

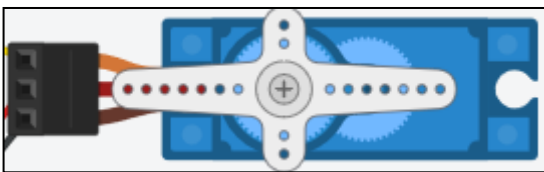


Arduino koolituse üheksas tund – servomootorid ja nende kasutamine

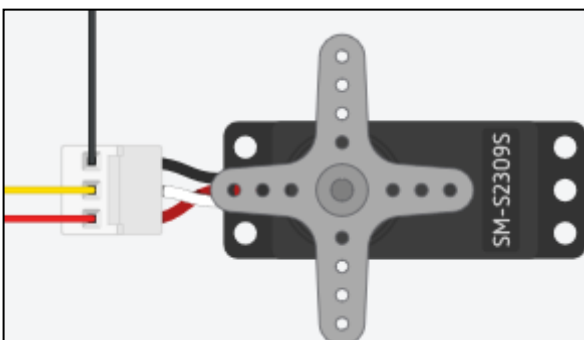
Õppematerjali koostajad: Indrek Karo ja Angela Leppik

Servomootorid ja nende kasutamine

Servomootor ehk täiturmootor on tavaliselt pisike mootor (on ka suuremaid, aga hetkel need ei ole olulised), mis muudab temaga käitatavate mehhanismide asendit vastavalt etteantud seadesignaalidele. Servomootor saab oma liikumissignaali läbi servovõimendi. Liikumist kontrollivad tahhomeeter ja positsiooni enkooder (ehk kontrollitakse liikumise kiirust ning asukohta), mis omakorda saavad häiringute korral signaali Arduinole, muutmaks kas suunda, kiirust või asukohta. Tavaliselt muudetakse programmiga ainult kiirust ja tulevast asukohta. Servomootoreid on mitmeid sorte, kuid tihti jaotatakse neid kaheks erinevaks tüübiks. Piiramatu liikumisega mootoriteks, mis saavad kogu aeg pöörelda (360 kraadi) ja piiratud liikumisega mootoriteks, mille liikumisulatus on maksimaalselt 180 kraadi. Tinkercadil on kaks erinevat servomootorit – väga pisike, mida nimetatakse mikroservoks (pilt nr. 61) ja suurem (pilt nr. 62). Mõlema puhul on tegemist piiratud liikumisega mootoriga, sest saab kasutada ka tavalist alalisvoolu mootorit (kuigi see ei ole eriti täpne).



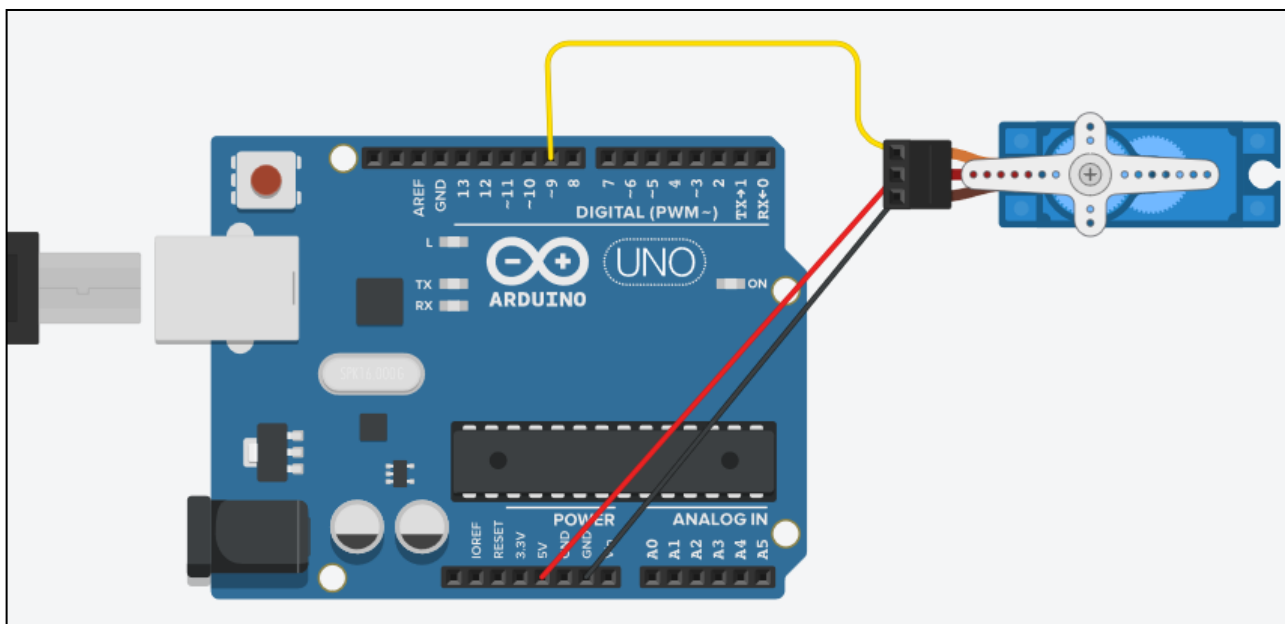
Pilt nr. 61



Pilt nr. 62

Servomootorite ühendamine

Arduino servomootorite ühendamise loogika on sarnane muudele seadmetele. Kollane juhe (mõnikord ka valge või oranž) on signaaliks ning punane ja must (harva ka pruun) on vastavalt 5V ja maandus. Servomootori katsetamiseks kasutame alguses väiksemat mootorit. Kuna servo ise sisaldab ka andureid, siis on selle kasutamiseks vaja lisada programmi servo kontrolliks vastav teek. Esimese programmina liigutame servot lihtsalt 90 kraadi. Servomootor alustab 0 kraadist ning läheb sealt edasi. Kui on vaja teisele poole liigutada, siis on arukas kõigepealt viia servo keskmisesse asendisse ning alles seejärel hakata liigutama. See on ka põhjuseks, miks tihti alguses liigutatakse servot 90 kraadi. Servo kiirust reguleeritakse viiteaja abil käsuga `delay ()`. Mootori ühendamise loogika on ära toodud pildil nr. 63.



Pilt nr. 63

Mikro servo kood 90 kraadi keeramiseks

```
#include <Servo.h>           //servo teegi lisamine
Servo mootor;                //ütleme, et Arduino arvaks, et Servo nimi on mootor

void setup ()
{
  mootor.attach (9);         //ütleme, et Arduinole on servomootor ühendatud väljaviiku nr. 9
}
```



```
void loop ()
{
mootor.write (90);           //anname servole käsu liikuda 90 kraadi
  delay(1000);               //anname servole viiteaja, iseenesest seda ei ole vaja, aga siis liigub
                             simulatsioon kiiremini
```

Liikumine edasi ja tagasi

Servomootoreid saab panna liikuma ka edasi-tagasi. Selle jaoks saab kasutada “for” tsüklit, mille abil saab samuti liikumise kiirust muuta. Siin saab vaadata ka kui palju servo on liikunud kasutades jadapordi monitори.

```
#include <Servo.h>           //servo kontrolliks vajaliku teegi lisamine
Servo mootor;               //anname servole nime

void setup ()
{
  Serial.begin (9600);       //pannakse tööle jadapordi monitor
  mootor.attach (9);         //ütleme, et servo nimega „mootor“ on väljaviigus nr. 9
}

void loop ()
{
  for (int i=0; i<180; i++){ //paneme tööle “for” tsükli, liigutamaks servot edaspidi 1 kraadi
                             haaval kuni 180 kraadini

    mootor.write (i);        //paneme mootori tööle ning see liigub muutuja “i” võrra
    Serial.println (i);      //näitame jadapordi monitoris, mis on “i” väärtus, sellega on näha ,kui
                             kaugele servo on juba liikunud

    delay (10);              //ütleme kui kaua peab ootama enne, kui järgmise tsükli osaga jätkata,
                             põhimõtteliselt saab sellega reguleerida mootori liikumise kiirust
                             (mida väiksem number, seda vähem ootab enne igat liigutust ja seega
                             liigub kiiremini)
  }                           //“for” tsükli lõpp

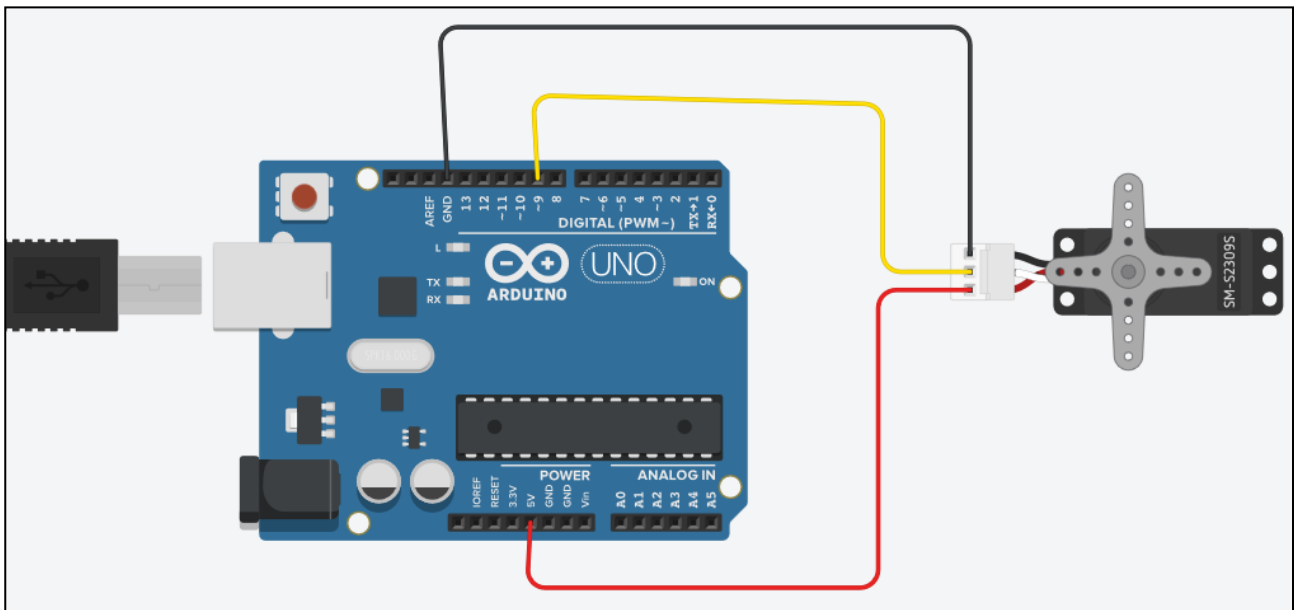
  for (int i=180; i>0; i--){ //paneme tööle “for” tsükli, liigutamaks servot tagurpidi 1 kraadi
                             haaval kuni 180 kraadini
```



```
mootor.write (i); //paneme mootori tööle ning see liigub muutuja "i" võrra
Serial.println (i); //näitame jadapordi monitoris, mis on "i" väärtus, sellega on näha kui
//kaugele servo on juba liikunud
delay (500); //ütleme, kui kaua peab enne järgmise kraadi keeramist ootama, nagu
//näha on siin number suurem (500) ning seetõttu liigub servo tagasi
//aeglasemalt
} //“for” tsükli lõpp
} //korduse lõpp
```

Suurem servomootor

Suuremat servomootorit kasutatakse Arduino puhul üsna sarnaselt väikesele, erinevus on lihtsalt võimsuses ning kinnitusosades. Ühendamise loogika on sama, mis väiksemal servomootoril ning seda on näha pildil nr. 64. Kood on ka sama, mis väiksemal servol.



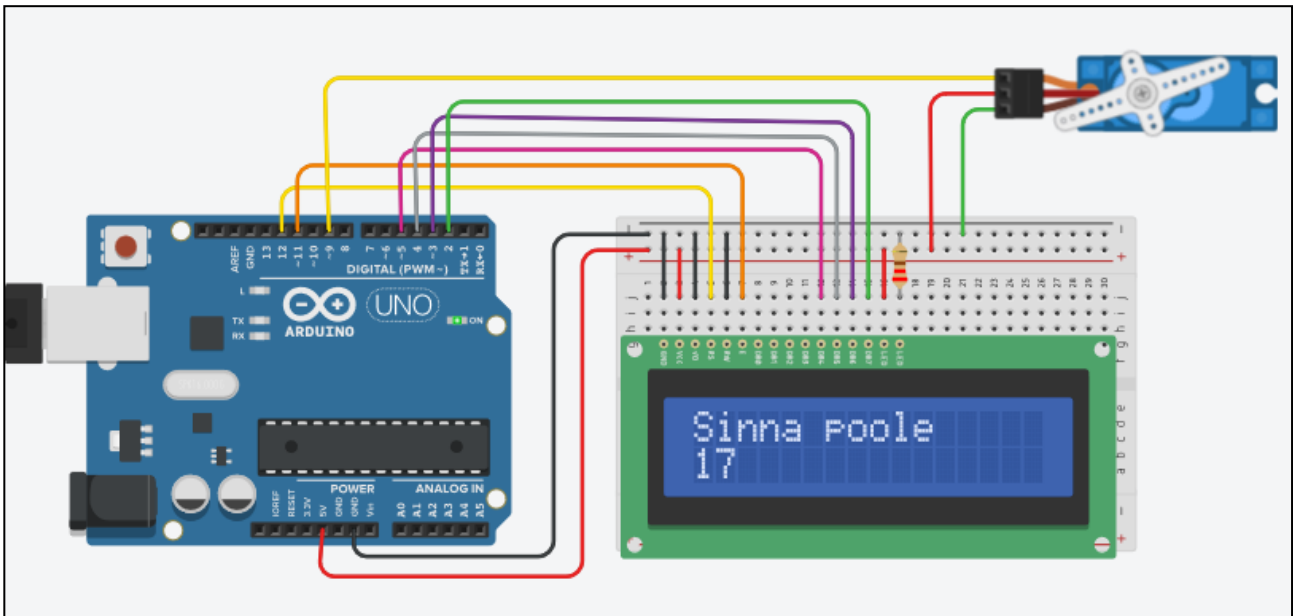
Pilt nr. 64

Servomootor ja ekraan

Servomootorit kasutatakse tihtipeale erinevate asjadega koos. Järgnevalt on pildil nr. 65 ära toodud näidisprogramm servomootori ja ekraaniga, kus ekraanil on näha millisele poole mootor parajasti keerab ning kui kaugele on juba keeratud. Selliselt erinevaid asju kombineerides on arukas tähele panna ühendusi, mis on küll sarnased ekraani peatükis vaadatuga. Siiski on mõistlik hoida need



võimalikult vähesed ja selged, sest nii on väiksem võimalus, et midagi sassi läheb. Samuti tuleb tähele panna seda, et ekraanil nädataval arvul on numbrite kogus muutuv (kord on üks number, siis jälle kaks või kolm), seetõttu tuleb peale numbri kirjutamist see ka ära kustutada, sest muidu jätab ekraan selle meelde ning hakkab näitama arusaamatut teavet.



Pilt nr. 65

Servo ja ekraani programm

```
#include <LiquidCrystal.h> //lisame ekraani teegi
#include <Servo.h> //lisame servomootori teegi

LiquidCrystal lcd (12, 11, 5, 4, 3, 2); //ütleme, milliseid väljaviike ekraan kasutab
Servo mootor; //anname servole nime "mootor"

void setup ()
{
  lcd.begin (16, 2); //ütleme, et LCD-l on 16 veergu ja 2 rida
  mootor.attach (9); //ütleme, et mootor on pesas 9
}

void loop ()
{
```



```
For (int i=0; i<180; i++){
  mootor.write (i);
  lcd.setCursor (0, 0);

  lcd.print ("Sinna poole ");

  lcd.setCursor (0, 1);
  lcd.print (i);

  lcd.print (" ");

  delay (10);

}
For (int i=180; i>0; i--){
  mootor.write (i);
  lcd.setCursor (0, 0);
  lcd.print ("Teisele poole");
  lcd.setCursor (0, 1);
  lcd.print (i);

  lcd.print (" ");

  delay (10);
}
}
```

//“for” tsüklil servo keeramiseks ühes suunas
//ütleme, et servo liigub “i” kraadi
//pannakse paika kirjutamise alguse koht ekraanil
(esimene rida ja esimene veerg, täitsa algusest vasakult)
//kirjutatakse ülemisele reale “Sinna poole” ning
jäätakse tühimik, sest järgmine kirjutis on pikem
//hakatakse kirjutama teise rea algusse
//näidatakse, mis on hetkel “i”, ehk mitu kraadi on
servo keeranud
//kirjutatakse teise rea algusse tühjus, et varem
kirjutatu ära kustutada
//ootab 10 millisekundit, sellega saab reguleerida servo
keeramise kiirust (mida väiksem number, seda
kiiremini keerab)

//“for” tsüklil servo keeramiseks teises suunas
//ütleme, et servo liigub “i” kraadi
//pannakse paika kirjutamise alguse koht ekraanil
//kirjutatakse ülemisele reale “Teisele poole”
//hakatakse kirjutama teise rea algusse
//näidatakse mis on hetkel “i”, ehk mitu kraadi on servo
keeranud
//kirjutatakse teise rea algusse tühjus, et varem
kirjutatu ära kustutada
//ootab 10 millisekundit