SULAUTETUT JÄRJESTELMÄT

NÄYTTÖ PROJEKTI

Roope Saario

[Kuvaus laitteesta 1](#_Toc61865334)

[Elektroniikka 1](#_Toc61865335)

[Ohjelman kuvaus 2](#_Toc61865336)

[Lämpötila 2](#_Toc61865337)

[Lcd näyttö 2](#_Toc61865338)

[Ajastus 3](#_Toc61865339)

[Projektin valitseminen, työskentely ja linkit 3](#_Toc61865340)

[Projekti 3](#_Toc61865341)

[Kirjastot ja linkit 4](#_Toc61865342)

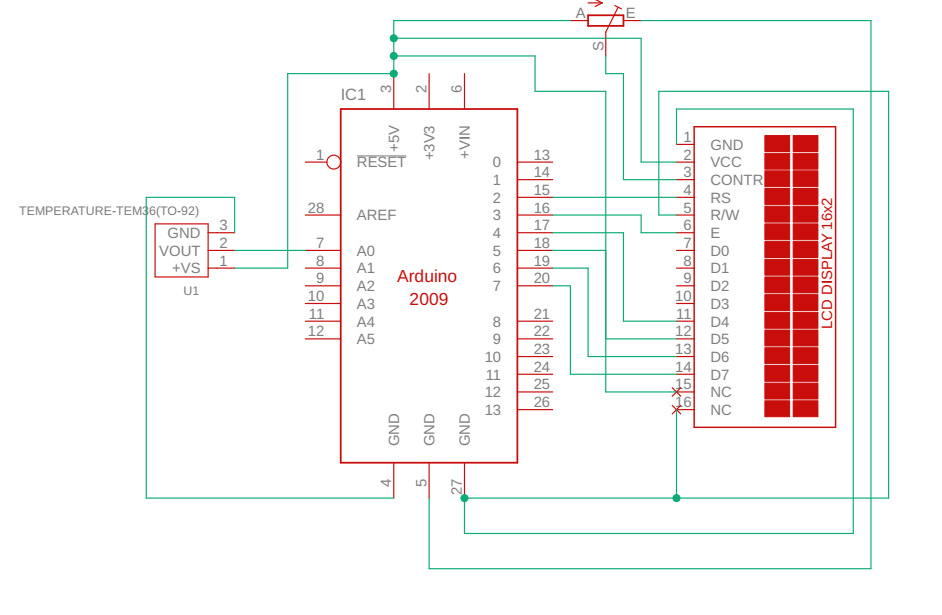
[Projekti kansio 4](#_Toc61865343)

# Kuvaus laitteesta

Laite on rakennettu mittaamaan huoneen tai vaikkapa vauvanrattaitten sisällä olevan lämpötilan. Lämpötila ilmoitetaan 16x2 Lcd näytöllä luettavaksi ja lämpötilan saamiseksi hyödynnetään TMP36 sensoria. Potentiometrillä on mahdollista säätää näytön sektoreitten ledin tulostus kirkkautta valoisuuteen sopivammaksi.

# Elektroniikka

Alla Autodeskin Eagle ohjelmalla piirretty schema laitteen kytkennöistä.



# Ohjelman kuvaus

Lämpötila Minulla ei ollut aikaisempaa kokemusta kyseisestä tmp36 sensorista ja jouduin ensitöiksi tutustumaan sen toimintaan. Lämpötilojen saamiseksi löytyy monia eri ratkaisuja osa tarkempia ja osa ei niinkään käyttökelpoisia.   
  
Saadakseen sensorin ilmoittamaan lämpötiloja liitin sen A0 porttiin ja 5V virtalähtöön. Ensiksi lämpötilan saamiseksi piti muuttaa analogisen portin

0 - 1023 lukema väli millivolteiksi, jonka jälkeen piti millivoltit muuttaa lämpötilaksi.  
  
Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti  
  
Kyseinen arvo silti vielä heittelee hieman ja tarkemman lukeman saamiseksi Janin ehdotuksesta päätin laskea keskiarvon mitatuille arvoille. Löysin Matt Fryerin tekemän Smoothed arduino kirjaston tätä tehtävää varten ja kyseinen kirjasto oli melko yksinkertainen ottaa käyttöön esimerkki mallin kautta.  
  
Muuttujat keskiarvon laskua varten  
Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti  
Setup lisäykset  
  
Lämpötila funktion lisäykset

Lcd näyttö Käytössämme olleen 16x2 Lcd näytön ohjauksessa on käytetty yleistä Liquidcrystal kirjastoa. Näytön käyttöönotto onkin melko suoraviivainen tehtävä kirjaston avulla, kunhan muistaa lukea ohjeistusta hieman ja on tarkkana kytkentöjen kanssa. Suosittelen luomaan vaihtoehtoisen sanan kirjaston nimelle käytettäväksi koodiin selkeyden ja kirjoittamisen helpottamisen lisäämiseksi. Ääkkösiä ja erikoismerkkejä varten löytyy taulukkoja tai apua voi niitten tulostamiseksi näytölle voi hakea Arduinon forumilta, josta löytyykin melkein asiaan kuin asiaan vinkkiä.

Ajastus Ilman minkäänlaista ajastusta näyttö päivittyisi koko aika hallitsemattomaksi ja saamatta selvää tulostetuista arvoista. Tätä varten loin oman ajastin kirjaston millis funktion avulla. Sen myötä on helppo ajastaa moniakin toimintoja halutessaan tapahtumaan saman aikaisesti ja muokkaamaan niiden suoritus väliä haluamakseen.   
  
Ylimääräisen näytön päivittämisen karsimiseksi ja näytön toimivuuden parantamiseksi loin vielä yhden lisä funktion. Kyseinen funktio vertaa uuden lasketun lämpötilan keskiarvon edelliseen laskettuun arvoon ja mikäli arvo ei ole muuttunut ei näyttöä turhaan päivitetä.  
  
Vertailu funktion sisältö  
Kuva, joka sisältää kohteen teksti

Kuvaus luotu automaattisesti

# Projektin valitseminen, työskentely ja linkit

Projekti Projektia valittaessa halusin työskennellä ja tehdä laitteen, jolle olisi olemassa todellinen käyttötarkoitus. Jokin toinen projekti olisi varmasti mahdollistanut monipuolisemman koodin luomisen ja muokkaamisen, mutta valitsemani aihe ja komponentit antoivat mielestäni riittävän laajan alueen osoittamaan osaamiseni aiheen parissa.  
  
Jani on muistuttanut pitkin kurssia yksinkertaisuuden kauneudesta ja muistutellut ettei yksinkertaista asiaa kannata tehdä liian vaikeasti. Monesti harjoitus tehtävissä porukassa ajatus karkasi liian pitkälle tehtävän lopulliseen tulokseen. Mikä aiheutti ongelmia saada koodia toimivaksi kokonaisuudeksi. Nyt yksin työskennellessä oli mukava rauhassa edetä askel askeleelta eteenpäin ja luoda/rakentaa yksi osa kerrallaan koodiin. Toimivan kokonaisuuden saaminen helpottui mielestäni tämän myötä. Janilta sai hyviä väli kommentteja, miten mittariani voisi kehittää todellista käyttöä varten ja loppujen lopuksi mietinkin, miten itse haluaisin lämpömittarin toimivan. Olisi mielenkiintoista päästä lisäämään mittariin vielä äänimerkki ilmoitus ja/tai lähetys toiminto vastaanottimelle raja-arvojen ylityksistä.

Kirjastot ja linkit Arduino ohjelmassa löytyy kattava kirjo valmiita kirjastoja käytettäväksi ja niitä on hyvin yksinkertaista lisätä käyttöön sitä kautta. Moni asia helpottuu tätä kautta ja eihän pyörää kannata uudestaan keksiä, jos on mahdollista hyöty käyttää olemassa olevaa. Ihan kaikkia sieltä ei kuitenkaan löydä ja keskiarvon laskemiseen hainkin Githubista Matt Fryerin luoman Smooted kirjaston. Näitten lisääminen onkin hieman erilaista koodiisi ja kun ohjelmasta löytyvän kirjaston voi lisätä muutamalla klikillä sketsi->sisällytä kirjasto polun kautta. Niin esimerkiksi githubista ladattu kirjasto pitää ensin siirtää arduino kansion alla olevaan libraries kansioon, jotta saadaan se käyttöön ohjelmaan. Yritin käyttää tmp36 sensoriin myös valmista kirjastoa, mutta sillä saadut tulokset heittelivät yli 20 asteella ja lähdin etsimään vaihtoehtoista tapaa mittarin lukemiseen.  
  
<https://github.com/MattFryer/Smoothed>  
<https://www.makerguides.com/tmp36-arduino-tutorial/>

Projekti kansio <https://github.com/rsaario/Lampomittari>