

INSTITUTT FOR ELEKTRONISKE SYSTEMER

IELS1001 - INGENIØRPROSJEKT

Motor-Modul

August 4, 2023

I denne labben skal vi se på styring av forskjellige typer motor ved bruk av Arduino. Vi skal se på tre ulike typer motorer og konseptene rundt de. Disse er DC-motor, Stepper-motor og Servo-motor.

DC-Motor

En DC-motor er en motor som går på likestrøm, og kan rotere i begge retninger basert på strømretningen. DC står for directional current, altså likestrøm. En DC-motor har som regel to tilkoblingspunkter på en side, og en aksling på motsatt side. Man kan si det ene tilkoblingspunktet er spenningsforsyning og det andre jord, men om man bytter på disse vil forskjellen være at motoren spinner motsatt vei.

Innsiden av en DC-motor På innsiden av en DC-motor er det et tilnærmet uniformt magnetfelt, skapt av permanente magneter. I magnetfeltet ligger det en rotor rundt akslingen, som har to eller flere elektromagnetiske poler. Disse polene magnetiseres i tur og orden slik at rotoren, og dermed akslingen, blir rotert i riktig retning basert på magnetfeltet. Denne magnetiseringen blir styrt av en kommutator som gjør at polene magnetiseres i riktig rekkefølge. Kommutatoren har kontakt med de to tilkoblingspunktene ved hjelp av to børster av mykt ledende materie.

H-Bro

For å styre en DC-motor i begge retninger uten å måtte koble om kretsen trenger vi underkretser som kontrollerer strømretningen. En av disse er en H-bro, som er en underkrets som tillater å endre hvilket tilkoblingspunkt som blir positivt.

I figur 1 er det tegnet et veldig enkelt eksempel på en H-bro. Den fungerer ved at når de røde bryterne er lukket vil de blå være åpne, og motsatt. Dermed vil strømmen ha forskjellig retning i broen mellom de to sidene. Om man setter DC-motoren midt i den vil man da kunne snu strømretningen til DC-motoren.

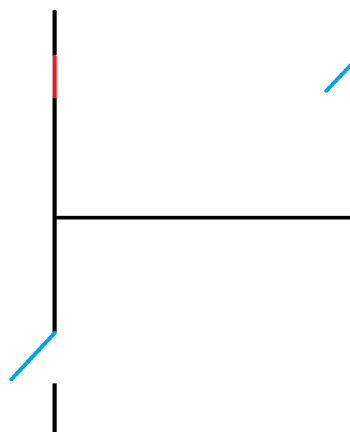


Figure 1: Eksempel på en H-bro

Motor-driver

En motordriver er en integrert krets som tillater oss å styre DC-motorer. De bruker logiske signaler, fra for eksempel en mikrokontroller, til å styre en sterkere spenningskilde.

L293D

Motordriveren vi skal bruke heter [L293D](#). På figur 2 kan dere se pinneoversikten til motor-driveren. På den er det et par ting å få med seg:

- V_s er tilkoblingen til den eksterne spenningsforsyning (9V)
- V_{ss} er tilkoblingen til mikro-kontrollerens spenningsforsyning (5V)
- Det er totalt fire *output* pinner, dette tillater blant annet at man kan koble opp to DC-motorer, en på output 1 og 2, den andre på output 3 og 4.
- Man bestemmer hastighet ved å skrive et PWM signal til ENABLE.
- Man bestemmer retning motoren skal kjøre ved å sende et høyt signal til INPUT.

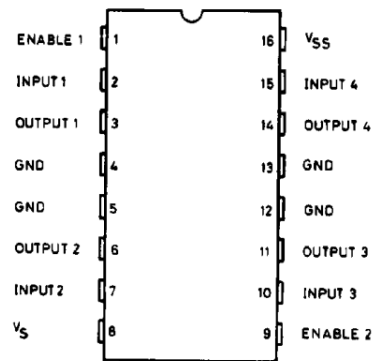


Figure 2: Pinne-oversikt over L293D

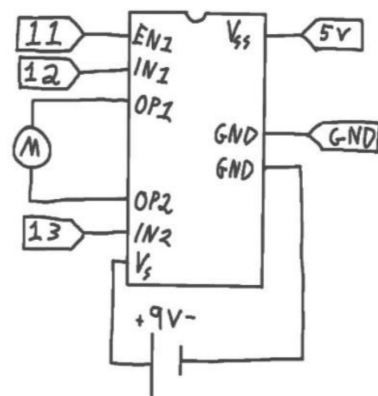


Figure 3: Eksempel på oppkobling av L293D

Stepper-Motor

En stepper motor er en type DC-motor som tillater mer presisjon enn en vanlig DC-motor. I motsetning til en vanlig DC-motor, kan en stepper motor rotere til et bestemt steg og stoppe. På grunn av dette kan man lettere stille akkurat hvor mye motoren skal rotere.

Oppkobling

For å styre stepper-motoren benytter vi oss igjen av motordriveren L293D. Figur 4 viser en oppkobling av en steppermotor mot L293D. Her kan du se at steppermotoren krever bruk av alle OUTPUT-pinnene til motordriveren.

Stepper-bibliotek

For å styre steppermotoren bruker vi som sagt motor-driveren, men i tillegg tar vi i bruk [Arduino sitt stepper bibliotek](#). Dette biblioteket forenkler styringen av stepper-motoren veldig og gjør at vi kan jobbe med mer meningsfulle verdier. Det anbefales å lese seg opp og forstå referansen lenket over, men det viktigste er å vite er:

- `#include <Stepper.h>` må brukes for å inkludere biblioteket etter å ha lastet det ned.
- `Stepper myStepper(pin1, pin2, pin3, pin4)` brukes til å sette opp ett stepper-objekt kalt myStepper.
- `myStepper.setSpeed(speed)` setter hastigheten til motoren
- `myStepper.step(step)` beveger motoren et satt antall steg.

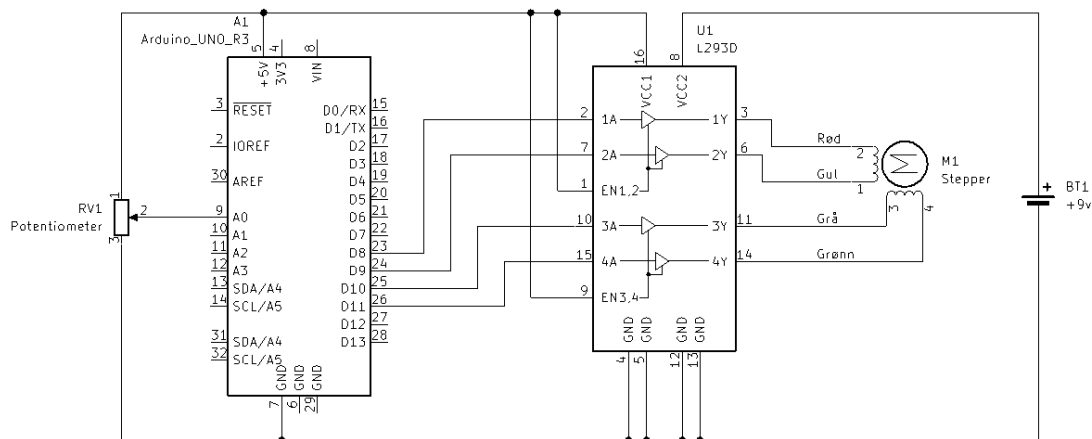


Figure 4: Oppkobling stepper-motor

Servo-Motor

En servomotor er en type motor som er laget for presisjon av bevegelser. I likhet med stepper-motoren er den lagt for å kunne gå til bestemte punkter og holde seg der. Servo-motoren styres av en kontroller som vet hvordan den er orientert. Det finnes servomotorer som kan rotere kontinuerlig, men de vi skal jobbe med kan kun rotere i 180°.

Styre en Servo

En servo har som regel tre tilkoblingspunkter: Spenningsforsyning, Jord og input. Disse er som regel Rød for spenningsforsyning, Svart/Brun for jord og gul/hvit/annen farge for input.

Som input tar Servo-motor imot et PWM-signal på 50Hz. Lengden av den høye pulsen i PWM-signalet er hva som bestemmer hvilken vinkel den skal bevege seg til. Kort puls ($\sim 1\text{ms}$) tilsvarer 0°, mens lang puls ($\sim 2\text{ms}$) tilsvarer 180°. Pulslengde mellom disse vil tilsvare en vinkel mellom 0° og 180°. Ingen signal vil gjøre at Servo-motor står stille.

Servo-bibliotek

For å styre Servo-motoren bruker vi [Arduino sitt servo-bibliotek](#). Dette biblioteket tillater bruk av funksjoner som tillater oss å bestemme hvilken vinkel Servoen skal stå i, oppgitt i grader. Igjen anbefales det å lese referansen, og prøve å forstå det, men viktigst er at:

- `#include <Servo.h>` må brukes for å inkludere biblioteket etter å ha lastet det ned
- `Servo myservo` brukes til å sette opp ett servo-objekt kalt `myservo`.
- `myservo.attach(pin)` brukes for å koble en pinne til servoen.
- `myservo.write(degrees)` brukes for å flytte servoen til gitt antall grader.

1 Oppgave: DC-Motor

a) Sett opp DC-Motor

Nå skal dere sette opp DC-motoren og koble den opp til motor-driveren. I dokumentasjonen over kan man finne pinneoversikten og et eksempel på oppkobling.

OBS! Husk å skille mellom Vs og Vss!

b) Styr motor

For å styre motoren må man sende signal til INPUT-pinnene og ENABLE-pinnen. Hastigheten til motoren er bestemt ved å sende inn et PWM signal til ENABLE, ved hjelp av `analogWrite()`. Begrens gjerne dette PWM-signalet til å være maksimalt 55, ettersom motoren kan gå veldig fort. Retning er bestemt ved å sende høyt logisk signal til en av INPUT-ene.

c) Vifte

Som nevnt tidligere kan man justere hastighet og retning i koden ved å bruke motordriveren sine pinner. Et enkelt produkt man kan lage med dette er en justerbar vifte. Lag en vifte som kan styres av en valgfri styringsmekanisme som tillater å styre hastigheten til motoren. Det skal være minst tre hastighetsnivåer man kan velge mellom.

Eksempel på styringsmekanismer er:

- Knapper
- Seriel kommunikasjon
- Potensiometer
- Temperatursensor ++

2 Oppgave: Stepper-Motor

a) Sett opp Stepper

Nå skal dere se på Stepper-motoren og koble den opp mot motor-driveren. I dokumentasjonen kan man finne pinneoversikten og et eksempel på oppkobling.

Det er vært å merke at steppermotoren krever alle fire INPUT pinnene, i motsetning til DC-motoren som kun trenger to per motor.

b) Få Stepper til å gå til bestemt step

Som nevnt er fordelen med stepper-motoren at man kan til et bestemt steg, som tillater mye mer kontroll over bevegelsen til motoren. Dette styres av biblioteket og man bruker `myServo.step()` for å bestemme antall steg den skal gå. Hvor mange steg en stepper-motor har burde være oppgitt i databladet eller ligge ved stepper-motoren. Den vi skal bruke har 200 steg per omdreining, som vil si om man skriver `step(200)` vil stepperen rotere en hel omdreining.

c) Fartsmåler med potmeter

Man kan også justere hastighet som stepperen roterer med ved bruk av `setSpeed()` funksjonen. Lag et program som justerer hastigheten til stepper-motoren basert på vridningen til et potensiometer. Husk at verdiene må omgjøres ettersom potensiometeret vil gi verdier opp til 1023.

OBS! `step()` funksjonen er blokerende, som vil si at koden venter med og gå videre til `step()` er ferdig. Om du setter hastigheten for lav og ber stepperen gå mange steg vil dette ta lang tid

Denne oppgaven skal også brukes i servo-delen, så det anbefales å la oppkoblingen stå om det er plass.

3 Oppgave: Servo-Motor

a) Sett opp Servo og ADALM2000 oscilloskop

Nå skal vi se på Servo-motoren og koble den opp. Denne krever ikke en motordriver ettersom den ikke krever egen underkrets for å fungere. Vi skal også koble den opp mot ADALM2000 for å se nærmere på hvordan Servo-motoren får signal.

Koble rett inngang på Servo-motor, og koble ADALM2000 sitt oscilloskop på PWM-inngangssignalet.

b) Styr Servo med bibliotek og se på ADALM

For å styre Servo-motoren skal vi bruke Servo-biblioteket, dette biblioteket tillater oss å sette vinkelen til servo-motoren. Lag et program som bruker biblioteket for og sveipe servoen fram og tilbake mellom 0° og 180°.

Mer info om biblioteket finner man i dokumentasjonen over.

Ved å koble ADALM2000 sitt oscilloskop på servoen kan man se PWM signalet når det sendes.

c) Sett opp Servo som et speedometer for Stepperen

Servo-motoren sin største fordel over andre motorer er sin presisjon. Den kan lett stilles til en bestemt posisjon og holde seg i den posisjonen. Bruk dette for å lage et speedometer til stepper-motoren. Altså skal Servo-motoren stå i posisjon 0 ved ingen fart og posisjon 180 i maks fart.

d) Selvstyrende solcellepanel

Nå skal dere bruke to Servo-motorer for å sette opp en prototype for en selvstyrt solcelle, som orienterer seg etter retningen det er sterkest sollys.

Utlevert skal dere få to servo-motorer, en festet over den andre, og en "solcelle", som består av fire foto-resistorer. Bruk verdiene fra foto-resistorene til å styre hver av servoene etter lyset.

Det kan være lurt å begynne med en servo før den andre. Det kan også være lurt å legge til en `millis()` funksjon som gjør at Servo-motoren ikke beveger seg for hurtig.

4 Oppgave: Case

Bra bruk av motorer er en sentral del av flere prosjekter man kan finne på. I dette delkapittelet er det oppgaver hvor man skal lage flere av disse.

a) Lag heis

Lag en heis som kan løfte en mengde vekt. Velg en passende motor og lag en styringsmekanisme for den.

b) Lag mobil vifte

Kombiner to motorer for å lag en vifte som kan styres. Den kan styres manuelt eller automatisk.