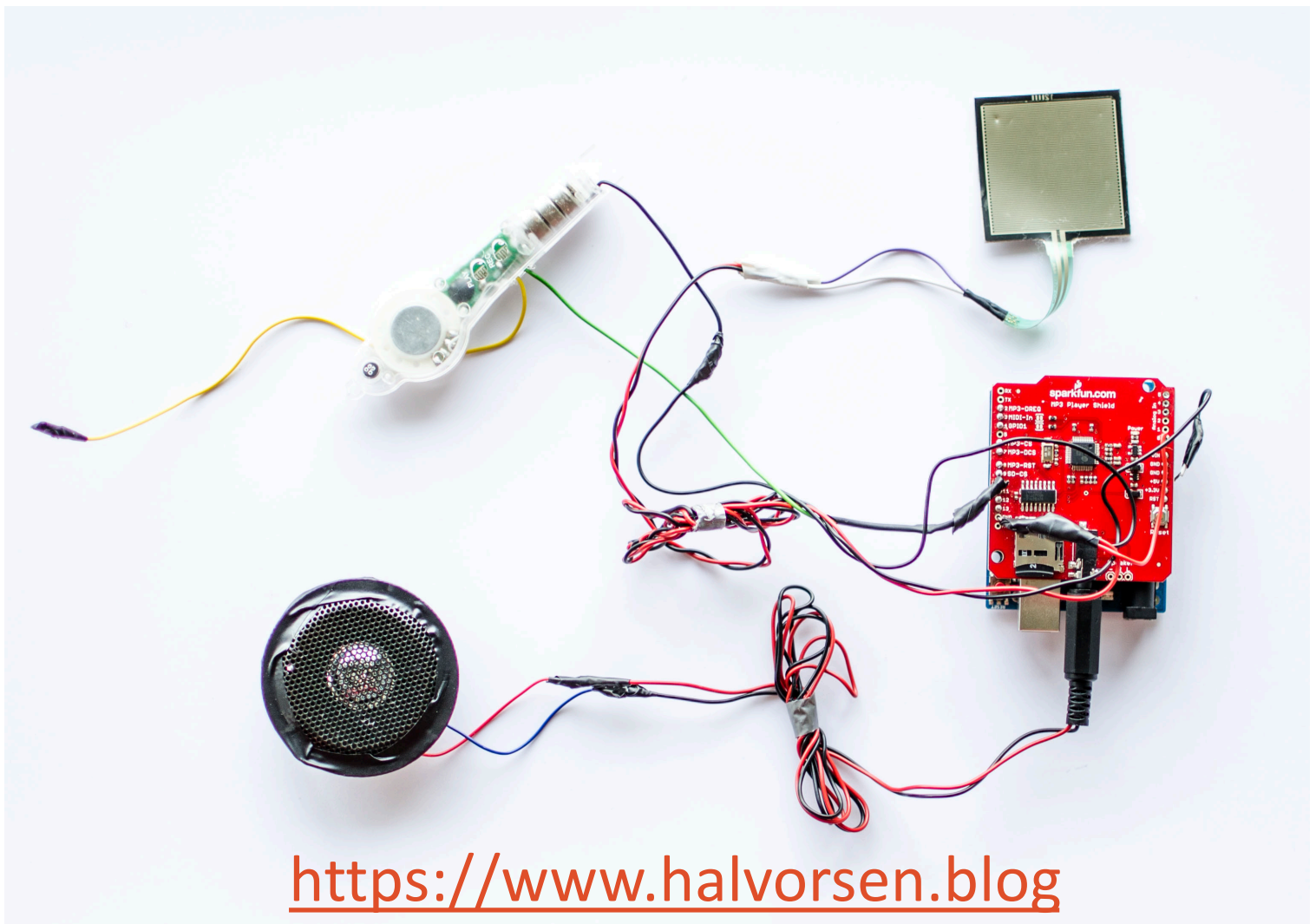


| Arduino Programming |

Hans-Petter Halvorsen



<https://www.halvorsen.blog>

| Arduino Programmering |

ISBN: 978-82-691106-2-3

Forlagsnummer: 978-82-691106

Hans-Petter Halvorsen

Copyright © 2018

Ved hjelp av dette opplæringsheftet vil du lære grunnleggende bruk av datamaskiner, grunnleggende elektronikk, sensorer, måleteknikk og programmering.

Vi vil også lære hvordan man kan kombinere software og hardware og hvordan vi kan kommunisere med omverdenen vha dataprogrammer og lage enkle prototyper.

I dette heftet vil vi bruke Arduino som en rød tråd til å lære disse tingene.



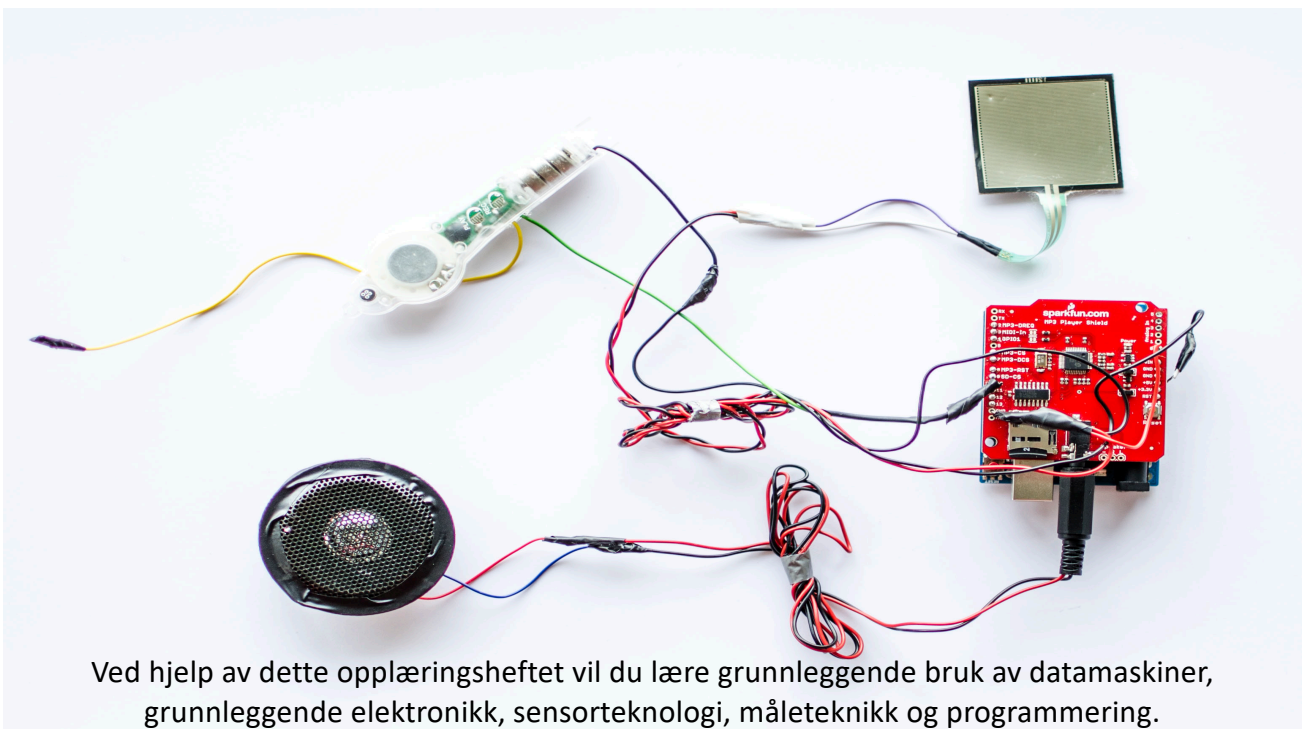
<https://www.halvorsen.blog>

<https://www.halvorsen.blog>

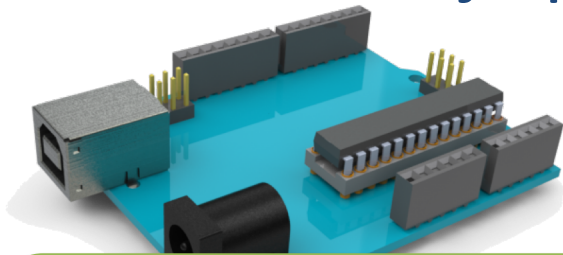


| Arduino Programmering |

| Hans-Petter Halvorsen |



Lær Teknologi og Programmering ved hjelp av Arduino

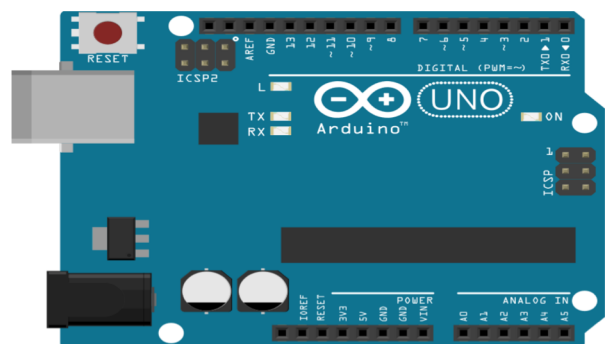


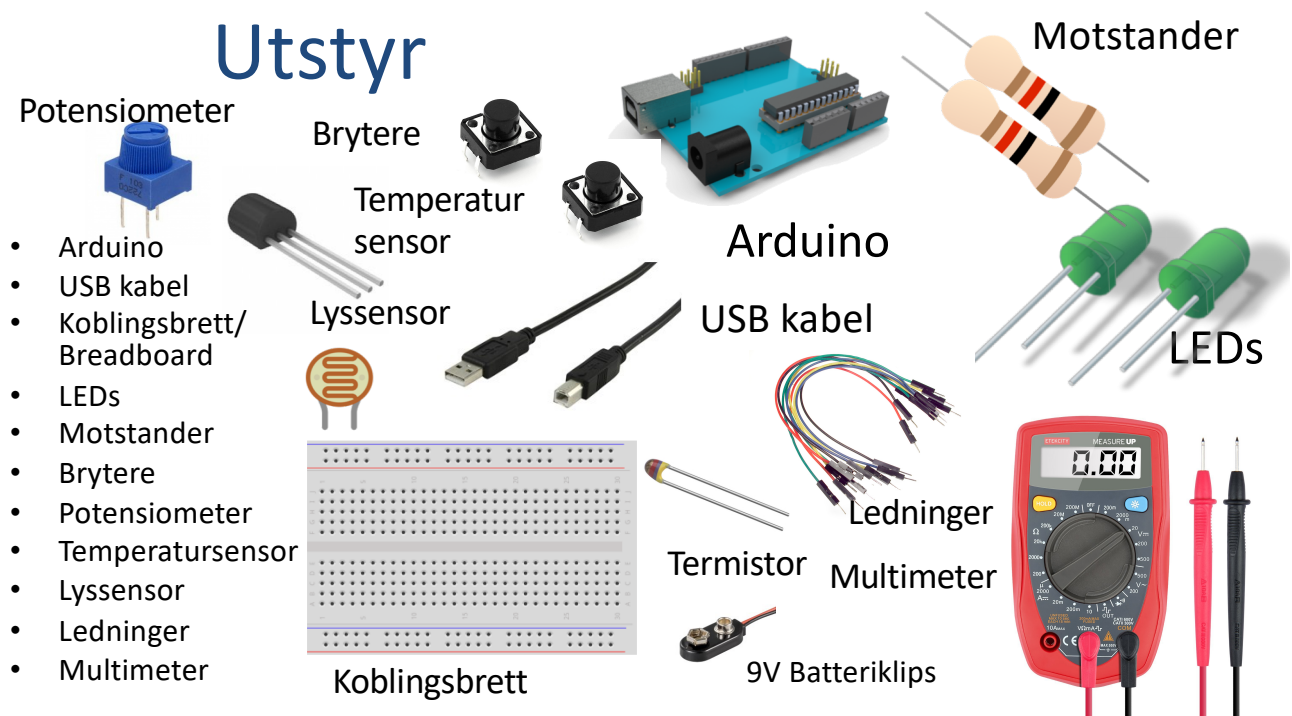
Arduino er en liten "datamaskin" som kan registrere og styre fysiske omgivelser og komponenter ved at man kobler den til ulike sensorer. Arduino er velegnet både for nybegynnere og for mer avansert bruk.

Hva trenger du?

For å komme i gang med Arduino trenger du følgende:

- PC (Windows, Mac, Linux)
- Arduino UNO (koster ca. 200 kr) eller en startpakke (ca. 800) med alt man trenger for å komme i gang
- Programvare (gratis)
- Diverse elektriske komponenter





Komponentliste

Her er eksempler på anbefalt utstyr:

Arduino:

- Arduino UNO, 1stk
- USB Kabel, 1stk
- Koblingsbrett, 1stk
- Ledninger, 10stk

Annet utstyr:

- Multimeter, 1stk

Elektronikk-komponenter:

- LED, rød, 2stk
- LED, grønn, 1stk
- Bryter, 2stk
- TMP36, 1stk
- Termistor, 1stk
- Lyssensor, 1st
- Potensiometer, 1stk
- Motstand 270 ohm, 2stk
- Motstand 330 ohm, 2stk
- Motstand 10k ohm, 2stk
- 9v Batteriklips, 1stk



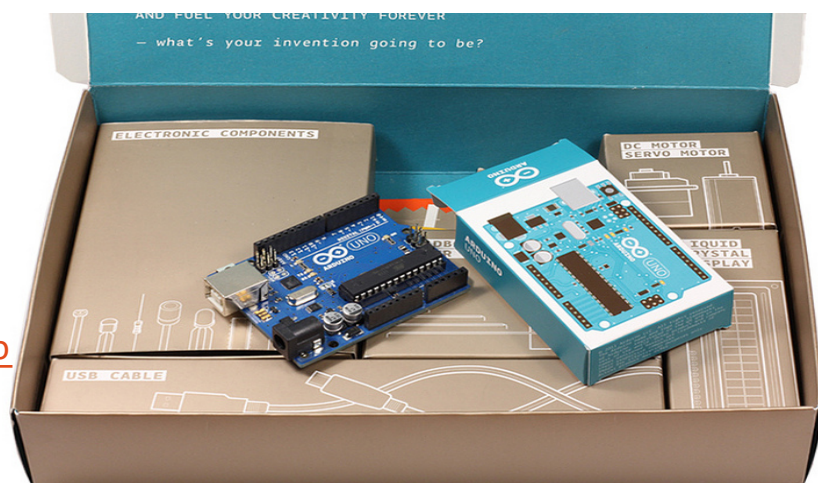
Ønsker du å kjøpe din egen Arduino?

Kjøp f.eks. et "Starter Kit" som inneholder alt du trenger

Disse kan kjøpes mange steder på nettet.

F.eks:
Kjell & Company:
<https://www.kjell.com/no>

Pris ca. 800-900,-



Innhold

De forskjellige delene kan gjennomgås/
leses uavhengig av hverandre



<https://www.halvorsen.blog>



Del 1: Introduksjon

| Hans-Petter Halvorsen |

<https://www.halvorsen.blog>



Introduksjon

| Hans-Petter Halvorsen |

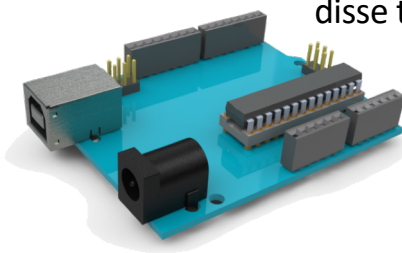


Introduksjon

Ved hjelp av dette opplæringsheftet vil du lære grunnleggende bruk av datamaskiner, grunnleggende elektronikk, sensorer, måleteknikk og programmering.

Vi vil også lære hvordan man kan kombinere software og hardware og hvordan vi kan kommunisere med omverdenen vha dataprogrammer og lage enkle prototyper.

I dette heftet vil vi bruke Arduino som en rød tråd til å lære disse tingene.



Arduino er en open source plattform for prototyping av elektronikk. Plattformen består av en familie med små "datamaskiner", et programmeringsspråk med tilhørende utviklingsmiljø (IDE), og en rekke sensorer, aktuatorer og "shields" som gjør at man kan bygge grensesnitt mot den fysiske verden.

Arduino

Rundt omkring oss er det massevis av små datamaskiner, såkalte mikrokontrollere, som utfører enkle oppgaver. De sitter i tidsbrytere, fjernkontroller, mikrobølgeovner, termostater, kortlesere og så videre. En mikrokontroller programmeres generelt for å utføre én bestemt oppgave. Arduino er en plattform for mikrokontrollere som gjør det lettere for programmereren å komme i gang.

Ved hjelp av Arduino kan en mikrokontroller programmeres til å utføre akkurat den oppgaven (eller oppgavene) som programmereren vil. Det fine med Arduino er at alt er basert på åpen kildekode. Hvem som helst kan bruke og forbedre maskinvaren og programvaren. Ved hjelp av Arduino kan man for eksempel bygge en robotgressklipper, en dørlås med RFID, en automatisk plantevanner eller en GPS-sporer, osv. Mulighetene er ubegrensede.

Maskinvare

Det du trenger for å komme i gang, er en datamaskin som kjører Windows, Mac eller Linux, pluss et Arduino utviklingskort. Et Arduino utviklingskort er et kretskort med en mikrokontroller og all annen elektronikk du trenger for å bruke kortet. Utviklingskort kobles til datamaskinen (via en USB kabel) og finnes i flere varianter, for eksempel Arduino UNO. Dette er i dag Arduinos referansekort, og de fleste utvidelser utvikles til nettopp UNO. En overkommelig pris og inn- og utganger nok for de fleste prosjekter gjør UNO til et godt valg for de fleste.

Utviklingskortet Arduino Uno har 14 digitale inn- og utganger, og disse kalles IO- (Input Output) eller GPIO-pinner (General Purpose Input Output). IO-pinnene brukes til å koble til digitalt tilbehør, for eksempel lysdioder og trykknapper.

Mikrokontroller

- Arduino er en mikrokontroller
- En mikrokontroller er en integrert datamaskin på en chip

En mikrokontroller (μC) er en integrert chip som ofte er en del av et innebygd system. Den inneholder en CPU, RAM, ROM, I/O-porter, mm. som en vanlig datamaskin. Fordi de er laget for å utføre kun én bestemt oppgave og å styre et enkelt system, er de mye mindre og forenklet slik at de kan inkludere alle funksjonene som kreves på en enkelt brikke. En mikroprosessor (μP), som er en mer generell chip som brukes til å lage en multi-funksjons datamaskin eller enhet, og krever flere chips for å kunne håndtere ulike oppgaver. Mikrokontrollere er ment å være mer selvstendige og uavhengige og brukes mye i såkalte embedded systemer.

TRY IT OUT!

Dette opplæringsheftet består av mange praktiske eksempler som du bør prøve ut underveis i ditt eget tempo.

Den beste måten å lære på er gjennom å koble selv og skrive koden selv fra bunnen av.

Play and Explore: Deretter er det lurt å "eksperimentere" litt selv, forandre på noen verdier, prøve å lage koden bedre, osv.

| Programmering og Arduino |

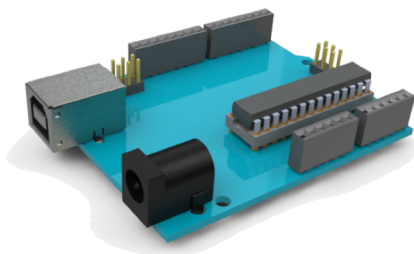
<https://www.halvorsen.blog>



Introduksjon til Arduino

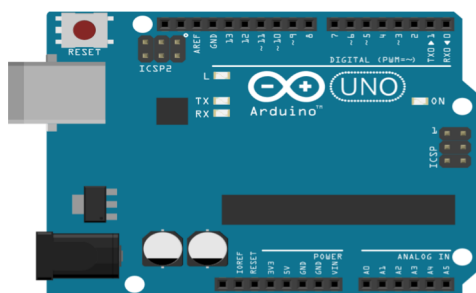
| Hans-Petter Halvorsen |

Arduino



Arduino er en "open source" plattform for "prototyping" av elektronikk. Plattformen består av en familie med små "datamaskiner", et programmeringsspråk med tilhørende utviklingsmiljø (IDE), og en rekke sensorer, aktuatorer og "shields" som gjør at man kan bygge grensesnitt mot den fysiske verden.

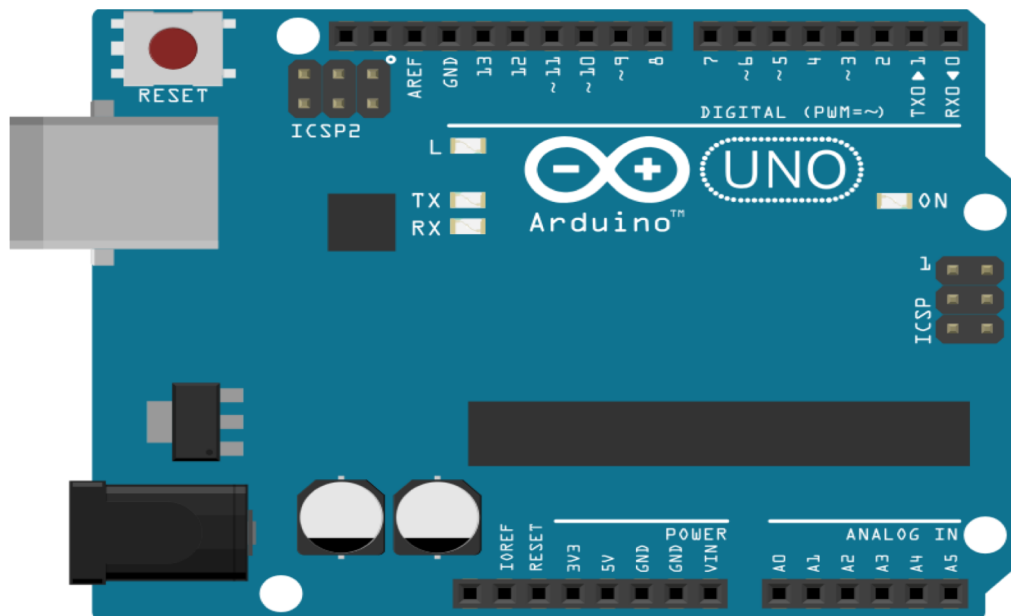
Det finnes mange forskjellige Arduino-brett. Det mest vanlige Arduino brettet er Arduino UNO.



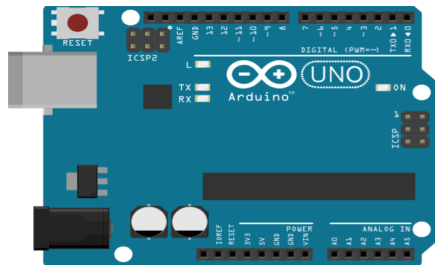
Les mer om de forskjellige typene på websiden til Arduino:

www.arduino.cc

Arduino UNO

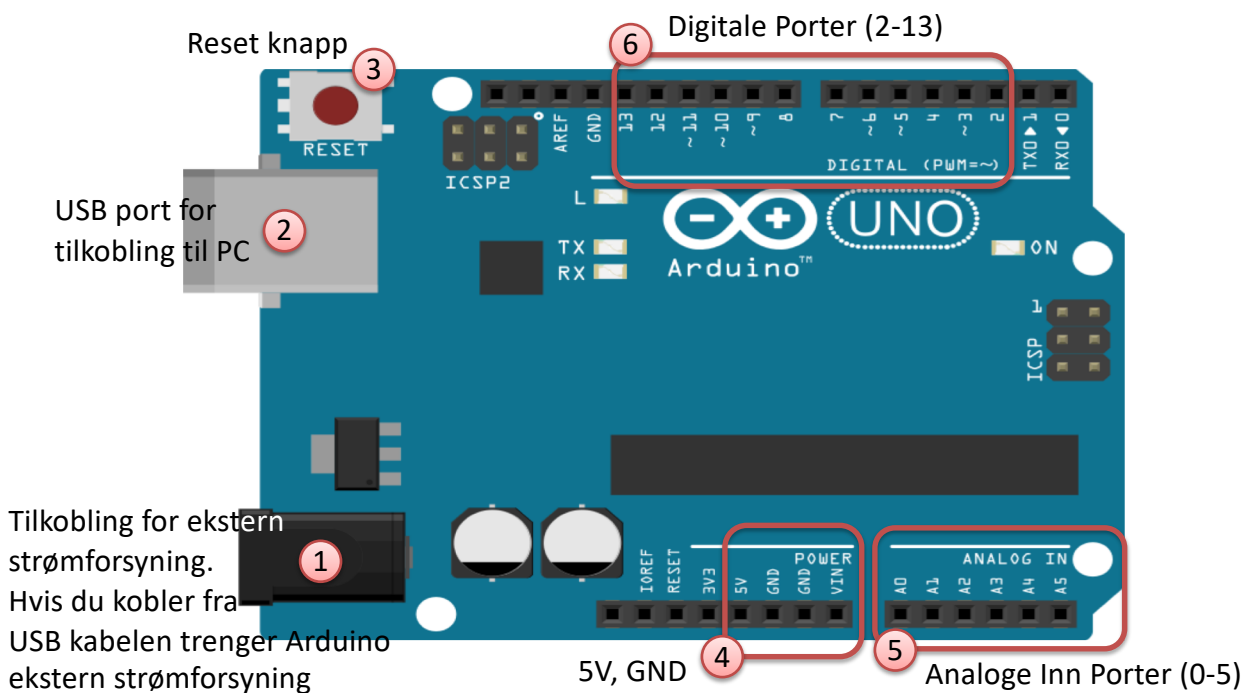


Arduino UNO



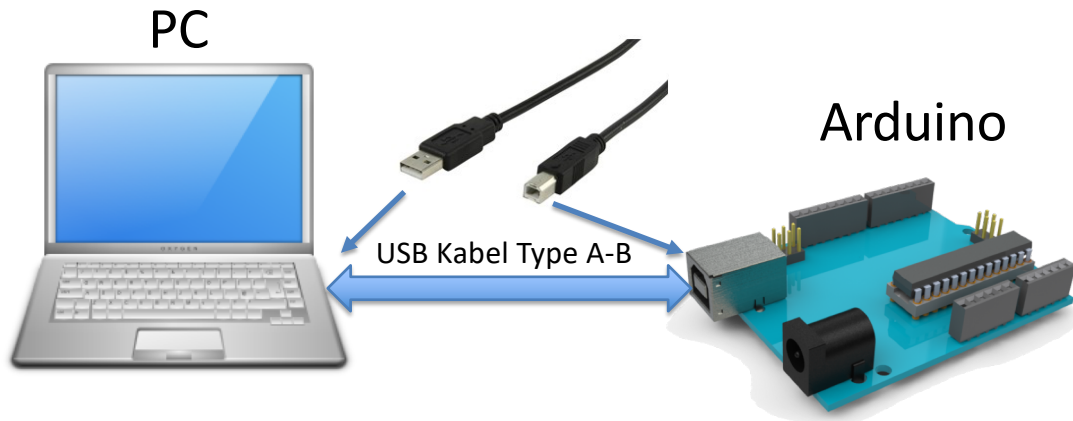
Tekniske spesifikasjoner

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g



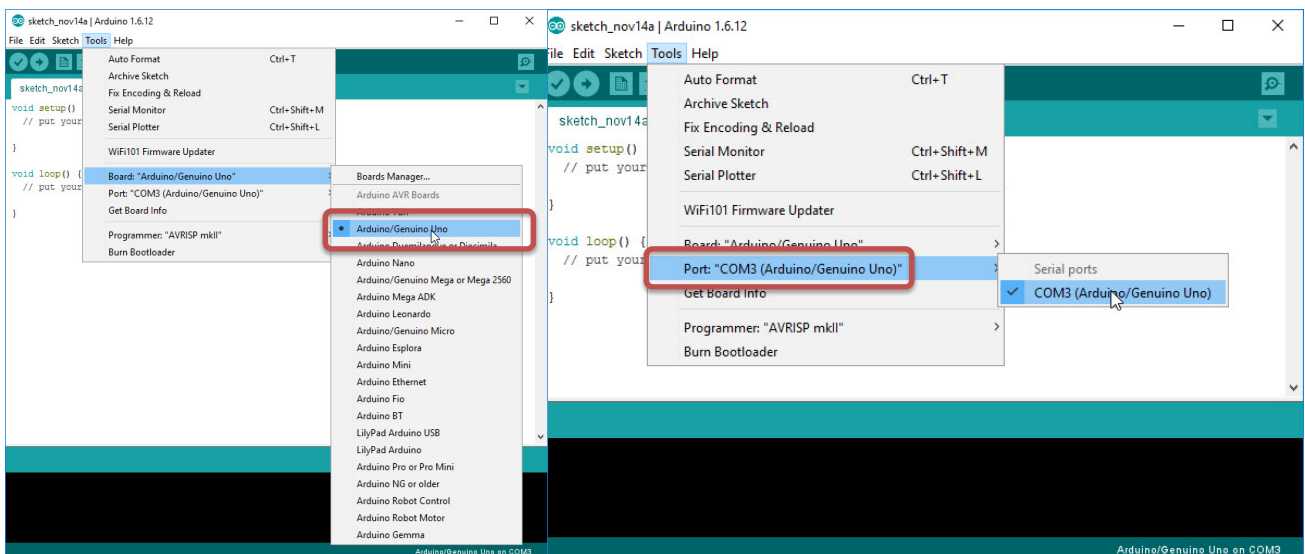
Fysisk Oppkobling

Slik kobler du Arduinoen til PCen:

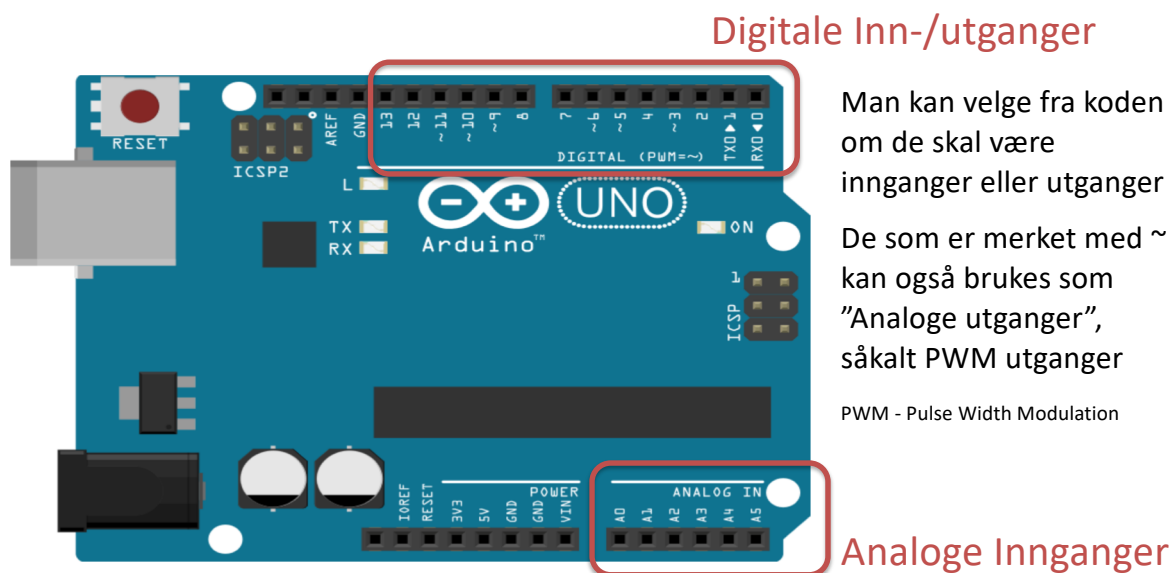


Arduino Software Oppkobling

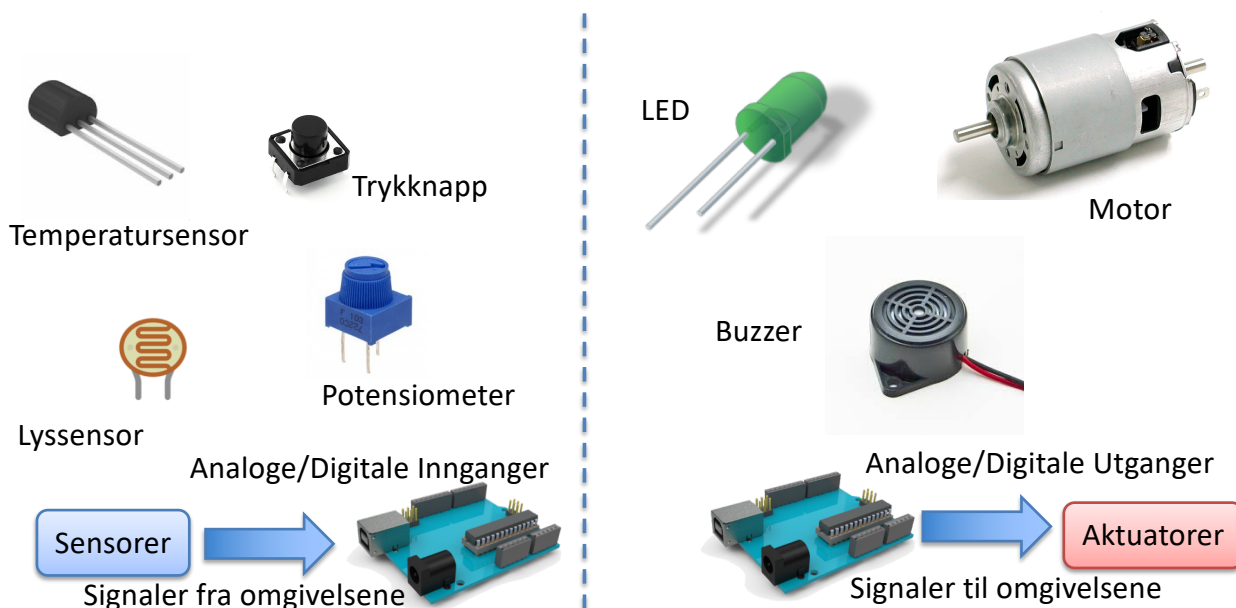
Velg riktig Kort (Arduino UNO) og serieport for riktig kommunikasjon



Innganger/Utganger (Analoge og Digitale)



Sensorer og Aktuatorer



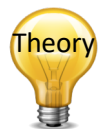
Sensorer og Aktuatorer

Arduino kortet kommuniserer med omverdenen via inn- og utganger. Disse kan være digitale eller analoge. Digitale inn- eller utganger er enten "høy" eller "lav", dvs "på" eller "av", og representeres som 5V eller 0V. Arduino kortet har ogs analoge innganger, dvs. at de kan lese en spenning mellom 0V og 5V, og gjør det om til et tall mellom 0 og 1023.

Sensorer er innretninger som kan forsyne Arduino med inn-signal. Vanlige sensorer er temperatursensorer, lysfølsomme sensorer, ulike bevegelsessensorer, ulike gass-sensorer osv. I prinsippet alt som kan gi Arduino beskjed om en tilstand i den fysiske verden via et elektrisk signal.

Aktuatorer er innretninger som kan motta signaler fra Arduino og gjøre det om til en mekanisk bevegelse. Eksempler på aktuatorer er servoer og andre motorer. Mange aktuatorer krever mer strøm og/eller spenning enn Arduino kan levere. Da trengs en eller annen form for forsterker som for eksempel releer, transistorer og driver-kretser.

Sensors and Actuators



- En **sensor** er en omformer som måler en fysisk størrelse og konverterer den til et signal som kan leses av et instrument, datainnsamlingsenhet, eller en Arduino.
Eksempler: temperatursensor, trykksensor, osv.
- En **aktuator** er en slags motor for å bevege eller styre en mekanisme eller et system. Den drives av en energikilde, typisk elektrisk strøm, hydraulisk fluidtrykk, eller lufttrykk, og omdanner denne energien til bevegelse.
Eksempler: Motor, Pumpe, Ventil, osv.

<http://en.wikipedia.org/wiki/Sensor>

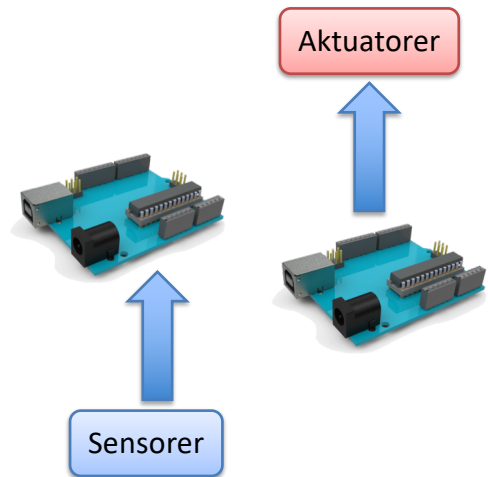
<http://en.wikipedia.org/wiki/Actuator>

Sensorer og Aktuatorer

Felles for sensorer og aktuatorer er at de kan være digitale eller analoge.

Noen sensorer og aktuatorer er ferdig tilpasset Arduino, mens andre må kobles opp i en eller annen krets for å kunne fungere riktig sammen med Arduino.

Mange av disse kommer med ferdige biblioteker for Arduino, slik at de er enkle å bruke.



Play and Explore

| Programmering og Arduino |

<https://www.halvorsen.blog>



Innføring i grunnleggende Elektronikk

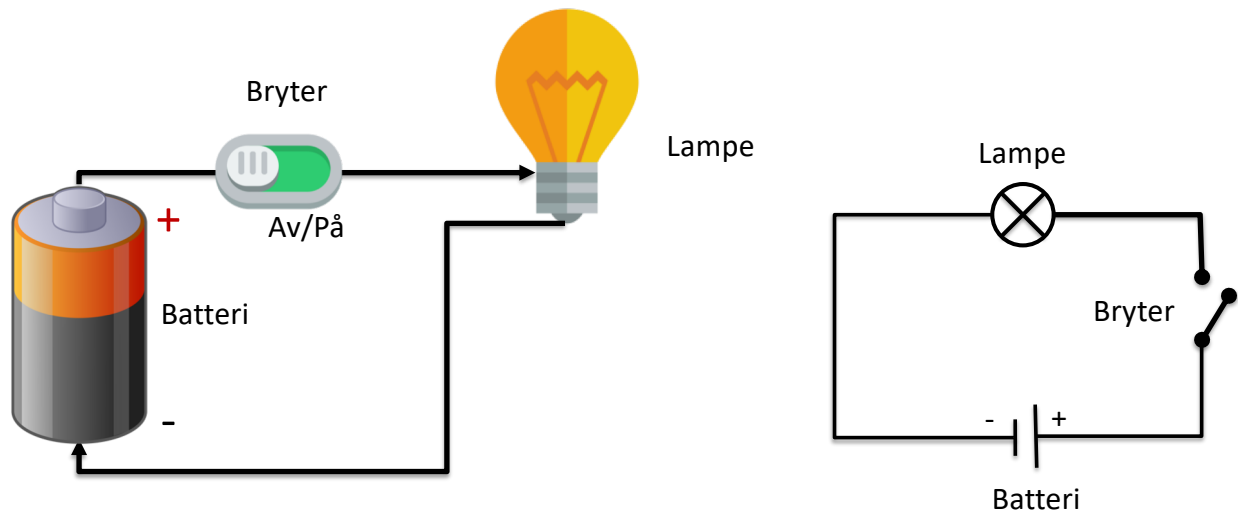
| Hans-Petter Halvorsen |

Elektrisk krets



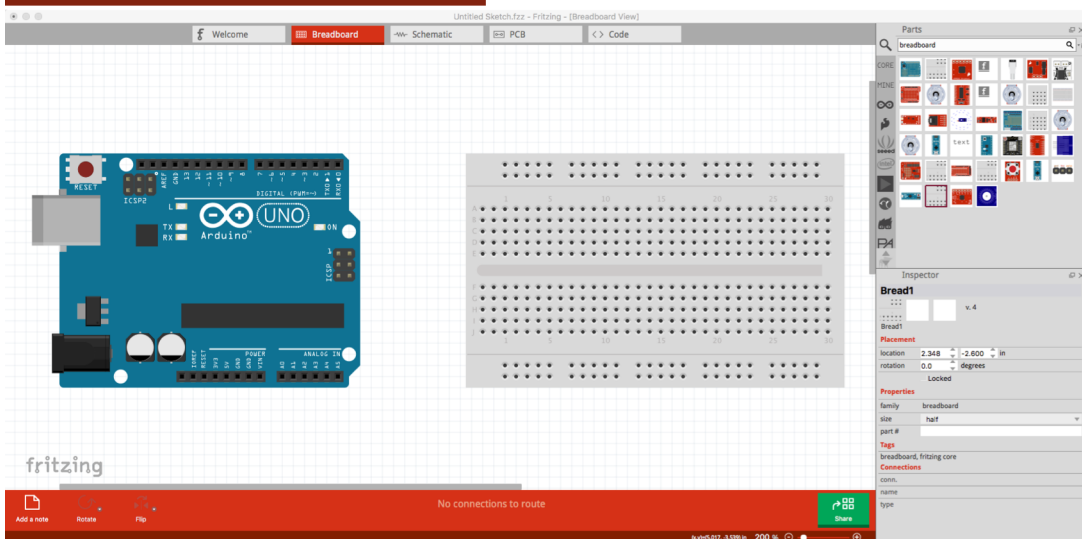
For at det skal gå strøm, må man koble en leder fra positiv side på batteriet, via en elektrisk komponent som bruker strømmen på veien, og til negativ side på batteriet

Elektrisk krets med bryter



Fritzing er Open Source software for tegning av elektriske kretser, m.m.

[<http://www.fritzing.org>]



Kortslutning



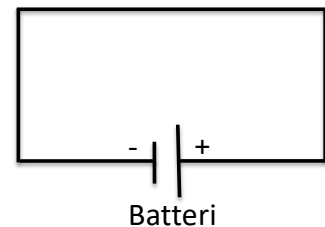
Vi må aldri koble sammen positiv og negativ side på en strømkilde uten at vi har en elektrisk komponent i mellom.

Dersom man gjør det kalles det en kortslutning.

Kortslutter man f.eks. et batteri, vil batteriet bli veldig varmt og batteriet blir tomt veldig raskt.

Enkelte batterier kan også begynne å brenne.

Kortslutning!!



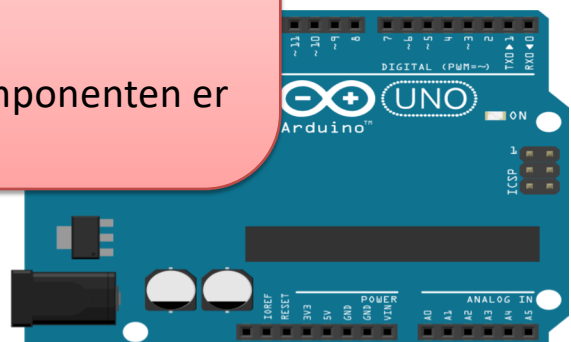
Kortslutning - Arduino



Dersom vi kortslutter portene på Arduinoen kan det i verste fall begynne å ryke fra den.

Når det begynner å ryke fra elektriske komponenter, skjer det fordi den har blitt for varm.

I de aller fleste tilfeller betyr det at komponenten er ødelagt.



Ohms Lov

Ohms lov er som følger:

$$U = RI$$



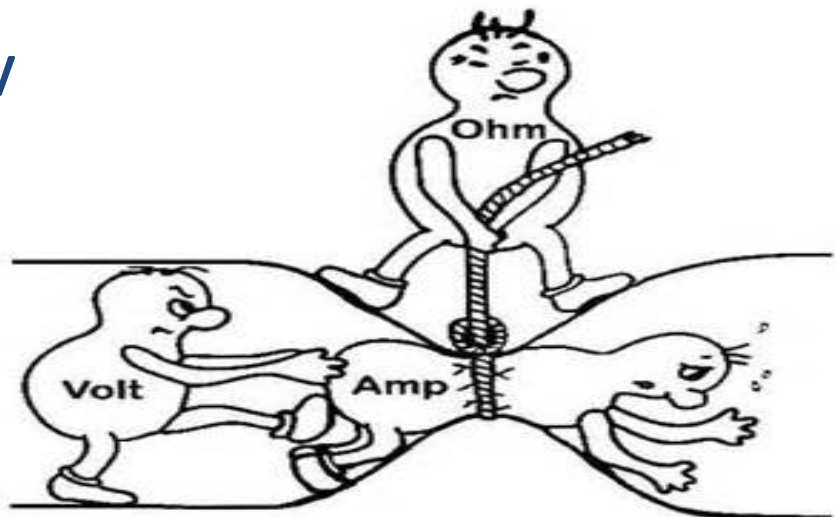
Ohms lov beskriver sammenhengen mellom elektrisk spenning, motstand og strømstyrke

U – Spenning, måles i Volt [V]
 R – Motstand, Måles i Ohm [Ω]
 I – Strøm, Måles i Ampere [A]

$$R = \frac{U}{I} \quad I = \frac{U}{R}$$

Ohms Lov

$$U = RI$$



Spenningen er den kraften som skal til for å flytte strømmen gjennom et kretsløp. Motstanden forteller oss hvor mye materialet elektrisiteten beveger seg gjennom bremses strømmen. Strømstyrken forteller oss hvor mange elektroner som beveger seg per sekund.

Multimeter

Et multimeter brukes til å måle strøm, spenning, m.m. i en elektrisk krets.

Vi vil bruke multimeter til å måle spenning og strøm i forbindelse med koblingene vi lager for Arduino, måle verdier på motstander, osv.



<https://learn.sparkfun.com/tutorials/how-to-use-a-multimeter>



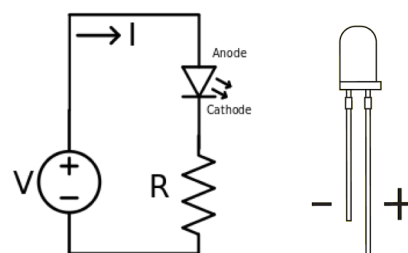
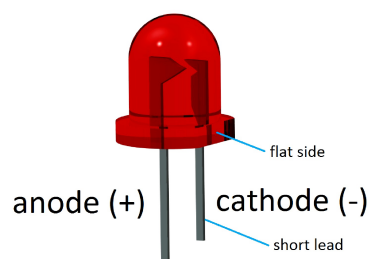
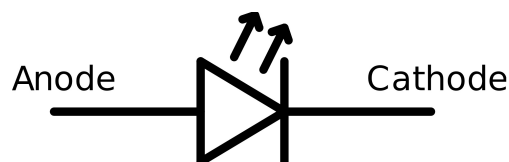
LED

En lysdiode (eller LED, fra engelsk light-emitting diode) er en halvleder-diode som utstråler lys under tilførsel av elektrisk strøm.

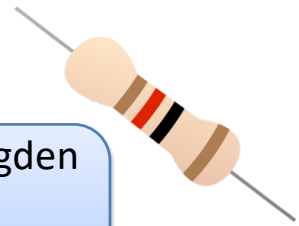
Hvis man setter på en for stor strøm så vil lysdioden brenne ut og bli ødelagt, for å forhindre dette bruker man en strømbegrensende motstand som setter strømmen igjennom lysdioden til ønsket verdi og passer på at spenningen er riktig.

Merk! Strømmen må gå fra pluss (lang pinne) til minus (kort pinne), ellers vil ikke lysdioden lyse.

[Wikipedia]



Motstand (Resistor)




En resistor eller motstand brukes til begrense mengden strøm som går gjennom en komponent. Ikke alle komponenter tåler like mye strøm, derfor kan vi bruke en motstand til å justere dette.

Motstand måles i Ohm (Ω)

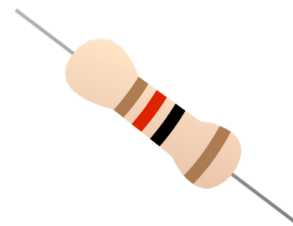
Motstander finnes i forskjellige størrelser, f.eks 220Ω , 270Ω , 330Ω , $1k\Omega$ $10k\Omega$, ...

Man kan finne ønsket motstand vha Ohms lov $U = RI$

Elektrisk symbol: 

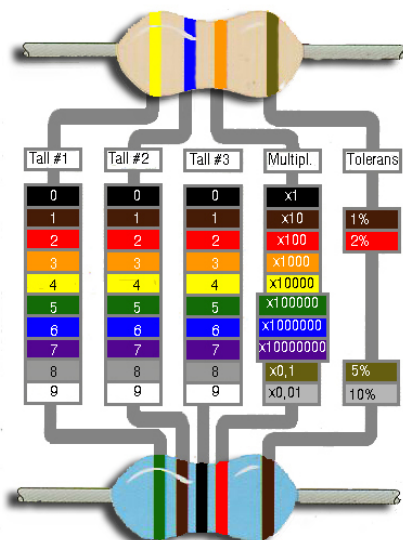
Fargekoder for motstander

Fargekoder for motstander					
Farge	1. Ring	2. (3.) Ring	3. (4.) Ring	4. (5.) Ring	6. Ring
	Tallverdi	Tallverdi	Multiplikator	Toleranse	Temperaturkoeffisient
Svart	0	0	* 1	20 %	200 ppm/K
Brun	1	1	* 10	1 %	100 ppm/K
Rød	2	2	* 100	2 %	50 ppm/K
Orange	3	3	* 1000	3 %	15 ppm/K
Gul	4	4	* 10.000	0...+100 %	25 ppm/K
Grønn	5	5	* 100.000	0,5 %	
Blå	6	6	* 1.000.000	0,25 %	10 ppm/K
Fiolett	7	7	* 10.000.0000	0,1 %	5 ppm/K
Grå	8	8	* 0,01		1 ppm/K
Hvit	9	9	* 0,1		
Sølv			* 0,01	10 %	
Gull			* 0,1	5 %	



<http://w3.elektrofag.info/elektronikk/resistoren/fargekoder>

Fargekoder for motstander



Gul = 4
 Blå = 6
 Orange = x1000
 Gull = 5%

Dette gir $R = 46000\Omega$,
 eller $R = 46\text{ k}\Omega$ med en toleranse på 5%.
 Toleransen forteller hvor stor margin det er på motstanden.

Huskereglen for 5 streker er som følger: Sett de tre første tallene sammen (510), multipliser dette med fjerde tallet som er multiplikatoren (x100), for så å oppnå motstandens verdi. Siste fargen var i dette tilfellet brun, noe som forteller oss at motstandsverdien har en toleranse på kun 1%.

Grønn = 5
 Brun = 1
 Svart = 0
 Rød = x100
 Brun = 1%

Dette gir $R = 51000\Omega$, eller $R = 51\text{ k}\Omega$ med en toleranse på 1%.

http://www.tek.no/artikler/elektronikkens_verden_-_del_1/97978/3

Fargekoder for motstander

TRY IT OUT!

Finn verdien på dine motstander ved hjelp av fargekodene på motstandene.

Bruk også et multimeter for å se om du får samme resultat.



Fargekoder for motstander TRY IT OUT!

Finn verdien på dine motstander ved hjelp av fargekodene på motstandene.

Bruk en "Motstandskalkulator" du finner mange steder på nettet, f.eks denne:



<http://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>

Seriekobling og Parallellkobling av motstander

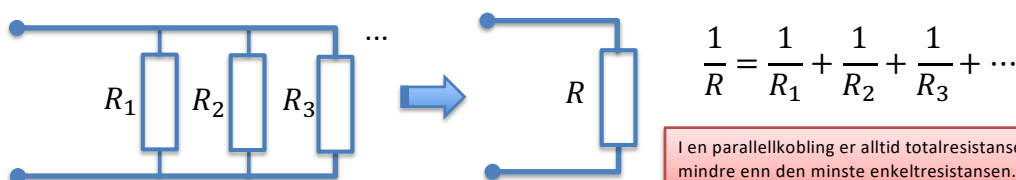
Seriekobling:



I en seriekrets deler spenningen seg, men strømmen er lik. I en krets der alle motstandene er like vil spenningen over hver motstand være gitt av den totale spenningen delt på antall motstander. Jo flere motstander i kretsen, jo flere ganger må spenningen deles og jo lavere spenning blir det over hver motstand.

Parallellkobling:

Merk: I en seriekobling er summen av delspenningene lik spenningen på spenningskilden.



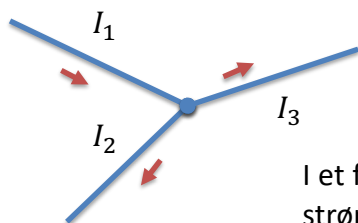
I en parallellkobling er alltid totalresistansen mindre enn den minste enkeltresistansen.

En parallellkobling er en kobling av elektriske komponenter ved siden av hverandre i en krets slik at spenningen over alle komponentene er lik. Strømmen inn til en parallellkobling vil være lik summen av strømmene i de enkelte komponenter.

Kirchhoffs lover

Kirchhoffs strømlov:

$$\sum_{k=1}^n I_k = 0$$



$$I_1 = I_2 + I_3$$

$$I_1 - I_2 - I_3 = 0$$

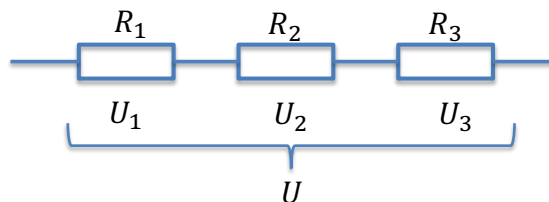
I et forgreningspunkt er summen av alle inngående strømmer lik summen av alle utgående strømmer siden ladning ikke kan hope seg opp i forgreningspunktet.

Kirchhoffs spenningslov:

Summen av alle elektriske potensialforskjeller (spenninger) i enhver lukket strømsløyfe er lik null.

$$\sum_{k=1}^n U_k = 0$$

$$U = U_1 + U_2 + U_3 + \dots$$

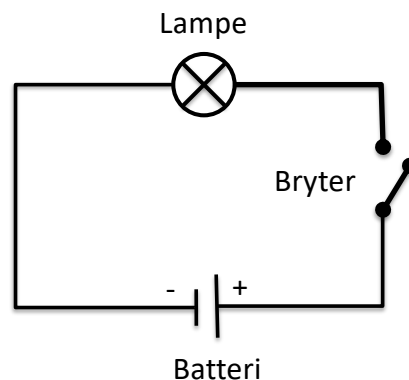


Bryter

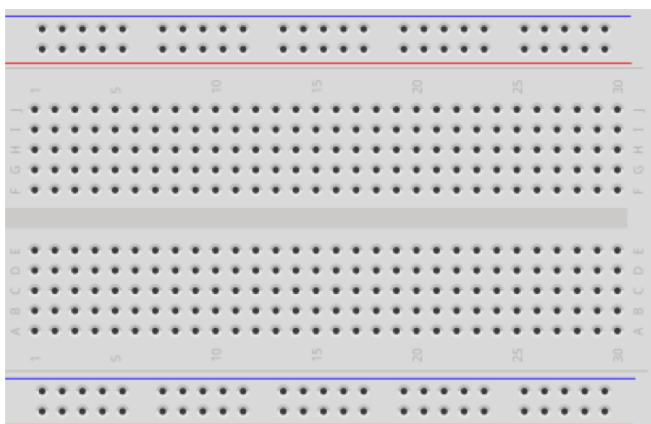
Den bryter flyten av strøm gjennom en krets når den er åpen. Når den er lukket vil strømmen flyte uhindret gjennom kretsen.



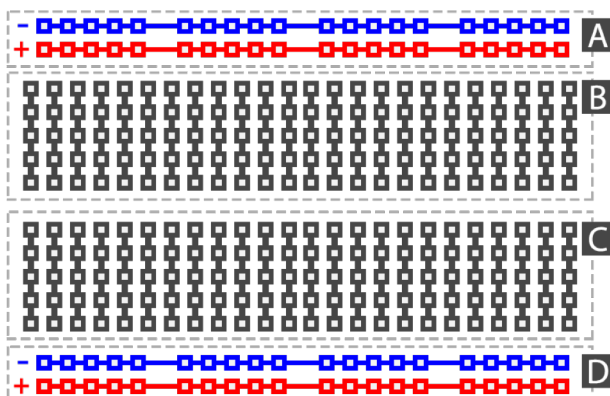
Disse finnes i mange former og fasonger



Koblingsbrett (Breadboard)

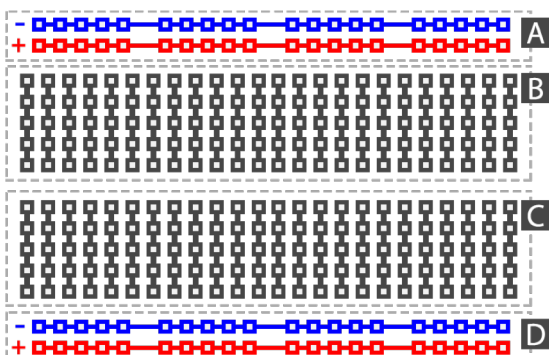


Brukes til å koble opp komponenter og elektriske kretser

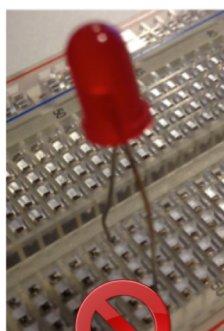


Koblingsbrett - Riktig oppkobling

Pass på at du ikke kortslutter komponentene som du kobler på koblingsbrettet



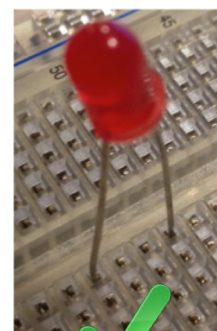
Ikke kortslutt komponentene dine



Kople over bruen på eksperimentkortet...



... eller på tvers av rader



Play and Explore

| Programmering og Arduino |

<https://www.halvorsen.blog>



Innføring i grunnleggende Programmering

| Hans-Petter Halvorsen |

Programmeringsspråk

Her er noen eksempler:

- C
- C++
- C#
- Java
- JavaScript
- LabVIEW
- ... Det finnes tusenvis av forskjellige programmeringsspråk

Programmeringsmiljø

Her er noen eksempler:

- Visual Studio
 - Programmeringsspråk: C++, C#, Visual Basic, ..
- XCode
 - Programmeringsspråk: Objective-C, Swift
- Android Studio
 - Programmeringsspråk: Java
- LabVIEW
 - Programmeringsspråk: LabVIEW (kalles noen ganger for "G")
- Arduino IDE
- ...

Programmering

Alle Programmeringsspråk har følgende grunnfunksjonalitet

- Variable
 - $x = 3$
- Betingelser
 - If .. Else
- Løkker
 - For løkker
 - While løkker
- Funksjoner
 - `writeAnalog()`

Variable

Eksempler:

```
...  
  
int variable1 = 0;  
int variable3 = 0;  
int variable3 = 0;  
  
variable1 = 2;  
variable2 = 3;  
  
variable3 = variable1 + variable2;
```

Dat typer:
int - Heltall
float – Flyttall
(tall med komma)
...

Betingelser

If – betingelse:

```
...
if(variable < 0)
{
    variable = 0;
}
```

Merk!

```
...
if(variable1 == 5)
{
    variable2 = 3;
}
```

If – Else - betingelse:

```
...
if(input == HIGH)
{
    digitalWrite(11, HIGH);
}
else
{
    digitalWrite(12, HIGH);
}
```

Betingelser

If – Else If - Else - betingelse:

Eksempel:

```
...
if(input == 1)
{
    digitalWrite(4, HIGH);
}
else if (input == 2)
{
    digitalWrite(5, HIGH);
}
else
{
    digitalWrite(6, HIGH);
}
```

Løkker

For løkke:

```
...  
for (int i=0; i<7; i++)  
{  
    ... // Do something  
}
```

While løkke:

```
...  
x = 0;  
while (x < 0)  
{  
    x++;  
}
```

```
...  
x = 0;  
do  
{  
    x++;  
} while (x < 5)
```

Do – While Løkke

Funksjoner

Eksempel:

```
void loop() //Hovedprogram  
{  
    z = calculate(2,3); //Her bruker vi funksjonen  
}  
  
float calculate(int x, int y) //Funksjonsdefinisjon  
{  
    return (x + y);  
}
```


Play and Explore

| Programmering og Arduino |

<https://www.halvorsen.blog>



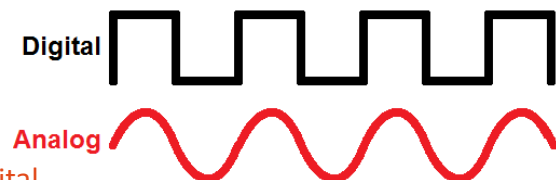
Analoge og Digitale Signaler

| Hans-Petter Halvorsen |

Analoge vs. Digitale Signaler

Det finnes to typer signaler som frakter informasjon: Analoge og digitale signaler

Forskjellen mellom analoge og digitale signaler er at analog er en kontinuerlig elektrisk signal, mens digital er en ikke-kontinuerlig elektrisk signal.

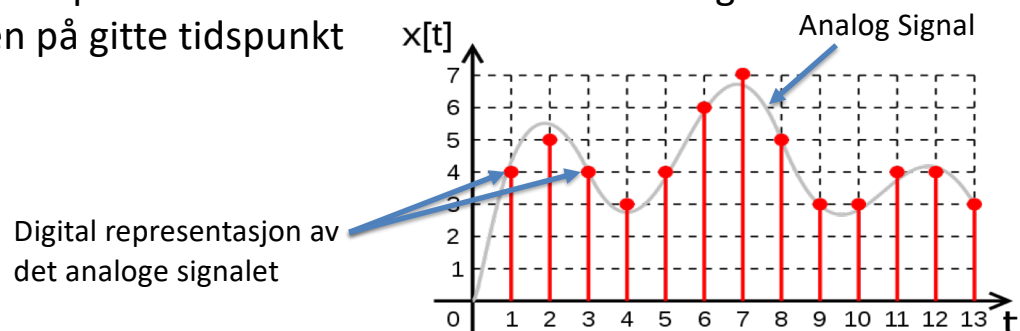


<https://learn.sparkfun.com/tutorials/analog-vs-digital>

Analoge Signaler

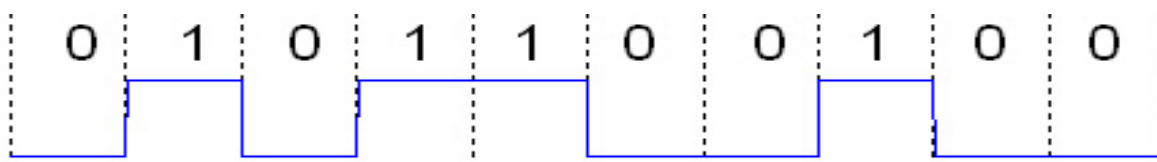
Analogt signal, signal, i form av for eksempel strøm og spenning, hvor styrken representerer andre fysiske størrelser (lydtrykk, lysstyrke osv.).

Et alternativt signalformat er digitalt signal hvor den fysiske parameteren representeres ved en rekke tall som angir signalstyrken på gitte tidspunkt

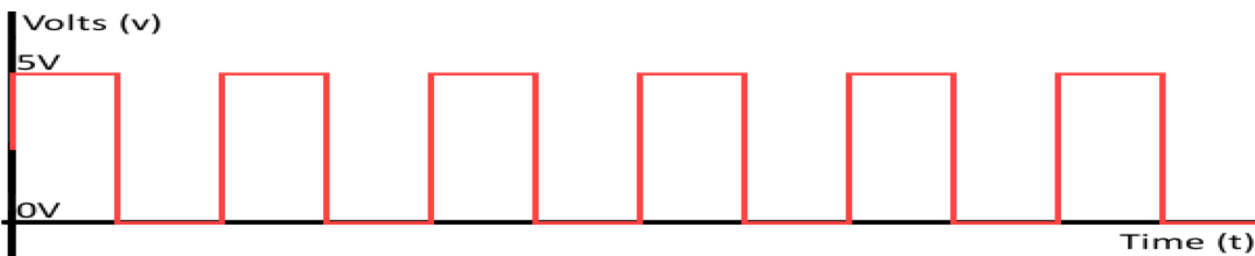


Digitale Signaler

Kun 2 forskjellige verdier: "0" og "1"



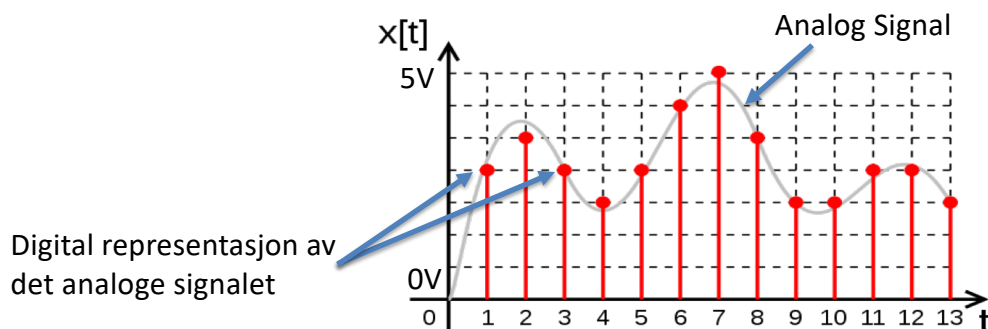
Arduino bruker 5V som "1" (True) og 0V som "0" (False)



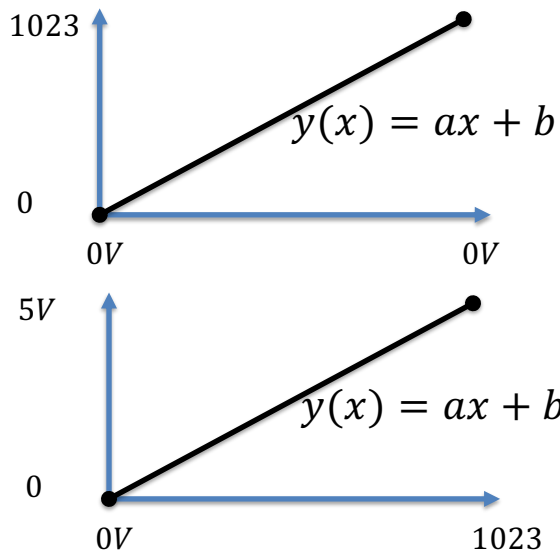
Analog til Digital konvertering

Analogt Signal ifm Arduino: 0-5V

Arduino konverterer dette analoge signal til en verdi mellom 0 og 1023 vha en 10bits ($2^{10} = 1024$) Analog til Digital konverter.



Analog til Digital konvertering



$$\begin{array}{l} 0V \rightarrow 0 \\ 5V \rightarrow 1023 \end{array} \quad y(x) = \frac{1023}{5}x$$

$$1.5V? \quad y(1.5) = \frac{1023}{5}1.5 = 307$$

$$\begin{array}{l} 0 \rightarrow 0V \\ 1023 \rightarrow 5V \end{array} \quad y(x) = \frac{5}{1023}x$$

$$400? \quad y(400) = \frac{5}{1023}400 = 1.96V$$

Play and Explore

| Programmering og Arduino |

<https://www.halvorsen.blog>



Arduino Utviklingsmiljø

| Hans-Petter Halvorsen |

Arduino Programvare

```
Blink | Arduino 1.6.10  
Blink $  
  
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW  
  delay(1000);           // wait for a second  
}
```

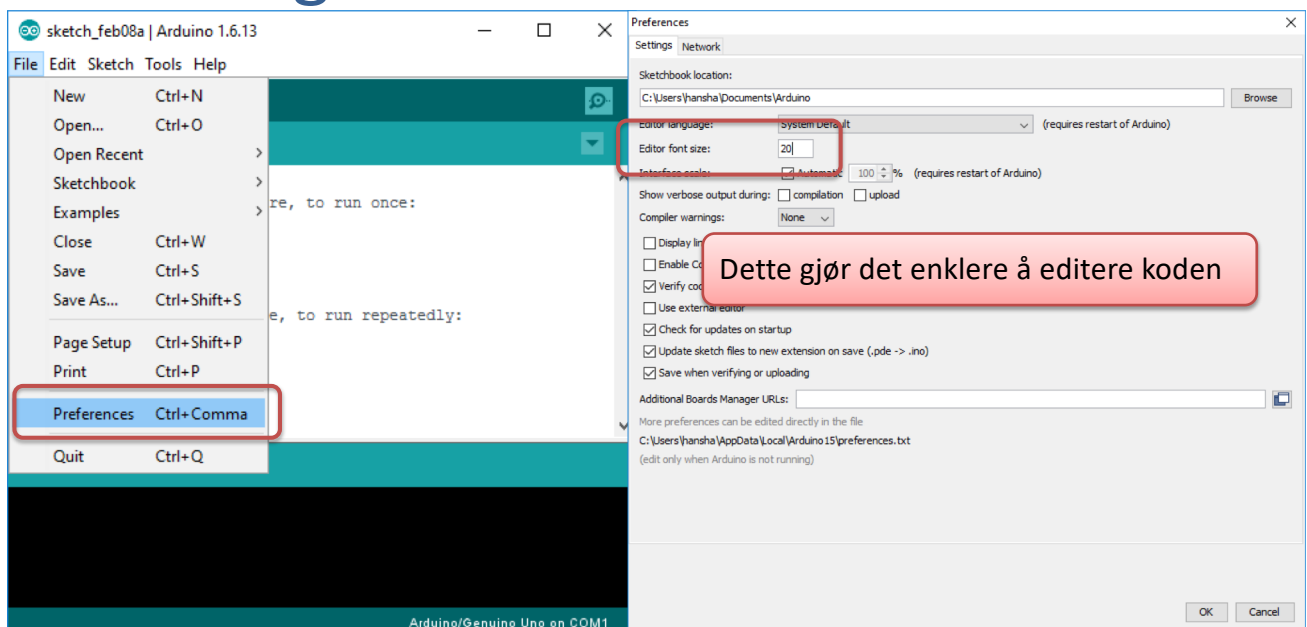
2 Arduino/Genuino Uno on /dev/cu.usbmodem1421

www.arduino.cc

Arduino IDE



Velg Skriftstørrelse i Editoren

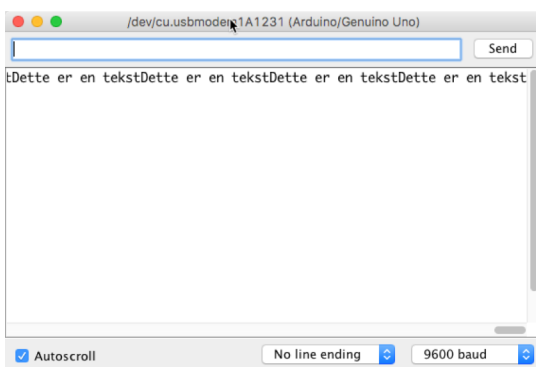


Serial Monitor

TRY IT OUT!

Du bruker Serial Monitor til å feilsøke Arduino programmer eller for å vise data/verdier fra programmet ditt til brukeren.

Du må ha en Arduino tilkoblet via USB til datamaskinen for å kunne aktivere Serial Monitor.

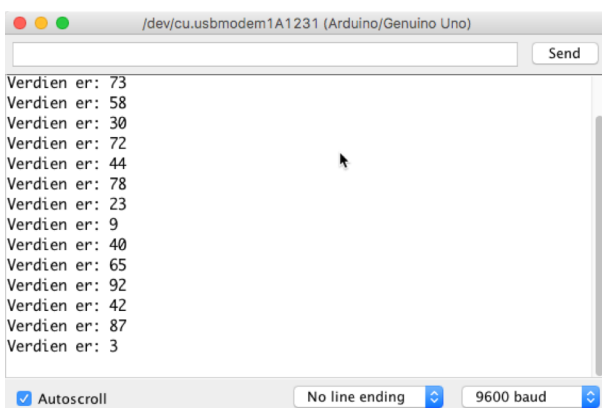


```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.print("Dette er en tekst");
  delay(1000);
}
```

Serial Monitor

TRY IT OUT!



Her ser vi hvordan vi kan skrive en verdi fra en variabel til Serial Monitor. Dette kan være en verdi fra en sensor, f.eks. en Temperatursensor.

```
int sensorValue = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  sensorValue = random(100);
  Serial.print("Verdien er: ");
  Serial.println(sensorValue);
  delay(1000);
}
```

Play and Explore

| Programmering og Arduino |

<https://www.halvorsen.blog>



Arduino Programmering

| Hans-Petter Halvorsen |

Arduino Programmer

Alle Arduino-programmer må følge følgende Hovedstruktur:

```
// Initialisering, definere variable, m.m.  
  
void setup()  
{  
    // Initialisering  
    ...  
}  
  
void loop()  
{  
    //Selve Hovedprogrammet som går i evig løkke  
    ...  
}
```

Arduino Program - Eksempel

```
void setup()  
{  
    pinMode(11, OUTPUT);    //Set the Pin as an Output  
}  
  
void loop()  
{  
    digitalWrite(11, HIGH); // Turn on the LED  
    delay(1000);           // Wait for one second  
    digitalWrite(11, LOW); // Turn off the LED  
    delay(1000);           // Wait for one second  
}
```

Arduino Program - Kommentarer

```
void setup()
{
    pinMode(11, OUTPUT);    //Set the Pin as an Output
}

void loop()
{
    digitalWrite(11, HIGH); // Turn on the LED

    /*
    ... alt mellom disse er kommentarer og vil ikke bli
    utført av programmet
    ...
    */
}
```

Bruk av funksjoner

```
int z;

void setup()
{
}

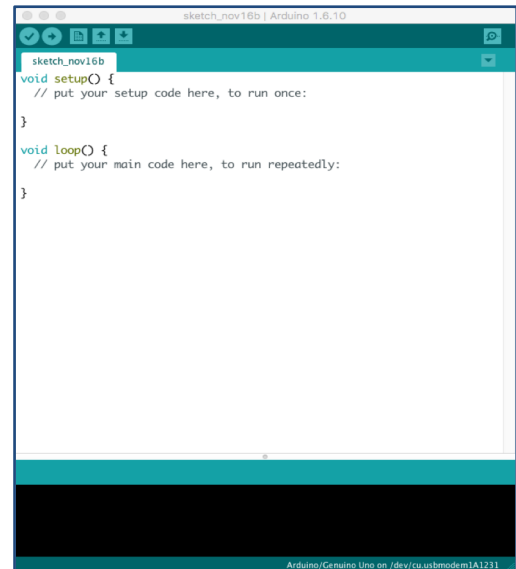
void loop()
{
    z = calculate(2,3);
}

float calculate(int x, int y)
{
    return (x + y);
}
```

TRY IT OUT!

Nå kommer noen enkle Arduino oppgaver som du skal prøve ut.

Sørg for at din Arduino enhet er koblet til PC-en og start opp Kode-editoren.



"Hello World" eksempel

TRY IT OUT!

Lag og test ut følgende "Hello World" program.

Åpne "Serial Monitor" for å se resultatet av programmet.

```
void setup()
{
  Serial.begin(9600);

  Serial.println("Hello, world!");
}

void loop()
{
}
```

”Hello World” eksempel

[TRY IT OUT!](#)

Lag og test ut følgende
”Hello World” program.

Åpne ”Serial Monitor”
for å se resultatet av
programmet.

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.println("Hello, world!");
  delay(1000);
}
```

Eksempel

I dette eksemplet bruker
vi variable, innebygde
funksjoner, vi lager og
bruker en egendefinert
funksjon, bruk av
kommentarer, samt at vi
skriver verdier til Serial
monitor

```
int z;int a;int b;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  a = random(100);
  b = random(100);
  z = calculate(a,b); //Vi summerer 2 tall

  //skriver verdier til Serial Monitor
  Serial.print(a);
  Serial.print(" + ");
  Serial.print(b);
  Serial.print(" = ");
  Serial.println(z);

  delay(1000);
}
float calculate(int x, int y)
{
  return (x + y);
}
```

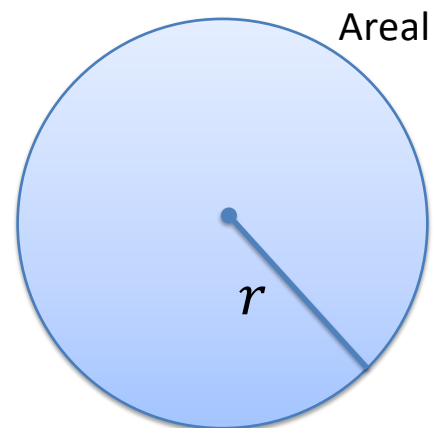
[TRY IT OUT!](#)

Oppgave om funksjoner

[TRY IT OUT!](#)

Lag en funksjon som beregner arealet til en sirkel med en gitt radius.

Skriv svaret til Serial Monitor.



Løsningsforslag

```
void setup()
{
  float area;

  Serial.begin(9600);
  // calculate the area of a circle with radius of 9.2
  float r=9,2;
  area = CircleArea(r);
  Serial.print("Area of circle is: ");
  // print area to 4 decimal places
  Serial.println(area, 4);
}

void loop()
{
}

// calculate the area of a circle
float CircleArea(float radius)
{
  float result;
  const float pi = 3.14;

  result = pi * radius * radius;

  return result;
}
```

[TRY IT OUT!](#)

Eksempel

I dette programmet bruker vi en for løkke til å summere 100 tilfeldige tall.

Deretter finner vi gjennomsnittet.

Sum og Gjennomsnitt skrives til Serial Monitor.

```
int x; int sum = 0; float gjennomsnitt = 0;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  sum = 0;
  for (int i = 0; i<100; i++)
  {
    x = random(100);
    sum = sum + x;
  }

  gjennomsnitt = sum / 100;
  Serial.print(" Sum = ");
  Serial.print(sum);
  Serial.print(" ,
  Gjennomsnitt = ");
  Serial.println(gjennomsnitt);
  delay(1000);
}
```

TRY IT OUT!

Eksempel

Her viser vi bruk av tabeller (arrays) i Arduino programmer

```
const int arraysize = 100;
int x;
int sum = 0;
float gjennomsnitt = 0;
int myarray[arraysize];

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  sum = 0;
  for (int i = 0; i<arraysize; i++)
  {
    x = random(200);
    myarray[i] = x;
  }
  sum = beregnsum(myarray);
  gjennomsnitt = sum / 100;
  Serial.print(" Sum = ");
  Serial.print(sum);
  Serial.print(" , Gjennomsnitt = ");
  Serial.println(gjennomsnitt);
  delay(1000);
}

int beregnsum (int sumarray[])
{
  for (int i = 0; i < arraysize; i++)
  {
    sum = sum + sumarray[i];
  }
  return sum;
}
```

TRY IT OUT!

Arduino Programmering



Her finner du fullstendig oversikt over Arduino Programmeringsspråket:

<https://www.arduino.cc/en/Reference/HomePage>

Play and Explore

| Programmering og Arduino |

<https://www.halvorsen.blog>



Del 2: Praktiske Arduino Eksempler

| Hans-Petter Halvorsen |

<https://www.halvorsen.blog>



Praktiske Arduino Eksempler

Hans-Petter Halvorsen

Arduino Eksempler

Nå kommer det en del eksempler/oppgaver som du skal gjøre steg for steg slik som angitt i teksten.

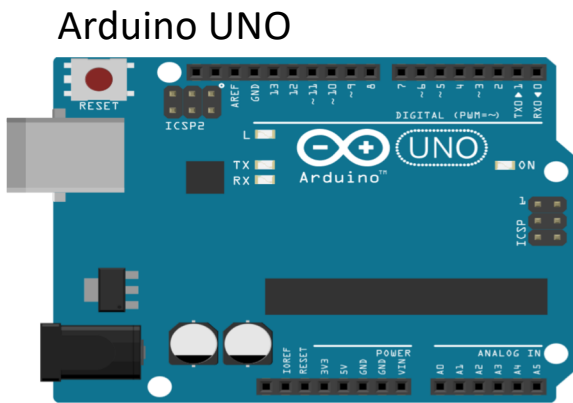
Når du er ferdig med oppgaven, vil jeg anbefale at du leker deg litt, gjør små forandringer i programmet, osv.

Arduino Eksempler

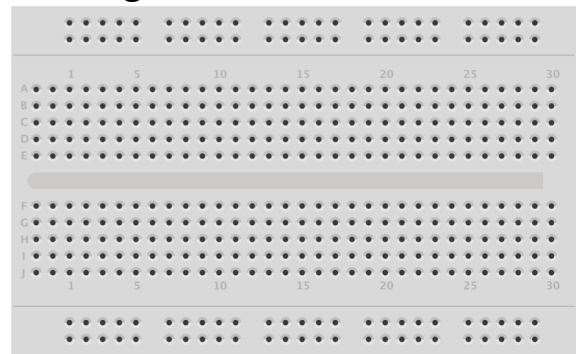
Her er noen eksempler som vi vil se nærmere på:

1. [Elektrisk krets](#)
2. [Blinkende LED](#)
3. [Bryter](#)
4. [Potensiometer](#)
5. [Temperaturmåling](#)
6. [Lyssensor](#)
7. [Termistor](#)

Vi trenger følgende utstyr



Koblingsbrett

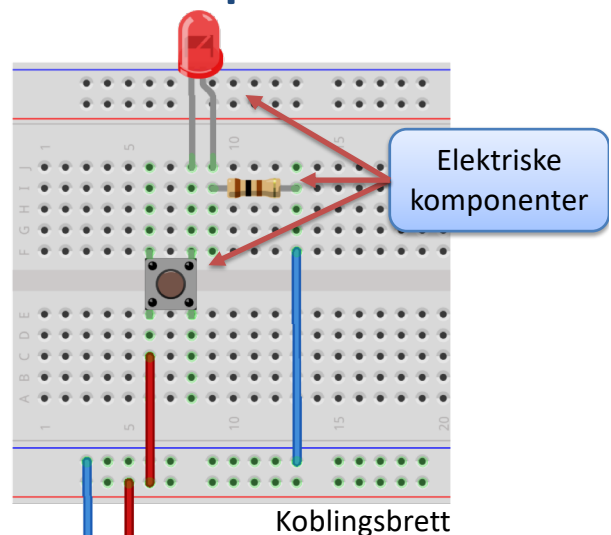
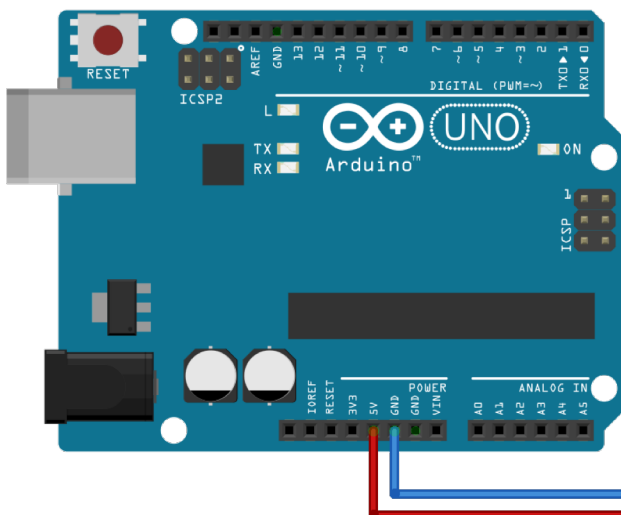


+

+ LED, motstander, ledninger, m.m.

Koblingsbrettet brukes til å koble opp komponenter og elektriske kretser

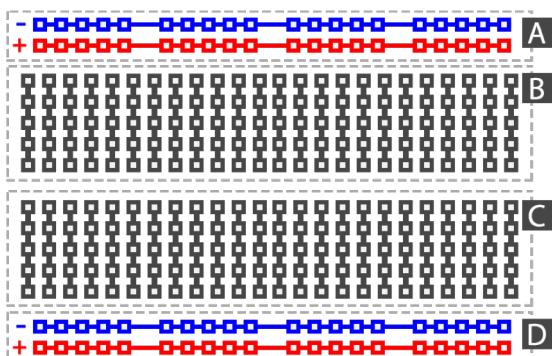
Koblingsbrett - Eksempel



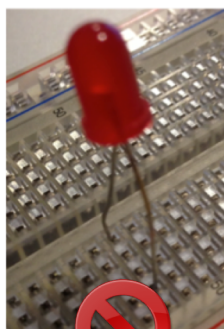
Ledninger som brukes til å koble de elektriske komponentene sammen til en lukket krets

Koblingsbrett - Riktig oppkobling

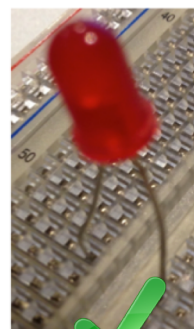
Pass på at du ikke kortslutter komponentene som du kobler på koblingsbrettet



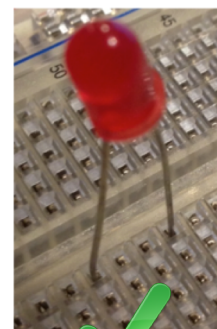
Ikke kortslutt komponentene dine



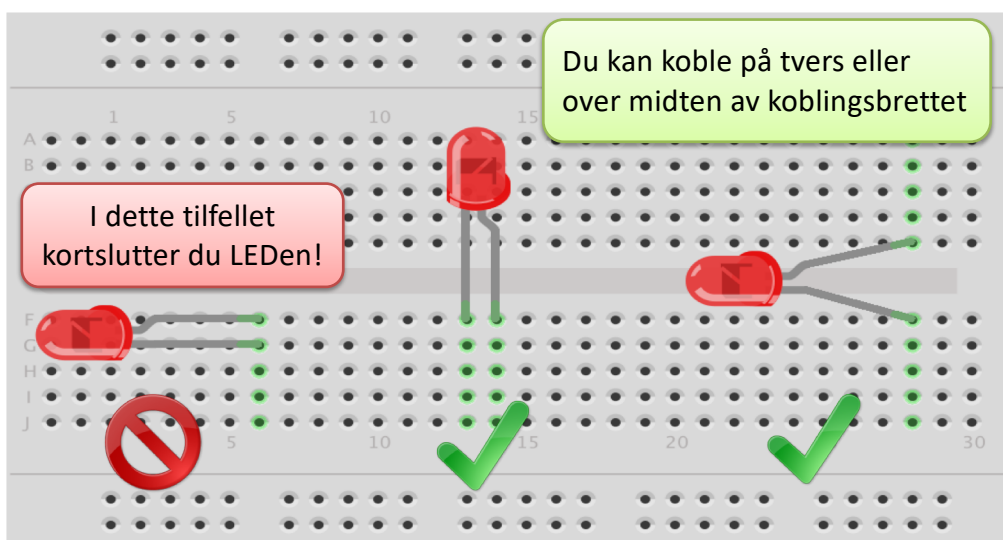
Kople over bruen på eksperimentkortet...



... eller på tvers av rader



Koblingsbrett - Riktig oppkobling



fritzing

Eksempel 1

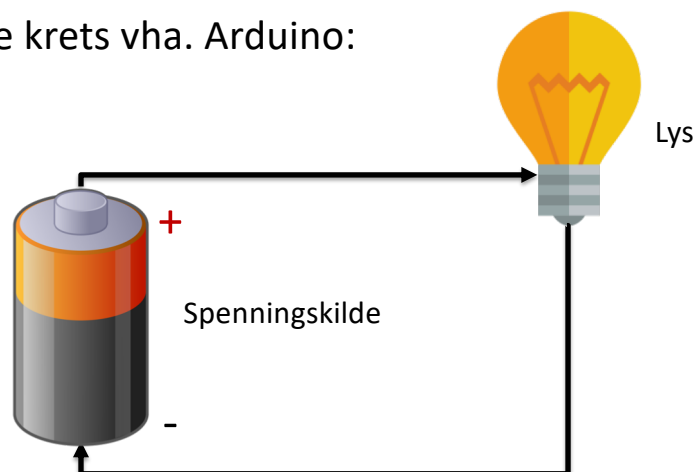
TRY IT OUT!

Elektrisk krets

Elektrisk krets

Vi skal lage følgende krets vha. Arduino:

Istedenfor et batteri vil vi bruke Arduino som en spenningskilde (5V)



Eksempel 1

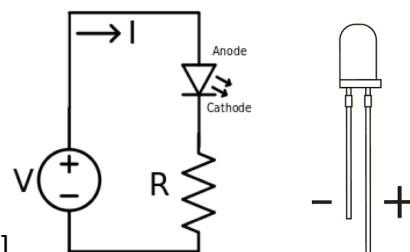
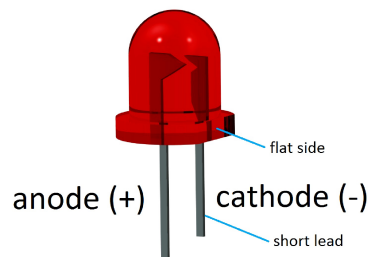
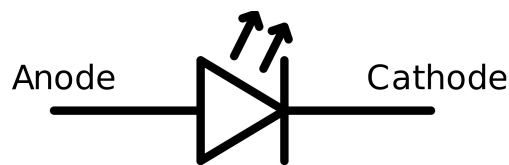


LED

En lysdiode (eller LED, fra engelsk light-emitting diode) er en en halvleder-diode som utstråler lys under tilførsel av elektrisk strøm.

Hvis man setter på en for stor strøm så vil lysdioden brenne ut og bli ødelagt, for å forhindre dette bruker man en strømbegrensende motstand som setter strømmen igjennom lysdioden til ønsket verdi og passer på at spenningen er riktig.

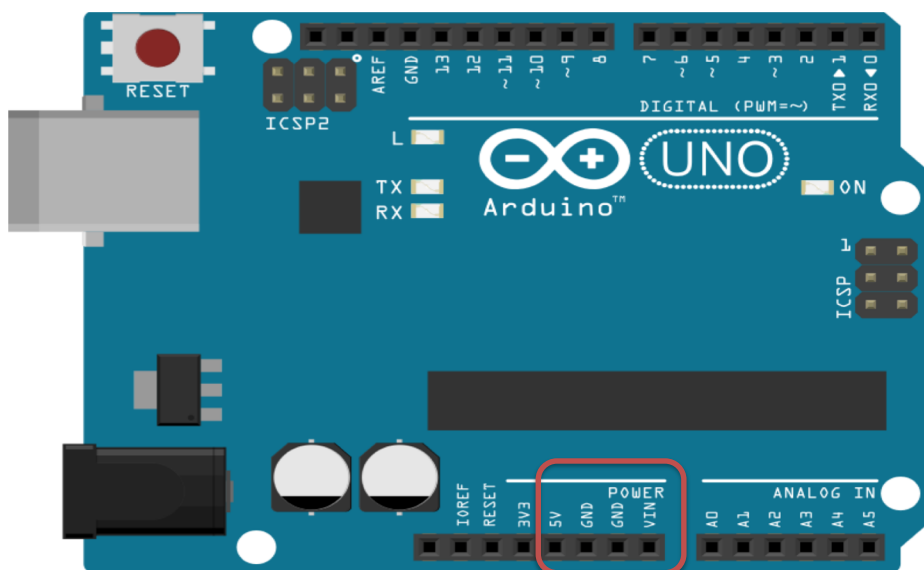
Merk! Strømmen må gå fra pluss (lang pinne) til minus (kort pinne), ellers vil ikke lysdioden lyse.



Eksempel 1

[Wikipedia]

Introduksjon



Vi vil begynne med å lage en elektrisk krets vha Arduino sine "Power"-porter "5V" og "GND"

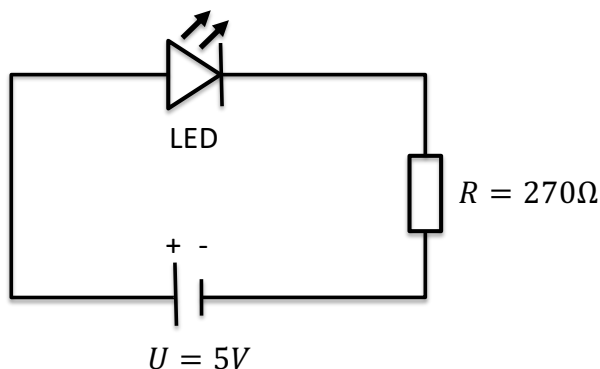
GND = Ground (Jord)

Eksempel 1

Elektrisk krets

TRY IT OUT!

Lag følgende krets vha Arduino og et koblingsbrett:



Utstyr:

- Koblingsbrett
- LED
- Motstand
- Noen ledninger
- Multi-meter

Eksempel 1

NB! Ingen Arduino kode kreves i dette tilfellet

Hvorfor trengs en Motstand?



Hvis man setter på en for stor strøm så vil lysdioden brenne ut og bli ødelagt, for å forhindre dette bruker man en strømbegrensende motstand som setter strømmen igjennom lysdioden til ønsket verdi og passer på at spenningen er riktig.

Motstandens størrelse?

En LED ønsker typisk en strømstyrke på 20mA (dette finner du i databladet til LEDen).

Vi bruker da Ohms lov:

$$U = RI$$

Arduino gir $U=5V$ og motstanden ønsker $I=20mA$. Vi får da følgende:

$$R = \frac{U}{I}$$

Dvs. $R = \frac{5V}{0.02A} = 250\Omega$. Motstander på 250Ω er ikke så vanlige, derfor bruker vi den som er nærmest, f.eks. 270Ω

Eksempel 1

Finn størrelsen på motstanden

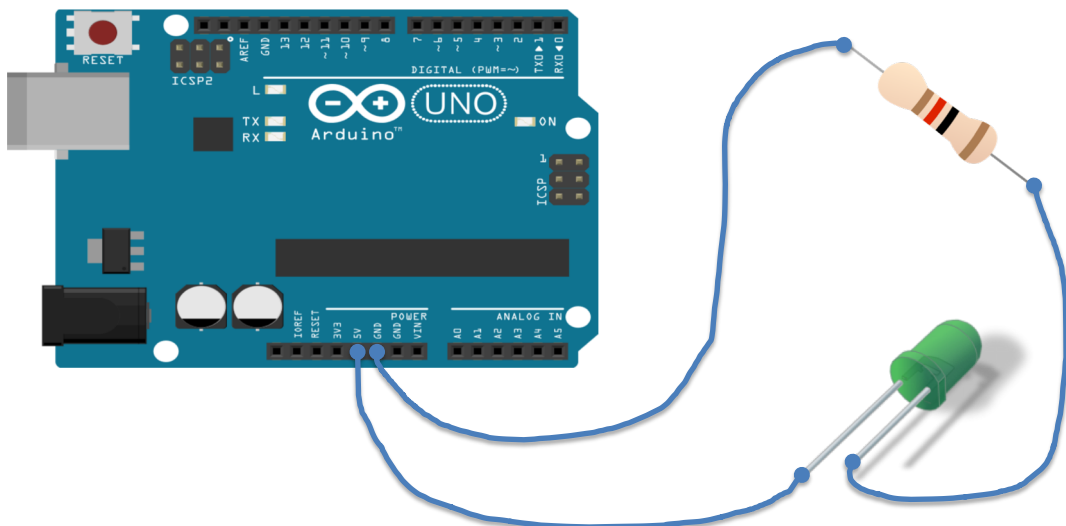
Finn verdien på dine motstander ved hjelp av fargekodene på motstandene eller bruk et multimeter:

Still inn hjulet på Multimeteret slik at det måler Ohm. Juster hjulet til du finner riktig måleområde

Du kan også bruke en "motstands-kalkulator" som du finner på internett:

<http://www.allaboutcircuits.com/tools/resistor-color-code-calculator/>

Eksempel 1

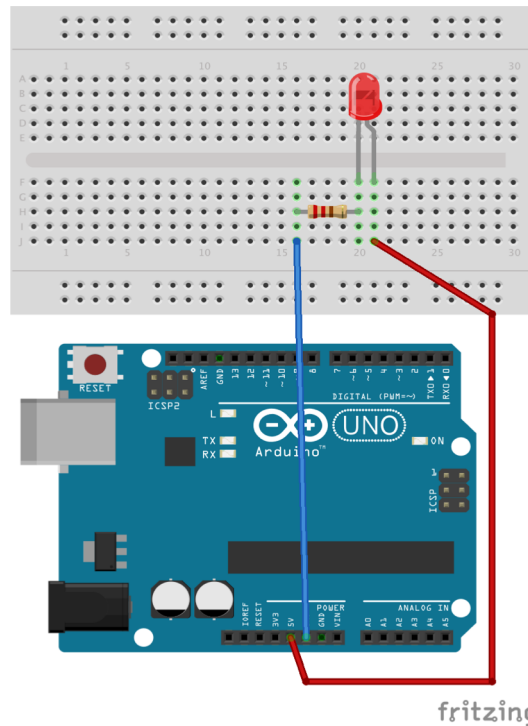


Husk riktig vei når dere kobler LEDen

Dere bruker et koblingsbrett. Se neste side hvordan dere gjør dette.

Eksempel 1

Oppkobling



TRY IT OUT!

Husk riktig vei når dere kobler LEDen – ellers vil den ikke lyse

Merk! Du må bruke en motstand på $R = 270\Omega$ for ikke å ødelegge LEDen. Bruk f.eks et Multimeter.

Eksempel 1

fritzing

Ekstraoppgaver

Prøv ut følgende:

- Seriekobling av 2 motstander ($R_1 = 270\Omega$, $R_2 = 330\Omega$)
- Parallellkobling av 2 motstander

Regn ut den totale motstanden.

Sjekk med et multimeter for å se om det stemmer.

Mål også strøm og spenning vha et multimeter.

Stemmer det ihht. teorien (Ohms lov/Kirchoffs lover)?

Eksempel 1



|Slutt på eksemplet|

Eksempel 2

TRY IT OUT!

Blinkende LED





LED

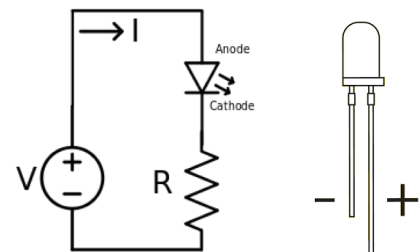
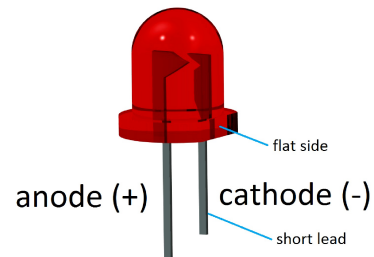
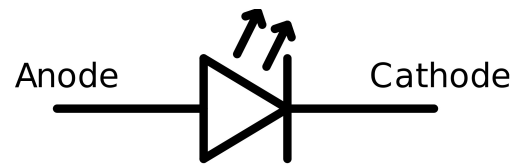
En lysdiode (eller LED, fra engelsk light-emitting diode) er en en halvleder-diode som utstråler lys under tilførsel av elektrisk strøm.

Hvis man setter på en for stor strøm så vil lysdioden brenne ut og bli ødelagt, for å forhindre dette bruker man en strømbegrensende motstand som setter strømmen igjennom lysdioden til ønsket verdi og passer på at spenningen er riktig.

Merk! Strømmen må gå fra pluss (lang pinne) til minus (kort pinne), ellers vil ikke lysdioden lyse.

Eksempel 2

[Wikipedia]



Introduksjon

Her skal vi programmere Arduinoen slik at vi får et lys (LED) til å blinke.

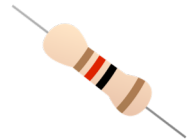
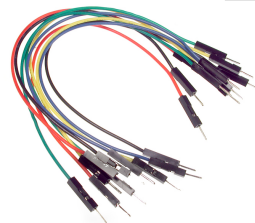
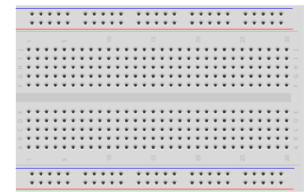
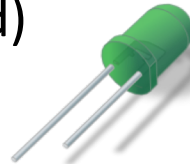
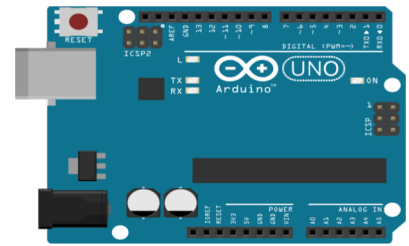
Fremgangsmåte:

1. Først kobler vi kretsen med nødvendige komponenter
2. Deretter lager vi programmet

Eksempel 2

Nødvendig utstyr

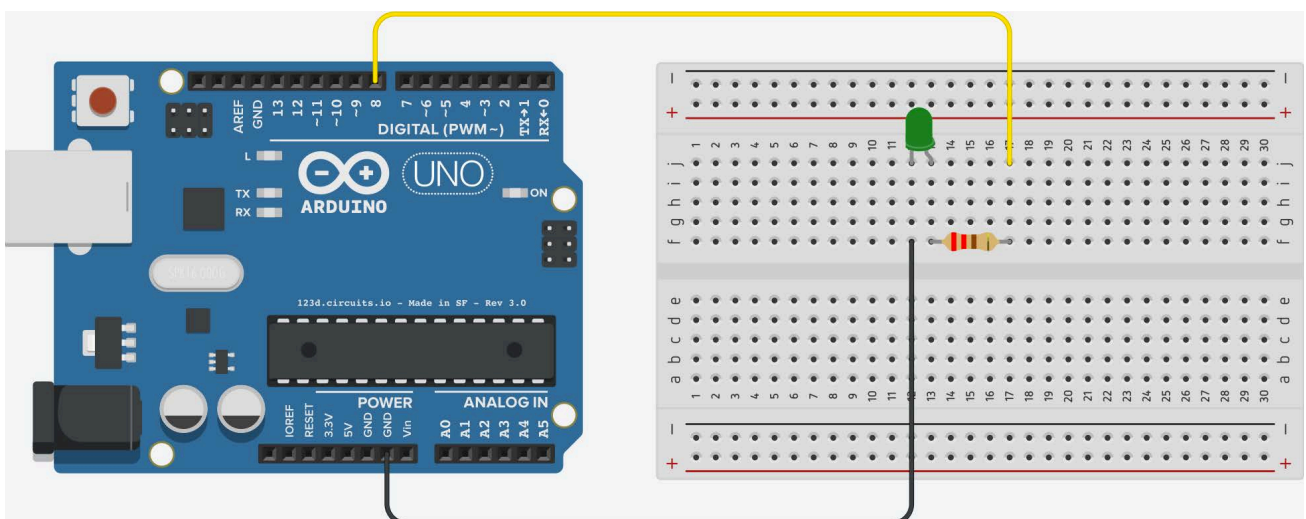
- Arduino UNO
- Koblingsbrett (Breadboard)
- LED
- Motstand, $R = 270\Omega$
- Ledninger (Jumper Wires)



Eksempel 2

Oppkobling

TRY IT OUT!



Eksempel 2

Programkomponenter

Generell Programstruktur:

```
//Globale variable
...

void setup()
{
  //Inisialisering
}

void loop()
{
  //Hovedprogrammet
}
```

Eksempel 2

Arduino Language Reference:
<https://www.arduino.cc/en/Reference>

Funksjoner som du må bruke for å lage programmet:

Hvilken pinne bruker du (0, 1, 3, ...)?

```
pinMode (pin, mode);
```

Alle digitale pinner kan enten være innganger (INPUT) eller utganger (OUTPUT). Vi skal bruke den til utgang.

```
digitalWrite (pin, value);
```

LEDen lyser LEDen lyser ikke

En digital pinne kan kun ha 2 verdier, enten HIGH eller LOW

```
delay (ms);
```

Funksjonen delay() pauser programmet i et gitt antall millisekunder (ms), f.eks. delay(1000) pauser programmet i 1 sekund

Arduino Program

```
void setup()
{
  pinMode(8, OUTPUT);
}

void loop()
{

  digitalWrite(8, HIGH); // Turn on the LED
  delay(1000);           // Wait for one second
  digitalWrite(8, LOW); // Turn off the LED
  delay(1000);           // Wait for one second
}
```

TRY IT OUT!

Eksempel 2

Arduino Program

```

int ledPin = 8;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, HIGH); // Turn on the LED
  delay(1000);                // Wait for one second
  digitalWrite(ledPin, LOW);  // Turn off the LED
  delay(1000);                // Wait for one second
}

```

TRY IT OUT!

En ørliten forbedring. Vi bruker en variabel til å definere pinne-nummeret

Eksempel 2

Ekstraoppgaver

TRY IT OUT!

Prøv ut følgende:

- Få LEDen til å blinke fortere/saktere
- Bruk en annen Digital port på Arduinoen. Hvilke forandringer må du gjøre i programmet ditt?
- Koble fra USB kableen. Bruk et 9V batteri istedenfor (hvis du har). Evt. et eksternt Power Supply (hvis du har).
- Har du noen forslag?

Eksempel 2

Ekstern Strømforsyning

Du kan koble fra USB kablen slik at Arduinoen kjører helt uavhengig av PCen.

Da må Arduinoen ha ekstern strømforsyning.

Her er noen alternativer:

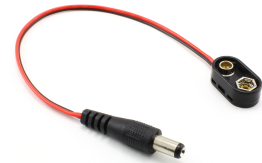


9-12V Power Supply

Eksempel 2



9v Battery connector 2.1mm



9v Battery connector

Ekstern Strømforsyning

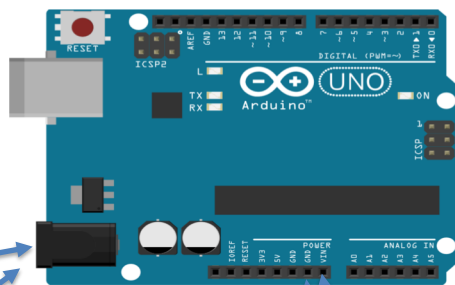
9-12V Power Supply



9v battery connector 2.1mm



Eksempel 2



9v battery connector



|Slutt på eksemplet|

Eksempel 3

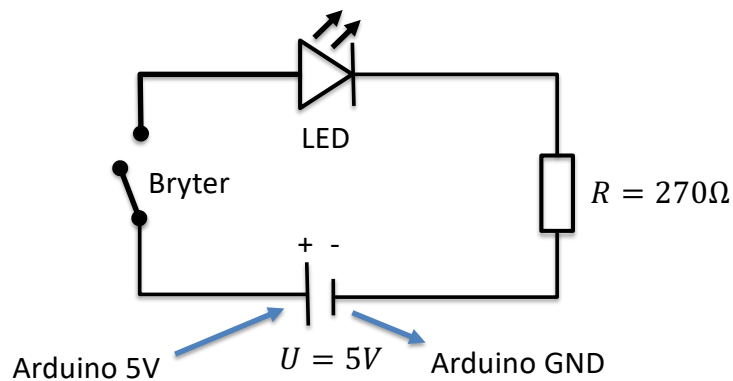
TRY IT OUT!

Bryter

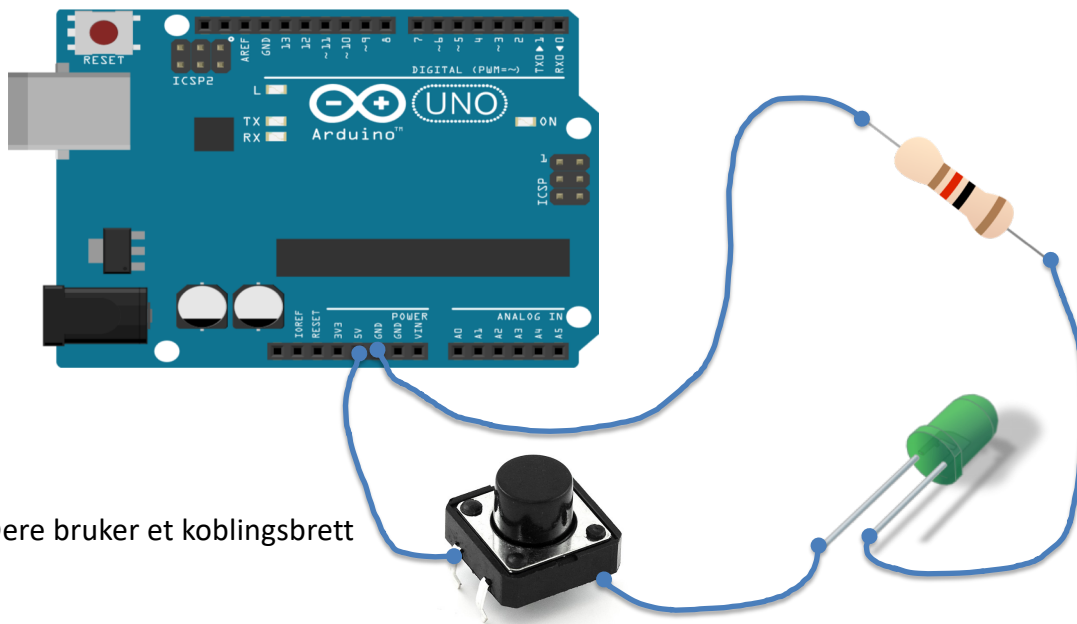


Introduksjon

Lag følgende krets ved hjelp av Arduino:



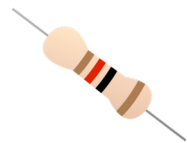
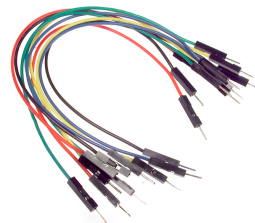
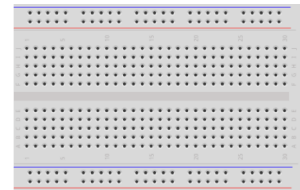
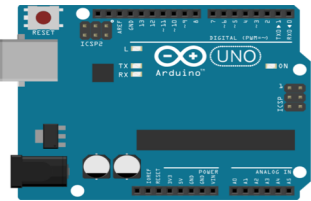
Eksempel 3



Eksempel 3

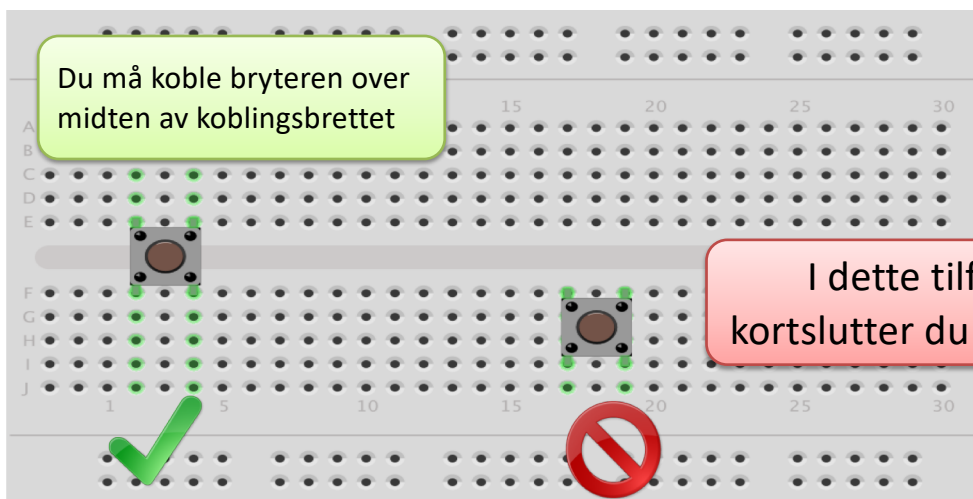
Nødvendig utstyr

- Koblingsbrett
- LED
- Bryter
- Motstand, $R = 270\Omega$
- Ledninger (Jumper Wires)



Eksempel 3

Husk riktig plassering av bryteren

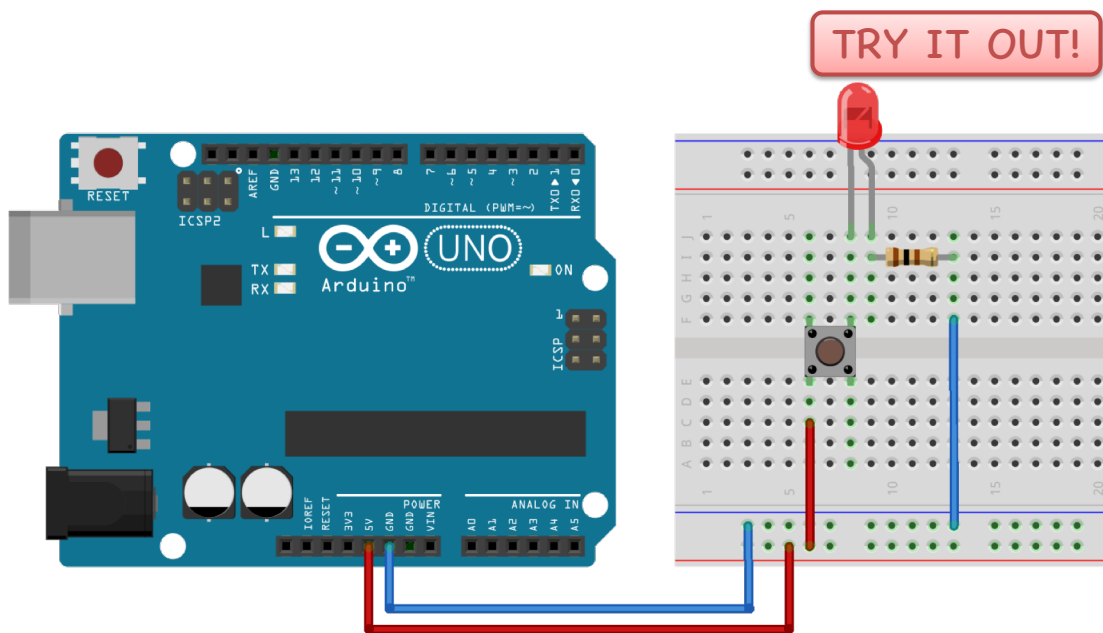


Unngå Kortslutning!!

fritzing

Eksempel 3

Oppkobling



NB! Ingen Arduino kode kreves i dette tilfellet

Ekstraoppgaver

TRY IT OUT!

Prøv ut følgende:

- Bruk et Multimeter for å måle strøm og spenning i kretsen før og etter du trykker på bryteren
- Prøv å koble 2 brytere i serie
- Prøv å koble 2 brytere i parallell

Eksempel 3

Eksempel 3b

TRY IT OUT!



Bryter

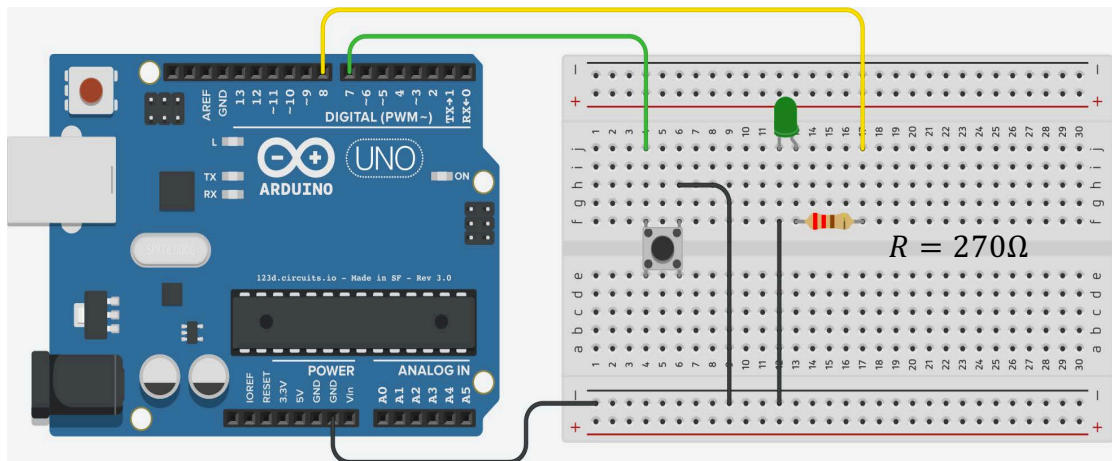
Vi ønsker å bruke en bryter for å kunne slå lyset av og på vha Arduino programmering

Nødvendig utstyr

- Arduino
- Breadboard
- LED
- Bryter
- Motstand, $R = 270 \Omega$
- Ledninger (Jumper Wires)

Eksempel 3b

Oppkobling



Merk! Ved denne konfigurasjon forutsetter vi at vi bruker vi en intern "Pull-up" motstand for å hindre "kortslutning".

pinMode settes da slik:

Eksempel 3b

```
pinMode (pin, INPUT_PULLUP);
```

Arduino Program

```
const int buttonPin = 7;
const int ledPin = 8;

int buttonState = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop()
{
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

TRY IT OUT!

Arduino har en intern "pull-up" motstand (motstand som kobler til power internt). Vi kan bruke denne istedenfor å koble på en ekstern "pull-up" motstand (se neste eksempel).

Eksempel 3b

Bryter – Alternativ løsning

https://en.wikipedia.org/wiki/Pull-up_resistor

- Virkemåten er helt lik, men her bruker vi en ekstern "pull-up" motstand ($R = 10k\Omega$) ifm oppkobling av bryteren.
- Vi må her bruke en "pull-up"/ "pull-down" motstand i forbindelse med bryteren for å unngå "kortslutning" når bryteren trykkes ned
- Den fysiske koblingen blir litt forskjellig, samt at programmet blir bitte litt annerledes

Eksempel 3b

<http://playground.arduino.cc/CommonTopics/PullUpDownResistor>

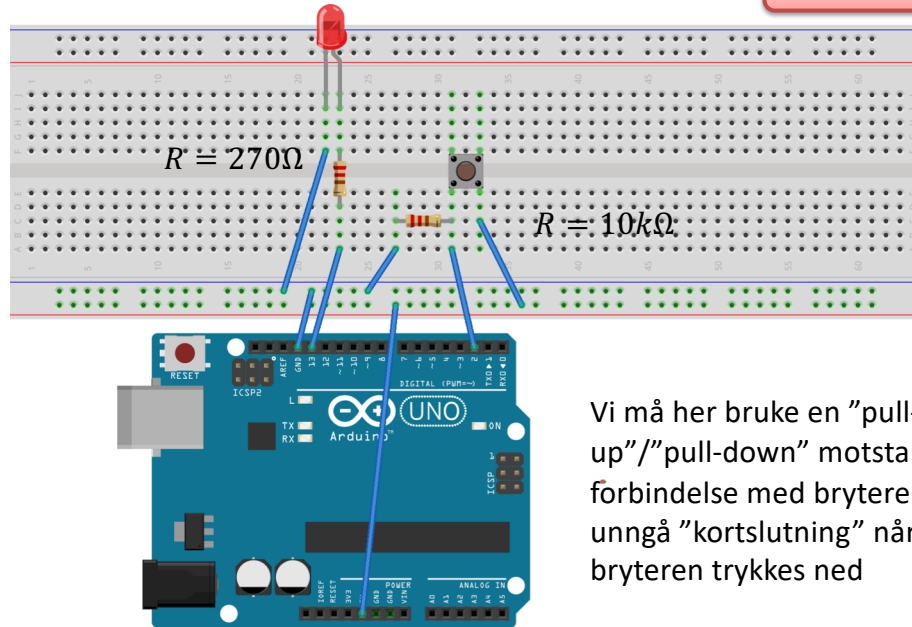
Nødvendig utstyr

- Arduino
- Breadboard
- LED
- Bryter
- Motstander $R = 270\Omega$ og $R = 10k\Omega$
- Ledninger (Jumper Wires)

Eksempel 3b

Oppkobling

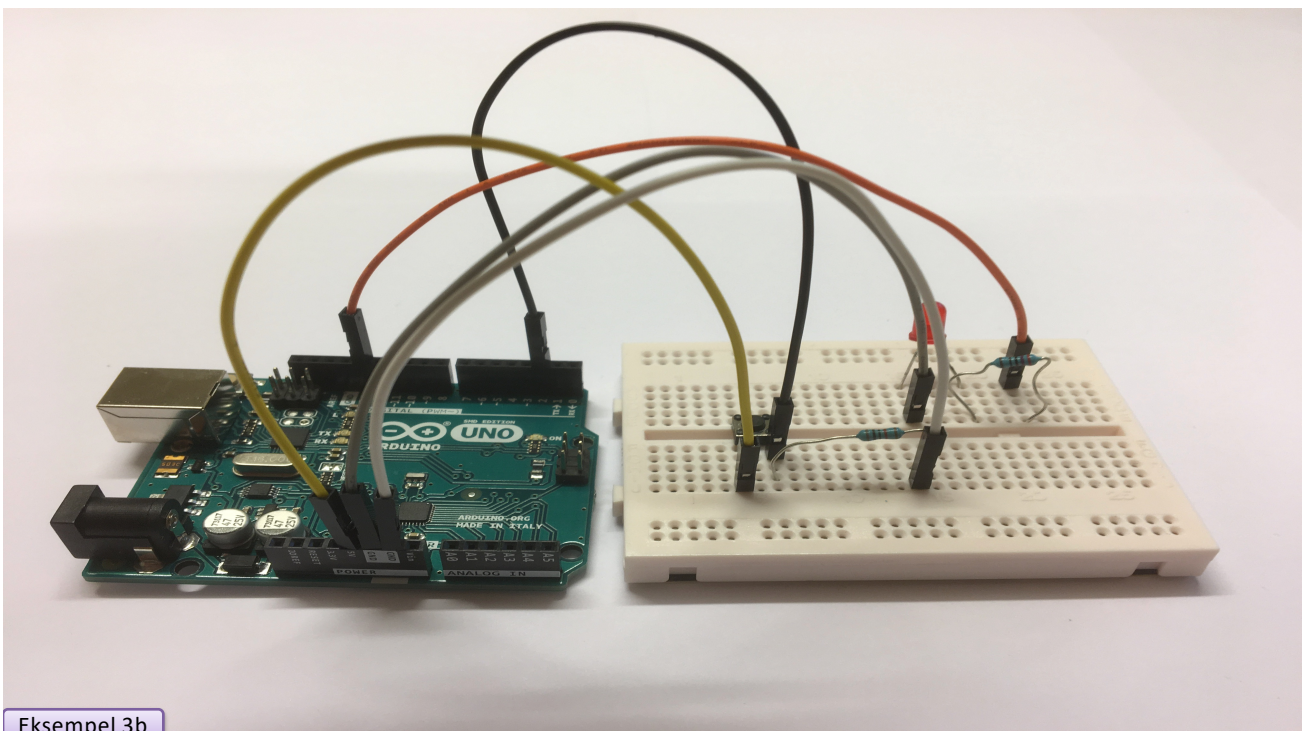
TRY IT OUT!



Vi må her bruke en "pull-up"/"pull-down" motstand i forbindelse med bryteren for å unngå "kortslutning" når bryteren trykkes ned

Eksempel 3b

fritzing



Eksempel 3b

Arduino Program

Eksempel 3b

```
const int buttonPin = 2;
const int ledPin = 13;

int buttonState = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT);
}

void loop()
{
  buttonState = digitalRead(buttonPin);

  if (buttonState == HIGH)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

TRY IT OUT!

Ekstraoppgaver

TRY IT OUT!

Prøv ut følgende:

- Kan du få lysdioden til å blinke når knappen trykkes inn?
- Kan du få lysdioden til å skrus på ved å trykke på knappen? Og deretter skru av lysdioden ved å trykke på knappen en gang til?

Eksempel 3b

Ekstraoppgaver -Løsninger

Prøv selv før du titter på løsningsforslaget!

Eksempel 3b

Kan du få lysdioden til å blinke når knappen trykkes inn?

Arduino
Program

Eksempel 3b

```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;
boolean ledState = HIGH;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH)
  {
    digitalWrite(ledPin, ledState);
    ledState = !ledState;
    delay(100);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

TRY IT OUT!

Kan du få lysdioden til å skrus på ved å trykke på knappen? Og deretter skru av lysdioden ved å trykke på knappen en gang til?

Arduino Program

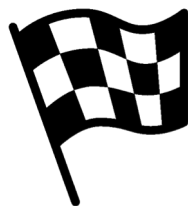
Eksempel 3b

```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;
boolean ledState = HIGH;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH)
  {
    digitalWrite(ledPin, ledState);
    ledState = !ledState;
  }
  else
  {
    //Do nothing
  }
}
```

TRY IT OUT!



| Slutt på eksemplet |

Eksempel 4

TRY IT OUT!

Potensiometer

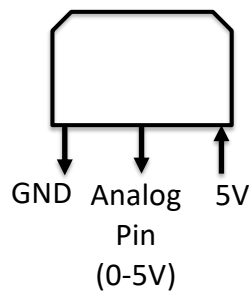
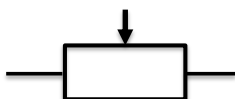


Potensiometer

Et potensiometer fungerer som en variabel motstand, dvs du kan justere motstanden ved å skru på et hjul.

Et potensiometer brukes f.eks til å justere lydvolum eller lysstyrke (dimmer).

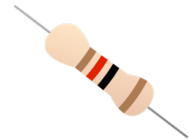
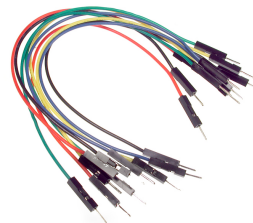
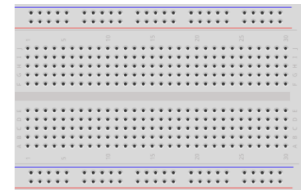
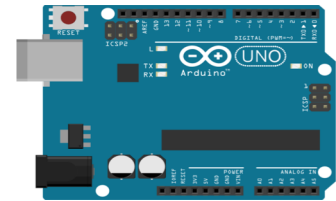
Elektrisk symbol:



Eksempel 4

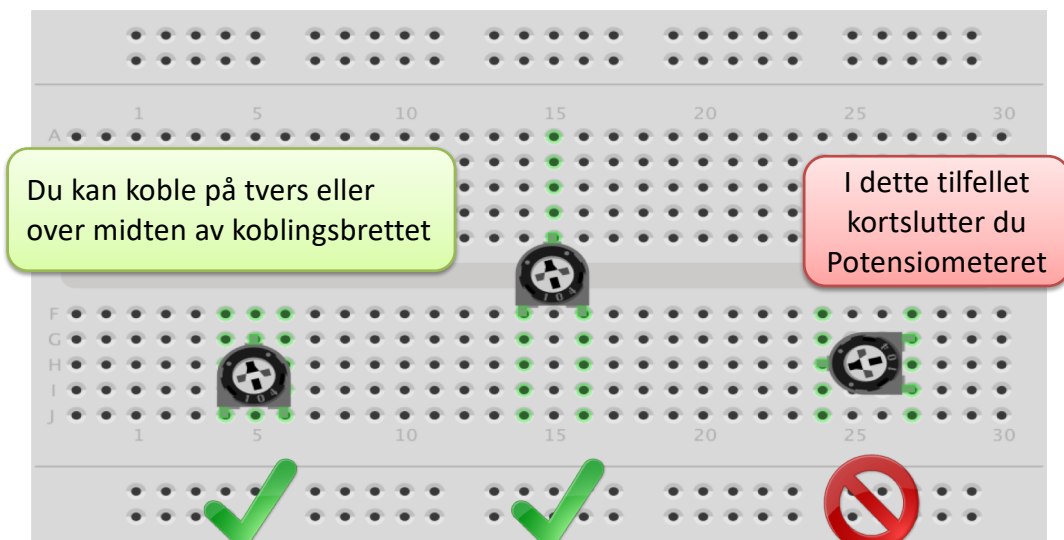
Nødvendig utstyr

- Arduino
- Breadboard
- Potensiometer
- LED
- Motstand, $R = 330\Omega$
- Ledninger (Jumper Wires)



Eksempel 4

Koblingsbrett - Riktig oppkobling



Eksempel 4

fritzing

Dimmer



I dette eksemplet vil vi lage en enkel dimmer vha et potensiometer som kontrollere lysstyrken til en LED.

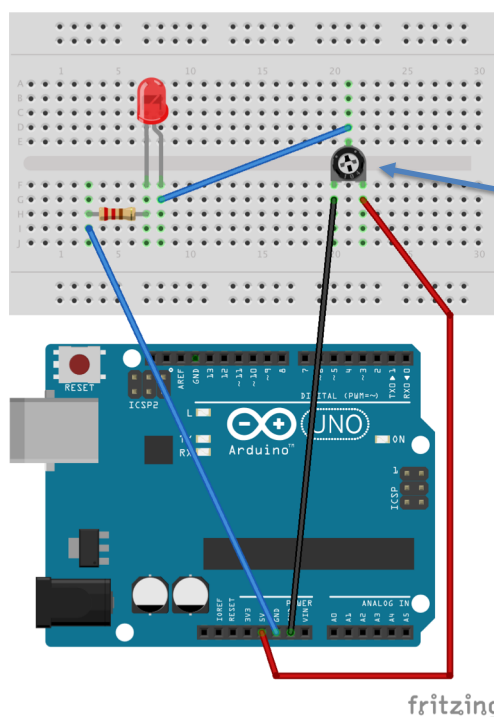
Dette er et typisk eksempel på hva man kan bruke et potensiometer til. Alle har en slik dimmer hjemme i stua eller andre steder i huset.

Rent teknisk øker man spenningen i kretsen slik at lyser lyser sterkere, eller motsatt.

NB! Ingen Arduino kode kreves i dette tilfellet

Eksempel 4

Oppkobling



TRY IT OUT!

Skru på potensiometeret (dimmeren) for å justere lysstyrken til LEDen

NB! Ingen Arduino kode kreves i dette tilfellet

Eksempel 4

Eksempel 4b

Potensiometer og blinkende LED

- I dette eksemplet vil vi bruke et potensiometer til å kontrollere blinkingen (raskt/sakte) til en LED.
- For å få til dette vil vi bruke funksjonen `analogRead()` til å lese den variable spenningen som Potensiometeret gir når vi skrur på det.
- Jo større spenningen er, jo saktere vil LEDen blinke og motsatt

`analogRead()` funksjonen i Arduino

Leser verdien fra en spesifikk analog pinne. Arduino brettet inneholder 6 analoge pinner (kanaler), og bruker en 10-bit analog til digital konverter.

Syntaks:

value = analogRead(analogPin);

value vil da være en verdi mellom 0 og 1023

Eksempel:

```
int sensorPin = 0;
int sensorValue;

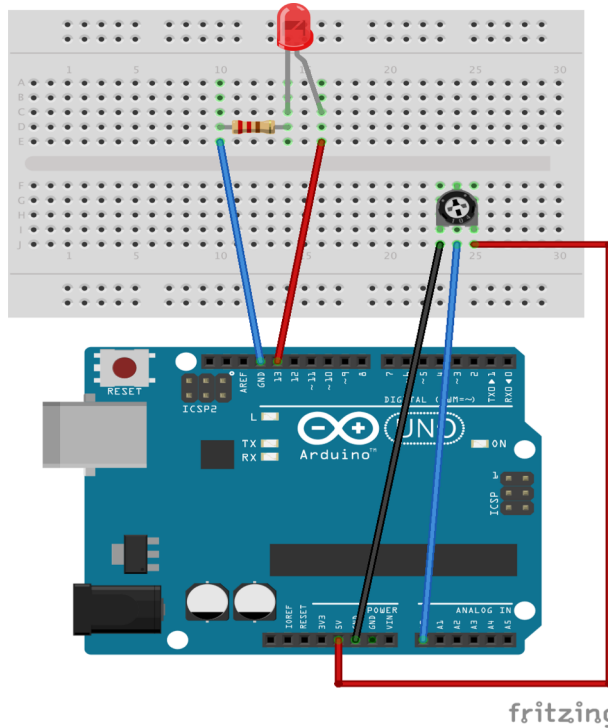
void setup()
{
}

void loop()
{
  sensorValue = analogRead(sensorPin);
}
```

<https://www.arduino.cc/en/Reference/AnalogRead>

Eksempel 4b

Oppkobling



TRY IT OUT!

Eksempel 4b

Arduino Program

```
int sensorPin = 0;
int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int sensorValue;

  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);

  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
}
```

TRY IT OUT!

Eksempel 4b

Ekstraoppgaver

TRY IT OUT!

Prøv ut følgende:

- Skriv verdien til Serial Monitor
- Bruk Potensiometeret til å øke/reducere lysstyrken til en LED.
 - Alternativ 1 (uten programmering): Du kan få til dette til å koble LEDen på utgangen av Potensiometeret. Du bruker da Potensiometeret til å direkte justere spenningen som går til LEDen
 - Alternativ 2 (med programmering): Du bruker funksjonen `analogRead` til å lese spenningen fra Potensiometeret, deretter bruker du funksjonen `analogWrite()` til å justere spenningen til LED lyset.

Eksempel 4b

analogWrite

Denne funksjonen skriver en "analog verdi" (PWM signal) til en pinne. Kan f.eks få en LED til å lyse med forskjellig intensitet.

Syntaks:

`analogWrite(pin, value)`

der value må være en verdi mellom 0 to 255

<https://www.arduino.cc/en/Reference/AnalogWrite>

Eksempel 4b

Merk! Du må bruke en av pinnene merket med ~ (tilde)

Eksempel:

```
int ledPin = 9;
int value = 0;
void setup()
{
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    value = random(256);
    analogWrite(ledPin, value);
    delay(1000);
}
```



|Slutt på eksemplet|

Eksempel 5

TRY IT OUT!

Temperatur



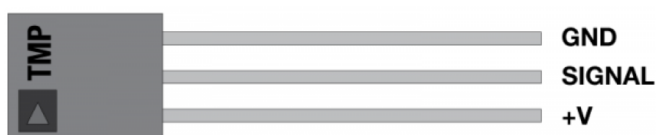
Introduksjon

I dette eksemplet vil vi bruke en liten temperatursensor til å lese temperaturen i rommet.

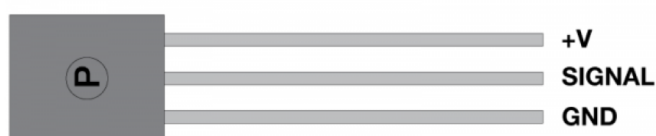
I dette eksemplet vil vi bruke en av de "Analoge Inn" portene på Arduino

Eksempel 5

TMP36 Temperatur-sensor



FRONT



BACK



Technical data	
Temperature measurement range	-40...+125 °C
Accuracy	±2 °C (0...70 °C)
Power supply	2.3...5.5 V
Package	TO-92
Temperature sensitivity, voltage	10 mV/°C

Eksempel 5

TMP36 Temperatursensor



TMP36 sensoren er en såkalt spenningsbasert sensor, dvs. sensoren bruker et "solid-state" prinsipp for å bestemme temperaturen. Det vil si, de bruker ikke kvikksølv (som gamle termometre), eller temperaturfølsomme motstander (såkalte termistorer). I stedet bruker de det faktum at hvis temperaturen øker, vil spenningen over en diode øke med en kjent rate.

Ved nøyaktig å forsterke spenningsforandringen, kan man generere et analogt signal som er direkte proporsjonalt med temperaturen.

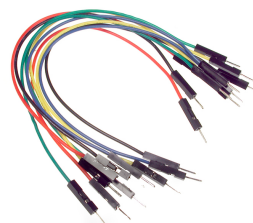
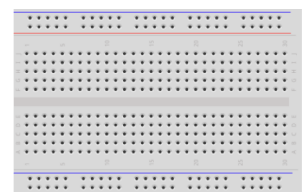
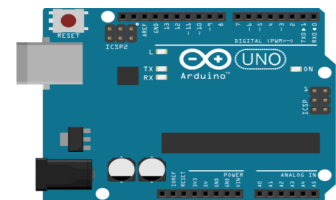
Fordi disse sensorene ikke har noen bevegelige deler, er de robuste, de slites aldri ut, trenger ikke kalibrering, de virker under mange miljøforhold. Dessuten er de veldig billig og ganske enkel å bruke.

Eksempel 5

<https://learn.adafruit.com/tmp36-temperature-sensor>

Nødvendig utstyr

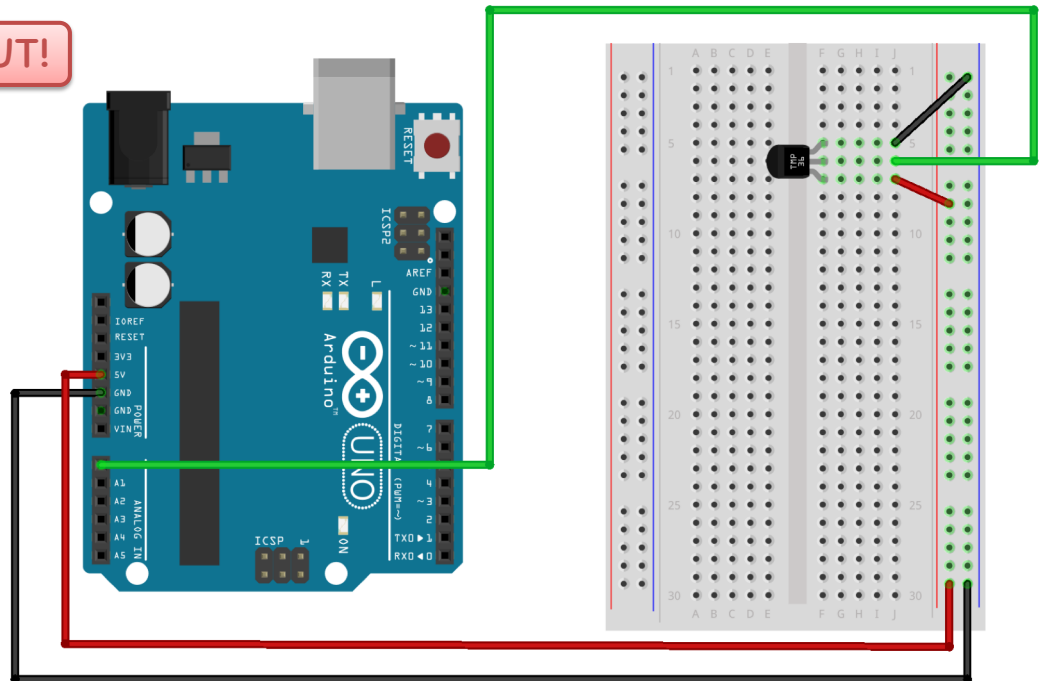
- Arduino
- Breadboard
- TMP36
- Ledninger (Jumper Wires)



Eksempel 5

TRY IT OUT!

Oppkobling



Eksempel 5

fritzing

analogRead

Leser verdien fra en spesifikk analog pinne. Arduino brettet inneholdet 6 analoge pinner (kanaler), og bruker en 10-bit analog til digital konverter.

Syntaks:

value = analogRead(analogPin);

value vil da være en verdi mellom 0 og 1023

Eksempel:

```
int sensorPin = 0;
int sensorValue;

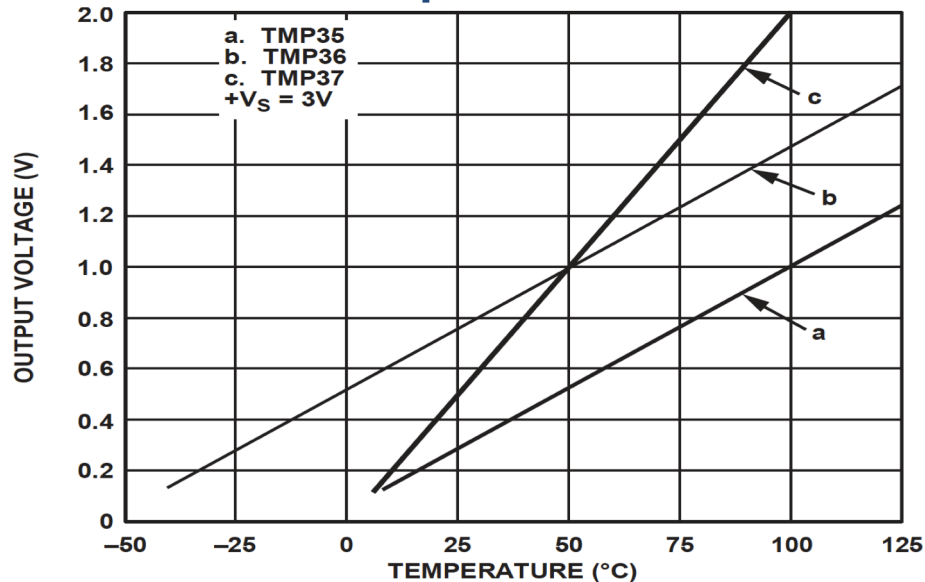
void setup()
{
}

void loop()
{
  sensorValue = analogRead(sensorPin);
}
```

Eksempel 5

<https://www.arduino.cc/en/Reference/AnalogRead>

Datablad Temperatur-sensor



Eksempel 5

[\[http://no.rs-online.com/webdocs/14cd/0900766b814cd0a1.pdf\]](http://no.rs-online.com/webdocs/14cd/0900766b814cd0a1.pdf)

Temperatur konvertering

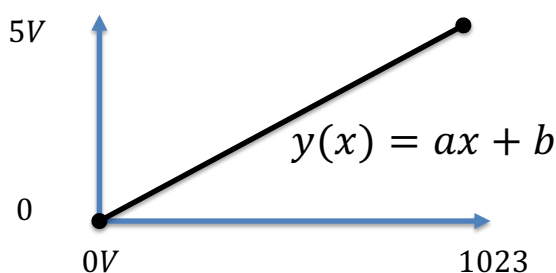
For at verdien fra sensoren skal vises i grader Celsius må vi gjøre noen konverteringer og beregninger:

1. `analogRead()` gir ut en verdi mellom 0 og 1023
2. Denne må konverteres til spenning (0-5V)
3. Deretter må denne konverteres til grader Celsius ihht grafen fra databladet (forrige side)

Eksempel 5

ADC-verdi til Spenningsverdi

`analogRead()` gir ut en verdi mellom 0 og 1023.
Denne må konverteres til spenning (0-5V)



Eksempel 5

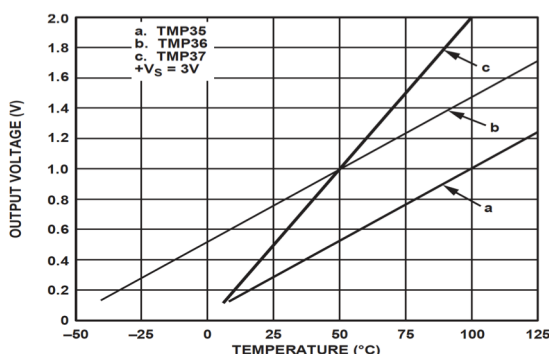
Vi har at:

$$\begin{aligned} 0 &\rightarrow 0V \\ 1023 &\rightarrow 5V \end{aligned}$$

Dette gir følgende formel:

$$y(x) = \frac{5}{1023}x$$

Sammenheng mellom Spenning (V) og grader Celsius (°C)



Dette gir følgende:

$$y - 25 = \frac{50 - 25}{1 - 0.75}(x - 0.75)$$

Dette gir følgende formel for relasjonen mellom spenning og grader Celsius:

Eksempel 5

Fra figuren i databladet har vi følgende:

$$\begin{aligned} (x_1, y_1) &= (0.75V, 25^\circ C) \\ (x_2, y_2) &= (1V, 50^\circ C) \end{aligned}$$

Det er en lineær relasjon mellom spenning og grader Celsius:

$$y = ax + b$$

Vi kan finne a og b vha. følgende formel:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

$$y = 100x - 50$$

Arduino Program

Eksempel 5

```
const int temperaturePin = 0;

float adcValue;
float voltage;
float degreesC;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  adcValue = analogRead(temperaturePin);

  voltage = (adcValue*5)/1023;
  degreesC = 100*voltage - 50;

  Serial.print("ADC Value: ");
  Serial.print(adcValue);

  Serial.print(" voltage: ");
  Serial.print(voltage);

  Serial.print(" deg C: ");
  Serial.println(degreesC);

  delay(1000);
}
```

TRY IT OUT!

Konvertering fra ADC-verdi til spenning

Konvertering fra spenning til grader Celsius

Konvertering fra Celsius til Fahrenheit

I Norge bruker vi typisk måleenheten Celsius i forbindelse med temperatur, mens f.eks i USA brukes måleenheten Fahrenheit.

Konverteringen mellom disse måleenhetene er som følger:

$$T_F = \frac{9}{5}T_C + 32$$

TRY IT OUT!

Utvid programmet slik at temperaturen vises både i grader Celsius og grader Fahrenheit

Eksempel 5

Arduino Program

Eksempel 5

```
const int temperaturePin = 0;
float adcValue;
float voltage;
float degreesC;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  adcValue = analogRead(temperaturePin);
  voltage = (adcValue*5)/1023;

  degreesC = 100*voltage - 50;
  degreesF = degreesC * (9.0/5.0) + 32.0;

  Serial.print("ADC Value: ");
  Serial.print(adcValue);
  Serial.print(" voltage: ");
  Serial.print(voltage);
  Serial.print(" deg C: ");
  Serial.print(degreesC);
  Serial.print(" deg F: ");
  Serial.println(degreesF);

  delay(1000);
}
```

TRY IT OUT!

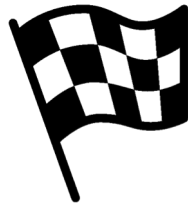
Ekstraoppgaver

TRY IT OUT!

Prøv ut følgende:

- Konverteringen fra Celsius til Fahrenheit bør lages som en funksjon som du kaller fra hovedprogrammet.
- Få en LED til å lyse hvis temperaturen blir for lav eller for høy

Eksempel 5



|Slutt på eksemplet|

Eksempel 6

TRY IT OUT!

Lysmåling



Introduksjon

I dette eksemplet vil vi bruke en lyssensor til å måle lysintensiteten i rommet.

Hvis det er mørkt, vil vi slå på lyset (LED)

Hvis det er lyst, vil vi slå av lyset

Eksempel 6

Lyssensor



eller



Merk! Utseendet kan variere

Lyssensor, Fotocelle (Photoresistor), LDR (light dependent resistor)

En Lyssensor/Fotocelle er en sensor som brukes for å registrerer lys.

Den virker slik at motstanden avtar med økende lysintensitet (sterkere lys).

Eksempel 6

Lyssensor - Bruksområder

Fotoceller har mange anvendelser i måleteknikk (solbatterier, lysmålere), signaltransmisjonsteknikk (bl.a. billedrør for fjernsyn), og reguleringsteknikk, bl.a. automatisk inn- og utkobling av gatelys, lyskontrollerte åpnings- og lukkemekanismer for dører.

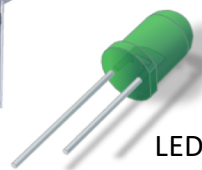
Eksempel 6

Nødvendig utstyr

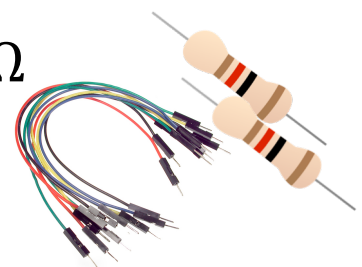
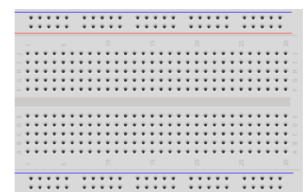
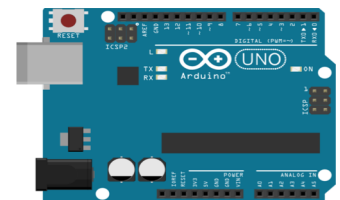
- Arduino
- Breadboard
- Lyssensor
- LED
- Motstander, $R = 330\Omega$, $R = 10\text{ k}\Omega$
- Ledninger (Jumper Wires)



eller

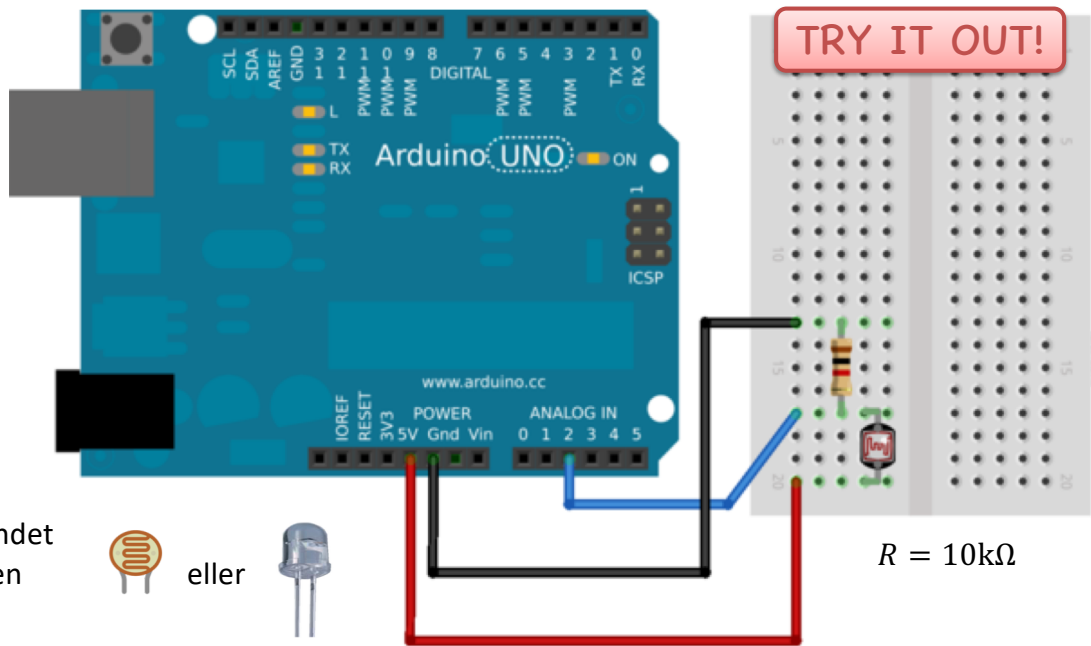


LED



Eksempel 6

Oppkobling



Merk! Utseendet på lyssensoren kan variere



eller



TRY IT OUT!

R = 10kΩ

På de fleste lyssensorer spiller det ikke noen rolle hvilken vei du putter pinnene, men hvis det ikke virker prøv å bytt om de 2 pinnene på koblingsbrettet

Made with Fritzing.org

Eksempel 6

Arduino Program

```
int photocellPin = 2;
int photocellReading;

void setup(void)
{
  Serial.begin(9600);
}

void loop(void)
{
  photocellReading = analogRead(photocellPin);

  Serial.print("Analog reading = ");
  Serial.println(photocellReading);

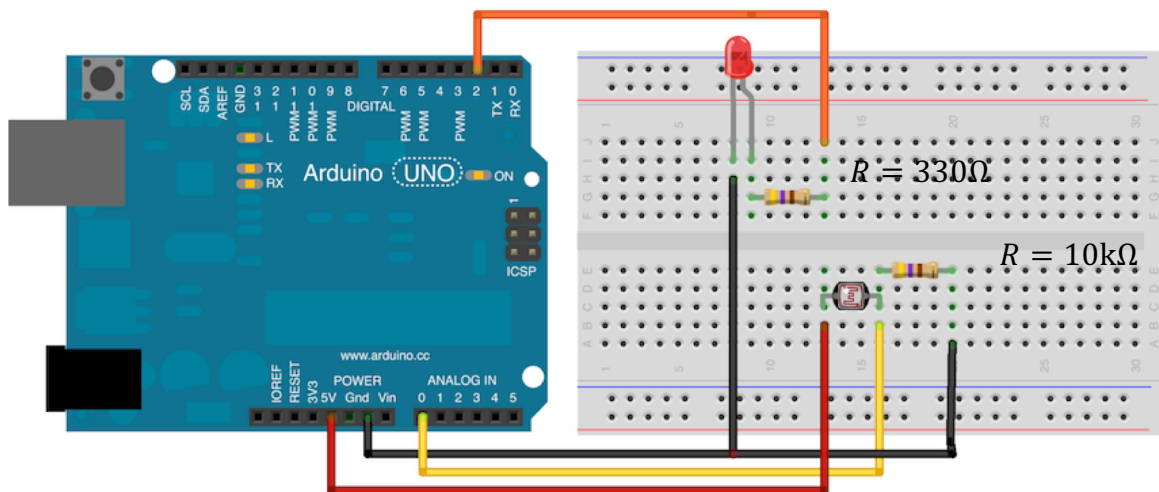
  delay(1000);
}
```

Dette programmet leser verdien fra lysmåleren og skriver denne verdien til Serial Monitor

TRY IT OUT!

Eksempel 6

Oppkobling



Made with Fritzing.org

Merk! Utseendet på lysensoren kan variere



Eksempel 6

Arduino Program

```
int photocellPin = 0;
int ledPin = 2;
int photocellReading;
const float limit = 100;

void setup(void)
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}

void loop(void)
{
  photocellReading = analogRead(photocellPin);
  Serial.print("Analog reading = ");
  Serial.println(photocellReading);

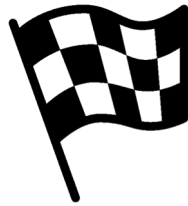
  if (photocellReading < limit)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }

  delay(1000);
}
```

Dette programmet leser verdien fra lysmåleren og skriver denne verdien til Serial Monitor. Hvis verdien er mindre enn en grenseverdi, slås lyset på. Hvis verdien er større enn denne grensen, slås lyset av.

TRY IT OUT!

Eksempel 6



|Slutt på eksemplet|

Eksempel 7

TRY IT OUT!

Termistor



Introduksjon

I dette eksemplet vil vi bruke en liten termistor til å lese temperaturen i rommet.

I dette eksemplet vil vi bruke en av de "Analoge Inn" portene på Arduino.

Eksempel 7



Termistor

En termistor er en elektronisk komponent som endrer motstand med temperaturen – såkalt Resistance Temperature Detectors (RTD). Den brukes ofte som temperatursensor.

Vår Termistor er en såkalt NTC (Negative Temperature Coefficient). I en NTC Termistor minsker motstanden når temperaturen stiger.

Det er en ulineær sammenheng mellom motstand og spenning. For å finne Temperaturen kan vi bruke følgende likning (Steinhart-Hart equation):

$$\frac{1}{T} = A + B \ln(R) + C(\ln(R))^3$$

der A, B, C er konstanter

[Wikipedia]

$$A = 0.001129148, B = 0.000234125 \text{ and } C = 8.76741E - 08$$

Eksempel 7



NTC Termistor - Datablad

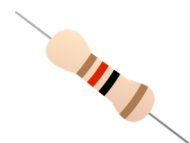
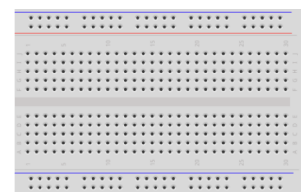
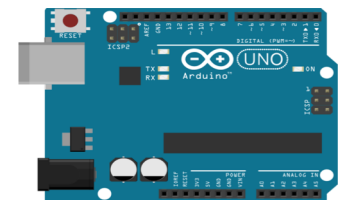
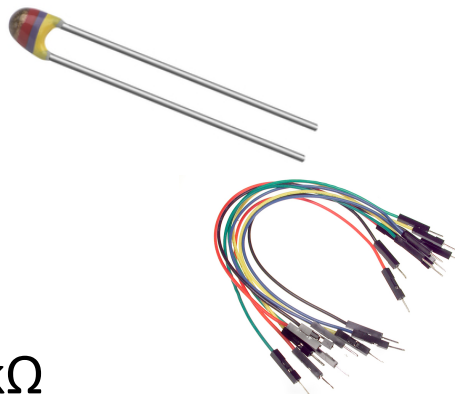
Technical data	
Resistance @ 25°C	10 kΩ
Temperature range	-40...+125 °C
Power max.	500 mW
Pitch	2.54 mm
Resistance tolerance	±5 %
W _{25/100} value	3977 K
B value tolerance	±0.75 %
Thermal time constant	15 s

Datablad: <https://www.elfadistelec.no/no/ntc-motstand-kablet-10-kohm-vishay-ntcle100e3103jb0/p/16026041?q=160-26-041&page=1&origPos=1&origPageSize=50&simi=98.0>

Eksempel 7

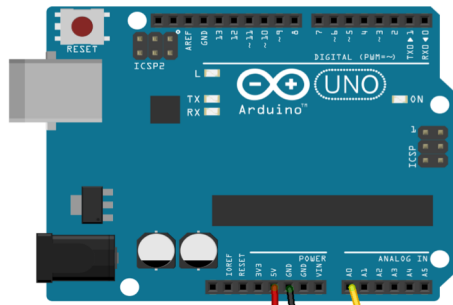
Nødvendig utstyr

- Arduino
- Breadboard
- Termistor
- LED
- Motstand 10 kΩ
- Ledninger (Jumper Wires)

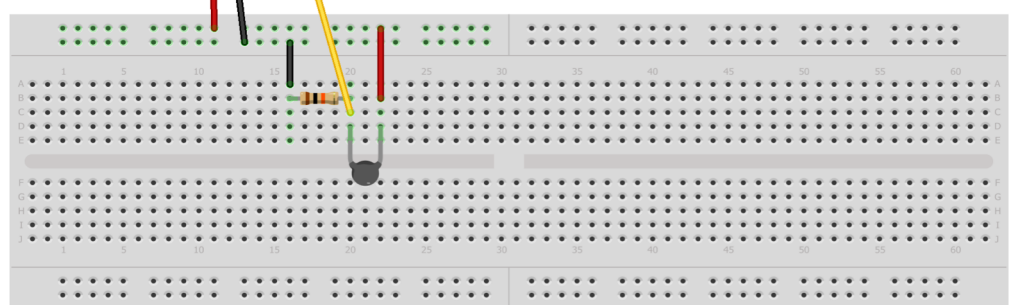


Eksempel 7

Oppkobling



TRY IT OUT!

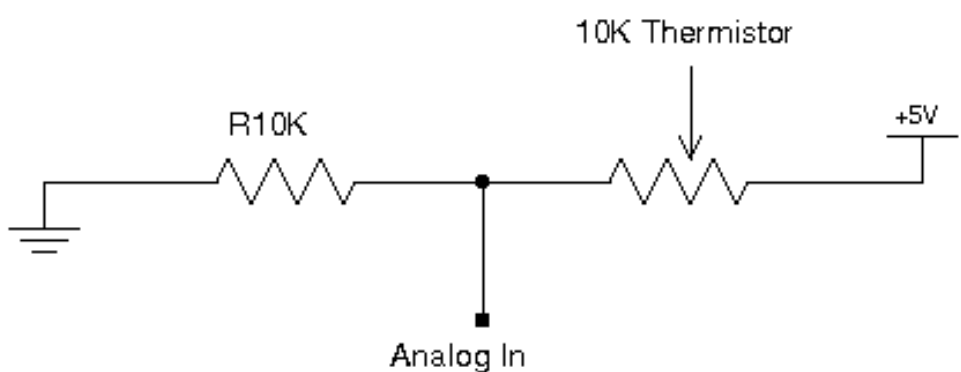


Eksempel 7

Made with Fritzing.org

Dette kalles en spenningsdeler:

Oppkobling



Eksempel 7

[<https://no.wikipedia.org/wiki/Spenningsdeler>]

Arduino Program

Eksempel 7

```
const int temperaturePin = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int temperature = getTemp();
  Serial.print("Temperature Value: ");
  Serial.print(temperature);
  Serial.println("*C");
  delay(1000);
}

double getTemp()
{
  // Inputs ADC Value from Thermistor and outputs Temperature in Celsius

  int RawADC = analogRead(temperaturePin);
  long Resistance;
  double Temp;

  // Assuming a 10k Thermistor. Calculation is actually: Resistance = (1024/ADC)
  Resistance=((10240000/RawADC) - 10000);

  // Utilizes the Steinhart-Hart Thermistor Equation:
  // Temperature in Kelvin = 1 / {A + B[ln(R)] + C[ln(R)]^3}
  // where A = 0.001129148, B = 0.000234125 and C = 8.76741E-08

  Temp = log(Resistance);
  Temp = 1 / (0.001129148 + (0.000234125 * Temp) + (0.0000000876741 * Temp * Temp * Temp));
  Temp = Temp - 273.15; // Convert Kelvin to Celsius
  return Temp; // Return the Temperature
}
```

TRY IT OUT!

Ekstraoppgaver

TRY IT OUT!

Prøv ut følgende:

- Bruk både TMP36 Temperatursensoren vi brukte tidligere og Termistoren.
- Sammenlign verdiene fra Termistoren med TMP36 Temperatursensoren vi brukte tidligere. Vis begge verdiene i Serial Monitor.
- Tegn krets og oppkobling vha Fritzing programvaren

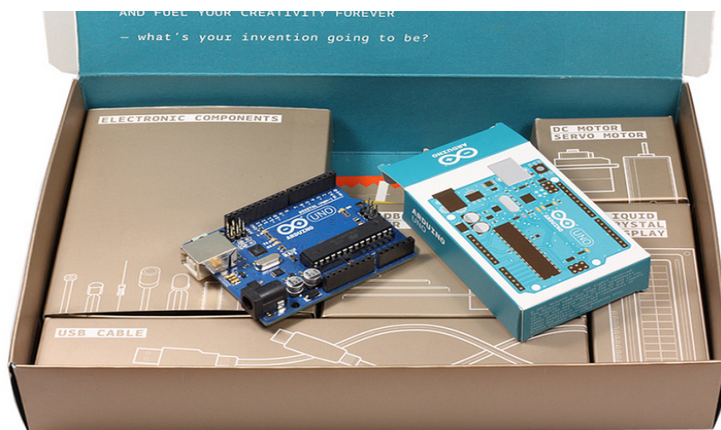
Eksempel 7



| Slutt på eksemplet |

Ønsker du å kjøpe din egen Arduino?

Kjøp f.eks. et “Starter Kit” som inneholder alt du trenger



Disse kan kjøpes mange steder på nettet. Lykke til med videre utforskning av Arduino, elektronikk, sensorteknologi, måleteknikk og programmering

Play and Explore

| Programmering og Arduino |

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

