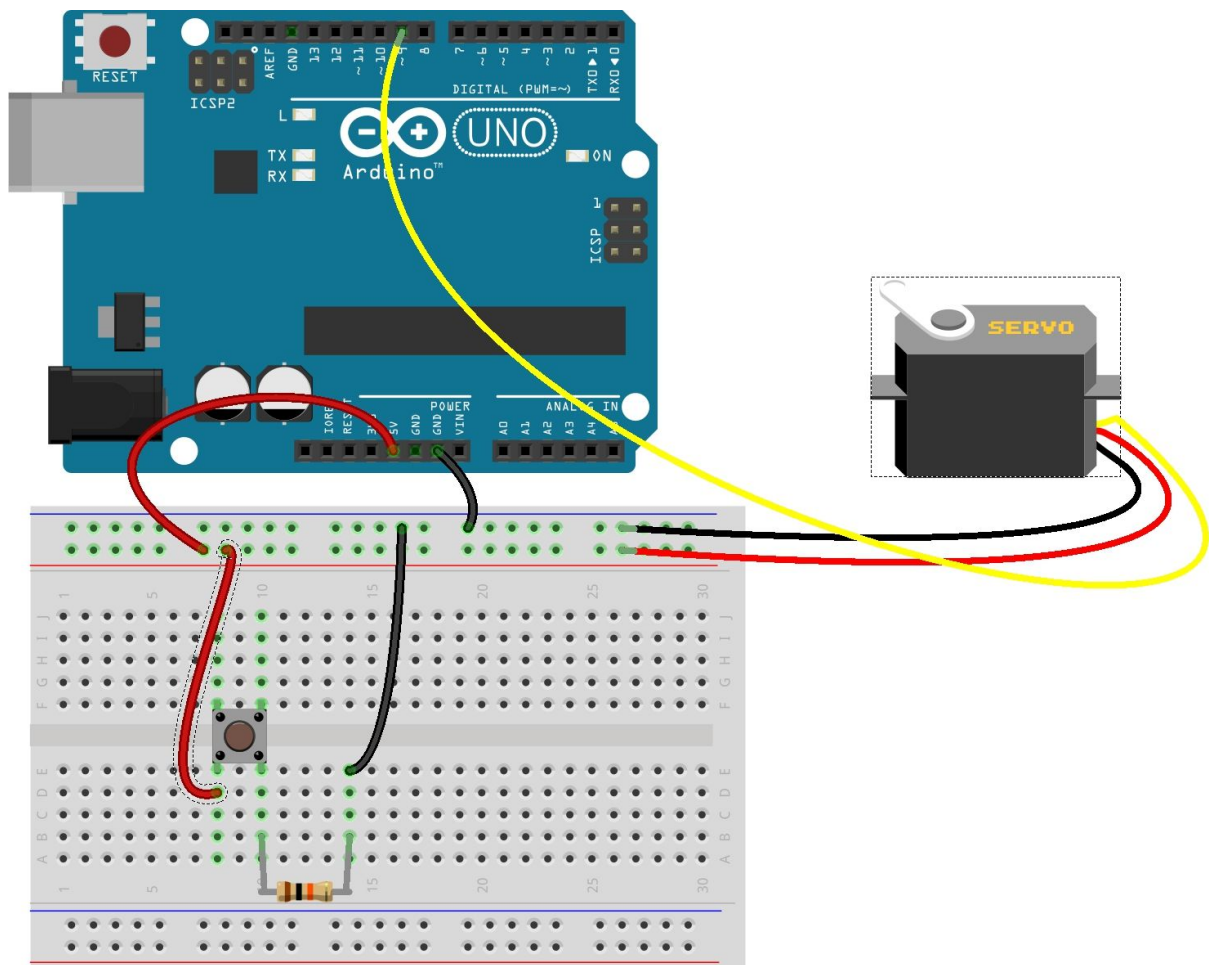
```
    myservo.write(pos); // tell servo to go to position in variable 'pos'
```

```
    delay(15); // waits 15ms for the servo to reach the position
```

```
  }
```

```
}
```

Button ja servomoottori



fritzing

Koeta rakentaa koodi itse, leikkaamalla ja liimaamalla sopivat osat Button, eli painokytkin-ohjeista. Vaihda ledin tilalle servon ohjauksen.

```

#include <Servo.h>

const int buttonPin = 2; // the number of the pushbutton pin ADDED

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0; // variable to store the servo position
int buttonState = 0; // variable for reading the pushbutton status ADDED

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object

  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT); //ADDED
}

void loop() {

  // read the state of the pushbutton value: ADDED
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn servo on:

    for (pos = 180; pos >= 120; pos -= 1) { // goes from 180 degrees to 0 degrees
      myservo.write(pos); // tell servo to go to position in variable 'pos'
      delay(10); // waits 10ms for the servo to reach the position
    }

    delay(10);

    for (pos = 120; pos <= 180; pos += 1) { // goes from 0 degrees to 180
degrees
      // in steps of 1 degree
      myservo.write(pos); // tell servo to go to position in variable 'pos'
      delay(10); // waits 10ms for the servo to reach the position

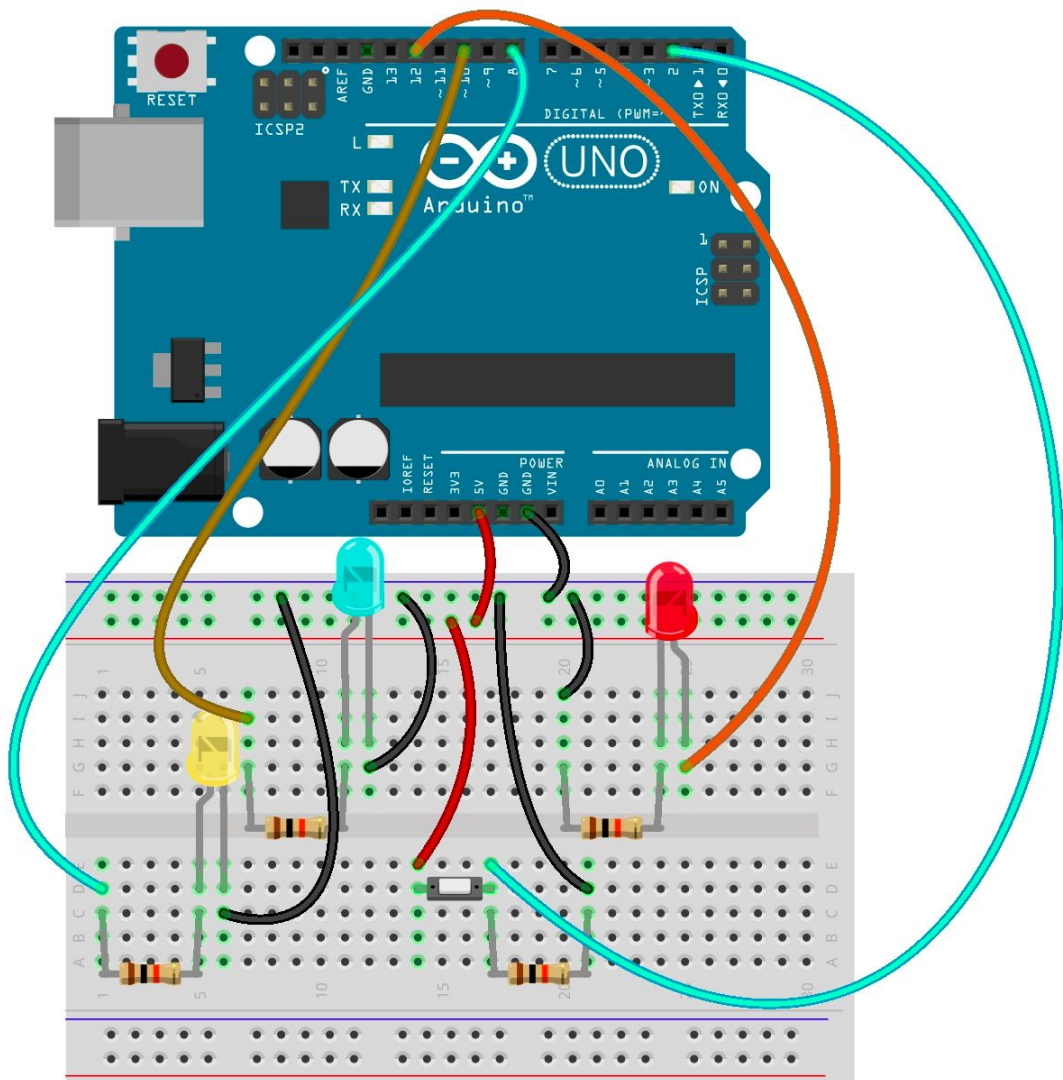
    }

  }

}
}

```

Rock-Scissor-Paper / kolikonheittolaite kolmella ledillä.



Sample code:

```
int randomNumber;
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin1 = 12; // the number of the LED pin, each ledpin needs own integer
const int ledPin2 = 10;
const int ledPin3 = 8;
int buttonState = 0; // variable for reading the pushbutton status

void setup(){
  Serial.begin(9600); // for communicating with Serial monitor (not mandatory)
  pinMode(ledPin1, OUTPUT); // initialize the LED pin as an output
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(buttonPin, INPUT); // initialize the pushbutton pin as an input
}

void loop() {

  buttonState = digitalRead(buttonPin); // read the state of the pushbutton value:

  if (buttonState == HIGH) { // check if the pushbutton is pressed. If it is, the buttonState is HIGH

//give random number
    randomNumber = random(1,4); //random numbers scale needs to be 1 more

    if (randomNumber == 1)
    {
      digitalWrite(ledPin1, HIGH);
      Serial.println("rock");
    }

    if (randomNumber == 2)
    {
      digitalWrite(ledPin2, HIGH);
      Serial.println("scissor");
    }

    if (randomNumber == 3)
    {
      digitalWrite(ledPin3, HIGH);
      Serial.println("paper");
    }

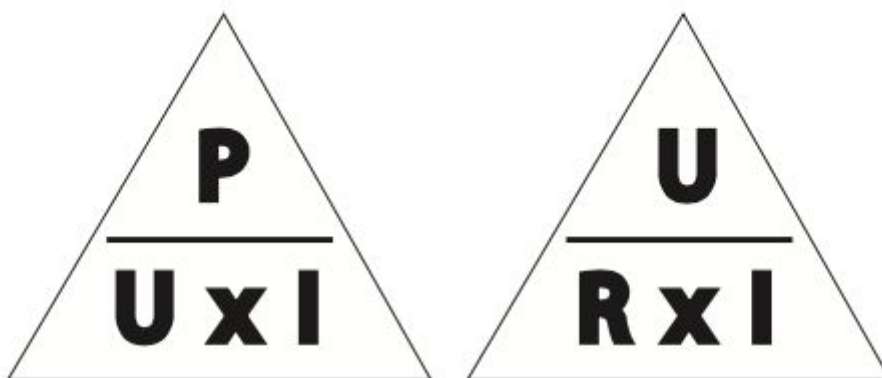
    }

  delay(1000);
  digitalWrite(ledPin1, LOW); //turn off all the leds
  digitalWrite(ledPin2, LOW);
  digitalWrite(ledPin3, LOW);
  }
}
```

LED ja sähkövirran rajoittaminen etuvastuksella.

Tämä on tärkeää, jotta voit käyttää ledejä turvallisesti erilaisilla ja erijännitteisillä paristoilla! Samalla saat ledien toiminta-aikaa pidemmäksi.

Ledin etuvastuksen laskukaava perustuu Ohmin lakiin: $P = U \times I$ tai $U = R \times I$



Saat halutun mittayksikön selville, kun tiedät jotkin kaksi mittayksikköä. Peitä selvittävä yksikkö ja saat selville pitääkö sinun jakaa vai kertoa. Koska mahdollisesti selvittäviä yksiköitä on neljä, tarvitsee sinun valita aina sopiva muistikolmio kahdesta.

$P = \text{teho} = \text{watti (W)}$
 $U = \text{jännite} = \text{voltti (V)}$
 $I = \text{virta} = \text{ampeeri (A)}$
 $R = \text{resistanssi} = \text{ohmi } (\Omega)$

Perusledien virta, I (ampeeri), on yleensä 20 - 30 mA. Voit yleensä laskea virran ledin kohdalla 20 mA:lla, eli 0,02A.

Seuraavaksi pitäisi selvittää, mikä on ledin kynnysjännite, eli pienin jännite ledin navoilla, jolla led syttyy täysin kirkkaaksi. Seuraava lista pitää yleensä melko hyvin paikkansa:

Punainen 1.7 V
Superkirkas punainen 2.2V

Vihreä ja keltainen 2.1V
Superkirkas sininen 4.5V

(Voit testata arvoja hyvin esim 3V kolikkoparistolla, jolloin saat helposti selville syttyykö led 3V jän- nitteellä. Huom. Kolikkopariston sisäinen resistanssi on niin suuri, ettei sen kanssa tarvita etuvastusta. Ts. voit laskea kynnyksjännitteeksi usein 3V.)

Aloita laskemalla vastuksen navalla vaikuttava jännite. Tämä lasketaan vähentämällä syöttöjännit- teestä (eli virtalähteen jännite, esim. 9V paristo) ledin kynnyksjännite (esim. 1,7V, jos punainen led).

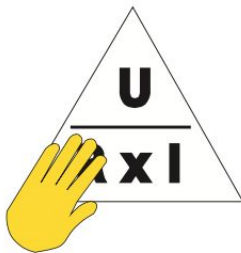
$$9V - 1,7V = 7,3V$$

Laskeaksemme vastuksen, eli resistanssin, eli ohmit, asetetaan saatu luku laskukaavaan: $U = R \times I$, jolloin $R = U / I$

$$R = 7,3V / 0,02A \quad R = \mathbf{365 \text{ ohmia}}$$

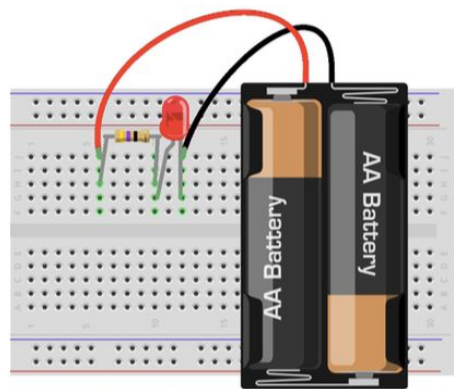
Tällöin etsitään seuraava isompi sopiva vastus. Tässä tapauksessa esim. **390 ohminen**, 0,25W hiilikalvovastus. (Hiilikalvovastus on yleisin "perusvastus".)

Ja kaiken tämän voit lopuksi varmistaa tai laskea suoraan netistä löytyvillä laskureilla. Esimerkiksi www.ledcalc.com



Laskukaava mene siis näin:

1. Jos lasket vastusta, peitä muistikolmion R eli Resistanssi, eli vastus.
2. Laske virtalähteen ja ledin kynnyksjännitteen välinen erotus. ($3V - 2.2V = 0.8V$)
3. Sijoita luvut muistikolmioon. ($0,8V / 0,02 A = 40$ ohmia.)
4. Etsi lähin tulosta vastaava seuraavaksi suurin vastus. (Tässä tapauksessa 47 ohmia).



Punaisella ledillä etuvastuksena 47 ohminen vastus (huomaa väritys).