***Rapport***

Idag så er blir husene våre smartere og smartere og det har blitt et større og større fokus på å spare ressurser og være mer effektiv med strømbruken vår. Dette innebærer blant annet å ha et god temperert rom og fortsatt holde strømregningen nede. Vi kan oppnå dette ved smart bruk av temperatur sensorer som lar oss varme rommene våre forskjellige så vi ikke sløser med bort energi.

Før i tiden hvis du ikke ville fryse hjemme så fyrte du i peisen. En peis har en viss sjarm til seg, men er ikke svert egnet til effektiv ressurs bruk, man må kutte ned skog som slipper ut CO2 som har blitt lagret av treet og når man brenner det så slipper vi ut klimagasser i atmosfæren. Heldigvis trenger vi ikke lengere å brenne trær for å oss varme. I dag så har vi varmekabler, varmepumper og elektriske ovner.

For å måle temperatur så har jeg brukt en NTC termistor. En NTC termistor fungere ved at når termistoren blir eksponert for varme så minker den elektriske motstanden og på den måten kan vi få Arduinoen til å regne om motstanden til temperatur.

For å finne temperaturen må vi vite motstanden, Arduinoen kan ikke måle motstand bare spenning. Så for å måle motstanden i termistoren må vi gjøre om formelen for en spenningsdeler som ser slik ut.

.

V*out*= spenningen mellom termistoren og kondensatoren

V*in* = Vcc (5v)

R1 = Resistansen vi vet om

R2 = Resistansen til termistoren.

Vi er ikke interessert i å finne Vout så vi gjør om formelen til å gi oss R2.

Da får vi

Nå som vi har motstanden så må vi bruke Steinhart-Hart formelen for å regne ut temperaturen. Formelen går som følger

T = temperaturen i Kelvin

R = resistansen i termistoren i ohms

A, B og C = Steinhart-Hart koeffisienter som er tre verdier som oppgis av produsenten

Steinhart-Hart koeffisienter kan variere basert på hvilken temperatur termistoren er laget for og hvilken modell og type termistor er. Formelen bruker også , men denne verdien er ofte så liten at man trenger ikke å ha den med i regningen. For å bruke formelen så må man ta motstanden på tre forskjellige temperaturer å regne ut de tre verdiene som skal blir til A, B og C. Som oftest så får du vite A, B og C verdiene til termistoren av produsenten. For en 10Kohm termistor så kan verdien til A, B og C typisk være:

A = 0.001125308852122

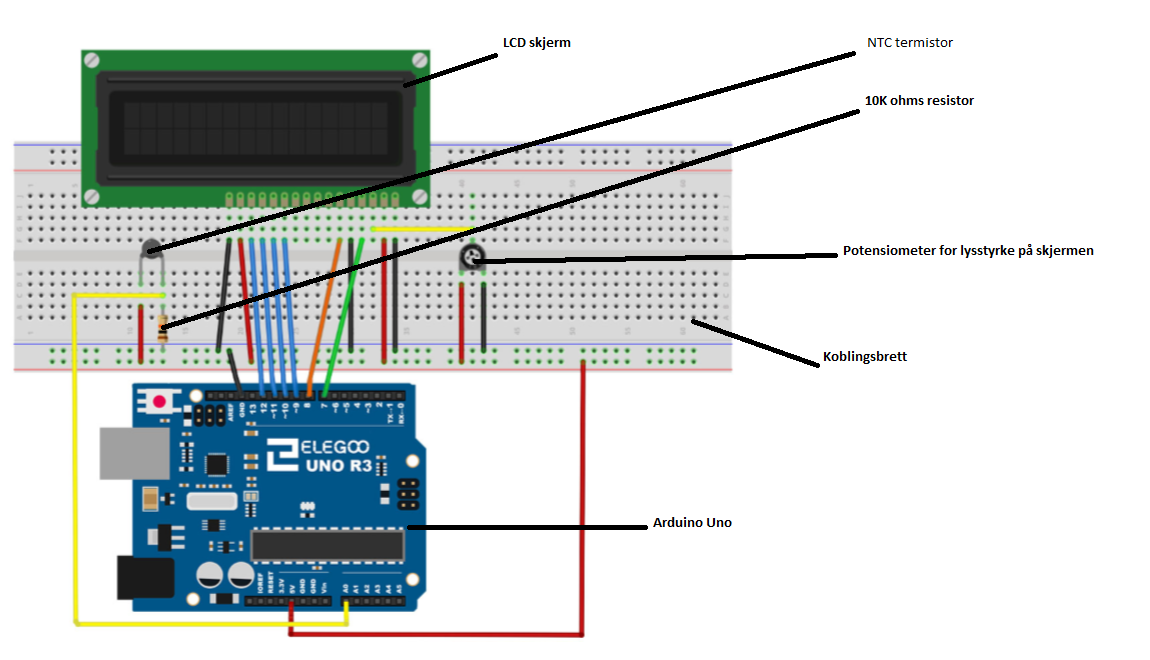
B = 0.000234711863267

C = 0.000000085663516

Jeg startet smarthus prosjektet med å lage et enkelt knappe program med en microkontroller. Mens jeg gjorde dette så lærte jeg meg nyttig informasjon om: grunnleggende koding i C, hvordan koble ting på et breadboard (koblingsbrett), hva de forskjellige elektriske komponentene gjør og hvordan knytte alt sammen. For å tilegne meg all den kunnskapen så fikk jeg hjelp fra læreren, men i all hovedsak så jeg på videoer som var anbefalt av min og læreren på NTNU[[1]](#footnote-1).

Planen min originalt var å lage temperatursensoren med samme microkontroller som jeg brukte til knappe programmet. Dette hadde vært den mest optimale løsningen, men på grunn av tidsrammen jeg fikk valgte jeg heller å bruke en Arduino Uno. Både mikrokontrolleren og Arduinoen har mange likheter, men Arduinoen har en rekke innebygde biblioteker, er mindre tidskrevende å jobbe med og har flere online resurser.

For å bygge termometret har jeg brukt dette koblings skjema:



Delene jeg brukte for dette prosjektet innebærer:

* NTC sensor
* LDC skjerm
* Potensiometer
* Koblingsbrett
* 10K ohms resistor
* Arduino Uno + overgang til PC-en
* Kabler

Her er koden jeg har brukt til prosjektet. Koden har blitt hentet fra en online ressurs, men jeg har kommentert og forstått koden. Arduinoen tar i bruk C kode som er en relativt lav kode (det betyr at koden er nærmere binærkoden som PC-en bruker enn teksten du skriver på PC-en). For å kommentere koden så må jeg skrive «//», dette forteller Arduinoen at den skal ignorere alt som står bak det.

//Her importerer Arduinoen biblioteket som printer ut teksten på LCD skjermen

#include <LiquidCrystal.h>

int tempPin = 0;

//                BS  E  D4 D5  D6 D7

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup()

{

//Setter i gang LCD skjermen

lcd.begin(16, 2);

}

void loop()

{

//Her tar Arduinoen inn dataen den får fra NTC sensoren

int tempReading = analogRead(tempPin);

//Her gjør Arduinoen om spenningen til motstand

double tempK = log(10000.0 \* ((1024.0 / tempReading - 1)));

//Her regner Arduinoen ut temperaturen

  tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 \* tempK \* tempK )) \* tempK );

 // Her gjør den om temperaturen i kelvin om til Celsius

float tempC = tempK - 273.15;

 // Her printer Arduinoen ut temperaturer til LCD skjermen

  lcd.setCursor(0, 0);

  lcd.print("Temp         C  ");

  lcd.setCursor(6, 0);

//Her venter Arduinoen 500 millisekunder.

  delay(500);

}

Når jeg hadde koblet alt opp og koblet det til Pc-en så lyste LCD skjermen opp og viste temperaturen. Arduinoen registrerer temperaturen hver halve sekund og dette gjør at Arduinoen måler nøyaktig uten å overlaste Arduinoen.

Smarthuset prosjektet har vært et gøy, men krevende prosjekt. Det har gitt meg en større forståelse og takknemlighet for all småelektronikken vi har rundt oss. Det har også lært meg mye om hvordan hardware og software jobber sammen og gir oss et lettere liv.

Utregningen jeg brukte viste seg å stemme i praksis. Det var ingen store avvik fra det jeg skrev i teorien og formelene som ble brukt holdt opp.

En feilkilde var at hvis NTC sensorens bein berørte hverandre så viste den høye utslag. Dette kommer nok av at Arduinoen veldig liten motstand og det resulterer i at den tror at NTC termistoren er super varm.

Hvis jeg hadde hatt en større tids ramme så hadde jeg gjort følgende:

* Lagt til flere sensorer så man kunne få et bedre bilde av miljøet i rommet. Jeg hadde implementert dem på samme måte som men NTC termistoren, men selvfølgelig skrevet egen kode til dem og gjort produktet mitt til en HUB hvor man kan skreddersy miljøet man vil ha hjemmet.
* Jeg hadde laget et system som gjorde at termometeret kunne styre den faktiske temperaturen i huset
* Jeg hadde implementert et system så brukeren kan manuelt innstille temperaturen de vil ha. Jeg ville ha gjort dette ved å legge til en større eller flere skjermer og brukt en kombinasjon av knapper og potensiometere for å la brukere stille inn temperaturen selv. Etter hvert ville jeg også ha gjort at brukeren kunne ha stilt inn temperaturen på forhånd eller mens de er borte.

Kilder:

1. John S. Steinhart, Stanley R. Hart, Calibration curves for thermistors, Deep-Sea Research and Oceanographic Abstracts, Volume 15, Issue 4, August 1968, Pages 497–503, ISSN 0011-7471, [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1016/0011-7471(68)90057-0](https://doi.org/10.1016%2F0011-7471%2868%2990057-0).
2. [**^**](https://en.wikipedia.org/wiki/Steinhart%E2%80%93Hart_equation#cite_ref-2) ["Memorial resolution of the faculty of the University of Wisconsin-Madison on the death of professor emeritus John S. Steinhart"](https://web.archive.org/web/20100610052139/http:/www.secfac.wisc.edu/senate/2004/0405/1775(mem_res).pdf) (PDF). University of Wisconsin. 5 April 2004. Archived from [the original](http://www.secfac.wisc.edu/senate/2004/0405/1775(mem_res).pdf) (PDF) on 10 June 2010. Retrieved 2 July 2015.
3. [**^**](https://en.wikipedia.org/wiki/Steinhart%E2%80%93Hart_equation#cite_ref-3) ["Dr. Stan Hart,"](http://www.whoi.edu/science/GG/people/shart/index.htm). Woods Hole Oceanographic Institution. Retrieved 2 July 2015.
4. Circuit Basics, (18. Nov. 2015), Arduino temperature sensor, <http://www.circuitbasics.com/arduino-thermistor-temperature-sensor-tutorial/>
5. Project Hub, (31. Mars. 2019), Digital Thermometer, <https://create.arduino.cc/projecthub/Anti_Spoofed/digital-thermometer-for-beginners-1f455d?fbclid=IwAR13k2Y7qm9D-7ieV_UqUGe7PWB4FkAhd1ul1b3084aNnbjSrEWQWPhglpM>
6. SKYLE INSTRUMENTS LTD, lest (11.02.2020), <https://www.skyeinstruments.com/wp-content/uploads/Steinhart-Hart-Eqn-for-10k-Thermistors.pdf>

1. <https://www.newbiehack.com/MicrocontrollerTutorial.aspx> [↑](#footnote-ref-1)