



Aalto-yliopisto
Sähkötekniikan
korkeakoulu

ELEC-C5070 – Elektroniikkapaja

Loppuraportti

Optinen sykemittari



Syksy 2018 - Ryhmä numero 18

Santeri Suitiala
santeri.suitiala@aalto.fi

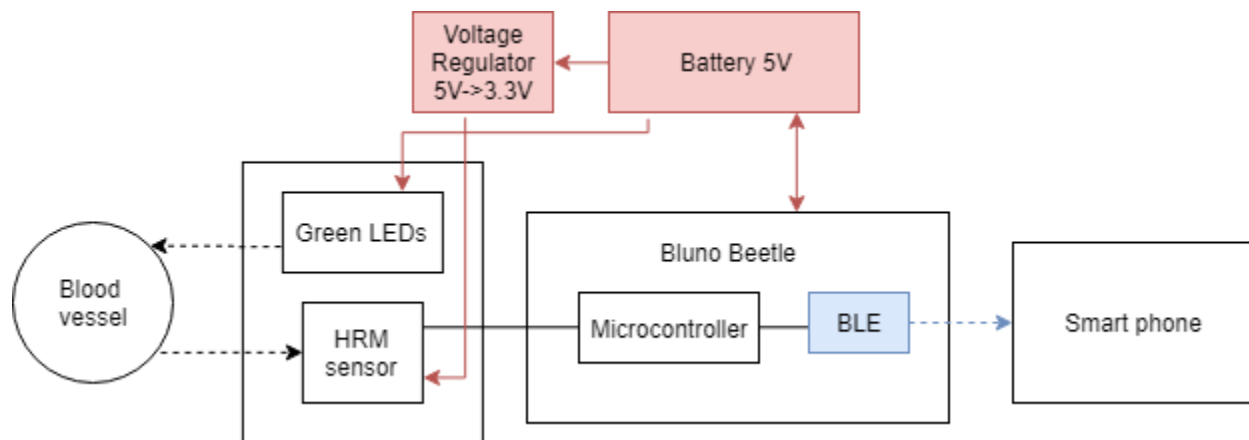
Sergei Kaukiainen
sergei.kaukiainen@aalto.fi

Aapo Linjama
aapo.linjama@aalto.fi

JOHDANTO

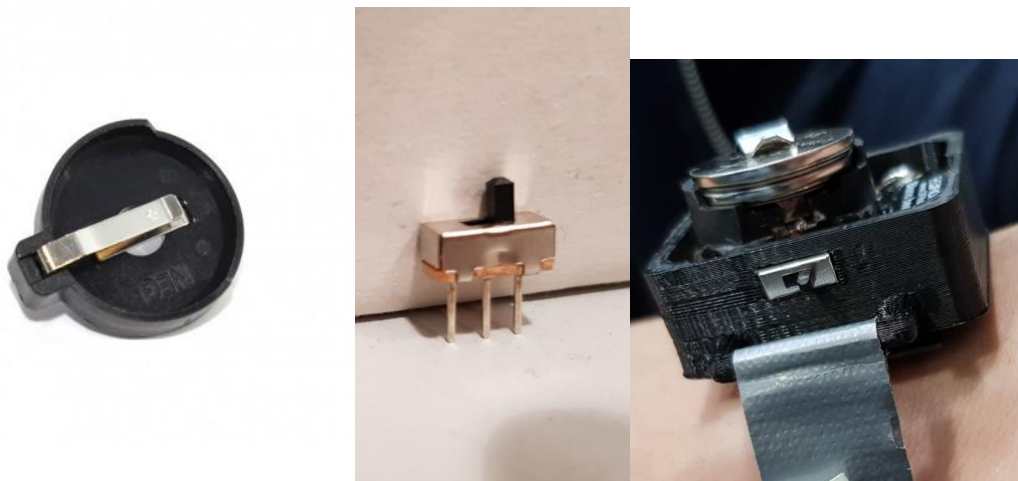
Ryhmämme tavoitteena on rakentaa mahdollisimman pienikokoinen optinen sykemittari, joka mittaa sykettä ranteesta. Lisäksi haluamme laitteen toimivan langattomasti jolloin syke tulisi näkymään bluetooth:in avulla esimerkiksi älypuhelimien ruudulla. Laite toimii akulla tai paristolla jolloin toteutus on kokonaan langaton ja laitetta voidaan pitää ranteessa mittaamassa sykettä esimerkiksi urheiluvoimistelun aikana.

1 TOTEUTETTU LAITTEISTO



Kuva 1: Sykemittarin komponentit ja niiden vuorovaikutukset

Kuva 1 esittää sykemittarin toimintaperiaatteen. Punaiset osat kuljettavat tehoa, musta viiva informaatiota, musta katkoviiva valoa ja sininen katkoviiva bluetooth-signaalia. Vasen ympyrä on verisuoni josta syke mitataan. Vasen suorakulmio jossa LEDit ja sensori on on sykemittarimoduuli (Kuva 5). Bluno beetle (Kuva 4) on mikrokontrolleripiiri jossa on sisäänrakennettu BLE-moduuli. Oikealla on älypuhelin johon on ladattu BlunoBasicDemo android-applikaatio mikä tunnistaa blunon bluetoothin välityksellä automaattisesti. Ylhäällä on esitetty patteripidike josta tulee 5V ja erillinen jänniteregulaattori jota tarvittiin jännitteen heikentämisessä sensorille.



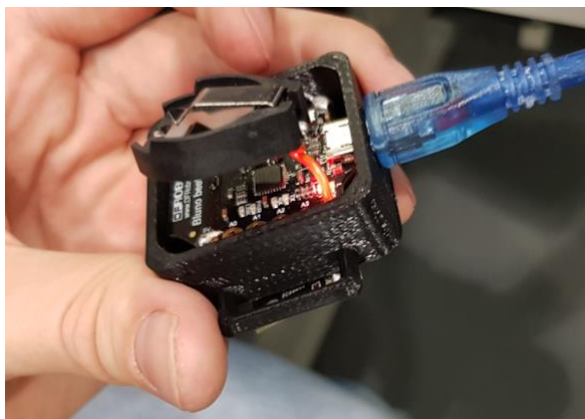
Kuva 2: Paristopidike ja kytkin

Tehonsyöttö

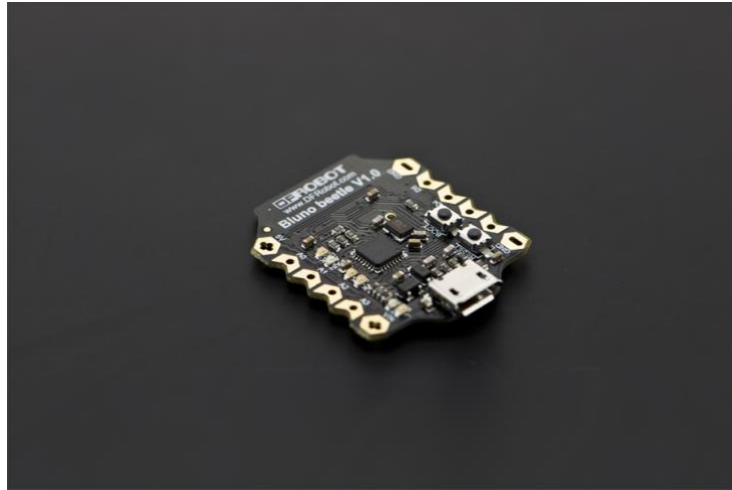
Sykemittarimme saa virtansa kahdesta nappiparistosta. Nämä paristot on kytketty sarjaan paristopidikkeen⁶ avulla (kuva 2, vasemmalla), joka on kytketty kytkimen kautta mikro-ohjaimeen. Käytetyt paristot antavat molemmat 3V eli mikro-ohjaimen tulojännitteeksi saadaan 6V. Työssä puhutaan viidestä voltista mutta todellisuudessa paristot antavat 6V. Yhden voltin ero ei kuitenkaan aiheuta toiminnallisia eroja. Pariston ja muiden laitteiden välillä on kytkin jotta laite ei jatkuvasti kuluttaisi paristoja. Kytkin (kuva 2, keskellä) on helposti käytettävissä laitteen kotelon reunassa (kuva 2, oikealla).

Ulkoisen jänniteregulaattori pudottaa nappiparistoilta tulevan 5V tulojännitteen 3.3V:iin. Se jaetaan sellaisenaan LEDeille ja regulaattorin³ (kuva 3, oikealla) kautta sensorille. Sensorin käyttöjännite on 3.3V ja mikrokontrolleripiirille menee suoraan 5V. Blunolle on myös mahdollista antaa jännite micro-usb kautta. Regulaattori on juotettu irrallisena osana litteämmän toteutuksen mahdollistamiseksi. Regulaattori tulisi kuitenkin integroida osaksi mikrokontrolleripiiriä mikäli tuotteesta toteutettaisiin täydellisempi versio.

Micro-USB sisääntulo on huomioitu koteloinnissa ja USB voidaan syöttää suoraan kotelossa olevan reiän läpi (Kuva 3, vasemmalla).



Kuva 3: Micro-usb sisääntulo ja regulaattori



Kuva 4: Bluno Beetle -mikrokontrolleripiiri

Mikrokontrolleripiiri

Sykemittaria ohjaa arduino pohjaisesta Bluno Beetle mikro-ohjain², (Kuva 4) johon on liitetty optinen sykesensori¹. Mikro-ohjain on sopivan pieni ja sopii hyvin käyttötarkoitukseemme. Bluno:ssa on valmis I2C-väylän tuki jonka kautta sensori kommunikoi. Saimme sensorin helposti yhteyteen Blunon kanssa juottamalla molempien SCL ja SDA väylät yhteen. Blunolla on myös valmis 5V sisääntulo ja maa jolloin käyttäjännitettä on helppoa syöttää suoraan paristoilta. Käytetyn koodin voi löytää sykemittarin sivuilta¹ kohdasta "HR Calculation Algorithm". Mikrokontrolleriin ladattu ohjelmisto on ZIP-paketista löytyvä HeartRate.ino. Se tarvitsee toimiakseen muita ZIP-paketissa olevia kirjastoja ja arduinon valmiita kirjastoja.

BLE

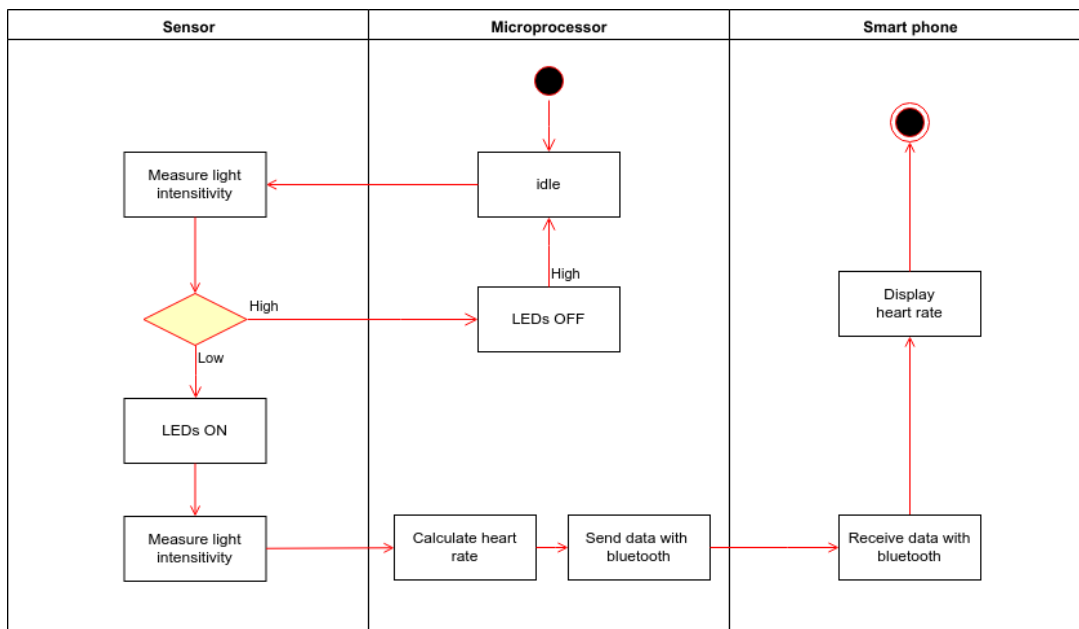
Beetlen mikrokontrolleripiirissä on myös valmiiksi BLE-moduuli (Bluetooth Low Energy), jonka avulla lähetämme käsitellyn datan, eli lasketun sykkeen, puhelimeen. BLE on huomattavasti energiatehokkaampi ratkaisu kuin perinteinen bluetooth mutta on huono tilanteissa, joissa laitteen pitää lähettää paljon dataa. Meidän luoma sykemittarimme lähettää kuitenkin vain yhden viestin sekunnissa joten BLE on enemmän kuin tarpeeksi ja lisäksi se säästää akkua langattomassa toteutuksessa.



Kuva 5: BH1790GLC sykesensori

Anturi

Anturiksi valikoitui ROHM semiconductor BH1790GLC sykesensorimoduuli (Kuva 5). Valinnan pääsyyt olivat suhteellisen halpa hinta, helppokäyttöisyys ja valmis lähdekoodi. ROHM:in anturimoduuli on suhteellisen helppokäyttöinen. I2C-väylä ei ollut entuudestaan tuttu käsite mutta onneksi arduinopohjainen wire-kirjasto hoitaa väylän toiminnalla ongelmitta ja lisäksi ROHM:in valmis lähdekoodi hoiti sensorin mittaukset ilman sen kummempia huolia.



Kuva 6: Sykemittarin prosessilohkokaavio

Toimintaperiaate

Kuvan 6 prosessilohkokaavio esittää sykemittarin perimmäisen toimintaperiaatteen. Mikäli "display heart rate"-lohkoon ei päästä, muistiin kirjoitetaan luku 0 ja se vaikuttaa keskimääräiseen sykkeeseen. Mikroprosessorin mittaustaajuus on 32Hz joten idle eli tyhjäkäyntiaika on noin 31ms. Mikäli lohkoavio suoritetaan loppuun, se alkaa taas alusta olettaen että laite on yhä päällä.

Laite siis mittaa valon intensiteettiä 32Hz taajuudella. Jos valo on liian kirkasta, laite olettaa että laite ei ole käytössä vaan esimerkiksi pöydällä eikä se silloin käsittele dataa. Jos sensori on tarpeeksi himmeässä ympäristössä, esimerkiksi kiinni käyttäjän ranteessa, se alkaa lähettämään ledeillä vihreää valoa ja mittaa vihreän ledin vastetta. 32Hz mittauksesta saadaan hyvin selville korkeakin pulssi sillä Nyquistin rajataajuuden määritelmällä $32\text{Hz} \cdot 60\text{s} / 2 = 960\text{bpm}$. 960bpm on siis suurin syke mitä

mittarillamme voidaan teoriassa mitata. Ihmisen maksimisyke on noin 200bpm joten laitteemme riittää aivan hyvin mittaamaan kaikenlaiset sykkeet.

Mikroprosessori laskee raakadatasta sykkeen arduino pohjaisella algoritmilla. Algoritmi toimii seuraavalla tavalla:

- Selvitä raakadatasta huippukohdat
- Laske huippukohtien taajuus
- Pidä kirjaa viimeisimpien huippukohtien taajuuksista ja laske keskiarvo tietyllä aikavälillä
- Palauta taajuuskeskiarvo kerrottuna 60 sekunnilla jolloin tulos on bpm (beats per minute)

Tämän jälkeen data lähetetään BLE:n välityksellä puhelimelle ja puhelin voi tehdä tiedolla mitä haluaa. Tässä työssä käytimme android-aplikaatiota, joka yksinkertaisesti näyttää tulevan datan ruudulla.

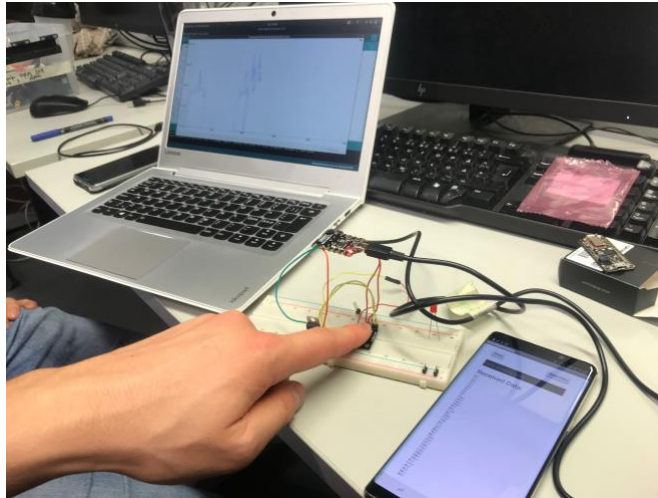
2 MITTAUKSET JA TESTAUS

Aluksi meillä ei ollut tarkkaa tietoa, miten tilaamamme komponentit toimivat. Aloitimme sykesensorin testaamisesta.

Sykesensorin toimivuuden todentaminen

Testasimme aluksi hyvin yksinkertaisella alustalla pelkästään sensorin toimintaa ja sen tulosten oikeellisuutta. Testi toteutettiin leipälaudalla ja tekijät, jotka mahdollisesti voisivat luoda häiriöitä, minimoitiin. Kuva 7 on otettu testistämme. Testissä haluttiin ainoastaan tietää toimiiko sensori vai ei. Tuskailun ja ihmettelyn jälkeen saimme graafit arduino IDE:n näytölle. Tämän jälkeen

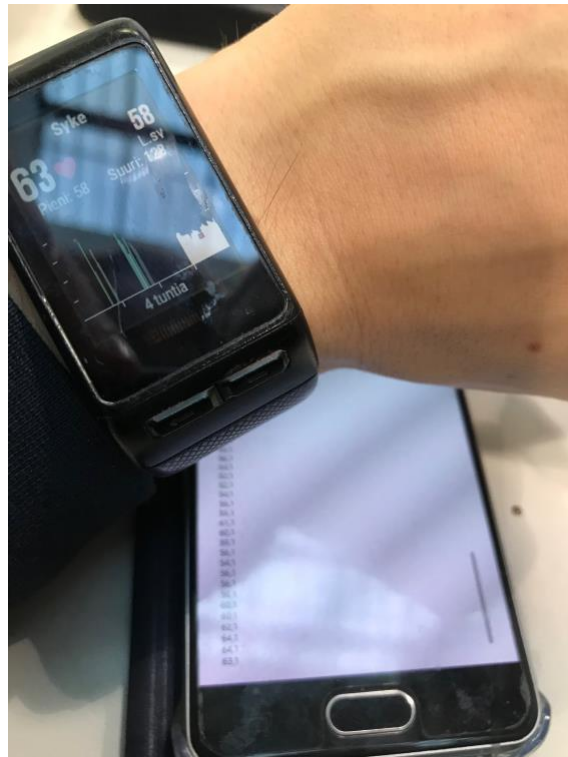
saimme samalla todennettua BLE:n toimivuuden lähettämällä datan eteenpäin puhelimelle mikä näkyy kuvassa 7. Sensori ja BLE todettiin siis toimiviksi heti projektin alkuvaiheella.



Kuva 7: Sykesensorin toimivuuden todentaminen

Toiminnallisuustestit

Molemmissa testeissä Garmin-aktiivisuusranneke oli referenssinä. Delta kertoo taulukoissa sen kuinka paljon positiiviseen tai negatiiviseen suuntaan testeissä oli eroa referenssitason. Mittauslaitteistona käytettiin älypuhelinta, johon oli ladattu BlunoBasicDemo -ohjelma. Ohjelman avulla syke saatiin näkyviin puhelimen näytöllä. Bluno-mikrokontrolleripiiri siis yhdistettiin BLE:llä puhelimeen BlunoBasicDemoon.



Kuva 8: Referenssitaso (Garmin) ja laitteemme tuottama ulostulo näytöllä

TESTI 1:

Kaikki testit olivat 30 sekunnin pituisia.

Kaikki arvot kirjattiin viiden beats per minute (bpm) tarkkuudella.

Testikerta	Max delta neg. (bpm)	Max delta pos. (bpm)	Arvio oikeellisuudesta	Kommentteja
1.	10	80	Ei järkevä	Heittelyä paljon
2.	3	45	Puolet arvoista ok	Tasaantui loppua kohden
3.	10	10	Erittäin hyvä	Odotimme tasaantumista

TESTI 2:

Sensori suojattiin liinalla paremmin taustavalolta.

Kaikki testit olivat 30 sekunnin pituisia.

Kuten testi 1:ssä arvot kirjattiin viiden tarkkuudella.

Testikerta	Max delta neg. (bpm)	Max delta pos. (bpm)	Arvio oikeellisuudesta	Kommentteja
1.	3	40	ok	Tasaantui lähelle referenssiä
2.	5	10	Erittäin hyvä	Ei suurta heittoa referenssiin
3.	5	5	Erittäin hyvä	Ei suurta heittoa referenssiin

Johtopäätöksiä testeistä:

- Vaikuttaa siltä että algoritmi toimii kuten pitääkin, algoritmi siis poimii ja parsii sensorin tuottamasta datasta sykkeen
- Peittämällä sensori liinalla, saatiin häiriöitä ja eroa referenssiin pienennettyä
- Molemmissa testitilanteissa päästiin järkeviin tuloksiin
- Kun dataa alkaa saapua bluethoothin välityksellä esim. puhelimeen, täytyy odottaa hetki että ero referenssiin alkaa tasaantua, tämä johtuu algoritmin laskutavasta, joka perustuu useamman mittauksen keskiarvoon

- Hyviin tuloksiin päästään siis odottamalla, että sensori on tuottanut useamman mittaustuloksen

3 JOHTOPÄÄTÖKSET

Kun vertailee osapalautus 2:ssa asettamiamme vaatimuksia, onnistuimme saavuttamaan kaikki tavoitteet. Päävaatimuksemme olivat mm. sykemittarin koko, tehonsyöttö, datan ulostulo sekä budjetti.

Pienen mikrokontrollerin ansiosta saimme tulostettua 3D-tulostimella sykemittarille kätevän kotelon, jonka sai kiinnitettyä hihnalla ranteeseen. Tehonsyöttöön valitsimme paristot ja näin saimme sykemittarin ranteeseen, jonka kanssa pystyy liikkumaan vapaasti. Kotelolla saimme myös estettyä liian valon mittarille, jolloin mittaaminen onnistuu tarkasti valoisassakin huoneessa.

Mikro-ohjaimen BLE-yhteyden avulla onnistuimme vaivatta lähettämään datan puhelimeen ja näin myös sen osalta luopumaan ylimääräisestä datajohdosta. Jos ei lasketa sykemittarin saamista toimivaksi, koko paketin saaminen ranteeseen ilman ylimääräisiä johtoja oli varmasti meidän tärkein tavoite työssä. Annetun budjetin ja ajan raameissa työhön olisi tuskin saanut enää tätä monimutkaisempia ominaisuuksia.

4 LÄHTEET

1. ROHM semiconductor, *Optical Sensor for Heart Rate Monitor - BH1790GLC*,
<https://www.rohm.com/products/sensors-mems/optical-sensor-for-heart-rate-monitor/bh1790glc-product>
2. DFRobot, Beetle BLE - The smallest Arduino bluetooth 4.0 (BLE),
<https://www.dfrobot.com/product-1259.html>
3. L78L33ACZ regulaattori,
<https://www.mouser.fi/ProductDetail/STMicroelectronics/L78L33ACZ?qs=sGAEpiMZZMtUqDgmOWBjgLAmlD5B%2FjmHpfabeMkxbt0=>
4. ROHM semiconductor, *BH1790GLC-EVK-001 Manual*,
http://rohmfs.rohm.com/en/products/databook/applinote/ic/sensor/optical_sensor_for_heart_rate_monitor/bh1790glc-evk-001-manual-e.pdf
5. DFRobot, Bluno Basic Demo,
https://www.dfrobot.com/wiki/index.php/Bluno_SKU:DFR0267#Bluno_Basic_Demo
6. Coin cell battery holder, <https://www.mouser.fi/ProductDetail/Keystone-Electronics/1061TR?qs=sGAEpiMZZMtT9MhkajLHrnU1d13jcSgSc6jJguRFDPO%3d>

LIITE 1: TYÖNJAKO JA MITÄ OPITTIIN

Näin kirjoitimme ensimmäiseen osapalautukseen:

“Ryhmäläisten elektroniikka osaaminen on samalla tasolla eli alkeissa. Tavoitteena on oppia elektroniikkasuunnittelun perusteita: protopiirin teko, tarvittavien osien tilaaminen järkevään hintaan, mahdollinen valmiin piirin teettäminen, mikrokontrollerin tms. ohjelmointi, tuotteen koteloointi sekä verifiointi. Eli tavoitteenamme on oppia elektroniikan tuotesuunnittelu kokonaisuutena.”

Kaikki tavoitteet saavutettiin ja uskomme että tunnemme tuotesuunnittelun käytäntöjä ja yleisiä ongelmia hieman paremmin kuin kurssin alussa. Protopiiri toteutettiin leipälaudalle jolla todettiin tuotteen idean toimivuus. Tarvittavat osat tilattiin ja yksi täyttilaus piti suorittaa jotta kaikki sopivat osat olivat kasassa. Valitsimme osat koon ja hinnan mukaan jolloin tuote on samalla halpa sekä pieni. Tuotteemme on niin sanottu valmis piiri joka on koteloitu 3D-tulostetulla kotelolla jossa on kotelo ja kansi. Projektimme ei itsessään ole lopullinen tuote vaan prototyyppi. Jos tuote aiottaisiin oikeasti toteuttaa, siihen todennäköisesti suunniteltaisiin yksi integroitu piiri ja parempi kotelo ohuemmasta muovista. Verifiointi tapahtui ennalta suunnitelluilla testeillä jotka on kuvailtu kohdassa “mittaukset ja testaus”.

Ryhmäläisten oppimaa:

- I2C-väylä ja sen käyttäminen
- 3D-suunnittelu tulostamista varten
- Yleistä tuotesuunnittelua
- Tuotteen verifiointi
- Piirien yhdistäminen isommaksi kokonaisuudeksi
- Erilaisten komponenttejen toiminnallinen ja hinnallinen vertailu
- Elektroniikan manuaalien tulkitseminen

Työnjako

Kokoonnuimme työstämään projektiamme kerran viikossa jolloin töitä ei oltu etukäteen selkeästi jaettu vaan ne määräytyivät tiimiläisten halujen mukaan. Pääpiirteittäin kuitenkin työtaakka jakautui seuraavalla tavalla:

Santeri:

Ohjelmiston ja I2C-väylän ymmärtäminen ja käyttäminen, 3D-suunnittelu ja koteloointi

Sergei:

Komponenttien tilaaminen, kolvaaminen ja mikrokontrolleripiirien vertaileminen

Aapo:

Kolvaaminen ja komponenttivalintoihin tutustuminen

Kukaan ei kuitenkaan tehnyt mitään täysin yksin vaan ryhmä oli joka kerta kokoontunut ja ryhmäläiset auttoivat ja tukivat toisiaan koko projektin ajana.

Lisäksi työssä oli töitä johon osallistuivat tasaisesti kaikki ryhmän jäsenet.

Näitä olivat mm.:

Kokonaisuuden suunnittelu, komponenttien valitseminen, tuotteen testaaminen, testipiirien kokoaminen, osapalautusten ja raportin kirjoittaminen

Vertaus sähköpajaan

Verrattuna aikaisempaan Sähköpaja-kurssiin työskentely oli nyt tehokkaampaa ja osasimme paremmin arvioida kuinka haastavaksi työ kannattaa suunnitella ominaisuuksien ja työmäärän osalta. Aloitimme myös työskentelyn tehokkaasti heti alusta lähtien, joten kiirettä ei missään vaiheessa tullut kurssin aikana toisin kuin Sähköpaja-kurssilla. Yleensä tällaisissa projekteissa tulee aina jotain komplikaatioita vastaan, joten työskentelyn aloittaminen ajoissa on tärkeää. Jos työtä olisi halunnut hioa vielä pidemmälle esimerkiksi pienempään kokoon, olisi kannattanut suunnitella ja toteuttaa useampia komponentteja itse, mutta se olisi taas kasvattanut merkittävästi virheiden ja komplikaatioiden todennäköisyyttä.