

# Guide til bruk av PCB Joystick

---

EN GUIDE TIL BRUK OG KALIBREIRNG



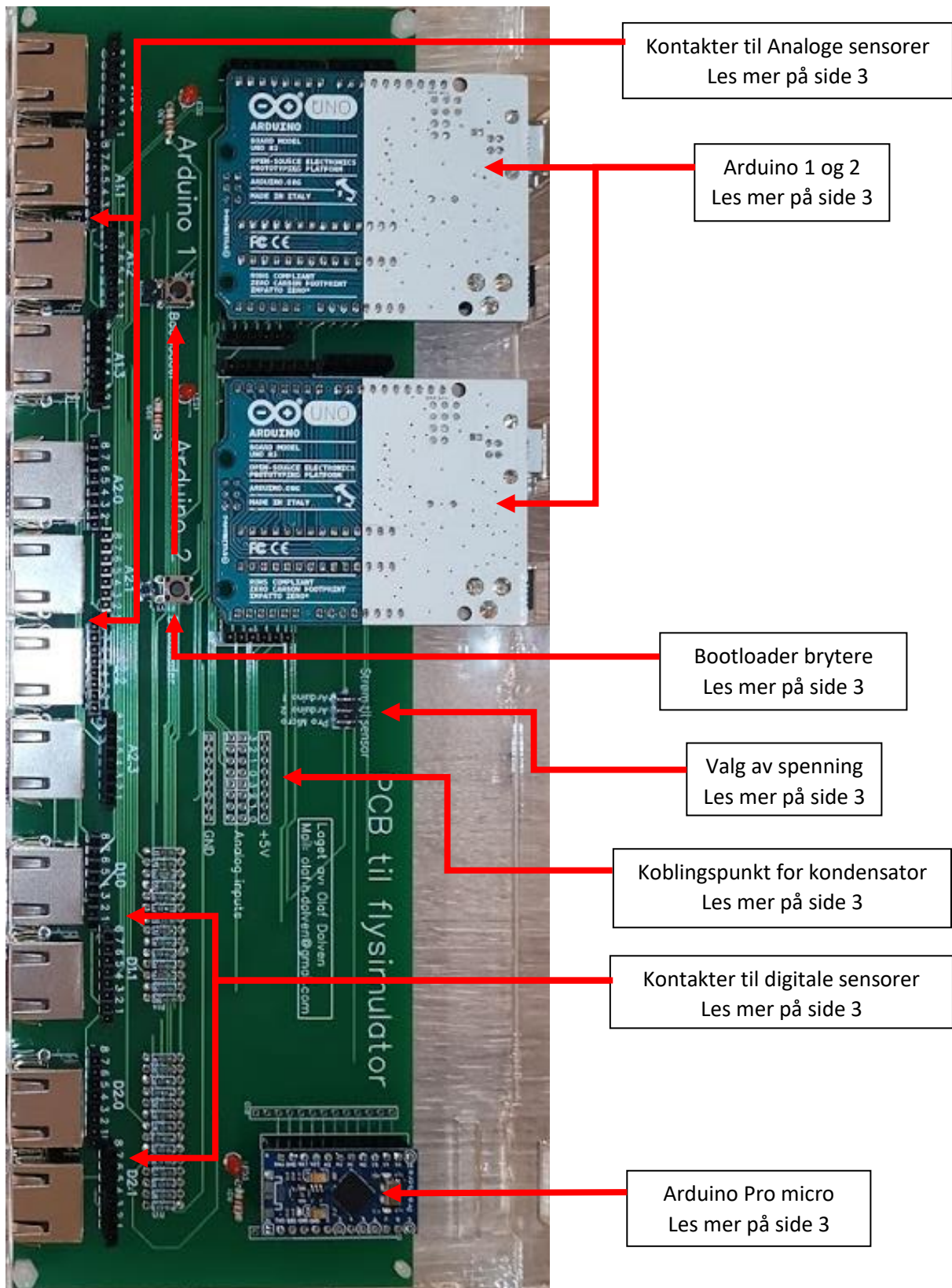
Skrevet av : Olaf Dolven  
Mail: [olaf.h.dolven@gmail.com](mailto:olaf.h.dolven@gmail.com)

## Innhold

PCBens komponenter .....	2
Kontakter til Analoge sensorer .....	3
Arduino 1 og 2 .....	3
Bootloader brytere .....	3
Valg av spenning .....	3
Koblingspunkt for kondensator .....	3
Kontakter til digitale sensorer .....	3
Arduino pro micro .....	3
Hjelp til oppkobling av sensorer .....	4
Analoge porter .....	4
Digitale porter .....	5
Kalibrering .....	6
Kalibrerings måter .....	6
Kalibrering i Windows .....	6
Kalibrering i Arduino kode .....	6
Kalibrering i Windows .....	7
Kalibrering i Arduino koden .....	8
Steg 1 - DFU mode .....	8
Steg 2 - Flashe Arduino Firmware .....	8
Steg 3 - Finne endepunkter .....	9
Steg 4 - Ny Joystick kode .....	10
Steg 5 - Flashe Joystick Firmware .....	10
Steg 6 - Ferdig .....	11

## PCBens komponenter

PCBen består av mange forskjellige komponenter, her er en kort forklaring av hver av dem.



Figur 1 Overblikk bilde av PCB

### Kontakter til Analoge sensorer

Dette er hvor du plugges inn kablene som går til de analoge sensorene. Som plugg har jeg valgt å bruke RJ45 (Dette er samme plugg som brukes til internett). Disse pluggene har 8 ledere. Bare 3 brukes om du bare har en analog sensor per plugg. Hvordan dette kobles opp kan du lese om på side 4

### Arduino 1 og 2

Dette er de to Arduinoen som brukes til å gjøre om sensordataen til joystick data som PCen forstår. Det brukes to stykker fordi det trengs for å ha nok analoge og digitale inputs. Disse trenger en kabel hver til PCen, her kan det eventuelt brukes en USB HUB.

### Bootloader brytere

Dette er bryterne som kan brukes til å sette Arduinoene til «**DFU mode**». Dette brukes under kalibrering.

### Valg av spenning

Her velger du hvor spenningen til sensorene skal komme fra, enten Arduino 1, Arduino 2 eller Arduino pro micro. Dette brukes om man eventuelt bare skulle brukt Arduino 2. Da må det settes en jumper her. Vanligvis skal denne stå på Arduino 1. Hver Arduino kan levere 500mA, dette skal være nok til sensorene.

Jeg anbefaler ikke å koble to av Arduinoen samtidig. Dette kan føre til at det går strøm om ikke spenningene er helt like. Ikke gjør dette om Arduinoene ikke er plugget i samme PC.

### Koblingspunkt for kondensator

Hvis det er elektrisk støy i sensorene så er det mulig å glatte ut dette med kondensatorer. I midten er det to pins for hver analog input. På hver side av dette er det +5V og GND. En kondensator loddas mellom +5V og sensoren, og mellom GND og sensoren. Dette vil fjerne noe støy

Om man bruker en for stor kondensator vil sensorene bli mindre responsive, jeg anbefaler å ikke bruke kondensatorer om du ikke må.

### Kontakter til digitale sensorer

Dette er hvor du plugges inn de digitale sensorene, dette er gjerne brytere eller lignende. Her brukes alle 8 pinsene. En pin til +5V og resten har en digital input hver. Du kan du lese om dette på side 5

### Arduino pro micro

Dette er en ekstra Arduino som er her om man trenger noen ekstra inputs eller noe annet. Arduinoen er ikke koblet til noen ting unntatt at den har samme 0V potensial som de andre Arduinoene.

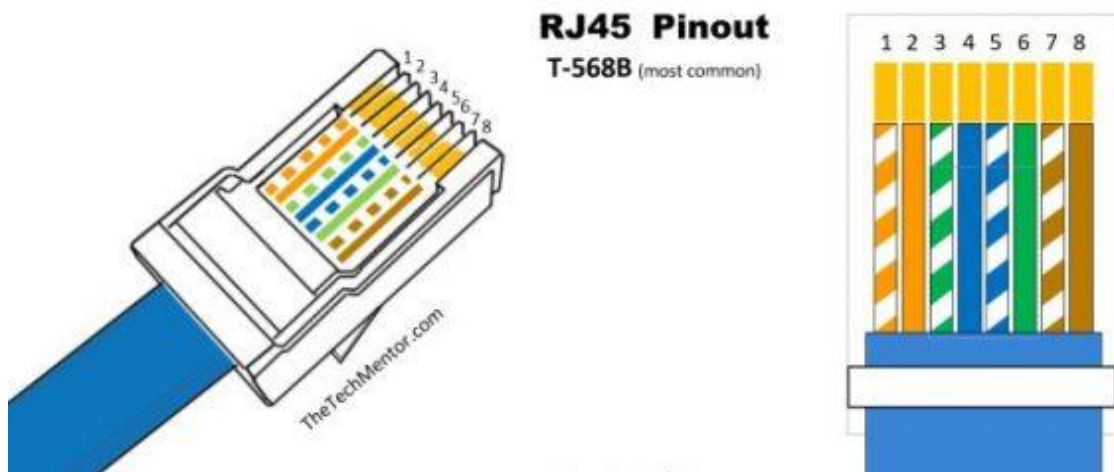
En Arduino Pro micro har muligheten til å emulere et tastatur. Dette betyr at man kan bruke denne til å styre en meny eller lignende. Men dette kan også gjøres med de vanlige Arduinoene ved hjelp av noen ekstra programmer.

## Hjelp til oppkobling av sensorer

Pluggene jeg har valgt å bruke på PCBen er RJ45 plugger. Disse pluggen brukes vanligvis med internettkabler. **IKKE KOBLE DENNE INN I EN RUTER ELLER LIGNDE.**

På PCBen er det to forskjellige typer plugger, A og D porter. A portene er til analoge sensorer og D portene er til Digitale sensorer

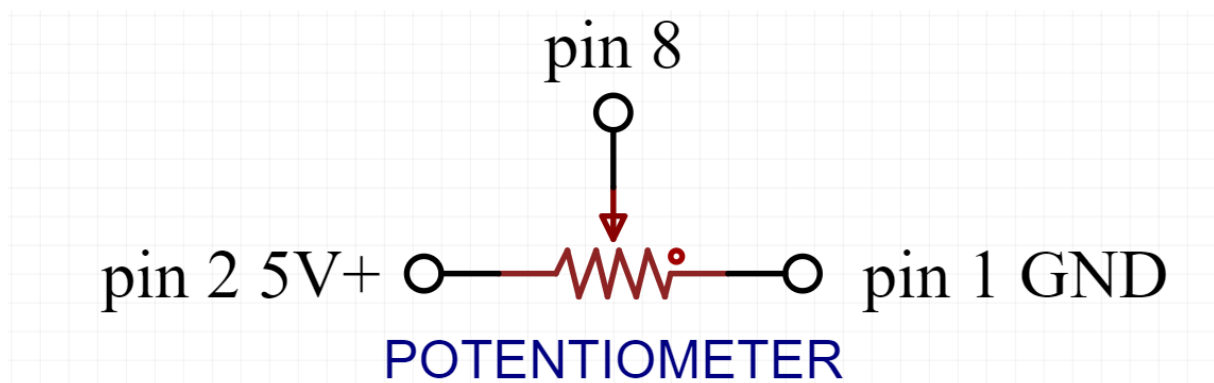
Når man skal koble kabler mellom PCBen og selve sensoren så er det lurt å bruke en standardisert fargerekkefølge. Jeg anbefaler å bruke T56B standard siden den blir brukt de fleste andre steder. Denne kan du se på bildet under.



Figur 2 T56B standardisert kobling

## Analoge porter

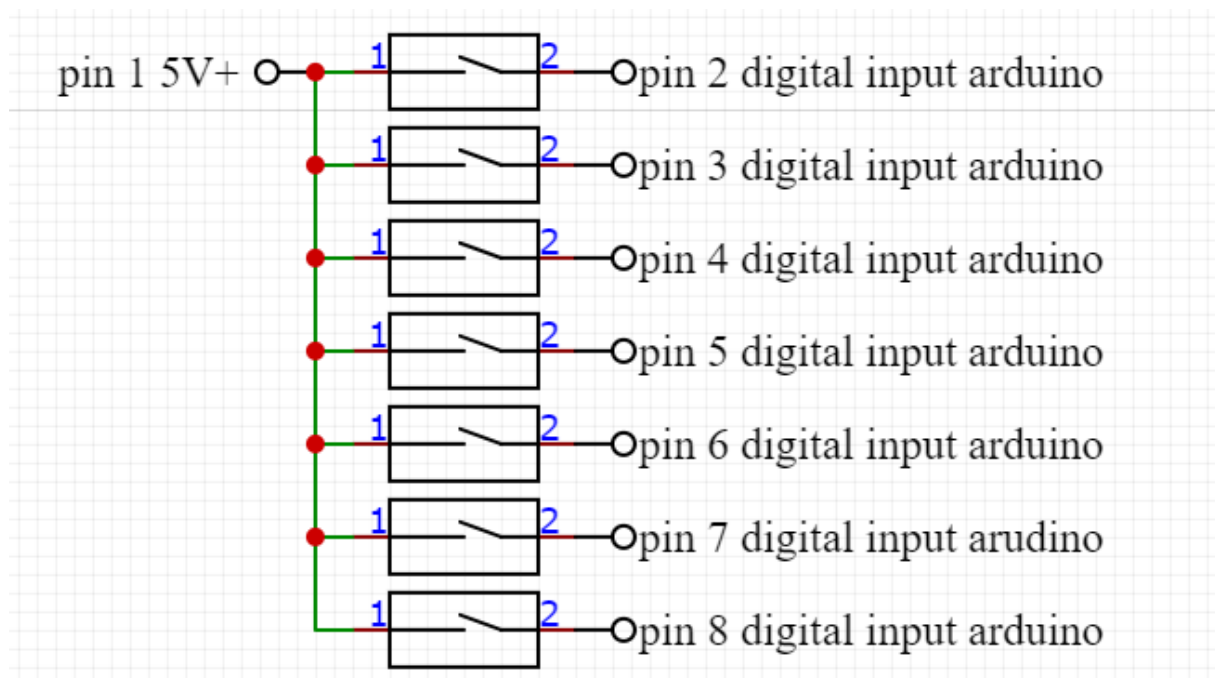
I de analoge portene så er pin nummer 1 koblet til GND og pin nummer 2 koblet til VCC (5V+). Pin nummer 8 er den som går til den analoge pinen på Arduinoen. Dette betyr at for å koble til et potensiometer så gjøres det slik:



Figur 3 Kobling av potensiometer

## Digitale porter

I de digitale portene så er pin 1 koblet til VCC 5V+. Alle de andre pinsene går til hver sin digitale input på Arduinoen i tillegg til en pull-down-resistor (10K ohm). For å koble brytere gjør slik:



Figur 4 Kobling av digitale inputs

## Kalibrering

Kalibrering av sensorer er viktig for at simulatoren skal fungere som den skal. Hvis Sensorene ikke er kalibrert, vill ikke kontrollene gi riktig utslag.

### Kalibrerings måter

For å få høyest mulig oppløsning og nøyaktighet er det to forskjellige måter man kalibrerer på:

#### Kalibrering i Windows

Denne kalibreringen skal gjøres når man merker at sensorene ikke har riktige endepunkt eller når de ikke sentrerer riktig. Denne gjøres inne i Windows sine innstillinger og er veldig lett å fullføre, men nøyaktigheten blir ikke den beste om man bare bruker denne metoden, derfor kalibreres det i Arduino koden en gang først. Kalibreringer gjort i ettertid kan gjøres i Windows.

#### Kalibrering i Arduino kode

Denne kalibreringen gjøres en gang når sensorene er koblet opp. Denne kalibreringen er mer teknisk og tar lengre tid, men vil få høyere nøyaktighet og oppløsning. Derfor gjøres denne kalibrering første gang simulatoren skal brukes.

## Kalibrering i Windows

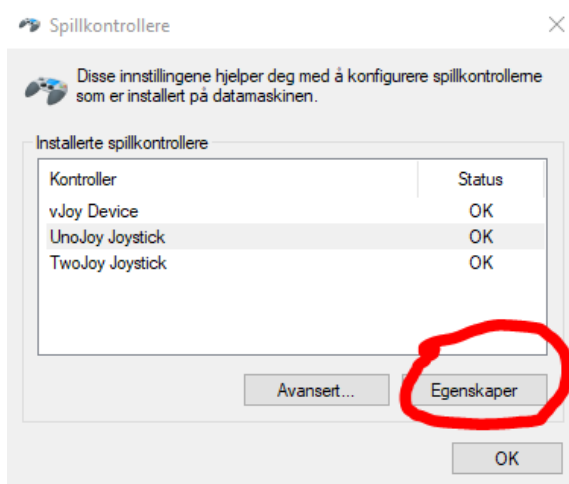
Kalibrering i Windows gjøres ved hjelp av Windows sitt kalibreringsverktøy for spillkontrollere. For å komme inn på dette går du inn på:

**Kontrollpanel → Maskinvare og Lyd → Enheter og Skrivere**

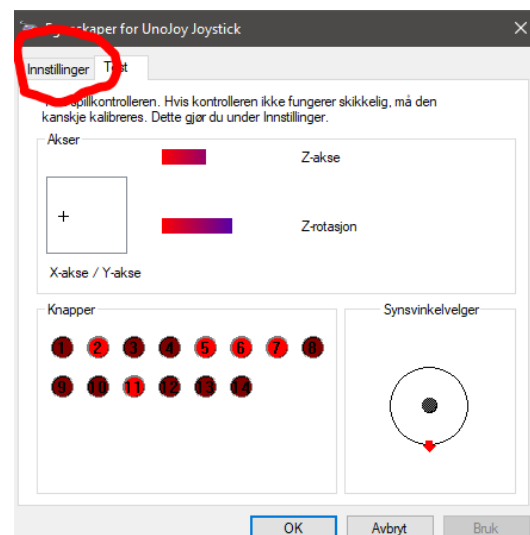
Her får du en liste med alle enheter tilkoblet til systemet. Du velger nå enten «**UnoJoy Joystick**» eller «**TwoJoy Joystick**». Hvis man skal gjøre en full kalibrering må det gjøres på begge enhetene.

Høyre klikk på enheten og trykk deretter på «**innstillinger for spillkontrollere**». Nå vil du få opp et nytt vindu. Her velger du igjen enheten du skal kalibrere, også trykker du **egenskaper**. Nå kommer det et nytt vindu. Her kan du se alle aksene og knappene koblet til denne enheten. For å kalibrere gå til fanen «**innstillinger**» øverst til venstre. Trykk så på knappen «**Kalibrer**». Det vil nå åpnes et nytt vindu med en guide til kalibrering. Gjør det guiden sier og enheten vil være ferdig kalibrert.

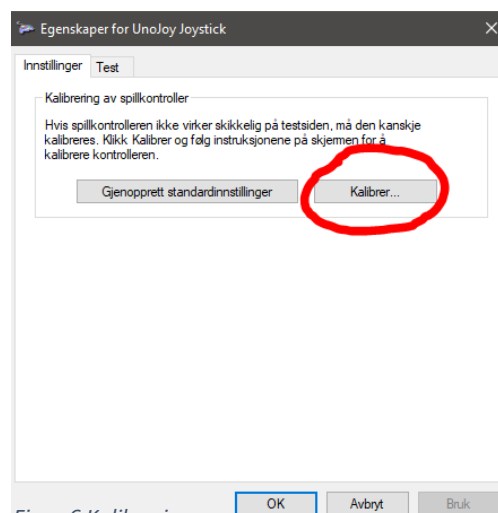
**Kontrollpanel → Maskinvare og lyd → Enheter og skrivere → høyre klikk på enheten → Innstillinger for spillkontrollere → velg enhet → egenskaper → innstillinger → kalibrer**



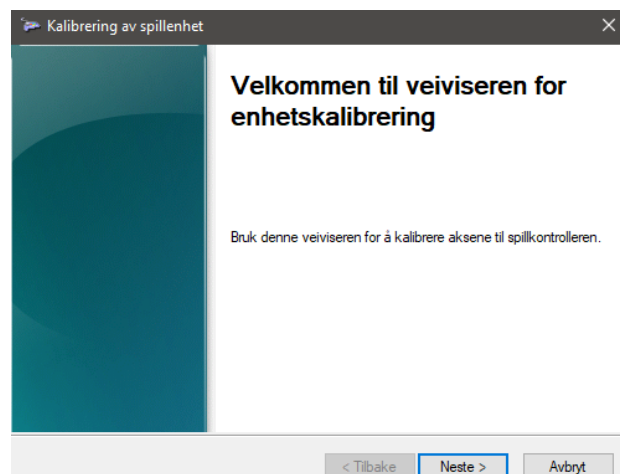
Figur 5 Liste over spillkontrollere



Figur 7 Visning av alle sensorer og brytere til kontrolleren



Figur 6 Kalibrering



Figur 8 Kalibrerings verktøyet innebygget i Windows

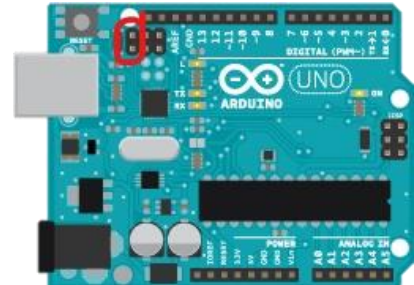


## Kalibrering i Arduino koden

Denne måten å kalibrere på har flere steg og krever litt mer data kunnskaper og litt programmeringskunnskaper. Å kalibrere på denne måten er bare nødvendig å gjøre første gang man setter opp simulatoren.

### Steg 1 - DFU mode

Når Arduinoen brukes som en joystick har de en annen **firmware** installert. Dette betyr at for å endre på Arduino koden må vi flash tilbake den **originale Arduino firmwares**. For å gjøre dette må Arduinoen settes i «**DFU mode**». For å gjøre dette må du kortslette de to metallpinnene øverst ved USB porten. Bruk en liten metallbit eller lignende for å gjøre dette. Når du har kortslettet så vil lysene på Arduinoen blinke og du kan fjerne kortslutningen (tar under et sekund). Om Arduinoen er koblet til PCBen så kan du også trykke på knappen som heter «**bootloader**»



Figur 9 pinnene som skal kortslettes

På PCen vil det nå stå at det er tilkoblet en «**ATmega16U2 DFU**».

### Steg 2 - Flashe Arduino Firmware

Nå skal vi flashe Arduinoen med den originale firmwares. For å gjøre dette gå i mappen tilhørende prosjektet

**Dokumentasjon-Simulator → Arduino (Velg 1 eller 2) → Conversion**

Her inne kjører du filen som heter «**TurnIntoAnArduino.bat**». Denne filen vil åpne ledeteksten. Når programmet har kjørt ferdig står det på bunn «**Press any key to exit**». Nå kan du lukke programmet.

Nå er Arduinoen flashet med den originale **Arduino Firmwares**. For at Arduinoen skal gå ut av «**DFU mode**» slik at vi kan programmere den må vi plugge den ut og inn igjen.

```
Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Reading bootloader version..... PASS
Erasing..... PASS
Selecting FLASH..... PASS
Blank checking..... PASS
Parsing HEX file..... PASS
Programming memory..... PASS
Verifying memory..... PASS
Starting Application..... PASS

Summary: Total 11 Passed 11 Failed 0
Now, you need to unplug the Arduino and plug it back in,
and it'll show back up as an Arduino. Press any key to exit....
```

Figur 10 Programmet som flasher Firmwares

### Steg 3 - Finne endepunkter

For å kalibrere sensorene må vi finne endepunktene. Nå som Arduinoen har den originale Bootloaderen så kan vi laste opp kalibrerings koden. Denne koden finner du i

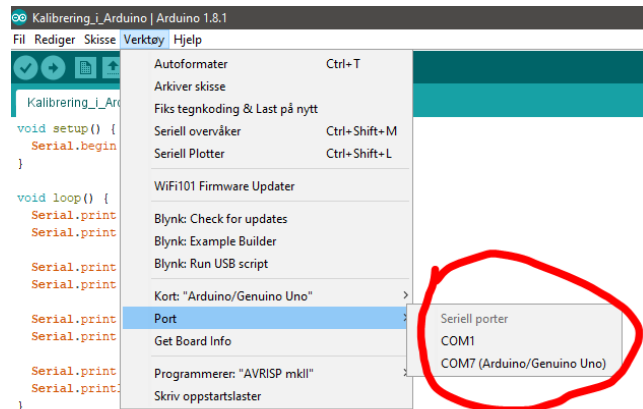
**Dokumentasjon-Simulator → Arduino (Velg 1 eller 2) → Kalibrering og Testing → Kalibrering\_i\_Arduino**

Dobbelklikk på «**Kalibrering\_i\_Arduino.ino**». Dette vil åpne Arduino IDEen. Gå opp i Verktøy menyen og sjekk at du har valgt riktig **COMport**.

Trykk deretter den blå pilen øverst til venstre eller trykk Ctrl+U. Dette laster opp koden til Arduinoen.

Når koden er ferdig lastet opp vil det stå «Opplasting ferdig» nede til venstre

Opplasting ferdig.



Figur 11 Velge COMport

Nå skal vi åpne Seriell overvåkeren. Dette er et program som viser all data sendt mellom Arduinoen og PCen. Mens du fortsatt er inne på Arduino IDEen trykk på den blå knappen øverst til høyre i programmet.



Dette vil åpne et nytt vindu. Her må du velge riktig **baudrate**. Dette gjør du nederst til venstre i rullegardinmenyen. Velg 9600 baud.

Nå vil du se mange linjer tekst som oppdaterer seg hele tiden. Dette er verdien til hver enkelt analog sensor. A10 er verdien til sensoren koblet til A1.0 på kretskortet.

```
A10 = 291  A11 = 292  A12 = 293  A13 = 287
A10 = 297  A11 = 290  A12 = 293  A13 = 291
A10 = 289  A11 = 295  A12 = 288  A13 = 294
A10 = 296  A11 = 290  A12 = 295  A13 = 286
A10 = 292  A11 = 293  A12 = 289  A13 = 296
```

Figur 12 Eksempel på data i Seriell overvåkeren til Arduino

Ta alle analoge sensorer (stikka Pedaler eller lignende) og dytt de til det ytterste punktet de klarer. Skriv så opp alle verdiene på en lapp eller lignende. Dette vil være det ene endepunktet (eller minste verdi til sensoren). Ta så alle sensorene til ytterste punkt. Skriv så opp alle verdiene igjen. Et eksempel ser du på høyre side. Dette er minste punkt og makspunkt til alle sensorene, disse verdiene skal brukes i joystick koden for å kalibrere den.

```
min A10 = 300
max A10 = 1012
```

```
min A11 = 400
max A11 = 800
```

```
min A12 = 250
max A12 = 678
```

```
min A13 = 0
max A13 = 1023
```

Figur 13 Eksempel på notering av endepunkter

#### Steg 4 - Ny Joystick kode

Nå som vi har funnet endepunktene så skal disse verdiene skrives inn i selve joystick koden. Denne joystick koden finner du i:

**Dokumentasjon-Simulator** → **Arduino** (Velg 1 eller 2) → **UnoJoy\_kode** (eller TwoJoy)

Åpne «UnoJoy\_kode.ino». Dette er selve koden som kjører på Arduinoen. Øverst i koden er det definert 8 «double» variabler, dette er endepunktene. Endre disse til de verdiene du skrev opp i steg 3. Eksempel:

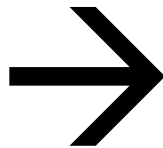
```
//      Kalibrering
// <-----
double minA10 = 0 - 10;
double maxA10 = 1023 + 10;

double minA11 = 0 - 10;
double maxA11 = 1023 + 10;

double minA12 = 0 - 10;
double maxA12 = 1023 + 10;

double minA13 = 0 - 10;
double maxA13 = 1023 + 10;
// <-----
```

Figur 15 Endepunkter i koden



```
//      Kalibrering
// <-----
double minA10 = 300 - 10;
double maxA10 = 1012 + 10;

double minA11 = 400 - 10;
double maxA11 = 800 + 10;

double minA12 = 250 - 10;
double maxA12 = 678 + 10;

double minA13 = 0 - 10;
double maxA13 = 1023 + 10;
// <-----
```

Figur 14 Endepunkter i koden etter endring

Trykk nå Ctrl+S for å lagre koden. Sjekk at du har valgt riktig **COMport** og last så opp koden (slik som du gjorde i «Steg 3 - Finne endepunkter»)

Når koden er ferdig lastet opp så må Arduinoen flashes med Joystick firmwaren.

#### Steg 5 - Flashe Joystick Firmware

For å flashe firmwaren må du først sette Arduinoen i «**DFU mode**» igjen. Hvordan man gjør dette er beskrevet i «Steg 1 - DFU mode». Det er viktig at du gjør dette først før du går videre med steg 5

Programmet for å flashe Arduinoen finner du i:

**Dokumentasjon-Simulator** → **Arduino** (Velg 1 eller 2) → **Conversion**

Her inne kjører du filen som heter «**TurnIntoAJoystick.bat**». Denne filen vil åpne ledeteksten. Når programmet har kjørt ferdig står det på bunn «**Press any key to exit**». Nå kan du lukke programmet.

Nå er Arduinoen flashet med **Joystick Firmwaren**. For at Arduinoen skal gå ut av «**DFU mode**» slik at den kan brukes som en joystick må vi plugge den ut og inn igjen.

```
Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Reading bootloader version..... PASS
Erasing..... PASS
Selecting FLASH..... PASS
Blank checking..... PASS
Parsing HEX file..... PASS
Programming memory..... PASS
Verifying memory..... PASS
Starting Application..... PASS

C:\Users\zipol\Documents\GitHub\Flysimulator\Full kode til flyet\Arduino 2\Conversion
Running batchisp 1.2.5 on Sat Feb 08 13:59:45 2020

ATMEGA16U2 - USB - USB/DFU

Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Reading bootloader version..... PASS
Erasing..... PASS
Selecting FLASH..... PASS
Blank checking..... PASS
Parsing HEX file..... PASS
Programming memory..... PASS
Verifying memory..... PASS
Starting Application..... PASS

Summary: Total 11 Passed 11 Failed 0
Now, you need to unplug the Arduino and plug it back in,
and it'll show back up as an Arduino. Press any key to exit....
```

Figur 16 Programmet som flasher Firmwaren

### Steg 6 - Ferdig

Nå er Arduinoen kalibrert og fungerer som en vanlig joystick igjen. Det er lurt å teste at endepunktene er riktig ved hjelp av «**JoytsickTest.exe**». Dette er et program som ligger i «**Dokumentasjon-Simulator**» mappen.