



LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

TULOSTAULU

Case: Lahden Mailaveikot

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Tietokone-elektronikka
Opinnäytetyö
Kevät 2015
Erkka Alamäki

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

ALAMÄKI, ERKKA:

Tulostaulu
Case: Lahden Mailaveikot

Tietokone-elektronikan opinnäytetyö, 40 sivua, 17 liitesivua

Kevät 2015

TIIVISTELMÄ

Opinnäytetyö käsittelee pesäpallon tulostauluprototyypin suunnittelua ja toteutusta. Työn toimeksiantajana oli pesäpalloseura Lahden Mailaveikot. Laite toteutettiin, jotta pesäpallopelien seuraaminen Lahden kisapuistossa helpottuisi katsojille.

Työn alkuun kartoitettiin, millainen tulostaulun tulisi olla. Spesifikaatioiden karttuessa ja työn yleisilmeen selvetessä alkoi elektroniikkakomponenttien valitseminen ja ulkoasun yksityiskohtaisempi suunnittelu. Työn käytännön osuudessa laitetta alettiin suunnitella ja toteuttaa osissa, jotka työn valmistuessa kasattiin toimivaksi kokonaisuudeksi. Laitteen keskeisimpinä osuuksina toimii Raspberry-pi-mikrotietokone ja opiskelijan suunnittelema piirilevy. Näiden osuuksien avulla ajetaan 20-tuumaisia seven-segment-näyttöjä, joita käytetään tulostaulun visuaalisen informaation esittämiseen.

Työn lopputuloksena saatiin suunniteltua ja toteutettua spesifikaatioiden mukainen prototyyppi. Laite on suunniteltu toimitettavaksi tilaajalle kesäksi 2015, joten siihen on talven 2014–2015 aikana mahdollisuus toteuttaa laajennuksia tai parannuksia.

Asiasanat: Raspberry-pi, elektroniikka, mikrokontrolleri, prototyyppi

Lahti University of Applied Sciences
Degree Programme in Information Technology

ALAMÄKI, ERKKA:

Scoreboard

Case: Lahden Mailaveikot

Bachelor's Thesis in Computer Electronics, 40 pages, 17 pages of appendices

Spring 2015

ABSTRACT

This Bachelor's thesis deals with the designing and building of a baseball scoreboard. The device was created for baseball club Lahden Mailaveikot. The device was built, so that watching baseball games in the Lahti sport park would be more enjoyable for the spectators.

The work started by examining what the scoreboard should be like. When the specifications started to accumulate and the overall picture started to be clearer, the choosing of the components and a more detailed layout plan could start. In the practical part of the thesis, the device was divided into sections, which were put then together as one unit. The main units of the device are a Raspberry-pi microcomputer and PCB board for the electronic parts, designed in this project. These units are used for driving 20" seven-segment displays, which are used for displaying visual information of the scoreboard.

As a result of this thesis, a prototype device was designed and implemented, which meets the specifications. The plan is to deliver the scoreboard to the customer in summer 2015, so in winter 2014 - 2015 it is still possible to make additions or improvements.

Key words: Raspberry-pi, electronics, microcontroller, prototype

SANASTO

AC-jännite	Vaihtojännite
AND-gate	AND-portti
BCD	Binäärikoodattu esitystapa
CAT6	Ethernet-standardi
CLK	Kellosignaali
C/C++	Ohjelmointikieli
DC	Pulssisuhde (Duty cycle)
DC-jännite	Tasajännite
EMI	Sähkömagneettinen häiriö
FLASH	Puolijohdemuisti
GND	Maapotentiaali
GPIO	Yleiskäyttöinen portti
HASL	Piirilevyn viimeistelytapa
IP-luokitus	Järjestelmä sähkölaitteiden tiiviiden määrittämiseksi
IC	Integroitu piiri
LED	Hohtodiodi
PCB	Piirilevy
RAM	Keskusmuisti
RPI	Raspberry-pi
SD	Muistikorttityyppi
SRAM	Staattinen RAM
SSD	Seven-segment-näyttö
UART	Sarjaliikennepiiri
VCC	Käyttöjännite
WIFI-hotspot	Kohde missä on internetyhteys

SISÄLLYS

1	JOHDANTO	1
2	PESÄPALLO JA PESÄPALLOTULOSTAULU	2
2.1	Pesäpallopele	2
2.2	Pesäpallotulostaulu	3
3	TULOSTAULUN ELEKTRONINEN OSUUS	5
3.1	Virtalähde	6
3.2	Käyttösähkökytkentä	8
3.3	Ohjauskytkentä	9
3.3.1	Regulointikytkentä	10
3.3.2	Mbed-mikrokontrolleri	11
3.3.3	74HC4511	12
3.3.4	SN74HC7001N	14
3.3.5	ULN2003	16
3.4	Piirilevyn toteutus	18
3.5	Kaapelointi	19
3.6	Seven-segment-näytöt	22
3.7	Raspberry-pi	24
3.8	RUT700-reititin	26
4	TULOSTAULUN MEKAANINEN RAKENNE	28
4.1	Vanerit	29
4.2	Pleksilasi	29
4.3	Kotelot	30
5	OHJELMALLINEN OSUUS	32
5.1	Raspberry-pin ohjelmakoodi	32
5.2	Mbed-mikrokontrollerin ohjelmakoodi	34
6	PROJEKTIN ONNISTUMINEN JA ARVIONTI	37
6.1	Haasteet	37
6.2	Jatkokehitys	38
7	YHTEENVETO	40
	LÄHTEET	41
	LIITTEET	

VIRHE. KIRJANMERKKIÄ EI OLE MÄÄRITETTY.

1 JOHDANTO

Opinnäytetyön tavoitteena oli suunnitella ja rakentaa pesäpallotulostaulu lahtelaiselle pesäpalloseuralahden Mailaveikoille. Työ sai alkunsa tarpeesta päivittää Lahden kisapuiston hiekkatekonurmikentän pesäpallotulostaulu. Ennen kentällä toimi manuaalisesti käytettävä tulostaulu, josta ei selvinnyt kaikki pesäpallopelille ominaiset tilannekohtaiset tiedot. Lisäksi entinen tulostaulu oli kokonsa ja informaationäyttöjensä puolesta epäkäytännöllinen. Valmiina pesäpallotulostaulu lisää tietoisuutta pesäpallosta Lahden kaupungissa sekä tuo helpotusta pelitilanteen seuraamiseen kentällä oleville katsojille ja pelissä mukana oleville

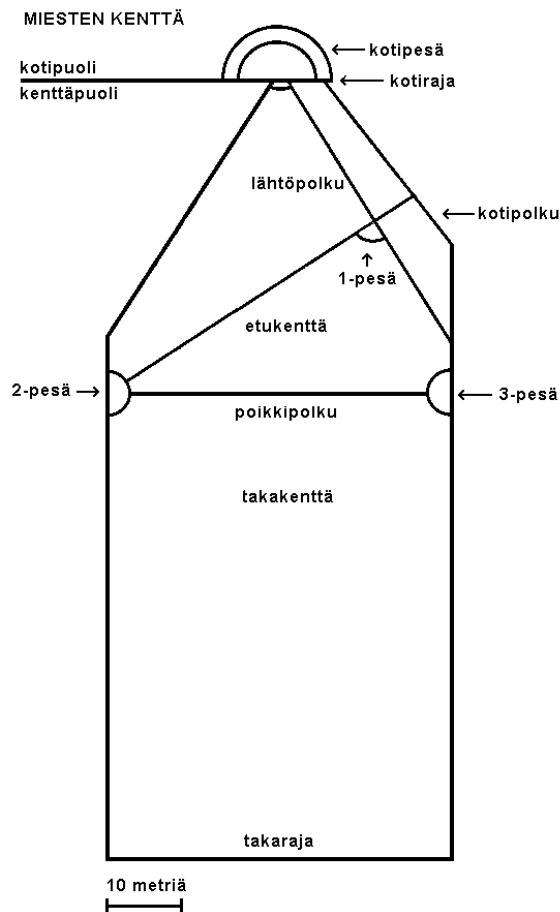
Tämä opinnäytetyö aloitettiin jo syksyllä 2013, jolloin pidettiin ensimmäiset projektikokoukset. Tästä alkoi laajamuotoinen taustaselvittely projektin eri osuuksien osalta. Selvittelyiden pohjalta pystyttiin aloittamaan kokonaisuuden suunnittelu, joka vastasi kokouksissa tilaajan luomia spesifikaatioita. Tärkeää projektin edetessä oli pitää toimeksiantaja mukana suunnitelmia luodessa.

Työ on suunniteltu niin, että valmiina tulostaulu toimii täysin ”pistoke seinään”-periaatteella, eli loppukäyttäjän ei tarvitsisi tehdä laitteella muuta kuin laittaa se päälle. Edellä esitetty suunnitteluideologia juontuu projektin toisesta osuudesta, jossa toinen projektiryhmä loi tulostaululle internetpohjaisen käyttöliittymän, jolla voidaan syöttää tieto näytettäväksi taululle. Käyttöliittymä on suunniteltu käyttäjäystävälliseksi, joten itse taulunkin tuli olla helppokäyttöinen loppukäyttäjälle.

2 PESÄPALLO JA PESÄPALLOTULOSTAULU

Pesäpallo on Suomen kansallispele. Sen juuret juontuvat 1900-luvun alkuun, ja se on kehitelty amerikkalaisen pesäpallon pohjalta. Pesäpallon kehitti Lauri ”Tahko” Pihkala 1920-luvulla. Sen erilaisia muunnoksia pelataan ympäri maailmaa.

Kuviossa 1 on esitetty suomalaisen pesäpallon pelikenttä.



KUVIO 1. Pesäpallokenttä (Wikipedia 2005)

2.1 Pesäpallopele

Pesäpallossa ideana on, että sisävuorossa oleva joukkue yrittää saada pelaajansa kierrätettyä kolmen pesän kautta takaisin kotipesään ja näin tehdä juoksun. Ulkovuorossa olevan joukkueen tehtävänä on estää toista joukkuetta saamasta juoksuja polttamalla etenijöitä pesille. Kolmen palon jälkeen tapahtuu vuoron vaihto ja joukkueiden roolit vaihtuvat. Pesäpallossa pelataan kaksi jaksoa ja nämä jaksot on jaettu neljään vuoropariin. Se joukkue, joka saa enemmän juoksuja

jakson aikana, voittaa kyseisen jakson, ja voittamalla jaksoja enemmän kuin toinen joukkue voittaa pelin. Jaksovoittojen mennessä tasan pelataan supervuoropari ja tarvittaessa kotiutusjuoksukisa, jotta pelille saadaan voittaja.

Pesäpalljoukkueessa on pelitilanteessa yhdeksän pelaajaa, jotka pelaavat ulkovuoron ja lyövät numerjärjestyksessä sisävuorossa. Lisäksi joukkueen pelaavassa kokoonpanossa on jokerilyöjiä, jotka voivat astua lyömään missä välissä tahansa sisävuorossa, kuitenkin maksimissaan kolme jokeria voi lyödä yhdessä sisävuorossa.

2.2 Pesäpallotulostaulu

Pesäpallopelissä tulostaululla halutaan ilmaista tilannetta katsojille sekä pelaaville joukkueille. Tulostaulu on yleensä sijoitettu kentän laidalle näkyvälle paikalle, josta sitä on helppo seurata pelin edetessä. Kuviossa 2 on esitetty Viininjärvellä sijaitseva tulostaulu.



KUVIO 2. Pesäpallo tulostaulu (Pietarinen 2014)

Pesäpallon tulostaulussa on yleensä esitetty melko lailla samat pelitilannetiedot riippumatta taulusta:

- kotijoukkueen juoksut
- vierasjoukkueen juoksut
- pelattavan vuoron palot
- se monesko jakso menossa
- molempien joukkueiden jaksovoitot
- se monesko vuoro jaksosta on menossa
- pelattavan vuoron viimeinen lyöjä
- lyöntivuorossa olevan lyöjän numero.

Projektissa rakennettu pesäpallon tulostaulu toteutettiin tilauksesta pesäpalloseura Lahden mailaveikoille ja Lahden kaupungin liikuntatoimelle. Lahden Mailaveikot (LMV) on vuonna 1929 perustettu pesäpalloseura. LMV on menestyksekkäin pesäpalloseura Suomessa. Joukkue on voittanut kymmenen Suomen mestaruutta, joista yhdeksän miesten joukkueella ja yhden naisten joukkueella. Nykypäivänä Lahden Mailaveikoilla on joukkueita kaikenikäisille. Lahden mailaveikkojen agendaan kuuluu lajiylpeys, liikunnan ilo, terve kilpailuhenki ja yhdessä tekeminen. (Lahden Mailaveikot 2014.)

Lahden kaupungilla on pitkäaikaiset perinteet yhtenä Suomen merkittävimpiä urheilukaupunkina. Liikuntatoimi vastaa Lahden kaupungin kaikista sen hallinnoimien liikuntakohteiden kunnossapidosta. Se myös vastaa myös kaiken ikäisten kaupunkilaisten liikunnallisuuden edistämisestä ja hyvinvoinnista. (Lahden kaupunki 2014.)

3 TULOSTAULUN ELEKTRONINEN OSUUS

Tässä luvussa käydään läpi, kuinka tulostaulun elektroniset osuudet toimivat ja kuinka ne toimivat keskenään. Laitteen ohjauselektronikkaosuudet on jaettavissa kuuteen eri toiminnalliseen lohkokon:

- virtalähde
- VCC-kytkentä
- ohjauskytkentä
- näyttoelementit ja niiden kytkentä
- raspberry-pi
- 3G-reititin.



KUVIO 3. Elektroniikan asennuskotelo

Kuviossa 3 on esitetty kaikki muut osuudet asennettuna asennuskoteloon, paitsi 3G-reititin. Valmiina käytössä laite toimii reaaliajassa päivittäen pelitiedot esille sitä mukaa kuin käyttäjä niitä syöttää päätteelleen. Toiminnallinen lohko alkaa, kun käyttäjä syöttää tiedon pelitilanteen muutoksesta Lahden ammattikorkeakoulun Niiskuneiti-palvelimella toimivaan käyttöliittymään. Kuviossa 4 on esitetty Niiskuneiti-palvelimen käyttöliittymänäkymä käyttäjälle.

Joukkue	1.jakso				2.jakso				Super	Tulos
LMV	1	0	0	0	1	0	0	0	0	0
Paukku	4	0	0	0	4	0	0	0	0	1

KOTIJOUKKUE	4. VUOROPARI	2. JAKSO
JUOKSUT		PALOT
- 0 +		- 0 +
Vuoroparin vaihto		
JOKERI	VIIMEINEN LYÖJÄ	LYÖJÄ
0	- 9 +	- 1 +

Muokkaus Kumoa vuoroparin vaihto Uusi peli Kirjautu ulos

KUVIO 4. Käyttöliittymä

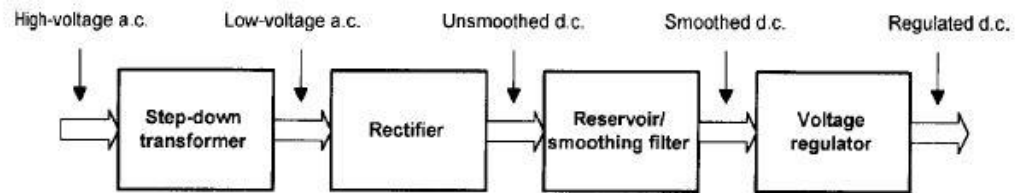
Palvelimelta pelitilanteen muutostieto siirtyy langattomasti 3G-reitittimen avulla tulostaululla olevalle Raspberry-pi:lle. Raspberry-pi:ssä tulkitaan saapuvaa data ja muokataan eteenpäin lähtevä data seuraaville osuuksille sopivaksi. Seuraavaksi tieto siirtyy Mbed-mikrokontrollerille UART-sarjayhteyttä pitkin. Mbed:ssä tapahtuu tiedon tulkinta siitä, miltä näyttöelementtien tulisi näyttää. Lisäksi Mbed:llä tuotetaan sopivat kello-signaalit näyttöelementtien ohjaukseen. Mbed toteuttaa näyttöjen ajon ohjaamalla jokaiselle näytölle osoitettua dekodierikomponenttia, jolta tieto siirtyy eteenpäin ohjauskomponenteille.

Elektroniset laitteet tarvitsevat myös käyttöönsä sähköä. Tämä tarve on toteutettu käyttämällä riittävä suurta teholahtetta, jolta siirtyy tarvittava määrä tehoa VCC-kytkennälle. VCC-kytkennässä muokataan sähköenergia sopivaksi näyttöelementeille ja ohjauskytkennälle. Ohjauskytkennässä on myös regulointipiiri, jolla saadaan ohjauselektronikalle sopivat käyttöjännitteet.

3.1 Virtalähde

Virtalähde, teholahteen ja jännitelähde ovat elektroniikassa käytettyjä kytkentöjä, joiden avulla tuotetaan erilaisiin kytkentöihin halutunlaista energiaa. Virta- ja jännitelähteen ero on siinä, millaista energiaa halutaan sen tuottavan riippumatta siitä, millainen kuorma on kytkennässä. Yleisesti kaikista elektronisista laitteista löytyy tasajännitelähde, jonka tuottamalla energialla laite toimii. Jännitelähteen

muuttaessa AC-jännitteen DC-jännitteeksi sen täytyy käydä läpi kuviossa 5 esitetty kytkentä läpi energiatehokkaasti ja minimaalisilla häviöillä. (Texas Instruments 2004a.)



KUVIO 5. Jännitelähteen toimintamallin lohkokaavio (Tooley 2006)

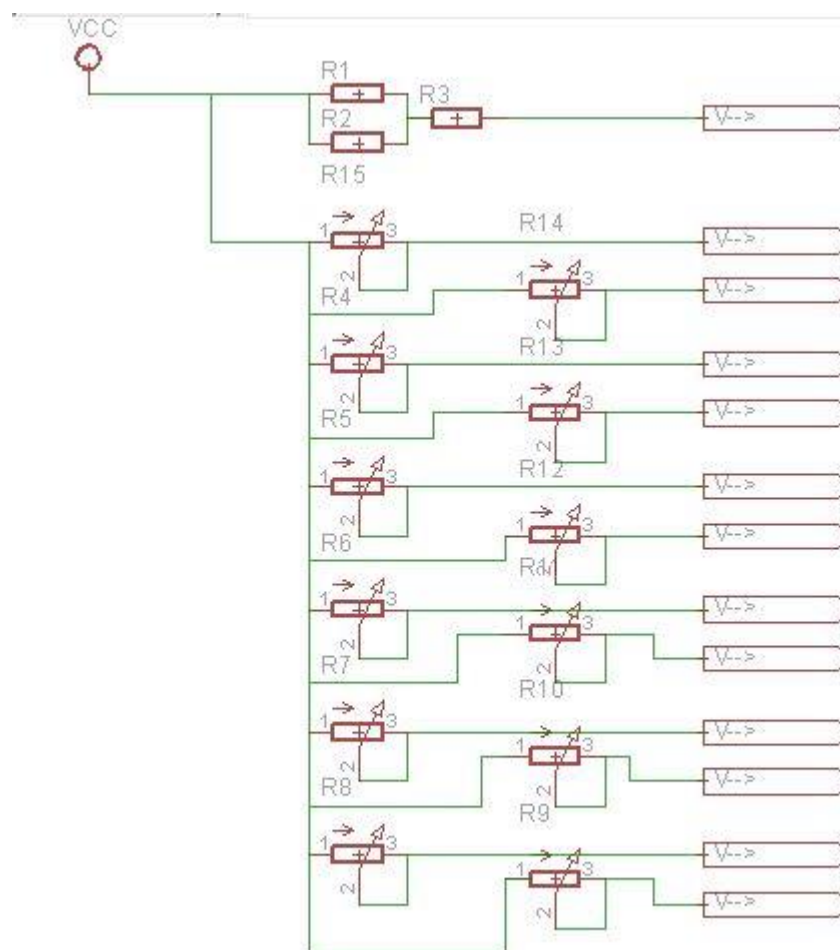
- muuntaja (transformer): laskee verkkojännitteen sopivaksi tasasuuntausta varten
- siltakytkentä (rectifier): muuntaa AC-jännitteen DC-jännitteeksi, mutta ei mitään muuta
- suodatus (filter): tasoittaa DC-jännitettä
- regulointi (regulator): tuottaa halutun DC-jännitteen.

Edellä esitetty toimintamalli on yksinkertaistettu visualisoinnin helpottamiseksi. Näiden kohtien lisäksi kaupallisessa virtalähteessä on suodatusta lähteen molemmissa päissä, vikavirtasuojasta ja kontrollipiirejä.

Laitteen jännitelähteeksi valittiin HLG-240H-15A, joka on ledien ajoon suunniteltu lähde. Kyseinen lähde tuottaa 15 A virran, 15 V jännitteen ja 225 W tehoa. Tämä laite valittiin projektiin muutamasta painavasta syystä. Ensimmäinen syy oli lähteen lähestulkoon täydellisesti sopivat spesifikaatiot, jotka olivat yli tai alle vain hieman laitteelle tarvittavista. Tulostaulussa käytetyt LED-näytöt tarvitsevat noin 14 V jännitteen, kun niiden halutaan loistavan halutulla teholla. Toinen syy valintaan oli hankintoja tehtäessä halutunlaisten lähteiden huono markkinatilanne. Jännitelähteiden hintaa ja haluttuja spesifikaatioita vertailtaessa oli lopulta edellä esitetty HLG-240H-15A ainoa projektiin sopiva valinta. (SHENZHEN YUHONG ELECTRONICS CO., LTD 2013.)

3.2 Käyttösähkökytkentä

Käyttösähkökytkentä muokkaa HLG-240H-15A:lta saatavan sähkön sopivaksi seven-segment-näytöille ja ohjauskytkennälle. Lisäksi siinä yhdistyy kaikki kytkennän maadoitukset. Käyttösähkökytkentä on toteutettu laitteen ollessa prototyypivaiheessa juovakuparoidulle piirilevyllä (Vero-levy), jonka reiät ovat 1mm halkaisijaltaan ja rasterijuovan leveys 2,54 mm. Ratkaisuun käyttää Vero-levyä projektissa ajaututtiin PCB-jyrsimen rikkoutuminen ja projektin työstön ajankohdan takia. Kuviossa 6 on esitetty käyttösähkökytkentä, missä ylin haara tuottaa käyttösähkön ohjauskytkennälle ja loput 13 seven-segment-näytöille.



KUVIO 6. Käyttösähkökytkentä

Laitteessa käytettyjen 20-tuumaisten seven-segmenttien käyttöjännite on 14 V ja ne vaativat virtaa 1,89 A, jos niitä ajettaisiin täydellä teholla. Todellisuudessa virran tarve on paljon pienempi normaalissa käytössä, mutta jännitteen täytyy olla

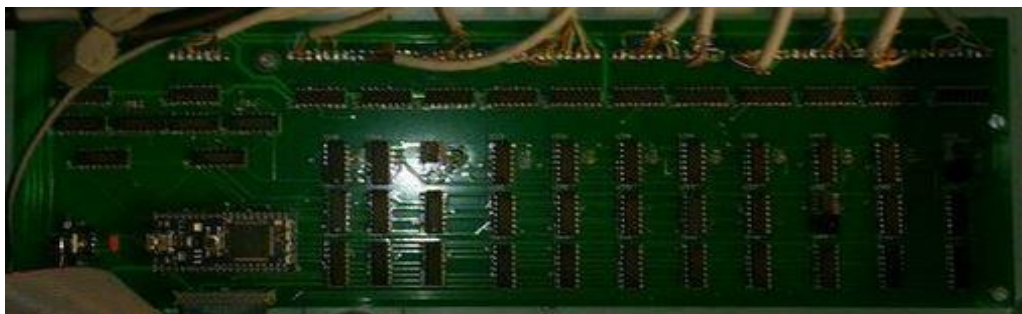
noin 14 V, jotta seven-segment-näytöt loistaisivat halutunlaisella kirkkaudella. Tätä varten on kytkennässä jokaista näyttökytkentää kohden oma 5 Ω :n potentiometri, jolla saadaan tuotettua sopivan tasoinen käyttöjännite näytöille.

Kytkenän käyttötarkoitus juontuu ohjauskytkennän regulaattorivalinnasta. Regulaattori LD1085V50 pystyy tuottamaan suuren määrän virtaa, jotta sitä riittää ohjauskytkennän kaikille komponenteille. Pystyäkseen tuottamaan halutun määrän virtaa tähän projektiin tulee regulaattorille syötettävän jännitteen määrää vähennetään 15 V:sta, minkä HLG-240G-15A virtalähde tuottaa. Regulaattorin datalehden sivulta 12 löytyvästä taulukosta 7 pystytään lukemaan tarvittava jännitearvo, jolla regulaattori syöttää ulos tarvittavan määrän virtaa. (STmicroelectronics 2013.)

Jännitetason laskua varten on kytkennässä kolme kappaletta tehovastuksia. Tehovastuksella tarkoitetaan vastusta, jonka tehonkesto on enemmän kuin 1 W. Kytkennässä käytetyt vastukset ovat arvoiltaan 8,2 Ω ja tehonkestoiltaan 8 W. Vastukset ovat kytketty kuvion 6 ylimmän haaran mukaiset, jotta vastusarvo kytkennälle on saatu sopivalle tasolle. Vastukset on valittu sen mukaan mitä oli käytettävissä kytkennän tätä osuutta rakentaessa laboratorion hyllyssä.

3.3 Ohjauskytkentä

Näyttöelementtien ohjaus tapahtuu kuviossa 7 esitetyllä elektroniikka piirilevyllä. Tämä kyseinen levy on 12,7 cm x 37,7 cm kokoinen ja 1,6 mm paksu. Levyn pinta on käsitelty HASL-tekniikalla. HASL-käsittelyssä PCB-levy kastetaan sulaan juotteeseen, jotta kaikkiin juotoskohtien kupareihin tulee pinta, johon on helpompi juottaa. (Wikipedia 2014b.)

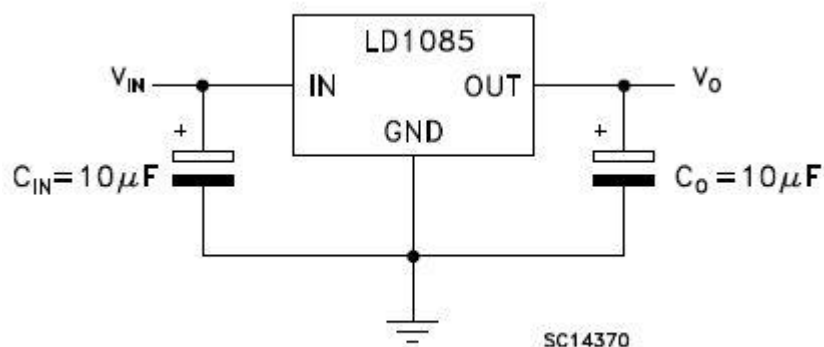


KUVIO 7. PCB-levy

Levyllä on regulointipiiri, jolla muokataan käyttösähkökytkennältä saapuva jännite sopivaksi kaikille ohjauskomponenteille. Koko kytkennän keskusyksikkönä toimii Mbed-mikrokontrolleri, jolla ohjataan kaikkia ohjauskomponentteja. Ensimmäisenä ohjauskomponenttina näyttöelementeille kytkennässä on 74HC4511-dekooderi. Seuraavaksi kytkennässä on SN74HC7001N-AND-portti ja viimeiseksi kytkennässä on ULN2003-transistoripiiri. Jokaista kolmeatoista LED seven-segment-näyttöä kohden on yksi edellä esitetty kytkentäketju, jolla suoritetaan näytön ohjaus. Levyllä löytyy myös kaksi kappaletta liittimiä: käyttösähköliitin, jossa on kaksi kappaletta napoja (VCC ja GND), ja 26-napainen sarjaliitin.

3.3.1 Regulointikytkentä

Elektronisissa kytkennöissä regulointipiirillä tuotetaan halutunlainen käyttöjännite kytkennän muille osille. Reguloinnilla pystytään säätämään kytkennälle menevien jännitteiden, virtojen ja tehojen määriä kytkentään sopiviksi. Tässä kytkennässä regulointiin käytettiin LD1085V50-regulaattoria. Kyseinen regulaattori on malliltaan kiinteää jännitettä tuottava, eli se tuottaa hyvin tarkasti 5 V antojännitteen, kunhan laitteelle syötetään tarpeeksi suuri jännite sisääntuloon. Kyseinen regulaattori valittiin laitteeseen sen suoman 3 A virran takia. Suhteellisen suurta virtaa tarvitaan kytkentään IC-komponenttien määrän takia. Kytkennässä on 39 IC-komponenttia rinnankytkettynä, Mbed-mikrokontrolleri ja 26 ylösvedettyä GPIO-pinniä.

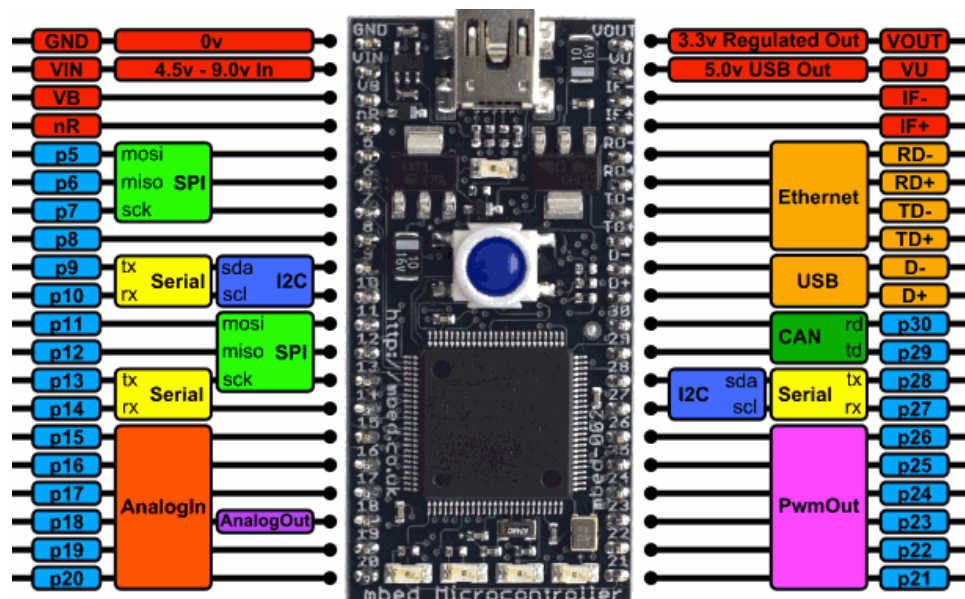


KUVIO 8. Regulointikytkentä (STmicroelectronics 2013)

Regulaattorin kanssa sarjassa on kaksi kappaletta suodatuskondensaattoreita, kuten kuviossa 8 on esitetty. Sisääntulossa on 330 nF alumiinielektrolyyttikondensaattori ja ulostulossa 10 μ F muovikondensaattori kytkettynä toisesta jalastaan maahan. Näiden kondensaattorien tehtävänä on suodattaa pois kaikki piikkijännitteet siten, että kytkennälle saataisiin mahdollisimman tasainen 5 V:n tasajännite.

3.3.2 Mbed-mikrokontrolleri

Projektin komponentteja valittaessa oli hyvä hakea komponentteja, jotka sopivat hyvin prototyyppien rakentamiseen. Tämän takia valinnassa päädyttiin NXP:n valmistamaan mbed LPC-1789 -mikrokontrolleriin. Kyseinen laite soveltuu projektiin hyvin, koska se on suunniteltu juuri nopeaan prototyyppirakentamiseen. Mikrokontrolleria ohjaa ARM Cortex-M3 -ydin, joka toimii 96 MHz:llä. Muistia laitteessa on 32 Kb RAM:a ja 512 Kb FLASH:ä.



KUVIO 9. Mbed LPC-1789 -mikrokontrolleri (NXP 2014)

Mbed:ssä on runsaasti liitäntöjä, jotka ovat esitetty kuviossa 9, ja näiden takia se myös soveltuu todella hyvin prototyyppien rakentamiseen:

- Ethernet
- USB

- 2 SPI-liitäntää
- 2 I2C-liitäntää
- 3 UART-liitäntää
- CAN-liitäntä
- 6 PWM-liitäntää
- 6 ADC-liitäntää
- GPIO-liitäntöjä.

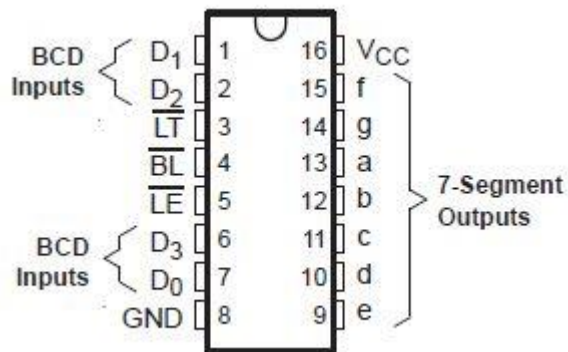
Mbed-mikrokontrolleri käyttää ohjelmointikielensä C/C++:aa. Ohjelmakoodin muokkaamiseen käytetään web-pohjaista kevyttä kääntäjää valmistajan sivuilla. Mbed:n ympärillä on myös hyvin laaja käyttäjäkunta, joka tuottaa ja muokkaa sivuille valmiita ohjelmakirjastoja kaikkien käyttöön ja tutkittavaksi. Myöskin NXP on tuottanut todella laajan kirjon kirjastoja valmiiksi mbed:iin ja täten sen käyttöönotto on todella helppoa. (NXP 2014.)

Pinni GND on kytketty kytkennän maahan ja VIN on kytketty regulointipiiriin, josta se saa 5 V käyttöjännitteen mbed:lle. Pinnit 5-8 on kytketty neljän ensimmäisten 74HC4511-dekooderien LE-pinneihin. Pinnit 9 ja 10 on kytketty UART-sarjaliittimeen, joka on yhdistettynä lattakaapelilla Raspberry-pi-mikrotietokoneeseen. Pinni 9 toimii TX:nä eli lähettää tietoa ulospäin mbed:stä ja pinni 10 toimii RX:nä, joka taas vastaanottaa tietoa Raspberry-pi:ltä. Pinnit 11 – 19 ovat kytkettynä loppuihin dekodereihin. Pinnit 21 – 24 toimivat PwmOut:na ja tuottavat ensisijaisesti 30 kHz:n taajuista kellosignaalia 20 % DC:llä (duty cycle) kytkennän SN74HC7001N-AND-portille. Pinnit 27 – 30 on kytketty 74HC4511-piirien BCD-tuloon. Nämä neljä pinniä tuottavat kytkennälle binäärisen datatiedon dekooderille.

3.3.3 74HC4511

74HC4511 on BCD-koodatun datan purkupiiri, joka on suunniteltu seven-segment-näyttöjen käyttöä varten. Kyseinen komponentti vastaanottaa 4-bittisen binäärisen datatiedon ja muuntaa vastaanotetun datan seven-segmentille sopivaksi ohjausdataksi. 74HC4511 tai muiden vastaavien purkupiirien käytön kannattavuus

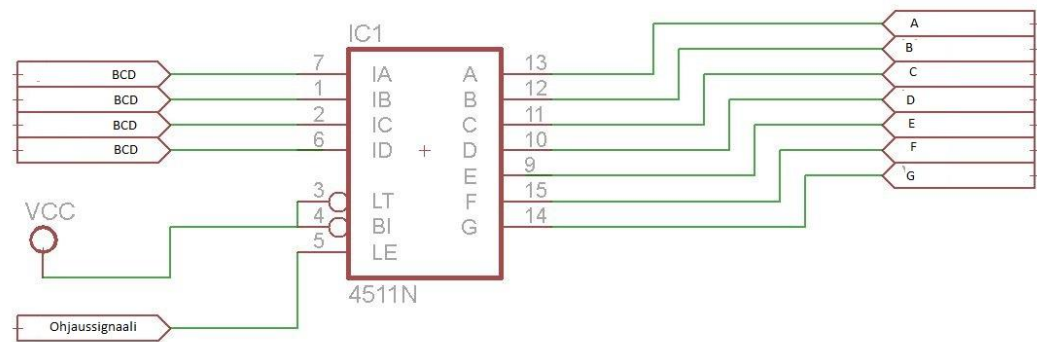
tulee siitä, että niiden avulla voidaan ajaa samanaikaisesti useita näyttöjä rinnakkain. (Texas Instruments 2003.)



KUVIO 10. 74HC4511 (Texas Instruments 2003)

Kuviossa 10 on esitetty 74HC4511:n pinnien järjestys ja niiden merkitykset. 74HC4511 toimii niin, että sille saapuvat numerot on koodattu mbed-mikrokontrollerissa binääriluvuiksi. Numeroiden 0 - 9 esittämiseen binäärisesti tarvitaan neljä bittiä, ja nämä neljä bittiä saapuvat piirille BCD-tuloihin. Neljällä bitillä on mahdollista esittää 16 lukua, mutta tässä piirissä ei ole hyödynnetty näitä 9:n yli meneviä esitystapoja, vaan kuusi suurinta binäärilukua sammuttaa ledit. Pinni LE toimii jokaisessa kytkennän purkupiirissä komponentin tunnistuspinninä. Kaikki kytkennän 13 74HC4511-komponenttia ovat kytkettynä samaan dataväylään pinneillä D0 – D3, mutta jokaiselle komponentille on oma yksittäinen datapinni LE. Tämän pinnin avulla pystytään tunnistamaan, mille purkupiirille halutaan tiedon menevän ajamalla se alatilaa ja tämän jälkeen komponentti lukee dataväylällä kulkevan tiedon. (Texas Instruments 2003.)

Komponentin ulostuloja ovat pinnit A – G, jotka kytketään jokaiseen näytön segmenttiin aakkosellisesti. Komponentissa on myös VCC- ja GND-pinnit, joiden avulla komponentti maadoitetaan ja siihen tuodaan 5 V:n käyttöjännite. Pinnit BL ja LT ovat testausta varten tai segmenttien nopeaan sammuttamiseen.

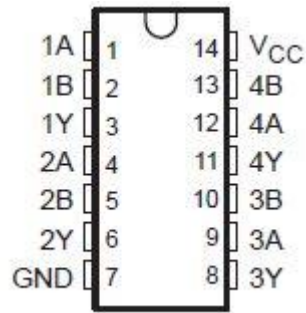


KUVIO 11. 74HC4511 kytkentä

Tässä projektissa komponentit 74HC4511 on kytketty kuvion 11 mukaisella tavalla. Pinnit D0 – D3 on yhdistetty mbed-mikrokontrollerin pinneihin 27 – 30. Pinni LE on kytketty johonkin Mbed:ssä määrättyyn ohjauspinniin. Pinnit BL ja LT on molemmat kytketty suoraan käyttöjännitteeseen, mikä pitää ne molemmat HIGH-tilassa koko ajan. Tämä on tehty siksi, että HIGH-tilassa kumpikaan pinni ei pääse vaikuttamaan komponentin toimintaan käytön aikana, kuten on komponentin datalehden toisella sivulla olevassa taulukossa esitetty. Pinnit A – G on kytketty komponenttiketjun seuraavien komponenttien SN74HC7001N:n sisääntuloihin. (Texas Instruments 2003.)

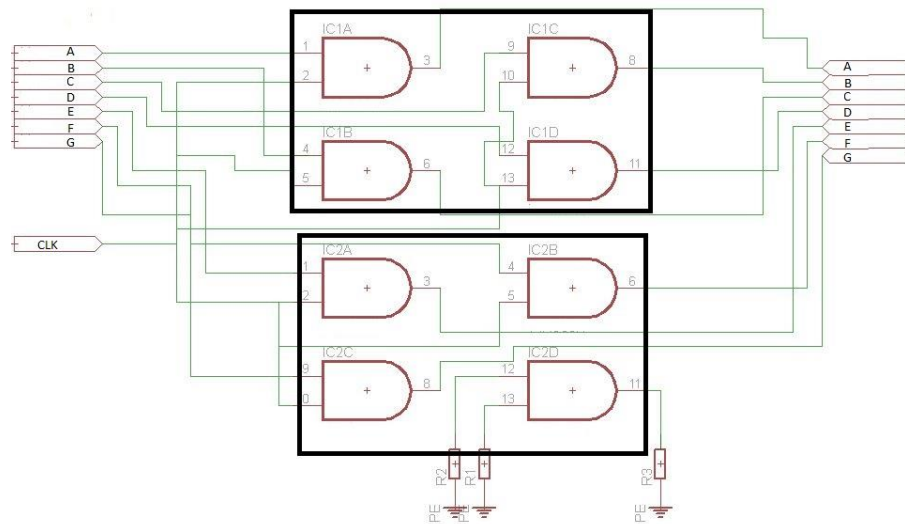
3.3.4 SN74HC7001N

SN74HC7001 on komponentti, joka sisältää neljä Schmitt-liipaisin-tulolla varustettua AND-porttia. AND-portti kuuluu loogisten komponenttien perheeseen, jonka avulla toteutetaan Boolean algebran lausekkeita elektronisesti. Portti päästää ulostulonsa ”1”-tilaan ainoastaan, kun molempien sisääntulojen tila on myös ”1”. Schmitt-liipaisin on yksi toimintapa loogisten komponenttien sisääntuloissa. Sen hyödyllisyys käytössä perustuu sisäänrakennettuun hystereesiin, jolloin komponentti ei ole kerkkä kohinalle. (Texas Instruments 2004b.)



KUVIO 12. SN74HC7001N (Texas Instruments 2004b)

SN74HC7001-komponentin jokaisella neljällä portilla on kolme kappaletta pinnejä XA, XB ja XY. Pinnit A ja B ovat molemmat sisääntuloja, ja pinni Y on ulostulo. Lisäksi komponentissa on VCC-pinni käyttöjännitteelle ja GND-pinni maahan kytkemistä varten. Kuviossa 12 on esitetty piirin pinnien järjestys.



KUVIO 13. SN74HC7001N-kytkentä

Projektissa SN74HC7001N kytkettiin kuvion 13 osoittamalla tavalla. Jokaista seven-segment-näytön ohjausta varten on kaksi kappaletta SN74HC7001N:iä. Komponentit on kytketty niin, että komponentilta 74HC4511 saapuvat datasiinaalit on kytkettynä yksitellen omaan AND-porttinsa A-pinniin. AND-portin toiseen sisääntulopinniin on kytketty mbed-mikrokontrollerilta saapuva kellosignaali. AND-portin ulostulopinni on kytkettynä ULN2003:n sisääntuloon. VCC-pinniin syötetään 5 V:n käyttöjännite, ja GND-pinni on kytkettynä maaverkoston.

Koska seven-segment-näytöissä on vain seitsemän kappaletta LED-elementtejä, niin yksi AND-portti jää kytkennässä käyttämättä. Nämä kolme pinniä, joita ei kytkennässä tarvita, on kytketty maahan käyttäen 1 k Ω :n vastuksia, jotta ne eivät jäisi turhaan kellumaan.

Komponentti toimii tässä kytkennässä ohjaussignaalin kellotaajuuden muuttajana. Mbed-mikrokontrollerilta lähtevä ohjaussignaali on tasajännitettä ja HC7411:ltä lähtevä signaali on myös tasajännitettä. Koska tässä projektissa ajetaan suurta määrää LED:jä, ei ole suotavaa pitää niitä 100 % ajasta päällä, vaan kytkeä niitä päälle ja pois kellosignaalin avulla. Kyseisellä toiminnalla pystytään käyttämään LED:jä pienemmällä määrällä energiaa. Lisäksi komponentin ULN2003:n datalehden sivulla 11, taulukossa 15 esitetään komponentin yhden ulostulon maksimi kollektorivirta suhteessa duty cycleen. Mikäli käytetään kaikki seitsemää komponentin ulostuloa, niin virransietokyky tipahtaa radikaalisti mentäessä 20 % DC:n yli. Tämän takia kytkennässä käytetyt kellosignaalit ovat 30 kHz:n taajuisia ja toimivat 20 % DC:llä, jotta voidaan ajaa kytkennän komponentteja turvallisesti. (Texas Instruments 2013.)

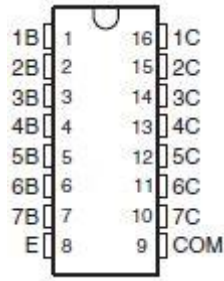
3.3.5 ULN2003

ULN2003 on suurta virtaa kestävä darlington-transistori piiri. Komponentti sisältää seitsemän kappaletta npn-transistoria. Darlington-transistori on kytkentä, jolla pystytään tuottamaan transistorille suurempi virrananto. Kytkennässä on kaksi kappaletta transistoreja, joilla on kollektorit rinnakkain ja toisen transistorin emitteri on kytkettynä toisen kantaan. (Wikipedia 2014a.)

ULN2003 kestää jopa 500 mA:n piikkivirtoja, ja tämän takia on valittu projektiin komponentiksi. Projektissa käytettyjen seven-segmenttien yhden segmentin virran tarve on maksimissaan 270 mA, joten ULN2003:n virranannossa on tarpeeksi käyttämätöntä piikkivirtojen varalle. Lisäksi yhdessä näytössä on kytketty useampi segmentti rinnakkain ja näin ollen virranantotarve on mitoitettu lähes optimaaliseksi. (Texas Instruments 2013.)

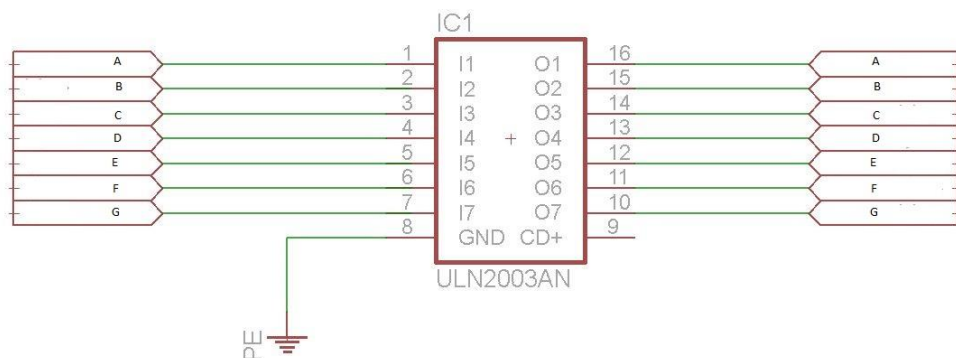
Kuviossa 14 on esitetty piirin pinnien järjestys. Samannumeroiset pinnit ovat pari. B-loppuiset pinnit on yhdistetty transistorien kantoihin (base). C-loppuiset

pinnit on yhdistetty transistorien kollektoreihin (collector) Kaikkien transistorien emitterit (emiter) on yhdistetty piirin E-pinniin. Komponentin toimintaperiaate on se, että pinniin 1B saapuva signaali saturoi transistorin kannan ja näin luo kontaktin maahan kollektorilta.



KUVIO 14. ULN2003 (Texas Instruments 2013)

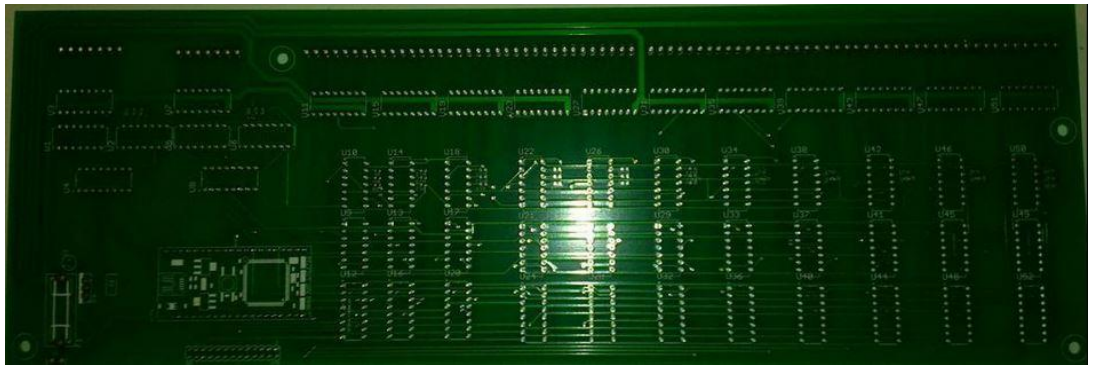
Projektissa ULN2003-komponenttia käytettiin kytkimenä seven-segmentin LED-segmenteille. Kuviossa 15 on esitetty sen kytkentä. SN74HC7001N:ltä saapuva ohjaussignaali on yhdistetty aakkosjärjestyksessä komponentin B-pinneihin. C-pinnit on yhdistetty seven-segmenttien LED:ien katodeihin. ULN2003 siis tuottaa LED:ille yhteyden maahan saturoituessaan ja saa ne palamaan. E-pinni on komponentin maapinni. COM-pinni toimii ULN2003:ssa yhteisenä tienä flyback-jännitteille, jos kytkennässä on induktiivisia kuormia ja tulisi se siinä tapauksessa kytkeä käyttöjännitteeseen. Tässä projektissa ei ole induktiivisia kuormia, joten COM-pinnille ei ole käyttöä ja näin ollen on se jätetty kytkemättä.



KUVIO 15. ULN2003-kytkentä

3.4 Piirilevyn toteutus

PCB-levyn tarkoitus on tukea elektronisia komponentteja, jotta ne eivät pääse liikkumaan ja elektronisesti yhdistää ne toisiinsa kuparijohtimilla. Levyjä voidaan suunnitella yksi- tai monikerroksisiksi. Tässä projektissa levy on kaksipuolinen, eli kuparijohtimia kulkee levyn ylä- sekä alapuolella. Kerrosten yhdistämiseen toisiinsa käytetään kuparisia läpivientejä. Kuviossa 16 on esitetty ohjauskytkennän piirilevy ilman komponentteja. Piirilevyn piirustukset löytyvät tämän opinnäytetyön liitteenä 3.



KUVIO 16. Piirilevy

Piirilevyn suunnittelussa on hyvä käyttää todettuja suunnittelusääntöjä ja pitää niistä kiinni järjestelmällisesti. Seuraavaa listausta on käytetty projektin PCB-piirilevyn suunnittelussa apuna (Toolean, 2006, 330):

:

1. Komponenttivalintojen jälkeen tarkista kaikkien komponenttien olemassaolo PCB-sunnitelmassa ja merkitse ne loogisesti, jotta ne ovat tunnistettavissa myöhemmin. Merkkää myös tarkasti kaikki levytä ulos kytkevät yhteydet.
2. Merkitse ylös kaikki väylä-yhteydet, joissa on useita ulospääsypisteitä ja sijoittele komponentteja siten, että näiden yhteyksien ei tarvitse hyppiä tai kierrellä mahdottoman paljon.
3. Merkitse ylös komponentit, jotka tarvitsevat erityistä huomiota ja sijoita nämä järkevästi huomioon ottaen niiden tarkoituksensa.

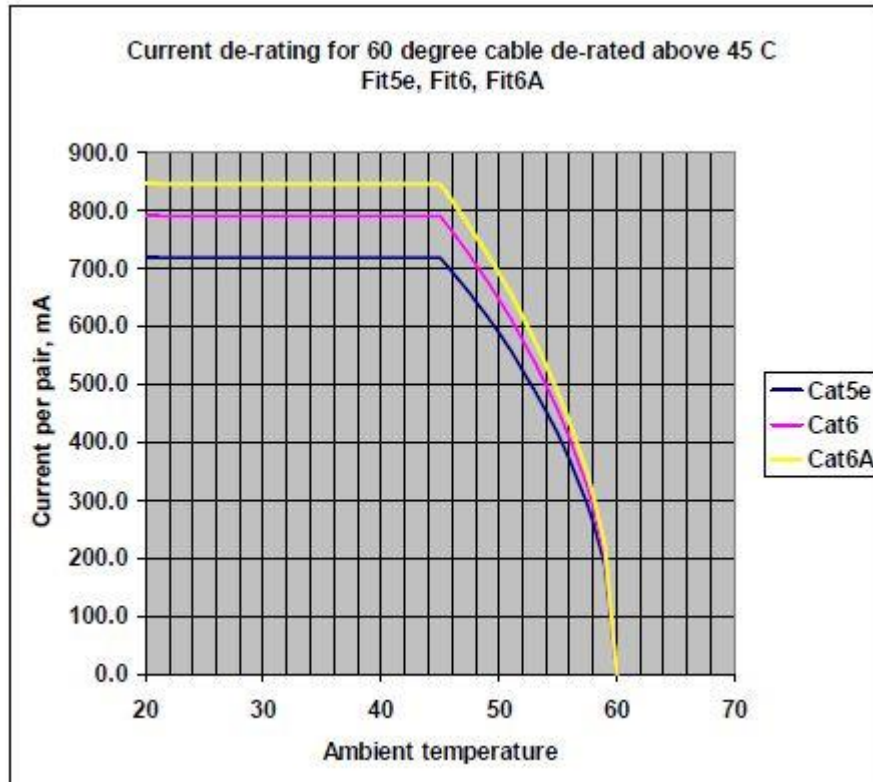
4. Merkitse ylös kaikki vedot, jotka tarvitsevat erityistä huomiota. Näitä vetoja ovat esimerkiksi mikrokontrollerien ohjaussignaalit tai muut vastaavat suurtaajuiset signaalit.
5. Sijoita sisään- ja ulostulot levyn eripäihin ja hyvänä suunnittelusääntönä voidaan pitää sähkön kulkemista vasemmalta oikealle.
6. Sijoita kytkennän eri osuudet erilleen ja lämpöä tuottavat komponentit järkevästi.
7. Aloita vetojen kytkeminen käyttöjännitteestä ja maasta.
8. Yritä sijoittaa komponentit mahdollisimman lähemmäs ja näin säästää levyn pinta-alaa.
9. Yritä saadaa maajohtimia mahdollisimman laajalle aluelle levyllä.
10. Suunnittele levyn mekaanista kiinnitystä. Mistä kohtaa ja miten levy kiinnitetään tukevasti alustaansa?
11. Pidä johdinten vedot mahdollisimman loogisina ja kulmat mahdollisimman loivina.
12. Tunnista kaikki vedot, joissa kulkee suuri määrä virtaa tai jännite on suuri ja levennä niitä tarvittaessa. Tässä projektissa suunnittelusääntönä pidettiin Toolean M. s. 332 taulukkoja 19.1 ja 19.2.
13. Mieti suodatuskondensaattorien määrää ja sijoitusta.
14. EMI-suodatus.

3.5 Kaapelointi

Kaapeloinnin tarkoitus elektronisissa kytkennöissä on yhdistää toisiinsa fyysisesti erillään olevat elektroniset laitteet ja tuoda laitteille virtalähteeltä laitteen vaatima käyttöjännite. Projektissa on käytetty neljää erityyppistä kaapelia, jotka toimittavat omaa käyttötarkoitustaan.

Projektissa parikaapelia käytettiin luomaan yhteys seven-segment-näyttöjen ja ohjauskytkennän välille. Parikaapelin sopivuus projektiin kaapelointiin juontuu tarvittavien johtimien määrästä. Seven-segmentissä on seitsemän liityntää, ja parikaapelissa on kahdeksan johdinta. Lisäksi parikaapelin hyvä taajuusalue oli otettava huomioon. Parikaapelin yhden johtimen tulee pystyä kuljettamaan tämän projektin tapauksessa maksimissaan 270 mA virtaa ja 3,8 W tehoa. Näihin

arvoihin riittää CAT5-kaapeli ja projektissa käytetty CAT6-kaapeli. Kuviosta 17 voidaan nähdä, että kyseiset kaapelit kestävät tarvittavat virtakuormat normaaleissa suomalaisissa lämpötiloissa. Kuviossa 17 esitetyt arvot ovat standardin IEEE 802.3at mukaiset. Kyseinen standardi sisältää ohjeistukset ”power over ethernet”-teknologiaan. (Vaden 2014.)



KUVIO 17. CAT-kaapeleiden virrankesto suhteessa ympäröivään lämpötilaan (Vaden 2014)

Parikaapelit on kytketty taulukon 1 mukaisella tavalla, jotta kytkentä olisi mahdollisimman looginen. Ainoa tästä kytkentätavasta poikkeava seven-segmentin kytkentä on vuoroparinäyttöjen näyttö, joka esittää tiedon, onko vuoroparin alku vai loppu. Tässä yksittäisessä näytössä on ohjauskytkennän päässä kaapelit kytketty taulukon 2 mukaisella tavalla. Mbed-mikrokontrollerin ja Raspberry-pin sarja-napojen välistä kommunikointia hoitaa 26-napainen lattakaapeli, jossa on 2,54 mm liittimet päässä. Kyseinen kaapeli on todella käytetty tällaisissa kytkennöissä yleisesti.

TAULUKKO 1. Parikaapelien kytkentä

Väri	Segmentin kirjain
Sininen	A
Valko-sininen	B
Oranssi	C
Valko-oranssi	D
Ruskea	E
Valko-ruskea	F
Vihreä	G

TAULUKKO 2. Vuoroparinäytön kaapeleiden kytkentä

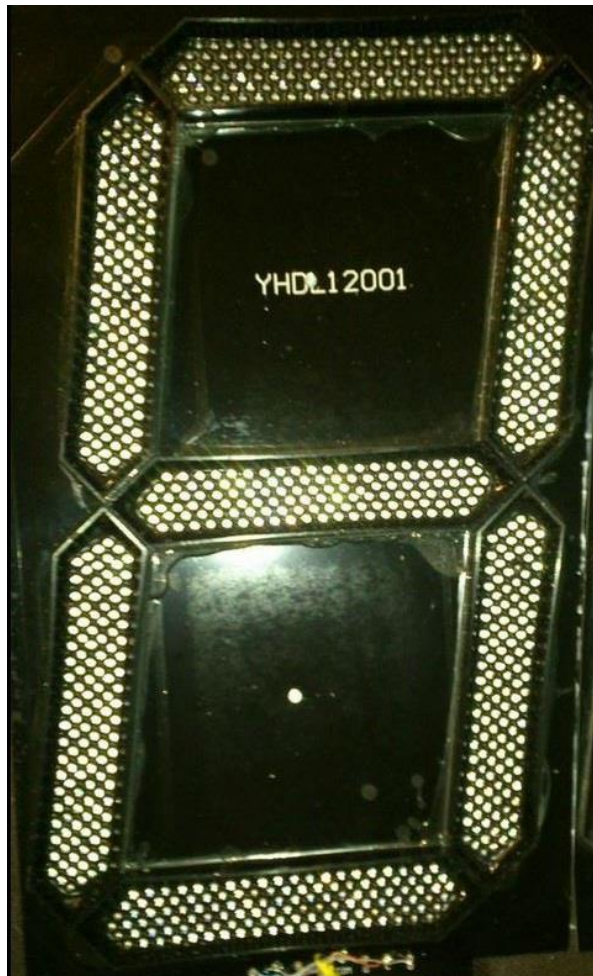
Väri	Segmentin kirjain
Valko-oranssi	A
Ruskea	B
Valko-ruskea	C
	D
	E
Sininen ja Valko-sininen	F
Oranssi ja Vihreä	G

Virtajohtoina kytkennässä toimii kaksinapainen kuparikaapeli, jossa on PVC-suojaus. Kaapeleita on laitteessa kahden värisiä: harmaa/harmaa mustalla raidalla ja punainen/musta. Kaapeleista harmaa ja punainen on kytketty niin, että ne kuljettavat käyttöjännitettä eteenpäin virtalähteeltä. Musta ja harmaa mustalla raidalla ovat kytkennässä maajohtimina. Kyseiset kaapelit on mitoitettu toimimaan hyvin alle 75 V:n jännitteellisissä kytkennöissä ja toimimaan hyvin suurtaajuisissa kytkennöissä.

Sähköverkkoon kytkettynä johtona toimii laitteessa 3-napainen virtajohto, jolla saadaan kytkettyä HLG-240H-15A-virtalähteelle käyttösähkö. Kaapelin johtimet ovat kytkettynä virtalähteen samenvärisiin johtimiin. Kaapelin johtimet ovat väreiltään sininen, musta ja keltavihreä. Sinisessä johtimessa on 0-vaihe, mustassa johtimessa ensimmäinen vaihe. Keltavihreä on maadoitusjohdin standardien mukaan. Kaapelin toisessa päässä on suojaluokkaan II kuuluva pistotulppa, jonka eriste on IP44-luokiteltu.

3.6 Seven-segment-näytöt

Seven-segment-näyttö on elektroninen näyttöelementti. SSD sisältää seitsemän kappaletta näyttösegmenttejä. Se pystyy esittämään 128 erilaista tilaa näytöillä. Yleisimmin SSD:tä käytetään numeroiden esittämiseen. Näytön segmentit on yleensä koodattu aakkosellisesti A – G ylimmästä alkaen ja kellon suuntaisesti siirtyen ja loppuen keskimmäiseen segmenttiin. Yleisimmin varsinkin pienemmissä näytöissä SSD on toteutettu LED:eillä, näin SSD:n koon kasvaessa tarvitsee vain tuoda samaan kytkentään vain lisää LED:ejä. SSD:n segmentin kytkentä on yleensä verrattavissa yksittäiseen LED:iin. Kytkennän käyttöjännite tuodaan anodille ja katodi kytketään maan suuntaan. Kytkemällä LED:jä rinnakkain voidaan nostaa tarvittavan virran määrää, mutta pitää jännitetarve entisellään. Kytkemällä LED:jä taas sarjaan nostetaan käyttöjännitteen määrää, mutta pidetään virran tarve entisellään

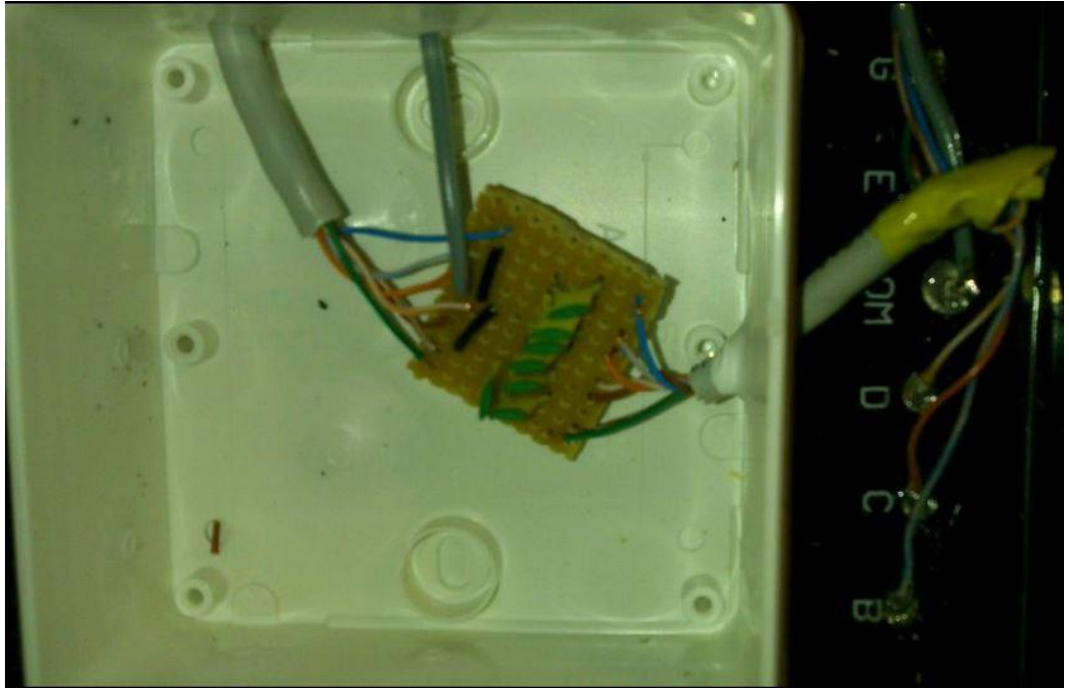


Kuvio 18. Seven-segment-näyttö

Projektissa on käytetty 13 kappaletta 20-tuumaista seven-segment-näyttöä ja kuviossa 18 on yksi sellainen esitetty. SSD:t on valmistanut kiinalainen elektroniikan komponenttien valmistamiseen erikoistunut yritys Shenzhen Yuhong Electronics Co., LTD. Yhdessä segmentissä on 126 LED:ä, jotka on kytketty 7 kappaleen sarjoihin, joita on 18 rinnakkain. Yhden LED:n käyttöjännitteeksi on suositeltu 2 V:n jännitettä, joten käyttöjännitteeksi SSD:lle saadaan 14 V. Jatkuvaksi virraksi datalehti suosittelee 15 mA:a yhtä sarjaa kohden, joten jatkuva I_c on yhdelle segmentille 270 mA, mikä on ylin suositeltu virta-arvo. Koko kytkennän Seven-segmenttejä ajettaessa täydellä virralla, kaikki LED:t päällä ja DC 100 % niin, näytöt voisivat teoreettisesti vaatia 24,5 A virtaa. Todellisuudessa arvo on murto-osan edellä esitetystä. (SHENZHEN YUHONG ELECTRONICS CO., LTD 2013.)

Tulostaulussa näytöt on kytketty niin, että jokaisessa näytössä anodit ovat kytkettynä yhteen ja tähän pisteeseen tuodaan käyttö sähkökytkennältä käyttöjännite. Jokaisen näytön kaikille segmenteille on oma liitin, johon yhdistyy kaikki rinnakkain segmentissä olevat katodit. Liittimeen on kytketty jokaiselle oma aakkos- ja värikoodattu johdin, joka on yhdistetty seven-segmentin suodatuskytkentään.

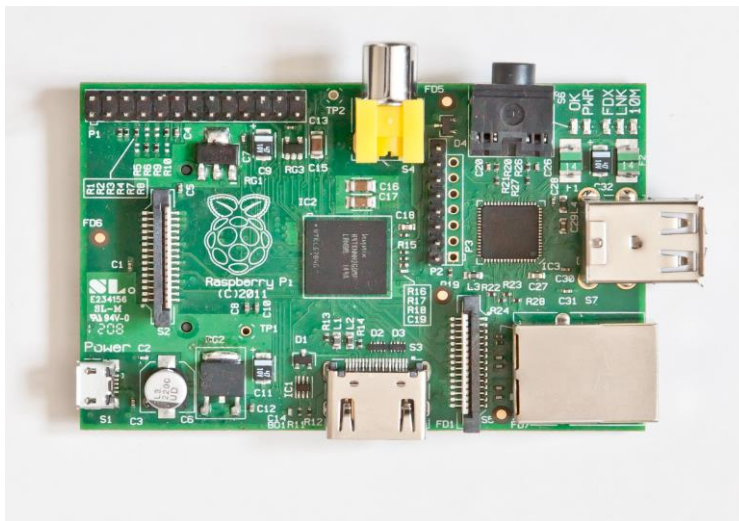
Suodatuskytkennän ideana on tuoda kytkentään lisää häiriöiden suodatusta suoraan seven-segmenttien jälkeen, jotta häiriökohinaa ei joutuisi ohjauspiirissä oleviin ULN2003-transistoreille. Suodatus on toteutettu liittämällä ylimääräinen piirilevy suoraan SSD:n jälkeen. Kuviossa 19 on esitetty yksi suodatuskytkentä. Kyseisellä piirilevyllä sijaitsee jokaista johdinta kohden 22nF keraaminen kondensaattori. Näitä suodatuskondensaattoreita kutsutaan myös Y-kondensaattoreiksi. Y-kondensaattorien koko alue on yleensä 470 pF – 48 nF ja tässä tapauksessa 22 nF antoi parhaan suodatustehon.



KUVIO 19. Suodatuskytkentä

3.7 Raspberry-pi

Raspberry-pi on luottokortin kokoinen mikrotietokone, jossa kaikki tietokoneen toiminnalle tärkeät komponentit on sijoitettu yhdelle piirilevyille. RPi julkaistiin vuonna 2012, ja siitä on kolme eri mallia saatavilla: malli A, malli B ja malli B+. RPi:n valmistajana ja jakelijana toimivat yritykset Premier Farnell ja RS Components. Kuviossa 20 on esitettyä Raspberry-pin malli B.



KUVIO 20. Raspberry-pi (Raspberry-pi 2012)

Raspberry-pi malli B pyörii ARM11-perheen 700 MHz:n ARM1176JZF-suorittimella. Siinä on 512 Mbittiä SRAM-muistia. Kaikki Raspberry-pin liittimet on esitetty taulukossa 3.

TAULUKKO 3. Raspberry-pin liittimet

Liitin	
USB 2.0	2 kpl
Videoliittimet	RCA ja HDMI
Ääniliittimet	HDMI ja 3,5mm blugi
massamuisti	SD-muistikortti
Verkkoliitin	10/100 Ethernet
Virtalähde	5V MicroUSB

Raspberry-pissä käyttöjärjestelmänä voi olla Archilinux ARM, OpenELEC, PIDORA, Raspbmc, RISC OS, Rasbian tai joku muu monista linux-kernel-pohjaisista käyttöjärjestelmistä. Edellä esitetyt käyttöjärjestelmät ovat valmiina asennettaviksi heti käyttöönotettaessa RPi:n asennusmanagerissa NOOB:ssa. Massamuistia RPi:ssä ei ole valmiina, joten käyttöönotettaessa täytyy RPi:hin asentaa SD-muistikortti, joka toimii RPi:n massamuistina.

Raspberry-pin käytön suosio perustuu sen aloittelijaystävällisyyteen, helppoon laajennettavuuteen ja asialleen omistautuneeseen yhteisöön. Yhteisössä kehitellään jatkuvasti uusia projekti-ideoita, ja ideoita jaetaan ja dokumentoidaan ahkerasti. Tämä perustuu Raspberry-pin Open-source toteutusmalliin. (Wikipedia 2014c.)

Tässä projektissa Raspberry-piin toimintaidea on toimia verkkosovittimena ja verkosta vastaanotettavan datan tulkitsijana mbed-mikrokontrollerille. RPi on yhteydessä verkkoon Element14 WIFI-donglen avulla, joka on yhdistetty yhteen RPi:n USB 2.0 -porteista. Kyseinen dongle on suunniteltu juuri RPi:lle ja on näin ollen helppokäyttöinen ja ongelmaton asentaa. RPi:ssä toimii Rasbian-käyttöjärjestelmällä.

Raspberry-pissä on taustalla pyörivä C++-kielinen skripti, joka kyselee palvelimalla olevalta tulostaulun ohjausohjelmalta pelitilannetta. Pelitilanne tiedot RPi lähettää GPIO-liittimen UART-sarjaliitännän kautta mbed-

mikrokontrollerille. sarjayhteudessa tarvitaan vain kahta liitinpiinniä: TX ja RX. TX-pinni lähettää tietoa Mbed:lle ja RX-pinni taas vastaanottaa tietoa Mbed:ltä. Kytkennällisesti pitää muistaa kytkeä nämä kaksi johdinta ristiin, jotta serial-yhteys toimii, eli RPi:n päässä oleva TX-pinni yhdistetään mbed:n päässä RX-pinniin ja RX-pinni TX-pinniin.

3.8 RUT700-reititin

RUT700 on Teltonika-yrityksen valmistama 3G-ulkoilmareititin. Kuviossa 21 on esitetty RUT700-reititin. Reititin tuottaa suurnopeuksista langatonta verkkoyhteyttä ja sisältää liitännämahdollisuuden langalliseen ethernetiin. Reititin toimii HSPA+ 3G-matkapuhelin ja pakettidatan siirtoprotokollalla. Reititin pystyy tuottamaan valmistajan mukaan jopa 21 Mbit latausnopeuden, joka on tarvittaessa jaettavissa reitittimen asetusten avulla. Reitittimen käyttöä on helppo säätää sen web-käyttöliittimen avulla. (Teltonika 2014)



KUVIO 21. RUT700-reititin (Teltonika 2014)

Projektissa RUT700-reititintä käytetään tuottamaan Raspberry-pi:lle yhteys verkkoon, jotta se voi saada palvelimelta tiedot pelitilannemuutoksista. RUT700-reititintä käytetään projektissa, koska tulostaulun loppusijoituspaikka sijaitsee sellaisella paikalla, että ethernet-kaapelin tuominen sille ei ollut toimiva ratkaisu. Lisäksi käyttämällä reititintä saadaan tuotettua hiekkatekonurmelle WIFI-hotspot pelejä varten, mikä on nykypäivänä hyvää asiakaspalvelua joukkueelta.

4 TULOSTAULUN MEKAANINEN RAKENNE

Pesäpallotulostaulun mekaanisen rakenteen pohjana toimii tässä työssä alumiinista valmistettu kehikko. Kehikko on kooltaan 3050 mm x 2050 mm x 180mm. Siinä kiertää etu- sekä takapinnassa neliskanttinen alumiiniputki ympäri. Tukevoittamassa kehikkoa on kolme poikittain kulkevaa alumiiniputkea etu- sekä takapinnassa, ja ne on sijoitettu ylös, alas ja keskelle. Lisätukea kehikkoon on saatu tukipilarien lisäämisellä kulmiin. Tulostaulun kaikki sivut ovat 180 mm leveää alumiinikiskoa. Kiskoissa on urat kahdelle vanerille ja pleksilasille. Kiskoissa on kaksi kappaletta tuuletusreikiä alhaalla vasemmassa nurkassa ja oikeassa sivussa ylhäällä. Tulostaulun pohjakisko on irrotettavissa, ja se pysyy kiinni neljällä kuusiokoloruuvilla.

Kehikon suunnittelu aloitettiin jo tammikuussa 2014, ja se oli alun perin tarkoitus toteuttaa teräksestä Lahden ammattikorkeakoulun metallintyöstölaboratoriossa. Suunnitelmiin konsultoitii useita metallintyöstämiseen erikoistuneita opettajia. Samalla myös tutkittiin, pystyttäisiinkö tätä projektin vaihetta ulkoistamaan, koska kukaan projektissa mukana ollut ei ole metallintyöstöön erikoistunut.

Tulostaulun kehikon on valmistanut lahtelainen painatukseen erikoistunut yritys Valopalkki. Tulostaulun kokoon vaikutti ensimmäisenä 20-tuumaisen seven-segment-näyttöjen koko. Yhden näytön leveys on 350 mm, ja ulkonäkösuunnitelmien mukaan näyttöjä tulee olla viisi kappaletta yhdessä rivissä. Näin ollen minimileveydeksi tulee 1750 mm. Lisäksi tilaa vie kytkennöissä käytetyt kaapelit, jotka on vedetty näyttöjen välistä. Korkeussuunnassa näytöt vievät tilaa vain 1650 mm, mutta pleksilasissa olevat näyttötekstit tarvitsevat tilaa, koska niiden tulee olla luettavia kaukaakin. Lisäksi tulostaulun ylä- ja alareunoissa on tilaa mainoksille ja Lahden Mailaveikkojen logolle. Toisena vaikuttavana tekijänä tulostaulun kokoon toimi se, että kyseisen kehikko on suurin mitä Valopalkki pystyi tuottamaan.

Kehikon alalaidassa on asennettuna lämmityselementti. Elementillä pystytään lämmittämään tulostaulua talvella, jos lämpötila on liian kylmä ohjauseletronikalle. Elementillä pystytään myös kuivattamaan taulun sisustaa, jos kosteutta alkaa kerääntyä. Elementti toimii verkkovirralla ja siihen kytketty

yksivaiheinen sähköjohto, jonka päässä on sähkötulppa. Kyseinen sähkötulppa on IP44-luokiteltu.

4.1 Vanerit

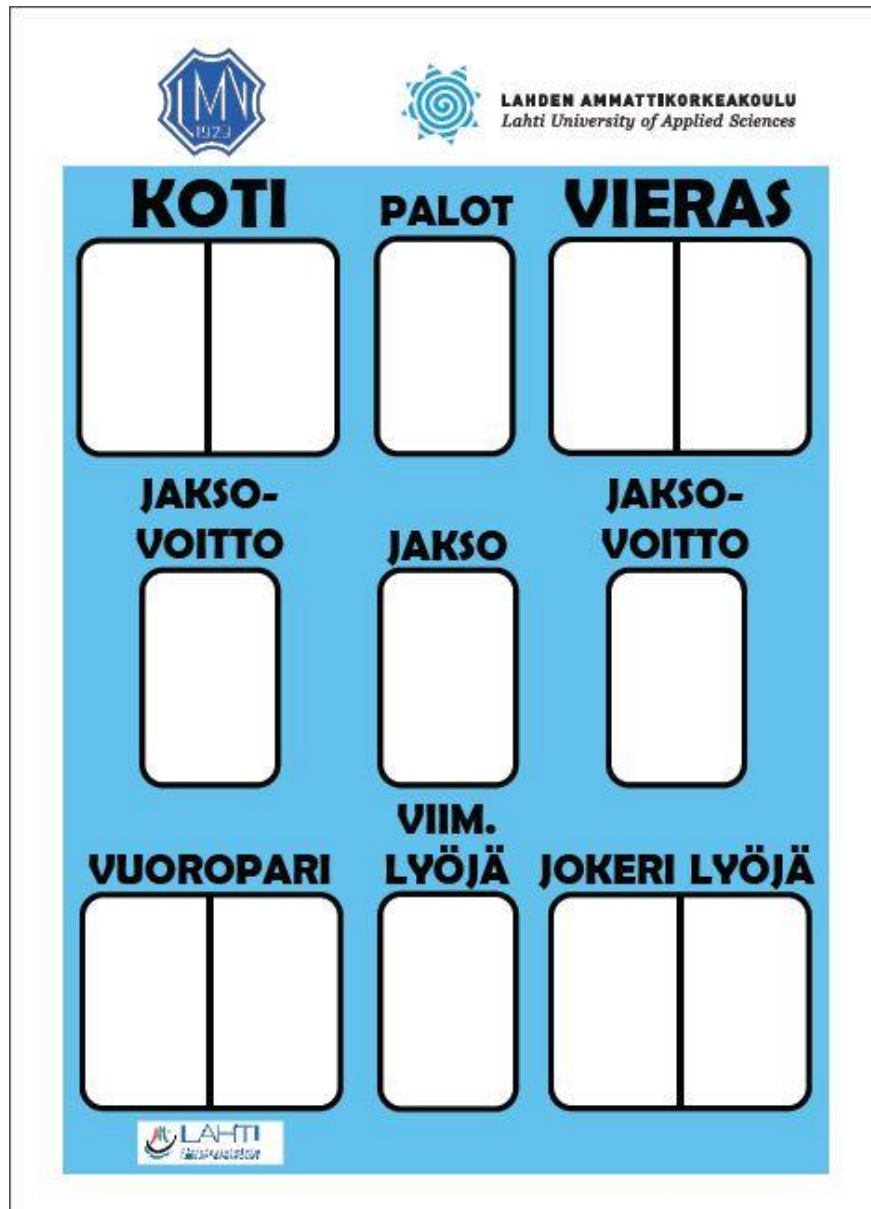
Tulostaulun takaseinänä ja seven-segment-näyttöjen asennuslevyinä projektissa toimii säänkestävä filmivaneri. Vanerien koot ovat 1500 mm x 2040 mm x 20 mm, joten takaseinä ja asennuslevy on koottu kahdesta vanerista.

Asennusvanerissa on kolme kappaletta lovia vanerin molemmissa sivuissa helpottamassa tulostaulun asentamista.

Projektissa päädyttiin käyttämään takaseinänä vanerilevyä sen hyvien säänkestäominaisuuksien ja suhteellisen sopivan painon takia. Vaneri myös kestää hyvin iskut kimmoisuutensa takia. Vanerin käyttöön asennuslevynä päädyttiin, koska tulostaulun keihikon sisäiset tukirakenteet ja pleksilasissa olevat näyttöaukot eivät linjautuneet hyvin, joten seven-segment-näytöt tarvitsivat asennusalustan. Lisäksi otettiin huomioon vanerin helppo muokattavuus ja seven-segment-näyttöjen helppo kiinnitys siihen.

4.2 Pleksilasi

Tulostaulun etupuolen visuaalisena seinämänä toimii pleksilasi, jossa on liimattuna suunniteltu ulkoasu. Pleksilasin koko on 3020 mm x 2000 mm x 5 mm. Pleksilasin toimitti projektiin Valopalkki ja sen teippauksesta huolehti myös Valopalkki. Ulkoasun pleksilasin teippaukselle on suunnitellut Elina Ylkänen Lahden ammattikorkeakoulusta. Kuviossa 22 on esitetty tulostaulun pleksilasin ulkoasu.



KUVIO 22. Pleksilasin ulkoasu

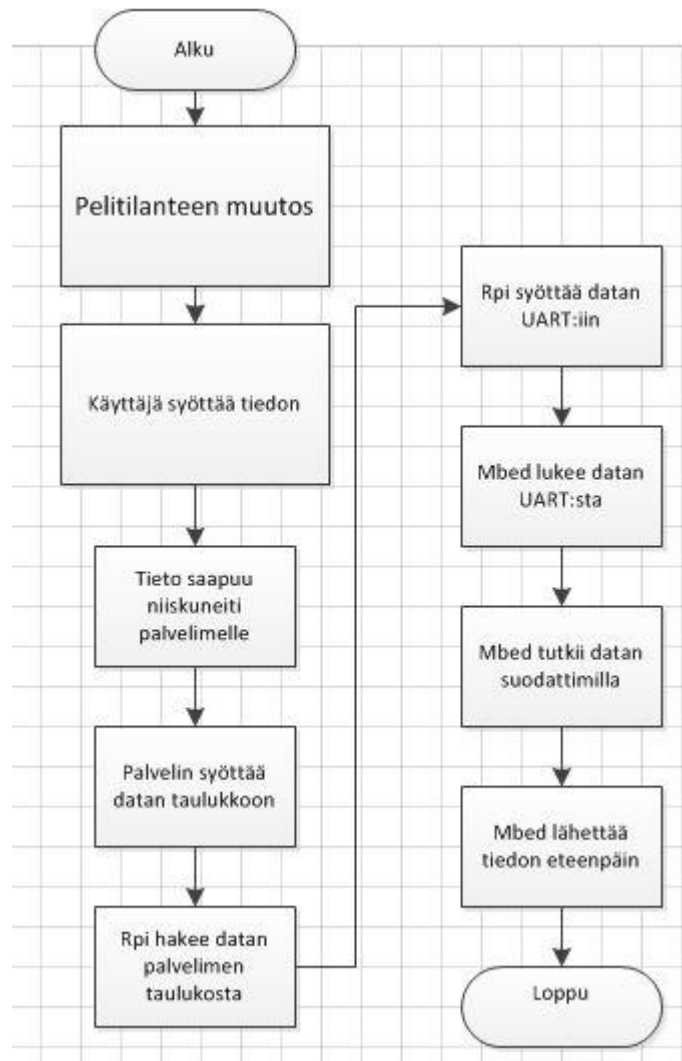
4.3 Kotelot

Tulostaulun ylälaudassa on asennuskotelo, johon on kaikki elektroniikka asennettu. Kyseinen kotelo on esitelty kuviossa 2, kun siinä on kaikki elektroniikka sisällä. Kyseinen kotelo on mallia EKP30T, ja siihen kuuluu saman sarjan kirkas läpinäkyvä kansi. Kotelon mitat ovat 380 mm x 280 mm x 30 mm. Kotelolla on IP66-luokitus, mikä tekee siitä hyvän valinnan elektroniikkaosille. Kotelossa on myös kalvotiivistelaippa, jonka läpi on vedetty kaikki parikaapelit ja käyttöjännitejohtimet. Laipassa on 43 läpivientä, ja se on IP65-luokiteltu. Seven-

segmen-näyttöjen suodatuspiirien suojausta varten on jokainen piiri sijoitettu jakorasiaan. Jakorasiat ovat kooltaan 86 mm x 86 mm x 39 mm, ja ne ovat IP55-luokiteltuja. (Fibox 2010.)

5 OHJELMALLINEN OSUUS

Projektin kaikki elektronisen osuuden laitekoodit on kirjoitettu C- ja C++-ohjelmointikielillä. Kirjoitettua ohjelmakoodia on kytkennän äly-yksiköissä Raspberry-pissä ja mbed-mikrokontrollerissa. Ohjelmien toimintaidea on esitetty kuvion 23 vuokaaviossa.



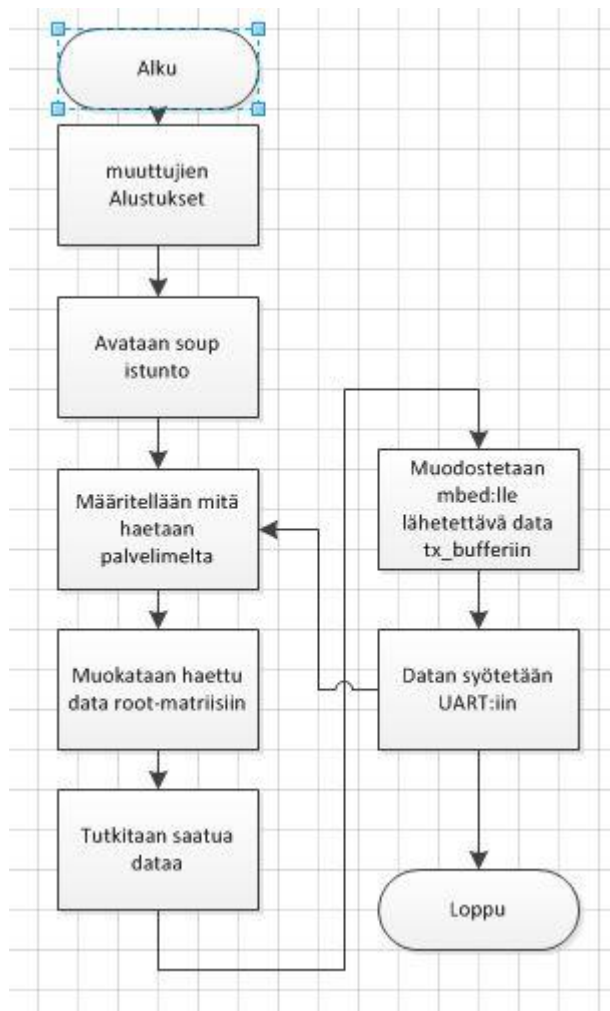
KUVIO 23. Kytkennän ohjelmien vuokaavio

5.1 Raspberry-pin ohjelmankoodi

Raspberry-pin käynnistyttyä se käy läpi tarvittavat laite- ja ohjelmistotestaukset, minkä jälkeen RPi kysyy käyttäjältä käyttäjätunnusta ja salasanaa. Tämä osuus on täytynyt ohittaa, jotta tulostaulu toimisi täysin automaattisesti. Kyseisen

toimenpiteen ohittaminen on toteutettu poistamalla aluksi käyttöjärjestelmätiedostosta inittab rivi koodia, joka ohjaa virtuaalikonsolin tty1:n ohjausta. Tämän jälkeen tiedostoon on lisätty komento, joka saa konsolin ohittamaan salasankyselyn ja kirjautuu suoraan sisään. (eLinux 2013.)

Toiseksi täytyi saada Raspberry-pissä toimiva ohjelmistokoodi käynnistymään automaattisesti, kun RPi käynnistyy. Tämä on toteutettu lisäämällä käyttöjärjestelmätiedostoon rc.local komento, joka suorittaa ohjelman. Rc.local on skripti, jonka linux ajaa käynnistyessään. Kyseiseen skriptiin lisätään juuri ohjelmat, joiden halutaan käynnistyvän tietokoneen käynnistyessä. (Raspberry projects 2013.)



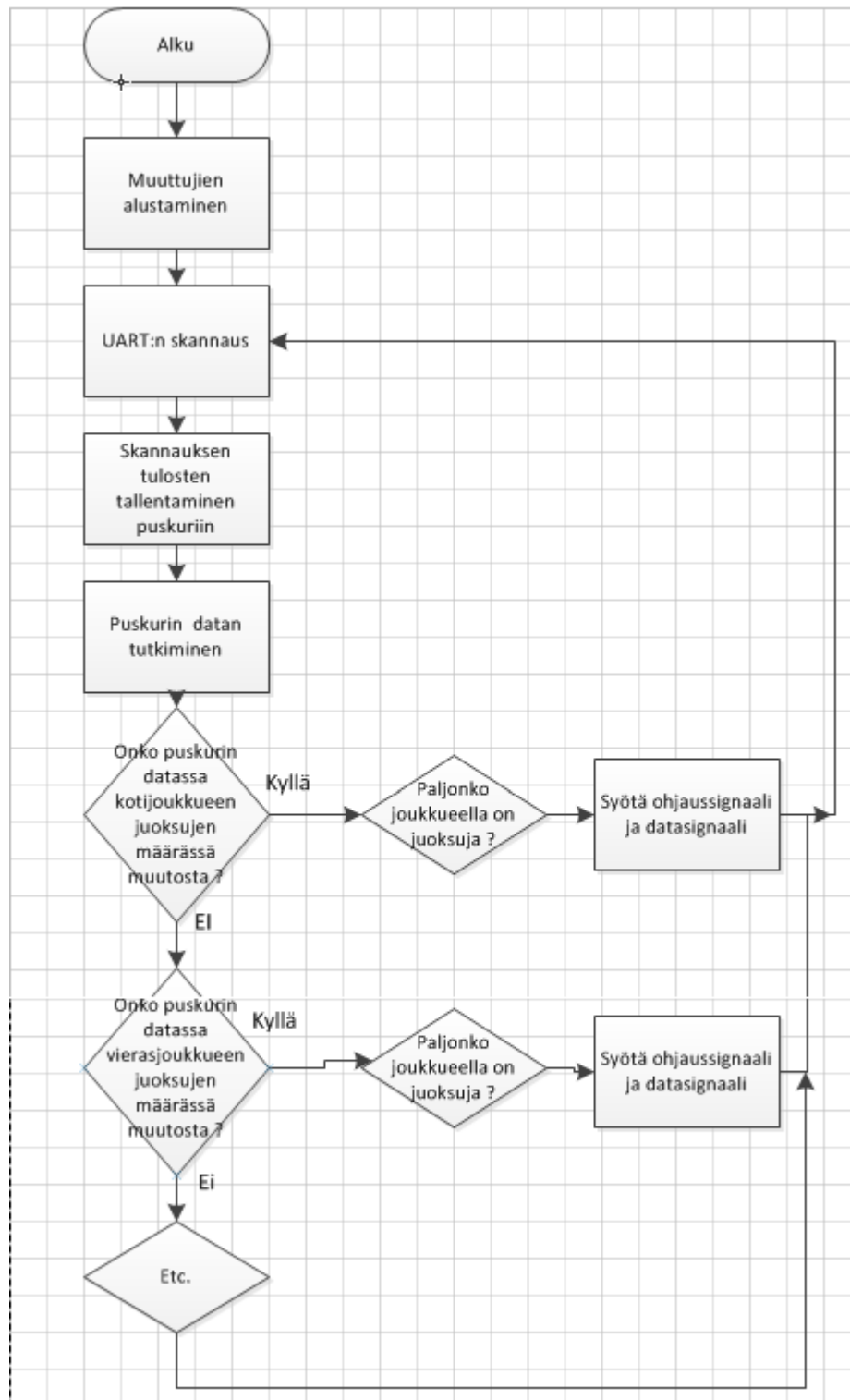
KUVIO 24. Raspberry-pin ohjelmakoodin vuokaavio

Tämän opinnäytetyön liitteenä 1 on kokonaisuudessaan Raspberry-pin ohjelmakoodi ja kuviossa 24 se esitetään vuokaaviona. Koodissa GET-

komennolla haetaan niiskuneiti-palvelimelta tietoa ja tutkitaan sitä while-silmukassa, kunnes ohjelmisto saa lopetuskäskyn. While-silmukassa tapahtuu tiedon liittäminen root-matriisiin. Root-matriisin tietoja tulkitaan seuraavaksi, ja jokainen ohjelma-askel kerrallaan etsii sen suodattimia vastaavaa dataa matriisista. Tämän jälkeen tieto syötetään lähetyspuskuriin tx_buffer string-tiedostona ja sen eteen laitetaan tunnistetieto mbed-mikrokontrolleria varten. Lopuksi lähetyspuskuri tx_buffer syötetään UART-linjalle ja lähetetään Mbed:lle. Ohjelman lopussa on UART:n alustustietoja.

5.2 Mbed-mikrokontrollerin ohjelmakoodi

Mbed-mikrokontrolleri on alun perin suunniteltu nopeaan prototyyppi-rakentamiseen, joten se on käyttövalmis lähestulkoon välittömästi saadessaan käyttäjännitteen. Näin ollen sen muistissa olevan ohjaukoodi on valmis ennen RPi:tä, mikä helpottaa tulostaulun ajamista.



KUVIO 25. Mbed-mikrokontrollerin ohjaukoodin vuokaavio

Tämän opinnäytetyön liitteenä 2 on kokonaisuudessaan Mbed-mikrokontrollerin ohjaukoodi ja kuviossa 25 on se esitetty vuokaaviona. Koodin alussa alustetaan Mbed:n sarjamuotoinen yhteys, kello-signaalien taajuuudet ja duty cyclet halutulle

tasolle. Seuraavaksi skannataan sarjamuotoinen yhteys ja tuodaan data str-datapuskuriin. Koodin pääosuus on str-datapuskurin tutkimisessa. Sen tutkimiseen käytetään if-lausekkeita, joissa jokaisessa on yksilöidyt ehdot. IF-lausekkeen ehtojen täytyessä, alkaa se toteuttaa silmukkaa. Silmukassa ohjataan neljää datapinniä ja ohjauspinnejä.

6 PROJEKTIN ONNISTUMINEN JA ARVIONTI

Lopputyön onnistumista voidaan tarkastella alussa asetettujen kriteerien avulla. Tarkoituksena oli tuottaa itsenäisesti toimiva pesäpallotulostaulu Lahden Mailaveikkojen käyttöön Lahden kisapuistoon. Tuotettu prototyyppi toimii laboratorio-olosuhteissa halutulla tavalla ja on läpäissyt tehdyt rasiustestit. Prototyypin mekaanista kestävyyttä pitkäaikaisessa käytössä pystytään testaamaan vasta kesällä 2015, kun se asennetaan paikalleen. Asennukseen saakka laitetta pystytään testaamaan ja kehittämään.

Arviotaessa opinnäytetyön onnistumista olisi myös hyvä miettiä tekijän tietotaitojen karttumista projektin edetessä. Tätä voidaan tutkia miettimällä tekijän elektroniikan ja yleisen tietotekniikan osaamisen karttumista projektia työstettäessä. Omasta mielestäni projekti on opettanut paljon itselleni ja lisännyt ja vahvistanut kyseisiä taitoja. Oppimiskokemukset ovat syntyneet erityisesti itselleni täysin uusien laitteiden käyttöönotossa ja vanhojen oppien vahvistamisessa, kun on päässyt toteuttamaan ennestään tuntemiaan oppeja konkreettisesti.

6.1 Haasteet

Tämän tyyppisissä projekteissa eteen tulee paljon tiedostettuja ja myös ennalta arvaamattomia haasteita. Suurin haaste suunnittelun kannalta oli alkuun tulostaulun mekaaninen osuus. Projektissa, missä on ulkopuolinen tilaaja, tulee ottaa tilaajan toivomukset huomioon tarkasti ja kompromissien kautta saada molempia osapuolia tyydyttävä ratkaisu aikaiseksi. Projektin edetessä jo pitkälle oli tulostaulun loppusijoituspaikka ja sen mekaaniset mitat vielä päättämättä. Tämä toi omat haasteensa tulostaulun ulkonäön ja elektroniikamsuunnitteluun. Lisäksi haastetta loivat käytetyt materiaalit ja toteutustavat, jotka olivat osaltaan entuudestaan tuntemattomia tekijälle. Projektin elektronisessa osuudessa haasteina olivat entuudestaan tuntemattomat laitteet ja toteutustavat.

6.2 Jatkokehitys

Projektissa tuotettu pesäpallotulostaulu on prototyypin laite, ja se on ensimmäinen tällä tavalla toteutettu laite. Tästä syystä laitteen valmistuessa tulee esiin osuuksia, jotka olisi voinut suunnitella tai toteuttaa toisella tavalla. Tässä luvussa tullaan esittämään jatkokehitysideoita ja yleisiä bugien korjauksia.

Pesäpallossa palot yleensä esitetään tulostaululla X:llä. Tässä tulostaulussa sama informaation esitys on toteutettu yhdellä seven-segment-näytöllä, joten palot esitetään numeraalisesti. X-esitystapa voidaan toteuttaa 16-segment-näytöllä, rakentamalla erityinen X-näyttö segmenteistä tai käyttämällä erillisiä LED-nauhoja. Projektin osia hankittaessa ei löytynyt sopivaa tuotetta tai tavarantoimittajaa markkinoilta sillä hetkellä, joten päädyimme käyttämään numeraalista esitystapaa.

Piirilevykortteja tilattaessa oli tarkoitus tuottaa käyttösähkön ja seven-segmenttien suodatuspiirilevyt elektroniikkalaboratorion PCB-jyrsimellä. Kyseisen jyrsimen toimimattomuudesta ja vuodenajan takia päädyttiin toteuttamaan edellä esitetyt piirilevyt Vero-levylle. Laitteen pitkäaikaisen kestävyyskannalta olisi suositeltavaa suunnitella uudestaan piirilevyt ja tilata ne kaikki piirilevyjä valmistavalta yritykseltä.

Laitteen elektroniikan suunnittelussa ja valmistuksessa löytyy muutamia prototyypivirheitä. Nämä virheet eivät estä laitteen käyttöä, mutta ne olisi silti hyvä korjata laitteen seuraavaan malliin.

Laitteen elektroniikan liitännät muuttuivat, kun jouduttiin ottamaan käyttöön Vero-levyt. Nämä liitännät olisi hyvä toteuttaa johtimille suunnitelluilla liittimillä, jotta ne voitaisiin helpommin irrottaa ja liittää. Seuraavaan malliin olisi myös hyvä ottaa 74HC4511:n tilalle toinen dekooderi, jolla pystyttäisiin esittämään myös kirjaimia numeroiden lisäksi. Seuraavaan malliin olisi myös hyvä lisätä EMI-suodatuksen määrää.

Mbed-mikrokontrollerin ja Raspberry-pin ohjelmistoihin olisi hyvä tehdä muutama bugi-korjaus ja yleisesti parannella koodia. Mbed:n ohjauksoodi toimii tällä hetkellä halutulla tavalla, mutta sen logiikkaa voisi yksinkertaistaa helposti.

Muuttamalla koodia sen sai mahtumaan paljon pienempään tilaan ja toimimaan joutuisammin. Lisäksi Mbed:n tallennustila ohjelmistolle on rajallinen ja tällä hetkellä ohjelmisto vie jo todella suuren osan tallennustilasta. Raspberry-pin ohjelmisto tällä hetkellä pyörii ikuisessa silmukassa, eikä näin ollen sammu ikinä ohjelmallisesti. RPi:n ohjelmistoon olisi hyvä luoda silmukka, joka sammuttaa laitteen, kun pesäpallopele tulee päätökseensä. Tämä ohjelmallinen lisä olisi hyvä lisätä, koska nykyaikaiset SD-muistikortit korruptoituvat helposti, jos niitä käyttävästä laitteesta poistaa sähkö vetämällä töpselin seinästä, mikä on tällä hetkellä ainoa tapa sammuttaa RPi.

7 YHTEENVETO

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa pesäpallotulostaulu pesäpalloseura Lahden Mailaveikkojen käyttöön. Työ toteutettiin oppilastyönä Lahden ammattikorkeakoulun lehtori Timo Turusen toimeksiannosta. Laitteen tarkoitus on parantaa pesäpallon pelin seuraamista katsomosta ja helpottaa pelaavien joukkueiden pelitilanteen seuranta.

Projekti aloitettiin kartoittamalla laitteen tarpeet ja se mitä erityisesti asiakas siltä halusi. Kartoitustyö aloitettiin syksyllä 2013 ja se jatkui vuoden 2014 puolelle. Talvella 2014 projekti eteni erilaisten toteutusmallien esittelyyn ja joidenkin ulkonäöllisten päätösten tekoon. Kevät 2014 eteni laitteen suunnittelun ja edistyksen seurantakokousten parissa. Kesällä 2014 laitteen kokoaminen voitiin aloittaa, kun kaikista yksityiskohdista oli päätetty. Syksyn 2014 alussa tulostaulun ensimmäinen prototyyppi valmistui.

Projekti oli haastava ja todella monipuolinen. Se piti sisällään monia osuuksia, joihin lukeutuu elektroniikkaa, ohjelmointia, piirilevysuunnittelua ja mekaanista suunnittelua. Lisäksi projektissa sai osallistua ahkerasti päätöstentekoon kokouksissa ja mikä esitteli myös, kuinka byrokratia toimii tämänlaisissa suunnitteluprojekteissa. Mielestäni työ sopi hyvin insinöörin opinnäytetyöksi. Siinä pääsi toteuttamaan monia erilaisia osaamisen alueita.

LÄHTEET

eLinux. 2013. RPI debian auto login [viitattu 5.11.2014]. Saatavissa:

http://elinux.org/RPi_Debian_Auto_Login

Pietarinen, E. 2014. Tulostaulu+Viinijärven+pesäpallokentällä+100512+EP.jpg

[viitattu 15.10.2014]. Saatavissa: <http://img.yle.fi/uutiset/pohjois->

[karjala/article6091510.ece/ALTERNATES/w960/Tulostaulu+Viinij%C3%A4rven+pes%C3%A4pallokent%C3%A4ll%C3%A4+100512+EP.jpg](http://img.yle.fi/uutiset/pohjois-karjala/article6091510.ece/ALTERNATES/w960/Tulostaulu+Viinij%C3%A4rven+pes%C3%A4pallokent%C3%A4ll%C3%A4+100512+EP.jpg)

Fibox. 2010. Fibox Kotelojärjestelmät ja ohjauskeskukset [viitattu 2.11.2014].

Saatavissa:

<http://www.fibox.fi/documents/Lataa/Fibox%20Tukkuri%20290110NET.pdf>

Lahden Mailaveikot. 2014. Seura [viitattu 15.10.2014]. Saatavissa:

<http://www.lahdenmailaveikot.fi/seura/>

Lahden kaupunki. 2014. Liikuntatoimi [viitattu 15.10.2014]. Saatavissa:

<http://www.lahti.fi/liikuntapalvelut>

Mean Well. 2010. HLG-240H SPEC. Mean Well Enterprises Co., Ltd. [viitattu

7.10.2014]. Saatavissa: [http://www.alldatasheet.com/datasheet-](http://www.alldatasheet.com/datasheet-pdf/pdf/347320/MEANWELL/HLG-240H-15A.html)

[pdf/pdf/347320/MEANWELL/HLG-240H-15A.html](http://www.alldatasheet.com/datasheet-pdf/pdf/347320/MEANWELL/HLG-240H-15A.html)

NXP. 2014. MBED LPC1768 [viitattu 20.10.2014]. Saatavissa:

<http://developer.mbed.org/platforms/mbed-LPC1768/>

Raspberry-pi. 2012. STARTERS GUIDE for Raspberry Pi, A, B, A+, B+ and RPI

2 B [Viitattu 1.2.2015]. Saatavissa:

<http://www.raspberrypi.org/forums/viewtopic.php?t=4751>

Raspberry projects. 2013. Auto running programs [viitattu 5.11.2014]. Saatavissa:

<http://www.raspberry-projects.com/pi/pi-operating-systems/raspbian/auto-running-programs>

SHENZHEN YUHONG ELECTRONICS CO., LTD. 2013. 20" Outdoor Red Seven Segment LED Display

- STmicroelectronics. 2013. LD1085 Datasheet. STmicroelectronics. [viitattu 9.10.2014]. Saatavissa:
<http://www.st.com/web/en/resource/technical/document/datasheet/CD00001883.pdf>
- Teltonika. 2014. Teltonika RUT700 [viitattu 6.11.2014]. Saatavissa:
<http://www.teltonika.lt/en/pages/view/?id=1068>
- Texas Instruments. 2004a. AN-556 Introduction to Power Supplies. [viitattu 9.10.2014]. Saatavissa: <http://www.ti.com/lit/an/snva006b/snva006b.pdf>
- Texas Instruments. 2003. 74HC4511 datasheet [viitattu 17.10.2014]. Saatavissa:
<https://www.sparkfun.com/datasheets/IC/74HC4511.pdf>
- Texas Instruments. 2004b. SN74HC7001N datasheet [viitattu 17.10.2014]. Saatavissa: <http://pdf1.alldatasheet.com/datasheet-pdf/view/27945/TI/SN74HC7001N.html>
- Texas Instruments. 2013. ULN2003N datasheet [viitattu 17.10.2014]. Saatavissa:
<http://www.ti.com/lit/ds/symlink/uln2003a.pdf>
- Tooley, M. 2006. Electronic Circuits – Fundamentals & Applications, third edition. Elsevier Ltd.
- Vaden, S. 2014. A contribution on cabling current de-rating vs temperature for PoE+ [viitattu 22.10.2014]. Saatavissa:
http://www.ieee802.org/3/at/public/2008/03/vaden_1_0308.pdf
- Wikipedia. 2014a. Darlington transistor [viitattu 20.10.2014]. Saatavissa:
http://en.wikipedia.org/wiki/Darlington_transistor
- Wikipedia. 2014b. Hot air solder leveling [viitattu 8.10.2014]. Saatavissa:
http://en.wikipedia.org/wiki/Hot_air_solder_leveling
- Wikipedia. 2014c. Raspberry-pi [viitattu 30.10.2014]. Saatavissa:
http://en.wikipedia.org/wiki/Raspberry_Pi

Wikipedia. 2005. Pesäpallo [viitattu 11.10.2014]. Saatavissa:
<http://fi.wikipedia.org/wiki/Pes%C3%A4pallo>

LIITTEET

Liite 1/1. Raspberry-pin ohjelmakoodi

```
#include <libsoup/soup.h>    // soup-kirjasto toetojen hakemiseksi netistä
#include <json/json.h>      // json kirjasto
#include <json/reader.h>    // json lukija

// json nimialua

using namespace Json;

// sarjaliikenteelle kirjastot

#include <stdio.h>           // standardi sarjaliikenne
#include <unistd.h>         // Used for UART
#include <fcntl.h>          // Used for UART
#include <termios.h>        // Used for UART

// järjestelmäkirjastoja
#include <sys/types.h>
#include <unistd.h>
#include <string.h>

// c++ konsoli i/o (ei tarvita lopullisessa)
#include <iostream>

using namespace std; // nimialue c++ konsolille

int main( void )
{
    // sarjaportin alustusfunktion prototyyppi
```

Liite 1/2.

```
int initUART( void );

    // taulukko merkkien esittämiseksi segmenteissä
unsigned char segments[] = {0x3f, 0x06, 0x5b, 0x4f, // 0-3
                            0x66, 0x6d, 0x7d, 0x07, // 4-7
                            0x7f, 0x6f, 0x77, 0x38, 0x00 }; // 8, 9, A, L, "

bool loppu = false; // prosessin lopetuslippu

// esitellään soupin muuttujat
SoupSession *session;
SoupMessage *msg;
uint status;

// esitellään jsonin muuttujat
Value root;
Reader reader;

// sarjaportin merkkien lähetyspuskuri
char tx_buffer[128];

// alustetaan sarjaportti
int uart = initUART();

g_type_init(); // tarvitaan soupin toimintaan alustettavaksi
```

Liite 1/3.

```
session = soup_session_sync_new(); // avataan soup istunto
// määritetään haettava tailutieto soupilla
msg = soup_message_new ("GET", "http://niisku.lamk.fi/~lmv/pesis/taulu");

while (!loppu)
{
    // haetaan taulutieto
    status = soup_session_send_message(session, msg);

    // cout << "data:\n" << msg->response_body->data << endl;

    // parsitaan vastaanotetusta tiedosta json-viesti root-matriisiin
    reader.parse( msg->response_body->data, root );

    sprintf(tx_buffer, "\nj%s\r",
            root[9]["vuoro"].asCString() );
    write( uart, tx_buffer, strlen(tx_buffer) );

    sleep( 0.05);

    sprintf( tx_buffer, "\na%s\r",           //kotijuoksut
            root[0][0]["kotijuoksut"].asCString() );
    write( uart, tx_buffer, strlen(tx_buffer) );

    sleep( 0.05 );
}
```

Liite 1/4.

```
printf( tx_buffer, "\nb%s\r\n",  
        root[1][0]["vierasjuoksut"].asCString() );  
write( uart, tx_buffer, strlen(tx_buffer) );
```

```
sleep( 0.05 );  
printf(tx_buffer, "\nc%s\r\n",  
        root[2][0][0]["palot"].asCString() );  
write( uart, tx_buffer, strlen(tx_buffer) );
```

```
sleep( 0.05 );  
printf(tx_buffer, "\nd%s\r\n",  
        root[3][0]["kotitulos"].asCString() );  
write( uart, tx_buffer, strlen(tx_buffer) );
```

```
sleep( 0.05 );  
printf(tx_buffer, "\ne%s\r\n",  
        root[4][0]["vierastulos"].asCString() );  
write( uart, tx_buffer, strlen(tx_buffer) );
```

```
sleep( 0.05 );  
printf(tx_buffer, "\nf%s\r\n",  
        root[5][0]["jakso"].asCString() );  
write( uart, tx_buffer, strlen(tx_buffer) );
```

```
sleep( 0.05 );
```

Liite 1/5.

```
    sprintf(tx_buffer, "\ng%s\r\n",
            root[6][0]["jokerit"].asCString() );
    write( uart, tx_buffer, strlen(tx_buffer) );
    sleep( 0.05 );
    sprintf(tx_buffer, "\nh%s\r\n",
            root[7][0]["lyoja"].asCString() );
    write( uart, tx_buffer, strlen(tx_buffer) );

    sleep( 0.05);
    sprintf(tx_buffer, "\ni%s\r\n",
            root[8][0]["viim_lyoja"].asCString() );
    write( uart, tx_buffer, strlen(tx_buffer) );

    sleep( 0.05 );
    sprintf(tx_buffer, "\n%s\r\n",
            root[9]["vuoro"].asCString() );
    write( uart, tx_buffer, strlen(tx_buffer) );

    sleep( 1 ); // 1 s odotus
}

return 0;
}
```

Liite 1/6.

```
/*
sarjaportin alustus
*/
int initUART( void )
{
    //-----
    //----- SETUP USART 0 -----
    //-----

    //At bootup, pins 8 and 10 are already set to UART0_TXD, UART0_RXD (ie
the alt0 function)

        // respectively

    int uart0_filestream = -1;

    //OPEN THE UART
    //The flags (defined in fcntl.h):
    //      Access modes (use 1 of these):
    //
    //          O_RDONLY - Open for reading only.
    //
    //          O_RDWR - Open for reading and writing.
    //
    //          O_WRONLY - Open for writing only.
    //
    //
    //      O_NDELAY / O_NONBLOCK (same function) - Enables
nonblocking mode. When set read
        //requests on the file can return immediately with a failure status
    //
    //
    //if there is no input immediately available (instead of blocking).
Likewise, write
```

Liite 1/7

```
// requests can also return
//

//immediately with a failure status if the output can't be written immediately.
//

//          O_NOCTTY - When set and path identifies a terminal device,
open() shall

//not cause the terminal device to become the controlling terminal for the process.

    uart0_filestream = open("/dev/ttyAMA0", O_RDWR | O_NOCTTY |
O_NDELAY);

//Open in non blocking read/write mode

    if (uart0_filestream == -1)

    {

        //ERROR - CAN'T OPEN SERIAL PORT

        return -1;

    }

//CONFIGURE THE UART

//The flags (defined in /usr/include/termios.h - see

//http://pubs.opengroup.org/onlinepubs/007908799/xsh/termios.h.html):

//          Baud rate:- B1200, B2400, B4800, B9600, B19200, B38400,
B57600, B115200, B230400, B460800,

// B500000, B576000, B921600, B1000000, B1152000, B1500000, B2000000,
B2500000, B3000000, B3500000, B4000000

//          CSIZE:- CS5, CS6, CS7, CS8

//          CLOCAL - Ignore modem status lines

//          CREAD - Enable receiver
```

Liite 1/8.

```
//          IGNPAR = Ignore characters with parity errors

//          ICRNL - Map CR to NL on input (Use for ASCII comms where you
want to auto correct end of

// line characters - don't use for binary comms!)

//          PARENB - Parity enable

//          PARODD - Odd parity (else even)

struct termios options;

tcgetattr(uart0_filestream, &options);

options.c_cflag = B9600 | CS8 | CLOCAL | CREAD;
//<Set baud rate

options.c_iflag = IGNPAR;

options.c_oflag = 0;

options.c_lflag = 0;

tcflush(uart0_filestream, TCIFLUSH);

tcsetattr(uart0_filestream, TCSANOW, &options);

return uart0_filestream; // no error - return uart stream
}
```

Liite 2/1. Mbed-mikrokontrollerin ohjaukoodi

```
#include "mbed.h"

Serial usb(USBTX, USBRX);

Serial pc(p9, p10); //tx,rx uart serial yhteys Raspberry-pi:n ja Mbed:n välillä

DigitalOut myled(LED1);

DigitalOut myled2(LED2);

DigitalOut myled3(LED3);

PwmOut kello1(p21); //kello signaali 1# 2# 3# segmenteille

PwmOut kello2(p22); //kello signaali 4# 5# 6# segmenteille

PwmOut kello3(p23); //kello signaali 7# 8# 9# segmenteille

PwmOut kello4(p24); //kello signaali 10# 11# 12# 13#segmenteille

DigitalInOut le1(p5); //Kotijuokkueen "kympit" segmentin ajo

DigitalInOut le2(p6); //Kotijuokkueen "ykköset" segmentin ajo

DigitalInOut le3(p7); //"Palot" segmentin ajo

DigitalInOut le4(p8); //Vierasjuokkueen "ykköset" segmentin ajo

DigitalInOut le5(p11); //Vierasjuokkueen "kympit" segmentin ajo

DigitalInOut le6(p12); //"kotitulos" segmentin ajo

DigitalInOut le7(p13); //"vierastulos" segmentin ajo

DigitalInOut le8(p14); //"jakso" segmentin ajo

DigitalInOut le9(p15); //"lyöjä" segmentin ajo

DigitalInOut le10(p16); //"jokerit" segmentin ajo

DigitalInOut le11(p17); //"viimeinen lyöjä" segmentin ajo

DigitalInOut le12(p18); //"vuoropari" segmentin numeron ajo

DigitalInOut le13(p19); //"vuoropari" segmentin kirjaimen ajo

DigitalInOut le14(p20);
```

Liite 2/2.

DigitalInOut le15(p25);

DigitalInOut le16(p26);

DigitalInOut d0(p30);

DigitalInOut d1(p29); //Dataginaalit 74HC4511 piireille. Segmenttien ajo
tapahtuu binäärisesti. d0=1 d1=2 d2=4 d3=8

DigitalInOut d2(p28);

DigitalInOut d3(p27);

```
int main( void )
```

```
{
```

```
    void init(void);
```

```
    void setDigit(DigitalInOut *dio, int val);
```

```
    char str[40]; //Datapuskuri
```

```
    char c; // kenttä
```

```
    int n; // kentän numeroarvo
```

```
    char ext[20]; // mahdollinen rivin loppumerkkijono - ainakin vuoroparissa  
    tarvitaan
```

```
    int cnt; // saatujen tietojen määrä
```

```
    init();
```

```
    while (1)
```

```
    {
```

```
        le14 = 0;
```

```
        le15 = 0;
```

Liite 2/3.

```
le16 = 0;
```

```
pc scanf("%s", str); //serial yhteyden skannus ja puskurin tallentaminen  
str-muuttujaan
```

```
cnt = sscanf(str, "%c%i%s", &c, &n, ext);
```

```
usb.printf("%s", str);
```

```
if (cnt < 2) continue; // hypätään silmukan lopun yli takaisin alkuun, jos ei  
saatu oikeaa tietoa
```

```
/*
```

```
#####
```

```
Raspberry-pi:ltä saapuu string puskurin joka on malli  
"a0b0c0d0e0f0g0h0i0j0" jokainen kirjain merkkää eri tietoa
```

```
a - Kotijoukkueen tulos/juoksut. 2 digittiä. Max. esitettävä numero "99"
```

```
b - Vierasjoukkueen tulos/juoksut. 2 digittiä. Max. esitettävä numero "99"
```

```
c - Palot. 1 Digitti. Max. esitettävä numero "9"
```

```
d - Kotijoukkueen jaksovoitot. 1 digitti. . Max. esitettävä numero "9"
```

```
e - Vierasjoukkueen jaksovoitot. 1 digitti. Max. esitettävä numero "9"
```

```
f - Pelattava jakso. 1 digitti. Max. esitettävä numero "9"
```

```
g - Vuorossa viellä käytettävissä olevien jokerien lukumäärä.1 digitti. Max.  
esitettävä numero "9"
```

```
h - Kyseisellä hetkellä lyöntivuorossa olevan pelaajan numero. 1 digitti.  
Max. esitettävä numero "9"
```

```
i - Vuoron viimeinen lyöjä. 1 digitti. Max. esitettävä numero "9"
```

```
j - Tällähetkellä pelattava vuoropari. 2 digittiä. Max. esitettävä numero "9 +  
"A" ja "L" ykköset digitissä"
```

```
#####
```

```
*/
```

Liite 2/4.

```
switch (c)
```

```
{
```

```
/*
```

```
#####
```

```
    Kotijuokkueen Tulosten esittäminen
```

```
##### */
```

```
    case 'a':
```

```
        setDigit(&le2, n / 10);
```

```
        setDigit(&le1, n % 10);
```

```
        break;
```

```
//#####
```

```
    //Vierasjuokkueen tulosten esittäminen
```

```
//#####
```

```
    case 'b':
```

```
        setDigit(&le5, n / 10);
```

```
        setDigit(&le4, n % 10);
```

```
        break;
```

```
//#####
```

```
    //Palojen Esittäminen
```

```
//#####
```

```
    case 'c':
```

Liite 2/5.

```
    setDigit(&le3, n % 10);
```

```
    break;
```

```
#####
```

```
    //Kotijuokkueen jaksovoittojen esittäminen
```

```
#####
```

```
    case 'd':
```

```
        setDigit(&le6, n % 10);
```

```
        break;
```

```
#####
```

```
    //Vierasjuokkueen jaksovoittojen esittäminen
```

```
#####
```

```
    case 'e':
```

```
        setDigit(&le7, n % 10);
```

```
        break;
```

```
#####
```

```
    //Pelattavan jakson esittäminen
```

```
#####
```

```
    case 'f':
```

```
        setDigit(&le8, n % 10);
```

```
        break;
```

Liite 2/6.

```
#####
```

```
//Jokerien esittäminen
```

```
#####
```

```
case 'g':
```

```
    setDigit(&le10, n % 10);
```

```
    break;
```

```
#####
```

```
//Vuorossa olevan lyöjän esittäminen
```

```
#####
```

```
case 'h':
```

```
    setDigit(&le9, n % 10);
```

```
    break;
```

```
#####
```

```
//Viimeisen lyöjän esittäminen
```

```
#####
```

```
case 'i':
```

```
    setDigit(&le11, n % 10);
```

```
    break;
```

```
/*#####
```

```
Vuoropari segmenttien esitys
```

```
#####*/
```

Liite 2/7.

```
case 'j':
    setDigit(&le12, n % 10);
    if (strstr(ext, "A"))
        setDigit(&le13, 4); // millä numerolla saadaan A-kirjain???? A = 4
    else
        setDigit(&le13, 7); // millä numerolla saadaan L-kirjain???? L = 7
    break;
}
}
}
/*
Hardwaren alustusfunktio
*/
void init(void)
{
    static DigitalInOut *leds[13] = { &le1, &le2, &le3, &le4, &le5, &le6, &le7,
&le8, &le9, &le10, &le11, &le12, &le13 };

    static DigitalInOut *ds[4] = { &d0, &d1, &d2, &d3 };

    int i;

    pc.baud(115200); //Serial yhteyden Baudraten alustaminen

    kello1.period(0.0000333); // f = 1 /T ==30khz
    kello1.write(0.8);
```

Liite 2/8.

```
kello2.period(0.0000333);
kello2.write(0.8); //kellojen taajuuksien ja duty cyclen alustaminen
kello3.period(0.0000333);
kello3.write(0.8);
kello4.period(0.0000333);
kello4.write(0.8);

for (i = 0; i < 13; i++)
    leds[i]->output(); // asetetaan ledit outputeiksi
for (i = 0; i < 4; i++)
    ds[i]->output(); // asetetaan datat outputeiksi
}

/*
Digitin kirjoitusfunktio
*/
void setDigit(DigitalInOut *dio, int val)
{
    static DigitalInOut *ds[4] = { &d0, &d1, &d2, &d3 };
    int i;
    *dio = 0;
    for (i = 0; i < 4; i++, val >>= 1)
        *ds[i] = val & 1;
    *dio = 1;
}
```

