



# IFI6203 ROBOOTIKA

Praktikum 5



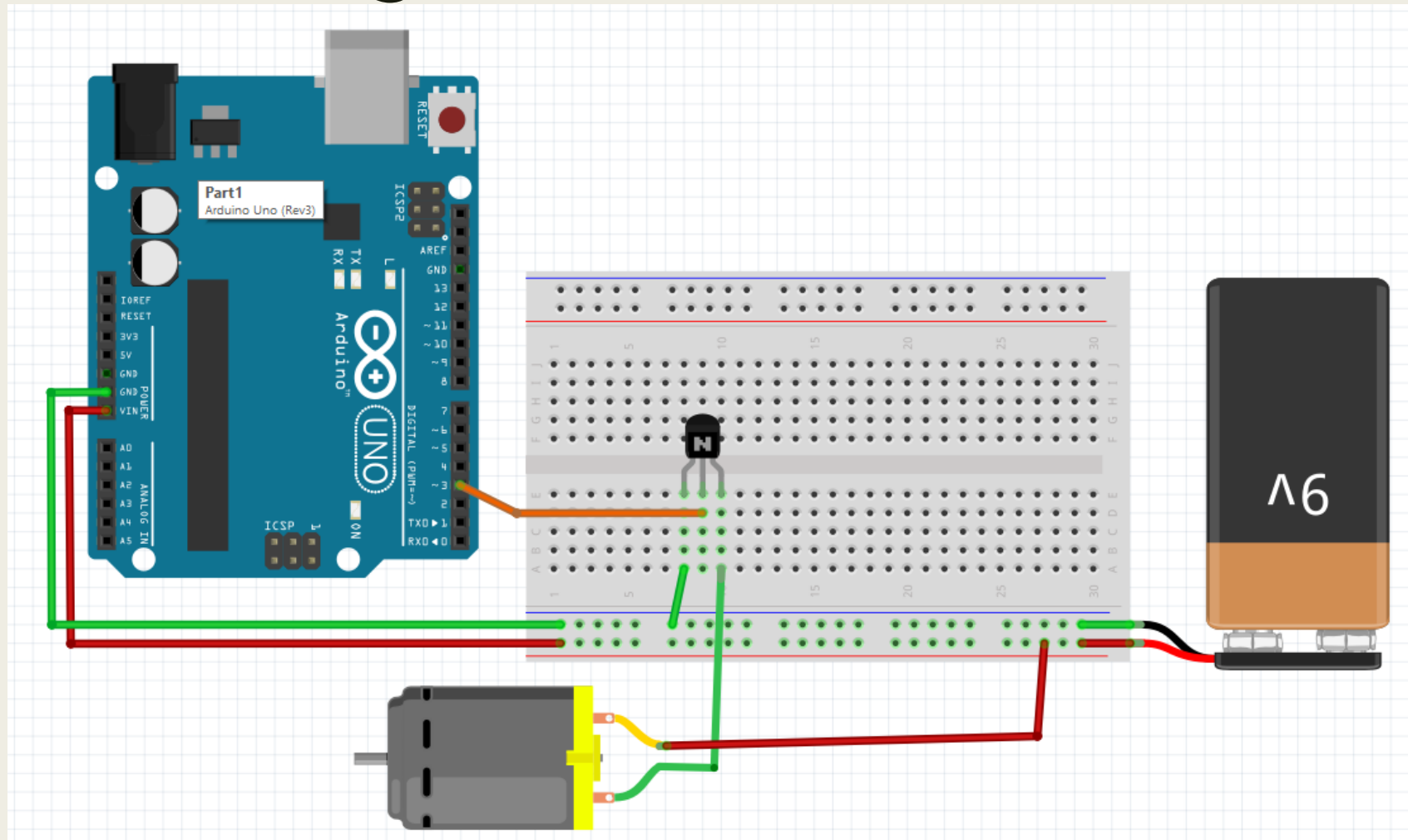
# Alalisvoolumootori kasutamine

- Alalisvoolumootor ei vaja oma tööks mingit erilist juhtloogikat
- Mootori töö kiiruse määrab talle mõjuv voolupinge (ei tohi ületada lubatud maksimaalset pinget).
- Mootori pöörlemise suuna määrab mootorit läbiva voolu suund.
- Alalisvoolumootorid vajavad tihti lisatoiteallikat, sest Arduino kontrolleri ei suuda neile piisava pinge ja/või voolutugevusega toidet pakkuda.

# Alalisvoolumootori juhtimine NPN transistoriga

- Kasutame NPN transistorit lülitina, mis ühendab mootori maandusega
- Transistori baasile antava pingega saame kontrollida mootorit läbivat voolu ning seega mootori töökiirust.
- Me ei saa muuta mootori töösuunda.

# Alalisvoolumootori juhtimine NPN transistoriga



# NPN transistoriga juhtimise näitekode

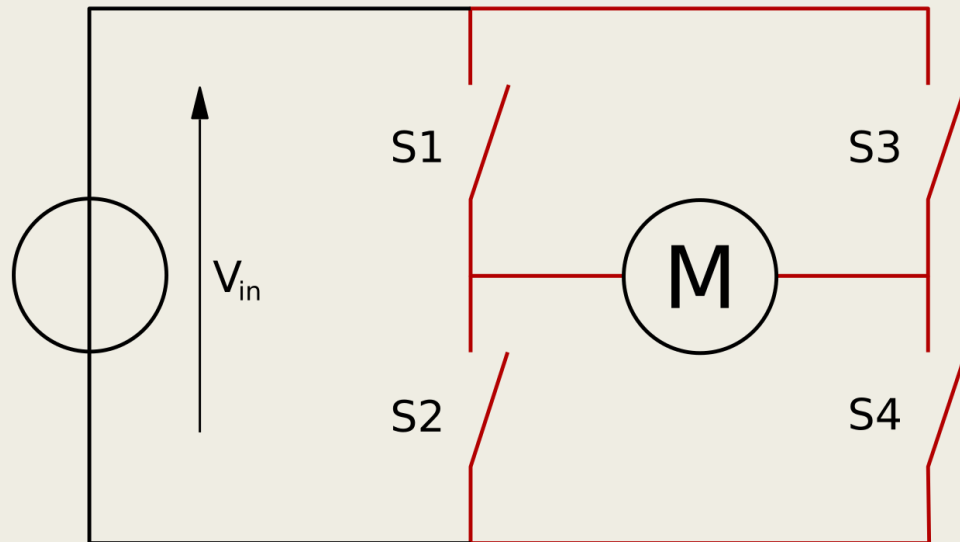
```
#define base 3 //transistori baas

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

void loop() {
  analogWrite(base, 0); //mootor ei tööta
  delay(3000);
  analogWrite(base, 50); //mootor töötab aeglaselt
  delay(3000);
  analogWrite(base, 100); //mootor töötab kiiremini
  delay(3000);
  analogWrite(base, 200); //mootor töötab veel kiiremini
  delay(3000);
}
```

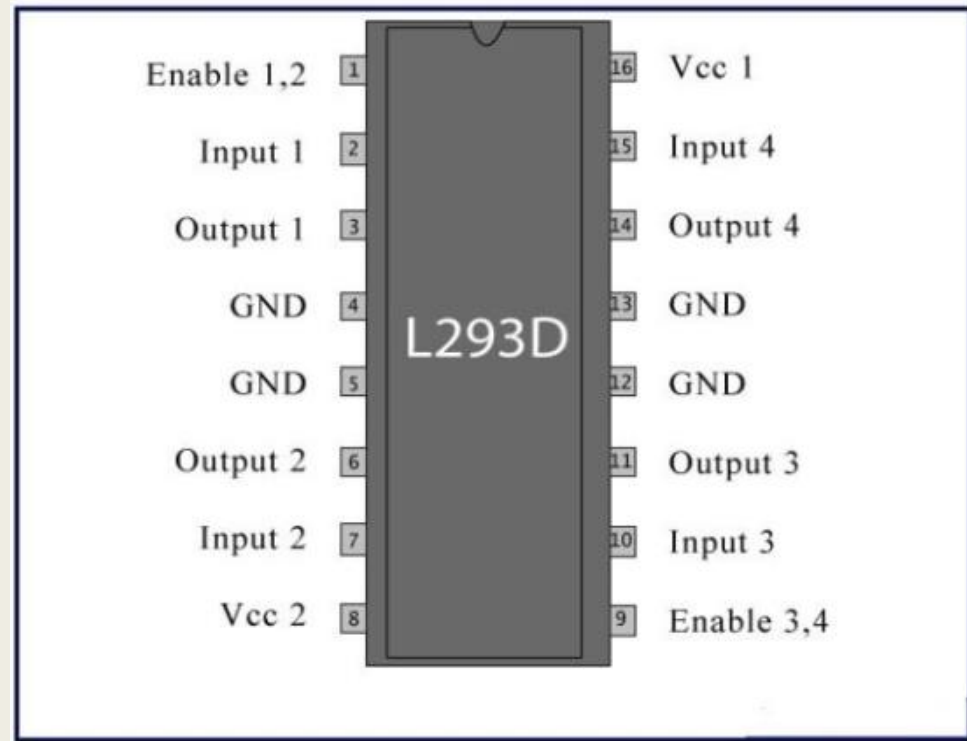
# H-sild

- H-sild on elektroonikalülitus, mis võimaldab seadmele rakendada pinget mõlemas suunas.
- Enamasti kasutatakse alalisvoolumootorite töötamissuuna muutmiseks.

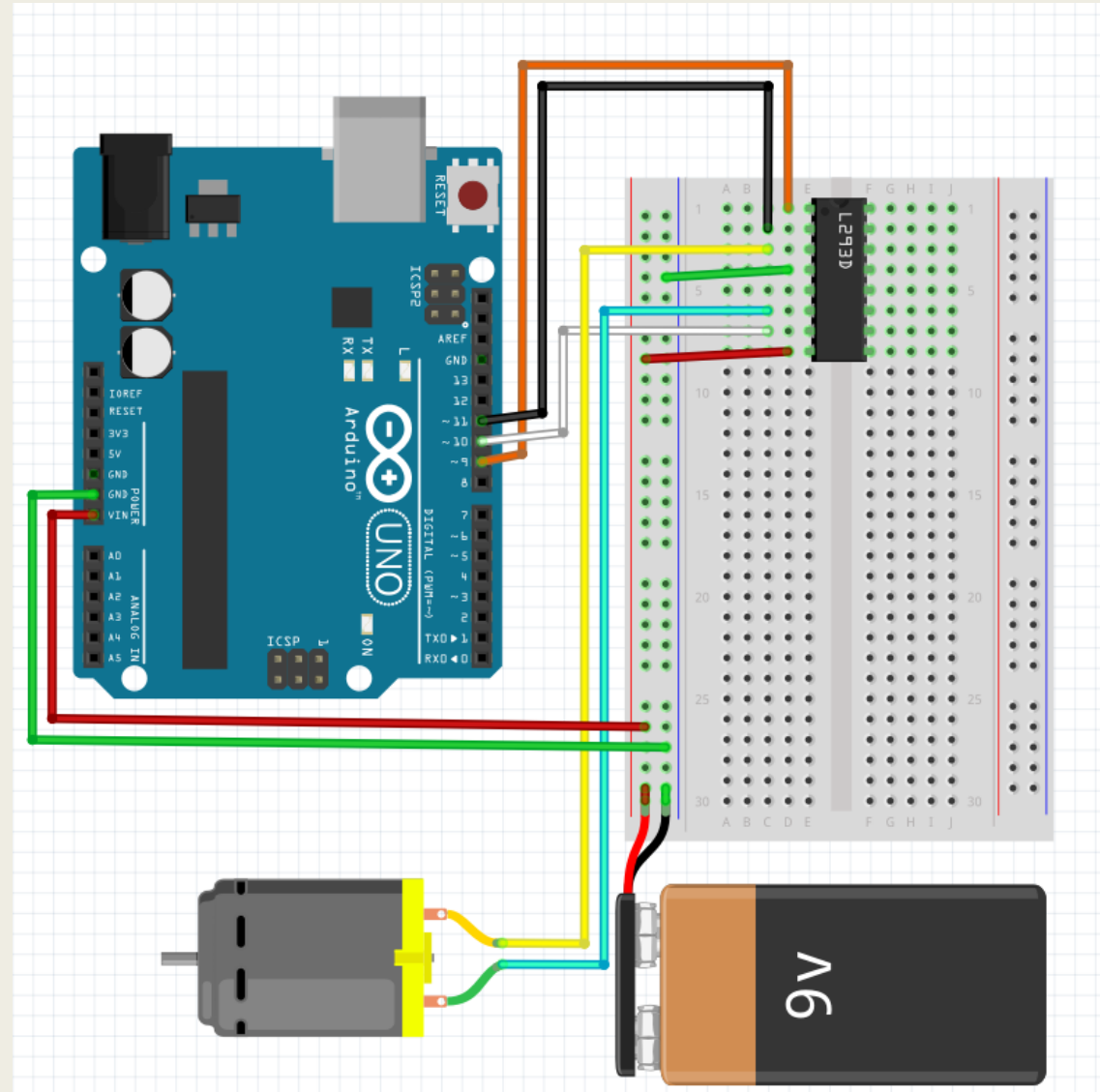


S1	S2	S3	S4	Tulemus
1	0	0	1	Mootor pöörleb paremale
0	1	1	0	Mootor pöörleb vasakule
0	0	0	0	Mootor veereb vabalt
0	1	0	1	Mootor pidurdab paremale
1	0	1	0	Mootor pidurdab vasakule
1	1	0	0	Lühis
0	0	1	1	Lühis
1	1	1	1	Lühis

# L293D kiip annab meile 2 H-silda



# Alalisvoolumootori juhtimine L293D kiibiga





# L293D näitekood

```
#define enablePin 9 //kontrollib mootori töökiirust limiteerides pinget.
#define controlPin1 11 // H-silla lülitus (HIGH või LOW)
#define controlPin2 10 // H-silla lülitus (HIGH või LOW)

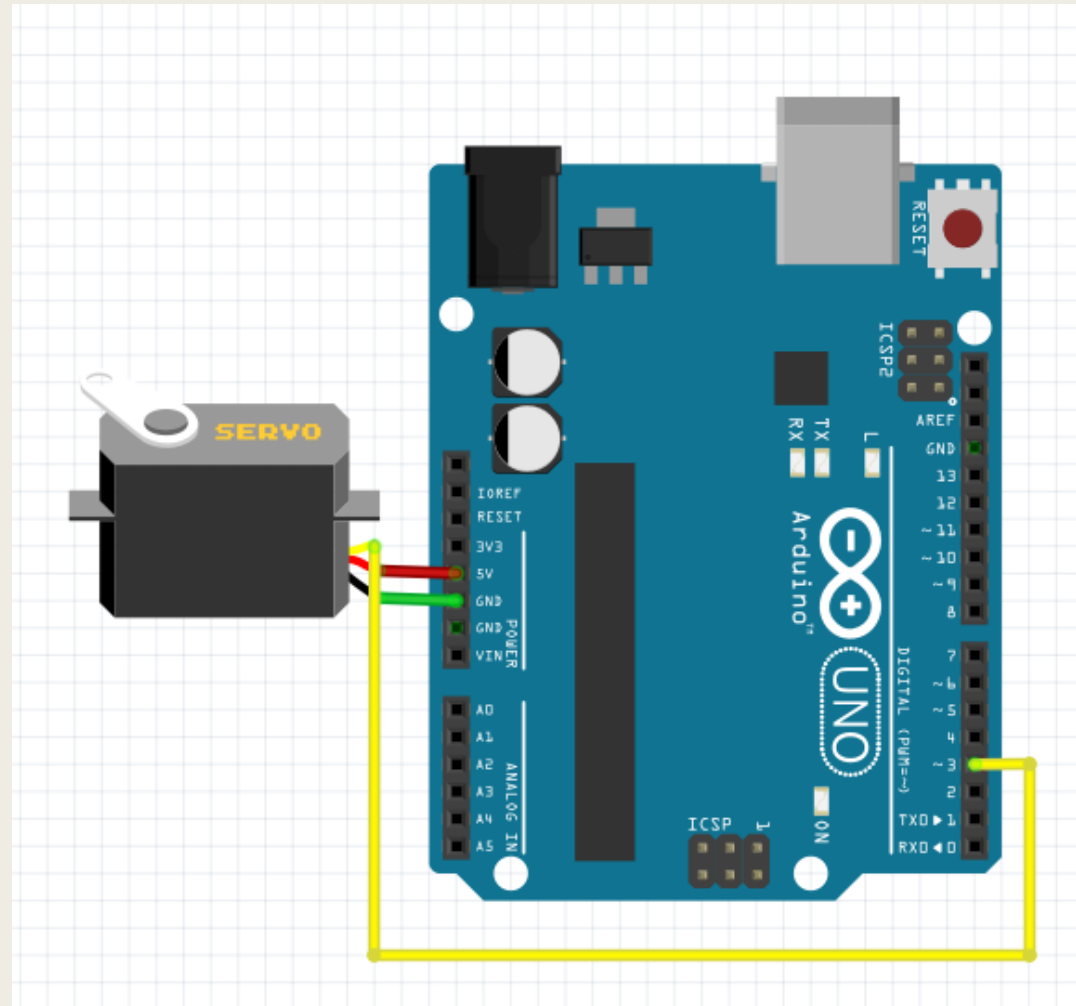
void setup() {
  pinMode(enablePin, OUTPUT);
  pinMode(controlPin1, OUTPUT);
  pinMode(controlPin2, OUTPUT);
}

void loop() {
  digitalWrite(controlPin1,HIGH); //määrame mootori töösuuna
  digitalWrite(controlPin2,LOW); //määrame mootori töösuuna
  analogWrite(enablePin, 50); //Paneme mootori tööle (võrdlemisi aeglaselt)
  delay(3000);
  digitalWrite(controlPin1,LOW); //muudame mootori töösuuna vastupidiseks
  digitalWrite(controlPin2,HIGH);//muudame mootori töösuuna vastupidiseks
  analogWrite(enablePin, 255); //Paneme mootori tööle (täie kiirusega)
  delay(3000);
}
```

# Servomootorid

- Servomootor e. täiturmootor on juhtimiseks kasutatav mootor, mis muudab temaga käitavate mehhanismide asendit vastavalt etteantud seadesignaale
- olulisimad näitajad on
  - *Pöördnurk* - mitu kraadi mootor suudab oma telge pöörata
  - *Nurkkiirus* - kui kiiresti mootor ennast pöörab
  - *Toitepinge* - millise pingega toidet on mootori tööks tarvis
  - *Pöördemoment* - kui palju jõudu suudab mootor rakendada
- Servomootori juhtimiseks on tarvilik riistvaralise servomootori draiver, kuid Arduino puhul võime me selle funktsionaalsust simuleerida kasutades sobivat teeki.

# KY66 servomootori näide (pöördenuk 120 kraadi)



# KY66 näitekood

```
#include <Servo.h> //kasutame Arduino Servo teeki

Servo servo; // deklareerime servo objekti
int minAngle = 0; // servo minimaalne nurk
int maxAngle = 120; //servo maksimaalne nurk

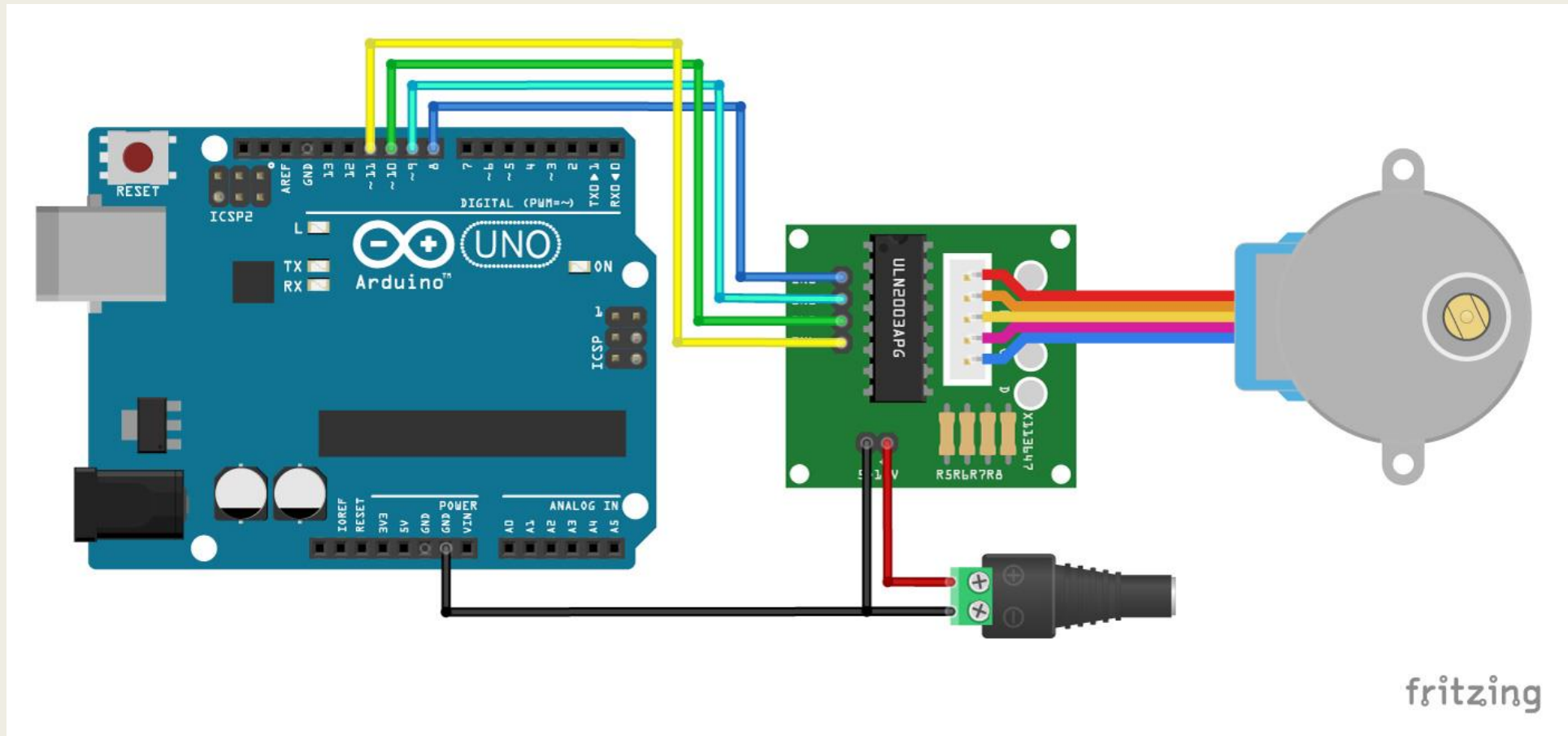
void setup() {
  servo.attach(3); //määrame servo kontrollviigu (PWM võimelise)
}

void loop() {
  servo.write(minAngle);
  delay(1000);
  servo.write(maxAngle);
  delay(1000);
}
```

# Samm-mootorid

- Samm-mootor on pingeimpulssidega käitav sünkroonmootor. Iga impulsi mõjul pöörduv mootori rootor kindla nurga - sammu - võrra.
- Samm-mootori puhul on korraga võimalik kontrollida nii nurkkiirust, pöördenurka kui pöörlemise suunda.
- Samm-mootor võib teha täispöördeid.
- Samm-mootor ei suuda taluda suuri koormusi
- Samm-mootori juhtimiseks on tarvis riistvaralist mootoridraiverit ja selle juhtimiseks enamasti ka eraldi teeki.

# Samm-mootori näide 28BYJ-48 mootori ja ULN2003 draiveri abil



# Samm-mootori näitekood

```
#include <Stepper.h> //kasutame Arduino Stepper.h teeki

const int stepsPerRevolution = 2048; // mitmest sammust koosneb mootori üks täispööre
//deklareerime stepper objekti, viigud 8-11 on sisendid
Stepper myStepper = Stepper(stepsPerRevolution, 8, 10, 9, 11); //viikude selline järjekord
selle mudeli puhul on oluline

void setup() {
  myStepper.setSpeed(5); //määrame mootori kiiruse, ühikuks on pööret/minutis
  Serial.begin(9600);
}

void loop() {
  // Teeme ühe täispöörde päripäeva:
  Serial.println("Päripäeva");
  myStepper.step(stepsPerRevolution);
  delay(500);

  // Teeme ühe täispöörde vastupäeva:
  Serial.println("Vastupäeva");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```