

Project 3

IN1080 - 2018

Mats Høvin

April 3, 2018

Intro

I dette prosjektet skal vi koble opp og programmere noen vanlige byggeblokker for mekatroniske system. Konkret skal vi se på styring av steppermotor, bruk av Arduino mikrokontroller, avlesning av analoge og digitale sensorer og regulering av temperatur.

Dette prosjektet består av 3 deler, og hver prosjektgruppe (på 2 studenter) kan velge en av delene etter interesse. Dvs. det holder å gjøre en av de 3 delene som her er beskrevet. Senere i kurset kan det bli aktuelt å koble delene sammen til større systemer.

Prosjektet godkjennes ved fysisk demonstrasjon av oppkoblet system og fungerende programkode for gruppelærer/labveileder. Gruppelærer/labveileder vil kunne stille spørsmål om hvordan programkoden virker, samt hvorfor man har koblet opp kretsene slik man har valgt. Man må kunne svare tilfredsstillende på spørsmålene for å få godkjent prosjektet. Begge deltagerne på gruppen må være tilstede ved demonstrasjonen. Programkoden samt koblingsskjema sendes inn på Delvilry på vanlig måte.

Dette prosjektet er utformet litt mer som en oppgave man vil kunne møte i en bedrift ute i den virkelige verden, dvs. det er ikke detaljspesifisert hvordan man skal koble opp kretsene eller skrive programkoden. For å vite hvordan man skal koble og programmere, må man benytte seg av den kunnskapen man får på forelesninger samt felles regneøvelse på mandager. Man må også påregne noe eget søk på nettet etter relevante eksempler og spesifikasjoner.

Alle obligatoriske oppgaver ved IFI skal følge instituttets reglement for slike oppgaver. Det forutsettes at du gjør deg kjent med innholdet i reglementet og at besvarelsen er i henhold til dette. Reglementet finner du for tiden på: <https://www.uio.no/studier/eksamen/obligatoriske-aktiviteter/mn-ifi-obliger-retningslinjer.html>.

Del 1 - styring av steppermotor

I denne delen skal vi styre en lineær aktuator fra PC. Aktuatoren består av en steppermotor samt en gjengeavstand med mutter. Gjengeavstand (pitch) er 0.85mm. Motoren skal kjøre fullstepp, og drives av en dobbel H-bru motordriver av type TB6612FNG (se datablad). Motordriveren styres fra Arduinokortet som igjen kommuniserer med PCen og Javakoden dere skriver i Processing.

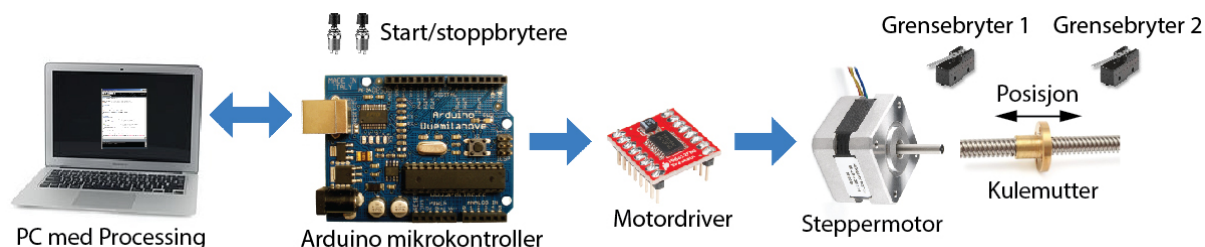


Figure 1: Systemet i del 1

Spesifikasjoner:

- I Processing skal det være mulig å spesifisere den lineære hastigheten på bevegelsen til mutteren i mm/sec. Hastigheten kan være konstant.
- Start og stopp av bevegelsen skal styres av to brytere som er koblet til Arduinokortet (en bryter for hver vei). Aktuatoren skal bevege seg kun når den korresponderende bryteren er trykket inn.
- For å unngå at mutteren hopper av skruen eller kiler seg fast i enden har vi 2 grensebrytere. Hvis/når disse bryterene aktueres må bevegelsen stoppe momentant og motoren skal gå 4 step tilbake i motsatt retning før normal operasjon gjenopptas.

Dere må selv foreta de nødvendige oppkoblingene. Husk å bruke riktige spenninger slik at komponentene ikke ødelegges.

Krav til innlevering:

- Fysisk demonstrasjon av fungerende system og programkode for gruppelærer/labveileder.
- Innlevert programkode (Devilry).
- Innlevert koblingsskjema over alle ledninger i systemet (Devilry). Dette kan tegnes for hånd.

Del 2 - avlesning av kraftsensor

I dette prosjektet skal vi lese av kraft fra en kraftsensor og i sanntid plote kraften som en funksjon av tid i Processing. Vi bruker en kraftsensor av type FX1901 som er basert på en strekklapp (load cell) koblet i Wheatstone-bru konfigurasjon. Denne sensoren gir ut en spenning som er proporsjonal med påført kraft og spenningen vil typisk bare variere med noen få millivolt (se datablad) - derfor må spenningen forsterkes før den konverteres til digital i Arduino kortet. Sensorens utgang har en relativ høy impedans slik at dere må tenke på impedanstilpassing når dere kobler sammen utgangen med inngangen på spenningsforsterkeren. Som spenningsforsterker kan dere bruke en MCP601 operasjonsforsterker (se datablad) i en egnet konfigurasjon.

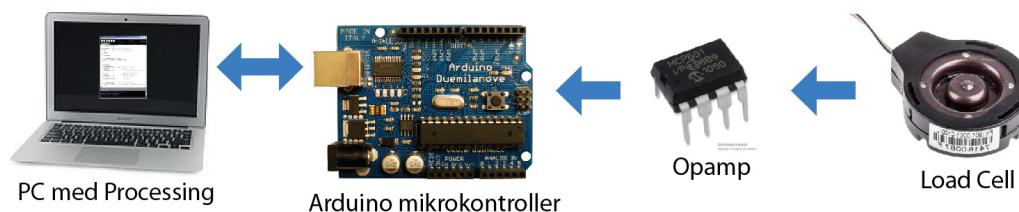


Figure 2: Systemet i del 2

Spesifikasjoner:

- Full skala trykk, plottet i Prosessering: 0 - 10kg.
- Plottetid: 20sec.
- Anbefalt analog spenningsområde inn til Arduino ADC er 1-4V.

Krav til innlevering:

- Fysisk demonstrasjon av fungerende system og programkode for gruppelærer/labveileder.
- Innlevert programkode (Devilry).
- Innlevert koblingsskjema over alle ledninger i systemet (Devilry). Dette kan tegnes for hånd.

Del 3 - regulering av temperatur

I dette prosjektet skal vi regulere temperaturen i et varmeelement. Til dette bruker vi et resitivt varmeelement HPG-2/12 (se datablad) sammen med en høypresisjons digital temperatur sensor. Varmeelementet styrer vi fra Arduino via en nMOSFET transistor av type FQP30N06L (se datablad). Temperatursensoren er av type HYT271 (se datablad) og er I2C basert. For å kommunisere med sensoren kan dere bruke Arduinos "wire" bibliotek. Dette biblioteket håndterer all lavnivå/bitnivå kommunikasjon, og dere kan fokusere på å sende/motta bytes.

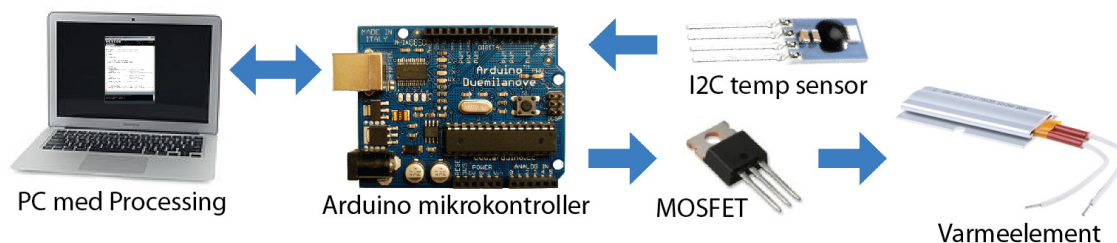


Figure 3: Systemet i del 3

Reguleringsalgoritmen dere bruker kan være så enkel som å slå på varmeelementet hvis målt temperatur er under spesifisert temperatur, og slå det av når målt temperatur er høyere en spesifisert temperatur. Hvis noen av dere er ekstra flinke kan dere også velge å programmere inn en mer nøyaktig regulering basert på PID/PWM kontroll (ref Google).

Spesifikasjoner:

- Spesifisert temperatur : 60 grader Celcius
- Målt temperatur skal plottes i Processing over en periode på 30sec
- Full spenning over varmeelementet skal være 12V

Krav til innlevering:

- Fysisk demonstrasjon av fungerende system og programkode for gruppelærer/labveileder.
- Innlevert programkode (Devilry).
- Innlevert koblingsskjema over alle ledninger i systemet (Devilry). Dette kan tegnes for hånd.

Lykke til, husk å ha det gøy !