

Fysik B - Temperaturmåling med Arduino - Introduktion til Arduino Uno

Dette undervisningsforløb består af (3 moduler á 2x45 minutter)

Forudsætninger: Ohms lov, kendskab til kredsløb

Dette undervisningsforløb tager eleverne igennem linearisering af en temperatursensor, skalering af analog input og lidt simpel programmering (C++, med god stilladsering)

Indholdsfortegnelse

| | |
|---|----|
| Installation af Arduino IDE styresystem til din PC | 2 |
| Opstart af Arduino-printet | 2 |
| Programmering af Arduino..... | 3 |
| Elektronikkomponenter, breadboard, modstand (resistor)..... | 5 |
| Opbygning af et eksternt kredsløb med en LED, ohms lov | 6 |
| Analog indgang og A/D-konverter | 9 |
| Betinget valg - anvendelse af A/D-count til at tænde LED | 12 |
| Digital til analog udgang (PWM)..... | 13 |
| Anvendelse af en variabel resistans til temperaturmåling..... | 16 |
| Fastlæggelse af overførselsfunktion (så der kan udlæses °C i stedet for A/D-count)..... | 19 |
| Programmering af overførselsfunktion | 21 |

Arduino er et billige, simpelt og robust microcontrollerprint (elektronik board) der kan anvendes til mange programmeringsopgaver i skolen og i DIY værkstedet (temperaturstyring, solfangertilt, højdemåling med tryksensor, robotter osv).

Bag Arduino står en non-profit organisation med en rigtig god hjemmeside.

Arduino er Open Source og alle hardware og software komponenter er fuldt tilgængelige. Der er et stort hjælpsomt community omkring Arduino (eksempelvis er der flere danske Arduino-grupper på Facebook)

Lektionerne er opbygget omkring et Arduino Uno microcontrollerboard der kan købes hos <https://arduinotech.dk/> til 91 kr. I øvelserne anvendes en NTC med ledninger og krympeflex (købt samme sted, 20 kr). NTC typen er ikke så vigtig fordi overførselsfunktionen lineariseres i øvelserne (af hensyn til skaleringen er 10 kΩ bedst)

Strømforsyning til Arduinoprintet foregår "automatisk" gennem USB-kabel fra PC.

Det fungerer fint med to elever omkring et board (tre kan gå, fire er for mange).

Hvis det første gang man som lærer skal i gang med Arduino, er det godt at begynde med at købe et starter kit (695 kr), hvor man får ledninger, breadboard, lysdioder, knapper osv med.

Man kan sagtens selv "læse op" og komme i gang. En lørdag formiddag med starterkit, youtube-videoer, Arduino hjemmesiden osv gør meget ved det.

Installation af Arduino IDE styresystem til din PC

Gå til: <https://www.arduino.cc/en/Main/Software>

Klik på dit styresystem.

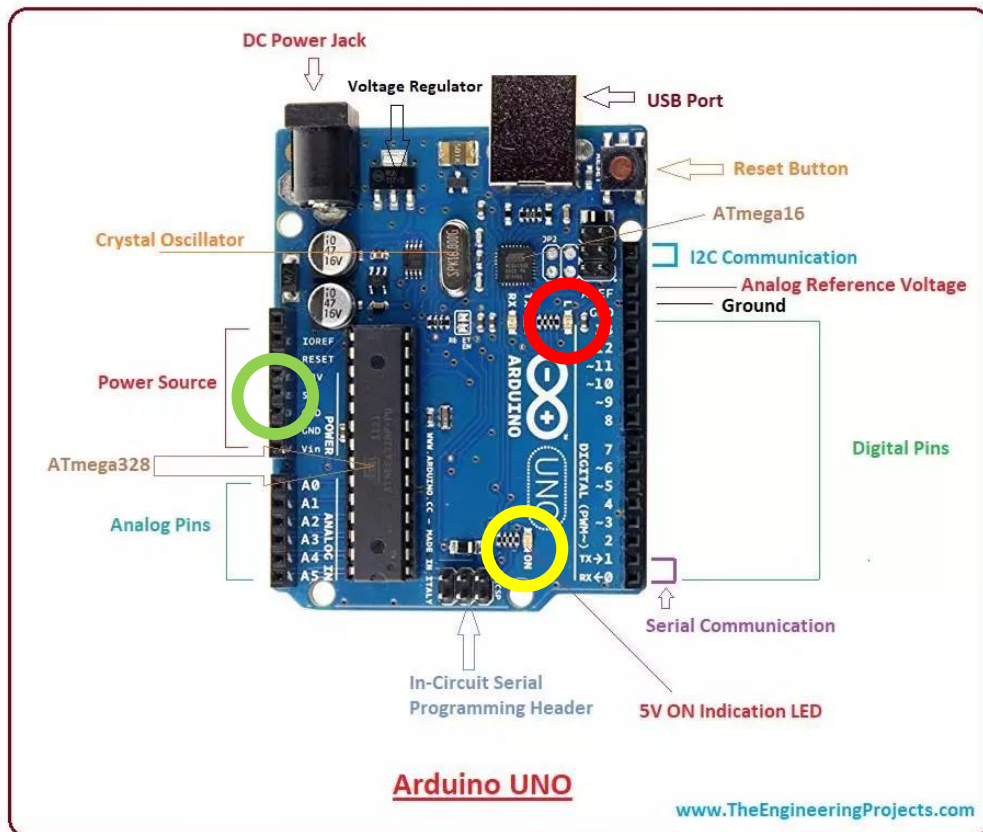
Klik "just download"

Opstart af Arduino-printet

Åbn Arduino appen.

Tilslut Arduino-printet med et USB kabel.

Kontrollere der er en grøn LED (ON) der lyser konstant - markeret med gul.



Billede af Arduino Uno printet (fra theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html)

Arduino-printet skal forbindes med appen.

I feltet "Select Board" vælges den USB-port som appen foreslår (fx COM3).



Når programmeringseditoren er åbnet, vælges den COM-port Arduino-printet er tilsluttet

Programmering af Arduino

Opgave 1a) - Første program - Tænd en lysdiode (LED)

Arduino printet er udstyret med en mikrocontroller som har forskellige ind- og udgange. En af udgangene er tilsluttet en LED der kan lyse. Udgangen kaldes "Pin 13".


Når Pin 13 anvendes som en udgang (OUTPUT), kan mikrocontrolleren sætte 5 volt eller 0 volt på den. Det kaldes HIGH eller LOW, 1 eller 0.

Når der er 5 V på Pin 13 vil LED lyse.

Det gøres i programmet ved, at der i koden (linje 8) skrives: `digitalWrite(13,HIGH);`

Først skal Pin 13 defineres som en udgang.

Det gøres ved at der i linje 3 skrives: `pinMode(13, OUTPUT);`

Nu er programmet færdig og skal tjekkes for fejl. Tryk på knappen "Verify" 

Efter koden er verificeret (kompileret) og uden fejl, skal den op i mikrocontrolleren.

Tryk på knappen "Upload" 

Vi er færdige. Kontrollere at LED lyser konstant.



Den røde cirkel markerer den LED der er forbundet til microcontrollerens pin 13

Opgave 1b) - Sluk LED

Ændr din kode så der nu i linje 8 står: `digitalWrite(13,LOW);`

Snak med din sidekammerat om hvad der sker med LED når programmet køres.

Kompiler koden (tryk på "Verify").

Læg koden op i mikrocontrolleren og tjek at LED er slukket.

Opgave 1c) - Blink med dioden

Hvis vi får programmet til at tænde LED, vente et sekund og så slukke LED igen (og gentage det) vil vi opleve, at LED blinker.

Et Arduino-program består af to dele.

En initialiseringsdel (kaldet setup) og hovedprogrammet/hovedløkken (kaldet loop).

I initialiseringen fortæller vi mikrocontrolleren om en Pin er en indgang eller en udgang. LED er tilsluttet pin 13 og det er en udgang (derfor skrev vi `pinMode(13, OUTPUT);`)

I hovedprogrammet gentages kommandoerne/linjerne igen og igen (det er en løkke).

Vores program skal tænde LED, vente, slukke LED, vente, tænde LED osv.
Fordi Arduino-koden i "loop" gentages, kan vi nøjes med at skrive:

```
Tænd LED,  
Vent 1000 millisekunder  
Sluk LED  
Vent 1000 millisekunder
```

1) Nu skal du ændre din kode (fra linje 8), så den ser således ud

```
digitalWrite(13,HIGH); // Her skrives "1" til udgangen (tænd LED)  
delay(1000);           // Vent 1000 ms  
digitalWrite(13, LOW); // Her skrives "0" til LED (sluk LED)  
delay(1000);           // Vent 1000 ms
```

Symbolerne "//" bruges af programmøren når der skal skrives en kommentar.

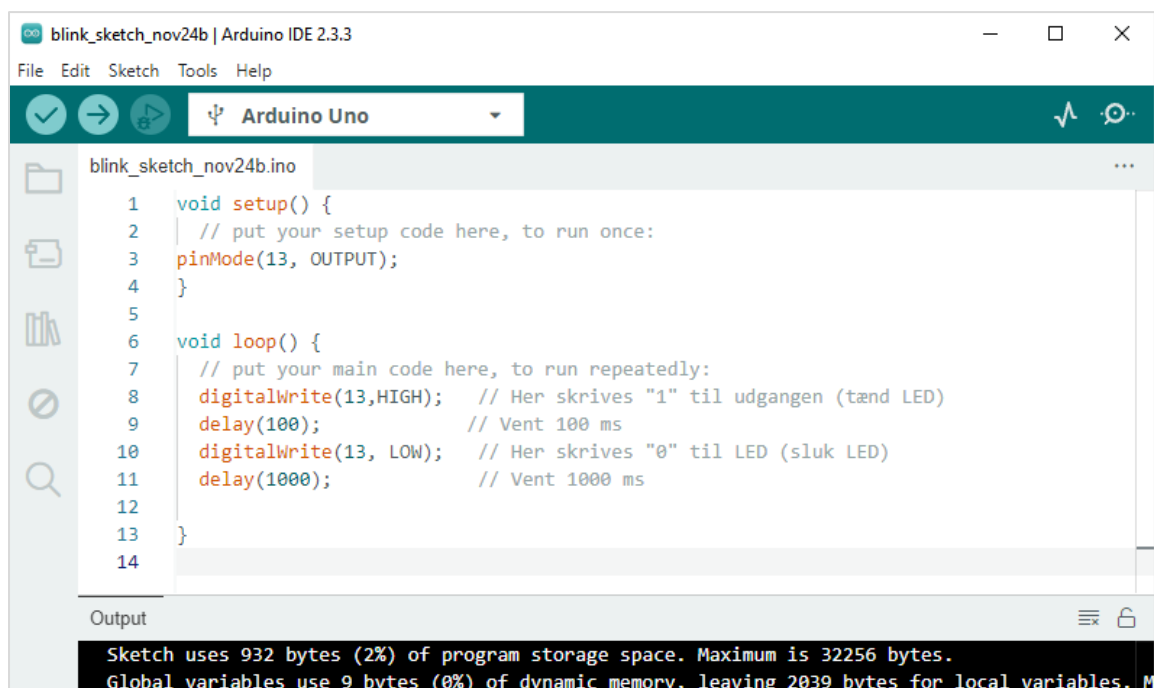
2) Når du har indtastet ovenstående, kompileres koden og lægges op i microcontrolleren.

3) Tjek at LED blinker.

4) Når koden virker, så husk at gemme den (File - Save As...)

Opgave 1d) Heart Beat

Lav din kode om således at blinket minder om "Heart Beat" - ændr tænd-tiden til 100 ms



Her er et screenshot af hele koden til heart-beat programmet.

Opgave 1e) Hvor hurtig er vores øjne

80 ms ON og 80 ms OFF svarer til en periodetid på 160 ms, eller $\frac{1}{0,16\text{ s}} = 6,25\text{ Hz}$

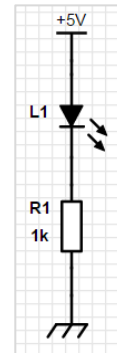
1) Prøv at indstille begge delay-tider til 80 ms - blinker LED ?

2) Undersøg hvor lave delay-tiderne skal være, inden vi ikke kan se at LED blinker.

Elektronikkomponenter, breadboard, modstand (resistor)

Den programmering vi lavede for at få LED til at blinke, er software.

Elektronikkomponenter er hardware. Komponenterne anvendes til at få sensorer, aktuatorer og lamper til at fungere. Komponenternes navne er eksempelvis kondensator (kapacitor), modstand (resistor), microcontroller, ledninger, transistor, gates, LED.

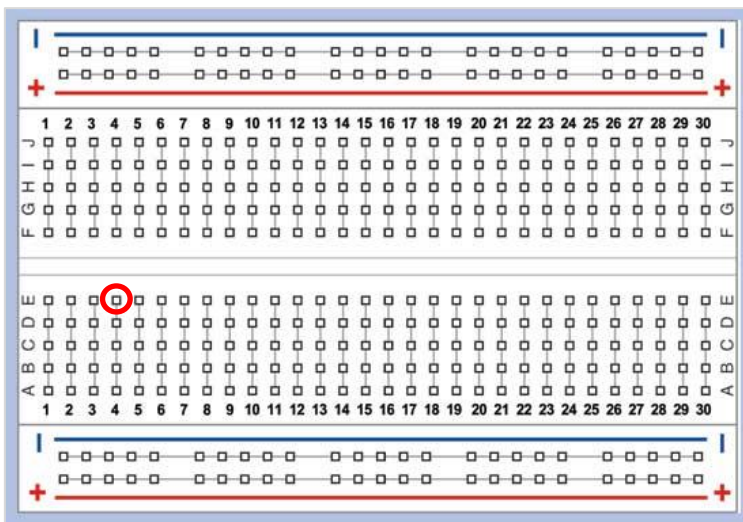


Her er et billede af forskellige elektronikkomponenter og til højre et eksempel på et el-diagram

Komponenterne forbindes i kredsløb. For at softwaren kan få en LED til at blinke, skal den forbindes til et ben på microcontrolleren. Det gøres med en ledning.

Den LED vi programmerede er fastmonteret på printet, til microcontrollerens ben 13. Når vi bruger Arduino i projekter, anvendes et breadboard (hul-board) så det er nemt at opbygge elektriske kredsløb. Et breadboard er en plastikblok med mange huller i.

På billedet ses hvorledes de enkelte huller er forbundet med hinanden



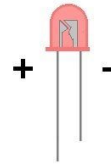
Her er et billede af et breadboard - hullet E4 er markeret med en rød ring

Komponenternes ben stikkes ned i hullerne og derved bliver de forbundet til hinanden.

Opbygning af et eksternt kredsløb med en LED, ohms lov

Det skal opbygges et elektrisk kredsløb med en LED der gennem en strømbegrænsende resistor er monteret til +5V dc spændingsforsyning, som vist på el-diagrammet.

Det lange ben på en LED komponent er anoden (plus).
Det korte ben er katoden, der skal tilsluttes resistoren.



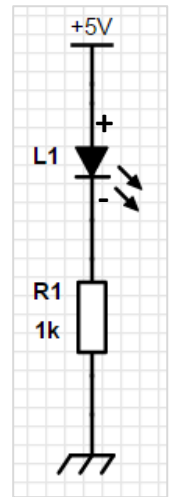
Kredsløbsbeskrivelse.

LED vil lyse konstant.

Spændingsfaldet hen over en LED er typisk $U_{LED} = 2,0 \text{ V}$.

En LED kan maksimalt tåle 20 mA (men lyser fint ved 2 mA).

Resistor $R1 = 1k\Omega$



Opgave 2a) - Beregning af den strømbegrænsende resistor

På el-diagrammet ses, at L1 (vores LED) er i serie med resistansen R1.

Kredsløbet forsynes med +5V dc, så der flyder den samme strøm (I_{R1}) i LED (L1) som der gør i R1.

Vi har fået oplyst at spændingsfaldet over LED er +2,0 V

1) Beregn spændingsfaldet over R1.

2) Beregn strømmen I_{R1} (vi husker ohms lov hvor $I = \frac{U}{R}$)

3) Kig på beregningerne - brænder LED af, når vi tænder for spændingen (altså: kontroller at strømmen, I_{R1} er mindre end 20 mA).

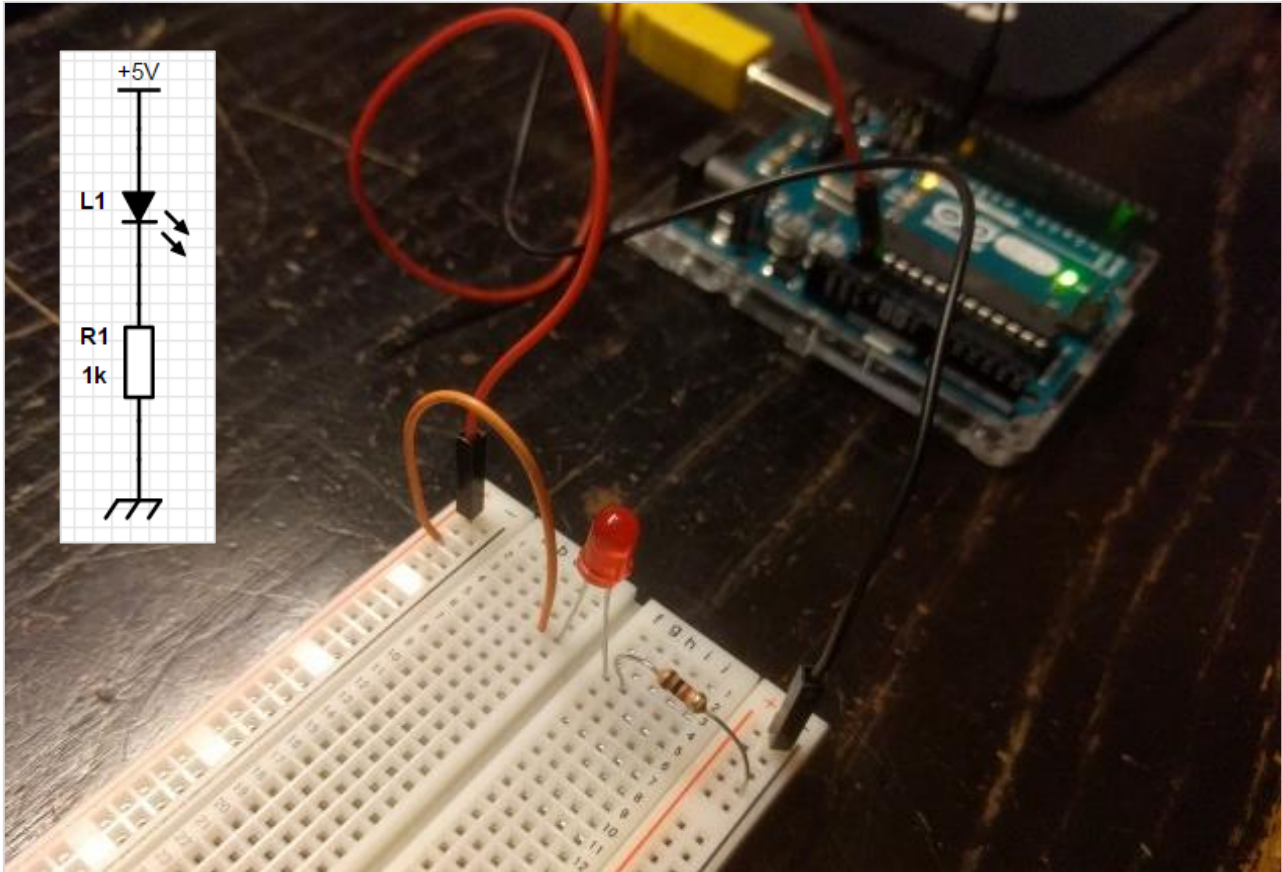
4) Hvis det er OK, kan vi gå videre og opbygge kredsløbet

Svar: $U_{R1} = 3 \text{ V}$. $I_{R1} = 3 \text{ mA}$, Ja det er OK: $3\text{mA} < 20 \text{ mA}$.

Opgave 2b) - Opbygning af elektronikkredsløbet på breadboard

Du skal bruge:

- 1 stk breadboard.
- 1 stk Arduino print (Arduinoen er +5V spændingsforsyning)
- 1 stk LED
- 1 stk resistor (1k Ω har farvekoden brun-sort-rød)
- 3 stk ledninger (en rød til +5V, en sort til 0V og yderligere en)



Det lange ben på den røde LED er tilsluttet +5V gennem den orange ledning.

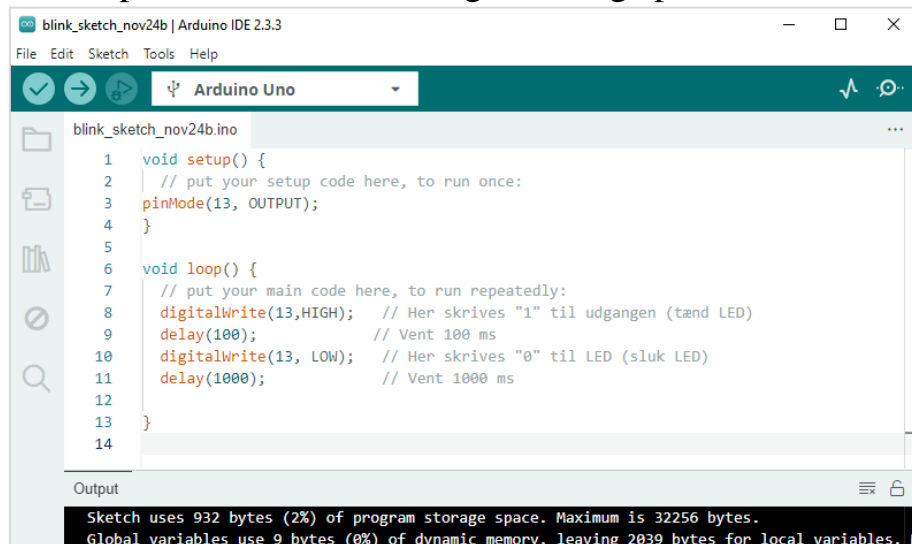
- 1) Placer LED'ens lange ben i hullet E4 og det korte ben i F4.
- 2) En modstand (1k Ω brun-sort-rød) skal placeres mellem LED og 0V. Resistorens ene ben skal i hullet G4 og det andet ben skal til 0V, hul - 4.
- 3) Arduinoen har en lang connector. Det er trykt teksten +5 ud for hul nr. 5 (markeret med en grøn cirkel på side 2).
Set en rød ledning fra Arduinoens +5 hul til breadboard hul +1
- 4) Den sorte ledning sættes i hullet markeret GND på Arduinoens lange connector. Den anden ende sættes i breadboard hul - 1.
- 5) For at forbinde LED med +5V sættes en ledning i breadboard hul D4 til hul + 4.
- 6) Tjek at LED lyser.

Opgave 2c) - lad Arduino tænde og slukke LED på breadboard

Nu skal LED på breadboard tilsluttes til Arduino pin 12. Fordi så kan vores software lade microcontrolleren lægge +5 volt på LED (lyser) eller 0 V (slukket).

1) Flyt den røde ledning fra den lange Arduino connector +5 (pin 5) til den lange connector på den modsatte side af Arduino. Ledningen skal i hul nr. 6 (der står 12 på connectoren).

2) Vi skal have opdateret softwaren til også at bruge pin 12. Indlæs dit program



```
blink_sketch_nov24b | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
blink_sketch_nov24b.ino
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(13, OUTPUT);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   digitalWrite(13,HIGH); // Her skrives "1" til udgangen (tænd LED)
9   delay(100);           // Vent 100 ms
10  digitalWrite(13, LOW); // Her skrives "0" til LED (sluk LED)
11  delay(1000);           // Vent 1000 ms
12 }
13
14
Output
Sketch uses 932 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. M
```

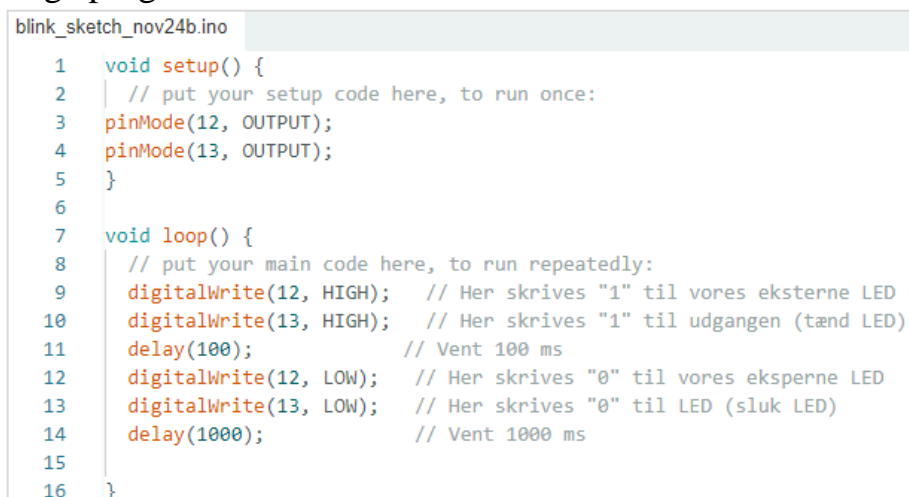
Der arbejdes videre på programmet (læg mærke til delay tiderne er "heart beat")

3) Vi skal have pin 12 med i programmet.

Tilføj en linje inden linje 3 og skriv `pinMode(12, OUTPUT);`

4) I "loop" indsættes to linjer hvor der skrives til pin 12.

Det færdige program ser således ud:



```
blink_sketch_nov24b.ino
1 void setup() {
2   // put your setup code here, to run once:
3   pinMode(12, OUTPUT);
4   pinMode(13, OUTPUT);
5 }
6
7 void loop() {
8   // put your main code here, to run repeatedly:
9   digitalWrite(12, HIGH); // Her skrives "1" til vores eksterne LED
10  digitalWrite(13, HIGH); // Her skrives "1" til udgangen (tænd LED)
11  delay(100);             // Vent 100 ms
12  digitalWrite(12, LOW);  // Her skrives "0" til vores eksterne LED
13  digitalWrite(13, LOW);  // Her skrives "0" til LED (sluk LED)
14  delay(1000);            // Vent 1000 ms
15
16 }
```

5) Læg programmet op i Arduino og tjek at vores eksterne breadboard LED blinker

Analog indgang og A/D-konverter

Microcontrolleren kan læse en analog værdi.

En analog spænding (fx 2,36 V) fra den virkelige verden, kan omsættes til et digitalt tal inde i Arduinoen som softwaren kan arbejde med.

Der tales om en "analogue to digital converter" forkortet ADC, eller bare A/D-konverter. En analog til digital konvertering foregår ved, at det analoge signal måles relativt til 5 V.

$$\frac{2,36 \text{ V}}{5,00 \text{ V}} \cdot 100 = 47,2 \%$$

De 2,36 V svarer med andre ord til 47,2 % af 5,00 V.

Arduinoen har en 10-bit A/D konverter, så den leverer et tal mellem 0 og 1024.

Et 10-bit binært tal har en opløsning på $2^{10} = 1024$ step (kaldet A/D-count).

0 A/D-count svarer til 0,00 V og 1024 A/D-count svarer til 5 V (100 %).

Det digitale tal der leveres fra A/D-converteren er 47,2 % af 1024 = 483 A/D-count

Opgave 3a) - A/D-konverterens opløsning

Der er 1024 A/D-count fordelt på 5 V.

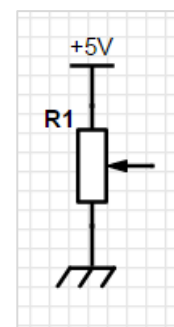
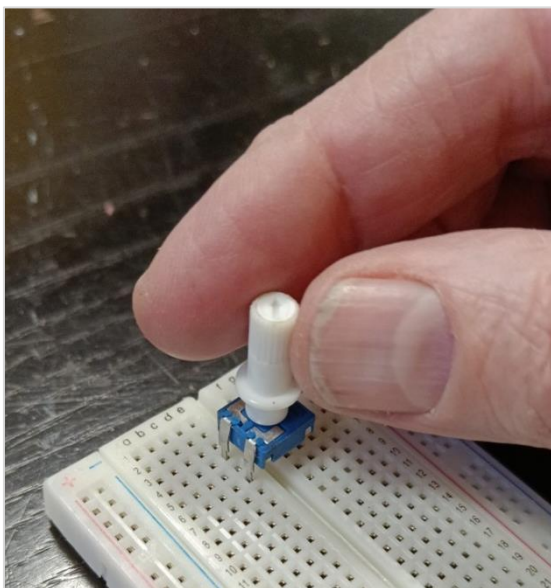
Beregn hvor mange mV der er på et A/D-count.

En variable resistans - et potentiometer

En gammeldags bilradio har en drejeknap til at skrue op og ned for lydstyrken.

Elektronisk gøres dette med en variable resistor, kaldet et potentiometer.

Et potentiometer har tre ben. To ben som en almindelig resistans og så et tredje ben, hvis resistans ændres i takt med den vinkel potentiometerarmen drejes.



Elektrisk symbol
for potentiometer.

Et potentiometer bruges til at omsætte en rotation på 0-270° til en variable spænding.

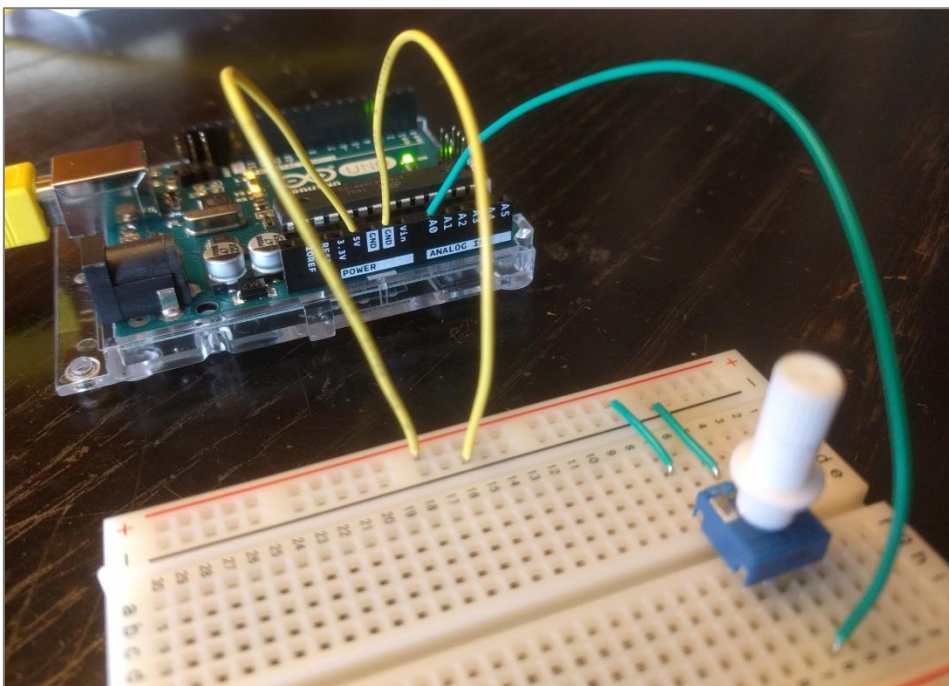
Et 10 kΩ potentiometer kan ændre resistansen på det midterste ben (udgangen) fra 0 Ω til 10 kΩ i forhold til et af de to andre ben.

Når potentiometeret påtrykkes en spænding på +5V (som vist herover på el-diagrammet), vil udgangsspændingen kunne ændres fra 0V til +5V (afhængig af, hvor meget potentiometer-armen er drejet).

Ved at bruge et potentiometer som variabel spændingsforsyning kan Arduinoens A/D-konverter afprøves.

Opgave 4a) - opbygning af et kredsløb med et potentiometer

Der skal opbygges et nyt kredsløb som kan levere en variabel spænding (0V til +5V) som tilsluttes en analog indgang på microcontrolleren (den indgang som hedder A0). Arduino-printet bruges til at levere spændingen til potentiometeret (så risikerer vi ikke at en overspænding brænder Arduinoen af).



Et potentiometer monteres mellem 5V og GND. Potentiometerets udgang forbindes til A/D (A0)

Et potentiometer sættes i E4, E7 (og den anden siden i G6)

Der skal bruges fem ledninger:

- En ledning fra 0V (hul -5) til potentiometeret (C5).
- En ledning fra +5V (hul +8) til potentiometeret (B7).
- En ledning Arduino pin A0 til potentiometeret (J6).

Strømforsyning (0V), fra Arduino GND til hul -16
Strømforsyning (+5V), fra Arduino hul +18

Opgave 4b) - Opsætning og programmering af A/D

For at vi som programmører kan se det tal A/D-converteren generer, anvendes et monitor-vindue. Monitorvinduet åbnes ved at vælge topmenuen (i Arduino IDE appen, når programmet er up-loadet til Arduino-printet): Tools - Serial Monitor.

Arduino-printet skriver i monitor-vinduet via en seriel kommunikationsport. Kommunikationsporten opsættes i programmet, i initialiseringsrutinen.

1) Kopier følgende over i Arduino editoren og up-load det til Arduino printet:

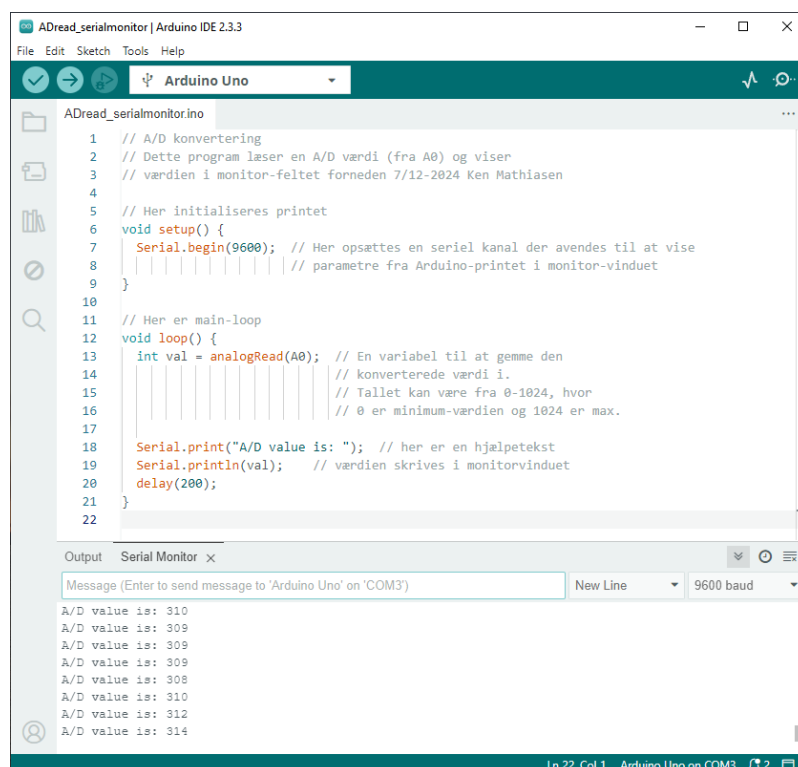
```
// A/D konvertering
// Dette program læser en A/D værdi (fra A0) og viser
// værdien i monitor-feltet forneden 7/12-2024 Ken Mathiasen

// Her initialiseres printet
void setup() {
  Serial.begin(9600); // Her opsættes en seriel kanal der avendes til at vise
                     // parametre fra Arduino-printet i monitor-vinduet
}

// Her er main-loop
void loop() {
  int val = analogRead(A0); // En variabel til at gemme den
                           // konverterede værdi i.
                           // Tallet kan være fra 0-1024, hvor
                           // 0 er minimum-værdien og 1024 er max.

  Serial.print("A/D value is: "); // her er en hjælpetekst
  Serial.println(val);           // værdien skrives i monitorvinduet
  delay(200);
}
```

Arduino-koden til læsning af en A/D værdi. A/D-værdien vises i serial monitor (Tools - Serial Monitor).



Monitorvinduet vist for neden. Når potentiometerarmen drejes ændrer tallene sig.

Betinget valg - anvendelse af A/D-count til at tænde LED

Nu skal der laves et lille program der tænder LED når potentiometeret er drejet til højre (skruet op, med uret) og som slukker LED når potentiometeret er drejet til venstre.

Det foregår ved at vi måler spændingen på den analoge indgang.

Når spændingen er over 2,5 V (når A/D-count > 512) tænder vi LED.

Når spændingen er under 2,5 V (når A/D-count ≤ 512), slukker vi LED.

Opgave 5a) - Betinget valg, If-then-else

1) Indsæt nogle linjer efter linje 21 (Tuborg-klammen "}" skal flyttes lidt længere ned)

2) Indsæt denne kode-stump. Begynd i linje 23 (inden "}")

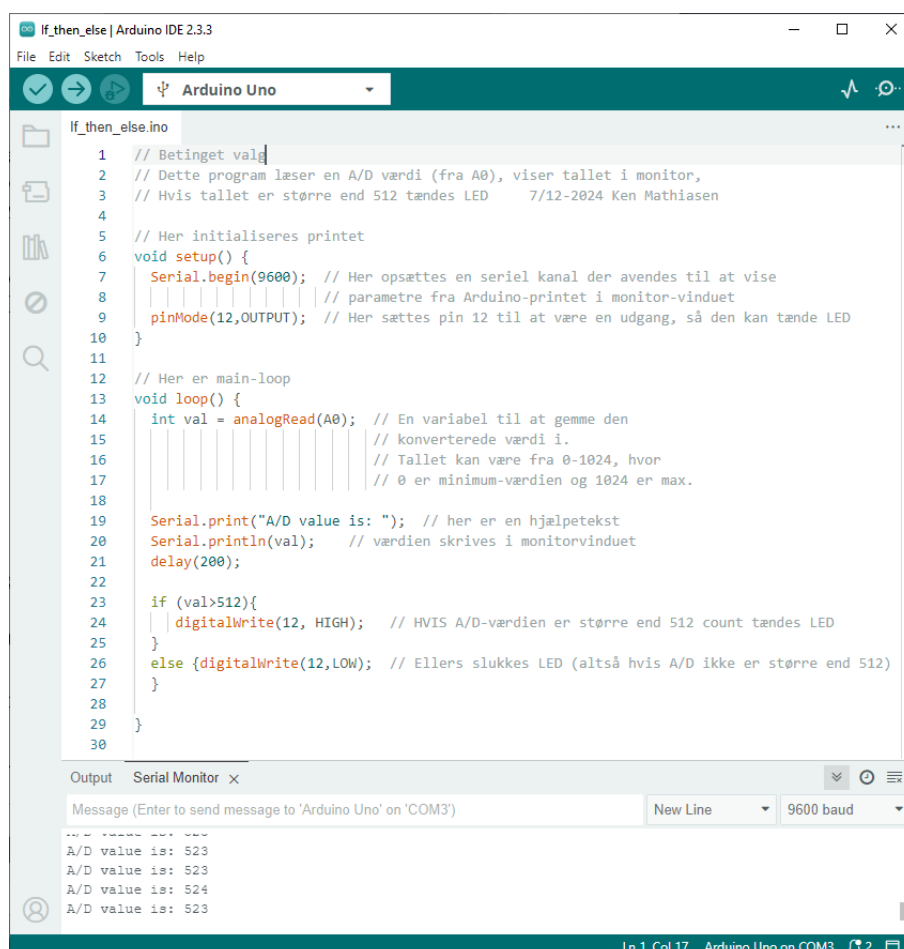
```
if (val>512){  
    digitalWrite(12, HIGH);  
}  
else {digitalWrite(12,LOW);  
}
```

3) Når du har indtastet ovenstående, kompileres koden og lægges op i microcontrolleren.

4) Husk at tænde for Serial Monitor (Ctrl-Shift-M) så vi kan se resultatet fra ADC.

5) Tjek at LED lyser, når "A/D-value is: 512" eller mere (drej på potentiometeret).

6) Når koden virker, så husk at gemme den (File - Save As...)



Her ses koden i Arduino IDE med monitorvinduet vist for neden.

Digital til analog udgang (PWM)

En digital udgang kan levere 0V eller 5 V - den kan tænde eller slukke LED.

Ved at tænde og slukke hurtigt, fx med 500 Hz (10 000 μ s ON og 10 000 μ s OFF) kan man lave en blinkefri LED som vi tidligere afprøvede. MEN lysstyrken vil være reduceret i forhold til når LED er tændt hele tiden. Fordi halvdelen af tiden er LED slukket og dermed kun tændt i halvdelen af tiden.

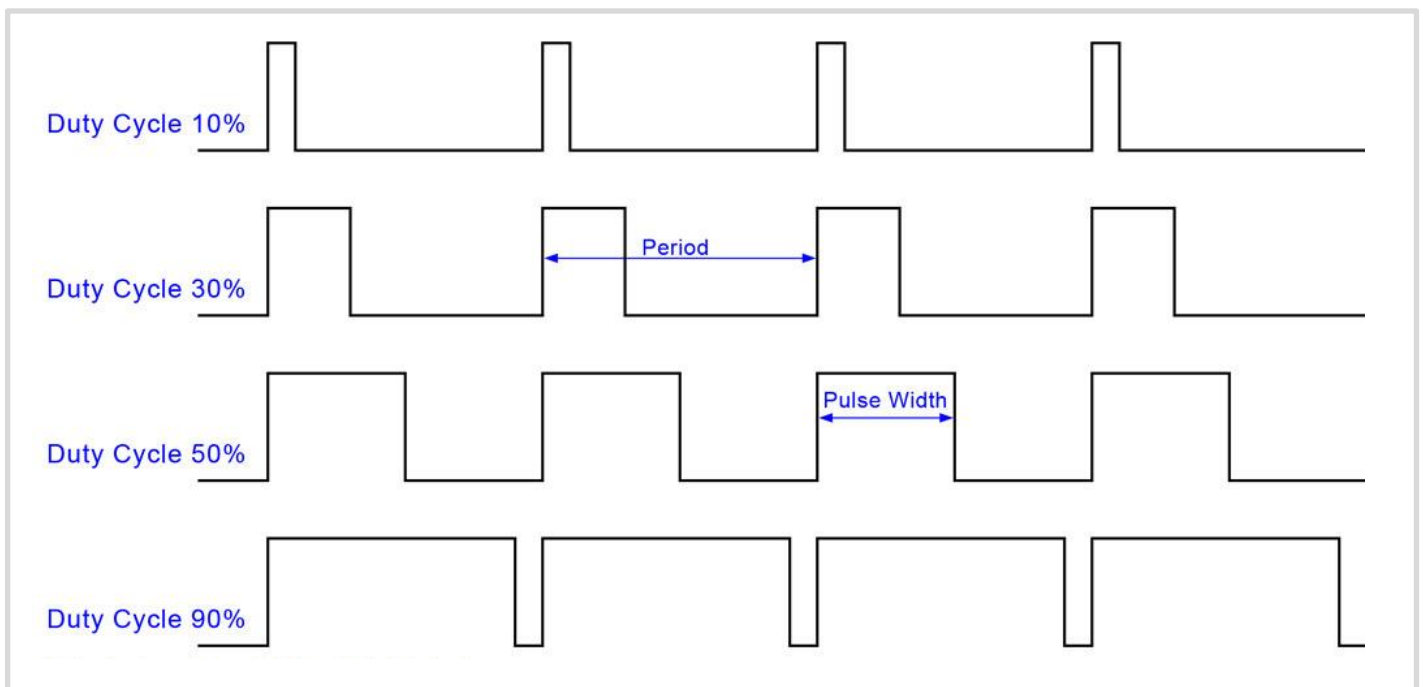
Ved at lade LED være tændt 95 % af tiden vil den lyse med næsten fuld styrke (LED vil være slukket i 5 % af tiden).

Ved at lade LED være tændt i 70 % af tiden (og slukket i 30 % af tiden) dæmpes lyset.

Fænomenet kaldes "Puls Width Modulation", forkortet til PWM.

Når LED er tændt i 70 % af tiden tales om at "duty-cycle" er 70 %

Her er fænomenet illustreret med fire pulser



Figuren viser forskellige duty-cycles - frekvensen er konstant.

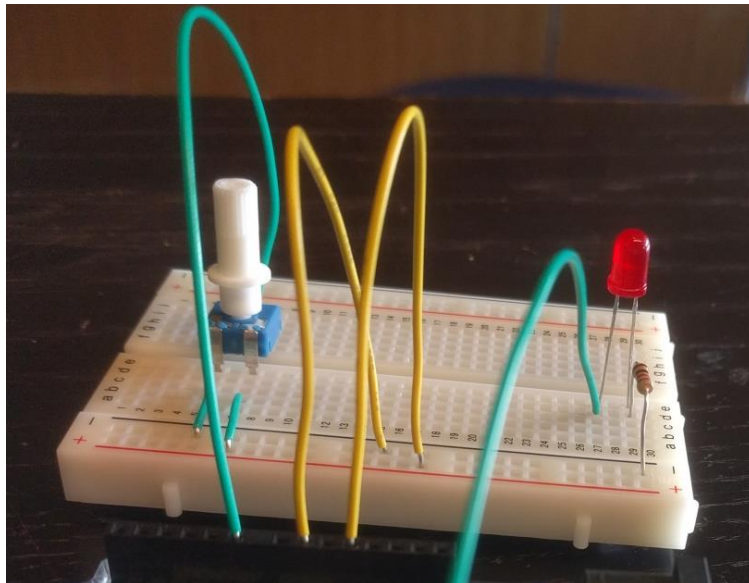
Ved at anvende pulsviddemodulation kan energien i fx en elektrisk radiator varieres, selvom man kun har en digital udgang til rådighed. Den afsatte energi svarer til duty-cycle-procentdelen af den maksimale energi (arealet under kurven).

Opgave 6a) - En LED tilsluttes en digital udgang der PWM modulers

1) Sæt en LED i breadboard'et.

Det lange ben skal forbindes til Arduino-printets pin 13 med en ledning.

Husk at sætte en resistor i serie med LED (mellem det korte ben og 0V).



På det breadboard hvor potentiometeret er, monteres en LED (med resistor)

Opgave 7a) - Programmering så en digital port kan anvendes analogt

Programmet indledes med at initialisere Pin12 som digital output (i void setup).

Se kodeeksemplet på næste side.

Men allerførst defineres en variabel der skal bruges til at gemme dutycycle værdien i.

I main-loop (void loop) læses A/D-converteren og en dutycycle-variable der markerer tænd-tiden udregnes (det er i princippet dutycycle-procenten omregnet til antal μ s).

Main-loop delay består at kommandoen `delayMicroseconds(xxyy)` der sætter tænd-tiden for LED. Sluk-tiden findes ved den maksimale tænd-værdi (10240μ s) fratrækkes tænd-tiden.

Tallet 10240 stammer fra begrænsninger i Arduinos operativ system.

Af praktiske hensyn anvendes laves koden ikke hele perfekt. Fx er udlæsningen i monitor ikke helt præcist og man vil kunne udlæse en duty-cycle på 102.

1) Kopier følgende ind i Arduino IDE (husk at gemme det)

```
// PWM moduleret udgang
// Dette program læser en A/D værdi (fra A0) og bruger tallet til en angivelse af
dutyCycle.
// Med et pot-meter skruer brugeren op/ned for lysstyrken i en LED. 8/12-2024 Ken
Mathiasen

// Her defineres variablene
int DutyCycle = 10; // Der oprettes en variable til at holde værdien for dutyCycle
// 0 svarer til 0 % dutyCycle (OFF). 10240 svarer til 100,0 % (ON)

// Her initialiseres printet
void setup() {
  Serial.begin(9600); // Her opsættes en seriel kanal til monitor
  pinMode(12,OUTPUT); // Her sættes pin 12 til at være en udgang, så den kan tænde LED.
}
// Her er main-loop
void loop() {
  int val = analogRead(A0); // En variabel til at gemme den A/D værdien i
  // Tallet kan være fra 0-1024. 1024 er max.

  DutyCycle = 10*val; // A/Dværdi gøres klar til delay (0-10240 mikro sek)
  digitalWrite(12,HIGH); // Her skrives "1" til udgangen (tænd LED)
  delayMicroseconds(DutyCycle); // Vent xxyy mikrosekund (puls-tid)
  digitalWrite(12,LOW); // Her skrives "0" til LED (sluk LED)
  delayMicroseconds(10240-DutyCycle); // Vent (pause-tiden beregnes udfra puls-tiden)

  Serial.print(" Dutycyclen er (%): "); // Skriv en hjælpetekst i monitor
  Serial.println(DutyCycle/100); // DutyCycleværdien (i procent) skrives i monitor
}
```

2) Kompiler koden og læg den op i microcontrolleren.

3) Husk at tænde for Serial Monitor, så vi kan se resultatet af udregningen.

4) Tjek at der kan skrues op og ned for LED styrken.

Opgave 7b) - Programmering, voltmeter med serial monitor

Vi ved at 5V repræsenterer 1024 A/D-count og 0 V svarer til 0 A/D count.

Så ved at kende antal A/D-count kan vi beregne den påtrykte spænding.

Der er en lineær sammenhæng mellem A/D-count og den spænding der blev målt med A/D-konverteren, $f(x) = ax + b$ hvor x er antal A/D-count og y er den målte spænding. Vi har to punkter:

$$\begin{aligned}(x_1; y_1) &= (0; 0) \\ (x_2; y_2) &= (1024; 5)\end{aligned}$$

Hældningskoefficienten, $a = \frac{y_2 - y_1}{x_2 - x_1}$ beregnes til cirka $a = \frac{1}{200}$

Begyndelsesværdien beregnes formelen, $b = y_1 - a \cdot x_1$ til 0

Dermed fremkommer funktionen $f(x) = \frac{1}{200}x$

1) Ændr linje 26, så det er "val/200" der udlæses.

2) Ændr teksten i linje 25 så brugeren kan se at det er spændingen der udlæses.

Opgave 7c) - Udvidelse

Ændr programmet således at den fastmonterede LED (pin 13) lyser i takt med LED.

Ændr programmet således de to LED er i modfase. Når der skues op, lyser LED kraftigere, men den fastmonterede LED lyser samtidig svagere.

Anvendelse af en variabel resistans til temperaturmåling

For at microcontrolleren kan registrere en temperaturændring, anvendes en 10 k Ω NTC modstand (en resistor med negative temperature coefficient).

Det er en komponent med to ben der opfører sig som en resistor. Men resistorens værdi ændrer sig, som funktion af temperaturen.

NTC har en resistans på omkring 10k Ω ved stuetemperatur.

Resistansen aftager med voksende temperatur, så den ved 80°C måske er 1,2k Ω .

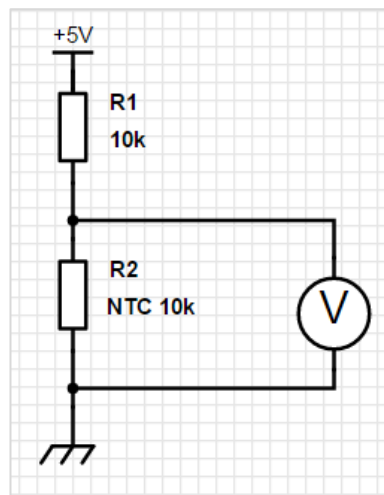
Microcontrolleren kan måle spænding (volt), ikke strøm (ampere) eller resistans (ohm). Derfor anvendes et lille kredsløb hvor NTC'ens modstandsændring omsættes til et spændingssignal imellem ca. 0,5 V til 3,5 V.

Ekstraopgave til de hurtige

Undersøg (mål) den resistans NTC har ved 80°C

Opgave 8a) - Dimensionering af elektrisk kredsløb

Det skal opbygges et elektrisk kredsløb med en NTC ($10\text{k}\Omega$ @ 20°C) der gennem en $10\text{k}\Omega$ resistor er monteret til $+5\text{V}$ dc spændingsforsyning, som vist på el-diagrammet.



Voltmeteret repræsenterer en analog indgang (A0) på microcontrolleren

På el-diagrammet ses, at $R2$ er i serie med resistansen $R1$.

Kredsløbet forsynes med $+5\text{V}$ dc, så der flyder den samme strøm i begge resistorer.

Antag at $R2$ $10\text{k}\Omega$ i de efterfølgende beregninger.

- 1) Beregn den samlede resistans ($R1+R2$)
- 2) Beregn strømmen I_{R1} (vi husker ohms lov hvor $I_{R1} = \frac{U_{R1}}{R_1}$)
- 3) Beregn spændingsfaldet hen over $R2$ (det er en serieforbindelse, så $I_{R1} = I_{R2}$).

Effektafsættelsen i $R2$ skal begrænses til 1mW (for at undgå egenopvarmning - det er jo en temperatursensor vi er ved at lave).

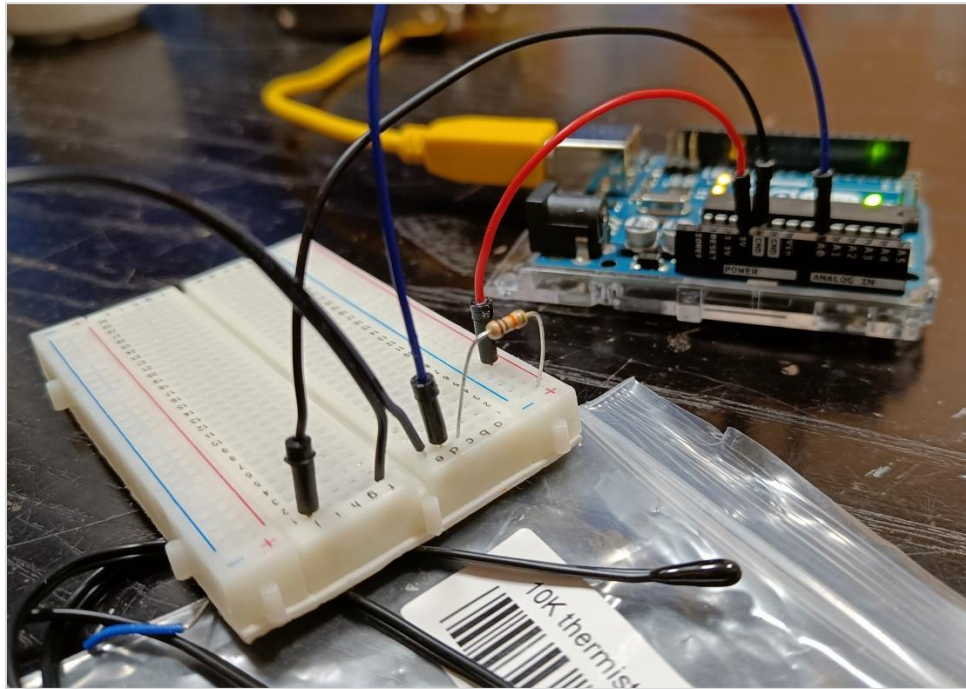
- 4) Undersøg om effektafsættelsen i $R2$ er mindre end 1mW

Opgave 8b) - Opbygning af kredsløb

Sørg for det er et rent breadboard du arbejder på (fjern komponenterne fra sidste øvelse)

Du skal bruge:

- 1 stk breadboard.
- 1 stk Arduino print (Arduinoen er +5V spændingsforsyning)
- 1 stk resistor (10k Ω har farvekoden brun-sort-orange)
- 1 stk NTC 10k Ω (helst i krympeflex, så vi kan stikke den i vand)
- 3 stk ledninger (en rød til +5V, en sort til 0V og yderligere en)



NTC-modstanden er loddet på et kabel, der sættes i breadboard

- 1) Begynd med at tilslutte +5V fra Arduino-printet til den røde "skinne".
- 2) Resistoren monteres med +5V på den ene side og NTC-kablet på den anden side.
- 3) Der monteres en ledning fra fællespunktet mellem NTC-kablet og 10k Ω resistoren. Ledningen tilsluttes Arduino-printets analoge indgang (A0).
- 4) Gennem den sorte ledning tilsluttes den anden side af NTC-kablet med 0V fra Arduino-printet (GND).

Opgave 8c) - Kontrol af kredsløbet

Her er et program der læser A/D-converteren og udlæser værdien på skærmen (seriel monitor)

1) Kopier denne testsoftware ind i Arduino-IDE.

```
// NTC temperaturmåling
// Dette program læser en A/D værdi (fra A0) og viser
// værdien i monitor-feltet forinden 8/12-2024 Ken Mathiasen

// Her initialiseres printet
void setup() {
  Serial.begin(9600); // Her opsættes en seriel kanal der avendes til at vise
                     // parametre fra Arduino-printet i monitor-vinduet
}

// Her er main-loop
void loop() {
  int val = analogRead(A0); // En variabel til at gemme den
                           // konverterede værdi i.
                           // Tallet kan være fra 0-1024, hvor
                           // 0 er minimum-værdien og 1024 er max.

  Serial.print("A/D value is: "); // her er en hjælpetekst
  Serial.println(val);           // værdien skrives i monitorvinduet
  delay(200);
}
```

2) Kontroller, at der kommer en værdi ud på skærmen. Læg mærke til tallet.

3) Varm nu NTC op (hold den mellem to fingre) og læg mærke til tallene.

4) Hvis tallene falder når temperaturen stiger, er alt OK

Fastlæggelse af overførselsfunktion (så der kan udlæses °C i stedet for A/D-count)

For at microcontrolleren skal kunne lave en præcis temperaturmåling skal kredsløbets karakteristisk findes. Problemer er at vi ikke umiddelbart kender sammenhængen mellem temperatur, NTC spændingsfald, og dermed A/D-count.

Så vi skal have fastlagt kredsløbets overførselsfunktion (vha. regression)

Det gøres ved at registrere kredsløbets respons til forskellige temperaturer.

Vi nedsænke NTC i vand med forskellige temperaturer og noterer temperatur og A/D-count i en tabel.

Så kan vi efterfølgende lave en eksponentiel regression og lægge formelen ind i Arduino.

Vi går i fysiklab

- 1) Fyld fire kopper med forskellig vandtemperatur (cirka: 10°C, 35°C, 60°C, 80°C).
- 2) Nedsæk NTC i den ene kop. Mål vandets temperatur med et præcist termometer. Temperaturen skal måles nøjagtig (så sørg for løbende omrøring). Temperatur og A/D-count aflæses, når tallene i serial monitor er stabile. Skriv resultaterne i skemaet.

| | kop 1 | kop 2 | kop 3 | kop 4 |
|--|-------|-------|-------|-------|
| Målt temperatur (°C) | | | | |
| Aflæst A/D-count (fra serial monitor) | | | | |

Skift til den næste temperatur (kop) og gentag øvelsen.

- 3) Gentag forsøget med de andre kopper/temperaturer.
- 4) Når tabellen er udfyldt, skal der laves en eksponentiel regression.
Husk at tjekke om det er en god regression og at residualerne ligger tilfældig fordelt
- 5) Kopier dette program ind i Arduino IDE

```
// NTC temperaturmåling
// Dette program læser en A/D værdi (fra A0) og omregner værdien til grader Celsius
// Resultatet vises i monitor-feltet.                                     8/12-2024 Ken Mathiasen

// Her initialiseres printet
void setup() {
  Serial.begin(9600); // Her opsættes en seriel kanal der avendes til at vise
                     // parametre fra Arduino-printet i monitor-vinduet
}

// Her er main-loop
void loop() {
  float var = analogRead(A0); // En variabel til at gemme A/D-værdien i
  float temperatur = 114.3135*pow(0.9968,var); // her udregnes temperaturen

  Serial.print("Temperature is calculated to: "); // hjælpetekst til monitor
  Serial.println(temperatur); // værdien udlæses temperaturen
  delay(200);
}
```

- 6) Formlen i linje 14 er: $y = 114,3135 \cdot 0,9968^x$
Det skrives som: `float temperatur = 114.3135*pow(0.9968,var);`
Ændr tallene i linje 14 til tallene fra din regressionsformel
- 7) Afprøv programmet. Prøv at varme NTC med fingrene og se om temperaturen stiger.
- 8) Kontroller at Arduino-termometeret virker. Dyp NTC ned i en kop med 40°C vand.

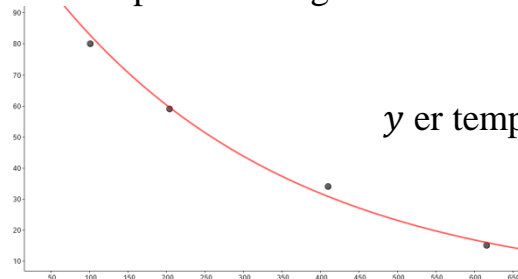
Eksempel

Følgende tal er målt i lab:

| | kop 1 | kop 2 | kop 3 | kop 4 |
|------------------------|-------|-------|-------|-------|
| Målt temperatur (°C) | 15°C | 34°C | 59°C | 80°C |
| Aflæst A/D-count på PC | 616 | 410 | 204 | 101 |

Der gennemføres en eksponentiel regression med GeoGebra med følgende resultat

| | A | B |
|---|-----|----|
| 1 | 616 | 15 |
| 2 | 410 | 34 |
| 3 | 204 | 59 |
| 4 | 101 | 80 |



$$y = 114,3135 \cdot 0,9968^x$$

y er temperaturen i °C og x er antal A/D-count
($R^2 = 0,9922$ hvilket er fint)

Programmering af overførselsfunktion

Formlen skal programmeres i Arduino IDE.

Det er tale om reelle tal (ikke heltal, integer), så der oprettes en variabel af typen float

De $0,9968^x$ indtastes som en potens-funktion. Det hedder power på engelsk (pow)

```
NTC_to_Celcius | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
NTC_to_Celcius.ino
1 // NTC temperaturmåling
2 // Dette program læser en A/D værdi (fra A0) og omregner værdien til grader Celcius
3 // Resultatet vises i monitor-feltet. 8/12-2024 Ken Mathiasen
4
5 // Her initialiseres printet
6 void setup() {
7   Serial.begin(9600); // Her opsættes en serial kanal der avendes til at vise
8   // parametre fra Arduino-printet i monitor-vinduet
9 }
10
11 // Her er main-loop
12 void loop() {
13   float var = analogRead(A0); // En variabel til at gemme A/D-værdien i
14   float temperatur = 114.3135*pow(0.9968,var); // her udregnes temperaturen
15
16   Serial.print("Temperature is calculated to: "); // her er en hjælpetekst til monitor
17   Serial.println(temperatur); // værdien udlæses temperaturen
18   delay(200);
19 }
20
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM3') New Line 9600 baud
Temperature is calculated to: 26.17
Temperature is calculated to: 26.17
Temperature is calculated to: 26.17
Ln 21, Col 1 Arduino Uno on COM3
```