

Nils Kr. Rossing

# Servicehefte – “CO<sub>2</sub> - Regnemaskin”



Denne siden er blank

Servicehefte –  
“CO<sub>2</sub>-Regnemaskin”

Nils Kr. Rossing

## Servicehefte – “CO2-Regnemaskin”

Trondheim 2016

ISBN 978-82-92088-56-3

Bidragstere:

*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](mailto: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 1.0 – 16.06.16

Utstillingen er bygget  
på oppdrag fra Skimuseet ved Holmenkollen



---

## Forord

Heftet gir en kortfattet orientering om bruk og vedlikehold av installasjonen “CO<sub>2</sub>-Regnemaskin” som er utviklet på oppdrag fra Skimuseet.

Modellen “CO<sub>2</sub>-Regnemaskin” er utviklet av Vitensenteret i Trondheim v/Nils Kr. Rossing som også har bygget, montert og testet ut elektronikken. Det er også han som har programmert styre-elektronikken. Kabinettet er bygget av **Steinar Løken-Olsen**. **Frode Willmann** og **Lars Pedersen** har bistått under byggingen. Topp-plata er skåret og gravert på vår EPILOG Fusion M2 laserkutter.

Heftet vil bli lagt ut på vår hjemmeside og oppdatert etter behov.

Vitensenteret  
Juni 2016  
Nils Kr. Rossing



---

## Innhold

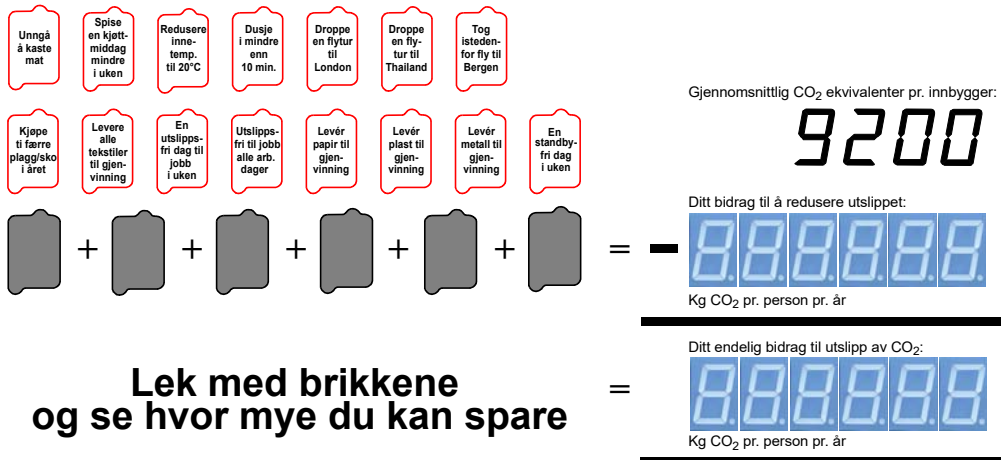
<b>1 Innledning .....</b>	<b>7</b>
<b>2 Oppstart og service .....</b>	<b>8</b>
2.1 Oppstart .....	8
2.2 Brikkene .....	8
2.3 Oppdatering av programvare .....	10
<b>3 Kort beskrivelse av installasjonens oppbygning .....</b>	<b>13</b>
<b>4 Hjelp til feilfinning ved ev. feil .....</b>	<b>14</b>
4.1 Feilregistrering av brikker .....	14
4.2 Ustabil registrering av en brikke .....	15
4.3 Brikker som forsvinner .....	15
5 Bilder av interiør .....	16
<b>Vedlegg A Styreprogram .....</b>	<b>19</b>



# 1 Innledning

Notatet gir en kort beskrivelse av utstillingsmodellen “CO<sub>2</sub>-Regnemaskin”, hvordan den er bygget opp og tips til vedlikehold.

## CO<sub>2</sub> REGNEMASKIN



15 brikker er påført ulike tiltak for å redusere CO<sub>2</sub>-utslippet. Disse ligger i 15 *passive* posisjoner for oppbevaring når de ikke er i bruk, og kan flytte ned til 6 *aktive* posisjoner når de skal inkluderes i CO<sub>2</sub>-regnskapet. Til hver brikke er det knyttet en CO<sub>2</sub>-ekvivalent som ligger skjult i programvaren, den er med andre ord ikke uten videre tilgjengelig for publikum. Når en brikke legges ned i en av de aktive posisjonene, vil den skjulte verdien til CO<sub>2</sub>-ekvivalenten adderes til summen på displayet til høyre. Det er mulig å velge inntil seks tiltak (brikker) som kan være med å redusere CO<sub>2</sub>-fotavtrykket for oss mennesker. 9200 kg CO<sub>2</sub>-ekvivalenter er gjennomsnittet for hver enkelt individ i den norske befolkningen. Dette inkluderer både det som hver enkelt bruker og summen av det som industrien slipper ut fordelt på alle innbyggerne i Norge.

Alle de 15 brikkene er tiltak som avviker fra gjennomsnittet og dermed kommer i fratrekk fra gjennomsnittet. Fratrekket er merket med: *Ditt bidrag til å redusere utslippet*: Dette er summen av de brikkene som ligger i de aktive posisjonene og vil bli trukket fra gjennomsnittet. Resultatet er merket med: *Ditt endelige bidrag til utslipp av CO<sub>2</sub>*.

### En betenkning mht. bruk av modellen

Siden 9200 kg er gjennomsnittet inkludert utslipp fra industrien, kan modellen noen ganger være vanskelig å tolke av publikum. Under er skissert noe situasjoner som kan skape feilslutninger:



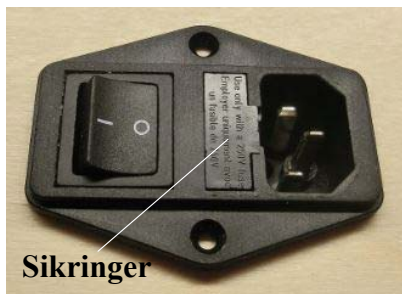
1. Noen vil kanskje reagere på at de reduserer “sitt” utslipp med 2000 kg når de dropper en reise til Thailand. Dersom de aldri har hatt planer om å reise til Thailand, så vil det kanskje oppleves feil å gjøre et slikt valg.
2. Dersom man har planlagt en reise til f.eks. London og dermed sparer 480 kg CO<sub>2</sub>-ekvivalenter, så betyr ikke det at gjennomsnittet på landsbasis går ned med 480 kg CO<sub>2</sub>-ekv. på grunn av *deres* valg, men ville ha gjort det dersom alle i Norge hadde gjort det samme valget. Forutsetningen er da at de i utgangspunktet hadde tenkt å ta en reise til London eller tilsvarende.
3. Dersom brukeren av utstillingen dusjer hver lørdag og ikke ellers, hva betyr det da at vedkommende reduserer forbruket til under 10 min. pr. dag?

Med andre ord, man må være nøye med hva man skriver i teksten som skal hjelpe de besøkende til å bruke utstillingen for å få en rett forståelse av konsekvensene av slike valg.

En kan risikere at slik modellen er utformet så hadde en liste over CO<sub>2</sub>-ekvivalenter knyttet til ulike aktiviteter vært mer opplysende. Dog skal man ikke undervurdere betydningen av å foreta seg konkrete handlinger ved å flytte på brikker.

## 2 Oppstart og service

### 2.1 Oppstart



Modellen startes ved å sette stikkkontakten i veggens og slå på hovedbryteren som er bak til venstre. 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 ikke tåler startstrømmene til strømforsyningen.

### 2.2 Brikkene

Det følger med 15 brikker merket som vist i tabellen under. Disse er kodet med magneter på baksiden. Magnetene er presset inn i ett eller flere av fire hull og er ikke limt. Om noen av magnetene skulle falle ut og forsvinner, kan de erstattes med sylindriske magneter Ø10 mm x 2 mm (uten hull) som kan kjøpes hos Clas Ohlson. Dersom en magnet faller ut og skal settes tilbake i hullet, så legg magneten over hullet med *sydpolen* inn mot hullet. Fester den med en tape, og leer en plate over magneten. Ved å slå hardt på plata, smetter magneten inn i hullet og blir liggende kant i kant med undersiden av brikken.

Dersom noen av brikkene forsvinner ta kontakt med Vitensenteret i Trondheim så lager vi nye brikker.



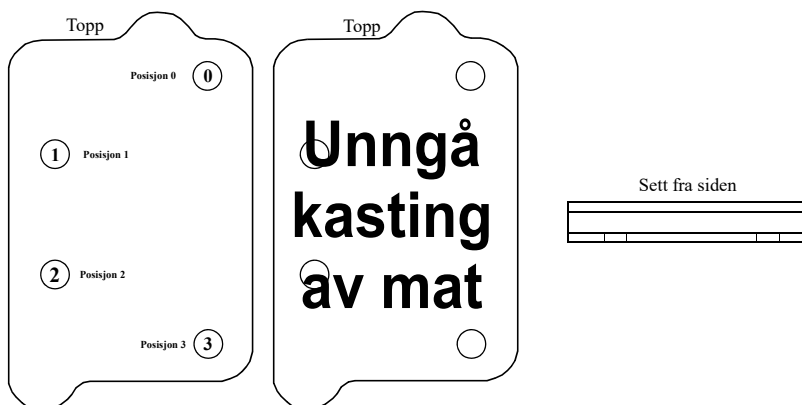


Tabellen under viser de tiltak som kan velges, hver brikke sin magnetkoding og hvor stor reduksjon i CO<sub>2</sub>-ekvivalent hvert enkelt valg medfører:

**Tabell 2.1 Forslag til bidrag til CO<sub>2</sub> reduksjon**

Nr.	Type	Koding Pos. 3,2,1,0	Mengde CO <sub>2</sub>	Kommentar
	Indikerer at posisjonen er tom	1111	-	Ingen magneter
1	<i>Unngå å kaste mat</i>	1110	- 75 kg	Magnet i pos 0
2	<i>Spise en kjøttmiddag mindre i uken</i>	1101	- 1 400 kg	
3	<i>Redusere innetemp. til 20°C</i>	1100	- 73 kg	
4	<i>Dusje i mindre enn 10 min.</i>	1011	- 46 kg	
5	<i>Droppe en flytur til London</i>	1010	- 480 kg	
6	<i>Droppe en flytur til Thailand</i>	1001	- 2 000 kg	
7	<i>Tog istedenfor fly til Bergen</i>	1000	- 106 kg	
8	<i>Kjøre ti færre plagg/sko i året</i>	0111	- 85 kg	
9	<i>Lever alle tekstiler til gjenvinning</i>	0110	- 90 kg	
10	<i>En utslippsfri dag til jobb i uken</i>	0101	- 108 kg	
11	<i>Utslippsfri til jobb alle arb. dager</i>	0100	- 529 kg	
12	<i>Lever papir til gjenvinning</i>	0011	- 22 kg	
13	<i>Lever plast til gjenvinning</i>	0010	- 6 kg	
14	<i>Lever metall til gjenvinning</i>	0001	- 34 kg	
15	<i>En standby-fri dag i uken</i>	0000	- 110 kg	Fire magneter

Tredje kolonnen indikerer i hvilke posisjoner det er plassert magneter. Figuren under viser plasseringen av hullene for magnetene i de ulike posisjonene på brikkene.



Det er bare laget hull der det skal plasseres magneter. Brikkene er sett fra tekstsiden.



Dersom det skulle være behov for å skifte verdier for CO<sub>2</sub>-ekvivalentene, så kan det gjøres ved at man endrer verdiene i programmet. Koden må legges inn på stedet der modellen befinner seg og gjøres i følge prosedyren beskrevet i avsnitt 2.3, eller man kontakter ViT for å få generert ny kode.

## 2.3 Oppdatering av programvare

Siden dette er første gang akkurat denne varianten av utstyret er bygget, er det ikke utenkelig at det kan oppstå situasjoner som gjør at modellen oppfører seg på en uventet måte. Ta kontakt med Vitensenteret i Trondheim og beskriv problemer som er oppstått. Oppgradert programvare kan da oversendes og installeres av lokale teknikere. Det er dog vanskelig å teste programvaren siden vi ikke har noen kopi av utstyret i Trondheim.

## Nedlasting og installasjon av programvare

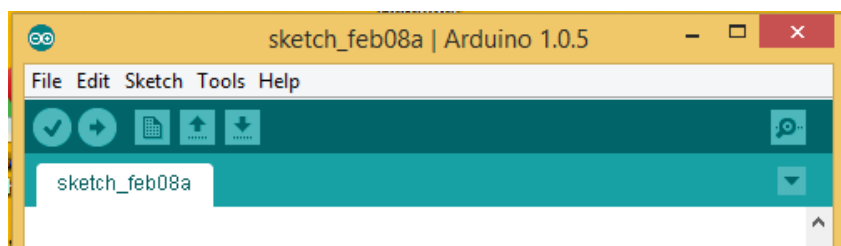
Det er programeditoren til Arduino som benyttes. Denne hentes ned fra

<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 1.6.9

Slå på PC og plugg USB-kablene fra modellen inn i en USB-port. Kabelen for installasjonen skal være fastet bak til høyre på kabinettet. Dra kabelen ut av hullet og koble den 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).

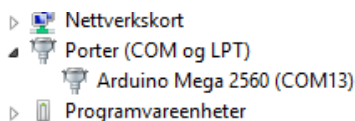


Ved å velge *Tools* fra menylinjen vil man få flere valg, deriblant *Board* og *Serial Port*.

Board – velg: *Arduino Mega 2560*

Serial Port – velg: *COM <høyeste nr.>*

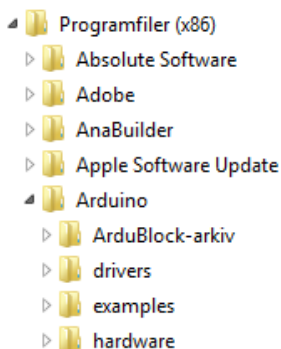
Dersom driverne ikke er installert riktig vil man hos PC'er bare finne *COM 1* eller ingen (Serial Port er "grå"). I så tilfelle kan man gå til Kontrollpanel/System/Enhetsbehandling her skal det komme opp et alternativ i listen som heter *Porter*. Velger man denne vil man se noe i likhet med det som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



Hvor Arduino Mega 2560 er tildelt en port (i vårt tilfelle COM13), eller man får 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

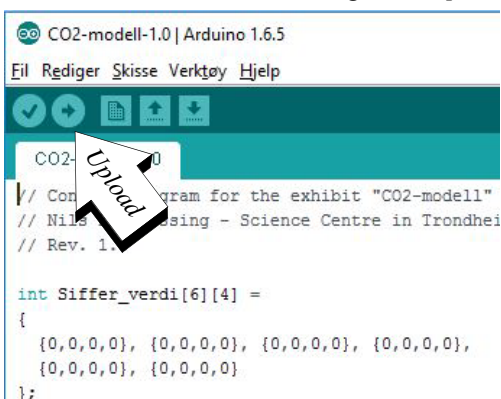


## Installer ny programvare i “CO<sub>2</sub>-REGNEMASKIN”

Programfilen for innlasting vil ha navnekonvensjonen:

CO<sub>2</sub>-modell-<Rev nr.>.ino den medfølgende er CO<sub>2</sub>-modell-1.0.ino

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, men displayene vil ikke virke med mindre at utstyret er slått på. Den kan selvfølgelig også lastes opp mens modellen er påslått.

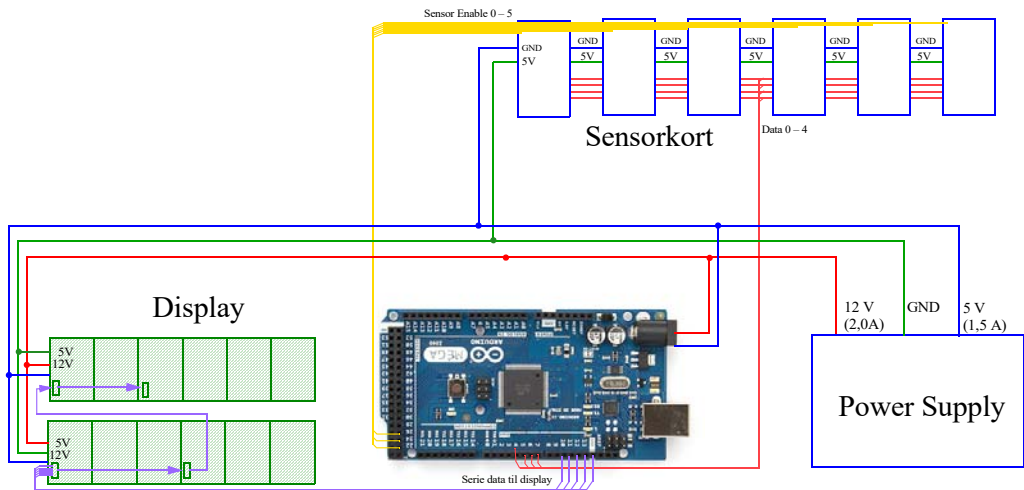
## Endring av CO<sub>2</sub>-ekvivalentverdier

På et tidspunkt kan det være aktuelt å endre på verdiene til CO<sub>2</sub>-ekvivalentene. Disse verdiene er skrevet inn i programmet. For å endre disse må man åpne programmet i editoren og legge inn nye verdier. I figuren til høyre ser man utsnittet av programmet der verdiene er lagt inn. En må da lete seg fram til dette punktet og gjøre de endringene man ønsker for så og compilere og legge inn programmet på nytt inn i styrekortet.

```
int conTabell[16][5] =
{
  {1,1,1,1,0}, // Ingen brikker plassert
  {1,1,1,0,75}, // Uungå å kaste mat
  {1,1,0,1,1400}, // Spis en kjøttmiddag mindre i uka
  {1,1,0,0,73}, // Reduser innetemperaturen til 20 C
  {1,0,1,1,46}, // Dusje i mindre enn 10 min.
  {1,0,1,0,480}, // Droppe en flytur til London
  {1,0,0,1,2000}, // Droppe en flytur til Thailand
  {1,0,0,0,106}, // Tog istedenfor fly til Bergen
  {0,1,1,1,85}, // Kjøpe ti færre plagg/sko i året
  {0,1,1,0,90}, // Lever alle tekstiler til gjenvinning
  {0,1,0,1,108}, // En utslippsfri dag til jobb i uken
  {0,1,0,0,529}, // Utslippsfri til jobb alle arb. dager
  {0,0,1,1,22}, // Lever papir til gjenvinning
  {0,0,1,0,6}, // Lever plast til gjenvinning
  {0,0,0,1,34}, // Lever metall til gjenvinning
  {0,0,0,0,110} // En standby-fri dag i uken
};
```

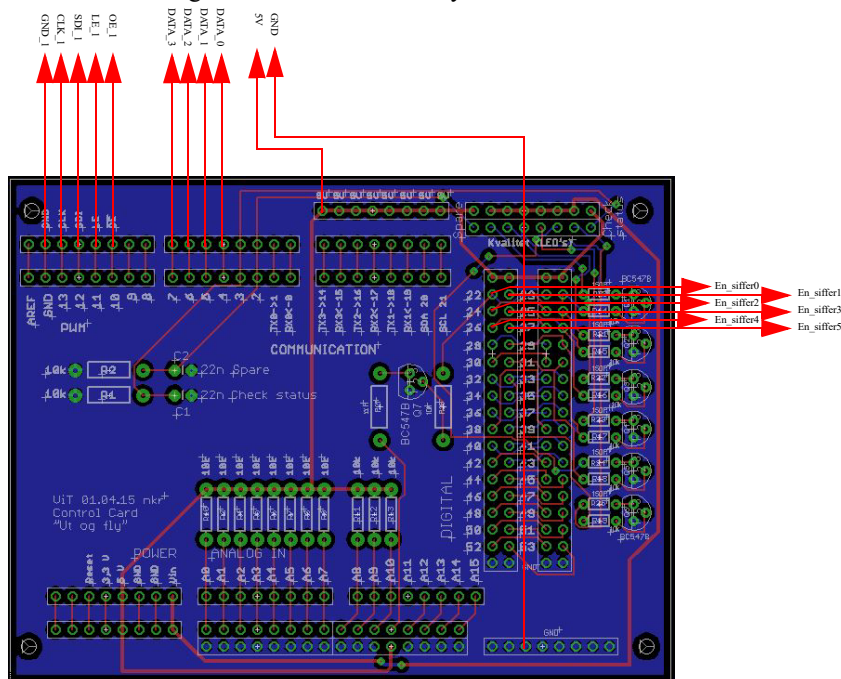
## 3 Kort beskrivelse av installasjonens oppbygning

Blokkdiagrammet under viser en overordnet skisse av hvordan maskinvaren er bygget opp.



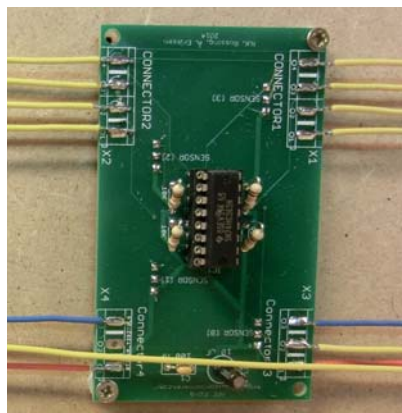
Styrekortet er en Arduino Mega 2650 med shieldkort for at det skal være lett å lodde til ledningene.

Figuren under viser tilkoblingen til shieldkortet for styrekortet.



## Sensorkortene

Sensorene er hall switcher som tilkobler (slutter strømmen) når en magnet kommer i nærheten. *Nordpolen på magnetene skal peke ned mot sensorene.* Hvert kort består av fire sensorer slik at en i prinsippet kan skille mellom 16 ulike magnetkonstellasjoner. Null-tilstanden (ingen magnet) brukes for å detektere at ingen brikke er plassert i posisjonen. En tri-state bussdriver gjør at sensorene kan leses av kort for kort (brikke for brikke) ved hjelp av enable linjer til hvert av kortene (gule ledninger).



## Displaykortene

Displaykortene er utviklet av Science Project Ltd - London. Disse leveres med fire siffer, men kan deles opp og settes sammen etter behov. Kortene kobles i kjede slik at dataene sendes inn på seriell form i kortet lengst mot høyre for så å riple gjennom kortene.



## 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 2,0 A. Arduino Mega forsynes med 12 V som reguleres ned til 5 V på kortet. Sensorkortene forsynes med 5 V og vil dermed ikke belaste regulatoren på kontrollert-kortet.

## 4 Hjelp til feilfinning ved ev. feil

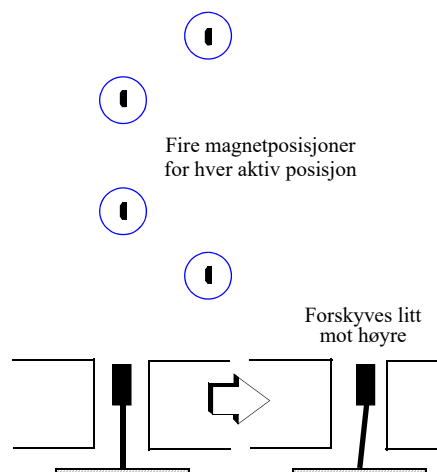
### 4.1 Feilregistrering av brikker

Det er oppdaget at det er mulig å registrere feil brikke akkurat idet brikken legges på plass. Dette kan skje dersom brikken legges langsomt ned. Da kan det skje at ikke alle magneten registreres samtidig, slik at registreringen et kort øyeblikk blir feil. Dette er imidlertid uten betydning da den riktige registreringen kommer på plass omtrent umiddelbart etter.

### 4.2 Ustabil registrering av en brikke

Dersom man oppdager at en brikke gir forskjellig verdi avhengig av hvilken aktiv posisjon den settes i, så er det stor sannsynlighet for at dette skyldes at man har fått en litt uheldig posisjonering av en av sensoren i plata under denne brikken slik at magneten ikke blir registrert av sensoren. Da gjør man som følger:

- Finn hvilken aktiv posisjon som gir feil.
- Om mulig finn hvilken av de fire mulige magnetposisjonen som gir feil.
- Skru av topp-plata og plastplata slik at du ser de svarte sensorene i de sirkulære utboringene
- Forskyv sensoren en mm mot høyre, men sørg for at den står vertikalt som vist nederst på figuren til høyre.
- Monter topp-plata og sett i skruene.



### 4.3 Brikker som forsvinner

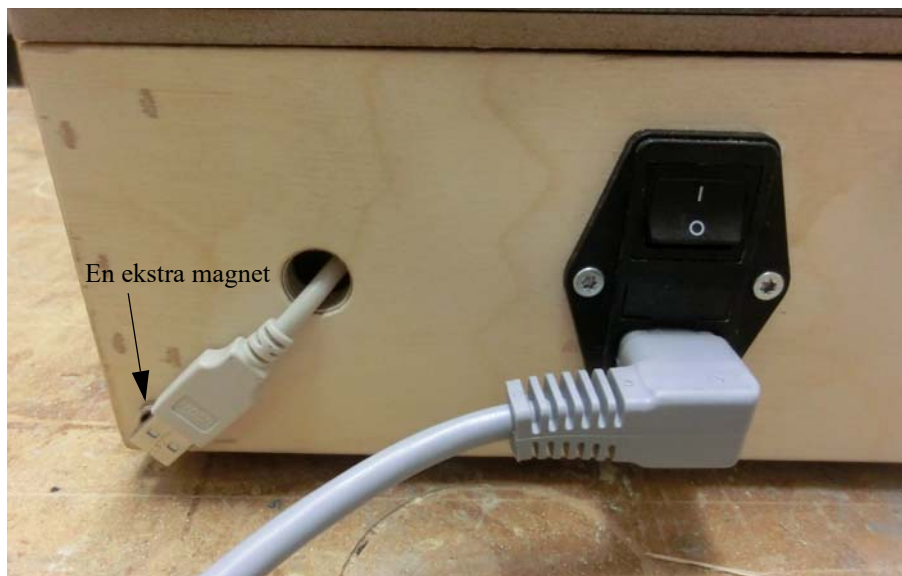
Dersom en av brikkene skulle forsvinne, ta kontakt med Vitensenteret i Trondheim så lager vi en ny. Bruk gjerne [nkr@vitensenteret.com](mailto:nkr@vitensenteret.com)

## 5 Bilder av interiør

Topp-plate med brikker



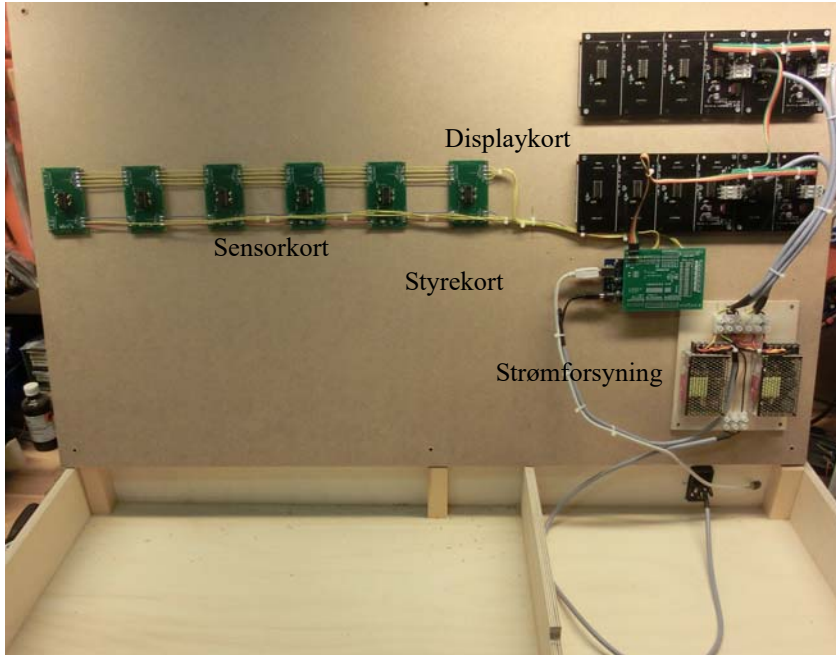
Bakside med USB-kontakt



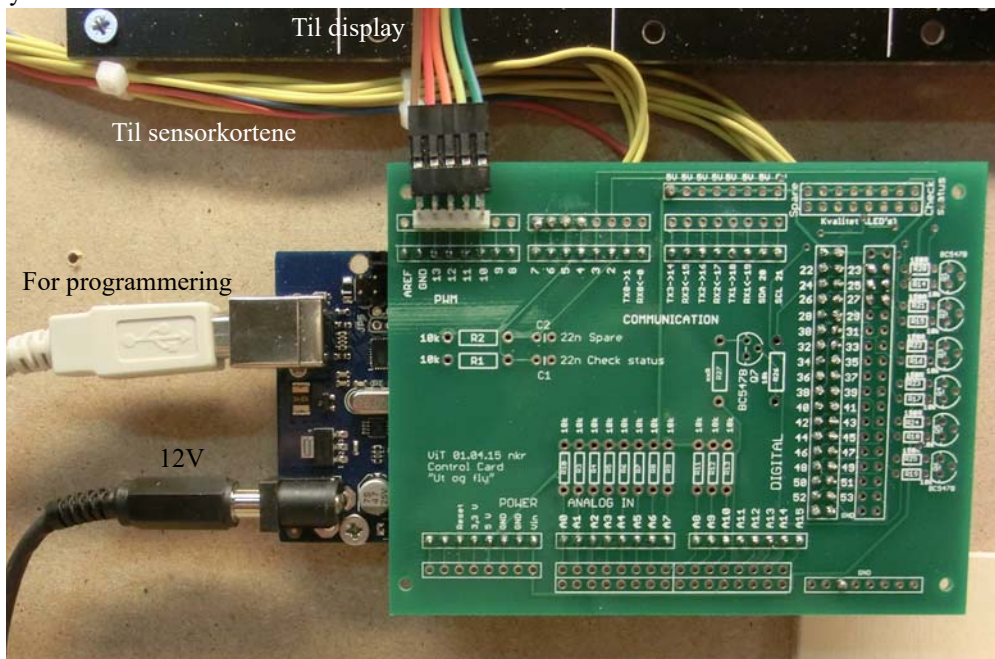
Under USB-kontakten ligger det ekstra magnet som kan taes av og brukes for å detektere følsomhetsområdet til sensorene. Plasseringen av følsomhetsområdet i forhold til magnetene på brikkene kan være en indikasjon på at det kan oppstå feildeteksjoner.



## Oversiktsbilde av elektronikken

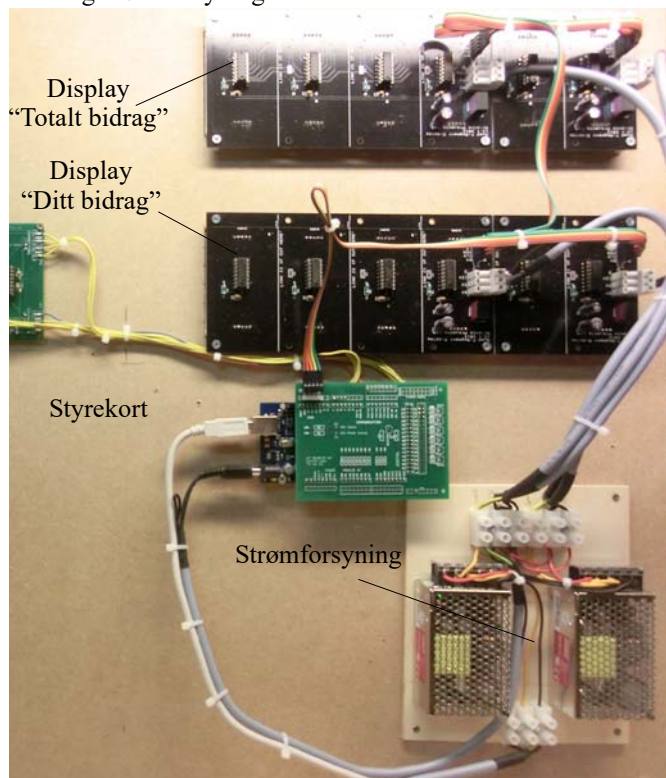


## Styrekort med shield





## Displaykort, styrekort og strømforsyning



## Nettilkobling og sikringer.





## Vedlegg A Styreprogram

Under er vist styreprogrammet som ligger i kontrolleren på styrekortet.

```
// Control program for the exhibit "CO2-modell"
```

```
// Nils Kr. Rossing - Science Centre in Trondheim 9. juni 2016
```

```
// Rev. 1.0
```

```
int Siffer_verdi[6][4] =
```

```
{
```

```
  {0,0,0,0}, {0,0,0,0}, {0,0,0,0}, {0,0,0,0},
```

```
  {0,0,0,0}, {0,0,0,0}
```

```
};
```

```
int pinData[4]   = {4, 5, 6, 7};
```

```
int pinEn_siffer[6] = {22, 23, 24, 25, 26, 27};
```

```
int En_siffer[6]  = {1, 1, 1, 1, 1, 1};
```

```
int bidrag[6]     = {0, 0, 0, 0, 0, 0};
```

```
int conTabell[16][5] =
```

```
{
```

```
  {1,1,1,1,0}, // Ingen brikker plassert
```

```
  {1,1,1,0,75}, // Unngå å kaste mat
```

```
  {1,1,0,1,1400}, // Spis en kjøttmiddag mindre i uka
```

```
  {1,1,0,0,73}, // Reduser innnetemperaturen til 20 C
```

```
  {1,0,1,1,46}, // Dusje i mindre enn 10 min.
```

```
  {1,0,1,0,480}, // Droppe en flytur til London
```

```
  {1,0,0,1,2000}, // Droppe en flytur til Thailand
```

```
  {1,0,0,0,106}, // Tog istedenfor fly til Bergen
```

```
  {0,1,1,1,85}, // Kjøpe ti færre plagg/sko i året
```

```
  {0,1,1,0,90}, // Lever alle tekstiler til gjenvinning
```

```
  {0,1,0,1,108}, // En utslippsfri dag til jobb i uken
```

```
  {0,1,0,0,529}, // Utslippsfri til jobb alle arb. dager
```



```
{0,0,1,1,22}, // Lever papir til gjenvinning
{0,0,1,0,6}, // Lever plast til gjenvinning
{0,0,0,1,34}, // Lever metall til gjenvinning
{0,0,0,0,110} // En standby-fri dag i uken
};

int tidSet_enable = 2;
int tidLes_data = 2;

long gjennomsnittligBidrag = 9200; // Setter gjennomsnittsverdi pr. person
long egetBidrag = 0;
long totaltBidrag = 0;

int i, m, n, p;
int antallSiffer = 6;
int antallPosisjoner = 16; //Antall posisjoner i konverteringstabell

// Declaration for display

// unsigned char Sif[5];

long Siffer_1 = 0;
long Siffer_10 = 0;
long Siffer_100 = 0;
long Siffer_1000 = 0;
long Siffer_10000 = 0;

int pinCLK_1 = 13;
int pinSDI_1 = 12;
int pinLE_1 = 11;
int pin_OE_1 = 10;
```



---

```
int Sif[12];
byte SifCon[12];

// Sif[0] Minst signifikante siffer egetBidrag
// Sif[5] Mest signifikante siffer egetBidrag
// Sif[6] Minst signifikante siffer totaltBidrag
// Sif[11] Mest signifikante siffer totaltBidrag

void setup()
{

  for (i = 0; i < 4; i++)
  {
    pinMode(pinData[i], INPUT);
    digitalWrite(pinData[i], HIGH); // Set pullup resistor
  }

  for (i = 0; i < 6 ; i++)
  {
    pinMode(pinEn_siffer[i], OUTPUT);
  }

  pinMode(pinCLK_1, OUTPUT);
  pinMode(pinSDI_1, OUTPUT);
  pinMode(pinLE_1, OUTPUT);
  pinMode(pin_OE_1, OUTPUT);

  digitalWrite(pin_OE_1, LOW);
  digitalWrite(pinLE_1, 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; // 9 (ABCDEFG DP)
SifCon[10] = B00000010; // - (ABCDEFG DP)
SifCon[11] = B00000000; // '' (ABCDEFG DP)
```

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

```
delay(1000); // This delay should guarantee that the
              // display is initialised before the program
              // start communicating with the display cards
```

```
}
```

```
void loop()
```

```
{
```

```
// Les siffer m og digit n
```

```
for (m = 0; m < (antallSiffer); m++)
```

```
{
```

```
digitalWrite(pinEn_siffer[m], LOW); // Read digit m
```

```
delay(tidSet_enable);
```

```
for (n = 0; n < 4; n++)
```

```
{
```

```
Siffer_verdi[m][n] = digitalRead(pinData[n]); // Read position n
```

```
delay(tidLes_data);
```



```
}
digitalWrite(pinEn_siffer[m], HIGH); // Finish reading digit m

}

// Converting from magnetpositions to digits

for (m = 0; m < antallSiffer; m++)
{
  for (p = 0; p < antallPosisjoner; p++)
  {
    if ((Siffer_verdi[m][0] == conTabell[p][0]) &&
        (Siffer_verdi[m][1] == conTabell[p][1]) &&
        (Siffer_verdi[m][2] == conTabell[p][2]) &&
        (Siffer_verdi[m][3] == conTabell[p][3]))
    {
      bidrag[m] = conTabell[p][4];
    }
  }
}

egetBidrag = bidrag[0] + bidrag[1] + bidrag[2] + bidrag[3] + bidrag[4] + bidrag[5];
totaltBidrag = gjennomsnittligBidrag - egetBidrag;

// Converting the numbers to digits egetBidrag

Siffer_10000 = (egetBidrag/10000);
Siffer_1000 = (egetBidrag - 10000*Siffer_10000)/1000;
Siffer_100 = (egetBidrag - 10000*Siffer_10000 - 1000*Siffer_1000)/100;
Siffer_10 = (egetBidrag - 10000*Siffer_10000 - 1000*Siffer_1000 - 100*Siffer_100)/10;
Siffer_1 = (egetBidrag - 10000*Siffer_10000 - 1000*Siffer_1000 - 100*Siffer_100 -
10*Siffer_10);
```



```
Sif[0] = Siffer_1;  
Sif[1] = Siffer_10;  
Sif[2] = Siffer_100;  
Sif[3] = Siffer_1000;  
Sif[4] = Siffer_10000;  
Sif[5] = 10;      // Sett inn minustegn
```

```
// Removing leading zero's
```

```
if (Sif[4] == 0)  
{  
  Sif[4] = 11;  
  if (Sif[3] == 0)  
  {  
    Sif[3] = 11;  
    if (Sif[2] == 0)  
    {  
      Sif[2] = 11;  
      if (Sif[1] == 0)  
      {  
        Sif[1] = 11;  
      }  
    }  
  }  
}
```

```
// Converting the numbers to digits totaltBidrag
```

```
Siffer_10000 = (totaltBidrag/10000);  
Siffer_1000 = (totaltBidrag - 10000*Siffer_10000)/1000;  
Siffer_100 = (totaltBidrag - 10000*Siffer_10000 - 1000*Siffer_1000)/100;
```



---

```
Siffer_10 = (totaltBidrag - 10000*Siffer_10000 - 1000*Siffer_1000 - 100*Siffer_100)/10;  
Siffer_1 = (totaltBidrag - 10000*Siffer_10000 - 1000*Siffer_1000 - 100*Siffer_100 -  
10*Siffer_10);
```

```
Sif[6] = Siffer_1;  
Sif[7] = Siffer_10;  
Sif[8] = Siffer_100;  
Sif[9] = Siffer_1000;  
Sif[10] = Siffer_10000;  
Sif[11] = 0;
```

```
// Removing leading zero's
```

```
if (Sif[11] == 0)  
{  
Sif[11] = 11;  
if (Sif[10] == 0)  
{  
Sif[10] = 11;  
if (Sif[9] == 0)  
{  
Sif[9] = 11;  
if (Sif[8] == 0)  
{  
Sif[8] = 11;  
if (Sif[7] == 0)  
{  
Sif[7] = 11;  
}  
}  
}  
}  
}
```





```
}

BigDisplay ();
delay(500);

}

void BigDisplay ()
{

// Serial.println("BigDisplay is Called");

// unsigned char tall;
byte tall;
char c = 0;

digitalWrite(pinCLK_1, LOW);
delay(250);
for (n = 11; n > -1; n--)
{
tall = SifCon[Sif[n]];
// Serial.println(int(tall));
for (m = 0; m < 8; m++)
{
//Serial.println(int(tall));
if (tall & 128)
{
digitalWrite(pinSDI_1, HIGH);
}
else
{
digitalWrite(pinSDI_1, LOW);
```



---

```
    }
    delayMicroseconds(100);
    tall = tall << 1;
    digitalWrite(pinCLK_1, HIGH); // Clocking bit into shiftregister
    delayMicroseconds(100);
    digitalWrite(pinCLK_1, LOW);
  }
  delay(1);
}

digitalWrite(pinLE_1,HIGH); // Latch display data 1
delayMicroseconds(200);
digitalWrite(pinLE_1,LOW);
}
```



---

NOTATER:







Denne siden er blank



Heftet gir en kortfattet orientering om bruk og vedlikehold av installasjonen “CO<sub>2</sub>-Regnemaskin” som er utviklet på bestilling av Skimuseet. Dette er en av 6 modeller som Vitensenteret i Trondheim har utviklet våren 2016 for Skimuseet i forbindelse med deres klimautstilling som åpner høsten 2016.

