



# IFI6203 ROBOTIKA

Praktikum 1



# Aine tutvustus

- Õppejõud: Tanel Toova
  - A-432
  - [Tanel.Toova@tlu.ee](mailto:Tanel.Toova@tlu.ee)
- Aine materjalid
  - <http://www.cs.tlu.ee/IF16203>
  - *Tutvume kursuseprogrammiga*
- tinkercad.com keskkond
  - <https://www.tinkercad.com/classrooms/af3tLEDg7Mr>

# Arduino platvorm

- Arduino koduleht: <https://www.arduino.cc/>
- Arduino riistvara: <https://www.arduino.cc/en/Main/Products>
- Arduino tarkvara: <https://www.arduino.cc/en/Main/Software>
- Arduino põhifunktsioonide abimaterjal:  
<https://www.arduino.cc/reference/en/>
- Näiteid Arduinot kasutavatest projektidest:  
<https://create.arduino.cc/projecthub>

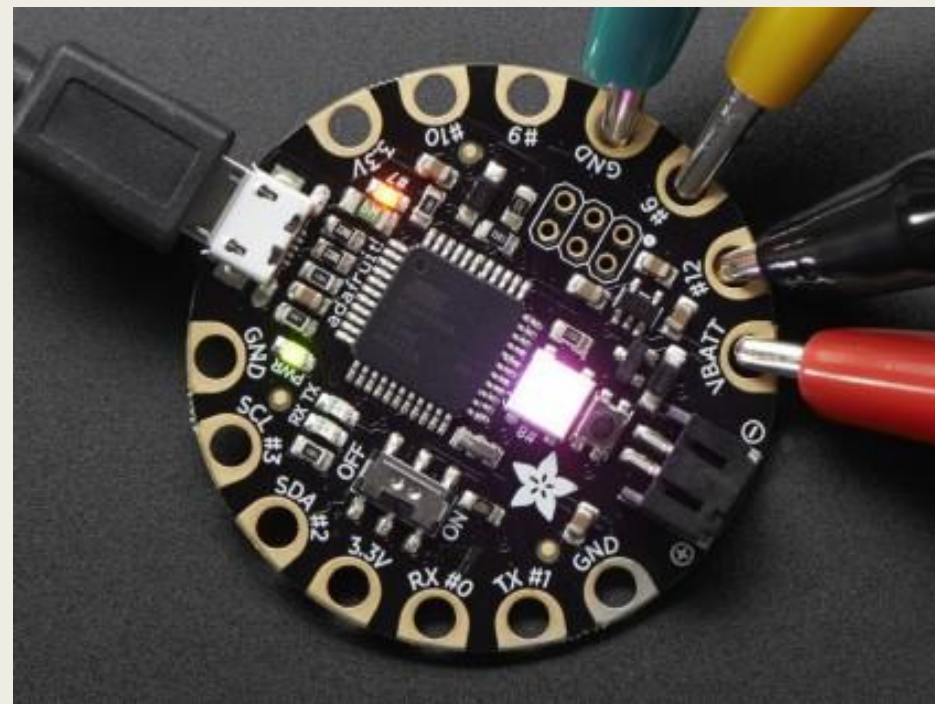
# Arduino Uno

- Programmeeritav riistvarakontroller
- Sobib hästi esmaste suuremate prototüüpide loomiseks.
- Lihtne teha muudatusi.



# Adafruit FLORA

- Programmeeritav riistvarakontroller
- Sobib hästi kantavate, riiete külge kinnitatud lahenduste, loomiseks
- Väike ja kompaktne



# Node MCU

- Programmeeritav riistvarakontroller
- WiFi võimekusega
- Sobib hästi lahendusteks, kus on vaja andmevahetust väliste süsteemidega

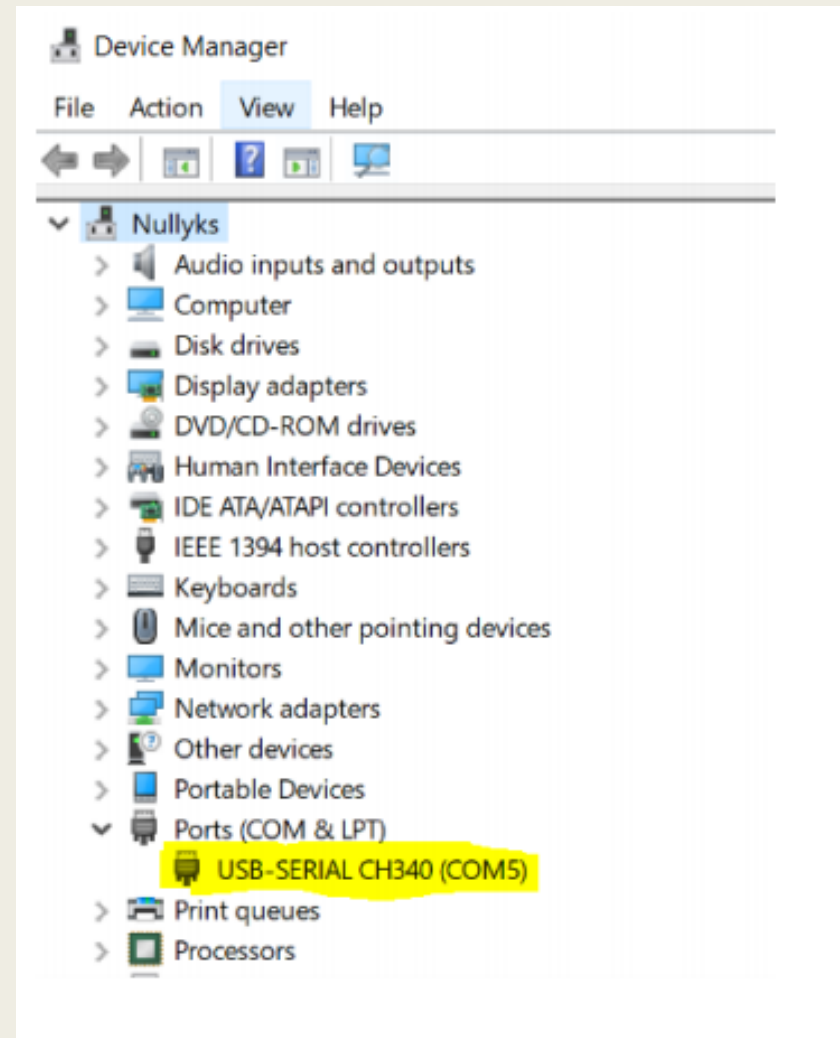


# Arduino IDE paigaldamine

- Lae alla Arduino IDE viimane stabiilne versioon:
  - <https://www.arduino.cc/en/Main/Software>
- CH340 driver:
  - <https://sparks.gogo.co.nz/ch340.html>
- Kui plaanid kasutada mõnda Adafruit'i kontrolleri, siis:
  - järgi seda õpetust (*Super Easy Installation*):  
<https://learn.adafruit.com/adafruit-arduino-ide-setup/arduino-1-dot-6-x-ide>

# Arduino kontrolleriiga ühenduse tekitamine

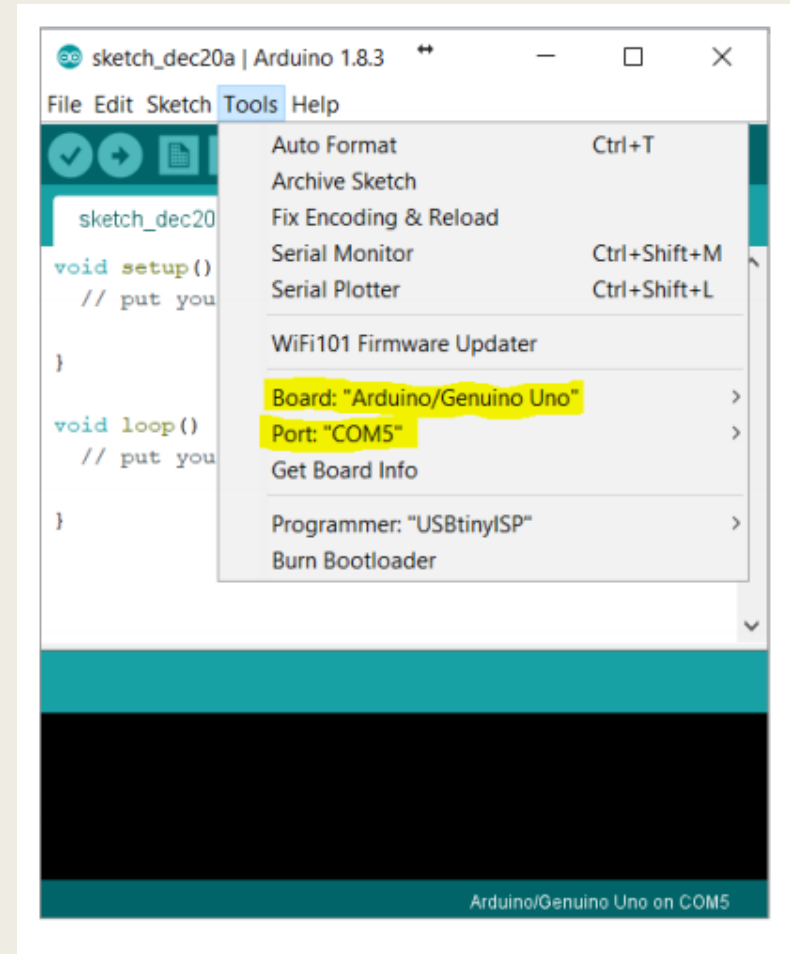
- Ühenda Arduino controller USB kaabli abil arvutiga
- Uuri välja, milline COM port seadmele määrati (vt. pilti)





# Arduino kontrolleriiga ühenduse tekitamine

- Seadista Arduino IDE tarkvaras:
  - *Board* - vali kontrolleri, mida kasutad
  - *Port* - vali COM port, mis sinu kontrolleri jaoks määrati. NB! See võib muutuda iga kord kui USB uuesti ühendad.



# Kontrollime, et kõik töötab

- Avame Arduino IDE rakenduses näite Blink
  - *File -> Examples -> 01.Basics -> Blink*
- Laeme näite oma kontrolleriile vajutades Upload nuppu.
- Veendume, et näite laadimine õnnestus
  - *Kontrolleril olev LED vilgub sagedusega 1 kord sekundis*

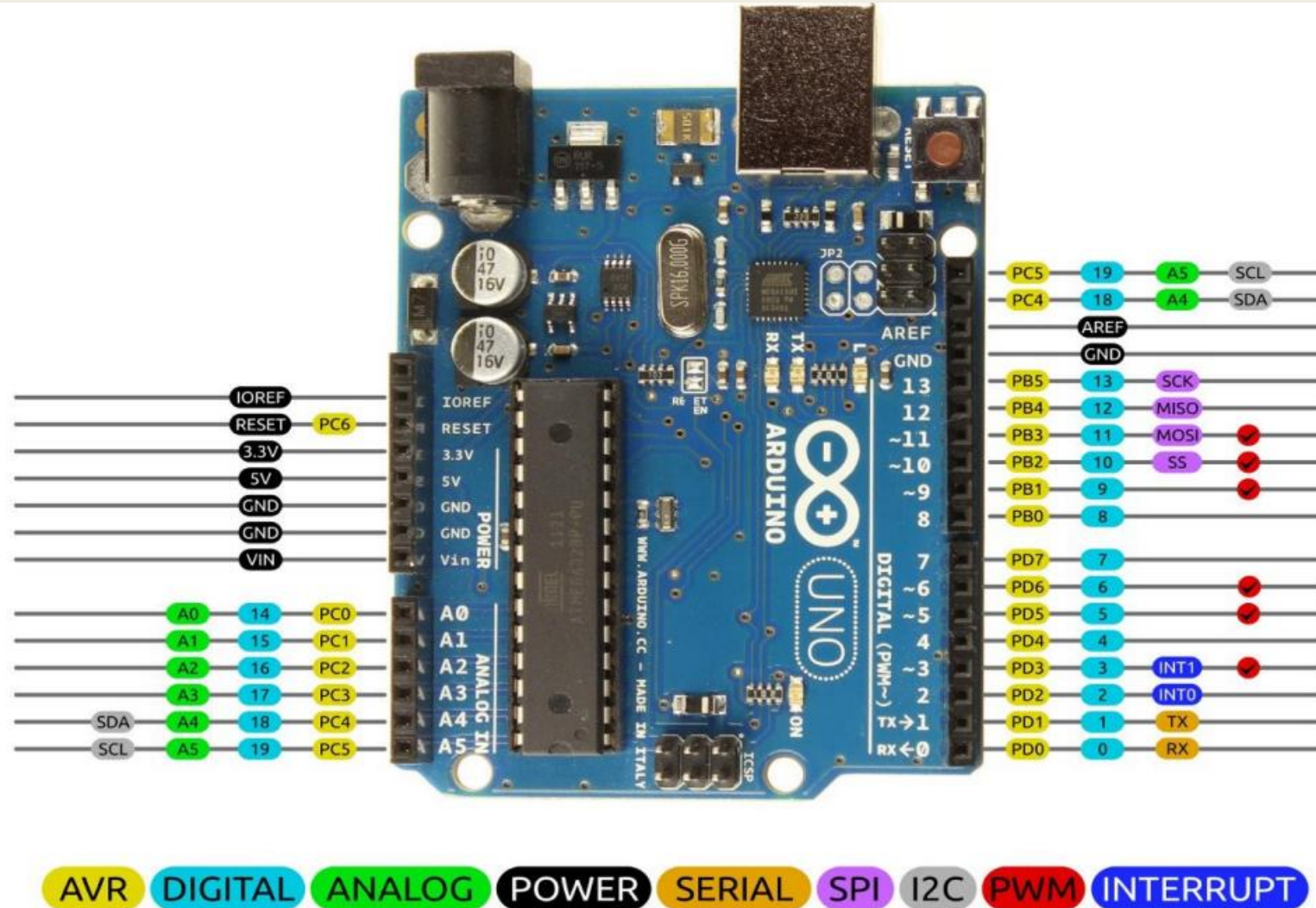
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

```
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

# Vaatame üle Arduino programmi olulisemad osad

- `setup ()` - see funktsioon käivitub kontrolleri käivitamisel ja jookseb ühe korra. Nagu nimigi ütleb, on see kasulik andmestruktuuride ja sensorite/seadmete algväärtuste seadistamiseks ning tööks valmispanekuks.
- `Loop ()` - see funktsioon käivitub peale `setup ()` funktsiooni ja jookseb lõpmatult kuni kontrolleri väljalülitamiseni või katkestussignaali saamiseni. Enamasti kirjutatakse kogu põhiprogramm siia sisse.
- Arduino programmeerimiskeele abimaterjal on kättesaadav siit:
  - <https://www.arduino.cc/reference/en/>

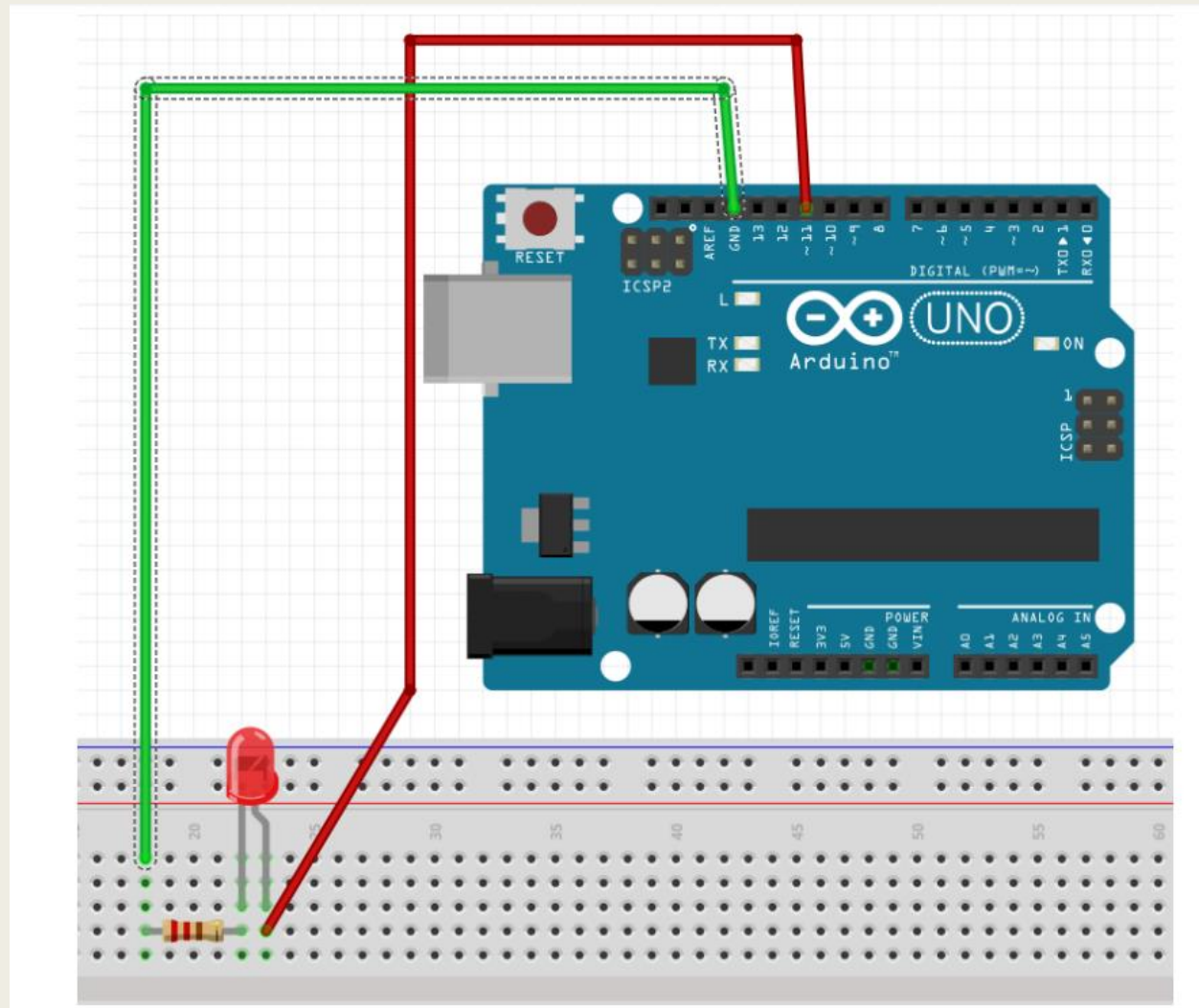
# Arduino IDE väljundviigud (ehk, mida ja kuhu ühendada)



# Olulisimad viigud

- **Power** (Toide) - püsiva pingega (3,3V või 5V) toide sensoritele ja seadmetele. GND on maadus.
- **Analog** (analoogsisend) - analoogsignaali (täpse volupinge) lugemiseks. Kümne-bitine analoog-digitaal konverter muundab pinge vahemikus 0 - 5V täisarvuks vahemikus 0 - 1023
- **Digital** (digitaalne sisend/väljund) - sisendrežiimis võimaldab lugeda kahendsignaali 5V=HIGH, 0V=LOW. Väljundrežiimis võib olla kontrollitav toiteallikas (või signaallikas) volutugevusega kuni 40 mA.
- **PWM** (pulse width modulation) - seda tüüpi digitaalsed viigud võimaldavad simuleerida erinevaid väljundpingeid vahemikus 0 - 5V

# Ehitame ise vilkuva LED-iga seadme



# LEDi vilgutav kood

```
int LEDpin = 11;
int time = 200;

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

void loop()
{
  digitalWrite(LEDpin, HIGH);
  delay(time);
  digitalWrite(LEDpin, LOW);
  delay(time);
}
```

# Muudame LEDi heledust kasutades PWM võimekusega viiku

- Eesmärgiks on panna LED vilkumise asemel graduaalselt helendama ja tuhmuma.
- Kuna me juba kasutame PWM võimekusega viiku, siis on füüsilist seadet me muutma ei pea.
- Koodis on oluliseks lahenduseks **analogWrite()** funktsiooni kasutamine koos **for()** tsükliga.



# LEDi heledust muutev kood (1)

```
int LEDpin = 11;
int time = 50;
int steps = 20;

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

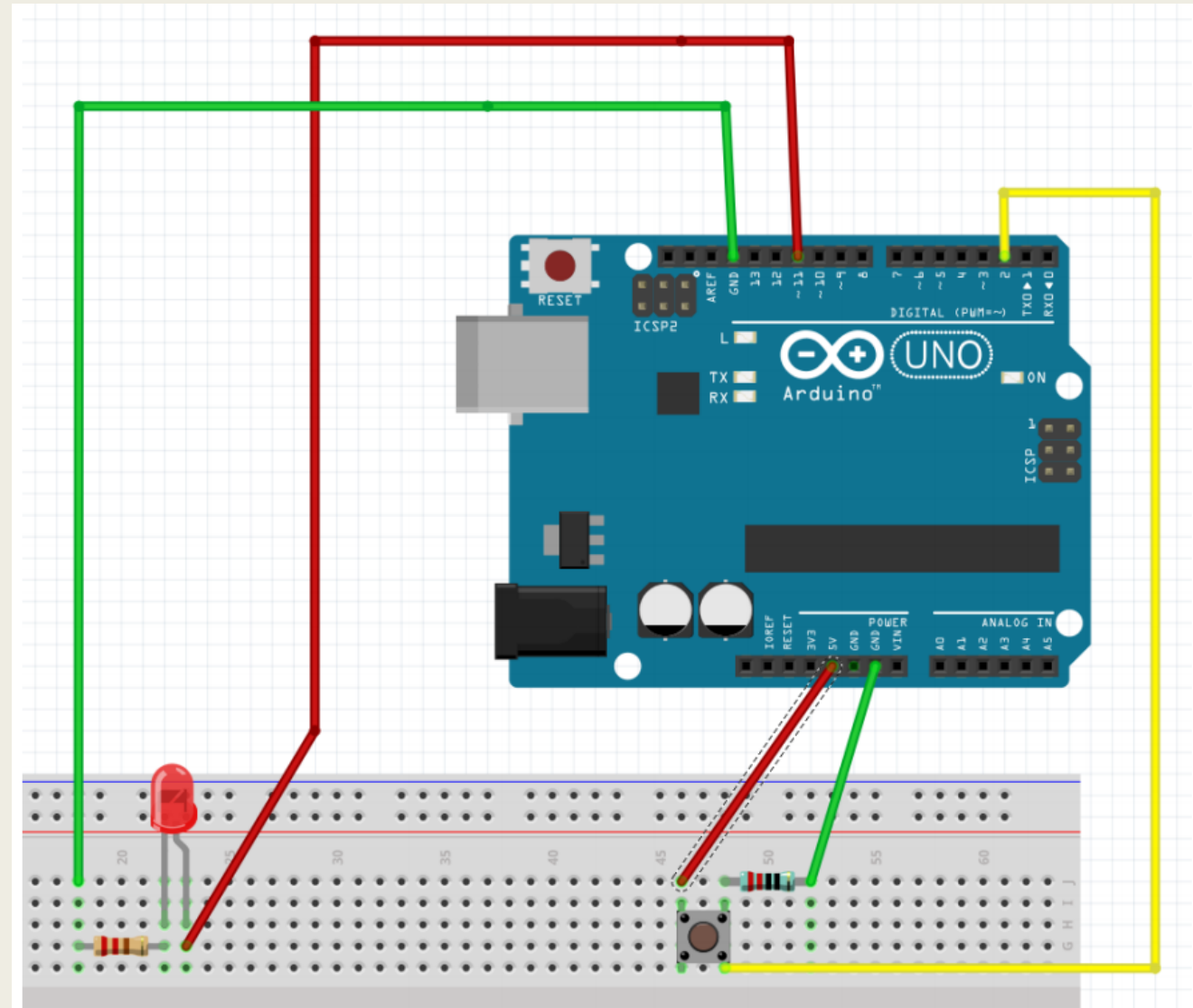
void loop()
{
  fadeIn(steps);
  fadeOut(steps);
}
```

# LEDi heledust muutev kood (2)

```
void fadeIn(int steps){
  int brightness=0;
  analogWrite(LEDpin, brightness);
  for(int i=0; i<=steps; i++){
    brightness=i*int(255/steps);
    analogWrite(LEDpin, brightness);
    delay(time);
  }
}
```

```
void fadeOut(int steps){
  int brightness=255;
  analogWrite(LEDpin, brightness);
  for(int i=steps; i>=0; i--){
    brightness=i*int(255/steps);
    analogWrite(LEDpin, brightness);
    delay(time);
  }
}
```

# Lisame oma seadmele nupu



# Muudame oma koodi, et see loeks nupuvajutusi (lisandunud kood on sinine)

```
int LEDpin = 11;
int buttonPin = 2;
int time = 50;
int steps = 20;

void setup()
{
  pinMode(LEDpin, OUTPUT);
  pinMode(buttonPin, INPUT);
  digitalWrite(LEDpin, LOW);
}

void loop()
{
  int buttonState=digitalRead(buttonPin);
  if(buttonState == HIGH ){
    fadeIn(steps);
    fadeOut(steps);
  }
}
```