



ABEL
PRISEN

Nils Kr. Rossing

Servicehefte – “Ut og fly – størst verdi”



Denne siden er blank

Servicehefte – “Ut og fly – størst verdi”

Nils Kr. Rossing

Servicehefte – “Ut og fly – størst verdi”

Trondheim 2015

ISBN 978-82-92088-55-5

Bidragstyttere:

Nils Kr. Rossing, (nkr@vitensenteret.com) Vitensenteret i Trondheim

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 1.4 – 15.06.15



ABEL
PRISEN

Prosjektet er gjennomført i med støtte fra Niels Henrik Abels Minnefond.



Forord

Heftet gir en kortfattet orientering om bruk og vedlikehold av installasjonen “Ut og fly – størst verdi” som er utviklet som et delprosjekt under Abel-fond-prosjektet som startet i 2011 og avsluttes ved årsslutt 2015. Utvikling av rike interaktive installasjoner for bruk ved Vitensenter, har vært et samarbeid mellom Vitensenteret i Trondheim og VilVite i Bergen. I løpet av prosjektet er det utviklet to interaktive modeller ved VilVite og to ved Vitensenteret i Trondheim (ViT). Det har vært et krav at utstillingene skal handle om matematikk og utviklet i tråd med intensjonen om “Active prolonged engagement”, et prinsipp utviklet ved Exploratorium i San Fransisco.

ViT har utviklet modellene “Nærmest null”, hvorav åtte er levert i løpet av februar og mars 2015, og fire “Ut og fly – størst verdi” som er levert i løpet av juni 2015. Prototypene er utviklet ved ViT av **Nils Kr. Rossing**, **Steinar Løken-Olsen**, **Henning Lundh** og **Frøde Willmann**. Science Project Ltd. London ved **Guy Griffin** har utviklet og produsert de endelige kabinetene som er sendt til Trondheim og montert og testet av **Torgeir Rossing** og **Nils Kr. Rossing**.

Dette heftet gir en første innføring i bruk og vedlikehold av “Ut og fly – størst verdi”. Heftet vil bli oppdatert med tips til feilsøking etter hvert som vi får større erfaring med modellene i utstilling. Det vil også bli opprettet en hjemmeside hvor siste utgave av heftet vil kunne lastes ned.

Vitensenteret

Juni 2015

Nils Kr. Rossing





Innhold

1	Innledning	9
2	Montering, oppstart og service	11
2.1	Montering	11
2.2	Oppstart og reset	12
2.3	Lyssetting	12
2.4	Gjenstands-brikkene	12
2.5	Oppdatering av programvare	12
3	Kort beskrivelse av installasjonens oppbygning	15
3.1	Kontrollenhet	15
3.2	Lysensorene (LDR)	15
3.3	Displaykortene	15
3.4	Strømforsyningen (Power supply)	15
3.5	Vifta	16
4	Justering og feilretting	16
4.1	Feil belysning - justering av terskelnivå.	16
4.2	Endre på tid før tømning	16
5	Bilder av interiør	17
	Vedlegg A Styreprogram	20



1 Innledning

Kapittelet gir en kort beskrivelse av utstillingsmodellen “Ut og fly – størst verdi”.



Man skal ut å reise med fly og skal pakke kofferten. Det er mange ting man ønsker å ta med seg, men man må gjøre et utvalg siden det er 20 kg vektbegrensning på fly. En velger da å ta med seg de gjenstandene som gir størst verdi innen vektbegrensningen på 20 kg. I alt har man 11 gjenstander med ulik vekt og verdi som man skal velge blant.

Hver gjenstand (brikke) er merket med både vekt og verdi. Luftbordet er merket med den enkeltes gjenstands omriss, vekt og verdi.

Ved å trykke SJEKK-knappen på panelet i lokket av kofferten så får man en kort (5 sek.) statusrapport om hvordan man ligger an. Dette vises på en lysdioderekke til høyre på panelet. Grensen for de ulike statusverdiene er satt som følger:

<i>Kjempebra</i>	= 1700	Grønn (maksimal verdi)
<i>Bra</i>	=/ > 1600 og < 1700	Gul eller Grønn
<i>Kan bli bedre</i>	=/ > 1500 og < 1600	Gul eller Grønn
<i>Mye å gå på</i>	=/ > 1400 og < 1500	Gul eller Rød
<i>Prøv igjen</i>	< 1400	Rød

Dersom man trykker SJEKK-knappen mens det er overvekt vil alle lysdiodene blinke i ca. 5 sekunder.

Dersom brikkene på bordet ligger urørt på luftbordet i mer enn 60 sekunder vil bordet automatisk sørge for tømning. Brikkene skal da falle ned i oppsamlingsrommet foran. For å hindre tømning er det nok å flytte på en brikke slik at det skjer en forandring mht. vekt og verdi.



Hvordan går man frem for å løse oppdraget?

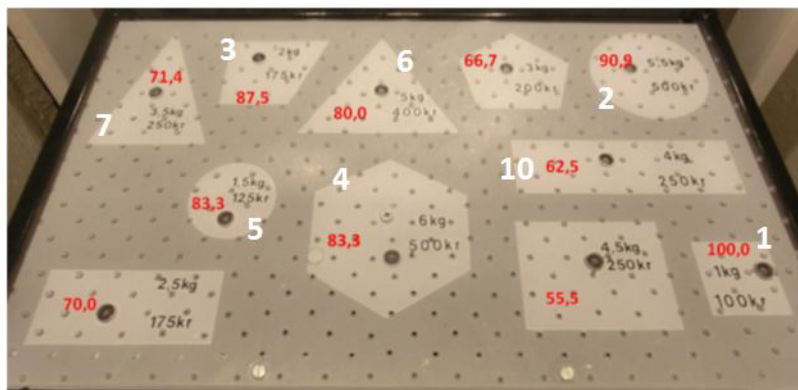
Dette er en typisk optimaliseringsoppgave hvor publikum må finne en strategi. En nærliggende strategi er å velge de gjenstandene som har størst verdi pr. vektenhet. Det betyr at deltagerne må gjøre et overslag over forholdet mellom verdi og vekt for hver gjenstand. Figuren under viser en oversikt over gjenstander og forholdstall for verdi og vekt.

Objects	Weight [kg]	Worth [NOK]	Worth/Weight	Choice	Total wight [kg]	Total worth [NOK]
A	4	250	62,50		0	0
B	6	500	83,33	1	6	500
C	2	175	87,50	1	2	175
D	5,5	500	90,91	1	5,5	500
E	5	400	80,00	1	5	400
F	1	100	100,00		0	0
G	3	200	66,67		0	0
H	2,5	175	70,00		0	0
I	1,5	125	83,33	1	1,5	125
J	4,5	250	55,56		0	0
K	3,5	250	71,43		0	0
				5	20	1700

Det finnes en optimal løsning som gir en maksimal verdi i kofferten på kr. 1700,- med 5 gjenstander på til sammen 20 kg.

Figuren under antyder en strategi for å komme fram til maksimal verdi.

... å velge de brikkene med størst egenverdi (kr/kg)



1 = 100kr/1kg 1+2 = 600kr/6,5kg 1+2+3 = 775kr/8,5kg 1+2+3+4 = 1275kr/14,5kg
1+2+3+4+5 = 1400kr/16kg 1+2+3+4+5+6 = **1800kr/21kg** 1+2+3+4+5+10 = **1650kr/20kg BRA**
2+3+4+5+6 = **1700kr/20kg MAKS**

Man velger gjenstander i henhold til hvilke som har størst egenverdi. Dette går greit helt til man har valgt de 5 mest verdifulle gjenstandene 1+2+3+4+5 (16 kg). Dersom man nå velger den 6. mest verdifulle gjenstanden vil man passere vektgrensen og får overvekt (21 kg). Da kan man gjøre et kompromiss ved å velge den gjenstanden som fyller opp til vektbegrensningen. Dette er gjenstand 10 (4.0 kg 250 kr.). En oppnår da et resultat som er BRA (20,0 kg, 1650,- kr.), men ikke MAKS.

For å oppnå maksimal verdi så må man ta ut gjenstand 1, som har høyest egenverdi, da kan man legge opp i gjenstand 6 (5,0 kg, 400 kr.) som gir 20,0 kg og 1700,- kr. i verdi.

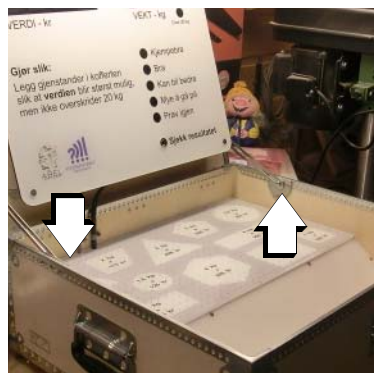
2 Montering, oppstart og service

2.1 Montering

Lokket er festet til sideveggene i kofferten ved hjelp av to avstivinger. Stram skruene slik at lokket blir stående i riktig stilling. **VIKTIG! Stram skruene (tork) slik at det ikke er fare for at lokket uforvarende faller ned å lukker kofferten og knuser små fingre (se piler for lokalisering av skruer på figuren til høyre).**

Det følger med to teleskopiske bein til kofferten slik at den kan skrånstilles. Dette er nødvendig for at det skal være mulig å tømme bordet effektivt ved hjelp av trykkluft.

Juster teleskopbeina slik at den svarte og grå delen blir ca. 9 cm lange og skru dem inn i bakkant av kofferten i særskilte hull for dette formålet (se bildet under). Test om helningen er stor nok slik at gjenstandene glir av bordet ved tømning. Testing gjøres ved at modellen startes, man legger på gjenstander og venter ca. 1 min. til vifta starter. Om de glir lett av så er bordet justert riktig. Om de blir liggende så økes lengden på teleskopbeina slik at helningen på bordet blir større. Det er sannsynligvis mest kritisk om tømning skal lykkes, når det er få brikker som ligger på bordet (f.eks. den minste brikken ligger igjen alene). Om denne ikke skulle gli av så er det jo imidlertid ingen krise da en slik konfigurasjon vil neppe være til noen stor hjelp for neste bruker.

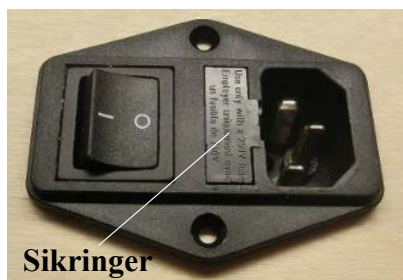


Dersom helningen på bordet er for stor, vil brikkene gli av uten trykkluft. Sørg for brikkene ligger stabilt når luftstrømmen ikke er aktiv.

Brikkene legges i oppsamlingsrommet i forkant med teksten opp slik at alle gjenstandene er lett synlig med vekt og verdi.



2.2 Oppstart og reset



Modellen startes ved å sette stikkontakten i veggen og slå på hovedbryteren som er bak til høyre. Mellom bryteren og nett-tilkoblingen sitter sikringsholderen. Ved skifte av sikring vippes holderen ut med et skrujern og sikringene erstattes med *trege* 1 A sikringer (5 x 20 mm). Det er viktig at de er trege, da raske sikringer kan ha problemer med å håndtere startstrømmene til strømforsyningen.

2.3 Lyssetting

Siden hver gjenstand registreres ved at optiske sensorer blir tildekket, er det viktig at man bruker litt ekstra omtanke på belysning av modellen. Det er ønskelig at modellen belyses med så diffust lys som mulig, dvs. at det dannes minst mulig skarpe skygger når noen rekker armen ut over kofferten. Dette kan lett oppnås ved å belyse bordet fra både høyre og venstre side av modellen. Det trengs ikke særlig sterkt lys for at modellen skal fungere tilfredsstillende.

NB! Ikke bruk lysstoffrør, disse blinker ca. 100 ganger i sekundet. Det er derfor sannsynlig at en eller flere av sensorene kalibreres idet lysstoffrøret er “mørkt”.

Deteksjonen av gjenstandene skjer ved at en optisk sensor (LDR) midt under brikken detekterer at den er lagt på plass slik at den skygger for lyset. Ved oppstart av modellen vil alle sensorer bli automatisk kalibrert mht. den belysningen modellen har. **NB! Det er viktig at luftbordet er tømt når modellen slås på, og at ingenting skygger for belysningen av luftbordet. Dette gjelder bare idet modellen slås på.**

Dersom det oppstår problemer med at gjenstander som er lagt på bordet ikke registreres eller at skyggen fra en hånd gir uønsket registrering av gjenstand, så kan man vurdere å justere terskelnivået for deteksjon (se avsnitt 4.1), ev. justere belysningen av bordet.

2.4 Gjenstands-brikkene

Det følger med 11 gjenstands-brikker (eller bare gjenstander eller brikker) merket med vekt og verdi. Brikkene er malt med svart lakk for å gi bedre marginer med hensyn til optisk deteksjon. Lakken er en to-komponent type som skal være spesielt egnet for tøff bruk.

Dersom noen av brikkene forsvinner ta kontakt med Vitensenteret i Trondheim. Blir de svært slitt kan man legge på et nytt lag med lakk. En bør imidlertid sørge for at lakkflaten er mest mulig plan slik at luften løfter brikken.

2.5 Oppdatering av programvare

Siden dette er første gang utstyret er i salg, vil programvaren sannsynligvis ikke være optimal til tross for at installasjonen er langtidstestet i utstillingen ved Vitensenteret i Trondheim. Ta kontakt med Vitensenteret og beskriv ev. problemer som er oppstått (nkr@vitensenteret.com). Oppgradert



programvare kan da oversendes og installeres av lokale teknikere. Vi vil også legge ut programvaren på nettsiden for modellene.

Nedlasting og installasjon av programvare

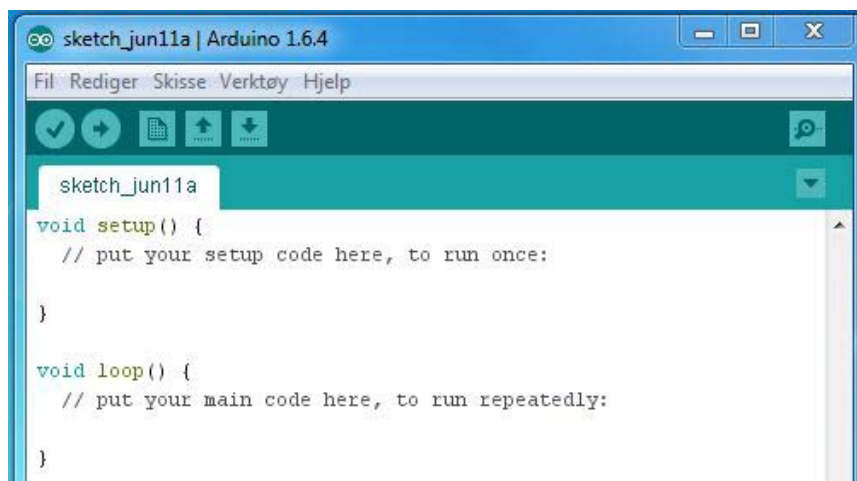
Det er programeditoren til Arduino som benyttes. Denne hentes ned fra

<http://arduino.cc/en/pmwiki.php?n=main/software>

pakkes ut og installeres på en PC eller MAC. Aktuell versjon er i skrivende stund Arduino 1.6.4.

Slå på PC og plugg USB-kablene fra modellen inn i en USB-port. Kabelen for installasjonen skal være festet til høyre under panelet. Dra kabelen ut av hullet og koble den til en av USB-inngangene til PC'en/MAC'en. Om du er heldig vil den automatisk be deg installere drivere. Følg da anvisningen.

For å sjekke om driverne er installert, kan du starte Arduino-editoren ved å velge ikonet vist til høyre. Man kommer da inn i editoren og får opp følgende vindu (bare øverste del er vist i figuren under).

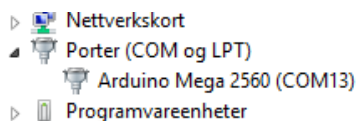


Ved å velge *Verktøy (Tools)* fra menylinjen vil man få flere valg, deriblant *Kort (Board)* og *Port (Serial Port)*.

Kort – velg: *Arduino Mega or Mega 2560*

Port – velg: *COM <høyeste nr.>*

Dersom driverne ikke er installert riktig vil man hos PC'er bare finne *COM 1* eller ingen (Serial Port er "grått"). I så tilfelle kan man gå til *Kontrollpanel/System/Enhetsbehandling* her skal det komme opp et alternativ i listen som heter *Porter*. Velger man denne vil man se noe i likhet med det som er vist på figuren over:

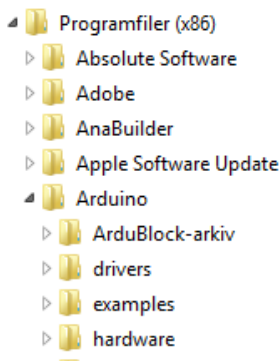




Hvor Arduino Mega 2560 er tildelt en COM-port (f.eks. COM13), eller man skal få varsel om at det er oppdaget en ukjent port. I så fall høyre-klikker man på den ukjente porten og velger:

Oppdater driver programvare

... og man kommer inn i programvaren for installasjon av drivere. Her kan det være lurt å *peke på driverprogramvaren* i stedet for å velge *Automatisk*. Driverne ligger under Programfiler (x86)/Arduino/Drivers



For ytterligere hjelp gå til følgende side:

<http://www.arduino.cc/en/Guide/UnoDriversWindowsXP>

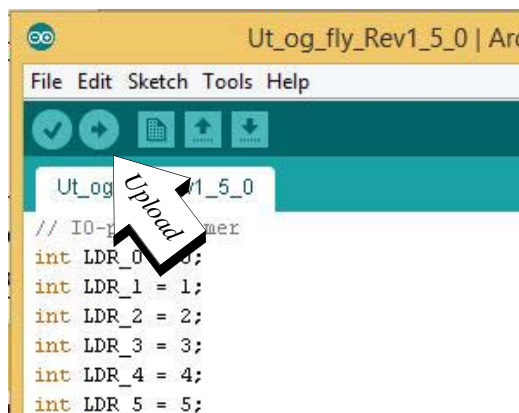
Installer ny programvare i “Ut og fly - størst verdi”

Programfilen for innlasting vil ha navnekonvensjonen:

Ut_og_fly_<Rev nr.>.ino den medfølgende er Ut_og_fly_<Rev nr.>.ino

For tiden er *Ut_og_fly_Rev 1.5.0.ino* gjeldende revisjon.

Programfilen hentes inn i Arduino editoren (IDE) ved å velge *File/Open*. Deretter kompiles og lastes filen over til den interne Arduino MEGA ved å velge *File/Upload* (pil mot høyre).

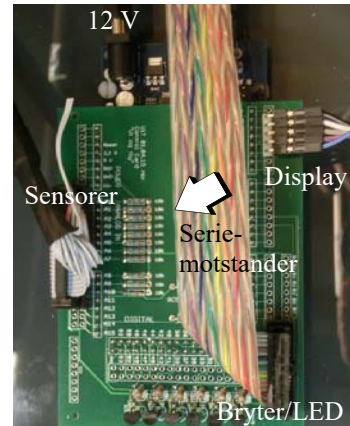


Programvaren kan lastes opp mens modellen er avslått siden Arduino Mega kortet får spenning via USB-kabelen. Den kan selvfølgelig også lastes opp mens modellen er påslått.

3 Kort beskrivelse av installasjonens oppbygning

3.1 Kontrollenhet

Kontrollenheten består av et Arduino MEGA 2650 kort med et skreddersydd shield-kort montert på toppen. Shield-kortet er forsynt med tre flatkabel tilkoblinger: Sensorer, Display og kabel til Bryter/LED. I tillegg tilføres Arduino-kortet 12 V like-spennning. Om nødvendig kan shield-kortet løftes av hovedkortet for inspeksjon. Ved montering er det viktig å sjekke at alle pinnene er kommet i riktig hylsekontakt (svart). Inspeksjon kan gjøres med et speil.



3.2 Lyssensorene (LDR)

Under hver gjenstands-markering på luftbordet er det montert en lyssensor. Denne ligger inne i en plastsyylinder limt fast på baksiden av bordet med dobbeltsidig tape. Lyssensorene er tilknyttet shield-kortet via den “rund” flatkabelen (merket Sensorer). LDR'en danner en spenningsdeler sammen serie-motstandene på shield-kortet.



Hver LDR er montert på et lite kretskort som vist i figuren til venstre. Kortet sitter fast til sensorholderen ved at beina på LDR'en er loddet til kortet. Ved bytte av sensor løsnes holderen fra plata og beina loddes fra kortet. LDR'en kan så tas ut på framsiden av LDR-holderen.

3.3 Displaykortene

Displaykortene er utviklet av Science Project Ltd. Kortene er koblet i kjede slik at dataene sendes inn på seriell form i kortet lengst mot venstre for så å “riple” over til kortet til høyre (slik kortene sees når panelet er løsnet og ligger over luftbordet).

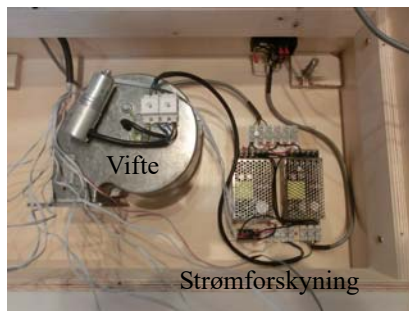
3.4 Strømforsyningen (Power supply)

Strømforsyningen består av en 5 V forsyning som kan levere inntil 1,5 A og en 12 V forsyning som kan levere inntil 3,0 A. Arduino Mega forsynes med 12 V som reguleres ned til 5 V på kortet. Displaykortene forsynes med 5 og 12 V. Sammen med strømforsyningen er det montert et rele som slår av og på vifta.



3.5 Vifta

Vifta er ei standard sentrifugalvifte som trekker inn luft fra undersiden. Det er derfor viktig at modellen har “fri luftveier” fra undersiden og ut i lokalet der modellen står.



4 Justering og feilretting

Dette kapitlet oppsummerer kjente problemer som kan oppstå. Kapitlet vil bli supplert etter som det ev. skulle dukke opp nye feil under bruk:

4.1 Feil belysning - justering av terskelnivå.

Ved uheldig belysning kan en få skarpe skygger fra en arm som strekkes ut mot panelet for å trykke på SJEKK-knappen. Dette vil i så fall kunne endre på resultatet og dermed også gi feil status.

Problemet løses enklest ved å sørge for å belyse luft-bordet slik at det ikke oppstår så skarpe skygger.

I enkelte spesielle tilfeller kan det være nødvendig å justere *terskelmarginen* i programmet. Lys-sensorene som ligger under luftbordet gir verdier fra 0 (svært lyst) til 1023 (helt mørkt). Ved oppstart måles lysintensiteten for hver enkelt sensor. Terskelverdien er den verdien der lysdetektoren detekterer en gjenstand. Terskelmarginen (LimMargin) er differansen mellom verdi som indikerer gjenstand på plass og nivå målt uten gjenstand. Dersom denne er for liten så vil modellen kunne registrere skygger fra f.eks. en arm, som en gjenstand.

For justering av terskelmargin (LimMargin), gjør slik:

1. Last opp programmet i IDE og identifiser følgende kodelinje:
int LimMargin = 300; // Her kan terskelnivået for deteksjon justeres
2. Dersom enkelte brikker ikke detekteres når de ligger på bordet, forsøk å *reduser verdien*
3. Dersom skyggen fra en hånd detekteres, forsøk å *øk verdien*

4.2 Endre på tid før tømning

Dersom det ikke har skjedd noe på bordet i løpet av 60 sekunder så vil vifta slås på og bordet tømmes. Om denne tiden oppleves for kort (eller for lang) kan den lett endres i programmet.

Finn parameteren TimeLimit i programmet (bruk søkefunksjonen). Default verdi er 150 som tilsvarer ca. 60 sek. Øke eller minke med 50 for å øke eller minke med 20 sek,

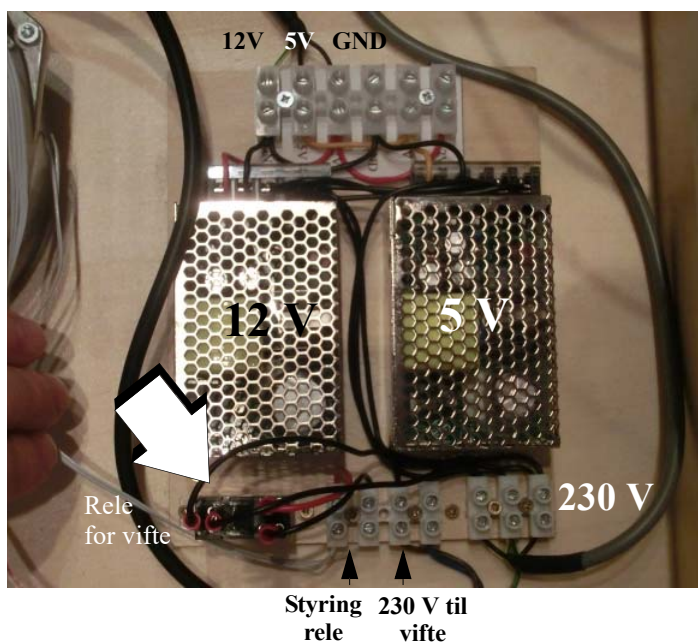
Etter endringen lastes programmet inn og modellen er klar for bruk.

5 Bilder av interiør

Nett-tilkobling og sikringer.



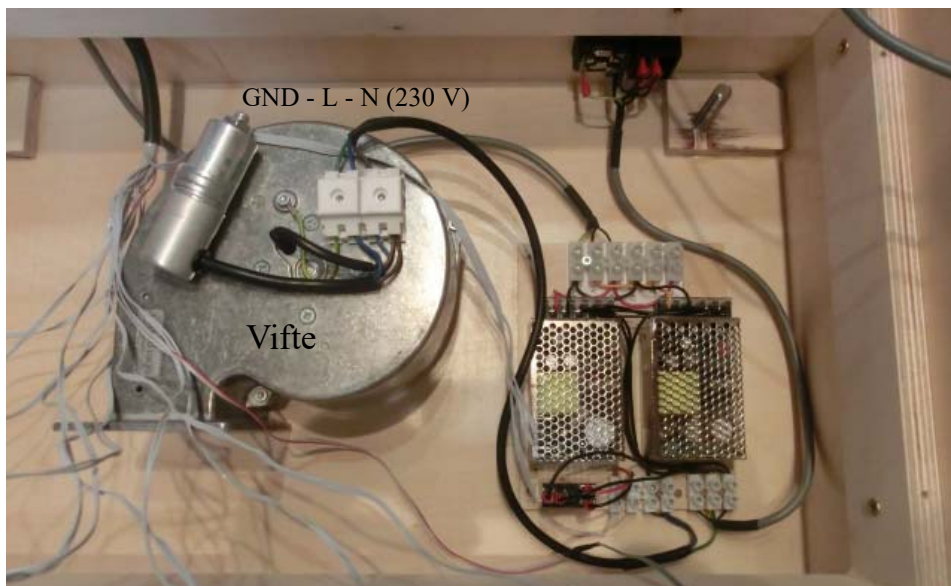
Power supply



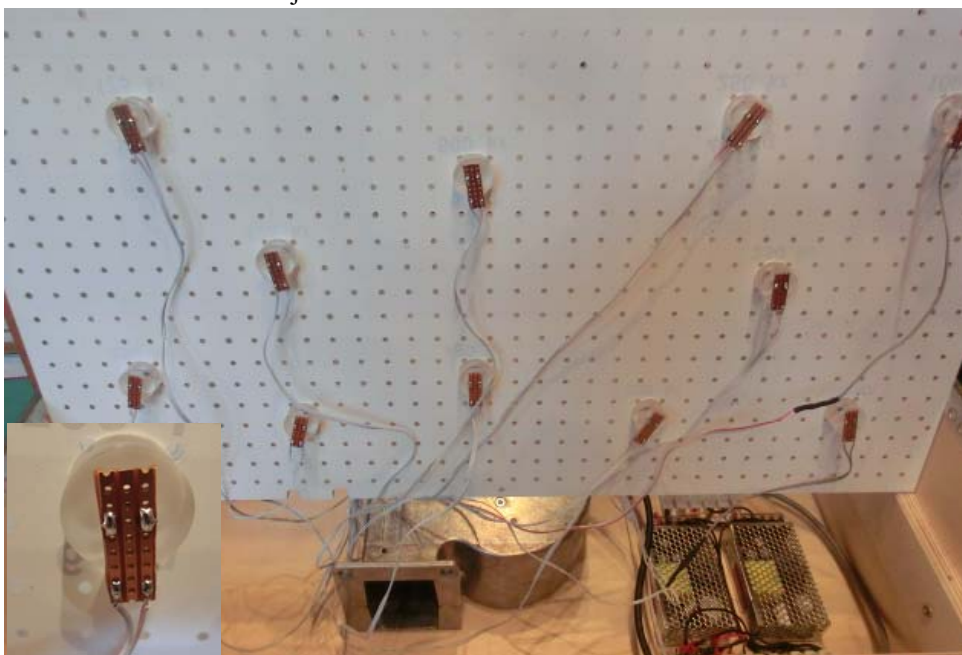
Vi ser at det her er et sett med rekkeklemmer for tilkobling av GND, 5 V og 12 V. Normalt vil det kun være ett sett.



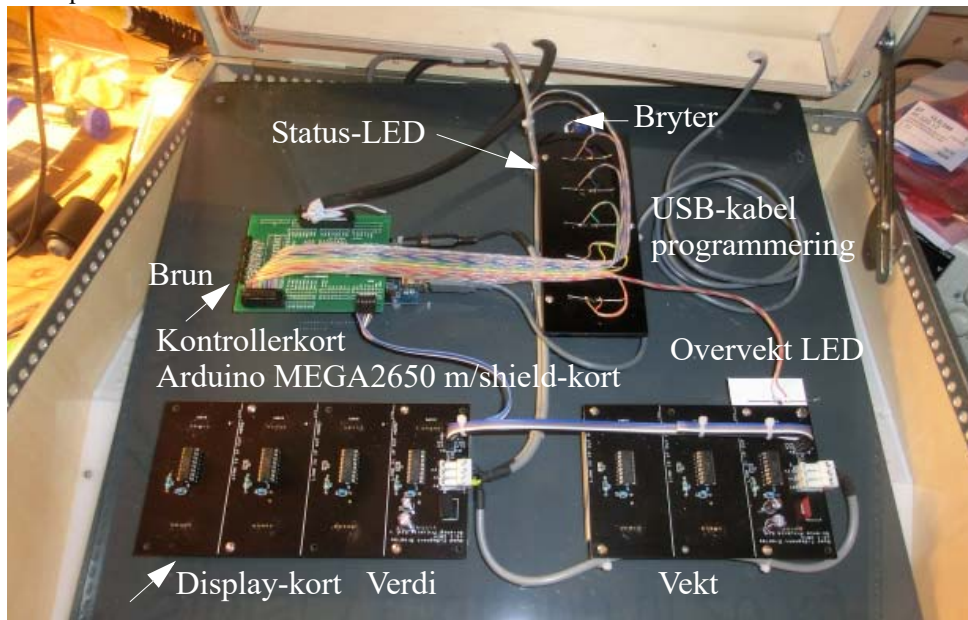
Sentrifugalvifte - legg spesielt merke til hvilke konnektorer som skal forbindes med 230 V og jord.



Sensorer. Bildet viser påmonterte lyssensorer under hver av markeringene av gjenstandene på luftbordet. Nederst i venstre hjørne er vist et nærbilde av sensoren montert i holderen.



Kontrollpanel.



Bryter/LED flatkabelen er koblet opp slik (brun lengst mot venstre). Hvert par består av en farget ledning i henhold til standard fargekoding og en grå ledning. Merk at den fargede ledningen er pluss 12 V, mens den tilhørende grå ledningen er koblet til kollektoren på drivertransistoren via en seriemotstand:

- Brun/Grå -> Bryter
- Rød/Grå -> *Overvekt* (Rød lysdiode)
- Oransje/Grå -> *Kjempebra* (Grønn lysdiode)
- Gul/Grå -> *Bra* (Gul (Grønn) lysdiode)
- Grønn/Grå -> *Kan bli bedre* (Gul (Grønn) lysdiode)
- Blå/Grå -> *Mye å gå på* (Gul (Rød) lysdiode)
- Grå/Grå -> *Prøv igjen* (Rød lysdiode)



Vedlegg A Styreprogram

// IO-port nummer

```
int LDR_0 = 0;
int LDR_1 = 1;
int LDR_2 = 2;
int LDR_3 = 3;
int LDR_4 = 4;
int LDR_5 = 5;
int LDR_6 = 6;
int LDR_7 = 7;
int LDR_8 = 8;
int LDR_9 = 9;
int LDR_10 = 10;
```

// Flagg for posisjoner som er tildekket

```
int LDR_0_F = 0;
int LDR_1_F = 0;
int LDR_2_F = 0;
int LDR_3_F = 0;
int LDR_4_F = 0;
int LDR_5_F = 0;
int LDR_6_F = 0;
int LDR_7_F = 0;
int LDR_8_F = 0;
int LDR_9_F = 0;
int LDR_10_F = 0;
```

// Avleste verdier for lyssensorer

```
int LL_0;
int LL_1;
int LL_2;
int LL_3;
int LL_4;
int LL_5;
int LL_6;
int LL_7;
int LL_8;
int LL_9;
```



```
int LL_10;

// Vekt, verdi og terskelverdier
int W_1_0 = 10;
int V_1_0 = 100;
int Lim_1_0 = 200; //750; // Terskelverdi LDR - Vekt 1,0 kg, verdi 100 kr.
int W_1_5 = 15;
int V_1_5 = 125;
int Lim_1_5 = 200; //900; // Terskelverdi LDR - Vekt 1,5 kg, verdi 125 kr.
int W_2_0 = 20;
int V_2_0 = 175;
int Lim_2_0 = 200; //900; // Terskelverdi LDR - Vekt 2,0 kg, verdi 175 kr.
int W_2_5 = 25;
int V_2_5 = 175;
int Lim_2_5 = 200; //950; // Terskelverdi LDR - Vekt 2,5 kg, verdi 175 kr.
int W_3_0 = 30;
int V_3_0 = 200;
int Lim_3_0 = 200; //600; // Terskelverdi LDR - Vekt 3,0 kg, verdi 200 kr.
int W_3_5 = 35;
int V_3_5 = 250;
int Lim_3_5 = 200; //900; // Terskelverdi LDR - Vekt 3,5 kg, verdi 250 kr.
int W_4_0 = 40;
int V_4_0 = 250;
int Lim_4_0 = 200; //700; // Terskelverdi LDR - Vekt 4,0 kg, verdi 250 kr.
int W_4_5 = 45;
int V_4_5 = 250;
int Lim_4_5 = 200; //950; // Terskelverdi LDR - Vekt 4,5 kg, verdi 250 kr.
int W_5_0 = 50;
int V_5_0 = 400;
int Lim_5_0 = 200; //600; // Terskelverdi LDR - Vekt 5,0 kg, verdi 400 kr.
int W_5_5 = 55;
int V_5_5 = 500;
int Lim_5_5 = 200; //600; // Terskelverdi LDR - Vekt 5,5 kg, verdi 500 kr.
int W_6_0=60;
int V_6_0 = 500;
int Lim_6_0 = 200; //900; // Terskelverdi LDR - Vekt 6,0 kg, verdi 500 kr.
int LimMargin = 400; // Her kan terskelmarginen for deteksjon justeres
```



```
// Variable som holder de ulike
// sifferne som skal vises på display
int Siffer_1_kg = 0;
int Siffer_10_kg = 0;
int Siffer_100_kg = 0;
int Siffer_1_kr = 0;
int Siffer_10_kr = 0;
int Siffer_100_kr = 0;
int Siffer_1000_kr = 0;

// Variable for ulik bruk
int Total_Weight = 0;
int Total_Verdi = 0;
int Old_Weight = 0;
volatile int OnTime = 0;
int vent = 100;

int pinCLK = 13;
int pinSDI = 12;
int pinLE = 11;
int pin_OE = 10;
int pinSpare = 2;
int pinCheck = 3;

int Check;

volatile int Overweight = 33;
volatile int Run_flag = LOW;

int Best = 31;
int Best_Verdi = 1700;
int Nest_Best = 29;
int Nest_Best_Verdi = 1600;
int Bra = 27;
int Bra_Verdi = 1500;
int Daarlig = 25;
int Daarlig_Verdi = 1400;
int Daarligst = 23;
```



```
int Vifte = 22;

int Rest = 0;
int TimeLimit = 150; // 50 tilsvare ca. 20 sek. 150 tilsvare ca. 60 sek.
int ActiveFlagg = 0;
int BlinkTid = 2;
int Sjekk = 0;

volatile char n, m;
volatile int i;
int ant_loop = 0;
int Sif[7];
byte SifCon[10];

void setup()
{
  pinMode(Daarligst, OUTPUT); // Pinne 23 (Rød)
  pinMode(Daarlig, OUTPUT); // Pinne 25 (Gul)
  pinMode(Bra, OUTPUT); // Pinne 27 (Gul)
  pinMode(Nest_Best, OUTPUT); // Pinne 29 (Gul)
  pinMode(Best, OUTPUT); // Pinne 31 (Grønn)
  pinMode(Overweight, OUTPUT); // Pinne 33 (Rød)
  pinMode(Check, INPUT); // Pinne 8
  pinMode(pinCLK, OUTPUT); // Pinne 13
  pinMode(pinSDI, OUTPUT); // Pinne 12
  pinMode(pinLE, OUTPUT); // Pinne 11
  pinMode(pin_OE, OUTPUT); // Pinne 10
  pinMode(Vifte, OUTPUT); // Pinne 22
  pinMode(pinSpare, INPUT); // Pinne 2 (Interrupt 0 inngang SPARE)
  pinMode(pinCheck, INPUT); // Pinne 3 (Interrupt 1 inngang CHECK)

  attachInterrupt(1, Status, FALLING); // Attach interrupt to pin 3

  digitalWrite(pin_OE, LOW);
  digitalWrite(pinLE, LOW);

  SifCon[0] = B11111100; // 0 (ABCDEFG DP)
  SifCon[1] = B01100000; // 1 (ABCDEFG DP)
```



```
SifCon[2] = B11011010; // 2 (ABCDEFG DP)
SifCon[3] = B11110010; // 3 (ABCDEFG DP)
SifCon[4] = B01100110; // 4 (ABCDEFG DP)
SifCon[5] = B10110110; // 5 (ABCDEFG DP)
SifCon[6] = B10111110; // 6 (ABCDEFG DP)
SifCon[7] = B11100000; // 7 (ABCDEFG DP)
SifCon[8] = B11111110; // 8 (ABCDEFG DP)
SifCon[9] = B11110110; // 4 (ABCDEFG DP)
```

```
Serial.begin(9600);
```

```
CalibrationLDR();
```

```
}
```

```
void loop() {
```

```
    // Sjekk om sensor er tildekket,
```

```
    // legg til eller trekk fra riktig vekt og verdi
```

```
    // Serial.print("LOOP: ");
```

```
    // Serial.print(ant_loop);
```

```
    // ant_loop++;
```

```
LL_0 = analogRead(LDR_0); // Read analog LDR 0
```

```
if ((LL_0 > Lim_1_0) && (LDR_0_F == 0))
```

```
{
```

```
    Total_Weight = Total_Weight + W_1_0;
```

```
    Total_Verdi = Total_Verdi + V_1_0;
```

```
    LDR_0_F = 1;
```

```
    delay(vent);
```

```
}
```

```
else if ((LL_0 < Lim_1_0) && (LDR_0_F == 1))
```

```
{
```

```
    Total_Weight = Total_Weight - W_1_0;
```

```
    Total_Verdi = Total_Verdi - V_1_0;
```

```
    LDR_0_F = 0;
```




```
    delay(vent);
}

LL_1 = analogRead(LDR_1); // Read analog LDR 1

if((LL_1 > Lim_1_5) && (LDR_1_F == 0))
{
    Total_Weight = Total_Weight + W_1_5;
    Total_Verdi = Total_Verdi + V_1_5;
    LDR_1_F = 1;
    delay(vent);
}
else if((LL_1 < Lim_1_5) && (LDR_1_F == 1))
{
    Total_Weight = Total_Weight - W_1_5;
    Total_Verdi = Total_Verdi - V_1_5;
    LDR_1_F = 0;
    delay(vent);
}

LL_2 = analogRead(LDR_2); // Read analog LDR 2

if((LL_2 > Lim_2_0) && (LDR_2_F == 0))
{
    Total_Weight = Total_Weight + W_2_0;
    Total_Verdi = Total_Verdi + V_2_0;
    LDR_2_F = 1;
    delay(vent);
}
else if((LL_2 < Lim_2_0) && (LDR_2_F == 1))
{
    Total_Weight = Total_Weight - W_2_0;
    Total_Verdi = Total_Verdi - V_2_0;
    LDR_2_F = 0;
    delay(vent);
}

LL_3 = analogRead(LDR_3); // Read analog LDR 3
```



```
if((LL_3 > Lim_2_5) && (LDR_3_F == 0))
{
    Total_Weight = Total_Weight + W_2_5;
    Total_Verdi = Total_Verdi + V_2_5;
    LDR_3_F = 1;
    delay(vent);
}
else if((LL_3 < Lim_2_5) && (LDR_3_F == 1))
{
    Total_Weight = Total_Weight - W_2_5;
    Total_Verdi = Total_Verdi - V_2_5;
    LDR_3_F = 0;
    delay(vent);
}

LL_4 = analogRead(LDR_4); // Read analog LDR 4

if((LL_4 > Lim_3_0) && (LDR_4_F == 0))
{
    Total_Weight = Total_Weight + W_3_0;
    Total_Verdi = Total_Verdi + V_3_0;
    LDR_4_F = 1;
    delay(vent);
}
else if((LL_4 < Lim_3_0) && (LDR_4_F == 1))
{
    Total_Weight = Total_Weight - W_3_0;
    Total_Verdi = Total_Verdi - V_3_0;
    LDR_4_F = 0;
    delay(vent);
}

LL_5 = analogRead(LDR_5); // Read analog LDR 5

if((LL_5 > Lim_3_5) && (LDR_5_F == 0))
{
    Total_Weight = Total_Weight + W_3_5;
```



```
Total_Verdi = Total_Verdi + V_3_5;
LDR_5_F = 1;
delay(vent);
}
else if((LL_5 < Lim_3_5) && (LDR_5_F == 1))
{
Total_Weight = Total_Weight - W_3_5;
Total_Verdi = Total_Verdi - V_3_5;
LDR_5_F = 0;
delay(vent);
}

LL_6 = analogRead(LDR_6); // Read analog LDR 6

if((LL_6 > Lim_4_0) && (LDR_6_F == 0))
{
Total_Weight = Total_Weight + W_4_0;
Total_Verdi = Total_Verdi + V_4_0;
LDR_6_F = 1;
delay(vent);
}
else if((LL_6 < Lim_4_0) && (LDR_6_F == 1))
{
Total_Weight = Total_Weight - W_4_0;
Total_Verdi = Total_Verdi - V_4_0;
LDR_6_F = 0;
delay(vent);
}

LL_7 = analogRead(LDR_7); // Read analog LDR 7

if((LL_7 > Lim_4_5) && (LDR_7_F == 0))
{
Total_Weight = Total_Weight + W_4_5;
Total_Verdi = Total_Verdi + V_4_5;
LDR_7_F = 1;
delay(vent);
}
```



```
else if ((LL_7 < Lim_4_5) && (LDR_7_F == 1))
{
    Total_Weight = Total_Weight - W_4_5;
    Total_Verdi = Total_Verdi - V_4_5;
    LDR_7_F = 0;
    delay(vent);
}
```

```
LL_8 = analogRead(LDR_8); // Read analog LDR 8
```

```
if ((LL_8 > Lim_5_0) && (LDR_8_F == 0))
{
    Total_Weight = Total_Weight + W_5_0;
    Total_Verdi = Total_Verdi + V_5_0;
    LDR_8_F = 1;
    delay(vent);
}
else if ((LL_8 < Lim_5_0) && (LDR_8_F == 1))
{
    Total_Weight = Total_Weight - W_5_0;
    Total_Verdi = Total_Verdi - V_5_0;
    LDR_8_F = 0;
    delay(vent);
}
```

```
LL_9 = analogRead(LDR_9); // Read analog LDR 9
```

```
if ((LL_9 > Lim_5_5) && (LDR_9_F == 0))
{
    Total_Weight = Total_Weight + W_5_5;
    Total_Verdi = Total_Verdi + V_5_5;
    LDR_9_F = 1;
    delay(vent);
}
else if ((LL_9 < Lim_5_5) && (LDR_9_F == 1))
{
    Total_Weight = Total_Weight - W_5_5;
    Total_Verdi = Total_Verdi - V_5_5;
}
```



```
LDR_9_F = 0;
delay(vent);
}

LL_10 = analogRead(LDR_10); // Read analog LDR 9

if ((LL_10 > Lim_6_0) && (LDR_10_F == 0))
{
    Total_Weight = Total_Weight + W_6_0;
    Total_Verdi = Total_Verdi + V_6_0;
    LDR_10_F = 1;
    delay(vent);
}
else if ((LL_10 < Lim_6_0) && (LDR_10_F == 1))
{
    Total_Weight = Total_Weight - W_6_0;
    Total_Verdi = Total_Verdi - V_6_0;
    LDR_10_F = 0;
    delay(vent);
}

// Serial.print("LL_0: "); Serial.print(LL_0); Serial.print(" Lim_1_0: "); Serial.println(Lim_1_0);
// Serial.print("LL_1: "); Serial.print(LL_1); Serial.print(" Lim_1_5: "); Serial.println(Lim_1_5);
// Serial.print("LL_2: "); Serial.print(LL_2); Serial.print(" Lim_2_0: "); Serial.println(Lim_2_0);
// Serial.print("LL_3: "); Serial.print(LL_3); Serial.print(" Lim_2_5: "); Serial.println(Lim_2_5);
// Serial.print("LL_4: "); Serial.print(LL_4); Serial.print(" Lim_3_0: "); Serial.println(Lim_3_0);
// Serial.print("LL_5: "); Serial.print(LL_5); Serial.print(" Lim_3_5: "); Serial.println(Lim_3_5);
// Serial.print("LL_6: "); Serial.print(LL_6); Serial.print(" Lim_4_0: "); Serial.println(Lim_4_0);
// Serial.print("LL_7: "); Serial.print(LL_7); Serial.print(" Lim_4_5: "); Serial.println(Lim_4_5);
// Serial.print("LL_8: "); Serial.print(LL_8); Serial.print(" Lim_5_0: "); Serial.println(Lim_5_0);
// Serial.print("LL_9: "); Serial.print(LL_9); Serial.print(" Lim_5_5: "); Serial.println(Lim_5_5);
// Serial.print("LL_10: "); Serial.print(LL_10); Serial.print(" Lim_6_0: "); Serial.println(Lim_6_0);
// delay(1000);

// Beregner sifrene i vekten
Siffer_100_kg = (Total_Weight/100);
Siffer_10_kg = ((Total_Weight - 100*Siffer_100_kg)/10);
Siffer_1_kg = (Total_Weight - 100*Siffer_100_kg - 10*Siffer_10_kg);
```



```
Sif[6] = char(Siffer_100_kg); // Adder 48 for å angi ASKII-verdien
Sif[5] = char(Siffer_10_kg); // Adder 128 for å angi punkt ved sifferet
Sif[4] = char(Siffer_1_kg); // Adder 48 for å angi ASKII-verdien

// Beregner sifrene i krondeverdien
Siffer_1000_kr = (Total_Verdi/1000);
Siffer_100_kr = ((Total_Verdi - 1000*Siffer_1000_kr)/100);
Siffer_10_kr = ((Total_Verdi - 1000*Siffer_1000_kr - 100*Siffer_100_kr)/10);
Siffer_1_kr = (Total_Verdi - 1000*Siffer_1000_kr - 100*Siffer_100_kr - 10*Siffer_10_kr);

Sif[3] = char(Siffer_1000_kr); // Adder 48 for å angi ASKII-verdien
Sif[2] = char(Siffer_100_kr); // Adder 48 for å angi ASKII-verdien
Sif[1] = char(Siffer_10_kr); // Adder 48 for å angi ASKII-verdien
Sif[0] = char(Siffer_1_kr); // Adder 48 for å angi ASKII-verdien

BigDisplay (); // Skriv vekt til display

if (Old_Weight != Total_Weight)
{
// Serial.println("Vekten er forandret");
OnTime = 0;
ActiveFlagg = 1;
Old_Weight = Total_Weight; // Oppdaterer gammelverdi
}
else
{
OnTime = OnTime + 1;
}

// Avslutt og tøm bordet dersom det er gått tilstrekkelig lang tid siden noe har skjedd

if ((OnTime > TimeLimit) && (ActiveFlagg == 1))
{
Rydd_bordet ();
ActiveFlagg = 0;
OnTime = 0;
}
```



```
Old_Weight = 0;
}
else if ((OnTime > TimeLimit) && (ActiveFlagg == 0))
{
OnTime = 0;
}

// Serial.print(" ActiveFlagg = ");
// Serial.print(ActiveFlagg);
// Serial.print(" Total vekt: ");
// Serial.print(Total_Weight);
// Serial.print(" Old vekt = ");
// Serial.print(Old_Weight);
// Serial.print(" OnTime = ");
// Serial.println(OnTime);

// Registrer overvekt

if (Total_Weight > 200)
{
digitalWrite(Overweight, HIGH);
}
else
{
digitalWrite(Overweight, LOW);
}

delay(100);
}

void Status ()
{

if (Total_Weight > 200)
{
for (n = 0; n < 4; n++) // Blinkende lys ved overvekt
{
digitalWrite (Best, HIGH); // Alarm
```



```
digitalWrite (Nest_Best, HIGH);
digitalWrite (Bra, HIGH);
digitalWrite (Daarlig, HIGH);
digitalWrite (Daarligst, HIGH);
digitalWrite (Overweight, HIGH);

for(i=0 ; i<30 ; i++) {delayMicroseconds(16383);} // ca. 0,5 sek.

digitalWrite (Best, LOW); // Alarm
digitalWrite (Nest_Best, LOW);
digitalWrite (Bra, LOW);
digitalWrite (Daarlig, LOW);
digitalWrite (Daarligst, LOW);
digitalWrite (Overweight, LOW);

for(i=0 ; i<30 ; i++) {delayMicroseconds(16383);} // ca. 0,5 sek.
}
}
else
{
if (Total_Verdi == Best_Verdi)                digitalWrite (Best, HIGH);
if ((Nest_Best_Verdi <= Total_Verdi) && (Total_Verdi < Best_Verdi)) digitalWrite (Nest_Best, HIGH);
if ((Bra_Verdi <= Total_Verdi) && (Total_Verdi < Nest_Best_Verdi)) digitalWrite (Bra, HIGH);
if ((Daarlig_Verdi <= Total_Verdi) && (Total_Verdi < Bra_Verdi))  digitalWrite (Daarlig, HIGH);
if (Total_Verdi < Daarlig_Verdi)                digitalWrite (Daarligst, HIGH);

for(i=0 ; i<200 ; i++) {delayMicroseconds(16383);} // ca. 3 sek.

digitalWrite (Best, LOW);
digitalWrite (Nest_Best, LOW);
digitalWrite (Bra, LOW);
digitalWrite (Daarlig, LOW);
digitalWrite (Daarligst, LOW);
}
}

void Rydd_bordet ()
```




```
{
digitalWrite (Vifte, HIGH);
Serial.print("Rydder bordet");
delay(8000);
digitalWrite (Vifte, LOW);
}

void CalibrationLDR()
{
  Lim_1_0 = analogRead(LDR_0) + LimMargin; // Read analog LDR 0
  Lim_1_5 = analogRead(LDR_1) + LimMargin; // Read analog LDR 1
  Lim_2_0 = analogRead(LDR_2) + LimMargin; // Read analog LDR 2
  Lim_2_5 = analogRead(LDR_3) + LimMargin; // Read analog LDR 3
  Lim_3_0 = analogRead(LDR_4) + LimMargin; // Read analog LDR 4
  Lim_3_5 = analogRead(LDR_5) + LimMargin; // Read analog LDR 5
  Lim_4_0 = analogRead(LDR_6) + LimMargin; // Read analog LDR 6
  Lim_4_5 = analogRead(LDR_7) + LimMargin; // Read analog LDR 7
  Lim_5_0 = analogRead(LDR_8) + LimMargin; // Read analog LDR 8
  Lim_5_5 = analogRead(LDR_9) + LimMargin; // Read analog LDR 9
  Lim_6_0 = analogRead(LDR_10) + LimMargin; // Read analog LDR 10
}

void BigDisplay ()
{
  // unsigned char tall;
  byte tall;
  char c = 0;

  digitalWrite(pinCLK, LOW);
  delay(250);
  for (n = 6; n > -1; n--)
  {
    tall = SifCon[Sif[n]];
    if (n == 5) tall = tall + 1;
    // Serial.println(int(tall));
    for (m = 0; m < 8; m++)
    {
```



```
//Serial.println(int(tall));
if (tall & 128)
{
    digitalWrite(pinSDI, HIGH);
}
else
{
    digitalWrite(pinSDI, LOW);
}
delayMicroseconds(100);
tall = tall << 1;
digitalWrite(pinCLK, HIGH); // Clocking bit into shiftregister
delayMicroseconds(100);
digitalWrite(pinCLK, LOW);
}
delay(1);
}

digitalWrite(pinLE,HIGH); // Latch display data
delayMicroseconds(200);
digitalWrite(pinLE,LOW);
}
```











Heftet gir en kortfattet orientering om bruk og vedlikehold av installasjonen "Ut og fly - størst verdi" som er utviklet som et delprosjekt under Abel-fond-prosjektet som startet i 2011 og avsluttes ved årsskiftet 2015/16. *Utvikling av rike interaktive installasjoner for bruk ved Vitensenter* har vært et samarbeid mellom Vitensenteret i Trondheim og VilVite i Bergen. I løpet av prosjektet er det utviklet to interaktive modeller ved VilVite og to ved Vitensenteret i Trondheim (ViT). Det har vært et krav at utstillingene skal handle om matematikk og være utviklet i tråd med intensjonen om "Active prolonged engagement", et prinsipp utviklet ved Exploratorium i San Fransisco.



ABEL
PRISEN