Termostaatti

(joka ohjaa tuuletinta)

Aku-Oskari Vilkki

Tieto- ja tietoliikennetekniikan PT, ICT-asentaja

Careeria

Sisällysluettelo

[1. Laitteen toiminta 1](#_Toc31198521)

[2. Laitteen elektroniikka 2](#_Toc31198522)

[3. Ohjelman sisältö ja rakenne 3](#_Toc31198523)

[3.1 Funktiot 3](#_Toc31198524)

[3.2 Keskiarvoistus 3](#_Toc31198525)

[3.3 Suoran yhtälö 5](#_Toc31198526)

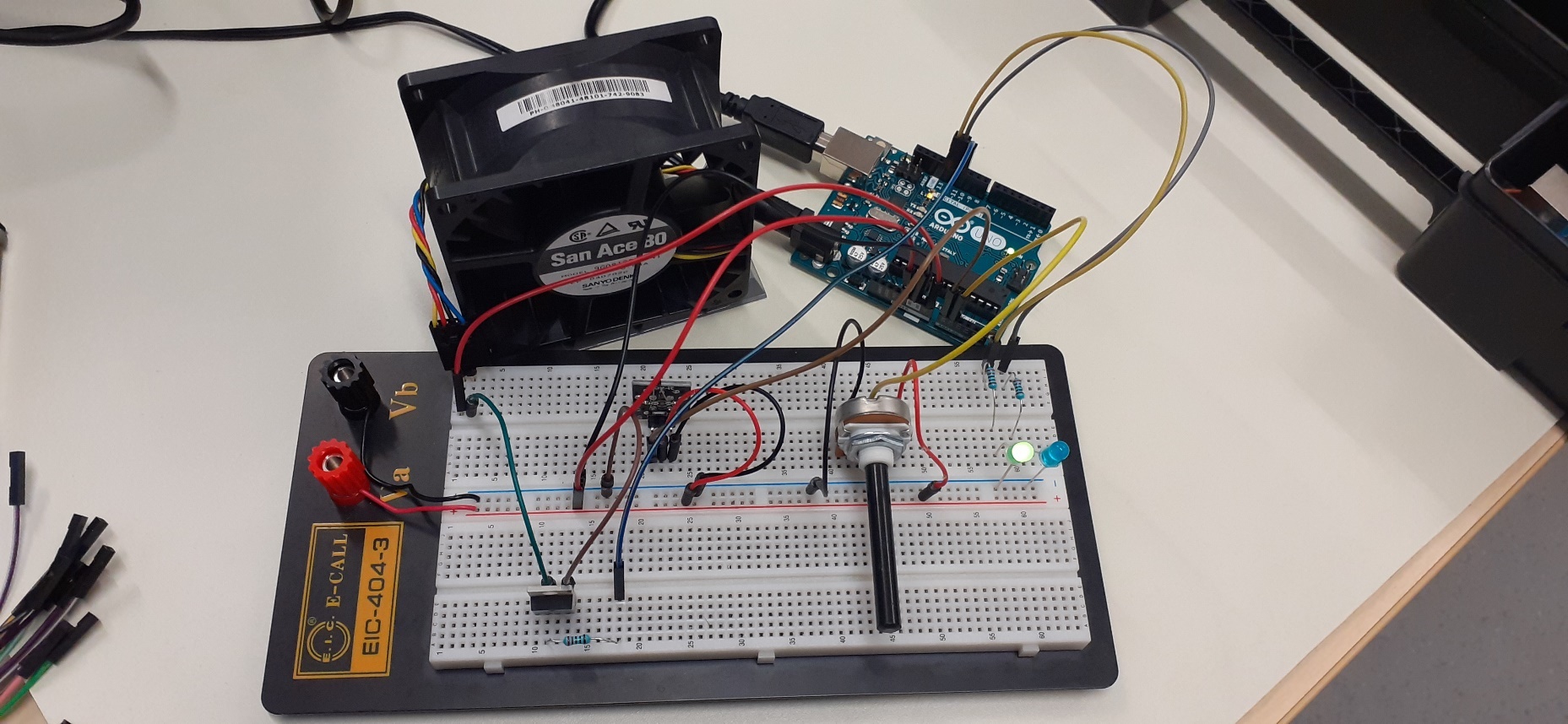
[3.4 PWM-ohjaus 6](#_Toc31198527)

[4. Oma tavoitteeni 7](#_Toc31198528)

[5. Projektikansio 8](#_Toc31198529)

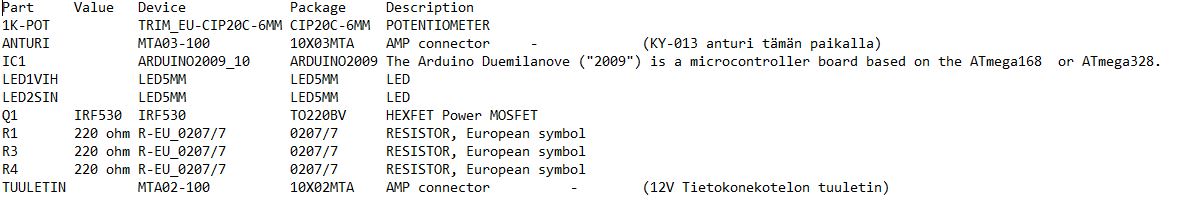
# 1. Laitteen toiminta

Laite reagoi lämpötilan nousuun käynnistämällä tuulettimen. Mitä suuremman lämpötilan nousun laite havaitsee, tai mitä suuremmaksi valitaan lämpötilan erotus potentiometrillä, sitä suuremmalla nopeudella puhallin pyörii. Laitteessa on vihreä ja sininen merkkivalo; aina kun puhallin jäähdyttää, palaa sininen ledi. Muutoin palaa vihreä ledi.



# 2. Laitteen elektroniikka

Lämpötila-anturina on käytössä KY-013. Anturissa on kolme pinniä A0, plus ja miinus. Se mittaa lämpötilan muutokset jännitteen vaihteluna, jotka arduino lukee asteikolla 0-1023. Tuulettimen asteittaista PWM-ohjaamista varten lisättiin laitteeseen MOSFET-piiri IRF530N. Tuuletin tarvitsi myös toimiakseen 12V jännitettä, jota varten lisättiin erillinen virtalähde arduinoon. Piirilevy suunniteltiin EAGLE-ohjelmalla.



Lämpötila-anturin peruskäyttöönottoon käyttämäni ohje: <https://arduinomodules.info/ky-013-analog-temperature-sensor-module/> .

Tuulettimen käyttöönottoon ja sen PWM-ohjaamiseen käyttämäni ohje: <https://www.youtube.com/watch?v=Pw1kSS_FIKk> .

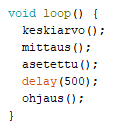
# 3. Ohjelman sisältö ja rakenne

Yksityiskohtaiset selostukset koodin toiminnoista löytyy ohjelmasta kommenteiksi lisättynä.

## 3.1 Funktiot

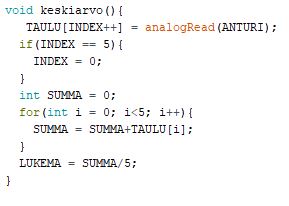
Ohjelman selkeyttämiseksi ohjelmaan lisättiin omat funktiot seuraaville toiminnoille: keskiarvo, mittaus, asetettu ja ohjaus.

* keskiarvo funktiossa luetaan lämpötila-arvo 5 kertaa ja siitä lasketaan keskiarvo lämpötilalle.
* mittaus funktiossa muunnetaan lämpötilan AD-arvo vastaaviksi celsius-asteiksi.
* asetettu funktiossa luetaan potentiometrin AD-arvo, ja mäpätään ääriasennot halutuiksi maksimiarvoiksi.
* ohjaus funktiossa lämpötilan ja asetusarvon erotus määrää sen, milloin puhallin lähtee pyörimään. Ledit syttyvät vihreästä siniseen riippuen siitä onko tarvetta jäähdyttää.



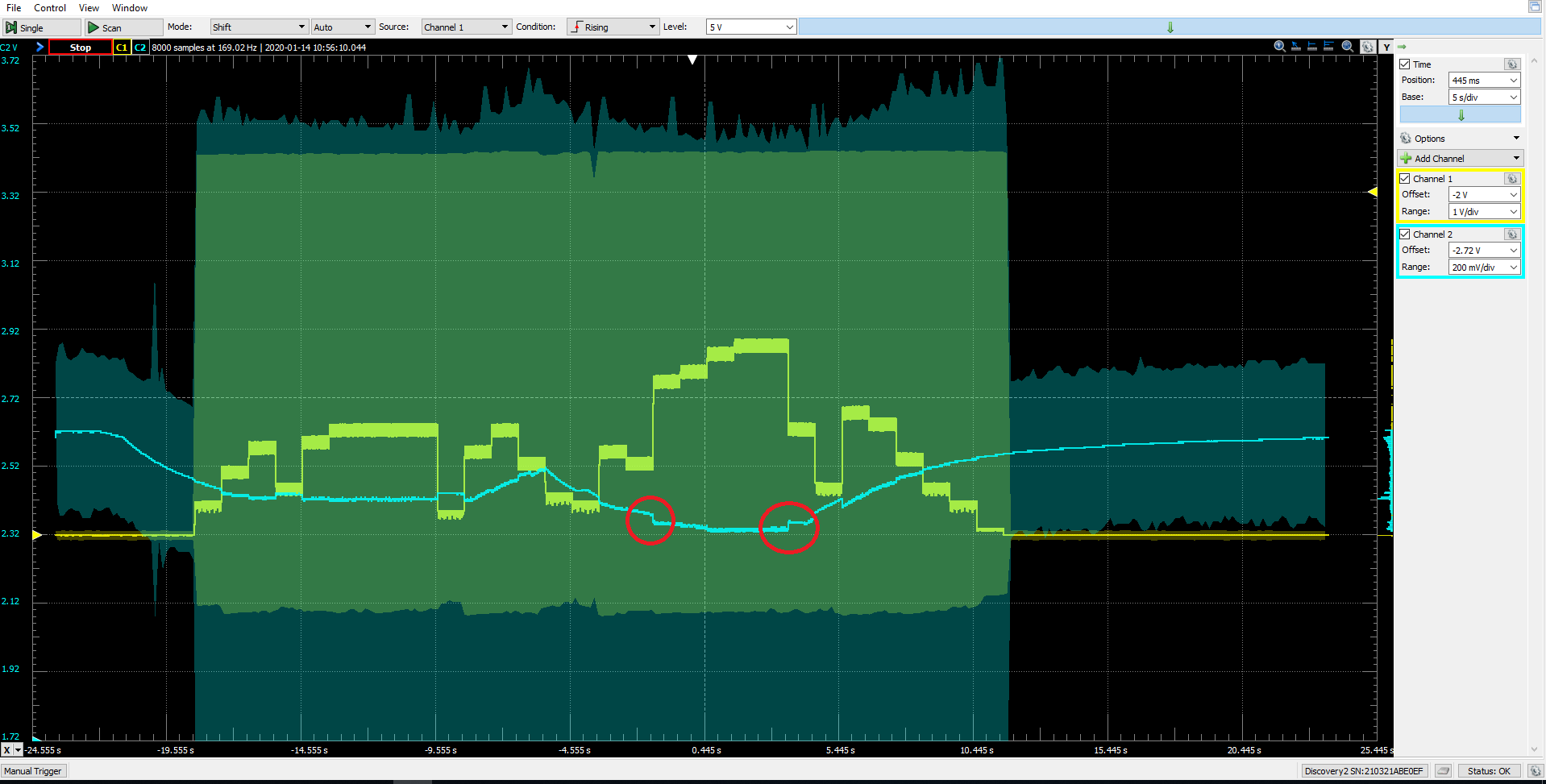
## 3.2 Keskiarvoistus

Ongelmaksi ohjelmassa muodostui anturin mittausten lukemien suuri vaihtelu, joka aiheutti laitteen epävakaata toimintaa. Toiminnan parantamiseksi lisättiin seuraava keskiarvoistus:

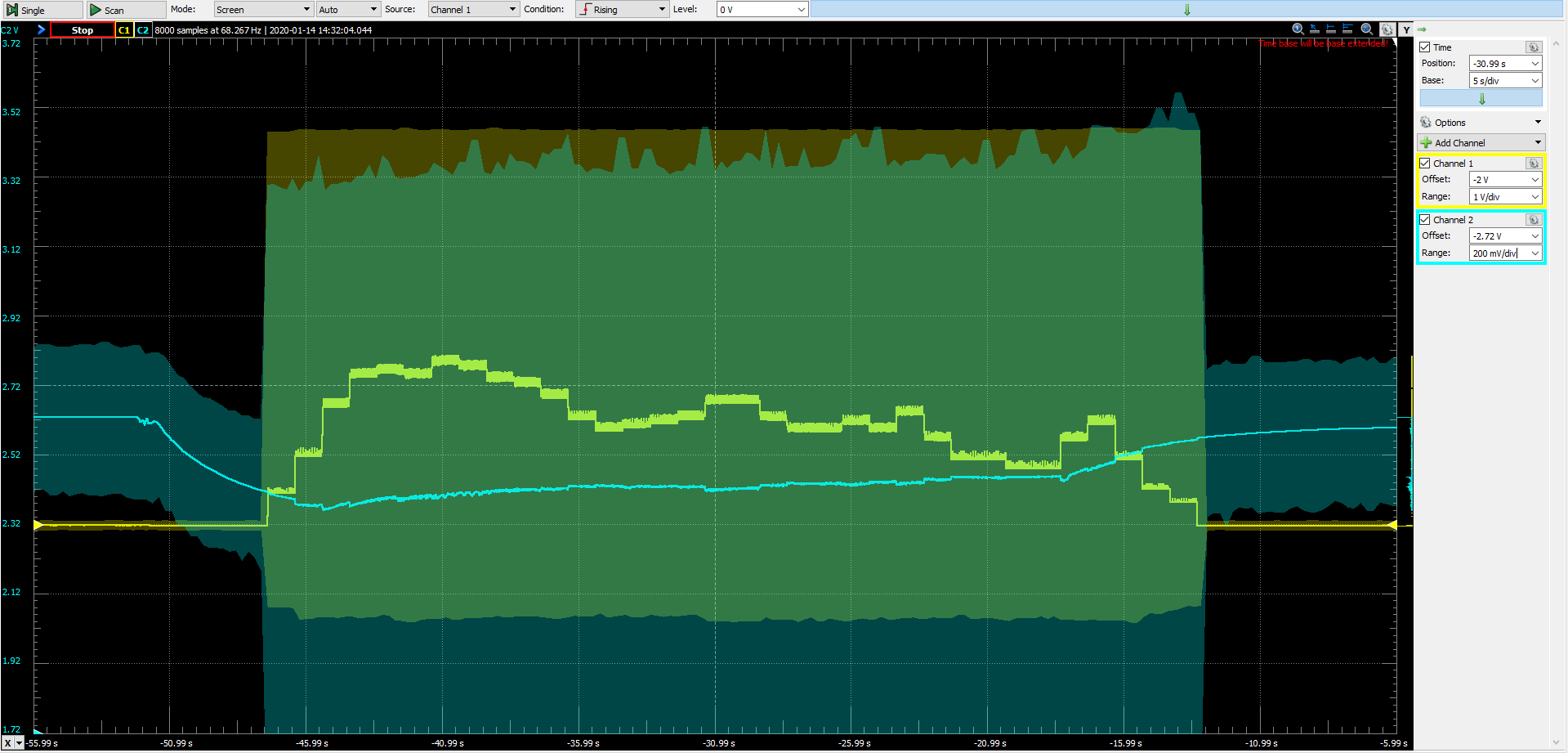


Sininen käyrä kuvaa lämpötilan nousua ja keltainen käyrä puhaltimen toimintaa. Kun lämpötila nousee - oskilloskoopin lukema vastus-arvo laskee tällöin – alkaa puhallin puhaltamaan kovempaa riippuen siitä kuinka suuri lämpötila muutos on.

Ensimmäisessä kuvassa näkyy ympyröitynä lämpötilan poukkoilua, joka näkyy myös tuulettimen PWM-ohjauksen vastaavana poukkoiluna. Puhallin vaihteli nopeuttaan jatkuvasti reagoiden pienimpiinkin lämpötilan vaihteluihin.



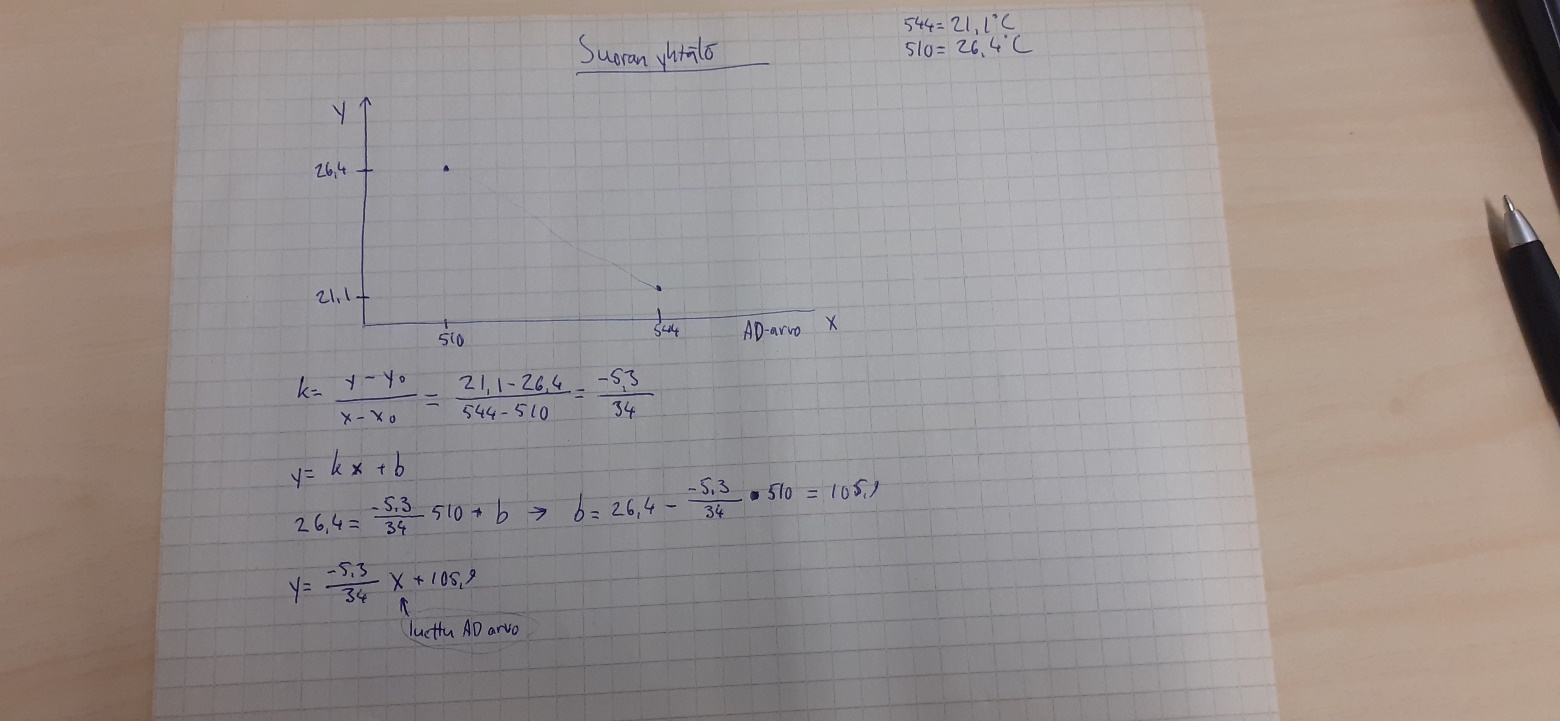
Toisessa kuvassa lisätty keskiarvoistus, jossa laitteen toiminta on selvästi rauhoittunut.



## 3.3 Suoran yhtälö

ASTE = (-5.30/34.00)\*LUKEMA+105.9;

Luettujen analogiarvojen muuttaminen celsius asteikkoon tehtiin hyödyntäen suoran yhtälöä. Mitattaessa huonelämpötila 21,1 celsius astetta, vastasi AD arvoa 544, ja kädenlämpötila 26,4 celsius astetta, vastasi AD arvoa 510. Näitä lukuja ja kuvassa näkyviä kaavoja hyödyntäen voitiin ratkaista yhtälö, joka lisättiin ohjelmaan. LUKEMA on keskiarvoistuksella saatu luku.

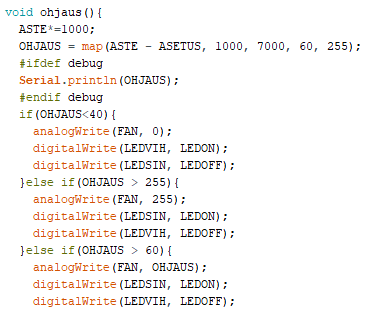


ASTE = map(LUKEMA, 510.0, 544.0, 26.4 , 21.1);

Saman ”muunnoksen” voi myös tehdä näin mäppäämällä, kunhan tietää AD arvoja vastaavat celsius asteet. Suoran yhtälöllä tarkkuus oli kuitenkin parempi, joten käytin mieluummin sitä.

## 3.4 PWM-ohjaus

Puhaltimen ohjauksesta olisi voinut tehdä myös paljon yksinkertaisemman, jossa puhallin olisi alkanut puhaltaa täydellä teholla heti kun lämpötila olisi noussut yli raja-arvon. Jotta laitteen toiminta olisi ollut monipuolisempaa ohjelmoinnin kannalta, hyödynnettiin arduino unon PWM-ohjausta. Näin puhaltimen ohjaus saatiin porrastetuksi. Anturin lukeman arvon ja potentiometrin asetusarvon erotus vaikuttaa siihen, miten nopeasti puhallin pyörii; mitä suurempi erotus sitä kovempaa puhallin pyörii. PWM-ohjaus toimii pulssimaiseen tapaan, jolloin 100%:lla (255 AD-arvo) jännitettä syötetään jatkuvana syöttönä. 50%:lla (127 AD-arvo) jännite vaihtelee symmetrisesti pulssina täyden jännitteen ja 0V:n välillä.



# 4. Oma tavoitteeni

Alkuperäinen ideani oli kehittää jonkunlainen termostaatti. Tein ennen tätä projektia vastaavanlaisen lämpötilahälyttimen, josta sain perusajatuksen tähän projektiin. Alunperin projekti alkoi siitä, että halusin kokeilla lämpötila-anturin toimintaa. Siitä projekti kehittyi, kun aloin lisäämään siihen osia vähän kerrallaan. Aiempi projekti oli melko helppo tähän verrattuna, joten halusin jatkokehittää sitä. Halusin myös oppia lisää koodin kirjoittamisesta ja tästä projektista opinkin paljon. Tein ohjelmasta käännöksen kirjastoksi, päädyin silti käyttämään vanhempaa versiota ohjelmasta, koska osaan kertoa sen toiminnasta paljon paremmin.

Ennen tätä kurssia minulla ei ollut minkäänlaista kokemusta koodaamisesta, ja kun aloitimme kurssin, koin koko aiheen melko ylivoimaiseksi. Vähän kerrallaan kuitenkin aloin hahmottamaan koodiin sisältöä, ja nyt minusta tuntuu, että tästä voi olla myös jonkunlaista hyötyä minulle. Olen alkanut hahmottamaan miten sulautettuja sovelluksia on hyödynnetty monissa elektronisissa laitteissa, aiemmin asiaa ei ole osannut ajatellakaan.

# 5. Projektikansio

<https://github.com/aov-0/termostaatti.git>

* BOM.txt
* Dokumentointi.docx
* Schematic.pdf
* termostaatti.ino