



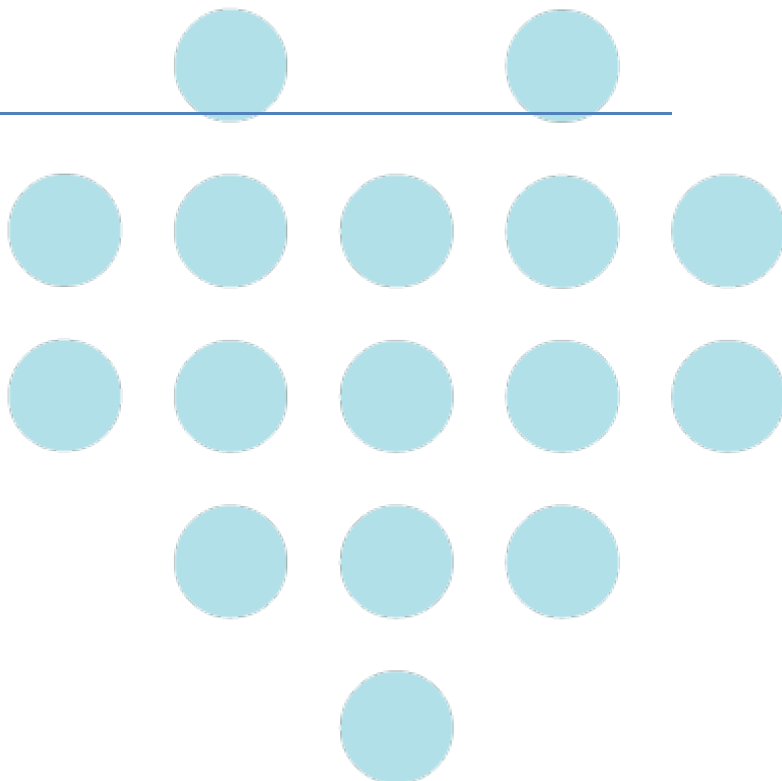
## super:bit-oppgraget

---

### Lærerveiledning

---

Versjon 1, august 19.



## Innhold

1	Oppvarming – analog programmering (10 minutter) .....	3
2	Kom i gang med micro:bit (15 minutter) .....	5
3	Kjør en meter med BitBot.....	6
4	Kjør ei løype med BitBot .....	7
5	Programmering av trafikklys.....	8
6	Robotplenklipper .....	10

## Om opplegget

Opplegget gir en introduksjon for elevene i praktisk bruk av programmering, der de vil løse forskjellige oppdrag knyttet til smarte byer. Hvor lang tid de bruker på de forskjellige oppdragene vil erfaringsmessig variere stort, og det kommer an på mange faktorer. Det er ikke meningen at alle elevene skal bli ferdige med alle disse oppdragene i løpet av to timer, men heller at de som trenger ekstra utfordringer kan få dette.

Som lærer vil en i en del tilfeller oppleve at en ikke har alle svarene, og det er heilt greit! Her skal en lære ved å prøve og feile, dele erfaringar med resten av klassen, og få det til i lag.

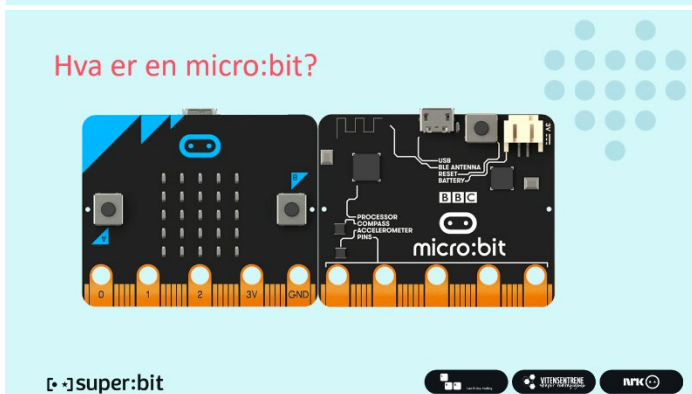
Det er lagt opp til at de to første oppdragene, oppvarming og introduksjon til micro:bit blir gjennomført med felles progresjon, og at etter at de er satt i gang med oppdrag 3 går de bare videre etter hvert som de blir ferdige.

Når det gjelder oppdraget med trafikklys, vil det være en fordel om en har bygget 2-3 trafikklys på forhånd, da dette kan ta litt tid. Gjerne gjør dette sammen med elevene. Det er ikke nødvendig å ha klar et helt klassesett med trafikklys, da ikke alle vil ha kommet så langt samtidig.

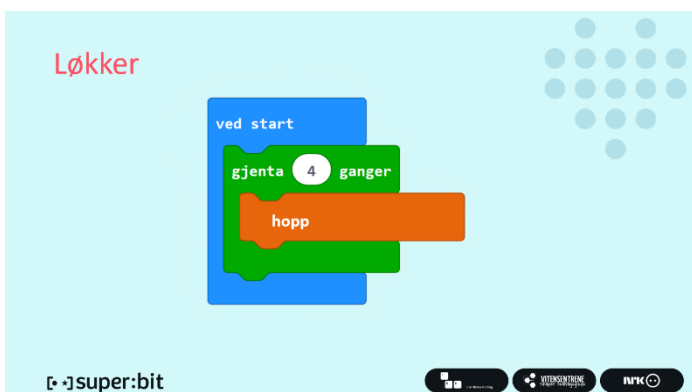
## 1 Oppvarming – analog programmering (10 minutter)

Her skal læreren programmere elevene! Målet her er at elevene skal få repetert eller få en introduksjon til hvordan blokkprogrammering kan se ut. De vil også få sett hva løkker, uendelege løkker, vilkår og variabler er. Gjerne snakk om dette underveis.

[sett inn link til presentasjon]



Her kan en ha framme dette bildet og snakke litt om hva en micro:bit er. En liten datamaskin som kan programmeres, har innebygde sensorer, og kan kobles til andre sensorer eller styre annet utstyr ved hjelp av tilkoblingene nederst.



Elevene venter til læreren sier «start», og så skal de gjøre det som står på skjermen. De skal altså hoppe 4 ganger. Dette er eksempel på ei løkke.

**Pause**

[•]super:bit

Her skal de hoppe 4 ganger, sette seg ned, vente 5 sekunder, og så klappe.

En micro:bit forholder seg helst til millisekunder, så her er en fin anledning til å snakke om hvor lenge 5000 millisekunder er.

**Gjenta for alltid**

[•]super:bit

I stedet for å utføre noe en gang når maskinen skrur på, kan vi be den om å gjenta noe for alltid, helt til den skrur av. Her vil elevene måtte reise seg, klappe to ganger, og sette seg igjen, for så å gjenta dette for alltid. Dette er et fint tidspunkt for læreren til å ta seg en tur ut og hente kaffe. Så kan en snakke om roboter, og at de ikke blir lei av å utføre en oppgave mange ganger.

**Løkke i løkke**

[•]super:bit

Denne sida inneholder en referanse til det lokale fotballaget, og teksten her kan med fordel tilpasses målgruppa. Får de til denne vil det høres ut som om man er på kamp og heier.

**Variabler**

[•]super:bit

Variabler er nyttige i programmering, og kan beskrives som noe vi kan gi navn, men som egentlig er tall. Elevene må her utføre et par regneoperasjoner, og så rope ut «Ja!» eller «Nei!» alt etter hva de kommer fram til.

## 2 Kom i gang med micro:bit (15 minutter)

Nå er det på tide å begynne med å programmere micro:biten. Del ut ark med introduksjonsoppgaven på. [Ha med link til pdf]

Få elevene til å åpne makecode, og sett dem i gang.

Her er det meningen at elevene skal leke seg litt på egen hånd med egne figurer, tekst og annet. De som er ivrige kan prøve seg på side 2, der de må bruke variabler.

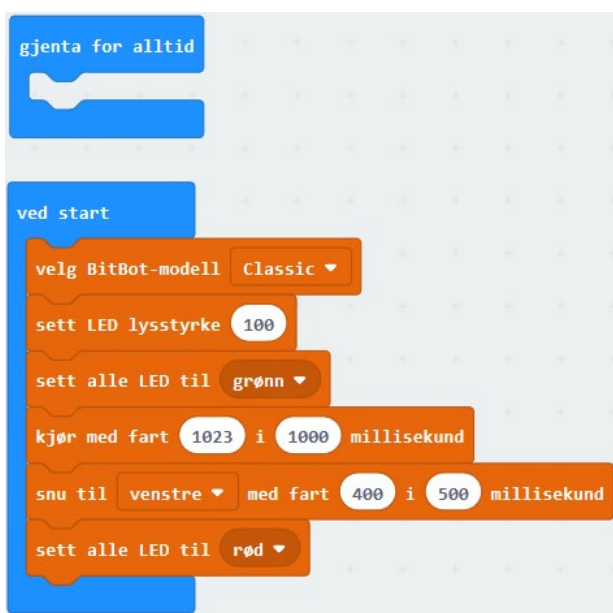
Hvis de ikke allerede har gjort det, er dette tiden for å lære elevene hvordan de overfører et program til micro:bit. Dette kommer an på hva slags plattform skolen bruker, så se egne veiledninger for dette [Link til veiledning for PC/iPad/Chromebook]

### 3 Kjør en meter med BitBot

Her skal elevene bli kjent med BitBot, og det første oppdraget er å få den til å kjøre en meter fram, snu, og så kjøre tilbake igjen.

For å enkelt komme i gang med kjøringen har vi laget ferdige programmer som gjør litt av det vi vil BitBoten skal gjøre, slik at elevene ser hvilke blokker som kan brukes for å kjøre fram, svinge osv.

Programmene de trenger ligger her [Ha med link til programmene]. Om elevene laster ned programmene på forhånd, eller om de skriver av, er opp til læreren.



Gjenta for alltid er tom, det vil si at den kun skal utføre programmet en gang.

Dette programmet vil få BitBoten til å kjøre framover litt, og så svinge litt til venstre. For å få den til å kjøre en meter må elevene justere på hvor lenge den kjører framover. Dette er lettere enn å justere på farten, og den største farten er uansett 1023. (Her regulerer man egentlig hvor stor spenning hver av motorene får, og 1023 betyr full gass)

[https://makecode.microbit.org/\\_RjoRmPK1RXHT](https://makecode.microbit.org/_RjoRmPK1RXHT)

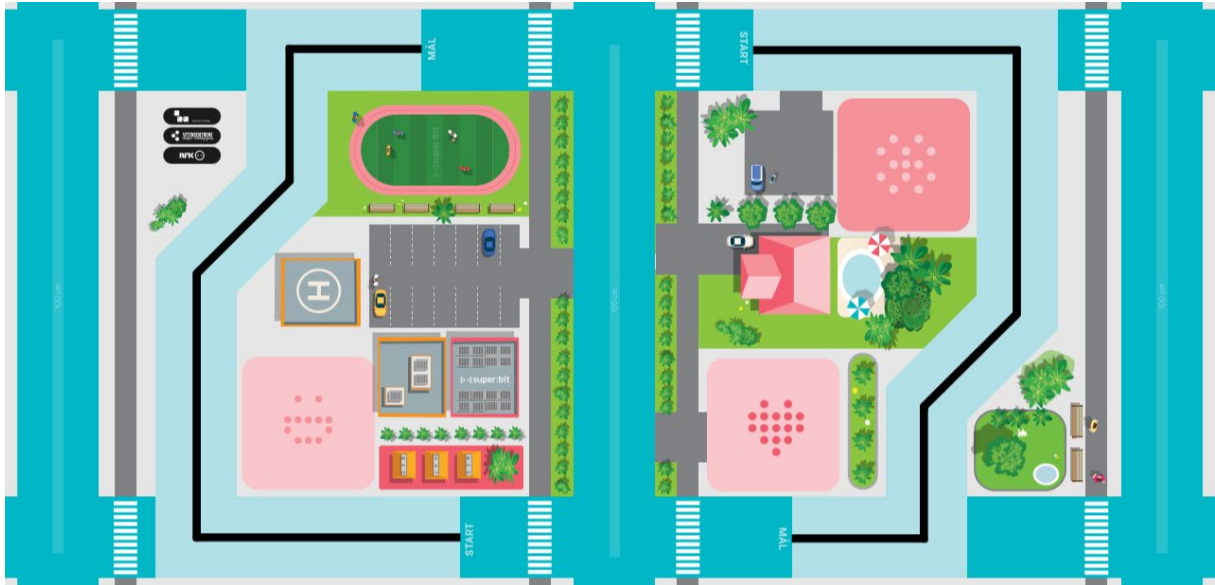
Her er det lurt å få elevene til å måle hvor langt den kjører når den kjører i ett sekund. Og så spør hvor langt de tror den kjører hvis den kjører i to sekunder. Det samme gjelder for rotasjonen. Hvor mange grader snur den når den snur i et halvt sekund?

Til slutt må de kopiere den blokken som får BitBoten til å kjøre framover, og legge den etter snu-blokken.

Når de har kontroll på hvordan de kan få den til å kjøre en meter, og å snu 180 grader kan elevene skrive ned parameterene de har brukt. Det vil være nyttig i neste oppdrag, som er å kjøre ei løype med BitBot.

## 4 Kjør ei løype med BitBot

Her skal elevene få BitBoten til å kjøre ei av løypene som er markert på smartby-matta:



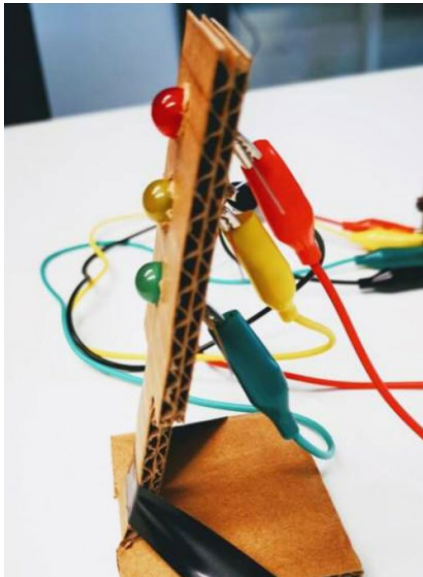
Her må de måle hvor lange de enkelte strekningene er, og anslå vinklene. Her kommer det godt med å vite hva som skal til for å få den til å kjøre en meter! Spør hvor langt de tror den kjører hvis den kjører halve tida, og det samme med svingene.

Har dere ikke matte tilgjengelig, går det også helt fint å teipe opp en liten bane på gulvet.

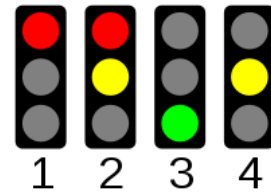


BitBoten har også linjesensor på undersiden, og i etterarbeidet vil elevene få mulighet til å få den til å følge linja.

## 5 Programmering av trafikklys



Her er det en fordel å ha satt sammen 2-3 trafikklys på forhånd, eller ha liggende instruksjoner så elevene kan sette dem sammen selv. Det tar litt tid å sette dem sammen. Se egen veiledning for dette her [link til trafikklysbygging]



Når trafikklyset er satt sammen og koblet til micro:biten med rødt lys til pin 0, gult til pin 1 og grønt til pin 2, er det klart til å programmeres.

```

gjenta for alltid
  skriv digital til P0 ▼ verdi 1
  pause (ms) 500 ▼
  skriv digital til P0 ▼ verdi 0
  pause (ms) 500 ▼

```

Elevene laster inn eller skriver av dette halvferdige programmet for å komme i gang og se hvilke blokker de trenger. Skriv digital-blokkene fins under Avansert -> Tilkobling.

Programmet vil gjøre at trafikklyset står og blinker rødt. Det er jo ikke sånn det skal være, så oppdraget er å fikse det, sånn at det viser rett sekvens.

[https://makecode.microbit.org/\\_Fwq0mfJXuDwb](https://makecode.microbit.org/_Fwq0mfJXuDwb)

Her er et løsningsforslag, men vent med å gi det til elevene før de har fått prøve selv.



```
gjenta for alltid
  skriv digital til P0 verdi 1
  skriv digital til P1 verdi 0
  skriv digital til P2 verdi 0
  pause (ms) 2000
  skriv digital til P0 verdi 1
  skriv digital til P1 verdi 1
  skriv digital til P2 verdi 0
  pause (ms) 1000
  skriv digital til P0 verdi 0
  skriv digital til P1 verdi 0
  skriv digital til P2 verdi 1
  pause (ms) 2000
  skriv digital til P0 verdi 0
  skriv digital til P1 verdi 1
  skriv digital til P2 verdi 0
  pause (ms) 1000
```

[https://makecode.microbit.org/\\_UDFAaA9sT2dq](https://makecode.microbit.org/_UDFAaA9sT2dq)

## 6 Robotplenklipper

Her tar en i bruk ultralydsensoren til BitBot. Målet er å få BitBoten til å fungere som en robotplenklipper eller robotstøvsuger. Den skal stoppe når den møter en hindring, for så å snu og kjøre en annen vei.

Elevene laster inn eller skriver av dette halvferdige programmet:



[https://makecode.microbit.org/\\_dpa3Hf8eH66r](https://makecode.microbit.org/_dpa3Hf8eH66r)

Her møter vi vilkår igjen. Den leser hele tiden avstanden til nærmeste hindring, og hvis denne er større enn 15 cm skal BitBoten kjøre framover. Blir derimot avstanden mindre enn 15 cm, vil den stoppe.

Oppdraget her er å fikse på programmet så den snur i stedet for å stoppe.

Her kan en i tillegg legge inn at den skal snu tilfeldig høyre eller venstre, med tilfeldig antall grader, bremse ned forsiktig før en hindring, eller lignende variasjoner. Hvordan den skal fungere kan elevene være med på å bestemme selv.