



TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

3D PRINDITUD ELEKTRIMASINATE JUHTIMISMUUNDURI EDASIARENDUS

FURTHER DEVELOPMENT OF DRIVE FOR 3D PRINTED MOTOR

MAGISTRITÖÖ

Üliõpilane: Aleksei Tsikin

Üliõpilaskood: 176835

Juhendaja: Anton Rassõlkin, teadur

Tallinn 2019

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“.....” 201.....

Autor:

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

“.....” 201.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

“.....”201... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

LÕPUTÖÖ LÜHIKOKKUVÕTE

<i>Autor:</i> Aleksei Tsikin	<i>Lõputöö liik:</i> Magistritöö
<i>Töö pealkiri:</i> 3D prinditud elektrimasinate juhtimismuunduri edasiarendus	
<i>Kuupäev:</i> 24.05.2019	64 lk
<i>Ülikool:</i> Tallinna Tehnikaülikool	
<i>Teaduskond:</i> Inseneriteaduskond	
<i>Instituut:</i> Elektroenergeetika ja mehhatroonika instituut	
<i>Töö juhendaja(d):</i> teadur Anton Rassõlkin	
<i>Töö konsultant (konsultandid):</i> vanemteadur Ants Kallaste	
<i>Sisu kirjeldus:</i> Töö eesmärgiks on uurida 3D prinditud elektrimasinate erijuhtimine, ning arendada juhtimis süsteemi kasutades Arduino mikrokontrollerit. Lõputöö käigus arendatud muunduri prototüübil peab olema lihtne ja intuitiivne kasutajaliides, ning tagasiside andurid mis annavad pidevat tagasisidet mootorite talitlustest. Lõputöö on jagatud viieks osaks. Esimeses osas kirjeldatakse millest koosneb juhtimis muundur ja turu uuring, selleks et aru saada mis on vaja muunduri kokkupanemiseks. Teises osas on vajalik moodulite valik ja moodulite tehnilise kirjeldus. Kolmas osa on skeemide koostamine ja prototüübi kokkupanek. Neljas osa on algoritmide koostamine ja prototüübi programmeerimine. Viimases osas kirjeldatakse prototüübi katsetamise. Töö tulemuseks on juhtimismuundur samm mootorite jaoks.	
<i>Märksõnad:</i> Elektrimootorite juhtimine, 3D printimine, mikrokontrolleri programmeerimine	

ABSTRACT

Author: Aleksei Tsikin

Type of the work: Master Thesis

Title: Further development of drive for 3D printed Motor

Date: 24.05.2019

64 pages

University: Tallinn University of Technology

School: School of Engineering

Department: Department of Electrical Power Engineering and Mechatronics

Supervisor(s) of the thesis: resercher Anton Rassõlkin

Consultant(s): senior resercher Ants Kallaste

Abstract:

The aim of the work is to study the control possibilities of 3D printed electrical machines, and to develop a control system using an Arduino microcontroller. The converter prototype developed in the thesis must have a simple and intuitive user interface, and feedback sensors that provide continuous feedback on motor performance.

The thesis is divided into five parts. The first section describes what driver converter consists off and a market survey to understand what is needed to assemble the converter. In the second part, a selection of modules and a description of the modules. The third part is the design of the schemes and the assembly of the prototype. The fourth part is compiling algorithms and programming the prototype. The final part describes prototype testing.

The work results in a stepper motor drive.

Keywords: Electric motor control, 3D printing, microcontroller programming

Elektroenergeetika ja mehhatroonika instituut

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Aleksei Tsikin, 176835
Õppekava, peeriala: AAAM, Energiamuundus- ja juhtimissüsteemid
Juhendaja(d): Anton Rassõlkin, Teadur, 6203807
Konsultandid: Ants Kallaste, vanem teadur, 6203807

Lõputöö teema:

(eesti keeles) 3D prinditud elektrimasinate juhtimismuunduri edasiarendus
(inglise keeles) Further development of drive for 3D printed Motor

Lõputöö põhieesmärgid:

1. Uurida turul olevaid elektromasinate juhtimis lahendusi
2. Arendada 3D prinditud mootori juhtemis süsteemi
3. Panna kokku juhtimis muunduri

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Skeemide koostamine, komponentide tellimine	22.02.2019
2.	Prototüübi programmeerimine, teoreetilise osa kirjutamine	08.04.2019
3.	Lõplik koostamine ja testimine	22.04.2019

Töö keel: eesti **Lõputöö esitamise tähtaeg:** "24" mai 2019a

Üliõpilane: Aleksei Tsikin "....."201....a
/allkiri/

Juhendaja: Anton Rassõlkin "....."201....a
/allkiri/

Konsultant Ants Kallaste "....."201....a
/allkiri/

SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE	3
ABSTRACT	4
EESSÕNA.....	7
LÜHENDITE JA TÄHISTE LOETELU	8
SISSEJUHATUS.....	9
1. TEHNIKA TASEME ANALÜÜS JA KASUTATAV TOPOLOOGIA	10
1.1 Turu uuring	10
1.2 Arduino.....	12
1.2.1 PWM.....	13
2 NÕUDED JA NENDE PÕHJENDUSED	17
2.1 Erinevad võimalikud valikud.....	17
2.1.1 Arduino	17
2.1.2 HMI valik.....	19
2.1.3 Draiverite valik	20
2.1.3 Vooluandurite valik	22
2.1.4 Pingeandur.....	23
2.2 Seadmete valik ja konstruktsioon.....	24
2.3 Valitud seadmete konstruktsioon.....	26
3 PROTOTÜÜBI KOKKUPANEK	29
4 PROTOTÜÜBI PROGRAMMEERIMINE JA SEADISTAMINE.....	32
4.1 HMI Programmeerimine	32
4.2 Mikrokontrollerite programmeerimine.	42
5 KATSETAMINE	51
KOKKUVÕTE.....	57
CONCLUSION	58
KASUTATUD KIRJANDUSE LOETELU	59
LISAD	61

EESSÕNA

Käesolev töö on 3D prinditud elektrimasinate uuringu projekti osa. Töö oli pakutud Tallinna Tehnikaülikooli elektroenergeetika ja mehhatroonika instituudi teaduri Anton Rassõlkini poolt.

Töö käsitleb 3D prinditud elektrimasinate juhtimismuunduri kokkupanemist, ning juhtimisalgoritmi loomist.

Elektrimootorite juhtimine, 3D printimine, mikrokontrolleri programmeerimine,

LÜHENDITE JA TÄHISTE LOETELU

3D - kolmemõõteline

PWM – Pulsilaiusmodulatsioon

HMI – Inimese-masina liides

WGM – Signaali generaator

CS – Kella vali

UART – universaalne asünkroonne vastuvõtja/saatja

NMOS – n-kanali MOSFET

SPI - sünkroonse järjestiksuhtluse liides

DC – alalisvool

I/O – sisend ja väljund

EEPROM - elektriliselt kustutatav programmeeritav püsimälu

RAM – suvapöördusmälu

CW – ClockWise, kellaosuti

CCW – CounterClockWise, vastu kellaosuti

ADC - analoog digitaal konverter

SISSEJUHATUS

Tänapäeval sammootorid kasutatakse igal pool ja muidugi 3D printerites. Elektrimasinate 3D printimine on huvitav ja innovaatiline, kuna tegemist on metallidega. Samal ajal 3D printimise tehnoloogia areneb väga kiiresti ja muutub odavamaks. Mõned ettevõtted juba pakuvad 3D printimise teenuseid, seetõttu on vaja leida odav lahendus, millega saab juhtida 3D prinditud elektrimasinaid.

Turul on palju erinevaid sammootori kontrollereid, millega saaks juhtida 3D prinditud mootorit. Tavaliselt juhtimismuundur koosneb: kontrollerist, taktgeneraatorist ja draiverist.

Arduino mikrokontroller on üks lahendus, mida hakkab kasutama selles töös. Kõige suurem ja võimsam mikrokontroller Arduino seeriast on Arduino Mega, kuhu saab ühendada kõik vajalik andurid ja ekraani. Arduino kontrolleriiga saab ka juhtida draiverit, kuna on PWM ja digitaalsed väljundid.

Magistritöös "3D PRINDITUD ELEKTRIMASINATE JUHTIMINE ARDUINO MIKROARVUTIGA" on mitu puudutuseid. Näiteks: puudub kasutajaliides, parameetrite sisestamiseks kasutatakse 4 potentsiomeetrit, PWM-i edastamiseks L298n draiverile kasutatakse analogwrite() funktsiooni, mis ei ole kõige parem variant, kuna PWM-i sagedus on 490hz ja muunduris puuduvad andurid. Kood on keeruline, kuna puuduvad kommentaarid. Muunduri eelised on järgmised: kompaktne ja on kasutatud ainult üks Arduino UNO.

Nüüd kui mul on selged eelmise aasta töö puudutused, võin arendada uue muunduri. Andmete sisestamiseks tahan kasutada 2 potentsiomeetrit: üks nendest juhtimis impulsi laiuse sisestamiseks ja teine impulsi sageduse sisestamiseks. Ekraanis saab valida kas tegemist on 6/4 või 8/6 mootoriga. Samuti saab näha: impulsi laiust protsentides, impulsi sagedus (Hz), iga mähise vool ja pinge. Tööde jaoks on vaja vähemalt 2 Arduinot, üks draiveri juhtimiseks ja teine parameetrite sisestamiseks ja informatsiooni ekraanile väljatrükkimiseks.

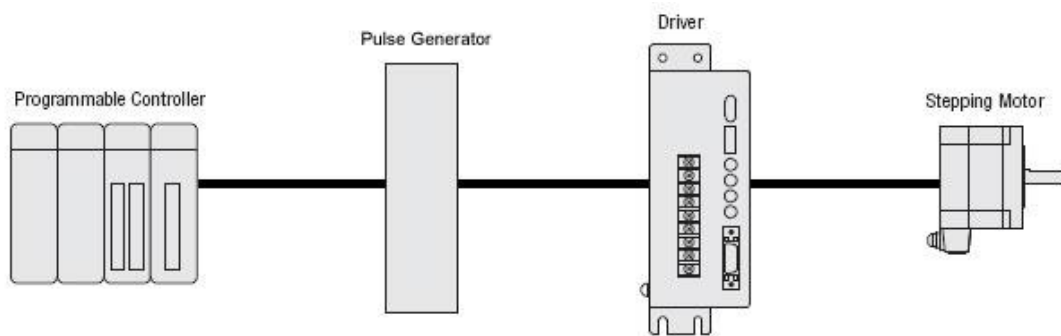
Lõpptoode peab olema kompaktne ja kindel. Erinevate moodulite valik ja Arduino programmeerimine on ka huvitav aspekt, miks ma valisin selle töö. Muunduri loomine, kokkupanek ja testimine on põnev protsess. Samal ajal sain rohkem teada elektrimasinate 3D printimisest ja sammootorist.

Tööeesmärk on 3D prinditud sammootori juhtimise muunduri loomine ja kokkupanek.

1. TEHNIKA TASEME ANALÜÜS JA KASUTATAV TOPOLOOGIA

Samm-mootori juhtimis muundur on elektrooniline jõuseade, mis juhtsignaalide põhjal juhib samm-mootori kõrgvoolu mähiseid ja võimaldab samm-mootoril astuda samme (pöörata).

Samm-mootorit on palju raskem juhtida, kui tänapäevast kollektormootorit – on vaja sisse- ja välja lülitada mootori mähised, antud järjekorras ja täpse impulsi laiusega. Selleks kasutatakse juhtimis muundurit.



Joonis 1.1 Samm-mootori juhtemis muundur [1]

Joonisel 1.1 on kontrolleri (Programmable controller), taktigeneraatori (pulse generaator) draiveri (driver) ja samm-mootori (stepping motor). Kontrolleri on süsteemi osa, mis juhib automaatselt süsteemi ja kogub ning analüüsib anduritelt saadud informatsiooni süsteemi oleku kohta. Taktigeneraator genereerib pulsilaiusmodulatsiooni. Pulsilaiusmodulatsiooni väljundiks on konstantse pingeamplituudiga elektriliste impulsside jada, kus vajaliku kujuga signaali saamiseks muudetakse impulsside kestust (laiust) konstantse perioodi korral [2]. Draiver on seade, mis muudab väikese võimsusega juhtsignaalid vooluks, mis on piisav mootorite juhtimiseks.

Samm-mootori juhtimiseks, kasutaja sisestab vajalikud parameetrid kontrollerrisse, mis omakorda edastab sisestatud parameetrid taktigeneraatorisse. Taktigeneraator edastab juht signaalid draiverile, mis omakorda lülitab väljundid teatud sagedusega ja pulsi laiusega.

1.1 Turu uuring

Tänapäeval turul võib leida erinevaid lahendusi samm-mootori juhtimiseks. Need lahendused on saadaval erineva suurusega ja on ettenähtud erineva samm-mootori jaoks, aga peamised parameetrid on: nimivõimsus, nimivool, toitepinge, sammu pikkus. Selleks, et aru saada millest koosneb samm-mootori muundur, vaatame turul kättesaadavaid lahendusi.

ABB NextMove e100 motion controller ja ABB MicroFlex Servo Drive

NextMove e100 tugineb tõestatud NextMove'i kontrolleri tooteperekonnale, integreerides reaajas Etherneti juhtimise tööstusstandardi Etherneti POWERLINK abil. Kõiki draivereid, I / O-seadmeid, andureid, saab POWERLINK-võrku lisada ja juhtida MINT-programmeerimiskeeles. See lihtsustab oluliselt süsteemi projekteerimist ja paigaldamist ning samal ajal laiendab NextMove e100 võimalusi masina juhtplatvormina.

NextMove e100 suudab interpoleerida kuni 16 telge, kas ühe koordinaatgrupina või sõltumatu alamrühmana mitmetasandilise koordinaatide süsteemis interpoleerides Etherneti POWERLINK võrgus või ehitatud astmik / servo teljed. [3]

ABB NextMove e100 motion controller on suure jõudlusega controller mis sisaldab:

- 20 digitaalsisendit
- 12 digitaalset väljundit ja 1 rele
- 2 analoogsisendit
- 4 analoogväljundit
- USB port
- Seeriaport HMI-ühenduse jaoks
- Arvutiga ühendamiseks kasutatavad erinevad pordid
- Programmeeritakse MINT-s

Analoog I/O kasutatakse analoogseadmete ühendamiseks. Digitaalsed sisendid ja väljundid on loodud töötama digitaalsete seadmetega. Enkooderi ühendamiseks on spetsiaalsed väljundid.

Mootori otsejuhtimiseks kasutatakse ABB MicroFlex Servo Drive'i[4]. Peamised parameetrid:

- Ühendab otse toiteallikaga.
- Voolud kuni 3A, 6A ja 9A sõltuvalt mudelist
- Pöörlemis sammu ja pöörlemissuunda on võimalik vahetada.

TB6600 samm-mootori driver

TB6600 samm-mootori draiver moodul loodi professionaalsete ja amatöör CNC süsteemide loomiseks suure võimsusega samm-mootoritel.

PWM ja mootori mähiste intelligentse voolujuhtimise abil saab ühendada samm-mootorid laia toitepinge ja vooluga - kiip piirab mootori maksimaalset mähisvoolu määratud tasemel. [5]

Peamised parameetrid:

- Toitepinge: 48V
- Sisendvool: 0-5A

- Väljundvool: 0,5-4A
- Juhtimissignaali: 3,3-24V
- Võimsus (max): 160W

Schneider electric SD315DN10B400 [6]

SD315DN10B400 on muundur samm-mootorite jaoks. Peamised parameetrid:

- Toitepinge 19-60VDC
- Võimalik valida samm ja pöörlemisuuna
- Väljundvool kuni 10A faasi kohta
- Nimi võimsus 590W
- Vähem kui 7W võimsuskadu
- Kohalikud tööhäired

Tabel 1.1 Erinevate muunduri peamised parameetrid

	ABB	TB6600	SD315DN10B400
Väljundvool, A	3, 6, 9	4	10
Toitepinge, V	230AC	48DC	19-60DC
Võimsus, W	500-300	160	590

Tabelis 1.1 on erinevate muundurite peamised parameetrid, selleks et saab kohe näha millised tehnilised parameetrid on vaja muunduri arendamiseks. Nüüd, saades lühidalt ülevaate turul saadaval muunduritest, saab teha loendi muunduri vajalikest võimalustest:

- HMI (ekraan), et kuvada vajalik teave
- Toitepinge kuni 50 V
- Väljundvool kuni 5A
- Voolu- ja pingeadurid
- Võime valida sammu, pöörlemisuuna ja faaside arvu

1.2 Arduino

Arduino on avatud lähtekoodi mikrokontroller, mida tänapäeval kasutakse erinevates projektides: robotid, mootori juhtimine, erinevad mõõteseadmed ja palju muud. Lihtsamalt öeldes Arduino on pisike arvuti millele saab ühendada vajalikke elektroonika komponente. Tegemist on programmeeritava loogikakontrolleriga, mis tähendab seda, et kasutaja võib kirjutada ise vajalikud

programmid valmis ning anda Arduinole teatud käsklusi. Kuna tegemist on avatud lähtekoodi mikrokontrolleriga, siis igauks võib ühendada, ehitada, programmeerida kuidas iganes ta seda soovib. Arduino on saadaval trükkplaadina, kus kasutatakse erinevaid ATmel kiipe. Programmeerimiseks kasutatakse C keele loogikat mis on kättesaadav erinevates teegid, mis omakorda lihtsustab koodi kirjutamist.

1.2.1 PWM

Pulsilaiusmodulatsioon-signaal on konstantse amplituudiga impulsside jada, kus periood on konstantne, aga impulsside kestus muutub. PWM-i peamine kasutusala on elektriliste seadmete võimsuse kontrollimine. [7]

Kuna tänapäevased muundurid kasutavad PWM-i, et muuta faasidele rakendatud pinge, siis selle töö jaoks pinge muutmiseks kasutatakse ka PWM. Pulsilaiuse modulatsiooni (PWM) saab Arduino-s rakendada mitmel viisil.

analogWrite funktsioon.

Arduino programmeerimiskeel muudab PWM-i kergesti kasutatavaks; lihtsalt kutsume `analogWrite (pin, dutyCycle)` funktsiooni, kus `dutyCycle` on väärtus vahemikus 0 kuni 255 ja `pin` on üks PWM-pinist (3, 5, 6, 9, 10 või 11). Funktsioon `analogWrite` pakub lihtsa liidese riistvara PWM, kuid ei anna mingit kontrolli sageduse üle. (Pange tähele, et vaatamata funktsiooni nimele on väljund digitaalne signaal.) [8]

Bit-banging PWM.

PWM-i saab käsitsi rakendada mis tahes digitaal väljundis, keerates väljund sisse ja välja soovitud ajaks. Nt:

```
digitalWrite(pin, HIGH);  
delay(1); // ligikaudu 10% dutyCycle  
@1Khz  
digitalWrite(pin, LOW);  
delay(9);
```

Selle meetodi eelis on see, et võib kasutada mis tahes digitaal väljundi pinnis. Lisaks on teil täielik töötsükli ja sageduse kontroll. Üks peamisi puudusi on see, et mis tahes katkestused mõjutavad ajastust. Teine puudus on see, et väljund ei saa muuta oma oleku, kui protsessor teeb midagi muud. Lõpuks on raske kindlaks määrata konkreetse töötsükli ja sageduse jaoks sobivaid konstante, kui te ei loe tsükleid hoolikalt või ostilloskoopi vaadates väärtusi. [8]

ATmega PWM registrite otse kasutamine.

ATmega168P / 328P kiibil on kolm PWM-taimerit, mis kontrollivad 6 PWM-väljundit. Kiibi taimeriregistreid manipuleerides saate rohkem kontrolli kui analogWrite funktsioon pakub.

AVR ATmega328P andmeleht pakub PWM-taimerite üksikasjalikku kirjeldust, kuid paljude erinevate juhtimis- ja väljundrežiimide tõttu võib andmeleht olla raskesti arusaadav.

ATmega328P-l on kolm taimerit 0, Timer 1 ja Timer 2. Igal taimeril on kaks väljundi võrdluse registrit, mis reguleerivad PWM-i laiust ajastaja kahe väljundi puhul: kui taimer jõuab võrdluse registri väärtuse juurde, lülitatakse vastav väljund. Mõlema taimeri kaks väljundit on tavaliselt sama sagedusega, kuid neil võib olla erinev töösükel (sõltuvalt vastavate väljundite võrdluse registrist). Igal taimeril on taktijagur, mis genereerib taimerikella, jagades süsteemi kella taktijaguriga, nagu näiteks 1, 8, 64, 256 või 1024. Arduinol on süsteemi kella 16MHz ja taimerikella sagedus on süsteemi kella sagedus jagatud taktijaguri teguriga. Pange tähele, et taimeril 2 on teistsugused taktijaguri väärtused teistest taimeritest.

PWM peamised režiimid on "Kiire PWM" ja "Phase-correct PWM", mida kirjeldatakse allpool. Taimer võib töötada vahemikus 0 kuni 255 või 0-st fikseeritud väärtusele. (16-bitisel taimeril 1 on täiendav režiim, mis toetavad taimeri väärtusi kuni 16 bitti.) Igat väljundit saab samuti inverteerida. [8]

Taimeri registrid

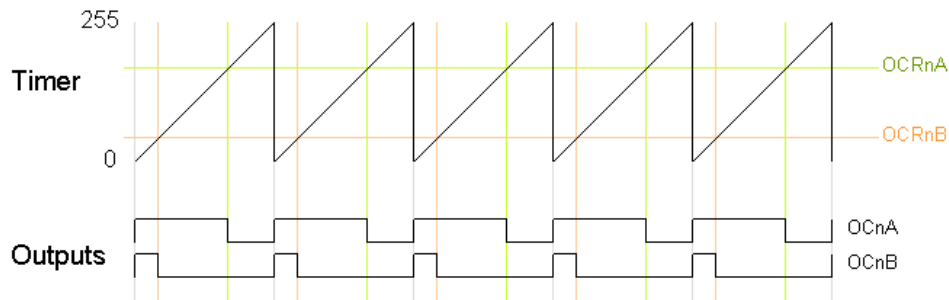
Iga taimeri juhtimiseks kasutatakse mitmeid registreid. Taimer / loendurite kontrollregistrid TCCRnA ja TCCRnB hoiavad taimerile peamisi juhtimisbitte. (Pange tähele, et TCCRnA ja TCCRnB ei vasta väljunditele A ja B.) Need registrid sisaldavad mitmeid bitirühmi:

- WGM bittid : need reguleerivad taimeri üldrežiimi. (Need bitid jagatakse TCCRnA ja TCCRnB vahel.)
- CS bittid: need juhivad kella taktijagurit
- Väljundi võrdle A bitti (COMnA): need lubavad / keelavad / inverteerivad väljundit A
- Väljundi võrdle B bitti (COMnB): need lubavad / keelavad / inverteerivad väljundi B

Väljundi võrdluse registrid OCRnA ja OCRnB määravad tasemed, mis mõjutavad väljundid A ja B. Kui taimeri väärtus vastab registri väärtusele, muudetakse vastavat väljundit režiimi poolt määratud viisil. [8]

Kiire PWM

Lihtsaimas PWM režiimis loeb taimer korduvalt 0-st kuni 255-ni. Väljund lülitub sisse, kui taimer on 0 ja lülitub välja, kui taimer vastab väljundi võrdluse registrile. Mida suurem väärtus väljundite võrdluse registris, seda suurem on töötükkel. Seda režiimi nimetatakse **kiireks PWM-režiimiks**. Joonis 1.2 näitab väljundeid kahe konkreetse OCRnA ja OCRnB väärtuse jaoks. Pange tähele, et mõlemal väljundil on sama sagedus, mis vastab täieliku taimeritsükli sagedusele.



Joonis 1.2 kiire PWM-i väljundi sõltuvus taimerist

Järgmine koodifragment loob kiire PWM-i pinnil 3 ja 11 (Taimer 2). Registri seadete kokkuvõtmiseks valib WGM bitid väärtuseks 011. COM2A-bitide ja COM2B-bitide seadistamine 10-le annab mitte-inverteeritud PWM-i väljunditele A ja B. CS-bittide seadistamine 100-le seab taktijagur jagama kella 64-ni. Väljundi võrdluse registrid on suvaliselt seadistatud 180 ja 50, et kontrollida väljundite A ja B PWM töötükkli.

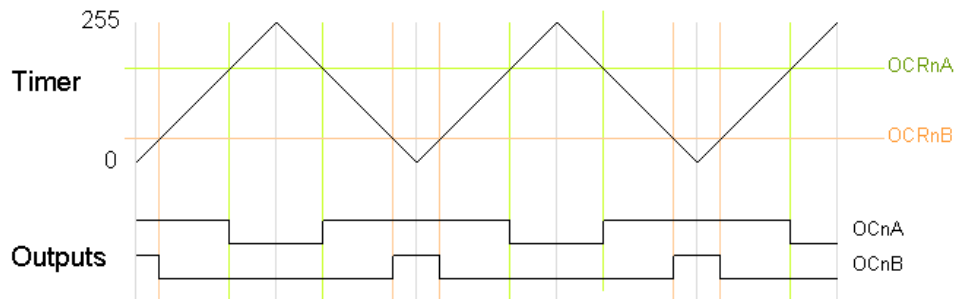
```
pinMode (3, OUTPUT);
pinMode (11, OUTPUT);
TCCR2A = _BV (COM2A1) | _BV (COM2B1) | _BV (WGM21) | _BV (WGM20);
TCCR2B = _BV (CS22);
OCR2A = 180;
OCR2B = 50;
```

Väljundsagedus on 16MHz süsteemikellade sagedus, mis on jagatud prescaler väärtusega (64) jagatud 256 tsükliga, mis kulub taimeri ümbritsemiseks. Pange tähele, et kiire PWM hoiab väljundit ühe tsükli võrra pikemaks kui võrdluse registri väärtus. [8]

Phase-Correct PWM

Selles režiimis loeb taimer 0-st 255-ni ja seejärel tagasi 0-ni. Väljund lülitub välja, kui taimer tabab väljundi võrdluse registri väärtust ülespoole ja lülitub tagasi, kui taimer tabab väljundi võrdluse

registri väärtust tee alla. Tulemuseks on sümmeetriline väljund. Väljundsagedus on umbes pool kiirest PWM-režiimist, sest taimer töötab nii üles kui alla. Joonisel 1.3 on Phase-Correct PWM-i väljundi sõltuvus taimerist.



Joonis 1.3 Phase-Correct PWM

Järgmine koodifragment loob faasikorrektse PWM-i pinnil 3 ja 11 (Taimer 2). WGM bitid seadistatakse faasikorrektse PWM jaoks 001. Teised bitid on samad kui kiirel PWM-il.

```
pinMode (3, OUTPUT);
pinMode (11, OUTPUT);
TCCR2A = _BV (COM2A1) | _BV (COM2B1) | _BV (WGM20);
TCCR2B = _BV (CS22);
OCR2A = 180;
OCR2B = 50;
```

Phase-Correct PWM jagab sageduse kiirega PWM võrreldes kahega, sest taimer läheb nii üles kui alla. Mõnevõrra üllatav on, et sagedus jagatakse 256 asemel 255 ja töötsükli arvutused ei lisa ühtegi kiiret PWM-i. [8]

Kuna C-programmeerimiskeelt kasutatakse kontrolleri programmeerimiseks ja suur hulk teegid lihtsustab seda protsessi, pole mõtet kasutada PWM-tüüpi, mis nõuab registre parameetrite muutmist. Funktsiooni analogWrite kasutamist takistab asjaolu, et impulsside sagedust ei ole võimalik muuta ja funktsiooni kujutatakse teatud sagedusega impulssidena, võib esineda häireid. Sellisel juhul kasutatakse meetodi Bit Banging.

2 NÕUDED JA NENDE PÕHJENDUSED

Kuna antud muundurit kasutatakse 3D trükitud mootorite testimiseks, peaks olema võimalik muuta PWM-sagedust, samuti impulsi laiust.

Impulsside sageduse muutmise võimaldab muuta mootori võlli pöörlemiskiirust, mis on vajalik 3D trükitud mootori parameetrite edasiseks häälestamiseks ja muutmiseks.

Impulsi laiuse muutmise võimaldab kontrollida jõudu, mis pöörleb mootori võlli.

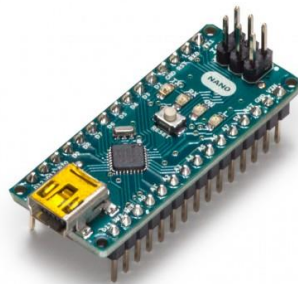
2.1 Erinevad võimalikud valikud

Kõik vajalikud moodulid peavad olema kättesaadavad Eesti turul.

2.1.1 Arduino

Arduino on kättesaadav erineva suurusega: Nano, UNO, Mega 2560 ja paljud muud. Kuna arduinos kasutatakse kiipe, mis võimaldavad samal ajal käivitada ainult ühe programmi, siis on vaja vähemalt kaks arduinot: üks nendest PWM genereerimiseks ja teine mõõtmiseks.

Arduino Nano (joonis 2.1) on mikrokontrollplaat, mis põhineb ATmega328P-I (Arduino Nano 3.x) kiibil. Nano-I on 22 digitaalsisendit (millest 6 saab kasutada PWM-väljunditena), 8 analoogsisendit, 16 MHz kvartsi sagedus, Mini - B USB-ühendust. [9]



Joonis 2.1 Arduino Nano [9]

Arduino Uno (joonis 2.2) on mikrokontrollplaat, mis põhineb ATmega328P kiibil. UNO-I on 14 digitaalsisendit (millest 6 saab kasutada PWM-väljundina), 6 analoogsisendina, 16 MHz kvartsi sagedus, USB-ühendus, toitepistik ja reset-nupp. UNO sisaldab kõike, mis on vajalik mikrokontrolleri toetamiseks, lihtsalt ühendage see USB-kaabli abil arvutiga või käivitage see AC-DC adapteriga või akuga.[10]



Joonis 2.2 Arduino UNO [10]

Arduino Mega 2560 (joonis 2.3) on ATmega2560-l põhinev mikrokontrollplaat. Mega-l on 54 digitaalsisendit (millest 15 saab kasutada PWM väljundina), 16 analoogsisendit, 4 UART, 16 MHz kvartsi sagedus, USB-ühendus, toitepistik reset-nupp. Mega sisaldab kõike, mis on vajalik mikrokontrolleri toetamiseks; lihtsalt ühendage see USB-kaabli abil arvutiga või käivitage AC-DC adapteriga või akuga. [11]



Joonis 2.3 Arduino Mega 2560 [11]

Tabelis 2.1 on toodud erinevate Arduino parameetrid

Tabel 2.1 Arduino Nano/Uno/Mega 2560 tehnilised parameetrid

Nimi	Protsessor	Toitepinge, V	Protsessori kiirus, MHz	Analoog I/O	Digitaal I/O	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
Nano	Atmega 328P	7-9	16	8/0	14/6	1	2	32	Regulaar	1
Uno	Atmega 328P	7-12	16	6/0	14/6	1	2	32	Regulaar	1
Mega 2560	ATmega2560	7-13	16	16/0	54/15	4	8	256	Regulaar	4

2.1.2 HMI valik

HMI on kasutajaliides, mis ühendab inimese masina, süsteemi või seadmega. PWM sageduse ja impulsi laiuse sisestatud parameetrite ning tarbitava voolu ja pinge väljundi kuvamiseks on vaja HMI-i. Sel põhjusel peaks ekraan olema kas suur, mitte vähem kui 2" või võimalus vahetada lehekülgi.

Adafruit FeatherWing 3,5" (joonis 2.4) on TFT-puutetundlik ekraan, sisse ehitatud microSD-kaardi pesaga, millel on 6 valget LED-taustvalgustust ja 480 x 320 pikslit. FeatherWing 3,5-tolline TFT-ekraan pakub 16-bitist värvi ja pakub vastupidavat puutetundlikku ekraani. Resistiivne puutetundlik ekraan suudab ekraanil tuvastada sõrmejälgi. [12]



Joonis 2.4 Adafruit 3.5" ekraan [12]

Nextion Enhanced NX3224K024_011R 2,4" (joonis 2.5) on võimas 2.4-tolline HMI TFT ekraan, millel on 32MB Flash andmesalvestusruum, 1024 baiti EEPROM, 8192 baiti RAM. Nextion sisaldab riistvara osa (TFT-plaatide seeria) ja tarkvara osa (Nextion-Editor). Nextioni TFT-plaat kasutab suhtlemiseks ainult ühte jadaporti. See võimaldab kasutajal vältida juhtmestikku. Nextioni redaktoril on liidese disaini jaoks palju komponente nagu: nupud, tekst, number, liugur, instrumentide paneel jne. [13]



Joonis 2.5 NX3224K024_011R 2.4" [13]

4D Systems gen4-uLCD-43DCT-CLB-AR 4,3 " Diablo16 (joonis 2.6) integreeritud ekraani moodul on osa uusimatest 4D-süsteemide moodulitest. Gen4 seeria on mõeldud spetsiaalselt integreerimise ja kasutamise hõlbustamiseks, võttes hoolikalt arvesse ruumi nõudeid ja funktsionaalsust. Antud moodul on varustatud mahutavuse puuetundlikku 4,3-tollise värvilise TFT LCD-ekraaniga. Neid toidab 4D Systems Diablo16 graafikaprotsessor, mis pakub hulgaliselt funktsioone ja võimalusi igale disainerile, insenerile, kasutajale. [14]



Joonis 2.6 gen4-uLCD-43DCT-CLB-AR [14]

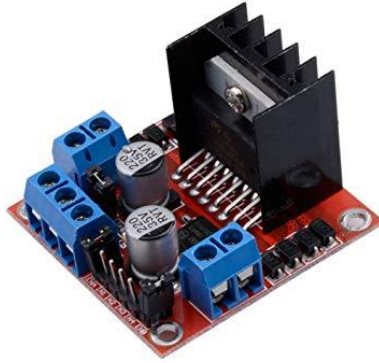
2.1.3 Draiverite valik

Mootorite juhtimiseks on vaja seadet, mis muudab väikese võimsusega juhtsignaalid vooluks, mis on piisav mootorite juhtimiseks. Sellist seadet nimetatakse mootori draiveriks. Need erinevad nii võimsuse kui ka elementide baasist, mille alusel need valmistatakse.

Antud tööde jaoks, on kokkulepitud et, põhilised tehnilised nõuded: kuni 50V ja 5A. Samuti peaks draiver olema üsna väike, selleks et muundur oleks kompaktne.

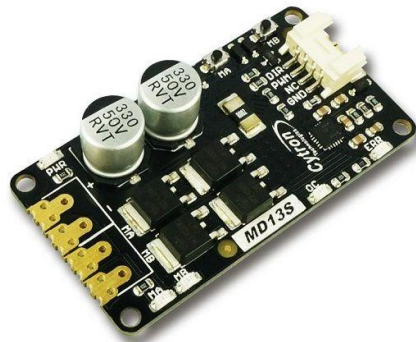
Moodul L298 (joonis 2.7) on integreeritud monoliitne ahel. See on suure voolu täissilladraiver, mis on ettenähtud standardse TTL-loogika taseme ja saab kasutada induktiivse koormusega: releed, solenoidid, alalisvoolu- ja samm-mootorid. L298n mooduliga saab juhtida kaks mootorit. Iga väljundi jaoks on ette nähtud oma loogika sisendid, mis võimaldavad üksteisest sõltumatult juhtida. [15]

Moodulis on paigaldatud stabilisaator 7805, mis on ühendatud + 5 V väljundiga, ning võimaldab moodulit kasutada muunduri teiste komponentide toitmiseks.



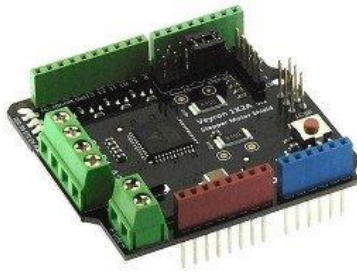
Joonis 2.7 Moodul L298n [15]

MD13S (joonis 2.8) on ette nähtud suure vooluga alalisvoolumootorite juhtimiseks, kuni 13 A. MD13s on täissilla draiver, millega saab juhtida 1 alalisvoolu mootorit. Väljundi juhtimiseks on ettenähtud PWM ja Dir sisendid. Täielik NMOS H-sild tagab parema tõhususe ja jahutusradiaatorit pole vaja. Maksimaalne vool kuni 13A pidev ja 30A piik (10 sekundit). [16]



Joonis 2.8 Moodul MD13s [16]

DFRobot DRI0035 (joonis 2.9) on alalisvoolu mootori draiver. Antud draiveris kasutatakse TMC260 kiipi, mis on kõrge täpsusega mikrolülituse draiver, bipolaarsete samm-mootorite jaoks, millel on integreeritud võimsus MOSFET. TMC260 saab juhtida nii Step & Direction signaalidega kui ka seeria SPI liidesega. Täielik kaitse- ja diagnostikafunktsioonid muudavad selle seadme väga kindlaks. Draiveriga saab juhtida samm-mootoreid kuni 2A ilma jahutuseta. [17]



Joonis 2.9 Moodul DRI0035 [17]

2.1.3 Vooluandurite valik

Iga faasi voolu mõõtmiseks ja mootori poolt tarbitava võimsuse arvutamiseks on vaja vooluandurit. Põhilised tehnilised nõuded: kuni 50V ja 5A

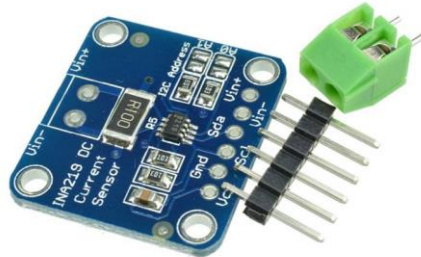
ACS712ELC (joonis 2.10) pakub ökonoomset ja täpset lahendust vahelduvvoolu või alalisvoolu mõõtmiseks. Seade koosneb täpsest, madalalt nihutatud, lineaarsest Hallist andurist. Seadme kaudu voolav vool tekitab tundliku magnetvälja mis teisendatakse proportsionaalseks pingeks. Turul on saadaval ACS712 mis on ettenähtud voolu tugevusele: 5A, 20A, 30A. [18]



Joonis 2.10 Voolu andur ACS712ELC [18]

Voolutugevuse moodul INA219 (joonis 2.11) on tugevvoolu ja võimsuse andur I2C liidesega. INA219 jälgib nii šundi langust kui ka toitepinget, programmeeritava teisendamisaaja ja filtreerimisega. Programmeeritav kalibreerimisväärtus, kombineerituna sisemise kordajaga, võimaldab otseseid näitajaid amprites. Lisa korrutustegur arvutab võimsuse vattides. I2C liides sisaldab 16

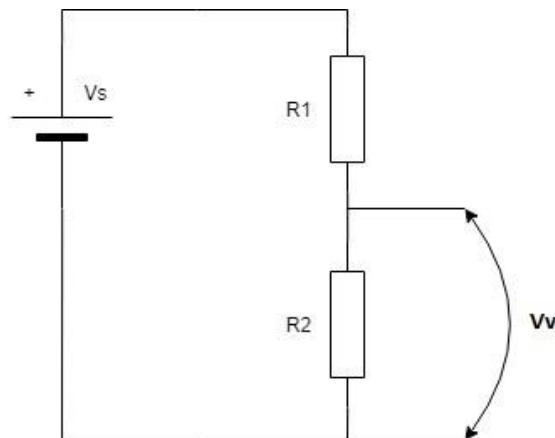
programmeeritavat aadressi. INA219 tunneb pinged vahemikus 0-26V. Kiibi toide on vahemikus +3,3V kuni +5V-ni. Toitevool on maksimum 1mA.[19]



Joonis 2.11 Voolu andur INA219 [19]

2.1.4 Pingeandur

Pingejaotur kasutatakse pingemõõturiks. Joonisel 2.12 on pingajaoturi skeem, kus pinge väljundis (V_v) sõltub takistuse R_1 ja R_2 väärtusest.



Joonis 2.12 Pinge jaguri skeem

Takistite väärtused peaksid olema piisavalt suured, et mitte oluliselt mõjutada voolutarbimist. Kuna Arduino analoogsisendid arvutatakse 5V DC-le, on vaja arvutada takistite väärtused valemiga:

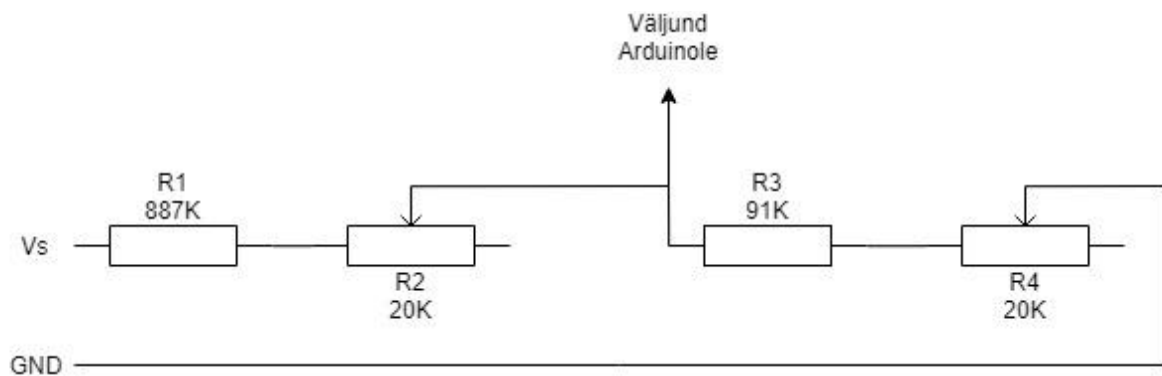
$$V_v = \frac{V_s \cdot R_2}{R_1 + R_2} \quad (2.1)$$

kus V_v – väljundpinge, V,

V_s – sisendpinge, V,

R1 ja R2 on takistuse väärtused, Ohm.

Saame väärtused R1 = 900k ja R2 = 100k. Kuna takistitel on sageli viga, kasutatakse täpsemate väärtuste saamiseks trimmereid. Trimmeri väärtuse muutmisega saavutatakse V_v 5V väärtus. Joonisel 2.13 on antud tööde jaoks vajalik pingeaotur. Potentsiomeetriga R2 ja R4 saab reguleerida pingeaoturi parameetrid.



Joonis 2.13 Pinge jaguri skeem

Tööde jaoks takistite ja trimmerpotentsiomeetrite loend on tabelis 2.2.

Tabel 2.2 Takistuse loend

Skeemi tähistus (joonis 16)	Väärtus, kOhm	tk.
R1	887	5
R2	91	5
R2, R4	20	10

2.2 Seadmete valik ja konstruktsioon

Nüüd, kui on teada erinevad komponendid, koostatakse tabeli ja lõpuks valime vajaliku mooduli

Tabelis 2.3 on välja toodud erinevate HMI-i peamised parameetrid

Tabel 2.3 HMI loend

	Diagonal	Piksliit	Värvid	Toitepinge, V	Ühendamiseks vajalik pinnide arv, tk
Adafruit 3,5" FeatherWing	3,5"	480x320	16-bit	3.3	10
NEXTION NX3224K024_011R	2,4"	320x240	16-bit	5	4
4D Systems gen4-uLCD-43DCT-CLB-AR	4,3"	480x272	16-bit	5	4

Adafruit ei sobi, kuna kasutab liiga palju pinne. Samuti tuleb ka eraldi kokku panna +3.3V toite lati. Nextion sobib, vaid ekraani suurus võiks olla suurem. 4D süsteem sobib ka, aga hind on liiga kõrge. Kokkuvõtte: valitud NEXTION NX4832K035 2,4". Põhjuseks: madal hind, kasutab ainult 4 pinni (Rx, Tx, +5V, GND), Nextion Editor tarkvara.

Tabelis 2.4 on välja toodud erinevate draiverite peamised parameetrid

Tabel 2.4 Draiverite loend

	Tööpinge, V	Vool, A	Väljundite arv
L298n	6-35	2	2
MD13s	6-30	13	1
DRI0035	6-40	2	2

L298n meile sobib, kuna mooduli tehnilised parameetrid vastavad töö nõuetele. Samuti on kõige madalam hind. MD13s sobib ka, aga tuleb osta 4 tükki, kuna ühe draiveriga saab toita ainult ühe faasi. DRI0035 sobib, aga see on kõige kallim moodul.

Kokkuvõtte: valitud L298n moodul. Põhjuseks: madal hind, üsna väike, tööpinge ja vool vastavad töö nõuetele.

Tabelis 2.5 on välja toodud erinevate voolu andurite parameetrid

Tabel 2.5 Voolu andurite loend

	Nimivool, A	Nimipinge, V	Väljund
ACS712ELC	5/20/30	2400	Pinge kuni 5V
INA219	3,2	26	I2C

ACS712 sobib, kuna moodulite parameetrid vastavad töö nõuetele. INA219 sobib, aga kuna ta kasutab I2C liidest raskendab programmeerimist.

Kokkuvõtte: valitud ACS712.

Arduino valimiseks on vaja teada I/O arv. Tabelis L2.1 on vajalik ühendused Arduino ja andurite vahel. Tabelis 2.6 on analoog ja digitaalse I/O arv vajalik tööde jaoks

Tabel 2.6 Analoog ja digitaalsed I/O

	Analog I/O	Digital I/O
Arduino 1	13	15
Arduino 2	3	17

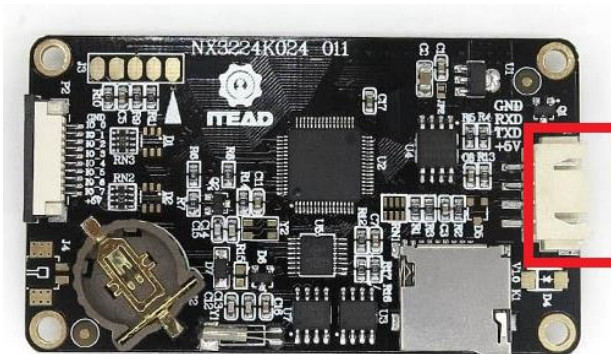
Kokkuvõtte, et sobib ainult Arduino Mega 2560.

2.3 Valitud seadmete konstruktsioon

NEXTION NX3224K024_011R 2,4"

Mooduli mõõdud: 74,4 mm x 42,72mm x 5,8mm

Ekraani pind: 48,96 mm x 36,72 mm

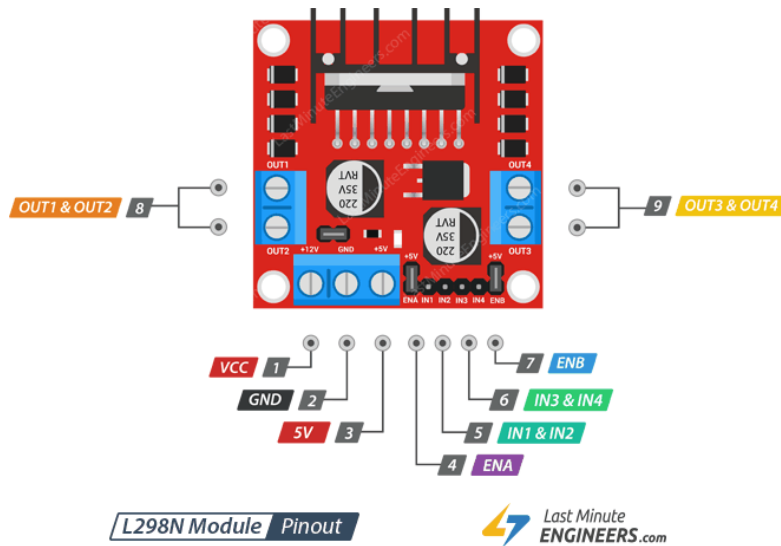


Joonis 2.14 GND - maandus, RXD - võtab vastu info Arduino-lt, TXD - saadab info Arduino-le, +5V - 5V toite pinge

Joonisel 2.14 on Nextion HMI tagakül. Punases ruudus on ühenduseks vajalikud pinnid. Kinnitamiseks kasutatakse M3 polte ja mutreid.

Moodul L298n

Mooduli mõõdud: 43 x 43 x 26mm. Kinnitamiseks kasutatakse M3 polte ja mutreid



Joonis 2.15 Moodul L298n

Joonisel 2.15 on välja toodud L298n-i juhtimis ja jõu osa pinnid ja pistikud:

- VCC on mootorite toitepinge. Peab olema vahemikus 6-35V DC
- GND on maandus pin
- 5V on L298n kiibi loogika elementide toide. Kui 5V jumper on paigas, antud pinni saab kasutada teiste moodulite toitmiseks
- ENA pin kasutatakse mootori A juhtimiseks. Kui antud pinnil on kõrge signaal ja PWM on edastatud IN1 või IN2 pinnile, rootor pöörleb. Kui signaal on madal, rootor ei pöörle
- IN1 & IN2 pinne kasutatakse pöörlemis suuna valimiseks (mootor A). Kui ühel nendest on kõrge signaal ja teisel pinnil on madal signaal, rootor pöörleb ühes suunas. Kui on teistmoodi, siis mootor pöörleb teises suunas. Kui mõlemad signaalid on kõrged või madalad, siis rootor ei pöörle.
- IN3 & IN4 pinne kasutatakse pöörlemissuuna valimiseks (mootor B). Kui ühel nendest on kõrge signaal ja teisel pinnil on madal signaal, rootor pöörleb ühes suunas. Kui on teistmoodi, siis mootor pöörleb teises suunas. Kui mõlemad signaalid on kõrged või madalad, siis rootor ei pöörle.
- ENB pinni kasutatakse mootori B juhtimiseks. Kui antud pinnil on kõrge signaal ja PWM on edastatud IN1 või IN2 pinnile, siis rootor pöörleb. Kui signaal on madal sellisel juhul rootor ei pöörle.
- OUT1 & OUT2 mootori A väljundid mähise ühendamiseks.
- OUT3 & OUT4 mootori B väljundid mähise ühendamiseks.

ACS712ELC

Mooduli mõõdud: 31 (mm) x13 (mm)

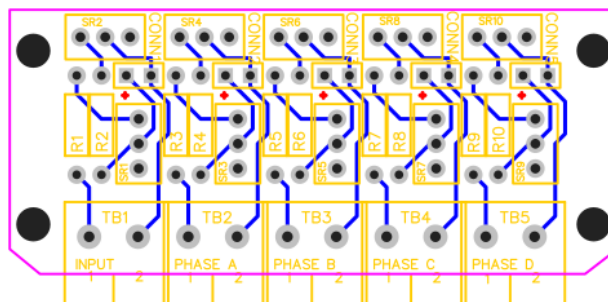
ACS712ELC kinnitamiseks kasutatakse M3 polte ja mutreid.

Joonisel 2.10 on välja toodud ühendamiseks vajalikud pinnid:

- VCC on kiipi toidepinge 5V
- GND on maandusklemm.
- OUT on mõõdetud voolu väljund

Pinge jaotur

Trükkplaadi koostamiseks kasutatakse interneti lehekülge www.easyeda.com. Trükkplaadi mõõdud: 62,23 mm x 27,05 mm. Trükkplaadi valmistamiseks kasutatakse CNC freespink ProtoMat S63. Pingejaoturi kinnitamiseks kasutatakse M3 polte ja mutreid.



Joonis 2.16 Pinge jaoturi trükkplaat, vaade ülevalt

Joonisel 2.16 on pingeaoturi trükkplaat, kus:

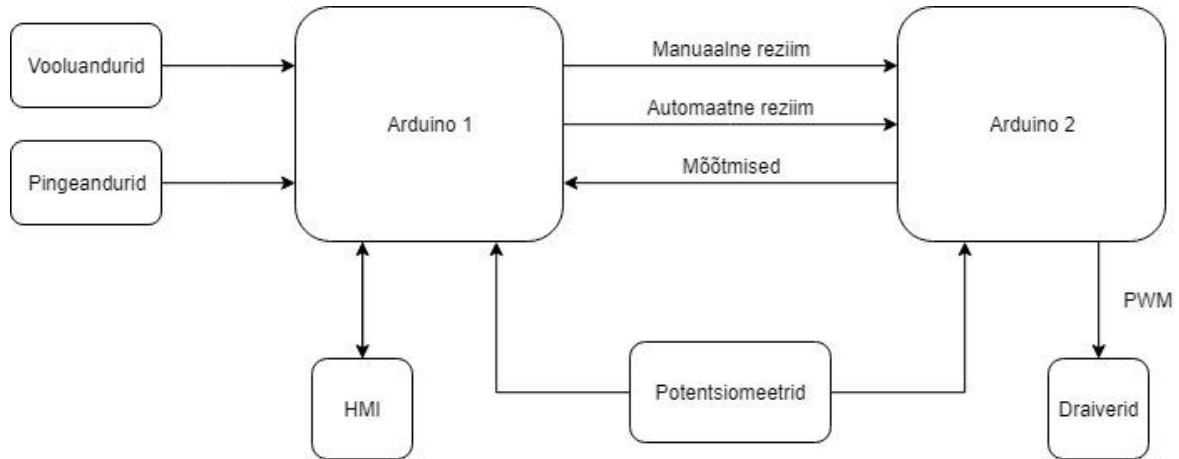
- TB1 – TB5 on mõõdetava pingeaoturi ühendusklemmid (1 on positiivne pinge, 2 on GND)
- SR1 – SR10 trimmer potentsioomeetrid
- R1-R10 – resistorid
- “+” on väljund Arduino-le

Arduino Mega 2560

Mõõdud: 101,52(mm) x 53,3(mm). Arduino Mega 2560 pinnide numbrid on välja toodud lisas L2.2. Arduino toitmiseks saab ühendada +5V ja GND ükskõik millised pinnid Arduino plaadil. See tähendab, et ei ole vaja ehitada lisa toiteploki. Arduino kinnitamiseks kasutatakse M2.5 polte ja mutreid.

3 PROTOTÜÜBI KOKKUPANEK

Alustuseks koostatakse juhtsignaali skeem.

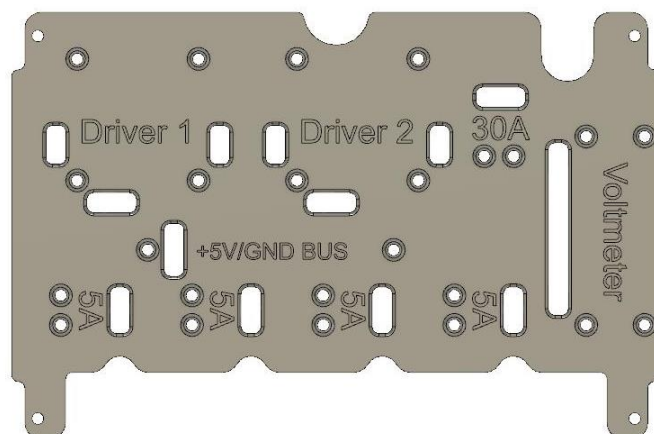


Joonis 3.1 Juhtsignaali skeem

Joonisel 3.1 on juhtsignaal skeem, kus pinge ei ületa +5V. Arduino 1 saab signaalid voolu- ja pingeandurist, samuti ka mõõtmise jaoks vajalikud signaalid Arduino 2-lt. Juhtimiseks vajalikud parameetrid saavad mõlemad Arduinod potentsiomeetrist. Arduino 2 saab juhtimissignaale Arduino 1-lt ja edastab PWM signaale draiverile.

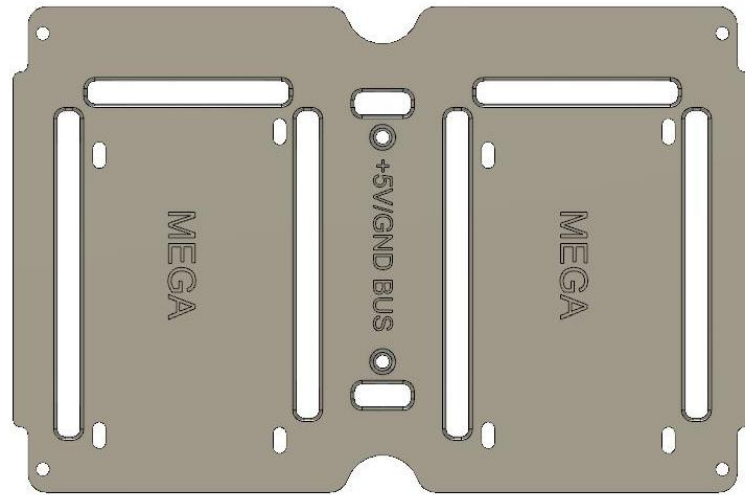
Muunduri kokkupanemiseks on vajalik: kere, pistikupesad ja potentsiomeetrid. Kere on valitud mõõduga: 220mm x 145mm x 75mm, kuna muundur peab olema kompaktne ja jagatakse muundur kaheks tasandiks: ülemine tasandil - 2 Arduino Mega, alumine tasandil - moodulid L298n, vooluandurid, pinge jaotur, ja kere kaanes on potentsiomeetrid ja ekraan.

Iga mooduli jaoks on selle spetsiaalselt määratud koht. Selleks koostan Autodesk Fusion360 3D-mudeli. Kõik mudelid on trükitud 3D-printeriga, kus on näha, kuhu ja millist moodulit paigaldada.



Joonis 3.2 Muunduri alumine tasand

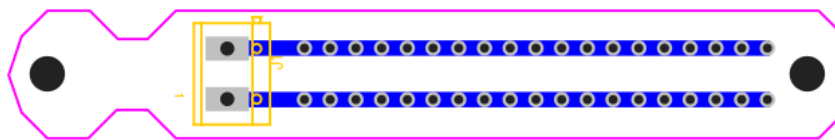
Joonisel 3.2 on alumine tasandi mudel: siia paigaldatakse mõlemad L298n draiverid (Driver 1 ja Driver 2), voolutugevuse andurid (5A ja 30A), pingejaotur (Pingeandur) ja +5V/GND latt (+5V/GND BUS). Kõik juhtmed paigaldatakse mudeli alla. Selleks mudelis on ettenähtud rudulised augud.



Joonis 3.3 Muunduri ülemine tasand

Joonisel 3.3 on ülemine tasandi mudel. Siia paigaldatakse mõlemad Arduinod (Mega) ja +5V/GND latti.

Taseme eraldamiseks kasutatakse pikendajat M3.

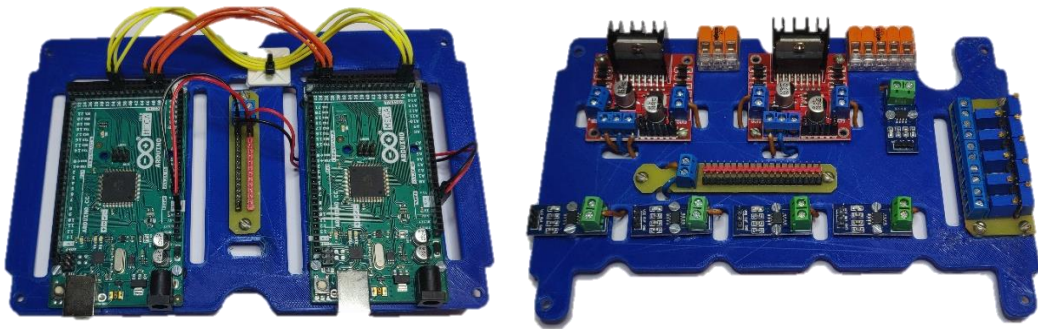


Joonis 3.4 +5V/GND latti trükkplaadi skeem, ülemine rada on +5V, allmine rada GND

Joonisel 3.4 on +5V/GND latti trükkplaadi skeem. +5V/GND latti valmistamiseks kasutatakse CNC freespink ProtoMat S63. Antud latti kasutatakse mooduli toitepinge ühendamiseks. Latti saab pinge L298n moodulist, +5V väljundist. GND on ühendatud muunduri GND-ga

Moodulite kinnitamiseks trükitud mudeliga kasutatakse polte ja mutreid: M3 ja M2.5.

Potentsiomeetrid on valitud nimiväärtusega 10 kOm. Kokku on vaja 3 potentsiomeetrit: PWM, sagedus, ramp. Joonisel 3.5 on moodulid paigaldatud ettenähtud kohas.



Joonis 3.5 Muunduri alumine tasand (paremal) ja ülemine tasand (vasakul)

Nüüd, kui kõik komponendid on paigas, saame moodulid kergesti ühendada. Muunduri ühendus skeem on L3.1.

Kere külgsel peal paiknevad iga faasi väljundid ja toite sisendid, L3.2, L3.3, L3.4.

Muundur on jagatud kaheks osaks: tugevvoolu osa ja juhtimise osa. Summaarne voolutarve muunduri oote režiimis mõõdetakse ampermeetriga ja on 0,561A.

Kuna maksimaalne voolutarve on umbes 9A (seadmete voolutarve ooterežiimis pluss nelifaasid 2A iga) siis jõu osa kaabli pindala on 1mm^2 . [20]

Juhtseadme ühendused on tehtud $0,2\text{ mm}^2$ juhtmega. Kuna kõigil moodulitel on väljundiks ettenähtud pinnid, nende ühendamiseks teiste komponentidega, kasutatakse 3 tüüpi pistikuid: isa / isa, isa / ema, ema / ema.

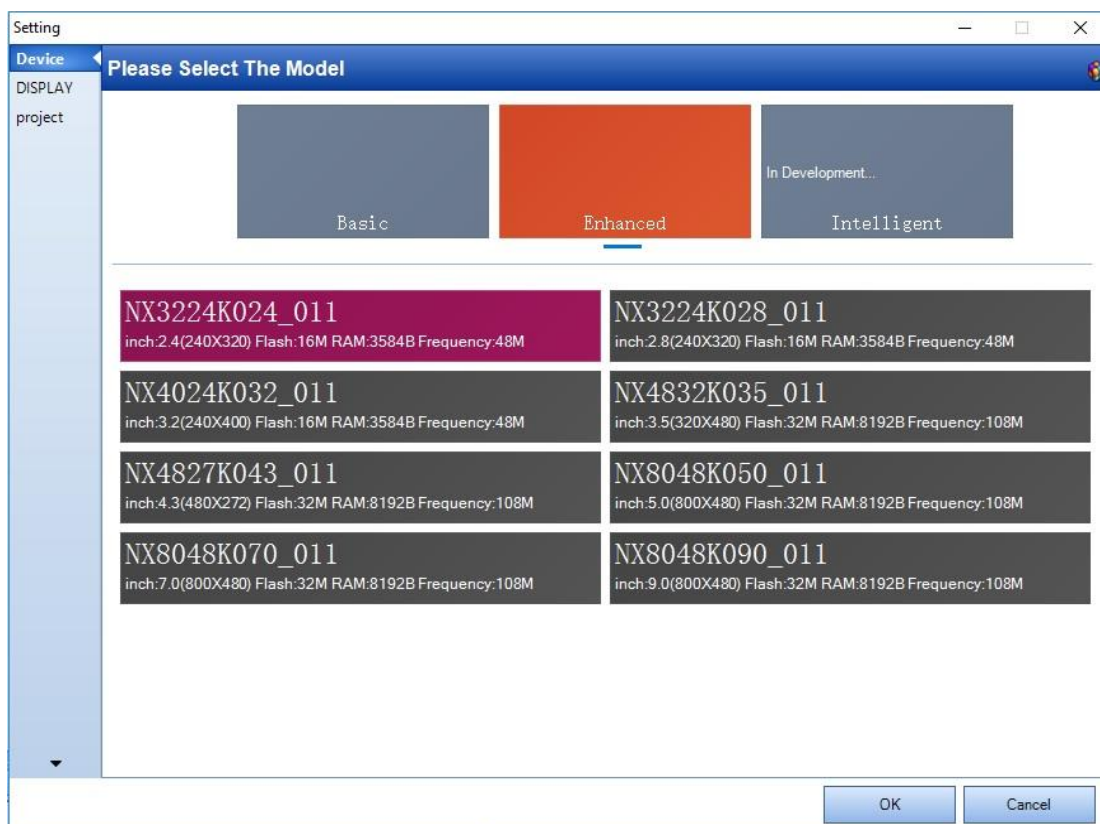
4 PROTOTÜÜBI PROGRAMMEERIMINE JA SEADISTAMINE

Programm on kirjutatud C programmeerimis keele loogikaga, kasutades Arduino Ide tarkvara. Kood laaditakse Arduinole USB-kaabli kaudu. Ekraani programmeerimiseks kasutatakse tarkvara Nextion Editor. Programmi alla laadimiseks ekraani kontrollerrisse on võimalik kasutada kahte varianti:

- Laadida valmis programm tühja microSD-kaardile kuni 8 GB. Lülitage ekraani toitepinge välja, sisesta kaart, lülitage ekraan sisse, oodake, kuni programm on laetud, lülitage toide välja, eemaldage kaart.
- Kasutage USB-kaablit, kuid selleks tuleb kasutada täiendavat TTL-adapterit, mis põhineb näiteks PL2303 mikrokontrolleril

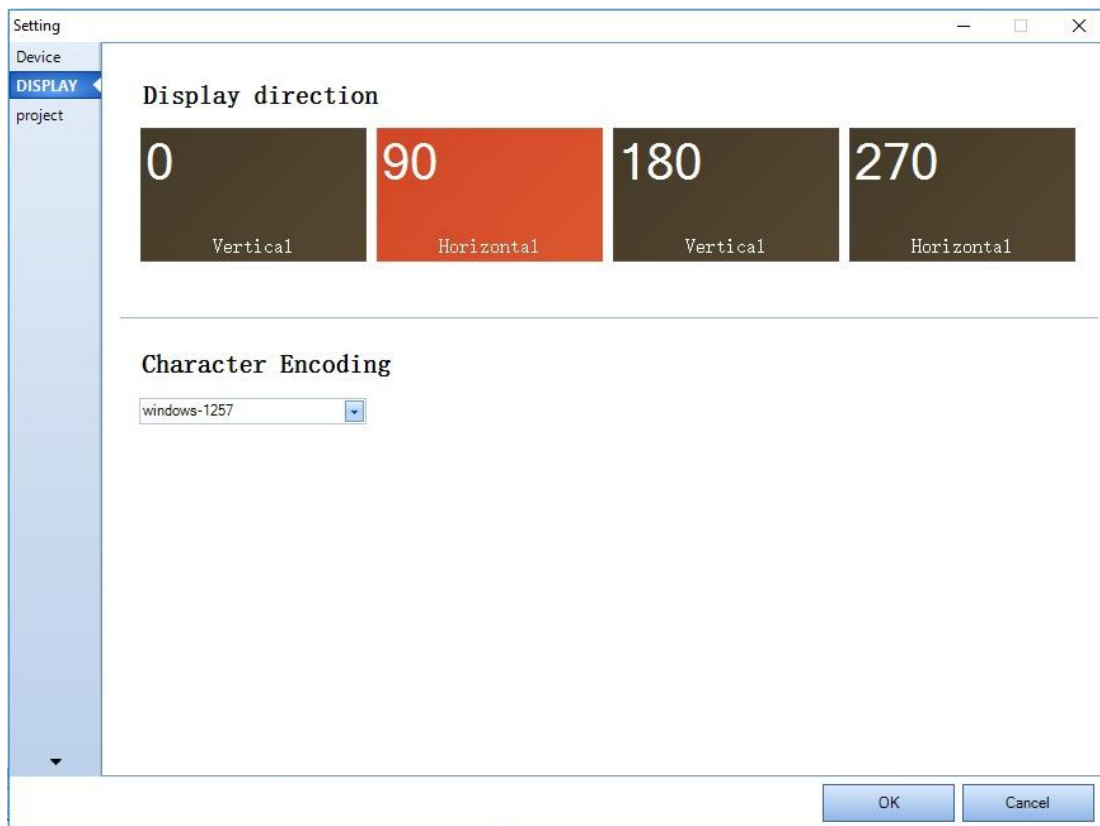
4.1 HMI Programmeerimine

Esiteks programmeerime ekraani. Selleks, Nextion Editor-is, tuleb luua uus projekt File> New. Valitakse kuvatavas aknas projekti nimi ja vajutatakse nuppu Save. Kuvatavas aknas tuleb valida vajalik ekraanimudel, joonis 4.1 (antud juhul NX3224K024 Enchanted).



Joonis 4.1 Ekraanide valik

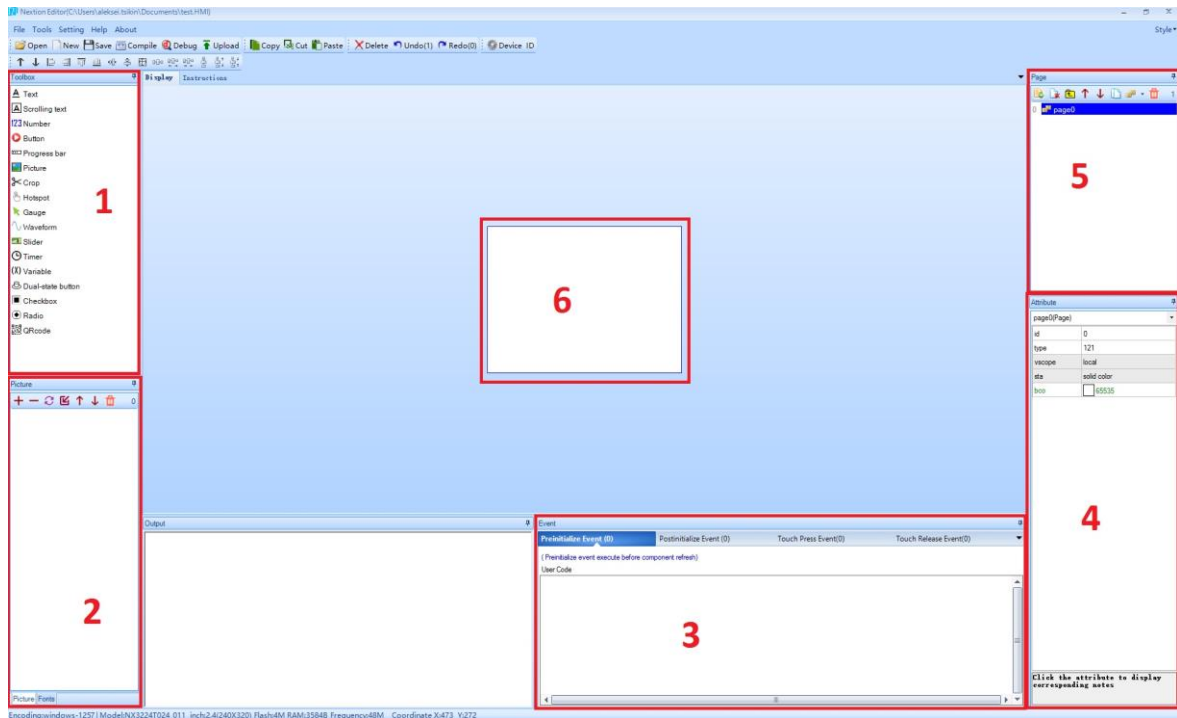
Jaotises DISPLAY tuleb valida kuvatud liidese positsioon, joonis 4.2.



Joonis 4.2 Ekraanide positsioon

Kuna NektionEditor ei ole väga kasutajasõbralik, vaatleme programmi põhielemente (joonis 4.3):

- 1 – Toolbox, siit saab valida vajalikud elemendid, programmi koostamiseks (nupud, pildid, tekstid ja muud)
- 2 - Pictures. Kõik selle projektiga seotud pildid asuvad siin. Pildi loomiseks, selles aknas tuleb vajutada „+” ja valida soovitud pilt.
- 3 – Event. Sündmuste aken, kus saab antud elemendi jaoks, koostada vajalikke tegevusi.
- 4 – Attribute. Selles aknas määratakse iga asukoha elemendi atribuudid.
- 5 – Page. Antud aknas on programmiga seotud leheküljed. Uue lehekülje loomiseks vajutage „+” nuppu.
- 6 – ekraani töötsoon

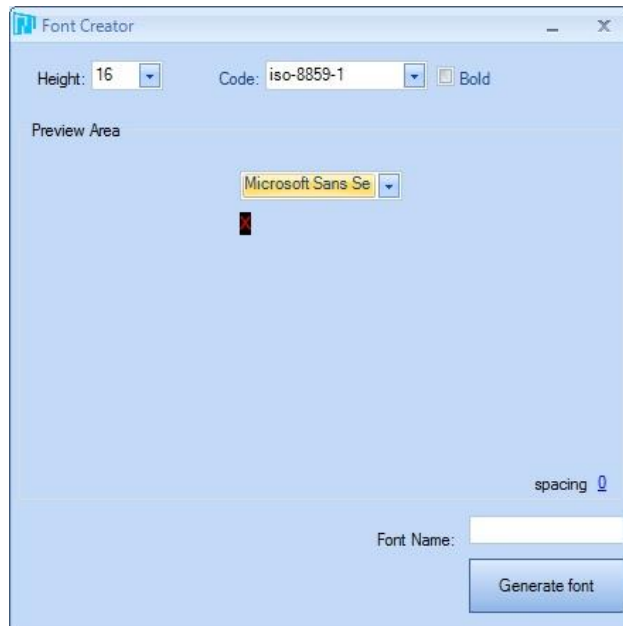


Joonis 4.3 Nextion Editori liides

Kuna ekraanis ei mahu kõik vajalikud elemendid, luuakse 5 lehekülge:

- Avaleht
- Manuaalne režiim
- Auto režiim
- Häälestus
- Mõõtmised

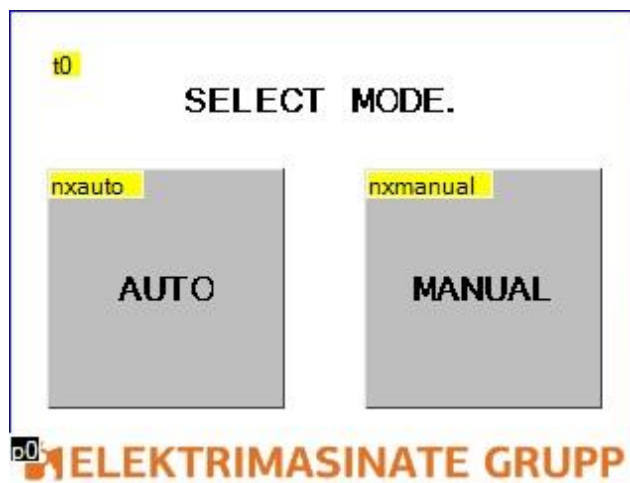
Samuti on vaja luua ekraanile fonte (joonis 4.4). Selleks kasutatakse Tools> Font Generator. Ilmuvas aknas tuleb määrata fondi kõrguse (Height), valida loendist fonte, sisesta nime (Font Name) ja salvesta, mugavuse huvides, programmi kausta. Selles töös on kasutatud Microsoft Sans Sheriffi font, kõrgus 24 ja 36.



Joonis 4.4 Font creator

Kuna UART kasutatakse ekraani Arduino-ga suhtlemiseks, iga ekraani elemendi jaoks on vaja määrata selle unikaalne nimi.

Avaleht. Režiimi valik.



Joonis 4.5 Avaleht

Avaleht numbriga 0 (page0, joonis 4.5). Antud leheküljel on 4 elementi:

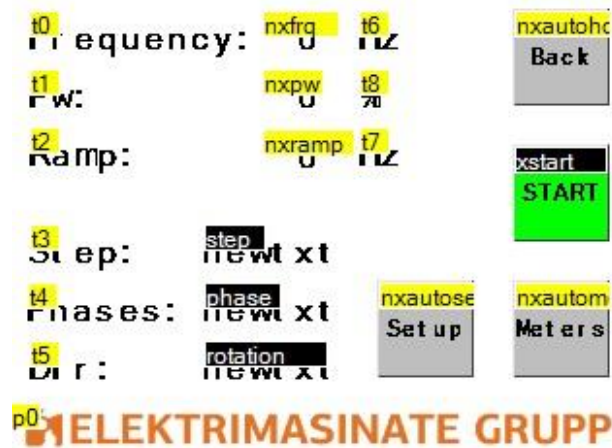
- Tekst "Please select mode"
- Nupud Auto ja Manual
- Pilt

Tabelis 4.1 on antud lehekülje elementide atribuudid ja sündmused. Kui midagi ei ole tabelis määratud, pole vaja antud komponendi jaoks midagi muuta.

Tabel 4.1 Avalehe komponentide atribuudid

Element	Sündmuse aken	Atribuudid	
	Send Component ID	objname	txt
Tekst		t0	Select mode
Nupp	✓	nxauto	AUTO
Nupp	✓	nxmanual	MANUAL

Automaatne režiim



Joonis 4.6 Auto Mode lehekülg

Lehekülg number 1, automaatne režiim (joonis 4.6). Leheküljel on 20 elementi:

- Tekstid "Frequency", "Hz", "PW", "%", "Ramp", "Hz", "Step", "Phases", "Dir"
- 3 numbrit (nxfrq, nxpw, nxramp)
- Nupud Back, Setup, Meters
- Kahe olekuga nupp START/STOP
- Pilt

Tabelis 4.2 on antud lehekülje elementide atribuudid ja sündmused. Kui midagi ei ole tabelis määratud, pole vaja antud komponendi jaoks midagi muuta.

Tabel 4.2 Automaatse režiimi lehekülje elementide atribuudid

Element	Sündmuste Aken		Atribuudid				
	Touch Press Event (Send Component ID)	Preinitialize Event	objname	vscope	bco0	bco1	txt
Lehekülg 1		Kood 1					
Tekst			t0	local			Frequency
Tekst			t1	local			PW
Tekst			t2	local			Ramp
Tekst			t6	local			Hz
Tekst			t8	local			%
Tekst			t7	local			Hz
Tekst			t3	local			Step
Tekst			t4	local			Phases
Tekst			t5	local			Dir
Tekst			Step	global			
Tekst			Phases	global			
Tekst			rotation	global			
Nupp			nxautohome	local			Back
Nupp	✓		nxautosetup	local			Setup
Nupp	✓		nxautometers	local			Meters
Kahe olekuga nupp	✓, Kood 2		xstart	global	2016	63488	START
Number			nxfrq	local			0
Number			npxpw	local			0
Number			nrxramp	local			0

Kood 1

```
if(xstart.txt=="START")
{
vis 1,1
}else
{
vis 1,0
}
step.txt=page4.step.txt
phase.txt=page4.phase.txt
rotation.txt=page4.rotation.txt
```

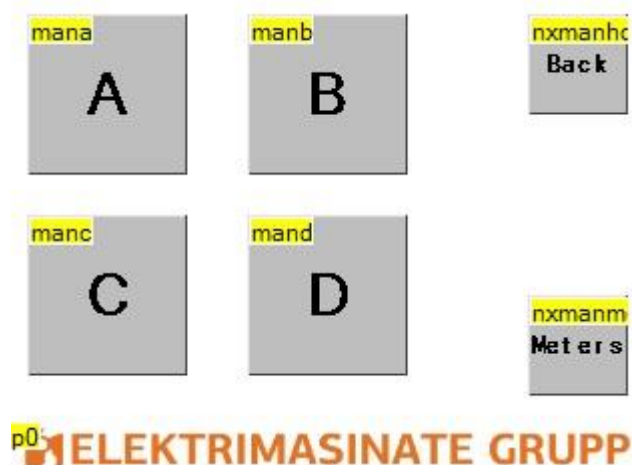
Kood 1 kasutatakse selleks, et igakord, kui antud lehekülg initsialiseerib, ta kontrollib nuppu „START/STOP“ olekut ja kopeerib teksti „Setup“ leheküljest.

Kood 2

```
if(xstart.txt=="START")
{
xstart.txt="STOP"
vis 1,0
}else
{
xstart.txt="START"
vis 1,1
}
```

Kood 2 on ettenähtud selleks, et kui vajutakse „START“ nuppu, ta muudab oma oleku. Värv muutub punaseks, paiknev tekst muutub „STOP“-le ja vastupidi. Samuti, eemaldatakse ekraanist „Back“ nupp, sest enne väljumist, mootor peab peatuma. Loogika on selline: Kui tekst antud nupul on „START“, „Back“ nuppu on näha, kui tekst on „STOP“, „Back“ nuppu ei ole näha,

Manuaalne reziim



Joonis 4.7 Manuaalne reziim lehekülg

Lehekülg number 2 (Joonis 4.7). Leheküljel on 7 elementi:

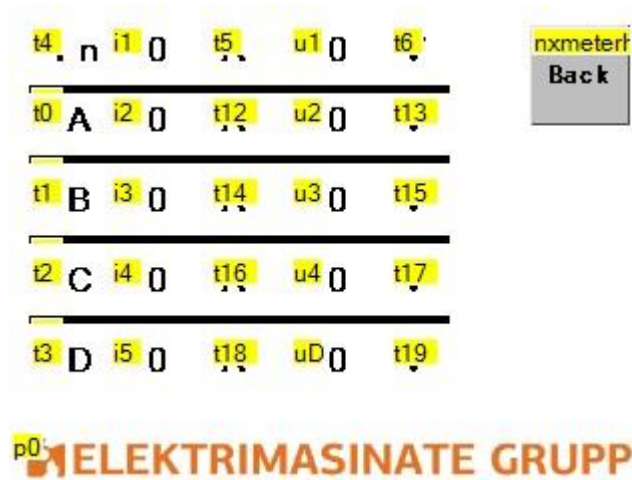
- 2 nuppu (Back ja Meters)
- 4 kahe olekuga nupud (A, B, C, D)
- pilt

Tabelis 4.3 on antud lehekülje elementide atribuudid ja sündmused. Kui midagi ei ole tabelis määratud, pole vaja antud komponendi jaoks midagi muuta.

Tabel 4.3 Manuaalne režiimi lehekülje elementide atribuudid

Element	Sündmuste aken	Atribuudid	
	Send Component ID	objname	txt
Nupp	✓	nxmanhome	Back
Nupp	✓	nxmanmeters	Meters
Kahe olekuga nupp	✓	mana	A
Kahe olekuga nupp	✓	manb	B
Kahe olekuga nupp	✓	manc	C
Kahe olekuga nupp	✓	mand	D

Mõõtmised



Joonis 4.8 Mõõtmise lehekülj

Lehekülj number 3 (Joonis 4.8). Leheküljel on 27 elementi:

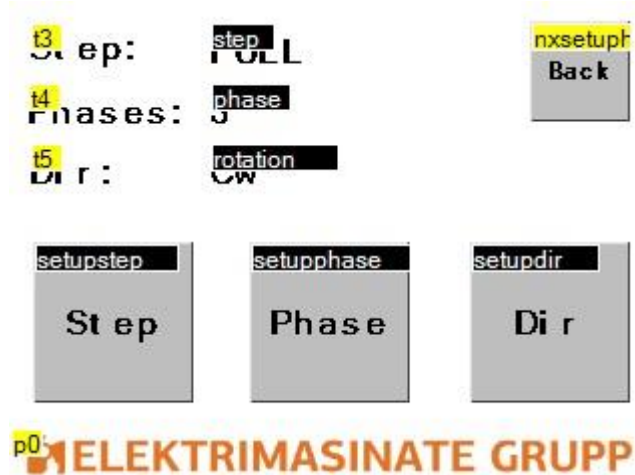
- 15 teksti
- 10 numbrit
- 1 nupp
- 1 pilt

Tabelis 4.4 on antud lehekülje elementide atribuudid ja sündmused. Kui midagi ei ole tabelis määratud, pole vaja antud komponendi jaoks midagi muuta.

Tabel 4.4 Mõõtmise lehekülje elementide atribuudid ja sündmused

Element	Atribuudid		Element	Sündmuste aken	Atribuudid	
	objname	txt		Send Component ID	objname	txt
Tekst	t1	In	Number		i1	0
Tekst	t2	A	Number		i2	0
Tekst	t3	B	Number		i3	0
Tekst	t4	C	Number		i4	0
Tekst	t5	D	Number		i5	0
Tekst	t6	A	Number		u1	0
Tekst	t7	A	Number		u2	0
Tekst	t8	A	Number		u3	0
Tekst	t9	A	Number		u4	0
Tekst	t10	A	Number		u5	0
Tekst	t11	V	Nupp	✓	nxmeterhome	Back
Tekst	t12	V				
Tekst	t13	V				
Tekst	t14	V				
Tekst	t15	V				

Häälestus lehekülg



Joonis 4.9 Setup lehekülg

Lehekülg number 4 (joonis 4.9) kasutatakse muunduri häälestuseks. Siin saab valida: samm (täis/pool), faaside arv ja pöörlemissuunda. Leheküljel on:

- 4 Nuppu: Back, Step, Phases, Dir
- 6 Teksti: Step, Phases, Dir, FULL/HALF, CW/CCW
- Pilt

Tabelis 4.5 on antud lehekülje elementide atribuudid ja sündmused. Kui midagi ei ole tabelis määratud, pole vaja antud komponendi jaoks midagi muuta.

Tabel 4.5 Häälestuse lehekülje elementide atribuudid ja sündmused

Element	Sündmuse Aken		Atribuudid		
	Touch Press Event (Send Component ID)	Preinitialize Event	objname	vscope	txt
Lehekülg		Kood 3		local	
Tekst			t3	local	Step:
Tekst			t4	local	Phases:
Tekst			t5	local	Dir:
Tekst			step	global	FULL
Tekst			phases	global	3
Tekst			rotation	global	CW
Nupp	✓		setupstep	global	Step
Nupp	✓		setupphase	global	Phases
Nupp	✓		setupdir	global	Dir
Nupp	✓		nxsetupback	local	Back

Kood 3

```
if(page1.xstart.txt=="STOP")
{
vis 9,0
}else
{
vis 9,1
}
```

Kood 3 on ettenähtud selleks, et kui mootor töötab, siis ei saa valida faaside arvu, saab ainult pöörlemis suunda ja sammude valida.

4.2 Mikrokontrollerite programmeerimine.

Kuna Nextion.h teeki on vajalik ekraaniga töötamiseks, tuleb see lisada. Seda saab teha, klõpsates Arduino IDE-s Sketch> Include Library> Add Zip library ja valides alla laaditud nextion.zip teeki. Seejärel tuleb selle teeki koodi lisada käsuga #include <Nextion.h>. Samuti on vaja muuta teekides oleva faili „NexConfig.h“. Selleks tuleb leida antud fail arvutis, avada, leida lause #define DEBUG_SERIAL_ENABLE ja kommenteerida seda, // #define DEBUG_SERIAL_ENABLE“. Samas failis, tuleb leida lause „#define nexSerial Serial2“ ja kustutada „2“, et saaks „#define nexSerial Serial“ Järgmiseks tuleb määrata kõik muutujad, samuti kõik nupud, mille kohta tahame infot ekraanilt saada (selleks määratakse Send Component ID Nextion Editor-is). Näide:

```
NexButton Button = NexButton(0, 1, "b1");
```

NexButton – Objekti nimi, antud juhul nupp

Button – objekti nimi, mis me tahame kasutada Arduino IDE-s

NexButton(0, 1, "b1"); - Objekti tüüp(lehekülg, kus antud element asub, elementi ID (atribuutide aken), objname (atribuutide aken)

Seejärel on vaja luua loend, kus paiknevad kõik vajalik nupud.

```
NexTouch *nex_listen_list[] =
{&Button, //Nupp on lisatud
&Button1, //Nupp on lisatud
&Button2, //Nupp on lisatud
NULL // string terminated
};
```

Seejärel tuleb iga nupuvajutus sündmuse jaoks funktsioone luua.

```
void ButtonPushCallback(void *ptr){
  teemidagi();
}
```

Kuna ekraani ja Arduino vahel suhtlemise jaoks on kasutatud UART-i, siis Arduino IDE-s, setup() funktsioonides on vaja käivitada Serial() protokoll kiirusega 9600 baud. Samuti on vaja määrata digitaalsete pinnide tüüp (INPUT, OUTPUT). Samuti on vaja lisada kõik sündmused uuesti.

```
void setup(){
  Serial.begin(9600);
  pinMode(pin, OUTPUT);
  Button.attachPush(ButtonPushCallback);
}
```

Nüüd peate Arduino loop () funktsioonis kirjutama, nexLoop (nex_listen_list);, mis „kuulab“ kõik elemendid, mis on määratud NexTouch *nex_listen_list[] loendis

Manuaalne juhtimine

Manuaalne juhtimine toimub järgmises järjekorras. Kui kasutaja vajutab A, B, C või D nuppu, siis, Arduino 1, edastab signaali Arduinole 2, mis omakorda lülitab sisse väljundi, mis on ühendatud draiverite IN pinniga. Faasi väljalülitamiseks tuleb vajutada sama nupp. Nupud on kahe olekuga, mis tähendab, et on näha mis faas on sisse/välja lülitatud.

Automaatne režiim

Tabel 4.6 Juht signaali

	Juhtimissignaali	
	KÕRG	MADAL
Pöörlemissuund	CW	CCW
Faaside arv	4	3
Samm	Pool	Täis
Režiim	Auto	Manual
START/STOP	START	STOP

Tabelis 4.6 on toodud juht signaalid, mis on vajalikud mootori juhtimiseks. Signaalid tulevad Arduino 1-st Arduino 2.

Draiverite juhtimiseks on vaja teha kolm sammu:

1. Andmete maha lugemine: sagedus, pulsi laius, ramp
2. Arvutada kõik vajalikud viivitused
3. Edastada juhtimisimpulsid Arduino 2-lt draiverile.

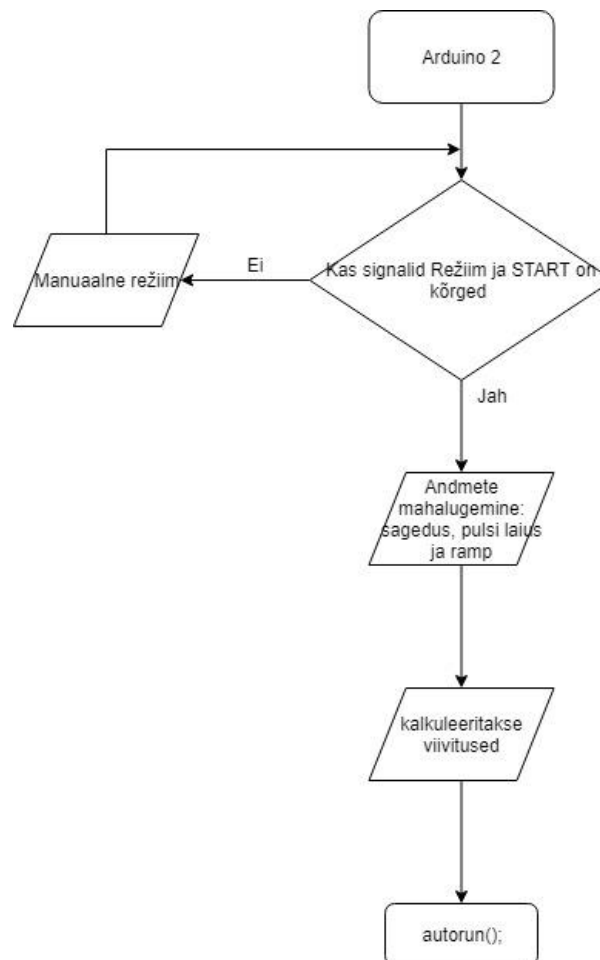
Samuti Arduino-lt 1 on vaja edastada Arduino-le 2 kõik vajalikud signaalid häälestuse kohta: faaside arv, samm, pöörlemis suunda ja start/stop.

Vajalikud on ka funktsioonid Arduino 2-s, mis käivitatakse ainult siis, kui 2 tingimust on täidetud:

1. Auto režiim on valitud
2. START on vajutatud

Meie juhul, see on programmi põhi funktsioon loop(). Programmi algoritm on toodud joonisel 4.10.

```
void loop() {  
  if (digitalRead(reziim) == HIGH && digitalRead(start) == HIGH) {  
    readdata();  
    calcdelays();  
    autorun();  
  } else  
    Manual();  
}
```



Joonis 4.10 Arduino 2 juhtimis loogika

Funktsioon readdata() loeb sisse kõik vajalikud parameetrid potentsiomeetrist. Funktsioon calcdelays() arvutab kõik viivitused ja funktsioon autorun() saadab juhtimis impulsid draiverile. Vaatleme funktsiooni calcdelays () Kõigepealt tuleb arvutada kogu perioodi pikkus millisekundites, sest selleks kasutame valemit:

$$T = \frac{1}{f} * 1000 \quad (2.1)$$

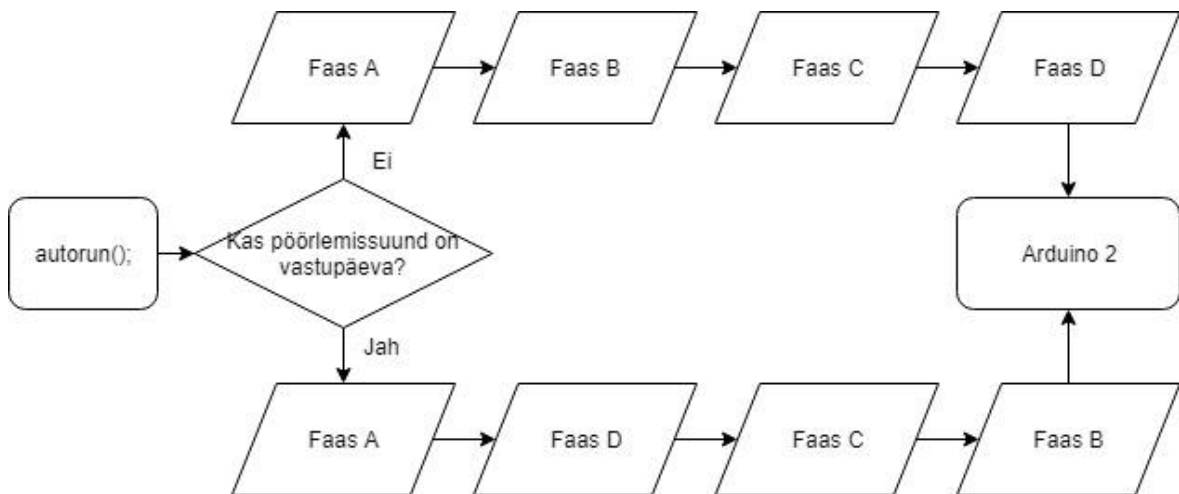
kus T – periood, ms
f – sagedus, hz

Nüüd arvutatakse iga impulsi pikkuse. Seda saab teha, korrutades perioodi pikkuse impulsi laiusega protsentides.

$$PWtime = T * \frac{PW}{100} \quad (2.2)$$

kus PWtime – ühe impulsi pikkus, ms
PW – andmed potentsiomeetrist, %

Nüüd kutsutakse autorun(); mis omakorda kutsub iga faaside funktsiooni. Pöörlemisuund sõltub sellest, mis järjekorras kutsutakse faaside funktsioone.

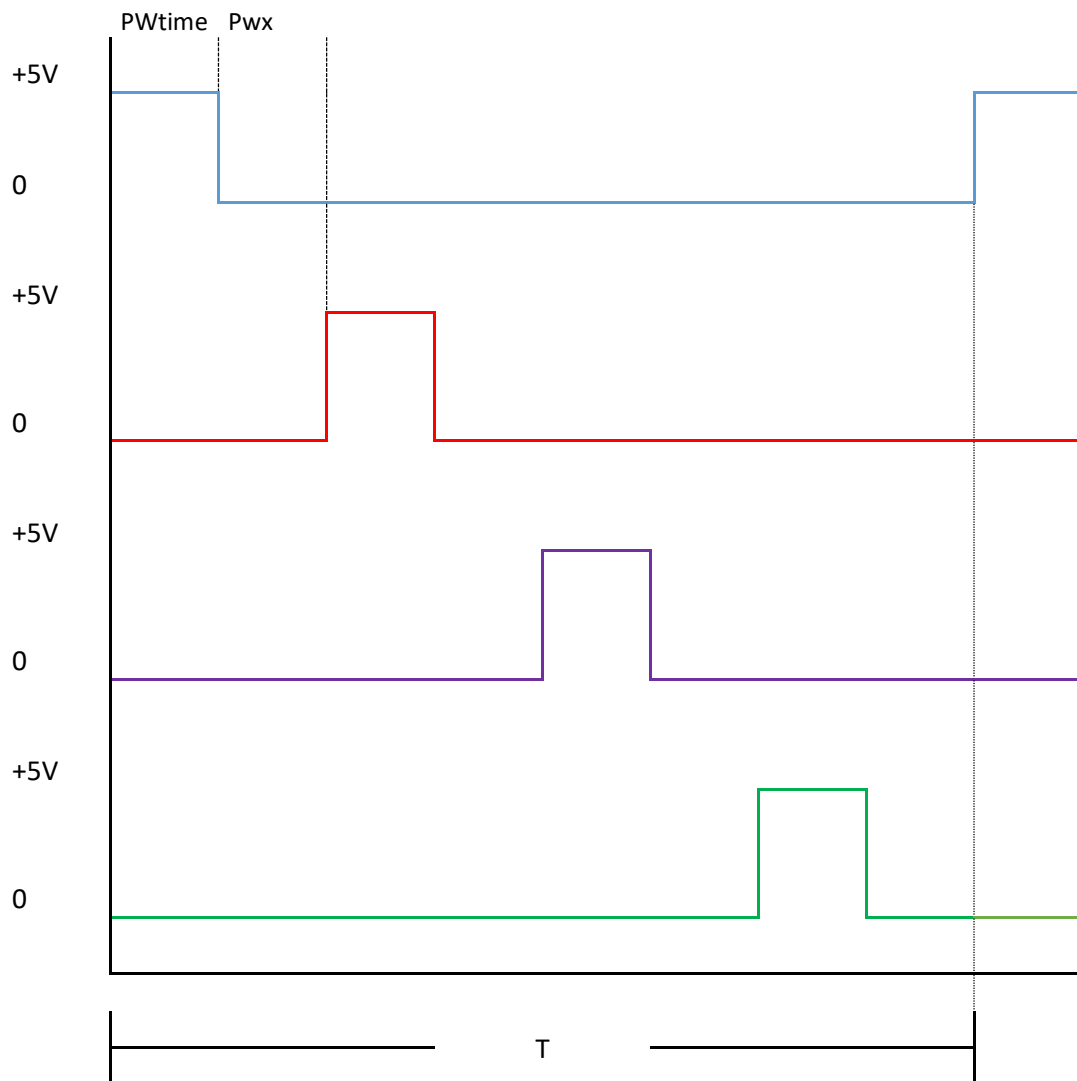


Joonis 4.11 Pöörlemisuuna valimis algoritm

Joonisel 4.11 on pöörlemisuuna valimis algoritm, kui faaside arv on 4. Juhul kui faaside arv on 3, faasi D ei kasuta.

Täissamm, 4 faasid.

Antud juhul pulsi laius peab olema maksimum 25%. Juhtimise impulsi pikkused ja viivitused impulsi vahel on toodud joonisel 4.12



Joonis 4.12 4 faasi täissammu juhtimis impulsi pikkus ja viivitused impulsi vahel

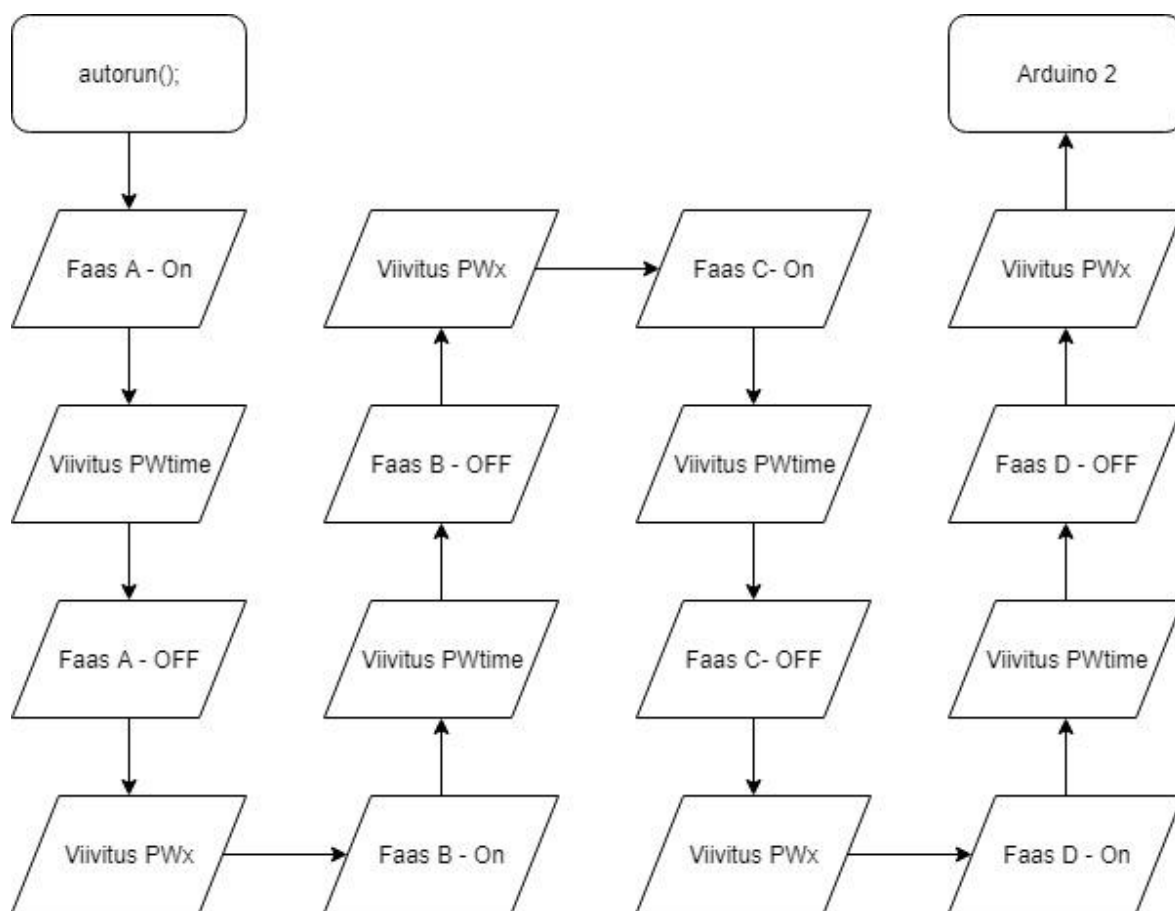
Kuna me leidsime pulsi pikkuse valemiga 2.2, on vaja arvutada faaside vaheline viivitus. Selleks esitame veel ühe muutuja PW_{25} , mis on impulsi laius 25% juures:

$$PW_{25} = T * 0,25 \quad (2.3)$$

Nüüd toimub faaside vahelise viivituse arvutamine kui 25% pikkuse impulsi ja pika impulsi vaheline erinevus:

$$PW_x = PW_{25} - PW_{time} \quad (2.4)$$

Joonisel 4.13 on 4 faasi täissammu faaside lülitus algoritm.

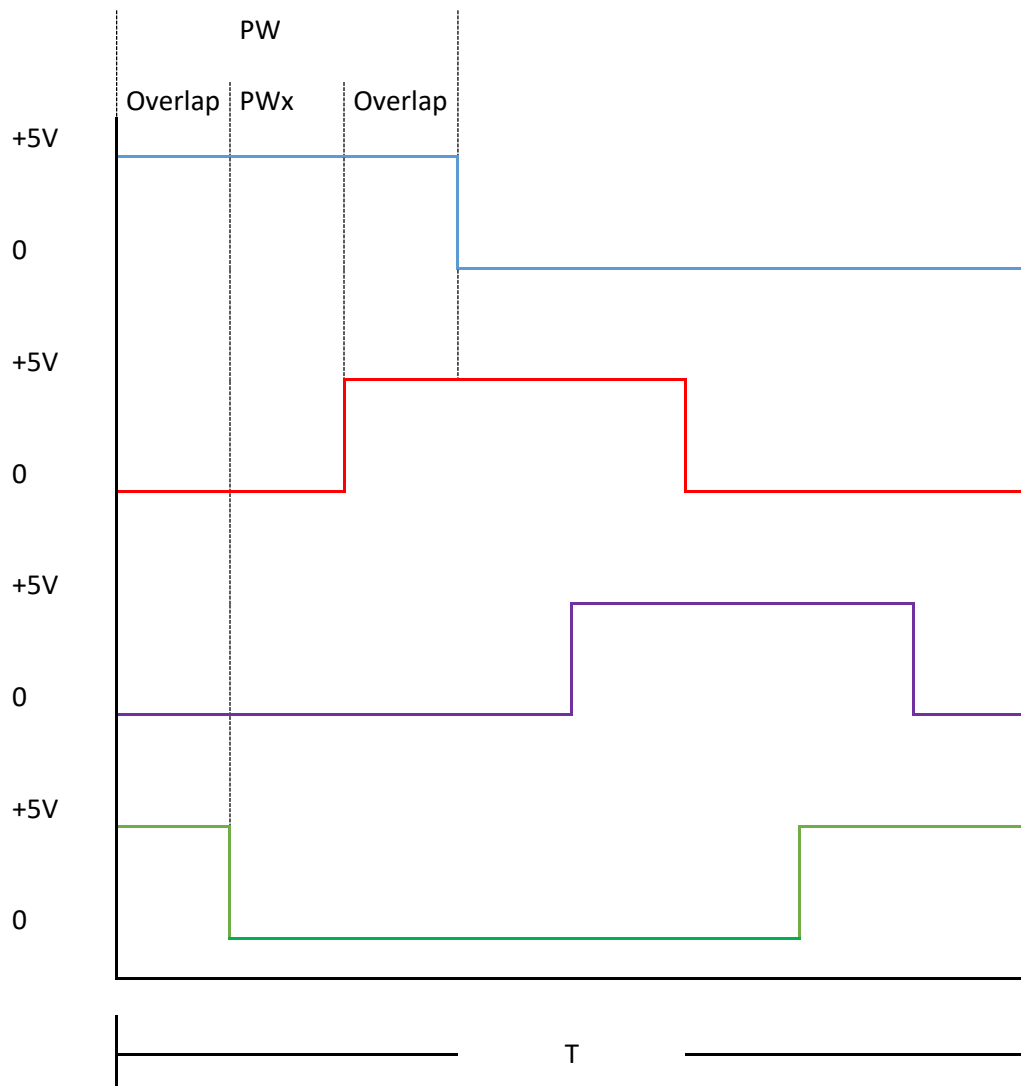


Joonis 4.13 4 faasi, täissammu faaside lülituse algoritm

Juhul kui pulsi laius on 25%, viivitus faaside vahel ei ole.

Poolsamm, 4 faasid:

Antud juhul pulsi laius on vahemikus 26-49%.



Joonis 4.14 4 faasi poolsammu impulsi pikkused ja viivitused

Viivituse arvutus loogika on järgmine

1. Arvutada perioodi pikkus, kasutades võrrandit 2.1
2. Arvutada pulsi laius, võrrand 2.2
3. Arvutada pulsi laius 25% juures, võrrand 2.3

Nüüd arvutatakse ülekatet, selleks kasutatakse valemit:

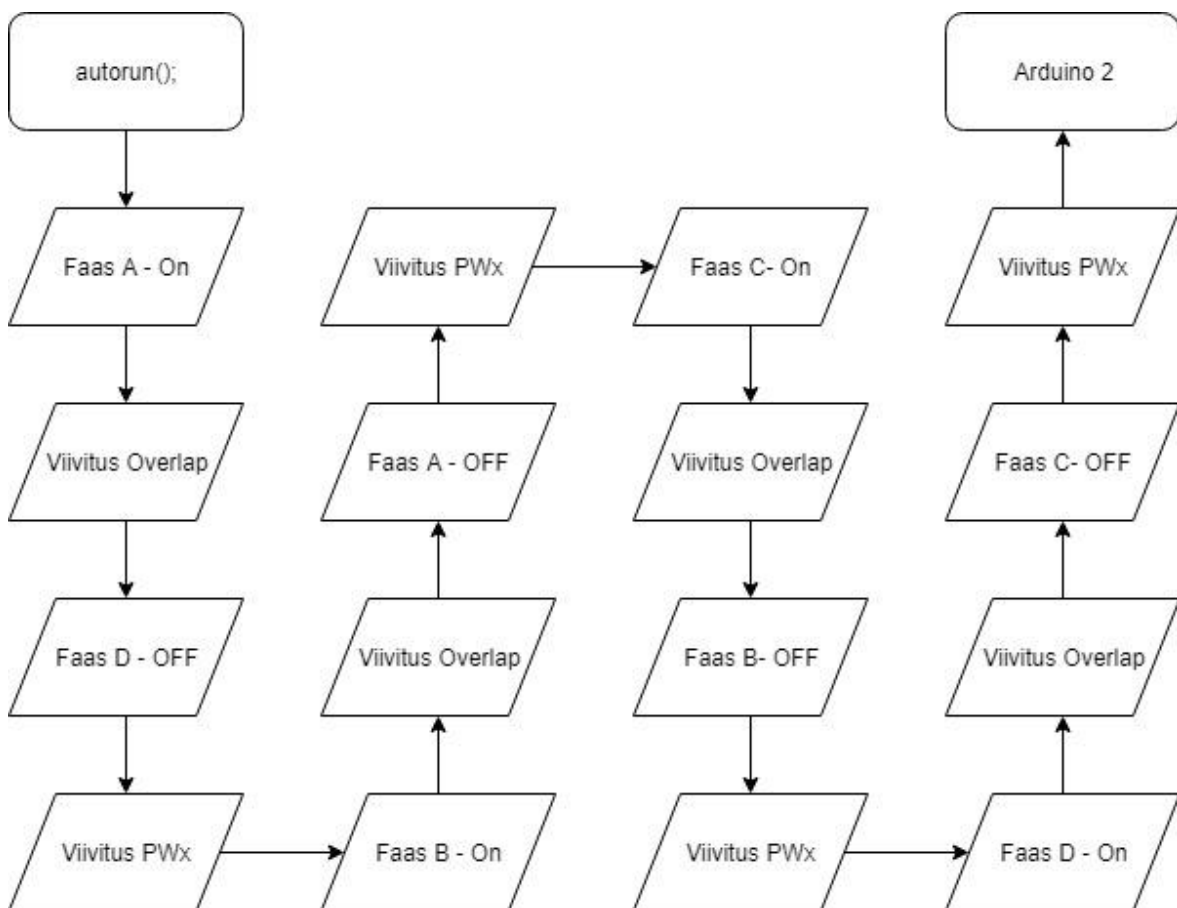
$$Overlap = PWtime - PW25 \quad (2.5)$$

Samuti, eelmise faasi väljalülitamiseks on vaja teada *overlap*-i ja pulsi laiuse vahe 25% juures. Selleks kasutame võrrandit:

$$PWx = PW25 - Overlap \quad (2.6)$$

Joonisel 4.14 on 4 faasi poolsammu juhtimis impulsi pikkus ja viivitused impulsi vahel

Joonisel 4.15 on 4 faasi poolsammu faaside lülitus algoritm.



Joonis 4.15 4 faasi, poolsammu faaside lülituse algoritm

Täissamm, 3 faasid.

Antud juhul pulsi laius on vahemikus 1-33,3%. Juhtimis loogika on sama nagu joonisel 4.13, vaid puudub faas D. Valemi 2.3 asemel kasutatakse järgmist valemit:

$$PW33 = T * 0,333 \quad (2.7)$$

Ja valemi 2.4 asemel kasutatakse järgmist valemit:

$$PWx = PW33 - PWtime \quad (2.8)$$

Poolsamm, 3 faasid.

Antud juhul pulsi laius on vahemikus 34-66,6%. Juhtimis loogika on sama nagu joonisel 4.15, vaid puudub faas D. Valemi 2.5 asemel kasutatakse järgmist valemit:

$$Overlap = PWtime - PW33 \quad (2.9)$$

Ja valemi 2.6 asemel kasutatakse järgmist valemit:

$$PWx = PW33 - Overlap \quad (2.10)$$

Möötmine

Manuaalses režiimis, mõõdetakse kohe iga faasi pinge ja vool, kuna vool ja pinge on konstantsed. Auto režiimis mõõdetakse perioodi jooksul voolu ja pinge kesk väärtused. Selleks programm peab arusaama, kust alustada mõõtmist ja mis hetkel lõpetada. Selleks Arduino 2, enne impulsi edastamist draiverile, saadab ka impulsi Arduino 1-le (joonis 3.1). Funktsioonis on kasutatud lõpmatu tsüklil while(true). Idee on järgmine:

1. Kui signaal on madal ja muutuja a = 0, siis muutuja a = 1. Seda kasutatakse selleks, et leida impulsi algus
2. Kui signaal on kõrge ja muutuja a = 1, siis see tähendab, et mähises on nüüd vool ja tuleb salvestada andmed: vool ja pinge. b = 1.
3. Kui signaal on madal ja b = 1, siis mähises voolu ei ole, aga ikka salvestame voolu ja pinge väärtused, ja a=2
4. Kui signaal on kõrg ja a = 2, siis mähises on jälle vool ja nüüd kasutame break-l ja väljastame andmed antud faasi kohta.
5. Punkt 1, vaid nüüd mõõdame järgmise faasi

5 KATSETAMINE

Nüüd, kui muundur on kokkupandud, saab seda katsetada. Katsetamiseks kasutatakse ostsilloskoopi GWINSTEK GDS-1104B ja 3D printeriga prinditud elektrimootor. Alustame juhtimise impulsiga.

3 faasid, sagedus 1Hz ja pulsi laius 10%

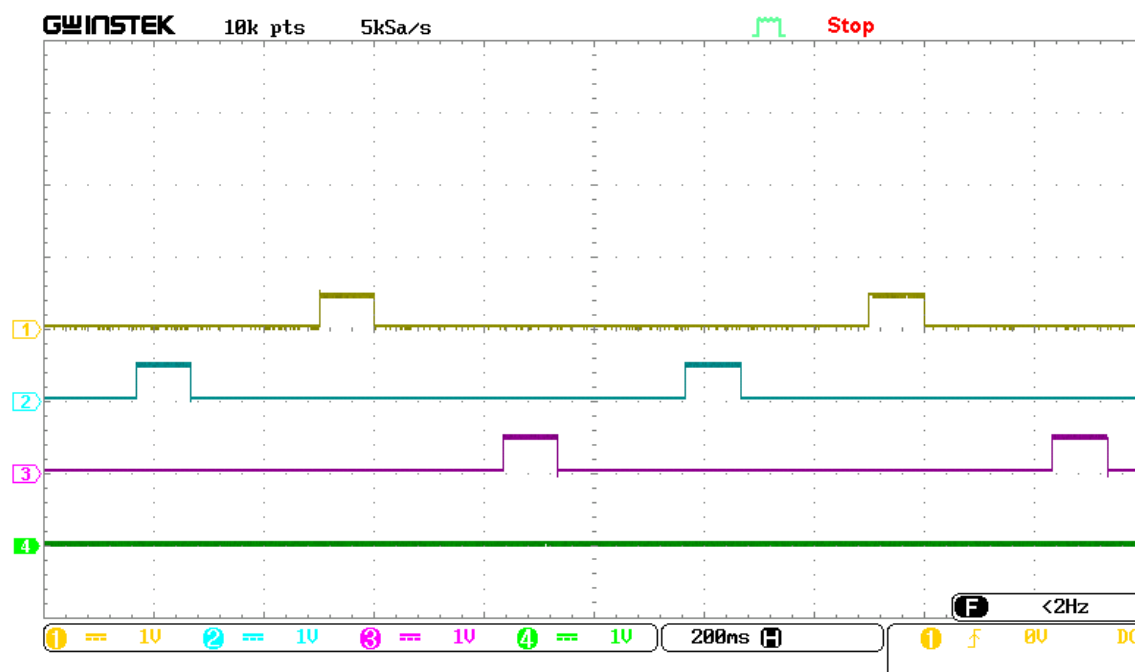
Viivituse kalkuleerimiseks kasutatakse valemid 2.1, 2.2, 2.7 ja 2.8.

Perioodi pikkus: $T = \frac{1}{1} * 1000 = 1000 \text{ ms}$

Pulsi laius: $PW_{time} = 1000 * \frac{10}{100} = 100 \text{ ms}$

Pulsi laius 33% korral: $PW_{33} = 1000 * 0,333 = 333 \text{ ms}$

Viivitus pulsi vahel: $PW_x = 333 - 100 = 233 \text{ ms}$



Joonis 5.1 3 faasid, sagedus 1Hz ja pulsi laius 10%

Joonisel 5.1 on näha, et pulsi laius on 100ms ja viivitus pulsi vahel on umbes 233ms, mis tähendab, et juhtsignaalid on õiged.

3 faasi, sagedus 1Hz ja pulsi laius 40%

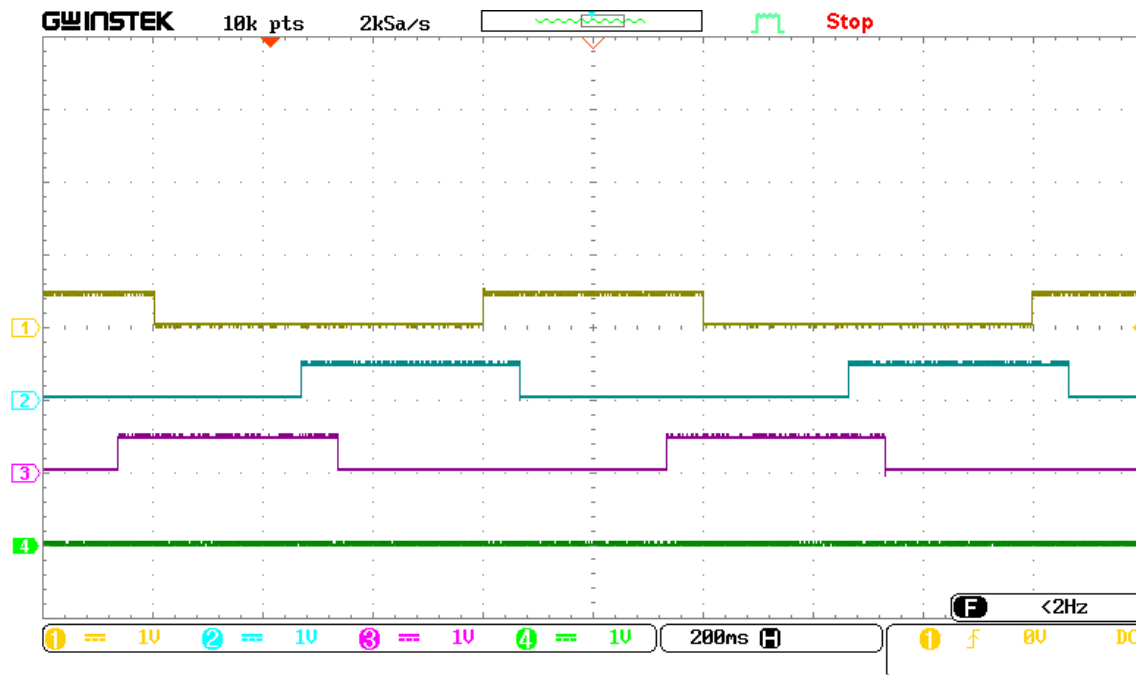
Viivituse kalkuleerimiseks kasutatakse valemid 2.1, 2.2, 2.9 ja 2.8.

Perioodi pikkus: $T = \frac{1}{f} * 1000 = 1000 \text{ ms}$

Pulsi laius: $PWtime = 1000 * \frac{40}{100} = 400 \text{ ms}$

Pulsi laius 33% korral: $PW33 = 1000 * 0,333 = 333 \text{ ms}$

Ülekate on: $Overlap = 400 - 333 = 67 \text{ ms}$



Joonis 5.2 3 faasi, sagedus 1Hz ja pulsi laius 40%

Joonisel 5.2 on näha, et pulsi laius on 400ms ja ülekate on umbes 67ms, mis tähendab, et juhtsignaalid on õiged.

4 faasi, sagedus 1Hz ja pulsi laius 10%

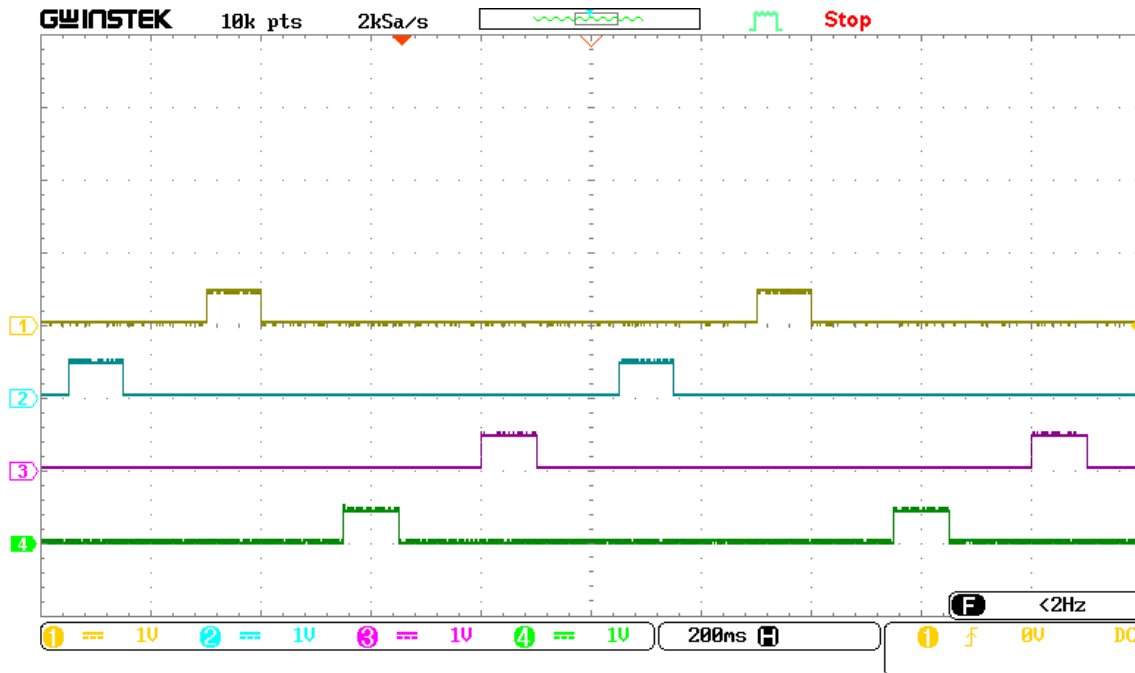
Viivituse kalkuleerimiseks kasutatakse valemid 2.1, 2.2, 2.3 ja 2.4.

Perioodi pikkus: $T = \frac{1}{f} * 1000 = 1000 \text{ ms}$

Pulsi laius: $PW_{time} = 1000 * \frac{10}{100} = 100 \text{ ms}$

Pulsi laius 25% korral: $PW_{25} = 1000 * 0,25 = 250 \text{ ms}$

Viivitus pulsi vahel: $PW_x = 250 - 100 = 150 \text{ ms}$



Joonis 5.3 4 faasi, sagedus 1Hz ja pulsi laius 10%

Joonisel 5.3 on näha, et pulsi laius on 100ms ja viivitus pulsi vahel on 150ms, mis tähendab, et juhtsignaalid on õiged.

4 faasi, sagedus 1Hz ja pulsi laius 40%

Kasutades valemid 2.1, 2.2, 2.3, 2.5 ja 2.6 saime väärtused:

Perioodi pikkus: $T = \frac{1}{1} * 1000 = 1000 \text{ ms}$

Pulsi laius: $PWtime = 1000 * \frac{40}{100} = 400 \text{ ms}$

Pulsi laius 25% korral: $PW25 = 1000 * 0,25 = 250 \text{ ms}$

Ülekate on: $Overlap = 400 - 250 = 150 \text{ ms}$

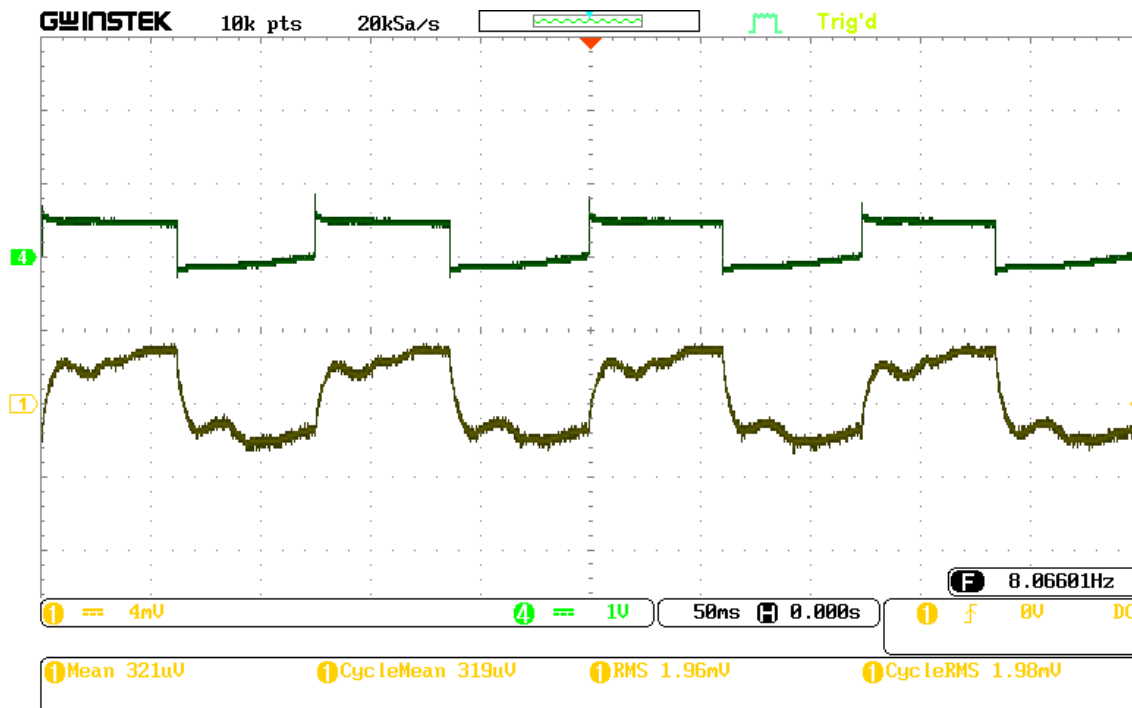


Joonis 5.4 4 faasi, sagedus 1Hz ja pulsi laius 40%

Joonisel 5.4 on näha, et pulsi laius on 400ms ja ülekate on 150ms, mis tähendab, et juhtsignaalid on õiged.

Mootori testimiseks ühendame 4 faasilise mootori muunduriga. Kuna kõik faasid on sümmeetrilised, siis mõõdame ainult ühe faasi voolu ja pinget. Joonisel 5.5 on A faasi tarbitav vool (1) ja pinget impulss (4). Joonisel 5.5 on ka näha, et vool ei ole väga stabiilne. Põhjus on siirdeprotsessid ja kuna koormus on induktiivne, siis on vaja kokku panna ka filtrid, selleks et stabiliseerida mootori tööd. Filtrite arvutamine ja kokkupanek, ei olnud selle töö eesmärk.

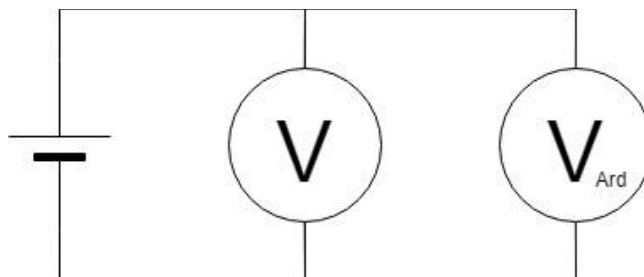
Mootori konstruktsioon ei ole ideaalne, seetõttu mootor pöörles suure raskusega.



Joonis 5.5 Faasi A vool (1) ja pinge (4)

Pingeanduri kalibreerimine

Pingeanduri kalibreerimiseks koostatakse skeem.



Joonis 5.6 Pingeanduri kalibreerimise skeem

Joonisel 5.6 on toite allikas, multimeeter KEITHLEY 2100 6 1/2 DIGIT MULTIMETER (V) ja pinge jaotur (Vard). Pinge väärtuse saamiseks Arduinol, on vaja korrutada ADC-i väärtus 20,46-ga. Põhjus on see, et 5V võrdub väärtusele 1023 ADC-s, mis tähendab, et iga 1 väärtus ADC-s võrdub 20,48mV Tabelis 5.1 on mõõte tulemused

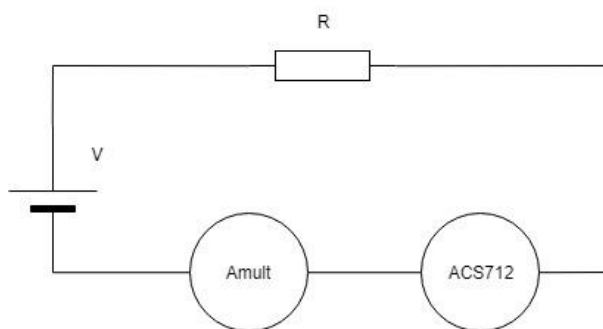
Tabel 5.1 Pinge mõõte tulemused

Multimeeter, V	Arduino, V	Viga, %
5.9944	5.97	0.407
12.006	12.06	0.450
17.999	18.15	0.839
23.977	24.22	1.013
29.97	30.32	1.168
34.981	35.4	1.198

Maksimaalne mõõte viga on 1.2%, aga tuleb võtta arvesse et, multimeetril on ka teatud mõõtmise viga, 0,0038%

Vooluanduri kalibreerimine

Vooluanduri kalibreerimiseks koostatakse skeem (joonis 5.7). Joonisel on toite allikas, multimeeter KEITHLEY 2100 6 1/2 DIGIT MULTIMETER (Amult) ja voolusensor ACS712. Kuna voolu sensor mõõdab +-5A, siis 0 väärtus voolu sensori väljundis on 2.5V, mis tähendab et kui vool on 0, ADC-i väljundis on 512. Seega tuleb Voolu väärtus Arduinol on $(2,5V - (ADC \text{ väärtus} * (5/1024)))/0,185$. Testimiseks muudame toite allika pinget ja takistuse väärtus ei muuta.



Joonis 5.7 Vooluanduri kalibreerimise skeem

Tabelis 5.2 on mõõte tulemused ja on näha et, viga ei ületa 5%, aga tuleb võtta arvesse et, multimeetril on ka teatud mõõtmise viga, 0,120%.

Tabel 5.2 Voolu mõte tulemused

Pinge, V	Multimeeter, A	Arduino, A	Viga, %
6	0.169	0.172	1.78
12	0.339	0.347	2.36
18	0.509	0.524	2.95
24	0.681	0.711	4.41
30	0.855	0.895	4.68

KOKKUVÕTE

Töö käigus sain teada, millest koosneb samm-mootori juhtimismuundurid. Samuti sain teada, et turul on kättesaadavad väga palju juhtimismuundureid erineva võimsusega, ja kuna selle töö eesmärk oli arendada, ehitada ja katsetada juhtimismuundurit, siis turu analüüs oli väga kasulik. Arduino-st sain teada, et on erinevat PWM-i saamise võimalusi, aga osasid nendest on keerulisem programmeerida. Sain teada et samm-mootori juhtimiseks on vaja tegelikult ainult kolm juhtimisparameetrit: pulsilaius, sagedus ja samm.

Moodulite valimise ajal sain teada, et mooduleid on väga palju ja enne tuleb teada millist koormust on vaja juhtida, ja selle koormuse parameetreid. Turul on väga palju erinevaid ekraane, draivereid, sensoreid ja muid mooduleid, mis lihtsustasid prototüübi kokkupaneku, kuna ei olnud vajadust koostada ja joota trükkplaate, aga samal ajal, kui analoogse prototüübi jaoks koostada trükkplaate, siis see saab olema kompaktsem. Moodulite valiku ajal, ei tekkinud suuremaid raskusi.

Muunduri kokkupanek oli lihtne, tuleb vaid õppida, kuidas paigaldada erinevaid komponente piiratud ruumis. 3D printeri kasutamine lihtsustas selle osa, kuna oli võimalus joonistada vajalikud mudelid iga mooduli jaoks.

Suurem osa antud töös on programmeerimine. Sellega tekkisid raskused, kuna parameetreid on palju, juhtsignaale on palju ja lisaks oli kolm mikrokontrollerit: Arduino 1, Arduino 2 ja Nextion ekraan. Kuna kõik kontrollerid peavad oma vahel suhtlema, siis on vaja koostada juhtsignaali skeem enne programmeerimist, see lihtsustab programmeerimist. Pole mõtet ka koostada terve süsteemi algoritm, kuna see on keeruline ja väga suure mahuline töö, algoritmid tuleb alati jagada.

Katsetamisega oli raskusi, kuna mootor, millega testitakse muundurit ei olnud ideaalne. Samuti muunduri häälestamiseks on vaja ehitada filtreid, kuna koormus on induktiivne ja tekivad siirdeprotsessid.

Ehitatud muundurit saab ka kasutada harjavaba alalisvoolu mootoriga, kuna nende tööpõhimõtte on sarnane ja juhtimiseks on vaja pulsilaiust ja sagedust.

Lõpuks võin öelda, et tööeesmärk on saavutatud. Muunduriga saab juhtida 3D printitud mootorit. Töö käigus oli saanud lai praktiline kogemus, ning uued teadmised ja oskused.

CONCLUSION

In the course of my work I learned more about stepper motor control. I also found out that there are a lot of control systems available on the market with different parameters, and because the purpose of this work was to develop, build and test a control driver, the market analysis was very helpful. I learned about Arduino that there are different ways to get PWM, but some of them are more difficult to program. I learned that only three control parameters are needed to control the stepper motor: pulse width, frequency and step size.

At the time of module selection, I found out that there are a lot of modules and before choosing right one we need to know what load needs to be controlled and load parameters. There are a lot of different screens, drivers, sensors and other modules on the market that made it easier to assemble a prototype because there was no need solder the PCB, but at the same time PCB with all components for a similar prototype, would be more compact. During the choice of modules, no major difficulties arose.

The assembly of the converter was easy, just needed to learn how to install different components in a limited space. Using a 3D printer made it easier because I could draw the necessary models for each module.

Biggest part of this work is programming. This caused difficulties because of the many parameters, the number of control signals and three microcontrollers: Arduino 1, Arduino 2 and Nextion. Since all controllers need to communicate between each other, it is necessary to draw up a control signal scheme before programming, which makes programming easier. It also makes no sense to compile an entire system algorithm, because it is a complex and very large volume of work, algorithms must always be separated to smaller ones.

The testing was difficult because the motor that used for testing the converter was not ideal. It is also necessary to build filters for tuning the converter as the load is inductive and transient processes occur.

The built converter can also be used with a brushless DC motor, as their operation is similar and requires a pulse width and frequency for control.

Finally, I can say that the work goal has been achieved. The converter can be used to control a 3D print motor. In the course of my work I got huge practical experience also new knowledge.

KASUTATUD KIRJANDUSE LOETELU

- [1] "Stepper Motor Drivers - AC & DC Input Stepper Motor Driver Products." [Võrgumaterjal]. Saadaval: <https://www.orientalmotor.com/stepper-motors/stepper-motor-drivers.html>. [Kasutatud: 13-May-2019].
- [2] "E. Brindfeldt, E. Pettai Täituriid robotikas."
- [3] "NextMove e100." [Võrgumaterjal]. Saadaval: <https://new.abb.com/drives/low-voltage-ac/motion/nextmove-e100>. [Kasutatud: 15-May-2019].
- [4] "MicroFlex analog - compact servo drive." [Võrgumaterjal]. Saadaval: <https://new.abb.com/drives/low-voltage-ac/motion/legacy-motion-drives/microflex-analog>. [Kasutatud: 24-May-2019].
- [5] "TB6600." [Võrgumaterjal]. Saadaval: <https://arduino.ua/prod2872-draiver-shagovogodvigatelya-tb6600-4a>. [Kasutatud: 15-May-2019].
- [6] "SD315DN10B400 - motion control stepper motor drive." [Võrgumaterjal]. Saadaval: <https://www.schneider-electric.com/en/product/SD315DN10B400/motion-control-stepper-motor-drive---sd315---pulse-direction-without-oscillator/>. [Kasutatud: 15-May-2019].
- [7] "Pulsilaiusmodulatsioon." [Võrgumaterjal]. Saadaval: <https://sisu.ut.ee/elektroonika/56-pulsilaiusmodulatsioon>. [Kasutatud: 15-May-2019].
- [8] "Secrets of Arduino PWM." [Võrgumaterjal]. Saadaval: http://www.righto.com/2009/07/secrets-of-arduino-pwm.html?m=1&fbclid=IwAR0SasVWay_DUMJz4lqpUenVK1Nm_fKW29Qka2AvsDw3IS5U7nJoBWtvrY+. [Kasutatud: 15-May-2019].
- [9] "Arduino Nano." [Võrgumaterjal]. Saadaval: <https://store.arduino.cc/arduino-nano>. [Kasutatud: 13-May-2019].
- [10] "Arduino Uno Rev3." [Võrgumaterjal]. Saadaval: <https://store.arduino.cc/arduino-uno-rev3>. [Kasutatud: 13-May-2019].
- [11] "Arduino Mega 2560 Rev3." [Võrgumaterjal]. Saadaval: <https://store.arduino.cc/mega>

2560-r3. [Kasutatud: 13-May-2019].

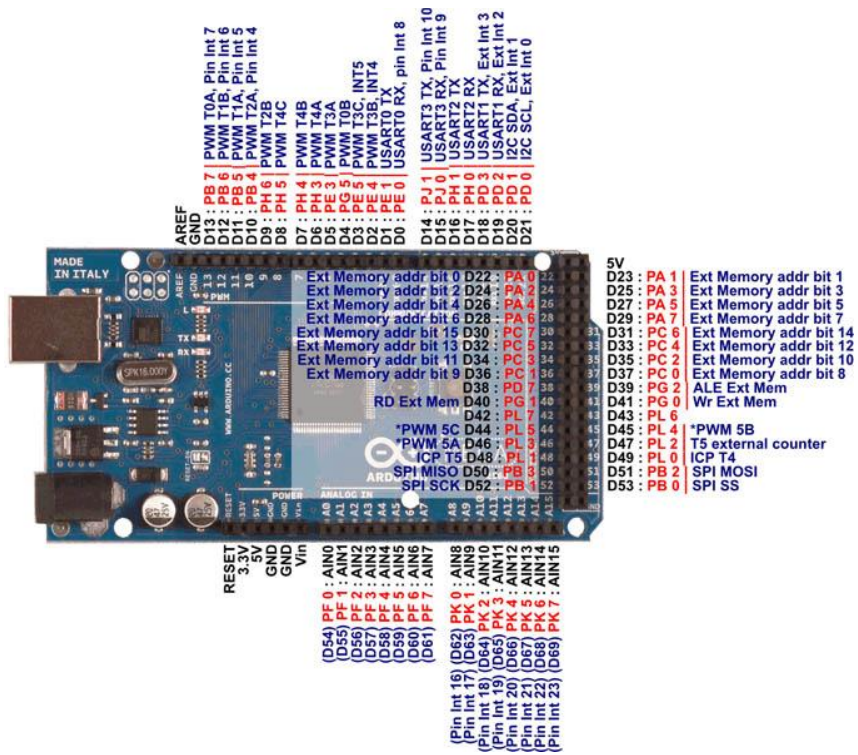
- [12] "FeatherWing 3.5." [Võrgumaterjal]. Saadaval: <https://ru.mouser.com/new/adafruit/adafruit-featherwing-tft/>. [Kasutatud: 14-May-2019].
- [13] "Nextion NX3224K024_011R." [Võrgumaterjal]. Saadaval: <https://www.itead.cc/nextion-nx3224k024.html>. [Kasutatud: 14-May-2019].
- [14] "4D Systems | gen4-uLCD-43D." [Võrgumaterjal]. Saadaval: https://www.4dsystems.com.au/product/gen4_uLCD_43D/. [Kasutatud: 14-May-2019].
- [15] "L298N." [Võrgumaterjal]. Saadaval: <http://pdf1.alldatasheet.com/datasheet-pdf/view/22440/STMICROELECTRONICS/L298N.html>. [Kasutatud: 14-May-2019].
- [16] "MD13s." [Võrgumaterjal]. Saadaval: <https://www.cytron.io/p-13amp-6v-30v-dc-motor-driver>. [Kasutatud: 14-May-2019].
- [17] "TMC260 Stepper Motor Driver Shield For Arduino-DFRobot." [Võrgumaterjal]. Saadaval: <https://www.dfrobot.com/product-1360.html>. [Kasutatud: 14-May-2019].
- [18] "ACS712." [Võrgumaterjal]. Saadaval: <http://pdf1.alldatasheet.com/datasheet-pdf/view/168326/ALLEGRO/ACS712.html>. [Kasutatud: 14-May-2019].
- [19] "INA219." [Võrgumaterjal]. Saadaval: <http://pdf1.alldatasheet.com/datasheet-pdf/view/249609/TI/INA219.html>. [Kasutatud: 14-May-2019].
- [20] "Current Carrying Capacity Table." [Võrgumaterjal]. Saadaval: <https://www.sab-cable.com/cables-wires-harnessing-temperature-measurement/technical-data/cables-and-wires/instructions-for-the-safe-application-of-cables/boundary-conditions/calculate-wire-cross-section-current-carrying-capacity-table.html>. [Kasutatud: 17-May-2019].

LISAD

Lisa 2.1 Ühendused Arduino ja komponentide vahel

I/O Digital		Arduino 1	Arduino 2	I/O Digital		Arduino 1	Arduino 2
0	D0	Ekraan TX, In		35	D35		Manual B, In
1	D1	Ekraan RX, Out		36	D36	Measure A, In	Manual C, In
2	D2			37	D37	Measure B, In	Manual D, In
3	D3			38	D38	Measure C, In	Phase A, Out
4	D4			39	D39	Measure D, In	Phase B, Out
5	D5			40	D40	Manual A, Out	Phase C, Out
6	D6			41	D41	Manual B, Out	Phase D, Out
7	D7			42	D42	Manual C, Out	
8	D8			43	D43	Manual D, Out	
9	D9			44	D44		
10	D10			45	D45		
11	D11			46	D46		
12	D12			47	D47		
13	D13			48	D48		
14	D14			49	D49	START, Out	
15	D15			50	D50	Phases, Out	Measure A, Out
16	D16			51	D51	Dir, Out	Measure B, Out
17	D17			52	D52	Step, Out	Measure C, Out
18	D18			53	D53	Mode, Out	Measure D, Out
19	D19			54	A0	PW pot, In	PW pot, In
20	D20			55	A1	frq pot, In	frq pot, In
21	D21			56	A2	ramp pot, In	ramp pot, In
22	D22			57	A3		
23	D23		START, In	58	A4		
24	D24		Phases, In	59	A5		
25	D25		Dir, In	60	A6	Current In, In	
26	D26		Step, In	61	A7	Current A, In	
27	D27		Mode, In	62	A8	Current B, In	
28	D28			63	A9	Current C, In	
29	D29			64	A10	Current D, In	
30	D30			65	A11	Voltage In, In	
31	D31			66	A12	Voltage A, In	
32	D32			67	A13	Voltage B, In	
33	D33			68	A14	Voltage C, In	
34	D34		Manual A, In	69	A15	Voltage D, In	

Lisa 2.2 Arduino Mega pinnid



Lisa 1.2 Muundur, vaade ülevaet



Lisa 3.3 Muundur, vaade vasakult



Lisa 3.4 Muundur, vaade paremalt

