## TinkerCad EP 3 Sensor og analoge avlesinger

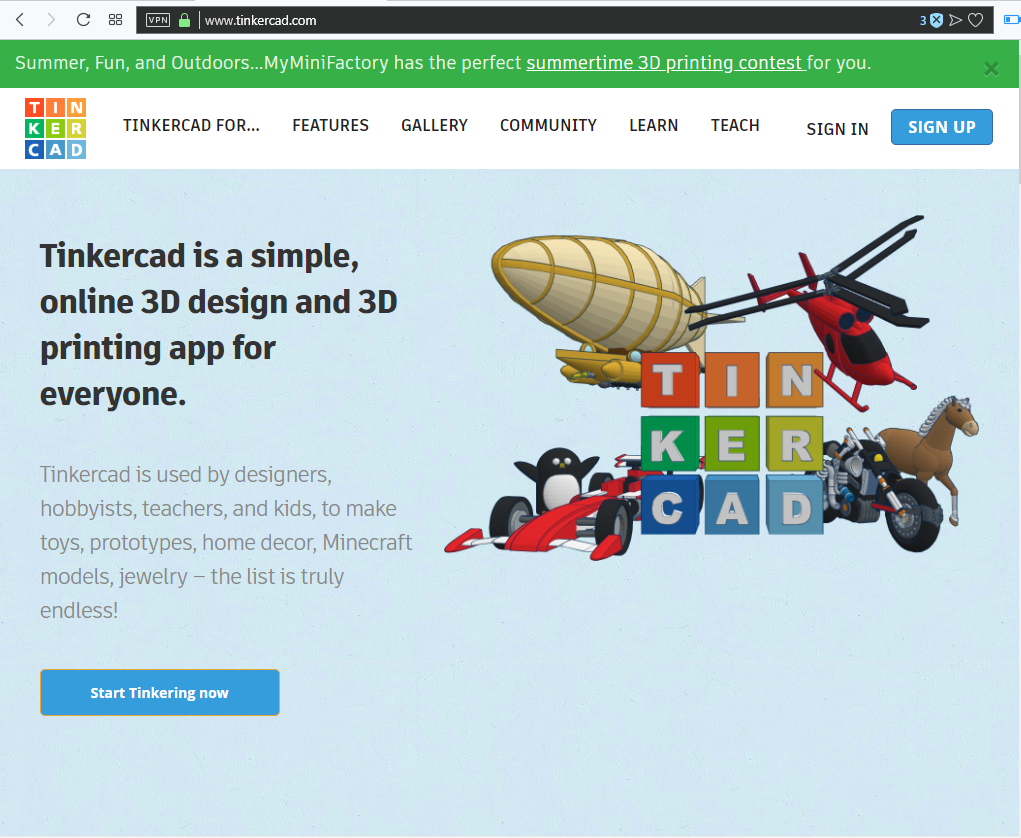
Nå skall vi bruke arduinoen for og automatisk regulere en LED avhengig av hvor mørkt det er i omgivelsene. Da trenger vi en sensor for å måle omgivelsene også trenger vi lage en skecth som bruker grunnleggende syntaks, variabler, konstanter så da er det kjekt og vite litt om datatyper.  
Det blir også nødvendig og se lit på hvordan en krets er tegnet for å se hvordan ting henger sammen.

-Grunnleggende syntaks   
-Variabler og konstanter  
-Datatyper

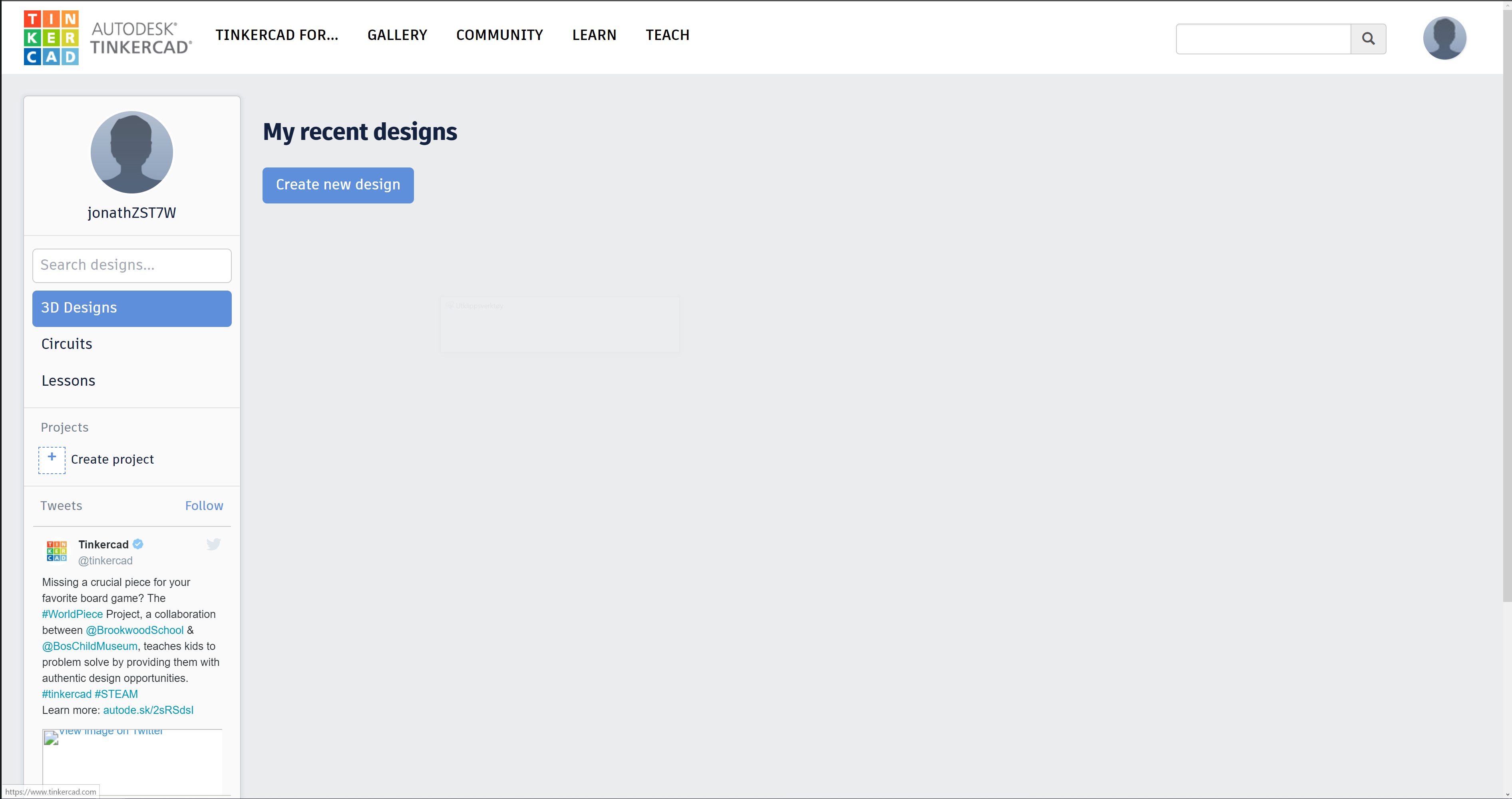
## Steg 1

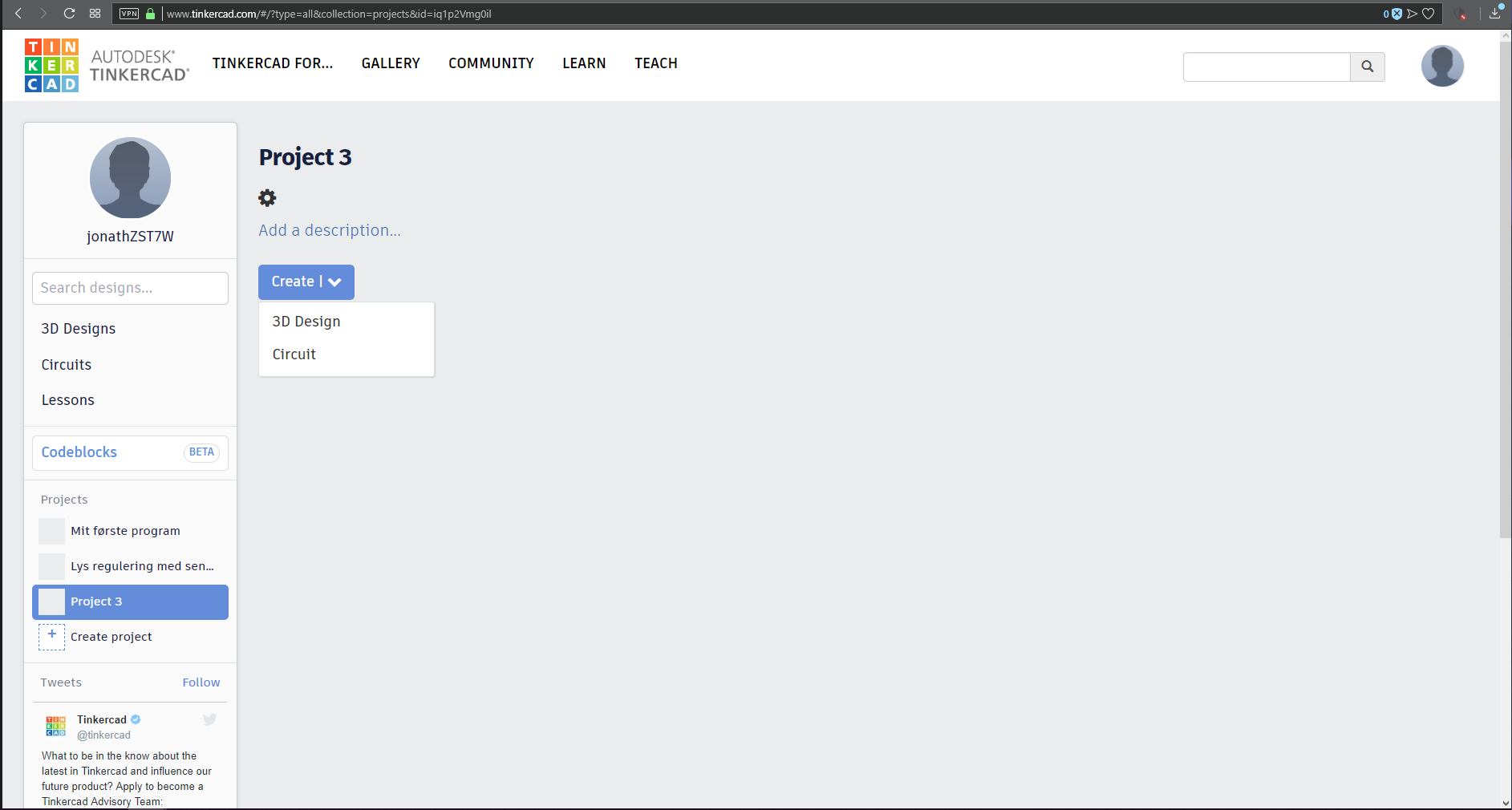
Starter med å gå til TinkerCad -Logg in – Create project – (gi det et navn) Lysregulering med sensor- create curuit.

Gå til <https://www.tinkercad.com> og logg inn.

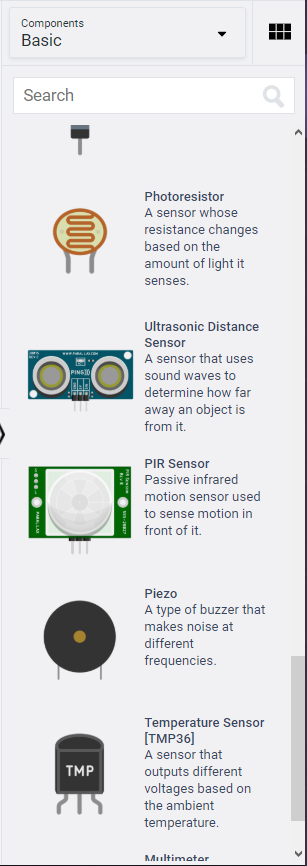


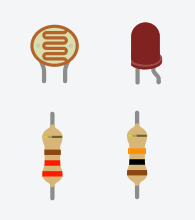
Velg crate project – klikk på tannhjulet og gi det et navn og gjerne en beskrivelse. Og velg cuircuit.





## Steg 2 – sette sammen komponentene

Vi trenger en Arduino, en LDR sensor (lys følsom motstand), to motstander og en LED. Kan også ta med et protobrett om man ønsker.

TIP: klikk på knappen ved siden av dropdown menyen for å få en mer detaljert beskrivelse av komponentene.

Dette er en krets som benytter seg av spenningsdeling og hvor spenningen endre seg over en variabel motstand. Dette kan vi bruke for å måle verdien av lyset i rommet rundt oss. Ved hjelp av en av de analoge portene på Arduinoen og noe smart kode kan vi følle med på lysnivået i rommet.

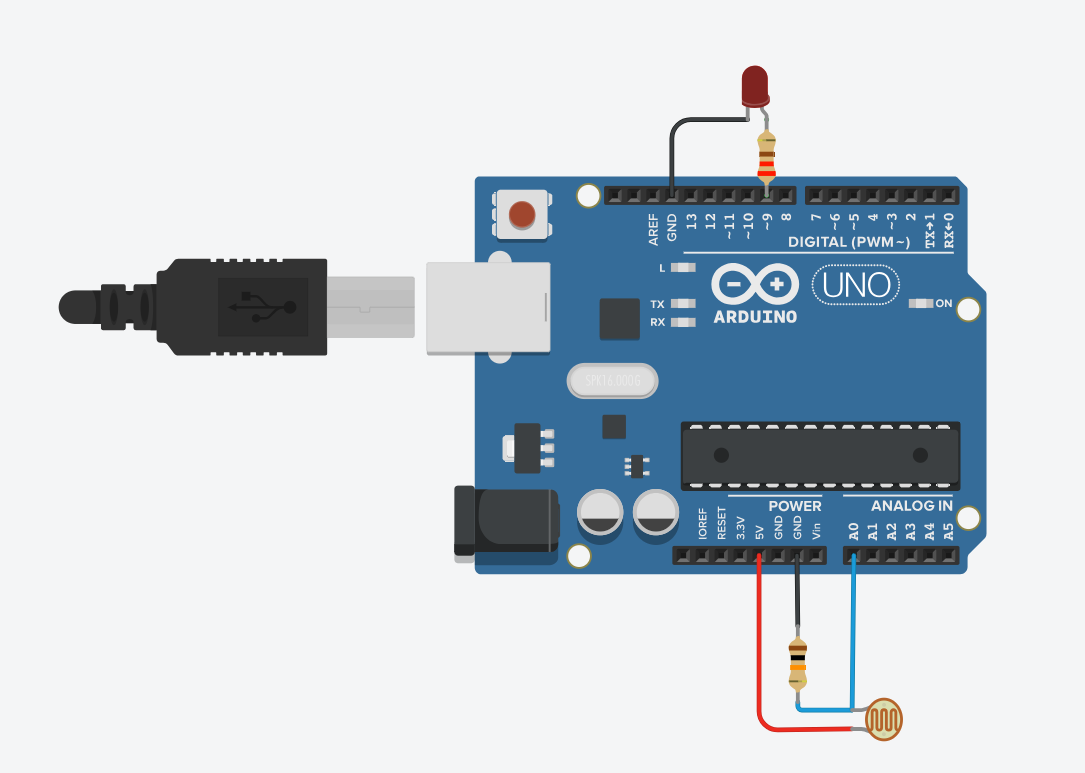
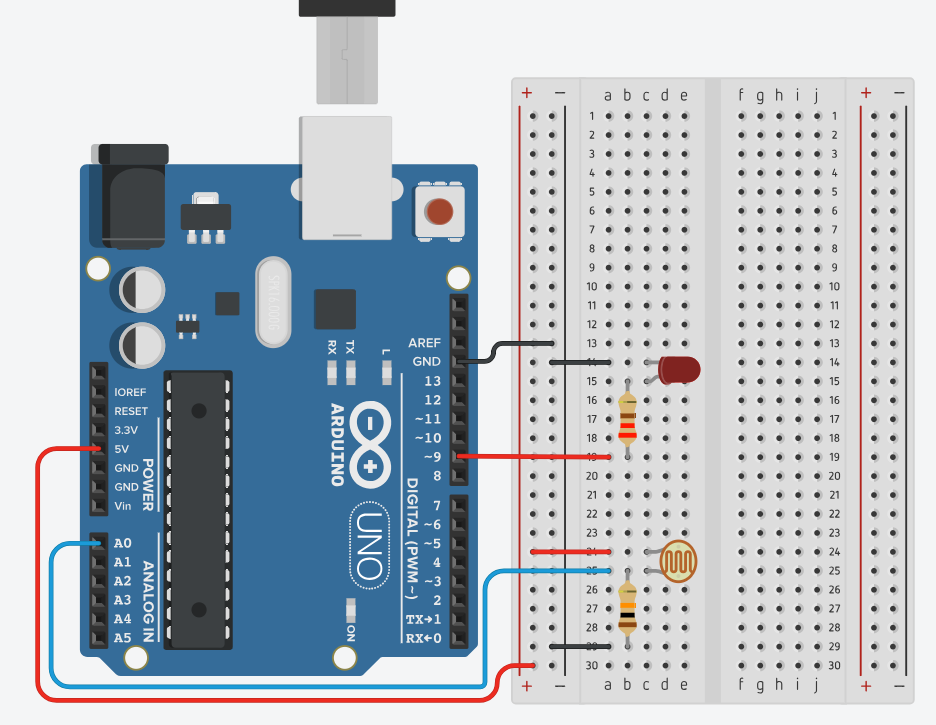


Beskrivelse som er generert med svært lav visshetTip: <https://www.youtube.com/watch?v=u9Riurh4y9U> hvordan en fotoresistor fungerer

Men først må vi se lit på hvordan vi setter opp  
kretsen slik at det vill fungere som vi ønsker.  
Til høyre så ser vi et kretsskjema. Og dette skjema   
illustrere hvordan sammenhengen mellom alle   
komponentene vi skal bruke. Vi ser at her så brukes   
det lit andre symboler til å illustrere de komponenten   
vi skal bruke. **R3** blir for oss photoresistoren, den er   
forment som en motstand men med en pil igjennom   
som indikerer at den er variabel. **LED1** er leden vår   
og den er illustrert som en diode men med to piler som   
forlater som indikerer lys.  
R2 og R1 er vanlige   
motstander og de forskjellige   
porten er illustrert med en Box og jord er en   
vertikal strek og ofte **GND** i tillegg.

Når det kommer til hvordan skjema ser ut og hvordan   
man kobler det opp i virkeligheten så trenger ikke det å   
være 1 til 1, somregel er det ikke likt og vi kan se på none   
eksempler nå med hvordan koble opp denne kretsen.

**GND**

Her har vi nå to oppsett som begge er like gyldig og stemmer med kretsskjemaet, men de ligner kanskje ikke så mye på kretsskjemaet med det første. Og de ser også ganske forskjellige ut selv om det er akkurat det samme.

## Steg 3 – hvordan skrive koden

For å dra nytte av kretsen vi har bygd så må vi skrive en kode som er lit annerledes en hva vi har gjort tidligere. Vi må ha kjennskap til datatyper og vite va vi skal bruke samtidig så må vi ha noen nye funksjoner.

Vi kan dele opp Arduino programmer i 3 hoved deler.

* Funksjoner
  + Digital og analog I\O
  + Matte
  + Kommunikasjon
  + Tid
* Variabler
  + Konstanter
  + Data typer
  + Scop og Qualifiers
* Strukturer
  + Sketch
  + Control strukturer
  + Aritmetisk, bolsk og comparison operatorer
  + Bit vis operatorer

En mer utfyllende liste finer du her: <https://www.arduino.cc/reference/en/>

Når man bruker variabler er det viktig å velge riktig data type son at man beholder dataen slik den skal være. Vi vil nå se på noen datatyper og konstanter slik at vi vet va vi velger når vi skal skrive koden vår.

Int, String, Bool og float brukes ofte så vi kan se lit nærere på dem.

int er en forkortelse av integer som er en datatype for nummer.   
 int variabel = 13;

String bruks til ord, tegn og setninger og kan brukes f.eks son:  
 String setning1 = "Min setning";

bool brukes for å lagre bolkse verdier, ja nei san\usan osv.   
 bool running = false;

float brukes for å lagre data som er tall med desimaler.  
 float kommatall = 1.117;

Det finnes altså flere nytteområder for de forskjellige typer men siden vi skall bruke enkle tallverdier så kan vi holde os til int da dette holder for vårt bruksområde. Til å starte med så setter vi opp alle variablene våre, disse må på plass først og ved å skrive dem øverst i koden så sørger vi for at programmet har fått det med seg samtidig som at det er oversiktlig or oss og lese.

En av grunnen til at vi bruker pinne 9 på arduinone er at den har egenskapen PWM som enkelt forklart justerer hvor fort en kan skru seg av og på. Dette bruker vi for å justere svakere og sterkere lys på LED’en som vi bruker i kretsen vår.

Så alle variablene vi trenger kan være int’er. Og for å finne ut hva vi trenger så kan vi se på kretsen lit, her ser vi at vi er koblet til en LED og de går til pinne 9 så da vet vi at vi må ha den, men pohotoresitoren er koblet til A0 så da må vi ha med den også. Men er det nok og bare ha de to int’ene(variablene)? Nei vi trenger også et sted og lagre verdien i mellomtiden og det både til sensoren og til LED’en så vi må totalt ha 4 variabler.

Så er det Void setup’en vår. Den må inneholde alt vi skal bruke. Så hva bruker vi egentlig. Vell det er flere komponenter i kretsen her men bare to vi gjør noe med sett fra arduinoen sin side. Det er LED’en, den regulerer vi lysstyrken på og så har vi sensoren vår. Den leser vi av verdien på. Så da er det bare de to vi gjøre noe med fra kretsen sin side og disse blir jo pinmode output og input. Men så vill vi jo også se på datan(sensorverdien og led verdien) og da må vi bruke serial kommunikasjon. Det betyr jo at vi må ha med serial. begin i tillegg.

Til slutt har vi void setup og her skjer jo alt som må gjøres hele tiden. Nå kommer vi til å bruke alle variablene og noen ekstra funksjoner for og får det vi vil skall skje til og skje. Så hva er det vi vill, jo vi vill at en LED skall justere seg etter mengden lys som er i rommet. Jo lysere det er i rommet jo svakere skall LED’en lyse og motsatt. Så hva må vi gjøre da?   
først så må vi lese av verdien og lagre den i variabelen vår. Videre så må vi gi denne verdien til LED variablene vår men vi må også invertere verdien slik at høyeste verdi blir lavest og motsatt.

Det letteste å gjøre da er og bruke map funksjon som vill mappe verdiene vi får inn fra sensoren og over til verdien til LED utgangen. Funksjonen map(value, fromLow, fromHigh, toLow, toHigh) ser slik ut og vi vill da bruke sensorverden som value og verdien vi skall bruke er da 1024 bits oppløsning, det er ikke så viktig og forstå alt det men vi kan se på det som at sensoren har 0 til 1023 verdier som representerer mengden lys i rommet. Denne verdien vill vi fordele likt utover på LED’en vår og oppløsningen på leden er 256 eller fra 0 til 255 som vi kommer til å skrive. Map funksjonen kan vi skrive til variabelen vår for LED styrken.

