



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

TAPIO DAHLMAN  
VERKOTETTU MAALITAU LUJEN PISTELASKU- JA HALLINTA-  
JÄRJESTELMÄ

Diplomityö

Tarkastaja: professori Timo Hämäläinen  
Tarkastaja ja aihe hyväksytty  
Tieto- ja sähkötekniikan tiedekunta-  
neuvoston kokouksessa 8. lokakuuta  
2014

## TIIVISTELMÄ

**DAHLMAN, TAPIO:** Verkotettu maalitaulujen pisteenlasku- ja hallintajärjestelmä

Tampereen teknillinen yliopisto

Diplomityö, 45 sivua, 4 liitesivua

Joulukuu 2014

Signaalinkäsittelyn ja tietoliikennetekniikan diplomi-insinöörin tutkinto-ohjelma

Pääaine: Sulautetut järjestelmät

Tarkastaja: professori Timo Hämäläinen

Avainsanat: verkotettu järjestelmä, WLAN

Tässä työssä kuvataan tuotekehitysprojekti, jonka tavoitteena oli toteuttaa verkotettu maalitaulujen pisteenlasku- ja hallintajärjestelmä muovikuula-aseille. Järjestelmän tarkoitus on toimia harrastus- ja harjoitteluvälineenä sekä tehdä tarkkuusammunnasta mielekkäämpää toteuttamalla automaattisen pisteenlaskun ja ohjattuja ammuntatoimintoja.

Työssä selvitetään ensin vastaavien järjestelmien kaupallinen tilanne, määritetään toteutettavan järjestelmän vaatimukset ja kuvataan järjestelmän suunnitteluvaiheet. Tämän jälkeen kuvataan järjestelmän käytännön toteutus sekä valmiiseen järjestelmään tehdyt testit ja mittaukset. Työn lopussa pohditaan järjestelmän jatkokehitysmahdollisuuksia.

Toteutettu järjestelmä koostuu maalitaulujen ohjainlaitteista, niitä ohjaavasta hallintalaitteesta sekä laitteiden välisestä radiolinkistä. Tauluohjainlaitteiden toteutusta varten suunniteltiin elektroninen kytkentä, jonka tärkeimpinä komponentteina on mikro-ohjain sekä radiomoduuli. Mikro-ohjaimen ohjelmisto toteutettiin C-kielellä Atmel Studio 6 – ohjelmistokehitysympäristöä käyttäen.

Hallintalaitteena käytettiin valmista laitetta, johon toteutettiin hallintaohjelmisto. Ohjelmisto toteutettiin Windows Phone –laitteille käyttäen Visual Studio 2013 – ohjelmistokehitysympäristöä. Ohjelmalogiikka toteutettiin C#- ja käyttöliittymä XAML-kielellä. Radiolinkin tyyppinä käytettiin langatonta lähiverkkoa.

Vaatimuksena järjestelmän vasteajalle ammutusta osumasta taulun kaatumiseen määritettiin 0,2 sekuntia ja osumatiedon päivittymiseen käyttöliittymän pistelaskuriin 0,5 sekuntia. Työssä toteutetun valmiin järjestelmän mittauksissa taulun kaatuminen tapahtui noin 0,10 sekuntia, ja pistelaskurin päivittyminen noin 0,16 sekuntia osuman jälkeen. Järjestelmälle asetetut tavoitteet saavutettiin siis hyvin.

## ABSTRACT

**DAHLMAN, TAPIO:** A Networked System for Target Scoring and Management

Tampere University of Technology

Master of Science Thesis, 45 pages, 4 Appendix pages

December 2014

Master's Degree Programme in Signal Processing and Communications Engineering

Major: Embedded systems

Examiner: professor Timo Hämäläinen

Keywords: networked system, WLAN

This thesis describes a development project, the goal of which was to implement a networked system for target scoring and management for airsoft guns. The system is designed to operate as a leisure and training tool and make precision shooting more meaningful by adding automatic scoring and controlled shooting functions.

First the commercial state of similar systems is investigated, system requirements are defined and the design process is described. Then the practical implementation and tests and measurements of the implemented system are detailed. Finally, the possibilities for further development are considered.

The system consists of target control devices, a target management device and a radio link. For the target control device an electronic circuit was designed containing a microcontroller and a radio module as the main components. The software for the microcontroller was developed with Atmel Studio 6 development environment using C-language.

A management software was designed for an existing commercial device. The software was developed for Windows Phone devices with Visual Studio 2013 development environment. The program logic were implemented using C# language and the user interface using XAML language. The radio link type used in the system is wireless LAN.

The response time requirements for the system were set to 0.2 seconds for the target toggle and 0.5 seconds for the score counter to update after a target hit. Measured response time for the first requirement was 0.10 seconds and for the second requirement 0.16 seconds. Hence, the goals set to the system were achieved.

## **ALKUSANAT**

Tässä diplomityössä tehty tuotekehitysprojekti tehtiin oman yritykseni Blue Particle Media ja Helsinkiläisen yrityksen PiQ-Wing yhteistyönä. Haluan kiittää PiQ-Wingin yrittäjiä yhteistyöstä projektin aikana. Kiitokset myös työn ohjaajalle, professori Timo Hämäläiselle. Lisäksi haluan kiittää Janne Kaarlaa järjestelmän testausavusta sekä Annina Päivärintaa tuesta projektin aikana.

Tampereella, 22.12.2014

Tapio Dahlman

## SISÄLLYSLUETTELO

1.	JOHDANTO .....	1
2.	JÄRJESTELMÄN TOIMINNOT JA VASTAAVAT KAUPALLISET TUOTTEET 2	
2.1	Järjestelmän käyttötarkoitus .....	2
2.2	Kaupallinen tilanne .....	3
2.3	Hintatavoitteet.....	4
3.	VAATIMUSMÄÄRITTELY .....	5
3.1	Järjestelmän vaatimukset ja tavoitteet .....	5
3.2	Tauluohjaimen vaatimukset.....	5
3.3	Hallintalaitteen ohjelmiston määrittely.....	6
3.4	Järjestelmän käyttöympäristö .....	6
3.5	Kommunikointi hallintalaitteen ja tauluohjaimen välillä .....	7
3.6	Järjestelmän suorituskyky.....	7
4.	ARKKITEHTUURISUUNNITTELU .....	8
4.1	Järjestelmän osat ja fyysiset lohkot .....	8
4.1.1	Anturointilohko maalitaulun osumatarkastelun .....	9
4.1.2	Tauluohjauslohko .....	10
4.1.3	Radiolinkin toteuttava lohko .....	11
4.1.4	Teholähde .....	13
4.1.5	Ohjainlohko .....	14
4.2	Työnjako järjestelmän osien välillä sekä rajapinnat .....	15
4.3	Tauluohjaimen ohjelmistosuunnittelu .....	16
4.4	Hallintalaitteen ohjelmiston kuvaus .....	22
4.4.1	Hallintalaitteen ohjelmistosuunnittelu.....	22
4.4.2	Hallintalaitteen käyttöliittymäsuunnittelu .....	26
5.	KÄYTÄNNÖN TOTEUTUS.....	29
5.1	Tauluohjaimen elektroniikkapiiri ja –komponentit .....	29
5.2	Tauluohjaimen ohjelmiston toteutus.....	32
5.3	Tauluohjaimen mekaniikka .....	32
5.4	Hallintalaitteen ohjelmiston toteutus .....	33
5.5	Tietoliikenneyhteyksien toteutus .....	35
5.6	Valmistuskustannukset .....	35
6.	KÄYTTÖTESTIT JA MITTAUKSET SEKÄ NIIDEN MERKITYS .....	37
6.1	Testikäyttö .....	37
6.2	Mittaukset .....	37
6.2.1	Vasteaika- ja nopeusmittaukset.....	38
6.2.2	Sarja-ammuntatesti.....	39
6.3	Mittaustulosten merkitys .....	40
7.	JATKOKEHITYS .....	42

7.1	Tauluohjaimen jatkokehitys.....	42
7.2	Hallintalaitteen ohjelmiston jatkokehitys .....	42
8.	YHTEENVETO .....	44
	LÄHTEET.....	45

LIITE A: KYTKENTÄKAAVIO

LIITE B: JOHDOTUSKUVAT

LIITE C: OSASIJOTTIELUKUVA

LIITE D: KOMPONENTTILUETTELO

## LYHENTEET JA MERKINNÄT

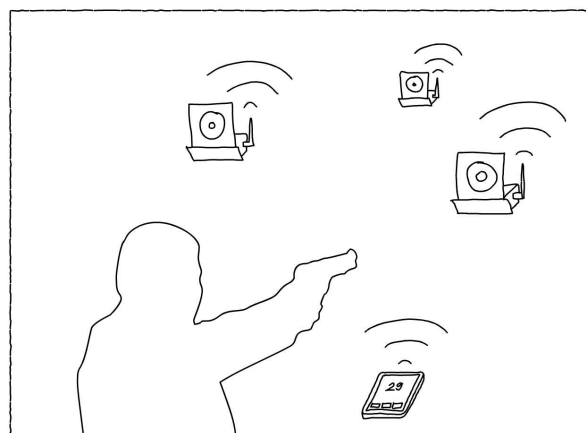
AD	engl. Analog to Digital, analogisesta digitaaliseen
ASCII	engl. American Standard Code for Information Interchange, tietokoneiden merkistö
BJT	engl. Bipolar Junction Transistor, bipolaaritransistori
EEPROM	engl. Electrically Erasable Programmable Read-Only Memory, haihtumaton muisti, jota pyyhitään ja luetaan elektronisesti
I/O	engl. In/Out, sisään- ja ulostulo
IP	engl. Internet Protocol, internetprotokolla
LCD	engl. Liquid Crystal Display, nestekidenäyttö
LED	engl. Light-Emitting Diode, diodi, joka säteilee valoa, kun sen läpi johdetaan sähkövirta, eli hohtodiodi
PWM	engl. Pulse Width Modulation, pulssinleveysmodulaatio
TCP	engl. Transmission Control Protocol, tietoliikenneprotokolla yhte-yksien muodostamiseen tietokoneiden välillä
UART	engl. Universal Asynchronous Transmitter/Receiver, asynkroninen sarjaliikennepiiri
WLAN	engl. Wireless Local Area Network, langaton lähiverkkotekniikka
WPS	engl. Wi-Fi Protected Setup, standardi, jonka tavoitteena tarjota helppo tapa suojata langaton lähiverkko
XAML	engl. Extensible Application Markup Language, merkintäkieli

# 1. JOHDANTO

Airsoft on peli, jossa pelaajat taistelevat käyttäen muovikuulia ampuvia aseita. Taiste- luissa pelaajat ampuvat aseilla toisiaan. Muovikuula-aseiden käyttö ei rajoitu kuiten- kaan pelkästään taistelupeleihin, vaan niitä voidaan käyttää myös tarkkuusammuntaan. Maalitauluihin ampuminen on helppo tapa harjoitella lajia taistelupelien ulkopuolella, sekä mahdollistaa aseiden käytön niille, jotka eivät taistelupelien harjoittamiseen halua ryhtyä.

Helsinkiläinen yritys PiQ-Wing oli saanut idean automaattisesti nousevan harjoittelu- taulun tekemiseen ja oli keskustellut asiasta lajin harrastajien kanssa. Ideasta kuultuaan harrastajat kannustivat laitteen toteuttamiseen, koska laite olisi hyödyllinen apu lajin harjoittelussa. Vuoden 2014 alussa yritys ryhtyikin suunnittelemaan maalitaulun proto- tyyppejä, joka saatiin valmiiksi kesään mennessä. Prototyypin toimintoina oli taulun au- tomaattinen kaataminen osumasta sekä ajastin, jolla määriteltiin aika ennen taulun ta- kaisin ylös nostoa.

Saman vuoden syyskuussa Pietarsaarelainen yritys Blue Particle Media tuli mukaan projektiin, ja päätti verkottaa maalitaulut sekä ohjata niitä erillisellä hallintalaitteella. Tärkein etu verkottamisessa oli, että järjestelmään voitaisiin toteuttaa automaattinen pisteenlasku osumista. Verkottaminen, pisteenlasku sekä hallintalaitteen käyttö edellytti maalitaulujen elektroniikan toteuttamista digitaalisesti, aiemman analogisen kytkennän sijaan. Tuotekehitysprojekti tehtiin diplomityönä ja on kuvattu tässä työssä. Työssä kes- kitytään järjestelmän elektroniikan sekä ohjelmiston kehittämiseen. Kuvassa 1.1 on ha- vainnollistettu järjestelmän käyttötarkoitus.



**Kuva 1.1** Havaintokuva järjestelmän käyttötarkoituksesta



## 2. JÄRJESTELMÄN TOIMINNOT JA VASTAAVAT KAUPALLISET TUOTTEET

Verkotetun maalitaulujen pistelasku- ja hallintajärjestelmän kehittämiseen ryhdyttiin, koska muovikuula-aseharrastajien kokemusten mukaan vastaavia järjestelmiä ei ole kaupallisesti saatavilla. Kehitettävän järjestelmän vaatimustasot määriteltiin harrastajien näkemysten ja toiveiden perusteella, sekä huomioiden kirjoitushetkellä markkinoilla olevien järjestelmien ominaisuuksia ja puutteita. Aliluvussa 2.1 kuvataan, mihin käyttötarkoitukseen järjestelmä toteutettiin ja mitä ongelmia se ratkaisee. Aliluvussa 2.2 käsitellään saatavilla olevia kaupallisia järjestelmiä sekä niiden ominaisuuksia.

### 2.1 Järjestelmän käyttötarkoitus

Järjestelmää käytetään muovikuula-aseilla ammuskelun harrastus- ja harjoitteluvälineenä. Käyttöön tarvitaan hallintalaite sekä yksi tai useampi maalitaululaite. Maalitaululaitteet asetetaan sopivalle etäisyydelle käyttäjistä ja ne paritetaan hallintalaitteen kanssa. Toteutettavan järjestelmän vaatimuksina on maalitaulujen ohjaaminen automaattisesti ennalta määrättyin ehdoin ja osumien laskemisen automatisointi. Sekä ohjausehtojen asettaminen että osumien lukumäärän lukeminen on pystyttävä tekemään erilliseltä hallintalaitteelta siten, että ammuntatilanne helpottuu perinteisen osumasta kaatuvien maalitaulujärjestelmien käyttöön nähden.

Maalitaulujen ohjaaminen tapahtuu ennalta määritettyjen hallintalaitteelle tallennettujen ohjelmien nk. ammuntasekvenssien mukaisesti. Ammuntasekvenssi sisältää esimerkiksi tiedon ammunnan kestosta, nousujen lukumäärästä sekä taulujen laskemiseen vaadittujen osumien lukumäärä. Sekvenssi käynnistetään hallintalaitteella, jonka jälkeen ammunta tapahtuu. Ammunnan tulos voidaan lukea hallintalaitteelta. Järjestelmää suunniteltaessa voidaan olettaa, että vain yksi käyttäjä ampuu maalitauluja kerrallaan. Osumia voi kuitenkin tulla yksittäiseen tauluun nopealla tahdilla käytettäessä sarja-ammunta-asetusta. Tällaisessa tilanteessa järjestelmä ei saa hukata osumia vaan sen on kyettävä pitämään lukua niiden määrästä muistissaan.

Tässä työssä toteutettava järjestelmä on vielä prototyypitasolla, mutta kuitenkin niin, että tuotteistaminen on mahdollista vain pienellä lisätyöllä. Tämän vuoksi suunnittelu on tehtävä toiminnoiltaan määritetyn vaatimustason mukaisesti.

## 2.2 Kaupallinen tilanne

Kirjoitushetkellä markkinoilla on maalitaulujärjestelmiä, jotka ovat toiminnallisuudeltaan huomattavasti suppeampia kuin toteutettava järjestelmä. Tarkastellaan nyt kaupallisia tuotteita työn kannalta olennaisten toiminnallisten ominaisuuksien ja niiden toteutusmenetelmien näkökulmasta. Näitä ominaisuuksia ovat osumien tunnistus, maalitaulujen esille ja piiloon kääntyminen sekä pisteenlaskuautomaatiikka.

Eräs kaupallisesti saatavilla oleva tuotetyyppi on järjestelmä, joka koostuu yksittäisestä geelipäällysteisestä maalitaulusta. Geelipinnoitteen tarkoitus on pysäyttää muovikuula osumakohtaan. Tällöin osumakohta voidaan havaita kuulan paikasta taululla. Esimerkkejä geelipäällysteistä maalitaulujärjestelmistä ovat valmistajan *Crosman* valmistamat *Airsoft Game Board Gel-Trap Target* sekä *Electronic Airsoft Target Game Board* [1]. Näistä jälkimmäinen toteuttaa pisteenlaskuautomaatiikan, joka näyttää pisteet nestekide- eli LCD-näytöllä (Liquid Crystal Display). Kuvassa 2.1 on esitetty geelipinnoitetut maalitaulut.



*Kuva 2.1 Kaupallisesti saatavilla olevia geelipäällysteisiä maalitauluja. Vasemmalla Airsoft Game Board Gel-Trap Target ja oikealla Electronic Airsoft Target Game Board. [1]*

Toinen kaupallisesti saatavilla oleva tuotetyyppi on mekaanisesti osumasta kaatuvat maalitaulut, jotka nousevat automaattisesti. Esimerkkejä tällaisista järjestelmistä ovat valmistajan *Crosman* valmistama *Airsoft Auto Reset Target* [1] sekä valmistajan *MT Company* valmistama *Electronic Auto-Reset Airsoft Gun Target* [2]. *Airsoft Auto Reset Target* koostuu kolmesta vierekkäisestä taulusta, ja toteuttaa taulujen noston mekaanisesti vasempaan tauluun ammuttaessa. *Electronic auto-reset airsoft gun target* koostuu myös kolmesta vierekkäisestä taulusta, mutta toteuttaa taulujen noston sähköisesti paristojen avulla. Kuvassa 2.2 on esitetty mekaanisesti osumasta kaatuvat maalitaulujärjestelmät.



**Kuva 2.2** Kaupallisesti saatavilla olevia mekaanisesti osumasta kaatuvia maalitauluja. Vasemmalla Airsoft Auto Reset Target [1] ja oikealla Electronic Auto-Reset Airsoft Gun Target. [2]

Markkinoilla on saatavilla myös edistyneimpiä järjestelmiä, mutta niitä ei ole suunniteltu muovikuula-aseille, vaan muille ilma-aseille. Taulukossa 1 on luetteloitu yhteenvetona kaupallisia tuotteita ja niiden ominaisuuksia.

**Taulukko 1.** Kaupallisia muovikuula-aseille tarkoitettuja maalitaulujärjestelmiä [1], [2].

Järjestelmä	Antu- rointi	Kääntö- mekanismi	Pisteenlasku- automaatiikka	Hinta
Electronic Airsoft Target Game Board	geeli	-	LCD-näyttö	\$43,99
Airsoft Game Board Gel-Trap Target	geeli	-	-	\$15,33
Airsoft Auto Reset Target	-	mekaaninen	-	\$12,99
Electronic Auto-Reset Airsoft Gun Target	-	sähkömoottori	-	-

### 2.3 Hintatavoitteet

Hintatavoitteeksi suunniteltavalle järjestelmälle määriteltiin, että yksittäinen maalitauluohjain tulisi maksamaan kuluttajalle noin 80-120 euroa. Tavoitteelliset maalitauluohjaimen materiaali- ja komponenttikustannukset ovat noin 30-40 euroa. Hintatavoitteet on kuitenkin määritelty lopullista tuotetta varten, eikä tavoitteeseen pääseminen ole vielä tässä työssä toteutettavan järjestelmän kannalta kriittistä. Suunnittelu on kuitenkin pyrittävä tekemään siten, että hintatavoitteisiin olisi mahdollista päästä muuttamatta toteutusta suuresti ennen kaupallisten erien valmistamista.

## 3. VAATIMUSMÄÄRITTELY

Tässä luvussa kuvataan työssä toteutettavan järjestelmän vaatimuksia ja tavoitteita. Luvussa 3.1 käsitellään järjestelmän yleisiä vaatimuksia. Luvussa 3.2 tauluohjaimen vaatimuksia ja luvussa 3.3 hallintalaitteen ohjelmiston vaatimuksia

### 3.1 Järjestelmän vaatimukset ja tavoitteet

Maalitauluohjainlaitteet eli tauluohjaimet tulee voida parittaa hallintalaitteen kanssa siten, että järjestelmän osat eivät ole riippuvaisia toisistaan ja käytettävien tauluohjainten lukumäärää voidaan vaihdella eri käyttökertojen välillä. Myös hallintalaitte on voitava vaihtaa helposti toiseen. Vähimmäisvaatimukseksi yhdellä hallintalaitteella käytettävien tauluohjainten maksimilukumäärälle asetettiin 20 kappaletta.

Hallintalaitteena tulee voida käyttää jotain kohtuullisella hinnalla saatavilla olevaa valmista laitetta, esimerkiksi älypuhelin tai tablettitietokonetta. Hallintalaitteelle on voitava asentaa järjestelmän käyttämiseen tarkoitettu pistelasku- ja hallintaohjelmisto ilmaiseksi siten, että asentaminen on helppoa eikä vaadi erityisosaamista.

### 3.2 Tauluohjaimen vaatimukset

Yksittäisen tauluohjaimen tärkein tehtävä on taulun tarkkailu siihen tulleiden osumien havaitsemiseksi sekä taulun nostaminen ja laskeminen. Lisäksi tauluohjain kommunikoi järjestelmän hallintalaitteen kanssa toteuttaen tämän antamia komentoja.

Tauluohjaimen elektroniikalle määriteltiin vaatimukseksi toimivuus 6 voltin jännitteellä. Tämä päätös tehtiin siksi, että taulun kääntäminen toteutetaan servolaitteella, joka toimii 4,8 - 6,0 voltin jännitteellä, ja käytettäväksi haluttiin neljä 1,5 voltin AA-paristoa. Samaa ratkaisua oli käytetty edellisen tauluohjainlaitteen toteutuksessa. Tauluohjaimen toiminnot asettavat elektroniikan vaatimuksiksi myös ohjelmoitavuuden sekä radiolähttimen, anturointiin ja servo-ohjainpiirin sisällyttämisen kytkentään.

Muita tauluohjaimen elektroniikalle asetettuja vaatimuksia on uudelleenohjelmoinnin mahdollisuus sekä mahdollisuus liittää vähintään yksi ulkoinen näppäin ja kolme ulkoista LED-valoa (engl. Light Emitting Diode). Fyysisiä rajoituksia tauluohjaimen elektroniikalle ei määritetty. Laitteen lopullinen kotelo suunnitellaan elektroniikan koon ja muiden vaatimusten, kuten ulkoisten näppäinten ja LED-valojen määrän sekä antennin tyypin mukaan.

Suurin osa järjestelmän ylläpitämästä datasta tallennetaan hallintalaitteen ohjelmistoon. Tauluohjaimen ohjelmisto vertailee anturiltaan saatua dataa johonkin referenssidataan, jonka perusteella voidaan päätellä, onko osuminen maalitauluun sattunut. Tämä referenssidata on kuitenkin pidettävä tauluohjaimen haihtumattomassa muistissa.

### 3.3 Hallintalaitteen ohjelmiston määrittely

Hallintalaitteen ohjelmiston tehtävä on hallita tauluohjaimia sekä tarjota käyttäjälle ohjattuja ammuttoimintoja. Tauluohjainten hallintaan liittyy yksittäisen tauluohjaimen nimeäminen, kalibrointi ja testaaminen. Ammuttoimintoja on kahdenlaisia, vapaa ammunta sekä ammutasekvenssit joita käynnistämällä käyttäjä voi suorittaa valitsemansa ennalta määrätyn ammuntaohjelman.

Vapaa-ammuttoiminnon tarkoitus on tarjota helppo tapa harjoitella ampumista. Ammunta on jatkuva toiminto, jossa maalitauluja nostetaan ja lasketaan ennalta määritettävissä olevien asetusten perusteella. Näitä asetuksia ovat:

- Aika, jonka yksittäinen maalitaulu pysyy piilossa sen piilottamisen jälkeen
- Aika, jonka yksittäinen maalitaulu pysyy esillä sen esille kääntämisen jälkeen
- Piilottamiseen vaadittu osumien lukumäärä

Vapaa-ammuttoiminnossa maalitauluihin ammunnan aikana tapahtuneet osumat lasketaan pistelaskuriin.

Sekvenssiammunnan tarkoitus on tarjota käyttäjälle pelimäinen tapa kilpailla muiden käyttäjien kanssa tai haastaa itseään ennätysten tekemiseen. Sekvenssiammunnan asetuksia ovat vapaan ammunnan asetusten lisäksi:

- Vaihtoehdot ajastettu ammunta tai määrätty maalitaulujen näyttöjen lukumäärä
- Ammunta-ajan tai maalitaulujen näyttöjen lukumäärän asetukset

Valittaessa ajastettu ammunta, ammunnan kesto määritetään vakioksi, jolloin käyttäjä voi asettaa tavoitteekseen ampua mahdollisimman paljon osumia maalitauluihin ammunnan aikana. Kun taas maalitaulujen näyttöjen lukumäärä valitaan vakioksi määritettäväksi, käyttäjä voi pyrkiä suorittamaan ammunta mahdollisimman pienessä ajassa. Sekvenssiammuttoiminto tarjoaa pistelaskurin lisäksi ajan näytön. Ammutasekvenssin alkaminen ja loppuminen indikoidaan äänimerkillä.

Hallintalaitteen ohjelmisto tarjoaa lisäksi toiminnot yksittäisen tauluohjaimen kalibroimiseen. Hallintalaitteen ohjelmiston käyttöliittymä toteutetaan englannin kielellä.

### 3.4 Järjestelmän käyttöympäristö

Järjestelmän käyttö tapahtuu sisätiloissa avarassa tilassa, joten käyttöympäristö voidaan olettaa kuivaksi ja lämpötilan olevan nollan celsiusasteen yläpuolella. Hallintalaitteen

voidaan olettaa olevan alle 30 metrin päässä maalitaululaitteista. Lisäksi niiden välillä ei ole suuria esteitä, kuten seiniä.

### 3.5 Kommunikointi hallintalaitteen ja tauluohjaimen välillä

Hallintalaitteen tulee kyetä valitulla kommunikointimenetelmällä kommunikoida langattomasti jopa 20 tauluohjainlaitteen kanssa. Kommunikointiyhteys on pidettävä järjestelmän normaalissa käyttötilanteessa jatkuvasti auki siten, että datan lähettäminen tapahtuu nopeasti ja luotettavasti. Hallintalaitteen ja tauluohjaimen välinen kommunikointi on kaksisuuntaista. Lähetettävän datan määrä on kuitenkin pieni.

### 3.6 Järjestelmän suorituskyky

Järjestelmän osat tulee kyetä kommunikoimaan toistensa kanssa kohtuullisilla vasteajoilla. Vasteajat ovat kriittisimpiä käyttömukavuuden takaamiseksi toiminnoissa, joissa järjestelmän halutaan reagoivan johonkin fyysiseen herätteeseen. Tällainen heräte on esimerkiksi osuma tauluun, jonka jälkeen taulun tulisi kääntyä piiloon. Tavoitteena oli, että kääntyminen tapahtuu niin nopeasti, että osuma näyttäisi kaatavan maalitaulun, vaikka todellisuudessa se tehdään sähköisesti. Vasteaikatavoitteeksi maalitaulun kaatumiselle osumasta määriteltiin 0,2 sekuntia. Pistelaskurin vaste-ajat eivät ole järjestelmän käytettävyyden kannalta yhtä kriittiset. Tavoitteeksi vaste-ajalle määritettiin 0,5 sekuntia. Taulukossa 2 on listattu järjestelmän eri toimintojen tavoitteelliset vasteajat yhden tauluohjaimen osalta.

*Taulukko 2. Järjestelmän vasteajat*

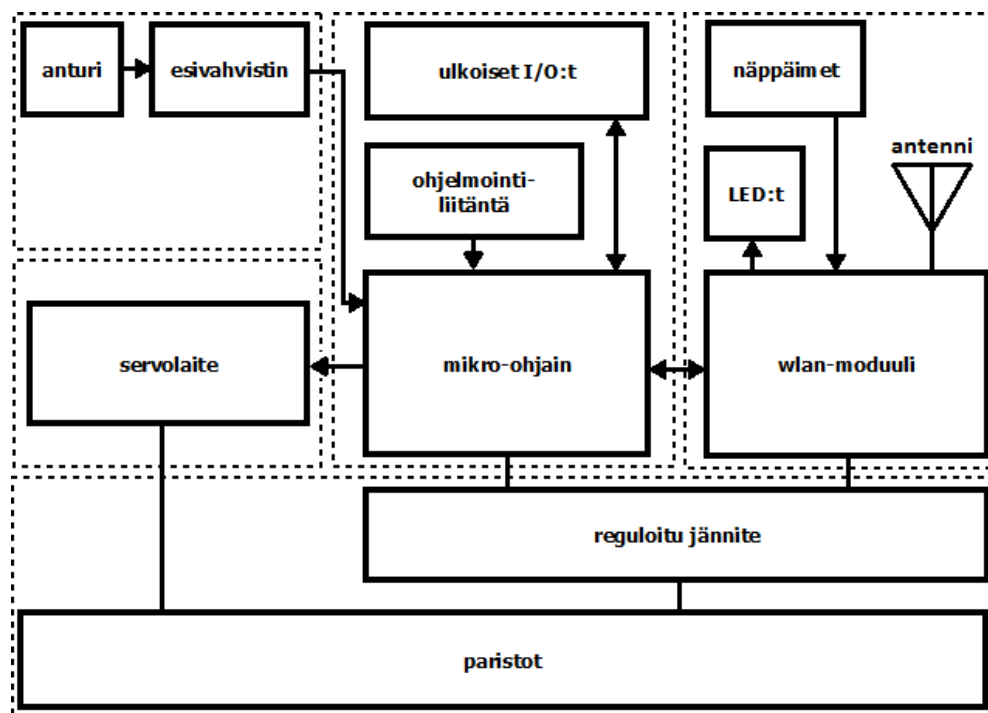
Toiminto	Vasteaikatavoite [s]
Maalitaulun nosto hallintalaitteen komennosta	< 0,1
Maalitaulun piilotus hallintalaitteen komennosta	< 0,1
Maalitaulun kaatuminen osumasta	< 0,2
Osumatiedon päivittyminen pistelaskuriin	< 0,5
Tauluohjaimen paritus hallintalaitteen kanssa	< 10,0

## 4. ARKKITEHTUURISUUNNITTELU

Tässä luvussa kuvataan järjestelmän jako fyysisiin ja ohjelmallisiin osiin sekä kuvataan osien väliset riippuvuudet sekä rajapinnat. Luvussa 4.1 kuvataan maalitaululaitteiston lohkot sekä niiden tehtävät ja luvussa 4.2 tehtävänjako tauluohjainten ja hallintalaitteen välillä. Luvussa 4.3 kuvataan maalitaululaitteiston ohjelmiston ja luvussa 4.4 hallintalaitteen ohjelmiston rakenne.

### 4.1 Järjestelmän osat ja fyysiset lohkot

Toteutettavasta järjestelmästä voidaan erottaa kolme pääosaa, tauluohjaimet, hallintalaitte sekä radiolinkki. Tauluohjaimien elektronikasta voidaan edelleen erottaa eri toiminnon toteuttavat lohkot. Näitä ovat maalitaulun anturointilohko, tauluohjainlohko, radiolinkin toteuttava lohko, ohjainlohko sekä tehollähde. Näiden lohkojen tehtävät sekä toiminnallinen toteutus on kuvattu luvuissa 4.1.1–4.1.5. Kuvassa 4.1 on esitelty tauluohjaimen lohkokaavio.



*Kuva 4.1 Tauluohjaimen elektronikan lohkokaavio*

Hallintalaitteena käytetään valmista laitetta, joten sen osalta työssä keskitytään vain siihen toteutettavaan hallintaohjelmistoon. Kappaleessa 4.3 kuvataan hallintaohjelmiston suunnittelu. Radiolinkin tyyppi riippuu tauluohjaimelle valitusta radiolähtimestä.

### 4.1.1 Anturointilohko maalitaulun osumatarkastelun

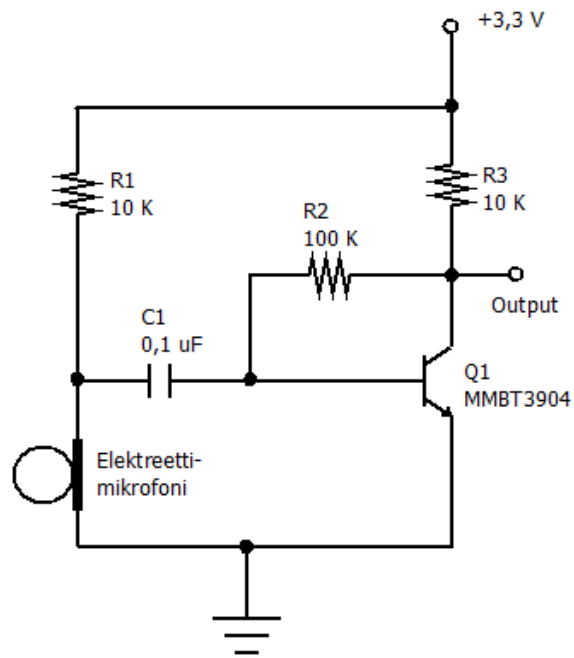
Anturointilohkon tehtävänä on toteuttaa elektroninen piiri järjestelmän maalitaulujen tarkkailuun. Lohkon sisäänmenona on mitattavissa oleva fysikaalinen suure, jossa tapahtuu muutos osuman sattuessa maalitauluun. Ulostulona on sähköinen signaali, jota analysoimalla voidaan todeta osuman tapahtuneen.

Osumatarkastelun anturointia valittaessa rajoittavana tekijänä oli maalitaulun materiaali. Koska prototyyppivaiheessa taulutyypiksi oli valittu perinteinen pahvitaulu, päädyttiin harkitsemaan kahta erilaista anturointitapaa, värinään ja ääneen perustuvaa. Värinään perustuvan anturoinnin etuna olisi osuman havaitsemisen yksiselitteisyys. Jos maalitaulun voidaan todeta värähtäneen, on osuma tapahtunut. Värinään perustuvan anturoinnin haittana voidaan kuitenkin pitää sitä, että anturi on sijoitettava liikkuvaan osaan, taulua liikuttavaan varteen. Lisäksi värinään perustuva anturointi ei toimisi luotettavasti ulkotiloissa, koska se olisi altis tuulen vaikutukselle. Tämä ei kuitenkaan olisi este, koska järjestelmää määriteltäessä ulkokäytön mahdollisuutta ei pidetty vaatimuksena.

Ääneen perustuvan anturoinnin etuna oli se, että sitä oli käytetty jo aiemman maalitauluohjainversion anturointina hyvin tuloksin. Anturityyppinä käytössä oli ollut edullinen mikrofoni. Jokainen osuma maalitauluun tuottaa hyvin samankaltaisen ja -vahvuisen äänen. Näin osumatarkastelua tehdessä voidaan osumääni erottaa hyvin muista ympäristön äänistä jo äänen voimakkuuden perusteella kun mikrofoni asetetaan tarpeeksi lähelle maalitaulua.

Osumatarkastelun anturiksi valittiin elektreettimikrofoni. Elektreettimikrofoni on edullinen, yleisesti esimerkiksi matkapuhelimissa käytetty mikrofonityyppi, jonka tuottama signaali on millivolttiluokkaa [3]. Signaalin pienen amplitudin vuoksi, toteutettuun kytkentään suunniteltiin esivahvistin. Esivahvistimeksi valittiin yksinkertainen yhden BJT-transistorin (engl. Bipolar Junction Transistor) lisäksi vain muutaman passiivikomponentin vaatima kytkentä [4]. Työssä käytetty esivahvistinkytkentä on esitetty kuvassa 4.2. Mikrofonin tuottama analoginen signaali muutetaan digitaaliseen muotoon AD-muuntimella (engl. Analog to Digital).



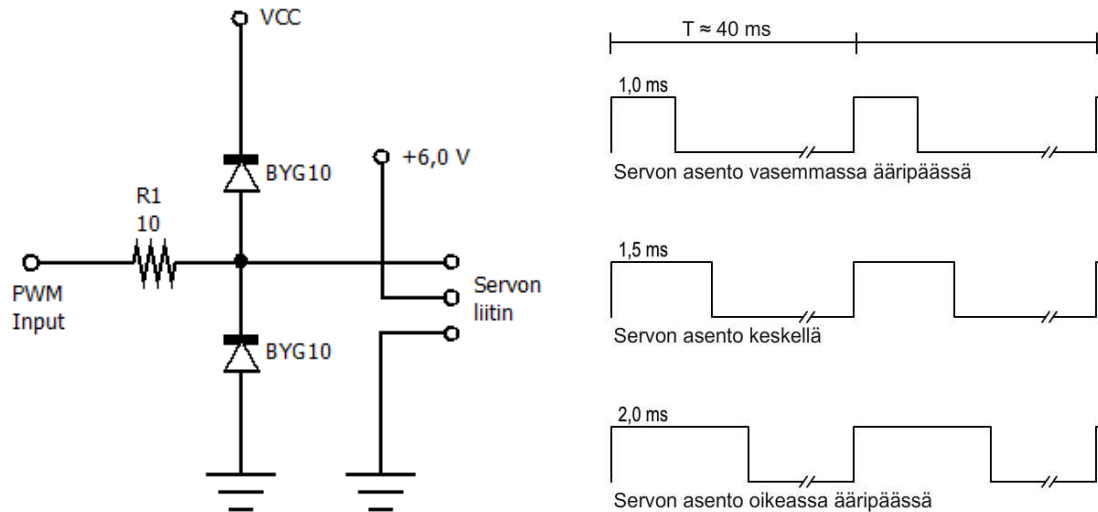


*Kuva 4.2 Mikrofonin esivahvistinkytkentä*

#### 4.1.2 Tauluohjauslohko

Tauluohjauslohkon tehtävänä on nostaa maalitaulu näkyviin ja laskea se näkymättömiin. Lohkon sisäänmenona on ohjaussignaali, jonka perusteella valitaan maalitaulun asento. Ulostulona on maalitaulun fyysikaalinen liike näkyvästä tilasta näkymättömään.

Taulun liikuttamiseen valittiin analoginen servo, koska sellaista oli käytetty maalitauluohjaimen aikaisemmassa versiossa. Servo on laite, jolla voidaan toteuttaa fyysinen kiertoliike pulssisuhdemoduloidun signaalin eli PWM-signaalin (engl. Pulse Width Modulation) avulla. Tavallisimmin servo kykenee liikkumaan 90-180 astetta. Ohjaussignaalin pulssisuhde määrää servon asennon. Signaalin taajuuden tulee olla noin 25 hertsiä. Kuvassa 4.3 on selitetty servon ohjaussignaalin vaikutus servon asentoon. [5]



**Kuva 4.3** Servon kytkentä ja ohjaukseen käytettävä PWM-signaali

Servon käyttöjännite on 4,8 – 6,0 voltia, joten sen jännite otetaan kytkennässä suoraan paristoista. Ohjaussignaalin jännite saa vaihdella välillä 3,0 – 5,0 voltia, joten suoraan mikro-ohjaimelta saatu PWM-signaali on toimiva [5].

### 4.1.3 Radiolinkin toteuttava lohko

Radiolinkin toteuttavan lohkon tehtävä on hoitaa kommunikointi pisteenlasku- ja hallintaohjelmistoa ajavan hallintalaitteen ja tauluohjainten ohjainlohkon välillä. Kommunikointi on kaksisuuntaista. Lohkon sisäänmenona voi olla radiosignaalin hallintalaitteelta tuleva data, jolloin ulostulona on sähköinen datasiignaali, tai sähköinen datasiignaali ohjainlohkolta, jolloin ulostulona on data radiosignaalin muodossa.

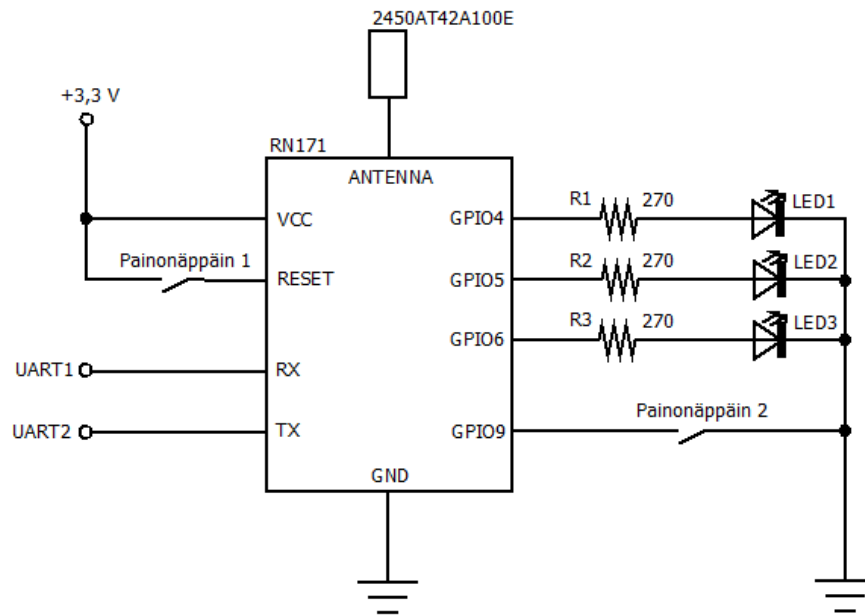
Radiolinkin toteutustekniikan valinnassa harkittavia vaihtoehtoja oli useita. Projektin alkuvaiheessa radiolinkin toteutustekniikaksi harkittiin Bluetooth-, langaton lähiverkko eli WLAN- (engl. Wireless Local Area Network) ja ZigBee-tekniikkaa sekä protokollatonta digitaalista lähetystä. Ajatuksena oli, että maalitaulut voitaisiin verkottaa käyttämällä erillistä isäntä-laitetta (engl. master device), joka hoitaisi tauluohjainten hallinnan ja pisteenlaskun. Hallintalaitteella voitaisiin tällöin ottaa yhteys jollain laitteen tukemalla radiolinkkityypillä pelkästään isäntä-laitteeseen, joka tarjoaisi tietoa tauluohjainten tilasta. Isäntälaitteeksi olisi voinut olla tarkoitukseen suunniteltu tauluohjain tai erillinen ”kapulamainen” lisälaitte. Kolmanneksi vaihtoehtoksi isäntälaitteen tyyppiä mietittiin monimutkaisempaa laitetta, joka myös toteuttaisi järjestelmän käyttöliittymän ja toimisi hallintalaitteena. Laitteen valmistaminen olisi kuitenkin ollut kohtuuttoman työläs. Muutenkin erillisen isäntälaitteen toteuttaminen järjestelmään arvioitiin turhaan kasvattavan järjestelmän monimutkaisuutta sekä vaikeuttavan sen käyttöä. Hallintalaitteena päätettiin käyttää valmista kannettavaa laitetta, älypuhelin tai tietokonetta. Näin radiolinkin toteutustekniikan vaihtoehtoiksi jäi kaupallisten laitteiden yleisesti tukemat Bluetooth- ja WLAN-tekniikat.

Piirikytkentään liitettävät Bluetooth-moduulit ovat WLAN-moduuleja edullisempia. Niiden virrankulutus on myös keskimäärin WLAN-moduulien kulutusta pienempi. Molempien radiotekniikoiden kantomatkat ovat riittävän suuret sisäkäyttöön suunniteltavalle järjestelmälle. Ratkaisevaksi ominaisuudeksi jäi hallintalaitteen kanssa samanaikaisesti kommunikoitavissa olevien laitteiden lukumäärä. Järjestelmän toiminta edellyttää jatkuvaa kommunikointia hallintalaitteen ja tauluohjainten välillä. Bluetooth-tekniikka vaatii kommunikoiden laitteiden parittamisen toistensa kanssa. Samanaikaisesti paritettuna olevien laitteiden lukumäärä on kuitenkin rajoitettu. Bluetooth 4.0 -standardin mukaan laite voidaan pitää paritettuna maksimissaan seitsemän laitteen kanssa samanaikaisesti [6]. Lisäksi hallintalaitteen valmistaja voi ohjelmistossaan määritellä vielä tiukempia rajoituksia.

WLAN-tekniikkaa käytettäessä laitteita ei pariteta suoraan toistensa kanssa, vaan kommunikointi tapahtuu tukiaseman välityksellä (engl. access point). Tällöin riittää, että laitteet on yhdistetty samaan tukiasemaan, jona käytännössä toimii WLAN-reititin. Koska järjestelmää määriteltäessä tauluohjaimia haluttiin voida käyttää samanaikaisesti jopa 20 kappaletta, tekniikaksi valittiin WLAN.

WLAN-moduulia valittaessa huomiota kiinnitettiin komponentin hintaan sekä sen helpokäyttöisyyteen. Hallintalaitteelta tauluohjaimelle lähetettävän datan määrä on hyvin pieni. Lisäksi tauluohjaimen toteuttavat toiminnot radiokommunikaation lisäksi ovat yksinkertaisia, joten mikro-ohjaimen vaatimukset ovat vähäiset sekä datan määrän että laskentatehon suhteen. Näin ollen WLAN-moduuliksi päätettiin valita sellainen, joka toteuttaa mahdollisimman suuren osan WLAN-yhteyden vaatimasta työstä sisäisesti. Moduuliksi valittiin valmistajan *Microchip Technology Inc* RN171-moduuli. Moduuli toteuttaa tietoliikenne- ja internetprotokollat (engl. Transmission Control Protocol/Internet Protocol, TCP/IP) sisäisesti ja toteuttaa piirin universaaliin asynkroniseen datan lähettämiseen ja vastaanottoon (engl. Universal Asynchronous Receiver/Transmitter, UART) radioteitse lähetettävän ja siitä vastaanotetun datan kaittamiseen mikro-ohjaimelle [7].

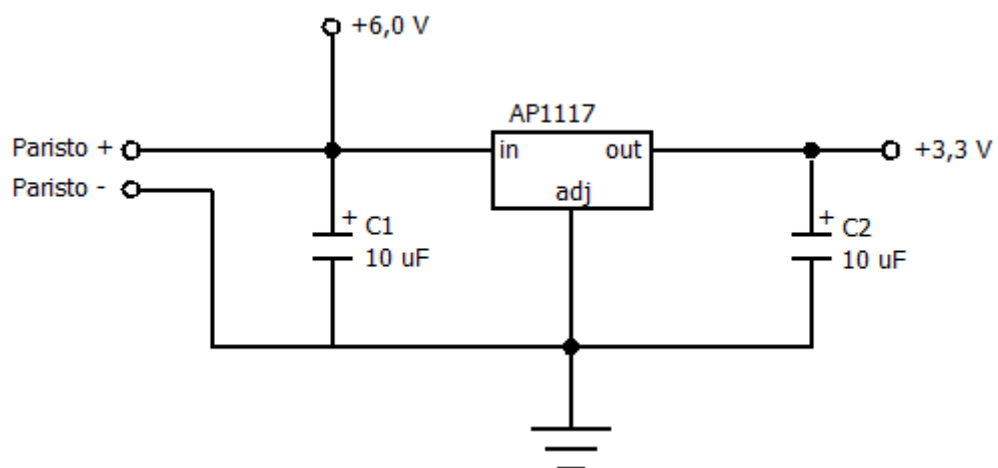
RN171-moduuli voidaan liittää piirikytkentään yksinkertaisimmillaan viemällä mikro-ohjaimelle kaksi johdinta, lähetys- ja vastaanotto-johtimet sekä tarjoamalla sille käyttöjännite ja maataso. Lisäksi se tarvitsee antennin. Antenniksi valittiin valmistajan *Johanson Technology* siruantenni 2450AT42A100E. Tauluohjaimen kytkennässä RN171-moduulille liitettiin vielä kolme LED-valoa radioyhteyden tilan näyttämiseen, sekä painonäppäimet moduulin nollaamiseen sekä web-sovelluksen käynnistämiseen. Kuvassa 4.4 on esitelty radiolinkin toteuttavan lohkon piirikytkentäkaavio.



*Kuva 4.4 Radiolinkin toteuttavan lohkon kytkentä*

#### 4.1.4 Teholähde

Teholähdelohkon tehtävänä on tuottaa maalitauluohjaimen laitteiston vaatimat käyttöjännitteet. Teholähteenä käytettiin neljää AA-paristoa, joista voidaan suoraan syöttää tehoa servolle. WLAN-moduuli vaatii käyttöjännitteeksi 3,3 voltia. Myös prosessori voi toimia tällä jännitteellä, joten paristojen jännitteestä saatiin käyttöjännite molemmille käyttämällä 3,3 voltin jänniteregulaattoria. Kuvassa 4.5 on esitelty teholähdelohkon piirikytkentäkaavio.



*Kuva 4.5 Teholähdelohkon kytkentä*

### 4.1.5 Ohjainlohko

Tauluohjaimen ohjainlohko toteuttaa taulujen osumatarkkailun sekä nostamisen ja laskeamisen ja kommunikoi hallintalaitteen kanssa. Nämä toteutetaan käyttämällä edellä esiteltyjen lohkojen ulostuloja ja sisäänmenoja. Ohjainlohko toteuttaa tauluohjaimen toiminnot digitaalisesti mikro-ohjaimen avulla. Mikro-ohjaimen valinta määräytyi tauluohjaimen lohkojen vaatimuksista sekä siihen toteutettavan ohjelmiston monimutkaisuudesta. Piirilevyille päätettiin toteuttaa kaksi LED-valoa ja näppäin ohjelmiston testaamiseen. Lisäksi tauluohjaimen haluttiin toteuttaa mahdollisuus ulkoisten lisälaitteiden, kuten näppäinten ja LED-valojen käyttämiseen. Taulukossa 3 on esitelty prosessorin valintaan vaikuttaneet tekijät, kuten vaaditut erikoistoiminnot sekä tarvittavien sisäänmeno- ja ulostulo-, eli I/O-pinnien (engl. In/Out) määrä.

*Taulukko 3. Prosessorin valintaan vaikuttaneet ominaisuudet*

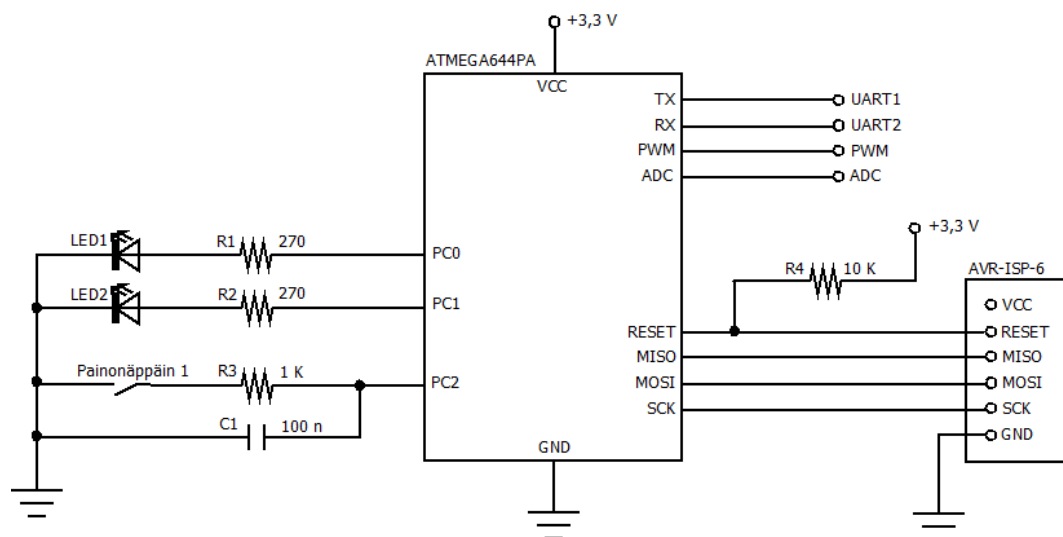
Toiminto tai ominaisuus	Toiminto	Tarvittavat I/O-pinnit	Arvo
Käyttöjännite			3,3 V
Ulkoiset näppäimet (2 kpl)		2	
Ulkoiset LED-valot (5 kpl)		5	
Piirilevyn LED-valot (2 kpl)		2	
Piirilevyn näppäimet (1 kpl)		1	
Servon ohjaus	PWM	1	
Radiolähettimen kommunikointi	UART	2	
Anturin lukeminen	ADC	1	

Tarvittavat erikoistoiminnot (PWM, UART ja AD-muunnin) on sisäänrakennettuna useimmissa mikro-ohjaimissa. Oleellisempaa valintaa tehdessä oli huomioida I/O-pinnien määrä sekä arvio tarvittavasta ohjelmamuistin määrästä. Mikro-ohjaimeksi valittiin valmistajan *Atmel* valmistama Atmega644PA. Mikro-ohjaimen valmistaja ja tuotesarja valittiin aiemman käyttökokemuksen perusteella, sekä siksi, että käytössä oli laitteen ohjelmointia varten tarvittava ohjelmointilaite. Ohjaimen versio valittiin I/O-porttien lukumäärän ja ohjelmamuistin määrän perusteella. Taulukossa 4 on esitetty kyseisen komponentin keskeiset ominaisuudet.

**Taulukko 4.** Valitun mikro-ohjaimen keskeisimmät ominaisuudet

Ominaisuus	Arvo
Käyttöjännitealue	2,7 – 5,5 V
Suurin kellotaajuus	20 MHz
I/O-pinnien lukumäärä	32
Pinnien lukumäärä	44
Kotelotyyppi	TQFP
Ohjelmamuistin koko	64 kilotavua
Datamuistin koko (EEPROM)	2 kilotavua
AD-muuntimen resoluutio	10 bittiä
Datarajapinnat	I2C, SPI, USART
PWM-kanavien määrä	6

Ohjainlohkon kytkentä sisältää mikro-ohjaimen lisäksi kaksi LED-valoa, painonäppäimen ja ohjelmointiliitännän sekä niiden kytkennässä käytetyt passiivikomponentit. Kuvasssa 4.6 on esitetty ohjainlohkon kytkennän hieman yksinkertaistettu piirikaavio.

**Kuva 4.6** Ohjainlohkon kytkentä

## 4.2 Työnjako järjestelmän osien välillä sekä rajapinnat

Suunniteltaessa työnjakoa hallintalaitteen ja tauluohjainten välillä, päätettiin tauluohjaimen ohjelmisto pitää niin yksinkertaisena, kuin mahdollista ja toteuttaa kaikki mahdolliset järjestelmän toiminnot hallintalaitteen ohjelmistoon. Syitä tähän päätökseen oli monia. Koska tauluohjaimen ohjelmiston tulee jatkuvasti tarkkailla maalitaulua osuimien havaitsemiseksi luotettavasti, ohjata maalitaulua ohjaavaa servoa sekä olla valmiina vastaanottamaan tietoa hallintalaitteelta, pyrittiin minimoida sen muut tehtävät. Mitä pienemmästä määrästä tehtäviä tauluohjaimen mikro-ohjain vastaa, sitä pienempi on sen tehon tarve ja siksi myös tehtävään valittavan mikro-ohjaimen hinta. Lisäksi, jos tauluohjaimen ohjelmisto on yksinkertainen ja toteuttaa vain selvät fyysisen laitteiston vaa-

timat toimenpiteet, voidaan järjestelmään helpommin toteuttaa lisäominaisuuksia myöhemmin päivittämällä pelkästään hallintalaitteen ohjelmistoa.

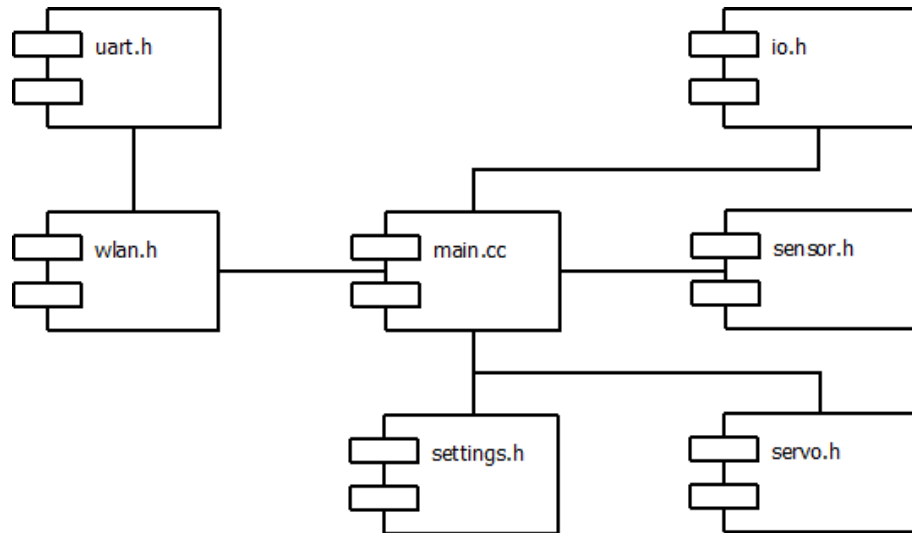
Hallintalaitteen ja tauluohjainten väliseen kommunikointiin määriteltiin yksinkertaisia komentoja toimintojen suorittamiseen ja tiedon siirtämiseen. Koska tauluohjaimen vastualue tehtävien suorittamisessa pidettiin pienenä, jäi myös lähetettävien komentojen määrä pieneksi. Komennot ja data lähetetään WLAN-verkossa TCP/IP-paketteina. Sekä hallintalaitteen käyttöjärjestelmän palvelut että tauluohjaimen WLAN-moduuli hoitavat kommunikoinnin edellyttämien protokollien toteuttamisen, voitiin tässä työssä keskittyä pelkästään itse paketin dataosaan, tässä tapauksessa ASCII-muotoiseen (eng. American Standard Code for Information Interchange) merkkijonoon. Komentoina päätettiin käyttää kolmen merkin pituisia merkkijonoja. Lisäksi komennon eteen päätettiin lisätä symboli '#' komennon havaitsemisen helpottamiseksi sekä hallintalaitteen että tauluohjaimen ohjelmistossa. Taulukossa 5 on esitetty kommunikointiin käytetyt komennot sekä niiden merkitykset.

**Taulukko 5.** Hallintalaitteen ja tauluohjaimen kommunikoinnin toteuttavat komennot

Komento	Lähetäjä	Vastaanottaja	Kuvaus
#TGH	Tauluohjain	Hallintalaite	Maalitaulu saanut osuman ( <i>target hit</i> )
#SDD	Tauluohjain	Hallintalaite	Anturi reagoiut ( <i>sound detected</i> )
#STU	Hallintalaite	Tauluohjain	Näytä maalitaulu ( <i>set target up</i> )
#STD	Hallintalaite	Tauluohjain	Piilota maalitaulu ( <i>set target down</i> )
#CRD	Hallintalaite	Tauluohjain	Komento vastaanotettu ( <i>command received</i> )
#ECM	Hallintalaite	Tauluohjain	Aloita kalibrointi ( <i>enter calibration mode</i> )
#XCM	Hallintalaite	Tauluohjain	Lopeta kalibrointi ( <i>exit calibration mode</i> )
#THT	Hallintalaite	Tauluohjain	Maalitaulu sai osuman ( <i>target hit true</i> )
#THF	Hallintalaite	Tauluohjain	Maalitaulu ei saanut osumaa ( <i>target hit false</i> )

### 4.3 Tauluohjaimen ohjelmistosuunnittelu

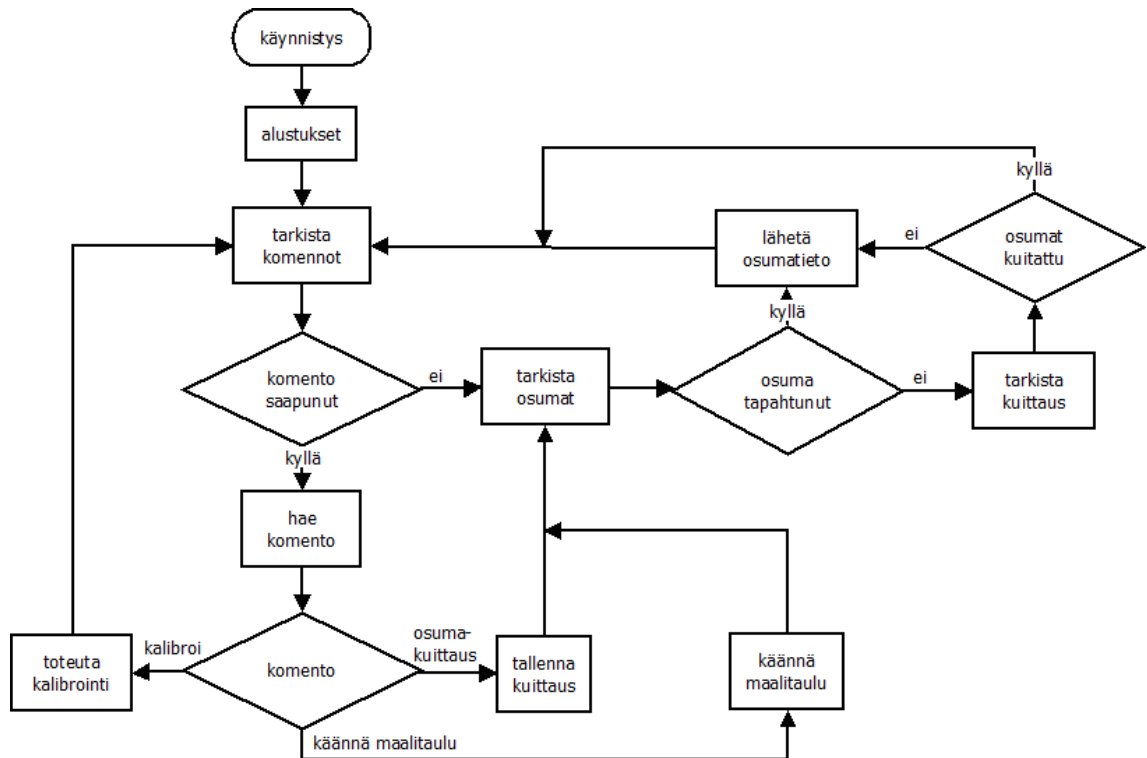
Maalitaululaitteen mikro-ohjaimen ohjelmiston toteutus jaetaan osiin, jotka toteuttavat oman erillisen tehtävänsä. Nämä tehtävät ovat servolaitteen ohjaaminen, anturin lukeminen, asetusten muistiin kirjoittaminen, prosessorin ulosmenojen ja sisään tulosten hallinta, UART-väylän hallinta sekä radiolinkin hallinta. Prosessorin ohjelmakoodi tuotetaan pääohjelmamoduulin lisäksi mainittujen tehtävien toteuttaviin moduuleihin. Kuvassa 4.1 on esitetty ohjelmiston moduulikaavio, josta nähdään ohjelmamoduulit ja niiden väliset riippuvuudet.



**Kuva 4.7** Tauluohjaimen ohjelmiston moduulit ja niiden väliset riippuvuudet

Ohjelma tekee käynnistyessään ensin tarvittavat asetukset I/O-porteille sekä ohjelmassa käytetyille keskeytyspalveluille. Tämän jälkeen suoritus siirtyy pääohjelman silmukkaan. Silmukassa tehdään kolme toimintoa, jonka jälkeen palataan silmukan alkuun. Ensimmäinen toiminto on hallintalaitteen komentojen vastaanoton tarkistaminen sekä saapuneiden komentojen lukeminen ja suorittaminen. Seuraava toiminto tarkistaa, onko maalitauluosuma tapahtunut. Mikäli osuma on tapahtunut, lähetetään tieto osumasta hallintalaitteelle. Pääohjelman silmukan viimeinen toiminto on osumatiedon kuittauksen tarkistaminen. Jos osumatieto on lähetetty, mutta hallintalaite ei ole sitä kuitannut määrätyn ajan kuluttua, lähetetään tieto uudelleen. Kuvassa 4.8 on esitetty ohjelman kulku pääpiirteittäin vuokaavion avulla.

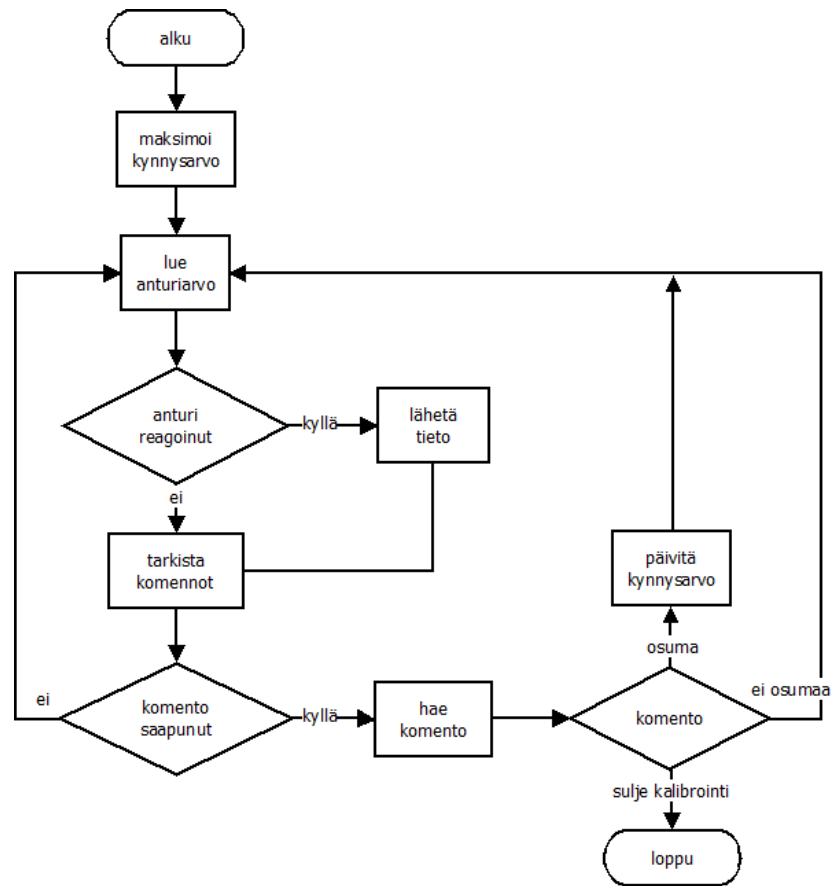




**Kuva 4.8** Tauluohjaimen ohjelmiston pääohjelmiasilmukan kuvaus vuokaavioesityksenä

Tauluohjaimen ohjelmisto tarkastelee maalitaulun osumia anturin suurinta arvoa tarkastelemalla. Osuma todetaan tapahtuneeksi, kun suurin arvo ylittää muistiin tallennetun kynnyksenarvon. Tarkastelu toimii siis laukaisumenetelmällä, jolloin suurin anturin lukema arvo on aina johduttava maalitaulun osumasta, jotta järjestelmä toimisi oikein. Tilanne on kuitenkin helppo saavuttaa mikrofonia käytettäessä anturina, jos mikrofoni on asennettu riittävän lähelle maalitaulua.

Tauluohjaimen ohjelmisto toteuttaa normaalin toiminnan lisäksi kalibrointitoiminnon. Kalibrointitilaan siirrytään hallintalaitteen komennosta, jolloin osumakynnyksen arvo maksimoidaan kalibrointia varten ja ohjelman suoritus siirtyy kalibrointisilmukkaan. Silmukassa luetaan ensin maalitaulun anturin arvo ja verrataan sitä ennalta määrättyyn minimiarvoon, jonka ylittyessä anturin voidaan todeta reagoineen johonkin ympäristön tapahtumaan. Jos minimiarvo ylittyy ja arvo on pienempi kuin osumakynnyksen arvo, lähetetään tieto siitä hallintalaitteelle. Tämän jälkeen tarkistetaan hallintalaitteelta saapuneet komennot. Jos komento on saapunut, luetaan sen sisältö, muussa tapauksessa palataan silmukan alkuun. Mikäli saapunut komento on osuman hylkäämiskomento, palataan silmukan alkuun. Mikäli taas saapunut komento on osuman hyväksymiskomento, päivitetään osumakynnyksen arvo uuteen ja palataan silmukan alkuun. Kalibrointitilan sulkemiskomennon saapuessa ohjelman suoritus siirtyy takaisin pääohjelmiasilmukkaan. Kuvassa 4.9 on esitetty kalibrointitoiminnon kulku vuokaavion avulla.



**Kuva 4.9** Tauluohjaimen ohjelmiston kalibrointitilan kuvaus vuokaavioesityksenä

Pääohjelmiasilmukan toiminnot sekä kalibrointitoiminnot suoritetaan käyttämällä hyväksi ohjelmiston eri moduulien tarjoamia palveluita. Seuraavissa kappaleissa kuvataan ohjelmiston moduulien tarjoamat palvelut sekä niiden käyttöön määritellyt rajapinnat.

## IO.H

Otsikkotiedoston *io.h* kautta päästään käsiksi rajapintaan, joka tarjoaa palvelut I/O-porttien alustamiseen sekä piirilevyllä olevien LED-valojen sytyttämiseen ja sammuttamiseen. Lisäksi rajapintaan voidaan toteuttaa tarvittaessa piirilevyn ulkoisten laitteiden ohjaamisen sekä lukemisen toteuttavat funktiot.

## SENSOR.H

Otsikkotiedosto *sensor.h* esittelee rajapinnan, joka tarjoaa funktiot osumien tarkistukseen sekä anturin lukutoiminnon alustamiseen. Lisäksi rajapinta esittelee keskeytyspalvelun AD-muuntimen arvojen lukemiselle.

Anturin lukutoiminnon alustava aliohjelma suorittaa anturin lukemista edellyttävät laitteistoasetukset. Näitä ovat sisäisen referenssijännitteen käyttöönotto, AD-muuntimen kanavan valinta, esijakajan asettaminen, muunnoskeskeytyksen käyttöönotto sekä itse muuntimen käyttöönotto. Lisäksi aliohjelman lopussa käynnistetään ensimmäisen anturiarvon muuntaminen.

AD-muuntimen keskeytys tapahtuu, kun muunnos on valmis luettavaksi. Keskeytyspalvelussa luetaan muuntimen arvo, tarkastetaan edellisen maalitauluosuman jälkeen kulunut aika sekä päivitetään maksimiarvo tarvittaessa. Mikäli edellisestä maalitauluosumasta on kulunut määrätty aika, anturimuunnoksen arvo tallennetaan maksimiarvon tallentavaan muuttujaan jos arvo ylittää edellisen maksimiarvon. Keskeytyspalvelun lopuksi käynnistetään vielä seuraavan anturiarvon muuntaminen.

Osumien tarkistusfunktio tarkastaa, onko tauluosumia tapahtunut edellisen tarkistuksen jälkeen ja palauttaa tiedon totuusarvona. Mikäli edellisestä osumasta ei ole kulunut määrättyä aikaa, on paluuarvo aina *epätosi*. Muussa tapauksessa AD-muuntimen keskeytyspalvelussa tallennettua maksimiarvoa verrataan osuman kynnysarvoon. Jos maksimiarvo on suurempi kuin osuman kynnysarvo, asetetaan osuma-ajastimelle määrätty arvo, nollataan anturimuunnosten maksimiarvo ja palautetaan *tosi*.

## SERVO.H

Otsikkotiedosto *servo.h* esittelee rajapinnan, joka tarjoaa palvelut maalitaulun liikuttamiseen. Näiden palveluiden toteuttamiseen kuuluvat funktiot servon ohjauksen alustamiseen, maalitaulun liikuttamiseen sekä maalitaulun tilan lukemiseen. Lisäksi rajapinta esittelee keskeytyspalvelut maalitaulua ohjaavan servon ohjaukseen tarvittavan PWM-signaalin tuottamiseen.

Servon ohjauksen alustava aliohjelma suorittaa prosessorin ajastimen alustamisen siten, että sen avulla voidaan tuottaa PWM-signaalia. Näihin alustamisiin kuuluu ajastimen esijakajan asettaminen, ajastimen ylivuotokeskeytyksen käyttöönotto (engl. *overflow interrupt*) sekä keskeytyksen käyttöönotto tilanteissa, jossa ajastimen arvo saavuttaa vertausarvon (engl. *output compare match*). Lisäksi aliohjelma asettaa vertausarvon vastaamaan arvoa, jossa PWM-signaalin pulssisuhde määrittää servon asennon sellaiseksi, että maalitaulu on ala-asennossa.

Maalitaulun liikuttamisen toteuttava aliohjelma saa parametrina maalitaulun halutun asennon totuusarvona. Käytännössä aliohjelma liikuttaa maalitaulun haluttuun asentoon asettamalla ajastimen vertausarvo sellaiseksi, että PWM-signaalin pulssisuhde määrittää maalitaulua liikuttavan servon asennon maalitaulun haluttua asentoa vastaavaksi. Maalitaulun tila myös tallennetaan muutoksen jälkeen, josta se voidaan lukea maalitaulun tilan lukevan funktion avulla.

Ajastimen ylivuotokeskeytys tapahtuu, kun ajastimen laskuri on ohittanut maksimiarvonsa ja muuttunut nolllaksi. Tällöin ylivuodon keskeytyspalvelu asettaa PWM-signaalin ulostulopinnin jännitteelliseksi. Toinen ajastimen keskeytys tapahtuu, kun ajastimen laskuri saavuttaa vertausarvon. Tällöin ohjelman suoritus siirtyy keskeytyspalveluun, jossa asetetaan PWM-signaalin ulostulopinni jännitteettömäksi. Näin vertausarvoa muuttamalla voidaan muuttaa signaalin pulssisuhdetta, ja saadaan toteutettua kuvassa 4.3 esitellyt signaalit servon ohjaukseen.

## SETTINGS.H

Otsikkotiedosto *settings.h* esittelee rajapinnan, joka tarjoaa palvelut kalibrointiasetuksen lukemiseen ja kirjoittamiseen mikro-ohjaimen sisäiseen muistiin. Kalibrointiasetus on yksittäinen 16-bittinen muuttuja, joka sisältää osumaksi tulkittavan anturimuunnoksen kynnysaron. Muuttujan arvo säilytetään haihtumattomassa EEPROM-muistissa (engl. Electronically Erasable Programmable Read-Only Memory), josta se voidaan lukea, ja johon uusi arvo voidaan kirjoittaa.

## UART.H

Otsikkotiedosto *uart.h* esittelee rajapinnan, joka tarjoaa palvelun datan lähettämisen aloittamisen UART-väylään sekä keskeytyspalvelut merkin vastaanottamiseen ja lähettämiseen. Vastaanotettava data kirjoitetaan vastaanottokeskeytyksen palvelussa globaalin vastaanottopuskuriin.

Lähetettävä data kirjoitetaan globaaliin lähetyspuskuriin, josta se kirjoitetaan UART-väylään keskeytyspalvelussa. Lähetyskeskeytys tapahtuu, kun prosessorin sisäinen lähetyspuskuri on tyhjä. Tällöin keskeytyksen palvelussa tarkistetaan, onko globaalissa lähetyspuskurissa merkkejä jäljellä. Jos merkkejä on jäljellä, kirjoitetaan merkki prosessorin sisäiseen lähetyspuskuriin ja poistetaan globaalista lähetyspuskurista. Tilanteessa, jossa molemmat lähetyspuskurit ovat tyhjiä, ei uutta lähetyskeskeytystä enää tapahdu, koska keskeytys tapahtuu vain, kun prosessorin sisäinen lähetyspuskuri tyhjenee. Tämän vuoksi rajapinta tarjoaa aliohjelman datan lähettämisen aloittamiseen. Lähettämisen aloittava aliohjelma kirjoittaa globaalin lähetyspuskurin ensimmäisen merkin prosessorin sisäiseen lähetyspuskuriin ja poistaa sen globaalista puskurista.

## WLAN.H

Otsikkotiedosto *wlan.h* esittelee rajapinnan, joka tarjoaa palvelut tiedon lähettämiseen tauluohjaimelta hallintalaitteelle sekä hallintalaitteen lähettämien komentojen lukemiseen. Ohjelmamoduuli toteuttaa kaksi rajapintaa, tauluohjaimen ohjelmiston käyttämän rajapinnan komentojen lähettämiseen ja vastaanottamiseen, sekä hallintalaitteen rajapinnan tauluohjaimen käyttämiseen.

Tiedon lähettäminen suoritetaan rajapinnan kautta kutsumalla aliohjelmia, jotka lähettää tiedon tauluosumasta ja anturin reagoinnista kalibrointitilassa. Aliohjelmat toteuttavat toiminnon hallintalaitteen sekä tauluohjaimen rajapinnan mukaisia komentoja käyttäen. Komentojen lähettäminen tehdään kirjoittamalla komennot globaaliin lähetyspuskuriin sekä käyttämällä otsikkotiedoston *uart.h* esittelemää datan lähettämisen aloittavaa palvelua.

Komentojen vastaanottaminen tapahtuu käyttämällä aliohjelmaa, joka tarkistaa, onko komentoja tullut hallintalaitteelta ja palauttaa tiedon totuusarvona. Vastaanotetut komennot tarkistetaan tutkimalla globaalia vastaanottopuskuria. Mikäli komento on tullut, aliohjelma tallentaa sen globaaliin komentomuuttujaan sekä poistaa sen globaalista vas-

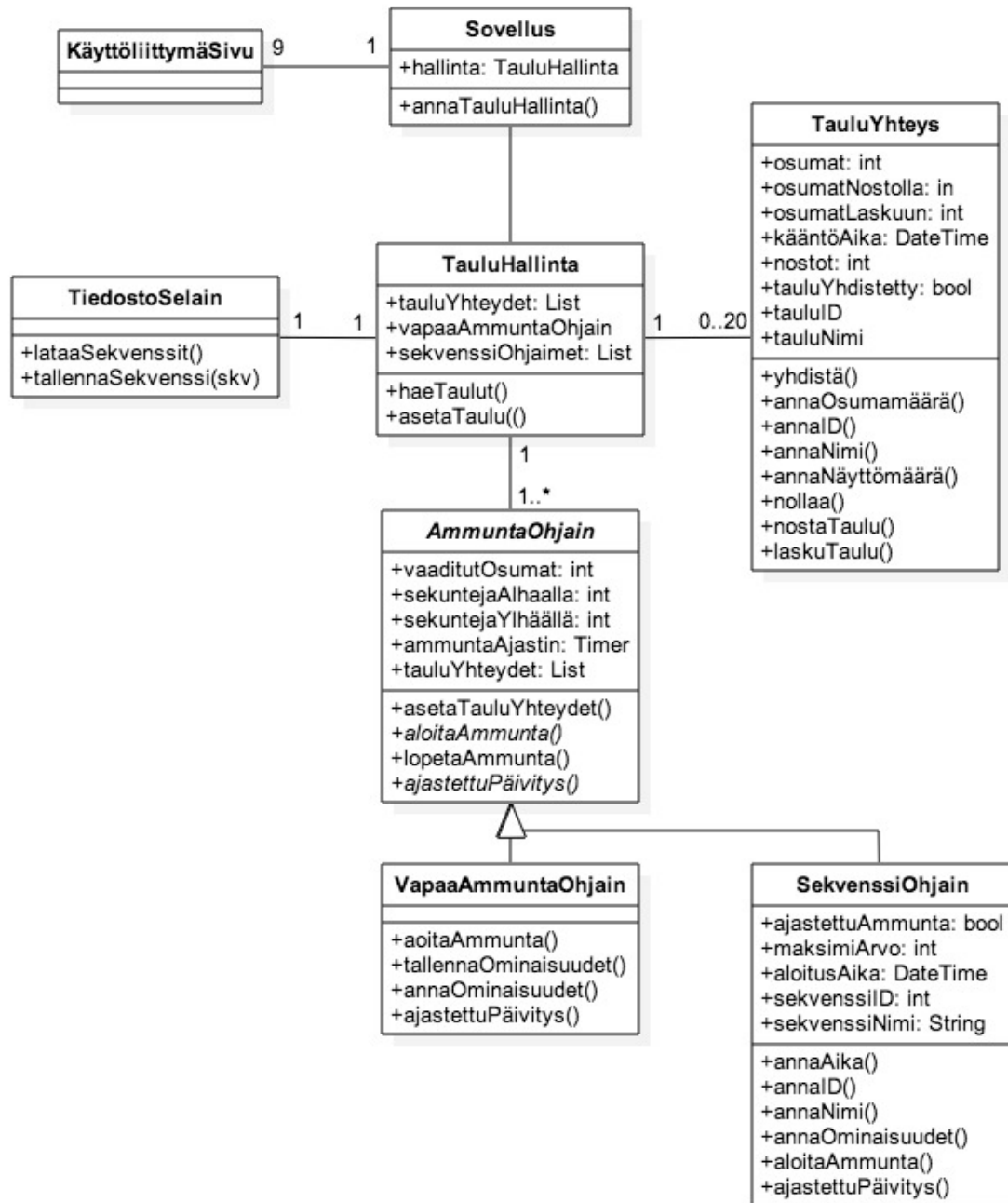
taanottopuskurista. Tämän jälkeen vastaanotettu komento voidaan lukea globaalista komentomuuttujasta.

## **4.4 Hallintalaitteen ohjelmiston kuvaus**

Hallintalaitteen ohjelmiston suunnittelu koostuu kahdesta erillisestä osa-alueesta, ohjelmistosuunnittelusta sekä käyttöliittymäsuunnittelusta. Aliluvussa 4.4.1 kuvataan hallintalaitteen ohjelmistosuunnittelu ja aliluvussa 4.4.2 käyttöliittymän suunnittelu.

### **4.4.1 Hallintalaitteen ohjelmistosuunnittelu**

Hallintalaitteen ohjelmiston päätehtävinä on hallita tauluohjaimia sekä toimia pisteenlasku- ja hallintajärjestelmän käyttöliittymänä. Ohjelmisto toteuttaa myös tauluohjainten osumien laskemisen, sekä ennalta määritettyjen ammutasekvenssien ohjaamisen. Ohjelmiston suunnittelussa käytettiin olio-ohjelmointimenetelmiä. Toteutettavan ohjelmiston sisällöstä voitiin erottaa selvästi erilliset käsitteet, taulujen hallinta, yksittäisen taulun datayhteys, vapaa-ammuntatila, sekvenssiammuntatila sekä muistiin kirjoittaminen. Lisäksi tarvitaan käyttöliittymää varten joukko luokkia käyttöliittymäsivuja varten sekä pääohjelma. Kuvassa 4.10 on esitetty ohjelmiston sisältöä ja rakennetta kuvaava luokkakaavio.



**Kuva 4.10** Hallintalaitteen ohjelmiston luokkakaavio

Luokkakaaviossa esitellään ohjelmiston luokat ja niiden väliset riippuvuudet sekä toteutuksen kannalta oleelliset attribuutit ja palvelut. Seuraavissa kappaleissa kuvataan luokkien tehtävät. Ohjelmiston toiminnan kannalta kriittisten tehtävien toteutukset on esitetty tarkemmin algoritmien avulla.

## SOVELLUS JA KÄYTTÖLIITTYMÄSIVUT

Sovellusluokka toimii ohjelmistossa pääohjelmana. Pääohjelma luo taulun hallinnan toteuttavan olion, jonka palveluja käyttämällä ohjelmiston käyttöliittymäsivujen eli näkymien toteuttavat oliot päivittävät tietoa ohjelman tilasta. Sovellusluokan ja käyttöliittymäsivujen suunnittelua ei tässä kuvata tarkemmin, koska se on vahvasti ohjelmistoke-

hitysympäristöstä riippuvainen. Lisäksi niiden toteutus on pääosin hoidettu valmiiksi ohjelmistokehitysympäristöjen sovelluskehitysissä (*template*).

## TAULUJEN HALLINTA

Luokka *TauluHallinta* hoitaa tiedostoselainolion, vapaa-ammuntaolion ja tauluohjainten yhteysolioiden luomisen sekä lataa muistiin tallennetut sekvenssiammunta-asetukset sekvenssiammuntaolioiden luomista varten. Tauluhallintaluokka tarjoaa palvelut tauluohjainten ja ammuntaolioiden tilan näyttämiseen ohjelmiston käyttöliittymässä.

## AMMUNTAOHJAINLUOKAT

Ammuntaohjainluokat toteuttavat hallintalaitteen ohjelmiston ohjatut ammuntatoiminnot. Luokka *VapaaAmmuntaOhjain* vastaa ohjelmiston vapaa-ammuntatoiminnon toteutuksesta ja luokka *SekvenssiOhjain* sekvenssiammuntatoiminnon toteutuksesta. Luokat periytetään abstraktista kantaluokasta *AmmuntaOhjain*, koska luokkien ominaisuuksissa ja toiminnoissa on paljon yhtäläisyyksiä. Kantaluokka määrittelee aliohjelmat ammunnassa käytettyjen tauluohjainten asettamiseen, ammunnan aloittamiseen ja lopettamiseen, pistelaskurin tiedon tarjoamiseen sekä pistelaskurin nollaamiseen. Kun ammunnan aloittavaa aliohjelmaa on kutsuttu, ammuntaohjain käynnistää ajastimen tauluohjainten ohjaamiseksi ajastetusti. Ajastinta käytetään herätteenä tauluohjaimia ohjaavan aliohjelman ajamiseen määrätyn ajanjakson välein.

Vapaa-ammuntatoiminnon toteuttava luokka tarjoaa palvelut vapaa-ammunta-asetusten lukemiseen ja tallentamiseen. Luokan toteutus tauluohjaimia ohjaavasta aliohjelmasta suorittaa ohjaamisen algoritmin 1 mukaisesti. Aliohjelmassa lasketaan tauluohjaimet, jotka ovat ylhäällä, ja joiden nostamisesta on kulunut vapaa-ammunnan asetusten määräämä yläasennossa pitämisen maksimiaika. Tauluohjaimet, jotka ovat alhaalla ja joiden laskemisesta on kulunut asetusten määräämä ala-asennossa pitämisen maksimiaika taas nostetaan.

```

1  for each taulu in taululista do
      if taulu.tila = "ylhäällä"
3      if taulu.muutos aika > ylhäälläoloaika then laskeTaulu();
      else taulu.tila = "alhaalla"
5      if taulu.muutos aika > alhaallaoloaika then nostaTaulu();
      end if
7  end for

```

*Algoritmi 1. Tauluohjainten päivittäminen vapaa-ammuntatilassa*

Sekvenssiammuntatoiminnon toteuttava luokka tarjoaa palvelut ammunta-ajan sekä sekvenssin tietojen ja asetusten hakemiseen. Palautettava ammunta-aika on jäljellä oleva ammunta-aika, jos sekvenssin asetuksissa on ammuntilaksi valittu ajastettu ammunta. Muussa tapauksessa palautettava ammunta-aika on aika, joka on kulunut ammunnan aloittamisesta. Luokan toteutus tauluohjaimia ohjaavasta aliohjelmasta suorittaa taulujen ohjaamisen algoritmin 2 mukaisesti.

```

1  for each taulu in taululista do
      if taulu.tila = "ylhäällä" then
3      if taulu.muutos aika > ylhäälläolo aika then laskeTaulu();
      end if
5  end for

7  ammuntaValmis := false;

9  if ajastettuAmmunta then
      if aloitusAika + ammuntaAika > tämänhetkinenAika then
11     ammuntaValmis := true;
      end if
13 else
      ammuntaValmis := true;
15     for each taulu in taululista do
          if taulu.näyttömäärä = maksiminäyttömäärä then
17             ammuntaValmis := false;
                break;
19             end if
          end for
21     end if

23     if ammuntaValmis then
        lopetaAmmunta();
25     else
        for each taulu in taululista do
27             if taulu.tila = "alhaalla" then
                if taulu.muutos aika > alhaallaolo aika then nostaTaulu();
29             end if
        end for
31     end if

```

*Algoritmi 2. Tauluohjainten päivittäminen sekvenssiammuntatilassa*

Maalitaulut lasketaan samoilla ehdoilla, kuin vapaa-ammuntatilassa. Nostaminen vaatii kuitenkin huomattavasti enemmän tarkastelua. Tämä johtuu siitä, että on tarkistettava ensin, onko ammunta suoritettu valmiiksi. Jos sekvenssin asetuksissa ammunta on määritetty ajastetuksi, ammunta on suoritettu valmiiksi jos aika loppuu. Jos ammunta ei ole määritetty ajastetuksi, on ammunnan loppumisen ehtona maalitaulujen näyttökertojen loppuminen.



## TAULUYHTEYS

Tauluyhteyden toteuttava luokka *TauluYhteys* hoitaa yhteyden muodostamisen ja datan lähettämisen ohjelmiston ja yksittäisen tauluohjaimen välillä. Lisäksi se pitää kirjaa hallitsemansa tauluohjaimen tauluosumien ja nostojen määrästä sekä viimeisimmästä taulun liikuttamisajasta. Käytännössä tauluohjaimen yhdistäminen tehdään avaamalla TCP-yhteys tauluohjaimen sen IP-osoitteen sekä käytettävän portin avulla.

## TIEDOSTOSELAIN

Tiedostojen käsittelyn hoitava luokka *TiedostoSelain* hoitaa ohjelmiston datan kirjoittamisen hallintalaitteen muistiin sekä muistin lukemisen. Se tarjoaa palvelut ammuntaskvenssien nimien ja asetusten tallentamisen sekä lukemisen muistista.

### 4.4.2 Hallintalaitteen käyttöliittymäsuunnittelu

Hallintalaitteen ohjelmiston käyttöliittymän kautta tulee voida suorittaa kaikkia järjestelmän käyttöön liittyviä toimintoja. Näitä toimintoja ovat:

- Käytettävien tauluohjainten valitseminen
- Yksittäisen tauluohjaimen nimeäminen, kalibrointi ja testaaminen
- Vapaan ammunnan suorittaminen
- Vapaan ammunnan asetusten valitseminen
- Sekvenssiammunnan suorittaminen
- Suoritettavan sekvenssin valitseminen
- Uuden sekvenssin luominen

Selvästi toisistaan eroteltavia käsitteitä on siis kolme: Tauluohjaimet, vapaa ammunta ja sekvenssiammunta. Jokaiseen käsitteeseen liittyy omat toimintonsa. Käyttöliittymään toteutettiin päävalikkonäkymä näiden kolmen toiminnon valitsemiseksi. Tauluohjainten toiminnot toteutettiin kolmen eri näkymän avulla, tauluohjainten valitseminen, nimeäminen ja testaaminen sekä kalibrointi. Vapaan ammunnan toimintoihin tarvitaan kaksi näkymää, ammuntanäkymä sekä asetusnäkyvä. Sekvenssiammunnan toiminnot toteutettiin kolmena näkymänä, sekvenssivalikko, uuden sekvenssin luominen sekä sekvenssiammuntatila.

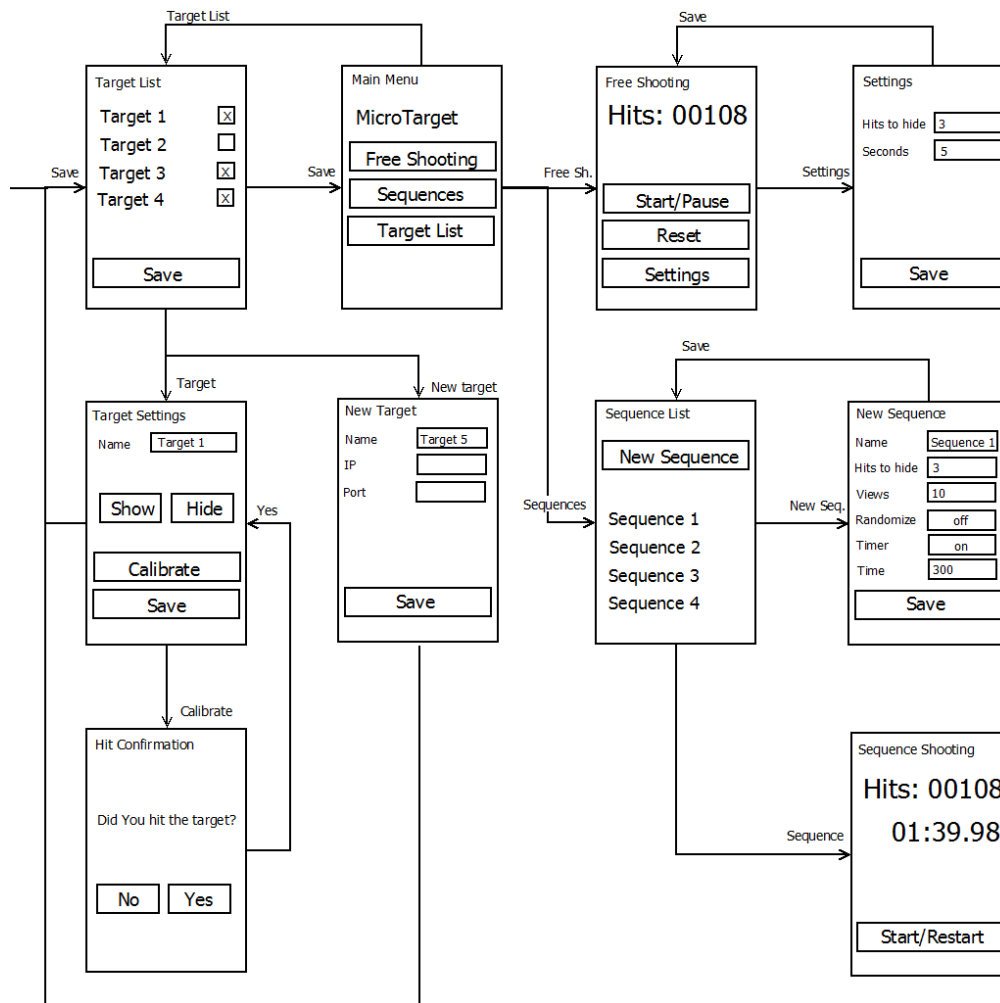
Hallintalaitteen ohjelmiston käyttöliittymä pyrittiin suunnitella mahdollisimman selkeäksi ja johdonmukaiseksi. Kun ohjelma käynnistetään, on ennen ammunnan aloittamista valittava käytettävät tauluohjaimet ja otettava niihin yhteys. Tämän vuoksi tauluohjainten valintanäkymää päätettiin käyttää ensimmäisenä näkymänä ohjelman käynnistyttyä. Valintanäkymässä tallennetut tauluohjaimet listataan. Lisäksi vaihtoehtoina on uuden tauluohjaimen lisääminen sekä yksittäisen tauluohjaimen asetuksiin siirtyminen. Valittaessa tauluohjaimen asetukset, aukeaa näkymä, jossa voidaan muokata tauluohjaimelle annettua nimeä sekä testata taulua painamalla nosto- ja laskunäppäimiä. Näitä painamalla hallintalaitte lähettää komennon tauluohjaimelle taulun nostamiseen tai laskemiseen. Valittaessa uuden tauluohjaimen lisääminen, aukeaa näkymä, jossa syötetään uuden

tauluohjaimen IP-osoite sekä käytettävä portti. Uusi tauluohjain tallennetaan valitsemalta *tallenna*.

Kun käyttäjä valitsee tauluohjainlistasta paritettavat laitteet, ohjelmisto suorittaa yhteydenmuodostamisen jokaiselle valitulle tauluohjaimelle ja siirtyy päävalikkoon. Päävalikossa valintavaihtoehtoina on *vapaa ammuskelu*, *ammuntasekvenssit* sekä *paritusasetukset*. Valittaessa toiminto *vapaa ammuskelu* aukeaa uusi näkymä, jossa näytöllä esitetään pistelaskuri sekä valintavaihtoehdot *aloita*, *tauko*, *nollaus* sekä *asetukset*. Valittaessa *aloita*, tauluja aletaan nostamaan asetuksissa määrättyin säännöin. *Tauko*-valinta pysäyttää taulujen ohjaamisen ja *nollaus*-valinta nolaa pistelaskurin. Valittaessa *asetukset*, voidaan valita maalitaulujen piilottamiseen vaadittu osumamäärä sekä sen, kuinka pitkän ajan kuluttua kaatumisen jälkeen taulu käännetään takaisin esille.

Kun hallintalaitteen ohjelmiston päävalikosta valitaan ammuntasekvenssit, aukeaa näytölle uusi näkymä, *sekvenssivalikko*, jossa näytetään hallintalaitteelle tallennetut ammuntasekvenssit sekä valinta *uusi sekvenssi*. Valittaessa uusi sekvenssi, aukeaa näytölle näkymä, jossa tehdään uuden sekvenssin asetukset. Asetuksia ovat sekvenssille valittava nimi, ammunnan ajan oton käyttöön otto tai sekvenssin keston määrittäminen, maalitaulun piilottamiseen vaadittu osumamäärä, sekä nostojen lukumäärä. Lisäksi voidaan valita, nousevatko taulut satunnaisesti vuorotellen, vai nousevatko taulut määritellyn ajan kuluttua uudelleen. Asetuksien valinnan jälkeen valitaan tallenna, jolloin ohjelma palaa sekvenssivalikkoon. Valittaessa jokin tallennettu sekvenssi, aukeaa uusi näkymä, jossa näytöllä esitetään pistelaskuri ja ajan näyttö sekä näytetään valinta *käynnistä*. Valittaessa *käynnistä* ammuntasekvenssi käynnistyy. Sekvenssin päätyttyä näytöllä esitetään osuimien lukumäärät tauluja kohden tai vaihtoehtoisesti sekvenssin suoritukseen kulunut kokonaisaika.

Kun päävalikosta valitaan paritusasetukset, palaa ohjelma samaan näkymään, kuin käynnistyttyään. Näkymästä edelliseen siirtymiseksi voidaan painaa hallintalaitteen virtuaalista tai fyysistä paluu-näppäintä. Kuvassa 4.11 on esitetty hallintalaitteen ohjelmiston käyttöliittymän näkymäkartta, josta nähdään käyttöliittymän tarjoamien näkymät ja navigointi niiden välillä.



*Kuva 4.11 Hallintalaitteen käyttöliittymän näkymäkartta*

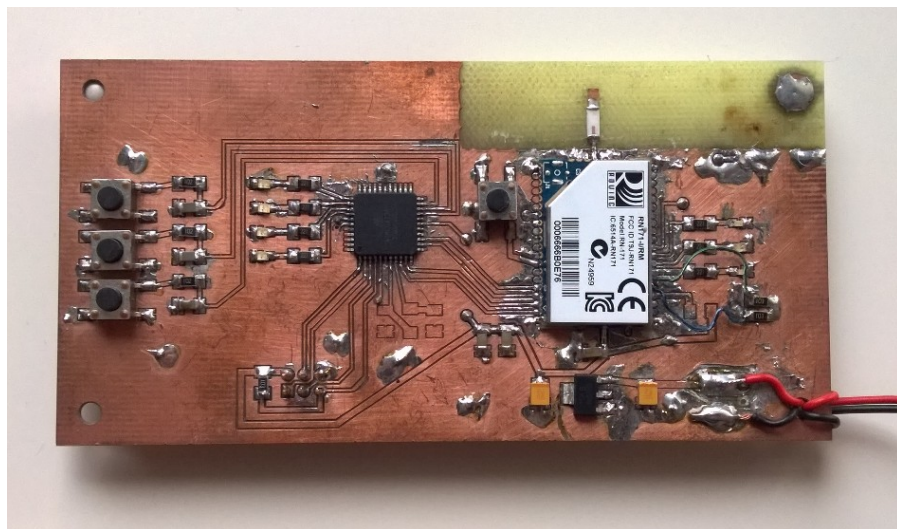
## 5. KÄYTÄNNÖN TOTEUTUS

Tässä luvussa kuvataan järjestelmän prototyypin käytännön toteutusprosessin menetelmät, kokemuksia sekä toteutusvaiheessa kohdattuja ongelmia. Luvussa 5.1 käsitellään tauluohjaimen elektroniikan toteutus, luvussa 5.2 tauluohjaimen ohjelmiston toteutus ja luvussa 5.3 mekaniikan toteutus. Luvussa 5.4 käsitellään hallintalaitteen ohjelmiston toteutus ja luvussa 5.5 radiolinkin toteutus.

### 5.1 Tauluohjaimen elektroniikkapiiri ja –komponentit

Laitteiston piirilevyjä suunniteltiin kaksi. Ensimmäinen piirilevy suunniteltiin, jotta valittua radiolähetintä voitiin testata jo varhaisessa vaiheessa. Toinen piirilevy suunniteltiin lopullisen prototyypin ohjainalustaksi. Piirilevyjen suunnitteluun käytettiin KiCAD-ohjelmistoa.

Ensimmäinen piirilevy suunniteltiin heti projektin alussa nopealla aikataululla. Piirikytkentään liitettiin mikro-ohjain, WLAN-moduuli RN171 sekä 3 painonäppäintä ja 4 LED-valoa. Näppäinten ja LED-valojen tarkoitus oli toimia käyttöliittymänä testiohjelmiston toimintojen käynnistämiseen ja radiovastaanottimelta saapuvan datan esittämiseen. Testikytkentä valmistettiin kaksipuoliselle piirilevyllä käyttämällä tietokoneella numeerisesti ohjattavaa jyrsintä (engl. Computer Numerical Control, CNC). Piirilevyn suunnittelussa tuotetut johdotuskuvion sisältämät tiedostot syötettiin jyrsintälaitetta ohjaavalle tietokoneelle, jolloin ne saatiin jyrsittyä piirilevyllä ohjatusti pientä terää käyttäen. Kuvassa 5.1 on esitetty testikäyttöön suunniteltu piirilevy komponentteineen.



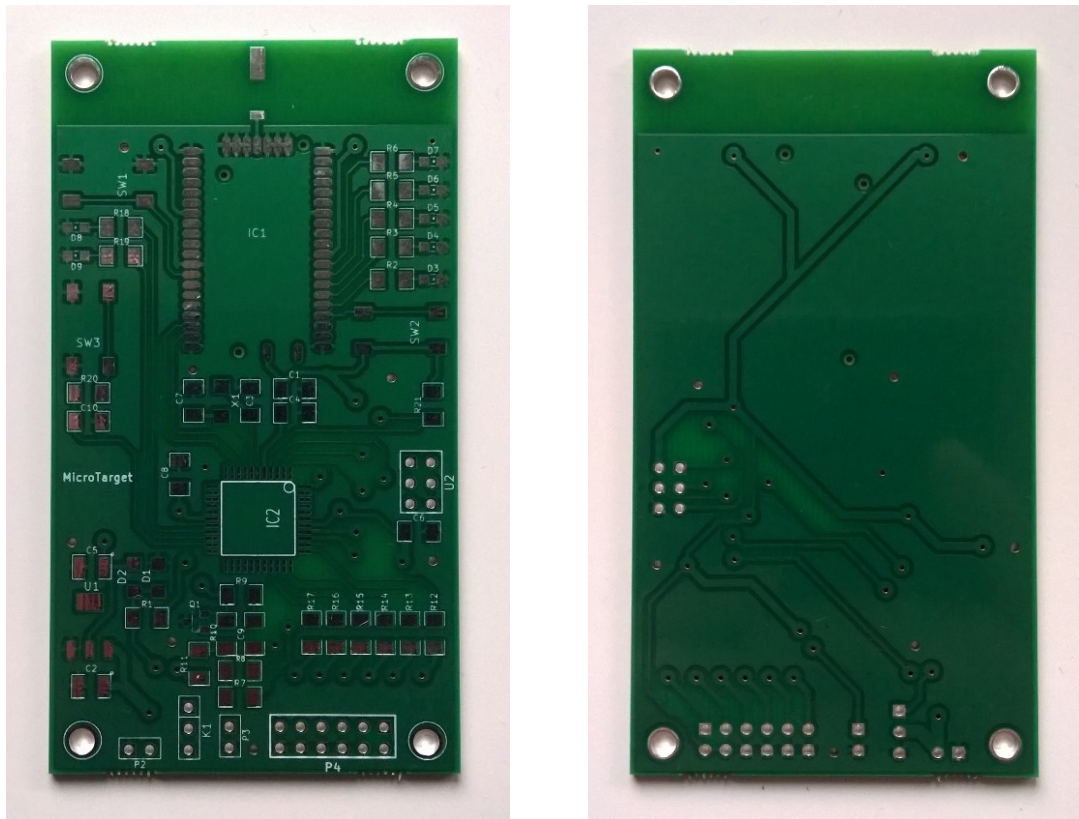
*Kuva 5.1 WLAN-moduulin toiminnan testaamiseen valmistettu piirilevy*

Testikytkentä toimi toivotulla tavalla, ja WLAN-moduulin toimintaa saatiin testattua. Testaaminen osoittautui hyväksi aloittaa mahdollisimman nopeasti, sillä moduulin käyttöön perehtyminen vei paljon aikaa. Käytön aloittaminen ei sujunut ongelmitta, ja moduulin ohjelmistoon olikin tehtävä tehdasasetusten palautus moneen otteeseen. Jossain vaiheessa moduuliin perehtymistä kävikin jo mielessä moduulin vaihtaminen johonkin toiseen vastaavaan. Moduulin käyttöä edellyttävät asetukset saatiin kuitenkin lopulta määritettyä ja moduulin käytön testaus saatiin käyntiin. Luvussa 5.5 on kuvattu moduulin käyttöön vaaditut asetukset.

Testausvaiheessa WLAN-moduulin kanssa kommunikointiin tietokoneen avulla käyttäen telnet-yhteyttä. Kun asetukset olivat oikeat, WLAN-moduuli toimi toivotulla tavalla ja data saatiin liikkumaan onnistuneesti tietokoneen ja testikytkennän välillä. Voitiin todeta, että testattu moduuli oli sopiva tauluohjaimen radiolähettimeksi ja sen toiminnot olivat riittävät järjestelmän toteuttamiseen.

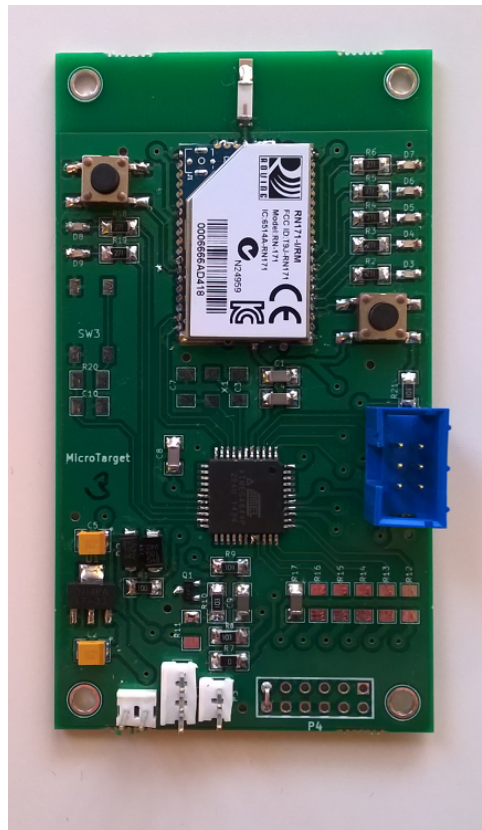
Toinen piirilevy suunniteltiin lopullisen prototyypin ohjainalustaksi, joten siihen toteutettiin kaikki luvussa 4.1 esitellyt lohkot ja toiminnot. Piirilevystä tehtiin kaksipuolinen. Tauluohjainten prototyyppien piirilevyt valmistettiin Pirkkalalaisessa yrityksessä Cri-dix OY. Teetetyn erän suuruus oli 12 kappaletta. Liitteessä A on esitetty tauluohjaimen piirilevyn kytkentäkaavio, liitteessä B johdotuskuvat, liitteessä C osasijoittelukuva ja liitteessä D komponenttiluettelo.

Tauluohjaimen prototyypin piirilevysuunnittelussa keskityttiin testilevyä enemmän komponenttien fyysiseen sijoitteluun sekä koteloon liitettävyyteen. Suunnittelussa kaikki koteloon asennettavien lisälaitteiden liittimet sijoitettiin piirilevyn alaosaan. Näitä liittimiä ovat paristoliitin, servon liitin, mikrofonin liitin sekä liitin ylimääräisiä ulkoisia laitteita varten. Liittimen tyyppinä käytettiin piikkirimaa. Mikro-ohjaimen ohjelmointiliitin sijoitettiin erilleen muista liittimistä piirilevyn reunalle, jotta sitä ei sekoiteta normaalissa käytössä käytettäviin liittimiin. WLAN-moduuli sijoitettiin piirilevyn yläosaan, koska siihen liitettävä siruantenni vaatii toimiakseen piirilevyltä kuparittoman alueen ympärilleen. Mikro-ohjain sijoitettiin keskelle piirilevyä, jotta sen ympärille jäisi mahdollisimman paljon tilaa siihen liitettäviä lukuisia johtimia varten. Käyttöjännitejohtimet suunniteltiin signaalijohtimia leveimmiksi ja ne sijoitettiin piirilevyn toiselle puolelle. Piirilevyn kulmiin tehtiin paikat ruuveja varten koteloon liittämiseksi. Lisäksi piirilevystä tehtiin hieman testilevyä pienempi. Kuvassa 5.2 on esitetty lopullinen tauluohjaimen prototyypin piirilevy.



*Kuva 5.2 Tauluohjaimen piirilevy*

Tauluohjainten prototyyppien valmistamista varten hankittiin komponentit neljän piirilevyn kalustamiseen. Komponenttien juottaminen tehtiin käsin. Kuvassa 5.3 on esitetty tauluohjaimen piirilevy komponentteineen.



*Kuva 5.3 Tauluohjaimen piirilevy komponentteineen*

## 5.2 Tauluohjaimen ohjelmiston toteutus

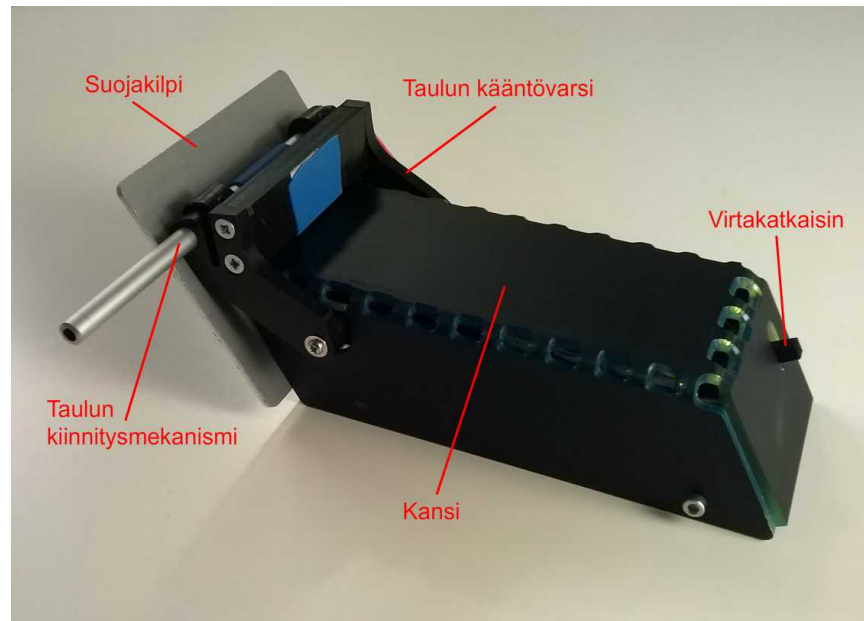
Tauluohjaimen mikro-ohjaimen ohjelmisto toteutettiin C-kielellä käyttäen mikro-ohjaimen valmistajan ohjelmistoa Atmel Studio 6. Ohjelman viemiseen mikro-ohjaimelle käytettiin AVRISP mkII -ohjelmointilaitetta.

Mikro-ohjaimen kellolähteenä käytettiin mikro-ohjaimen sisäistä 8 megahertsin resonaattoria. Valmiin ohjelmiston tiedostokokoksi jäi noin 10 kilotavua. Näin ohjelma mahtuisi myös mikro-ohjaimen valmistajan edullisemmän mallin mikro-ohjaimen Atmega164PA:n ohjelmamuistiin. Tämän ohjaimen ohjelmamuistin koko on 16 kilotavua, joten tilaa jäisi myös varalle. Mikro-ohjain on koteloiltaan, toiminnoiltaan sekä pinnijärjestykseltään sama kuin prototyypissä käytetty Atmega644PA, joten sitä voidaan käyttää jatkossa kytkennässä kustannuksissa säästämiseksi.

## 5.3 Tauluohjaimen mekaniikka

Tauluohjaimen kotelon suunnittelusta vastasi PiQ-Wing, joka myös valmisti osat. Prototyyppiä varten suunniteltu kotelo koostuu pohjalevystä, kannesta, taulun kiinnitysmekanismista ja kääntövarresta. Lisäksi siihen liitettiin paristokotelo, servo sekä virtakatkaisin. Pohjalevy valmistettiin 2 millimetriä paksusta alumiinilevystä siten, että siitä taitettiin kotelon etuosaan suojalevy. Kansi valmistettiin akryylilevystä joka maalattiin mus-

taksi, taulun kiinnitysmekanismi muovista sekä alumiinista ja kääntövarret muovista. Osat valmistettiin CNC-jyrsintä käyttäen. Kuvassa 5.4 on esitetty tauluohjaimen kotelo ja sen osat.



*Kuva 5.4 Tauluohjaimen piirilevy komponentteineen*

## 5.4 Hallintalaitteen ohjelmiston toteutus

Hallintalaitteena päätettiin käyttää prototyypivaiheessa Windows Phone 8.1 –laitetta. Hallintaohjelmisto ohjelmoitiin käyttäen Visual Studio 2013 –ohjelmistokehitysympäristöä. Ympäristö tarjoaa *Universal Apps* –ohjelmistokehityksen, jota käyttämällä voidaan ohjelmistosta helposti tehdä yhteensopiva sekä Windows Phone 8.1 –älypuhelimien että Windows 8.1 –tietokoneiden kanssa. Ideana on toteuttaa varsinainen ohjelmalogiikka nk. jaettuun tiedostoihin ja toteuttaa käyttöliittymä erillisiin tiedostoihin älypuhelinversiota ja tietokoneversiota varten. Tässä työssä käyttöliittymä toteutettiin kuitenkin pelkästään älypuhelimia varten.

Ohjelmisto toteutettiin kuvan 4.10 luokkakaavion mukaisesti. Ohjelmalogiikka toteutettiin C#- ja käyttöliittymä XAML-kielellä (engl. Extensible Application Markup Language). Toimintojen ohjelmoinnin mahdollistamiseksi ja ohjelmointia helpottamaan käytettiin monia valmiita ohjelmistoympäristön tarjoamia oliioita. Näistä tärkeimpiä ovat:

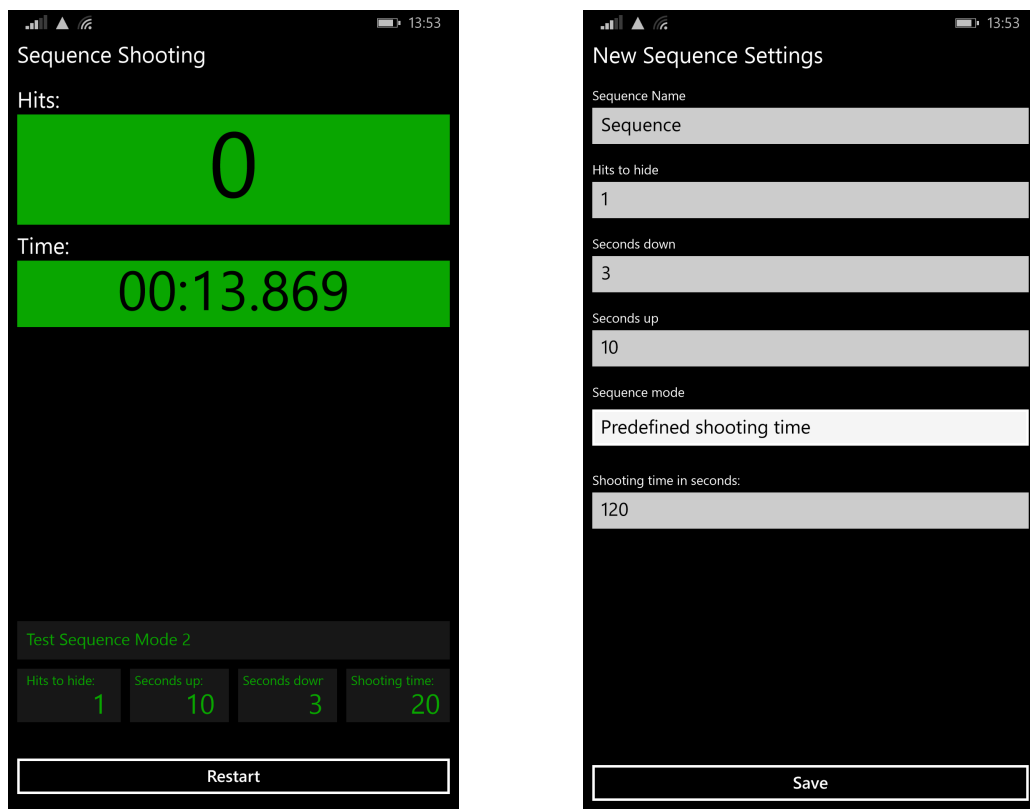
- *DispatcherTimer*, joka tarjoaa valmiin toteutuksen ajastimille. Ajastimia käytettiin ammuttoiminnon toteutuksissa toimimaan herätteenä tauluohjainten päivittämiseksi sekä käyttöliittymän päivittämiseen.
- *DateTime*, joka tarjoaa toteutuksen ajan esittämiseen. Oliota käytettiin maalitaulun osuma-ajan tallentamiseen.



- *StreamSocket*, joka tarjoaa palvelut yhteyden muodostamiseen langattomassa verkossa. Oliota käytettiin toteutetussa *TargetConnection*-luokassa yhteyden muodostamiseen tauluohjaimiin.
- *DataReader*, joka tarjoaa palvelut datavirran lukemiseen. Oliota käytettiin tauluohjaimen lähettämän datan lukemiseen.

Lisäksi käytettiin ohjelmistokehitysympäristön tarjoamaa kantaluokkaa *Page*, joka tarjoaa palvelut navigoitavien näkymien toteuttamiseen. Toteutuksessa omat käyttöliittymäsivut periytetään kantaluokasta. [8]

Prototyypivaiheessa hallintalaitteen ohjelmiston graafinen suunnittelu jäi vähäiselle huomiolle ja käyttöliittymän toteuttamiseen käytettiin suurimmaksi osaksi valmiita käyttöliittymäkomponentteja, kuten näppäimiä, tekstikenttiä ja valintalaatikoita. Tämä toisaalta lisää yhtenäisyyttä itse Windows Phone käyttöliittymään ja sen sisältämiin esiasennettuihin ohjelmiin. Kuvassa 5.5 on esitetty ohjelmiston valmis käyttöliittymä prototyypivaiheen lopulla. Kuvassa vasemmalla on sekvenssiammuntanäkymä ja oikealla näkymä, jossa uusi ammuntasekvenssi luodaan.



**Kuva 5.5** Hallintalaitteen käyttöliittymän sekvenssiammuntanäkymä ja uuden sekvenssin luominen

## 5.5 Tietoliikenneyhteyksien toteutus

Järjestelmän eri osien väliset tietoliikenneyhteydet vaativat oikeiden asetusten käyttöä toimiakseen. Tauluohjainten mikro-ohjaimen ja RN171-piirin välisen UART-väylän nopeutena käytettiin 38400 baudia. Väylän muut käytettävät asetukset olivat 8 databittiä, 1 stop bitti ja ei pariteettitarkistusta. Nämä asetukset määritellään mikro-ohjaimen ohjelmistossa.

Järjestelmän kommunikointi perustuu siihen, että kaikki sen osat liitetään samaan langattomaan lähiverkkoon. Hallintalaite on siis liitettävä käytettävään lähiverkkoon ennen hallintaohjelmiston käynnistämistä. Ennen hallintaohjelmiston käynnistämistä on myös kytkettävä tauluohjainten virrat, jolloin ne liittyvät lähiverkkoon ottamalla yhteyttä niiden muistiin tallennettuun liityntäpisteeseen.

WLAN-moduulin RN171 käyttö vaatii verkkoon liittämisen ja UART-asetusten lisäksi TCP/IP-pakettien lähetysasetukset. Asetusten tekeminen suoritettiin käyttämällä moduulin tarjoamaa web-sovellusta. Web-sovellus käynnistetään nostamalla jännite moduulin ulostulossa GPIO9 nolasta käyttöjännitteen tasolle hetkellisesti. Tätä varten kytketään liitettiin painonäppäin piirilevyn suunnitteluvaiheessa. Kun web-sovellus käynnistyy, muodostaa moduuli oman liityntäpisteensä, johon otetaan yhteyttä esimerkiksi tietokoneella, jonka jälkeen web-sovellukseen päästään käsiksi web-selaimella. Taulukossa 6 on esitelty kaikki piirin RN171 käyttöön määritellyt asetukset ja niiden selitykset [9].

**Taulukko 6.** Piiriin RN171 vaatimat asetukset [9]

Komento	Kuvaus
\$\$\$	Siirtyy komentotilan
set wlan phrase <i>salasana</i>	Asettaa käytettävän verkon salasanan
set wlan ssid <i>verkkotunnus</i>	Asettaa käytettävän verkon tunnuksen
set wlan tx 6	Asettaa lähetystehon desibeleissä noin puoleen maksimista
set wlan chan 0	Asettaa kanavan. Jos 0, verkkoa haetaan kaikilta kanavilta
set wlan join 1	Asettaa tallennettuun verkkoon liittymisen automaattiseksi
set uart baud 38400	Asettaa UART-nopeudeksi 38400 baudia
set uart flow 0x00	Poistaa toiminnot flow control ja pariteettitarkistus käytöstä
set comm match 0x0d	Asettaa paketin lähetyksen alkavaksi merkillä CR ( <i>carriage return</i> )
save	Tallentaa asetukset moduulin muistiin
reboot	Käynnistää moduulin uudelleen
exit	Poistuu komentotilasta (ei tarvita uudelleen käynnistyksen jälkeen)

## 5.6 Valmistuskustannukset

Tauluohjaimen prototyypin valmistuskustannukset koostuivat elektroniikan ja mekaniikan kustannuksista. Elektroniikan kustannukset koostuivat piirilevyjen valmistuskus-

tannuksista sekä komponenttien kustannuksista. Taulukossa 7 on esitetty yhteenveto tauluohjaimen kustannuksista.

***Taulukko 7. Yksittäisen tauluohjaimen kustannukset***

<b>Osa tai komponentti</b>	<b>Kustannukset [€]</b>
Mikrofoni	1,29
Diodi (2 kpl)	0,404
Transistori	0,149
Piirilevyn liittimet (4 kpl)	1,084
Tantaalikondensaattori (2 kpl)	0,362
Keraaminen kondensaattori (7 kpl)	0,107
Pintaliitosvastus (13 kpl)	0,199
WLAN-moduuli RN171	23,06
Siruantenni	0,574
Mikro-ohjain Atmega644PA	7,88
Painonäppäin (3 kpl)	0,819
Jänniteregulaattori	0,62
Virtakatkaisin	0,60
Piirilevy	20,33
Kotelon materiaalit	n. 10,00
Servo	7,50
Yhteensä:	<b>n. 75,00</b>

Yhden tauluohjaimen prototyypin valmistuksen kokonaiskustannukset ovat noin 75 euroa. Tämä on noin kaksinkertainen summa projektin alussa määriteltyihin materiaali- ja komponenttikustannuksiin nähden. Kustannuksissa pystytään kuitenkin säästämään huomattavasti, kun ostettavien erien suuruus kasvaa.

## 6. KÄYTTÖTESTIT JA MITTAUKSET SEKÄ NIIDEN MERKITYS

Tässä luvussa kuvataan valmiin prototyypin testikäytön kokemuksia sekä mittaukset jotka toteutettiin järjestelmän ominaisuuksien selvittämiseen. Luvussa 6.1 kuvataan ensin testikäyttö. Tämän jälkeen luvussa 6.2 kuvataan mittaukset ja luvussa 6.3 analysoidaan niiden merkitystä.

### 6.1 Testikäyttö

Järjestelmää testatessa käytössä oli neljä tauluohjainta. Hallintalaitteena käytettiin Nokia Lumia 1520 älypuhelin. Lisäksi testitilanteessa käytössä oli kolme erilaista asetta sekä pahvisia maalitauluja. Yksi aseista oli edullinen muovinen ase, joka kykenee ampumaan vaan yksittäisiä muovikuulia siten, että jokaisen kuulan välillä ase on ladattava manuaalisesti. Toinen ase oli vaativammalle harrastelijalle sopiva metallirunkoinen ase, jossa on myös sarja-ammuntatoiminto. Ase ei vaadi manuaalista lataamista, vaan toimii sähkömoottorin avulla ja saa käyttövoimansa uudelleenladattavasta akusta. Molemmat aseet ampuvat muovikuulia, joiden halkaisija on 6 millimetriä. Kolmas ase oli vanha metallinen manuaalisesti ladattava ase, joka ampuu kuparisia kuulia, joiden halkaisija on 4 millimetriä.

Kokemukset testikäytöstä olivat hyviä. Järjestelmän käyttö vastasi odotuksia ja helpotti harjoittelua. Ammuntasekvenssien käyttö teki ammuntaharjoittelusta mielenkiintoisemman ja onnistui innostamaan käyttäjän haastamaan itsensä parempien tulosten tekemiseen.

Järjestelmän käyttömukavuutta todettiin testissä rajoittavan tapa, jolla tauluohjaimet yhdistetään hallintalaitteen kanssa. Jotta tauluohjaimen voidaan yhdistää, on sen IP-osoite tunnettava. Tauluohjain saa IP-osoitteen liittyessään reitittimen liityntäpisteeseen. Näin IP-osoite saadaan selville vain avaamalla reitittimen tarjoama web-sovellus, josta verkkoon liitetyt laitteet ja niiden IP-osoitteet voidaan listata.

### 6.2 Mittaukset

Luotettavuus- ja vasteaikatestit toteutettiin sisätiloissa noin 20 metriä pitkässä ja 8 metriä leveässä avoimessa tilassa. Luvussa 6.2.1 kuvataan suoritettavat vasteaika- ja nopeusmittaukset sekä niiden tulokset ja luvussa 6.2.2 luotettavuustestit ja niiden tulokset.

## 6.2.1 Vasteaika- ja nopeusmittaukset

Vasteaikamittausten tarkoituksena oli selvittää, kuinka järjestelmälle määritellyt vasteaikatavoitteet toteutuivat prototyypilaitteissa. Mittaukset tehtiin käyttämällä kameraa, jolla voidaan kuvata videokuvaa taajuudella 240 Hz. Tällä taajuudella kuva otetaan noin 4,2 millisekunnin välein. Tapahtuma-aika voidaan määrittää tarkastelemalla videota kuva kovalta ja kirjaamalla ylös kuvakehyksen järjestysnumero tapahtumahetkellä.

Vasteaikamittauksissa mitattiin taulun kaatumisaika osuman jälkeen. Taulun kaatuessa tauluohjain havaitsee osuman ja lähettää siitä tiedon hallintalaitteelle. Hallintalaitteeseen vastaanottaa tiedon osumasta, ja lähettää tauluohjaimelle vahvistuksen tiedon saapumisesta sekä komennon piilottaa tauluohjain. Samalla mitattiin hallintalaitteen käyttöliittymän pistelaskurin reagointiaikaa. Mittaukset suoritettiin kahdelle tauluohjaimelle siten, että toistoja tehtiin kymmenen kummallekin. Taulukossa 8 on listattu vasteaikamittausten tulokset.

*Taulukko 8. Vasteaikamittausten tulokset*

Laite	Testi	Taulun kaatumisaika kuvakehyksinä	Pistelaskurin reagointiaika kuvakehyksinä
1	1	20	32
	2	21	47
	3	28	41
	4	28	46
	5	14	30
	6	70	64
	7	15	25
	8	21	42
	9	36	47
	10	30	55
2	1	41	52
	2	18	27
	3	24	42
	4	21	39
	5	25	15
	6	14	33
	7	17	28
	8	18	44
	9	16	31
	10	12	31

Vasteaikamittausten yhteydessä selvitettiin myös käytetyn aseiden sarja-ammunnanopeus. Mittaukset tehtiin samalla laitteistolla ja menetelmällä kuin vasteaikamittaukset. Ammunnanopeus selvitettiin ampumalla lyhyitä sarjoja ja mittaamalla aseesta lähtevien kuulien lähtöaika edellisen kuulan lähdön jälkeen. Taulukossa 9 on listattu mittausten tulokset.

**Taulukko 9.** Sarja-ammunta vasteajat

Sarja	Kuulanumero	Lähtöaika kuvakehyksinä
1	1	-
	2	26
	3	25
	4	24
	5	25
2	1	-
	2	25
	3	25
	4	26
	5	24

Kolmannessa testissä mitattiin tauluohjaimen servojen nopeutta. Tarkoitus oli selvittää taulun piilottamiseen ja esille kääntämiseen kuluva aika. Mittaukset tehtiin kahdella tauluohjaimella. Taulukossa 10 on listattu mitatut taulujen nousu- ja laskuajat.

**Taulukko 10.** Maalitaulua ohjaavan servon nopeus

Laite	Testi	Laskuaika kuvakehyksinä	Nousuaika kuvakehyksinä
1	1	92	92
	2	94	94
	3	92	95
2	1	110	124
	2	113	108
	3	110	118

## 6.2.2 Sarja-ammuntatesti

Sarja-ammuntatestin tarkoituksena oli tutkia järjestelmän luotettavuutta sarja-ammunnassa. Sarja-ammuntatoiminnon käyttö kasvattaa kommunikoinnin nopeuden vaatimuksia normaaliin ampumiseen nähden. Testi toteutettiin ampumalla käytettävissä olevan aseensa suurimmalla mahdollisella sarjalla tauluihin osumia ja kirjaamalla ylös tauluosumien määrä sekä hallintalaitteen rekisteröimä osumamäärä. Tarkoituksena oli selvittää kuinka hyvin järjestelmä kykenee rekisteröimään osumat nopeassa sarja-ammunnassa. Testissä käytettiin samaa nopeuskameraa kuin vasteaikatesteissä todellisten osumien lukumäärän laskemiseen. Taulukossa 11 on esitelty testin tulokset.

**Taulukko 11.** Osumien rekisteröinnin todennäköisyys sarja-ammunnassa

Laite	Testi	Sarjan koko	Osumien määrä	Rekisteröity määrä
1	1	29	26	26
	2	29	27	27
	3	29	29	29
2	1	29	29	11
	2	29	27	17
	3	29	26	12

### 6.3 Mittaustulosten merkitys

Taulukossa 8 esitetyistä mittaustuloksista voidaan laskea keskimääräinen aika, joka kuuluu maalitaulun laskemiseen osuman jälkeen sekä aika, joka kuluu hallintalaitteen pistelaskurin päivittymiseen. Ajat on kirjattu taulukkoon kuluneina kuvakehyksinä, joten tapahtuma-ajat voidaan määrittää kuvakehyksien määrästä ja jakamalla määrä näytteenottotaajuudella seuraavan kaavan mukaisesti:

$$T_t = \frac{N_{kuva}}{f_{näyte}}, \quad (1)$$

jossa  $T_t$  on tapahtuma-aika,  $N_{kuva}$  on kuvakehysten lukumäärä ja  $f_{näyte}$  on näytteenottoaajuus.

Keskiarvoinen kuvakehyksen järjestysnumero maalitaulun laskeutumisen hetkellä on ensimmäisellä tauluohjaimella 28,3. Ajaksi saadaan siis  $28,3/240 \text{ Hz} \approx 118 \text{ ms}$ . Toisen tauluohjaimen keskiarvoiseksi kuvakehyksen järjestysnumeroksi saadaan 20,6, ja ajaksi  $20,6/240 \text{ Hz} \approx 86 \text{ ms}$ . Keskiarvoinen kuvakehyksen järjestysnumero hallintasovelluksen käyttöliittymän pistelaskurin päivittymishetkellä on ensimmäisellä tauluohjaimella 42,9, jolloin päivittymisajaksi saadaan  $42,9/240 \text{ Hz} \approx 179 \text{ ms}$ . Toisen tauluohjaimen vastaava kuvakehyksen järjestysnumero on 34,2 ja aika on  $34,2/240 \text{ Hz} \approx 143 \text{ ms}$ .

Vasteaikamittausten tulosten perusteella voidaan todeta, että järjestelmälle määritellyt vaatimukset täytettiin sekä maalitaulun kaatumiseen kuluvan ajan että pistelaskurin päivittymisnopeuden osalta. Mittaustilanteessa molemmat tauluohjaimet selviytyivät molemmista tehtävistä käyttämällä vain noin puolet maksimijasta.

Testissä käytetyn aseiden sarja-ammuntanopeus voidaan laskea taulukon 9 tuloksista laskeamalla ensin keskiarvoinen lähtöhetki kuvakehyksinä ja muuttaa tulos ajaksi käyttämällä kaavaa (1). Ensimmäisen testisarjan keskiarvoiseksi kuvakehyksen järjestysnumeroksi saadaan 25,0, ja ajaksi saadaan noin 104 ms. Toisen testisarjan keskiarvoinen kuvakehyksen järjestysnumero on myös 25,0 ja näin ajaksi saadaan jälleen noin 104 ms. Ase kykenee siis ampumaan noin 10 kuulaa/s.

Maalitaulujen nousu- ja laskunopeudet voidaan laskea taulukon 10 tuloksista. Ensimmäisen tauluohjaimen maalitaulun keskiarvoinen laskuhetki kuvakehyksien lukumääränä on 92,7 ja nousuhetki 93,7. Näin laskuajaksi saadaan  $92,7/240 \text{ Hz} \approx 386 \text{ ms}$  ja nousuajaksi  $93,7/240 \text{ Hz} \approx 390 \text{ ms}$ . Toisen tauluohjaimen keskiarvoinen laskuhetki kuvakehyksien lukumääränä on 111 ja nousuhetki 117. Laskuajaksi saadaan  $111/240 \text{ Hz} \approx 460 \text{ ms}$  ja nousuajaksi  $117/240 \text{ Hz} \approx 490 \text{ ms}$ . Tuloksista voidaan todeta, että yksittäisen servon nopeus on lähes sama lasku- ja noususuuntiin. Servoyksilöissä on kuitenkin selviä nopeuseroja.

Taulukon 11 tuloksista voidaan helposti todeta, että ensimmäinen tauluohjain suoriutui sarja-ammuntatestissä osumien rekisteröinnistä 100 %:n tarkkuudella. Toinen tauluohjain kuitenkin rekisteröi 82:sta osumasta vain 40. Osumista rekisteröitiin siis hieman alle 50 %. Nämä tulokset kertovat siitä, että järjestelmä ei ole vielä luotettava sarja-ammuntatoimintoa käytettäessä. Ongelmaan on perehdyttävä järjestelmän jatkokehitysvaiheessa.



## 7. JATKOKEHITYS

Tässä luvussa pohditaan järjestelmän jatkokehitysmahdollisuuksia ja –tarpeita, etenkin kaupallistamisen näkökulmasta. Luvussa 7.2 käsitellään tauluohjainta ja luvussa 7.3 hallintalaitteen ohjelmistoa koskevaa jatkokehitystä.

### 7.1 Tauluohjaimen jatkokehitys

Suurin käytön mukavuutta ja helppoutta rajoittava tekijä on tauluohjainten verkkoon liittämisen monimutkaisuus. Kuten nykyiset reitittimet tavallisesti, myös työssä käytetty WLAN-moduuli kuitenkin tukee WPS-toiminnon painonäppäinmetodia, (engl. Wi-Fi Protected Setup), joka mahdollistaa helpon verkkoon liittymisen. Metodissa verkkoon voidaan liittyä painamalla samanaikaisesti reitittimen sekä liitettävän laitteen kotelossa olevia painonäppäimiä [10]. Tällainen näppäin on toteutettu tauluohjaimen piirilevyille, mutta käyttöä helpottaisi jos näppäin siirrettäisiin tauluohjaimen koteloon.

Tauluohjainten jatkokehitysvaiheessa tarvitaan enemmän testausta mahdollisten epäkohtien löytämiseen. Tärkeää olisi myös valmistaa tauluohjaimia suurempi määrä, jotta järjestelmän käyttöä voitaisiin testata samanaikaisesti ohjattavien tauluohjainten maksimimäärällä.

Järjestelmän mittauksissa ilmennyt tauluohjainten epävarmuus käytettäessä sarja-ammuntatoimintoa vaatii ongelmaan perehtymistä. Hyödyllistä olisi myös testata järjestelmää aseilla, joiden sarja-ammuntanopeus on suurempi kuin testeissä käytetyllä aseella.

### 7.2 Hallintalaitteen ohjelmiston jatkokehitys

Hallintalaitteen ohjelmistoon jäi paljon parantamisen varaa. Etenkin ulkoasuun ja käytettävyyteen onkin hyvä tehdä parannuksia ennen mahdollista järjestelmän kaupallistamista. Esimerkiksi grafiikan lisääminen vapaa-ammunta- ja sekvenssiammuntatoimintojen ammuntaäkymään käytössä olevien asetusten indikoimiseksi toisi käyttöliittymään selkeyttä. Myös käyttöliittymäelementtien, kuten näppäinten ja tekstikenttien kokoon olisi syytä kiinnittää enemmän huomiota, etenkin älypuhelinien suurten näyttökokoerojen vuoksi. Käyttöliittymän selkeyttä voitaisiin lisätä myös paremmilla värivalinnoilla.

Ohjelman vakauden kannalta olisi myös paljon parannuksia tehtävissä. Tärkein näistä lienee virhetilanteiden käsittely. Ohjelmisto ei huomioi kaikkia mahdollisia tilanteita. Yksi näistä on yhteyden katkeaminen tauluohjaimeen, esimerkiksi tilanteessa, jossa

tauluohjaimen virta katkeaa. Tällöin hallintalaitteen ohjelmiston tulisi havaita tauluohjaimen yhteyden katkeaminen ja huomauttaa siitä jollain tavalla käyttöliittymässä, ja antaa käyttäjälle mahdollisuus jatkaa ohjelman käyttöä jäljellä olevilla tauluohjaimilla.

Tarvetta olisi myös monien lisäominaisuuksien toteuttamiselle. Järjestelmä onkin suunniteltu siten, että järjestelmään voidaan lisätä ominaisuuksia pelkästään hallintalaitteen ohjelmistoa päivittämällä. Yhtenä hyödyllisenä lisäominaisuutena voitaisiin pitää erillisten tauluohjainten osumien laskemista. Nykyversiossa pisteet näytetään käyttöliittymässä vain kaikkien käytössä olevien tauluohjainten osumien kokonaispisteinä. Osumista voitaisiin myös tehdä jonkinlainen lokitiedosto, josta voitaisiin nähdä aikajärjestyksessä kaikki sekvenssiammunnan aikana tapahtuneet osumat ja niiden tapahtuma-ajat.

## 8. YHTEENVETO

Tässä työssä suunniteltiin ja valmistettiin verkotettu maalitaulujärjestelmän prototyyppi muovikuula-aseille. Toteutettu järjestelmä koostuu valmiista hallintalaitteesta, jolle suunniteltiin hallintaohjelmisto, tauluohjainlaitteista sekä laitteiden välisestä radiolinkistä. Radiolinkin tyyppinä käytettiin WLAN-verkkoa, jonka vuoksi järjestelmän käyttöön tarvitaan hallintalaitteen ja tauluohjainten lisäksi langaton liityntäpiste, johon laitteet liitetään. Liityntäpisteenä voidaan käyttää tavallista reititintä.

Järjestelmän käyttötarkoitus on toimia muovikuula-aseiden harrastus- ja harjoitteluvälineenä. Käyttäjä pyrkii ampumaan maalitauluihin osumia, jolloin hallintalaitte laskee osumapisteet sekä nostaa ja laskee maalitauluja automaattisesti käyttäjän määrittämin ehdoin. Ammunta voidaan suorittaa vapaana ammutana, jolloin se mahdollistaa on jatkuvan harjoittelun, tai sekvenssiammutana, jolloin käyttäjä määrittää ammunnan keston aikana sekunteina tai maalitaulujen näyttömääränä.

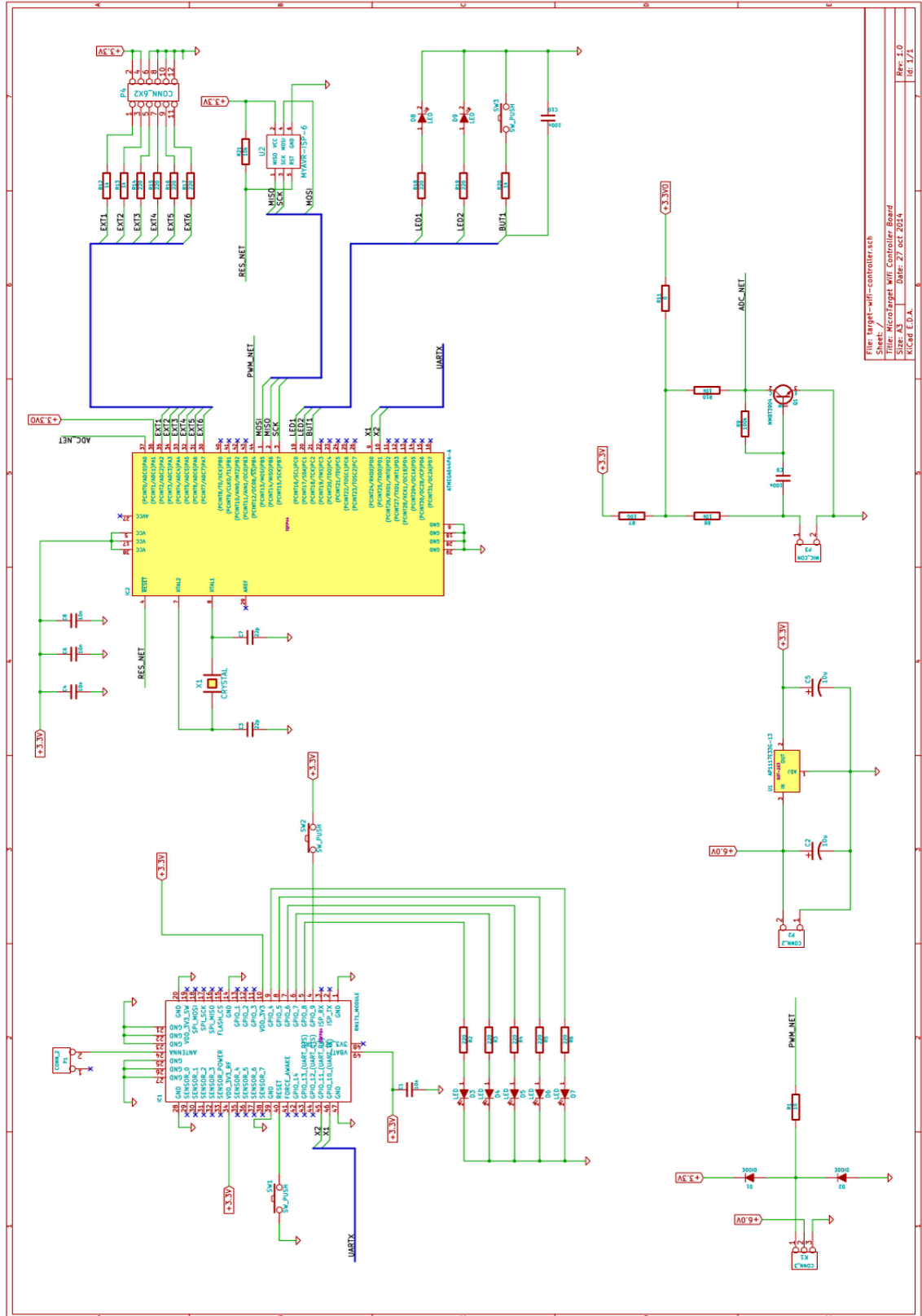
Hallintalaitteen ohjelmisto toteutettiin prototyyppivaiheessa Windows Phone –alustalle käyttäen Visual Studio 2013 –ohjelmistokehitysympäristöä. Ohjelmalogiikka toteutettiin C#-kielellä ja ohjelmiston käyttöliittymä XAML-kielellä. Tauluohjaimen mikroohjaimen ohjelmisto toteutettiin C-kielellä.

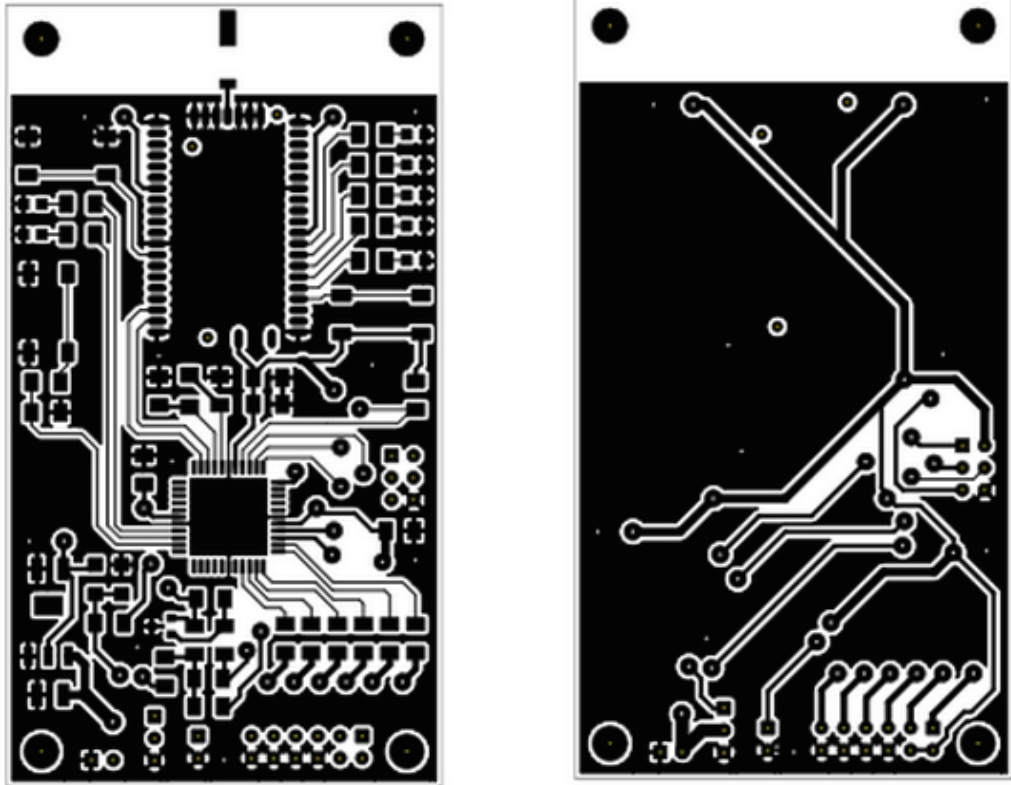
Tauluohjainten prototyyppijä valmistettiin testikäyttöön neljä kappaletta. Käyttötesteissä järjestelmän todettiin toimivan tavoitteeksi asetetulla tavalla ja käyttö ammutatilan-teessa vastasi odotuksia. Parantamisen varaa jäi kuitenkin osumien rekisteröinnin luotettavuuteen sarja-ammuntanopeuksilla. Myös menetelmä, jolla tauluohjaimet yhdistetään hallintalaitteen kanssa vaatii vielä kehitystä, ennen kuin järjestelmä saadaan tuntumaan valmiilta.

## LÄHTEET

- [1] *Airsoft Accesories*. Crosman. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://www.crosman.com/airsoft/accessories>
- [2] *MT Company's Blog: Electronic Auto-Reset Airsoft Gun Target*, 2011. MT Company. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://www.mtcompanystore.com/blog> > Articles
- [3] Arie van Rhijn, *Integrated Circuits for High Performance Electret Microphones*, Audio Engineering Society Convention Paper, 114<sup>th</sup> Convention, 2003. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://educyclopedia.karadimov.info/library/aes2003.pdf>
- [4] Storey, N. *Electronics – A System Approach*. Second edition. Pearson Prentice Hall. 1998. ISBN 0-201-17796-X
- [5] Sawicz, D. *Hobby Servo Fundamentals*. [WWW]. [viitattu 15.12.2014]. Saatavissa: <https://www.princeton.edu/~mae412/TEXT/NTRAK2002/292-302.pdf>
- [6] *Bluetooth Basics – A Look at the Basics of Bluetooth Technology*. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://www.bluetooth.com/Pages/Basics.aspx>
- [7] *RN171 2.4 GHz IEEE Std. 802.11 b/g Wireless LAN Module*, 2014. Microchip inc. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://ww1.microchip.com/downloads/en/DeviceDoc/70005171A.pdf>
- [8] *Dev Center – Windows: API Reference*. Microsoft. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://msdn.microsoft.com/library/windows/apps/bg124285.aspx>
- [9] *WiFly Module Command Reference, Advanced Features and Applications User's Guide*, 2014. Microchip inc. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://ww1.microchip.com/downloads/en/DeviceDoc/50002230A.pdf>
- [10] *How does Wi-Fi Protected Setup work?* Wi-Fi Alliance. [WWW]. [viitattu 15.12.2014]. Saatavissa: <http://www.wi-fi.org/knowledge-center/faq/how-does-wi-fi-protected-setup-work>

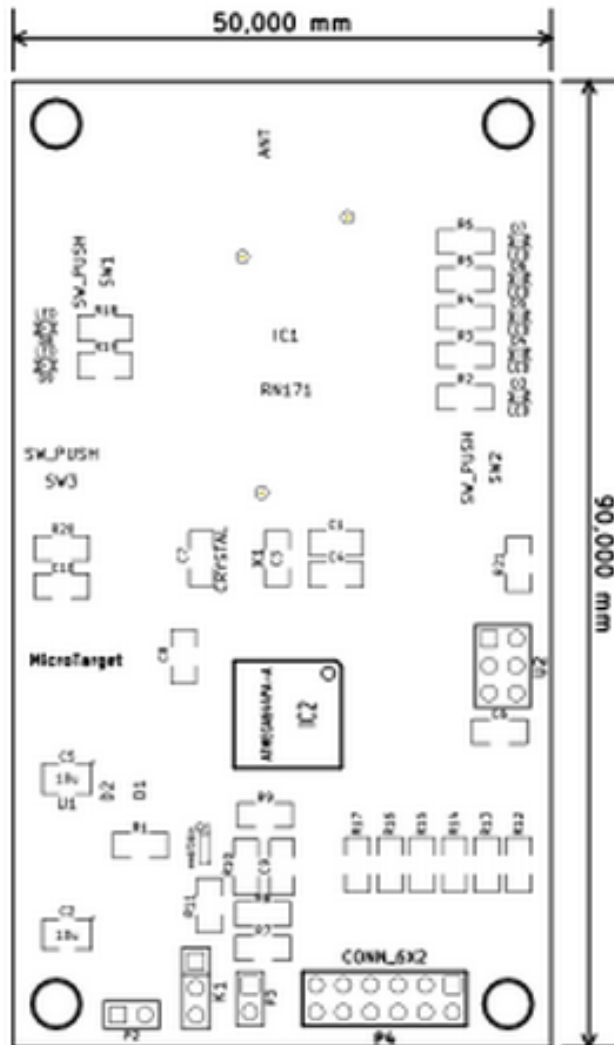
# LIITE A: TAULUOHJAIMEN KYTKENTÄKAAVIO



**LIITE B: TAULUOHJAIMEN PIIRILEVYN JOHDOTUSKUVAT**

*1. Tauluohjaimen johdotuskuvat. Oikealla piirilevyn etupuoli ja vasemmalla takapuoli.*

## LIITE C: TAULUOHJAIMEN PIIRILEVYN OSASIJOTTELUKUVA



## LIITE D: TAULUOHJAIMEN PIIRILEVYN KOMPONENTTILUETTELO

1	C1	10n	SM1206
2	C2	10u	SM1210
3	C3	22p	SM1206
4	C4	10n	SM1206
5	C5	10u	SM1210
6	C6	10n	SM1206
7	C7	22p	SM1206
8	C8	10n	SM1206
9	C9	100n	SM1206
10	C10	100n	SM1206
11	D1	DIODE	BYG10G-E3
12	D2	DIODE	BYG10G-E3
13	D3	LED	LED-0805
14	D4	LED	LED-0805
15	D5	LED	LED-0805
16	D6	LED	LED-0805
17	D7	LED	LED-0805
18	D8	LED	LED-0805
19	D9	LED	LED-0805
20	IC1	RN171_MODULE	RN171
21	IC2	ATMEGA644PA-A	TQFP44
22	K1	CONN_3	PIN_ARRAY_3X1
23	P1	CHIP_ANTENNA	CHIP_ANTENNA_PRINT
24	P2	CONN_2	PIN_ARRAY_2X1
25	P3	MIC_CONN	PIN_ARRAY_2X1
26	P4	CONN_6X2	PIN_ARRAY_6X2
27	Q1	MMBT3904	SOT23
28	R1	10	SM1206
29	R2	220	SM1206
30	R3	220	SM1206
31	R4	220	SM1206
32	R5	220	SM1206
33	R6	220	SM1206
34	R7	10G	SM1206
35	R8	10K	SM1206
36	R9	100K	SM1206
37	R10	10K	SM1206
38	R11	0	SM1206
39	R12	1K	SM1206
40	R13	1K	SM1206
41	R14	220	SM1206
42	R15	220	SM1206
43	R16	220	SM1206
44	R17	220	SM1206
45	R18	220	SM1206
46	R19	220	SM1206
47	R20	1K	SM1206
48	R21	10K	SM1206
49	SW1	SW_PUSH	SW_PUSH_PRINT
50	SW2	SW_PUSH	SW_PUSH_PRINT
51	SW3	SW_PUSH	SW_PUSH_PRINT
52	U1	AP1117E33G-13	SOT223
53	U2	AVR-ISP-6	PIN_ARRAY_3X2
54	X1	CRYSTAL	CRYSTAL_PRINT