



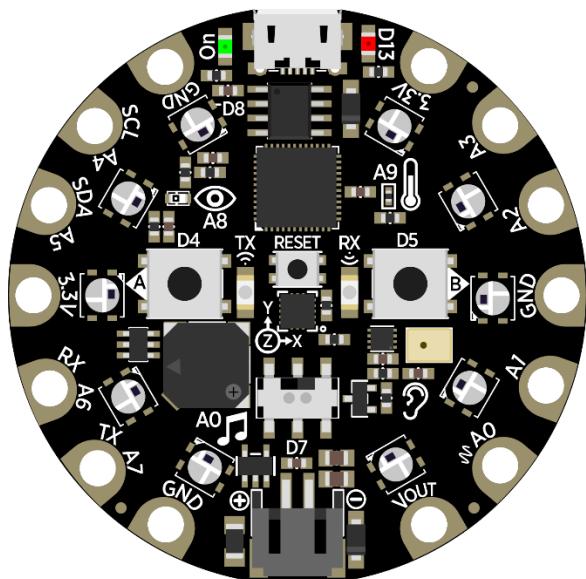
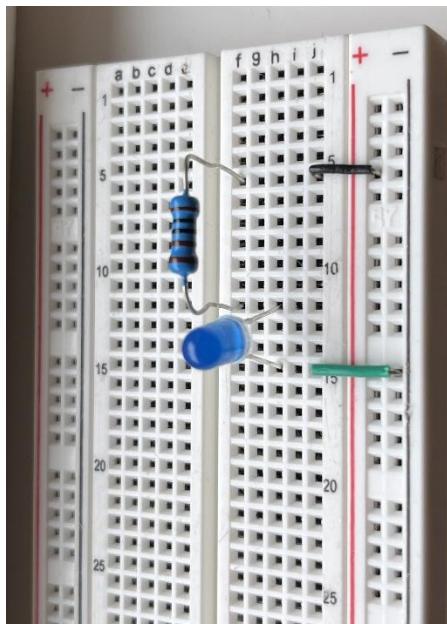
Velkommen til workshop med Akkodis!

Elektronikk- og kodeoppgaver

I dag skal du lære om kretser og transistorer! I tillegg til teori som introduserer deg for disse spennende konseptene, får du også selv bygge kretser som illustrerer hvordan de fungerer i praksis.

I tillegg har vi en ren programmeringsoppgave! Den passer både for deg som aldri har programmert før, og for deg som har litt eller mye erfaring.

Hvis du lurer på noe underveis, er det bare å spørre!



Oppgave 1: En enkel krets som skrur på en lysdiode

I denne oppgaven skal du bygge en enkel krets med en bryter, en motstand, og en lysdiode. Til dette trenger du:

- 1 koblingsbrett
- 2 motstander: $1\text{ k}\Omega$ og $47\text{ k}\Omega$
- 1 lysdiode
- 1 bryter
- Ledninger

Før du går i gang med oppgaven, skal du bli litt kjent med elektriske kretser og de ulike komponentene du skal bruke.

Bli kjent med elektriske kretser

Elektrisk krets

I en lukket elektrisk krets går det strøm. Man lager en lukket krets ved å koble pluss-siden til en strømforsyning (f.eks. et batteri) sammen med minus-siden via motstander og andre komponenter. Hvis pluss og minus ikke er koblet sammen, vil det ikke gå noe strøm. En bryter (slik som en vanlig lysbryter) er en effektiv og enkel måte å åpne en lukket krets, slik at det slutter å gå strøm. Når knappen trykkes inn, lukkes kretsen, og strøm vil begynne å gå gjennom den. Når knappen ikke er trykket inn, er kretsen åpen, og det går ingen strøm gjennom den.



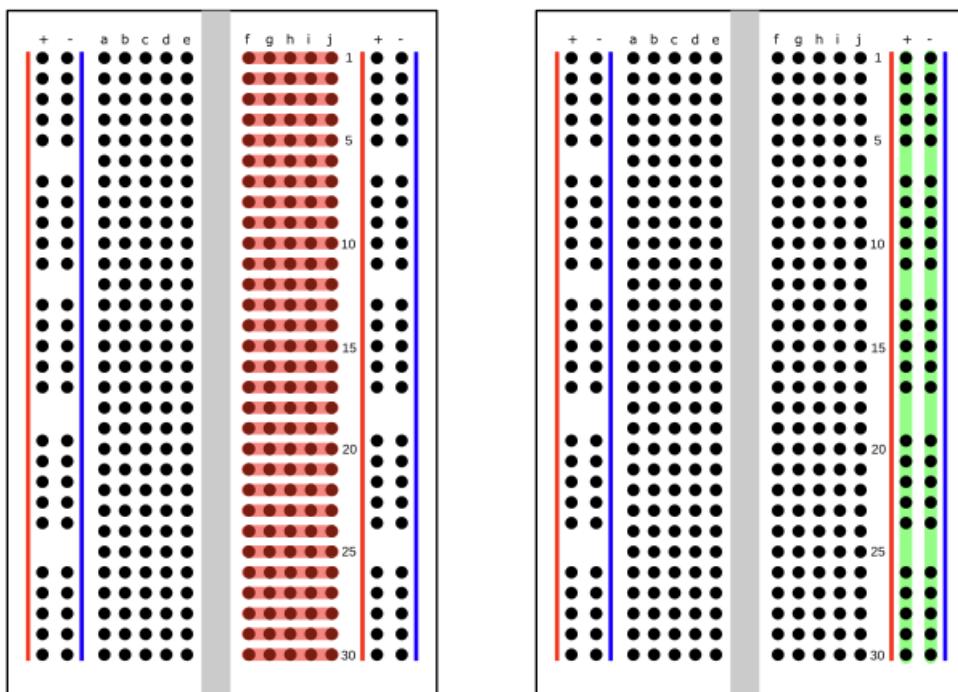
Pluss og minus må **IKKE** kobles sammen direkte, uten at det er koblet noen komponenter imellom, for da vil det gå veldig mye strøm. Dette kalles en kortslutning, og kan være farlig!

Koblingsbrett

Når man skal bygge og teste elektriske kretser, er det vanlig å bruke et koblingsbrett. Et slikt brett kalles også for et breadboard. Under kan du kan se to illustrasjonsbilder av brettet.

Brettet er satt sammen slik at fem og fem hull som ligger ved siden av hverandre på samme rad henger sammen på grunn av metallskinner som ligger inne i brettet (se de røde strekene på bildet ovenfor). Hvis man kobler to ulike elektriske komponenter i to hull som henger sammen, vil

det være det samme som å koble en ledning mellom dem slik at det kan gå strøm mellom dem. På hver side av brettet ligger det også to kolonner med hull, merket med + og -. Disse er også koblet sammen med metallskinner på undersiden (se de grønne strekene på bildet ovenfor). Her skal man koble på strømforsyningen som skal drive kretsen.



Bildene er hentet fra: <https://circuitfever.com/how-to-use-a-breadboard>

Man kobler den **positive spenningen** til den røde kolonnen merket +, og man kobler den **negative spenningen** til den svarte kolonnen merket -. Hullene langs hele kolonnen er koblet sammen slik at man kun trenger å tilføre spenning i ett av hullene, så vil det være samme spenning langs hele kolonnen.

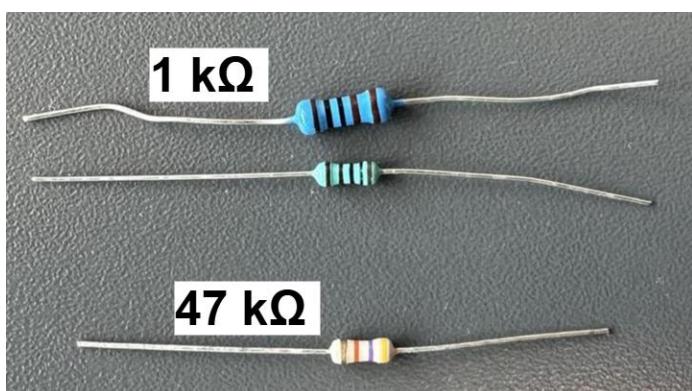
Motstander

Man vil ikke at det skal gå for mye strøm i kretsen, for da kan ting begynne å brenne! Derfor bruker man **motstander** for å begrense hvor mye strøm som går i kretsen. Jo større verdien på motstanden er, jo mindre strøm kan komme igjennom motstanden på én gang.

Bildet illustrerer forholdet mellom spenning, strøm og motstand. Spenningen er kraften som skyver elektrisk strøm gjennom en krets. Et batteri er en vanlig spenningskilde.

Strøm er flyten av elektrisk ladning, og motstand bremser denne flyten. Den røde mannen illustrerer hvordan motstand påvirker strømmen. Jo hardere drar han i tauet, jo større er motstanden.

Bildet under viser hvordan motstandene du skal bruke i dag ser ut.



De blå (eller turkise) motstandene er $1\text{ k}\Omega$, og den beige motstanden er $47\text{ k}\Omega$. Ω utales Ohm, og er enheten man måler størrelsen på motstanden i.

Dioder

Lysene du skal bruke er LED-lys, som du kanskje har hørt om. LED står for Light Emitting Diode, altså en diode som sender ut lys når det går strøm gjennom. På norsk heter det lysdiode.

Dioden fungerer som en enveisbryter for strøm. Det vil si at strømmen går lett gjennom den **ene veien**, men blir veldig begrenset den andre veien. Derfor er det viktig å sette dioden riktig vei. Hvis man setter lysdioden feil vei, vil den ikke lyse.

Som du ser på bildet på siden, har lysdioden to bein. Strømmen går lett gjennom fra det lange beinet til det korte beinet, så du må koble det lange beinet til pluss-siden og det korte beinet til minus-siden av kretsen. I tillegg må



du koble en motstand i serie med dioden, for å begrense hvor mye strøm som går gjennom den. Dioder tåler nemlig ikke så høy strøm. Om det slippes for høy strøm gjennom lysdioden, kan den bli ødelagt.

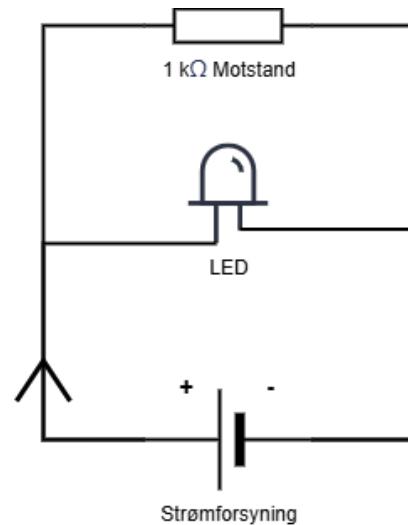
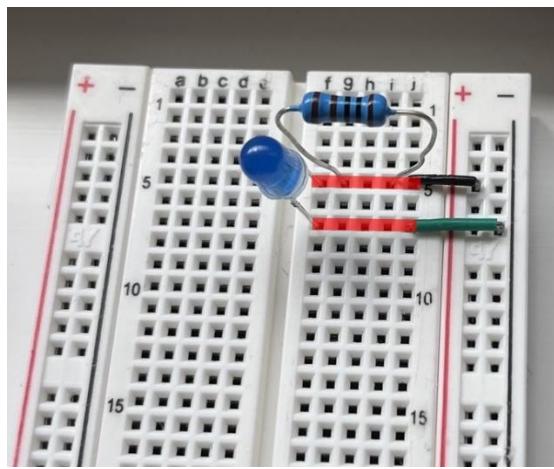
Litt om strøm

Som sagt, vil det gå strøm i en lukket krets, det vil si en krets der pluss-siden på en strømforsyning (for eksempel et batteri) er koblet sammen med minus-siden. Strømmen går fra + til -. I første oppgave kommer **en av veilederne til å koble strømforsyningen til kretsen** du skal koble sammen. Din oppgave når du bygger kretsene nå, er derfor å koble sammen komponentene og ledningene slik at det dannes en vei fra + til - på koblingsbrettet der strømmen kan gå.

Det som er viktig å huske på når du kobler, er at **strømmen alltid velger minste motstands vei**. Det betyr at hvis det finnes flere veier strømmen kan gå, vil mesteparten av strømmen gå gjennom den med lavest motstand. Derfor er det viktig at du tenker på hvor strømmen kommer til å gå mens du kobler.

Under kan du se to litt ulike kretser. I den første er en motstand og en diode koblet i samme rad, som igjen er koblet til +. Siden hele raden på koblingsbrettet er koblet sammen, kan strømmen velge om den vil gå gjennom motstanden. Fordi strømmen velger minste motstands vei, vil den hoppe over motstanden og bare gå gjennom dioden for å komme til minus. Dette kan skade dioden, som trenger en motstand for å begrense strømmen. Kretsen må derfor endres litt på.

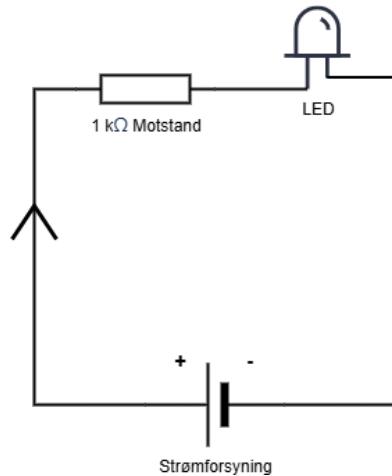
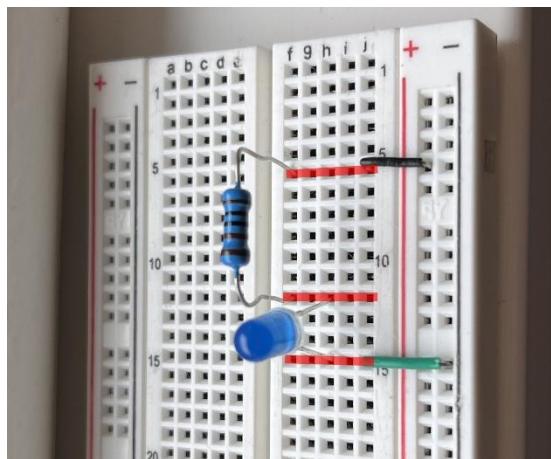
1)



Se nå på den andre kretsen. Her er det ene beinet til motstanden koblet til samme rad som det ene beinet til lysdioden. Det andre beinet til motstanden er koblet til en annen rad som også er koblet til +. Nå må

strømmen gå gjennom motstanden før å nå dioden, og man unngår skade på komponentene.

2)



Når strømmen kan ta flere ulike veier gjennom kretsen, slik som i den første kretsen, kalles det en parallelkobling. Mens når komponentene står etter hverandre slik som i den andre kretsen, slik at det bare fins én vei strømmen kan gå gjennom kretsen, kalles det en seriekobling. Som du skal få se i oppgavene seinere, kan både parallelkobling og seriekobling være veldig nyttig, men de brukes i ulike kretser.



Du skal bruke en Circuit Python Express (CPX) som strømforsyning til kretsen. CPX-en får strøm fra PC-en din via en USB-kabel. Første gang CPX-en kobles til koblingsbrettet kommer en veileder til å hjelpe deg. Etter det må du alltid trekke ut USB-kabelen fra PC-en før du gjør noen endringer på koblingsbrettet, slik at det ikke går noe strøm i kretsen! Du leder også strøm, og hvis du er uheldig kan du lage en lukket krets med deg i midten, og få strøm gjennom deg. Det kan være farlig!

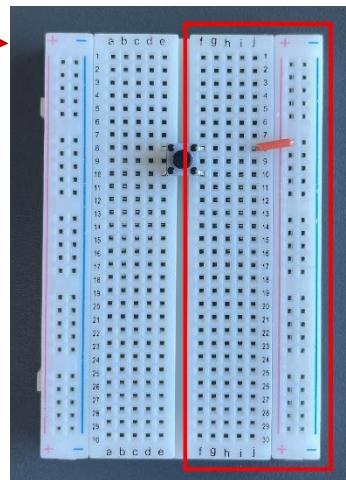
Oppgavebeskrivelse

Du skal nå koble sammen en krets med en bryter, en motstand og en lysdiode (fargen har ikke noe å si). Strømmen skal gå fra pluss-kolonnen på breadboardet ditt via bryteren, motstanden og lysdioden (og eventuelt en ledning), og så til slutt til minus-kolonnen. Vi har startet litt for deg, så bryteren er allerede koblet til pluss-kolonnen.

Kretsen du starter med ser slik ut →

a) Din oppgave er å fullføre kretsen. Du skal bruke motstanden på $1\text{ k}\Omega$ (den som er blå/turkis), bøy beina slik at motstanden passer der du vil plassere den, men vær oppmerksom på at beina er litt skjøre. **Du skal bare koble på høyre side av midten på koblingsbrettet.**

Når du har koblet opp alt, kan du si ifra til en av oss, så skal vi hjelpe deg å koble strøm til brettet ditt.



Nå skal lysdioden lyse når du trykker ned bryteren. Da blir kretsen lukket. Når du slipper bryteren igjen, er kretsen åpen, og det vil ikke gå noe strøm gjennom dioden.

b) Trekk ut USB-kabelen. Nå kan du prøve å bytte ut motstanden på $1\text{ k}\Omega$ med den mye større motstanden på $47\text{ k}\Omega$ (den som er beige). Koble så USB-kabelen inn igjen. Trykk ned bryteren.



Hva observerer du? Kan du tenke deg hvorfor?

c) Trekk ut USB-kabelen. Behold den store motstanden på $47\text{ k}\Omega$, men prøv nå å snu dioden så den står feil vei. Koble så USB-kabelen inn igjen. Trykk ned bryteren.



Hva observerer du? Kan du tenke deg hvorfor?

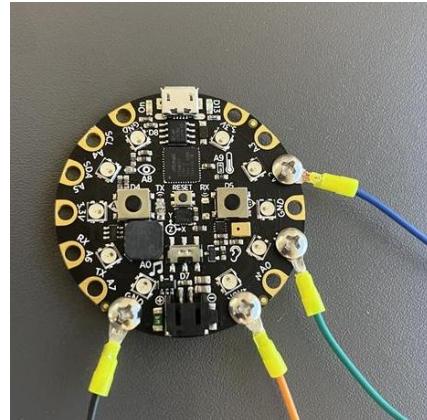
Oppgave 2: Logiske porter

I denne oppgaven skal du lære hvordan AND- og OR-porter fungerer. Du skal bygge to ulike kretser med transistorer, som illustrerer virkemåten til henholdsvis AND- og OR-porten.

Til denne oppgaven trenger du:

- 1 koblingsbrett
- 4 transistorer
- 2 motstander: $1\text{ k}\Omega$
- 2 lysdioder
- Ledninger

Du skal i de neste oppgavene bruke knappene på CPX-en som styringssignalene til logiske porter. Nedenfor ser du et bilde av CPX-en. Den har to knapper merket med A og B, og en mindre knapp i midten som er merket med RESET. Du skal bruke knappene A (koblet med den grønne/gule ledningen) og B (koblet med den blå/lilla ledningen).



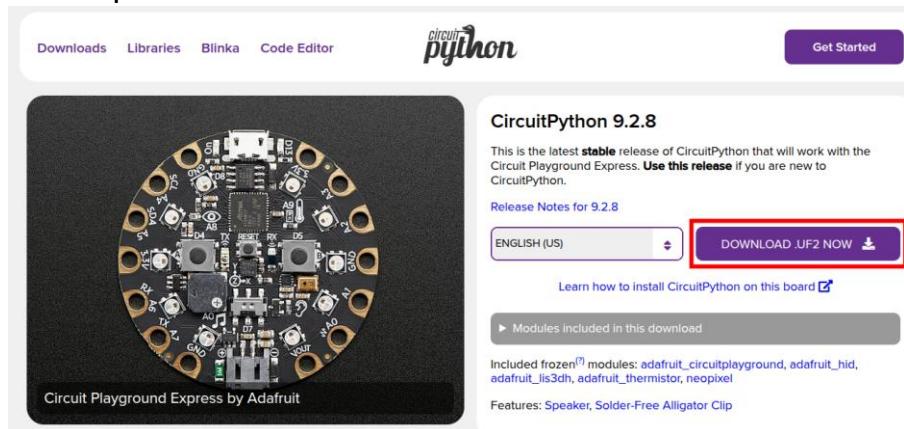
i

Før du beveger deg videre til de neste oppgavene, skal du sjekke om det dukker opp en fil som heter "CIRCUITPY" i filutforskeren når du kobler CPX-en til PC-en din med USB-kabelen. Hvis en slik mappe dukker opp, kan du hoppe videre til "[Oppgavebeskrivelse - OR-port](#)". Hvis det i stedet dukker opp en mappe som heter "CPLAYBOOT" er du nødt til å installere CircuitPython for å kunne bruke knappene på CPX-en som inngangssignalene til de logiske portene du skal bygge. For å gjøre dette må du gjøre følgende:

1. Gå til nettsiden:

https://circuitpython.org/board/circuitplayground_express/

2. Klikk på "Download .UF2 NOW"



3. Dra den nedlastede .uf2-filen til "CPLAYBOOT"-mappen

Etter noen sekunder vil "CPLAYBOOT"-mappen forsvinne, og en ny mappe kalt "CIRCUITPY" vil dukke opp.

Inne i "CIRCUITPY"-mappen skal det ligge en fil som heter code.py. Denne filen inneholder koden som setter opp knappene som inngangssignaler.

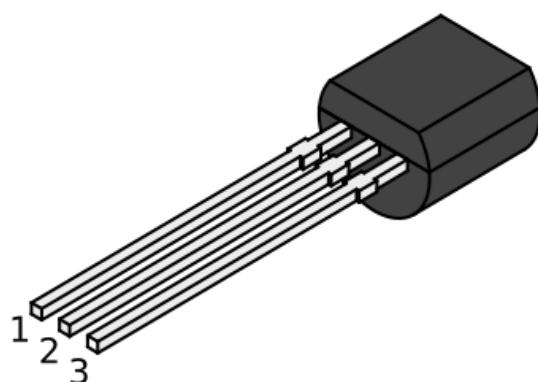
Hvis du ikke finner code.py-filen, eller trenger hjelp underveis, spør en av veilederne.

Transistorer og logiske porter

Hva er en transistor?

En transistor er en elektrisk komponent som fungerer som en bryter eller forsterker for elektrisk strøm. Transistorer brukes overalt, for eksempel i datamaskiner og mobiltelefoner.

1. Kollektor
2. Base
3. Emitter



Som du ser på bildet, har transistoren tre bein: **kollektor, base og emitter**. Basen fungerer som en styringsinngang. Hvis man sender strøm inn i basen, vil det begynne å gå strøm mellom kollektoren og emitteren. Når det ikke sendes strøm inn i basen, vil det heller ikke gå noe strøm

mellan kollektoren og emitteren. Slik kan transistoren brukes som en bryter i en elektrisk krets.

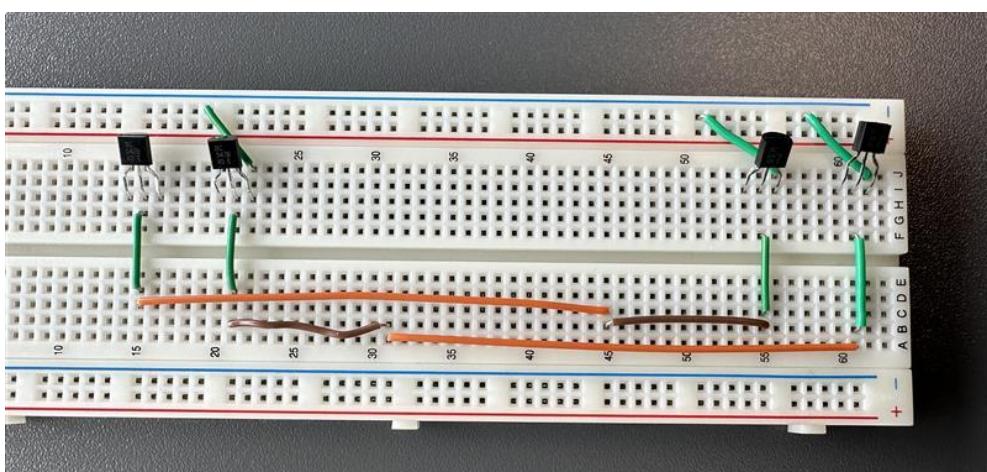
Det finnes flere ulike typer transistorer. I de transistorene du skal bruke i dag går strømmen **fra kollektoren til emitteren** når transistoren er aktivert (dette heter en NPN transistor).

Logiske porter

En logisk port er en grunnleggende byggestein i digital elektronikk. En logisk port utfører en enkel logisk operasjon på ett eller flere inngangssignaler, og gir ut ett utgangssignal. En logisk operasjon er en regel som bestemmer hva utgangssignalet skal være basert på inngangen(e). Inngangene og utgangen kan ha én av to verdier: 0 eller 1. Disse verdiene kan ha forskjellige representasjoner avhengig av systemet. I dag skal vi bruke en lysdiode til å illustrere hvordan utgangssignalene fra ulike logiske porter varierer når vi endrer inngangssignalene. Vi skal koble utgangssignalet fra porten til en lysdiode, og vi tenker oss at signalet har verdien 1 når dioden lyser, og 0 når den ikke lyser.

Oppgavebeskrivelse

I dag skal du lære om to logiske porter, AND og OR, og du skal få lage logiske porter med transistorer. Kretsen du starter med ser slik ut:



OR-port

Hva er en OR-port?

OR betyr ELLER. En slik port gir utgangssignalet 1 dersom **minst ett** av inngangssignalene er 1. For eksempel, hvis man har to inngangssignaler, A

og B, så må signal A **ELLER** signal B være 1. OR-porten kan man tenke på som et plusstegn: $1 + 0 = 1$

Under ser du en oversikt over hva slags utgangssignal porten vil gi ved forskjellige inngangssignaler.

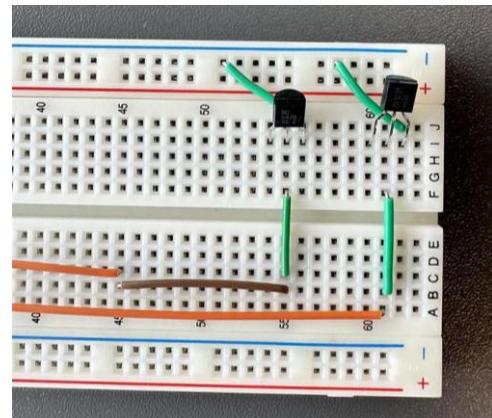
Inngangssignal A	Inngangssignal B	Utgangssignal
0	0	0
0	1	1
1	0	1
1	1	1

Oppgave: Bygg kretsen

Du skal bygge en krets som illustrerer hvordan en OR-port fungerer, ved hjelp av to transistorer, en motstand og en lysdiode. Du skal bruke kretsen vist på bildet på neste side som utgangspunkt (kretsen til høyre på breadboardet).

Du vet nå at en transistor fungerer som en bryter. Når det sendes strøm til basen, lukker transistoren kretsen, og strøm kan gå fra kollektoren til emitteren. I denne oppgaven skal du bruke knappene på CPX-en (A og B) som inngangssignaler til basene på de to transistorene. Disse har vi koblet for deg. Når du trykker inn en av knappene vil det sendes strøm gjennom ledningen fra CPX-en og inn i basen på den tilkoblede transistoren. Forestill deg derfor at hver knapp er et inngangssignal som er lik 1 når knappen trykkes inn og 0 ellers. Vi har i tillegg koblet emitterene til minus på koblingsbrettet for deg. Din oppgave er å **lukke kretsen** ved å koble ledninger til kollektorene, med en lysdiode, slik at kretsen oppfører seg som en OR-port. Når kretsen er koblet til riktig, skal lysdioden lyse så lenge du trykker på minst én av knappene.

Når du har koblet sammen alt, skal du be om hjelp til å koble til knappene og strømmen fra CPX-en.





Husk på det du lærte i forrige oppgave:

- Hvilken retning går strømmen?
- Hvilken vei må lysdioden stå?

Husk også på at lysdioden ikke tåler så mye strøm, så du er nødt til å begrense hvor mye strøm som går gjennom den. Husker du hva slags komponent vi bruker for å begrense hvor mye strøm som går i en krets?



Oppfører lyset seg som du forventer nå, basert på hva du vet om en OR-port? Har du laget en seriekobling eller en parallelkobling?



Hint (prøv selv før du leser)

Du er nødt til å koble en ledning i kollektoren til hver av de to transistorene, og koble den andre enden av disse ledningene til samme rad på koblingsbrettet. Da vil utgangssignalene fra begge transistorene gå til samme rad på brettet, og det vil gå strøm til denne raden så lenge det går strøm gjennom minst én av transistorene. Lysdioden kan så kobles til denne raden for å lukke kretsen.

AND-port

Hva er en AND-port?

AND betyr OG. En slik port gir utgangssignalet 1 dersom **begge** inngangssignalene er lik 1. For eksempel, hvis man har to inngangssignaler, må signal A **OG** signal B være 1. Det går også an å tenke på denne porten som et gangetegn: $1 \cdot 1 = 1$

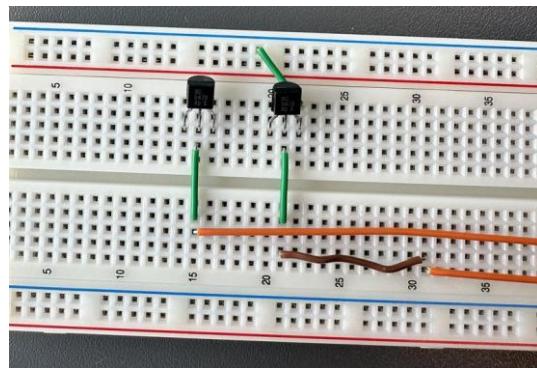
Under ser du en oversikt over hva slags utgangssignal porten vil gi ved forskjellige inngangssignaler.

Inngangssignal A	Inngangssignal B	Utgangssignal
0	0	0
0	1	0
1	0	0
1	1	1

Oppgave: Bygg kretsen

Du skal bygge en krets som illustrerer hvordan en AND-port fungerer, ved hjelp av to transistorer, en motstand og en lysdiode. Denne gangen skal du starte med kretsen på bildet til høyre (kretsen til venstre på breadboardet).

Start med å trekke ut USB-kabelen til CPX-en fra PC-en din.



Denne kretsen er nesten helt lik kretsen i forrige oppgave, med ett unntak - her er det bare **emitteren på den ene transistoren** som er koblet til minus på brettet. Din oppgave er å koble kollektorene til begge transistorene slik at lysdioden du skal koble til i kretsen kun lyser når begge knappene er trykket inn. De må altså kobles litt annerledes enn i forrige oppgave.

Når du er ferdig å koble sammen kretsen, kan du plugge inn USB-kabelen igjen.



Trykk på knappene. Oppfører lyset seg som du forventer nå, basert på hva du vet om en AND-port? Har du laget en seriekobling eller en parallellekobling?



Hint 1

I forrige oppgave var tilkoblingen av ledningene til hver transistor uavhengig av hverandre ettersom lysdioden lyste når **minst én av knappene** var trykket inn. Men i denne oppgaven er transistorene avhengig av hverandre fordi lysdioden kun lyser når **begge transistorene er åpne**.

**Hint 2**

Du må koble kollektoren i den ene transistoren til emitteren i den andre. Deretter kan du koble lysdioden og motstanden på samme måte som i oppgave a), men denne gangen skal dioden kun kobles til kollektoren på den ene transistoren.

Ekstraoppgave: Monitorér arbeidsmiljø

Til nå har du jobbet med to generiske inngangssignaler A og B som du kunne endre på ved å trykke på knappene på CPX-en. Nå skal vi se hva disse signalene kan være i praksis, og vise hvordan en AND- eller OR-port kan brukes i en virkelig situasjon!

Forestill deg at du skal lage et system som monitorerer arbeidsmiljøet på et kontor. Et godt arbeidsmiljø handler blant annet om **god belysning** og **passer støy nivå**. I denne oppgaven skal du måle både **lys** og **lyd**, og bruke de logiske kretsene du lagde i forrige oppgave for å varsle om noen av nivåene blir for høye.

CPX-en har allerede sensorer som måler lys og lyd. De er markert med et øye og øre på chipen, men vi har i tillegg markert dem i gult på bildet til høyre.

I sted lasta du ned et program på CPX-en. Det fungerer slik at den sender ut '1' på ledningen koblet til A1 når knapp 'A' er trykket inn, og '0' når den ikke er trykket inn. Tilsvarende sender ledningen koblet til A2 ut '1' når knapp 'B' er trykket inn, og '0' når den ikke er trykket inn. For å lage en arbeidsmiljømonitor ønsker vi å lage et *nytt program* som kan lastes inn på CPX-en. I dette programmet skal A1 **sende ut '1' når lydnivået er for høyt**, og A2 skal **sende ut '1' når lysnivået er for lavt**. Tabellene under viser hvordan programmet skal fungere:

Utgangssignal A (grønn eller gul farge på ledning):

	Lydnivå over 130	Lydnivå under 130
Verdi på signal i ledning fra A1	1 / høy spenning	0 / lav spenning

Utgangssignal B (blå eller lilla farge på ledning):

	Lysnivå over 70	Lysnivå under 70
Verdi på signal i ledning fra A2	0 / lav spenning	1 / høy spenning

Koble til og pare CPX-en til PC-en din

Gå inn på <https://makecode.adafruit.com/#editor>

Før du kan laste opp kode direkte til CPX-en via nettsiden, må du koble til og pare den med PC-en din:

1. Koble til CPX-en:

- Koble USB-kabelen til PC-en din.
- Åpne filutforskeren på PC-en og se hva CPX-en vises som:
 - Hvis den vises som "CPLAYBOOT", gå videre til neste steg.
 - Hvis den vises som "CIRCUITPY", må du dobbeltklikke på "RESET"-knappen på CPX-en. CPX-en skal nå vises som "CPLAYBOOT" i filutforskeren.

2. Skriv testkode:

På skjermen ser du en grønn "forever"-blokk. Denne blokken kjører koden gjentatte ganger så lenge programmet kjører. For øyeblikket er "forever"-blokken tom, som vil si at det ikke skjer noe når du kjører programmet. Du skal nå teste om du klarer å laste inn kode på CPX-en. Gå til menyen til venstre og klikk på kategorien "LIGHTS". Finn en blokk som lar deg sette fargen på ett eller flere av lysene på CPX-en. Dra denne blokken inn i "forever"-blokken.

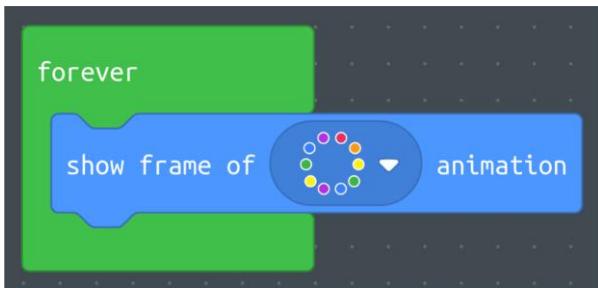


Tips!

Du kan få mer informasjon om en blokk ved å holde musepekeren over den.

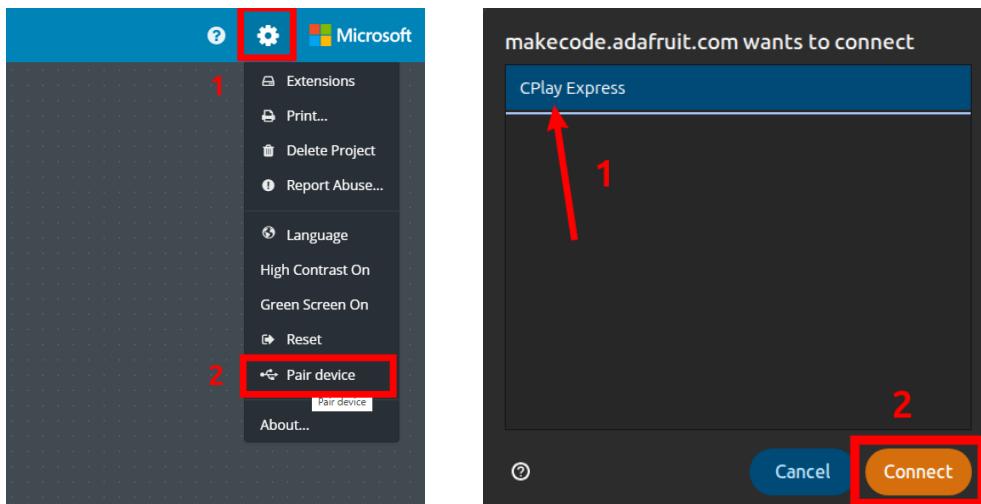


For at koden skal kunne kjøre, må formen på blokken du legger inn passe med formen til åpningen på "forever"-blokka. Et eksempel er vist her:



3. Pare CPX-en din:

Trykk på tannhjulet øverst i høyre hjørne i nettleseren, og trykk så på "Pair device". Trykk på enheten som dukker opp, og deretter "Connect". (Hvis paring mislykkes, gå til "**Hvis du ikke klarer å pare enheten din**").



4. Test om paringen var vellykket:

Nå kan du trykke på den rosa knappen som heter "Download" for å laste over koden. Lyser noen lys på CPX-en? Da har du fått til å laste inn kode!



Hvis du ikke klarer å pare enheten din

Noen typer PC'er kan slite med å finne enheten. For de det gjelder, må koden lastes opp **manuelt**.

Manuell nedlasting:

1. Hvis du ikke finner en mappe som heter "CPLAYBOOT" på PC-en din, så må du trykke på "RESET"-knappen slik at CPX-en lyser grønt.
2. Trykk "Download" på nettsiden, slik at det lastes ned en fil som slutter på .uf2.
3. Gå inn på applikasjonen som viser filene på PC-en din (Finder, File Explorer, Files)
4. Finn mappen som heter "CPLAYBOOT"
5. Legg filen du nettopp lastet ned inn på CPX-en ved å dra den inn i mappen.

Stegene over må gjøres hver gang du skal legge inn et nytt program



Hvis du ikke har så mye erfaring med koding, er det lurt å bruke **blokker** når du skal løse denne oppgaven. Denne metoden gir en enkel og intuitiv måte å lage koden din på ved å dra og slippe blokker. Hvis du er kjent med **tekstkoding** eller har lyst til å prøve deg på det, kan du hoppe videre til "[Oppgavebeskrivelse - koding med tekst](#)".

Oppgavebeskrivelse - koding med blokker

a) Legg inn "if-else"-blokker

Nå som du har fått til å laste inn kode på CPX-en, er neste steg å lage *logikken* til programmet. Start med to "if-else"-blokker, som du finner i "LOGIC"-kategorien i menyen til venstre.



Pass på at du fjerner det som var inne i "forever"-bokken fra forrige deloppgave. Pass også på at du har riktig tall på pixelene. Vi har lyst til å bruke pixelene som er nærmest de to inngangssignalene som går over til koblingsbrettet, fordi det gjør det lettere når vi senere i oppgaven skal sjekke hva som skjer i koden.

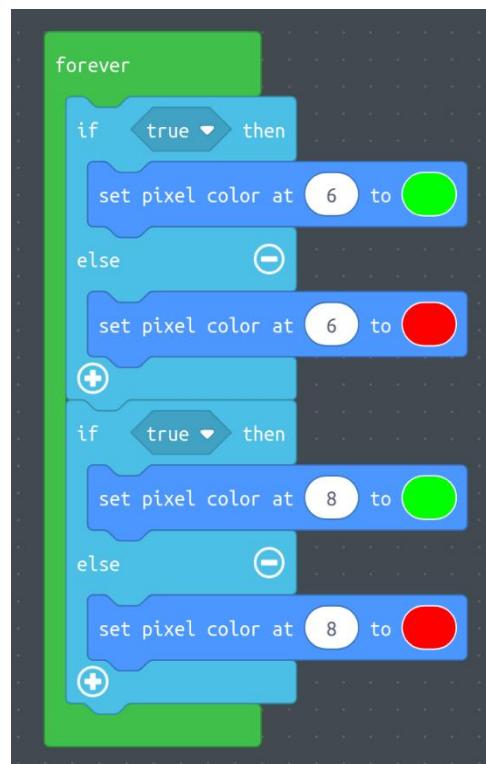
For å forstå hvordan en slik blokk fungerer, starter vi med å legge inn blokker for å sette to pixeler til grønt eller rødt. Slik skal det se ut:

Last ned den nye koden, og se hva som skjer.



Hvilken farge lyser pixelene? Rødt eller grønt?

En "if-else"-blokk fungerer slik at den har to ting den kan velge mellom. Den velger hva den skal gjøre, basert på om



det som står i det diamant-formede feltet (etter "if") er sant eller usant. I vårt tilfelle velger blokken om den skal sette en pixel (et lys) til rødt eller grønt. Den første "if-else"-blokka styrer pixel nr. 6 (ved A1), den andre "if-else"-blokka styrer pixel nr. 8 (ved A2). Fordi begge "if-else"-blokkene er "true", velger begge å sette pixelen til grønn i stedet for rød.

Endre den første "if-else" blokka slik at den går fra "true" til "false". Last ned den nye koden.



Hvilke farger lyser pixelene nå?

b) Legg inn sammenlikningsblokker

I de diamant-formede feltene kan man også legge inn andre ting enn "true" og "false", så lenge det er et *logisk uttrykk* som kan regnes ut til å være sant eller usant. Et eksempel er " $0 < 3$ ", som er *sant*, og " $0 = 3$ " som er *usant*.

Finn fram en "comparison"-blokk fra "LOGIC"-kategorien og dra den inn i den første "if-else"-blokken, slik →



Prøv å endre på tallverdiene og det matematiske symbolet, for å skjønne hvordan det fungerer.

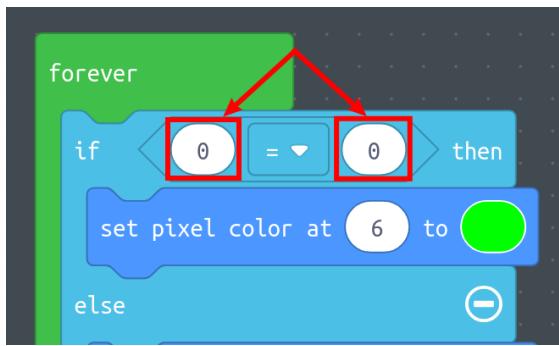


Hvilken farge lyser pixel nr. 6 hvis du for eksempel putter inn " $10 < 5$ "?

c) Mål lydnivå og lysnivå

De runde feltene som er inne i sammenlikningsblokken (vist på bildet under) kan også ta inn andre tallverdier, for eksempel fra sensorer. Gå i "INPUT"-kategorien og finn fram sensorblokkene som mäter lyd og lys. De skal ha runde kanter, slik at de passer inn i de runde feltene i sammenlikningsblokken. Blokkene fra sensorene vil hele tiden bli oppdatert og representer de siste verdiene som har blitt målt.

Putt blokken som mäter *lydnivå* inn som **ett** av inputene i sammenlikningen i den øverste "if-else"-blokka. Klarer du å endre om på uttrykket slik at lyset lyser **grønt** om lydnivået er **under 130**, og **rødt** dersom det er **over 130**? Last over koden, og sjekk ved å blåse på lydsensoren (det oppfattes som en høy lyd). Pixel nr. 6 skal da endre seg fra grønt til rødt.

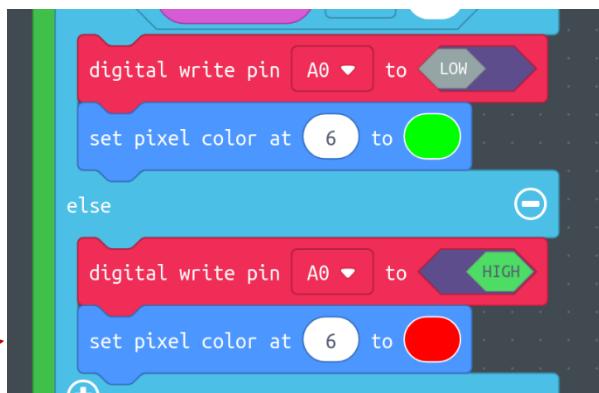


Putt blokken som mäter *lysnivå* inn som **ett** av inputene i sammenlikningen i den nederste "if-else" blokka. Klarer du å endre om på uttrykket slik at lyset lyser **grønt** om lysnivået er **over 70**, og **rødt** dersom det er **under 70**? Last over koden, og sjekk ved å legge en finger over lyssensoren, og se om pixel nr. 8 endrer seg fra grønt til rødt. Når du fjerner fingeren skal pixelen gå tilbake til grønt.

d) Sett spenning på ledningene

Hensikten med oppgaven er å lage en arbeidsmiljø-monitor, som varsler om lydnivået har blitt for høyt **eller** lysnivået har blitt for lavt. Vi ønsker at varselet skal være at en av lysdiodene som er på koblingsbrettet lyser.

Derfor må vi ha en måte å overføre signalene fra CPX-en via de to signalledningene som er mellom CPX-en og koblingsbrettet. Finn derfor fram blokken som kan *skrive* et digitalt signal til en *utgang* på CPX-en. Du finner den i "ADVANCED" → "PINS". Den ser slik ut



Plasser to slike blokker inni hver "if-else"-blokk, over hvert sted fargen på en pixel blir satt (se bildet over).

Utgangene (også kalt "pin") er hullene som er plassert rundt kanten på CPX-en. De utgangene man kan skrive til er nummerert fra A0 til A7, med liten skrift. Vi ønsker at den første "if-else"-blokken skal skrive til utgangen som er koblet til den grønne/gule ledningen. Se på CPX-en for å finne ut hva denne utgangen heter, og gjør en endring på de to blokkene i den første "if-else"-blokken slik at de ikke skriver til A0, men til utgangen som

er koblet til grønn/gul ledning. Endre også slik at den kommer til å skrive "low" når lydnivået er akseptabelt (lydnivå under 130) og "high" ellers.

Vi ønsker at den nederste "if-else"-blokken skal skrive til utgangen som er koblet til den blå/lilla ledningen. Se på CPX-en for å finne ut hva denne utgangen heter, og gjør en endring på de to blokkene i den nederste "if-else"-blokken slik at de ikke skriver til A0, men til utgangen som er koblet til blå/lilla ledning. Endre også slik at den kommer til å skrive "low" når lysnivået er akseptabelt (lysnivå over 70) og "high" ellers.

Sjekk om koden er riktig ved å utføre følgene test:

- 1) Dekke til lyssensoren. Da skal kun én av lysdiodene på koblingsbrettet lyse.
- 2) Blås på lydsensoren. Da skal også kun én lysdiode på koblingsbrettet lyse.
- 3) Prøv nå å gjør begge samtidig. Hva skjer?



Hvilken av de to kretsene på koblingsbrettet ville du brukt for å varsle om minst én av nivåene var uakseptable? Hvilken krets ville du brukt hvis du kun skulle varsle om begge nivåene var uakseptable?

Gratulerer - du har fullført alle oppgavene! 😊 🎉

Hvis det er mer tid igjen, spør gjerne veilederne om flere oppgaver.

Oppgavebeskrivelse - koding med tekst

For å kode med tekst, trykk på boksen hvor det står "**{}** Javascript". Om du aldri har kodet med tekst før, er det et par ting du burde kunne. Hopp over forklaringene hvis du er kjent med dette fra før av.

Funksjon

En funksjon er en kodesnutt som gjør en bestemt oppgave for deg. Den kan ha det man kaller "inputs", som er inngangsverdier du gir den. Den kan også ha "outputs", som er utgangsverdier du får fra funksjonen. Du kan lage dine egne funksjoner, men for denne oppgaven skal du bare bruke ferdiglagde funksjoner. Du kan kjenne igjen funksjoner ved at de har parenteser () etter seg. Hvis parentesen er tom, trenger ikke funksjonen noen inngangsverdier. Her kan du se eksempel på to funksjoner:

```
light.setPixelColor(6, Colors.Red)  
let temperature = input.temperature(TemperatureUnit.Celsius)
```

Den første funksjonen setter et lys til en gitt farge. Den har to inputs. Det første innputtet, "6", spesifiserer hvilket lys som skal bli satt. Det andre innputtet, "Colors.Red", spesifiserer fargen på lyset. Med andre ord vil denne funksjonen få lys nr. 6 til å lyse rødt.

Den andre funksjonen er koblet til en temperaturmåler på CPX-en. Den har en input, som bestemmer hvilken enhet den skal måle temperaturen i. Her blir den satt til grader Celsius. Den har en utgangsverdi, som er temperaturen. For å kunne bruke denne verdien videre, blir den lagt inn i en *variabel*, "temperature".

Variabel

En variabel er en beholder for data. Du gir beholderen et navn, og så kan du fylle den med noe, for eksempel et tall eller en tekst. Senere kan du hente dataen ut igjen.

If-else-setning

En "if-else-setning" brukes for å lage logikk i et program. Den starter først med en "if" og et uttrykk som enten er sant eller usant:

```
if (lightLevel > 10) {  
    music.playTone(440, 500)  
}
```

Dette betyr at hvis lysnivået er over 10, vil CPX-en spille av en lyd, fordi funksjonen "music.playTone" blir kalt på. En if-setning kan stå alene, eller den kan kombineres med en "else", for å si hva som skjer *hvis ikke* det første uttrykket er sant.

```
if (lightLevel > 10) {
    music.playTone(440, 500)
} else {
    music.stopAllSounds()
}
```

Kodesnutten over vil også spille av en lyd hvis lysnivået er over 10, men hvis lysnivået går under eller er lik 10, vil den stoppe å spille. Den første kodesnutten med kun en "if" hadde fortsatt å spille lyd, selv om lysnivået hadde gått tilbake ned igjen.

a) Mål lydnivå og lysnivå

- 1) Først må vi hente ut de riktige sensordataene. Finn ut hvordan man leser av lydnivået, og legg verdien inn i variabelen "soundLevel". Ta utgangspunkt i denne koden:

```
forever(function () {
    let soundLevel = /*Fyll inn kode her i oppgave A*/
    let lightLevel = /*Fyll inn kode her i oppgave A*/

    if (soundLevel > 130) {
        light.setPixelColor(6, Colors.Red)
        /*Fyll inn kode her i oppgave B*/
    } else {
        light.setPixelColor(6, Colors.Green)
        /*Fyll inn kode her i oppgave B*/
    }

    if (lightLevel < 70) {
        light.setPixelColor(8, Colors.Red)
        /*Fyll inn kode her i oppgave B*/
    } else {
        light.setPixelColor(8, Colors.Green)
        /*Fyll inn kode her i oppgave B*/
    }
})
```

Kodeforslag

- 2) Finn så ut hvordan man kan lese av lysnivået og legg dette inn i variabelen "lightLevel".
- 3) Last ned koden, og sjekk om den fungerer. Hvis du dekker over lyssensoren (se tidligere illustrasjon for å finne ut hvor den er), skal

pixelen ved A1 skifte fra grønt til rødt. Hvis du blåser på chipen, skal pixelen ved A2 skifte fra grønt til rødt.

b) Sett spenning på ledningene

Hensikten med oppgaven er å lage en arbeidsmiljø-monitor, som varsler om lydnivået har blitt for høyt *eller* lysnivået har blitt for lavt. Vi ønsker at varselet skal være en av lysdiodene som er på koblingsbrettet. Vi må derfor ha en måte å overføre signalene fra CPX-en via de to signalledningene som er mellom CPX-en og koblingsbrettet. For å gjøre dette, kan man bruke denne funksjonen:

```
pins.AX.digitalWrite(true)
```

Innputtet til funksjonen kan enten være "true" eller "false". "True" gjør at funksjonen setter en utgang til 1 / høy spenning. "False" gjør at funksjonen setter en utgang til 0 / lav spenning. Utgangene (også kalt "pin") er hullene som er plassert rundt kanten på CPX-en. Totalt må man legge inn fire digitale skrivinger, to i hver "if-else"-setning. Se på eksempekkoden for å finne ut hvor de skal legges inn.

De utgangene man kan skrive til er nummerert fra A0 til A7, med liten skrift. Vi ønsker at den første "if-else"-setningen skal skrive til utgangen som er koblet til den grønne/gule ledningen. Se på CPX'en for å finne ut hva denne utgangen heter, og gjør om funksjonene i den første "if-else"-setningen slik at de skriver til riktig utgang (bytt ut 'AX' med riktig utgang). Gjør også en endring på innputtet til den ene funksjonen slik at "if-else"-setningen setter en **1 / høy spenning** på den grønne/gule ledningen når lydnivået er uakseptabelt (over 130) og setter en **0 / lav spenning** på den grønne/gule ledningen ellers.

Se på CPX'en og finn ut hva utgangen som er koblet til den blå/lilla ledningen heter. Gjør om funksjonene i den andre "if-else"-setningen slik at den skriver til riktig utgang (bytt ut 'AX' med riktig utgang). Gjør også en endring på innputtet til den ene funksjonen slik at "if-else"-setningen setter en **1 / høy spenning** på den blå/lilla ledningen når lysnivået er uakseptabelt (under 70) og setter en **0 / lav spenning** på den grønne/gule ledningen ellers.

Sjekk om koden er riktig ved å utføre følgende test:

- 1) Dekke til lyssensoren. Da skal kun én av lysdiodene på koblingsbrettet lyse.

- 2) Blås på lydsensoren. Da skal også kun én lysdiode på koblingsbrettet lyse.
- 3) Prøv nå å gjør begge samtidig. Hva skjer?



Hvilken av de to kretsene på koblingsbrettet ville du brukt for å varsle om minst én av nivåene var uakseptable? Hvilken krets ville du brukt hvis du kun skulle varsle om begge nivåene var uakseptable?

Gratulerer - du har fullført alle oppgavene! 😊 🎉

Hvis det er mer tid igjen, spør gjerne veilederne om flere oppgaver.