

Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Tehnoloogiainstituut

Kaur Kullamäe

Elektroonikalahendus ja püsivara robotkäte juhtimiseks sotsiaalsel humanoidrobotil SemuBot

Bakalaureusetöö (12 EAP)
Arvutitehnika eriala

Juhendaja:
Karl Kruusamäe, PhD

Tartu 2024

Resüme

SemuBoti käe elektroonika süsteemi loomine

Tänapäeva tehnoloogia areng on viinud meid punkti, kus humanoidrobotid on muutumas üha enam reaalsuseks. Üks oluline osa sellest on robotite võime liigutada oma jäsemeid sarnaselt inimesele. Kuigi robotitehnoloogia on kiiresti arenenud, on endiselt väljakutseid, eriti seoses humanoidsete robotite käe liigutamisega.

Selle probleemi lahendamiseks on loodud tudengiprojekti SemuBoti esimene versioon, kus on valminud süsteem, mis võimaldab liigutada humanoidse roboti käsi.

Bakalaureuse töö tulemusena valmis elektroonika süsteem, mis võimaldab liigutada SemuBoti käes olevaid mootoreid ja kasutada tajureid käe asukoha saamiseks.

Järgmised sammud hõlmavad süsteemi täiustamist ja optimeerimist, et parandada roboti käe liikumise täpsust ja sujuvust. Samuti on plaanis süsteemi integreerida enda arendatud trükkplaat ja ROS operatsioonisüsteem.

CERCS: T125 Automatiseerimine, robotika, juhtimistehnika, T170 Elektroonika

Märksõnad: Robotika, humanoidne robot

Abstract

Building the SemuBot arm electronics system

Advances in technology today have brought us to the point where humanoid robots are becoming more and more of a reality. One important part of this is the ability of robots to move their limbs in a human-like manner. Although robotics technology has advanced rapidly, there are still possibilities, especially humanoid robot arm movement.

To solve this problem, the first version of the student project SemuBot was created, where a system was completed that allowed the movement of the arms of a humanoid robot.

Bachelor's thesis: a good, ready-made electronics system that can move the motors in SemuBot's hand and use the jubaras at hand.

The next steps include improving and optimising the system to improve the accuracy and smoothness of the robot arm's movement. It is also planned to integrate the self-developed printed circuit board and the ROS operating system into the system.

CERCS: T125 Automatiseerimine, robootika, juhtimistehnika, T170 Elektroonika

Keywords: Robotics, humanoid robot

Sisukord

Resümee	2
Abstract	2
Sisukord	4
1. Sissejuhatus	5
1.1 SemuBot	5
1.2 Lõputöö eesmärk	6
1.3 Tulemused	6
2. Kirjanduse ülevaade	7
2.1 Sotsiaalne robotika	7
2.2 Varasemate humanoidrobotite ülevaade	8
2.2.1 Poppy	8
2.2.2 Igus Humanoid avatud platvorm (Igus Humanoid Open Platform)	9
2.2.3 Robot Pepper	10
3. Loodud lahendus	12
3.1 Loodud lahenduse eesmärk ja nõuded	12
3.2 Loodud lahendus	13
3.2.1 Mootorite draiverid	15
3.2.2 Absoluutne kodeerija	19
3.2.3 Püsivara	21
3.3 Mikrokontroller	24
3.3.1 Arduino ja Nucleo võrdlus	24
3.4 Loodud lahendus Arduinoga	25
3.5 Loodud lahendus Nucleoga	27
Arutelu	30
Kokkuvõte	31
Tänuavaldused	32
Viited	33

1. Sissejuhatus

1.1 SemuBot

Tudengiprojekti SemuBot eesmärgiks on luua humanoidrobot, mis viiakse ellu üliõpilaste poolt. Selle viivad täide erinevate koolide ja instituutide tudengid. Projekti lõpp tulemusena valmib Eesti esimene humanoidrobot, mis suudab iseseisvalt liikuda, väljendada ennast eesti keeles. SemuBoti peamine valdkond on toetada laste suhtlusoskuste arengut. Robotit saaks kasutada psühholoogi, arsti või logopeedi juhendamisel. SemuBoti eeliseks on see, et robotid võivad luua sotsiaalse sideme patsientide ja meditsiinitöötajatega, samuti saavad suhelda loomulikult viisil inimestega. Robotid on kasulikud ka teistes meditsiini valdkondades, näiteks vanemate inimeste abistamisel. Eakad vajavad igapäevaseid kes vajavad igapäevaseid toiminguid tehes tuge, mingite asjade osutamine või pärast operatsiooni taastumisperiodil abi.

SemuBoti planeerimisel pandi paika mitu nõuet. Üheks nõudeks on see, et robot peab olema välimuselt nunnu ja lihtsasti kasutatav, näiteks logopeed oskaks seda kasutada. Lisaks kehakeele kasutamiseks peab SemuBot saama liigutada oma käsi, liikuda iseseisvalt ruumis ringi ja suhelda inimestega.

1.2 Lõputöö eesmärk

Humanoidrobot, nagu SemuBot, koosneb mitmest erinevast kehaosast. Lisaks roboti jalgadele, puusadele, kehale ja peale, on oluline et robot suudaks ka oma käsi liigutada ja kasutada. Käte liikumine võimaldab robotil ennast kehakeelega väljendada, mis võimaldaks suhtluse teha meeldivamaks. Lisaks sellele saab kätega haarata esemeid, hoida, tõsta ja paigutada asju. Nende kasutamine võimaldab ka robotil suhelda inimestega, näiteks žestide ja puudutustega.

Minu lõputöö eesmärgiks on luua tehniline lahendus, millega on võimalik liigutada SemuBoti käsi. Lisaks kirjeldada liidesed, et kõrgema taseme tarkvara saaks roboti käsi juhtida.

Selle tulemusena on minu lõputöö ülesanded järgmised:

- valida sobivad mootorid ja koodrid käele.
- liidestada mootorid ja koodrid mikrokontrolleriga.
- luua tarkvara, mis oleks vahelüli mootorite ja peaarvuti vahel.

1.3 Tulemused

Lõputöö tulemusena valmis süsteem, millega saab ühendada viis erinevat mootorit ja andurit. SemuBoti käes kasutatakse kokku viit erinevat mootorit ja nendega koos on tajurid. Viikude arvu tõttu oli lõpuks kasutusel Nucleo-F303RE[1]. Selle mikrokontrolleriga kontrollitakse igat mootorit eraldi, tajurid tagavad roboti stabiilsuse ja reageerimise väliskeskkonnale. Mikrokontroller oli läbi USB ühenduse ühendatud peaarvutiga, mille ülesandeks oli saata mikrokontrollerile käsklusi roboti liikumiseks. Peaarvutiga saab juhtida igat mootorit eraldi ja tajurid teevad kindlaks, et mootorid ei liigutaks kätt millegi vastu.

2. Kirjanduse ülevaade

Roboti ehitamine on kompleksne ja ressursimahukas protsess, mis nõuab märkimisväärset aega ning pingutust mehaaniliste, tarkvaraliste ja elektrooniliste aspektide omandamiseks. Läbi nende teadmiste rakendamise on võimalik lõpuks ehitada töötav robot. Selle protsessi käigus võivad ilmned spetsiifilised lahendused või disainivalikud, mis oluliselt parandavad roboti funktsionaalsust. Sageli dokumenteeritakse ja jagatakse selliseid avastusi avalikult, et aidata teisi sarnaste projektidega tegelevaid inimesi. Järgnevas tekstis käsitletakse lähemalt eelnevalt loodud roboteid ning analüüsitakse nende komponente.

Iga projekti alguses on vajalik teha põhjalik uurimistöö, et selgitada välja, kas sarnane projekt on varem teostatud. Õnneks on aastate jooksul loodud palju erinevaid roboteid, mis pakuvad väärtuslikku teavet ja inspiratsiooni. Käesolevas peatükis keskendutakse varasematele robotite projektidele, analüüsides nende omadusi ja näitajaid.

2.1 Sotsiaalne robotika

Sotsiaalne robotika on põnev valdkond, mis keskendub robotitele, mis suudavad inimestega suhelda ja nendega suhteid luua. Näiteks on olemas robotid nagu Pepper[2] ja Nao[3], mis on loodud selleks, et suhelda inimestega. Need robotid kasutavad tajureid, et tõlgendada inimeste kehakeel, näoilmeid ja kõnet. Lisaks saab kasutada selliseid roboteid meditsiini valdkonnas. Näiteks saab neid kasutada suhtlemisoskuste arendamisel väikeste laste seas, kellel võib olla rääkimisega raskusi. Robotid saavad pakkuda ka emotsionaalset tuge või aidata arstidel jälgida patsientide tervist. Robotite kasutamine meditsiinis on oluline mitmel põhjusel, esiteks robotid suudavad pakkuda järjepidevat ja väsimatut tuge, mis on eriti kasulik pikaajasel ravi või teraapia puhul. Teiseks on robotid võimelised koguma ja analüüsima suurel hulgal andmeid reaalsajas, mis aitab arstidel teha täpsemaid diagnoose ja jälgida patsientide tervislikku seisundit.

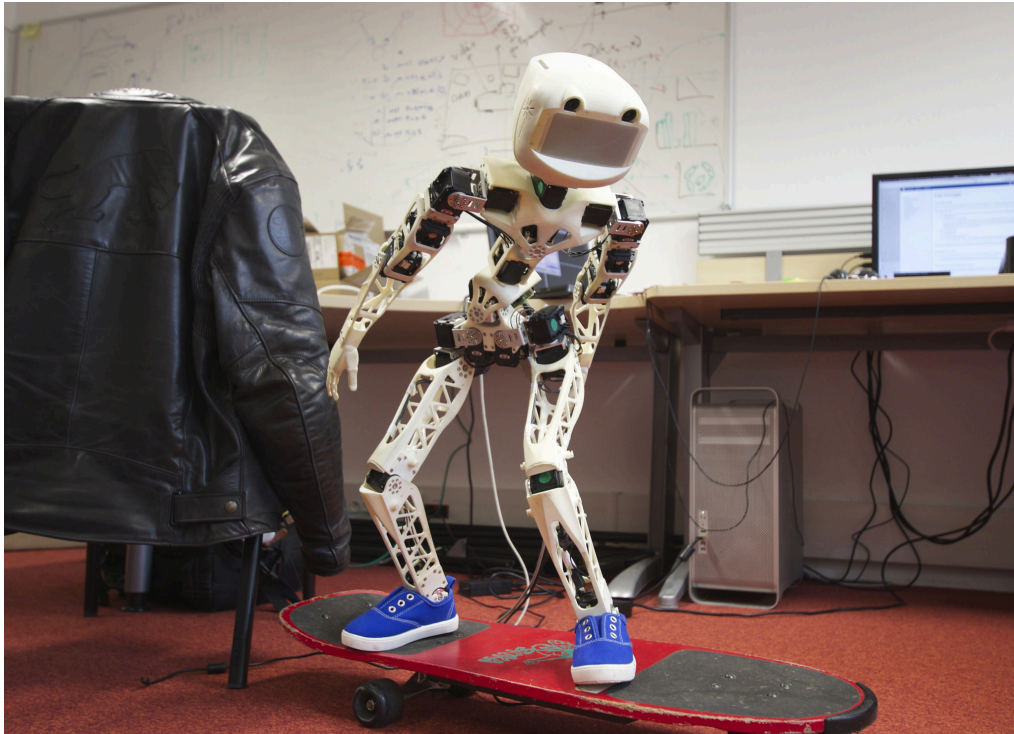
SemuBot võib hakata tegelema sarnaste teemadega, arendades välja võimekuse suhelda inimestega empaatilisel ja mõistval viisil. See võib olla näiteks võimet tuvastada ja reageerida inimese emotsioonidele, aidata inimestel lahendada probleeme ja pakkuda seltsi.

2.2 Varasemate humanoidrobotite ülevaade

Enne Semuboti on humanoidrobotid olnud juba tükk aega kasutusel olnud erinevates valdkondades. Peaaegu iga eelnev humanoidrobot on olnud erinev, igäühel oma unikaalne omadus. Selle tulemusena on need robotid hästi dokumenteeritud ja välja toodud milliseid lahendusi on kasutatud, et saaks sellist robotit teha. Nagu näiteks robotid Poppy[4], Pepper[2] ja Igus Humanoid robot[5].

2.2.1 Poppy

Poppy (joonis 1) on robotika maailmas üks humanoidsetest robotitest. Robot on loodud nii, et teda oleks võimalikult lihtne kasutada ja ta oleks modulaarne. Ükskõik, kes tahab, saaks osta endale ühe komplekti ja robot tükkidena kokku panna. Poppy kogumaksumus on 3D prinditud tükkidega ligikaudu 9588€ ja ilma 3D prinditud osadeta 7560€ [6]. Hind läheb nii kõrgeks, kuna robotis on kasutusel 25 servo mootorit. Nendest 21 on tippklassi *Robotis Dynamixel MX-28*, kaks MX-64 ja kaks AX-12 mootorit. Lisaks on kasutusel ka 16 jõutundlikut takistit [7]. Selle komplektiga saab endale lõpuks valmis ehitada ühe roboti, mis kaalub kõigest 3.5 kg ja on 83 cm[7]. Poppy on mõeldud teadlastele katsetamiseks, õpetajatele õpetamiseks ja teistele entusiastidele oma loominguga jagamiseks[6]. Koolides oleks hea kasutada sellist robotit, et õpetada õpilastele informaatikat läbi robotika ja visuaalse programmeerimise.



Joonis 1. *Poppy robot* [4]

2.2.2 Icus Humanoid avatud platvorm (Icus Humanoid Open Platform)

Mõned aastad tagasi tootis humanoidroboti Saksamaal olev ettevõtte nimega Icus, mis keskendub just plastidest valmistatud toodetele [8].



Joonis 2. *Icus Humanoid robot* [8]

Sellele ei ole otseselt nime antud, kuid see loodi, et oleks saadaval avatud lähtekoodiga robot, millega teha teaduslikke eksperimente [8]. Võrreldes Poppy'ga, on see natukene pikem ja raskem, 90 cm ja 6 kg [8]. See on natukene võimekam ka, toetab WiFi ühendust üks kaamera ja erinevad tajurid. Andurite hulka kuuluvad güroskoop, kiirendusmõõtur, magnetomeeter ja enkoodrid [5]. Erinevate tajurite juhtimiseks kirjutati kontrolleriile uus püsivara, et robot saaks kiiremini andmeid kätte. Samuti on olemas akutoide ja kogu süsteem kasutab ROS (*Robot Operating System*) tarkvara, mis on robotika maailmas suhteliselt populaarne. Antud robot ja selle tarkvara on loodud jalgpalli võistluse jaoks, kus mängijateks olid robotid ise [9].

2.2.3 Robot Pepper

SemuBoti arendamisel on suureks inspiratsiooniks olnud robot nimega Pepper. Seda just sellepärast, et Pepperit on massiliselt toodetud ja sarnaneb mitmes aspektis SemuBoti visioonile.



Joonis 3. Robot Pepper [2].

Pepperil on inimesega suhtlemiseks rinnale asetatud suur ekraan. Maailmas manööverdamiseks on paigaldatud nii 3D tajurid kui ka kaamerad. Lisaks nendele anduritele, on Pepperil veel tajureid (3D, laser, sonar, infrared andur), et robot oleks ohutum ja ei sõidaks kogemata kuhugi sisse [2]. Nende tajurite hulka kuulub mikrofonid, inertsiaalandur, puuteandur, infrapuna- ja ultraheliandur. Roboti kõrgus on 1210 mm, laius 480 mm, sügavus 425 mm ja kaalub tervenisti 28 kg[2].

Turvalisuse poole pealt on Pepperi käte liigestesse asetatud andurid, et saaks kinnitada, et jäsemed ei läheks valesse asendisse või isegi teeks inimesele liiga[2].

3. Loodud lahendus

3.1 Loodud lahenduse eesmärk ja nõuded

Lõputöö eesmärgiks on luua mootoritest, kodeerijatest ja mikrokontrollerist koosnev süsteem. Selle jaoks tuleb luua elektroonikasüsteem, püsivara lahendus ja seda loodud lahendust tuleb testida. Eesmärgiks ei ole panna robotit konkreetseid tegevusi tegema, vaid luua süsteem, mis võimaldab neid tegevusi. Näiteks võib loodud süsteemi kasutada järgmiste ülesannete täitmiseks:

- Objektide tõstmine ja asetamine
- Joonistamine
- Emotsioonide edasiandmine kehakeele miimikaga

Kehakeele miimika saavutatakse, kui luua lahendus, kus mootorid saavad liigutada keha mehaanilisi osi nii, et kasutaja saab igat mootorit eraldi juhtida. See tähendab, et iga liigutus ja poos on täpselt programmeeritav ja juhitav.

Kuna roboti käe liigesed on disainitud niimoodi, et need ei liigu täiesti 360°, siis näiteks tajurid peavad tagama, et mootorid ei kahjustaks mehaanilisi osi.

Elektroonika üldnõuded:

- esialgses versioonis peab saama osta kõiki kasutatavaid motoreid, draivereid, kodeerijaid ja juhtmeid
- kätt peab saama liigutada mootoritega niimoodi, et kasutaja saab suuna ja kiiruse ise valida
- kokku viis mootorit ühes käes
- tuvastada kui palju mootor on liikunud, näiteks kodeerijatega
- igat mootorit peab saama juhtida eraldi

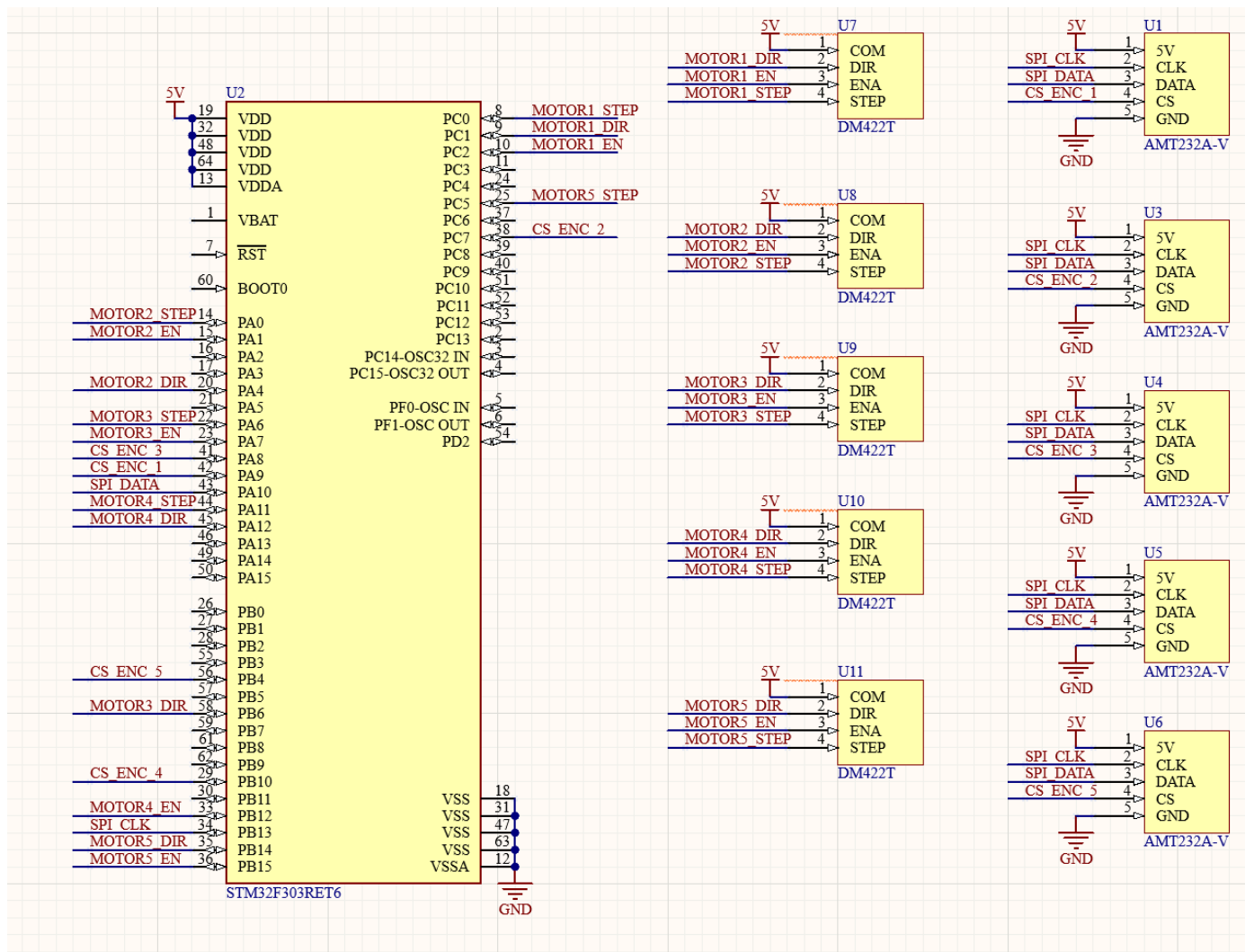
3.2 Loodud lahendus

Lõpplahenduses valmis elektroonika süsteem, mis koosnes kokku viiest Nema 17 Sammuja mootorist, viiest DM422T mootori draiverist, viiest AMT232A-V kodeerijast ja ühest mikrokontrollerist, mis testimise käigus oli nii Arduino Uno, Arduino Mega ja lõpuks NUCLEO-F303RE. Jooniselt 3.2.2 on näha, kuidas näeb välja üldine käe elektroonika skeem.

Kuna mootorite juhtimiseks on vaja draivereid, mida ilma spetsiaalse trükkplaadita saab juhtida läbi mikrokontrolleri, siis oli plaanis panna kätte ka üks mikrokontroller, mis saab käsklusi peaarvutist. Arvutiga saab läbi mikrokontrolleri liigutada mootoreid, mille tulemusena hakkab käsi liikuma.



Joonis 3.2.1. Käes kasutatud komponendid. DM422T draiver, mootor, mikrokontroller ja kodeerija



Joonis 3.2.2. Viikude ühendused mikrokontrolleri, draiveri ja kodeerija vahel.

Viie mootori juhtimiseks läheb vaja kokku 15 viiku, lisaks andmete edastamisele on vaja ka kodeerijatele nii 5V kui ka GND ühendust. Selle tulemusena sai valitud mikrokontroller Arduino Mega, millel on kokku 54 erinevat digitaalset viiku, mis võimaldab korraga juhtida kõiki viite mootorit ja nendega kaasas olevad kodeerijaid.

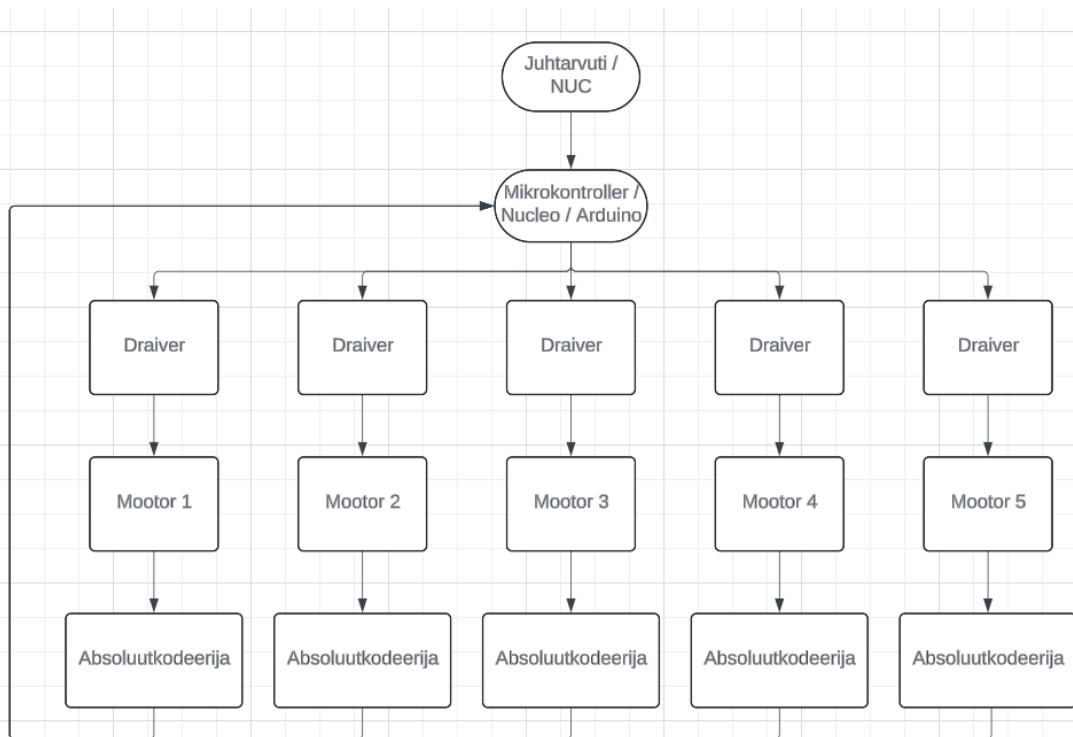
3.2.1 Mootorite draiverid

Mootori juhtimiseks on kasutusel DM542T[10] mootorite draiverid. Draiveril on kokku 8 erinevat viiku, mida saab kontrollida. Nendeks on PUL (mõndadel teistel draiveritel STEP), DIR, ENA ja ALM (Joonis 3.2.3). Igal viigul on olemas positiivne pool ja negatiivne pool, praegusel juhul said kõik negatiivsed pooled ühiselt ära maandatud. Nagu jooniselt 3.2.3 näha, siis paremal ääres keskel on mitu SWx valikut. Nende lülititega saab ära määrata kui suure sammuga mootor liigub ja kui palju voolu võimaldatakse väljastada mootorile.



Joonis 3.2.3. Mootor draiveri DM542T [10] ühendused

Jooniselt järgnevad ühendused, GND ja +Vdc annavad draiverile voolu. Prototüüpimise käigus sai tööd tehtud 24 V ja 3 A. Järgnevad ühendused, A+,A-,B+ ja B- tuleb ära ühendada vastavalt mootori faasidele. Ühendatud mootor ja draiverit on kujutatud joonisel 3.2.6.



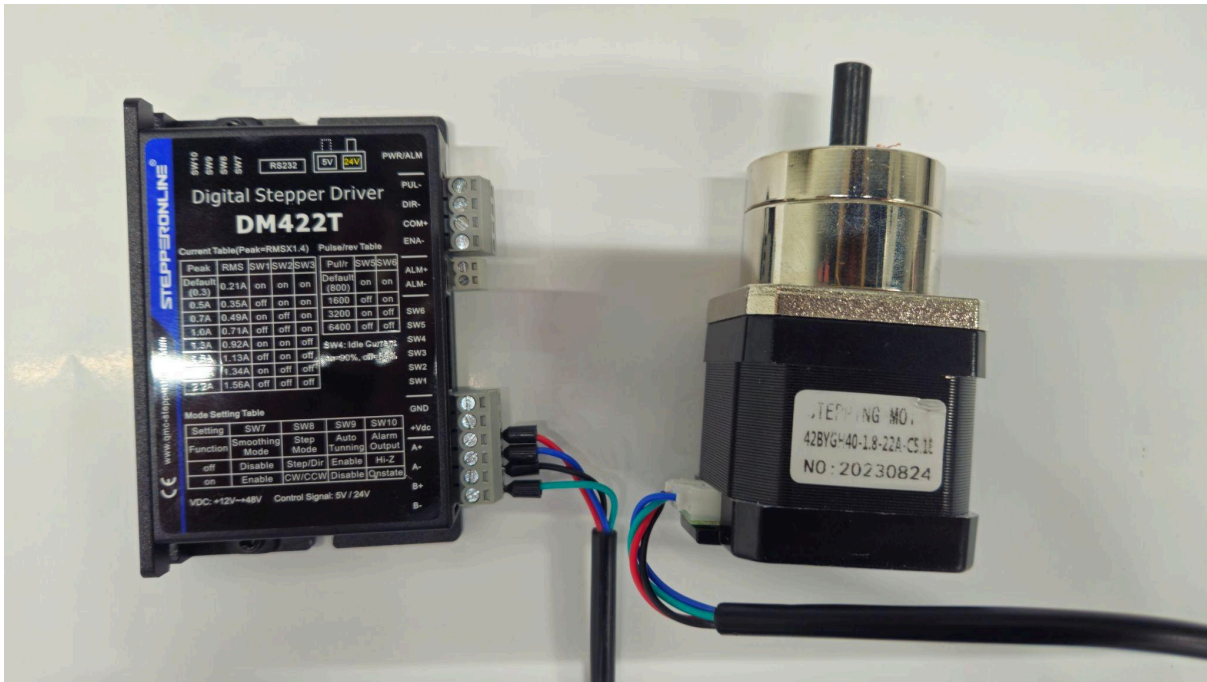
Joonis 3.2.4. Käe süsteemi komponendid

Nimelt ühe mootori ja kodeerija juhtimiseks läheb kokku kuus S/V viiku. Mootorile on vaja *step*, *direction*, *enable* ja kodeerijale *data*, *clock* ja *chip select(CS)* ühendust mikrokontrolleriga. Kodeerija kasutab SPI-ühendust, mis tähendab seda et *data* ja *clock* ühendust saab kasutada kõikidel kodeerijatel ühist, aga erinev CS ühendus, mis säästab meil viikude numbrit.



Joonis 3.2.5. *Bullet connector ühendused mootorite ja draiverite vahel*

Igast draiverist läheb mootorisse neli juhet. Need juhtmed on mootorite juurest lõigatud pooleks ja juurde paigaldatud *bullet connectorid*, mis võimaldavad lihtsalt ühendada mootorid lahti draiveritest, ilma et peab lahti võtma terve pika juhtme



Joonis 3.2.6. Mootor draiveri DM422T[11] ühendamine mootoriga

Üleüldiselt on meil kasutusel olevate mootorite juhtmete värviskeem samasugune. Punane juhe on A+, Sinine A-, Must B+ ja roheline B-.

Punane	A+
Sinine	A-
Must	B+
Roheline	B-

Tabel 1. Mootorite ühendid vastavalt värviskeemile

Kuna SemuBotil oli alguses plaanis kasutada käe liigutamiseks erineva suurusega mootoreid, siis telliti nii Nema 17 kui ka Nema 23 jaoks omad draiverid. Töö tegemise käigus ilmnas, et käte liigutamiseks piisab väiksematest mootoritest, siis läksid kasutamiseks väiksemad draiverid, kuid nendest mõlemale käele ei piisa. Valikus on suuremad draiverid, mis on mõeldud Nema 23 mootorite juhtimiseks, DM542T ja teistsugused draiverid, DM422T [11]. Teiste draiverite kasutamine on juhtimise poole pealt samasugune. DM422T peamine erinevus on see, et draiver vajab sisemise loogika jaoks eraldi 5V toidet. DM542T vajab see-eest ühist maandust, kõikide teiste viikude vahel.

3.2.2 Absoluutne kodeerija

Kodeerijatelt saab täisarvulisi andmeid, väärtuses 0-4000. Antud andmetega saab täpselt teada, kus kohas mootorid asuvad. Pärast andmete uuendamist kontrollitakse kas mootorid asuvad keelualas. Kui asuvad keelatud alas, siis mootor jääb seisma ja samas suunas enam liigutada ei saa, ainult vastassuunas.



Joonis 3.2.7. Kodeerijad AMT232A-V[12]

Kuna SemuBoti kasutatavad mootorid ei sisalda kodeerijat, siis on kodeerijad paigaldatud mootoritest eraldi. Kodeerijad on kasutusel täitmaks nõuet mootorite asukoha kohta, et mootorid ei keera kätt liiga palju ja ei kahjustaks käe mehaanikat või teisi mootoreid.



Joonis 3.2.8. Kodeerijad jaoks loodud Molex ühendus. Pruun: CS viik, punane GND, sinine 5V, roheline DATA, oranž CLK.

AMT232A-V[12] kodeerijal on kokku kuus erinevat viiku, kuid meil on kasutusel ainult 5. Kuues viik on mõeldud kodeerija programmeerimiseks, mida saab teha ostetava programmatoriga.. Toite jaoks vajab see 5V ja maandust, andmete edastamiseks DATA, CLK ja CS viike. Kuna tegemist on SPI ühendusega, siis kõikide kodeerijate peale on DATA ja CLK viigud ühised ja CS ühendused erinevad. Ühenduse skeem on nähtaval jooniselt 3.2.4. Kodeerijate andmevahetuse jaoks sai loodud eraldi kaablid, mille ühes otsas olid Molex 502578-0600[13] ühendus ja teises otsas 2.54 mm *pin header*.

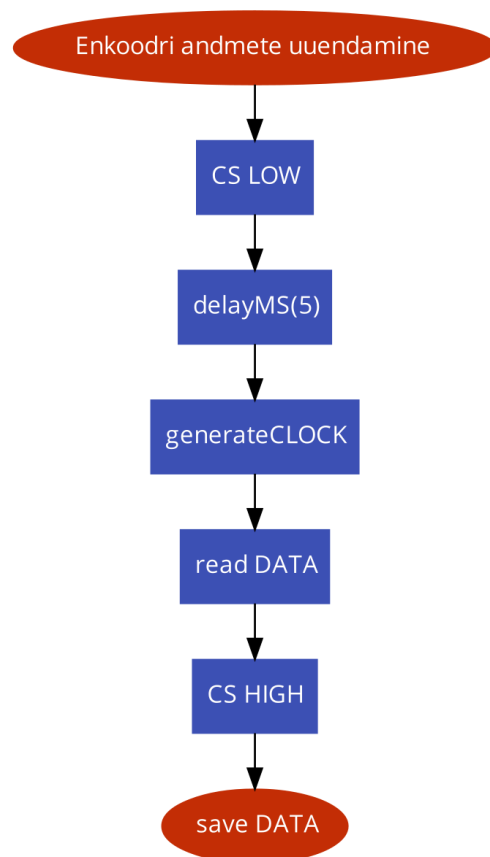
3.2.3 Püsivara

Loodud lahenduse juures on mõeldud mikrokontroller ühenduma peaarvuti külge läbi USB ühenduse. Seejärel toimub neil omavahel suhtlus läbi *serial monitori*, kust peaarvuti saadab mikrokontrollerile andmeid mootori liigutamiseks ja mikrokontroller saadab arvutile tagasi kodeerijate hetkeväärtusi.

Mootorite liigutamine mikrokontrolleriga on selliselt, et peaarvutist saadetakse mikrokontrollerile käsk, mis peab sisaldama järgmist informatsiooni:

- mootori nimetus, mida soovitakse liigutada
- päripäeva või vastupäeva liikumine
- sammude arv
- kiirus

Näiteks, kui soovitakse liigutada esimest mootorit päripäeva, 500 sammu jagu kiirusega 50, siis tuleb saata mikrokontrollerile järgnev sõne: “m1_ccw_500_50”.

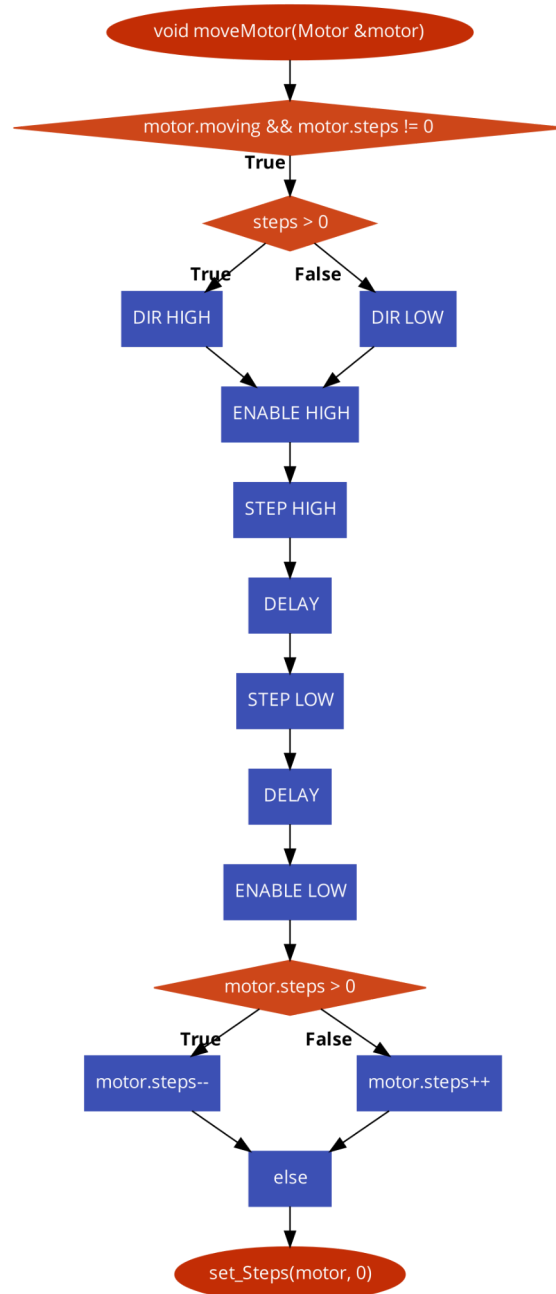


Joonis 3.2.9. Kodeerija AMT232A-V andmete uuendamine

Meie kasutatavad AMT232A-V kodeerijatelt uuendatakse andmeid koodis pidevalt. Andmete uuendamist graafiliselt saab jälgida jooniselt 3.2.9, kuid protsess käib järgmiselt:

- CS viik pannakse madalaks ja oodatakse 5 mikrosekundit
- saadetakse kodeerijatele kellsignaali
- loetakse andmete viigult andmed
- CS viik tagasi kõrgeks
- kontrollitakse bittidega, kas andmed on õiged
- andmed salvestatakse kodeerijate *structi*

Antud informatsioon salvestatakse programmeerimiskeeles C++, *struct*-idesse, mis on eelnevalt ära isikustatud iga mootori jaoks. Iga mootorile pannakse kirja mitu sammu on vaja liikuda, peale mida hakatakse iga tsükli väljakutsumise korral kontrollima, kas antud mootorit on vaja liigutada.



Joonis 3.2.10. Mootori liigutamise loogika programmeeritud

Seega koodi põhiselt käib mootori liigutamine järgmiselt:

- kontrollitakse kas mootorit on vaja liigutada alles olevate sammude järgi
- mootor lülitatakse tööle, määrates *ENA* viik kõrgeks
- sammude järgi otsustatakse, kas mootor peab liikuma päripäeva või vastupäeva, ette antakse vastavalt positiivne või negatiivne arv
- vastavalt eelnevale punktile pannakse *DIR* viik kõrgeks või madalaks
- *PUL* viik kõrgeks

- oodatakse nii mitu mikrosekundit, kui mootoril on kiiruseks antud
- *PUL* viik madalaks
- mootor lülitatakse välja, määrates *ENA* viik madalaks

Selle tulemusena tekib PWM signaal, mis on sobilik mootori draiveri etteantud nõuetele ja mootor hakkab liikuma. Visualiseeringut näeb jooniselt 3.2.10.

3.3 Mikrokontroller



Joonis 3.3.1. Vasakult paremale: NUCLEO F303RE ja Arduino Mega

3.3.1 Arduino ja Nucleo võrdlus

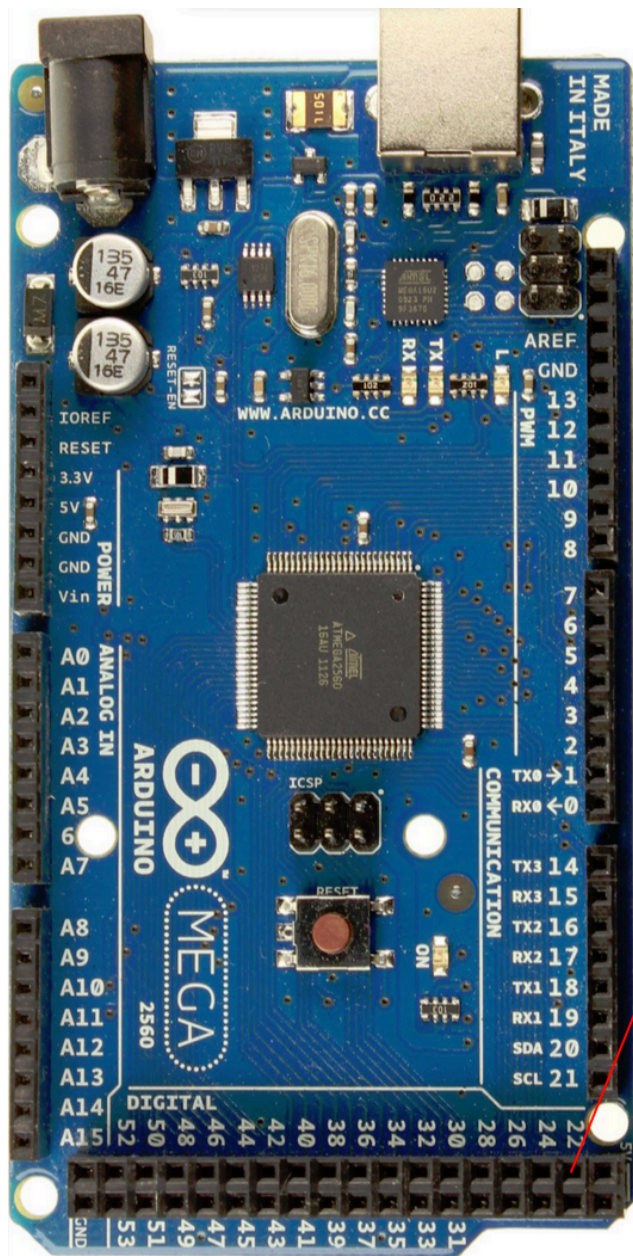
Roboti juhtimiseks mõeldud mikrokontroller peab vastama mitmetele tehnilistele nõuetele, et tagada süsteemi toimimine. Peamiselt peab see sisaldama piisavalt S/V viike, et ühendada draiverid ja tajurid mikrokontrolleriga. Selle jaoks sobivad hästi Arduino Mega ja NUCLEO F303RE.

Arduino on tuntud oma lihtsuse poolest, mis kasutab C-keelel põhinevat lihtsalt programmeerimiskeelt, millele on internetist saadaval palju erinevaid tööriistu ja õpetusi. STM mikrokontrollerid on STMicroelectronics'i poolt välja töötatud protsessorid. Neil on kasutusel 32-bitine arhitektuur, mis tähendab, et nad suudavad andmeid kiiresti töödelda, samas kui Arduino Mega[14] kasutab 8-bitist arhitektuuri. F303RE[1] töötab 72 MHz, mis on oluliselt suurem kui Arduino Mega 16 MHz. Lisaks pakub Nucleo võimsamat riistvara ja rohkem arenenud funktsioone. Näiteks F303RE-l on Cortex-M4 protsessor, võrreldes Arduino Mega Atmega2560-ga mis on vähem võimekas. Nucleol on samuti rohkem mälu ja täpsemad analoogsisend- ja -väljundkanalid.

Nucleo üheks heaks eeliseks on see, et mootorite juhtimiseks saab kasutada taimereid, mis võimaldab meil luua automaatselt PWM signaali, mis saadetakse mootori draiverisse ja mille tulemusena hakkab mootor liikuma. Iga taimer saab tööle panna erineva aja tagant, mis võimaldab meil igat mootorit erineval ajal tööle panna. Nii nagu Arduino peal, on draiverid ühendatud otse mikrokontrolleri viikude külge. Seekord on ühendused tehtud nii, et draiverite *PUL* viigud oleksid ühendatud taimerite külge, mis võimaldavad luua PWM signaali.

3.4 Loodud lahendus Arduinoga

Prototüüpimise käigus loodud esimene versioon kasutas mikrokontrolleriks Arduino Mega. Seda just sellepärast, et meil on kasutada palju S/V viike, millega saab lihtsasti mootoreid juhtida. Lisaks on Arduino prototüüpimise käigus hea lihtne mikrokontroller.

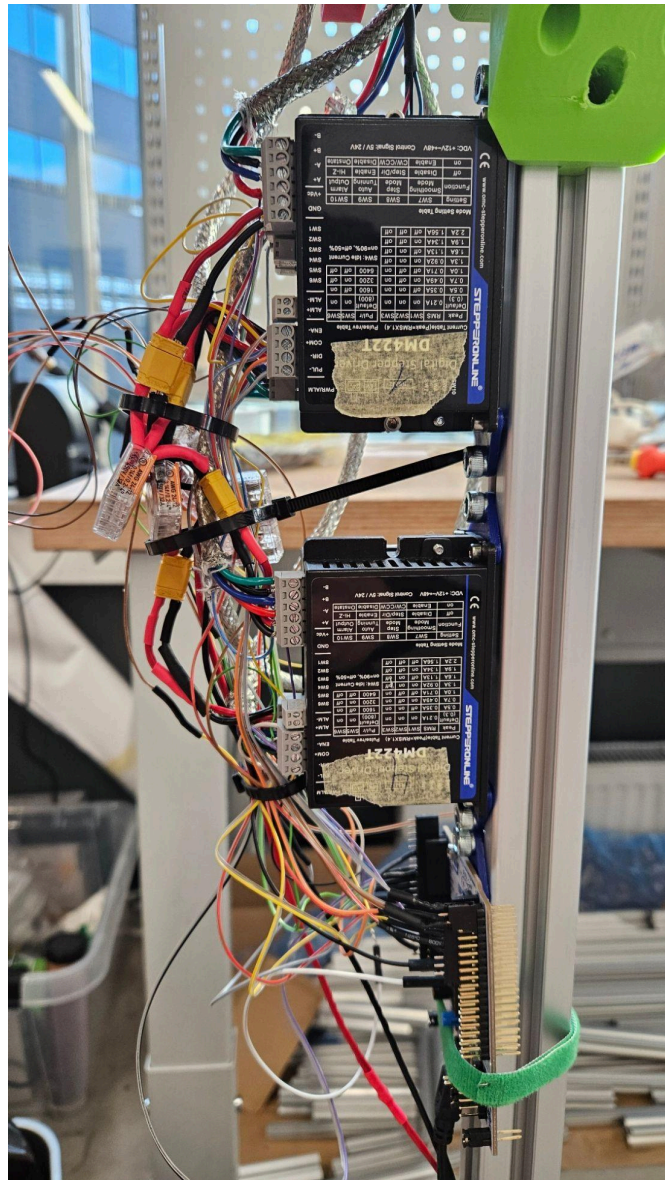


- 13 - MOTOR4_EN_PIN
- 12 - MOTOR4_STEP_PIN
- 11 - MOTOR4_DIR
- 10 - MOTOR3_EN_PIN
- 9 - MOTOR3_STEP_PIN
- 8 - MOTOR3_DIR
- 7 - MOTOR2_EN_PIN
- 6 - MOTOR2_STEP_PIN
- 5 - MOTOR2_DIR
- 4 - MOTOR1_EN_PIN
- 3 - MOTOR1_STEP_PIN
- 2 - MOTOR1_DIR

- 22 - MOTOR5_DIR
- 24 - MOTOR5_STEP_PIN
- 26 - MOTOR5_EN_PIN
- 28 - ENCODER_CS_1
- 30 - ENCODER_CS_2
- 32 - ENCODER_CS_3
- 34 - ENCODER_CS_4
- 36 - ENCODER_CS_5
- 38 - ENCODER_SCK
- 40 - ENCODER_SDO

Joonis 3.4.1. Arduino Mega[14] S/V ühendused

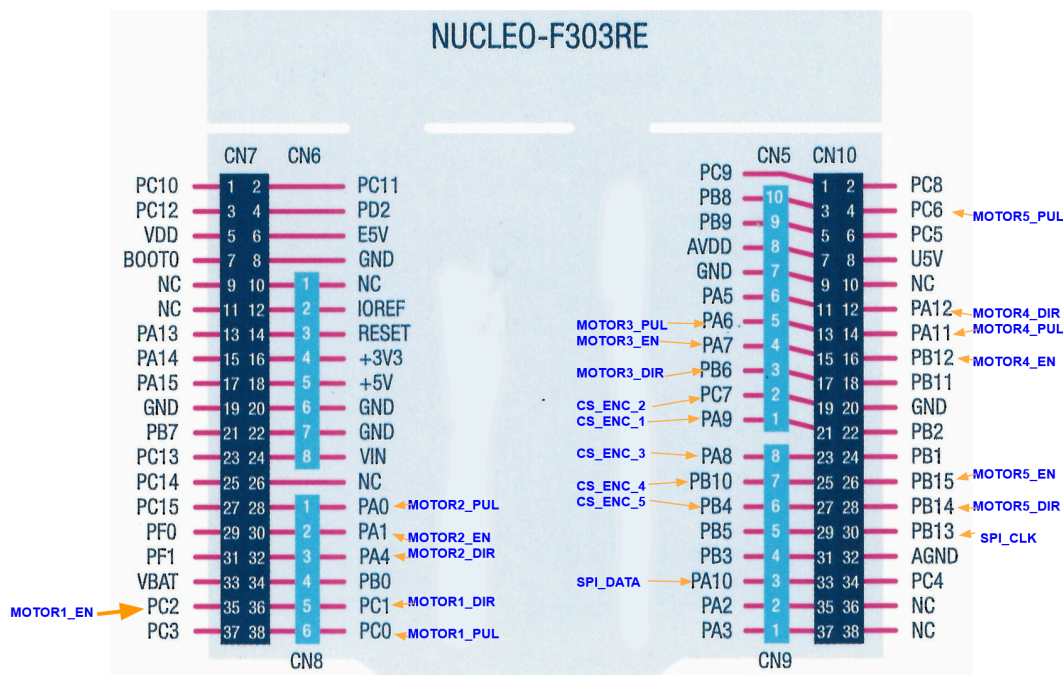
Arduino mikrokontrollerid on lihtsalt programmeeritavad, kuid võimekuse poole pealt jäävad meile võimekuse poole pealt väheks. Pärast Arduino lahenduse tegemist tuli otsus vahetada Arduino Mega STM32F303RET6 mikrokontrolleri vastu, mida saab kasutada juba valmis tehtud arendusplaadi Nucleo-F303RE-ga.



Joonis 3.4.2. Loodud juhtmed kasutusel draiverite ja mikrokontrolleri ühendamisel.

3.5 Loodud lahendus Nucleoga

Lõpptulemuses võimaldab Nucleo F303RE mootori juhtimissüsteemide loomist, kus iga mootorit saab individuaalselt juhtida taimerite abil. See annab kasutajale suurema kontrolli ja paindlikkuse, mida saab juhtida valmis kirjutatud koodiga, mis töötab peaarvuti peal ja saadab mikrokontrollerisse andmeid.



Joonis 3.5.1. NUCLEO-F303RE Viikude ühendused mootorite juhtimiseks

STM32 mikrokontrolleril on hea see, et saab kasutada taimereid, mis genereeriks meile vajaliku PWM signaali. Sealhulgas on juba sisseehitatud SPI võimekus, mis genereerib meile vajaliku sagedusega kellasignaali. Ühtlasi on Nucleo peal saadaval 51 erinevat S/V viiku, millega kõiki mootoreid juhtida.

STM32 protsessori programmeerimise liideses STM32CubeIDE[15] sai aktiveeritud iga mootori jaoks oma taimer, ehk kokku viis erinevat. Iga taimeri esimene kanal sai määratud looma PWM-i, mis loodi sagedusel 150 KHz. Mootori draiverite maksimaalne sissevõetav sagedus on küll 200 KHz, aga meil ei ole vajadust kõige maksimaalsema järgi ja seega sai veidi väiksem sagedus valitud. Teised ühendused, mis on draiveritel vaja, ehk *enable* ja *direction*, sai valitud selle järgi, et need asetseksid PWM-i toodetavatele liikudele lähedal, sest PWM-i toodetavat viiku ei saa vahetada, aga teisi saab. Seega valmis lõpuks viikude asetus, mis on nähtav joonisel 3.5.1.

Taimeritele sai määratud eelskaalajaks 479 ja arvestusperioodiks 99. Selle tulemusena genereeritakse esimesse kanalisse PWM-i, sagedusega 150 KHz. Pulsi pikkust, millal signaal on kõrge, saab teha nii, kui muudetakse taimeri esimese kanali capture/compare register 1

väärtust. Mida väiksem väärtus kirjutatakse sinna registrisse, seda kiiremini mootor liigub ja mida suurem, seda aeglasemalt mootor liigub.

Testimise jaoks sai valmis kirjutatud Python[16] programmeerimiskeeles testimisprogramm. Antud programm võimaldas kasutada DualSense[17] konsoolipulti, et liigutada igat mootorit. Selle jaoks kasutati *pydualsense*[18] teeki, millega on võimalik Pythonis kätte saada puldil olevate nuppude väärtused ja tuvastada millal juhtkange on liigutatud. Pult ühendati arvuti külge juhtmevabalt. Olenevalt mis suunas juhtkange liigutati, vastavalt sellele liigutati kindlat mootorit päripäeva või vastupäeva. Näiteks kui vasakut juhtkangi liigutada üles, siis liigutati esimest mootorit päripäeva ja kui alla, siis liikus mootor vastupäeva. Vastavalt juhtkangi asetusele, saadeti läbi USB juhtme vastavad andmed mikrokontrollerisse, mis pani mootorid liikuma.

Arutelu

Lõputöö tulemusena valmis elektrooniline süsteem, mis võimaldab liigutada robotkäe mootoreid individuaalselt. Sellegipoolest selgusid tulemust luues mõned vajakajäämised, mida saab järgmises SemuBoti versioonis paremini teha.

Lõputöös loodud lahenduses kasutatakse ainult valmistooteid, mida saab elektroonikapoest igaüks osta. Sinna hulka kuuluvad mootorite draiverid, kodeerijad ja mikrokontrollerid. Siit järgmine samm oleks luua valmis oma disainitud trükkplaat, kus asetseksid mootorite draiver ja mikrokontroller. See säästab robotis palju ruumi, sest lõputöös kasutusel olevad draiverid on mõõtmete poolest suured ja võtavad roboti kehas ruumi.

Teine puudujääk praeguses lahenduses on see, et kasutusel on hästi palju juhtmeid. Praegusel kujul on juhtmete arv liiga suur, lisaks võtavad need samuti palju ruumi ja nende haldamine on raske. Lõputöö tegijal kulus kodeerijate juhtmete tegemisele tunde, sest nende tegemine on ajakulukas ja täpne töö. Võimalikult lähedal olevad ühendused vähendavad juhtmete pikkusi ja kohti, kust võib midagi katki olla. Siis kui juhtmeid on vähem, on võimalik kasutada ka lahendusi, mida on võimalik elektroonikapoest osta.

Trükkplaadi arenduses saab luua süsteemid oma käe järgi, mitte ei pea aegunud andmelehtedest aru saama, kuidas mingisugune komponent töötab. Selle asemel saab trükkplaadi disainija ise lahti kirjutada, kuidas antud komponent töötab tema disainitud süsteemis. Nagu näiteks kasutusel olevad mootorite draiverite andmetelehed vajavad informatsiooni poolest täiendust.

Kokkuvõte

Bakalaureusetöö eesmärgiks oli välja töötada lahendus SemuBotile, mis võimaldaks liigutada roboti käsi, kasutades selleks mootoreid ja mikrokontrollereid. Valmis sai loodud kaks erinevat süsteemi, millel ühel on mikrokontrolleriks Arduino ja teisel Nucleo. Arduino mikrokontrolleri peal luuakse PWM signaal koodis manuaalselt, tõstetakse üks viik madalaks ja siis tagasi kõrgeks. Nucleo peal kasutatakse selle jaoks ära mikrokontrolleris olevaid taimereid.

Kokku on võimalik liigutada ühes käes korraga viis erinevat mootorit. Nende juhtimiseks tuleb saata peaarvutist käsk, mis võimaldab liigutada kätt. Liigutamise testimiseks kasutatakse ära mängukonsooli pulti, et nuppe vajutades saaks igit mootorit eri suunas liigutada.

Tänuavaldused

Täna lõputöö juhendajaid Karl Kruusamäe ja Veiko Vunder. SemuBoti meeskonnast soovin eraldi tänada Sander Saarpere ja Priit Rooden.

A handwritten signature in black ink that reads "Kauwk". The letters are cursive and connected, with a prominent 'K' at the beginning and a 'k' at the end.

Viited

- [1] „NUCLEO-F303RE“. [Online]. Available at: <https://www.st.com/en/evaluation-tools/nucleo-f303re.html>
- [2] A. K. Pandey, R. Gelin, ja A. Robot, „Pepper: The first machine of its kind“, *IEEE Robot. Autom. Mag.*, kd 25, nr 3, lk 40–48, 2018.
- [3] „NAO the humanoid and programmable robot.“ Vaadatud: 8. mai 2024. [Online]. Available at: <https://www.aldebaran.com/en/nao>
- [4] „POPPY HUMANOID: ADVANCED AND EASY TO USE OPEN SOURCE HUMANOID ROBOT“. Vaadatud: 20. oktoober 2023. [Online]. Available at: <https://www.poppy-project.org/en/robots/poppy-humanoid/>
- [5] „The igus Humanoid Open Platform: A Child-sized 3D Printed Open-Source Robot for Research“. Vaadatud: 25. oktoober 2023. [Online]. Available at: <https://arxiv.org/abs/1809.11110>
- [6] „The story behind the Poppy project“. Vaadatud: 20. oktoober 2023. [Online]. Available at: <https://www.poppy-project.org/en/about/>
- [7] „Poppy, a 3D-printed humanoid robot that defies conventions“. Vaadatud: 25. oktoober 2023. [Online]. Available at: <https://newatlas.com/poppy-3d-printed-humanoid-robot-kit/29497/>
- [8] „Humanoid OpenPlatform - Modular open-source robot for research purposes“. Vaadatud: 20. oktoober 2023. [Online]. Available at: <https://www.igus.eu/info/news-robocup-2014>
- [9] P. Allgeuer, H. Farazi, G. Ficht, M. Schreiber, ja S. Behnke, „The igus Humanoid Open Platform“, *KI - Künstl. Intell.*, kd 30, nr 3, lk 315–319, okt 2016, doi: 10.1007/s13218-016-0448-6.
- [10] „DM542T“. [Online]. Available at: <https://www.omc-stepperonline.com/de/digitaler-schrittmotortreiber-1-0-4-2a-20-50vdc-fuer-nema-17-23-24-schrittmotor-dm542t>
- [11] „DM422T“. Vaadatud: 6. mai 2024. [Online]. Available at: <https://www.omc-stepperonline.com/digital-stepper-drive-0-3-2-2a-12-50vdc-for-nema-11-14-17-stepper-motor-dm422t>
- [12] „AMT232A-V“. Vaadatud: 6. mai 2024. [Online]. Available at: <https://www.cuidevices.com/product/resource/amt23-v.pdf>
- [13] „Molex 502578-0600“. Vaadatud: 6. mai 2024. [Online]. Available at: <https://www.mouser.ee/ProductDetail/Molex/502578-0600?qs=3OKVfsn1b5Ax%252B4TT0aiBNw%3D%3D>
- [14] „Arduino mega“. [Online]. Available at: <https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [15] „Integrated Development Environment for STM32“. Vaadatud: 15. mai 2024. [Online]. Available at: <https://www.st.com/en/development-tools/stm32cubeide.html>
- [16] „python“. Vaadatud: 16. mai 2024. [Online]. Available at: <https://www.python.org/>
- [17] „DualSense wireless controller“. Vaadatud: 16. mai 2024. [Online]. Available at: <https://www.playstation.com/en-fi/accessories/dualsense-wireless-controller/>
- [18] „pydualsense“. Vaadatud: 16. mai 2024. [Online]. Available at: <https://github.com/flok/pydualsense>

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Kaur Kullamäe

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

“Elektroonikalahendus ja püsivara robotkäe juhtimiseks sotsiaalsel humanoidrobotil SemuBot”,

mille juhendaja on Karl Kruusamäe,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Kaur Kullamäe

20.05.2024