

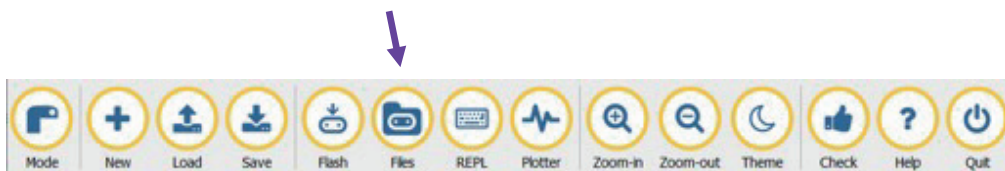
# Opplegg 12 – Momentan vekstfart og speedometer

Ein sonar sender ut eit lydsignal, og måler kor lang tid det tek før signalet blir reflektert attende til sonaren. Dersom ein veit kva lydfarten er, kan ein enkelt rekne ut avstanden til det som reflekterer lydsignalet. Sonaren veit kva lydfarten er, og vi kan dermed bruke han til å måle avstandar med. For å kunne bruke ein sonar i Python Mu, må vi importere eit bibliotek på sjølve micro:biten. Dette biblioteket er eigentleg eit program som gjer det enklare for oss å programmere sonaren, utan å måtte tenke på korleis alle detaljane i målingane.

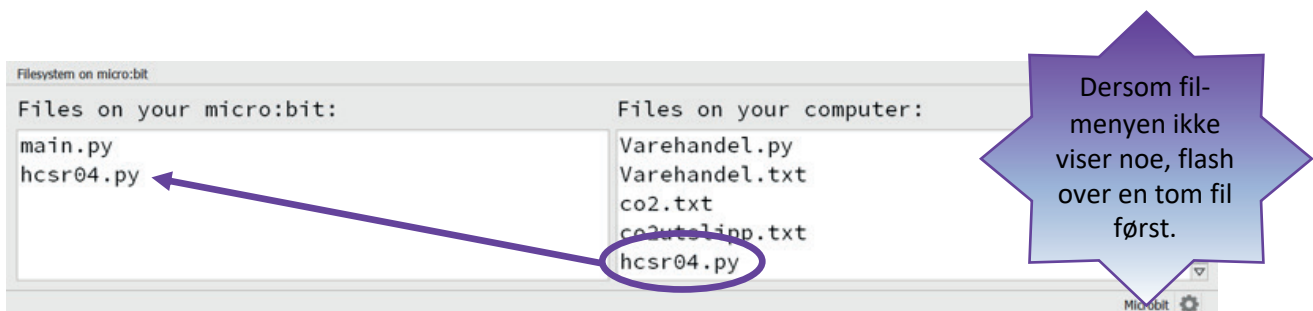
Det første vi må gjere er å få tak i biblioteket til sonaren, og det kan finnast på [www.kunnskapsfilm.no](http://www.kunnskapsfilm.no)

Kopier koden i filen «hcsr04.py» over i en ny fane (fil) i Python Mu og lagre den som «hcsr04.py» ved å trykke på «Save» og lagre den i katalogen som dukker direkte opp.

Deretter må denne fila leggst over på sjølve micro:biten, og det gjerest ved å kople micro:biten til datamaskinen, og trykke på «Files».



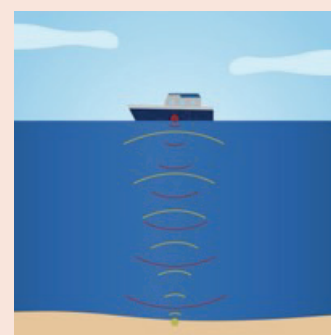
Da kjem denne menyen opp i Mu, og du må leite i feltet på høgre side til du finn fila som du nett lagra: «hcsr04.py». Når du finn henne, dreg du henne over i det venstre feltet. Da vil lyset på baksida av micro:biten blinke til programmet er overført.



Da står det att å lage sjølve programmet som bruker sonaren til å måle avstandar med. Under er det ei puslespill-oppgåve der kodelinene må ryddast i rett rekkefølge for å lage dette programmet.

## Puslespill-oppgave

```
print((HCSR04(),))  
from hcsr04 import HCSR04  
sleep(1000)  
while True:  
from microbit import *
```



# Lag et speedometer med micro:bit

## Oppgave

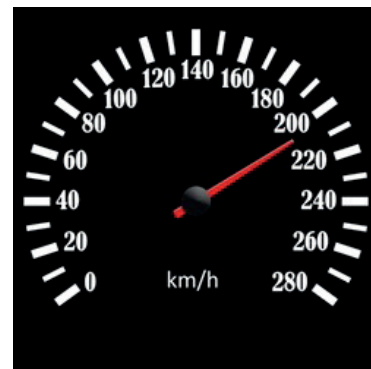
Bruk ein micro:bit med ein tilkoplta sonar til å lage eit speedometer. For å få til dette må de kalibrere avstandsmålingane frå sonaren slik at de får oppgitt avstanden i meter. Sjå tips nedst på sida for kalibrering.

**Fase 1:** Gjennomfør informasjonsinnhenting for å få idéar. Kva slags tilkoplingar trengs?

## Fase 2: Idémyldre og planlegge

Det er viktig at de er opne for alle slags idear og ikkje er for kritiske, da kan nyttige framlegg bli kutta ut for tidleg.

1. Tenk sjølv først og teikn gjerne skisser.
2. Forklar ideen din for dei andre på gruppa.
3. Heile gruppa diskuterer dei ulike ideane, og lagar ein felles plan for bygging.



## Fase 3: Gjennomføring

- Lag speedometeret.
- Lag programmet for å måle avstanden med hjelp av sonaren. Hugs at resultatet av kalibreringa må inkludrast i programmet
- Utvid programmet slik at det kan berekne momentanfarten frå dei kalibrerte avstandsmålingane.

**Fase 4:** Test om speedometeret, inkludert programmeringa, verkar som planlagt.

**Fase 5:** Samanlikn resultatata med andre i klassen.

- Verkar svara deira fornuftige?

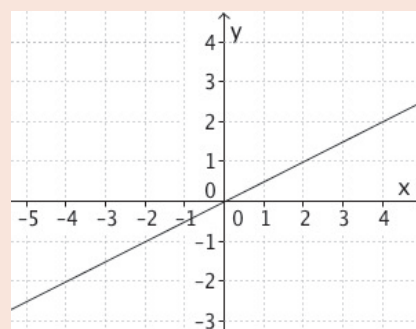
**Fase 6:** Gå attende til dei andre fasane for å gjere eventuelle forbetringar.

**Fase 7:** Dokumenter prosjektet på ein valfri måte.



## Kalibreringsoppgåve – regresjon

1. Plott alle micro:biten sine avstandsmålingar og dei tilhøyrande målingane de gjer med linjal eller metermål i Geogebra.
2. Finn en matematisk modell med å føreta ein regresjon
3. Korleis ser funksjonen ut?
4. Passar han bra med datapunkta dykkar?



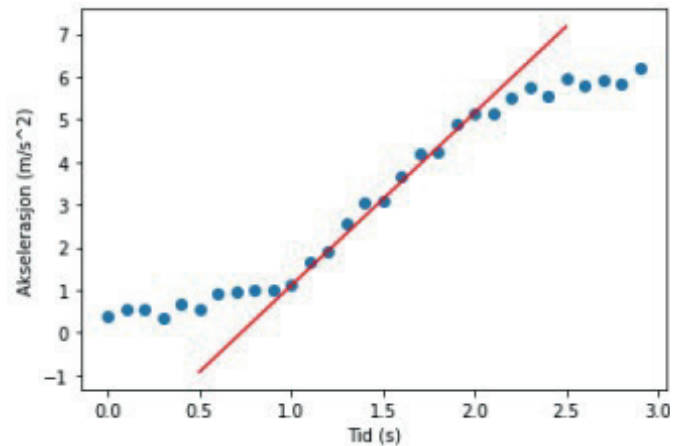
# Momentan vekstfart

## To former for vekstfart

Vi skil mellom gjennomsnittleg og momentan vekstfart. Den gjennomsnittlege vekstfarten fortel oss kor raskt noko aukar mellom to x-verdiar, medan den momentane vekstfarten fortel oss kor rask aukinga er i eit visst punkt. Vi byrjar med den gjennomsnittlege.

## Gjennomsnittleg vekstfart

I følgjande kode går vi ut frå at vi alt har importert eit datasett med x- og y-verdiar. Vi startar med å plote punkta. Deretter bereknar vi den gjennomsnittlege akselerasjonen med å bruke  $a = \Delta x / \Delta y$ . I figuren vår var  $N = 30$ , og vi ønsker å berekne den gjennomsnittlege vekstfarten for indeksverdiar mellom 10 og 20.



```
scatter(x,y)
xlabel("Tid (s)")
ylabel("Akselerasjon (m/s^2)")

aks_gj = (y[20]-y[10])/(x[20]-x[10])
```

Plottar x- og y-verdiane. Bereknar deretter den gjennomsnittlege akselerasjonen.

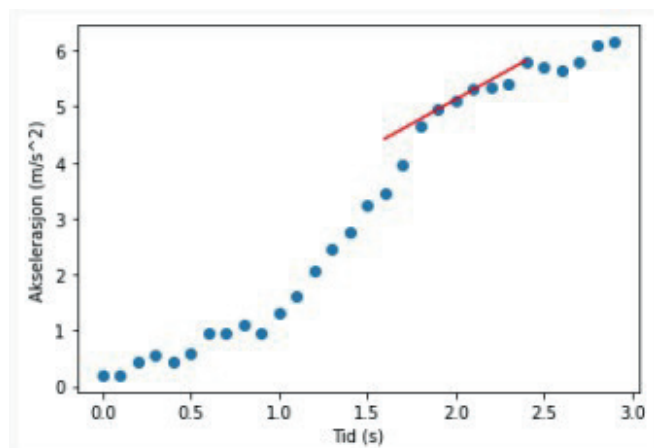
```
def f(x,a,x1,y1):
    return a*(x-x1)+y1

s = linspace(0.5,2.5)
t = f(s,aks_gj,x[10],y[10])
plot(s,t,'r')
```

Bruker eittpunktsformelen til å finne funksjonsuttrykket til den raude lina på figuren. Til sist plotter vi lina.

## Momentan vekstfart

Dersom vi vil finne den momentane vekstfarten, gjer vi akkurat som for den gjennomsnittlege, men vi vel berre to punkt som ligg rett i nærleiken av kvarandre. Dermed får vi eit uttrykk for korleis funksjonsverdiane er i ferd med å endre seg akkurat der. I dømet nedanfor har vi valt indeksverdiar 19 og 21 for å sjå på veksten rundt  $i=20$ , altså  $x=2,0$ .



Plottar grafen og bereknar  
gjennomsnittleg akselerasjon for  
x-verdiar i nærleiken av kvarandre.



```
scatter(x,y)
xlabel("Tid (s)")
ylabel("Akselerasjon (m/s^2)")
aks_gj = (y[21]-y[19])/(x[21]-x[19])
```

Bruker eittpunkts-  
formelen og plottar  
den raude linja.

```
def f(x,a,x1,y1):
    return a*(x-x1)+y1
s = linspace(1.6,2.4)
t = f(s,aks_gj,x[19],y[19])
plot(s,t,'r')
```

