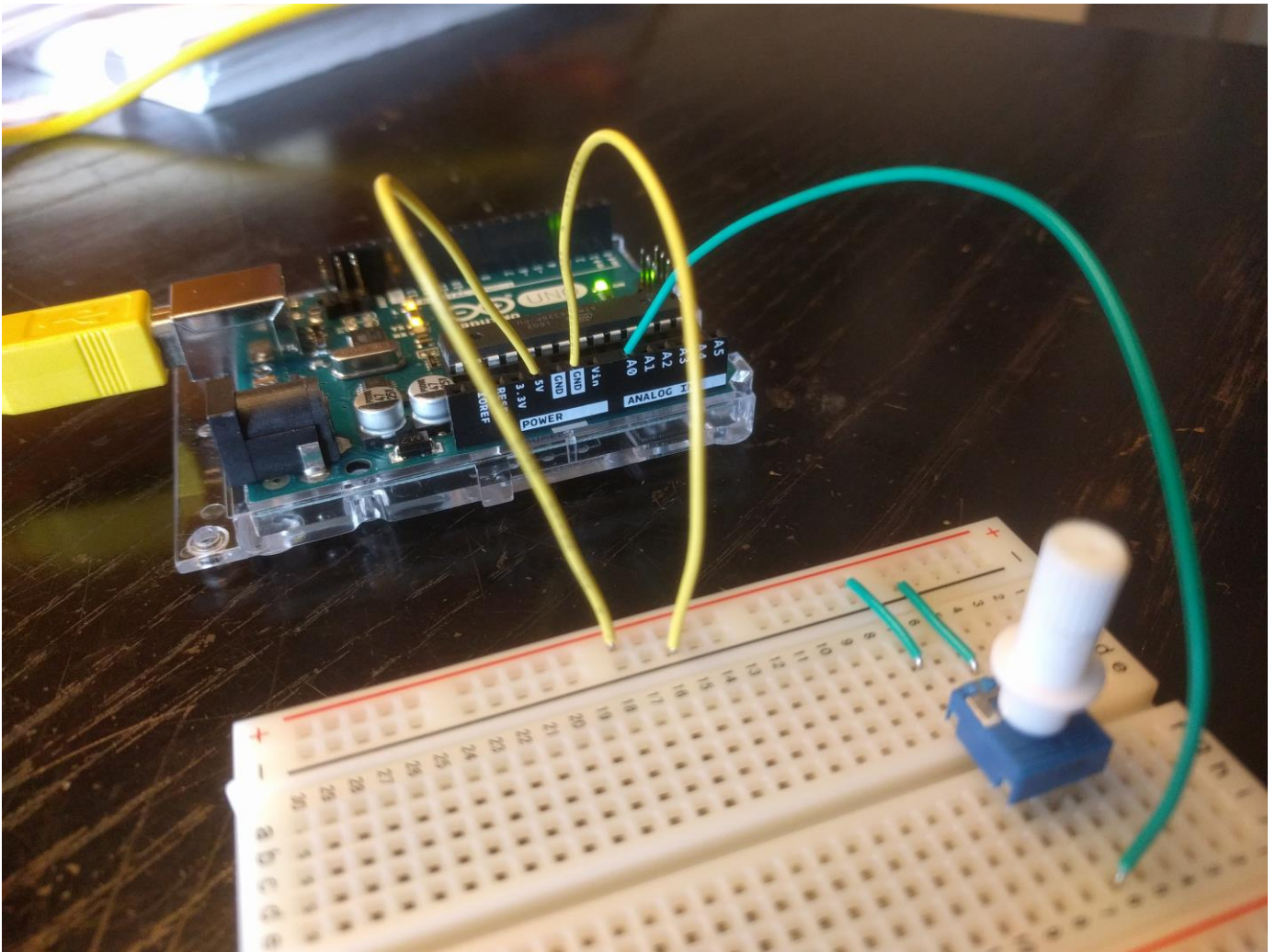


Arduino Introduktion

Hands-On programmering i fysik og informatik



Ken Mathiasen

Må gerne kopieres, men husk referencen

Arduino Introduktion

Hands-On programmering i fysik og informatik (stx, htx, eux)

Hent hæftet i en pdf-udgave (gratis): www.mathiasen.eu → books.

© 2024 Ken Mathiasen

Omslag, layout, tegninger og foto: Ken Mathiasen

Grafik: Helle Lauvring

Hæftet er lavet med hjælp fra velvillige sponsorer og er dermed gratis i pdf-udgaven.

Kopiering af denne bog må gerne finde sted (også selvom du ikke har købt hæftet) - det skal bare lige registreres hos Copydan Tekst & Node. Så send en mail til kopi@copydan.dk, hvor du vedhæfter en pdf-kopi af hæftet, med angivelse af de 10 sider du bruger mest.

Det er gøres nemt (fx i september, når der er faldet lidt ro på) ved at sende en mail:

Tekst til din e-mail: Korsbæk Handelsskole (skole nr. kendes desværre ikke).
 Kopieret fra "Arduino Introduktion", Ken Mathiasen ISBN 978-87-973415-4-4
 Siderne 10-20 til 18 elever (se vedlagte).
 mvh Overlærer Andersen.

Se mere på Copydan Tekst og node på <https://tekstognode.dk>

Pdf version 0.3 den 3/3-2024

1. udgave, 1. oplag. Printet hos TopTryk Grafisk A/S, Gråsten, Danmark, januar 2022

ISBN 978-87-973415-4-4

Vi har gjort vores bedste for at undgå fejl og mangler. Støder du på nogle, så send en mail (det er nemt at rette)

Ken Mathiasen
Povlstoft 16
6400 Sønderborg
mail: mathiasen@runbox.com

Kommentar til underviseren.

Lektionerne er opbygget omkring The Arduino Starter Kit (kr. 700,- hos El Supply, <https://el-supply.dk/shop/741-arduino-mainboards/68519-arduino-starter-saet/>).

Det er ikke vigtigt hvilken version af Arduino-printet der anvendes i undervisningen. Softwaren er den samme. Langt de fleste input/output-ben hedder det samme og er forbundet til de samme stik på printet.

Det fungerer fint med to elever omkring et board.

I sidste del af bogen gives anvisninger til, hvorledes den serielle kommunikation kan opbygges, hvis Arduino'en skal bruges til at opsamle data til Excel analyse.

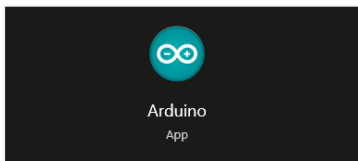
INDHOLDSFORTEGNELSE

1	Opstart og installation.....	4
1.1	Atmel mikrocontroller og Programmering	5
1.2	Introduktion til hardwarekomponenter og kredsløb	6
2	Programmering af digitale ind- og udgange	10
2.1	Anvendelse af input	12
3	Anvendelse af analoge ind og udgange	15
3.1	Analog indgang	15
3.2	Analog udgang.....	19
4	Linearisering af NTC.....	21
4.1	Måling af to punkter til kalibrering af sensor	21
5	Projekt - vejrstation	25
5.1	Temperatursensor	25
5.2	Vindsensor	25
6	Overførsel af data fra Arduino til fil på PC	27



1 OPSTART OG INSTALLATION

Programmøren arbejder på sin PC. Den software der skrives, afvikles på et Arduino Uno board (hardware - i daglig tale kaldet *printet*). Arduino-printet er forbundet til PC'en med et USB-kabel.



Arduino App er et programmeringsværktøj

Der anvendes en gratis app til programmeringen

ØVELSER - INSTALLATION

Opgave 1.1 - hent appen

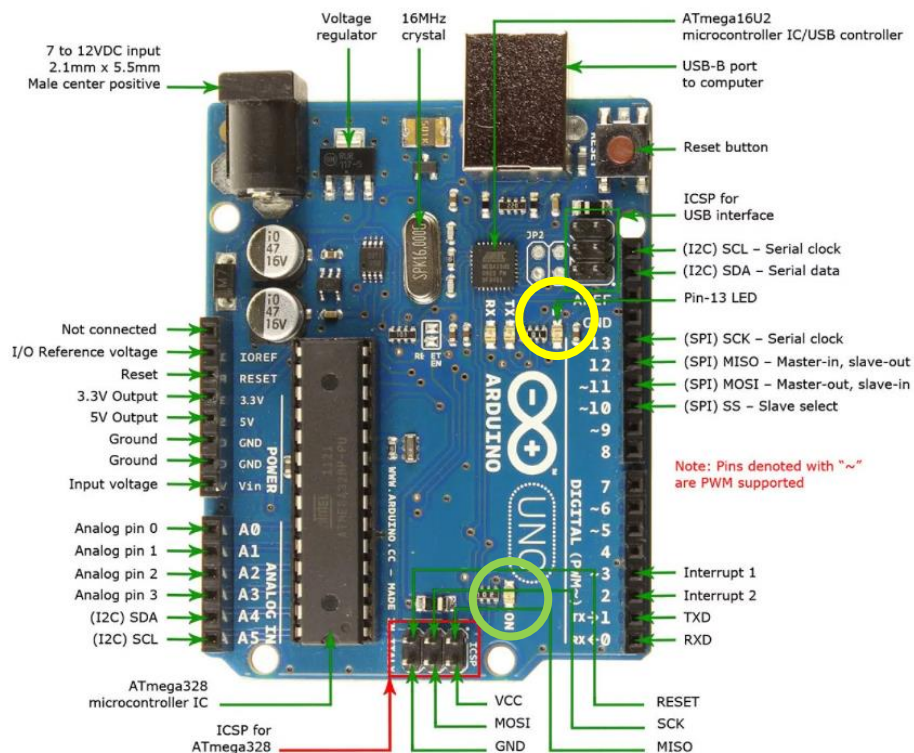
Fra elevens PC hentes og installeres Arduino IDE: <https://www.arduino.cc/en/Main/Software>

Opgave 1.2 - tilslut Arduino-printet

Åbn Arduino appen.

Tilslut Arduino-printet med et USB kabel.

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



Billede af Arduino Uno printet (fra <https://www.robomart.com/arduino-uno-online-india>).
Den LED vi som programmører kan bruge, er markeret med gul.


1.1 ATMEL MIKROCONTROLLER OG PROGRAMMERING

Arduino-printet er udstyret med en Atmel singlechip mikrocontroller, ATMEGA328. Mikrocontrolleren har forskellige indgange og udgange. På Arduino-printet er en af mikroprocessorens ben (en udgang) tilsluttet en LED. Udgangen kaldes "Pin 13". Pin 13 kan bruges som en digital indgang eller som en digital udgang. Når Pin 13 anvendes som en digital udgang, kan mikrocontrolleren sætte 5 volt på Pin 13 eller 0 volt på den. Altså HIGH eller LOW, 1 eller 0. Når der er 5 V på Pin 13 vil LED "L" lyse.

I programmet gøres dette ved, at der i koden skrives: `digitalWrite(13,HIGH);`

ØVELSE - PROGRAMMERING AF DET FØRSTE PROGRAM - SLUK/TÆND LED (L)

Opgave 1.3 - Der arbejdes i appen på PC'en

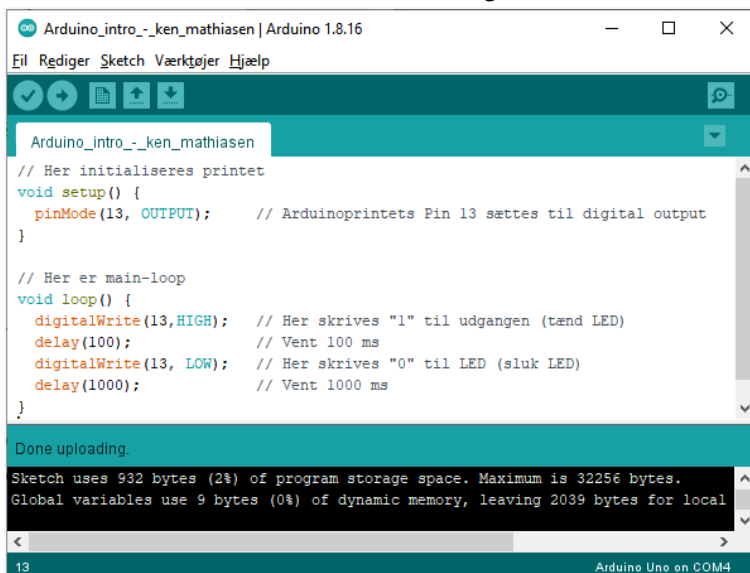
a) Opret et nyt dokument i Arduino editoren - det kaldes sketch. Tryk på "ny" 

b) Indtast følgende:

```
// Her initialiseres printet
void setup() {
  pinMode(13, OUTPUT);    // Arduinoprintets Pin 13 sættes til digital output
}

// Her er main-loop
void loop() {
  digitalWrite(13,HIGH);  // Her skrives "1" til udgangen (tænd LED)
  delay(100);             // Vent 100 ms
  digitalWrite(13, LOW);  // Her skrives "0" til LED (sluk LED)
  delay(1000);            // Vent 1000 ms
}
```

c) Når du har indtastet koden, så husk at gemme din Sketch (fil - gem som...)



Her er et screenshot af koden til det første program

d) Når koden er indtastet skal den kompileres (tjekket for fejl og oversættes til maskinkode). Det gøres ved at trykke på "Verificer"

Hvis koden kan kompileres, skal den lægges ned i mikrocontrolleren.

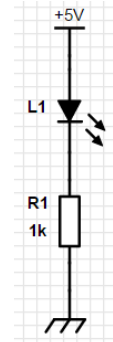
Det gøres ved at trykke på "Upload"

e) Kontroller at "L" blinker

f) "L" blinker med en "heart-beat" rytme (kort blink lang pause). Modificer koden, så det er en konstant langsom blinken.

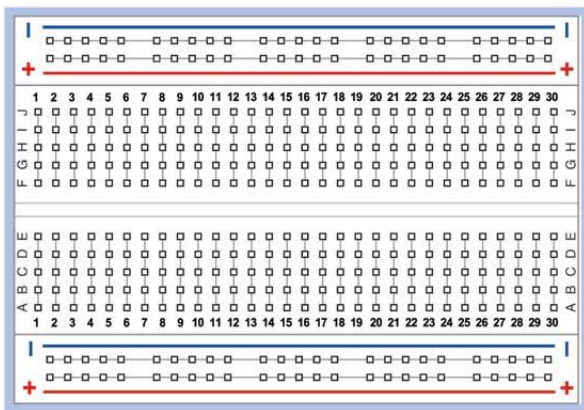
1.2 INTRODUKTION TIL HARDWAREKOMPONENTER OG KREDSLØB

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



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 LED forbindes til et ben på microcontrolleren. Det gøres med en ledning. I prototype-fasen (og til undervisning) anvendes breadboard når elektriske kredsløb skal opbygges. Et breadboard er en plastikblok med mange huller i.



Billedet viser hvorledes de enkelte huller er forbundet med interne ledninger

Med et breadboard kan komponenterne hurtig forbindes til hinanden og det er efterfølgende nemt at ændre forbindelserne.

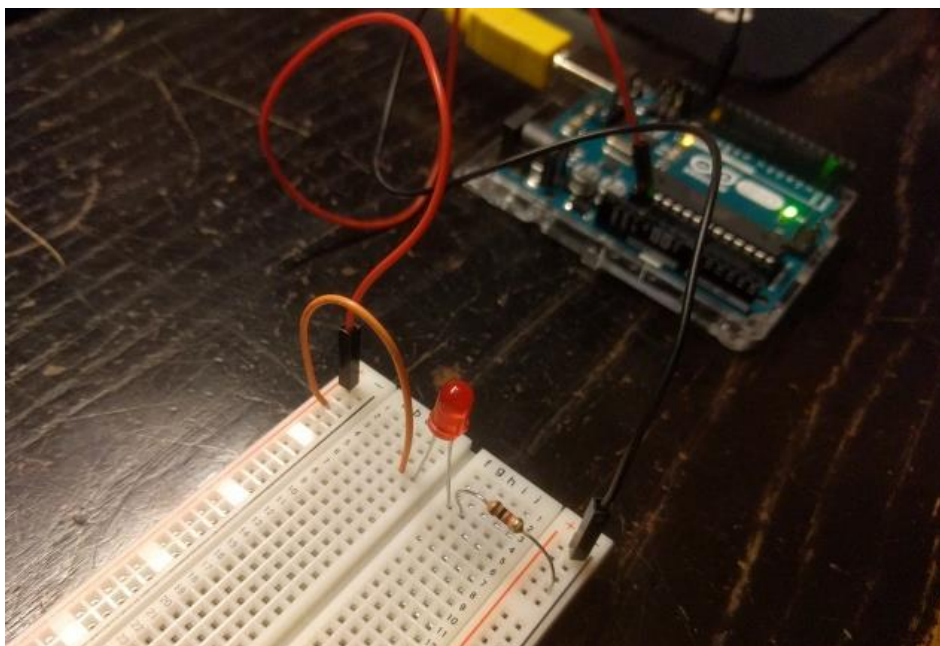
En ulempe ved breadboard er at en ledning kan hoppe ud af hullet når man arbejder med eller at der kan opstå løse forbindelser.

ØVELSE - VI FÅR EN LED TIL AT LYSE

Opgave 1.4 - LED tilsluttes

Nu skal vi have en LED til at lyse. El-diagrammet er som vist øverst på denne side, til højre.

- Placer LED'ens lange ben i e4 og det korte ben i f4.
- En modstand ($1k\Omega$ brun-sort-rød-guld) skal placeres mellem LED og 0V.
- De ene ben i g4 og det andet ben i -4.
- Set en ledning i d4 og op til +5V, +4



Det lange ben på den røde LED er tilsluttet +5V

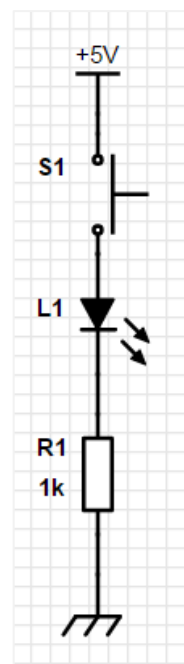
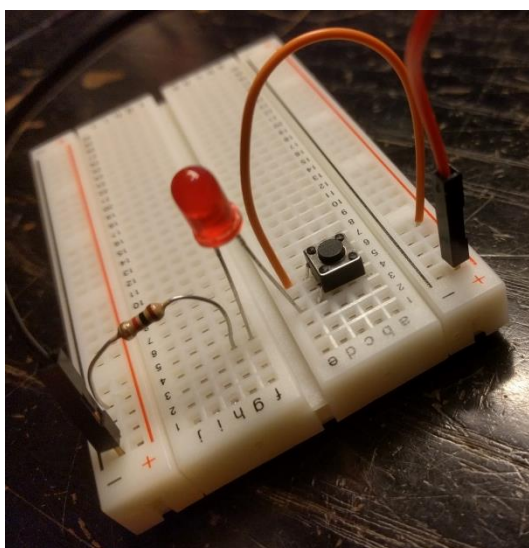
Nu mangler vi bare at tilslutte en strømforsyning. Det bruger vi Arduino Uno til.

e) Træk en rød ledning fra +1 til "5V" på den korte sorte connector på printet.

f) Og træk en sort ledning fra -1 til "GND" på den lange Arduino connector.

Opgave 1.5 - LED tilsluttes

Nu indsættes en trykknop. Når man trykker på knappen, skal LED lyse.



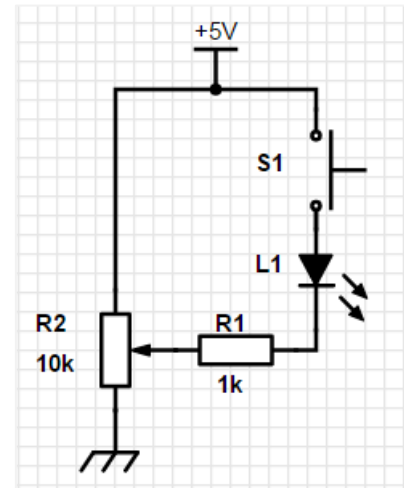
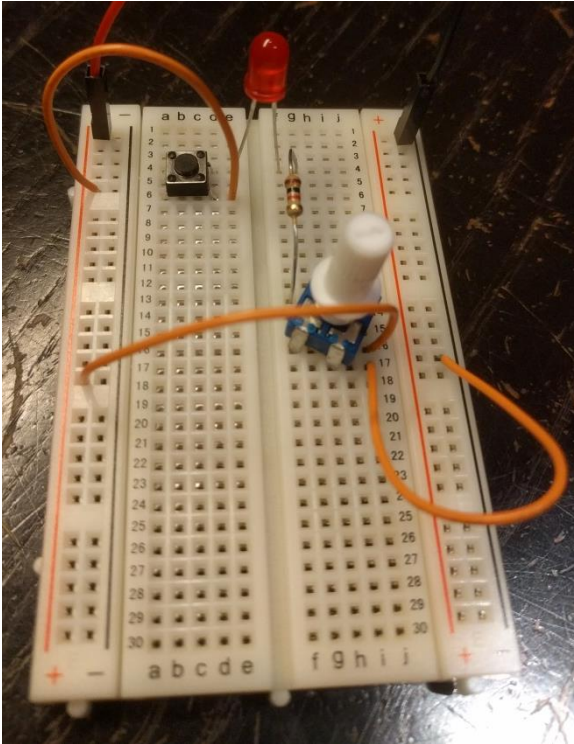
Opgave 1.6 - en knap monteres i serie med LED

a) Indsæt en knap (push button) i a4, d4 og a6 og d6.

b) Den ledning der før gik til det lange ben på LED flyttes, så den nu går fra e6 til +6.

Opgave 1.7 - lysdæmper

Nu skal kredsløbet ændres så man kan skrue op og ned for lyset. Det gøres med en variabel modstand, kaldet et potentiometer.



- Sæt potentiometeret i breadboard - det skal monteres lidt på skrå. Det midterste ben i i13, et ben i f16 og et i h17.
- Det skal være en ledning fra j17 til -17.
- En ledning fra j16 til +18.
- Modstanden flyttes, så der er et ben i g4 og et i f13.

Opgave 1.8 - Fri leg

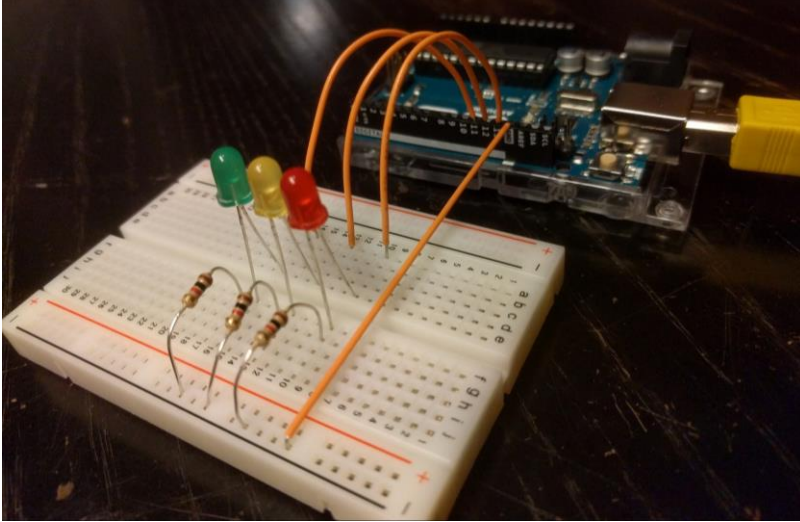
- Prøv at montere endnu en LED og se om du også kan komme til at skrue op for lyset.
- Prøv at lave et trafiklys med en rød, gul og grøn LED. Kan man lave det således at når der er skruet op for grøn, så er der skruet ned for gul og rød. Og omvendt når der er skruet op for gul og rød, skal grøn være skruet ned.

2 PROGRAMMERING AF DIGITALE IND- OG UDGANGE

Alternativt til at trykke på en knap, kan vi programmere Arduino til at gøre det for os

Øvelse - Programmering af tre digitale udgange

Opgave 2.1 - Opbygning af hardware til lyskryds



Tre LED skal tilsluttes Arduino-printet - hver med en 1 kΩ resistor i serie til stel (0V)

- Det lange ben på en rød LED tilsluttes (igennem en ledning) til Arduino pin 13.
- Det lange ben på en gul LED tilsluttes (igennem en ledning) til Arduino pin 12.
- Det lange ben på en grøn LED tilsluttes (igennem en ledning) til Arduino pin 11.
- LED'ens korte ben sættes sammen med modstanden (brun, sort, rød, guld) til 0V.
- 0V forbindes med en ledning, til 0V på Arduino (den pin der hedder GND).

Opgave 2.2 - Initialisering af microcontrollerens digitale udgange.

Før softwaren kan tænde og slukke de LED'er der er tilsluttet, skal vi fortælle microcontrolleren hvilke pinde der anvendes og hvad de anvendes til. Det kaldes initialisering.

- Opret et nyt dokument på PC'en (i Arduino appen) der hedder blinklys
- Indtast eller kopier følgende tekst ind i en ny Sketch (slet alt tekst i sketchen, inden du indsætter den nye)

```
// Blinklys
// Dette program kontrollerer tre lamper (rød, gul og grøn).
// 21/11-2021 Ken Mathiasen

// Her initialiseres printet
void setup() {
  pinMode(11, OUTPUT); //Arduinoprintets Pin 11 sættes til digital output
  pinMode(12, OUTPUT); //Arduinoprintets Pin 12 sættes til digital output
  pinMode(13, OUTPUT); //Arduinoprintets Pin 13 sættes til digital output

  digitalWrite(11,LOW); // Der skrives "0" til alle udgangene - det hele slukkes
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);
}
```

Opgave 2.3 - Programmering af lys-sekvensen

De forskellige LED skal tændes og slukkes - en ad gangen. På følgende vis:

Først tændes den røde.

Så ventes der et sekund, hvorefter den røde slukkes og den gule tændes.

Igen ventes et sekund og den gule LED slukkes, hvorefter den grønne tændes.

....og så fremdeles

a) Indtast blinksekvensen så dit program til sidst ser således ud:

```

Lyskryds_-_ken_mathiasen.ino | Arduino 1.8.16
Fil Rediger Sketch Værktøjer Hjælp
Lyskryds_-_ken_mathiasen.ino
// Lyskryds
// Dette program kontrollerer tre lamper (rød, gul og grøn).
// 21/11-2021 Ken Mathiasen

// Her initialiseres printet
void setup() {
  pinMode(11, OUTPUT); //Arduinoprintets Pin 11 sættes til digital output (grøn LED)
  pinMode(12, OUTPUT); //Arduinoprintets Pin 12 sættes til digital output (gul LED)
  pinMode(13, OUTPUT); //Arduinoprintets Pin 13 sættes til digital output (rød LED)


  digitalWrite(11,LOW); // Der skrives "0" til alle udgangene - det hele slukkes
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);
}

// Her er main-loop
void loop() {
  digitalWrite(13,HIGH); // Her skrives "1" til udgang 13 (tænd rød LED)
  delay(1000); // Vent 1000 ms
  digitalWrite(13, LOW); // Sluk rød LED (skriv "0" til pim 13)
  digitalWrite(12,HIGH); // Tænd gul LED
  delay(1000); // Vent 1000 ms
  digitalWrite(12, LOW); // Sluk gul LED
  digitalWrite(11,HIGH); // Tænd grøn LED
  delay(1000); // Vent 1000 ms
  digitalWrite(11, LOW); // Sluk grøn
}

Done uploading.
Sketch uses 1022 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum
1 Arduino Uno on COM4

```

Når koden opbygges bruger man ofte Ctrl-C og Ctrl-V for at kopiere tekst, hvorefter tallene rettes til

b) Gem dit program og afprøv koden ved at trykke på  (Arduino skal være tilsluttet).

c) Disko-lys. Modifier koden således lyset løber hurtigere.

d) Frem og tilbage. Modifier koden således lyset løber tilbage (ikke i ring som før)

e) Fri leg. Lav selv alternative lys-sekvenser.

En ny variabel

Der skal oprettes en variabel der afbilder den digitale indgang.

Ved at anvende en variabel kan vi anvende booleske test i vores kode.

b) Som det første i "void loop()" indsættes følgende:

```
int var;           // en variabel til at gemme om knappen er trykket
                  // var = 1 betyder at knappen ikke er trykket
                  // var = 0 betyder at knappen er trykket

var = digitalRead(10);
```

Blink-sekvensen modificeres

Når der trykkes på knappen skal LED lyse.

Hvis der ikke er trykket, er LED slukket. Det gøres med en While-sætning.

c) Modificer din kode så den indeholder en While-sætning. Fx på følgende vis:

```
Blinklys_med_starttryk.ino | Arduino 1.8.16
Fil Rediger Sketch Værktøjer Hjælp

Blinklys_med_starttryk.ino
// Blinklys med starttryk
// Dette program kontrollerer tre lamper (rød, gul og grøn).
// 21/11-2021| Ken Mathiasen

// Her initialiseres printet
void setup() {
  pinMode(10, INPUT); //Arduinoprintets Pin 10 sættes til digital indgang
  pinMode(11, OUTPUT); //Arduinoprintets Pin 11 sættes til digital output
  pinMode(12, OUTPUT); //Arduinoprintets Pin 12 sættes til digital output
  pinMode(13, OUTPUT); //Arduinoprintets Pin 13 sættes til digital output

  digitalWrite(11,LOW); // Der skrives "0" til alle udgangene - det hele slukkes
  digitalWrite(12,LOW);
  digitalWrite(13,LOW);
}

// Her er main-loop
void loop() {
  int var;           // en variabel til at gemme om knappen er trykket eller ej
                    // var = 1 betyder at knappen ikke er trykket
                    // var = 0 betyder at knappen er trykket

  var = digitalRead(10);
  while(var < 1)    // her tjekkes om knappen er trykket (om var = 0)
                  // Så længe knappen er trykket gennemføres koden
  {
    digitalWrite(13,HIGH); // Her skrives "1" til udgang 13 (tænd rød LED)
    delay(200);           // Vent 1000 ms
    digitalWrite(13, LOW); // Sluk rød LED (skriv "0" til pin 13)
    digitalWrite(12,HIGH); // Tænd gul LED
    delay(200);           // Vent 1000 ms
    digitalWrite(12, LOW); // Sluk gul LED
    digitalWrite(11,HIGH); // Tænd grøn LED
    delay(200);           // Vent 1000 ms
    digitalWrite(11, LOW); // Sluk grøn LED
    digitalWrite(11,HIGH); // Her skrives "1" til udgang 13 (tænd rød LED)
    delay(200);           // Vent 1000 ms
    digitalWrite(11, LOW); // Sluk rød LED (skriv "0" til pin 13)
    var = digitalRead(10); // vi slutter af med at se om knappen er trykket
  }
}

Done uploading.
Sketch uses 1206 bytes (3%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum
3 Arduino Uno on COM4
```

Blinksekvensen er gjort hurtigere ved at ændre delay til 200 ms

Læg mærke til at der til sidst inde i While-løkken testes om knappen er trykket.

Husk løbende at gemme din kode

d) Modifier koden således alle LED altid blinker, men når der bliver trykket på knappen, må den grønne LED ikke lyse.

e) Modifier koden således at alle LED lyser konstant når knappen trykkes. Når knappen ikke er trykket, så skal løbelyset fungere som normalt.

f) Fri leg

Lav nye kombinationer af kontakt og blink

PS - gå ind på <https://www.arduino.cc/reference/en/> for at se hvorledes de forskellige Arduino-kommandoer fungerer

3 ANVENDELSE AF ANALOGE IND OG UDGANGE

Inde i mikroprocessoren foregår tingene digitalt (HIGH eller LOW). I den virkelige verden er tingene analoge. For at forbinde den indre mikrokontrollers digitale verden med den analoge virkelige verden, anvendes convertere (en omsætter). A/D-convertere og D/A-convertere

3.1 ANALOG INDGANG

Microcontrolleren kan læse en analog værdi. Et analogt spændingssignal (fx 2,36 V dc) fra den virkelige verden, omsættes til et digitalt tal. Det kaldes konvertering. Der tales om en "analog til digital converter" forkortet til A/D. En analog til digital konvertering foregår ved, at det analoge signal måles relativt til 5 V. I procent bliver signalet på 2,36 V til

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

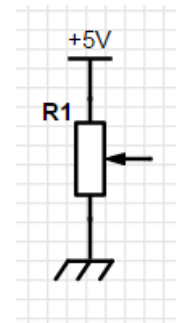
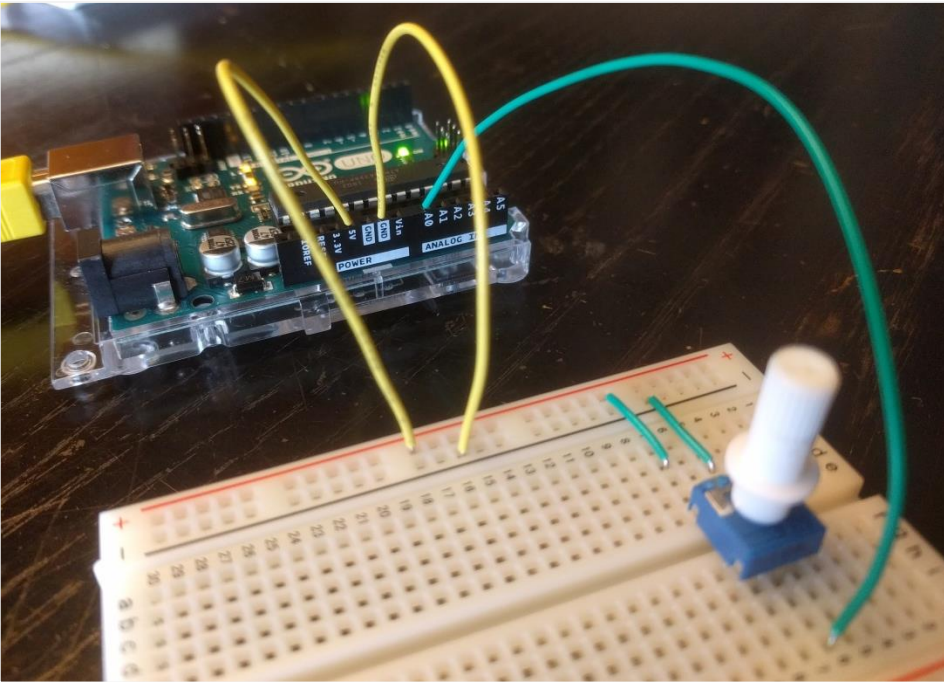
De 2,36 V svarer med andre ord til 47,2 % af 5,00 V. Microcontrolleren anvender et 10-bit binært tal. $2^{10} = 1024$, så A/D-converteren leverer et digitalt tal mellem 0 og 1024, hvor 0 svarer til 0,00 V og 1024 svarer til 100 %.

I eksemplet svarede de 2,36 V til 47,2 %. Det digitale tal der leveres fra A/D-converteren er således

$$47,2 \% \text{ af } 1024 = 483 \text{ count}$$

ØVELSE - ANVENDELSE AF A/D-CONVERTER

Opgave 3.1 - Opbygning af ny hardware



Et potentiometer monteres mellem 0 og +5V og potentiometerets udgang forbindes til A/D (A0)

a) Opbyg følgende

Et potentiometer sættes i e4, e7 (og den anden siden i g6)

Der skal bruges fem ledninger:


En ledning fra 0V (-5) til potentiometeret (c5).
 En ledning fra +5V (+8) til potentiometeret (b7).
 En ledning Arduino pin A0 til potentiometeret (j6).

Strømforsyning (0V), fra Arduino GND til -16

Strømforsyning (+5V), fra Arduino +18

Opgave 3.2 - En ny kode, så en A/D konvertering kan laves

For at kunne læse det tal A/D-converteren generer anvendes et monitor-vindue. Så kan vi på PC'en se hvad der foregår nede i Arduino'en

a) Åbn et monitor-vindue. Det gøres ved at klikke på  i øverste højre hjørne.

Arduino kan skrive i monitor-vinduet igennem en serial kommunikationsport. Kommunikationsporten opsættes i initialiseringsrutinen.

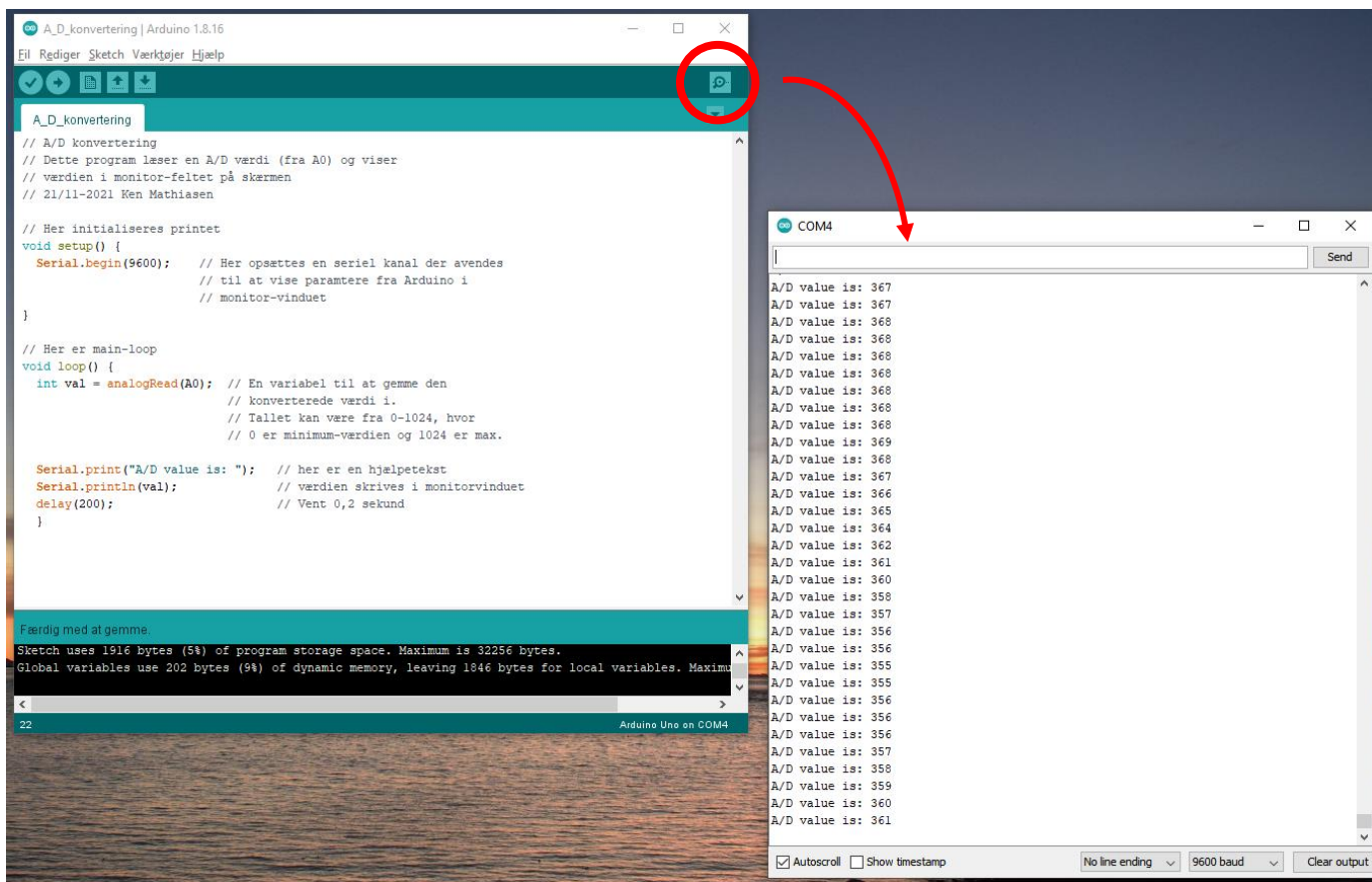
b) Kopier følgende over i Arduino editoren:

```
// A/D konvertering
// Dette program læser en A/D værdi (fra A0) og viser
// værdien i monitor-feltet på skærmen
// 21/11-2021 Ken Mathiasen

// Her initialiseres printet
void setup() {
  Serial.begin(9600); // Her opsættes en serial kanal der avendes
                    // til at vise parametre fra Arduino 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.

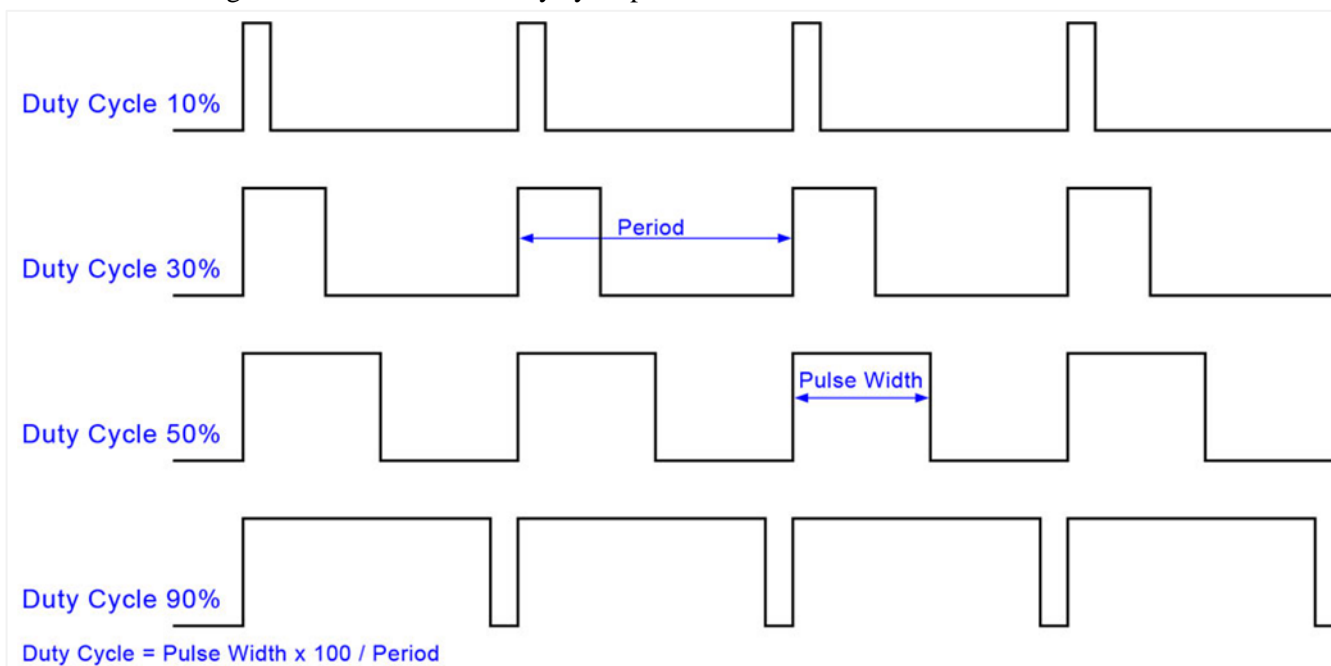
  Serial.print("A/D value is: "); // her er en hjælpetekst
  Serial.println(val);           // værdien skrives i monitorvinduet
  delay(200);                   // Vent 0,2 sekund
}
```



Her er kode til læsning af en A/D værdi. Samtidig er monitorvinduet vist (her COM4).

3.2 ANALOG UD GANG

En microcontrolleren kan ikke direkte lave et analogt udgangssignal. Microcontrolleren kan kun skrive 0 og 1. Ved at skifte fra 0 til 1, 1000 gange i sekundet, genereres et puls-tog, svarende til en frekvens på 1 kHz. 50 % af tiden er signalet 1. Det kaldes en duty cycle på 50 %.

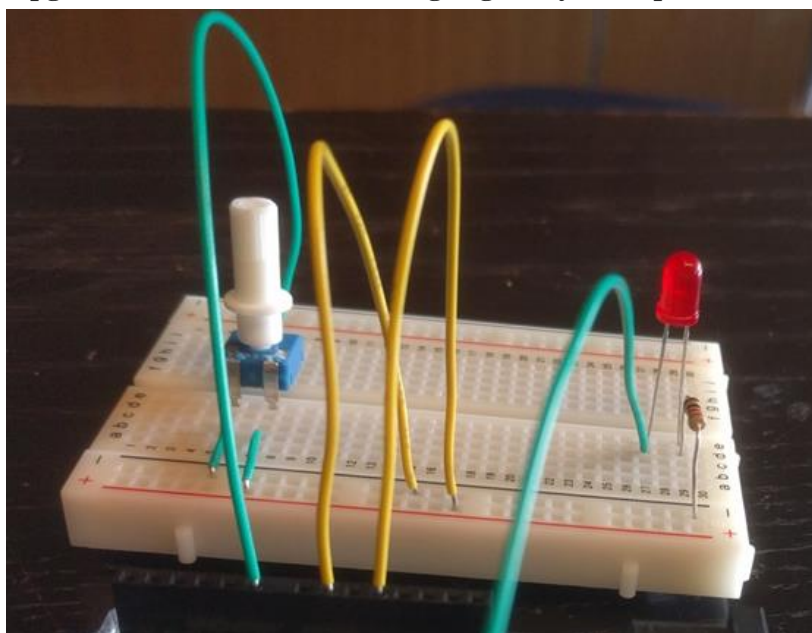


Figuren viser forskellige duty cycles - frekvensen er konstant.

Ved at ændre pulstiden kan der genereres en analog værdi.

ØVELSE - ANVENDELSE AF D/A KONVERTER

Opgave 3.3 - LED til at vise analogt signal (lys dæmper)



En LED skal tilsluttes en digital udgang, pin 13

a) Tilslut en LED. Husk at sætte en modstand i serie med LED (mellem det korte ben og 0V). Det lange LED ben skal tilsluttes Arduino pin 13.

Opgave 3.4 - Programmering så en digital port kan anvendes analogt

a) Begynd med at initialisere Pin13 som digital output.

b) I main-loop ændres programmet (efter koden med A/D konverteringen) således, at LED tændes og slukkes afhængig af A/D-værdien: Den tid LED skal tændes, sættes til A/D værdien.
Den tid LED skal slukkes, sættes til A/D-værdien minus 1024

```
digitalWrite(13,HIGH); // Her skrives "1" til udgang 13 (tænd LED)
delay(val);           // Vi venter det antal ms som er bestemt af potentiometeret
digitalWrite(13, LOW); // Her skrives "0" til udgang 13 (sluk LED)
delay(1024-val);      // Delayet gøres igen afhngig af A/D værdien
```

Forslag til kode der ændrer duty cycle afhængig af variabelen "val"

Ovenstående er lavet for at vise princippet i en PWM udgang.

```
// D/A konvertering
// Dette program læser en A/D værdi (fra A0) og bruger
// værdien til at bestemme dutycycle på udgangen Pin13
// Pin13 fungerer som PWM udgang
// 21/11-2021 Ken Mathiasen

// Her initialiseres printet
void setup() {
  Serial.begin(9600); // Her opsættes en seriel kanal der avendes
                    // til at vise parametre fra Arduino i
                    // monitor-vinduet
  pinMode(13, OUTPUT); // Arduinoprintets Pin 13 sættes til digital output
  digitalWrite(11,LOW); // Der skrives "0" til udgangen (slukket)
}

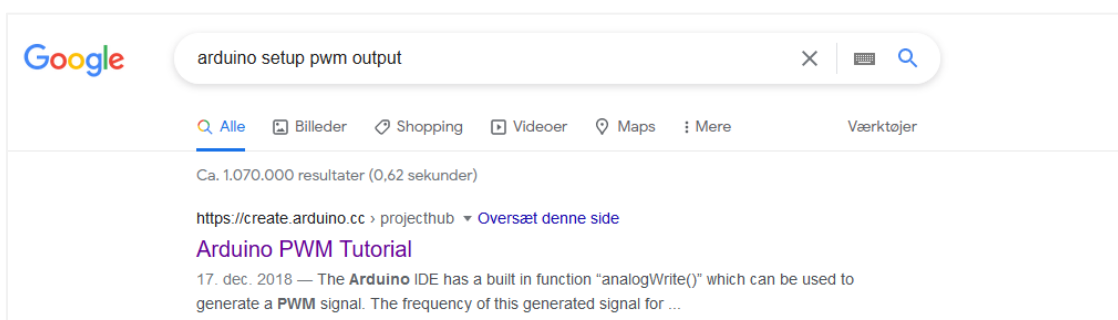
// Her er main-loop
void loop() {
  int val = analogRead(A0); // En variabel til at gemme den konverterede værdi i.
                          // Tallet kan være fra 0-1024, hvor
                          // 0 er minimum-værdien og 1024 er max.

  Serial.print("A/D value i procent er (ca): "); // her er en hjælpetekst til den
  Serial.println(val/10); // værdi der skrives i monitorvinduet

  digitalWrite(13,HIGH); // Her skrives "1" til udgang 13 (tænd LED)
  delay(val);           // Vi venter det antal ms som er bestemt af potentiometeret
  digitalWrite(13, LOW); // Her skrives "0" til udgang 13 (sluk LED)
  delay(1024-val);      // Delayet gøres igen afhngig af A/D værdien
}
```

Princippet i en simpel PWM-udgang

PWM-udgangen er langt fra 1 kHz . Den er for langsom. I praksis anvender man en "automatisk" PWM-udgangsfunktion Arduino er udstyret. Den kigger vi på en anden gang - eller søg selv på:



4 LINEARISERING AF NTC

For at microcontrolleren kan registrere en temperaturændring, anvendes man en 10 k Ω NTC modstand (negative temperature coefficient). Det er en komponent med to ben der opfører sig som en modstand. Modstandens værdi ændrer sig som funktion af temperaturen. Her er en simpel model af modstandens temperaturafhængighed:

$$R(T) = b \cdot a^T$$

R er komponentens modstand (resistans) med enheden ohm (Ω).
 T er komponentens temperatur i grader celsius ($^{\circ}\text{C}$).
 b er begyndelsesværdi
 a er fremskrivningsfaktor

Modstandens værdi ændrer sig omvendt proportionalt med temperaturen.

Ved 5 $^{\circ}\text{C}$ er modstanden omkring 23 k Ω

Ved stuetemperatur (25 $^{\circ}\text{C}$) er komponentens modstand 10 k Ω .

Ved 100 $^{\circ}\text{C}$ er den omkring 1 k 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 4,5 V.

Problemer er, at vi ikke umiddelbart kender sammenhængen mellem temperatur og spændingssignal. Så vi skal have undersøgt kredsløbets karakteristik.

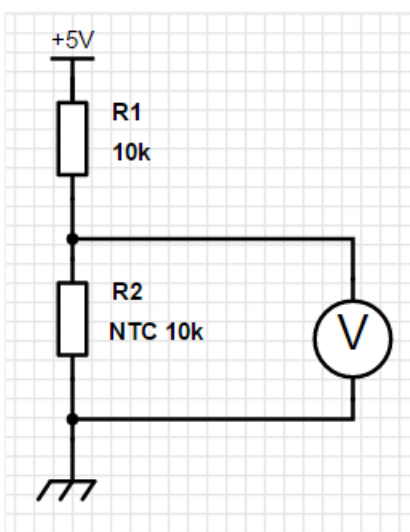
4.1 MÅLING AF TO PUNKTER TIL KALIBRERING AF SENSOR

Problemer er at vi ikke umiddelbart kender sammenhængen mellem temperatur og spændingssignal.

Så vi skal have undersøgt kredsløbets karakteristik. I denne undervisningssammenhæng gøres dette ved at måle to punkter og efterfølgende estimere funktionsværdierne

Opgave 4.1 - opbygning af kredsløbet

a) Brug breadboard og opbyg følgende - placer NTC yderst, så den kan stikkes ned i en kop vand.



Voltmeteret repræsenterer en analog indgang på microcontrolleren

Det er kun NTC der må komme ned i vandet - ikke breadboard eller noget af de andre komponenter.

ØVELSE - SIMPEL KARAKTERISERING AF NTC MODSTAND

Opgave 4.2 - karakterisering i fysiklab

For at microcontrolleren skal kunne lave en temperaturmåling skal kredsløbets karakteristik findes. Det gøres ved at måle kredsløbets respons på forskellige temperaturer og efterfølgende lave en eksponentiel regression. NTC komponenten nedsænkes i vand med forskellige temperaturer.

Det er vigtigt at kredsløbets spændingsforsyning er nøjagtig indstillet på 5,00 V

a) **Fyld fire** kopper med vand med forskellig temperatur (fx 10°C, 50°C, 75°C, 100°C).

b) **Nedsænk NTC** i den ene kop. Mål vandets temperatur (nøjagtigt - brug et nøjagtigt termometer, sørg for løbende omrøring) og aflæs samtidig spændingen på voltmeteret. Skriv resultaterne i skemaet.

	stuetemperatur	kop 1	kop 2	kop 3	kop 4
Målt temperatur (°C)					
Målt spænding (V)					

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

Opgave 4.3 - A/D konvertering og temperatur beregning

Formålet med hele øvelsen er at gøre microcontrolleren i stand til at udregne og vise en temperatur i grader celsius. Det gøres med Arduino og den serielle monitor.

Vi anvender en 10 bit A/D converter som måler en analog værdi mellem 0 og 5 V, svarende til 0 count til 1024 count (fordi $2^{10} = 1024$).

Anvendes tallene fra eksemplet, vil en målt spænding på

3,50 V svare til en temperatur på 5°C

2,50 V svarer til en temperatur på 25°C

0,40 V svarer til en temperatur på 100 °C

Dermed kan vi udregne det antal A/D-count microcontrolleren vil måle:

Ved 5°C er det $\frac{3,50 V}{5,00 V} \cdot 1024 = 716,8$. Når der er 5°C viser microcontrolleren 717 count

Ved 25°C er det $\frac{2,50 V}{5,00 V} \cdot 1024 = 512$. Ved 25°C viser microcontrolleren 512 count

Ved 100°C er det $\frac{0,40 V}{5,00 V} \cdot 1024 = 81,92$ Ved 100°C viser microcontrolleren 82 count

a) Gennemfør tilsvarende beregninger med tallene fra jeres forsøg og indsæt resultaterne i dette skema:

	stuetemperatur	kop 1	kop 2	kop 3	kop 4
Beregnet antal count					
Skriv den temperatur der skal vises					

Opgave 4.4 - Regression

Der gennemføres en eksponentiel regression med tallene i skemaet.

Her er et eksempel med opgavens tal:

	A	B	C	D
=				=ExpReg(
1	717.	5.	Titel	Ekspone...
2	512.	25.	RegEqn	a*b^x
3	82.	100.	a	164.993
4			b	0.995522
5			r ²	0.941276

Det viser sig at følgende sammenhæng er gældende mellem A/D-count og temperatur:

$$T(\text{count}) = 165 \cdot 0,9955^{\text{count}}$$

Dermed har vi skaleret spændingssignalet så 0 count er 0°C og 1024 count er 102,4°C.

- Gennemfør en tilsvarende eksponentiel regression med tallene fra jeres forsøg. Noter R²-værdien.
- Prøv også at lave en lineær regression og en potensregression.
Lav en graf med jeres målepunkter og de fundne funktioner fra regressionerne.
Vælg den regression som har den højeste R²-værdi og hvis graf passer bedst på de givne punkter.
- Denne formel kan implementeres i microcontrolleren. Her er et eksempel:

```

temperaturudregning | Arduino 1.8.16
Fil Rediger Sketch Værktøjer Hjælp

temperaturudregning $
// Her afprøves om lineariseringsformlen kan programmeres
// Der læses en A/D værdi (fra A0) og den tilsvarende temperatur udlæses
// 21/11-2021 Ken Mathiasen

#include <math.h> // Her inkluderes et matematik-bibliotek der gør anvendelsen af
                // forskellige matematiske formler og funktioner mulig

void setup() {
  Serial.begin(9600); // Her opsættes en seriel kanal der avendes
                    // til at vise parametre fra Arduino i monitor-vinduet
}
// Her er main-loop
void loop() {
  float AD0 = 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

  AD0 = constrain(AD0, 1, 1024);
  float temperatur=100;
  temperatur=-96*log10(AD0/830);

  Serial.print("A/D value: ");
  Serial.print(AD0);
  Serial.print(". Temperaturen (grader C) er udregnet til: ");
  Serial.println(temperatur);
  delay(800);
}

Sketch uses 4042 bytes (12%) of program storage space. Maximum is 32256 bytes.
Global variables use 256 bytes (12%) of dynamic memory, leaving 1792 bytes for local variables.

COM4
A/D value: 358.00. Temperaturen (grader C) er udregnet til: 35.06
A/D value: 357.00. Temperaturen (grader C) er udregnet til: 35.18
A/D value: 358.00. Temperaturen (grader C) er udregnet til: 35.06
A/D value: 360.00. Temperaturen (grader C) er udregnet til: 34.83
A/D value: 363.00. Temperaturen (grader C) er udregnet til: 34.48
A/D value: 362.00. Temperaturen (grader C) er udregnet til: 34.60
A/D value: 358.00. Temperaturen (grader C) er udregnet til: 35.06
A/D value: 357.00. Temperaturen (grader C) er udregnet til: 35.18
A/D value: 358.00. Temperaturen (grader C) er udregnet til: 35.06
A/D value: 360.00. Temperaturen (grader C) er udregnet til: 34.83
A/D value: 362.00. Temperaturen (grader C) er udregnet til: 34.60
A/D value: 363.00. Temperaturen (grader C) er udregnet til: 34.48
A/D value: 359.00. Temperaturen (grader C) er udregnet til: 34.94
A/D value: 357.00. Temperaturen (grader C) er udregnet til: 35.18
A/D value: 358.00. Temperaturen (grader C) er udregnet til: 35.06
A/D value: 359.00. Temperaturen (grader C) er udregnet til: 34.94
A/D value: 361.00. Temperaturen (grader C) er udregnet til: 34.71
A/D value: 364.00. Temperaturen (grader C) er udregnet til: 34.37
A/D value: 362.00. Temperaturen (grader C) er udregnet til: 34.60
A/D value: 358.00. Temperaturen (grader C) er udregnet til: 35.06
A/D value: 357.00. Temperaturen (grader C) er udregnet til: 35.18
A/D value: 359.00. Temperaturen (grader C) er udregnet til: 34.94
A/D value: 361.00. Temperaturen (grader C) er udregnet til: 34.71
A/D value: 363.00. Temperaturen (grader C) er udregnet til: 34.48
A/D value: 364.00. Temperaturen (grader C) er udregnet til: 34.37
A/D value: 360.00. Temperaturen (grader C) er udregnet til: 34.83
A/D value: 357.00. Temperaturen (grader C) er udregnet til: 35.18
Autoscroll Show timestamp No line ending 9600 baud

```


5 PROJEKT - VEJRSTATION

Der skal laves en vejrstation med måling af vindhastighed og udetemperatur

Vindhastighed (i m/s) og temperatur (°C) skal vises på skærmen via serial monitor.

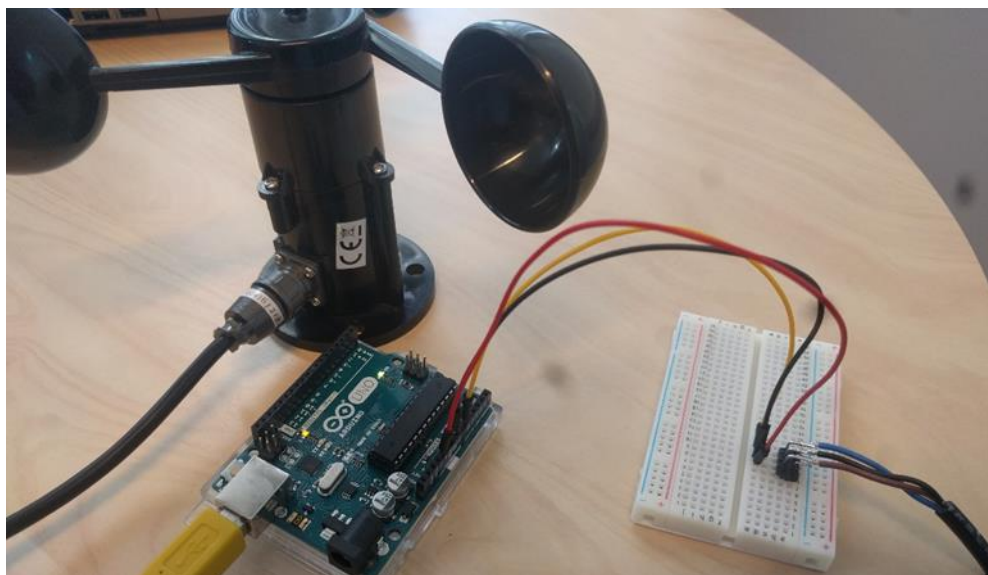
5.1 TEMPERATURSENSOR

Temperatursensoren skal laves med en NTC modstand (og kalibreres) som beskrevet i kapitel 4.

5.2 VINDSENSOR

Vindsensoren skal have en forsyningsspænding på 5V dc.

Vindsensoren afgiver et udgangssignal på 0-5V, hvor 0V er 0 m/s og 5V er 32 m/s



Vindsensoren får forsyningsspænding direkte fra Arduino-printet

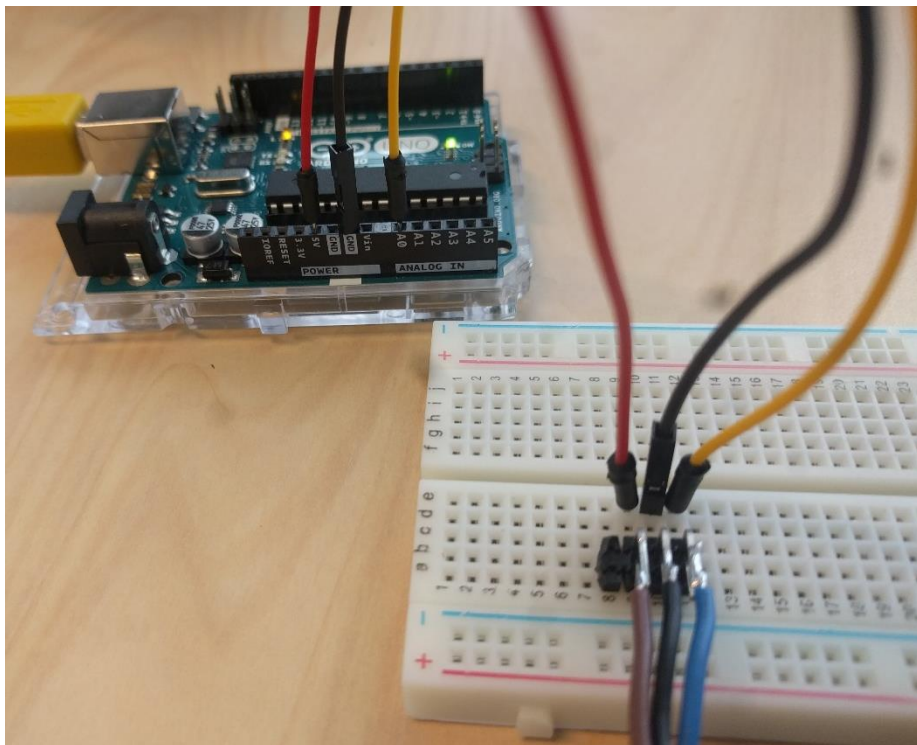
Målesignalet er lineært proportionalt med vindhastigheden.

Der er tre ledninger

sort: 0V forsyningsspænding
brun: +5V forsyningsspænding
blå: Udgangs signal fra vindmåler

Opgave 5.1 - Opbygning af kredsløb

a) Opbyg følgende (det er A0 der anvendes som analog ind)



b) Her er Arduino-koden som indlæser et signal fra AD0 indgangen og leverer et antal AD-count der vises i monitor.

```
// Her læses AD0 indgangen og skrives på skærmen
// 21/11-2021 Ken Mathiasen

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

// Her er main-loop
void loop() {
  int AD0 = analogRead(A0);   // en variabel til at gemme den læste værdi i
                              // tallet kan være fra 0-1024,
                              // hvor 0 er minimum-værdien og 1024 er max

  Serial.print("A/D value: ");
  Serial.println(AD0);

  delay(300);
}
```

c) Koden skal ændres således vindhastigheden udlæses i m/s og vises i monitoren på PC.

d) Temperatursensor

Koden fra kapitel 4 bruges som udgangspunkt for en spændingsmåling (AD1) fra et NTC-kredsløb. Signalet skal konverteres til en udlæsning i °C i monitoren, samtidig med vindhastigheden.

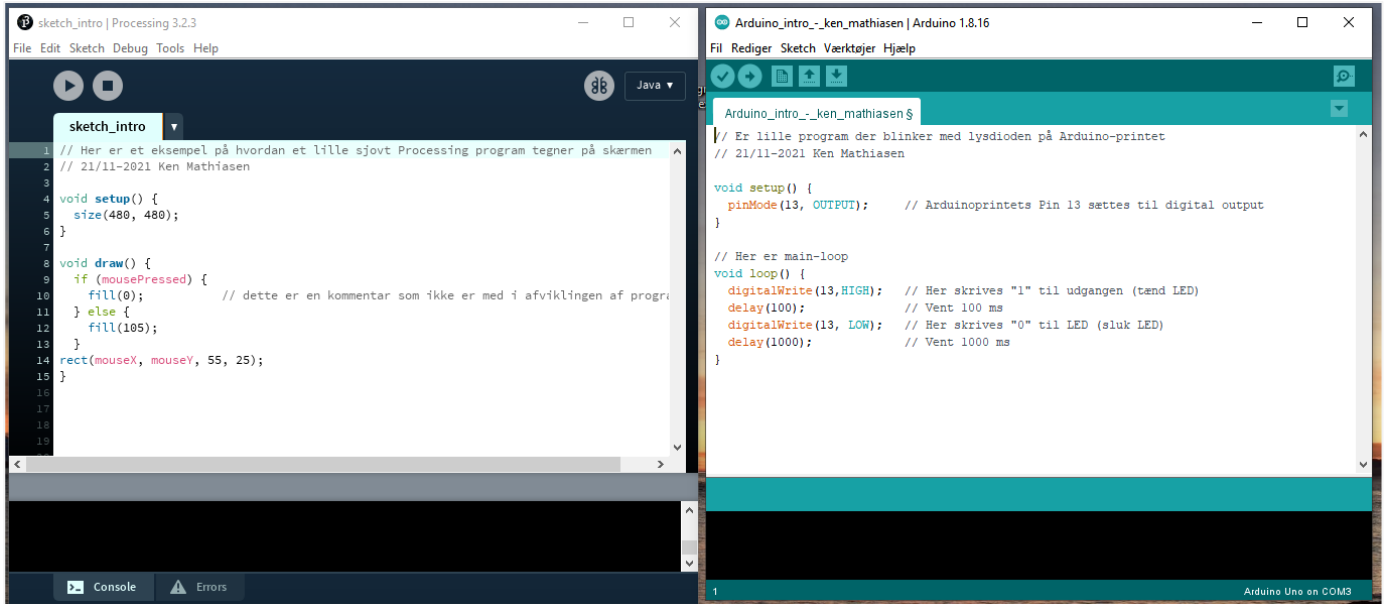
e) Sæt mange forklarende tekster/kommentarer ind i koden.

f) Den færdige program skal afleveres som en ino-fil i Lectio.

6 OVERFØRSEL AF DATA FRA ARDUINO TIL FIL PÅ PC

Ligesom Arduino er nem at gå til (man kan finde mange gode eksempler og videoer på nettet), er der et godt community omkring programmeringssproget Processing.

Det er (alt andet lige) nemmere at programmere i Processing end i det hardware-nære Arduino-sprog.



Programmeringsmiljøet for Processing og Arduino minder meget om hinanden

En ting Processing er godt til, er at modtage data fra en seriel kanal (fx serial monitor fra Arduino) og gemme data som en komma-separeret fil på harddisken. Denne fil vil senere kunne importeres i Excel.

I det efterfølgende vises nogle Processing-programmer som kan bruges til sammen at opsamle og gemme data.

OPGAVER - INSTALLATION AF PROCESSING

Opgave 1.1 - hent appen

Gå ind på <https://processing.org>

Klik download og installer appen.



OPGAVER - ARBEJDE MED PROCESSING

Opgave 1.1 - afprøv Processing

a) Indtast (eller kopier) følgende og afprøv programmet

```
// Her er et eksempel på, hvordan et lille sjovt Processing program tegner på skærmen
// 21/11-2021 Ken Mathiasen

void setup() {
  size(480, 480);      // dette er størrelsen af det grå vindue man kan tegne i
}

void draw() {
  if (mousePressed) {
    fill(0);          // hvis musen er klikket, tegnes med en anden farve
  } else {
    fill(105);
  }
}

rect(mouseX, mouseY, 55, 25);
}
```

Opgave 1.2 - skriv til en fil, med Processing

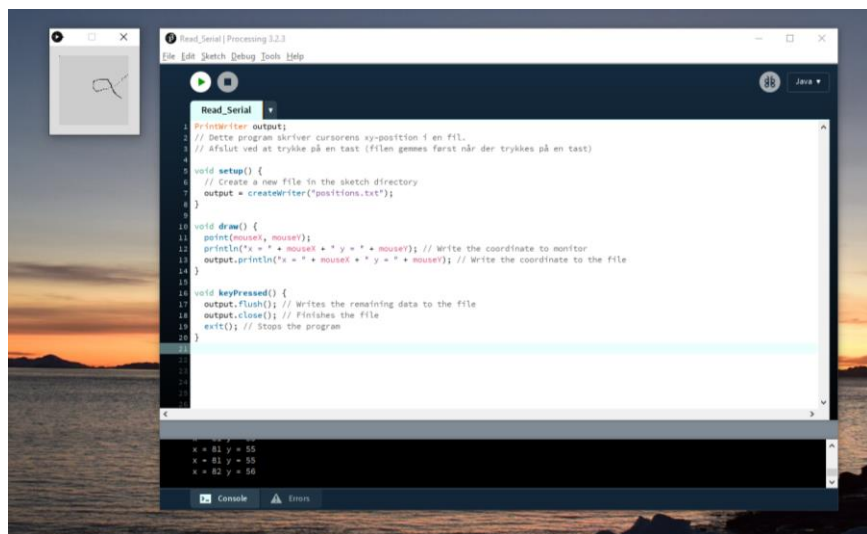
a) Indtast (eller kopier) følgende og afprøv dette program "file_write.pde" fra <https://processing.org>. Programmet bruges ved at lade PC'ens cursor tegne i det grå felt og efterfølgende trykke på en tast. Herefter oprettes en fil (positions.txt) med cursorens (x;y) positioner. Filen kan åbnes med NotePad.

```
PrintWriter output;
// Dette program skriver cursorens xy-position i en fil.
// Afslut ved at trykke på en tast (filen gemmes først når der trykkes på en tast)

void setup() {
  // Create a new file in the sketch directory
  output = createWriter("positions.txt");
}

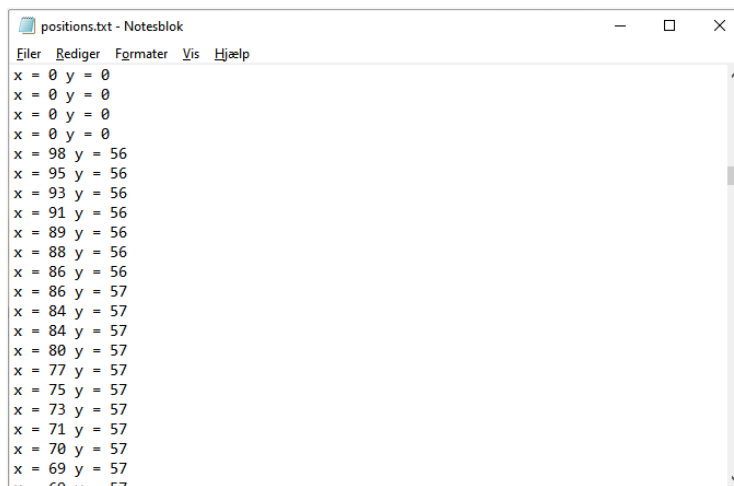
void draw() {
  point(mouseX, mouseY);
  println("x = " + mouseX + " y = " + mouseY); // Write the coordinate to monitor
  output.println("x = " + mouseX + " y = " + mouseY); // Write the coordinate to the file
}

void keyPressed() {
  output.flush(); // Writes the remaining data to the file
  output.close(); // Finishes the file
  exit(); // Stops the program
}
```

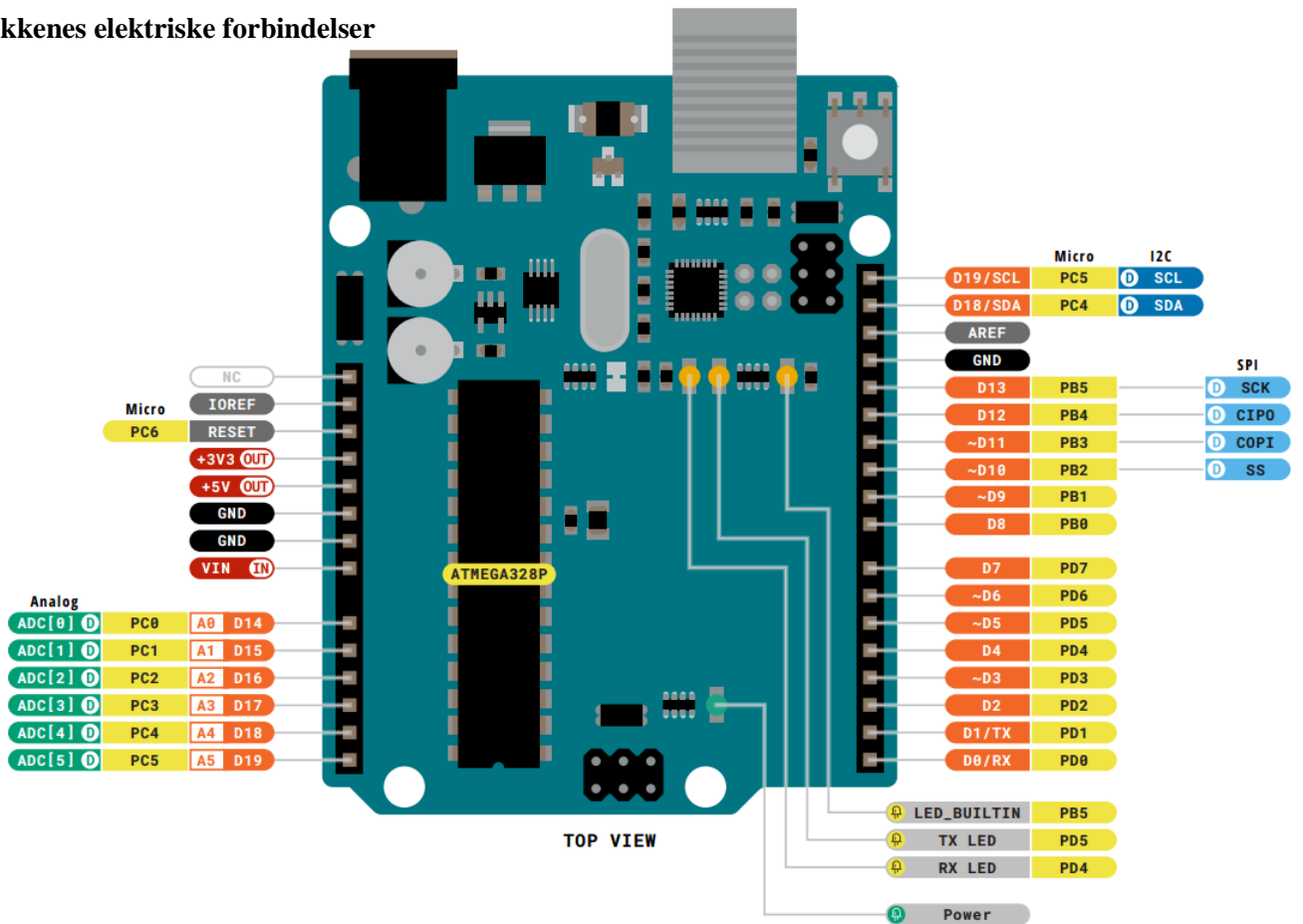


Her køres programmet der opsamler data (de ses i bunden af vinduet). Når man peger med musen i det lille grå felt, registreres cursorens xy-koordinat

Her er indholdet af filen "positions.txt" vist

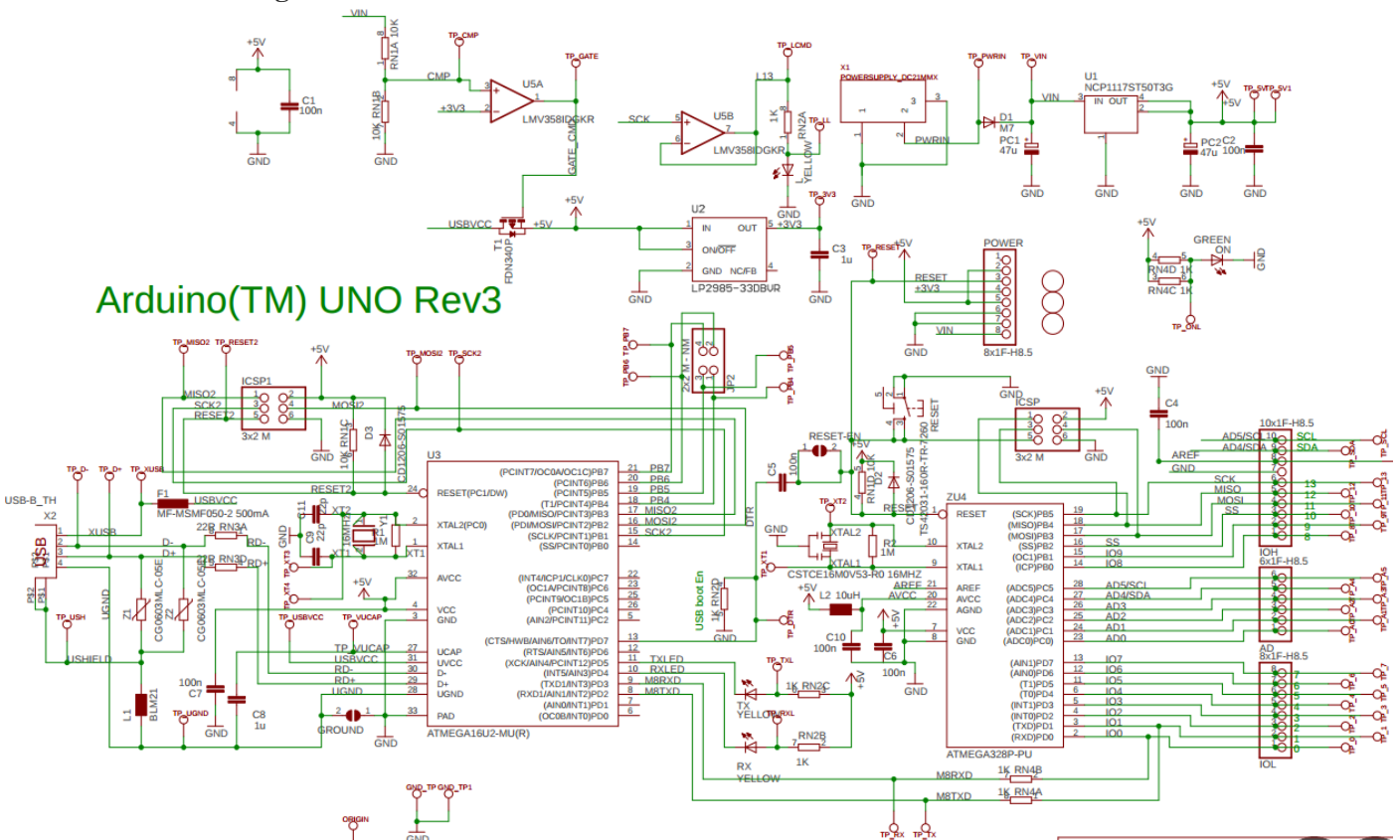


Stikkens elektriske forbindelser



Arduino Uno, Connector Pinouts (fra <https://docs.arduino.cc/hardware/uno-rev3>)

Der elektrisk diagram



Arduino Uno, schematics (fra <https://docs.arduino.cc/hardware/uno-rev3>)

Opgave 1.3 - læs fra den serielle port

Efter ide fra <https://processing.org>

Inden denne Processing kode afvikles, skal Arduino'en være opstartet og køre. Processing-programmet fungerer ikke, hvis Arduinos monitor program kører samtidig (Win10 tillader tilsyneladende kun, at der er et program som lytter med på den serielle kanal ad gange).

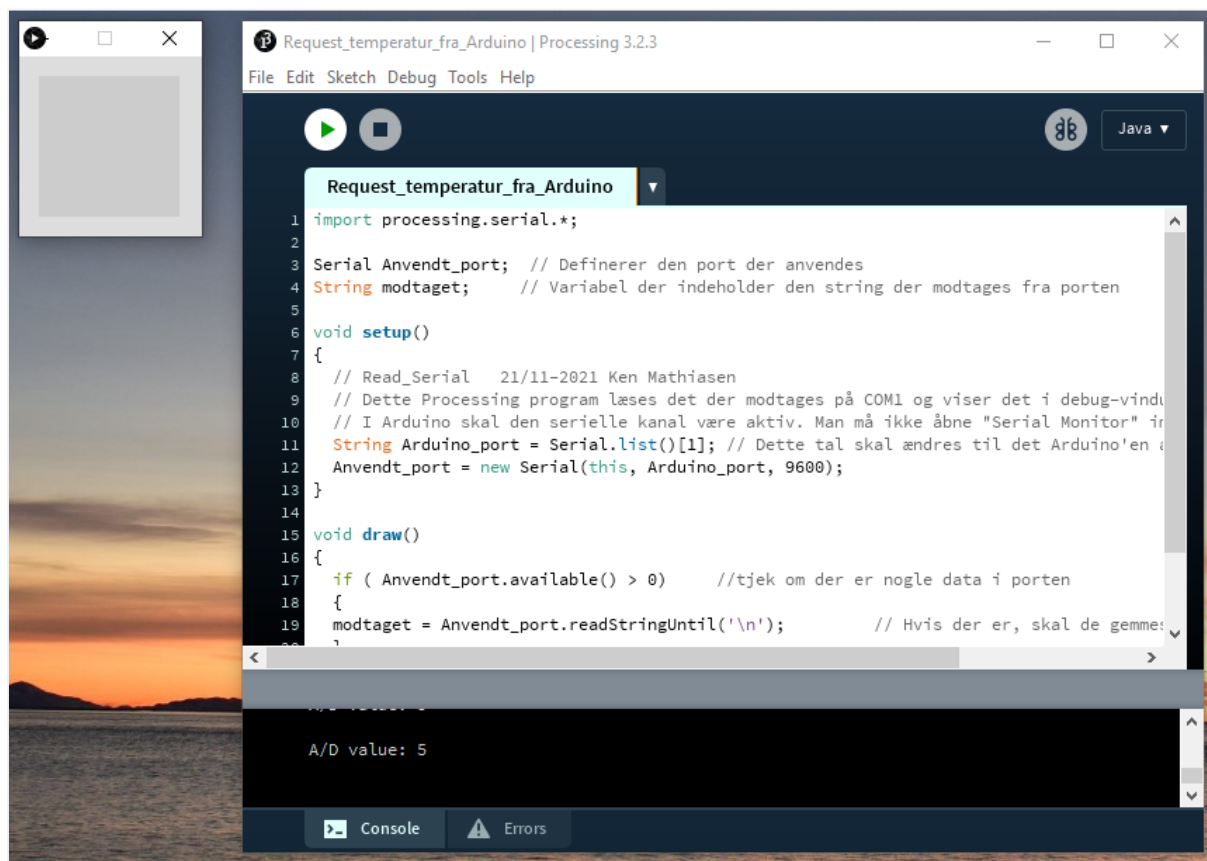
Man skal manuelt indstille den serielle kanal Arduino anvender i Processing koden. Det gøres ved at ændre "[1]" i SerialList til et andet tal (se i koden).

```
import processing.serial.*;

Serial Anvendt_port; // Definerer den port der anvendes
String modtaget;     // Variabel der indeholder den string der modtages fra porten

void setup()
{
  // Read_Serial 21/11-2021 Ken Mathiasen
  // Dette Processing program læses det der modtages på COM1 og viser det i debug-vinduet
  // I Arduino skal den serielle kanal være aktiv. Man må ikke åbne "Serial Monitor" imens.
  String Arduino_port = Serial.list()[1]; // Dette tal skal ændres til det Arduino'en anvender
  (0,1,2,3,4...)
  Anvendt_port = new Serial(this, Arduino_port, 9600);
}

void draw()
{
  if ( Anvendt_port.available() > 0) //tjek om der er nogle data i porten
  {
    modtaget = Anvendt_port.readStringUntil('\n'); // Hvis der er, skal de gemmes
  }
  println(modtaget); //print it out in the console
}
```



Her er et screenshot fra en kørsel - i bunden af vinduet ses de A/D værdier der er modtaget fra Arduino

Arduino Introduktion

Hands-On programmering i fysik og informatik (stx, htx, eux)

© 2024 Ken Mathiasen

Bogen giver en introduktion til minicomputeren Arduino. Arduino programmeres med tekstbaseret kode. Bogen er lavet, så den umiddelbart kan udleveres til eleverne, som arbejder selvstændigt efter den. Den pædagogiske tilgang er "Worked Example". Der viser et eksempel, som eleven efterfølgende modificerer.

Bogen må gerne kopieres

Bogen er lavet med hjælp fra velvillige sponsorer og er dermed gratis i pdf-udgaven, så kopiering af denne bog må gerne finde sted (også selvom du ikke har købt den) - det skal bare registreres med en mail hos Copydan Tekst & Node. Der sendes en mail til kopi@copydan.dk, hvor man vedhæfter bogen i pdf-udgaven og angiver de 10 sider der bruges mest. Her er et forslag til mailen:

Til kopi@copydan.dk

Vedr Registrering af klassesæt

Besked Korsbæk Handelsskole (skole nr. kendes desværre ikke).
Kopieret fra "Arduino Introduktion", Ken Mathiasen ISBN 978-87-973415-4-4
Siderne 10-20 til 18 elever (se vedlagte).
mvh
Overlærer Andersen.

Hæftet kan købes hos mathiasen@runbox.com for 50,- kr pr. stk (inkl moms, plus fragt)

Hent hæftet i en pdf-udgave (gratis): www.mathiasen.eu → books.

P	T	G	M	k		m	μ	n	p	f
peta	tera	giga	mega	kilo	1	milli	mikro	nano	pico	femto
$1 \cdot 10^{15}$	$1 \cdot 10^{12}$	$1 \cdot 10^9$	$1 \cdot 10^6$	$1 \cdot 10^3$	$1 \cdot 10^0$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-9}$	$1 \cdot 10^{-12}$	$1 \cdot 10^{-15}$

ISBN 978-87-973415-4-4



9 788797 341544 >