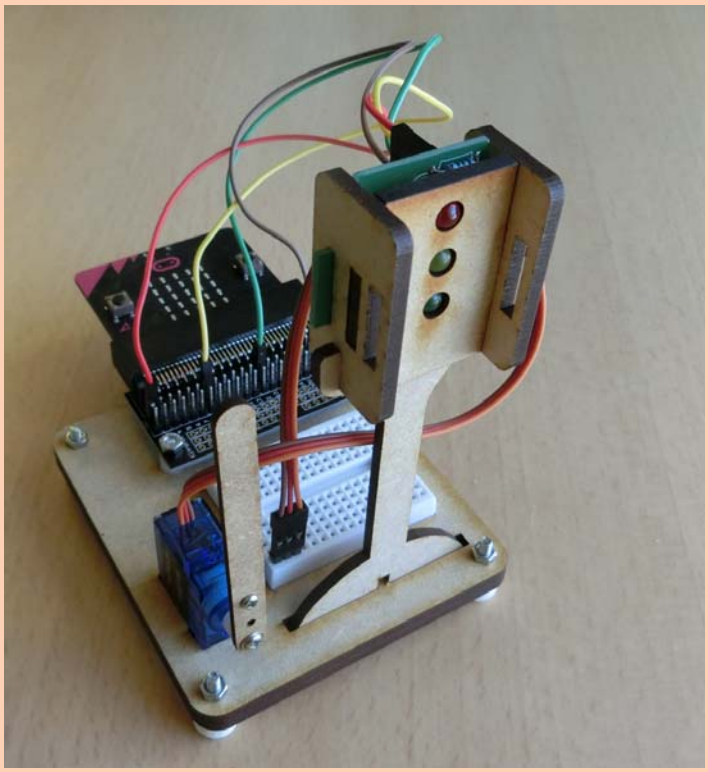
VITENSENTERET
TRONDHEIM

Rev 1.9

Nils Kr. Rossing, Ola Kleiven, Anne-Birgitte Belboe, Rannvei Sæther, Eva H. Hagen

Grunnkurs Micro:bit – DeKom



Oktober 2020

Denne siden er blank

Grunnkurs Micro:bit
DeKom

Nils Kr. Rossing, Ola Kleiven, Anne-Birgitte Belboe,
Rannvei Sæther, Eva H. Hagen

Grunnkurs Micro:bit – DeKom

Trondheim 2020

ISBN 978-82-92088-66-1

Bidragsytere:

Layout og redigering: *Nils Kr. Rossing, Vitensenteret i Trondheim*

Tekst og bilder: *Nils Kr. Rossing, Vitensenteret i Trondheim*
Anne Birgitte Belboe, Vitensenteret i Trondheim
Rannvei Sæther, Vitensenteret i Trondheim
Ola Kleiven, Vitensenteret i Trondheim

Korrekturlesning: *Eva H. Hagen, Vitensenteret i Trondheim*

Faglige spørsmål rettes til:
Vitensenteret i Trondheim
v/Nils Kr. Rossing
nkr@vitensenteret.com

Kongensgate 1
7011 Trondheim

Postboks 117
7400 Trondheim

Vitensenteret i Trondheim
Telefon: 72 90 90 07
http://www.vitensenteret.com/

Rev 1.9 – 01.10.20

Forord

Heftet er skrevet som en hjelp til gjennomføring av 3. samling av DeKom-tilbudet: *Skapende aktivitet i klasserommet*, som ble gitt til Okstad skole, Tomasskolen og Vikhammer/Vikhammeråsen skole høsten 2020.

Målsetingen med denne første og andre samlingen er å gi deltakerne en grunnleggende innføring i programmering med micro:bit samtidig som det skal være et eksempel på hvordan en kan bruke programmering for å bygge opp et produkt (prosjekt). Det er lagt vekt på sentrale begreper fra læreplanen og å vise hvordan man kan lage og teste små programbiter som etter hvert kan settes sammen til et større program.

Autentisitet har også vært viktig. Ved å bruke et trafikklys brukes noe som alle kjenner samtidig som svært få har reflektert over teknologien som ligger bak. Det er også lagt vekt på å tilrettelegge for tverrfaglighet ved å utfordre lærere og elever til å tenke trafikksikkerhet.

Det er foreløpig noe uklart hvordan man ved dette undervisningsopplegget kan fokusere på bærekraftmålene. De to mest nærliggende er bærekraftsmål 9: *Innovasjon og infrastruktur* og bærekraftsmål 11: *Bærekraftige byer og samfunn*. Deltakerne oppfordres til å se andre koblinger til bærekraftsmålene.

Heftet er ment som en støtte under arbeidet på kursdagen, men også som en hjelp i det etterfølgende arbeidet i klasserommet, dog ikke uten videre for utdeling til elevene.

Tilbudet er initiert av Trondheim og Malvik kommune og finansiert av DeKom (Desentralisert Kompetansehevning) midler fra Udir.

Vitensenteret i Trondheim
Oktober 2020

Nils Kr. Rossing
Ola Kleiven
Rannvei Sæther
Anne Birgitte Belboe
Eva H. Hagen

Innhold

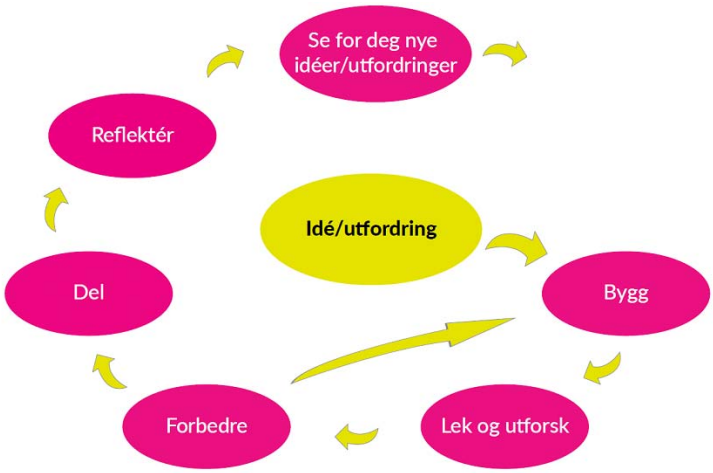
1 Innledning

1.1 Kompetansepakken: Skapende aktivitet i klasserommet

På Vitensenteret ønsker vi å etablere en kultur for helhetlig læring gjennom skapende aktiviteter. Kompetansepakken bygger på vår erfaring og grunnleggende tankegang om å anvende teori som et redskap for å realisere produktideer, kombinert med interessevekkende oppgaver og kreative prosesser som vist på figuren under.

Vårt fokus er blant annet å øke kunnskapen om tradisjonelle og digitale teknikker for å gi lærere og elever en mer rikholdig "verktøykasse" for å bli i stand til å realisere kompetansemålene i Fagfornyelsen 2020 i en "digitalisert verden".

Vi ser det som helt avgjørende at størstedelen av kompetansehevingen hos lærerne skjer i klasserommet sammen med elevene under veiledning. Vi ønsker derfor at lærere og elever skal arbeide med egne prosjekter som støtter læringsprosessen til alle involverte.



Målsetninger for kompetansepakken

Det primære målet med kompetansepakken er å stimulere læreres og elevers skapende evner gjennom å legge til rette for utforskende og engasjerende arbeidsmåter ved å ta i bruk gammel og ny teknologi i den skapende prosessen.

Beskrivelse av læringsprosesser i skolenes arbeid med kompetansepakkene

Kompetansepakken Skapende virksomhet i klasserommet skal fokusere på praktisk aktivitet som gir rom for lærernes og elevenes skapende og utforskende evner og søker å stimulere til nysgjerrighet og utvikle deres kreativitet, utholdenhet og evne til å løse problemer. De skal utvikle og realisere egne produktideer ved bruk av teknologiske løsninger basert bl.a. på enkel elektronikk og egenutviklet programvare, kombinert med bruk av tøy, tre- og plastmaterialer formgitt ved hjelp av digitale verktøy som f.eks. tegne- og modelleringsverktøy, 3D-printing og laserkutting.

Det vil derfor bli lagt vekt på at deltagerne sammen med elevene sine definerer prosjekter, små eller store, som skal ende opp i produkter som både lærere og elever har et eierforhold til. Dette skal også sikre at elevene skal oppleve at de arbeider med prosjekter som er relevante for deres livsverden.

Gjennom designprosessen skal de få et bevisst forhold til det produktet de har valgt å utvikle og de materialene og komponentene de velger å bruke, og på den måte få et etisk og miljøbevisst forhold til det produktet de utvikler. Dette kan også omfatte gjenbruk av materialer og bruk av digitalt verktøy for reparasjon eller utbedring av eksisterende gjenstander. På denne måten ønsker vi at elevene utvikler en kritisk holdning til eget forbruk og egne valg.

Det er også et mål at det bringes inn kunnskap fra ulike fagfelter der det er naturlig som en hjelp til å løse de utfordringer designprosessen skaper. Det kan være måletekniske, elektriske, magnetiske eller mekaniske utfordringer hentet fra fysikken. Det kan være regnetekniske, geometriske, romlige eller logisk-analytisk resonnering hvor man anvender prinsipper fra matematikken. Men det kan også være håndverksmessige teknikker, materialkunnskap eller blikk for det estetiske hentet fra Kunst og håndverksfaget. Under veilednings- og opplæringsprosessen prioriteres tilføring av ny kunnskap etter prinsippet "Just in time" framfor "Just in case".

Med bakgrunn i disse mer generelle prinsippene hentet fra Fagfornyelsen 2020, vil det legges vekt på å anvende programvare og utstyr som skolene har tilgang til, eller kan få tilgang til via nett og ved bruk av Skaperverkstedet med tilhørende utstyr ved Vitensenteret i Trondheim.

1.2 Programmering i klasserommet

Det er mange måter å lære seg bruk og programmering av micro:bit. Siden DeKom-pakken: *Skapende aktivitet i klasserommet* legger opp til at elever og lærere skal arbeide prosjektbasert, så vil vi også gjennom denne opplæringen vise hvordan små enkle øvelser til sammen kan bli et større prosjekt eller et produkt. Det finnes en rekke slike opplegg, både små og store, innen Skaperskolen og på nettet ellers. Under denne kursdagen har vi valgt å ha som mål å utvikle et trafikklys knyttet til det fiktive prosjektet "Byplanlegging". Det er flere årsaker til at vi har valgt akkurat dette prosjektet som eksempel:

Kjent: Prosjektet er valgt ikke først og fremst for at det er så spennende, men at trafikklyset er noe alle har et forhold til, og noe som de fleste av oss kommer i kontakt med til daglig og kan ha meninger om. Dessuten er det, for de fleste en skjult kompleksitet i et trafikklys som nettopp handler om programmering.

Kreativitet: Dette er et godt utgangspunkt for en kreativ prosess hvor man kan begynne med det kjent og videreutvikle dette i uventede retninger, gjerne på bakgrunn av egne erfaringer og behov.

Relevans: Dessuten er det et eksempel på et prosjekt som kan være utgangspunkt for nyttige diskusjoner i klasserommet om trafikksikkerhet og hvordan man skal komme seg trykt til skolen[1]. Som en del av prosjektet kan

man også sende barn ut med passende oppdrag ut til en nærliggende fotgjengerovergang eller et veikryss med lys. Det er heller ikke vanskelig å knytte matematikk (statistikk) til slike observasjoner.

Realisering: Trafikklys-prosjektet har også en annen interessant side ved at elevene kan utforme og lage ulike modeller av trafikklys som kan realiseres på forskjellige måter, f.eks. ved hjelp av laserkutting eller 3D-printing.

Rikt: Prosjektet kan dessuten lett utvides til å omfatte et større lyskryss som involverer biler og fotgjengere. Det har derfor den nyttige egenskapen at man kan begynne svært enkelt og ende opp med noe særdeles komplekst. Dette gjør det også enklere å differensiere undervisningen.

Det finnes selvfølgelig mange andre prosjekter man kan velge. Her vil vi vise til forslag presentert under www.skaperskolen.no.

1.3 Hvorfor jobbe prosjektbasert?

Skapende aktivitet i klasserommet handler i denne sammenhengen å arbeide målrettet mot å realisere et produkt. Produktet kan være en nyttegjenstand, et praktisk hjelpemiddel, en kunstinstallasjon eller noe annet som kan stilles ut og vises fram. Det kan være et produkt klassen har samarbeidet om å lage, men det kan også være en rekke små produkter, eller kanskje aller helst en rekke små produkter som viser prosessen fram mot et samlende produkt. Her står lærere og elever ganske fritt, dog med noen få rammebetingelser:

- *Valg av prosjektene gjøres i samarbeid med elevgruppene*
  Vi anser det som særdeles viktig at elevene får være med i prosessen med å velge prosjekt etter prosjekter, da vi tror at dette vil være med å gi elevene et eierforhold til prosjektet.
- *Det skal utvikles ett eller flere produkter hvor det skal legges vekt på form og funksjon.*
  Om det er praktisk mulig så vil det være gunstig at grupper av elevene velger ulike prosjekter, slik at flest mulig får anledning til å utvikle sine spesielle interesser og talenter, noe som kan være svært krevende for den som skal veilede.
- B*ruk digitale teknikker/hjelpemidler deriblant programmering.*
  Ett eller flere av prosjektene skal ta i bruk digitale verktøy, som f.eks. digitale tegneprogrammer, 3D-printere, laserkuttere, vinylkuttere o.l. Like så skal elevene på en eller annen måte bruke programmering og mikrokontrollere, eksempelvis micro:bit, i ett eller flere av prosjektene.
- *Integrert del av undervisningen*
  Prosjektene skal inngå som en integrert del av undervisningen den tiden prosjektene gjennomføres og være forankret i fagfornyelsen, L20. Det er vårt håp at lærerne skal oppleve at arbeidet med prosjektene skal være en hjelp i planleggingen av den ordinære undervisningen og ikke en ekstra byrde som legges på dem.

1. Bildet er hentet fra: https://www.hopscotch.in/blog/10-road-safety-rules-you-should-teach-your-children/

- *Samarbeid*
  Det oppmuntres til samarbeid mellom deltagerne på samme skole, ev. andre skoler som arbeider med samme kompetansepakke
- *Arbeidet med prosjektene skal primært skje mellom samlingene.*
  Samlingene vil bli brukt til opplæring i teknikker og bruk av digitale og andre verktøy for bruk i klasserommet. Tiden under samlingene er svært begrenset og vil være utenfor klasserommet, det er derfor viktig at deltagerne bruker tiden mellom samlingene og i klasserommet til å prøve ut og praktisere det de har lært. I den grad vi har ressurser til det vil vi gjerne bidra med veiledning og støtte også med det som skjer i klasserommet
- *Tverrfaglighet*
  Prosjektene skal i størst mulig grad være tverrfaglige ved at man kombinerer praktisk estetiske fag, naturfag og matematikk, ev. andre fag. Det bør derfor være en målsetning at lærere med forskjellig faglig bakgrunn arbeider sammen.
- *Inkluder bærekraftmål*
  Under arbeidet med prosjektene skal man ha bærekraftmålene i tankene og kunne peke på hvordan ett eller flere av bærekraftmålene berører prosjektene.
- *Oppsummerende utstilling*
  Ha som mål at produktene som utvikles i løpet av prosjektperioden skal kunne inngå i en utstilling som oppsummerer arbeide i klasserommet. En utstilling som vel så gjerne kan vise prosessen som det endelige produktet.

Lærerens rolle

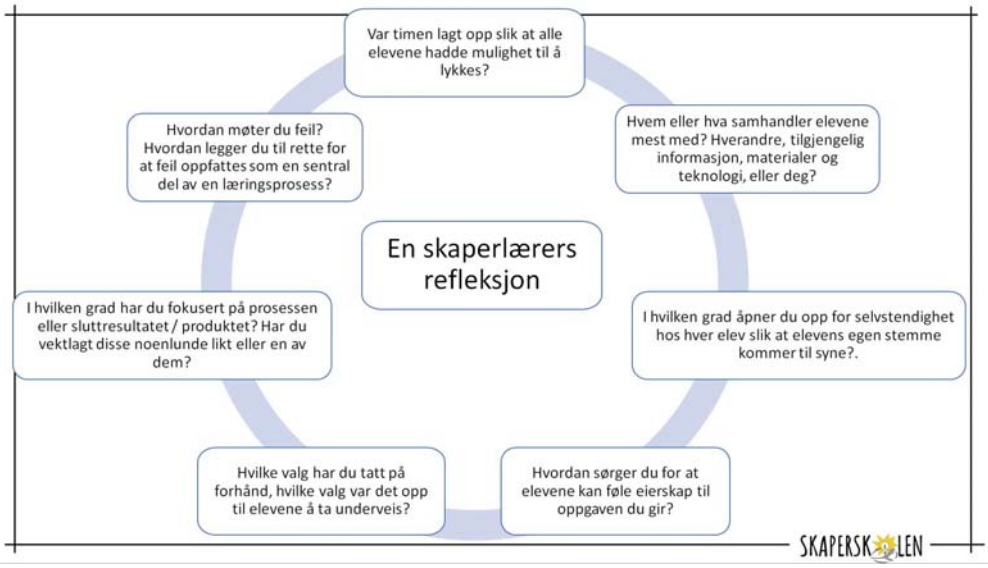
Vi har tro på at prosjekter som har en veldefinert målsetning som klassen kan samles om, vil være motiverende og gi arbeidet en retning. Det er viktig at elever og lærere har *eierskap til produktet*, hver arbeidsgruppe sitt produkt eller delprodukt. Vi har også tro på at ny kunnskap tilføres i størst mulig grad etter behov (”Just in Time”), ikke løsrevet fra de oppgavene som til en hver tid skal løses (”Just in case”).

Det er lett å sette seg for ambisiøse mål, barn er flinke til å skape visjoner langt ut over det som med rimelighet kan la seg realisere. I slike sammenhenger kreves en klok lærer som ikke tar motet fra elevene, men heller ikke gir urealistiske forhåpninger. I slike tilfeller kan det være lurt å sette seg *realistiske delmål som i seg selv oppleves som små seire på veien til de store målene.* Gjennom løsningen av delmålene vil også elevene etter hvert få et mer realistisk forhold til det endelige målet og være istand til å justere ambisjonene.

Lærerens rolle i denne sammenhengen vil være svært krevende og sammensatt og det er mange hensyn å ta og mye å holde rede på som antydet i figuren under. Her er det mange ting en kan reflektere over som lærer:

- I hvilken grad får hver enkelt elev anledning til å utvikle sine spesielle evner?
- Får alle elevene anledning til å føle mestring og erfare å lykkes?
- Stimulerer oppgavene til samhandling mellom elever eller arbeider de selvstendig og på egen hånd?

- Opplever elevene eierskap til det de arbeider med?
- Hvordan håndteres feil, blir de sett på som en viktig ressurs i læringsprosessen?



Fra presentasjon holdt av Liv Oddrun Voll Naturfagsenteret

Noen tips til prosjektarbeid

- *Sett opp en målsetning for prosjektet*
  Hva er målet med prosjektet, hva skal sluttproduktet være? Gjør elevene oppmerksom på at målet kan endre seg under veis. Det er også nødvendig å få en oversikt over rammevilkårene. Hvor mye tid og penger kan man bruke, hva har man av verktøy og materialer o.l., og hva må man skaffe til veie. Husk at leveranse av bestilte deler kan ta lang tid.
- *Idedugnad*
  Når rammene og målsetningen er fastlagt, kan man lage kreative prosesser hvor mange ideer får lov til å komme fram. Her kan boka *Nyskapning* av *Erik Lerdahl* være til nytte. Den kreative prosessen avsluttes gjerne med en mer kritisk gjennomgang hvor det tas noen beslutninger. Slike prosesser kan ta noe tid.
- *Lag realistiske delmål.*
  Hva trenger man å finne ut, teste eller prøve ut for å løse oppgaven? Kan man realisere deler av produktet? Dersom målet er å lage et avansert trafikklys, så kan det være lurt å finne ut hvordan man slår av og på lysene, hvilken rekkefølge lysene skal skifte, hvordan man kan skifte til grønt ved å trykke på en knapp osv. Hver av disse delmålene kan testes ut enkeltvis.
- *Lag en tidsplan*
  Lag en plan for når de ulike delmålene bør være uttestet og ferdige. Fordel oppgaven mellom ulike arbeidsgrupper og sett opp en rekkefølge. Juster planen etter behov.

- *Identifiser utfordringer*
  Om mulig bør man forsøke å finne ut hvilke delmål som er spesielt krevende eller krever kunnskaper det tar tid å tilegne seg. Det kan være lurt å finne ut så fort som mulig, om disse lar seg løse med de ressursene man har. Viser det seg at et slikt delmål ikke lar seg løse, så må man kanskje endre hele prosjektet. Det er leit om dette er noe man oppdager helt mot slutten av prosjektet.
- *Lag skisser og del tanker*
  Det er ofte lurt å lage skisser eller tegninger som viser en ide eller en utforming av et produkt. Slike skisser er også nyttige for å dele tanker og ideer med hverandre. Å dele en ide med andre gjør at man får nye ideer. Det er også nyttig å fortelle andre om utfordringer man har, bare det å fortelle om utfordringen gjør ofte at man ser løsningen.
- *Dra nytte av hverandre - Tverrfaglig kompetanse er ofte nødvendig i prosjektarbeid*
  Det er ofte krevende å arbeide i et prosjektteam, da man ofte har forskjellige oppfatninger av tingene. På den annen siden vil team-medlemmer som har ulike kompetanser og interesser kunne utfylle og støtte hverandre. En skal imidlertid ikke undervurdere utfordringen med god kommunikasjon.
- *Lag prototyper*
  Dersom man skal lage noe som andre skal ha nytte av, så er det viktig å teste om produktet fungerer etter hensikten. Da kan man lage enkle utgaver av produktet som brukerne får teste for så å si hva de mener. Dette er som oftest svært nyttig og man blir ofte overrasket hvor forskjellig man tenker. Prototyper kan være svært enkle, som f.eks. pappmodeller som styres av elevene, og helt fram til noe som er funksjonelt og er nær et ferdig produkt.
- *Dokumenter! – Tenk utstilling*
  Det er viktig å skrive ned, filme eller ta bilder av det som gjøres slik at man tar vare på det man finner ut slik at det kommer til nytte og ikke glemmes under veis. Slik dokumentasjon er også viktig når man i etterkant skal beskrive prosessen eller dersom noen andre skal gjenta prosjektet. Noen skriver en rapport som beskriver prosess og produkt, vi ønsker å lage en utstilling som viser både produktene og prosessen fra ide til produkt med bilder tekst og gjenstander. Vi håper også at utstillingen kan inneholde demonstrasjoner.

1.4 Skaperskolen

Bruk gjerne oppleggene som er presentert under "Skaperskolens" hjemmeside: www.skaperskolen.no. Her ligger det flere gjennomarbeidede opplegg for bruk i klasserommet hvor micro:bit er sentral, og som gjerne setter micro:bit og programmering inn i en sammenheng.

Vi klipper fra Skaperskolens hjemmeside:

> *Fra høsten 2020 skal de nye læreplanene tas i bruk i skolen. Da kommer programmering og teknologi på timeplanen i mange fag, alt fra matematikk og naturfag til kunst og håndverk og musikk. Samtidig går debatten høyt om bruk av læringsteknologi i skolen, og mange lærere og elever lurer på hvordan dette skal gjøres.*
>
> *Skaperskolen tar mål av seg til å vise hvordan programmering og teknologi kan brukes kreativt og tverrfaglig.*
>
> *Gjennom kurs og konferanser får lærere opplæring i hvordan dette kan skje. Skaperskolen utvikler også egne pedagogiske opplegg som kan tas i bruk i skolen. Gjennom skaperfestivaler møter elevene kreativitet og skaperglede i alle fasonger. Her på skaperskolen.no får du som lærer tips og råd til hvordan du kan bygge opp et skaperverksted på din skole.*

Videre leser vi:

> *Skaperskolen finnes i hele landet og er et samarbeid mellom de regionale vitensentrene og Naturfagsenteret. Skaperskolen – tar skolen inn i det 21. århundre. Prosjektet er finansiert av Sparebankstiftelsen DNB.*

Her finner man alt fra enkle "Kom-i-gang" prosjekter til prosjekter som i større grad enn trafikklys-prosjektet kaller på fantasi og kreativitet f.eks. knyttet til fag som kunst og håndverk og musikk. Spesielt vil vi anbefale: Musikkverksted for 5. – 7. trinn[2], som er et prosjekt som absolutt kaller på den frie fantasi og kreativitet satt inn i en spektakulær ramme.

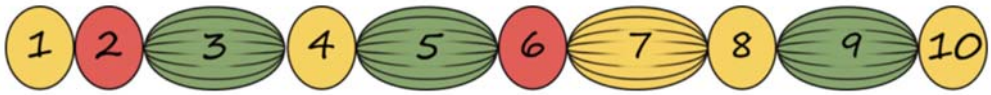
1.4.1 Bruk av Boblediagram

Når man skal i gang med å planlegge for kreative og skapende prosjekter i skolen, er det greit å ha noen rammer å jobbe i. Enkelt forklart er boblediagrammet, som er utviklet av Skaperskolen, et nyttig planleggingsverktøy for læreren. Det gir bl.a. læreren oversikt over rekkefølgen i pro-

2. Musikkverksted, 5. - 7. trinn: https://skaperskolen.no/oppfinnerverksted/

sjektet, altså hvilke faser prosjektet kan bestå av, hvilke steder i prosjektet det er nødvendig å «samle» elevene og i hvilken grad aktivitetene er tilrettelagt for elevenes kreative og skapende evner, altså i hvor stor grad det er lagt opp til elevfrihet i de ulike fasene.

Figuren under viser et eksempel på et prosjekt med 10 faser. De små boblene viser faser hvor lærer har kontrollen og har hele gruppen samlet. Det er for eksempel i forbindelse med innledning av prosjektet der hvor lærer har behov for å gi felles informasjon eller instrukser, eller felles gjennomgang av utstyr/teknikker som trengs på veien, eller spesifisering av kriterier man skal jobbe under osv. De store boblene symboliserer faser hvor elevene jobber mere selvstendig i gruppene og hvor lærer tilrettelegger og har mere en veiledende rolle. Disse aktivitetene drives av elevene selv. Fargene symboliserer i hvor stor grad elevene jobber kreativt og har stor grad av frihet (dvs. aktiviteten preges av å være mer skapende der elevene har mange valgmuligheter) i motsetning til aktiviteter som f.eks. å følge en oppskrift eller en instruks der de selv ikke tar mange valg (dvs. mer en gjøre-aktiviteter).



Rødt symboliser lite grad av elevfrihet, gul noen grad av elevfrihet og grønn er der hvor elevfriheten er veldig stor (selvsagt innenfor skolens rammer).

Gode kreative og skapende prosjekter bærer ofte preg av å ha flere bobler, som veksler mellom å være små og store, og hvor en del av boblene er grønne eller i det minste gul.

1.4.2 Bruk av idemyldrings- og virkemiddelkort

Ordet kreativitet kommer fra latin «creare» som betyr «å lage» eller «å skape». Ordet brukes ofte om personer, det vil si skapende mennesker der kreativitet er en fremtredende egenskap ved personligheten. I de fleste definisjoner av ordet forekommer stikkord som originalitet, fantasi, problemløsning, oppfinnsomhet, gode ideer, se muligheter osv. Kort fortalt kan man si at kreativitet er å sette sammen ting på nye måter og at kreativitet er en "muskel" som må trenes, akkurat som andre muskler i kroppen. Øver man seg ikke, og kun bruker den i ny og ne, blir den kreative "muskelen" slapp og vi synes det er vanskelig å være kreativ. Det er nemlig ikke slik at å være en kreativ person er en medfødt nådegave for de få. Alle kan være kreative, også voksne, selv om barn kanskje kommer lettere i situasjoner hvor denne "muskelen" trenes.

For å øve opp den kreative "muskelen" må man begynne i det små. Da trenger man gjerne en del hjelpemidler. Forestill deg at du får oppgaven; Lage en ny oppfinnelse som skal revolusjonere transportsystemet vårt. Litt vanskelig å gå løs på den oppgaven sånn med det samme? Noen ganger trenger hjernen vår å bli tvunget inn i et hjørne for å klare å se nye sammenhenger, og det kan hjelpe å ha noen rammer rundt seg. Er rammene for løse «lage oppfinnelsen akkurat som du vil», så blir kreativiteten vår ofte begrenset.

I Skaperskolen bruker man ulike *prosesskort* for å «presse» hjernen inn i det kreative hjørnet. Dette har vist seg å fungere svært godt, samtidig som et snev av humor bringes inn i skaperprosessen.

Idègenererende spørsmålskort har som mål nettopp å stille spørsmål slik at mange ideer dukker opp. I aller enkleste form kan det være en bunke kort med ulike substantiv og en bunke kort med ulike verb. Elevene trekker et substantivkort og et verbkort og så skal man skal sette sammen ordenen på en ny måte. F.eks. Hoppe og stol blir en «Hoppestol». Videre kan man finne ut hvordan den fungerer, hvem den brukes av og hvilket problem den løser osv. Da kommer fantasien og kreativiteten i gang. Videre kan man sette oppgaven inn i den konteksten man ønsker f.eks. Transport.

Man kan også ha mange ulike spørsmålskort[3] som går litt mere i dybden, med konkrete spørsmål knyttet til:

- funksjonen til oppfinnelsen
- brukeren og brukerens behov
- problemer som kan dukke opp

Disse kortene kan elevene gruppevis bruke til å stille hverandre spørsmål for å utdype ideene sine. De må da muntlig forklare nærmere og svare på spørsmålene så godt det lar seg gjøre. Mange ganger er det ting man ikke har tenkt over, som de da må ta stilling til. Dette er med på å utvikle ideene videre. Ofte er det lurt å lage seg en skisse av oppfinnelsen, særlig dersom man til slutt også skal lage en prototyp av den.

Virkemiddelkort er også veldig nyttig for å utvikle ideer videre, og da særlig dersom man f.eks. er i en fase hvor man lager skisserer av oppfinnelsen sin. Disse tvinger nemlig fram en endring på skissen ved at hver deltager på gruppen skal lage en ny skisse der de tar hensyn til en konkret endringsoppgave knyttet til antall, størrelse, form og plassering.

F.eks. slik dette kortet sier; man skal øke antall med 2, beholde størrelsen, endre formen, snu om på noe.

Etter at hver enkel i gruppa har tegnet sin nye individuelle skisse med disse endringene, sammenligner gruppen de ulike skissene og diskuterer hva som skjer med funksjonen til oppfinnelsen med disse endringene. Blir det bedre eller dårligere? Skaper det kanskje noen nye ideer for gruppa som de vil gå for til slutt? Blir det tydeligere for alle hva som ikke fungerer?

Det finnes mange ulike prosess- og virkemiddelkort for arbeidet med prosjektet vårt og som lar seg endre på og tilpasses til deres ulike prosjekt.

1.5 Teori og praksis

Det har alltid vært en utfordring å kombinere teori og praksis på en god måte i skolen. Mange mener at skolen er blitt for teoretisk og for lite praktisk noe som gjør at mange faller utenfor. Det er gjort mange forsøk på å bøte på dette gjennom flere år. Vi kan nevne satsingen på det tverrfag-

3. Eksempler på slike kort finnes på skaperskolen.no

lige emnet *Teknologi og design*[4] og *valgfagene* som ble gjeninnført i grunnskolen for noen år tilbake.

> Uttalt av ungdomsskoleelev:
>
> *Teori er når man forstår alt, men ingen ting virker*
> *Praksis er nå alt virker, men man forstår ikke hvorfor*
> *Teori og praksis i kombinasjon er når ingen ting virker og man skjønner ikke hvorfor*
>
> Fritt gjengitt etter Runar Baune
> Naturfagsenteret

Det gjennomføres mange gode "gjøre"-aktiviteter i skolen i et forsøk på å bøte på teoretiseringen, men ikke alle aktiviteter beveger seg over i en skapende fase.

Gjennom pakken: "Skapende aktivitet i klasserommet" ønsker vi å fokusere på den vanskelige prosessen fra *gjørende aktiviteter* til *skapende aktiviteter*. Det er ikke tvil om at det trengs mange gjøre-aktiviteter, spesielt i starten, for å lære ulike teknikker og tilegne seg kunnskaper som er redskaper som den kreative prosessen trenger for å komme igang. Da er det ofte heller *ikke nok å vite hvor* kunnskapen finnes, man må ha en god del *på plass i hodet.* For at det skal bli slik *må man trene* ved å utføre gjørende aktiviteter (se avsnitt 1.5.1).

Gjennom styrte kreative prosesser ønsker vi å komme et skritt videre og tenke kreative muligheter og legge grunnlaget for å finne nye løsninger og lage produkter som ikke finnes fra før. Hos Skaperskolen finnes hjelpemateriell for å gjennomføre slike prosesser, det samme finner vi i boka *Nyskapning* av ***Erik Lerdahl***[5].

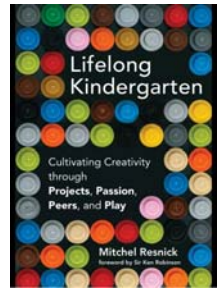
Leken er nettopp den arenaen hvor ingen prosjekter mislykkes siden all erfaring er nyttig erfaring. ***Sonja Kibsgård*** skriver i boka *Mens leken er god*[6]:

> *Lek blir etter min mening ikke noe som avtar med alder og år, men en måte å forske, arbeide og leve på som skaper engasjement, glød, motivasjon og hengivelse; en drivkraft til livet som gir fiksjon forrang for fornuften!*

Lek er en livsstil, en holdning, snarere enn noe vi gjør en gang i blant på idrettsplassen eller når vi er sammen med barn.

4. Anbefalt litteratur: Dahlin, Svorkmo, Voll, "Teknologi og design i skolen", Cappelen Damm Akademisk 2013
5. Lerdahl E., *Nyskapning – Arbeidsbok i kreative metoder*, Gyldendal Akademisk 2017
6. Kibsgaard & Wosstryck (red.), *Mens leken er god*, Tano forlag 1999

Dette er også grunnen til at vi har valgt boka: *Life long kindergarden – Cultivating creativity through Projects, Passion, Peers, and Play* av ***Mitchel Resnick*** som støttelitteratur for denne kompetansepakken.



1.5.1 Om "å gjøre" og om "å skape"

Et skaperverksted er ikke nødvendigvis et eget rom, slik Trigger skaperverksted på Vitensenteret i Trondheim er. Det er nok med et bord med masse materialer som gjør at elever blir nysgjerrige og inspirert til skapende aktivitet i et vanlig klasserom eller en krakk med en gjennomsiktig boks plasser ved siden av tavla – så er du i gang! I boksen legger du det du finner i roteskuffa, og så lar du det utvikle seg derfra. Det er lurt å samle på ting gjennom året.

Når vi driver med skaperaktiviteter er det viktig å vite forskjellen på *skapende* og *gjørende* aktiviteter. Det er viktig at vi selv er bevisst på det, og at vi også bevisstgjør elevene.

Gjørende aktiviteter er aktiviteter der vi følger en arbeidstegning eller en oppskrift gitt av læreren.
Skapende aktivitet er der eleven selv designer og lager sine egne arbeidstegninger eller oppskrifter, og bruker sin kreativitet i hele eller deler av prosessen.

Forskjellen på disse to typene aktivitet er at skapende aktivitetene har større sjanse til å lykkes med å åpne dørene inn til læring: Mestring, Engasjement og Interesse.

Hvorfor skapende aktivitet i klasserommet istedenfor gjørende aktiviteter:

Den første grunnen handler om hvilke kompetanser elever har bruk for i framtida. Framtida er jo usikker, men man trenger personer som *tørr å ta risiko*, *tørr å feile og prøve på nytt*.

På Vitensenteret er vi "superglade" når elever gjør noen feil. Vi har et eget begrepet for det: *Fantastiske feil*. Feil er jo fantastisk, det er jo da vi lærer! Elevene trenes i å ta risiko og å tørre å feile med skapende aktiviteter i klasserommet.

Samarbeid er også en viktig grunn til å drive med skapende aktivitet. I et skaperverksted, får elevene god trening i samarbeid. Det er alltid slik at de jobber sammen i grupper med varierende størrelse (2 og 2, eller 4 og 4). Bruk av samarbeidsteknikker som f.eks. idemyldring og bruk av virkemiddelkort, gjør at elevenes tanker og gjøringer blir mer synlige.

Elevene lærer også å løse praktiske utfordringer som f.eks. å sage, lime, klippe, spikre, forskjellige sammenføyningsteknikker og lignende. Slike utfordringer kan elever lære å takle både gjennom gjørende og skapende aktiviteter, men de er ofte mer motiverte til å greie utfordringen dersom de arbeider skapende, siden det ofte er elevens egne ideer som skal realiseres. I tillegg trenes elevene i å løse tidsaktuelle problemer innen ulike tema. De lære seg også å settes seg inn i et tenkt scenario og leve seg inn i ulike roller.

Som lærer streber man etter å finne utfordringer og arbeidsmåter som oppleves autentiske for elevene. I åpne oppgaver som omfatter skapende aktiviteter er det ofte lettere for elevene å finne reelle utfordringer som de ikke oppfatter som oppkonstruerte. Oppgavene eller problemene kan

derfor oppleves veldig virkelighetsnære og kan bli svarene på de visjonene elevene har. De arbeidsmetodene eleven bruker i en skapende aktivitet har ofte også stor *overføringsverdi til virkeligheten*.

Det at elevene arbeider tverrfaglig gjør det mer realistisk i forhold til de problemstillingene elevene skal løse i framtiden. Realfagslærere har her en gyllen mulighet til samarbeid, spesielt med K&H lærerne, men også lærere som underviser i andre fag. Å arbeide tverrfaglig er i tråd, både med fagfornyelsen og i skaperskolens ånd.

Eierskap til egen læringsprosess oppnås også lettere gjennom skapende aktiviteter. Elevene har vært gjennom en skaperprosess, hvor de støter på ulike utfordringer og finner egne løsninger. Eleven bruker både hodet, hånd og hjerte som gir en indre motivasjon som fører til et eierforhold til utfordringen de skal løse, og til læringsprosessen.

Skapende aktivitet i skolen øver også elevenes evne til kritisk tenkning og ikke bare kildekritikk. Dessuten vil en slik aktivitet trene elevene til å forstå hva kunnskap er, og gjør dem i stand til å vurdere sine egne forestillinger, løfte blikket og tenke før man kritiserer seg selv og andre. Eleven trenes i å forstå, tolke og analysere argumenter fra medelever og andre.

Kanskje det viktigste punktet av alt: *Det finnes mere enn en løsning på et problem*. Elevene trener på, og må bli vant til, at det ikke bare er ett rett svar.

Eksempel fra 5. trinn i sløyd:

Bygg fuglekasser etter oppskrift.

Elevene har en oppskrift med ferdig lengder og design. Her er et klassisk eksempel på en gjørende aktivitet.

Med å trene seg som lærer på å bruke skapende aktivitet i klasserommet gjør at resultatet vil kunne se mer ut som vist nederst på figuren:

Det er dette vi håper skal være resultatet. Bildeeksemplene er ikke slik det nødvendigvis vil se ut, men kan illustrere forskjellen på "gjørende" og "skapende" aktiviteter.

Eksempel fra 5. trinn i sløyd:

Bygge fuglekasser som skapende aktivitet.

Eleven designer sitt eget fuglehus. Undersøker størrelsen på innflygningshullet for den fuglesorten de ønsker i fuglekassen. Kan rovdyr komme lett til? Må taket være flatt? Må inngangen være foran? osv. De lager arbeidstegning og beregner lengder, vinkler og valg av materialer. Man kan legge begrensninger på valgene, som for eksempel at man kun har 6 standard dimensjoner på de plankene man har til rådighet, noe som ofte er tilfelle ved en skole.

Likevel vil vi alltid slite med dilemma: Gjøreaktivitet vs. skaperaktivitet. Gjøreaktiviteter blir nødvendige når man må lære seg en spesiell teknikk eller kunnskap for å bli istand til å være kreativ. Man kan ikke be en elev, innenfor for eksempel temaet arkitektur, å lage en prototype av en bygning uten først å ha lært ulike sammenføyningsteknikker. Kanskje må man tillate seg en liten "timeout" i skaperprosessen for å lære de forskjellige teknikkene.

Når vi driver med skapende aktivitet i undervisninga, så kan det hende vi får slike gylne øyeblikk. Når disse aha-opplevelsene kommer, så har vi lærere en gyllen mulighet til å stoppe opp og gå litt utenfor læringsmålene.

(David Mark Shaw)

Dersom vi skal våge oss på en oversettelse av *Serendipity* så kan vi kanskje kalle det *snubleflaks, lykketreff* eller *tilfeldighetshell*.

1.6 Bruk av loggbok og nettsted for deling

Etter hver samling vil vi gi deltagerne noen utfordringer. Disse kan være knyttet til egen kunnskapsutvikling ved at de utfordres til å arbeide med oppgaver for egen del, mens noe vil også være oppgaver som handler om å teste ut undervisningsopplegg i klasserommet. Dessuten ønsker vi at de skal utvikle egne prosjekter og produkter.

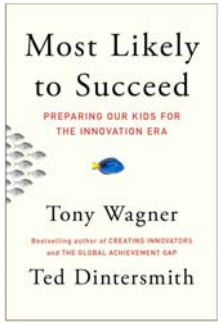
Loggboka, som vil være digital, skal dokumentere hva som skjer mellom samlingene. Hva som har gått bra og hva som utvikler seg annerledes enn forventet. Vi tenker oss at vi skriver disse spørsmålene inn i Loggboka og at deltagerne på de ulike skolene skriver inn sine erfaringer og spørsmål.

Alle deltagerne, også fra andre skoler, har mulighet til å lese hverandres dokumentasjon. Vi vil også legge opp til at deltagerne deler fra loggboken i starten av hver samling. Loggbøkene er også tilgjengelig for involverte medarbeidere ved Vitensenteret.

Inntil videre ser vi for oss at hver skole har hver sin mappe på en server hos kommunen, f.eks. Google doc hvor også loggbøkene ligger. Disse mappene kan også brukes til å dele informasjon internt på skolen.

1.7 “Most likely to succeed”

Som en hjelp til å tenke nytt ønsker vi å gi deltakerne en “kikk off” ved å vise filmen “Most likely to succeed” som forteller om et amerikansk skolesystem som i stadig større grad preges av å måle prestasjoner framfor å fremme læring, men som likevel viser at det finnes unntak som nettopp legger vekt på å stimulere barn og ungdoms skapende evner og kreativitet. En film som viser at det går an å bryte ut av en usunn undervisningspraksis.



Det er ikke vår mening med å vise filmen å hevde at slike forhold dominerer norsk skole. Vi er heldigvis ikke der, men heller å stimulere til nytenkning og se mulighetene som ligger i å gi frihet til utfoldelse og eksperimentering. Filmen har fått mange priser og er blitt vist i ulike sammenhenger verden over også i Norge.

Vi anbefaler også boka med samme navn: “Most likely to succeed” av ***Tony Wagner*** og ***Ted Dintersmith*** som er en utdypning av filmen.

2 Hva er et program?

Et dataprogram er en *oppskrift* for hvordan en jobb skal gjøres. Det er ikke så forskjellig fra å følge en kakeoppskrift. Istedet for at arbeidsoppgavene utføres manuelt, så er det elektroniske komponenter som utfører hver arbeidsoperasjon, styrt av programmet.

Et typisk eksempel er en *vaskemaskin*. Slike styres i dag av små datamaskiner, såkalte *mikrokontrollere*. Vi er godt kjent med arbeidsoperasjonene til en vaskemaskin: Fyll på vann, kjør forvask, tøm ut vann, fyll på nytt vann, varm opp vannet, ha i såpe, kjør trommelen, tøm ut vann, ha i skyllevann, gjerne med flere skyllinger for til slutt å slippe ut vannet og sentrifugere. Dette er en rutinejobb hvor hver operasjon er styrt av et program i mikrokontrolleren, et program som gjerne er skrevet av en ingeniør.

Aktuatorer

For å få til dette må mikrokontrolleren kunne slå av på strømmen til mange ulike elektriske innretninger, som f.eks. en ventil som slipper inn vann, en motor som snurrer trommelen, elektroniske låser som holder døra lukket og åpner når det er tid for det, slår på vannpumpa slik at vannet pumpes ut og slår på varmeelementet slik at vannet blir varmet opp. Alle slik ting som utfører en jobb kaller vi *aktuatorer* (av "action"), og husk at de må kunne styres ved hjelp av elektrisk strøm siden arbeidsoppgavene skal utføres automatisk. Mikrokontrolleren er den enheten som slår av og på de ulike aktuatorene i riktig rekkefølge og til riktig tid.

Sensorer

Det er ikke alltid nok å kunne styre alt i riktig rekkefølge til riktig tid. Vi kan tenke oss at vanntemperaturen i springen er forskjellig fra sommer til vinter. Dermed vil det kreve lengre tid å varme opp vannet til 60°C om vinteren enn om sommeren. For å bøte på dette så inneholder vaskemaskinen en temperaturmåler som registrerer at vannet har nådd den ønskede temperaturen. En slik temperaturmåler kaller vi en *sensor* fordi den "føler" på vanntemperaturen og varsler mikrokontrolleren at den ønskede temperaturen er nådd og varmeelementet kan slås av.

En vaskemaskin inneholder flere slike sensorer som f.eks. sensor som registrerer at lokket er ordentlig lukket, uten at det er tilfelle starter ikke maskinen. Likeså registreres at det er fylt tilstrekkelig med vann i maskinen, derfor finnes det en sensor som forteller at ønsket vannstand er nådd. Før det er skjedd starter ikke oppvarmingen, osv.

Programmet

Alt dette styres av et dataprogram som avleser sensorer, tar beslutninger på bakgrunn av avleste verdier (f.eks. temperatur) og starter aktuatorer når det er på tide. Det kanskje mest fascinerende er at det skrevne ord i programmet kan være istand til å styre fysiske prosesser, at når det står i programmet *Start motoren!* så starter motoren, og når det står *Slipp inn vann!* så åpnes ventilen slik at vannet slippes inn gjennom kranen.

For å få til dette så må kommandoordene i programmet oversettes til elektriske impulser. Skal dette være mulig er det helt avgjørende at vi bruker de riktige ordene slik at *oversetteren* (kompilatoren) forstår hva vi mener. Skriver vi feil vil den si ifra og vi må rette opp feil. "De riktige ordene" er en del av et språk, et *dataspråk* og dem finnes det mange av. I denne sammenhengen skal vi bruke et dataspråk hvor kommandoordene er pakket inn i *byggeklosser* eller *blokker*. Språket kalles derfor *blokkode*.

La oss begynne litt forsiktig med å lage og "kjøre" programmer uten å bruke PC.

2.1 Programmering uten bruk av PC

2.1.1 "Kompisprogrammering"

Mål med øvelsen:

Her skal deltakerne programmere en kompis, som i denne oppgaven skal være "robot". For å få til dette må deltakerne lage en oppskrift ved hjelp av piler (algoritme). Oppgaven er kompleks og må deles opp i mindre ledd/sekvenser for at den skal kunne løses.

Varighet:

10 – 15 minutter

Utstyr:

11 stk. A4-ark med sifrene/tallene fra 0 til 10:

8 stk. programmeringskort til hver gruppe med kommandoene:

Gå ett skritt fremover

Gå to skritt fremover

Gå tre skritt fremover

Gå fire skritt fremover

Gå ett skritt bakover

Snu 90 grader til venstre

Snu 90 grader til høyre

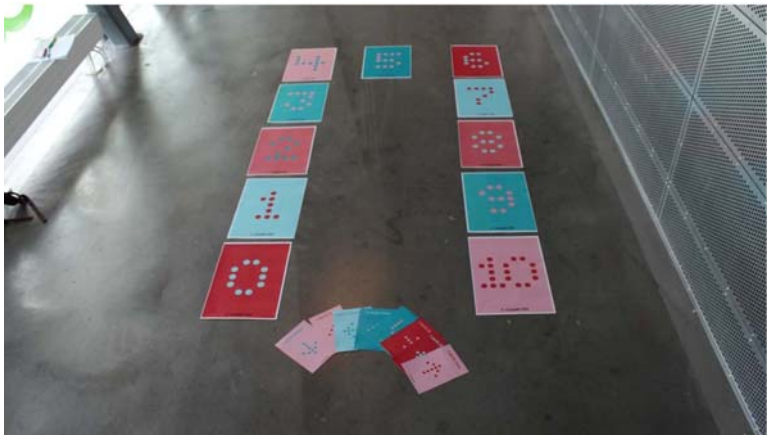
Snu helt om180 grader

Et ark som viser «banen» i miniutgave til hver gruppe

En spillebrikke (et viskelær eller en stein kan også brukes).

Gjennomføring:

Deltakerne går sammen i par. En skal være «robot», den andre skal programmere roboten.



Bygg opp banen med de nummererte kortene på gulvet slik bildet over viser.

Hvert par skal ha hver sin bunke med de åtte programmeringskortene, et A4-ark med banen i miniutgave og en spillebrikke.

Gruppen skal sammen lage en programmeringskode som programmerer «roboten» til å gå fra 0 til 10. De skal legge programmeringskortene i en rekkefølge slik at "roboten" ved å følge instruksjonene slavisk, akkurat skal nå fram til posisjon 10 når instruksjonen på det siste kortet er utført. Hvert kort skal kun brukes en gang.

Hver gruppe planlegger, lager og tester sitt program (en sekvens av kommandoer), før det prøves ut på den store banen. Testingen kan gjøres på A4-arket med banen i miniutgave, spillbrikken kan være "roboten". De som har programmert skal lese opp sekvensen og «roboten» skal følge instruksjonene ved å flytte spillebrikken. Start på 0.

Klarte alle å løse oppgaven? Fant de forskjellige løsninger eller hadde alle samme løsning?

Regler:

«Roboten»/eleven skal ta seg fra posisjon 0 til sluttposisjon 10.

Hvert programmeringskort kan kun brukes en gang.

Man må ikke bruke alle programmeringskortene.

Slik kan man øke vanskelighetsgraden:

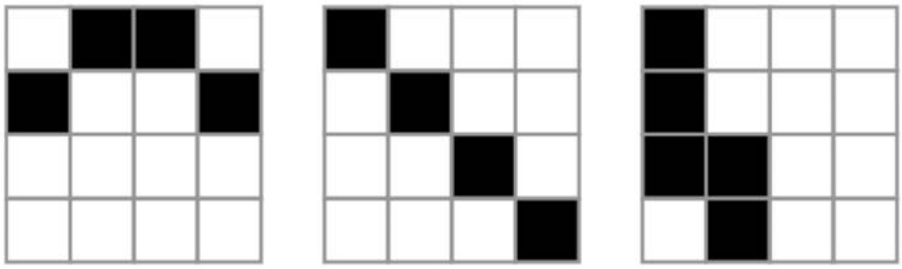
Ta bort programmeringskortet "Gå et skritt frem" og la gruppene prøve å løse oppgaven på nytt.

Ferdig opplegg med lærerveiledning til bruk i klasserommet finnes på: https://www.vitensenter.no/superbit/laerer/forarbeid/oppgave-3/

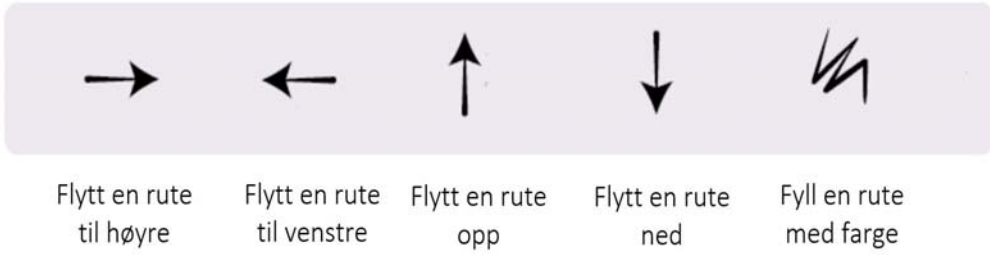
2.1.2 Rutenettprogrammering

I denne oppgaven får du prøve både å lage og lese programinstruksjoner.

Vi starter med å lage et program som steg for steg viser hvordan man tegner et enkelt bilde i et rutenett med 4 x 4 ruter. Du kan velge om du vil lage ditt eget bilde, eller velge ett av de tre bildene under:



Programmet skal bruke disse symbolene for å instruere den som leser programmet:



Programmet skal ta som utgangspunkt at man starter i øverste venstre rute. Skriv inn stegene i programmet ditt i rutene under (steg 1 i rute 1 og så videre).

Eksempel:

Steg 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Steg 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Steg 31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

Program 1

Steg 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Steg 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Steg 31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

Program 2

Steg 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Steg 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Steg 31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

Program 3

Steg 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Steg 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Steg 31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

Program 4

Steg 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Steg 16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Steg 31	32	33	34	35	36	37	38	39	40	41	42	43	44	45

3 Micro:bit

I dette kapittelet skal vi se nærmere på Micro:bit-kretsen.

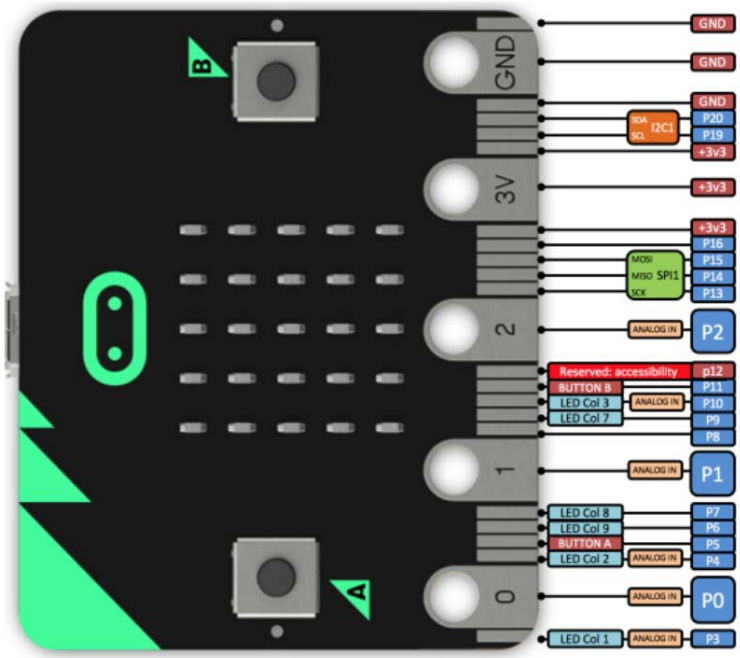
Micro:bit er en liten komplett datamaskin på størrelse med et kredittkort. Ved hjelp av dataprogrammer som vi skriver og som oversettes til et språk mikro:bit kretsen forstår, så kan vi få den til å gjøre det vi ønsker når vi sender programmet over til kretsen.

Den er imidlertid avhengig av å kunne kommunisere med omverdenen, både for å lese av sensorer (f.eks. temperatur) og for å styre aktuatorer (f.eks. motoren i vaskemaskinen). Dette gjør den ved hjelp av elektriske signaler som overføres via elektriske kontakter langs en *kantkontakt*, som kalles *porter* og som er nummerert fra P0 – P20.

Disse portene kan ha ulike funksjoner etter som hvordan vi bestemmer oss for å bruke dem. De kan enten ta imot signaler utenfra (innganger) eller sende ut signaler (utganger). De kan være *digitale*, dvs. forholde seg til full spenning (3,3V) eller ingen spenning (0,0V), "på" eller "av", "1" eller "0". Eller de kan være *analoge* og også kunne forholde seg til spenninger mellom 0 – 3,3 V, f.eks. 1,5V.

3.1 Tilkoblingspunkter – porter

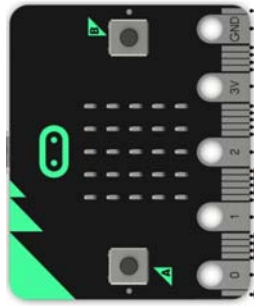
Ved tilkobling av kantkontakten til Micro:bit får man tilgang til et utvalg av porter, både inn- og utganger og analoge og digitale. Figuren under viser en oversikt over kantkontakten.



Som det framgår av figuren så har kretsen 20 porter hvorav 6 er analoge inn- eller utganger (P0, 1, 2, 3, 4 og 10), hvorav P0, 1 og 2 også kan brukes som berøringssensorer. De resterende 15 digitale portene (P5, 6, 7, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 20) kan settes opp som inn- eller utganger etter behov så fremt de ikke er reservert til spesielle oppgaver som f.eks. P12 som er reservert til annet formål. Dessuten er P3, 4, 6, 7, 9, 10 brukt til å styre displayet, men kan frigjøres til andre formål etter behov. Vi legger også merke til at P3, 4 og 10, kombineres med analoge innganger. Hvilket betyr at lysdiodene også kan fungere som lyssensorer. Knappene A og B er koblet til P5 og P11.

I tillegg ser vi at drivspenningen er oppgitt til 3,3 V, men Micro:bit kan også kjøres på 3 V (dvs. ett CR2032 eller 2 x AA eller 2 AAA). Av figuren over ser vi at forsyningsspenningen er tilknyttet tre kontakter. Dessuten er tre kontakter koblet til jord (– polen på batteriet).

Pinne P19 (SCL) og P20 (SDA) kan også brukes til kommunikasjon via seriebuss (I^2C). Dette er en særdeles anvendelig seriekommunikasjonslinje for å kommunisere med andre digitale sensorer o.l. Det er også gitt mulighet for trelinje SPI-buss (P13, 14 og 15).



Jord (–)
3V (+)
P2
P1
P0

Fem av tilkoblingspunktene er utvidet og gjennomhullet slik at det skal være mulig å koble til bananstikkere eller krokodilleklemmer. Dette gjelder batterispenningen 3V (+ og –) og portene P0, P1 og P2. Dette gjør at en, som vi skal se, kan komme igang svært fort med et minimum av tilleggsutstyr. Samtidig som det ligger et betydelig utviklingspotensial i kretsen.

Tabellen under gir en oversikt over hvordan vi kan bruke de ulike portene (P0 – P20).

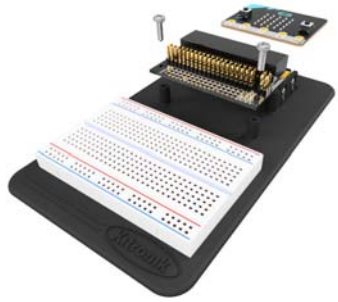
Port	Digital inn/utganger	Analog inn/utganger	Berøringssens.	Spesiell bruk	Kommunikasjon
P0		Rødt lys		Servo 1	
P1		Gult lys		Servo 2	
P2		Grønt lys		Servo 3	
P3				LED Kol. 1	
P4				LED Kol. 2	
P5		Knapp A		Knapp A	
P6				LED Kol. 9	
P7				LED Kol. 8	
P8		Lyd utgang			
P9				LED Kol. 7	
P10				LED Kol. 3	
P11				Knapp B	
P12	Reservert for spesiell bruk				
P13					SPI (MOSI)
P14					SPI (MISO)

Port	Digital inn/utganger	Analog inn/utganger	Berøringssens.	Spesiell bruk	Kommunikasjon
P15					SPI (SCL)
P16					
P19					I^2C (SCL)
P20					I^2C (SDA)

Kantkontakten

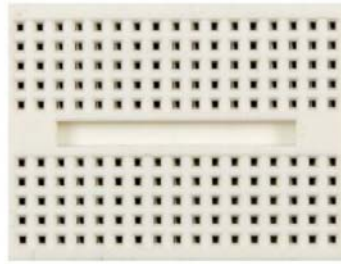
Det finnes kantkontakter som passer til Micro:bit og som selges til en overkommelig pris (typ. £4 montert på kretskort og *dobbel stiftlist*). Figuren over til venstre viser en kantkontakt uten kretskort. Bildet til høyre viser en kantkontakt og et kretskort som kan utstyres med dobbel stiftlist. Det er den siste vi skal bruke i dette prosjektet.

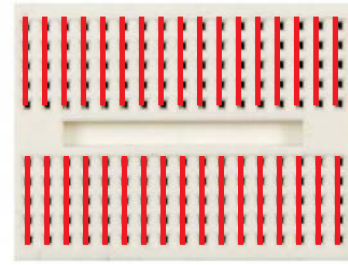
For den som ønsker å gå videre og utforske mulighetene til kretsen, kan en anskaffe et *Inventors kit* som gjør det lett å koble til eksterne komponenter på et koblingsbrett (se figuren til høyre).



I tillegg til byggeplata inneholder Inventors kit ledninger (jumpere) og en rekke sensorer og aktuatorer, men selges uten Micro:bit så den må kjøpes separat. Pris i Norge ca. kr. 500,- inkl. MVA. Mens Micro:bit koster ca. kr. 250,- inkl. MVA.

Koblingsbrettet

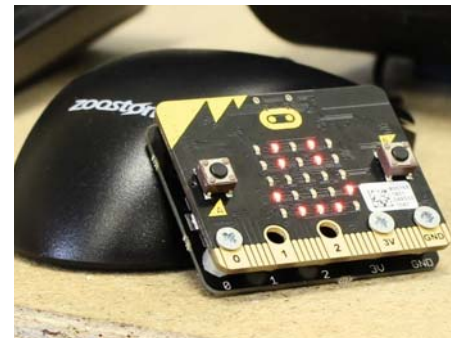




Et koblingsbrett er en komponent som gjør det mulig å koble sammen komponenter elektrisk uten at vi trenger å lodde. Det betyr at det er lett å gjenbruke komponenter og ledninger. Figuren til høyre viser et lite koblingsbrett som vi skal bruke. Dette er et plastbrett med en mengde huller som vi kan stikke ledninger og beina til komponenter ned i. Under hullene er det skjult metallskinner som forbinder de ulike hullene elektrisk. Huller som er forbundet elektrisk er markert med røde linjer på figuren over.

3.2 Batteripakke for Micro:bit

Dersom Micro:bit skal brukes uten å være tilkoblet PC, er det praktisk å montere på et kretskort som inneholder batteri (CR2032 – 3V), lydgiver (piezo elektrisk) og bryter. Kortet monteres til Micro:bit med tre maskinskruer med mutter og avstandsstykker.



3.3 Lydgiver/Høyttaler

Det finnes lydgivere. Den vesle svarte lydgiveren som følger med Super:bit kan egne seg godt i klasserommet dersom man ønsker å dempe lydnivået. Imidlertid egner den seg dårlig dersom man ønsker å formidle lyder eller musikk til et større publikum.

Vi har derfor gått til anskaffelse av en liten forsterker med høyttalere som gir kraftigere lyd. Denne skal tilføres 3V og jord i tillegg til signalet. Høyttaleren er gjengitt på figuren til høyre.

4 Læringsmål

Som nevnt er denne kursdagen planlagt som en veksling mellom *opplæring i programmering* av Micro:bit og *kreative prosesser*. Det er vårt håp at de kreative prosessene skal støtte opp om utviklingen av det gjennomgående prosjektet, selv om vi har lagt en rekke føringer.

Tabellen under beskriver sekvensen av delprosjekter som til sammen skal gi produktet *Trafikklys*. De ulike delprosjektene er farget grønn, gul og rød avhengig av hvor krevende vi anser dem å være. Det er også foreslått alternative delprosjekter avhengig av i hvilken retning prosjektet tar. Et viktig poeng har vært å vise hvordan et større prosjekt kan bygges opp av mange mindre delprosjekter for så gradvis å settes sammen til det endelige produktet.

Oppdrag	Beskrivelse	Alternativ	Læringsmål
1 side 39	Få en lysdiode til å blinke		Programstruktur. Skriv til port. Gjenta for alltid. Pause-funksjon Oppkobling Litt elektronikk
2 side 41	Få tre lysdioder til å blinke samtidig	Få flere lysdioder til å blinke i sekvens	Bruk av flere porter
3 side 44	Programmer trafikklyset slik at det får i riktig sekvens (rød, gul, grønn)		Bruk av ulike pauser
4 side 45	Bestille grønt lys for fotgjengere	Bruk av ekstern knapp for eksempel festet på det fysiske trafikklyset	Bruk av knapp A Funksjoner som interrupter loopen
5 side 45	Lag og programmer symboler for fotgjengere og vis dem på micro:bit skjermen		Skrive ut symboler på skjermen
6 side 46	Test ulike lyder med tanke på å gi lyd til trafikklyset som passer til situasjonen	Lage forskjellig lyd som passer rødt og grønt	Programmere lyd og musikk, takt og tone
		Lage forskjellig lyd som passer for fast grønt og blinkende grønt	
		Lag melodier som avspilles ved forskjellige farger	Programmere tonefølger, musikk. Bruk av høyttalerkort
7 side 49	Lage blinkende grønt lys mot slutten av grønn periode		*Gjenta-sløyfe* et visst antall ganger
8 side 50	Legge på passende lyd for synshemmede til det grønne lyset	Ev. lage forskjellige lyder for kontinuerlig og avbrutt grønt	Bruk ideene fra delprosjekt 5

9 side 50	Sett sammen ulike programmer for å gi et trafikklys med bestilling av grønt med lyd for synshemmede.		Ha flere vinduer opp samtidig
10 side 51	Bruke lyssensoren for å slå på blinkende gult ved mørkets frambrudd	Bestilling av grønt lys etter at det er blitt mørkt Radiostyrt blinkende gult lys	Avlesning av sensor. Bruk av variabel, sensorverdi. Hvis-setning og logisk sammenligning
11 side 53	Sammenstilling av flere programmer. Bestilling av grønt lys med lyd og blinkende grønt ved mørkets frembrudd	Bruk av funksjoner for å gi bedre oversikt.	Bruk av funksjoner
12 side 55	Planovergang ved kryssing av jernbanespor med bom ved bruk av 180° servo	To lys og to servoer som går ned på hver side av planovergangen, fjernstyrt av en tredje mikro:bit	Bruk av servoer. Ev. bruk av radio. Montere en fysisk bom.
		Fire trafikklys settes sammen til et komplett lyskryss. Kommunikasjonen mellom lysene gjøres med radio	Radiokommunikasjon med flere micro:bit bruk av en master.
		Bit:bot kjører mot bommen og stopper automatisk dersom bommen er nede	Enkel bruk av Bit:bot Bruk av avstandsmåler
13	Bestill grønt lys for biler ved bruk av magnetsensor i kjørebanen		En micro:bit med magnetsensor i kjørebanen detekterer magnet om bord i bil som styre lys trådløst

Oppdrag 13 er ikke tenkt å tilbys under denne opplæringen, men ev. peke fremover ved utvidelser av prosjektet.

5 Bruke av Make:bit grensesnittet – Makecode

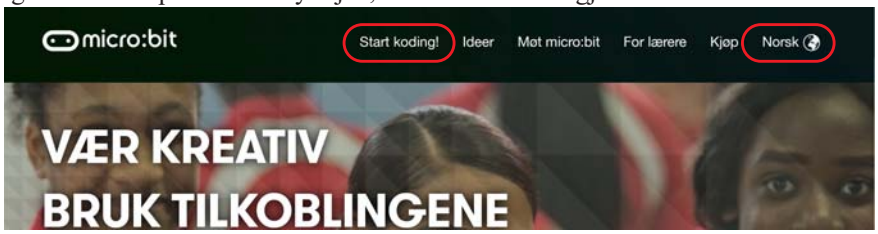
5.1 Programvare, arbeidsflaten og lagring av prosjektfiler

Micro:bit kan kodes med ulike språk bl.a. ved hjelp av Python, Java eller *blokkoding* som er basert på Java. Her skal vi konsentrere oss om å bygge opp programmet ved hjelp av *blokker*. Denne typen programmering kalles derfor *blokkoding*.

Nettressursen

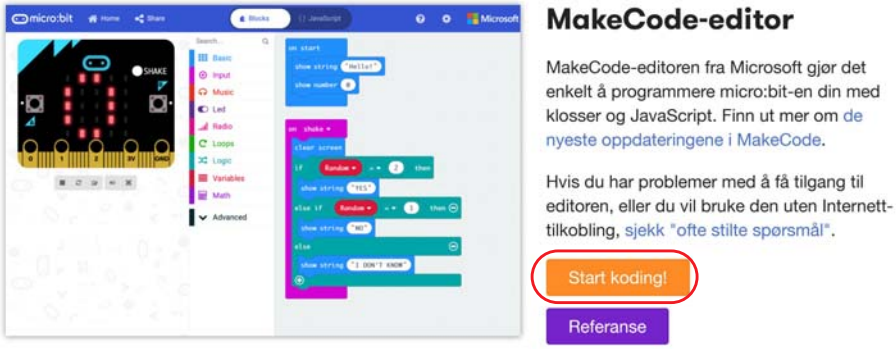
De fleste nettlesere kan brukes, men kan gi litt varierende prosedyrer for lagring og overføring av filer. For de som bruker Chromebook så er nettleseren gitt. Noen anbefaler bruk av Chrome, versjon 65 eller nyere, da fungerer webUSB som gjør overføring av filer fra datamaskin til microbit til en lek.

1. Gå til nettstedet: https://microbit.org/
2. Velg norsk som språk fra menylinjen, om det ikke alt er gjort.

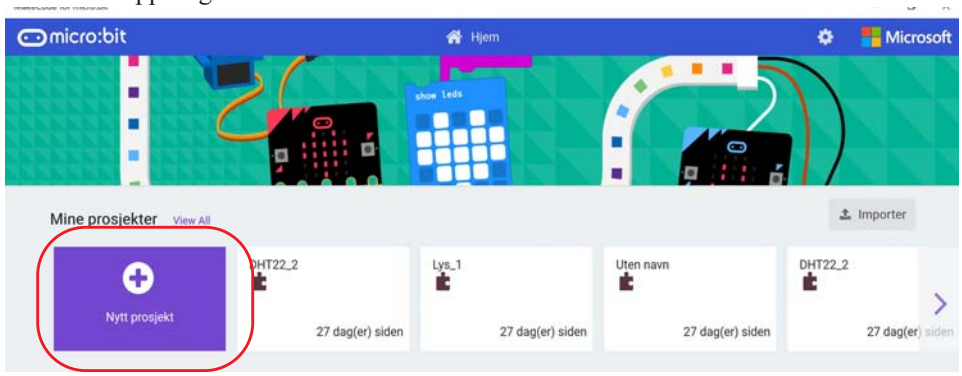


Start programmet

3. Velg *Start koding* fra menylinja. Gå ned til den delen av siden som er vist på bildet under, og trykk på *Start koding*:

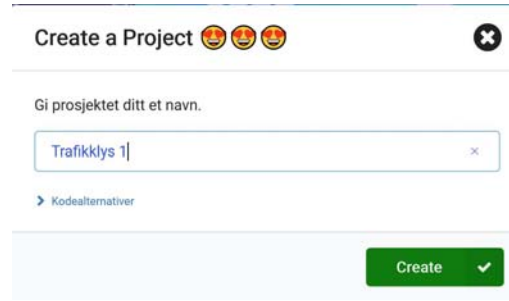


4. Du får da opp følgende vindu:

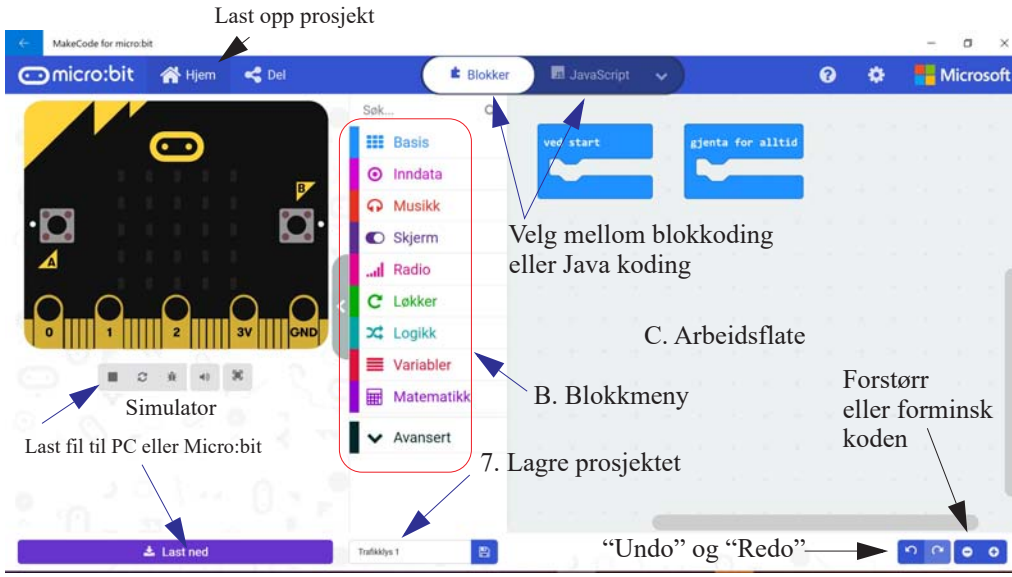


Velg *Nytt prosjekt*

5. Skriv inn navnet på ditt første prosjekt: *Trafikklys-1*
Vi ser for oss å lage mange Trafikklys delprogrammer (Trafikklys-1,-2,-3 ...) som vi til sist kan sette sammen til et større program etter ønske.



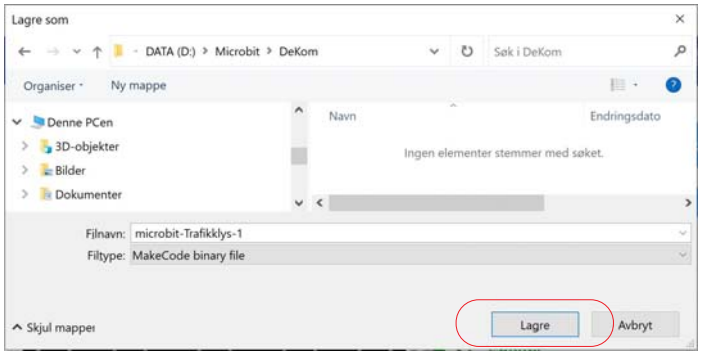
6. Du skal nå se et bilde omtrent som dette:



Her er en kort beskrivelse av gangen i arbeidet:

5.2 Lagre programfilen

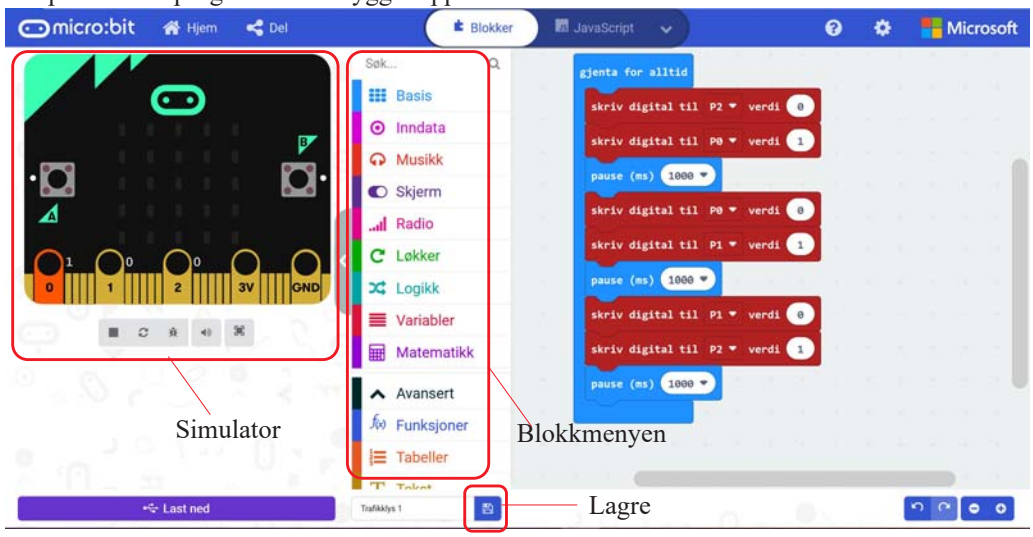
1. ***Lagre prosjektet***
   Du skrev inn prosjektnavnet når du gikk inn i programmet. Det kan være lurt å lagre prosjektet først som sist. Du trykker da på disketten til høyre for prosjektnavnet og får mulighet til å legge filen i en katalog. Pass på å opprett en katalog hvor du finner igjen filen senere. Trykk *Lagre* nederst til høyre.



Skriv programkode

2. **Skriv programmet**

   Ved å velge blant fanene i *blokkmenyen* får du opp lister over blokkommandoer. Disse dras fram av menyen og settes sammen på *arbeidsflata* slik at blokkene tilsammen blir det ønskede programmet som i eksempelet vist på figuren under. Vi skal ganske snart se nærmere på hvordan programmet er bygget opp



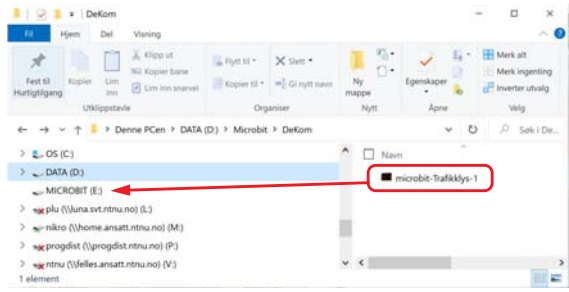
Legg spesielt merke til *simulatoren* til venstre i bildet. Denne vil vise hvordan programmet påvirker mikro:bit'en eller hvordan bruk av knapper eller bevegelse av micro:biten påvirker programmet. Det kan være praktisk å bruke simulatoren for å teste ut programmer før man overfører det til den fysiske micro:biten. Den oransje fargen lengst til venstre indikerer at denne kontakten har spenning (3V). Den skal kanskje slå på en lysdiode.

Overføring av programmet til micro:bit

3. **Lagre filen**
   Før du flytter filen over til micro:bit, må du lagre de forandringene du har gjort. Dette gjør du ved å trykke på disketten nederst midt på skjermbildet (se figuren over). Du må gjerne skrive over forrige versjon av programmet eller lagre det under et nytt navn.

4. **Koble til micro:bit**
   Det er nå på tide å koble til micro:bit til USB-inngangen. Bruk den medfølgende USB-kabelen.

5. **Overfør filen til micro:bit**
   Du kan bruke *Filutforskeren* i Windows til å flytte fila fra katalogen der du lagret den og slipp den over symbolet av micro:biten. Du vil se et blinkende gult lys på undersiden av micro:bit idet filen overføres.



Programmet skal nå være overført.

Ønsker du mer informasjon velger du spørsmålstegnet på menylinja øverst til høyre. Her har du mulighet til å stille spørsmål eller få opplæring.

6 Programmering av trafikklys

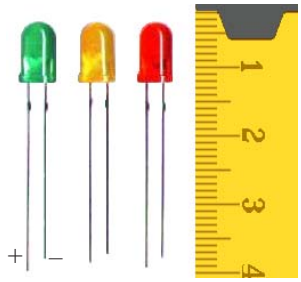
Vi tar utgangspunkt i tabellen i avsnitt , side 33 som viser et forslag til framdrift. Alle delprogrammene kan etter hvert støtte opp om det endelige produktet *Trafikklys*, dersom man ønsker det.

6.1 Oppdrag 1 – Lysdiode som blinker

Oppdrag 1: *Lag et program for micro:bit som slår av og på en lysdiode.*

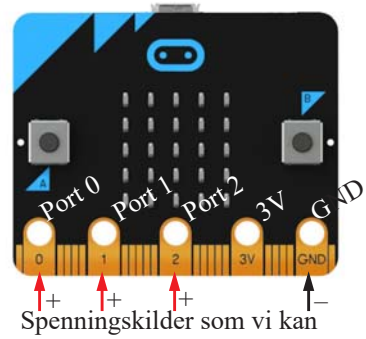
6.1.1 Oppkobling – bruk av krokodilleledninger

Vi kan enkelt koble opp kretsen ved hjelp av krokodilleledninger og en lysdiode. Micro:bit har kontakter langs kanten, noen brede og noen smale (til venstre på figuren under). Vi kan slå av og på en spenning mellom GND (–) og en eller flere av disse kontaktene ved hjelp av programmet.



Kort kobles til minus
Lang til pluss

Krokodilleledninger



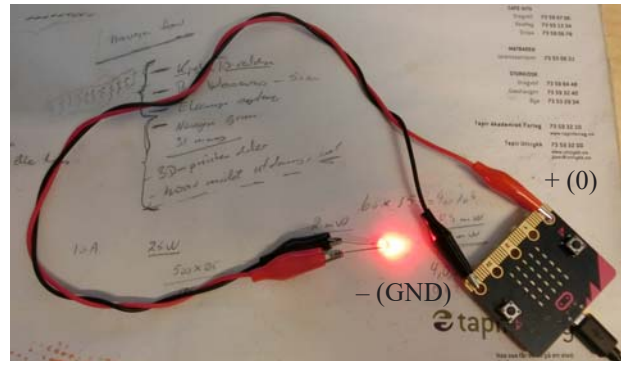
Spenningskilder som vi kan styre fra programmet

Lysdioder har en pluss- og en minus-pinne. Den *lange pinnen* på lysdioden er pluss og den korte er minus. Pluss-pinnen kobles til en av de brede kontaktene, 0, 1 eller 2. Den korte kobler du til minus lengst til høyre på kontakten (GND).

Det er viktig å koble riktig ellers lyser ikke dioden. Den blir heller ikke ødelagt om du kobler feil.

Fra elektrisitetslæra vet vi at vi både må koble oss til minus og pluss på batteriet, vi må lage en *sluttet krets*, Minusen på micro:bit finner du lengst til høyre på kontakten (GND – Ground (jord)). Portene 0, 1, 2 osv. er plusspoler som vi kan slå av og på med programmet.

Da er du klar til å koble opp kretsen som vist på figuren til høyre. *Til denne oppgaven bruker vi + (P0) og – (GND).*



For å kunne programmere micro:bit og få lys i lysdioden, må du koble deg til en USB-port på PC'en. USB-porten gir micro:bit spenning.

6.1.2 Lag programmet som slår av og på lysdioden

Ingen ting skjer før du ber programmet om å slå på spenningen på port 0 (P0).

Når du er klar kan du skrive inn programmet som er vist på figuren til høyre. Gå gjerne til avsnitt 5, side 35 dersom du er usikker på hvordan du skal komme i gang.

Menyene og blokkene har samme farge slik at det skal være lettere å finne ut under hvilken fane du finner blokken.

Forklaring av programmet.

1. Vi foreslår å bruke ***gjenta for alltid***-blokken (meny Basis) siden vi vil at blinkingen skal fortsette til vi slår av strømmen.
2. **Slå på lysdiode**
   I "gapet" til *gjenta for alltid*-blokken plasserer vi oppskriften på hva programmet skal gjøre. Velg blokken ***skriv digital til p0 verdi 1*** (meny *Tilkobling*). Det betyr at vi setter spenningen på port 0 til 3 V (logisk 1)[7]. Merk at med den vesle pila ved *P0* på blokken, kan du endre hvilken port du "slår på".
3. **Vent i 1 sekund**
   Etter at vi har slått på lysdioden på port 0, må vi la den være påslått en stund. Programmet tar en pause med kommandoen ***pause (ms) 1000*** (meny Basis). Som betyr at programmet venter i 1000 millisekunder som er lik 1 sekund.
4. **Slå *av* lysdioden**
   Så bruker vi den samme kommandoen ***skriv digital til p0 verdi 0*** (meny *Tilkobling*) for å slå av lysdioden. Legg merke til at vi denne gangen setter verdien til 0, vi slår av spenningen.
5. **Vent i 1 sekund**
   Deretter venter vi igjen i 1000 ms (1 sek.) (meny Basis) med kommandoen ***pause (ms) 1000.*** før vi begynner på nytt igjen med å slå på lysdioden.
6. **Lagre og legg programmet over på micro:bit'en**
   Lagre programmet og bruk filutforskeren til å legge programmet over på micro:bit (se ev. avsnitt 5.1, side 35 kulepunkt 11).

7. Det er vanlig å kalle kontaktene for porter, da de på en måte er en "port" mellom programmet og verden rundt oss.

7. **Lysdioden skal nå blinke**
   Om den ikke blinker så sjekk at du har koblet riktig, at lysdioden er satt rett vei og at du har koblet til og programmert port 0.

Løsningsforslag: Vedlegg A.1, side 60.

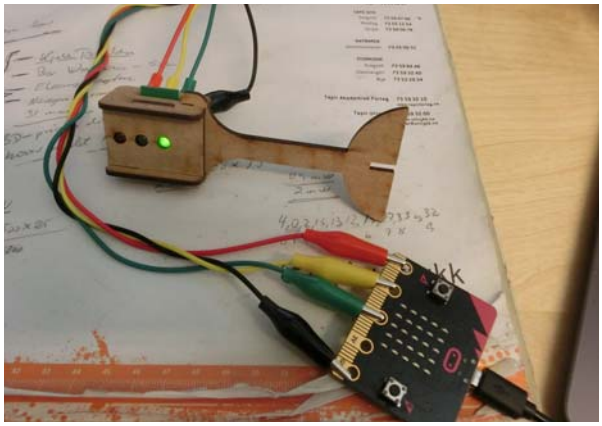
6.2 Oppdrag 2 – Tre lysdioder som blinker

Oppdrag 2: *Lag et program som slår av og på de tre lysdiodene i trafikklyset i tur og orden. Hver diode skal lyse i 1 sekund. De skal fortsette med det til strømmen blir brutt.*

I denne øvelsen skal vi koble opp *tre lysdioder* og få dem til å blinke. Denne gangen skal vi bruke det ferdiglagde trafikklyset som er lagt ved pakken.

6.2.1 Oppkobling av trafikklyset med krokodilleledninger

Det er laget modeller av trafikklys som har et lite kretskort som holder tre lysdioder, rød, gul og grønn. Kortene har hull for bananstikker på baksiden som også kan brukes for å koble til krokodilleklemmer. De tre store hullene har fått navnene *raud* (P0), *gul* (P1) og *grøn* (P2). Du skal nå koble krokodilleledninger til alle tre hullene i tillegg til jord (–) (GND) som vist på figuren under.



Det er viktig at krokodilleklemmene settes som vist på bildet over slik at de ikke kommer i kontakt med loddepunkter ved siden av de store hullene. **NB!** *Husk å koble den svarte ledningen til hullet som er merket GND (–).* Tilsvarende kobler du de andre endene av krokodilleledningene t*il portene 0, 1, 2 og GND* på micro:bit som vist på bildet over.

6.2.2 Lag programmet som slår av og på en og en lysdiode

Vi skal lage et program som slår av og på de tre lysdiodene i tur og orden. Hver diode skal lyse i 1 sekund. De skal fortsette med det til strømmen blir brutt. Her er "oppskriften":

1. Slå på rød lysdiode (P0)
2. La den lyse i 1 sekund
3. Slå av den røde lysdioden og slå på den gule lysdioden (P1).
4. La den gule lyse i 1 sekund
5. Slå av den gule lysdioden og slå på den grønne lysdioden (P2).
6. La den grønne lyse i 1 sekund
7. Begynn på nytt igjen

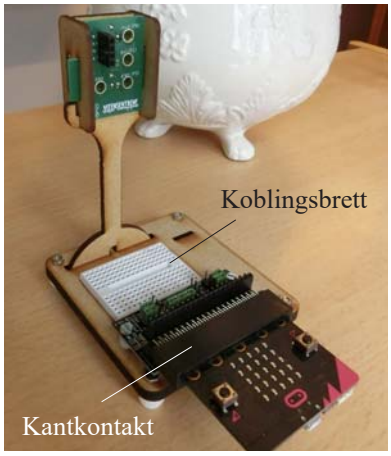
Tips: Husk å velg riktig port-nummer for de forskjellige lysdiodene.

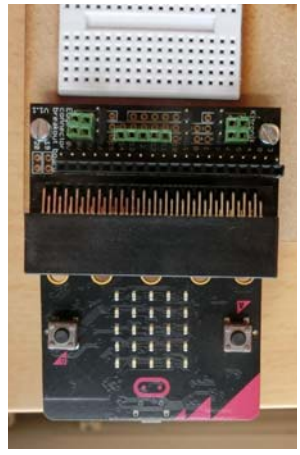
Skriv koden ut fra det du lærte i oppdrag 1. Lagre programfila under navnet Trafikklys-2.

Løsningsforslag: Vedlegg A.2, side 60

6.2.3 Oppkobling i jigg med jumpere

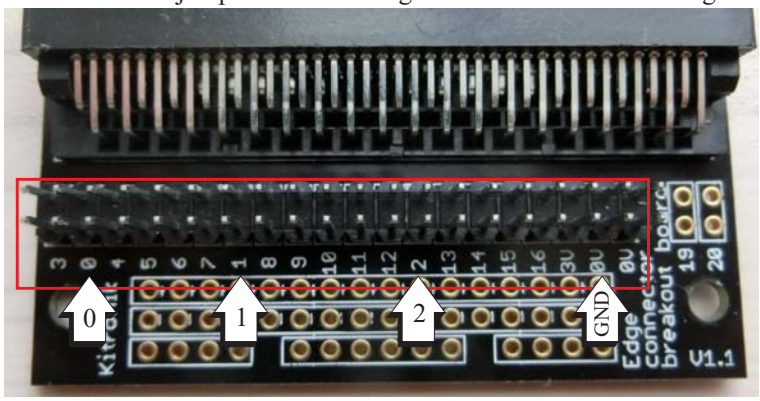
En *jigg* er en innretning som gjør det enklere å koble opp på en oversiktlig og stødig måte. Vi har laget en slik jigg for trafikklyset.



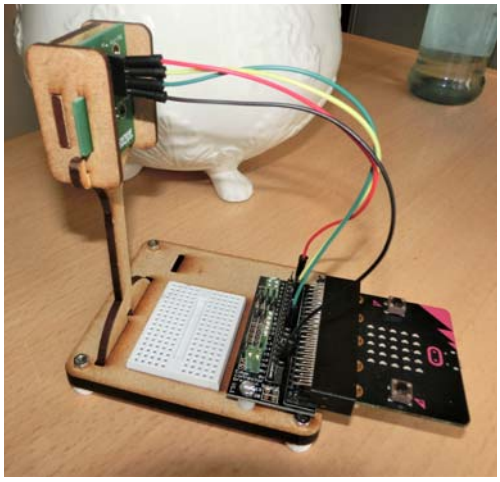


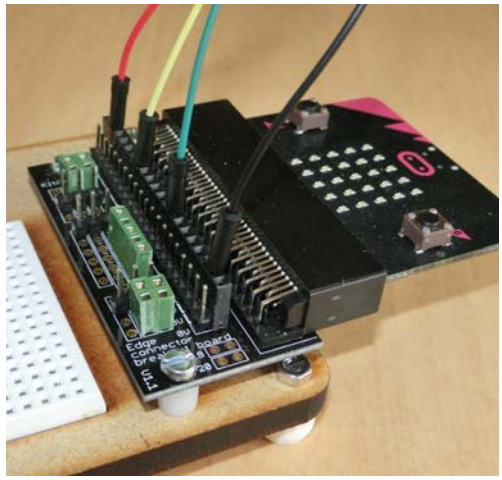
På kretskortet er det montert en sort *stiftlist* (se bilde over i midten). Dette er små stifter som vi kan stikke ledninger med en hylsekontakt inn på. Ledningene kaller vi *jumpere* (bildet nederst til høyre). På venstre side av stiftlista ser vi at det står P0, P1, P2 og GND altså det samme som for de store hullene.

1. Koble fra krokodilleledningene på lyskrysset og ta av tverrstaget i foten
2. Press trafikklyset ned i jiggen (bilde over til venstre)
3. Målejiggen inneholder trafikklyset, et hvitt *koblingsbrett* og en *kantkontakt* hvor micro:bit-kortet kan plasseres. Kantkontakten er montert på et kort med en *stiftlist*. Her kan du nå de samme kontaktene med jumpere som du tidligere brukte krokodilleledningene til.



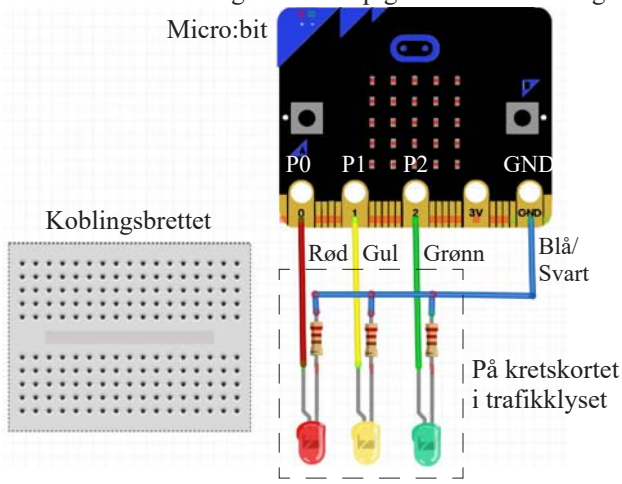
4. Bruk jumpere til å koble opp ledninger mellom stiftlista på trafikklyskortet og stiftlista på kantkontakten. Bildet under viser hvordan det kan ta seg ut.





5. Plugg i USB-kontakten og sjekk at lysdiodene blinker som forventet. Vi antar at programmet fra oppdrag 2 fortsatt ligger i micro:bit. Husk å koble til USB-kontakten slik at kortet får strøm.

Figuren under viser en skjematisk oppkobling. Lysdiodene og motstandene er montert på kretskortet på trafikklyset. Som vi ser så trenger vi foreløpig ikke bruke koblingsbrettet.



6.3 Oppdrag 3 – Programmer trafikklyset slik at det har riktig lyssekvens

Oppdrag 3: *Lag et program som gjør at lyset skifter fra rødt til grønt og tilbake til rødt slik som et virkelig trafikklys gjør. Bruk følgende tidsrom:*

- *Rødt i 5 sekunder*
- *Mellomtilstand i 1 sekund*
- *Grønt i 5 sekunder*
- *Mellomtilstand i 1 sekund*
- *Tilbake til rødt*
- *Gjenta sekvensen så lenge kortet har spenning.*

6.3.1 Oppkobling

Ingen ekstra oppkobling for dette oppdraget.

6.3.2 Programmering

Følgende kan være en god framgangsmåte:

1. Finn ut hvordan skiftet fra grønt til rødt og tilbake igjen egentlig skjer for et virkelig trafikklys
2. Endre på programmet fra oppdrag 2
3. Test og sjekk at det fungerer i henhold til oppdraget

Du trenger følgende type blokker:

Løsningsforslag: Vedlegg A.3, side 61.

Lagre programmet under navnet Trafikklys-3, overfør til micro:bit og test programmet.

6.4 Oppdrag 4 – Bestill grønt lys

I det neste oppdraget skal vi bruke knappene på micro:bit.

> **Oppdrag 4:** *Lyset skal vanligvis være rødt også når programmet starter. Når man trykker på knapp A skal det skifte til grønt og forbli grønt i 5 sek. før det blir rødt igjen. Husk at lyset skal fungere som et virkelig trafikklys med riktig skifte med gult.*

For å løse dette oppdraget så skal det settes igang en *aksjon* når vi trykker på knapp A på micro:bit. For å få til det bruker vi blokken "*når knapp A trykkes*". Det vi vil skal skje når knappen trykkes, legger vi inn i "*gapet*" til blokken.

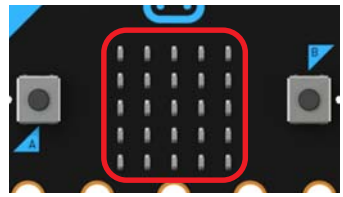
Blokken "*når knapp A trykkes*" finnes under fanen: "*Inndata*".

Løsningsforslag A.4, side 62.

6.5 Oppdrag 5 – Symboler for fotgjengerne

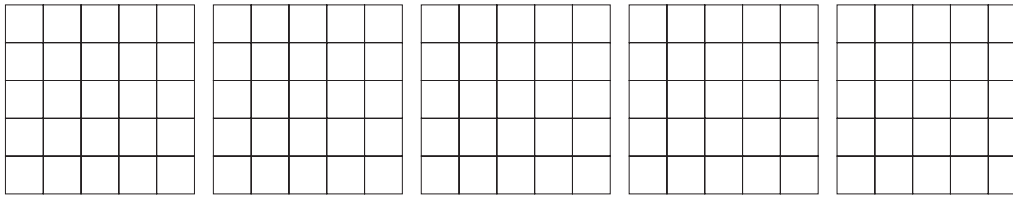
Lysdiodene vil normalt brukes overfor biltrafikk. For fotgjengeroverganger så brukes gjerne symboler som vist på bildet til høyre.

Micro:bit gir oss mulighet til å lage enkle symboler, 5 x 5 bildepunkter i displayet på micro:bit (se bildet lengst til høyre).

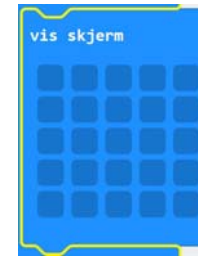

5 x 5 bildepunkter

> **Oppdrag 5a** *Tegn symboler som illustrerer ventende og gående fotgjenger. Tenk gjerne unisex. Symbolene skal være innenfor 5 x 5 bildepunkter. Lag et program som viser de to symbolene på displayet til micro:bit. La hvert symbol vises i 5 sekunder.*

1. Du kan bruke rutemønstrene under til å lage dine forslag på papir:



2. Dernest kan du bruke blokken vist til høyre for å legge inn de to symbolene. Trykk på rutene slik at de blir markert og tegn symbolene. Disse blokkene finner du under meny-fanen *Basis*.



Kombiner programmene

> **Oppdrag 5b** *Inkluder symbolene i programmet i oppdrag 4 slik at de skifter på riktig sted i sekvensen. Lysdiodene rød, gul og grønn skal virke som før.*

Dersom du tenker at lysdiodene styrer biltrafikken og symbolene på displayet styrer fotgjengerne, må du kanskje endre på sekvensen?

Løsningsforslag A.5, side 63.

Lagre filen med navnet Trafikklys-5.

6.6 Oppdrag 6 – Lag lyd

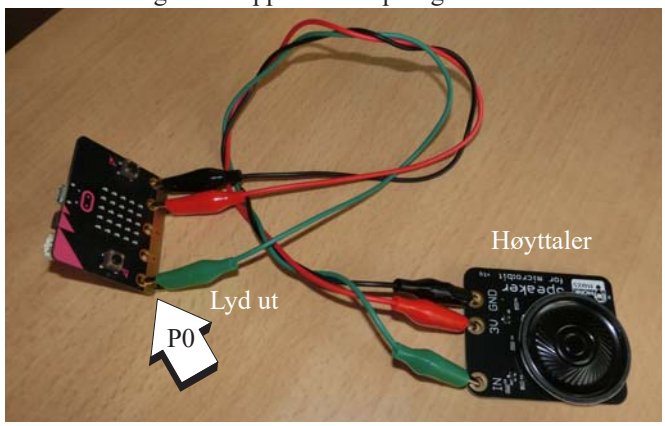
Mange trafikklys har også lyd for synshemmede slik at de kan bruke hørselen for å vite når det er grønt for gående. Det skal vi også gjøre for vårt trafikklys.

Det følger med en liten lydenhet (ABI-050). Siden den gir svært svak lyd har vi i tillegg en høyttaler som kan gi oss en litt kraftigere lyd om vi trenger det.

Legg merke til + tegnet på siden av lydenheten.

6.6.1 Oppkobling

Innledningsvis bruker vi krokodilleledninger når vi skal eksperimentere med lyder. Da kan vi dra micro:bit ut av kantkontakten og koble opp som vist på figuren under.



Før vi setter lyd til trafikklyset må vi lære hvordan vi bruker micro:bit for å lage lyd.

6.6.2 Programmering

Mens dere eksperimenterer med lyder er det lurt å ha oppdraget klart i tankene:

Oppdrag 6 *Test ulike lyder med tanke på å gi lyd til trafikklyset som passer til situasjonen.*

Vi tenker oss at dere skal lage lyder som passer til grønt og rødt lys for synshemmede fotgjengere. Lyden skal være slik at den ikke kan misforståes.

Diskuter gjerne med barna hvilke lyder de kan tenke seg.

- *Hvordan er lydene som vanligvis brukes i forbindelse med trafikklys? Hvorfor tror dere at de har valgt disse lydene?*
- *Kan en tenke seg andre alternativer?*
- *Tenker alle likt mht. hva som er en "rød lyd" (stopp) eller en "grønn lyd" (gå)?*
- *Bør det være en lyd for gult også, en gjør-deg-klar-lyd?*

Dersom du ikke gjør noen ting så vil lyden komme ut fra P0 (port 0), så derfor er det lurt å bruke den for disse innledende eksperimentene.

1. I *ved start-blokken* kan vi sette lydstyrken som vist i figuren under ved hjelp av *set volume-blokken*. Verdien 0 gir ingen lyd, og verdien 255 gir maksimal lyd. Sett maksimal lyd. Blokkene du trenger finner du i menyfanene *Basis* og *Musikk*. Du trenger også *gjenta for alltid-blokken* som skal fylles med innhold

2. Dersom du tenker svært tradisjonelt og ønsker å lage kortere eller lengre pipelyder, så kan du bruke disse blokkene

Spiller en valgt tone med en bestemt lengde (takt)

Tar en pause med en gitt lengde (takt)

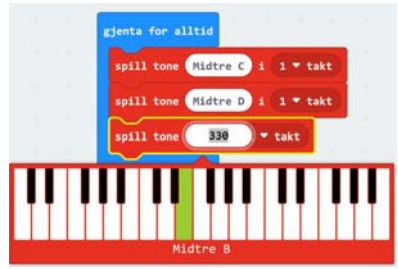
Sett volumet til avspillingen

3. Det er mange måter å lage lyd på. Her er noen flere blokker om du ønsker å lage en liten melodi:

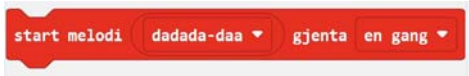
Spiller en sekvens av toner fra en meny

Sett tempoet til en avspilling

Velg en tone fra et tangentbordet



Sett opp en selvvalgt sekvens av toner med bestemte lengder (takt)



Spiller en forprogrammert melodi, valgt fra en meny

For nærmere forklaring av de ulike kommandoene se:
https://makecode.microbit.org/reference/music

Så er det bare å eksperimentere med ulike toner og melodier som kan brukes til “Grønn -” og “Rød fotgjenger”

4. Ta vare på ulike versjoner av programmet så du kan bruke dem senere i prosjektet. Du kan jo velge navn som *Trafikklys-6-gronn-tone og Trafikklys-6-rod-tone*.

Kanskje dere kan teste lydene overfor kollegaer (klassekammerater) og be dem fortelle om lyden indikerer *stopp* eller *gå*, eller kanskje *gjør-deg-klar*?

Et eksempel finnes i vedlegg A.6, side 64.

6.7 Oppdrag 7 – Blinkende grønt lys

Oppdrag 7: *Lag blinkende grønt lys mot slutten av grønn periode for fotgjengere*

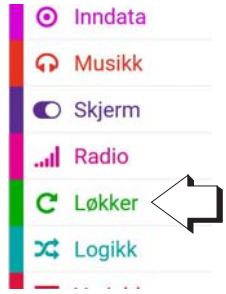
Vi vet at mot slutten av den grønne perioden begynner det grønne lyset for fotgjengere å blinke for å varsle fra om at nå er det ikke lenge igjen før det blir rødt. I USA og andre land er det heller ikke uvanlig at trafikklys viser gjenværende sekunder før de skifter til rødt ev. til grønt.

Vi tenker oss følgende spesifikasjon av oppdraget:

- Vanligvis er det rødt lys
- Når vi trykker knapp A så blir det grønt lys for fotgjengerne
- Det grønne lyset lyser uten å blinke i 5 sekunder
- Deretter blinker det tre ganger før det skifter til rødt (via gult), hvert blink varer i 1 sekund

6.7.1 "Gjenta-blokken"

Når vi ønsker å gjøre den samme tingen flere ganger bruker vi ofte en *"gjenta blokk"*, vi lager en *løkke*. Dette er for å forenkle koden vår slik at den skal ta mindre plass og bli mer effektiv. I dette tilfellet ønsker vi å blinke tre ganger med det grønne lyset før det skifter til gult og deretter rødt. Figuren under viser "gjenta blokken". Det som skal gjentas legges inn i gapet til blokken. Tallet på toppen av blokken viser hvor mange ganger vi vil at det i gapet skal gjentas før vi går videre i programmet.



Bruk det dere lærte i oppdrag 5 til å løse dette oppdraget.

Tips:

- Prøv gjerne ut "gjenta-blokken" alene med blinkende grønt før dere setter den inn i det store programmet.

Løsningsforslag finnes i vedlegg A.7, side 65.

Lagre programmet under navnet Trafikklys-7

6.8 Oppdrag 8 – Kombiner lys og lyd

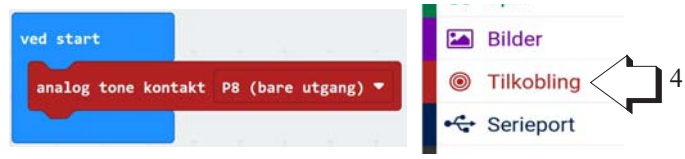
I dette oppdraget skal vi kombinere lys og lyd. *For at det ikke skal bli for vanskelig så lager vi et lite program som bare viser grønt (5 sek) og blinkende grønt (5 x 1 sekund) med lyd.*

Oppdrag 8 *Legge på passende lyd for synshemmede til det grønne lyset.*

Her er det opp til hver enkelt hvordan man vil lydsette lysene slik at det skal være tydelig for synshemmede hvilket lys de hører ved fotgjengerfeltet.

Tips:

- Normalt vil lyden komme ut av port 0 (P0). Siden vi bruker P0 til det røde lyset, må vi bruke en kommando som flytter lyden til en annen port f.eks P8. Dette gjør vi med *blokken analog tone kontakt P8 (bare utgang)* som vi finner i menyen *Tilkobling*.



  Årsaken til at P8 kan være et godt valg er at denne porten ikke brukes til noe annet (se tabellen i avsnitt 3.1, side 28).

- Du kan slå på lyset samtidig som du starter en tone med kommando-blokkene vist i figuren til høyre. I dette tilfellet bestemmes lengden til lydsignalet av verdien på *takt*.

Ekstra oppdrag

Oppdrag 8a: *Endre programmet slik at det høres et annet signal når lyset er kontinuerlig grønt. F.eks. kan man ha lange toner ved kontinuerlig grønt og korte lydstøt når det blinker grønt.*

6.9 Oppdrag 9 – Gi trafikklyset lyd

Oppdrag 9 *Sett sammen programmene så langt slik at du får et trafikklys med følgende egenskaper: Normalt skal det være rødt samtidig som det vises et symbol for fotgjengere. Når man trykker på knapp A skal trafikklyset skifte fra rødt til grønt samtidig som symbolet for fotgjengere viser at* ***nå*** *kan man gå. Etter 5 sekunder skal det grønne lyset begynne å blinke samtidig som det høres en lyd som signaliserer at det er kort tid igjen før lyset skifter til rødt.*

Tips:

- Det kan være lurt å ha flere vinduer med Makecode åpne samtidig slik at det er lett å flytte over kode. ***NB!*** *Du kan ikke elektronisk flytte en kode fra en Makecode til en annen, du må skrive av koden.* Det kan dere gjøre ved å åpne flere vinduer med Makekode (micro:bit-programmet) i nettleseren.
- Om dere synes det er unaturlig å vise grønt lys samtidig som trafikklyset viser symboler for fotgjengerne, så kan dere tenke dere at LED-lysene er for trafikantene, mens symbolene er for de gående. Hvordan vil dere da skifte om på programstrukturen?

Løsningsforslag finnes i vedlegg A.9, side 67.

Lagre det endelige programmet under navnet Trafikklys-9.

6.10 Oppdrag 10 – Blinkende gult lys

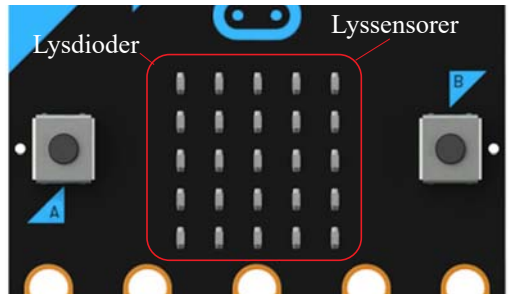
De fleste har merket at enkelte lyskryss får blinkende gult i alle retninger på kvelden og i helgene. Dette er for at trafikken skal gli smidigere når det likevel er lite trafikk. Denne gangen er oppdraget vårt:

> **Oppdrag 10** *Lag et program som skifter til blinkende gult lys når mørket faller på. Man kan anta at lyset i utgangspunktet er rødt. Hvert blink skal være på i 1 sekund og av i 1 sekund.*

Tips:

- Det er lurt å lage et lite program først som bare tester ut skifte fra rødt til blinkende gult når det blir mørkt.

6.10.1 Lyssensoren



Displayet på micro:bit består av 25 lysdioder som er ordnet i 5 x 5 rader og kolonner. Lysdioder kan også fungere som sensorer som registrerer lysstyrken.

> **NB!** ***Når vi bruker dem som lyssensorer kan vi ikke samtidig bruke dem som display.***

For å bruke dem som lyssensor, bruker vi en blokk fra fanen *Inndata* som heter *lysnivå*. Denne blokken holder det målte lysnivået som kan ha verdier fra 0 – 255. Hvor null er mørke og 255 er sterkt lys.

> ***NB! Vær klar over at i vanlig lys på kontoret så vil den målte verdien kunne være lavere enn 25.***

6.10.2 Oppkobling

Det trengs ingen ekstra oppkobling for å løse dette oppdraget.

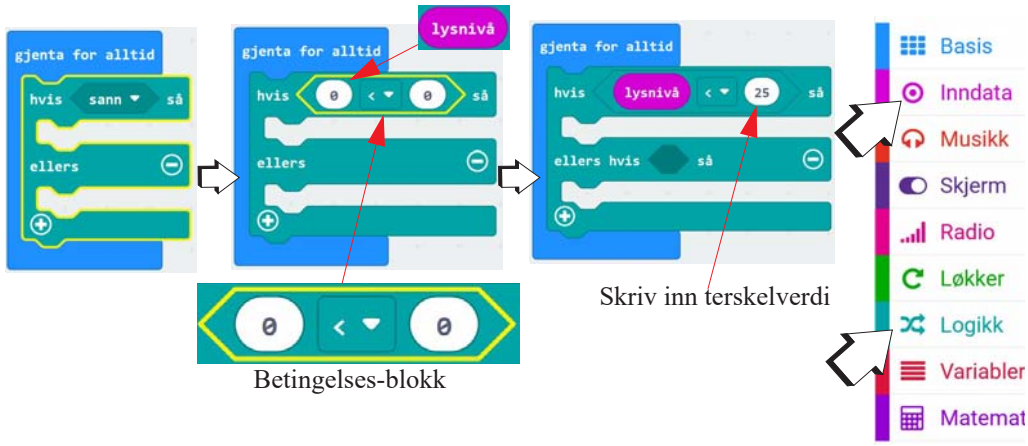
6.10.3 Programmet

La oss sette opp en sekvens for å bygge opp programmet:

1. I utgangspunktet skal lyset være rødt
2. Når lysstyrken i rommet blir lav nok, så slukkes det røde lyset og det gule lyset begynner å blinke. Man kan lage det mørkt ved å holde hånda over lysdiodene.
3. Dersom lysstyrken øker over grenseverdien så blir lyset rødt igjen.

For å løse denne oppgaven må vi bruke en *hvis-eller-blokk*. En slik blokk har en betingelse, dersom denne betingelsen er sann så skal programmet utføre det som vi legger i det øverste gapet (*hvis*), og dersom den er usann skal programmet gjøre det som befinner seg i det nederste gapet (*ellers*).

Ved å trykke på plusstegnet nederst til venstre i hvis-blokken kan vi få opp flere gap. Vi ser da at alternativet *ellers hvis* kommer opp, som også må inneholde en betingelse.



Betingelses-blokk

Skriv inn terskelverdi

Vi henter ut en *betingelses-blokk* (fra *Logikk-fanen*) som kan sammenligne to verdier som vist nederst på figuren over, og legger denne inn i betingelsen i *hvis-blokken* eller *hvis-ellers-blokken*. En slik betingelses-blokk sammenligner to tall. I dette tilfellet om det første tallet er mindre enn det andre tallet (se nederst på figuren over).

Vi ønsker å sammenligne verdien fra lysmålingen som vi finner i *lysnivå-blokken* (se fiolett blokk på figuren over), og sammenligner den målte verdien med en verdi som *vi* velger. Vi kan kalle denne for en *terskelverdi* som er den verdien lysstyrken har når vi vil at trafikklyset skal skifte til blinkende gult. Er den målte verdien *under* denne terskelverdien (la oss sette den til 25), så skal trafikklyset blinke gult, dersom den er større enn denne verdien, så skal det lyse rødt.

- Sett inn de siste blokkene i de to gapene slik at lyset fungerer som ønsket og ...
- ... varier terskelverdien slik at lyset begynner å blinke gult ved et passende lavt lysnivå. Prøv dere fram for å finne en passende terskelverdi.

Test programmet.

Løsningsforslag finnes i vedlegg A.10, side 68.

Lagre programmet under navnet Trafikklys-10.

6.11 Oppdrag 11 – Kombinerer lys, lyd og blinkende gult ved mørke

Denne sammenstillingen kan være ganske krevende og kan droppes eller man kan vente med den til slutt. Vi definerer følgende oppdrag:

> **Oppdrag 11** *Vi skal lage et trafikklys som normalt lyser rødt for fotgjengere, dvs. grønt for biltrafikk (vi skal lage fotgjenger-delen). Når fotgjengeren trykker på knappen skal lyset skifte til grønt, som etter 5 sekunder skal gå over til blinkende grønt med lyd, før det går tilbake til rødt igjen. Ved mørkets frambrudd skal lyset begynne å blinke gult. Dersom noen ønsker å gå over gata så går det fortsatt an å bestille grønt, selv om det blinker gult. Det gule går da over til rødt for så å skifte til gult og så til grønt, for deretter å gå tilbake til rødt og etter et par sekunder skifte til blinkende gult igjen.*

Tips:

- Siden vi bruker lysdiodene som lyssensorer i dette oppdraget så må vi droppe symbolene for fotgjengere

Det er lurt å tenke gjennom hvilke forskjellige tilstander vi har med å gjøre i dette oppdraget. Her er et forslag:

1. ***Betingelser:*** Det er lyst og ingen fotgjenger trykker på knapp A
   Aksjon: –> Konstant rødt for fotgjenger
2. ***Betingelser:*** Det er lyst og en fotgjenger trykker på knapp A
   Aksjon: **–>** Lyset skifter til grønt og tilbake (Oppdrag 8)
3. ***Betingelse:*** Det er mørkt og ingen fotgjenger trykker på knapp A
   Aksjon: **–>** Lyset blinker gult (Oppdrag 10)
4. ***Betingelse:*** Det er mørkt og en fotgjenger ønsker grønt
   Aksjon: **–>** Lyset skifter til grønt og tilbake (Oppdrag 8)

Vi legger merke til at aksjonen når en fotgjenger trykker på knappen er den samme uansett om det er mørkt eller lyst. Vi kan derfor slå sammen tilstand 3 og 4 og sitter igjen med tilstandene 1, 2 og 3.

I figuren under har vi antydet hvordan vi kan bruke en *hvis-blokk* til å løse oppdraget. Legg også merke til at koden under hver betingelse representeres ved t*re blå blokker* med et beskrivende navn: *kjør FotgjengerGrønt*, *kjør BlinkendeGult* og *kjør KonstantRød.* Dette er det vi kaller *funksjoner* som samler kode under et felles navn.

Gå gjennom betingelsene og vær sikker på at dere forstår hva de betyr.

Betingelse: Fotgjenger trykker knappen og ber om grønt lys

Aksjon: Grønt for fotgjenger (Oppdrag 8)

Betingelse: Det er mørkt og ingen fotgjenger ber om grønt lys

Aksjon: Blinkende gult (Oppdrag 10)

Betingelse: Det er lyst og ingen fotgjenger ber om grønt lys

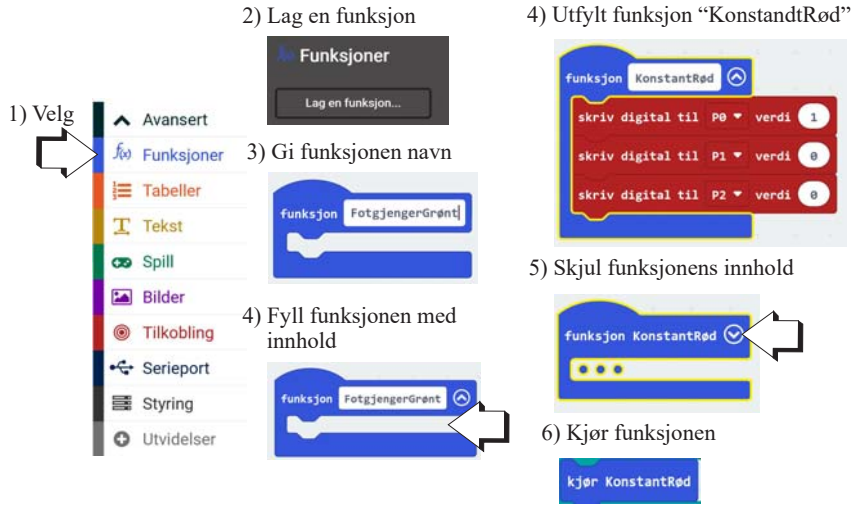
Aksjon: Konstant rødt

Bruk fargene på blokkene til å hjelpe dere å finne dem i menyfanene.

6.11.1 Funksjoner

La oss se hvordan vi kan samle kode i funksjoner for at det skal være lettere å få oversikt:

1. Velg *Funksjoner* fra menyen. Se figuren under for illustrasjon av hvert trinn.
2. Velg *Lag en funksjon*
3. Skriv inn et passende navn som beskriver det funksjonen skal gjøre
4. Fyll inn gapet i funksjonen med den koden du vil at funksjonen skal utføre
5. Lukk funksjonen med haken øverst i høyre hjørne slik at innholdet skjules
6. Velg *Funksjoner* fra menyen og hent den funksjonen du vil bruke dvs. *kjør <funksjonsnavnet>* og legg den inn i programmet der du vil at den skal utføres.



Nå kan man legge inn koden fra Oppdrag 8 under funksjonen *FotgjengerGrønt* og Oppdrag 10 under funksjonen *BlinkendeGult.*

Test programmet og se om det fungerer som ønsket.

Løsningsforslag finnes i vedlegg A.11, side 69.

Lagre programmet under navnet Trafikklys-11.

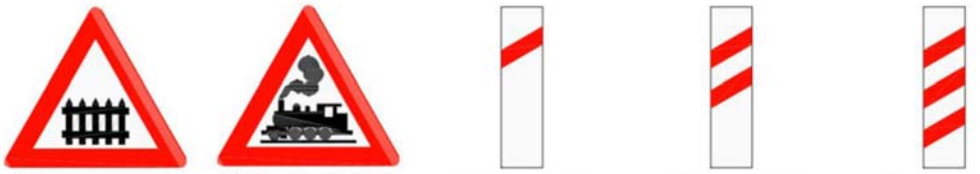
6.12 Oppdrag 12 – Planovergang

Det nye i dette oppdraget er å vise hvordan vi kan styre en mekanisk bevegelse ved hjelp av en servo.

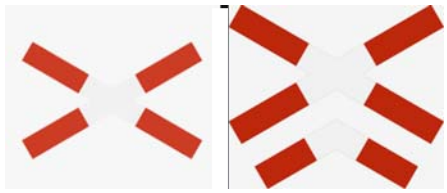
> **Oppdrag 12** *Vi kommer til en planovergang med kryssende tog. Denne planovergangen er forsynt med lys og bom. Vanligvis er lyset hvitt (grønt) når det ikke kommer tog.*
>
> *Varsel om at toget kommer kan gjøres ved å trykke på knapp B på micro:bit'en.*
>
> *10 sekunder før toget kommer skal lyset skifte fra grønt til rødt. 5 sekunder før toget kommer skal bommen gå ned. Toget bruker 3 sekunder til å passere planovergangen. 2 sekunder etter at toget har passert skal bommen gå opp og 2 sekunder etter at bommen har gått opp skal lyset skifte til grønt.*

I dette siste oppdraget skal vi lære å bruke en *servo* som kan heve og senke bommen over veien.

Kjenner dere betydningen av disse skiltene?



Eller hva betyr disse?



Spør barna:

- *Hva betyr skiltene og hvor har de ev. sett dem før?*
- *Hvilke lys kan du møte ved en planovergang?*

6.12.1 Servo

Det finnes flere typer servomotorer. Det er de som kun dreier en vinkel fra 0 – 180°, f.eks. SG90 som vi skal bruke her, og de som oppfører seg som en vanlig motor og går hele runden rundt. Disse kalles *360° servoer* eller *kontinuerlige servoer* (f.eks. FS90R) og kan f.eks. brukes til å lage biler og roboter som beveger seg. Det spesielle med servoer er at dreievinkelen eller hastigheten kan styres ganske nøyaktig fra programmet.

0 – 180°

SG90

360° – Kontinuerlig

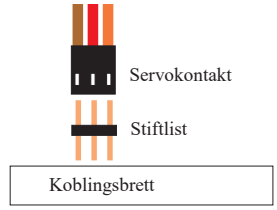
FS90R

Spør barna:

- *Hvordan beveger en bom seg ved en planovergang og hvor mange grader dreier bommen seg vanligvis fra lukket til åpen stilling?*

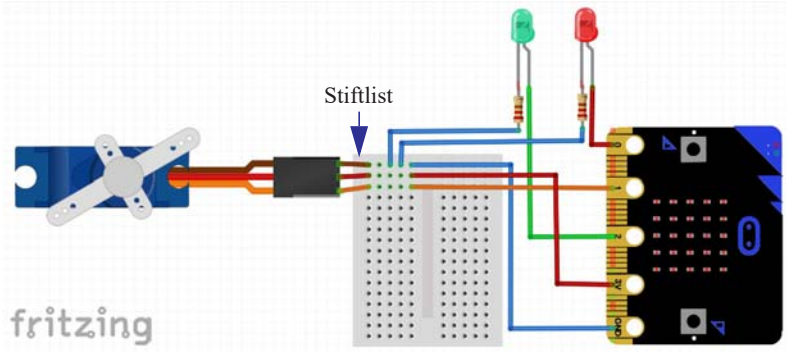
6.12.2 Oppkobling

Før vi lager programmet må vi koble til servoen. Denne har normalt tre ledninger som henger sammen (flatkabel). Vi bruker en liten *stiftlist* slik at det er mulig å stikke servokontakten ned i koblingsbrettet:

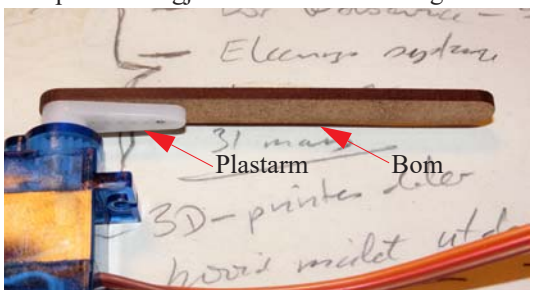


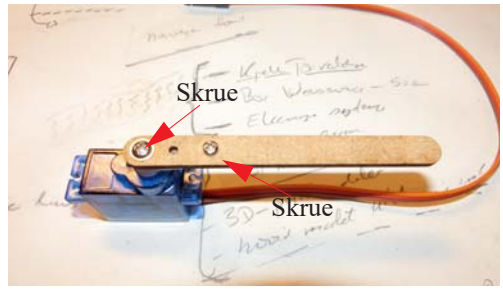
- Brun – Minus (jord)
- Rød – Pluss (+ 3,3 V)
- Oransje – Styresignal

Oppkoblingen er vist på figuren under. Her bruker vi kun rød og grønn lysdiode (egentlig burde den grønne ha vært hvit).



I tillegg må vi montere bommen på servoen. Dette gjøres ved hjelp av en liten plastarm, en skrue og en laserkuttet bom som vist på figuren under. Vi kan også bruke en liten avisolert ledningsstump som tres gjennom to av hullene og tvinnes.





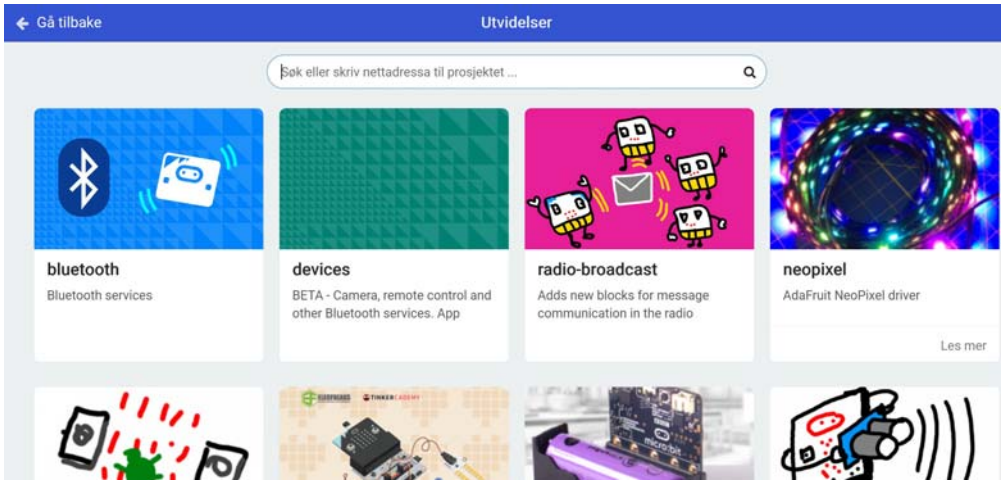
6.12.3 Installasjon av bibliotek

For å gjøre det enklest mulig å styre en servo har noen laget et sett med kommandoblokker spesielt tilpasset bruk av servoer. For å få tilgang til disse kommandoene må vi hente dem og legge dem til de kommandoene vi alt har. Det gjør vi ved å installere et *bibliotek* av kommandoer:

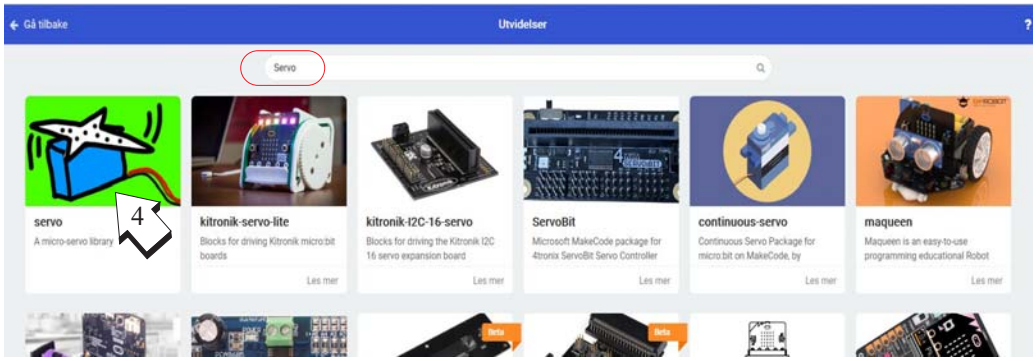
Biblioteket henter vi slik:

1. Velg *Utvidelser* under meny-fanen *Avansert* (se figur til høyre)

2. Man kommer da til en samling biblioteker



3. Skriv *Servo* i søkefeltet og velg søk (lupen). Da vil bl.a. følgende alternativer kommer opp



4. Velg så det første ikonet lengst til venstre merket: *Servo*
5. Vi vil da se at vi har fått en ekstra fane i menyen vår: *Servoer* (se figuren lengst til høyre).

6.12.4 Programmet

La oss nå bruke noen kommandoer fra servo-menyen for å løse oppdraget vårt som tidligere er omtalt kan uttrykkes slik:

> **Oppdrag 12** *Vi kommer til en planovergang med kryssende tog. Denne planovergangen er forsynt med lys og bom. Vanligvis er lyset hvitt (grønt).*
>
> *Vi "varsler" om at tog kommer ved å trykke på knapp B.*
>
> *10 sekunder før toget kommer skal lyset skifte fra grønt til rødt. 5 sekunder før toget kommer skal bommen gå ned. Toget bruker 3 sekunder til å passere planovergangen. 2 sekunder etter at toget har passert skal bommen gå opp og 2 sekunder etter at bommen har gått opp skal lyset skifte til grønt.*

Under denne fanen *Servoer* finner vi følgende kommando:

Figuren viser to kommandoblokker hvor vinkelen er satt til 90° og 0°.

> ***NB! Vi vil oppdage at det er bare er tre porter som kan brukes for å styre servoer. Dette er P0, P1 og P2. Vi velger å bruke P1 og fjerner det gule lyset. Dette er også logisk siden det som oftest kun er to lys ved en planovergang: Hvitt (grønt) og rødt.***

Bygg opp programmet rundt disse kommandoene slik at det tilfredsstiller oppdraget.

Test programmet å se om det oppfører seg som ønsket.

En tilleggsoppgave kan være:

> **Oppdrag 12A** *Løs oppdrag 12 med blinkende grønt lys når det ikke kommer noe tog. Normalt er det "grønne" lyset ved en planovergang blinkende hvitt.*

Løsningsforslag kan finnes i vedlegg A.12, side 71.

Lagre programmet under navnet Trafikklys-12

Vedlegg A Programforslag til oppdragene

A.1 Oppdrag 1 – Lysdiode som blinker

Figuren under viser et løsningsforslag til oppgaven.

A.2 Oppdrag 2 – Trafikklyset – Tre lysdioder som blinker

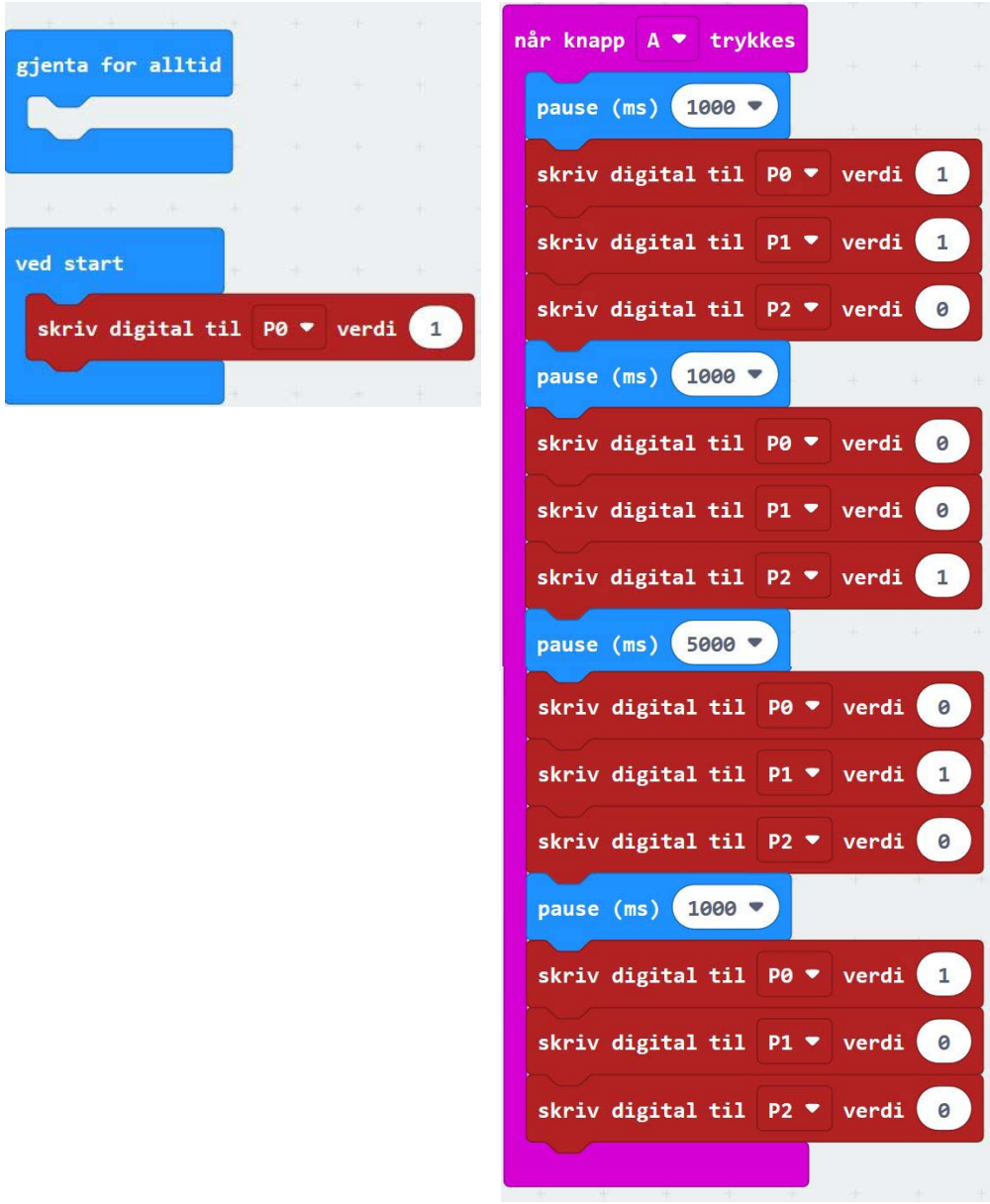
Figuren under viser et løsningsforslag til oppgaven.

A.3 Oppdrag 3 – Trafikklys med riktig sekvens

Figuren under viser et løsningsforslag til oppgaven.

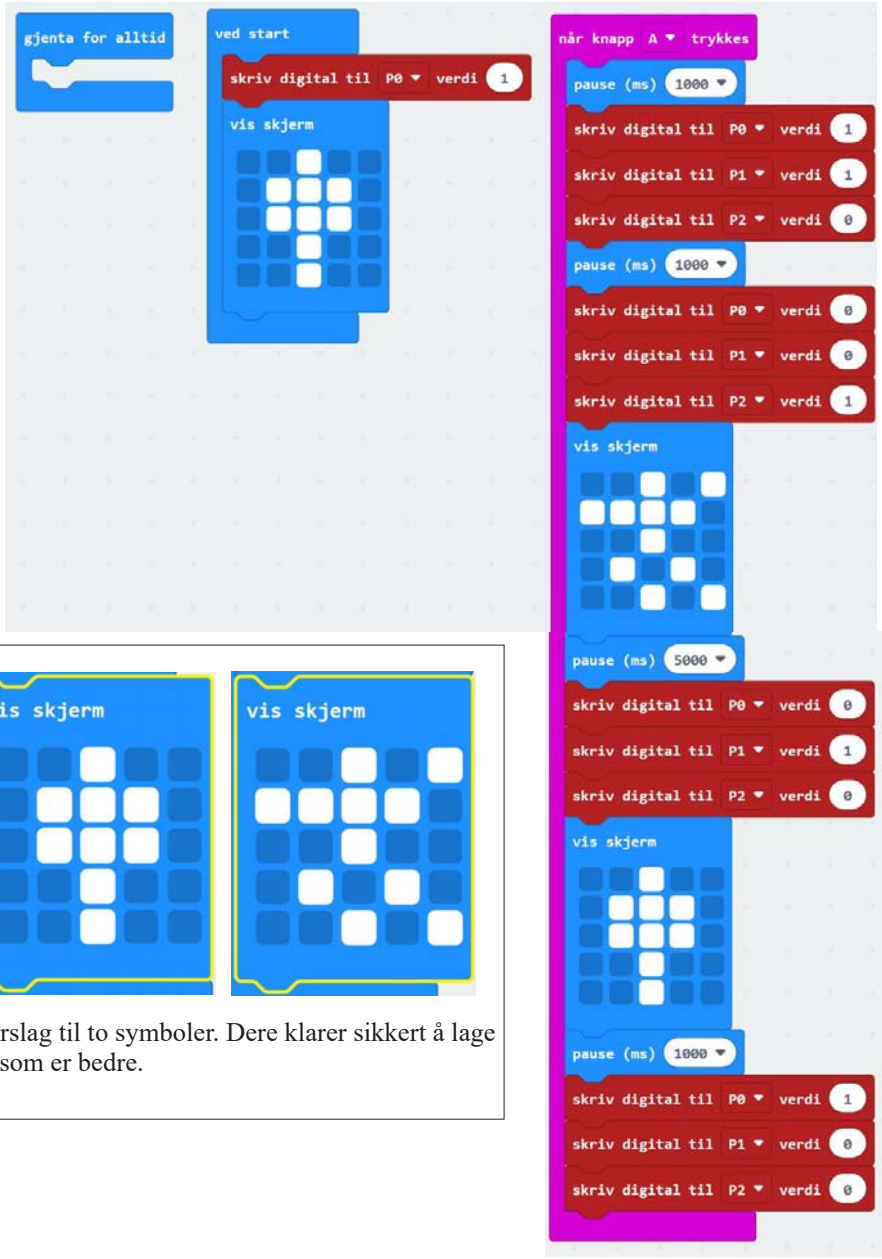
A.4 Oppdrag 4 – Bestill grønt lys ved å trykke knapp A

Figuren under viser et løsningsforslag til oppgaven.



A.5 Oppdrag 5 – Symboler for Vent og Gå for fotgjengere

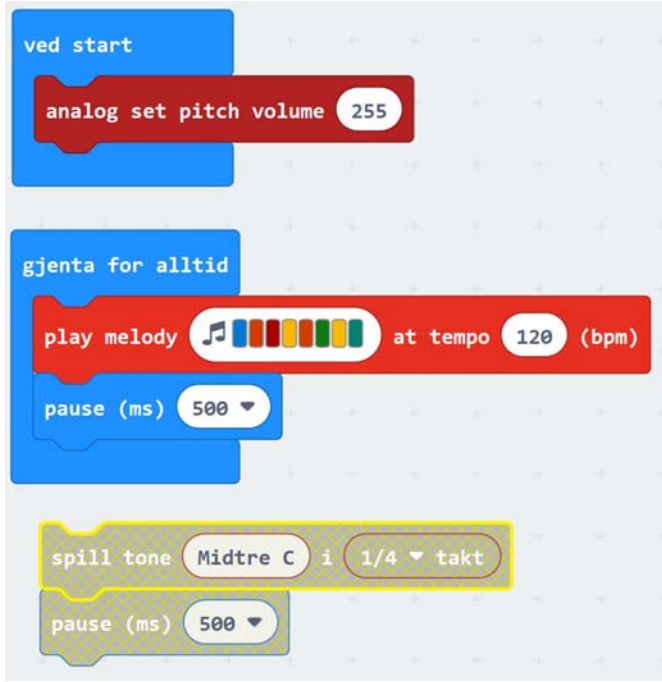
Figuren under viser et løsningsforslag til oppgaven



Forslag til to symboler. Dere klarer sikkert å lage to som er bedre.

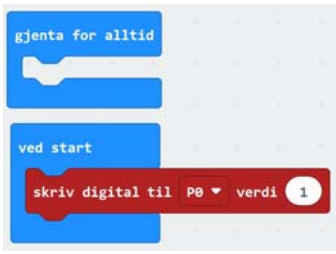
A.6 Oppdrag 6 – Lage lyd

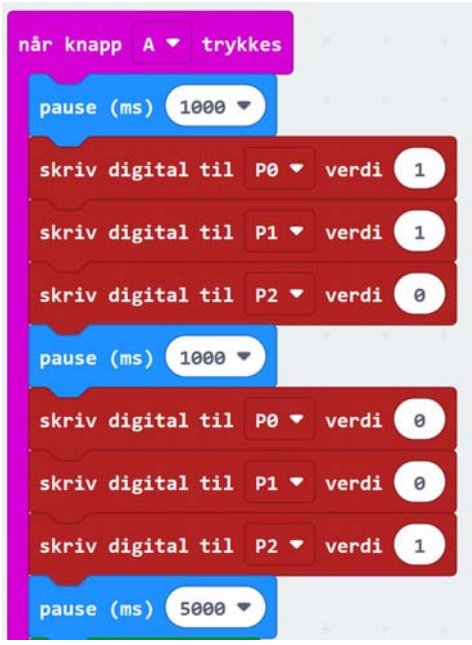
Figuren under viser et par eksempler. Det ene er tatt inn i gapet på *gjenta for alltid*-blokken. Denne kan ev. byttes ut med den som er vist under på skravert form.

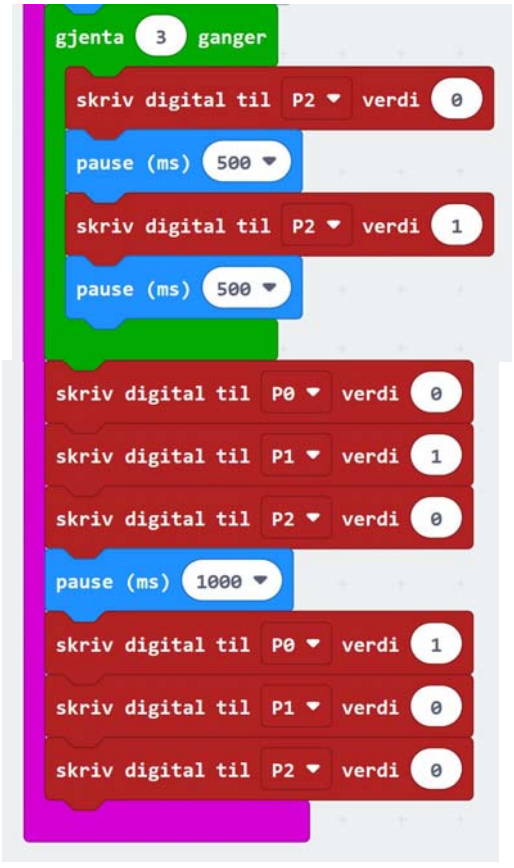


A.7 Oppdrag 7 – Blinkende grønt lys

Dette programmet gir 3 kort blink på slutten av den grønne perioden.

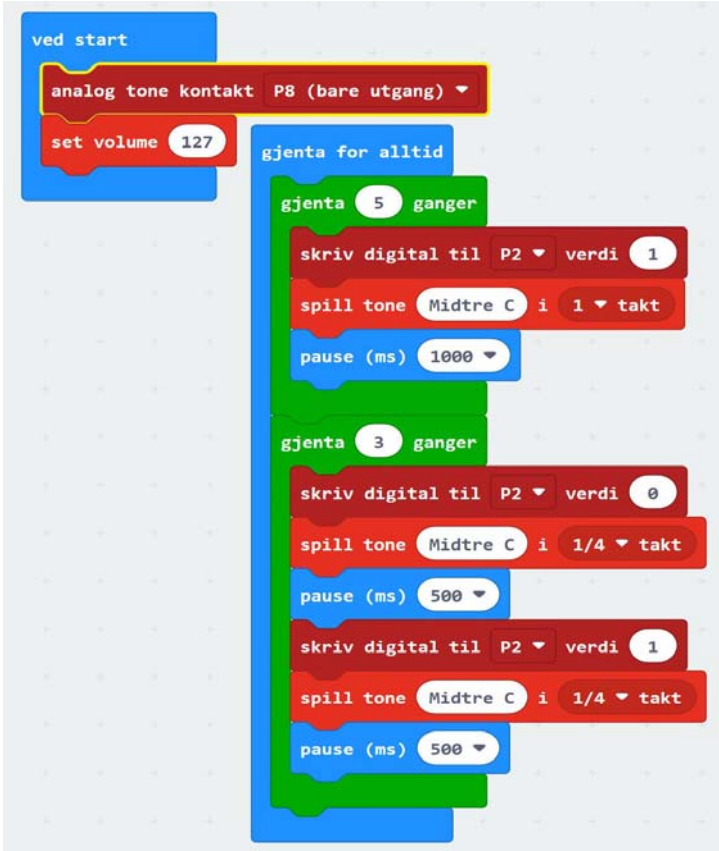






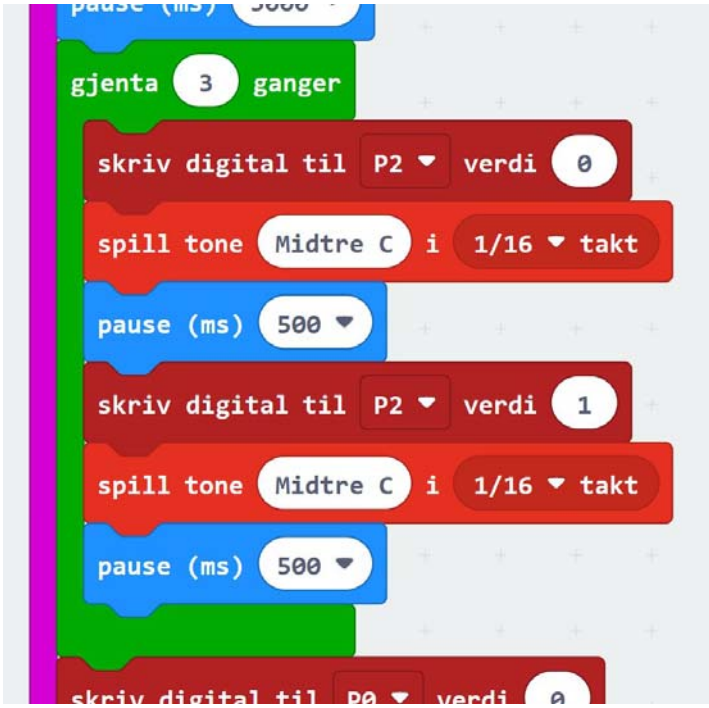
A.8 Oppdrag 8 – Kombiner lys og lyd

Vi lager et lite program som kun viser det grønne lyset og legger inn lyd. Vi har valgt å legge inn lengre toner når lyset er konstant grønt og kortere toner når lyset er blinkende grønt. Dessuten lager programmet to lyder pr. blink.



A.9 Oppdrag 9 – Gi trafikklyset lyd

Her har vi laget et trafikklys med blinkende grønt med lyd. Det gis to lydstøt for hvert blink. Vi har valgt kun å klippe ut den delen av programmet som er forskjellig fra Oppdrag 7.

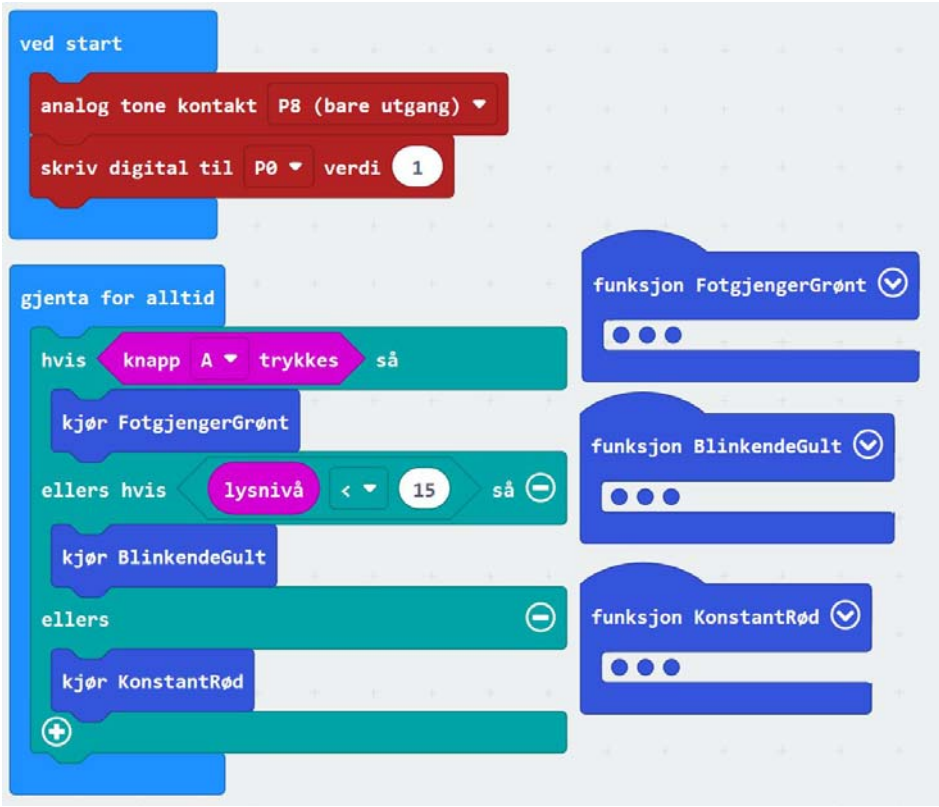


A.10 Oppdrag 10 – Blinkende gult lys

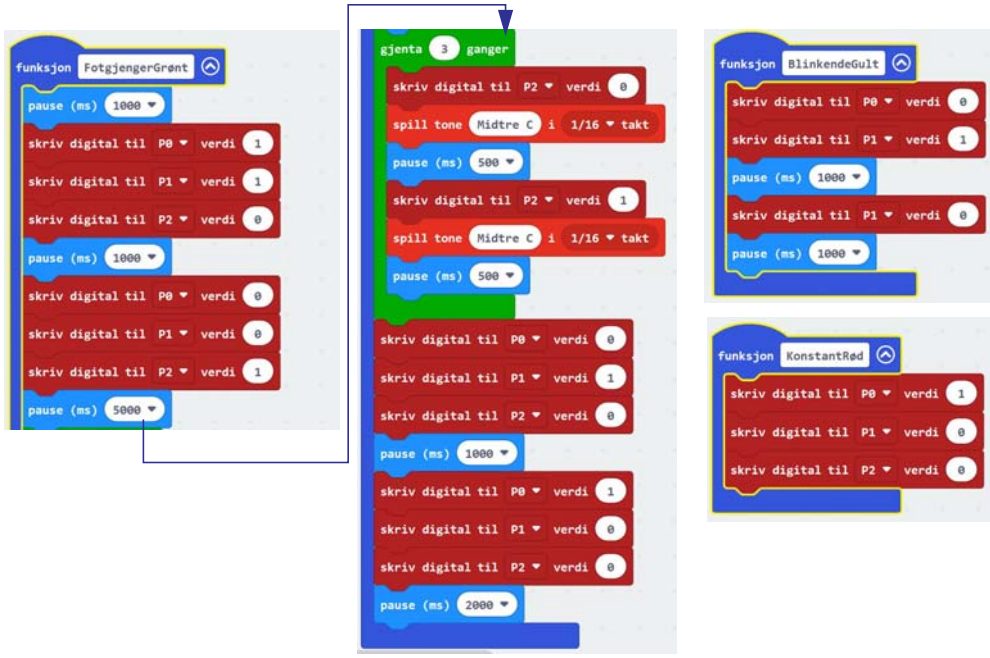
Programmet går fra rødt til blinkende gult ved mørkets frambrudd.

A.11 Oppdrag 11 – Kombinerer lys, lyd og blinkende gult ved mørke

Her kombinerer vi flere programbiter i et større program, hvilket er ganske vanlig. Figuren under viser Hovedprogrammet, mens på neste side er vist innholdet i de tre funksjonene.

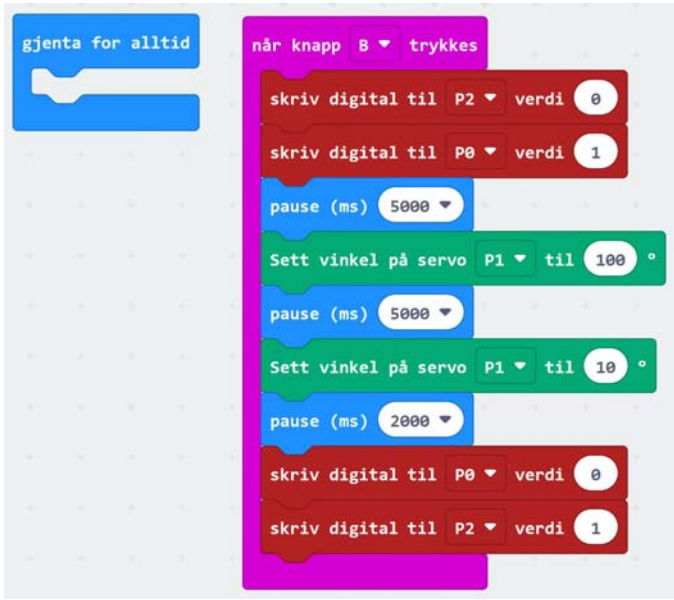


Figuren under viser innholdet i de tre funksjonene: *FotgjengerGrønt*, *BlinkendeGult* og *KonstantRød*.



A.12 Oppdrag 12 – Planovergang

Programmet styrer en planovergang.



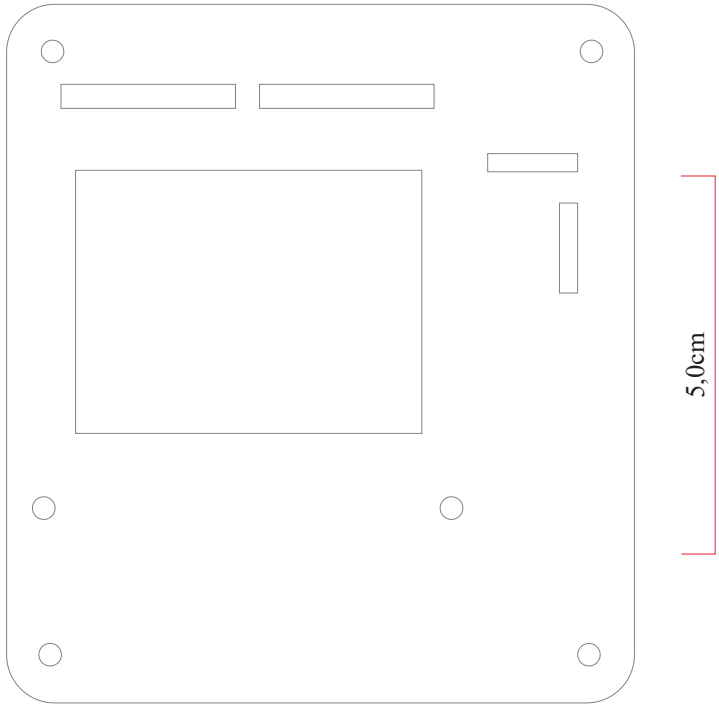
Vedlegg B Komponentlister

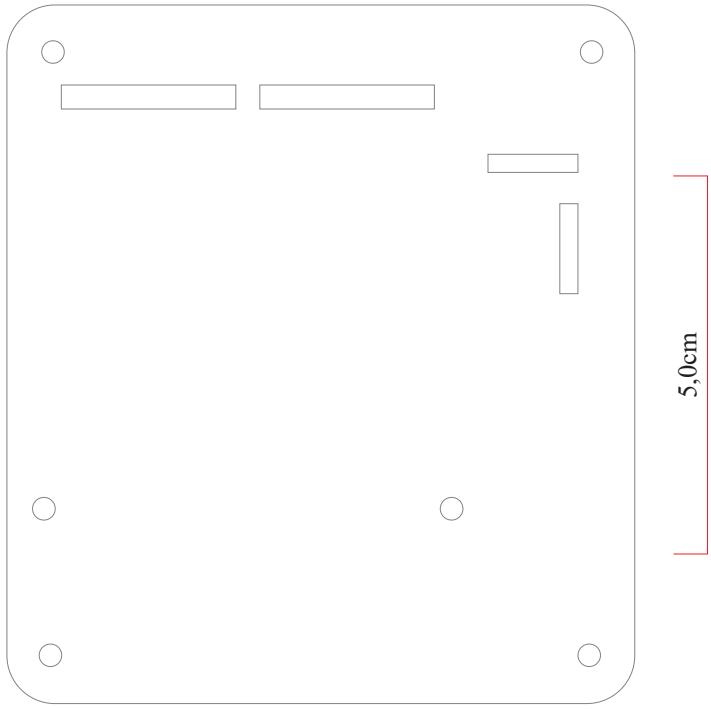
Tabellen under inneholder noen av komponentene som er brukt og hvor de kan skaffes fra:

Typebetegnelse	Type komponent	Leverandør	Nettadresse
Micro:bit	m/kabel	n00b	https://n00b.no/products/bbc-micro-bit-startpakke-inkl-usb-kabel-og-batteri-med-holder
Høyttaler	m/forsterker for micro:bit	n00b	https://n00b.no/products/speaker-for-bbc-micro-bit
Røde, grønne og gule	Lysdioder	Banggood	https://www.banggood.com/500PCS-5MM-LED-Light-White-Yellow-Red-Blue-Green-DIY-Assortment-Diodes-Kit-p-1073370.html
Mini	Koblingsbrett	Banggood	https://www.banggood.com/Mini-Solderless-Prototype-Breadboard-170-Points-For-Arduino-Shield-p-74814.html
Jumpere	Diverse	Banggood	https://www.banggood.com/40pcs-20cm-Male-To-Female-Jumper-Cable-Dupont-Wire-For-Arduino-p-973822.html?rmmds=search&cur_warehouse=CN
Kantkontakt	Kontakt for tilkobling til micro:bit-kortet	Kitronik	https://kitronik.co.uk/products/5601b-edge-connector-breakout-board-for-bbc-microbit-pre-built
USB-kabel	USB A —> micro	Banggood	https://www.banggood.com/BlitzWolf-BW-MC12-Micro-USB-Charging-Data-Cable-1ft0_3m-For-Samsung-S7-S6-Xiaomi-Redmi-Note-5-p-1339804.html
SG90	Servo 180°	Banggood	https://www.banggood.com/6PCS-SG90-Mini-Analog-Gear-Micro-Servo-9g-For-RC-Airplane-Helicopter-p-1078614.html
Buzzer	Passiv (høyttaler)	Banggood	https://www.banggood.com/30pcs-3V-12V-Buzzer-16R-Resistance-AC-Passive-Buzzer-12085-p-1479120.html
Buzzer	Aktiv 3V (gir pipelyd)	Banggood	https://www.banggood.com/50pcs-3V-Active-Buzzer-Electromagnetic-SOT-Plastic-sealed-Tube-Long-Sound-12mmx9_5mm-p-1458641.html
Skruer	M3 16 mm Phillips (200 stk)	ELFA	https://www.elfadistrelec.no/no/skrue-maskin-flathodet-phillips-ph1-m3-16mm-pakke-med-200-stykk-bossard-798503163/p/30071761
Muttere	M3 (100 stk)	ELFA	https://www.elfadistrelec.no/no/sekskantmutre-rustfrie-a2-m3-4mm-rustfritt-stal-bossard-bn-628-m3/p/30071978

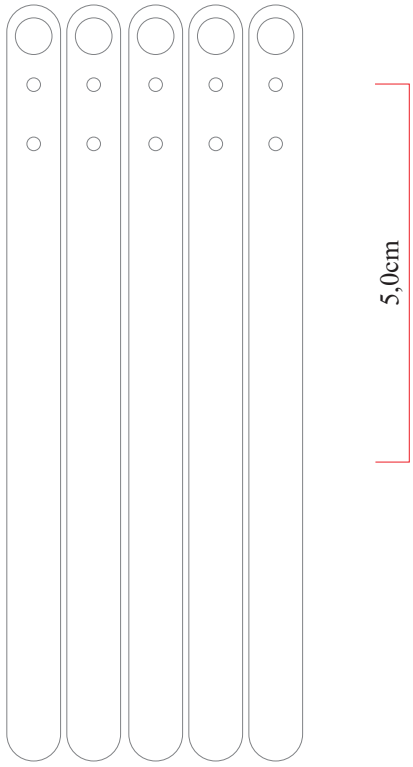
Vedlegg C Mal for laserkutting av målejigg

Mal for laserkutting av trafikklyset. Siden heftet kan være skalert så er en 5 cm lang målestrek inkludert i tegningene. Basen og trafikklyset er skåret i 3,3 mm MDF og bommen i 2 mm MDF. Baseplata er delt i to deler, en topp og en bunn, som skrues sammen med 3 mm skruer. Dette er for at koblingsbrettet skal få god støtte. Ellers er skjæremalene tilgjengelig ved Vitensenteret i Trondheim. Malen til trafikklyset er opprinnelig utviklet i forbindelse med Skaperskoleprosjektet.

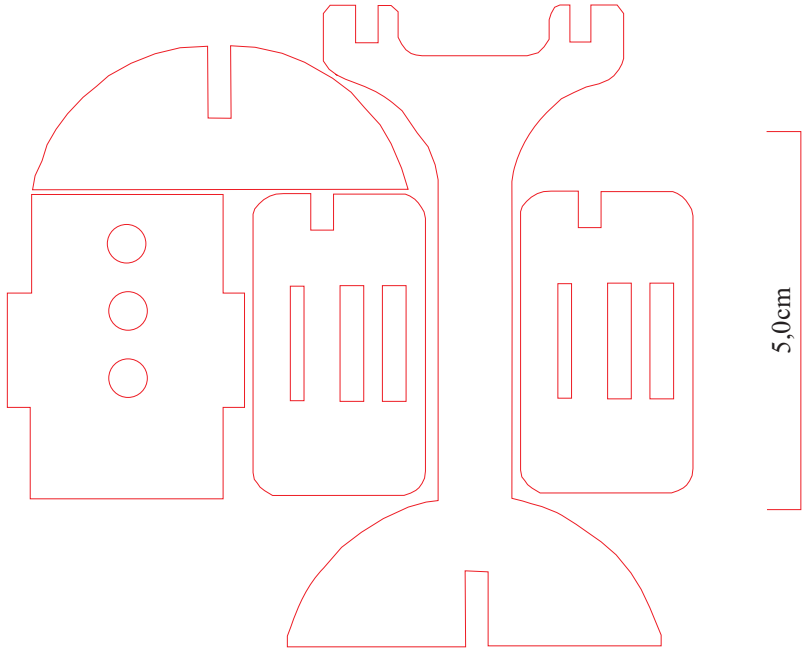



5,0cm

Bommen er skåret i 2 mm MDF



Holder for kretskort for trafikklyset. Trafikklyset er skåret i 3,3 mm MDF. Det ekstra hullet i sidevangene gjør det mulig å sette frontplata i ulik avstand til kretskortet ved bruk av bananstikker (lang avstand) eller jumpere (kort avstand). Kort avstanden gjør at lysdiodene kommer tydeligere fram i frontplaten.



Vedlegg D Bruk av micro:bit og nettbrett

Det er også mulig å bruke nettbrett istedet for PC eller Mac. Videoen på denne nettsiden beskriver hvordan dette lar seg gjøre:

https://www.nrk.no/skole/?page=search&q=super%3Abit+ipad

Her finner du to videoer som forteller hvordan du skal koble mikrobit opp mot iPad og hvilke utfordringer du kan møte:

super:bit - slik løser du vanlige iPad-problemer

super:bit - slik bruker du micro:bit med iPad

Vedlegg E Bruk av veiledning via nett

Makecode programvaren gir læreren mulighet til å veilede elevene over nett. I dette vedlegget beskriver vi hvordan dette kan gjøres.

<Her kommer en beskrivelse – Ola Kleiven?>

Hensikten med heftet er å gi deltakerne en grunnleggende innføring i programmering med micro:bit samtidig som det skal være et eksempel på hvordan programmering inngår i oppbyggingen av produkt. Det er lagt vekt på sentrale begreper fra læreplanen og å vise hvordan man kan lage og teste ut små programbiter som etter hvert kan settes sammen til et større program.

Autentisitet har også vært viktig. Ved å ta trafikklys som eksempel, brukes noe som alle kjenner, men som svært få kjenner virkemåten til. Det er også lagt vekt på å tilrettelegge for tverrfaglighet ved å utfordre lærere og elever til å tenke trafikksikkerhet.

Det er foreløpig noe uklart hvordan bærekraftmål kan løftes fram gjennom eksempelprosjektet. De to mest nærliggende er bærekraftsmål 9: *Innovasjon og infrastruktur* og bærekraftsmål 11: *Bærekraftige byer og samfunn*. Deltakerne oppfordres til å se andre kobler til bærekraftsmålene.

Nils Kr. Rossing (nkr@vitensenteret.com)
Dosent i naturfagdidaktikk ved NTNU og prosjektleder ved Vitensenteret i Trondheim

Ola Kleiven (ola@vitensenteret.com)
Lærer og prosjektleder for Super:bit-prosjektet ved Vitensenteret i Trondheim

Rannvei Sæther (rannvei@vitensenteret.com)
Pedagog ved Vitensenteret i Trondheim

Anne Birgitte Belboe (annebirgitte@vitensenteret.com)
Lærer og skaperlærer ved Vitensenteret i Tr.heim

Eva H. Hagen (eva@vitensenteret.com)
Leder formidleravdelingen Vitensenteret i Tr.heim

ISBN 978-82-92088-66-1
Rev. 1.9 – 01.10.20