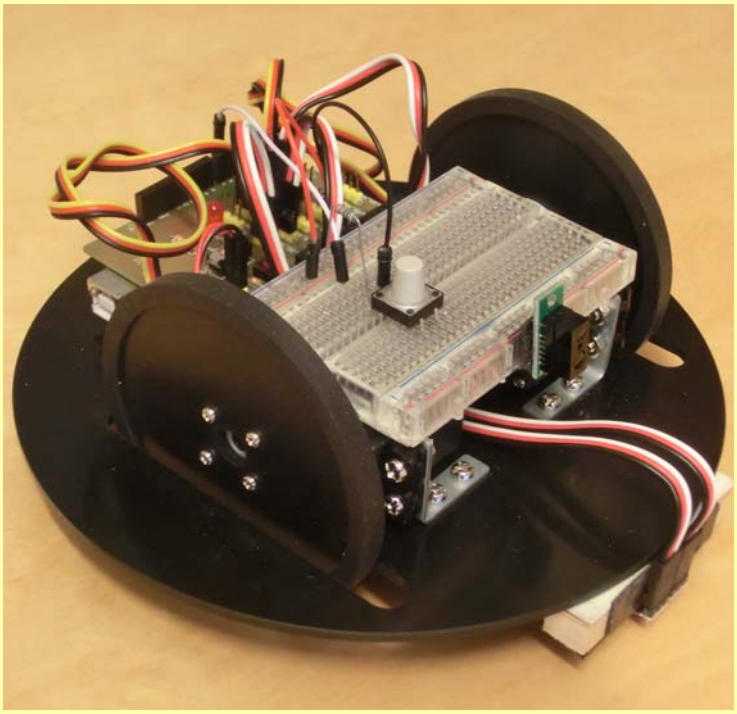
Vitensenteret

Nils Kr. Rossing, Adam Gajda

Veiledning – Mr. Abot
Elevhefte



Oktober 2014

Denne siden er blank

Veiledning – Mr. Abot
Elevhefte

Nils Kr. Rossing, Adam Gajda

Veiledning - Mr. Abot - Elevhefte

Trondheim 2014

ISBN 978-82-92088-53-1

Bidragsytere:
Nils Kr. Rossing, (nkr@vitensenteret.com) Vitensenteret i Trondheim
Adam Gajda (adam.gajda2@gmail.com)

Layout og redigering: Nils Kr. Rossing, Vitensenteret i Trondheim

Tekst og bilder: Nils Kr. Rossing, Vitensenteret i Trondheim

Faglige spørsmål rettes til:
Vitensenteret i Trondheim
v/Nils Kr. Rossing, 73 59 77 23
nils.rossing@vitensenteret.com

Kongensgate 1
7013 Trondheim

Postboks 117
7400 Trondheim

Vitensenteret i Trondheim
Telefon: 73 59 61 23
Telefaks: 73 59 61 20
http://www.vitensenteret.com/

Rev 1.2 – 03.10.14

Prosjektet er gjennomført i et samarbeid med **Atmel Norge AS** og **Skolelaboratoriet for Matematikk, Naturfarg og Teknologi ved NTNU** og **finansiert av Sparebankstiftelsen DNB NOR.**

Forord

Heftet er en veiledning for elever, og gir en kortfattet bruk av Mr. Abot som del av et aktivitets tilbud. Opplegget forutsetter en pedaggog som hjelper elevene gjennom oppgavene. I denne omgangen legges det vekt på å gi en god elevelse. Programmering og oppbygning av Arduino og Mr. abot tones ned i denne sammenhengen.

Programvare, en presentasjon og en eller flere arenaer følger med opplegget.

Vitensenteret
Oktober 2014

Nils Kr. Rossing

1 Innledning

1.1 Innhold i boksen

Følgende skal ligge i boksen

- En Mr. Abot med Arduino UNO og shield-kort
- En liten plastboks med:
  - 4 reflektanssensorer
  - 1 avstandssensor
  - 6 koblingsledninger
  - 1 bryter
  - 2 lysdioder
  - 2 motstander 220 Ohm
  - 1 motstand 10 kOhm
  - 1 høyttaler 100 Ohm
  - 5 flatkabler (3 leder)
  - 1 flatkabel (4 leder)
- En USB-kabel
- Et målbånd 3 m
- Et testark A3
- Et pappkrus (hindring)
- Et veiledningshefte (gult)
- En minnepinne
  - ***Arduino*** (Arduino programeditor/drivere)
  - ***Dokumentasjon*** (Veiledning)
  - ***Skisser*** (Prgrameksempler for Arduino)

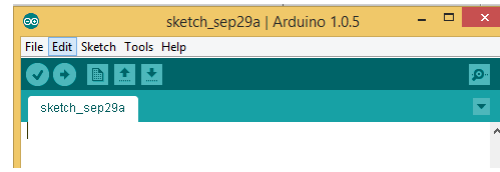
1.2 Installasjon

Programmet kan enten hentes på nettet: http://arduino.cc/en/main/software eller fra den vedlagte minnepinnen. Den aktuelle versjonen er pr. oktober 2014: Arduino 1.0.6. Programmet er tilgjengelig for Windows, MAC og Linux.

Programmet lastes ned og startes. Installasjonen fullføres.

1.3 Ved oppstart av programmet

Start programmet ved å trykke på ikonet:  .



Følgende vindu kommer opp:

- Velg “Tools/Board/Arduino Uno”

- Velg “Tools/Serial Port/COM<med høyeste nummer>”
- Velg “File/Open/Elev_program/Elev_program”

1.4 Last opp programmet

- Plugg ledningen inn i USB inngangen
- Legg programmet over på Arduino-kortet ved å trykke ikonet 
- Se side 10 for omtale av bruk av programeditoren

1.5 Installasjon av drivere

Normalt skal driverne installeres sammen med programmet. Dersom dette ikke skjer så vil en som oftest få beskjed om at det er funnet ny hårdvare når man plugger inn Arduino kortet, med spørsmål om man ønsker å installere drivere. Så følger man den anbefalte prosedyre.

Sjekk at driverne er installert:

- Start Arduino-editoren (IDE)
- Velg “Tools/Serial Port/COM<med høyeste nummer>”

Dersom ingen COM-porter vises eller den ene som er der har et meget lavt nummer (f.eks. 1), kan det skyldes at driverne ikke er installert. Forsøk da følgende:

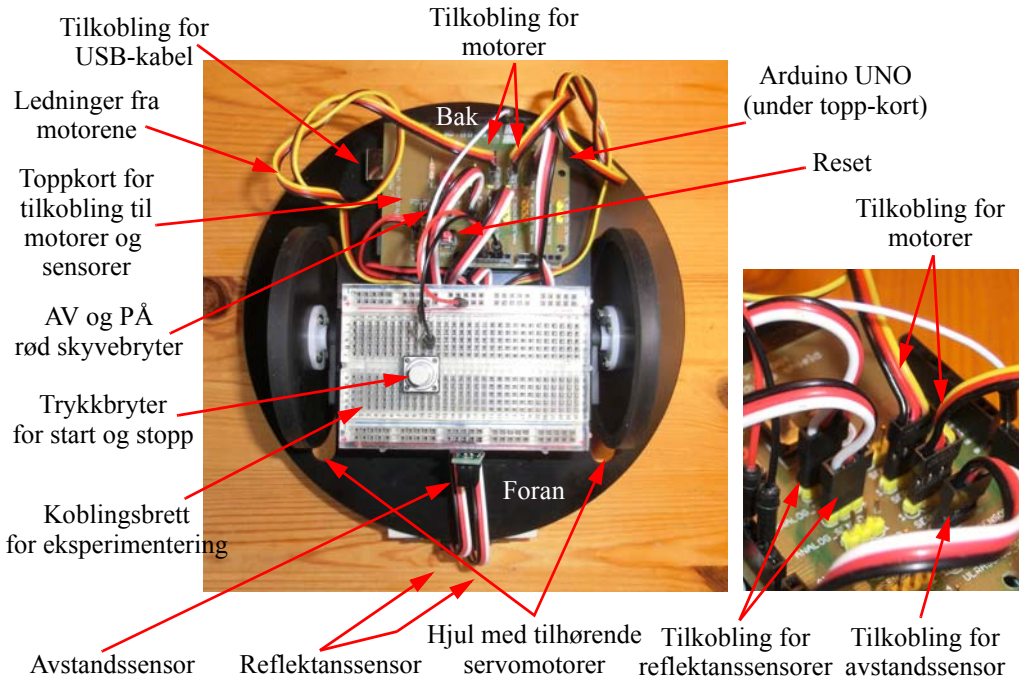
- Gå til “Kontrollpanelet”
- Velg “System”
- Velg “Enhetsbehandling” fra menyen til venstre
- Åpne “USB-kontrollere” ved å trykke på den vesle pila til venstre for teksten.
  Da kommer det opp en liste over USB-porter
- Høyreklikk på porten med ukjent hårdvare (markert med et spørsmålstegn), og du får opp en nedtrekksmeny
- Velg “oppdater driverepro gramvare” og følg prosedyren for installasjon av drivere
- Driverne ligger i en underkatalog til Arduino-programvaren:
  /Programfiler(x86)/Arduino/drivers

1.6 Dersom forbindelsen med kortet forsvinner

Det hender at forbindelsen med kortet forsvinner idet man setter i eller trekker ut ledningen i Arduino-kortet. Dette skyldes mest sannsynlig at robotens egen strømforskyning ikke er tilkoblet, dermed vil motorene i et kort øyeblikk trekke mer strøm en USB-porten klarer å tilføre og den vil koble ned porten. Det tar da en stund før den igjen kommer opp. En kan da trekke ut USB-kabelen, lukke programmet vente litt og så starte programmet på nytt, slå på Mr. Abots egen strømforskyning og plugge inn ledingen.

2 Oppbygging og montering av Mr. Abot

Figuren under viser Mr. Abot sett ovenfra:

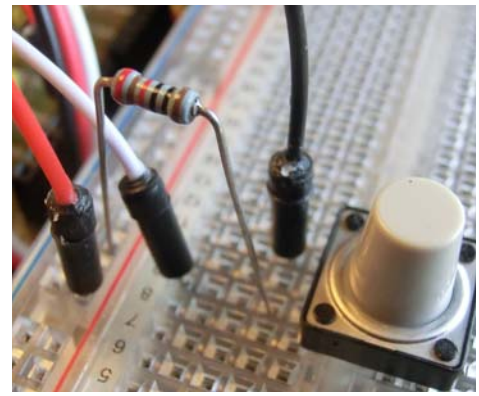


1. **Koble til servomotorene:**
   Høyre servomotor –> *Servo 1*
   Venstre servomotor –> *Servo 2*

2. **Koble til reflektansensorene**
   Høyre reflektanssensor –> *Analog_sensor_0*
   Venstre reflektanssensor –> *Analog_sensor_1*

3. **Koble til avstandssensor**
   Avstandssensor –> *Digital sensor*

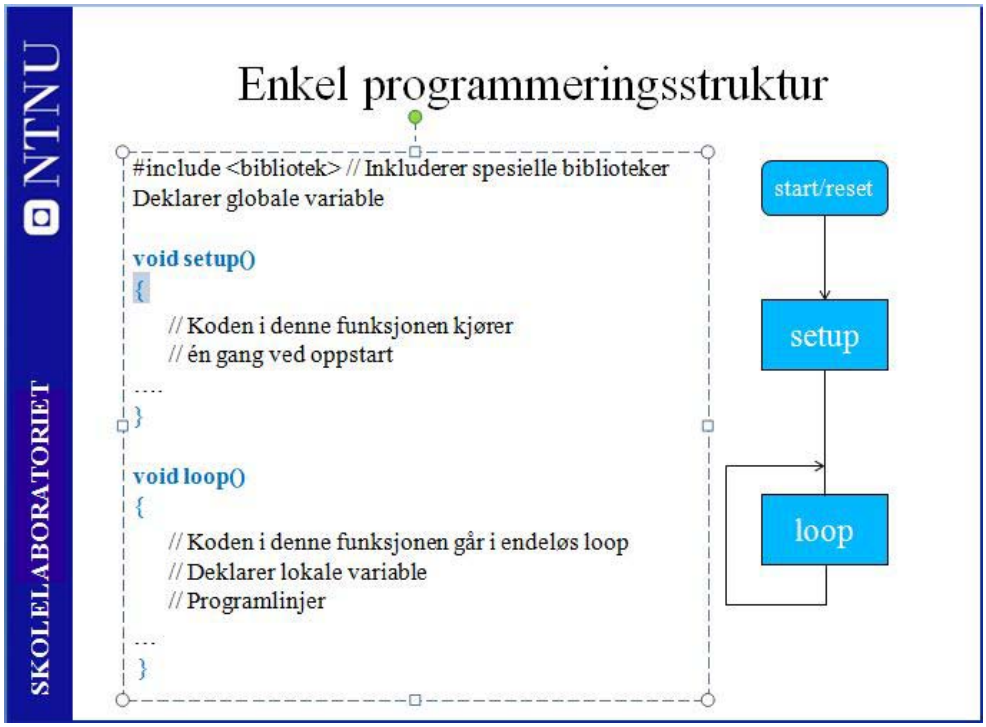
4. **Koble opp bryter**
   Trykkbryter, motstand (10 kOhm) og ledninger kobles opp som vist
   Sort ledning –> *Jord (GND)*
   Rød ledning –> *+ 5V*
   Hvit ledning –> *digital inngang 7*



Sorte ledninger skal alltid til kontakter merket 1

3 Grunnleggende Programstruktur

Dette arket viser den grunnleggende programstrukturen for et typisk Arduino program.



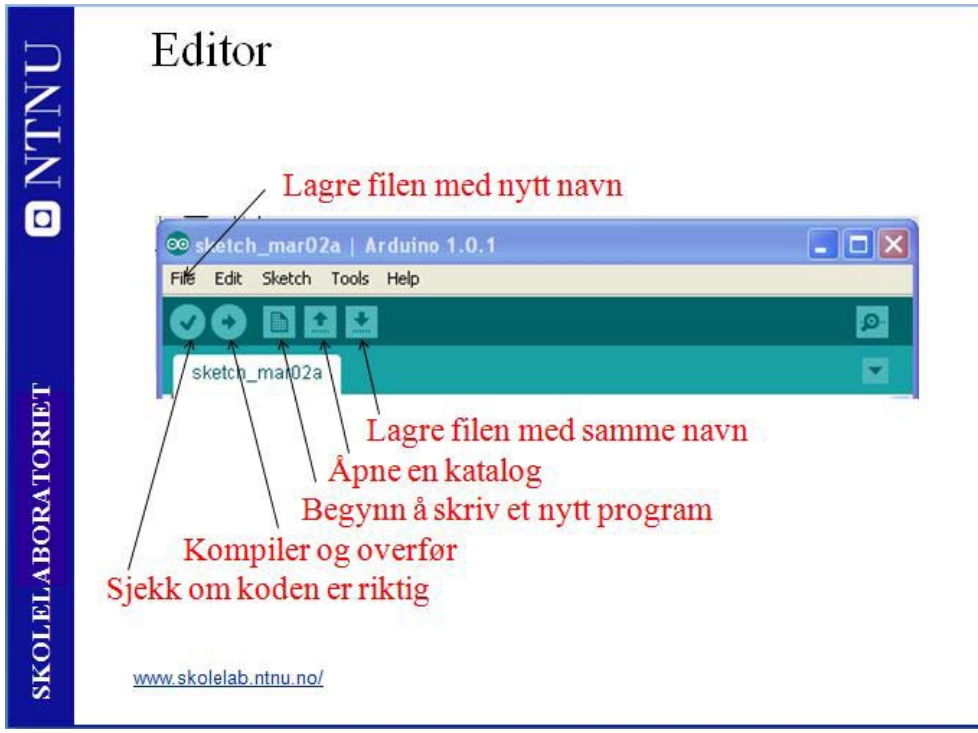
Start: Her legges:
- # funksjonsbiblioteker
- Deklarering av globale variable

Setup: Her legges:
- de kommandoene som bare skal utføres en gang i starten av hver kjøring

Loop: Her legges:
- de kommandoene som skal gjentas i en endeløs loop

4 Programeditoren

Programmet skrives i *programeditoren*. Når vi er ferdige, oversettes programkoden til en kode som Arduino-kortet forstår. Vi sier at programmet *kompileres*. Så snart programmet er overført til kortet, begynner det å utføre kommandoene i programmet.

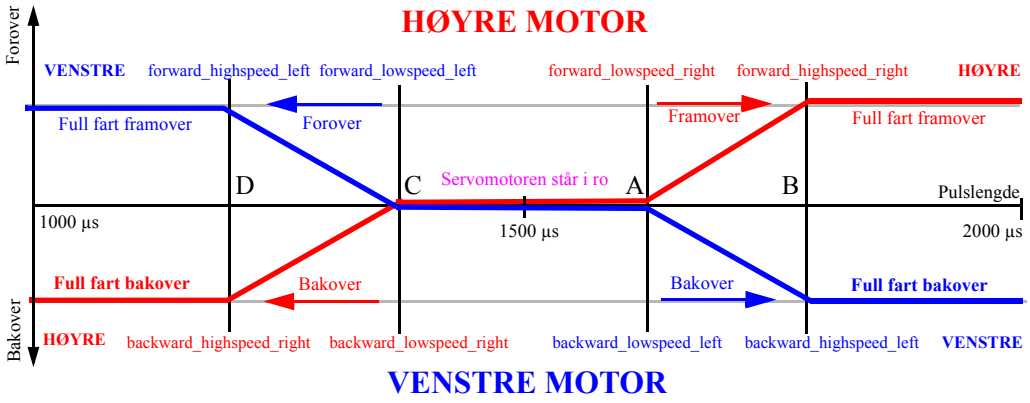


Gangen i programmering:

1. Skrivprogrammet
2. Sjekk om det er feil programkoden
3. Lagre programfilen
4. Overfør programmet til Arduino-kortet

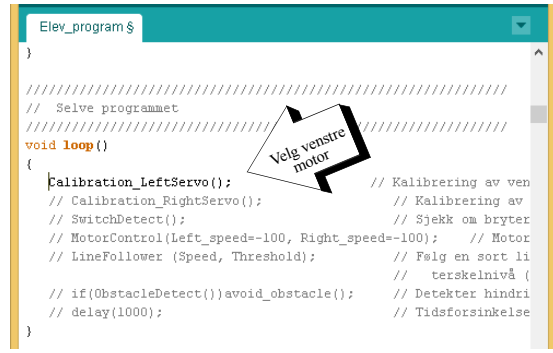
5 Kalibrering av motorene

Hastigheten til en servomotor styres av et tog av pulser. Lengden av pulsene bestemmer farten. Pulslengden er fra ca. 1000 µs (full fart bakover) til 2000 µs (full fart forover). I et område omkring 1500 µs står motoren stille. Som det framgår av figuren så kan farten bare endres i to områder, (A–B) og (C–D). Legg merke til at de to motorene må gå motsatt vei for å gå bent:



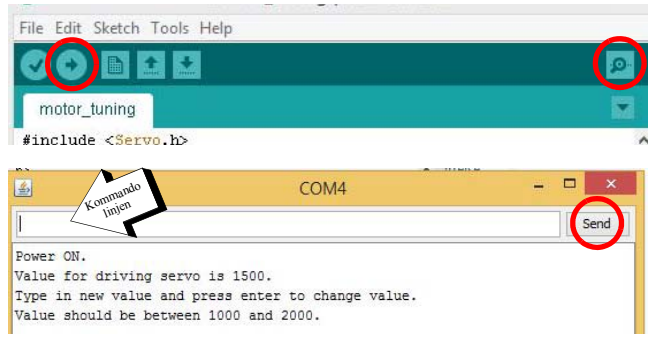
Verdiene i *knekkpunktene* A, B, C og D, er heller ikke like fra motor til motor. Vi må derfor finne disse fire grenseverdiene for høyre og venstre motor, og bruke dem til å kalibrere programmet som styrer roboten:

forward_lowspeed (A)
forward_highspeed (B)
backward_lowspeed (C)
backward_highspeed (D)

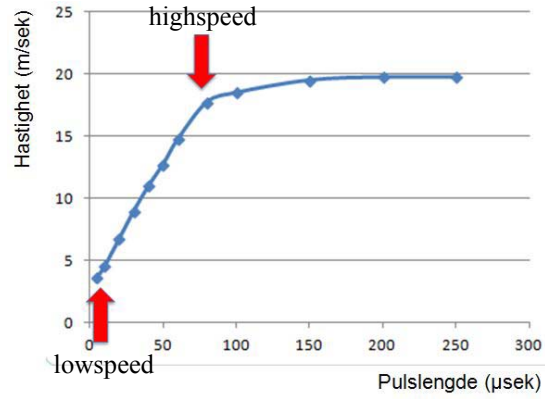


Slik bestemmes knekkpunktene:

1. Last opp programmet: *elev_program*
2. Gå til: void **loop**() og velg ***Calibration_Left.Servo();*** ved å fjerne // foran.
3. Overfør programmet til roboten ved å velge 
4. Åpne monitorvinduet 
5. Skriv inn verdier fra 1000 til 2000 på ***kommandolinjen*** og velg **Send**.



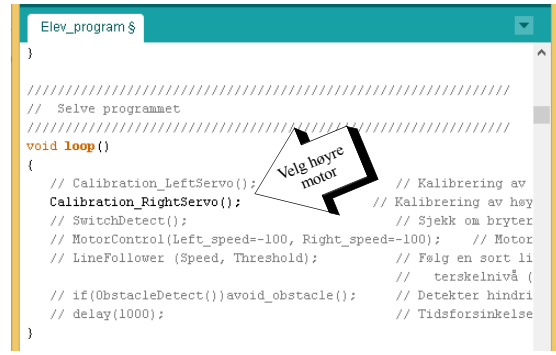
Punktet ***lowspeed*** er der motoren akkurat begynner å gå.
Punktet ***highspeed*** er det punktet hvor man når maks. fart, se figuren til høyre. Det enkleste er kanskje å lytte til forandrnger i lyden fra motorene. Legg også merke til at hastighetsendringene er noe langsommere nær toppen.



6. Finn knekkpunktene for venstre servomotor:

   Forward_highspeed_left: _____
   Forward_lowspeed_left: _____
   Backward_lowspeed_left: _____
   Backward_highspeed_left: _____

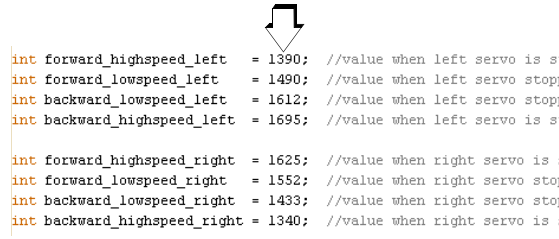
7. Velg ***Calibration_RightServo();*** i programmet ved å fjerne //. Husk å sette // foran Calibration_LeftServo();



8. Skriv inn verdier fra 1000 til 2000 på ***kommandolinjen*** og velg **Send**. Finn knekkpunktene som for den venstre motoren.

   Forward_highspeed_right: _____
   Forward_lowspeed_right: _____
   Backward_lowspeed_right: _____
   Backward_highspeed_right: _____

   Husk å sette // foran Calibration_RightServo(); når du er ferdig.

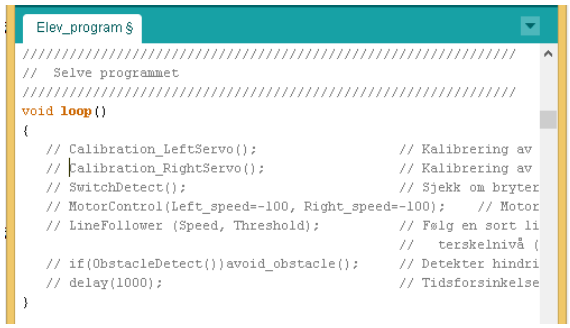


9. Gå så lengre fram i programmet, der hvor verdiene for knekkpunktene skal legges inn. Se figuren til høyre.

10. Skriv inn de nye verdiene og lagre programmet.

6 Oversikt over funksjoner

Programmet består av en rekke funksjoner som kan kalles opp innenfor programloopen (void loop();). Disse velges inn etter behov ved å fjerne // foran.



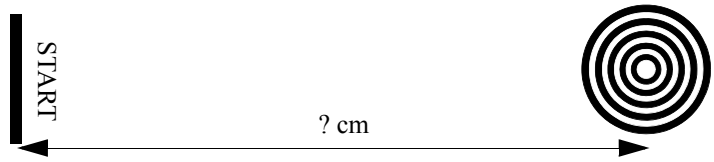
Følgende funksjoner ligger klare for å velges inn:

- ***Calibration_LeftServo();***
  For kalibrering av venstre servomotor.
- ***Calibration_RightServo();***
  For kalibrering av høyre servomotor.
- ***SwitchDetect();***
  Programmet starter når man trykker på knappen på koblingsbrettet, og stopper når man trykker en gang til.
- ***MotorControl(Left_Speed*** = <venstre verdi>***, Right_Speed*** = <høyre verdi>***);***
  Funksjonen setter farten på hver av servomotorene fra:
  <venstre/høyre verdi> = 1 - 100 –> venstre/høyre servo, sett fart *forover*
  <venstre/høyre verdi> = –1 - –100 –> venstre/høyre servo, sett fart *bakover*
  <venstre/høyre verdi> = 0 (full stopp)
  <venstre verdi> = (1 - 100), <høyre verdi> = –(1 - 100) –> sving mot høyre
  <venstre verdi> = –(1 - 100), <høyre verdi> = (1 - 100) –> sving mot venstre
- ***LineFollower (Speed*** = <fart verdi>***, Threshold*** = <terskelverdi>***);***
  Følg kanten av svart linje.
  <fart verdi> = 1 - 100 –> hastigheten til roboten
  <terskelverdi> = 0 - 1023 –> terskel for å skille mellom lyse (0) og mørke (1023) felt
- ***delay (***<millisekunder>***);***
  Forsinker programmet i et antall millisekunder gitt av argumentet i funksjonen.
- ***stop_running ();***
  Stopper programmet (og roboten) til man trykker reset eller slår AV og PÅ strømmen.
- ***avoid_obstacle ();***
  Funksjonen må programmeres i henhold til størrelsen på hindringen.
- ***obstacle_detect ();***
  Denne funksjonen detekterer hindringen og går til funksjonen ***avoid_obstacle ();*** slik at roboten kan bevege seg rundt og bak hindringen.
- ***if (obstacle_detect ()) avoid_obstacle();***
  Hvis avstandssensoren oppdager en hindring, vil funksjonen ***obstacle_detect ()*** returnere at en hindring er oppdaget (dvs. funksjonen returnerer true = 1). Når det er tilfelle vil funksjonen ***avoid_obstacle ();*** bli kalt opp slik at roboten går rundt og forbi. På baksiden må den på nytt lete etter den svarte stripen.

7 Oppgaver

7.1 Oppgave 1 - Nærmest målet

7.1.1 A - Nærmest målet i rett linje:



Oppgaven går ut på å bli kjent med roboten og finne ut hvor langt den går pr. tidsenhet. Oppgaven avsluttes med en konkurranse hvor dere fra et startpunkt skal komme nærmest et mål. Avstanden får dere oppgitt 5 min. før konkurransen starter.

1. **Kalibrer motorene** om ikke det alt er gjort, se informasjons-arket "Kalibrering av motorene"
   Bruk funksjonene:
   Calibration_LeftServo(); og ***Calibration_RightServo();*** for å kalibrere servomotorene
   SwitchDetect(); slik at dere kan starte og stoppe roboten med knappen
   Kommenter bort funksjonene når dere er ferdige

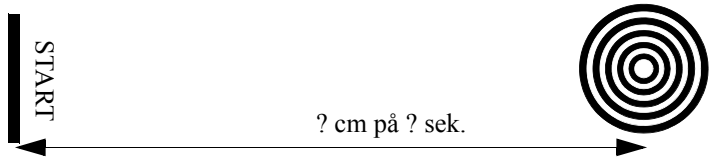
2. **Start og kjør servomotorene**
   Bruk funksjonen:
   MotorControl(Left_speed = <venstre verdi>***, Right_speed*** = <høyre verdi>***);***
   til å sette fart på høyre og venstre servomotor.
   <høyre/venstre verdi> kan ha verdier fra –100 til +100
   Funksjonen kan også brukes når roboten skal svinge til høyre og venstre:
   Se informasjons-arket "Oversikt over funksjoner".

3. **Kjøretid:**
   Bruk funksjonen:
   delay(<millisekunder>***);***
   Med denne funksjonen bestemmer dere hvor lenge foregående funksjon skal være aktiv før neste funksjon utføres.

4. **Start og stopp** av programmet:
   Bruk funksjonen:
   SwitchDetect();
   For manuell start og stopp av programmet
   stop_running ();
   Stopper programmet. Programmet startes på nytt ved å slå av og på strømmen eller trykke "Reset" (en liten rød knapp på topp-kortet).

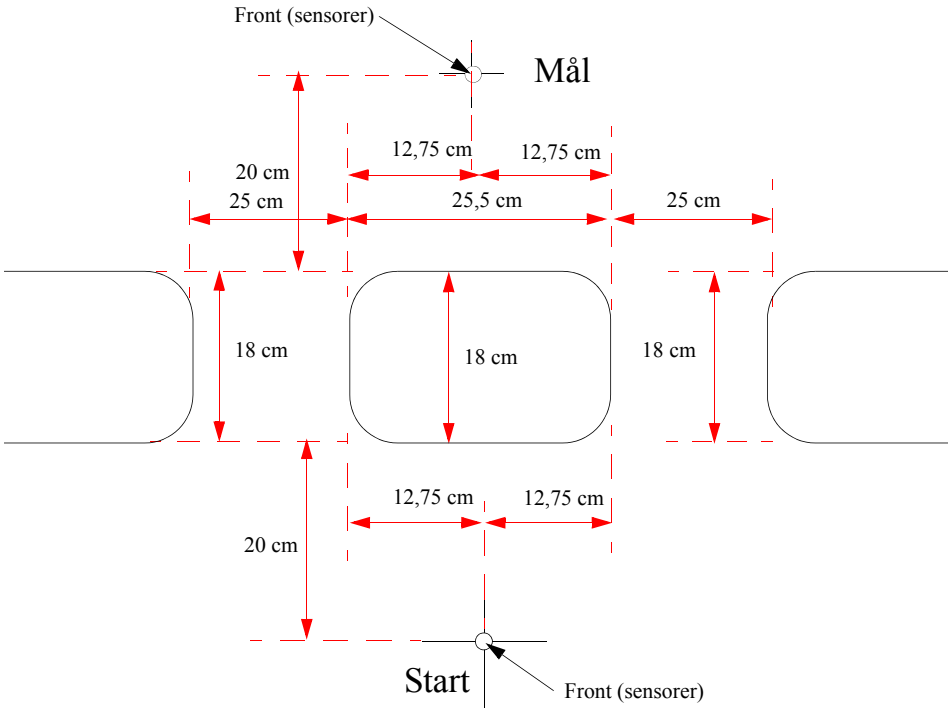
7.1.2 B - Nærmest målet på angitt tid:

Oppgave 1B er som oppgave 1A bare at nå skal dere komme nærmest målet innen en bestemt tid. Tid og ny avstand blir oppgitt 5 minutter før konkurransen.



7.2 Oppgave 2 - Nærmest målet bak hindring

Oppgaven går ut på å komme nærmest et mål som er skjult bak et hinder. Figuren under viser målene på banen som skal forsøres.



1. **Start og kjør servomotorene**
   Bruk funksjonen:
   MotorControl(Left_speed=-100, Right_speed=-100);
   til å sette fart på høyre og venstre servomotor.
   Funksjonen kan også brukes når roboten skal svinge til høyre og venstre:
   Se informasjons-ark "Oversikt over funksjoner"

2. **Kjøretid:**
   Bruk funksjonen:
   delay(<millisekunder>***);***
   Med denne funksjonen bestemmer dere hvor lenge foregående funksjon skal være aktiv før neste funksjon utføres.

3. **Start og stopp** av programmet
   Bruk funksjonen:
   SwitchDetect();
   For manuell start og stopp av programmet
   stop_running ();
   Stopper programmet. Programmet startes på nytt ved å slå av og på strømmen eller trykke "Reset" (en liten rød knapp på topp-kortet).

7.3 Oppgave 3 - Følg den svarte linjen

7.3.1 A - Raskest rundt

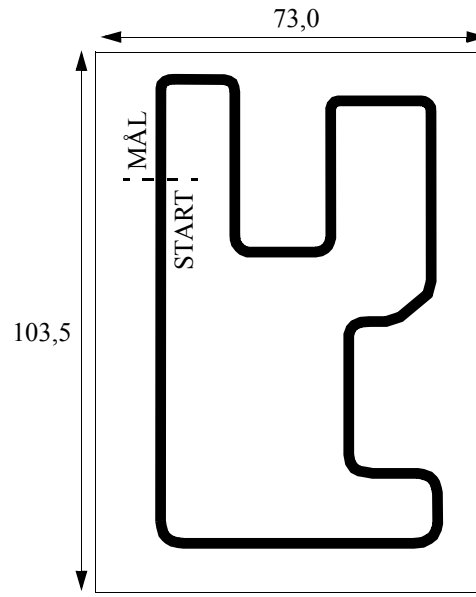
Oppgaven går ut på å følge den svarte linjen banen rundt på kortest mulig tid. Figuren til høyre viser formen på banen.

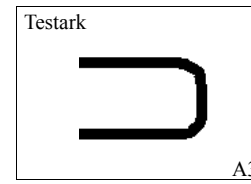
Følgende funksjoner vil være nyttige å bruke i denne oppgaven:

1. ***LineFollower (Speed =*** <fart verdi>***,***
   Threshold = <terskel verdi>***);***
   Følg kanten av svart linje.
   <fart verdi> = 1 - 100 –> hastighet
   <terskel verdi> = 0 - 1023 –> terskel for å skille mellom lyse (0) og mørke (1023) felt

2. ***SwitchDetect();***
   Programmet starter når man trykker på knappen på koblingsbrettet, og stopper når man trykker en gang til.

Tips: Bruk *testarket* for å teste ut sammenhengen mellom fart og robotens evne til å bevege seg rundt svingene.



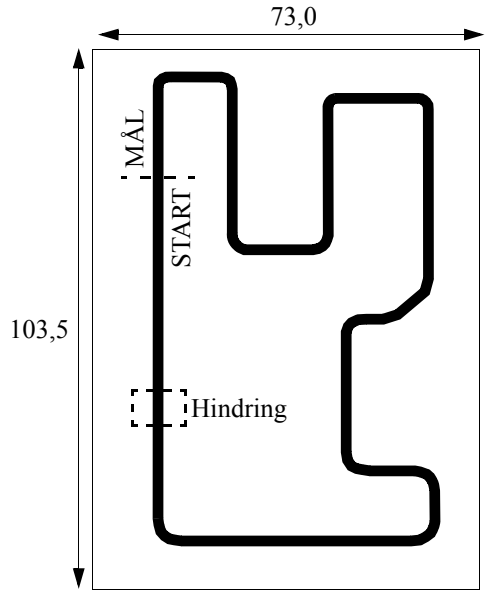


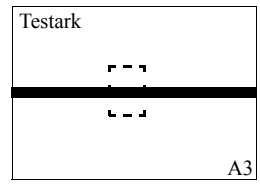
7.3.2 B - Raskest rundt og unngå hindring

Oppgaven går ut på å bevege seg raskest mulig langs den svarte linjen fra START til MÅL, detektere og omgå hindringen:

Følgende tilleggsfunksjoner vil være nyttige å bruke i denne oppgaven:

3. ***avoid_obstacle ();***
   Funksjonen må programmeres i henhold til størrelsen på hindringen som bør være kjent.
4. ***obstacle_detect ();***
   Denne funksjonen detekterer hindringen og går til funksjonen *avoid_obstacle ();* slik at en kommer seg rundt og bak hindringen.
5. ***if (obstacle_detect ()) avoid_obstacle();***
   Hvis avstandssensoren oppdager en hindring, vil funksjonen ***obstacle_detect ();*** returnere at en hindring er oppdaget (dvs. funksjonen returnerer true = 1). Når det er tilfelle vil funksjonen ***avoid_obstacle ();*** bli kalt opp slik at roboten går rundt og forbi. På baksiden må den på nytt lete etter den svarte linjen.

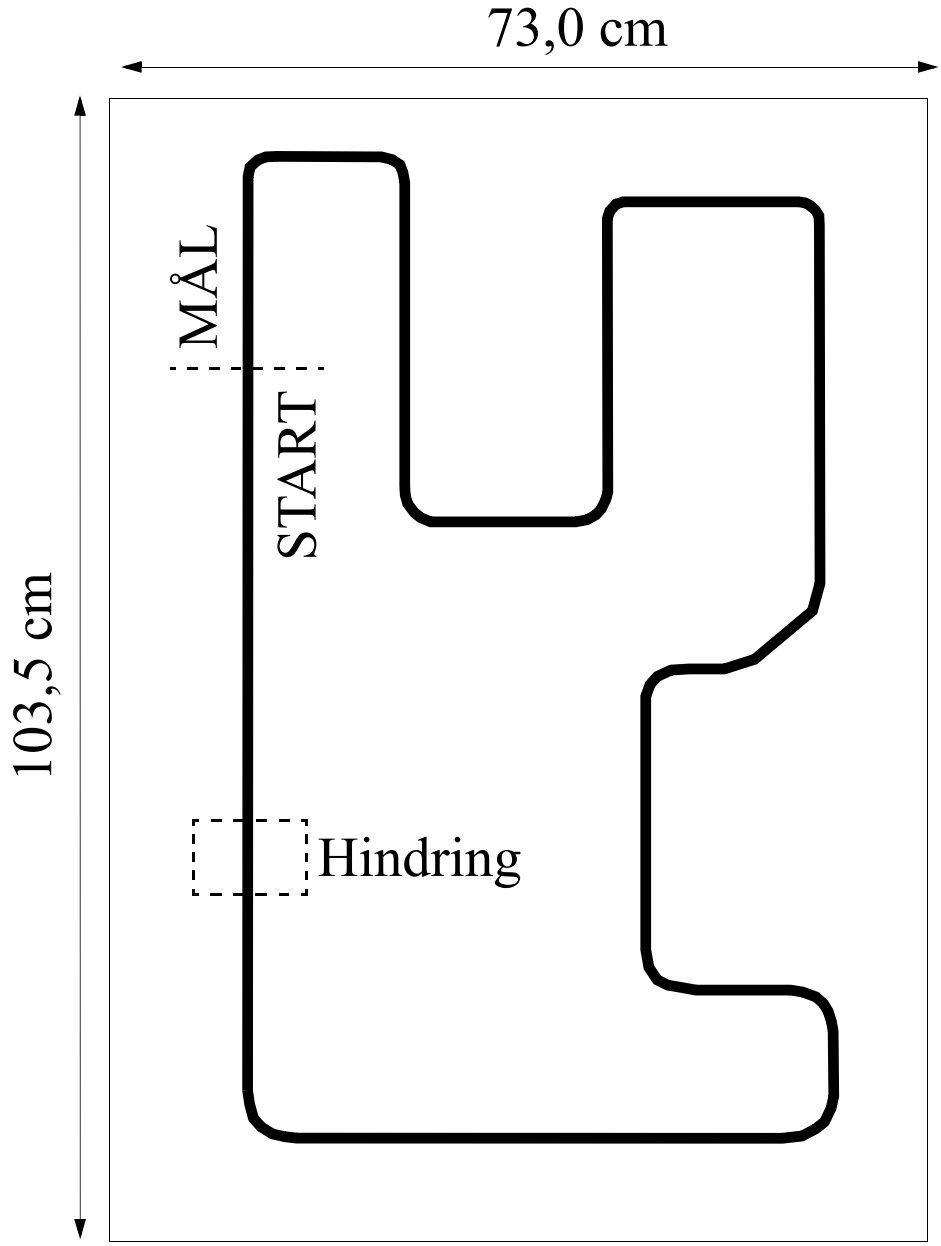




Vedlegg A Testarenaer

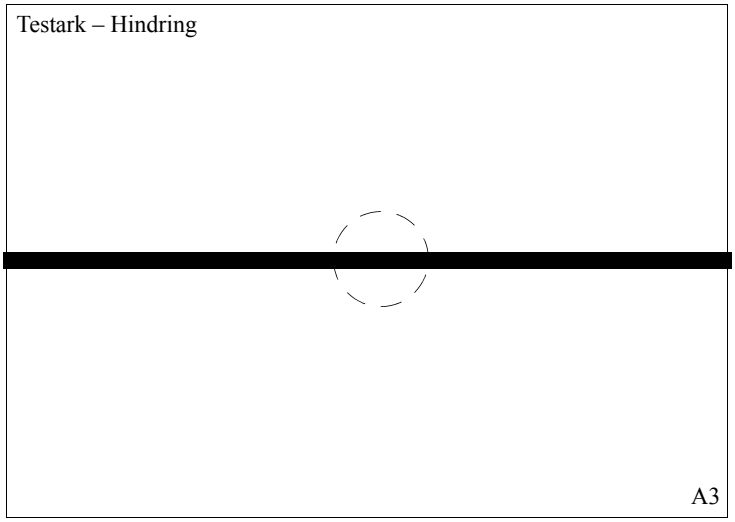
A.1 Følg linjen

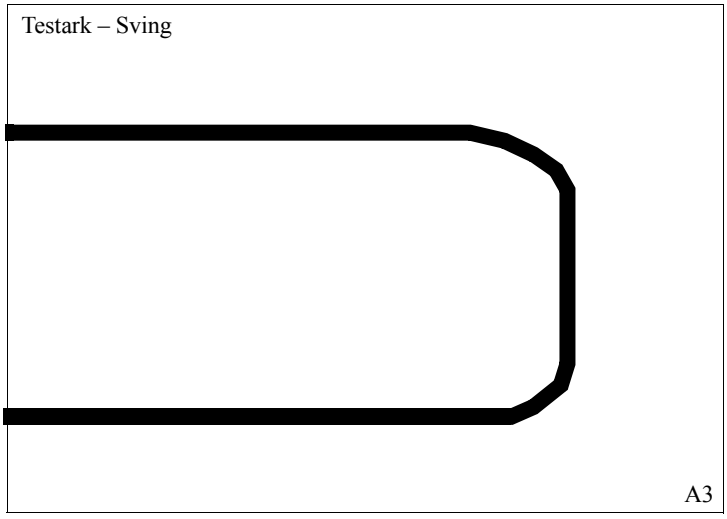
Bruk gjerne svart elektrikertape - f.eks. 3/4 Inch bred



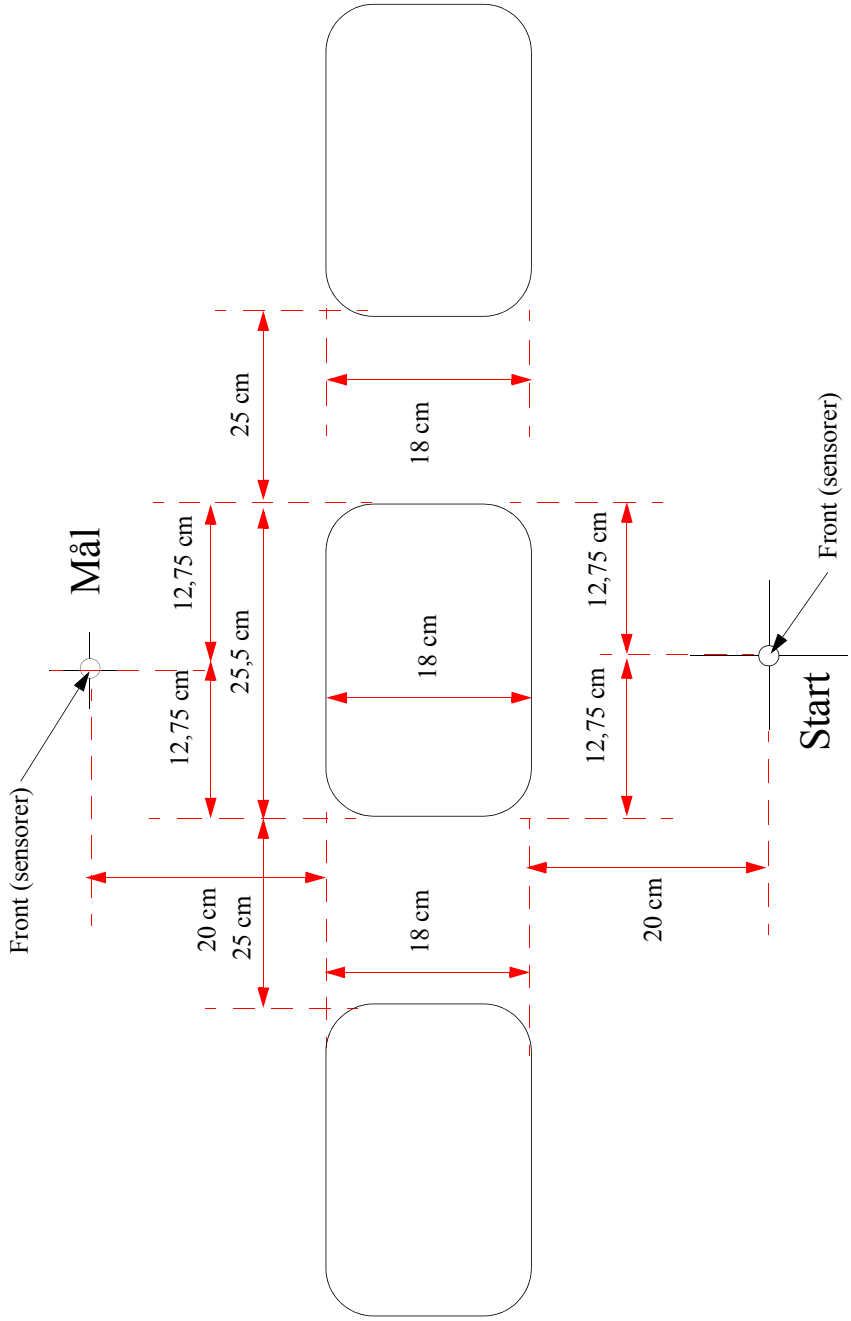
A.2 Testark A3

Bruk svart elektrikertape - f.eks. 3/4 Inch bred

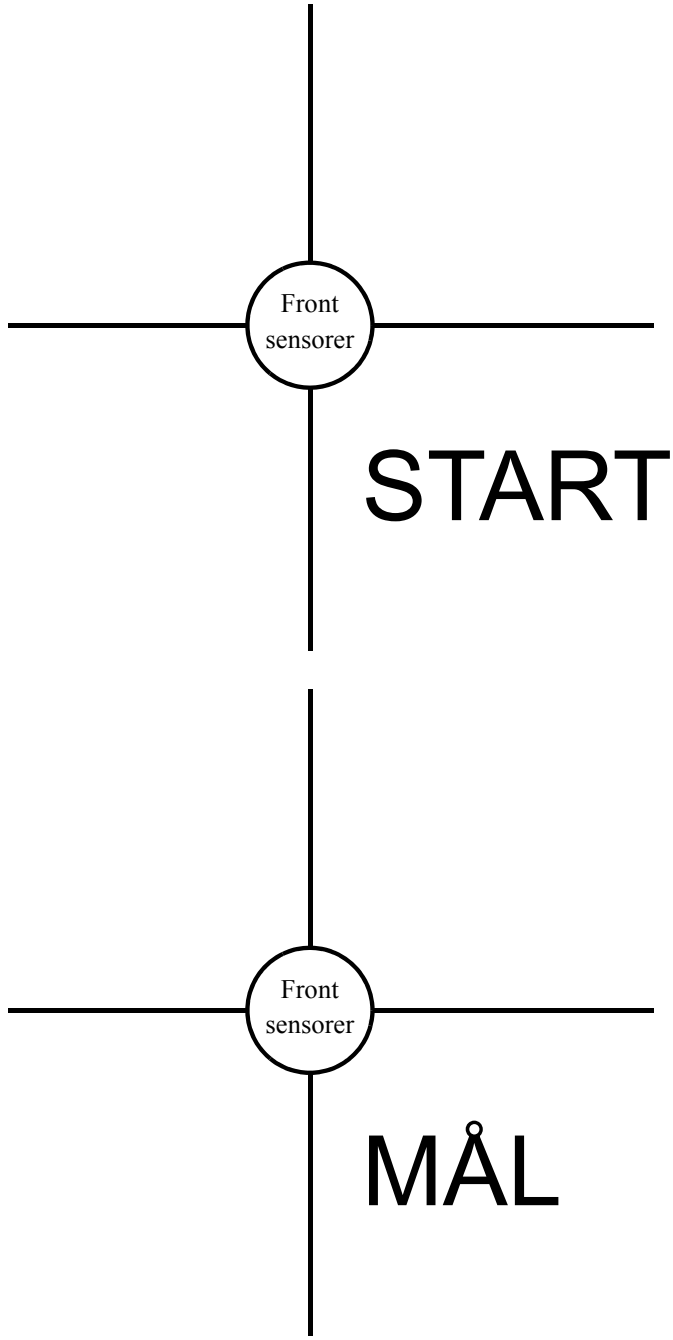




A.3 Arena med hindring



A.4 Start og stopp markering



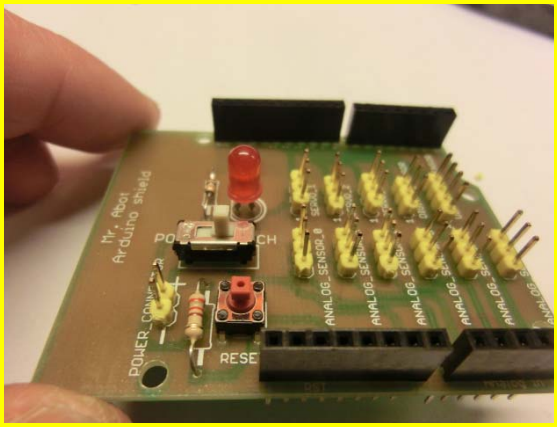
Vedlegg B Hjelpeark

Noter pulslengder for knekkpunkter. Bruk gjerne de typiske verdiene som utgangspunkt:

Venstre motor:		***Typiske verdier***
Forward_highspeed_left:	_____	*(1400)*
Forward_lowspeed_left:	_____	*(1490)*
Backward_lowspeed_left:	_____	*(1612)*
Backward_highspeed_left:	_____	*(1685)*

Høyre motor:		
Forward_highspeed_right:	_____	*(1615)*
Forward_lowspeed_right:	_____	*(1552)*
Backward_lowspeed_right:	_____	*(1433)*
Backward_highspeed_right:	_____	*(1350)*

Denne siden er blank



Heftet beskriver hvordan Mr. Abot kan kan brukes som aktivitet for ungdom i ungdomsskole og videregående. Opplegget er utviklet av ***Nils Kr. Rossing*** og IAESTE student ***Adam Gajda.***

Prosjektet har vært et samarbeid mellom Skolelaboratoriet ved NTNU, Atmel og Vitensenteret og er bl.a. finansiert av Atmel og Sparebankstiftelsen DNB NOR gjennom Breddegaveprosjektet.

ISBN 978-82-92088-53-1
Rev. 1.2 - 03.10.14