

# 1. Raket

---

Vi skal bygge en raket, som flyver ved at skyde vand ud. Raketten skal bære forskellige dele (sensorer) til at måle turen og luften omkring raketten. Da vægt og størrelse er afgørende for en raket, bruger vi små lette sensorer og computere.

Målingerne skal bruges til at undersøge, om vi kan forbedre raketten flyvetur, så den flyver højere, hurtigere eller mere sikkert. For at indholdet ikke skal blive ødelagt, når raketten rammer jorden, kan vi udstyre den med en faldskærm.

## Mulige opgaver i projektet:

- Konstruere en faldskærm til raketten og afprøve forskellige designs
- Konstruere en mekanisme, som kan udløse en faldskærm
- Måle flyvehøjden
- Måle acceleration og fart
- Undersøge effekten af vinger
- Undersøge effekten af raketten vægt, herunder mængden af vand
- Hvad vi ellers finder på

Under projektet skal I formidle jeres resultater og overvejelser via tekst og video på en blog.

Det er muligt at arbejde som en hel klasse eller i hold. Holdene kan konkurrere mod hinanden. Inden for hver klasse/hold, kan grupper tage sig af forskellige opgaver, som til sammen forbedrer raketten.

Kapitel 2 handler om en bevægelig del, en faldskærmsudløser. Kapitel 3 handler om at bestemme raketten flyvehøjde. De to kapitler kan læses uafhængigt af hinanden. Hvert afsnit i kapitlerne er markeret med \*, \*\* eller \*\*\* efter sværhedsgraden. I nogle afsnit er teksten suppleret med video, som kan ses, hvis teksten ikke giver mening.

<b>1. RAKET</b>	<b>1</b>
<b>2. BEVÆGELSE *</b>	<b>3</b>
2.1 STYRE EN SERVO *	4
2.3 HVAD NU HVIS...? *	8
2.4 FLYVER RAKETTEN OPAD ELLER NEDAD? **	9
2.5 BYG EN FALDSKÆRM **	12
2.6 FALDSKÆRM OG UDLØSER I RAKET ***	12
<b>3. HVOR HØJT OVER JORDEN? *</b>	<b>14</b>
3.1 HVOR HØJT FLYVER RAKETTEN? **	18
3.2 HVOR MEGET BETYDER VINGER? *	22
<b>4. ACCELERATION</b>	<b>23</b>
<b>5. KILDER:</b>	<b>24</b>

De fleste kan bruge teknologi for eksempel en smartphone eller ipad. I projektet skal vi prøve at skabe teknologi selv. Det kan være svært. Du skal derfor ikke blive frustreret, hvis der er noget

som driller i projektet. Prøv at tage det som en udfordring, som vi nok skal få løst. Søg hjælp i tekst/video, på holdet, YouTube, blogs eller hos din lærer.

Overordnede mål:

Selvstændig problemløsning

Skabe ved hjælp af teknologi

Anvende naturvidenskabelig metode (trial and error)

Tegn på at målene er opfyldt:

Raketten flyver og måler noget

Raketten flyver bedre (højere, hurtigere, mere sikkert)

Nogen oplever glæde

## 2. Bevægelse \*

Målet med denne opgave er at lave en del, som udløser en faldskærm, når raketten falder mod Jorden og ikke før. Udløseren er en bevægelig del.

Ligesom muskler får kroppen til at bevæge sig, er det motorer der giver robotter bevægelse. Når en robothånd skal gribe om et glas, skal leddene i hånden styres af en motor, som kan stoppe ved bestemte positioner og blive dér, til den får besked om at starte igen. Sådant en motor kaldes en servo-motor eller bare **servo**.

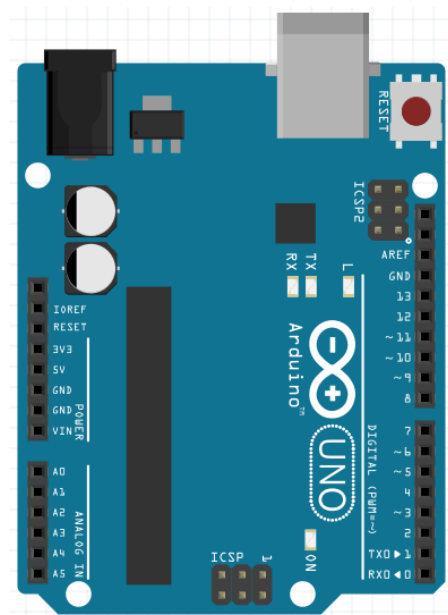


Figur: Servo

[https://commons.wikimedia.org/wiki/File:Micro\\_servo.jpg?uselang=da](https://commons.wikimedia.org/wiki/File:Micro_servo.jpg?uselang=da)

Af oomlout (SERV-03-MI (Micro Servo)) [CC BY-SA 2.0 (<http://creativecommons.org/licenses/by-sa/2.0>)], via Wikimedia Commons

En servo kan styres ud fra **sensor**-målinger. Det kan for eksempel være afstanden til et glas, der måles af en afstands-sensor. Målingerne sendes til og behandles i en **Arduino**, som er en lille computer, der også kan levere strøm (en strømforsyning). Hvis målingerne viser at glasset er tæt nok på, så starter en servo i robothånden, og den lukker.



Figur: Arduino UNO

## 2.1 Styre en servo \*

For at styre en servo, skal vi først prøve at bruge en **skydemodstand** (potentiometer). Den kan se ud på lidt forskellige måder, men har altid tre ben, som stikker ud:



[https://commons.wikimedia.org/wiki/File:10k\\_Ohm\\_Breadboard-Compatible\\_Potentiometer.jpg](https://commons.wikimedia.org/wiki/File:10k_Ohm_Breadboard-Compatible_Potentiometer.jpg)

eller



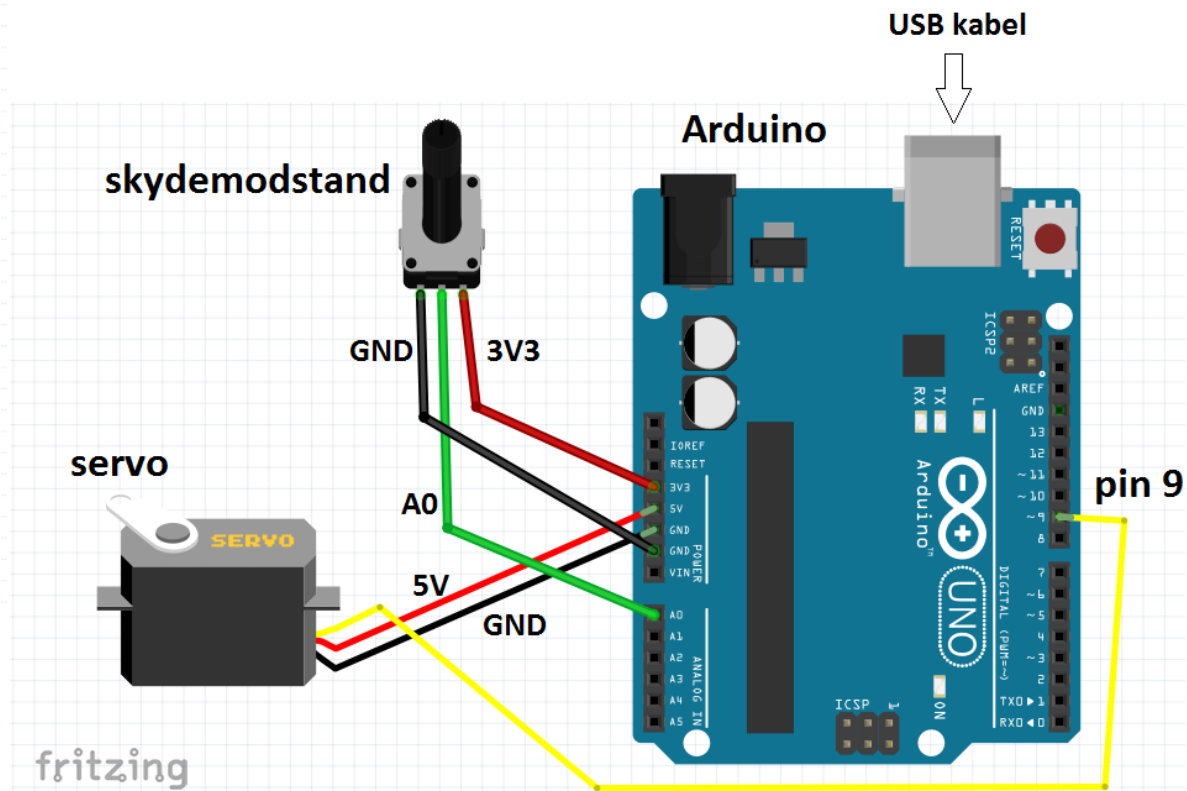
<https://commons.wikimedia.org/wiki/File:e%3APotentiometer.jpg>  
Af Iainf (Egen affotografering) [GFDL  
(<http://www.gnu.org/copyleft/fdl.html>),  
CC-BY-SA-3.0)

En skydemodstand skifter modstand, når den bliver drejet. Når modstanden skifter, ændres **spændingen** på det midterste ben ifølge **Ohms lov** ( $U=I \cdot R$ ). Spændingen måles af Arduinoen og omsættes til en værdi mellem 0 og 700, hvor værdien 0 svarer til en spænding på 0 volt (V) og værdien 700 til en spænding på 3 V.

Et program i Arduinoen lagrer værdien i en variabel og fortæller derefter servoen, hvor meget den skal dreje.

Arduinoen får i første omgang strøm via et USB kabel, og kan selv levere strøm (3 V eller 5 V). Alle elektriske **komponenter** (for eksempel en servo) skal forbindes i et **kredsløb**, så der sendes strøm fra Arduinoen gennem komponenterne og til **GND** (ground).

- Saml kredsløbet som vist nedenfor. Ledningerne er ikke nødvendigvis i de farver som er vist her.



Fra servoen skal den mørkeste ledning forbindes til GND, den røde ledning til 5V og den lyse ledning til pin 9.

Skydemodstanden (10 k $\Omega$ ) skal forbindes med det midterste ben til pin A0, det venstre til GND og det højre til 3V3 (3 V).

I videoen herunder kan du se:

**VIDEO: sætte kredsløbet op, analysere programmet**  
<https://youtu.be/xpHrYXh4PI0>



Når kredsløbet er samlet, skal vi bruge et **program** på Arduinoen, som er den opskrift, Arduinoen skal følge.

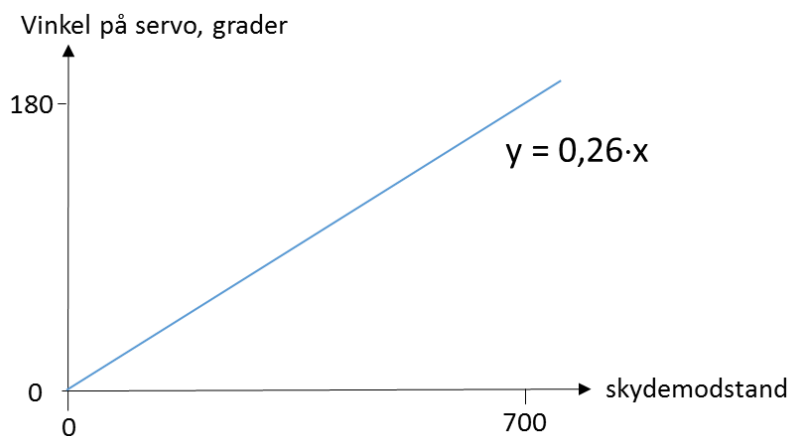
En Arduino kan programmeres fra en computer via software, som hedder **Arduino IDE**. Det er gratis, og kan downloades her: <http://www.arduino.cc/>. Du kan se her, hvordan man downloader og installerer det på PC: <http://somup.com/cDe01guv8>.

Et program til Arduino består af tre dele, som du kan se nedenfor.

I den grønne del oprettes variable og konstanter, og hjælpeprogrammer kaldes. I den blå del (void setup) oprettes forskellige forbindelser for eksempel mellem servo og Arduino. De to dele (grøn og blå) kører kun en gang, når programmet starter.

Den røde del (void loop) gentages, så længe programmet kører. Her måles spændingen over skydemodstanden, der beregnes, og servoen får besked om hvor meget den skal dreje.

Vores servo kan kun dreje 180 grader. Så værdien fra skydemodstanden mellem 0 og 700 skal omregnes til en vinkel mellem 0 og 180 grader.



Vi siger at de to størrelser er **proportionale**, dvs. følger en ret linje gennem 0,0. Kender vi værdien fra skydemodstanden (x), kan vi nemt beregne den vinkel servoen skal dreje til (y).

Betragt programmet nedenfor. Når der står to skrå streger i programmet //, betyder det at der kommer forklarende tekst.

- Hvor i programmet lagres målingen fra skydemodstanden?
- Har du fundet det sted hvor vinklen beregnes i programmet?
- Hvor får servoen at vide hvor meget den skal dreje?

### Program: Styre servo med skydemodstand

```
int value; // variabel som gemmer målingen fra skydemodstanden
int angle; // variabel til vinklen, som servo-motoren skal dreje
int pauseTid = 1000; //tid i millisekunder (1/1000 sekund)

#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9); // forbinder servo-motoren (objekt) med pin 9
  Serial.begin(9600); // åbner forbindelse til computeren
}
```

```

void loop() {
  value = analogRead(A0); // læser værdien fra skydemodstanden på analog pin 0
  Serial.print("value: "); // skriver tekst på computeren, hvis seriel monitor er åben
  Serial.print(value); // skriver værdien på computeren

  angle = 0.26*value; // omregner den målte værdi til et tal mellem 0 og 180 grader.

  Serial.print(", vinkel: ");
  Serial.println(angle); // printer vinklen på computeren

  myServo.write(angle);
  delay(pauseTid); // pause mellem hver måling
}

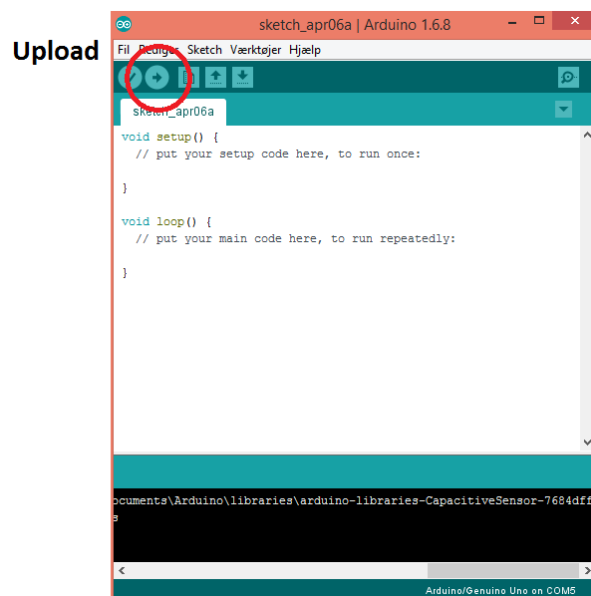
```

Nu skal vi have forbindelse mellem Arduino IDE på din computer og Arduinoen på bordet foran dig. Forbind dem med et USB-kabel.

- Forbind Arduino og computer med et USB-kabel.

Du skal ikke forsøge at installere drivere eller lignende. Åbn Arduino IDE. I menupunktet Værktøjer (Tools), vælg det rigtige Board (Arduino Uno) og Port, som vist i linksene nedenfor: PC (Windows): <https://www.youtube.com/watch?v=aECi1xoAjj0>. Mac OSX: <https://youtu.be/gxBKwNEQam4?t=4m38s>

- Åbn Arduino IDE, hvis du ikke har gjort det, og slet indholdet i det vindue som åbner.
- Kopier programmet ovenfor ind i Arduino IDE på computeren. Hvis du kopierer fra en Mac, skal du muligvis have teksten gennem en simpel editor for eksempel "Noter".
- I Arduino IDE: Klik på upload og vent.



I videoen herunder kan du se:

[Video : oploade til Arduino, dreje](#)

<https://youtu.be/O9boVISg49Y>

- Åbn **seriel monitoren** (vises som rød cirkel herunder) og se hvilke værdier der læses fra skydemodstanden, og lagres i value variabelen, samt hvilken vinkel servoen får besked om at dreje til (angle).



- Drej på skydemodstanden, og se hvad der sker.
- Hvor lang pause er der mellem hver måling?
- Gør pausen i programmet 10 gange kortere og upload programmet til Arduino igen. Hvordan opfører servoen sig nu?

I videoen herunder kan du se:

**Video: åbn seriel monitoren og ændre i programmet**

<https://youtu.be/uFl70j7zOzg>

Hvis skydemodstand eller servo bliver ved med at opføre sig underligt, kan du prøve en anden, evt. en helt anden type skydemodstand, som du kan se her:

<https://youtu.be/ahp14Fl5flU>

Du har nu lavet dit (måske) første kredsløb, og styret en servo-motor ved hjælp af et computerprogram. Servoen kan styres af mange input for eksempel lys, lyd, temperatur eller bevægelse.

## 2.3 Hvad nu hvis...? \*

Det er tit gavnligt, at kunne give betingelser for, hvornår noget skal ske. For eksempel skal vores raket udløse en faldskærm, når raketten falder nedad og ikke før.



Vi vil nu ændre programmet fra forrige afsnit, så servoen først drejer, hvis målingen fra skydemodstanden, når en vis tærskelværdi, som vi kalder **threshold**.

- Upload det nye program nedenfor til Arduinoen, drej skydemodstanden og se hvad der sker. Følg evt. med på seriel monitoren.

### Program: Styre servo, når skydemodstand viser vis værdi

```
int value; // variabel som gemmer målingen fra skydemodstanden
int angle; // variabel til vinklen, som servo-motoren skal dreje
int threshold = 90; // grænsen for hvornår servoen skal reagere

#include <Servo.h>
Servo myServo;

void setup() {
  myServo.attach(9); // forbinder servo-motoren (objekt) med pin 9
  Serial.begin(9600); // åbner forbindelse til computeren
}

void loop() {
  value = analogRead(A0); // læser værdien fra skydemodstanden på analog pin 0
  Serial.print("value: "); // skriver tekst på computeren, hvis seriel monitor er åben
  Serial.print(value); // skriver værdien på computeren

  angle = 0.26*value; // omregner den målte værdi til et tal mellem 0 og 180 grader.

  Serial.print(", vinkel: ");
  Serial.println(angle); // printer vinklen på computeren

  if (angle>threshold) {
    myServo.write(angle); // drejer serv-motoren til vinklen
  }

  delay(500); // 0,5 sekunders pause
}
```

- Skift værdien af konstanten threshold i den grønne del af programmet, og observer effekten.

I videoen herunder kan du se:

Video: indføre med en if-sætning

<https://youtu.be/061zFmAsjec>

Vi har nu set en forgrening (if-sætning), så programmet kun udfører kommandoer, hvis bestemte betingelser er opfyldt.

## 2.4 Flyver raketten opad eller nedad? \*\*

En drone kan nemt komme til at flyve skævt og styrte. En **gyro** er en sensor, der måler ændringer i vinkel. Det kan for eksempel bruges, til at holde en drone mere stabil i luften, da gyroen måler

om dronen tilter eller drejer. Dronen kan derefter undgå at styrte, ved at dreje nogle propeller mere end andre.

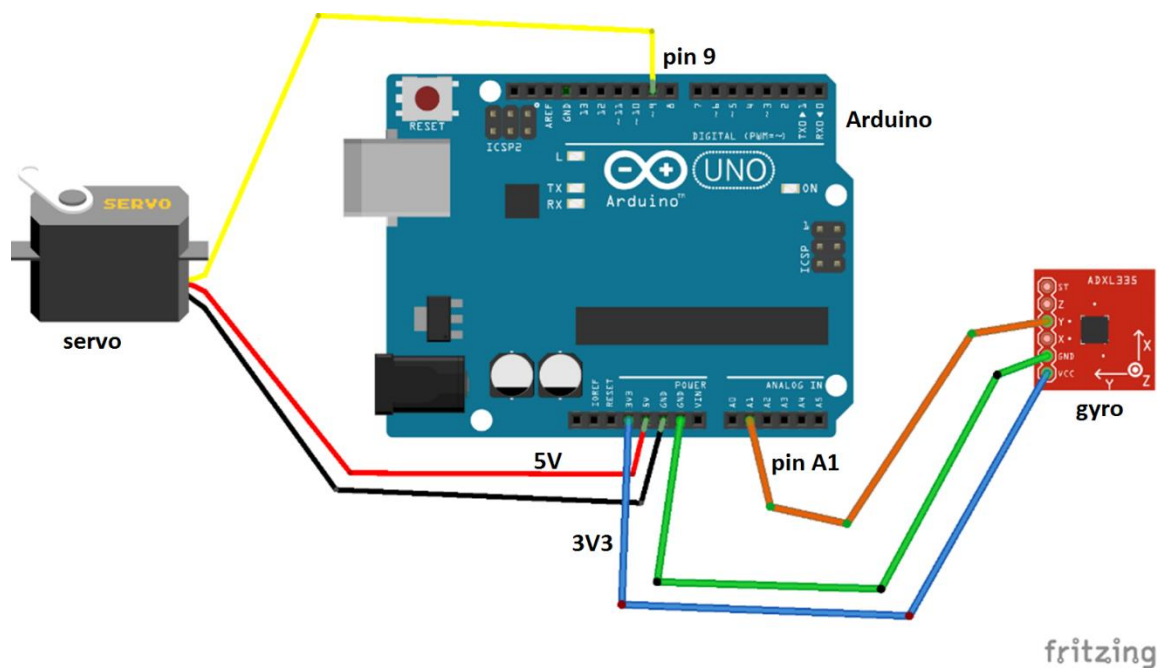


Figur: Drone

By Simon Jardine from Sufers Parasite, Australia (Quadcopter Uploaded by Yarl) [CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0/>)], via Wikimedia Commons [https://commons.wikimedia.org/wiki/File%3AQuadcopter\\_-\\_20091111.jpg](https://commons.wikimedia.org/wiki/File%3AQuadcopter_-_20091111.jpg)

Vi skal bruge en gyro, til at afgøre om vores raket flyver opad eller er på vej nedad. Ved affyring vil spidsen af raketten pege opad. Spidsen af raketten er tungest, fordi vi placerer batterier med mere dér. Derfor vil raketten også pege spidsen nedad, når alt vandet er ude, og den falder mod Jorden. Når det sker, skal en servo dreje, så en faldskærm bliver udløst.

- Saml kredsløbet nedenfor. Farverne er ikke nødvendigvis de samme som vist.



Forbindelserne mellem Gyro og Arduinoen:

På GY-61	På Arduino:
VCC	3V3 (3V spænding, IKKE 5V!)
GND	GND (ground)
Y_OUT	analog pin A1

I videoen herunder kan du se:

Video 4: samle kredsløbet med gyro

<https://youtu.be/ZIdAsbckpWQ>

Gyroen giver en spænding alt efter den vinkel den er placeret i. Spændingen kan omregnes til en vinkel mellem 0 og 180 grader. Vi vil dog bruge de rå data. Nedenfor kan du se programmet vi skal bruge.

### Program: Styre servo med gyro

```
int y; // variabel for gyroens vinkel
int threshold = 225; // variabel til den værdi hvor gyroen svinger fra opad til nedad
int vinkel = 110; // variabel til den vinkel servoen skal dreje til
int pauseTid = 500; // millisekunders pause

#include <Servo.h>
Servo myservo; // skaber servoen som et objekt

void setup() {
  myservo.attach(9); // forbinder servoen med digital pin 9
  myservo.write(0); // nulstiller servoens position
  delay(pauseTid);
  Serial.begin(9600);
}

void loop() {
  y = analogRead(A1); // gyroens vinkel læses fra analog pin A1
  Serial.print(" y = ");
  Serial.println(y);

  if (y>threshold)
  {
    myservo.write(vinkel);
  }

  delay(pauseTid);
} // end loop
```

Det afgørende sted i programmet ser sådan ud:

```
if (y>threshold)
{
  myservo.write(vinkel);
}
```

- Forklar hvad der sker i program-stumpen ovenfor.
- Kopier programmet ovenfor ind i Arduino IDE og upload det til Arduinoen.
- Hvad sker der på seriel monitoren? Find den akse, som ændrer værdien, når du langsomt drejer gyroen.
- Hvilken største og mindste værdi svinger variabelen y mellem, når gyroen drejes langsomt?
- Drej gyroen. Hvad sker der med servoen?
- Skift konstanten threshold i programmet og evt. > tegnet, så servoen drejer, når gyroen langsomt skifter fra opad til nedad.

I videoen herunder kan du se:

Video: aflæse max og min for gyroen. Ændre i programmet, så servo først drejer når..

<https://youtu.be/w4u1a8RTmtA>

Du har nu bygget et system, som kan afgøre om raketten flyver opad eller nedad, og er klar til at udløse en faldskærm, når raketten er på vej nedad.

## 2.5 Byg en faldskærm \*\*

Konstruer en faldskærm som vist her: <https://www.youtube.com/watch?v=nnZbji8Payc>

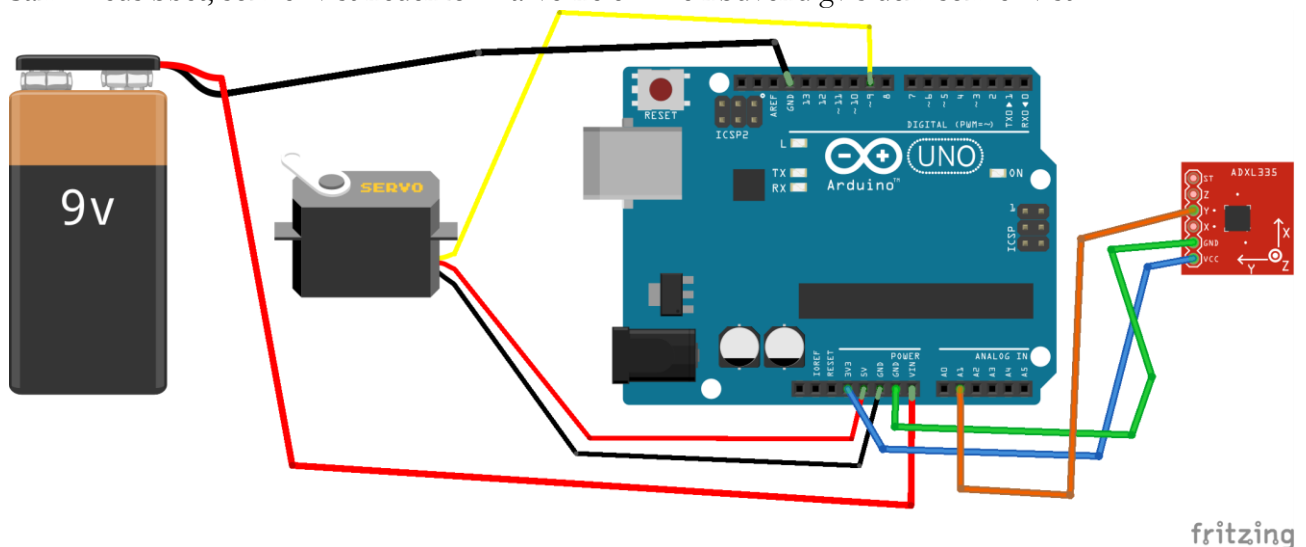
Test faldskærmen for eksempel ved at binde/tape et passende objekt under, og kaste den ud fra et højt sted.

Variér evt. faldskærmens design.

## 2.6 Faldskærm og udløser i raket \*\*\*

Når raketten flyver, er den ikke forbundet til Jorden via USB-kabel. Vi skal derfor gøre faldskærms-udløseren trådløs.

Saml kredsløbet, som er vist nedenfor. Farverne er ikke nødvendigvis dem som er vist.



Forbindelser:

9V batteri	På Arduino:
Rød ledning	Vin
Sort ledning	GND (ground)

En servo kræver høj spænding, særligt når den starter. Hvis den ikke virker, kan det være nødvendigt at bruge et nyt batteri eller parallel-forbinde en kapacitor (100  $\mu$ F).

- Fjern USB-kablet.
- Test at servoen stadig drejer, når gyroen drejes. Tryk evt. reset.
- Konstruer et rum, som faldskærmen kan ligge i, inden i en 1½ L vandflaske, som vist her: <https://www.youtube.com/watch?v=8JY6-vrV1Gg>

Husk at lave passende huller i flasken, så strømmen til Arduinoen kan afbrydes, og så elastikken, som skal holde døren, kan sættes rundt om servoen.

Husk at efterlade 3-4 cm uden indhold i flasken åbning, som kan passe oven på raketten.

I videoen herunder kan du se:

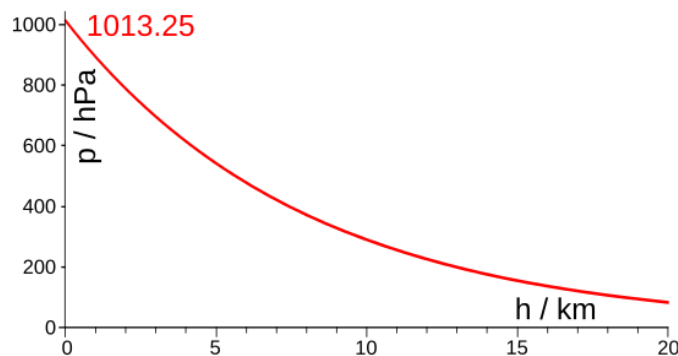
[Video: mit faldskærms modul](#)

Vi har nu fået bygget et modul til raketten, som kan udløse en faldskærm. Det betyder, at vi kan laste raketten med flere sensorer for eksempel kamera, vel vidende at de får en blød landing.

### 3. Hvor højt over Jorden? \*

Når vi ændrer på raketens design, er det blandt andet for at få den til at flyve højere. Det er meget svært at se, hvor højt en raket flyver. Derfor skal vi lave en dims, som kan måle det, så vi har et bedre grundlag for at sige, om vores ændringer i raketens design virker eller ej.

Luften omkring os har et **tryk**, som vi kalder  $p$ . Trykket findes fordi partikler i luften bl.a. ilt,  $\text{CO}_2$  og vand støder ind i alting. Højden over Jordens overflade kalder vi  $h$ . Jo højere man kommer op i luften over Jorden, desto lavere bliver trykket, fordi der bliver færre partikler. Du kan se sammenhængen mellem tryk og højde på grafen her:



Kilde: [https://commons.wikimedia.org/wiki/File:Pressure\\_air.svg#/media/File:Pressure\\_air.svg](https://commons.wikimedia.org/wiki/File:Pressure_air.svg#/media/File:Pressure_air.svg)

Sammenhængen mellem tryk og højde er udtrykt i en formel:

$$p = p_0 \cdot e^{-M \cdot g \cdot h / R \cdot T}$$

Hvor

$p$  = atmosfærens tryk i Pascal (Pa)

$p_0$  = starttrykket i Pascal (Pa)

$T$  = temperaturen i Kelvin (K)

$h$  = højde i meter

$R$  = gaskonstanten for luft =  $8,3 \text{ N} \cdot \text{m} / \text{mol} \cdot \text{K}$

$g$  = tyngdeaccelerationen =  $9.8 \text{ m} / \text{s}^2$

$M$  = luftens molarmasse =  $0,0289 \text{ kg} / \text{mol}$

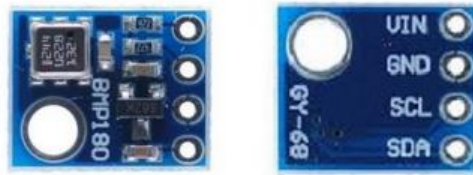
Kilde: [https://da.wikipedia.org/wiki/Barometrisk\\_h%C3%B8jdeformel](https://da.wikipedia.org/wiki/Barometrisk_h%C3%B8jdeformel)

Formlen kan omskrives til:

$$h = -R \cdot T / M \cdot g \cdot \ln(p / p_0)$$

Det betyder, at hvis vi kender starttrykket  $p_0$ , temperaturen  $T$  og trykket  $p$  i en vis højde  $h$ , kan vi beregne denne højde. Vi vil i det følgende antage at temperaturen er konstant.

For at måle trykket, bruger vi en tryk-sensor (trykmåler), som hedder BMP180.

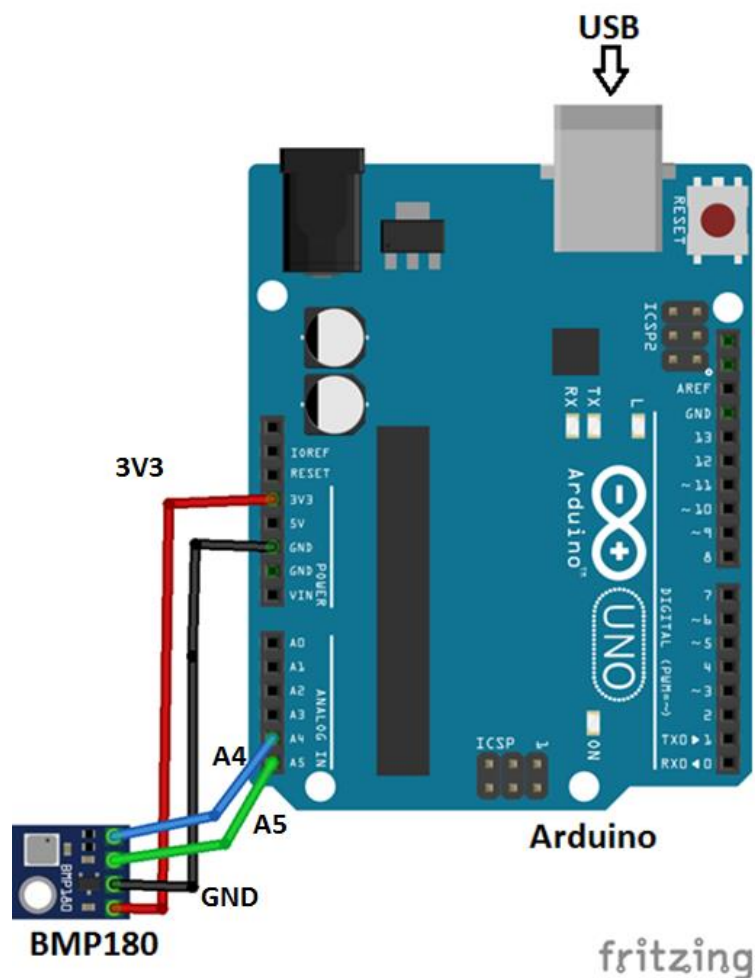


<http://artofcircuits.com/provict/bmp180-digital-barometric-sensor-movile-model-BMP180>

Målingerne fra tryksensoren sendes til og behandles i en **Arduino**, som er en lille computer, der også kan levere strøm (en strømforsyning).

Arduinoen får i første omgang sin strøm via et USB kabel. Alle elektriske **komponenter** (for eksempel en tryksensor) skal forbindes i et **kredsløb**, så der sendes strøm fra Arduinoen (3V eller 5V), gennem komponenterne og til GND (ground).

Tryksensoren BMP180 skal forbindes til Arduino som vist herunder.



Forbindelserne mellem BMP180 og Arduinoen:

På BMP180	På Arduino:
-----------	-------------

VIN	3V3 (3V spænding, IKKE 5V!)
GND	GND (ground)
SDA	analog pin A4
SCL	analog pin A5

- Saml kredsløbet ovenfor. Ledninger behøver ikke nødvendigvis være de farver som er vist.

I videoen herunder kan du se:

Video: samle kredsløb med BMP180

[https://youtu.be/51bra\\_XQBO0](https://youtu.be/51bra_XQBO0)

Når kredsløbet er samlet, skal vi bruge et **program** på Arduinoen, som er den opskrift Arduinoen skal følge.

En Arduino kan programmeres fra en computer via et program, som hedder **Arduino IDE**. Det er gratis, og kan downloades her: <http://www.arduino.cc/>. Du kan se her, hvordan man downloader og installerer det på PC: <http://somup.com/cDe01guv8>.

Et program til Arduino består af tre dele, som du kan se nedenfor.

I den grønne del oprettes variable og konstanter, og hjælpeprogrammer kaldes. I den blå del (void setup) oprettes forskellige forbindelser for eksempel mellem computer og Arduino. De to dele (grøn og blå) kører kun en gang, når programmet starter.

Den røde del (void loop) gentages, så længe programmet kører. Her måles trykket og højden beregnes.

Betragt programmet nedenfor. Når der står to skrå streger i programmet //, betyder det at der kommer forklarende tekst.

- I hvilken linje i programmet måles trykket?
- I hvilken linje i programmet beregnes højden?
- Hvorfor er der både en variabel til pressure og en til startpressure?
- Hvordan omregnes fra temperaturen fra grader Celcius til Kelvin?

### Program: Mål trykket og beregn højden

```
double pressure; //Variabel til tryk i Pascal (Pa)
double startpressure; //Variabel til tryk i Pascal (Pa)
double tempC; //Variabel til temperatur i grader Celcius
double starttempC; //Variabel til temperatur i grader Celcius
double hojde; // variabel til højden i meter over havet
double starttempK; //Variabel til temperaturen i Kelvin

const double gaskonstant = 8.31; // gaskonstanten i N*m/mol*K
const double tyngdeacceleration = 9.807; //tyngdeaccelerationen i m/s^2
const double molarmasse = 0.02896; //Molarmassen af luft i kg/mol

#include "Wire.h" // åbner biblioteker for at tale med BMP180 chippen (I2C)
#include "Adafruit_BMP085.h"
Adafruit_BMP085 mySensor; // laver et objekt kaldet mySensor

void setup(){
Serial.begin(9600); //åbner forbindelse til serial monitor
```



```

mySensor.begin(); // starter tryk-sensoren mySensor
starttempC = mySensor.readTemperature(); // Måler STARTtemperaturen i grader C fra BMP180
startpressure=mySensor.readPressure(); // Måler STARTtrykket i Pa
starttempK = starttempC + 273; // temperaturen i Kelvin
}

void loop() {
tempC = mySensor.readTemperature(); // Måler temperaturen i grader C
pressure=mySensor.readPressure(); // Måler trykket i Pa
hojde = -(starttempK*gaskonstant/(molarmasse*tyngdeacceleration)) * log(pressure/startpressure);
// beregner højden over start-niveau i meter

Serial.print("Tryk: ");
Serial.print(pressure);
Serial.println(" Pa");
Serial.print(" Højde: ");
Serial.print(hojde);
Serial.println(" meter");
delay(250);
}

```

Nu skal vi have forbindelse mellem Arduino IDE på din computer og Arduinoen på bordet foran dig.

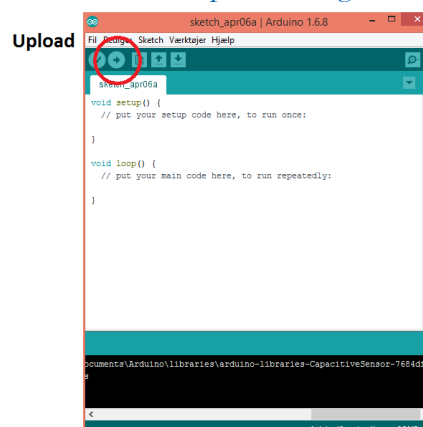
- Forbind Arduino og computer med et USB-kabel.

Du skal ikke forsøge at installere drivere eller lignende. Åbn Arduino IDE. I menupunktet Værktøjer (Tools), vælg det rigtige Board (Arduino Uno) og Port, som vist i linksene nedenfor:

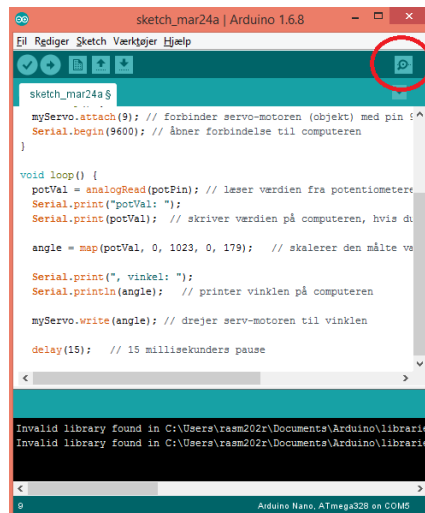
PC (Windows): <https://www.youtube.com/watch?v=aECi1xoAjj0>

Mac OSX: <https://youtu.be/gxBKwNEQam4?t=4m38s>

- Åbn Arduino IDE, hvis du ikke har gjort det, og slet teksten i det vindue, som åbner.
- Kopier programmet ovenfor til Arduino IDE, og upload til Arduino, som vist herunder. Hvis du kopierer fra en Mac, skal du muligvis have teksten gennem en simpel editor for eksempel "Noter". Du skal muligvis installere nogle ekstra biblioteker (Adafruit\_BMP085.h, som kan downloades her: <https://learn.adafruit.com/bmp085/using-the-bmp085>)



- Åbn **seriel-monitoren** (vises som rød cirkel herunder) og check at trykket måles og højde beregnes. Ser de rimelige ud? (Normalt tryk ved jordoverfladen er ca. 101300 Pa)



Åbn serial monitor

- Løft trykmåler og computer fra gulv til så højt I kan nå. Sæt derefter trykmåler og computer ned på gulvet, hvis det er muligt. Ændres målingerne som forventet?

I videoen herunder kan du se:

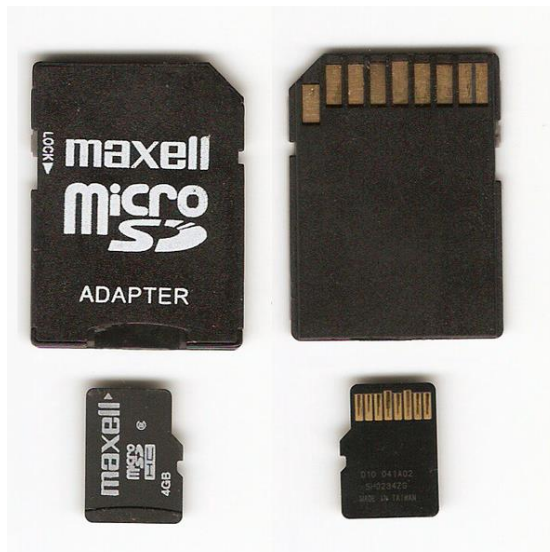
**Video: Måle tryk og højde med BMP180**

<https://youtu.be/jkV5IwlATAs>

Du har nu lavet dit (måske) første kredsløb, som kan måle trykket og beregne højden. Der er rigtig mange andre sensorer som kan forbindes til Arduino, og måle under en flyvetur i en raket.

### 3.1 Hvor højt flyver raketten? \*\*

En raket kan ikke forbindes til en computer på Jorden via USB-kabel. Derfor skal de højder vi beregner gemmes på et mikro SD-kort.

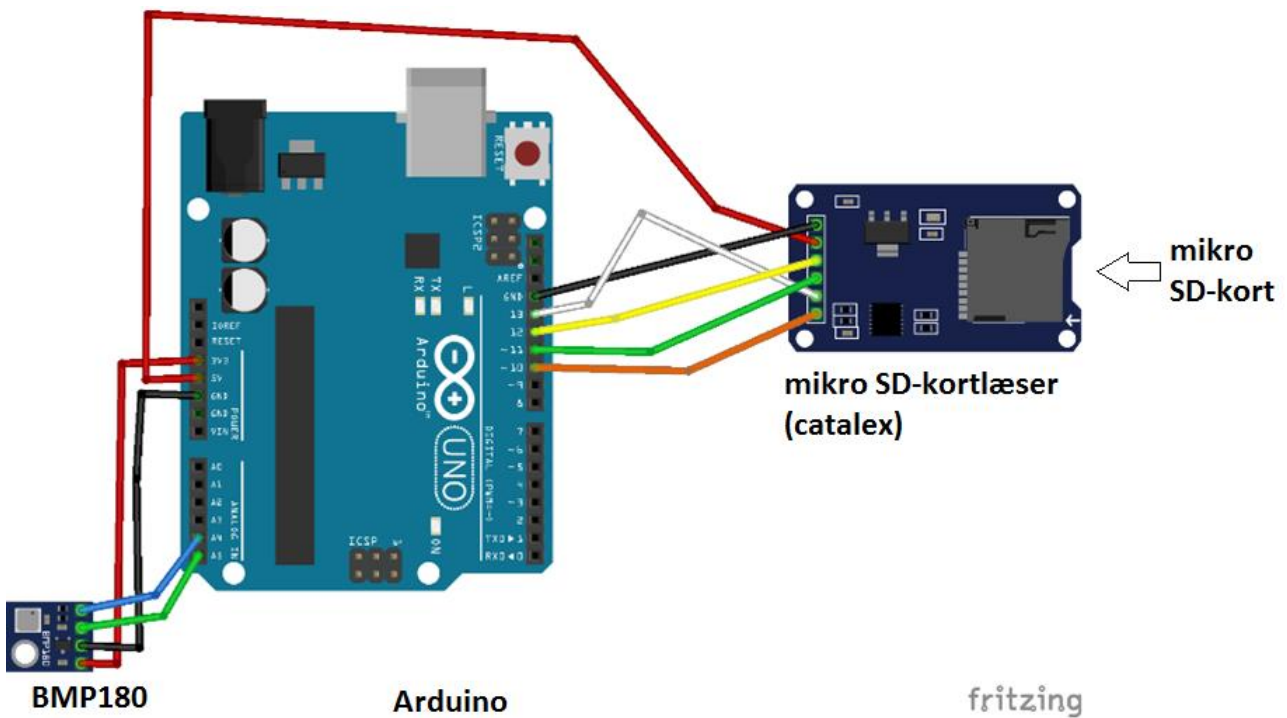


[https://commons.wikimedia.org/wiki/File:Adaptador\\_MicroSD.png?uselang=da](https://commons.wikimedia.org/wiki/File:Adaptador_MicroSD.png?uselang=da)

Et mikro SD-kort kan læses i de fleste computere via en adapter, som du kan se på billedet ovenfor. Vi vil gemme højderne i filen "HøjdeData.txt", som kan åbnes for eksempel i Notesblok (notepad) på PC og TextEdit på Mac OSX.

Mikro SD-kortet skal forbindes til Arduino gennem en **kortlæser**.

- Saml kredsløbet nedenfor:



Forbindelserne mellem trykmåler og Arduino kan ses i forrige afsnit.

Forbindelser mellem kortlæser og Arduino:

På SD-kortlæseren (Catalex)	På Arduino:
VCC	5V
GND	GND (ground)
MISO	Pin 12
MOSI	Pin 11
SCK	Pin 13
CS	Pin 10

I videoen herunder kan du se:

**Video: saml kredsløb med SD card-reader**

<https://youtu.be/AxDx4sWZsCM>

Programmet, som skal gemme højden, ses herunder.

- Hvor i programmet bliver højden gemt?
- Hvor tit gemmer programmet højden?

**Program: Mål trykket, beregn højden og gem det på SD-kort**

```

double pressure; //Variabel til tryk i Pascal (Pa)
double startpressure; //Variabel til tryk i Pascal (Pa)
double tempC; //Variabel til temperatur i grader Celcius
double starttempC; //Variabel til temperatur i grader Celcius
double hojde; // variabel til højden i meter over havet
double starttempK; //Variabel til temperaturen i Kelvin
int pauseTid = 5000; //tid mellem hver måling

const double gaskonstant = 8.31; // gaskonstanten i N*m/mol*K
const double tyngdeacceleration = 9.807; //tyngdeaccelerationen i m/s^2
const double molarmasse = 0.02896; //Molarmassen af luft i kg/mol

#include <SD.h> // åbner biblioteker for at tale med SD-kortlæser og GY68 chippen (I2C)
#include<SPI.h>
#include "Wire.h"
#include "Adafruit_BMP085.h"
Adafruit_BMP085 mySensor; // laver et objekt kaldet mySensor

int chipSelect = 4; // Pin til SD kortlæser
File mySensorData; // laver et fil-objekt, som kan gemme data

void setup(){
Serial.begin(9600); //åbner forbindelse til serial monitor
mySensor.begin(); // starter tryk-sensoren mySensor
starttempC = mySensor.readTemperature(); // Måler STARTtemperaturen i grader C fra BMP180
startpressure=mySensor.readPressure(); // Måler STARTtrykket i Pa
starttempK = starttempC + 273; // temperaturen i Kelvin

pinMode(10, OUTPUT); // Pin 10 reserveres til SD kortlæseren.
SD.begin(4); // starter SD kortlæseren
  if (SD.exists("Trykdata.txt")) // hvis filen "Trykdata.txt" findes på SD-kortet..
  {
    SD.remove("Trykdata.txt"); // ..slet filen.
  }
}

void loop() {
tempC = mySensor.readTemperature(); // Måler temperaturen i grader C fra BMP180
pressure=mySensor.readPressure(); // Måler trykket i Pa
hojde = -(starttempK*gaskonstant/(molarmasse*tyngdeacceleration))*log(pressure/startpressure);

mySensorData = SD.open("Trykdata.txt", FILE_WRITE); // åbn filen på SD-kortet
if (mySensorData) {
Serial.print("Tryk: ");
Serial.print(pressure);
Serial.println(" Pa");
Serial.print(" Højde: ");
Serial.print(hojde);
Serial.println(" meter");
delay(pauseTid);

mySensorData.println(hojde); //skriv højden til SD-kortet
mySensorData.close(); //luk filen på SD-kortet
}
}

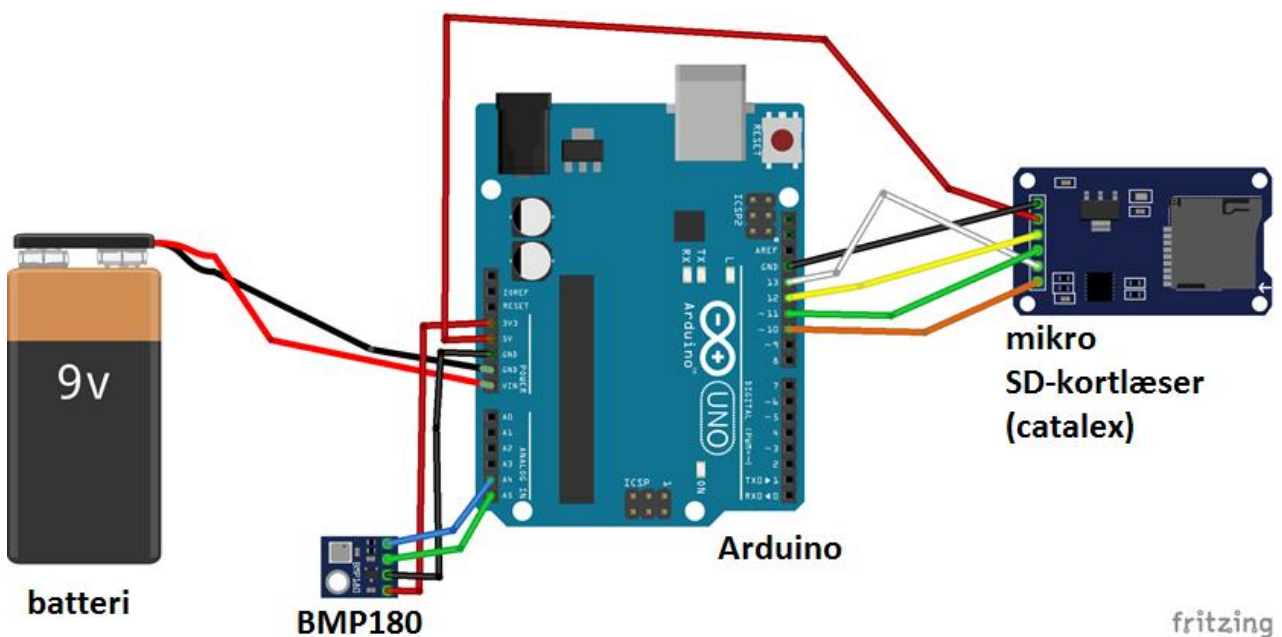
```

- Oplad programmet til Arduino. Du skal muligvis installere nogle ekstra biblioteker (SPI.h og SD.h)
- Check via seriel monitoren at programmet måler tryk og beregner højde, som forventet.
- Afbryd strømmen til Arduinoen, ved at fjerne USB-kablet. Tag *derefter* SD-kortet ud af kortlæseren.

- Sæt SD-kortet i en computer evt. ved hjælp af en adapter. Check at den beregnede højde er blevet gemt løbende på SD-kortet filen HøjdeData.txt. Filen kan åbnes i Notesblok (notepad) på PC og TextEdit på Mac OSX. Hvis der ikke er data på kortet, kan det være fordi, det er skrivebeskyttet og skal formateres i computeren først.
- Tag SD-kortet ud af computeren igen. Sæt det tilbage i SD-kortlæseren, mens strømmen til Arduinoen er afbrudt.

Kredsløbet er stadig ikke trådløst, når Arduinoen får strøm gennem USB-kablet fra computeren.

- Fjern USB-kablet, hvis du ikke allerede har gjort det. Forbind derefter Arduinoen til et 9V batteri som vist nedenfor.



Forbindelser:

9V batteri	På Arduino:
Rød ledning	Vin
Sort ledning	GND (ground)

Øvrige forbindelser mellem kortlæser, BMP180 og Arduino, kan ses i forrige afsnit.

- Check, som beskrevet ovenfor, at programmet stadig virker og gemmer nye højder på SD-kortet. Husk at afbryde strømmen til batteriet, før I tager SD-kortet ud.

I videoen herunder kan du se:

Video: check om der er data på Sdkort. Forbind til 9V og check igen.

<https://youtu.be/Qrmd8yt5os>

Nu skal alle delene til at bestemme højde sættes oven på raketten. Vi skal derfor lave et stativ af pap, eller et andet let materiale, som passer inde i bunden af ca. en halv 1½ L vandflaske. Der skal være 3-4 cm ved åbningen, som er uden stativ, da denne del skal sættes oven på raketten.

Alle dele til måling (batteri, Arduino, sensor, ledninger og SD-kortlæser) skal sættes fast på stativet –brug for eksempel ledning, ståltråd, tape eller elastik.

Sæt stativet fast inde i flasken, og sørg for at det ikke kan flyttes inde i flasken for eksempel ved at tilpasse størrelsen på flasken, så stativet sidder i spænd når flasken sættes oven på raketten.

Sørg for at man forholdsvis enkelt kan tage SD-kortet ud, og at man kan afbryde strømmen. Klip for eksempel små åbninger i flasken.

- Placer alle dele til måling på et stativ i en flaske, og fastgør det oven på raketten.

I videoen herunder kan du se:

**Video: placer det hele inde i en flaske.**

Vi har nu lavet en trådløs højdemåler, som kan placeres på raketten, og hjælpe med at afgøre om nye raketdesigns giver en bedre flyvehøjde.

### 3.2 Hvor meget betyder vinger? \*

---

Giv raketten vinger. Brug målinger til at afgøre betydningen for raketens egenskaber for eksempel hvor højt den flyver.

Varier vingernes placering, antal og/eller form. Brug igen målinger til at afgøre betydningen. Husk kun at variere en faktor ad gangen, så mængden af vand, trykket i raketten m.m. forbliver uændret.

Forsøg at forklare evt. forskelle mellem målingerne.

## 4. Acceleration

---

## 5. kilder:

---

<http://www.toptechboy.com> besøgt marts 2016

Shiloh, M. et al (red) ”The Arduino project book” Arduino 2013

Westerfield, M. ”Make: Rockets” Maker Media Inc. 2014