

Joonas Helava

OHJELMOINTIPULPETTI OHJELMOINNIN ALKEIDEN
OPETUKSEEN

Sähkötekniikan koulutusohjelma
2017

OHJELMOINTIPULPETTI OHJELMOINNIN ALKEIDEN OPETUKSEEN

Helava, Joonas
Satakunnan ammattikorkeakoulu
Sähkötekniikan koulutusohjelma
Tammikuu 2017
Ohjaaja: Asmala, Hannu
Sivumäärä: 31
Liitteitä:

Asiasanat: Ohjelmointi, Opetus, Apuvälineet

Vuoden 2016 syksyllä otettiin käyttöön uusi opetussuunnitelma, joka sisältää uuden oppiaineen Tieto- ja viestintäteknologinen osaaminen. Tämä synnytti aivan uudenlaisen tarpeen, sillä ohjelmointi on ollut osana opetusta vasta yläasteella ja silloinkin valinnaisten aineiden kautta.

Opinnäytetyössä suunniteltiin ja toteutettiin laite, jolla konkretisoitaisiin ohjelmoinnin periaatteita ja joka tulisi toimimaan sen opettamisen apuvälineenä. Työssä tutkittiin myös mitä ohjelmoinnista kannattaa opettaa ja millä tavoin ne opetettaisiin laitetta apuna käyttäen.

Opinnäytetyössä käytiin läpi laitteen kehityksen vaiheet ideasta toteutukseen ja sen käyttämistä opetuksen apuvälineenä

PROGRAMMING DESK FOR TEACHING FUNDAMENTALS OF PROGRAMMING

Helava, Joonas

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Electrical Engineering

January 2017

Supervisor: Asmala, Hannu

Number of pages:31

Appendices:

Keywords: Programming, Teaching, Assistive device

In autumn 2016 a new curriculum was introduced. It contained a new subject named information technology. This created a whole new demand as programming had only been part of education in a form of elective studies.

This thesis exhibits a device that can be used to concretize and teach programming skills. The paper also examines what should be taught of programming and how the device designed and built here could be utilized in this education.

In this thesis phases from the idea to execution and use in tuition are described.

SISÄLLYS

LYHENNELUETTELO	6
1 JOHDANTO.....	7
2 MÄÄRITELMÄ.....	8
2.1 Idea.....	8
2.2 Ohjelmoinnin opetus.....	9
2.2.1 Ohjelmointitaidot.....	9
2.2.2 Ohjelmoinnin rakenteet (edistynyt ohjelmoija).....	10
2.3 Suunnitelma	11
2.3.1 Laite	11
3 LAITTEET, MATERIAALIT JA TEKNIIKAT.....	13
3.1 Arduino Mega (alusta)	13
3.2 I ² C	13
3.3 EEPROM (muisti).....	14
3.4 3D-tulostus.....	15
3.5 3D-tulostin	16
3.6 3D tulostusmuovit.....	17
3.6.1 PLA	17
3.6.2 ABS	17
3.7 123D Design	18
4 TOTEUTUS	18
4.1 Kokonaisuus.....	18
4.1.1 Palikat	19
4.1.2 Alustat	21
4.1.3 Painikkeet ja kytkin	22
4.1.4 LCD	22
4.1.5 Toiminta	22
4.2 Elektroniikka ja ohjelmointi	23
4.2.1 Arduino	23
4.2.2 EEPROM	24
4.2.3 Bluetooth	25
4.3 Ohjelma.....	26
4.3.1 Palikan tunnistus.....	26
4.3.2 EEPROM kirjoitus ja luku.....	27
4.3.3 Bluetooth lähetysprotokolla.....	27
5 TESTAUS	28
5.1 Laitteen toiminta	28

5.2	Opetus	28
5.2.1	Ohjelmointitaito: ohjelma.....	28
5.2.2	Ohjelmointitaito: ohjelmallinen ajattelu.....	29
5.2.3	Ohjelmointitaito: algoritmi.....	29
6	LOPPUPÄÄTELMÄ.....	29
6.1	Yhteenveto	29
6.2	Jatkokehitys.....	30
	LÄHTEET.....	31

LYHENNELUETTELO

ABS – Acrylonitrile Butadiene Styrene

ADC – Analog to Digital Converter

DAC – Digital to Analog Converter

EPROM – Erasable Programmable Read-Only Memory

EEPROM – Electronically Erasable Programmable Read-Only Memory

GND – Ground

I/O – Input/Output

I²C – Inter-Integrated Circuit

ICSP – In Circuit Serial Programming

IDE – Integrated Development Environment

LSB – Least Significant Bit

MCU – MicroController Unit

MSB – Most Significant Bit

PLA – PolyLactic Acid

SCL – Serial Clock Line

SDA – Serial Data Line

STL - STereoLithography

UART – Universal Asynchronous Receiver Transmitter

VDC – Volts of Direct Current

1 JOHDANTO

Tämän opinnäytetyön aiheena on suunnitella, kehittää ja toteuttaa ohjelmointi- ja ohjauspulpetti, jolla opetetaan peruskoululaiselle ohjelmoinnin alkeita. Kyseessä on siis opetusväline, joka toimisi opetuksen tukena. Eli laite olisi apuväline, jonka avulla voitaisiin ohjelmoinnin periaatteita konkretisoida. Opinnäytetyössä tutkitaan myös, miten ohjelmointia kannattaa opettaa henkilölle, jolla ei ole aiempaa kokemusta asiasta.

Maailma on kovaa vauhtia kehittymässä yhä digitaalisemmaksi. Monet toiminnot ja palvelut ovat jo nyt digitalisoitu ja yhä enemmän asioita tullaan tekemään tietokoneilla. Tästä syystä on hyvä tutustuttaa ohjelmointi aikaisemmin nuoremmille. Sen lisäksi, että se laajentaa käsitystä teknologiasta, niin se tekee myös parempia tekijöitä tulevaisuudessa. Myös Opetushallitus on ottanut tämän huomioon, sillä uusi opetusohjelma, joka tulee käyttöön syksyllä 2016, tuo ohjelmoinnin perusopetukseen.

Aikaisempi kokemus monen tyyppisestä ohjelmoinnista auttaa minua suunnittelemaan ja kehittämään ideaa. Koska olen itse käynyt jo monesti läpi prosessin siitä, kuinka ohjelma luodaan monella eri ohjelmointi kielellä, on minulla hyvä käsitys siitä kuinka ohjelmointia kannattaa lähestyä.

Opintoni tukevat opinnäytetyötäni lähes kokonaan. Olen suorittanut automaatiomodulin, joka sisälsi paljon ohjelmointia. Sen lisäksi olen suorittanut kurssin ohjelmoitavasta digitaalitekniikasta, joka keskittyy ohjelmoitavan digitaalitekniikan suunnittelun perusteisiin.

Tämä opinnäytetyö sisältää ohjelmoinnin alkeiden opetteluun käytettävän laitteen kehitysvaiheet ideasta testaukseen. Kehitysvaiheissa käydään läpi, kuinka ideasta tuli toteutus kelpoinen, laitteen määritelmä, perusteellinen suunnittelu ja toteutus.

2 MÄÄRITELMÄ

Määritelmä osuuden tarkoituksena on määrittellä mitä laitteen olisi tarkoitus opettaa ja millä tavalla. Tässä vaiheessa kirjataan ylös, mitä ohjelmointi sisältää ajatustasolla ja miten ohjelmoija asiat näkee. Nämä asiat toimivat suunnittelun ytimenä ja laite tulee kehittymään näiden ympärille.

2.1 Idea

Ohjelmointipulpetin tarkoitus on toimia apuvälineenä ja opettavaisena leikkivälineenä. Laitteen on tarkoitus tulla tutuksi leikin avulla ja ohjelmoinnin perusidean opetus tapahtuu erilaisten tehtävien kautta. Tehtävät monimutkaistuvat samaan tapaan kuin missä tahansa muussa oppiaineessa.

Idea laitteesta syntyi, kun tulin miettineeksi, että ohjelmointi olisi helpompaa, jos ohjelman palasia voisi laittaa eteensä ja koota samaan tapaan kuin palapeliä. Palapelin kokoamista voi ajatella samalla tavalla kuin ohjelmoimista. Palapelin kokoaja aloittaa kokoamisen sillä, että hän lajittelee palat ja jakaa kokoamisen pienempiin helpommin koottaviin osiin. Tämän tyyppinen ajattelu on hyvin tärkeä osa ohjelmointia.

Laitteen ideana on siis auttaa visualisoimaan ohjelmointia siten, että ohjelman osat voidaan asetella ohjelmoijan eteen ja hän voi niitä siinä siirrellä. Laite tulisi sisältämään alustoja, joihin voitaisiin asetella ohjelman palasia eli tässä tapauksessa erilaisia robotin ohjaustoimintoja. Ohjaustoimintojen lisäksi alustoille voidaan asettaa palasia jotka sisältävät useita toimintoja ja jotka ovat ohjelmoijan itse tekemiä eli muistipaloja.

Laite sisältää useita alustoja joihin voidaan asettaa palikoita jotka esittävät robotin ohjaustoimintoja. Laite sisältää myös yhden erillisen alustan joka on tarkoitettu muistipalikan ohjelmointia varten.

Alussa opetellaan laitteen toiminnot liikuttelemalla robottia ilman muistipalikkaa. Testailtaan, miten robotti liikkuu ja kuinka paljon, miten se kääntyy ja muita tulevia toimintoja. Kun laite ja robotti on tullut tutuksi erilaisten leikkien kautta, on aika siirtyä tekemään tehtäviä sen kanssa.

Ensimmäiset tehtävät sisältävät 3 toiminnon peräkkäin laittamista (sarjarakenne) ja robotin ohjaamista suoraan. Tällä havainnollistetaan useamman toiminnon peräkkäistä toimintaa. Kun perusidea usean toiminnon toiminnasta on selvä, on aika tuoda muistipalikka tehtäviin mukaan. (Muistipalikka vastaa ns. oman funktion luomista.)

Muistipalikkaa käytetään aluksi yksinkertaisissa tapauksissa esim. laitetaan muistiin monta eteenpäin käskyä. Tällä opetetaan, että asioita joita toistetaan, voidaan laittaa muistiin ja käyttää muiden käskyjen kanssa samaan aikaan. (toistorakenne) Kun tämä on selvää, on aika opetella tehtävien jakamista pienempiin osiin. Tätä harjoitellaan yksinkertaisilla harjoituksilla. Harjoitusten jälkeen siirrytään harjoittelemaan opittuja asioita laitteen avulla.

Kaikkien näiden harjoitusten jälkeen ohjelmoinnin perusideat ovat opetettuna. Voidaan siirtyä laitteen monimutkaisempiin toimintoihin kuten analogisiin syöttöihin ja erilaisten ohjauksien päälle ja pois laittoon. Jos robotti sisältää esim. antureita niin laitteella voidaan opettaa yksinkertaista ehtolauseiden käyttöä.

2.2 Ohjelmoinnin opetus

Ohjelmointi on ohjelman kirjoittamista, joka taas on tietokoneelle tai vastaavalle laitteelle annettavia ohjeita. Ohjelmointia tarvitaan, jos halutaan toistoa tai halutaan ”keskustella” robotin kanssa sen omalla kielellä.

Miksi ihmeessä sisällyttää näinkin monimutkainen asia kuin ohjelmointi opetusohjelmaan? Siihen on montakin hyvää syytä. Sen lisäksi että ohjelmointi opettaa lapselle, miten esim. tietokone toimii ”juuritasolla”, niin se opettaa myös ajattelutapoja, joita voi hyödyntää muualla.

2.2.1 Ohjelmointitaidot

Asioita joita ohjelmoinnista tulisi ymmärtää:

Ohjelma – Ohjeistus jonka kone ymmärtää ja seuraa.

Ohjelmallinen ajattelu – Ajattelutapa jolla ohjelmoija valmistee ongelman digitaaliselle ratkaisulle

Algoritmi – Lista askelista jotka antavat sinun suorittaa tehtävän loppuun

Kaava – Teema joka toistuu monta kertaa

Paloittelu – Vaikean ongelman jakaminen pieniin, helppoihin ”ongelmiin”

Ongelman purkaminen osiin. Kun ohjelmoija kohtaa ongelman, jakaa hän sen pieniin osiin, jotka on helpompi ratkaista. Suoraviivaiset tehtävät jakautuvat usein komentosarjoiksi.

Algoritmin luominen. Algoritmi on kuvaus jonkin tehtävän suorittamiseksi tarvittavista toimenpiteistä. Ohjelmoija laatii algoritmin usein jakamalla annetun tehtävän pieniin osiin, jotka voidaan toteuttaa käytetyllä ohjelmointikielellä.

Kaavojen tunnistaminen. Algoritmia luodessaan ohjelmoija usein tunnistaa toistuvia kaavoja. Jos sama asia toistuu ohjelmassa monta kertaa peräkkäin, ohjelmoija hyödyntää toistorakennetta. Jos sama asia toistuu ohjelman eri paikoissa, kirjoitetaan funktio, joka tekee tämän asian.

Looginen päättely. Ohjelmoija joutuu jatkuvasti päättelemään, mitä tapahtuu eri tilanteissa, ja mistä seuraa mitä. Koska tietokone ei itse osaa päätellä mitään, on ohjelmoinnin laadittava päättelysäännöt, joita seuraamalla myös tietokone voi toimia näennäisen älykkäästi.

[1](Oppi & ilo 2016, 94-96)

2.2.2 Ohjelmoinnin rakenteet (edistynyt ohjelmoija)

Komentosarjat. Tietokone ymmärtää vain hyvin tarkkoja, yksityiskohtaisia ohjeita. Komentosarja on sarja peräkkäin suoritettavia komentoja.

Toistorakenteet. Toistorakenteeksi kutsutaan sitä, että tietokone toistaa jotakin asiaa n kertaa tai kunnes toinen asia tapahtuu.

Ehtolauseet. Ehtolauseiksi kutsutaan rakenteita, joissa tietokoneita opetetaan tekemään päätöksiä.

Funktiot. Funktiolla opetetaan kone tekemään jokin asia, jolle annetaan nimi. $f = \text{eteen}$, käänny oikealle. Jonka jälkeen robotti osaa suorittaa käskyt eteen, käänny oikealle aina kun se saa käskyn f .

[2](Oppi & ilo 2016, 97)

2.3 Suunnitelma

Koska helpoin ja halvin tapa tehdä projekteja on käyttää mikro-ohjainta, päätin että käytän projektissa suosittua Arduino alustaa, joka perustuu ATmega sarjan mikro-siruihin. Arduinolle voi kommunikoida muutamalla eri tavalla:

UART (Universal Asynchronous Receiver Transmitter) asynkroninen sarjaliikenne.

ICSP (In Circuit Serial Programming) piirin sisäinen sarjaliikenne.

Digitaaliset tulot.

Koska tarkoituksena ei ole käyttää toista laitetta kommunikointiin, niin ulkoisena kommunikointivälineenä käytetään digitaalisia tuloja. Digitaalinen tulo tunnistaa onko tulo HIGH vai LOW eli on/off tiloja, toisin sanottuna bittitietoa.

2.3.1 Laite

Ensimmäisenä laitteen suunnittelussa listataan toiminnot, jotka halutaan laitteen toteuttavan.

- Liikkuminen eteenpäin ja kääntyminen
- Muistiin kirjoittaminen
- Lähtöjen ohjaus ja anturien luku
- Analogisen tiedon käsittely

Koska työssä kehitetään prototyyppiä, toteutetaan näistä toiminnoista vain yksinkertaisimmat eli liikkuminen ja muistiin kirjoittaminen.

Seuraavana vaiheena suunnittelussa mietitään, kuinka toiminnot esitetään Arduinolle. Kun Arduinolle syötetään tietoa digitaalisen I/O:n kautta, se tehdään yleensä painikkeilla. Painikkeen painaminen vetää syötön joko GND tai +5VDC ja tämä ilmaistaan biteissä 0 tai 1. Painikkeen painaminen vaatii paljon fyysistä voimaa ja tässä toteutuksessa halutaan tietoa ilmaista kevyillä palikoilla, korvataan painike jollain muulla.

Koska painike yhdistää kaksi pinniä, voidaan se korvata jollain mikä tekee saman. Ajatuksena tuli mieleeni, että tässä tapauksessa palikka voisi toimia painikkeena. Ensimmäiseksi mietin, että palikka voisi painaa painikkeita alustassa, mutta palikka ei tule luultavasti painamaan tarpeeksi. Herkin mikrokytkin jonka löysin vaatii 0,3N voiman joka vastaa 30g.

Toisena ajatuksena tuli mieleeni, että palikka voisi toimia linkkinä kahden johtimen välillä. Eli palikassa olisi jalat, jotka koskisivat alustan koskettimiin. Koskettimia olisi niin monta kuin tarvittaisiin ja keskellä olisi jalka joka on aina GND tai +5VDC. Palikan sisällä yhdistettäisiin kulmajalan kosketin keskijalan koskettimeen, joka on joko GND tai +5VDC. Näin simuloidaan painiketta ilman että tarvitaan paljoa voimaa.

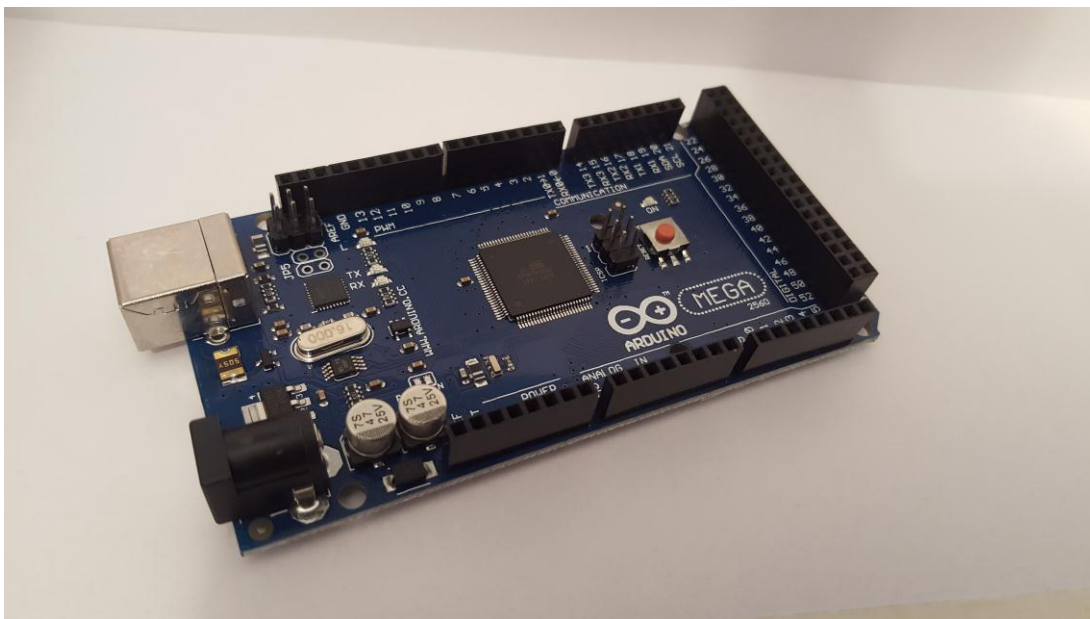
Muistipalikan ytimenä toimii EEPROM, koska se on helpoin ja käytännöllisin kaikista muisteista. EEPROMia käytetään I²C-väylän kautta ja se vaatii myös syöttövirran.

Seuraavana vaiheena suunnittelussa mietitään itse laitetta. Laitteessa olisi 4 alustaa, johon palikoita voisi laittaa. 3 alustoista olisi ohjauspalikoita varten ja 4. olisi muistipalikkaa varten.

3 LAITTEET, MATERIAALIT JA TEKNIIKAT

3.1 Arduino Mega (alusta)

Arduino Mega 2560 on ATmega2560 mikrosiruun perustuva mikro-ohjain. Se sisältää 54 digitaalista tuloa/lähtöä. Digitaalisten tulojen määrä tuo projektiin paljon mahdollisuuksia ja tekee laitteesta helposti laajennettavan. Arduino Mega sisältää myös 16 analogista tuloa/lähtöä joita voidaan käyttää erilaisissa analogisissa ohjauksissa. [3](Arduinon www-sivut 2016)



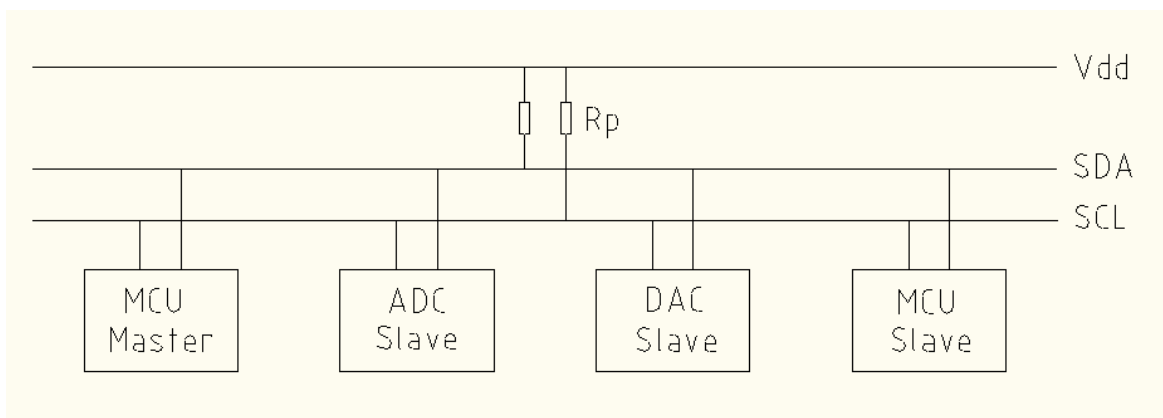
Kuva 1. Arduino Mega 2560

3.2 I²C

I²C (Inter-Integrated Circuit) eli integroitujen piirien välinen yhteys. I²C on sarjalii-
kenneväylä, jossa voi olla useita isäntiä (master) ja renkiä (slave). Tämä tarkoittaa sitä
että väylässä voi olla useita laitteita jotka voivat alustaa väylän ja tuottaa renkien tar-
vitsemaa kellosignaalia. I²C:tä käytetään yleensä yhdistämään hitaampia oheismikro-
piirejä prosessoreihin ja mikro-ohjaimiin. I²C on tarkoitettu lyhyen matkan tiedonsiir-
toon, yleensä piirien sisäiseen kommunikaatioon.

I²C käyttää ainoastaan kahta kaksoisuuntaista open-drain väylää, Serial Data Line (SDA) ja Serial Clock Line (SCL), joissa on ylösvetovastus (Rp). Tyypilliset käyttöjännitteet ovat +5V tai +3,3, mutta järjestelmää voidaan käyttää myös muilla jännitteillä.[4](Virallinen I²C määrittely 2014)

I²C väylään voidaan liittää mikro-ohjaimia, jotka voivat toimia kummassa tahansa roolissa. Mikro-ohjaimen lisäksi väylään voidaan liittää monenlaisia muuntimia ja näyttölaitteita.

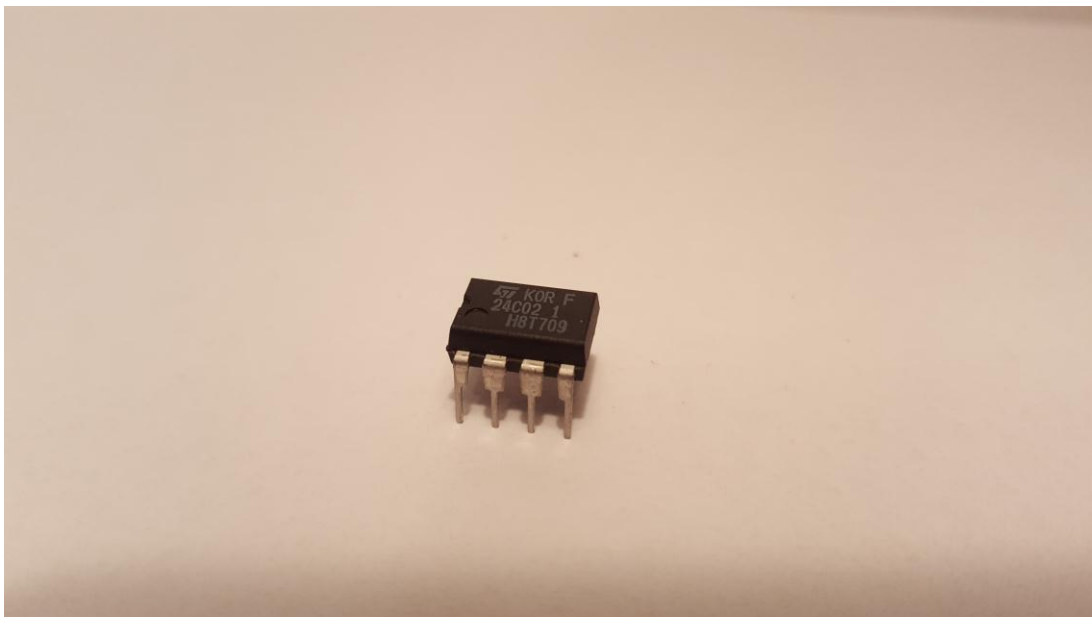


Kuva 2. Esimerkki I²C väylästä.

3.3 EEPROM (muisti)

EPROM (Erasable Programmable Read-Only Memory) haihtumatonta puolijohde-muistia, jolle voidaan kirjoittaa erillisellä laitteella normaalia käyttöjännitettä suurem-malla jännitteellä ja joka voidaan tyhjentää voimakkaan ultraviolettivalon avulla.

EEPROM (Electrically Erasable Programmable Read-Only Memory) niin kuin nimi-kin kertoo niin EEPROM:in voi tyhjentää elektronisesti ja voidaan uudelleen kirjoittaa 10000-100000 kertaa. EEPROM:ia ei tarvitse tyhjentää lohkoittain vaan uuden tiedon voi kirjoittaa vanhan päälle tavu kerrallaan.[5](Wikipedia artikkeli 2016)

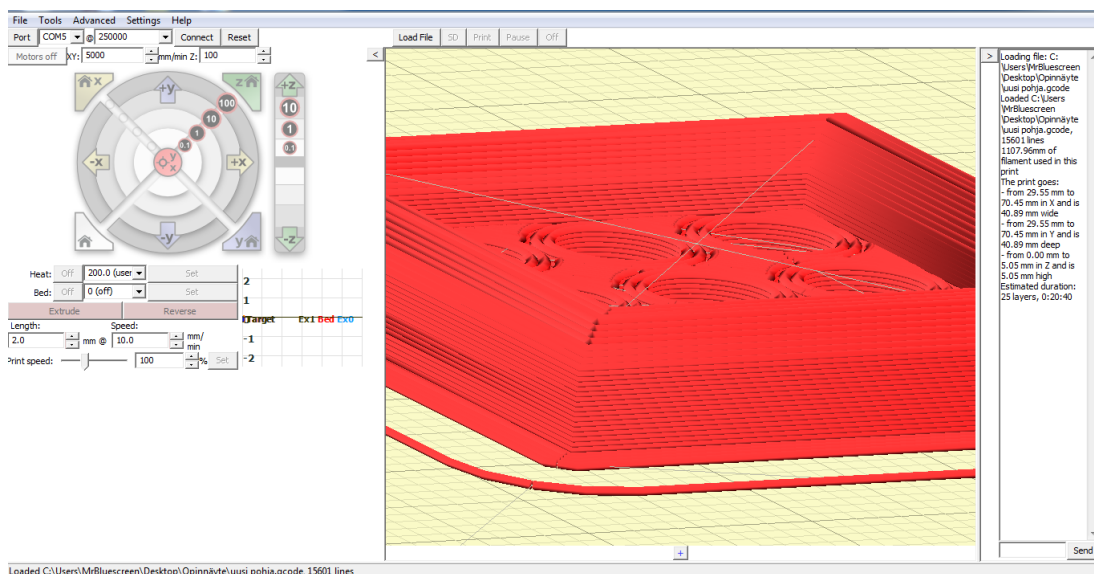


Kuva 3. EEPROM 24C02

3.4 3D-tulostus

3D-tulostus, tunnetaan myös materiaalia lisäävänä valmistuksena. Prosessi, jossa valmistetaan synteettisesti kolmiulotteinen objekti. Prosessissa 3D-malli muutetaan ensin STL-muotoon, joka on yksinkertaistettu malli, jota koneen on helpompi tulkita. Sen jälkeen STL-tiedosto vietään ”leikkuriksi” kutsutun ohjelman läpi, joka tuottaa mallista kerroskuvia, jotka tulkitaan G-koodiksi. [6](Wikipedia artikkeli 2016)

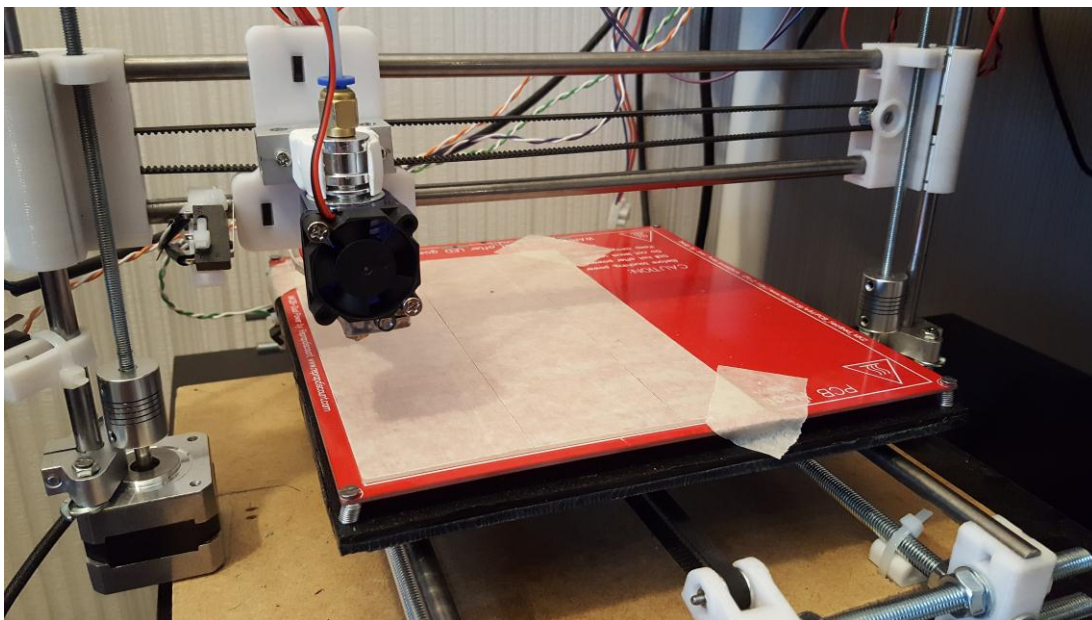
G-koodi on ohjelmointikieli, jota käytetään yleisesti erilaisissa työstökoneissa. G-koodi sisältää ohjeistuksen joka kertoo koneelle mihin suuntaan liikutaan ja kuinka paljon. Tätä samaista G-koodia käytetään 3D-tulostimessa.



Kuva 4. 3D malli josta näemme kerrokset, jotka tulostin tulostaa

3.5 3D-tulostin

Projektissa käytetty 3D-tulostin on tyypiltään materiaalia pursottava. Tulostin pursottaa termoplastista muovia kerros kerrokselta luoden lähes täysin yhtenäisen muovikappaleen.



Kuva 5. 3D-tulostimen voi rakentaa halvalla itse, jos kiinnostusta ja jaksamista riittää.

3.6 3D tulostusmuovit

3.6.1 PLA

PLA eli polyaktidi on uusiutuvista raaka-aineista valmistettu biohajoava termoplastinen alifaattinen polyesteri. Sen raaka-aineena voidaan käyttää esimerkiksi maissitärkkelystä tai sokeriruokoa. PLA-muovi on kovaa, eikä se kestä yhtä suuria lämpötiloja kuin ABS-muovi.[7](Wikipedia artikkeli 2016)



Kuva 6. PLA filamentti Octofiberilta

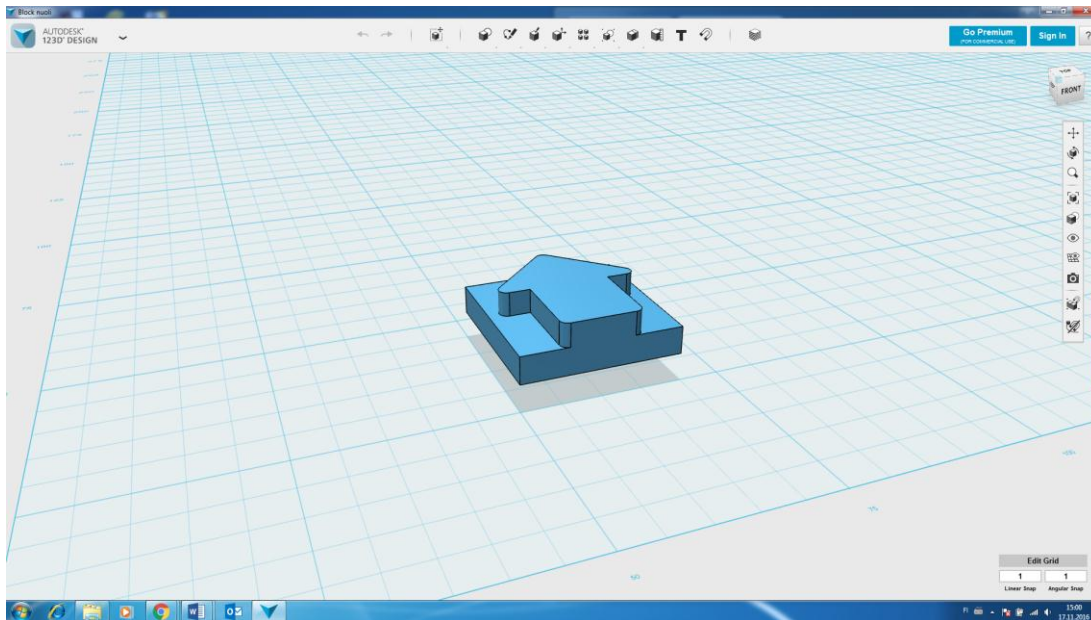
3.6.2 ABS

ABS eli akryylinitriilibutadieenistyreeni on eräs kestävä muovilaatu. Se on jäykkää mutta kevyttä ja hinnaltaan varsin edullista. Niinpä sen käyttökohteita ovat muun muassa erilaiset kypärät, kotitalous- ja konttorikoneet, kotelot, LEGO-palikat ja putket.

ABS-muovilla on hyvä sitkeyden, jäykkyyden ja sulajuoksevuuden yhdistelmä sekä hyvä pinnanlaatu ja kiilto. ABS-muovilla on myös pieni staattinen sähköisyys ja se kestää hyvin heikkoja happoja ja emäksiä. [8](Wikipedia artikkeli 2016)

3.7 123D Design

123D Design on ilmainen 3D-mallien luonti- ja muokkausohjelma Autodeskiltä. Se on erittäin helppokäyttöinen ja sopivan yksinkertainen 3D-tulostimen tulostustiedoston 3D-mallinnusta varten.



Kuva 7. 123D Design 3D-mallinnus ohjelma

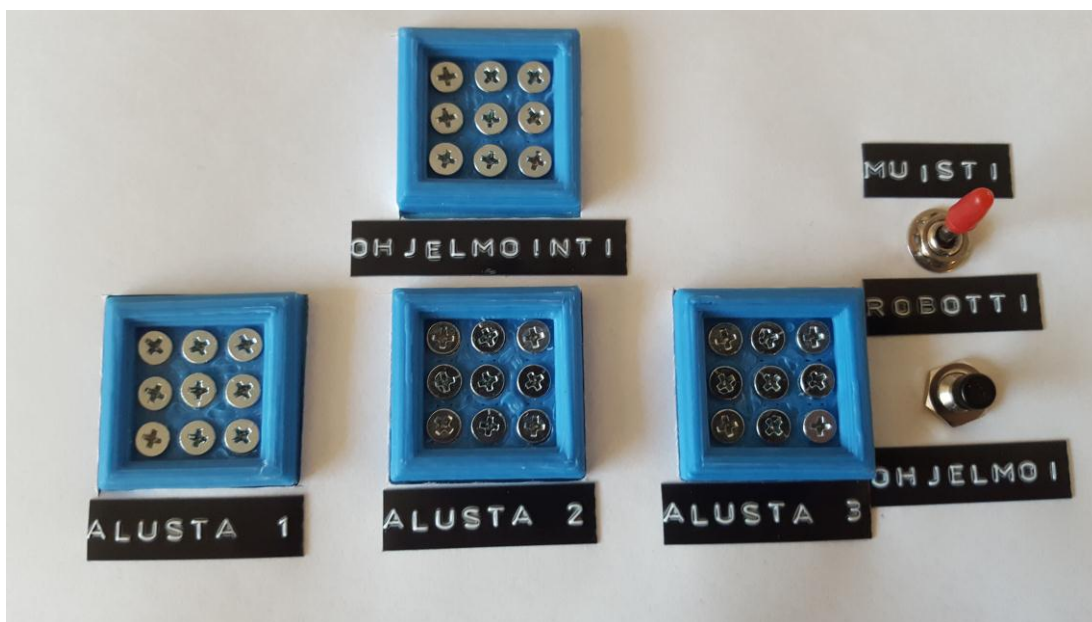
4 TOTEUTUS

Projektin kaikki muoviosat luotiin 3D-tulostustekniikkaa hyödyntäen ABS-filamentista, joka on todella kestävä muovilaatu. Valitsin 3D-tulostuksen, koska se on helppo tapa luoda omia tuotoksia ja erittäin halpa tapa tehdä pieniä prototyypppejä. 3D-mallit on tehty 123D Design ohjelmaa käyttäen. 123D Design on erittäin kevyt ohjelma ja juuri sopiva käytettäväksi esimerkiksi läppärillä.

4.1 Kokonaisuus

Laitteessa on 3 alustaa joihin erilaisia palikoita voidaan asettaa. 4. alusta on muistipalikan ohjelmointia varten. LCD-ruudun avulla pystyy tarkastelemaan muistipalikan sisältöä.

Laitteessa on myös 4 painiketta ja kytkin



Kuva 8. Ohjelmointipulpetti kokonaisuudessaan ilman LCD-näyttöä.

4.1.1 Palikat

Palikoita on tällä hetkellä 9 erilaista:

Liikkuminen eteenpäin ja kääntyminen oikealle ja vasemmalle

Muisti

Tulojen ohjaus

Sensorin luku

Kunnes

Jos

Analog

Liikkumis-palikat toimivat siten että eteenpäin käsky suoritetaan aina siihen suuntaan kuin robotin nokka on. Palikka voidaan asettaa alustalle miten päin tahansa ja näin helpotetaan asioiden hahmottamista.

Kääntymis-palikan käsky suoritetaan aina 90° käännöksinä ellei toisin määritellä (Lue: analog-palikka).

Muistipalikkalle voidaan tallentaa alustojen 1-3 toiminnot sen monille eri riveille tai sitä voidaan lukea ja suorittaa sen riveille tallennetut toiminnot. Muistipalikka sisältää 256 riviä, jolle kullekin voidaan tallentaa 1-3 toimintoa. Rivejä voidaan vaihtaa seuraava ja edellinen rivi painikkeilla. Ohjelmoi-painike tallentaa alustojen 1-3 toiminnot riville oli se tyhjä tai ei. Tällä tavoin ei tarvita erillistä rivin tyhjennyspainiketta ja jättämällä alustan tyhjäksi laite tyhjentää sen kohdan riviltä.

Näin monen rivin avulla pystytään luomaan monimutkaisia ohjelmia muilla palikoilla ja hahmottamaan kokonaisuutta.

Myöhemmin kehitetään laite, jolla pystytään näkemään koko ohjelma ja helpottamaan opetustilannetta.

Tulojen ohjaus-palikka ohjaa palikassa merkityn tulon päälle ja pois

Anturin luku-palikka lukee sensorin arvon.

Kunnes-palikkaa voidaan käyttää luomaan tyypillisen ohjelmarakenteen, jossa tiettyä toimintoa tehdään, kunnes saavutetaan ehto, jolla se lopetetaan.

Jos-palikalla luodaan toinen hyvin tyypillinen ohjelmarakenne, jossa odotetaan että 1. ehto täyttyy ja tehdään toiminto sen täytyttyä.

Analog-palikalla luodaan itse analogista tietoa palikan nuppia kääntämällä.



Kuva 9. Suoraan-, kääntymis- ja muisti-palikoita

4.1.2 Alustat

Alustassa on 9 kosketinta joista 4 on palikan tunnistusta varten, 2 on GND ja VCC, 2 on SDA ja SCL ja näiden lisäksi on 1 kosketin analogista mittausta varten. Alustoja on yhteensä 4. Kolmea ensimmäistä alustaa kutsutaan tässä projektissa riviksi ja 4. alusta on ohjelmointi alusta.



Kuva 10. Alustan koskettimina toimii 9 uraruuvia.

4.1.3 Painikkeet ja kytkin

Laitteen 4 painiketta ja kytkin:

Ohjelmointipainike, joka toimii joko käskynä suorittaa toiminnot tai käskynä tallentaa toiminnot muistipalikkaan

Seuraava rivi ja edellinen rivi painike, näiden avulla voidaan navigoida muistipalikan tallennusrivejä.

Tyhjennä painike, tyhjentää koko muistipalikan.

Valintakytkin, tämän avulla valitaan, että lähetetäänkö toiminnot ohjattavalle robotille vaiko ohjelmoidaanko ne muistipalikkaan.

4.1.4 LCD

LCD-näytöstä voidaan helposti lukea muistipalikan sisältö ja se helpottaa rivien välillä navigoimista

4.1.5 Toiminta

Alustoille 1-3 voidaan asettaa perustoimintoja, kuten eteenpäin ja käänny vasemmalle tai oikealle. Näillä voidaan ohjailla robottia suoraan. Alustoille voidaan asettaa myös muisti-palikka, jonka sisälle tallennetut toiminnot voidaan suorittaa. Laite tukee tällä hetkellä vain yhden muistipalikan lukua, mutta on jälkeinpäin laajennettavissa 3.

Alustoille voidaan asettaa tulojen ohjaus-palikoita, jotka ohjaavat tuloja päälle ja pois. Tälle toiminnolle on vain yksi palikka ja tämä pitää ottaa huomioon ohjattavaa robottia ohjelmoitaessa. Ohjelma pitää siis kirjoittaa, siten että sama käsky muuttaa digitaalisen tulon tilaa.

Laitteella voidaan lukea myös antureita anturin luku-palikalla, mutta tämän käyttö vaatii tuekseen lisä palikoita, joita ovat kunnes- ja jos-palikat.

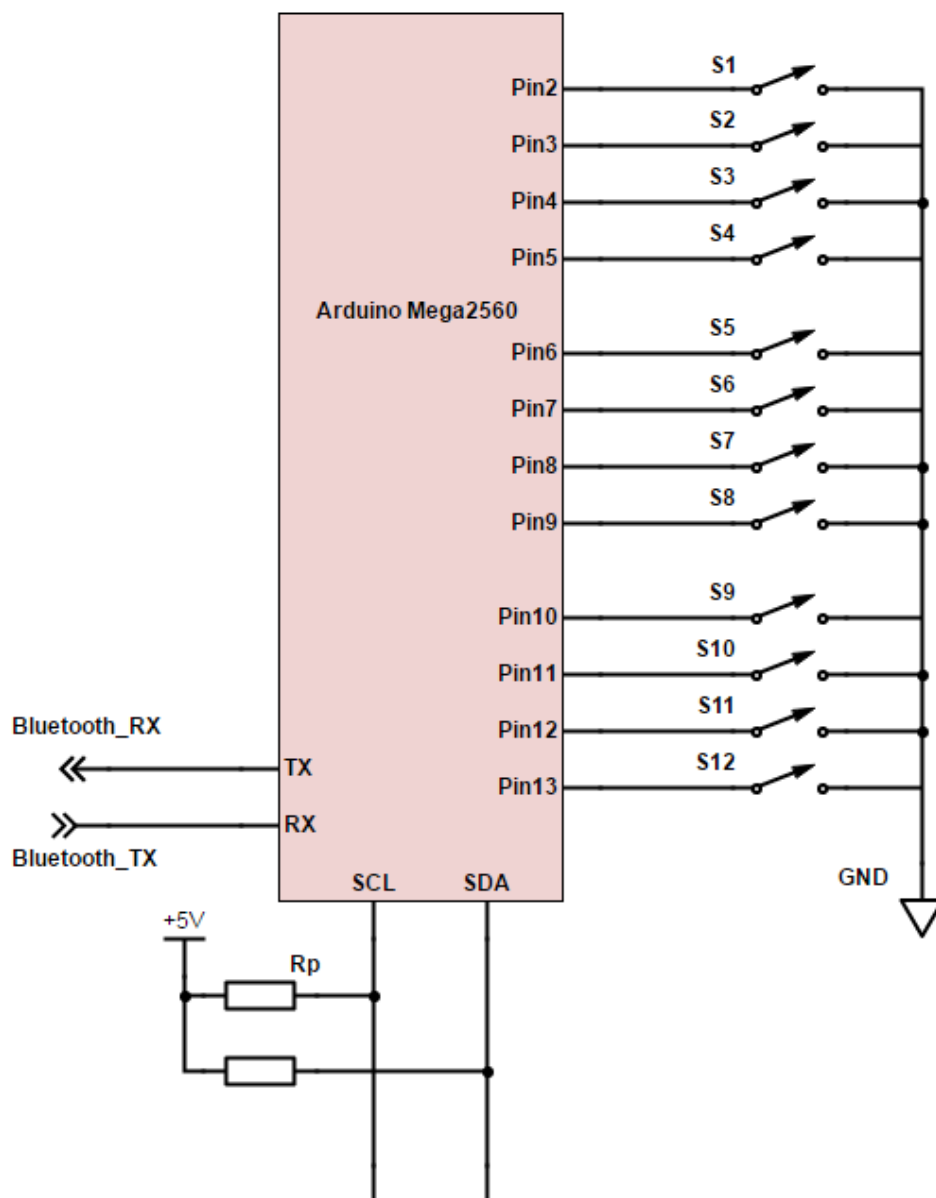
Kunnes-palikkaa käytettäessä se asetetaan 2. alustalle ja laite lukee rivin näin: alusta 1. kunnes alusta 3.

Edistyneemmät ohjelmoijat voivat käyttää myös monimutkaisempia palikoita. Näitä ovat jos-palikka ja analog-palikka. Jos-palikka muuttaa ohjelmointipulpetin lukutapaa. Lukutapa muuttuu siten, että kun jos-palikka asetetaan 1. alustalle laite lukee seuraavat 2 alustaa siten että ohjelman rivi näyttää tältä: jos alusta 2. niin alusta 3. Analog-palikka hyödynnetään analogista tietoa eri toiminnoissa. Kun analog-palikka asetetaan alustalle, laite lukee sen ja sitä seuraavan palikan ja yhdistää nämä yhdeksi toiminnoksi. Eli jos esimerkiksi asetetaan analog-palikka ja tämän nuppi on käännettynä keskelle ja siitä seuraavaksi palikaksi käännös-palikan näiden yhdistetty toiminto on 45° käännös.

4.2 Elektroniikka ja ohjelmointi

4.2.1 Arduino

Arduinon tullaan kytkemään 3 alustaa joista kukin käyttää 4 digitaalista pinniä eli yhteensä 12 pinniä. EEPROMia käytetään I²C väylän kautta, joka toimii Arduino Megassa sille osoitettujen pinnien kautta 20 (SDA) ja 21 (SCL). Bluetooth moduulia käytetään UARTilla, joka toimii pinneillä 0 ja 1. Seuraavassa kuvassa on kytkentä kokonaisuudessaan, painikkeet simuloivat alustoja.

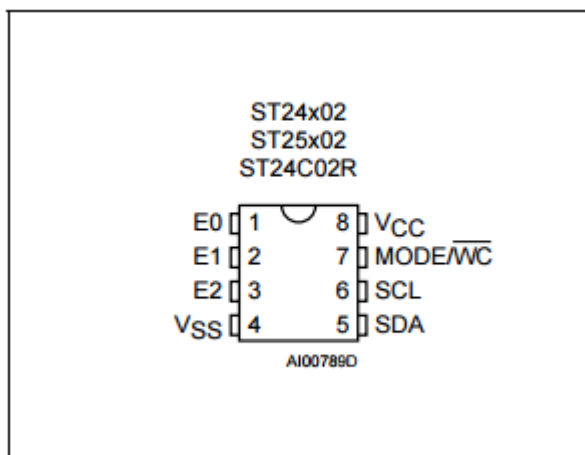


Kuva 11. Arduino kytkennät

4.2.2 EEPROM

Jokaisessa Arduinossa on sisäinen EEPROM, mutta tässä projektissa käytetään ulkoista EEPROMia. EEPROMin liittäminen Arduinoon on helppoa I²C väylän avulla. EEPROMeja on monenlaisia ja tässä projektissa käytetään 24C02 EEPROMia. 24C02 on 2Kbitin muisti, joka on jaettu 32 riviin josta kukin on 8 bittiä (256 sanaa). 24C02:ssa on [hard wire] osoite joka on 3 bitin pituinen (pinnit E0-E2), mikä tarkoittaa

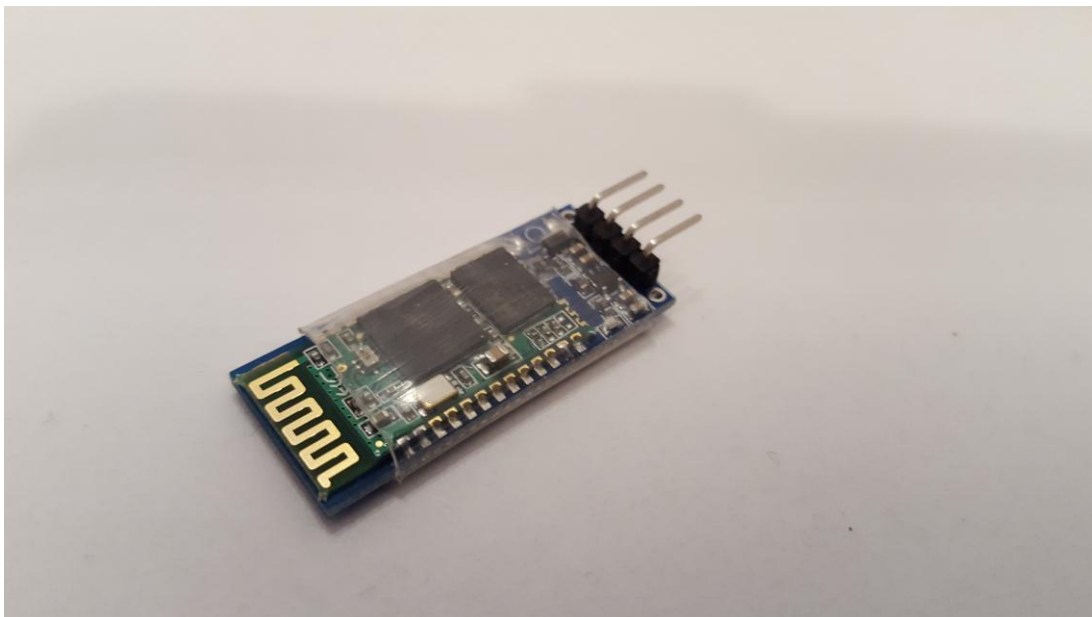
taa, että Arduinoon voidaan liittää maksimissaan 8 (2^3) muistia. Osoite annetaan muistille vetämällä pinni joko GND tai +5VDC esim. jos kaikki pinnit vedetään GND, muistin osoite on 0. 24C02 tarvitsee osoitteen alkuun tunnistus koodin, joka on 1010. Tähän lisätään perään pinnien tila 1010E2E1E0, niin saadaan koko osoitteeksi 1010000 (Heksana 50).[9](Datalehti 1997)



Kuva 12. EEPROM pinni layout.

4.2.3 Bluetooth

Bluetooth lisätään projektiin ulkoisen kommunikoinnin helpottamiseksi. Bluetooth-moduuli on helppo liittää Arduino projekteihin. Se käyttää sarjaliikenneväylää UART, mutta koska Arduino käyttää samaa väylää myös tietokoneen kanssa kommunikointiin on parempi käyttää ohjelmallista sarjaliikenneväylää (tälle löytyy valmis kirjasto Arduino IDE:stä SoftwareSerial.h). Ohjelmallinen sarjaliikenneväylä luo ”virtuaalisen” sarjaliikenneportin haluttuihin digitaalisiin I/O pinneihin.



Kuva 13. Bluetooth moduuli.

4.3 Ohjelma

4.3.1 Palikan tunnistus

Palikan tunnistamista varten on kirjoitettu yksinkertainen ohjelma. Ohjelma sisältää kaikkien alustojen lukemisen oikeassa järjestyksessä vasemmalta oikealle. Ohjelma tallentaa nämä muuttujaan, joka lähetetään joko muistipalikalke tai Bluetoothin kautta muualle. Ohjelma lukee pinnien tilat ja vertaa niitä IF-lauseessa valmiiksi annettuihin ehtoihin. Tässä tapauksessa palikan asennosta huolimatta, jos yhden pinnin tila on LOW ohjelma tunnistaa, että kyseessä on eteenpäin käsky. Ja vastaavasti jos kaksi pinniä on LOW niin ohjelma tietää, että kyseessä on kääntymis käsky oikealle, kolme tarkoittaa vasemmalle. Näin varmistetaan, että yksinkertaisten palikoiden välillä ei tule sekaantumisia.

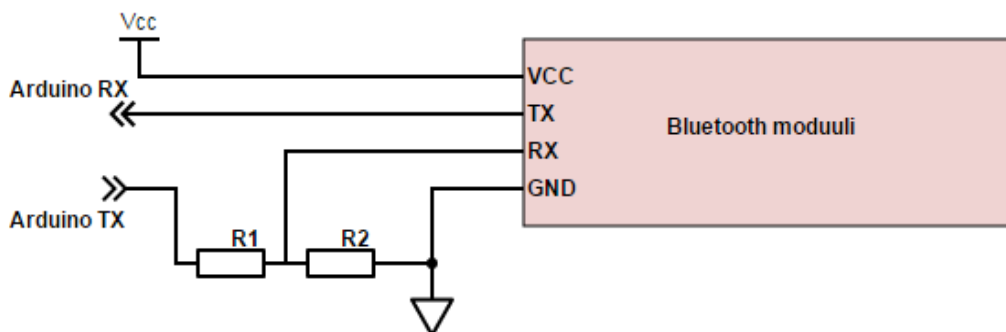
Palikan tunnistusohjelmaan kuuluu myös analogisen tiedon lukeminen, kun analogipalikka asetetaan alustalle.

4.3.2 EEPROM kirjoitus ja luku

EEPROMin käyttäminen on yksinkertaista Arduinolla, koska voidaan käyttää valmista kirjastoa Wire.h jonka avulla voidaan kommunikoida I²C laitteille. Wire.h kirjaston käyttäminen voi vaikuttaa hankalalta ensin, mutta pienen käytön jälkeen se on hyvin yksinkertainen. Ainoa hieman monimutkaisempi asia tässä käytössä on EEPROMin sisäisen osoitteen ”navigointi”. EEPROMilla on sisäisiä bitti osoitteita välillä 0-255. EEPROMin lähetyksen protokolla toimii niin että ensin lähetetään MSB eli eniten merkitsevä bitti ja sen jälkeen LSB eli vähiten merkitsevä bitti. Nämä ovat siis tässä tapauksessa 4 ensimmäistä bittiä ja 4 jälkimmäistä bittiä tavusta.

4.3.3 Bluetooth lähetyksen protokolla

Projektissa käytetään HC-06 Bluetooth moduulia, joka on helppokäyttöinen ja halpa tapa liittää Bluetooth projekteihin. HC-06 käytetään samaan tapaan kuin mitä tahansa muuta laitetta mikä liitetään UART väylään. Ainoana esteenä moduulin liittämällä suoraan laitteeseen on, että moduuli käyttää eri jännitetasoa kommunikoinnissa. HC-06 käyttää 3,3V kun taas Arduino käyttää 5V. Tämä ei ole ongelma, kun tietoa lähetetään moduulilta Arduinolle, mutta Arduinolta moduulille kommunikointi vaatii jännitetason muutoksen. HC-06 toimii kyllä ilman tätä, mutta kun moduulia käytetään pidemmän aikaa väärällä jännitteellä, sen vahingoittuminen on hyvin todennäköistä. [10](Datalehti 2006)



Kuva 14. Moduulin kytkentä, jännitetason muuttaja on tässä tapauksessa yksinkertainen jännitteen jakaja

5 TESTAUS

Testaaminen on asia mitä pitäisi tapahtua koko kehitys kaaren aikana. Jokaista osaa laitteesta kannattaa testata niiden valmistuttua ja aina kun toinen osa liitetään kokonaisuuteen on parasta testata, että edelliset osat toimivat edelleen niin kuin pitäisi.

Tämä osa ei ole pelkästään sitä, että katsotaan että laite toimii vaan myös sitä varten että tutkitaan laitteen toimintaa opetuksen apuvälineenä.

5.1 Laitteen toiminta

Laitteen testaus aloitettiin testaamalla palikan tunnistusta. Palikan tunnistus on helppo testata siten että palikoita asetetaan laitteeseen ja luetaan sarjamonitorista, että jokaisen alustan muuttuja vastaa oikeaa palikkaa. Samaan tapaan voidaan testata muistipalikan toiminta, mutta jotta voisin olla varma siitä, että laite kirjoittaa todella EEPROM:iin käytiin testaamisessa toista Arduinoa, joka oli ohjelmoitu lukemaan koko muisti.

Kun tiedetään että kaikki toiminnot toimivat, testataan lähettääkö laite nämä toivotusti eteenpäin kohdelaitteelle. Testaus toteutettiin siten, että laitteen lisäksi oli toinen Arduino, joka lukee Bluetoothin lähettämät tiedot ja kirjoittaa ne sarjamonitoriin.

5.2 Opetus

Laitteen testaaminen opetuksessa on monimutkainen prosessi, koska ohjelmointi sisältää paljon opetettavaa. Tästä syystä laitteella testattiin vain muutaman ohjelmointitaidon opettamista.

5.2.1 Ohjelmointitaito: ohjelma

Testasimme kohderyhmään kuuluvan testihenkilön kanssa, että pystyykö laitteen avulla selittämään mitä ohjelma tarkoittaa. Kävimme läpi lyhyitä kolmen askeleen tehtäviä, joissa yritetään saada robotti maaliin. Mietimme yhdessä mitä robotin pitää

tehdä, jotta se pääsee maaliin ja kuinka se kerrotaan robotille. Testihenkilö ymmärsi hyvin mitä ohjelma tarkoittaa.

5.2.2 Ohjelmointitaito: ohjelmallinen ajattelu

Ohjelmallisen ajattelun opettelu tapahtui samalla kun edellistä kohtaa opetettiin. Ohjelmallisesta ajattelusta opetettiin mitä asioita robotti ymmärtää ja miten ne pitää esittää robotille. Tähän ajattelutapaan kuuluu esimerkiksi se että robotti ei tiedä mihin suuntaan se on kääntyneenä.

5.2.3 Ohjelmointitaito: algoritmi

Algoritmin tekemistä opetettiin tekemällä tehtäviä siten, että kirjattiin kaikki askeleet joita robotin pitäisi tehdä, jotta se pääsisi maaliin. Tämän jälkeen kokeiltiin laitteella, että pääseekö robotti tosiaankin maaliin eli, tarkastetaan onko vastaus ”oikea”.

6 LOPPUPÄÄTELMÄ

6.1 Yhteenveto

Opinnäytetyössä suunniteltiin ja toteutettiin laite, joka tulisi toimimaan apuvälineenä ohjelmoinnin opetuksessa. Laitteen avulla konkretisoidaan ohjelmoinnin periaatteita ja siten tehdään siitä helpommin ymmärrettävä ja mielenkiintoisempi.

Laitteen avulla voidaan opetella ohjelmoinnin peruskäsitteitä ja sen monia ajattelutapoja. Laitteen kanssa opetellaan esimerkiksi ohjelmallista ajattelua, joka on hyvin olennainen tapa ajatella, kun halutaan ohjelmoida esimerkiksi robotteja.

Laitteen toteuttaminen oli nopeaa, koska suunnitelma laitteesta oli hyvin perusteellinen. Sen lisäksi että suunnitelma oli hyvä, projektia edisti oma kokemukseni ohjelmoinnista ja monet hyvin luodut ohjelmointi kirjastot.

Laitetta testattiin opettelukäytössä, mutta se jäi käytännössä paljon lyhyemmäksi ja suppeammaksi kuin laitteella olisi ollut potentiaalia. Syytä tähän on yksinkertaisesti se että aika joka ohjelmoinnin opetteluun käytetään kouluissa on paljon pidempi kuin mitä minulla on aikaa kirjoittaa tätä opinnäytetyötä.

6.2 Jatkokehitys

Tämä opinnäytetyö toimi idean testauksena ja ei ole tarkoitettu kaupallistamista varten.

Laitte on luotu siten, että siinä on riittävästi potentiaalia kehittyä paljon enemmän. Laitteeseen valittiin juurikin tästä syystä Arduino Mega, jossa on 54 digitaalista I/O:ta. Myös projektissa käytetty EEPROM on tarpeeksi suuri monimutkaisille ohjelmille, mutta se on myös helposti vaihdettavissa suurempaan muistiin. Kirjasin suunnitelmaani paljon enemmän toimintoja kuin toteutin, koska aika oli hyvin rajallinen. Laitetta olisi voinut kehittää vielä vaikka kuinka paljon, mutta johonkin oli vain pakko lopettaa.

LÄHTEET

- [1] Oppi & ilo 2016. Koodauspläjäys
- [2] Oppi & ilo 2016. Koodauspläjäys
- [3] Arduinon www-sivut. Viitattu 8.10.2016. <https://www.arduino.cc/en/Main/arduinoBoardMega>
- [4] I²C-väylän määrittely ja käyttäjä manuaali 2014. http://www.nxp.com/documents/user_manual/UM10204.pdf
- [5] Wikipedia artikkeli. Viitattu 15.10.2016. <https://fi.wikipedia.org/wiki/EEPROM>
- [6] Wikipedia artikkeli. Viitattu 17.11.2016. https://fi.wikipedia.org/wiki/Kolmiulotteinen_tulostus
- [7] Wikipedia artikkeli. Viitattu 18.11.2016 https://en.wikipedia.org/wiki/Polylactic_acid
- [8] Wikipedia artikkeli. Viitattu 18.11.2016 https://en.wikipedia.org/wiki/Acrylonitrile_butadiene_styrene
- [9] Mikropiirin datalehti. 1997. <http://pdf1.alldatasheet.com/datasheet-pdf/view/23727/STMICROELECTRONICS/24C02.html>
- [10] Bluetooth moduulin datalehti. 2006. <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>