

Niko Kaukonen

AIRSOFT-AJASTIN- SEKÄ PELILAITE

**Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Tietotekniikka
Syyskuu 2017**

TIIVISTELMÄ

Centria Ammattikorkeakoulu	Päivämäärä Syyskuu 2017	Tekijä Niko Kaukonen
Koulutusohjelma Tietotekniikka		
Työn nimi AIRSOFT-AJASTIN- SEKÄ PELILAITE		
Ohjaaja Sakari Männistö	Sivumäärä 22+2	
Työelämäohjaaja Jari Mustasaari		
<p>Mikrokontrollerit mielletään yleensä käytettäväksi eri tarkoituksissa, kuten suurempien kokonaisuuksien valvomiseen tai ohjaukseen pienellä toimilaitteella, niitä voidaan hyödyntää myös pienemmän skaalan projekteissa laitteiston päätoimisena yksikkönä. Tässä opinnäytetyössä keskitytään Arduino-mikrokontrolleri/elektroniikka-alustan hyödyntämiseen Airsoft-peliin tarkoitettussa ajastinlaitteessa.</p> <p>Tarveperusta ajastinlaitteelle ei sinänsä ole uusi idea, vaan on ollut yhteisöjen puheenaiheena jo jonkin aikaa, mutta vasta viimeisten vuosien aikana ajastimen rakentaminen pieneen kokoluokkaan on alkanut olla järkevää. Opinnäytetyössä ajastinlaite rakennettiin ensin realisoimalla sen tarve, kartoittamalla tarpeet täyttävän laitteen ominaisuudet ja toiminnallisuus, jonka jälkeen laitteen suunnittelu ja rakentaminen voitiin aloittaa. Laitteen testaaminen tehtiin sekä rakentamisen että laitteen valmistumisen jälkeen, jolloin laite todettiin valmiiksi pelikäyttöön.</p> <p>Opinnäytetyön tarkoitus oli tutkia ja perehtyä Airsoft-pelijaajastinlaitteen tarveperustaan, mahdollisiin olemassa oleviin vaihtoehtoihin sekä suunnitella ja rakentaa ajastinlaite Airsoft-pelejä järjestävälle ja toimintaa edistävälle West Coast Airsoft-yhdistykselle.</p> <p>Projekti saatiin lopulta valmiiksi ja valmistunut laite täyttää pelin luonteen asettamat vaatimukset, mukaan lukien laajennettavuuden tulevaisuuden laajennusten sekä muokkausten kannalta.</p>		

Asiasanat

Arduino, ajastin, Airsoft, West Coast Airsoft, Paintball

ABSTRACT

Centria University of Applied Sciences	Date September 2017	Author Niko Kaukonen
Degree Program Information Technology		
Name of thesis GAME TIMER AND GAME DEVICE		
Instructor Sakari Männistö		Pages 22+2
Supervisor Jari Mustasaari		
<p>Microcontrollers are widely used for multiple purposes, such as controlling or operating larger, more complex machinery or circuitry, it is also possible to use them for smaller-scale projects as a main operating unit to handle instructions and main functionality.</p> <p>This thesis will concentrate on using Arduino to create a timing device to be used in team-based airsoft survival games. The need for such a device was first realized long ago, but only now with small-scale embedded systems can it be implemented. In this thesis, a timing device was built by first realizing the need for such a device, defining its requirements and properties and then finally sketching and building the device. Testing was conducted both while building it and after it was deemed ready to be used in games.</p> <p>The purpose was to investigate the need of such a device, possible available alternatives in the market currently available and sketch and build a timing device for the West Coast Airsoft association. The project was successfully completed and the resulting device fulfilled all of the pre-determined requirements and properties set for the device, including future-proofing for expansion and modification.</p>		

<p>Key words Arduino, timer, Airsoft, West Coast Airsoft, Paintball</p>
--

KÄSITTEIDEN MÄÄRITTELY

LCD	Liquid Crystal Display. Nestekidenäyttö. Koostuu useista ohuista kalvomaisista tasoista, joiden tuloksena saadaan tuotettua kuvaa näytön pinnalta katsottavaksi.
WCAS	West Coast Airsoft. Keski-Pohjanmaan alueella vaikuttava rekisteröimätön Airsoftyhdistys. Jäseniä Keski-Pohjanmaan alueella yli sata.
Shield	Arduinon käyttämä lisämoduulijärjestelmä, jonka avulla Arduinoon voi lisätä toimintoja erilaisilla lisäpiireillä.
Airsoft	Joukkuepohjainen Japanissa kehitetty ryhmähenkeä kasvattava selviytymispeli. Rinnastettavissa jos Paintballin kaltaisiin lajeihin.
IDE	Integrated Development Environment. Ohjelmistoympäristö / kehitysympäristö, jonka avulla projekteja tuotetaan käytettäväksi.

TIIVISTELMÄ
ABSTRACT
KÄSITTEIDEN MÄÄRITTELY
SISÄLLYS

1 JOHDANTO	1
2 AIRSOFT LAJINA	2
2.1 Lajin historia ja kehitys Suomessa	2
2.2 Lajin eri pelimuodot ja lajille ominaiset piirteet	3
3 KEHITYSALUSTAT.....	5
3.1 Arduino	5
3.2 Raspberry Pi	6
3.3 Arduino Uno	7
4 KEHITYSYMPÄRISTÖT	8
4.1 Microsoft Visual Studio 2013	8
4.2 Arduino IDE	9
5 LAITTEEN KUVAUS SEKÄ TOIMINTA	11
5.1 Tarveperusta laitteelle	11
5.2 Laitteen toimintaperiaate	11
5.3 Laitteen toteutus.....	15
5.3.1 Tarpeet täyttävän laitteen suunnittelu ja rakentaminen.....	16
5.3.2 Keskeiset komponentit ja niiden perustoiminta.....	16
5.3.3 Ohjelmoinnin keskeiset seikat.....	18
5.4 Laitteen testaaminen	20
5.5 Jatkokehitysmahdollisuudet.....	21
6 JOHTOPÄÄTÖKSET JA POHDINTA.....	22
LÄHTEET	23

LIITTEET

KUVAT

KUVA 1. Arduino UNO	7
KUVA 2. Microsoft Visual Studio 2013	8
KUVA 3. Arduino IDE	10
KUVA 4. Salasanan syöttö ja tarkistus	13
KUVA 5.LCD-paneeli	16
KUVA 6.Matriisinäppäimistö	17
KUVA 7.Näppäimistön matriisikaavio	18
KUVA 8.Setup-funktio	19
KUVA 9.Ajanlaskentarakenne.....	19

1 JOHDANTO

Opinnäytetyö käsittelee Airsoft-harrastusta sekä sille kehitettävää ja toteutettavaa pelilaitetta, jolle syntyneen tarpeen ajattelin realisoida opinnäytetyönä Keski-Pohjanmaan alueella toimivalle rekisteröimättömälle West Coast Airsoft -yhdistykselle. Työn tavoitteena on kartoittaa kyseisen laitteen toimintatarpeet, suunnitella tarpeet täyttävä laite vastaamaan tarpeisiin mahdollisimman hyvin ja toteuttaa suunniteltu laite rakentamalla se Arduino-kehitysalustaa pohjana käyttäen. Samalla työssä käsitellään Airsoftia lajina sekä sen historiaa ja nykyistä tilannetta Keski-Pohjanmaan alueella ja koko Suomessa.

Aluksi työssä käsitellään Airsoftia harrastuksena ja esitellään laji pääpiirteittäin pureutumalla sen historiaan, luonteeseen sekä lajin kehitykseen ja tulevaisuusnäkyymiin. Samalla käydään lävitse lajissa käytettäviä varusteita sekä pelivälineitä ja yleisiä pelisääntöjä ja pelaajakantaa.

Työssä käsitellään eri vaihtoehtoja toteutukselle, ongelmia suunnittelussa ja rakennusprosessissa sekä rakennuksen aikana ja sen jälkeen tehtyjä havaintoja.

Työ käsittää laitteen suunnittelun, rakentamisen komponenttitasolta sekä laitteelle asetettujen toimintojen täyttävän, myöhemmin mahdollisesti päivitettävän ja kehitettävän ohjelmiston luomisen laitteelle. Työ soveltaa ja kartoittaa jo koulussa opittuja elektroniikan, ohjelmoinnin sekä suunnittelun taitoja sekä auttaa tutkimaan eri ratkaisuvaihtoehtoja kyseiselle laitteelle ja sen toiminnoille. Työn aikana on tarkoitus valita parhaiten tehtävään soveltuva vaihtoehto eri kehitysalustoista ja pohtia työn etenemistä ja suoritusta sekä rakennusvaiheessa että työn valmistuttua. Samalla selitetään myös käytettyjen komponenttien toimintaperiaate ja ominaisuudet, niiden sovellutukset työssä sekä mahdolliset vaihtoehdot ja miksi juuri nämä komponentit valittiin vaihtoehtojen joukosta.

2 AIRSOFT LAJINA

Tässä luvussa käsitellään Airsoftin historiaa ja kehitystä Suomessa. Luvussa tarkastellaan myös lajin eri muotoja ja lajille tyypillisiä piirteitä, tutustutaan lajissa käytettäviin varusteisiin ja pelivälineiden toiminnallisuuteen. Luku sisältää myös pikaisen katsauksen lajin luonteen asettamille vaatimuksille niin suojauksen kuin pelattavuuden ja turvallisuuden osilta.

2.1 Lajin historia ja kehitys Suomessa

Airsoft on alunperin Japanissa 80-luvulla keksitty laji. (Softaaja 2015.) Laji syntyi, koska Japanin lait kielsivät siviileiltä oikeiden aseiden hallussapidon, mutta sallivat replikoiden ja aseita jäljittelevien aseiden ja vastaavien laitteiden hallussapidon ja käytön. Japanilaiset replikoihin keskittyneet yritykset loivat green gas -kaasua, enimmäkseen propaania, käyttävän prototyypin, joka ampui halkaisijaltaan noin 6 millimetrin kokoisia muovipellettejä. Aseista tuli todella suosittuja japanilaisten harrastajien keskuudessa vaihtoehtona oikeille aseille ensin tarkkuusammuntatarkoitukseen ja myöhemmin ryhmäpelikäyttöön ja laji levisikin myöhemmin ympäri maailman, jopa Suomeen saakka. (YLE 2012.)

Suomeen Airsoft rantautui 90-luvun alussa. Laji alkoi kaveriporukkojen pienistä sunnuntaipeleiksi kutsutuista viikonloppupeleistä, joihin vähitellen eksyi aina vain enemmän pelaajia ja lopulta peleissä olikin jo kymmeniä pelaajia. Tämän myötä Suomeen syntyi Airsoft-aseita ja -varusteita myyviä yrityksiä, joista osa on vieläkin olemassa. 2000-luvulle tultaessa lajin suosio on vaihdellut alueellisesti. Vaikka Suomessa keskimääräisesti harrastuksen suosio on laskenut lajin alkuaikojen räjähdysmäisen kasvun myötä Keski-Pohjanmaan alueella laji ei koskaan ole ollut yhtä elinvoimainen kuin nykyään. Asiaa on auttanut paljon uusien pelaajien kiinnostus lajiin ja pelaajien vanhempien kiinnostus viettää aikaa lastensa kanssa, sekä Kokkolan ensimmäisen Airsoft-liikkeen, Airsofttrooperin perustaminen vuonna 2015. Näiden lisäksi yksityisten Airsoftiin tarkoitettujen pelialueiden löytäminen on vauhdittanut harrastusta.

Keski-Pohjanmaan alueella Airsoft on keskittynyt Kokkolaan. Kokkolassa pelialueita on tällä hetkellä kaksi; entinen puolustusvoimien asevarikon aidattu yksityisalue, jonka laajuus on noin 80 hehtaaria, sekä Lassila & Tikanoja -yrityksen läheisyydessä, aikaisemmin kettutarhana toimineen yksityisalueen aukio. Voitaisiinkin sanoa, että kirjoitushetkellä Kokkolan ulkopelialueet ovat helposti Suomen parhaimmista, ja tämän johdosta peli onkin niin elinvoimainen tällä hetkellä.

2.2 Lajin eri pelimuodot ja lajille ominaiset piirteet

Airsoftissa replikoita käytetään joukkuepeleissä hieman samalla tavalla kuin Paintballissa paintballaseita, jossa osumat merkataan maaliläiskillä, jotka syntyvät projektiilin osuessa tarpeeksi kovaan pintaan ja pallon hajotessa, jolloin sisältä purkautuu nestemäistä maalia, joka ”merkitsee” osumakohteen.(Ai-mag 2017.)

Airsoftissa maaliprojektiilin tilalla on vain muovista tai biohajoavasta muovimaisesta seosmassasta valmistettuja noin 5,95 mm:n muovikuulia, jotka osuessaan osuman ottaja tuntee osumana tai kuulee äänenä. Kuulien paino vaihtelee 0,12 g painavasta kuulasta aina yli puolen gramman tarkkuuskiväärikuuliin ja lajissa on tärkeää tietää aseeseen soveltuva kuula johtuen aseiden toimintaperiaatteen tuomasta lähtöenergiavaihtelusta. (Vyyryläinen 2015.)

Lajin harrastajalle tärkeä luonteenominaisuus on rehellisyys, sillä yleensä kuulan ampuja ei tiedä, osuiko kuula vastustajaan vai ei. Tästä johtuen onkin kehitelty erilaisia sääntöjä, joilla osumat myönnetään ja varmistetaan, kuten käden nostaminen pystyyn osuman huomattaessa ja huutamalla selkeällä äänellä: ”Osuma!”. Tämän lisäksi käytetään myös osumaliinoja, jotka ovat kirkkaita, silmiinpistäviä kankaankappaleita, jotka huomataan helposti myös maastossa.

Lajille ominaista on pelata kahden tai useamman joukkueen pelejä, joissa on tietyt ennalta määrätyt säännöt ja rajoitukset. Peleissä on yleensä tarkoituksena joko eliminoida vastapuolen joukkueen pelaajat pois kentältä osumilla, tai suorittamalla jokin ennalta-annettu tehtävä kentällä. Näihin tehtäviin voidaan laskea vaikkapa lippujen anastaminen toiselta joukkueelta, tai tietyn paikan haltuunotto. Tämän kaltainen ryhmätyöskentely edistää ryhmätyöskentelyä, luo uusia ihmissuhteita sekä tekee peliporukasta tiiviimmän. Muita pelimuotoja ovat esimerkiksi lipunryöstö, saattuetehävät ja suojelutehävät. (Softaaja 2015.) Käytännössä vain taivas on rajana, kun uusia pelimuotoja keksitään.

Airsoftia pelataan myös suurina tapahtumina, yhtenä mainittakoon Berget, joka on yksi Euroopan suurimmista tapahtumista, joka järjestetään vuosittain Ruotsissa. (Softaaja 2015) Tapahtumaan osallistuu vuosittain tuhansia pelaajia kymmenistä eri maista ja pelit kestävät yhtäjaksoisesti useamman vuorokauden. Bergetin kaltaisille tapahtumille on tyypillistä niissä esiintyvä armeijakalusto, joka vaihtelee pickup-mallisista autoista aina oikeisiin panssarivaunuihin ja helikoptereihin.

Skenaariopohjaisesta Airsoft-tapahtumasta puhutaankin yleensä ”milsiminä”, eli militariasimulaationa, jossa pyritään jäljittelemään oikeita tapahtumia tai tilanteita turvallisessa ympäristössä. Tyypillisessä suuren skaalan skenaariossa onkin oikean armeijan rakennetta jäljittelevä komentoketju ja suuri osa peliä saattaa mennä käskyjä odotellessa ja paikkaa vartioidessa. Joskus taas koko peliskenaario on toiminnantäyteinen tapahtuma. Lajissa tulee ehdottomasti käyttää kunnan suojaruustusta, jotta pelaajat välttyvät kuulien aiheuttamilta loukkaantumisilta. (Airedi 2015.)

Airsoftissa käytettävät kuulat matkaavat ilmassa parhaimmillaan sata metriä ja kuulan lähtönopeus voi olla suurimmillaan jopa 170m/s. Loukkaantumiset lajissa itse osumien johdosta ovat järkeä käyttämällä äärimmäisen harvinaisia kunnan suojaruustuksen ansiosta ja yleensä loukkaantumiset johtuvatkin kaatumisista tai muista varusteisiin liittymättömistä syistä. Joskus kuula toki osuu kohtaan, jossa vaatteet ovat kireällä tai joka on muuten vain heikosti peitetty ja osuma saattaa aiheuttaa kipeän mustelman tai pienen haavan. Nykypäivinä aseiden tekniikan kehittyttyä ja paremmin pakkasta kestävien kehittyneiden litium-polymeeriakkujen yleistyttyä Airsoft-peliä pelataan myös talvisin, jolloin liukastumisvaara kasvattaa loukkaantumisriskiä huomattavasti pelaajien juostessa liukkaalla alustalla.

3 KEHITYSALUSTAT

Tässä luvussa tutustutaan yleisesti erilaisiin pienprojekti- sekä prototyypityöskentelyissä käytettäviin kehitysalustavaihtoehtoihin, niiden ominaisuuksiin sekä tärkeimpiin eroavaisuuksiin työn kannalta. Luvussa tarkastellaan myös alustojen rajoja sekä soveltuvuutta työhön ja luodaan nopea katsaus valmistajiin. Luku käsittelee myös joissakin määrin alustojen teknisiä ominaisuuksia, enimmäkseen työlle oleellisia spesifikaatioita.

3.1 Arduino

Arduino-kehitysalustat ovat Italiassa suunniteltuja kehitysalustoja. Arduinon suunnittelu ja kehitys alkoi kun huomattiin että sen aikaiset kehitysalustat olivat kalliita ja kaupalliseen suunnitteluun ja toteutukseen tarkoitettuja ja joskus myös melko monimutkaisia käyttää. Ensimmäinen malli oli yksinkertainen piirilevy, jossa oli ATmega168 -mikrokontrolleri, jota ohjelmoitiin erillisen kehitysympäristön avulla, jolle oli kirjoitettu valmiita kirjastoja. Myöhemmin vuonna 2003 suunnittelijat siirtyivät käyttämään Atmega8-mikrokontrolleria ja nykyään eri Arduino-malleissa on käytössä joko ATmega 2560, ATmega328 tai jokin variantti. Melkein kaikissa Arduino-piirilevyissä on kuitenkin Atmelin suunnittelema mikrokontrolleri.

Arduino Uno käyttää ulkopuolisiin kytkentöihin 5V:n pinnejä ja 3V:n pinnejä. 5V:n Pinnit kestävät yksinään enintään 40mA kuormaa ja 3V pinnit taas 40mA kuormaa. Koko piirin virrankulutus ei suositusten mukaan saa ylittää 200mA. Omassa työssäni tarvitsin virtaa pääasiassa LCD-paneelille, kaiuttimelle ja led-valoile ja sain mitoitettuani kytkennän virrankulutuksen tulokseksi tuli alle 200mA, joten Arduino kelpaisi toteutukseen, mutta laajennusmahdollisuudet olisivat rajalliset.

3.2 Raspberry Pi

Raspberry Pi on Isossa-Britanniassa Raspberry Pi -hyväntekeväisyjärjestön toimesta kehitetty, vuonna 2016 julkaistu avoimen lähdekoodin yhden piirilevyn kehitysalusta. Alustan kehitys alkoi jo vuonna 2006 ideana kehittää yksinkertainen ja halpa alusta pääasiallisesti opetuskäyttöön kouluihin tietotekniikkakursseille ja myöhemmin myös yleiseen myyntiin.

Raspberry Pi pohjautuu yhden piirilevyn malliin, jossa kaikki komponentit on samalla piirilevyllä. Mallista riippuen Raspberryssä on 512-1024Mb muistia, HDMI-portti, USB-portti, verkkoliitäntä, 3,5mm audioliitäntä ja komposiittivideoliitäntä sekä muistikorttipaikka. Prosessorit vaihtelevat 1GHz yksityimisestä prosessorista aina 1,2GHZ neliytimiseen ARMv8 -prosessoriin. Joissakin malleissa on myös kameraliitäntä ja USB- ja HDMI-liitännät on voitu korvata miniversioilla. Piireistä löytyy myös langaton lähiverkkokortti ja bluetoothmoduuli. Kehittyneimmissä malleissa on näytönohjainpiirinä Broadcom Videocore 4, jota ajetaan joko 250, 300 tai 400 MHz kellotaajuudella.

Raspberry Pi käyttää Debianiin pohjautuvaa avoimen lähdekoodin Raspbian-käyttöjärjestelmää, jolla ajetaan muokattua LXDE-käyttöympäristöä. Kuitenkin myös muita käyttöjärjestelmiä on mahdollista asentaa Raspbianin tilalle tarpeen vaatiessa ja tähän toimenpiteeseen löytyy internetistä erilaisia ohjeita ja videoita.

Pi kestää 50 mA:n kuormaa jokaista 3.3 V:n pinniä kohden ja jopa 1A:n kuormaa suoraan 5 V:n USB-liitännästä. Joskin tähän pitää vielä laskea piirin muu kuorma mukaan. A-version Pi jaksaa tämän mukaan siis 500 mA:n piirin ulkopuolista kuormaa ja B-versio taas 300 mA:n kuormaa. Laskelmieni mukaan Raspberry Pi kestäisi vaadittavan kytkennän kuorman helposti mallista riippumatta ja laajennusvaraa jäisi mahdollisille lisäkomponenteille, kuten bluetooth-moduulille tai etäisyysensorille.

3.3 Arduino Uno

Päädyn käyttämään Arduino Uno sen yksinkertaisuuden, hinnan, shield-laajennusten sekä ohjelmointikielen vuoksi. Raspberry Pi olisi tarjonnut enemmän ominaisuuksia ja toiminnallisuutta, mutta suurin osa niistä olisi mennyt työssä hukkaan ja halusin pitää kustannukset mahdollisimman pieninä. Olin myös aikaisemmin ollut tekemisissä Arduinon kanssa, joten tiesin jo sen rajoituksista ja ominaisuuksista enemmän kuin Raspberryn.



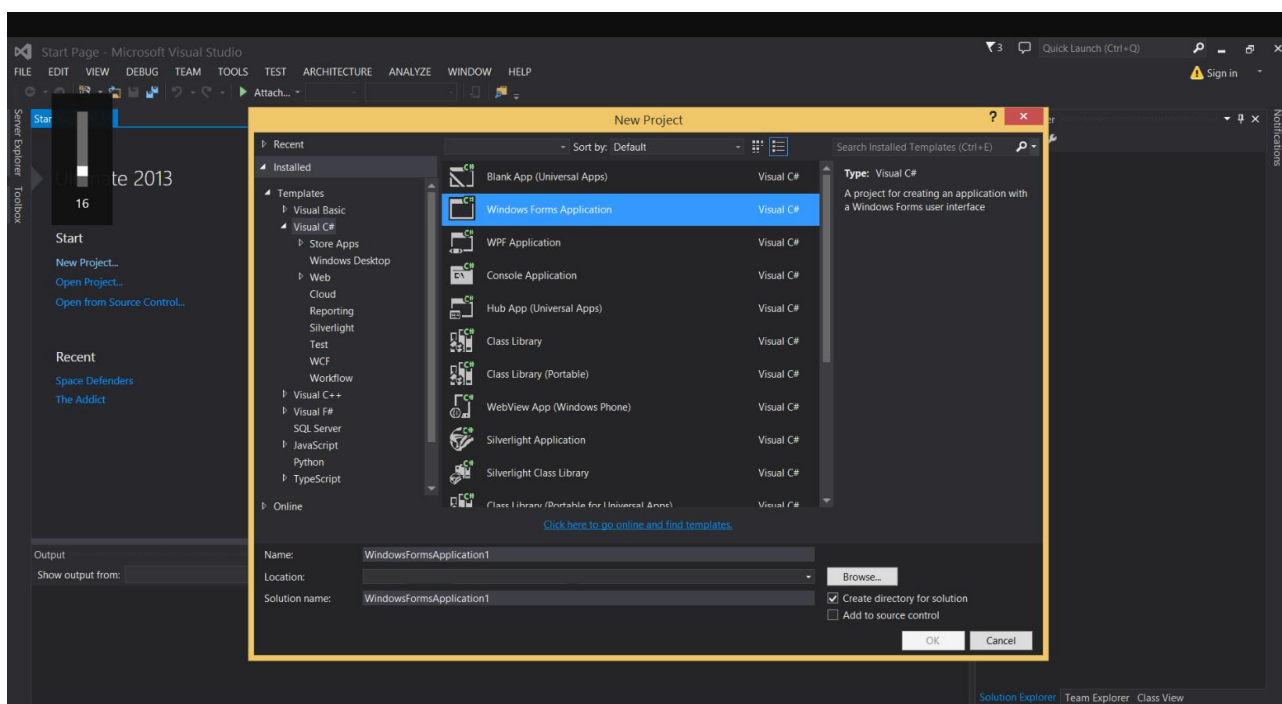
KUVA 1. Arduino Uno (mukaillen arduino.org 2016)

4 KEHITYSYMPÄRISTÖT

Arduino-piirejä voidaan ohjelmoida eri kehitysympäristöissä ja näistä kaksi yleisintä ja parhaiten tunnettua käydään seuraavassa luvussa läpi. Nämä tunnetuimmat ympäristöt ovat Microsoftin Visual Studio sekä Arduinon oma kehitysympäristö Arduino IDE.

4.1 Microsoft Visual Studio 2013

Microsoft Visual Studio on Microsoftin kehittämä yrityksille ja yksityisille suunnattu kehitysympäristö. Visual Studio tukee laajaa valikoimaa eri ohjelmointikieliä kuten C, C++, C# ja Visual Basic. Vuoden 2013 versiossa tuki Windows 8.1:lle sekä parannetut web-kehitysominaisuudet. Visual Studioa jo ennestään käyttäneenä 2013 olisi ollut varteenotettava vaihtoehto, mutta en sitä tullut käyttäneeksi, vaikka ehdinkin jo asentaa tarpeelliset liitännäiset, joiden avulla olisin kyennyt käyttämään Visual Studiota ohjelmointiin ja koodin testaamiseen. Myös mielenkiinto Arduinon omaa ohjelmistoa kohtaan oli niin suuri, että päädyin käyttämään sitä.

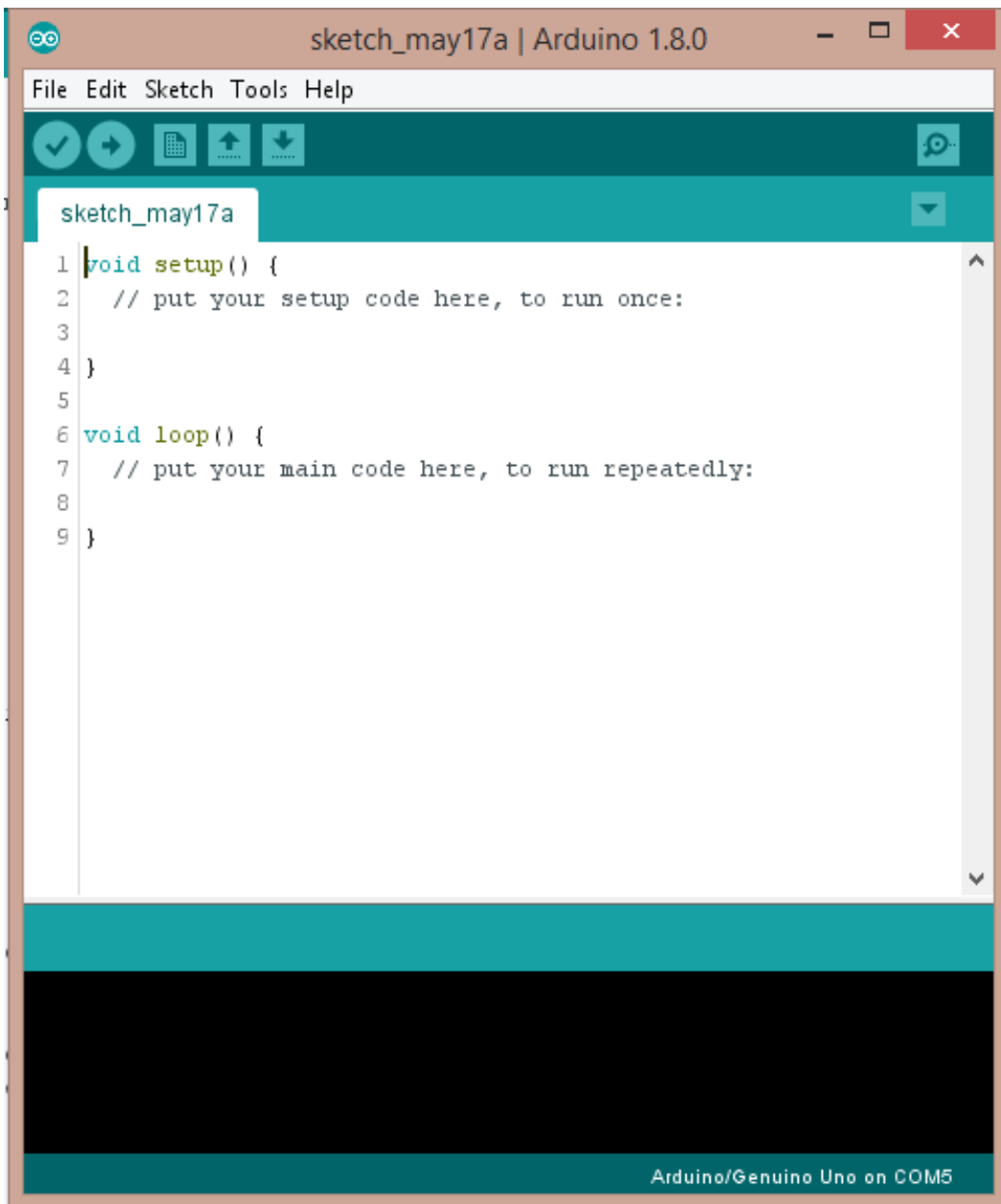


KUVA 2. Microsoft Visual Studio 2013

4.2 Arduino IDE

Arduino IDE on Arduinon oma kehitysympäristö Arduino-alustoille. Kuvassa 3 näkyy Arduino IDE:n aloitusnäky. Arduino IDE on kirjoitettu C/C++ ja Java -ohjelmointikieliä käyttäen ja on saatavilla Windows-, Mac OS X-, ja Linux-käyttöjärjestelmille. Arduino IDE-ohjelmistoa päivitetään jatkuvasti ja tulevaisuuden tukinäkyvät ovat hyvät.

Ympäristö tukee C- ja C++ -kieliä, joita myös kehitysalustat tukevat. Arduino-alustat itsessään voidaan ohjelmoida melkeinpä millä tahansa kielellä kääntäjän löytyessä, joka voi kääntää kielen binäärikonekieleksi prosessorille. Arduino IDEssä on itsessään parseri, jonka avulla kirjoitettu koodi voidaan tarkastaa ennen laitteeseen siirtämistä. IDE sisältää myös esikoodattuja esimerkkejä, joita voi käyttää joko pohjana tai mallina omassa työssään. Koodin siirtäminen laitteeseen toimii USB-kaapelin avulla tietokoneelta Arduinolle. Kaapelin tyyppi vaihtelee malleittain. (Banzi 2011, 20–21.) Arduino IDE tuli valituksi sen yksinkertaisuuden ja laiteläheisyyden vuoksi pienen perehtymisen jälkeen.



KUYA 3. Arduino IDE

5 LAITTEEN KUVAUS SEKÄ TOIMINTA

Seuraavaksi käydään läpi laitteen toimintaa, sen ominaisuuksia sekä yleistä tarvetta kyseessä olevalle laitteelle. Luvussa perehdytään myös itse koodin olennaisimpaan rakenteeseen, salasanajärjestelmään, sekä sen toteuttamistapaan.

5.1 Tarveperusta laitteelle

Idea ajastinlaitteelle tuli sattumalta oman vapaa-ajan harrastuksen puolelta Airsoftista. WCASin peleissä satuin kuulemaan keskustelua, kuinka olisi hienoa saada lisää pelimuotoja ja ne olisivat hieman realistisempia. Toisaalta myös tiesin että pelien ajastamisen kanssa on ollut hankaluuksia, sillä harvalla pelaajista on herätyskelloa tai muuta ajastimeksi soveltuvaa laitetta matkapuhelinta lukuun ottamatta mukana peleissä ja matkapuhelimetkin ovat henkilökohtaisia ja yleensä näppäinlukolla suojattuja, joten niiden kanssa ilmenee toiminnallisia ongelmia yksityisyyteen ja rikkoutumisiin liittyvistä asioista puhumattakaan.

Muutaman pelin johtajan kanssa asiaa tuumittuani huomasimme, että laitteelle olisikin käyttöä. Siitä alkoi suunnittelutyö, joskin ensin harrasteideana. Myöhemmin huomasin tehtävän sopivan oppinäytetyöaiheeksi, kun huomasin sen soveltuvan siihen sekä haastavuuden että luonteensa ansiosta. Aluksi ajattelin tehdä laitteesta pelkän yksinkertaisen ajastinlaitteen, mutta sitten laajensin idean kattamaan myös valikkorakenteen, jonka avulla voisi valita eri pelimuotoja pelattavaksi aikaa käyttäen apuna.

5.2 Laitteen toimintaperiaate

Laitteen toimintaperiaate on yksinkertainen: Laitteen käynnistyessä lcd-paneeliin tulee tekstiä, joka kertoo sen hetkisen valikkosijainnin ja näytöltä voi selkeästi päätellä, mitä seuraavaksi tulee tehdä. Jokaisen valinnan jälkeen valikko päivittyy ja ohjaa käyttäjää selkeästi eteenpäin.

Aluksi laite pyytää käyttäjää valitsemaan pelimuodon näppäimistön avulla. Tällä hetkellä pelimuotoja on kaksi: Ajastettu peruspeli, jossa laite on käytännössä vain herätyskello ja teemapeli, jossa laite toimii ajastimena, joka ajan kuluessa loppuun päästää äänimerkin ja ilmoittaa, kumpi joukkue voitti. Laitteen voi koodilla sulkea pois päältä ennen äänimerkkiä, tai koodin puuttuessa yrittää sammuttaa laitetta käynnistämällä toisen ajastimen, joka alkaa laskemaan aikaa pääajastimen rinnalla. Laite sammuu, jos

tämä lisäajastin saavuttaa nollan ennen pääajastinta, muulloin laite päästää taas äänimerkin, joka ilmoittaa toisen joukkueen voitokkaaksi.

Pelimuodon valinnan jälkeen tulee syöttää aika tunteina, minuutteina sekä sekunteina. Laite ottaa syötetyn ajan talteen ja päivittää sen välittömästi näytölle joka painalluksen jälkeen ja tallentaa syötetyn ajan muuttujaan, jota se päivittää ja muuttaa tietoa syötettäessä. Laite tarkistaa myös, että syötetty aika on oikeaa muotoa. Tästä esimerkkinä suurin sallittu peliaika, joka on 23 tuntia 59 minuuttia ja 59 sekuntia.

Seuraavaksi tulee salasanakysely, jos käyttäjä valitsi teemapelin. Muussa tapauksessa ajastin käynnistyy välittömästi. Salasanan syöttämisen jälkeen peli lähtee käyntiin ja laite piippaa joka sekunti, jotta pelaajat tietäisivät, missä suunnassa laite on ja vieläkö peliaikaa on jäljellä. Ajan loppumisen saa selville erillisistä pitkistä piippaussarjoista, jotka tulevat ajan saavuttaessa nollan.

Kun laite on käynnissä, ajastimen voi pysäyttää usealla eri tavalla. Yksi tapa näistä on syöttämällä siihen oikea koodi. Koodinsyöttö tapahtuu painamalla näppäimistön * -näppäintä, jolloin ajastin pysähtyy noin kolmen sekunnin ajaksi ja pelaajalla on mahdollisuus syöttää viisimerkkinen koodi näppäimistön avulla. Jos koodi on eri kuin ajastaessa syötetty koodi, laite ilmoittaa väärästä koodista ja salasanan syöttäjän joukkue häviää pelin. Jos koodi on oikea, aika pysähtyy ja salasanan syöttäneen pelaajan joukkue voittaa pelin. Jos käyttäjä ei ehdi syöttää koko koodia, ajanlasku jatkuu normaalisti kolmen sekunnin kuluttua. Tämä sen vuoksi, että salasanan syötöllä peliä ei voi jumittaa ja huijata aikaa.

Toinen tapa on painaa näppäimistön # -näppäintä, jolloin tausta-ajoksi saadaan aiemmin mainittu toinen ajastin, joka kisataa pääkelloa vastaan siitä, kumpi saavuttaa ensin nollan. Taustakello käyttää satunnaislukugenerointia käyttäen hyväksi 0-pinnin analogisignaalia, jolloin joka kutsu on eri arvolla koodissa määriteltyjen rajojen 30 ja 180 sekunnin väliltä. 0-pinni ei ole laitteessa käytössä eikä maadoitettuna, joten signaaliarvo sisältää kohinaa.

```

111 void displayCodePrompt()
112 {
113     clearRow(0);
114     lcd.setCursor(0,0);
115     lcd.write("CODE:");
116     givenPasswordDigits = 0;
117     while (givenPasswordDigits !=5 && timeToInput <300000)
118     {
119         timeToInput++;
120         char key3 = keypad.getKey();
121         if (key3 != NO_KEY && key3 != '*' && key3 != '#')
122         {
123             givenPassword[givenPasswordDigits] = key3;
124             givenPasswordDigits++;
125             lcd.print(givenPassword[givenPasswordDigits -1]);
126         }
127         if (givenPasswordDigits == 5)
128         {
129             checkPassword();
130         }
131         else
132         {
133             givenPasswordDigits = 0;
134         }
135     }
136     lcd.setCursor(0,0);
137     lcd.write("TIME REMAINING:");
138     timeToInput = 0;
139 }
140
141 boolean array_cmp(char *a, char *b)
142 {
143     int n;
144     for (n=0; n<5; n++)
145     {
146         if (a[n] != b[n])
147             return false;
148         else return true;
149     }
150 }
151 }
152
153
154 void checkPassword()
155 {
156     if (array_cmp(savedPassword, givenPassword) == true)
157     {
158         ctWin();
159     }
160
161     else if (array_cmp(savedPassword, givenPassword) == false)
162     {
163         terroristsWin();
164     }
165 }

```

KUVA 4. Salasanan syöttö ja tarkistus

Pelin päättyessä kuten aiemmin selvisi, laite palauttaa asetukset vakioarvoihinsa ja menee takaisin päävalikkoon, jossa se kysyy pelaajalta pelimuotoa ja näin uuden pelin voi aloittaa taas alusta.

Laitteen saa kirjoitushetkellä pois päältä irrottamalla USB-kaapeli joko Arduinosta tai sitten puhelimen varavirtapankin liittimestä. Tulevaisuuden kehitysmahdollisuuksia ovat esimerkiksi kytkimen kytkeminen USB-kaapelin liittimien välille tai vastaava kytkentämahdollisuus koteloinnin ulkopuolelle. Laite käynnistetään kytkemällä USB-kaapeli Arduinon ja akun välille, jonka jälkeen laite alustaa arvot ja jää päävalikkoon odottamaan näppäimistökomentoja.

5.3.1 Tarpeet täyttävän laitteen suunnittelu ja rakentaminen

Aloitin laitteen suunnittelun mielessäni perusseikat, jotka laitteelta vaadittiin ja ajatuksena oli rakentaa laite vaatimusten ympärille. Laitteessa tuli olla kaiutin, näyttö, näppäimistö sekä ilta- ja yöpeleihin LED-valo tai valot, jotta laitteen sekä pelin tilan näkee etäältä pimeässä. Laitteelle piti saada alunperin jonkinlainen kehikko, jonka ympärille kaikki rakennetaan. Myöhemmin päädyin alumiinisalkkuun enimmäkseen sen tuoman suojan kolhiintumista ja Suomen sääolosuhteita vastaan. Alumiinisalkkua joutui kuitenkin muokkaamaan, jotta Arduinon sai ruuvattua siihen kiinni. Myös varavirtapankille tuli oma velcronauhakiristyksellä varustettu paikka salkun seinämälle

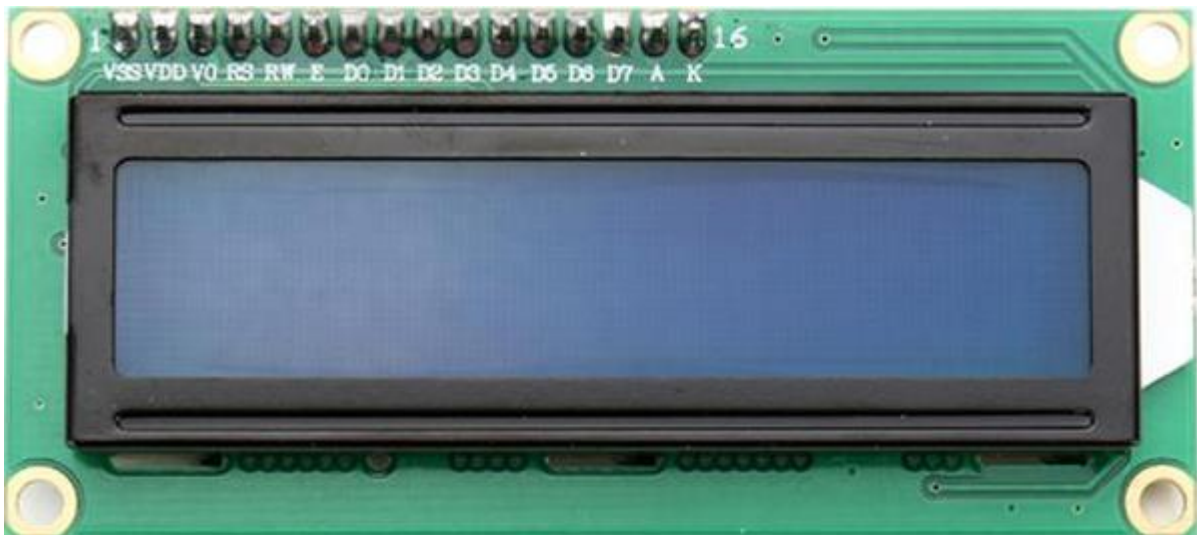
Laite pohjautuu Uno-malliseen Arduino-kehitysalustaan. Laitteeseen lukeutuvat itse prosessoripiiri, 2 kpl RGB-led -diodeja, joilla esitetään laitteen aktiivisuutta, piezosummeri, jolla luodaan ääntä, lcd-paneeli, jolla näytetään valikkovaihtoehdot sekä aikaa ja 4x3 matriisinäppäimistö ajan sekä parametrien syöttämistä varten. Kaikki nämä on yhdistetty Unoon ja toisiinsa ohuilla yksisäikeisillä johtimilla kytkentäalustan kautta ja on tarpeen vaatiessa muokattavissa avaamalla laitteen suojasalkun läpinäkyvä välipohja, jonka alla komponentit ovat suojassa säältä sekä häiriöiltä.

Kytkenät päätin toteuttaa koekytkentäalustalle, joka tuli Arduino Unon kehityspaketin mukana. LCD-paneelin sekä taustavalon voimakkuudensäätimen potentiometrin sijoitin pleksilevyyn, joka ruuvataan salkun kannen ja pohjan välille läpinäkyväksi kanneksi. Näppäimistö sijaitsee pleksin ulkopuolella ”kannen” päällä ja kytkennät on toteutettu pleksin alapuolella suojassa pölyltä, roiskeilta ja kytkennän manipuloinnilta pelitilanteessa.

5.3.2 Keskeiset komponentit ja niiden perustoiminta

Opinnäytetyössä keskeisessä osassa ovat 4x3 matriisinäppäimistö sekä 16x2 -lcd-paneeli. Komponentit on kytketty toisiinsa Arduino-piirin kautta ja niitä ohjataan koodin avulla ja matriisinäppäimistön painallukset näkyvät suoraan paneelissa muutoksena välittömästi, riippuen laitteen tilasta ja sen hetkisestä valikosta. Komponentit on asennettu kiinteästi välikanteen ruuveilla sekä matriisinäppäimistön tapauksessa liimapinnalla.

Lcd-paneelissa on 16 pinniä, jotka vaihtelevat valmistajan ja mallin mukaan. Omassa paneelissani pinnien järjestys on vasemmalta oikealle: GND, Vcc, VEE, RS, R/W, EN, joita seuraavat kahdeksan datapinniä DB0 - DB7 sekä lopulta Led+ ja Led-. Tärkeimmät ja pakolliset pinnit ovat käyttöjännitteen sekä maan lisäksi pinnit 4-14. Näiden avulla valitaan näytön toimintatila, syötetään näytölle data sekä päivitetään syötettävät merkit näytölle. Taustavaloa ei ole pakko käyttää ja kontrastia ei ole pakko säätää ja siihen voidaankin kolvata sopiva vastus kiinni ennalta, jolloin näyttö on aina yhtä kirkas. Halutessa voi käyttää potentiometriä, kuten itse päädyin tekemään. Potentiometrin ansiosta kirkkautta voidaan säätää pelin aikana sopivaksi ympäristön valotasoon nähden. Lcd-paneeli ei toimi täydellisesti ilman erillistä kirjastoa ja onkin suotavaa asentaa se ennen paneelin käyttöönottoa. Työn sisältäessä näin paljon suunnittelua, päädyin käyttämään valmista kirjastoa sen sijaan, että kirjoittaisin oman kirjaston näytölle.

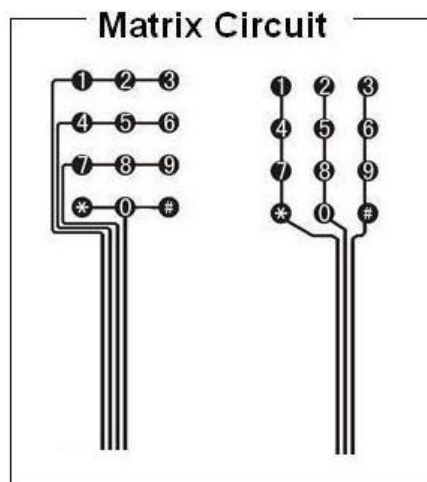


KUVA 5. LCD-paneeli (mukaiillen Communica.co 2017)

Matriisinäppäimistöissä on 7 pinniä. Vasemmalta ensimmäiset neljä pinniä määrittelevät rivit, kun taas viimeiset 3 pinniä määrittelevät sarakkeet. Rivien ja sarakkeiden risteyskohdissa on aina näppäin, jota painamalla kalvon kaksi pintaa koskettavat toisiaan ja syntyy oikosulku, jonka Arduino tulkitsee signaalina. Risteyskohtia on yhteensä 4x3 näppäimistöissä siis 12 kappaletta. Nämä risteyskohdat vaihtelevat näppäimistömallien ja tyyppien välillä ja joskus ne joudutaan tarkistamaan katsomalla näppäimistön takaosasta risteyskohdat ja kirjaamalla ne muistiin koodia varten. Näppäimistö ei toimi ilman erillistä kirjastoa.



KUVA 6. Matriisinäppäimistö (mukaillen Start2Arduino 2016)



KUVA 7. Näppäimistön matriisikaavio (mukaillen Start2Arduino 2016)

5.3.3 Ohjelmoinnin keskeiset seikat

Arduinon ohjelmoinnissa on muutama tärkeä seikka, joka liittyy ohjelmoinnin rakenteeseen. Suurin osa ohjelmista käyttää kolmeosaista rakennetta, joka koostuu yksinkertaisimmillaan setup-funktiosta sekä loop-funktiosta ja näitä edeltävästä osiosta, jossa esimerkiksi esitellään muuttujat. Setup-funktio on muuttujien alustamista sekä aloittamista varten. Yksinkertaisimmat ohjelmat eivät tarvitse muuta kuin setup-funktion. Kun ohjelma käynnistyy, Arduino ajaa setup-funktion ensimmäisenä ja tekee tämän vain kerran heti käynnistytyn jälkeen. Heti tämän jälkeen ohjelma siirtyy loop-funktioon, jota se ajaa kunnes ohjelma pysäytetään tavalla tai toisella, tai funktion ehto ei ole enää tosi. Käytännössä se kuitenkin on aina tosi, sillä loop-funktiota voidaan ajatella silmukkana, jonka ehto on koko ajan tosi.

```

297 void setup() {
298   // put your setup code here, to run once:
299   randomSeed(analogRead(0));
300   lcd.begin(16, 2);
301   pinMode(tubeLedsGreen, OUTPUT);
302   pinMode(tubeLedsRed, OUTPUT);
303   pinMode(buzzer, OUTPUT);
304   Serial.begin(9600);
305       setLedColour(0, 200); // 0 red, 255 green for unarmed
306
307 }
```

KUVA 8. Setup-funktio

Ohjelmointiprosessi alkoi suunnittelulla, jolla luotiin rakenne koodille. Päädyin tekemään osittain porrasmaisen rakenteen ohjelmalle, jossa koodi pysyy ”tasolla” tiettyjen ehtojen ollessa tosia ja siirtyy seuraavalle tasolle kun ehdot muuttuvat vaikkapa näppäimistön käytön seurauksena. Päätin toteuttaa myös ajanlaskennan käyttämällä Arduinon upitelaskuria apuna, jonka ansiosta kello pysyy tarkasti ajassa. Laskurin avulla ohjelma määrittelee lcd-paneelin ruudunpäivitysnopeuden, sekuntirekisterin päivittämisen, tärkeiden merkkiäänien ajankohdat sekä aikajärjestelmän toiminnan kokonaisuudessaan.

```

603 if (gameRunning == true && defusable == true && (currentMillis - previousMillis > timeInterval))
604 {
605     previousMillis = currentMillis;
606
607     gameSeconds--;
608     if (beingDefused == true)
609     {
610         defuseDuration--;
611     }
612
613     if (gameSeconds < 0)
614     {
615         gameMinutes--;
616         gameSeconds = 59;
617     }
618     if (gameMinutes < 0)
619     {
620         gameHours--;
621         gameMinutes = 59;
622     }
623     if (gameHours < 0)
624     {
625         gameHours = 0;
626     }
627     if (gameHours < 0)
628     {
629         gameHours = 0;
630     }
631     if (gameMinutes < 0)
632     {
633         gameMinutes = 0;
634     }
635     if (gameSeconds < 0)
636     {
637         gameSeconds = 0;
638     }
639     updateTime();
640
641     if (selectedGameMode == 2)
642     {
643         playBeep();
644     }
645 }

```

KUVA 9. Ajanlaskentarakenne

5.4 Laitteen testaaminen

Laitetta testattiin sekä rakentamisen, että Airsoft-pelin aikana. Rakennusvaiheessa toiminnallisuuden varmistaminen keskittyi yksittäisten komponenttien toiminnallisuuteen yksin sekä muiden komponenttien kanssa. Käytännössä samalla testattiin Arduinon toimintaa kyseisten komponenttien ollessa kytkettyinä ja varmistettiin, ettei virtarajat ylity kokoonpanossa. Tämä varmistettiin sekä laskennallisin keinoin, että mittalaitteiston avulla. Näin saatiin kaksinkertainen varmistus.

Laite otettiin mukaan erääseen WCAS:n järjestämään torstai-illan pelitapahtumaan. Tälle päivälle oli ennustettu vesisateita, joten samalla tarjoutui oiva tilaisuus varmistua koteloinnin säänkestosta. Laite oli tuona päivänä käytössä yhtäkestoisesti noin 3 tuntia, jonka aikana se altistui tuulelle, sateelle ja tärinälle ja selvisi näistä odotettua paremmin. Samalla testattiin myös äänien ja valojen toimintaa pelitilanteessa ja nämäkin ominaisuudet todettiin käyttökelpoisiksi ja riittäviksi pelitilanteisiin nähden. Ainoa asia, johon en ollut aivan täysin tyytyväinen, oli kaiuttimen äänenvoimakkuus, jolle ei juurikaan voi tehdä mitään, ellei kaiutinta vaihda suurempaan, jolloin Arduinon virtataso saattaa ylittyä ja tämä taas vaarantaa laitteen toiminnallisuuden. Tästä huolimatta kaiutin toimii kohtalaisen hyvin, vaikka ääni onkin tarkoitettua hiljaisempi.

5.5 Jatkokehitysmahdollisuudet

Jatkokehitystä ajatellen laitteeseen voisi ajatella rakennettavaksi bluetoothmoduulia, jolloin kauko-ohjaus olisi mahdollista aina tiettyyn etäisyyteen saakka. Tämä ei sinänsä olisi kovinkaan hankala toteuttaa ja voi olla, että myöhemmin sille tulee myös käyttöä. Arduinin modulaarisen Shield-lisäosien luonteen ansiosta siihen voisi asentaa myös etäisyysensorin lisätoiminnoille ja laajentaa ohjelmallisesti pelimuototukea kattamaan suuremman kirjon pelimoodeja. Ainoana rajoittavana tekijänä näenkin 200 mA:n virransyöttörajoituksen, mutta senkin voisi varmasti ohittaa ulkoisella virransyötöllä muille lisäkomponenteille erillisten ohjausyksiköiden avulla.

Koska vastaavanlaisia pelilaitteita ei ole juurikaan saatavilla kaupoista tai yksityisiltä rakentajilta opinnäytetyön kirjoitushetkellä muutamaa yksityishenkilöiden rakentamia poikkeuksia lukuun ottamatta, jotka nekin eroavat selkeästi työn laitteesta toiminnaltaan, voisi laitteen kaupalliset mahdollisuudet olla hyvä tutkia. Jos idean saisi myytyä jollekin Suomessa toimivista Airsoft-liikkeistä, voisi laitteita alkaa tuottamaan enemmänkin näin edistäen harrastustoimintaa.

6 JOHTOPÄÄTÖKSET JA POHDINTA

Tässä luvussa tarkastelen sitä, kuinka hyvin laite lunasti odotukset ja täytti sille asetetut vaatimukset. Luvussa pohditaan myös kokonaiskuvan onnistuneisuutta ja käsitellään opinnäytetyön hyötyjä kirjoittaneelle sekä peliyhteisölle.

Laitteen suunnittelu ja rakentaminen oli mielenkiintoinen ja opettavainen kokemus. Lähtökohtaisesti ajattelin projektin sisältävän vähemmän rakentamista ja teknistä työskentelyä kuin lopulta päädyin tekemään. Myös suunnitteluun meni enemmän aikaa kuin aluksi odotin. Jouduin myös käymään koodia ja komponenttien ominaisuuksia läpi uudelleen työn edetessä ja ongelmien noustessa esille, mutta selvisin niistä melko hyvin, enkä joutunut muuttamaan alkuperäistä suunnitelmaa juuri ollenkaan. Ainoat muutokset koskivat lähinnä valaistusjärjestelmää, johtuen Arduinon virransyötön ominaisuuksista ja rajoituksista, joista en ollut perillä ennen työhön perehtymistä.

Laitteen ominaisuudet ovat juuri ne, jotka alun perin oli tarkoituksena saada toteutetuksi, eikä työssä joutunut tekemään juurikaan kompromisseja valojärjestelmää ja kaiutinta lukuun ottamatta, joten määrittelin työn erittäin onnistuneeksi. Kirjoitushetkellä WCAS-järjestön työstä tietoiset henkilöt ovat erittäin tyytyväisiä laitteeseen ja laite otetaan pelikäyttöön lähiaikoina. Vaikka en saanut työstä rahallista korvausta, väittäisin, että työn tuoma kokemus ja oppiminen ovat lopulta rahaakin arvokkaampia.

LÄHTEET

Start2Arduino 2016. Matriisinäppäimistö, Matriisikaavio.

Saatavissa: <http://start2arduino.weebly.com/relay-control-panel.html>.

Viitattu 17.6.2017

Vyyryläinen, S. 2015. Lähtönopeudet.

Saatavissa: <https://airedi.fi/jutut/artikkelit/yleisesti/98-laehtenopeudet-atom-airsoft-teknikkanurkka>.

Viitattu-27.8.2017

Softaaja 2015. Airsoft.

Saatavissa: <http://www.softaaja.fi/airsoft/>.

Luettu: 2.8.2017

Monaf, P. 2011. What is airsoft.

Saatavissa: <http://www.ai-mag.com/what-is-airsoft/?v=79cba1185463>

Viitattu 29.8.2017

Huuki, M. 2016. Suomalainen airsoft, mihin olet menossa?

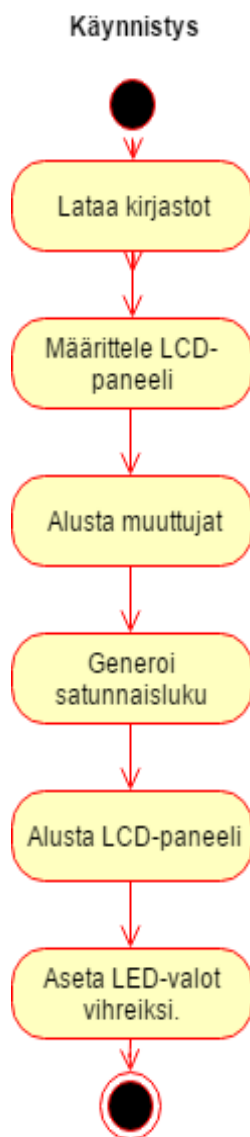
Saatavissa: <https://www.airedi.fi/uutiset/141-suomalainen-airsoft-mihin-olet-menossa>

Viitattu 28.7. 2017

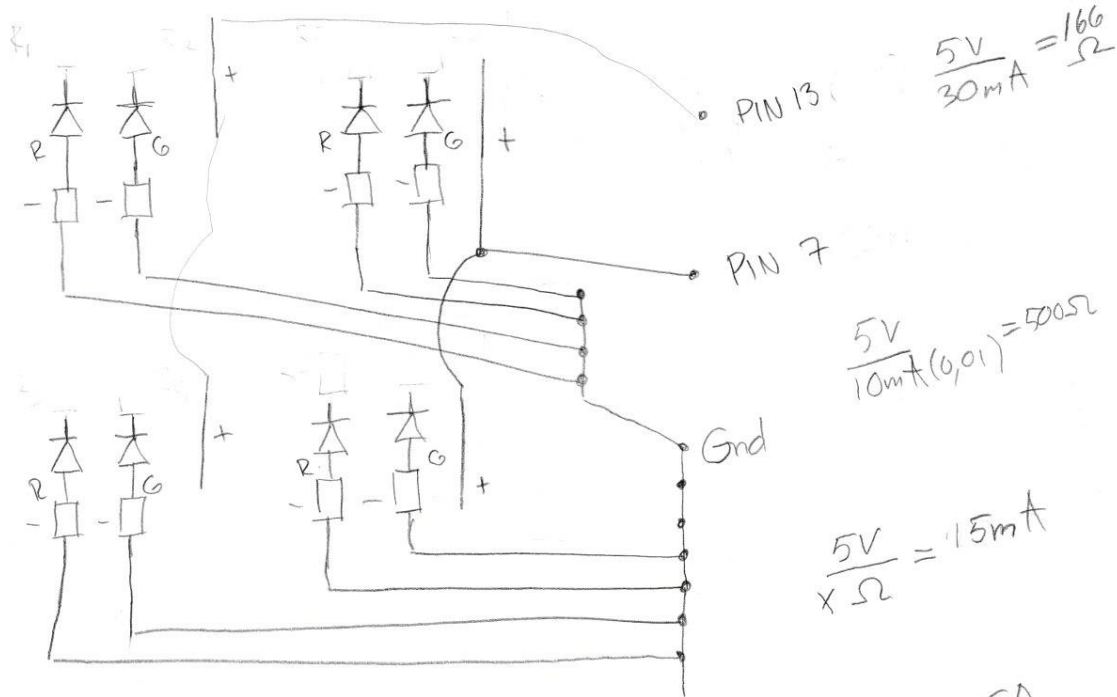
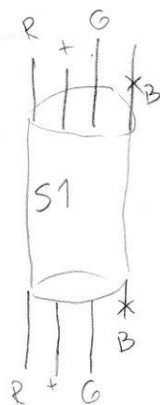
Lindfors, J. 2012. Airsoftissa on siisteintä, ettei kukaan kuole. Saatavissa:

<http://yle.fi/aihe/artikkeli/2012/06/01/airsoftissa-siisteinta-ettei-kukaan-kuole#media=80470>

Viitattu: 17.6.201

Perustilan ja ohjelman aloitusarvojen määrittäminen

Virta-arvojen laskeminen sekä LED-järjestelmän hahmotuskytkentä



$$\frac{5V}{30mA} = 166 \Omega$$

$$\frac{5V}{10mA(0,01)} = 500 \Omega$$

$$\frac{5V}{X \Omega} = 15mA$$

$$15mA / led = 30mA / pinni$$

$$\frac{5V}{\frac{1}{X} + \frac{1}{X} \Omega} = 0,015A$$