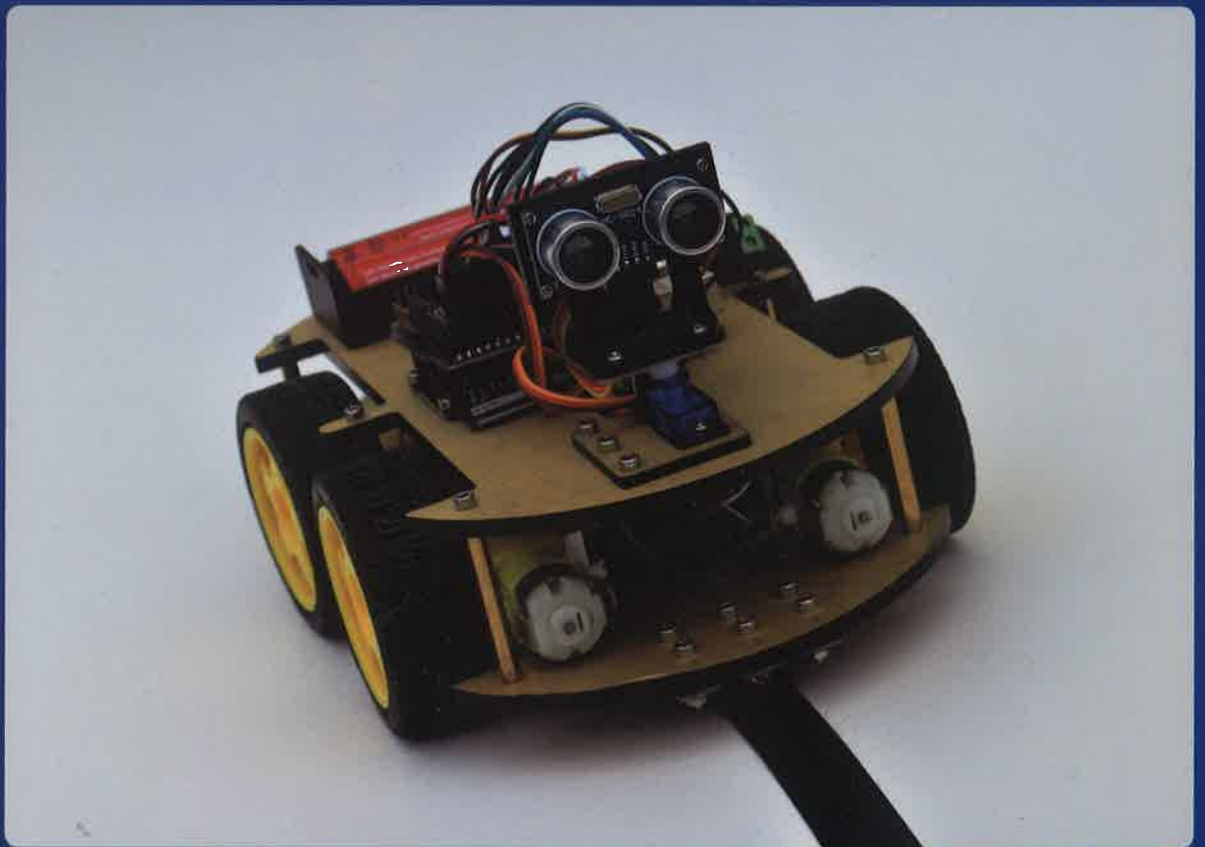


Ole Sørensen

Sensorer

Måling Styring Kontrol



Fysikforlaget 2019

ARDUINO

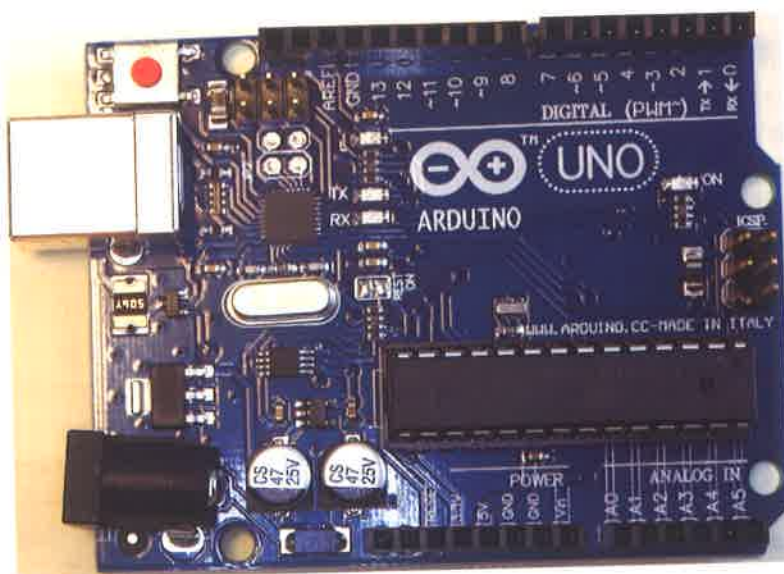
Beskrivelse af Arduino

Arduino er en minicomputer, der kan

- styre ting i dens omverden: Lamper, motorer, ventiler, højttalere m.m.
- måle på variable i dens omverden. Tryk på knap, elektriske spændinger m.m.

Signalerne kommer ofte fra en sensor, der omsætter fx en temperatur til en spænding

- programmeres, dvs. instrueres i at foretage bestemte handlinger.



Figur 4.1
Arduino UNO.

Arduino kan arbejde, når den er forbundet til en computer. Når den er programmeret og får strøm fra fx et batteri, kan den arbejde på egen hånd.

Der findes mange typer Arduino, der er specialiserede til bestemte formål. Vejledning og programmer er skrevet til basismodellen Arduino UNO.

Hvorfor arbejde med Arduino?

Et af de mange nye begreber, der tales om, er *Internet of Things*. At en lang række ting lige fra biler og køleskabe til løbesko og tøj forbindes til internet. Tingene kan udveksle informationer.

For eksempel kan dine løbesko registrere mulig overbelastning af anklerne og sende advarslen videre til dit løbe-ur.

Som almindelig borger er en vis indsigt i programmering og styring derfor vigtig for at følge med i udviklingen inden for it. Sidst men ikke mindst er it et område hvor beskæftigelsesmulighederne er særdeles gode.

At programmere eller kode

At skrive kode kræver en systematisk tankegang. Der er flere steder på nettet, hvor man kan lære kodning. Tag for eksempel et kig på hourofcode.com/dk. Her kan man blandt andet lære kodning mens man hjælper Disney-figuren Moana med at fange fisk og bekæmpe pirater: hourofcode.com/moana.

I gang med ARDUINO

Her følger en trin-for-trin vejledning i at komme i gang med Arduino.

0. Installation

Hvis du ikke har softwaren til ARDUINO på din computer, skal den først hentes: arduino.cc/en/Main/Software. Under *Download the Arduino IDE* vælges relevant version (Windows, MAC etc.). Følg anvisningerne og installer programmet.

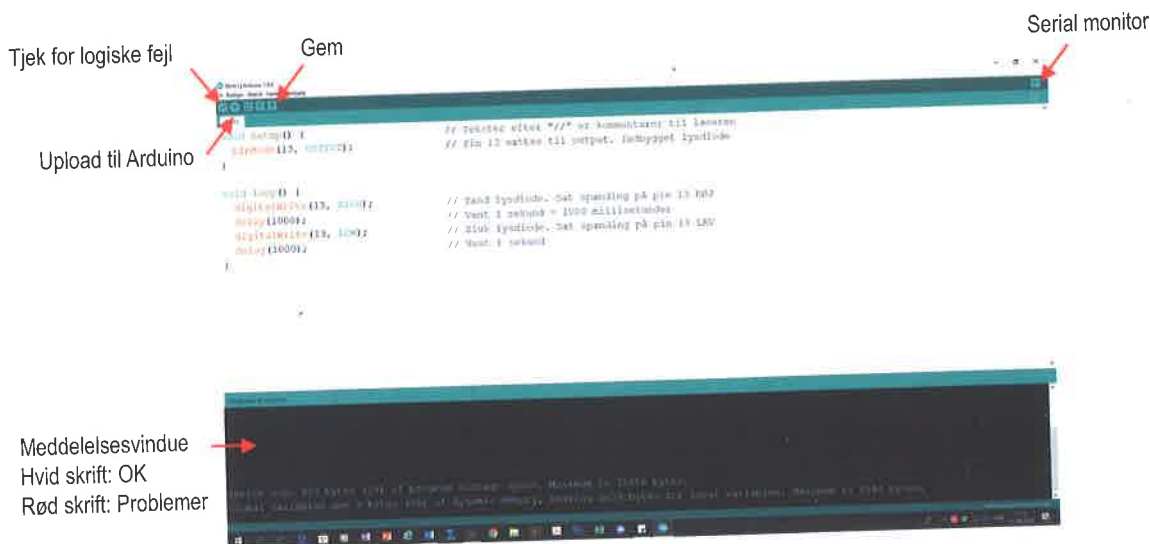
1. Opsæt kommunikationsport

- Åben Arduino programmet
- Tilslut Arduino til en USB port på computeren
- Vælg menupunkt Tools (eller Værktøjer, hvis computeren taler dansk)
- Vælg Port

På MAC vælges fx /dev/cu.usbmodem1421 (Arduino/Genuino Uno). I Windows vælges fx COM3 eller COM4. Windows 8 kan drille. Normalt genkendes Arduino automatisk lige som andre USB-enheder.

Nogle Arduinoer bruger en ikke-standard, men billig kommunikationschip. Den kan for eksempel hedde R3 CH340. Computeren kan have svært ved at genkende en sådan Arduino. Løsningen er at hente en driver på producentens (kinesiske) hjemmeside. Der er hjælp til dette på internet. Søg på chip-navnet.

Tjek evt. kommunikationen ved klik på Serial monitor øverst til højre i programvinduet.



Figur 4.2

Skærbillede af Arduino IDE (Integrated Development Environment – integrerede udviklingsmiljø). Her skrives og redigeres de programmer (Sketches), der styrer Arduino.

2. Skriv eller vælg et program

Der er mange gode eksempler i menupunkt Fil → Eksempler. Du lærer hurtigt at kopiere eksemplerne og lave dine egne programmer.

Kontroller programmet for logiske fejl (fx manglende krøllede parenteser { og }) ved at klikke på fluebenet øverst til venstre i programvinduet. Hvis du ændrer programmet, så husk at gemme det.

3. Opbyg et kredsløb

Arduino kan ikke meget i sig selv, ud over at blinke med en lille lysdiode.

Alt afhængigt af programmet skal Arduino tilsluttes *sensorer*, så den kan måle noget i dens omverden, eller *aktuatorer*, så den kan styre noget i dens omverden. En aktuator kan fx være en lysdiode, men også en højttaler, et relæ, en motor m.m. Mere om dette i eksemplerne herunder.

Hvis man vil være på den sikre side, kan man afbryde forbindelsen mellem Arduino og computeren, mens man opbygger kredsløbet.

4. Upload program til Arduino

Klik på højrepilen øverst til venstre i programvinduet. Arduino begynder at udføre programmet straks efter det er uploadet. Det vil være en god ide at lave eksperiment 4.1 og 4.2 inden du læser videre.

At programmere Arduino

Et program indeholder tre dele

1. definitioner
2. setup
3. loop

I definitionsdelen defineres konstanter og variable, fx

```
int led = 13; // Det svarer til kommandoen led:=13 i et CAS-program.  
// int betyder heltal (integer)
```

- Hvis der ikke er variable at definere, kan definitionsdelen undværes
- Setupdelen udføres kun én gang. Her defineres bl.a. konstanter
- Loopdelen gentages igen og igen til en evt. slutbetingelse er opfyldt, eller til Arduino slukkes eller resettes.

Nøgleordet *void* betyder, at programmet eller funktionen ikke forventes at give nogen information tilbage.

De krøllede parenteser { } afgrænser det, der skal gøres.

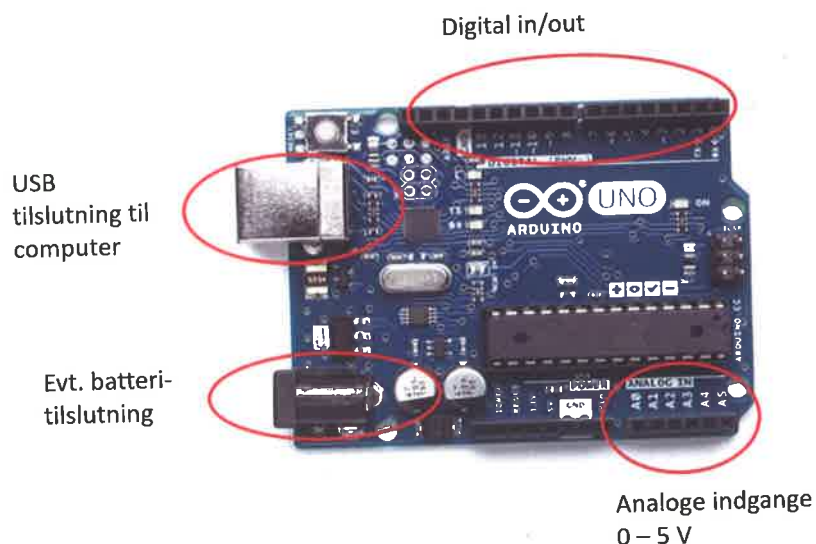
Grundstruktur i program

```
// Definitioner  
// Her kan du definere konstanter og variable  
//  
void setup( ) {  
    // Her skriver du setupkode, der kun afvikles én gang  
}  
  
void loop( ) {  
    // Her skriver du hovedprogrammet der gentages  
}
```

Læs mere på arduino.cc. Begynd under punktet *Learning*.

Arduinos ind- og udgange

Arduinos kommunikation med omverden sker gennem stikforbindelserne (*pins*), der ses øverst og nederst på figur 4.3. Øverst findes pins til digitale ind- og udgange. Man kan selv bestemme, om den enkelte pin skal være indgang eller udgang. I programmet i Eksperiment 4.2 bestemte vi, at pin 13 skal være udgang med kommandoen `pinMode(13, OUTPUT)`.



Figur 4.3
Arduinos indgange og udgange.



Figur 4.4
DuPont-kabler i mange farver.
Kablerne fås også med hunstik i den ene eller begge ender

Når pin 13 sættes høj med kommandoen `digitalWrite(13, HIGH)`, sættes en spænding på 5 V på pin 13.

Når vi taler om, at vi måler en elektrisk spænding i et digitalt system, mener vi i virkeligheden, at vi måler en spændingsforskel i forhold til et fastlagt nulpunkt. Dette nulpunkt betegnes GND (ground, jordforbindelse) og kan findes flere steder på Arduino. Det kan sammenlignes med minuspolen på et batteri.

Find på billedet, eller på din egen Arduino, 3 punkter med spændingen GND.

Et digitalt signal kan antage netop 2 værdier af spændingen, nemlig 0 V og 5 V – her underforstået spændingsforskel i forhold til GND. Andre pins har en fast spænding, 5 V eller 3,3 V. Find på billedet, eller på din egen Arduino, et punkt med spændingen 5 V.

Når vi i praksis skal have fat i signalet på en pin, bruger vi tynde farvede kabler (ofte kaldet jumpers eller duPont-kabler), der stikkes ned i pinnen. Den anden ende af kablet kan forbindes til en prøveplade, hvor vi kan eksperimentere.

Vi kan også forbinde kablet til et voltmeter eller et andet apparat. Det anbefales kraftigt at holde en "farvejustits", så alle kabler, der er forbundet til GND har samme farve, fx sort, og alle kabler forbundet til 5 V har samme farve, fx rød.

Eksperimenter



Eksperiment 4.1 Dit første Arduino-program

Skriv følgende program. Skriv krøllede parenteser { og } samt semikolon ; præcis som der står i teksten. Eventuelt kan du undlade "//" og al tekst efter "//".

Upload til Arduino og få en indbygget lysdiode til at blinke.

```
void setup( ) {                                // Tekster efter "//" er kommentarer til læseren
  pinMode(13, OUTPUT);                          // Pin 13 sættes til output. Indbygget lysdiode
}

void loop( ) {
  digitalWrite(13, HIGH);                       // Tænd lysdiode. Sæt spænding på pin 13 HØJ
  delay(1000);                                  // Vent 1 sekund = 1000 millisekunder
  digitalWrite(13, LOW);                        // Sluk lysdiode. Sæt spænding på pin 13 LAV
  delay(1000);                                  // Vent 1 sekund
}
```

Tillykke, du har nu skrevet dit første Arduino-program til at styre noget i din omverden. Måske ikke så ophidsende, men nu er du i gang.

Eksperiment 4.2

Prøv at ændre på den ene eller begge delay-tider og se, at du faktisk kan kontrollere blink-frekvensen. Sæt for eksempel delay til 100, dvs. 100 ms = 0,1 s.

Prøv at slutte et 9 volt batteri til Arduino. Nu kan du fjerne USB-kablet til computeren og overbevise dig om, at Arduino kan selv. Det er ikke din computer der styrer blinket.



Eksperiment 4.3

- Tilslut to kabler til et voltmeter. Det sorte kabel til 0 på voltmeteret (kaldes også COM, GND eller -), det røde kabel til V (kaldes også +).
- Sæt med krokodillenæb det sorte kabel til GND på Arduino og det røde kabel på 5 V
- Nu skulle voltmeteret gerne vise spændingsforskellen 5,0 V
- Forbind derefter det røde kabel til pin 13, mens blink-programmet fra Eksperiment 4.2 kører
- Nu skulle voltmeteret gerne skifte mellem 0 V og 5,0 V

Den røde ledning kan nu bruges som en "probe", der måler spændingen på en ønsket pin.

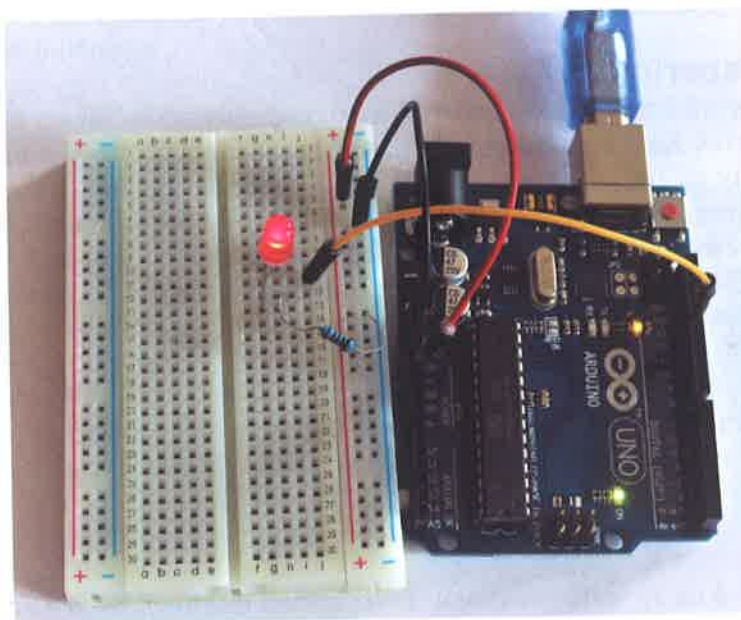
- Kontroller, at spændingen på pin mærket 3,3 V er, hvad den giver sig ud for
- Hvad er spændingen på en pin mærket GND?

Eksperiment 4.4 **Lysdiode på breadbord**

Vi går frem i små skridt. Næste skridt er at få en lysdiode til at lyse. Som bekendt leder en diode kun strøm i den ene retning.

Det længste ben på en lysdiode er + og det korte er -.

- Opbyg et kredsløb, der ser ud som på figur 4.5.



Figur 4.5

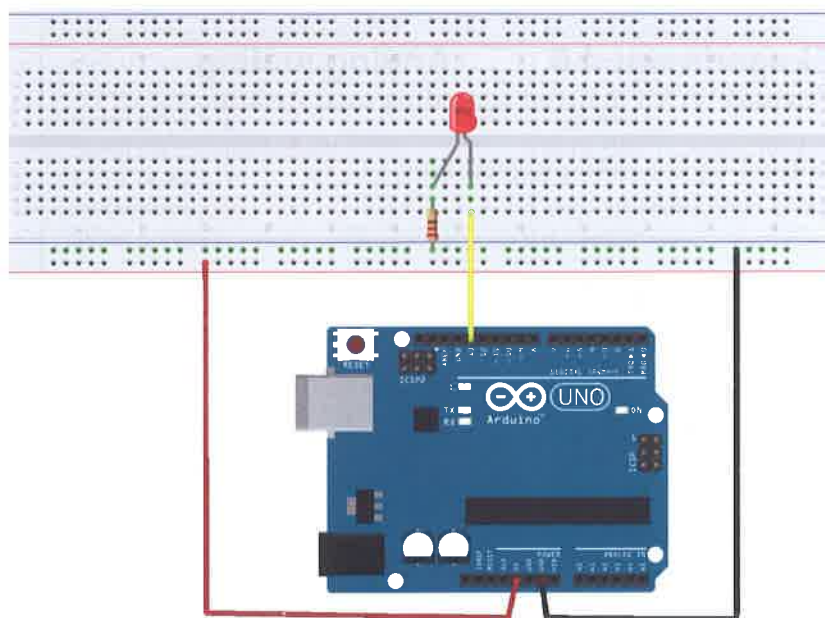
Foto af kredsløb med lysdiode på prøveplade. Den gule ledning er tilsluttet pin 13 på Arduino. Bemærk beskyttelsesresistoren, der forbinder lysdiodens minuspol med GND.

Bemærk den meget vigtige beskyttelsesresistor, som her er 220 Ω . Den anvendte prøveplade er bygget således, at pins yderst til venstre og højre er forbundet (markeret med rød eller blå streg).

Symbolerne + og – tolker vi som henholdsvis 5 V og 0 V (GND). Længere inde mod midten af breadboard er forbindelserne derimod vandrette, således at fem pins mærket a, b, c, d og e er forbundet. Tilsvarende er pins mærket f, g, h, i og j forbundet.

Vi kan referere til en bestemt pin med et tal og et bogstav lige som i et regneark. På figur 4.5 er lysdiodens –ben forbundet til pin g13, mens +benet er på g11. g11 er forbundet med j11, der via gul ledning er forbundet til Arduinoens pin 13.

På figur 4.5 er lysdiodens –ben via en 220 Ω resistor forbundet til den blå minusrække. Lysdiodens +ben (det lange) er via gul ledning forbundet til Arduinoens pin 13. Med opstilling som på billedet vil dioden blinke alt efter om pin 13 på Arduino er HØJ eller LAV.



Figur 4.6
Diagram af det samme kredsløb som i figur 4.5 med lysdiode på breadbord. Diagrammet er – lige som mange andre af bogens diagrammer – tegnet med Fritzing, fritzing.org.

Måling med Arduino

Nederst til højre på billedet af Arduino ses 6 pins mærket ANALOG IN. Det er som navnet antyder analoge indgange, hvor spændinger mellem 0 og 5 V kan måles.

Spændingen kan for eksempel komme fra et elektronisk termometer eller en anden sensor. Spændingen konverteres til et tal mellem 0 og 1023.

Processen kaldes Analog til Digital konvertering (Analog–Digital–Converting, ADC, se kapitel 3). Her omsættes en spænding, der kan variere kontinuert mellem 0 og 5 V, til ét ud af 1024 tal. For eksempel omsættes spændingen 2,5 V til tallet 512.



Det nytter selvfølgelig ikke meget at måle, hvis vi ikke kan se resultaterne. For at se resultaterne, skal de sendes til computeren via USB-forbindelsen.

I de næste eksperimenter arbejdes med et lille program, der måler spændingen på pin A0 og sender det til computeren.

Eksperiment 4.5 Design et voltmeter

- a) Skitser en opstilling, så spændinger vises med lysdioder. Jo højere spænding, desto flere dioder skal lyse.

I Eksperiment 5.15 forklares mere om, hvordan opstillingen bygges og programmeres

Eksperiment 4.6 Analog måling

Formål: Vi vil måle en spænding med Arduino.

Indtast og kørs måleprogrammet.

```
int count=0;                // Nulstil tæller

void setup() {
  Serial.begin(9600);        // Sæt op til seriel kommunikation via USB
}

void loop() {
  Serial.print(count);       // Skriv tællerens visning til seriel monitor
  Serial.print(" ");         // Mellemrum
  Serial.println(analogRead(A0)); // Skriv tal målt på analog indgang A0. Linjeskift
  delay(1000);               // Vent 1 sekund mellem hver måling
  count = count + 1;         // Læg 1 til tælleren
}
```

Web

Eksperiment4_6.ino

I Arduino-vinduet klikkes på Serial Monitor øverst til højre. Nu vises en skærm med måleresultater. Der er indbygget en tæller (count). Da der går et sekund mellem hver måling, fungerer count samtidig som tidtager.

Forbind pin A0 skiftevis til GND, 5 V og 3,3 V og forklar resultatet.

Mål nogle andre spændinger. Husk at forbinde minus på spændingskilden til GND på Arduino.

Eksperiment 4.7 Forbedret analog måling

Formål: I eksperiment 4.6 så vi kun de rå resultater. Nu vil vi gerne have vist spændingen i volt.

Indtast og kørs måleprogrammet. Se resultatet på Serial Monitor. Forbind pin A0 til GND, 5 V og 3,3 V og forklar resultatet.



```
int count=0; // Nulstil tæller
int sensorValue; // Definer variabel til at gemme værdi
float U; // Definer U som decimaltal

void setup() {
  Serial.begin(9600); // Sæt op til serial kommunikation via USB
}

void loop() {
  sensorValue=analogRead(A0); // måler værdien på analog indgang 0:
  U=5.00*sensorValue/1023; // Omregn fra heltal til spændingsværdi
  Serial.print(count); // Skriv tællerens visning til serial monitor
  Serial.print("\t"); // Tabulering
  Serial.print(sensorValue); // Skriv den "rå" værdi læst af A0
  Serial.print("\t");
  Serial.print(U,3); // Skriv spænding med 3 decimaler
  Serial.println(); // Linjeskift

  delay(1000); // 1 sekund mellem hver måling
  count = count + 1; // Forøg tæller med 1
}
```

Mål nogle andre spændinger. Husk at forbinde minus på spændingskilden til GND på Arduino.

Web

Eksperiment4_7.ino

Eksperiment 4.8 Bygge en alarm

Vi vil bygge et apparat, der reagerer på en ændring i omgivelserne. Ændringen i omgivelserne kan være en temperatur eller en vandhøjde, der overstiger en kritisk værdi. Reaktionen kan være, at en rød alarmlampe tændes.

I dette eksempel måles temperaturen med en komponent ved navn LM35.

LM35 tilsluttes 5 V til +Vs og afgiver da en spænding, V_{out} , der er proportional med temperaturen. Leverandøren oplyser, at V_{out} vokser med 10 mV/°C. Tegningen viser LM35 fra neden. Det midterste ben er V_{out} .

Herunder ses et program, der udfører det ønskede. Bemærk kommentarerne.

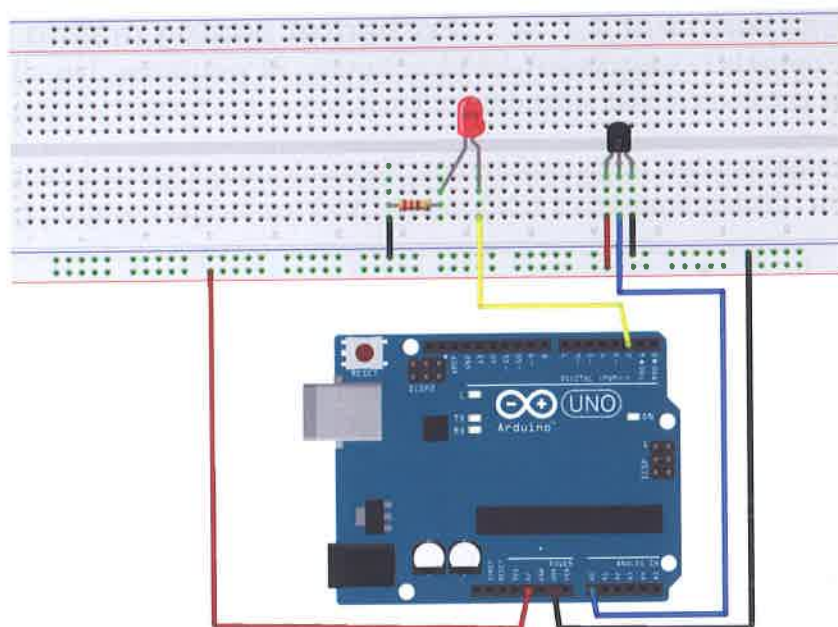


LP Package
3-Pin TO-92
(Bottom View)



Figur 4.7

Komponenten LM35 med tilhørende diagram, der viser komponenten set nedefra.



Figur 4.8

Gul ledning fører signal fra pin 2 til + på lysdiode. Blå ledning fører udgangssignal fra temperatursensor LM35 til analog indgang A0.

```
int critical=50; //Den kritiske værdi findes ved at prøve sig lidt frem

void setup() {
  pinMode(2,OUTPUT); // Pin 2 defineres til output. Skal styre lysdiode
}

void loop() {
  if (analogRead(A0) > critical) // Læs signal fra sensor på pin A0
  {                               // Hvis (IF) signal større end kritisk værdi, så aktiver alarm
    digitalWrite(2,HIGH);        // dvs. sæt pin 2 HIGH og tænd lysdiode
  }                               // slut på IF- kommando
  else
  {                               // Udføres hvis betingelse i IF ikke er opfyldt,
    digitalWrite(2,LOW);         // dvs hvis temperatur er ok, så sluk lysdiode
  }
}
```

Web

Eksperiment4_8.ino

Projekter med Arduino

I dette kapitel præsenteres en række små og større projekter. Målet er, at du skal lære at bruge Arduino til at styre forskellige ting i din omverden.

Når du arbejder med projekterne, får du samtidig udviklet din kreativitet og evne til at skrive kode.

På bogens websted fys.dk/sensorer findes halv- og helfærdige Arduino-programmer (sketches). Brug gerne disse sketches som udgangspunkt for projekterne.

K A P I T E L

5

