

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Thomas Johann Seebecki elektroonikainstituut

Tõnu Pajupuu 121093IAEB

KAABLITESTRI DISAIN

Bakalaureusetöö

Juhendaja: Eero Haldre
Diplomeeritud insener

Tallinn 2018

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Tõnu Pajupuu

20.05.2018

Annotatsioon

Käesoleva bakalaureusetöö eesmärk on tehase tootmisliinil esile kerkinud probleemi lahendamine. Probleemiks on testialal puuduv võimekus juhtmete testimiseks ja antud töös lahendatakse see probleem kaablitestri disainimise abil.

Töös pannakse esimesena paika disaini kriteeriumid ja valitakse sobiv meetod testimiseks. Peale vajaduste välja selgitamist disainitakse kriteeriumite põhjal elektriskeem ja valitakse välja sobivad komponendid. Elektriskeemi koostamise järel on kirjeldatud kuidas on testri trükkplaadid disainitud. Lisaks disaini füüsilisele poolele on kirjeldatud ka testri jaoks koostatud programmi.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti kahekümne üheksal leheküljel, kuus peatükki, üheksateist joonist, neli tabelit.

Abstract

Design of a cable tester

The aim of the following thesis is to solve a production issue that has risen to our attention. The source of the problem is the inability to individually test wires during the testing process. When assembled products fail in routine tests the issues are sometimes caused by faulty wires. Currently there is no acceptable way of checking the wires and that causes workers to make more mistakes when replacing components.

The most practical solution to this problem is a cable tester that would help workers better determine the cause of failure. This thesis finds the requirements for said tester and explains the design process that involves schematic and printed circuit board design based on the requirements.

The thesis is in Estonian and contains twenty-nine pages of text, six chapters, nineteen figures, four tables.

Lühendite ja mõistete sõnastik

LCD	<i>Liquid crystal display</i> , vedelkristallkuvar
USB	<i>Universal serial bus</i> , universaalne järjestiksiin
ADK	<i>Android development kit</i> , Androidi arenduskomplekt
IDE	<i>Integrated development environment</i> , integreeritud arenduskeskkond
LED	<i>Light-emitting diode</i> , valgusdiod
DIP	<i>Dual in-line package</i> , kaherealine pakend
CMOS	<i>Complementary metal-oxide-semiconductor</i> , komplementaarne metall-oksiid-pooljuht
ADC	<i>Analog-to-digital converter</i> , analoog-digitaalmuundur

Sisukord

1 Sissejuhatus	9
1.1 Taust ja probleem	9
1.2 Eesmärk	9
2 Vajaduste välja selgitamine	11
2.1 Tööpõhimõte.....	14
3 Elektriskeemi disain	16
3.1 Mikrokontrolleri valik [4].....	16
3.2 Mõõteahel	17
3.3 Toiteahel	21
3.3.1 Šunt takisti valik	21
3.4 Kasutajaliides.....	24
4 Trükkplaadi disain	27
5 Programmi kood	30
5.1 getButton funktsioon (Joonis 11)	30
5.2 menuSelection funktsioon (Joonis 12)	31
5.3 test funktsioon (Joonis 9).....	32
5.4 menuList funktsioon (Joonis 15)	35
5.5 compare funktsioon	35
6 Kokkuvõte	37
7 Kasutatud kirjandus	38
Lisa 1 – Testri elektriskeem.....	39
Lisa 2 – LCD mooduli elektriskeem	54
Lisa 3 – Testri fotod	56
Lisa 4 – Programmi kood	57

Jooniste loetelu

Joonis 1. Mõõtmise tööpõhimõtte skeem.	15
Joonis 2. Arduino MEGA ADK REV3 [4].	17
Joonis 3. Väljatransistor IRLD014PBF pakend ja skeemi tähis [6].	19
Joonis 4. Mõõteahela tööpõhimõtte skeem ühe kontakti näitel.	20
Joonis 5. LCD moodul DFR0009 [10].	25
Joonis 6. LCD mooduli klahvistiku tööpõhimõte.	26
Joonis 7. Põhiplaadi ülemine külg, kus asuvad Arduino, LCD moodul, transistorid ja MUXid.	27
Joonis 8. Põhiplaadi alumine külg, kus asuvad takistid ja kondensaatorid.	28
Joonis 9. Esimene sisekiht, mis on kaheks jaotatud.	29
Joonis 10. Pistikute plaat pealtvaates.	29
Joonis 11. Funktsiooni getButton tegevusdiagramm.	31
Joonis 12. Funktsiooni menuSelection tegevusdiagramm.	32
Joonis 13. Multipleksor MAX4617 kontaktide konfiguratsioon [7].	33
Joonis 14. Funktsiooni test tegevusdiagramm.	34
Joonis 15. Funktsiooni menuList tegevusdiagramm.	35
Joonis 16. Compare funktsiooni tegevusdiagramm.	36
Joonis 17. LCD mooduli ICSP ühendus, klahvistik ja LCD ekraani ühendused [13]. ...	54
Joonis 18. LCD mooduli LED indikaator, „Reset“ nupp ja piikriba ühendused [13]. ...	55
Joonis 19. Kokku pandud tester.	56

Tabelite loetelu

Tabel 1. Cirris Systems'i soovituslikud limiidid kontakti takistuse jaoks [1].....	12
Tabel 2. Erinevate juhtmete takistused.....	13
Tabel 3. Cirris Systems'i soovituslikud limiidid lühiste testimiseks [1].....	14
Tabel 4. Analoogsisendi väärtustele vastavad nupu väärtused.....	31

1 Sissejuhatus

Käesoleva bakalaureusetöö eesmärk on lahendada sagedusmuundureid komplekteerivas tehases testimisprotsessiga seonduv probleem. Probleemiks on puuduv võimekus juhtmete testimiseks ja selle probleemi lahendamiseks tuleb disainida abivahend, mida saaksid tootmistöölised testimise käigus kasutada.

1.1 Taust ja probleem

Peale sagedusmuundurite monteerimist läbivad seadmed mitmeid teste selleks, et kontrollida seadme funktsionaalsust ja koostekvaliteeti. Juhul kui testimisel tekib mõni viga peavad lõppmontaaži töötajad leidma selle põhjuse. Sageli on tegemist küllaltki keeruka ülesandega ja seetõttu tuleb ette olukordi, kus vahetatakse välja ka töökorras trükkplaate või juhtmeid. Üks põhjustest, miks vahetatakse välja töökorras komponent on see, et puudub võimalus korralikult testida tootes kasutusel olevaid juhtmeid. Hetkel on ainuke olemasolev vahend vigade otsimiseks multimeeter, millega testimine on ebamugav ja suuremate lintkaablite puhul äärmiselt aeganõudev. Samuti võib multimeetri kasutamine tekitada väga suure kasutajast tuleneva variatsiooni mõõtetulemustes. Seetõttu käsitsi kontrolli juhtmete puhul üldiselt ei teostatagi. Olukordades, kus ei saa kindel olla, kas põhjus on juhtmes või trükkplaadis vahetatakse sageli välja mõlemad. See põhjustab firmale nii rahalist kui ajalist lisakulu, mida oleks võimalik vältida.

1.2 Eesmärk

Töö eesmärgiks on testimisprotsessi osaks oleva veaotsimise võimekuse parandamine ja läbi selle rahaliste ja ajaliste kulutuste vähendamine. Selle saavutamiseks on vaja, et lõppmontaaži töötajatel oleks võimalik vigade otsimisel veenduda ka juhtmete töökorras. Kõige praktilisem viis selle saavutamiseks on kaablitester, mis võimaldaks antud tootes kasutusel olevate juhtmete testimist. Kuna tootes kasutatavatel juhtmetes

on kasutusel küllaltki lai valik erinevaid pistikuid, siis pole võimalik kasutada müügis olevaid standartseid testreid, vaid tuleb disainida spetsiaalne tester.

2 Vajaduste välja selgitamine

Hetkel on tootmise käigus avastatud defektsete juhtmete puhul olnud peamiselt kolm probleemi. Kõige sagedasemad defektid on lühis või halb kontakt, kuid jõukaablitel esineb ka ristühenduse probleemi. Esineb ka lühiseid kontaktide vahel, mis ühenduses olla ei tohiks. Lühiste puhul küll tootmises olev tester suudab vea tuvastada, kuid oleks vaja võimalust kontrollida kõrge ebavajalike ühenduste olemasolu juhtme peal eraldi. Peamiseks probleemide allikaks on lintkaablid, mille puhul võivad lühised ja halvad kontaktid väga kergelt tekkida. Kõiki kolme välja toodud viga on võimalik leida takistuste mõõtmise abil.

Järgnevalt on välja toodud kriteeriumid, millega tuleb disainiprotsessi juures arvestada. Esimene ja peamine nõue, millest disainil lähtuda, on lihtsus nii testimisel kui ka parandamisel kui selleks peaks tulevikus vajadus tekkima. Tester läheb kasutusse tootmisliinile, kus kasutavad seda tootmistöölised. Tehasetöölise näol on sageli tegemist elektrilase hariduseta isikutega ja seetõttu peaks tester olema väga lihtsalt kasutatav, et vältida testri valesti kasutamise või tulemuste valesti tõlgendamise võimalust. Komponentide valikul tuleks lähtuda nõudest, et võimalusel oleks komponendid kergesti vahetatavad ja tuleks kasutada juba olemas olevaid vahendeid. Tõenäoliselt tuleb tulevikus mehhaanilise või elektrilise kahjustuse tagajärjel ette ka vajadust komponente plaadi peal välja vahetada. Kergesti vahetatavate komponentide all on mõeldud komponente, mis on vahetatavad ilma jootekolbi kasutamata. See võimaldab tootmistöölistel ka iseseisvalt komponente välja vahetada kui viga on teada ja säästab selle arvelt nii montööride kui inseneri aega. Ka trükkplaadi eluiga on pikem kui komponente korduvalt peale ja maha ei joodeta.

Teiseks kriteeriumiks on universaalsus. Testri disain peaks olema selline, et hilisemalt on võimalik seda rakendada ka teistel tootmisliinidel. Erinevatel tootmisliinidel on kasutusel väga erinevad juhtmed ja pistikud, siis peaks pistikute osa olema testri põhiosast eraldatud. Lõplikku testrit peab olema selline, et on vajadusel võimalik lisada testimiseks uusi juhtmeid ja uusi pistikuid ilma, et oleks vaja muuta kogu disaini.

Kolmandaks küsimuseks on testi parameetrid. Alustuseks peab valima testi jaoks takistuse limiidid, mille põhjal otsustada kas juhe on defektne või mitte. Limiitide valimisel tuginen Cirris Systems'i [1] juhistele. Terve ühenduse kontrollimise testi jaoks on Cirris Systems välja pakkunud kolm kategooriat, mille vahel saab valida sõltuvalt sellest kui ranget testi soovitakse (Tabel 1). Kui juhtme mõõdetud takistus jääb alla määratud limiidi, siis tunnistatakse juhe testi läbinuks. Juhul kui takistus ületab määratud limiidi tunnistatakse juhe vigaseks.

Tabel 1. Cirris Systems'i soovituslikud limiidid kontakti takistuse jaoks [1].

Range test	Hea test	Keskpärase test
1,05% juhtme takistusest + 0,5Ω	Suurem väärtus arvudest 2Ω või 1,10% juhtme takistusest + 1Ω	Suurem väärtus arvudest 5Ω või 1,20% juhtme takistusest + 2Ω

Antud töö eesmärgiks on küllaltki lihtne tester, mis on mõeldud kasutamiseks abivahendina. Testri abil on vaja leida juhtmed, mis on põhjustanud testimisel vea. Kui juhe põhjustab juba testi läbi kukkumise, siis on defekt üldiselt lihtsalt leitav nii, et kolmest kategooriast osutub valituks keskpärase test, mille limiidid on kõige kõrgemad. Keskpärase testi limiidi määramisel on antud kaks varianti, mille hulgast peab valima kahest väärtusest suurema. Väärtus 5Ω on juba antud, kuid selleks, et teine võimalik limiit saada tuleb esmalt leida juhtmete tõeline takistus. Selleks kirjutan välja valemi, mille põhjal arvutada teine võimalik limiit. Muutuja r_{real} tähistab juhtme takistust ja muutuja x tähistab arvutatavat limiiti.

$$x = (r_{real} * 0,0120) + 2 \quad (1)$$

Juhtmete takistuste määramiseks on kõige lihtsam variant koostada tabel vajalike andmetega (Tabel 2). Reaalse takistuse leidmiseks on vaja kontakti ja kaabli takistust. Tõeliste väärtuste leidmine pole võimalik kuna puudub seade, millega oleks võimalik nii väikseid takistusi täpselt mõõta. Arvutuste tegemisel on kasutatud teoreetilisi väärtusi. Takistused tuleb leida kõigi unikaalsete juhtmete puhul. Juhtmetel, mille kontaktid ja kaabel on samad võib jätta välja kõik peale kõige pikema juhtme. Arvutuste abil otsime suurimat juhtme takistust ja lühemate juhtmete puhul on juba teada, et takistus on väiksem. Alles jääb üheksa erinevat juhet. Juhtmete takistuse arvutamiseks

on kasutatud Cirris Systems'i poolt loodud takistuse leidmise kalkulaatorit [2]. Pistikute puhul kasutan tootja poolt antud kontakti maksimaalse takistuse väärtuseid. Kaablikingade puhul selliseid väärtuseid pole tootjal võimalik anda, sest kontakti takistus sõltub sellest kuidas on kaabliking kliendi poolt pressitud. Seetõttu kaablikingade puhul tuleb lihtsalt võtta kontakti takistuseks keskmine väärtus teiste kaablite põhjal.

Tabel 2. Erinevate juhtmete takistused.

Juhtme Nr.	Pistiku kood	Kontakti takistus	Pistiku kood	Kontakti takistus	Kaabli pikkus ja läbimõõt	Kaabli takistus	Takistus kokku
1	3M 89134-0103	10 mΩ	3M 89134-0103	10 mΩ	110 mm, 28 AWG	6 mΩ	26 mΩ
2	TE 1-1827863-0	20 mΩ	TE 1-1827863-0	20 mΩ	700 mm, 26 AWG	37 mΩ	77 mΩ
3	TE 1-794617-2	20 mΩ	VHR-3N	20 mΩ	1105 mm, 22 AWG	59 mΩ	99 mΩ
4	TE 1445022-4	20 mΩ	TE 1445022-5	20 mΩ	90 mm, 22 AWG	5 mΩ	45 mΩ
5	TE 1445022-2	20 mΩ	M5 kaabliking	20 mΩ	435 mm, 22 AWG	23 mΩ	63 mΩ
6	TE 794617-2	20 mΩ	M6 kaabliking	20 mΩ	440 mm, 22 AWG	23 mΩ	63 mΩ
7	TE 1445022-3	20 mΩ	M4 kaabliking	20 mΩ	405 mm, 22 AWG	21 mΩ	61 mΩ
8	TE 1-794617-0	20 mΩ	TE 1-794617-0	20 mΩ	115 mm, 22 AWG	6 mΩ	46 mΩ
9	M5 kaabliking	20 mΩ	M5 Kaabliking	20 mΩ	280 mm, 22 AWG	15 mΩ	55 mΩ

Nüüd kui juhtmete takistused on leitud saab valemi (1) abil leida väärtuse, mida tuleks kasutada testi limiidina. Võrdluse jaoks on vaja leida ainult suurima takistusega juhtme tulemus. Kõrgeim takistus on kolmandal juhtmel. Järelikult piisab alustuseks kui arvutuse teen ainult kolmanda juhtme andmete põhjal.

$$x = (0,099 * 0,012) + 2 = 2,0012 \Omega$$

Tulemusest on näha, et lühikeste kaablite puhul on juhtmete enda takistus niivõrd väike, et ei mõjuta arvutuse tulemust peaaegu üldse tulemust. Kõige pikema kaabli puhul oli takistus liiga väike, mis tähendab, et teiste juhtmete puhul pole arvutusi vaja teha. Testi limiidiks saab 5 Ω .

Lisaks limiidile, mille abil panna paika terve kontakti takistusvahemik, on vaja leida sobiv limiit, mille põhjal otsustada, kas tegemist on lühisega. Cirris Systems on ka lühiste testi jaoks pakkunud välja kolm kategooriat (Tabel 3), mis aitavad valida testi jaoks limiidi [1]. Lühiste puhul piisab samuti keskpärasest testist, mis seab limiidiks 100 k Ω . Lühise testi puhul kontrollitakse ühendust kahe kontakti vahel, mis omavahel ühenduses olla ei tohiks ja limiidi abil kontrollitakse kahe kontakti vahelist isolatsioonitakistust. Juhul kui takistus peaks alla olema väiksem kui 100 k Ω loetakse seda juba lühiseks.

Tabel 3. Cirris Systems'i soovituslikud limiidid lühiste testimiseks [1].

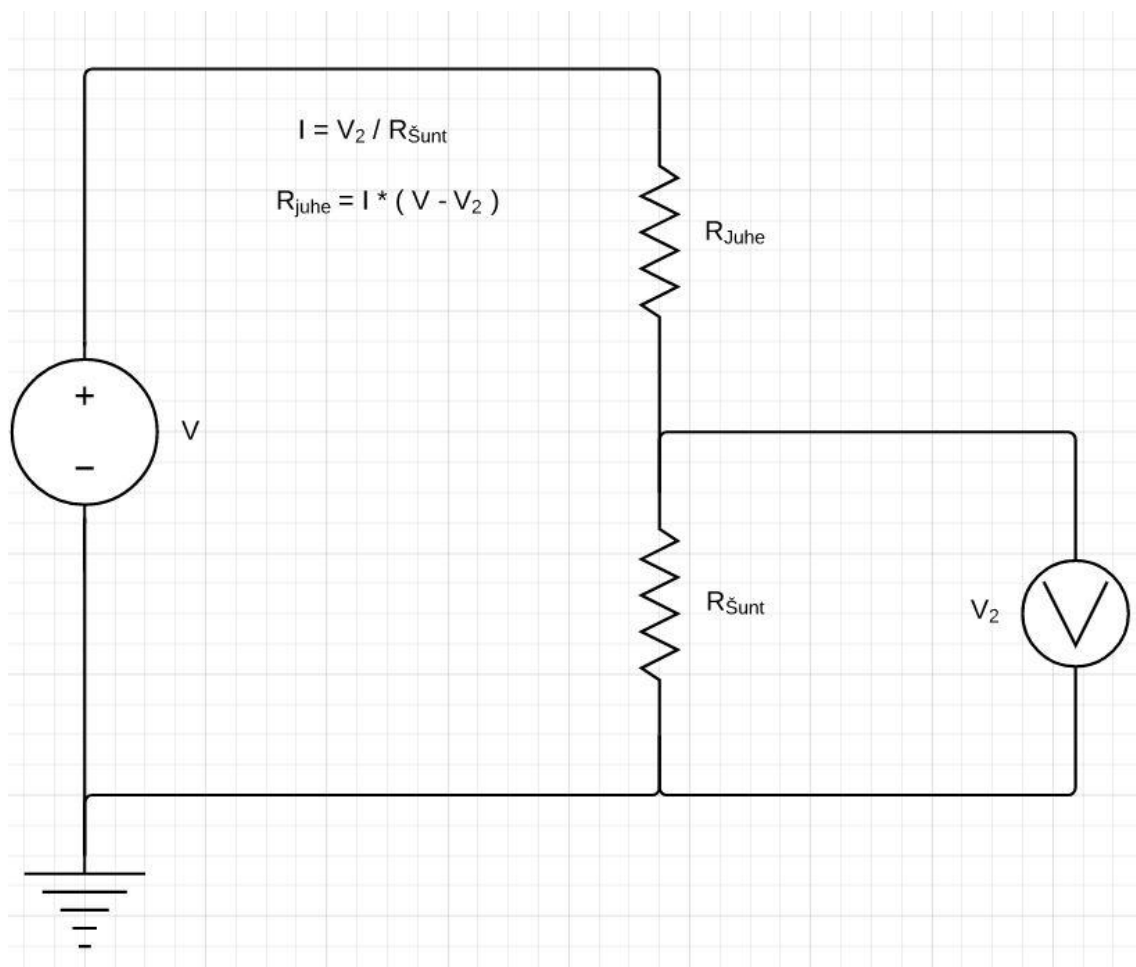
Range test	Hea test	Keskpärane test
5 M Ω	500 k Ω	100 k Ω

2.1 Tööpõhimõte

Uurides turul müügil olevate pidevuse testrite tööpõhimõtet, siis kerkib esile peamine meetod, mida kaablitestrite puhul kasutatakse. Testris on kasutusel konstantse voolu allikas. Juhe ühendatakse testri vooluallikaga ja mõõdetakse juhtme peal tekkivat pinget. Saadud tulemuse põhjal arvutab tester juhtme takistuse Ohm'i seaduse abil. Sellise tööpõhimõttega testrid jagunevad kahte kategooriasse. Odavamate testrite puhul kasutatakse kahe juhtme meetodit, mis tähendab, et pingelangu mõõdetakse kogu ahelas ja mõõdetakse ka testri enda ahelas tekkivat pingelangu. See pärsib mõõtetulemuste täpsust väiksemate takistuste puhul. Suurema täpsusega testrites lahendatakse see probleem nelja juhtme meetodi abil. Sellisel juhul on voltmeetri ühendused eraldi välja toodud ja mõõdetakse pingelangu ainult juhtme enda peal. Pingelang voltmeetri

kontaktide peal on väikse voolu tõttu piisavalt väike, et seda mitte arvestada. Nelja juhtme meetod on vajalik kui soovitakse täpselt mõõta takistusi, mis jäävad alla 1Ω . Eelmises peatükis seatud limiitide puhul piisab kui kasutada kahe juhtme meetodit [3].

Olemasolevate vahendite rakendamiseks tuleks käeoslevas testris vältida konstantse vooluallika kasutamist. Võttes vooluallika asemel kasutusele pingevallika on võimalik kasutada disainis keskse komponendina olemasolevat Arduino arendusplaati. Ka pingevallikaga on küllaltki lihtne mõõtetulemus saada. Ühendades järjestikku kaabli takistusega teise takistuse saab samamoodi teha arvutused Ohm'i seaduse abil, et leida kaabli takistuse voolu väärtuse abil (Joonis 1). Mõõtes pingelangu teada oleva takistuse pealt saab arvutada ahelas oleva voolu. Järjestikku ühendatud takistuste puhul läbib mõlemat takistust võrdne vool ja kui on teada juhete läbiva voolu väärtus, siis teades takisti peal olnud pingelangu saab arvutada ka pingelangu kaabli peal. Sellest on võimalik omakorda Ohm'i seaduse abil juhtme takistus kalkuleerida.



Joonis 1. Mõõtmise tööpõhimõtte skeem.

3 Elektriskeemi disain

Üks tingimus testri disaini juures oli olemasolevate vahendite kasutamine. Töökohal on võimalus kasutada arendusplaate, mille põhjal ka teisi testseadmeid on juba loodud. Sellel põhjusel baseerub ka käesoleva testri disain Arduino arendusplaadil. Skeemi mõõtmise tööpõhimõte vajab toimimiseks pingesallikat, takistit ja voltmeetrit. Pingesallikaks saab olema arendusplaadi enda pingeväljund ja šunt takisti peal pingelangu mõõtmiseks kasutan voltmeetri asemel plaadi analoogsisendit.

Kaablite täielikuks testimiseks on vaja ükskhaaval igast juhtmest signaal läbi lasta ja kontrollida vastaspoolel, kas ja kuhu signaal jõuab. Elektriskeem peab võimaldama ühelt poolt lülitada sisendpinget kõikide juhtme kontaktide vahel ja samamoodi lülitada analoogsisendisse pinget ükskhaaval kõigi juhtme sisendkontaktide pealt. Tootmisliinil kus kaablitester kasutusele võetakse on suurim ühenduste arv lintkaablil, millel on nelikümmend ühendust. See tähendab, et tester peab suutma lülitada ahelasse mõlemal pool nelikümmend erinevat kontakti.

Järgmine asi, mille peale tuleks mõelda on kasutajaliides. Tegemist ei ole automaatse testimisega ja on vaja anda testijale ka tagasiside testi tulemusest. Selles olukorras tuleb kaaluda erinevate lahenduste praktilisust ja leida tasakaal, et kasutajaliides oleks kasutaja jaoks võimalikult lihtsasti arusaadav, aga samas võimalikult informatiivne.

Toiteahelale mõeldes tuleks arvestada, et kõik võiks toimida läbi arendusplaadi, mis tähendab, et komponente ja testi pinget ja tarbitavat voolu peab valima selliselt, et mikrokontroller suudaks seda ka pakkuda.

3.1 Mikrokontrolleri valik [4]

Arendusplaadi valik on küllaltki lihtne seetõttu, et eesmärgiks on kasutada juba töökohal olemasolevat mikrokontrolleri plaati Arduino Mega ADK (Joonis 2).

Tegemist on ühega paindlikumatest mikrokontrolleri plaatidest, mida Arduino toodab ja see põhineb Atmeli Atmega2560 mikrokontrolleril [5]. Mõningad peamised parameetrid, mida disaini juures ka vaja läheb on:

- 54 digitaalset sisendit/väljundit
- 16 ADC sisendit 10-bitise resolutsiooniga
- USB ühendus
- Soovitatav sisendpinge 7-12 V
- 40 mA maksimaalne alalisvool digitaalsete sisendite/väljundite puhul
- Maksimaalne 750 mA väljundvool toiteadapteriga ja 500 mA USB toitega
- 256 KB välmälu



Joonis 2. Arduino MEGA ADK REV3 [4].

Plaadi programmeerimiseks on Arduino välja andnud oma tarkvara (IDE), mis võimaldab arvutis programmi koodi valmis kirjutada ja lihtsalt USB kaudu ühendades programmi üles laadida. Arduino kasutamine aitab väljundite lülitamiseks vajaliku loogika küllaltki lihtsalt teostada.

3.2 Mõõteahel

Alustuseks tuleb mõõtmise põhimõtte skeem sobitada Arduinoga kasutamiseks. Vajaliku pingesallikana tuleb kasutusele Arduino 5V pingeväljund. Pingelangu mõõtmise saab teostada arvutuste abil analoogsisendit kasutades. Juhtmete enda takistused on äärmiselt väikesed võrreldes analoogsisendi omaga, mis antud olukorras täidaks šunt takisti rolli. Sellise skeemi puhul jääks praktiliselt kogu pingelang

analoogsisendi peale ja väikseid muutuseid juhtme takistuses oleks pea võimatu tuvastada. Takistuse vähendamiseks saab paigutada analoogsisendiga paralleelselt šunt takisti, mis oleks oluliselt väiksema takistusega. Sellega kaasneb kaks muutust. Esimene ja positiivne muutus on see, et nüüd on pingelang juhtme peal märgatavalt suurem ja kaabli takistuse kasvamist on võimalik palju täpsemalt mõõta. Teine ja negatiivne tulemus on see, et mida väiksema šunt takisti valime, seda väiksemaks muutub ahela kogutakistus ja seda suuremaks läheb vool. Nüüd tuleks leida sobiv takistus, mis pakuks piisavalt täpsust arvutuste jaoks ja ei ületaks Arduino väljundvoolu limiite. Enne kui neid arvutusi teha saab tuleb läbi mõelda ka ülejäänud mõõteahela osad.

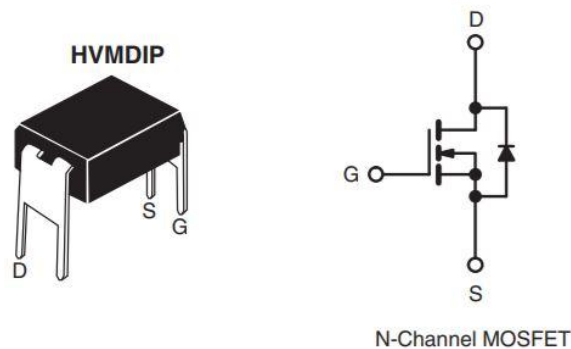
Järgmine aste skeemi disaini juures on ahela lülitusosa. Selleks, et kõik veavariandid saaksid testi käigus leitud on vaja kontrollida, kas ühendus on olemas kahe kontakti vahel, mis omavahel ühenduses peavad olema ja samuti kontrollida, et ei oleks lühiseid kõrval olevate kaablitega. Järelikult on vaja sisendi ja väljundi poolt eraldi lülitada. Selle jaoks vaja võimekust mõlemal pool lülitada neljakümme kanalit.

Arvestades iga lüliti jaoks eraldi komponendi, siis tähendab see seda, et trükkplaadile peab juba ainult selle jaoks mahutama kaheksakümmend komponenti. Komponentide valiku juures peaks jälgima ka seda, et komponentide suurus olema võimalikult väike. Üks kriteeriumeid oli komponendid, mida oleks võimalik vahetada ilma plaadi pealt maha jootmata, järelikult tuleks lülituskeemi disainil juba lähtuda sellest, milliseid sobivaid komponente selleks leidub. Mõeldes mõõteahela takistuse peale, siis antud olukorras oleks hea kasutada lülitamiseks releesid, kuid välja toodud tingimuste tõttu sobivat releed leida ei õnnestunud. Kõige paremini sobiv komponent õnnestus leida hoopis väljatransistoride hulgast.

Valituks osutus komponent IRLD014PBF (Joonis 3) [6]. Tegemist on n-kanali ja isoleeritud paisuga väljatransistoriga. Tähtsaimad omadused, millele valiku juures tähelepanu sai pööratud on järgnevad:

- DIP pakend, mida on võimalik soklisse asetada ja lisaks saab komponente otsapidi kokku laduda. See tähendab, et näiteks 16-DIP soklisse saab panna neli transistori.
- Maksimaalne võimsuserladus 1,3 W.

- Paisu juhtimine loogikanivoodega.
- Maksimaalne neelu-lätte pinge 60 V.
- Maksimaalne neelu vool 1,7 A.
- Paisu-lätte lävepinge maksimaalselt 2V.
- Maksimaalne neelu-lätte takistus 0,20 Ω .

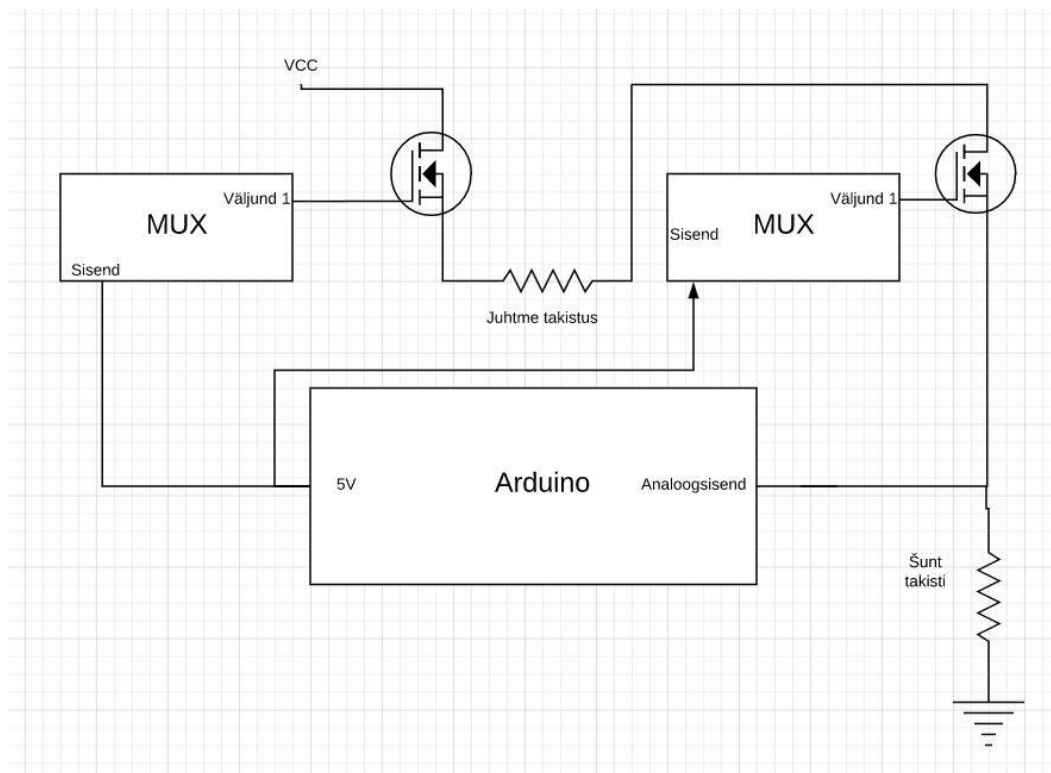


Joonis 3. Väljatransistor IRLD014PBF pakend ja skeemi tähis [6].

Vaatamata suurele hulgale digitaalväljunditele ei ole siiski Arduinol piisavalt väljundeid, et kõiki transistore juhtida ja seetõttu on vaja eraldi komponente, mille abil saaks transistore lülitada. Kokku on vaja juhtida kaheksakümmend transistori ja seetõttu sobiks hästi 8-kanaliga multipleksorid, mille abil saaks täpselt õige arvu kanaleid. Valituks osutus CMOS analoogmultipleksor MAX4617 [7]. Peamised parameetrid, millele valikul tähelepanu pöörata antud disaini juures on:

- Samuti DIP pakend, mida on võimalik soklisse asetada.
- Sobilik loogikanivoodega juhtimiseks.
- Toitepinge vahemik 2 V - 5,5 V.
- Sisselülitusaeg maksimaalselt 15 ns.
- Väljalülitusaeg maksimaalselt 10 ns.
- Maksimaalne vool kõigi terminalide puhul 75 mA.

Lisades algsesse mõõteahelasse sisse ka multipleksorid ja transistorid saame lõpliku mõõteahela (Joonis 4), mille põhjal on võimalik teha arvutused, et leida sobiv suurus šunt takisti jaoks.



Joonis 4. Mõõteahela tööpõhimõtte skeem ühe kontakti näitel.

Selliseks pidi mõõteahela skeem jääma, kuid hiljem kui juba prototüübiga esimesed katsed sai tehtud, ilmnas viga, mis skeemi disaini käigus jäi märkamata. Ühendades kaks transistori järjestikku oleks tulnud arvestada faktiga, et sel juhul pole ülemise transistori paisu-lätte pinge enam 5 V maa suhtes, vaid 5 V pinge VCC suhtes. Seetõttu ülemine transistor ei avane nii nagu vaja. Kõige lihtsam viis selle probleemi lahendamiseks on tuua allapoole VCC pinge, mis algselt pidi samuti 5 V olema. Arvestades, et transistori paisu-lätte lävepinge on maksimaalselt 2 V, siis piisab kui VCC pinge viia 5 V pealt 2,56 V peale. See võimaldab transistorid kindlalt avada ja samuti on Arduino puhul võimalik määrata analoogsisendi võrdluspingeaks plaadi sisemine 2,56 V [8].

Viimase asjana tuleb veel lisada vajalikud takistid ja kondensaatoreid. Lahtisidestamise kondensaatorid suurusega 100 nF on lisatud kõikide multipleksorite toitekontaktide juurde. Iga transistori paisu ja multipleksori digitaalsisendi juurde tuleb lisada ka mahalaadimise takisti selleks, et singaal ei jääks õhku rippuma. Selle jaoks piisab kui

panna takistid suurusega 10 k Ω . Samuti ühendan mahalaadimise takisti ka kõigi Arduino digitaalväljundite külge, mis kasutusele ei lähe. Nende puhul siiski iga väljundi jaoks eraldi takistit ei ole praktiline kasutada ja kõrvuti olevad väljundid jagavad takistit selleks, et komponentide arvu natuke vähendada. Järjestikku on ühendatud 1 k Ω takistid ka kõikide kasutusel olevate Arduino digitaalväljunditega selleks, et kaitsa neid liigse voolu eest.

3.3 Toiteahel

Arduino toite jaoks saab kasutada nii 5 V pinget arvuti USB pordist kui ka eraldi toiteadapterit. Testri kasutamiseks on parem kui alati ei pea arvuti kõrval olema ja seetõttu tuleks kasutusele võtta eraldi toiteadapter. Soovitatud toitepinge Arduino jaoks on 7-12 V [4]. Toiteadapterina sobib kasutada standartset 12 V AC-DC adapterit. Kasutusele tuleb adapter maksimaalse väljundvooluga 2 A, mis on rohkem kui piisav testri vajaduste täitmiseks.

Arduino 5 V väljundpinge peal hakkavad toimima LCD moodul, multipleksorid ja ka LED indikaator, kuid mõõteahela ülemise poole transistoride neelu peale on vaja anda pinge väärtusega 2,56 V. Arduino plaadi pealt on võimalik saada ainult 5 V ja 3,3 V väljundpingeid. Puudu oleva pingeniivo saamiseks tuleb kasutada lisakomponente ja lihtsaim viis selleks on Arduino 5 V väljundpinge alla toomine DC-DC pingemuunduri abil. Valituks osutus pingeregulaator PTH08080WAH [9]. Tegemist on reguleeritava pingeregulaatori mooduliga, mis vajab kasutamiseks ainult ühte lisatakistit. Tavalise takisti asemel on parem kasutada potentsiomeetrit. See võimaldab vajadusel võimaluse väljundpinget reguleerida ilma takistit vahetamata. Väljundpinge reguleerimise abil on võimalik soovi korral transistoridel tekkivat pingelangu vähendada ja selle arvelt analoogsisendisse jõudvat väärtust suurendada.

3.3.1 Šunt takisti valik

Šunt takistiks on esialgu valitud 15 Ω väärtusega ja kuni 1% variatsiooniga takisti. Soovitud täpsus mõõtmistulemuste jaoks on 1%, mis 5 Ω limiidi puhul annab veaks 0,05 Ω ja 100 k Ω limiidi puhul annab aksepteeritavaks veaks 1 k Ω . Takisti sobivust tuleb kontrollida arvutuste abil. Esimesene leitav väärtus on mõõteahela lisatakistus, mis tuleks mõõtmise tulemusest maha arvata enne testi tulemuse osas otsuse langetamist. Peamise lisatakistuse annavad mõõteahelasse kaks transistori, aga mõjutajateks on ka

DIP soklid ja piikribad. Transistorid sisestatakse trükkplaadi peal DIP soklitesse ja selline ühendusviis põhjustab väikse lisatakistuse sarnaselt pistikute kontakti takistusele. Väikse kontaktitakistuse lisavad ahelasse veel piikribad, mille abil pistikute trükkplaat ja ka Arduino põhiplaadi külge ühendatakse. Väiksema veaprotsendi saavutamiseks peab takistused üle mõõtma. Mõõtmise jaoks on pistikute plaat eemaldatud ja ühendus tehtud otse piikribade kontaktide vahel, mis võimaldab pistikute takistused ahelast eemaldada. Multimeeteriga mõõtmine annab sellise mõõteahela tulemusteks analoogsisendi peal pinge 2,311 V ja mõõteahelas pinge 0,249 V. Mõõtes lisaks veel üle šunt takisti saame väärtuseks 15,3 Ω ja on võimalik arvutada vool.

$$I = \frac{V_{\text{Sisend}}}{R_{\text{šunt}}} = \frac{2,311}{15,3} = 0,151 \text{ A}$$

Voolu ja mõõteahela pinge abil on leitav ka mõõteahela takistus. Mõõteahela takistuse leidmiseks tuleb leitud pingest veel lahutada väljundit ja sisendit ühendava juhtme peal olev pingelang. Juhtme peal oleva pingelangu väärtuseks annab multimeeter 0,009 V.

$$R_{\text{Mõõteahel}} = \frac{V_{\text{Mõõteahel}}}{I} = \frac{0,249 - 0,009}{0,154} = 1,558 \Omega$$

Kontrollides samade mõõtmistega ka teiste kanalite takistusi on näha, et variatsioon on minimaalne ja arvatud takistus sobib kasutamiseks kõigi kanalite puhul. Saab teha järelduse, et mõõteahela takistuse kompenseerimiseks tuleb uueks kontakti limiidiks panna 6,558 Ω .

Mõõteahela takistuse teadmine võimaldab välja arvutada ka väärtused soovitud limiitide jaoks. Teades, et šunt takisti väärtus on 15,3 Ω ja ahela lisatakistus on 1,558 Ω saab leida ka kogutakistuse.

$$R_{\text{Kogu}} = R_{\text{šunt}} + R_{\text{Mõõteahel}} = 15,3 + 1,558 = 16,858 \Omega$$

Kasutades lisaks kogutakistusele ja šunt takistusele Arduino analoogsisendi ADC väljundväärtust mõõdetud sisendpinge juures saab kasutades tuletada valemi, mille abil on võimalik leida ADC väljundväärtus kui juhtme takistus on 5 Ω .

$$\frac{R_{\text{šunt}}}{R_{\text{Kogu}}} = \frac{ADC_{\text{šunt}}}{ADC_{\text{Kogu}}} \Rightarrow ADC_{\text{šunt}} = \frac{R_{\text{šunt}} * ADC_{\text{Kogu}}}{R_{\text{Kogu}}}$$

Lahutades saadud ahela kogutakistuse ADC väljundväärtuse 715 juures ahela kogutakistuse ADC väljundväärtuse 716 juures saame takistuse, mille peale ADC väljundväärtus langes ühe ühiku võrra.

$$R_1 = R_{715} - R_{716} = 21,912 - 21,858 = 0,054 \Omega$$

Leitud väärtus näitab, et teoorias peaks tester tunnistama juhtme vigaseks kui takistus läheb $0,054 \Omega$ üle seatud limiidi. Hetkel valitud šunt takisti väärtuse juures on täpsus 5Ω limiidi jaoks ideaalne. Olukorras, kus ühte soovitud limiiti pole võimalik saavutada on eelistatud variandiks väiksem limiit lühise takistuse leidmisel kui see tähendab soovitud täpsust kontakti takistuse mõõtmisel. Valitud šunt takisti võib tunnistada sobivaks.

3.4 Kasutajaliides

Kasutajaliidese valikul tuleb mõelda selle peale, kuidas töölistel oleks kõige mugavam testrit kasutada ja ka selle peale, et testi tulemuste tõlgendamine oleks väga lihtne. Üks lihtsamaid variante kasutajale tagasisidet anda on värviliste LED-ide kasutamine. Esimene indikaator, mis kasutusele läheb on roheline LED. Indikaator läheb põlema kui mõõteahelas on toitepinge olemas ja annab kasutajale teada juhul kui testi tulemus oli negatiivne hoopis pinge puudumise tõttu. LED on ühendatud läbi takisti otse Arduino 5 V väljundpinge pealt maha ja kustub ära juhul kui pinge peaks mingil põhjusel mõõteahelast kaduma. Teiseks indikaatoriks saab punane LED, mis läheb põlema kui juhe testi ei läbi. Selle lülitamiseks võib kasutada ühte Arduino digitaalväljundit ja tule lülitamine toimub programmi koodi abil sõltuvalt testi tulemusest.

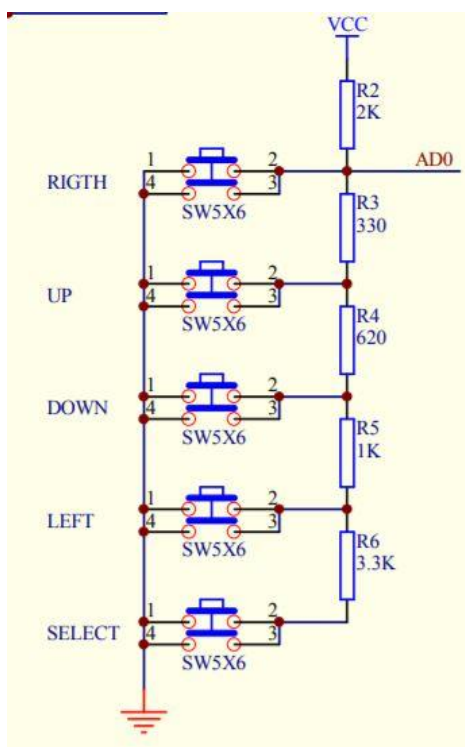
Lisaks algelistele LED indikaatoritele peaks kasutajaliides võimaldama ka valida testitavat juhet teda iseloomustava materjali koodi põhjal. Selle jaoks on hea võtta kasutusele ka LCD. Ekraani kasutamine võimaldab menüüde abil kasutajale kuvada kogu vajaliku info. Juurde on vaja veel liidest, mis võimaldaks kasutajal ka menüüs orienteeruda ja teha vajalikke valikuid. Selleks on kõige mugavam võtta kasutusele LCD moodul, millel on ka nupud juba olemas. Arendusplaatidele leidub mooduleid, mis on tehtud just selleks, et mooduli saaks lihtsalt Arduino plaadi pesadesse pista. Antud testi jaoks piisab küllaltki lihtsast displeist ja seetõttu osutus valituks DFRobot'i poolt toodetud DFR0009 LCD moodul (Joonis 5, Joonis 17, Joonis 18).



Joonis 5. LCD moodul DFR0009 [10].

Antud moodulil on 16 x 2 LCD displei, millel on 10 k Ω potentsiomeetri abil reguleeritav sinine taustavalgus. Klahvistik koosneb viiest nupust ja vajutatud klahvi tuvastamiseks kasutatakse 5-astmelist pingejagurit (Joonis 6). Klahvistiku väärtus leitakse kasutades Arduino analoogsisendit A0. Selleks tuleb testi programmi sisse panna alamfunktsioon, mis loeb sisendi väärtust ja võrdleb seda mooduli tootja poolt ette antud väärtustega, et tuvastada, millisesse vahemikku sisendi väärtus jääb [10].

Ekraani juhtimine toimib kasutades kuute Arduino digitaalväljundit/-sisendit. Antud displeid kasutatakse 4-bitises režiimis, mis tähendab, et läheb vaja nelja digitaalväljundit andmevahetuse jaoks. Lisaks on kasutusel veel lubav signaal ja väljund taustavalguse jaoks. Siin tuleb kasuks üks arendusplaatide boonustest, sest displei kontrollimiseks on võimalik kasutada juba olemas olevat Arduino teek LiquidCrystal [11].



Joonis 6. LCD mooduli klahvistiku tööpõhimõte.

Kuigi klahvistikul on saadaval viis nupp, siis vajadus on ainult nelja järgi ja seetõttu jääb „Select“ nupp kasutamata. Menüü struktuur tuleb võimalikult lihtne. Testri sisse lülitumisel prinditakse ekraani esimesele reale instruksioon, mis palub kasutajal valida juhtme, mida parasjagu testida soovitakse. Menüüs orienteerumiseks on kasutajal võimalik kasutada üles ja alla klahve. Teisele reale prindib programm välja esimese valiku. Samuti prinditakse teise rea lõppu üles ja alla nool või üks kahest. Nende abil antakse kasutajale märku kui ta on jõudnud valikute hulgas esimese või viimase variandi juurde. Kui kasutaja on oma valiku teinud, siis vajutab ta valiku klahvi, mille peale toimub automaatne testimine. Seejärel prinditakse ekraani esimesele reale kaabel, mida testiti ja teisele reale testi tulemus. Otsuse, kas antud tulemustega pääseb juhe testi läbi, teeb programm ise ja kasutajale prinditakse teisele reale ainult „PASSED“ või „FAILED“ tulemus. Peale testi on kasutajal võimalik vajutada tagasi klahvi, mille peale programm läheb tagasi algolekusse.

4 Trükkplaadi disain

Testri universaalsuse nõude täitmiseks tuleb eraldada komponendid, mis on unikaalsed ainult ühe tootmisliini jaoks. Selle nõude täitmiseks peab ühe trükkplaadi asemel kaks koostama ja disaini jaotama põhiliseks trükkplaadiks ja lisaplaadiks. Lisaplaadile peab paigutama ainult pistikud, sest kõik ülejäänud komponendid sobivad testris kasutamiseks ka teistel tootmisliinidel.

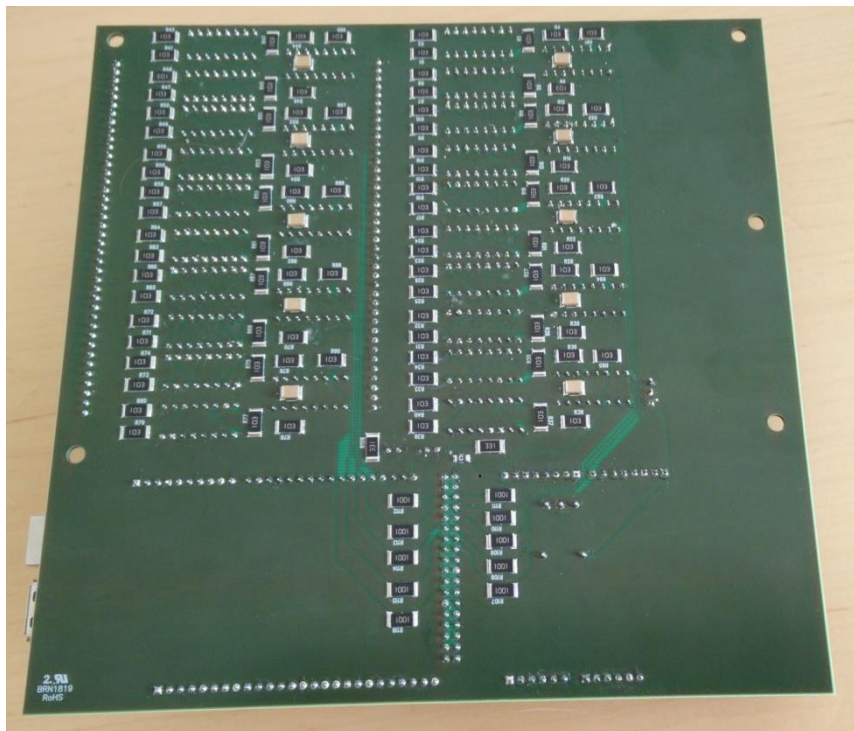
Põhiplaadi puhul on suuruse piiranguks ette antud dimensioonid. Testrile on vaja printida korpus ja selleks kasutatava 3D-printeriga on võimalik printida detaile maksimaalse suurusega 20 cm x 20 cm x 20 cm. Järelikult peavad trükkplaadi dimensioonid jääma nende limitide sisse.

Transistoride puhul on tegemist DIP-pakendiga, mida on võimalik paigutada DIP-soklitesse üksteise kõrvale. Transistorid lähevad nelja kaupa DIP-16 soklitesse ja paigutatakse neid lülitavate multipleksorite lähedusse, mis paigutatakse samuti DIP-16 soklitesse. Sisendi- ja väljundipoolsed komponendid jäävad üksteisest eraldi ja asetuse jookseb selliselt, et on võimalik väljundid ja sisendid järjest ühendada üherealise piikriba kontaktidega, mille abil ühendatakse külge pistikute plaat. Arduino ja LCD mooduli kohad jäävad kõrvuti trükkplaadi allosasse. LCD moodul on piisavalt kõrge, et sinna alla mahutada ka pingeregulaator (Joonis 7).



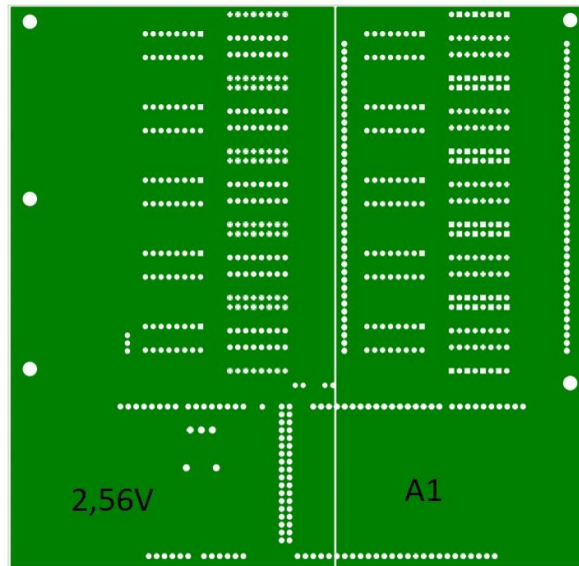
Joonis 7. Põhiplaadi ülemine külg, kus asuvad Arduino, LCD moodul, transistorid ja MUXid.

Läbivauk komponendid täidavad ära juba suurema osa trükkplaadi pealmisest kihist, kuid paigutada on veel vaja takistid ja kondensaatorid. Need on mõistlikum paigutada juba alumisele küljele, sest tegemist on pindmontaaži komponentidega, mida saab ka kegemalt piagutada avade lähedusse. Kondensaatorite paigutusel tuleb lähtuda sellest, et komponent oleks võimalikult lähedal multipleksori toitekontaktile ja takistite puhul võiks asukoht olla võimalikult lähedal transistori paisu kontaktile. Alumisel küljel on vaba ruumi piisavalt palju ja see võimaldab valida takistid oluliselt suurema pakendiga kui skeemi jaoks vaja oleks. Tegemist on tollimõõdustikus 2512 pakendiga takistitega, mille nimivõimsuseks on 2W. Suurema pakendi valik muudab lihtsamaks trükkplaadi jootmise, sest pindmontaaži komponendid tuleb peale paigutada käsitsi ja väikeste pakendite puhul on väga keerukas komponente paigutada ja jootepastat doseerida (Joonis 8).



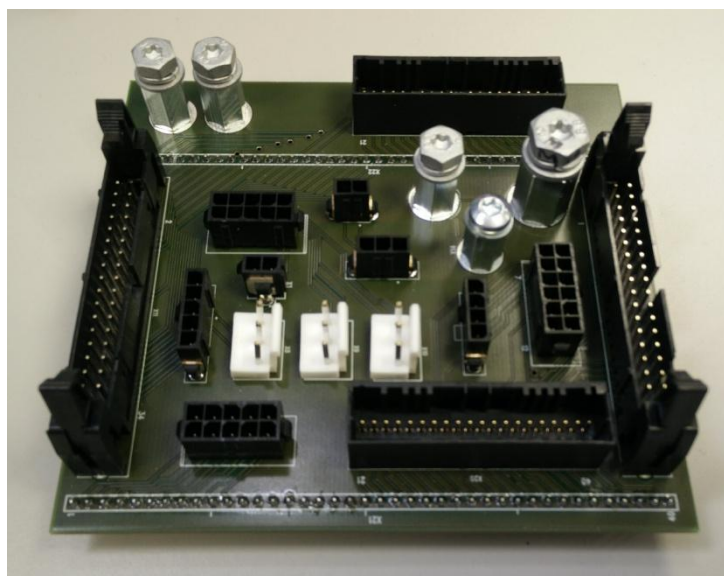
Joonis 8. Põhiplaadi alumine kül, kus asuvad takistid ja kondensaatorid.

Kaheksakümne transistori ühenduse radade jaoks kahekihilisel plaadil ruumi ei jagu ja seetõttu tuli kasutada nelja kihti. Esimene sisekihti sai jaotatud kaheks piirkonnaks. Vasak pool sisekihist on ühendatud pingeregulaatori väljundiga, mis annab transistoridele peale 2,56V pingeniivo ja parem pool ühendab analoogsisendisse kõik alumise poole transistoride lätted (Joonis 9). Teine sisekiht on kasutatud 5 V pingeniivo jaoks.



Joonis 9. Esimene sisekiht, mis on kaheks jaotatud.

Pistikute plaadi puhul tuli esmalt paika panna piikribad selliselt, et nende asukoht ühtiks põhiplaadi omadega ja võimaldaks pistikute plaadi hiljem põhiplaadi peale paigutada. Järgmisena sai paigutada pistikud ja nende asukohad pidid olema sellised, et kõiki kaableid saaks mugavalt ühendada. Kahe piikriba vahel asuvale alale siiski kõiki pistikuid mahutada ei õnnestunud ja plaati laiemaks tehes tuli jälgida, et kindlasti jääks laiem külg vasakule poole. Vastasel juhul oleks pistikute plaat ulatunud üle põhiplaadi ääre ja takistanud korpuse printimist. Selle plaadi puhul sai vajalikud ühendused tehtud kahe kihi abil (Joonis 10).



Joonis 10. Pistikute plaat pealtvaates.

5 Programmi kood

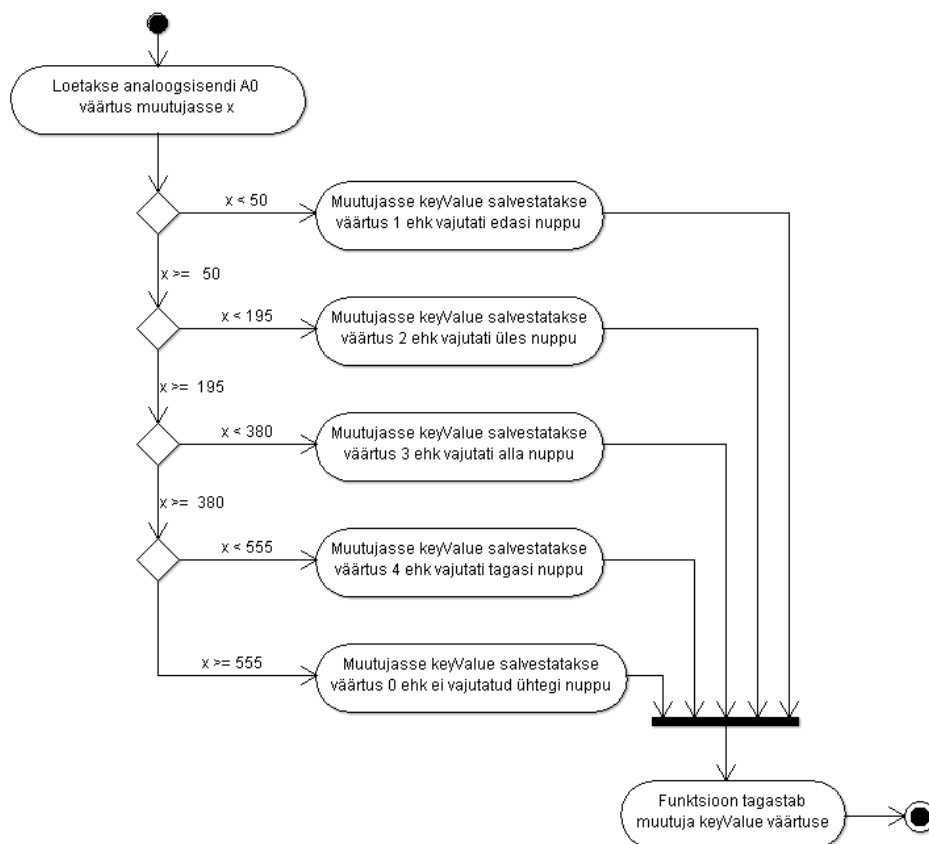
Arduino programmi kood põhineb C++ keelel ja funktsionaalsus on üles ehitatud kahe funktsiooni abil. Esimese funktsiooni sisse tuleb kirjutada koodi osad, mida programm peab ainult ühe korra läbi jooksutama ja teise funktsiooni sisse tuleb kirjutada koodi osa, mis jääb käima tsüklisse. Defineeritud on neli globaalset muutujat, millest kahte kasutatakse nupuvajutuste salvestamiseks, hetkel aktiivse juhtme määramiseks ja juhtme materjalikoodide salvestamiseks. Lisaks globaalsetele muutujatele on defineeritud ka juhtmete konfiguratsioonide maatriksid. Suurte mõõtude tõttu ei jagu Arduinol piisavalt mälu, et neid tavamuutujatena defineerida. Maatriksid tuli kirjutada programmi välkmälusse kasutades selleks PROGMEM funktsiooni [12]. Defineeritud on ka LCD jaoks kuus digitaalväljundit ja kaks erisümbolit, mida menüüs kasutatakse. Arduino initsialiseerimiseks mõeldud setup funktsioonis initsialiseeritakse LCD, seatakse Arduino vajalikud digitaalväljundid õigetele väärtustele ja printitakse välja algmenüü. Kasutusel on kümme väljundit multipleksorite lubavate signaalide ja kuus väljundit aadresside jaoks. Teine funktsioon sisaldab endas peamiselt alamfunktsioonide väljakutseid ja enamus programmi funktsionaalsusest on jaotatud alamfunktsioonidesse. Järgnevalt on kirjeldatud viie alamfunktsiooni eesmärki põhjalikumalt.

5.1 GetButton funktsioon (Joonis 11)

GetButton funktsioon on alamfunktsioon, mida kasutatakse klahvistiku vajutuste jälgimiseks. Funktsioonis on deklareeritud kaks muutujat. Esimest muutujat „x“ kasutatakse, et sisse lugeda ADC väärtust analoogsisendi A0 peal. Seejärel võrdleb funktsioon muutuja „x“ väärtust tootja poolt ette antud arvudega, et välja selgitada, millist nuppu vajutati. Klahvi väärtused pole määratud kindla arvuga, vaid iga nupu jaoks on eraldatud mingi sisendväärtuste vahemik, mille sisse väärtus jääb (Tabel 4 Tabel 4. Analoogsisendi väärtustele vastavad nupu väärtused.). Esmalt võrreldakse sisendväärtust kõige madalama limiidiga ja kui väärtus sellesse vahemikku ei mahu võetakse järgmine kõrgem limiit kuni leitakse sisendi väärtusele vastav vahemik. Kui sisendväärtus ületab klahvidele määratud vahemiku ilma, et väärtus mõne liimidi sisse mahuks, siis võib järeldada, et ühtegi klahvi vajutatud ei ole. Peale vajutatud nupu tuvastamist salvestab funktsioon nupule vastava väärtuse muutujasse „keyValue“ ja tagastab selle väärtuse põhifunktsioonile.

Tabel 4. Analoogsisendi väärtustele vastavad nupu väärtused.

Analoogsisendi A0 väärtus x	Vastav klahv
$x < 50$	Edasi klahv
$x < 195$	Üles klahv
$x < 380$	Alla klahv
$x < 555$	Tagasi klahv
$x > 554$	Klahvi ei ole vajutatud

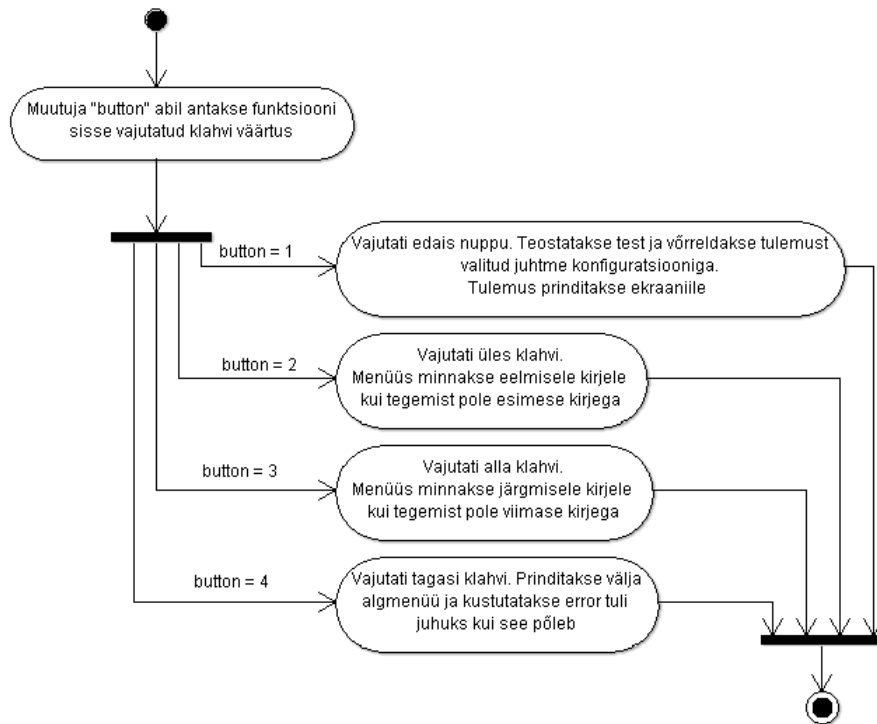


Joonis 11. Funktsiooni getButton tegevusdiagramm.

5.2 MenuSelection funktsioon (Joonis 12)

Funktsioon menuSelection kutsutakse välja kui programm tuvastab, et on vajutatud klahvi. Funktsiooni sisendiks on getButton funktsiooni väljund, mida kasutatakse *switch* väite abil nelja erineva menüü tegevuse vahel valimiseks. Üles ja alla klahvi vajutamise

puhul printitakse välja uus menüü kirje vastavalt valitud suunale. Tagasi klahvi vajutuse peale viiakse programm algolekusse. Kui vajutatud on edasi klahvi, siis printib funktsioon ekraanile valitud juhtme koodi ja kutsutakse välja teine alamfunktsioon, mis teostab mõõtmised. Kasutusel veel teine *switch* väide, mille sisendiks on valitud juhtme järjekorra number. Juhtmeid on kokku kaksteist tükki, aga nelja kaabli puhul saab kasutada kontrolliks sama konfiguratsiooni. Neli ühesugust valikuvarianti võib *switch* väite alt ära kustutada kui see konfiguratsioon lisada hoopis *default* tegevuse alla. Juhtme numbril põhjal annab funktsioon õiged väärtused sisse järgmisele alamfunktsioonile, mis teostab tulemuste kontrolli.



Joonis 12. Funktsiooni menuSelection tegevusdiagramm.

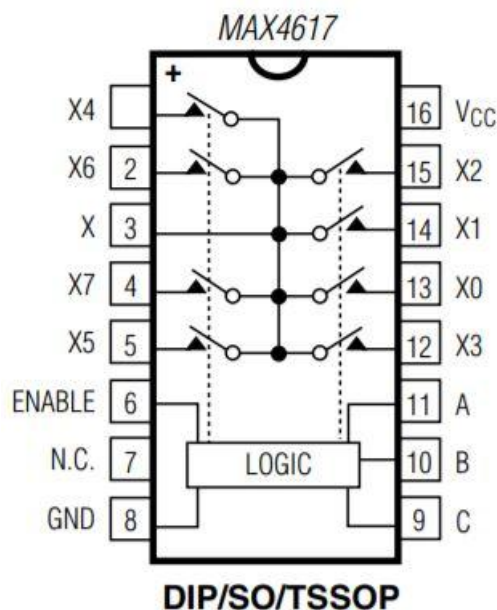
5.3 Test funktsioon (Joonis 14)

Selleks, et programmi testimisfunktsiooni ja kaablite kontrollimiseks koostatavate konfiguratsioonide koostamine oleks lihtsam on testimine universaalne kõigi kaablite puhul. Iga testi puhul kontrollitakse üle kõik võimalikud ühenduse kombinatsioonid.

Testril on nelikümmend väljundkontakti ja sisendkontakti mis tähendab, et kokku kontrollitakse üle kõik 1600 ühendust, mis on teoorias võimalikud. See meetod suurendab oluliselt testi aega, kuid lülitamise kiirus on piisavalt suur, et kasutaja jaoks käib testimine endiselt väga kähku. Iga kontakti väärtust kontrollitakse kolm korda ja salvestatakse kolme mõõtmise keskmine. Viiteks mõõtmiste vahel on pandud 200 μ s. Selle põhjal saab kalkuleerida testimiseks kuluva aja.

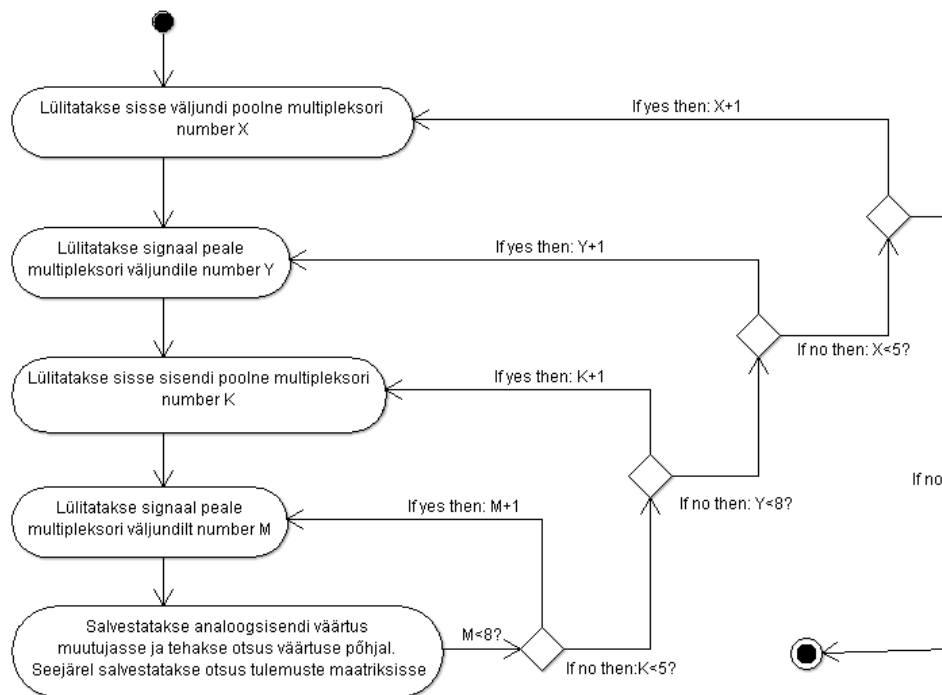
$$t = (3 * 200 * 10^{-6}) * 1600 = 0,96 \text{ sekundit}$$

Kanalite lülituste loogika on üles ehitatud kuue üksteise sisse paigutatud tsükli abil. Esmalt antakse pinge peale esimesele kontaktile ja kontrollitakse teises otsas üle kõik nelikümmend kontakti. Seejärel lülitatakse pinge peale teisele kontaktile ja teostatakse jälle mõõtmised teisel pool kõigi kontaktide peal.. Mõlema poole puhul toimub lülitamine järjekorras, kus sisemist tükklit kasutatakse selleks, et muuta väärtuseid multiplexsori kolme aadressi kontakti peal. Multiplexsori väljundeid kontrollitakse kolme kontakti abil kahendväärtustega. Multiplexsori pakendi puhul ei ühti kahjuks kontaktide järjestus kahendkoodis oleva väljundite järjekorraga (Joonis 13). Selleks, et kõrvuti reas olevaid transistore saaks lülitada järjest on koodi kirjutatud sisse maatriks, mille abil käib multiplexsori kanalite lülitamine järjekorras 5-7-6-4-2-1-0-3. Keskmine tsükkel kontrollib seda, milline multiplexsori väljaund on parasjagu aktiveeritud ja välimine tsükkel kontrollib seda, mitmes multiplexsori on aktiivne.



Joonis 13. Multiplexsori MAX4617 kontaktide konfiguratsioon [7].

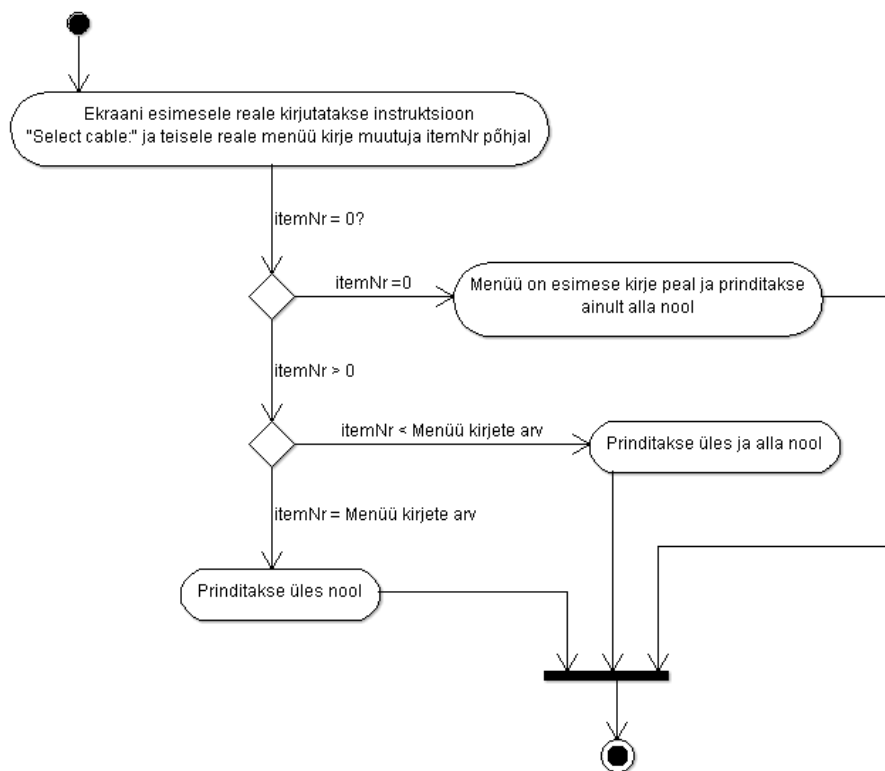
Peale mõõtmist täidetakse funktsiooni sisendisse antud kahemõõtmeline maatriks tulemustega. Sisendi väärtuse põhjal kirjutatakse maatriksisse tulemus 0, 1 või 2. Null tähistab puuduvat ühendust, üks tähistab tervet ühendust ja kaks tähistab kõrge takistusega ühendust.



Joonis 14. Funktsiooni test tegevusdiagramm.

5.4 MenuList funktsioon (Joonis 15)

MenuList funktsioon on alamfunktsioon, mis prindib põhimenüü kasutaja jaoks LCD ekraanile. Hetkel selekteeritud juhtme numbri ja selle juhtme nimetuse saab funktsioon gobaalsetest muutujatest. Lisaks prindib antud funktsioon ka menüü indikaator nooled, mis aitavad kasutajal menüüs orienteeruda. Noolte printimiseks on kolm varianti ja see, mida printida otsustatakse selekteeritud kaabli numbri ja menüü kirjade koguarvu võrdlemise abil.

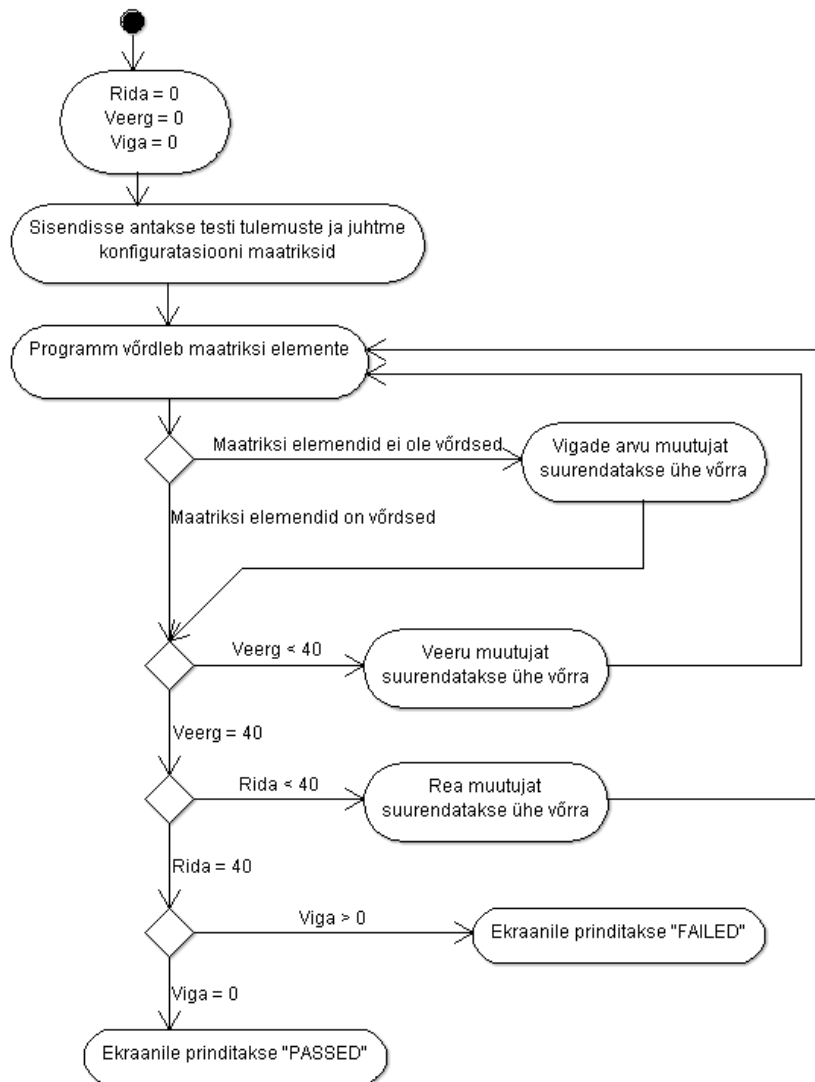


Joonis 15. Funktsiooni menuList tegevusdiagramm.

5.5 Compare funktsioon

Funktsioon compare teostab lõpliku otsuse testi tulemuste üle. Funktsiooni sisendiks on testi tulemuste maatriks ja valitud juhtme konfiguratsiooni maatriks. Kahe tsükli abil võrreldakse mõlema maatriksi väärtuseid ja kui leidub erinevusi kaabli konfiguratsiooni

ja testi tulemuste vahel, siis prinditakse kasutajale ekraanile välja tulemus „FAILED“.
 Kui võrdlusel erinevusi ei leidu, siis prinditakse tulemus „PASSED“.



Joonis 16. Compare funktsiooni tegevusdiagramm

6 Kokkuvõte

Bakalaureusetöö käigus sai projekteeritud kaablitester, mis sobib juhtmete testimiseks, mida kasutatakse tootmisliinil kokku monteeritavas seadmes. Disain põhineb Arduino arendusplaadil. Testri jaoks sai disainitud kaks trükkplaati, millest suurem on põhiplaat, mille peale piikribade abil kinnituvad LCD moodul, Arduino ja väiksem trükkplaat koos pistikutega. Algsest kahest limiidist, mida testimisel kasutada sooviti õnnestus saavutada ainult üks. Lihtsast disainist tuleneva piiratud mõõtevahemiku tõttu tuli lühise otsimiseks seatud limiiti märgatavalt vähendada, kuid ka madalama limiidiga peaks tester oma soovitud funktsiooni täitma.

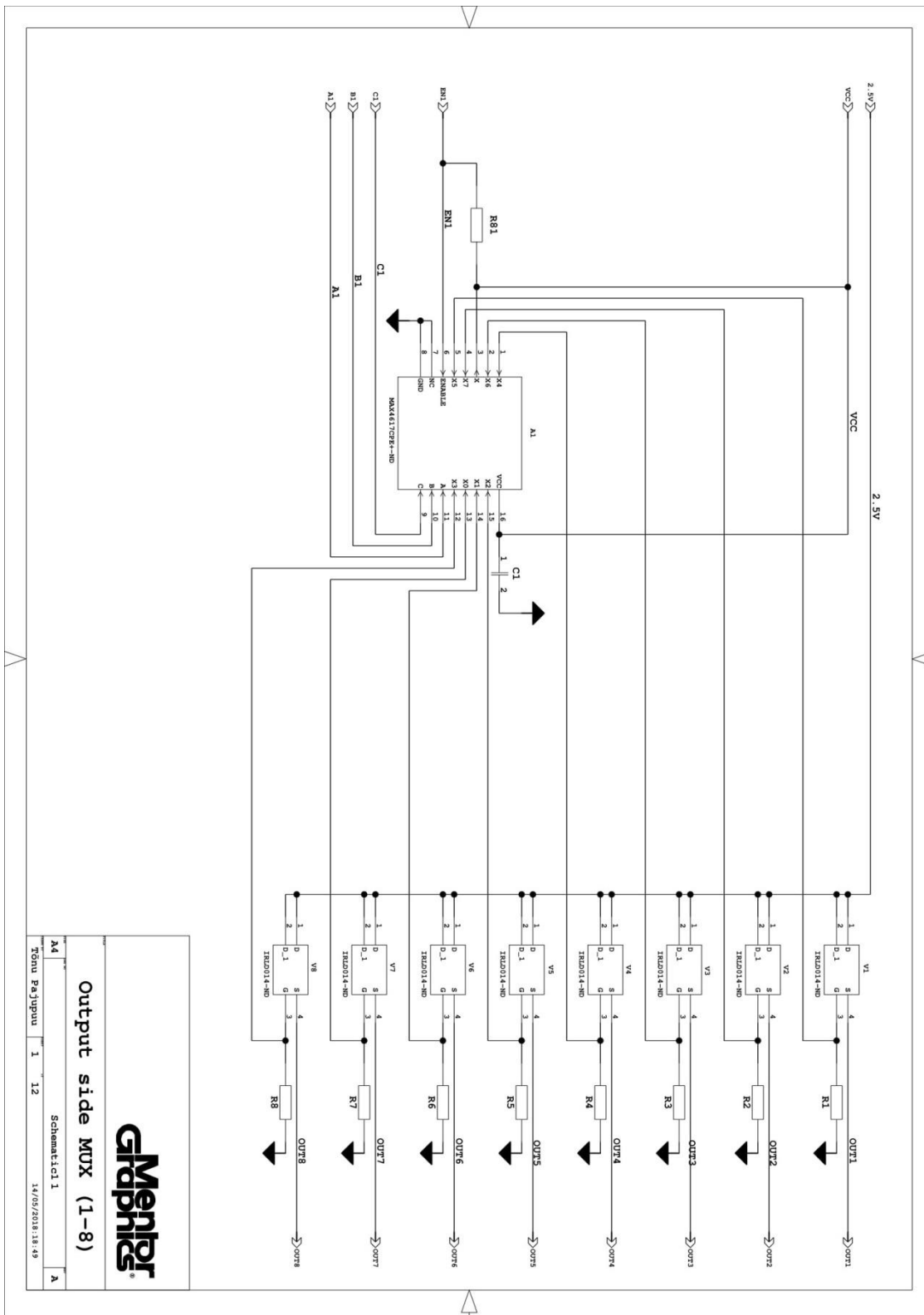
Testri esimene mudel on hetkel töökorras ja läheb peale karbi disainimist tootmisliinil kasutusse. Testri kasutusele võtmise järel tuleb hakata koguma andmeid valevigade kohta tootmisliinil ja võrreldes statistikat varasemate näitajatega saab teha järelduse, kas ja kui palju testri kasutusele võtmine valede komponentide välja vahetamist on vähendanud. Kui tester ennast ära tasub, siis saab seadme kasutamist ka teistele tootmisliinidele laiendada.

Lisaks testri tasuvusele on võimalik aja jooksul näha, kas seatud disaini kriteeriumid ja limiidid on põhjendatud. Juhul kui esineb probleeme testriga, siis on võimalik viia sisse parandused enne järgmise testri koostamist.

7 Kasutatud kirjandus

- [1] "Guidelines for setting resistance test thresholds," Cirris Systems, [Online]. Available: <https://www.cirris.com/learning-center/general-testing/guidelines/112-guidelines-for-setting-resistance-test-thresholds>. [Accessed 26 04 2018].
- [2] "Continuity test resistance calculator," Cirris Systems, [Online]. Available: <https://www.cirris.com/learning-center/calculators/126-continuity-test-resistance-calculator>. [Accessed 26 04 2018].
- [3] Jerry Janesch, "Two-Wire vs. Four-Wire Resistance Measurements: Which Configuration Makes Sense for Your Application?," May 2013. [Online]. Available: <http://www.tek.com/support/faqs/ohms-what-difference-between-2-wire-and-4-wire-ohms-measurements>. [Accessed 19 4 2018].
- [4] "Arduino MEGA ADK REV3," Arduino, 2018. [Online]. Available: <https://store.arduino.cc/usa/arduino-mega-adk-rev3>. [Accessed 04 05 2018].
- [5] "ATmega V-2560 microcontroller datasheet," Atmel, [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf. [Accessed 04 05 2018].
- [6] "IRLD014 Datasheet," Vishay Siliconix, 02 10 2012. [Online]. Available: http://www.farnell.com/datasheets/2046274.pdf?_ga=2.7743730.1564373310.1525940775-1651410510.1523033679. [Accessed 10 05 2018].
- [7] "MAX4617 Datasheet," Maxim Integrated, 2012. [Online]. Available: http://www.farnell.com/datasheets/1911551.pdf?_ga=2.35408163.306955065.1520238474-543861978.1496833116. [Accessed 10 05 2018].
- [8] "Arduino Analog reference," Arduino, [Online]. Available: <https://www.arduino.cc/reference/en/language/functions/analog-io/analogreference/>. [Accessed 11 05 2018].
- [9] "2,25A, Wide-input adjustable switching regulator," Texas Instruments, 09 2013. [Online]. Available: <http://www.ti.com/lit/ds/symlink/pt08080w.pdf>. [Accessed 11 05 2018].
- [10] "Arduino LCD keypad shield," DFRobot, [Online]. Available: [https://www.dfrobot.com/wiki/index.php/Arduino_LCD_KeyPad_Shield_\(SKU:_DFR0009\)](https://www.dfrobot.com/wiki/index.php/Arduino_LCD_KeyPad_Shield_(SKU:_DFR0009)). [Accessed 07 05 2018].
- [11] "LiquidCrystal library," Arduino, 2018. [Online]. Available: <https://www.arduino.cc/en/Reference/LiquidCrystal>. [Accessed 07 05 2018].
- [12] „PROGMEM,“ Arduino, 2018. [Võrgumaterjal]. Available: <https://www.arduino.cc/reference/en/language/variables/utilities/proGMEM/>. [Kasutatud 18 05 2018].
- [13] DFRobot, "LCD keypad shield schematic," 19 02 2009. [Online]. Available: https://www.dfrobot.com/wiki/images/a/a7/LCDKeypad_Shield_SCH.png. [Accessed 14 05 2018].

Lisa 1 – Testri elektriskeem



Output side MUX (1-8)

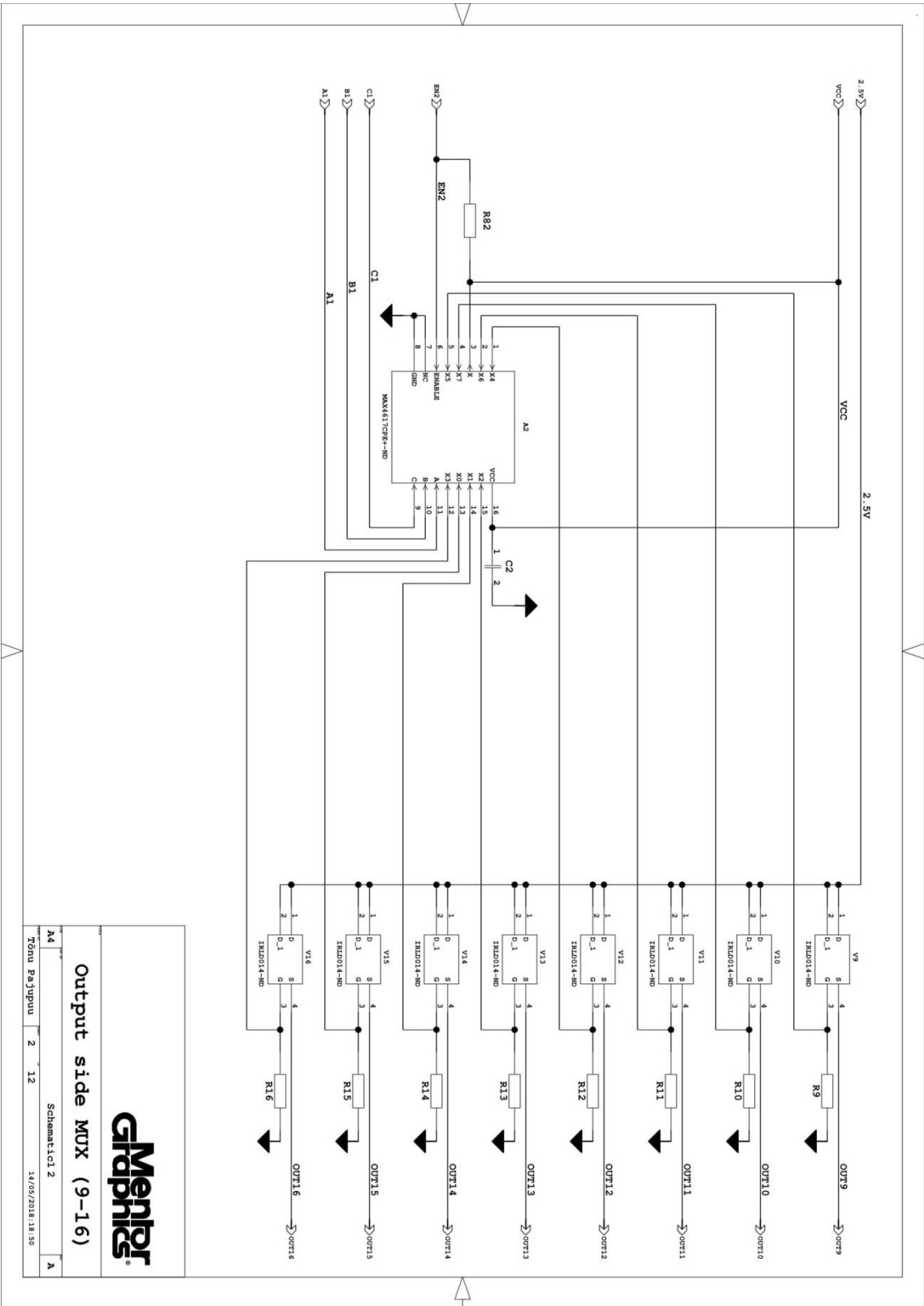
Schematic1.c1 1

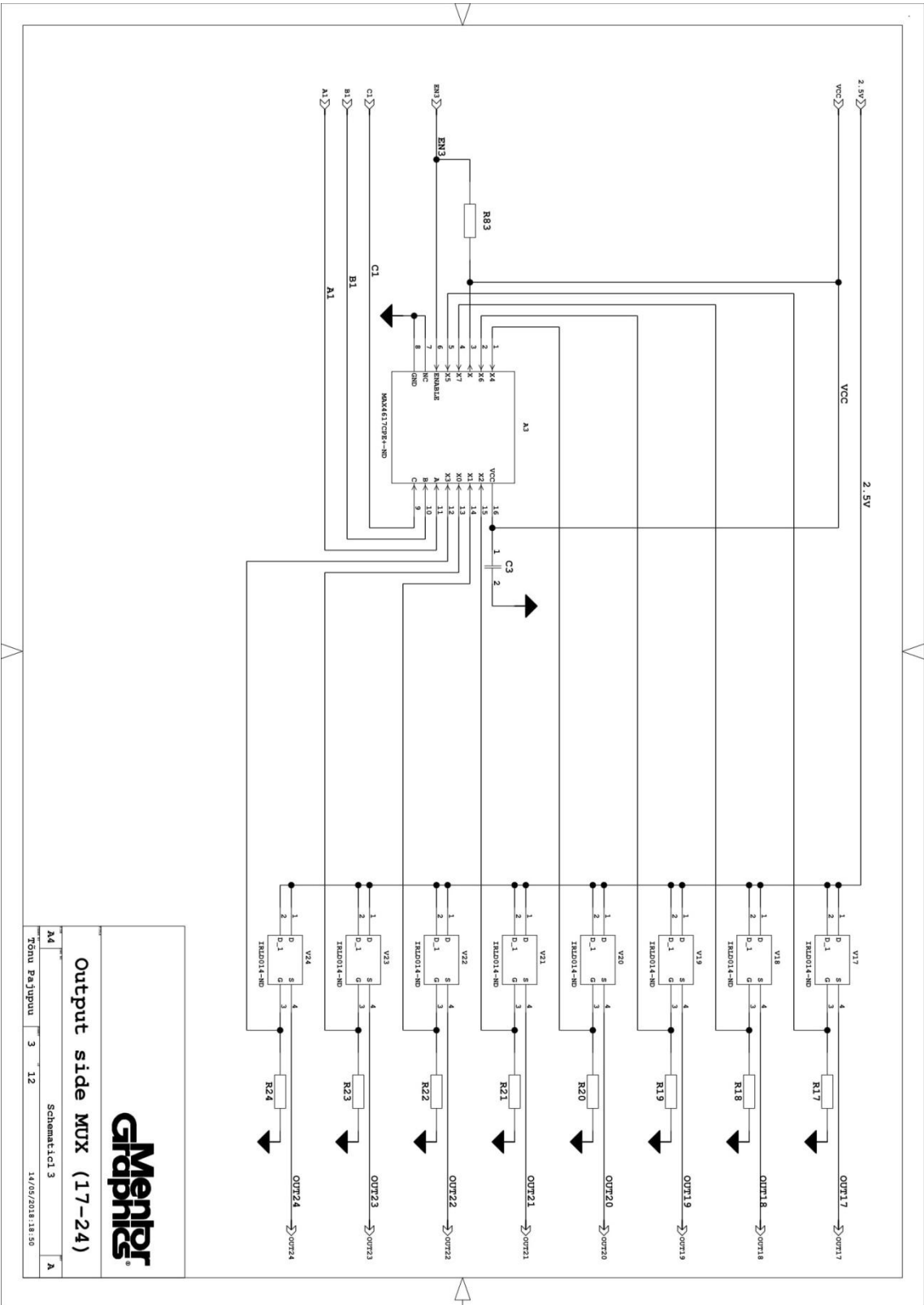
14/05/2018 18:49

A4

Temu Pa Jipuvu 1 12

A

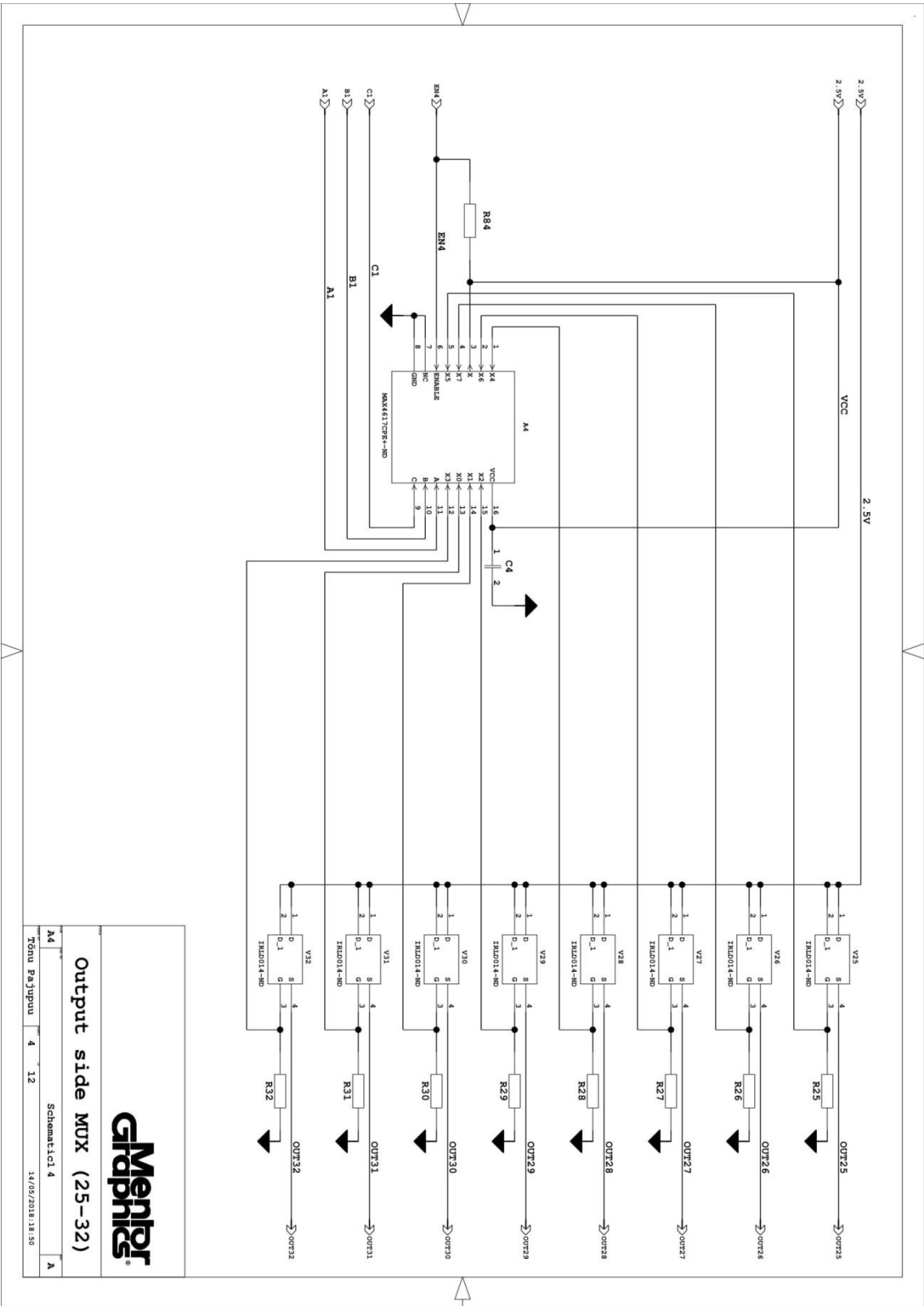




Mentor
Graphics

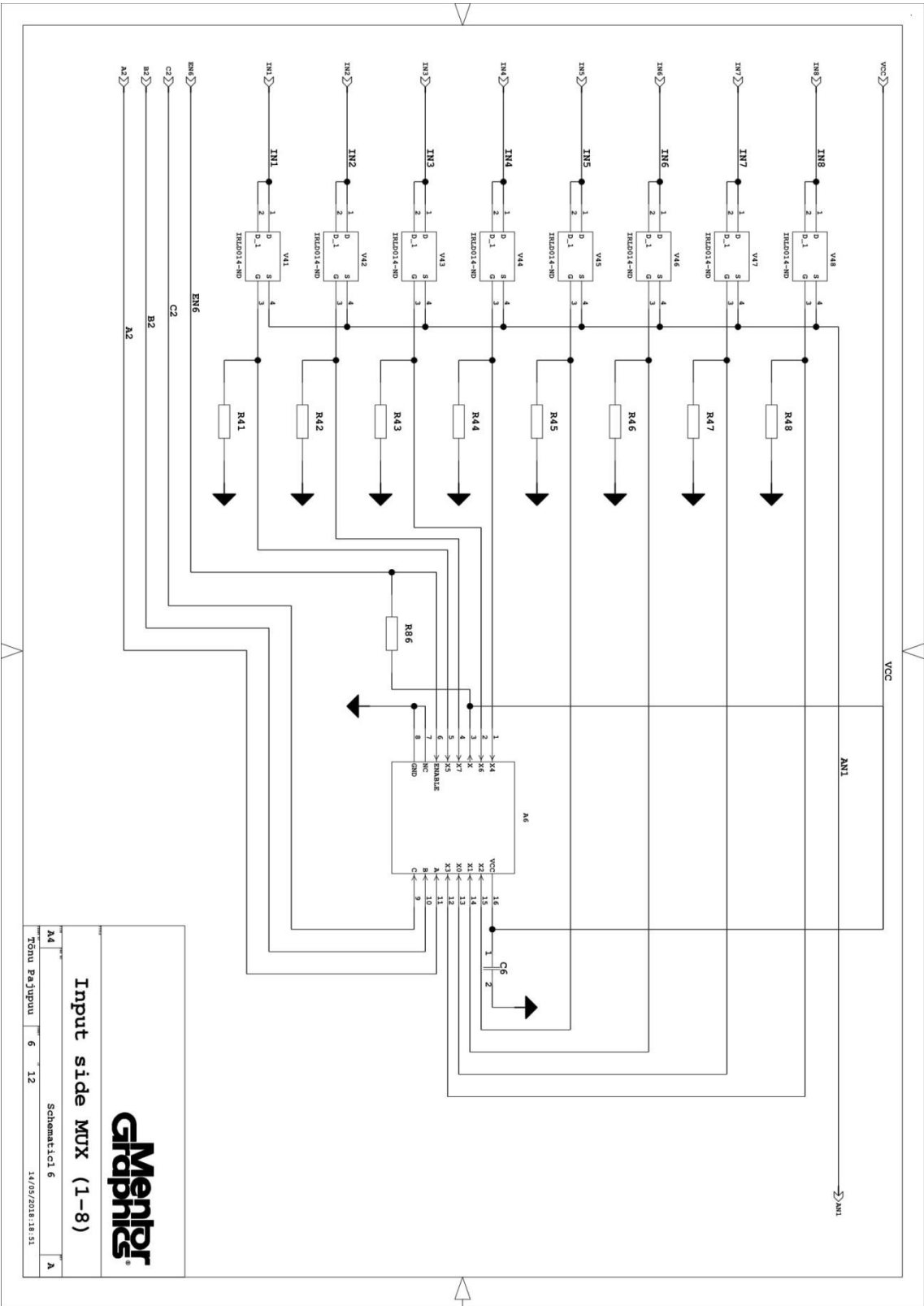
Output side MUX (17-24)

A4 Schematic:1 3 A
 Tonu PaJuppu 3 12 14/09/2018 18:50



Output side MUX (25-32)

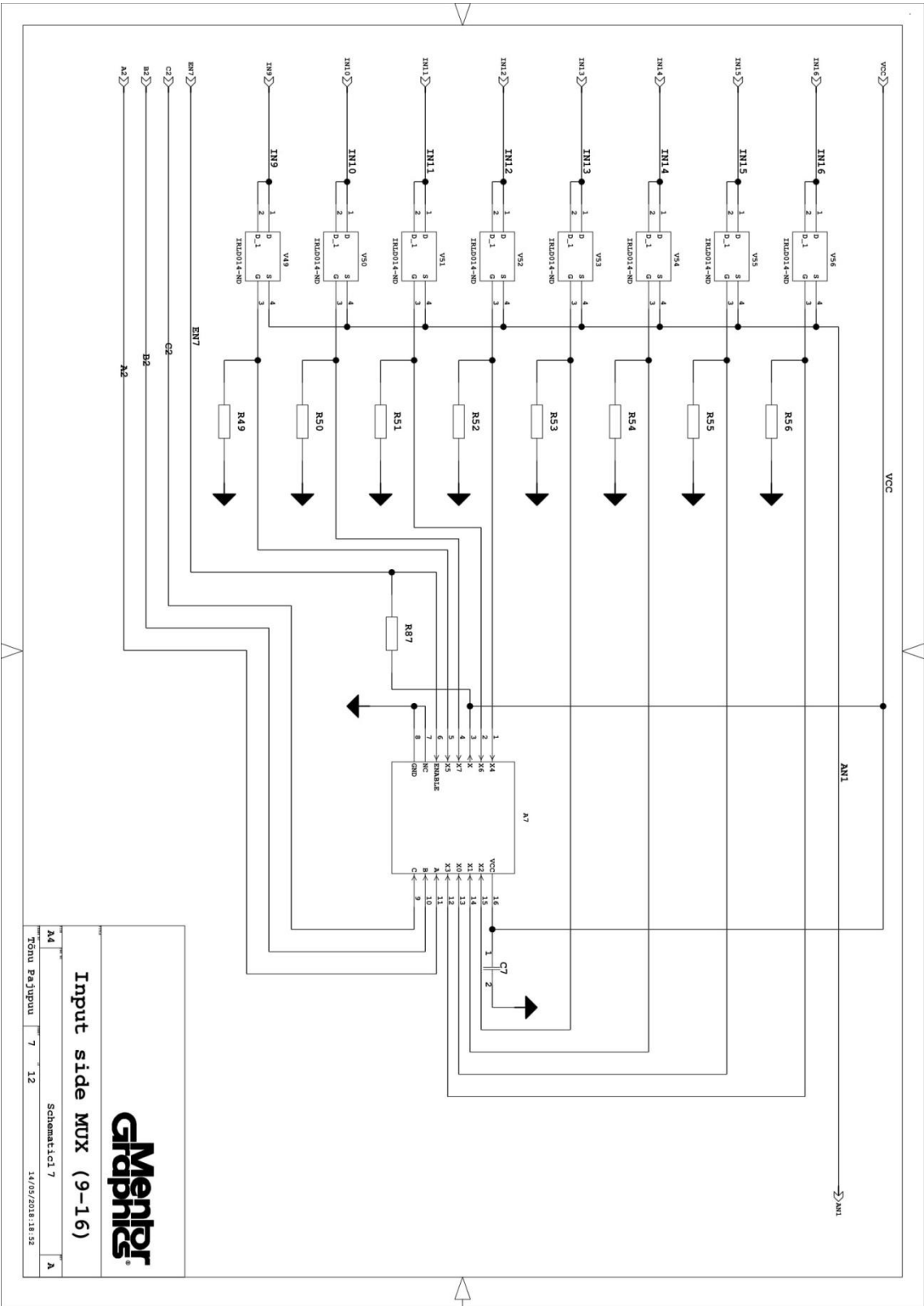
A4
 Tomu Pa-Juppu 4 12
 Schematic:CI 4
 14/09/2018 18:50
 A



Mentor
Graphics

Input side MUX (1-8)

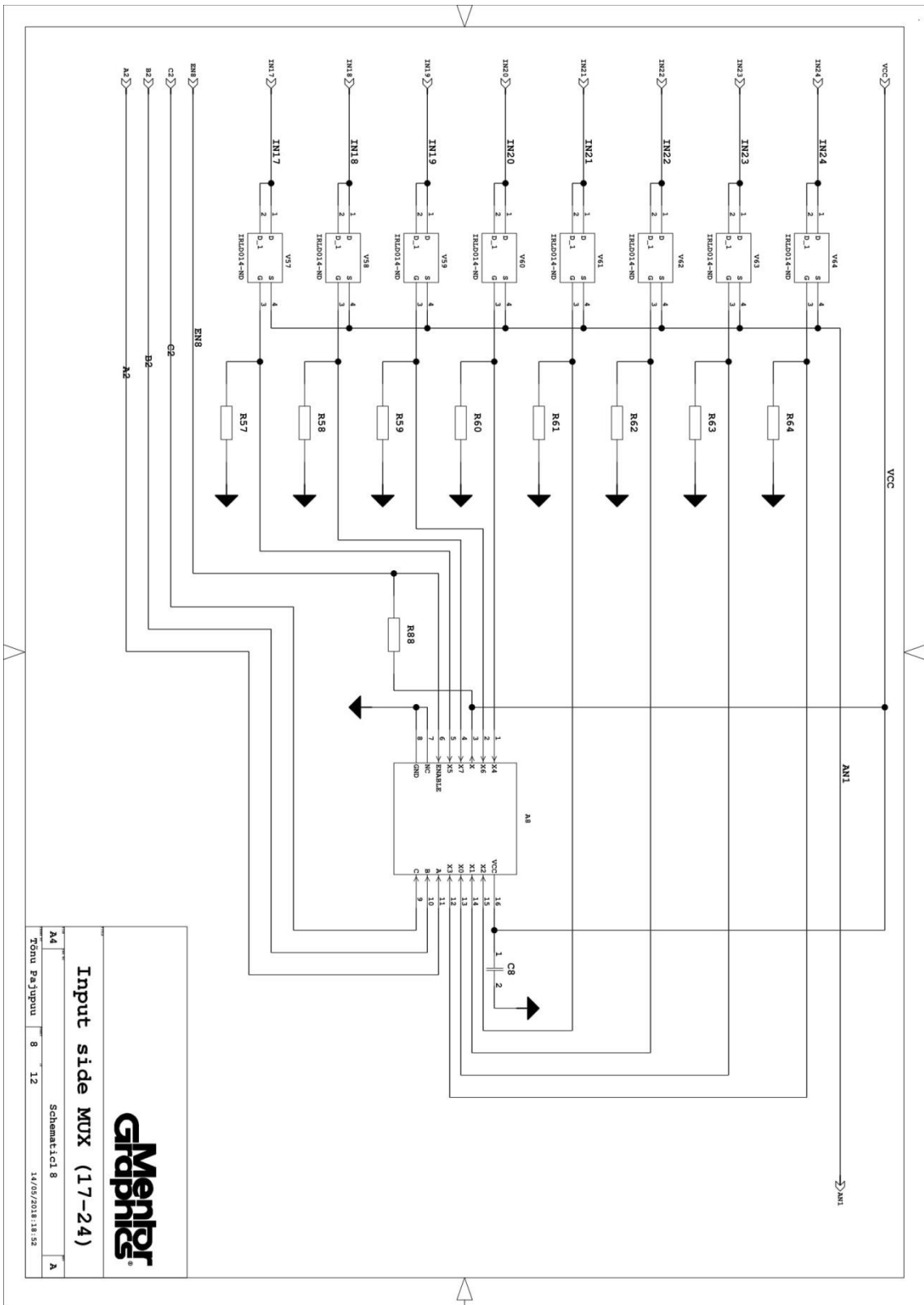
Schematic: 6
A4
Tonu PaJuppu 6 12
14/09/2018 18:51 A



Mentor
Graphics

Input side MUX (9-16)

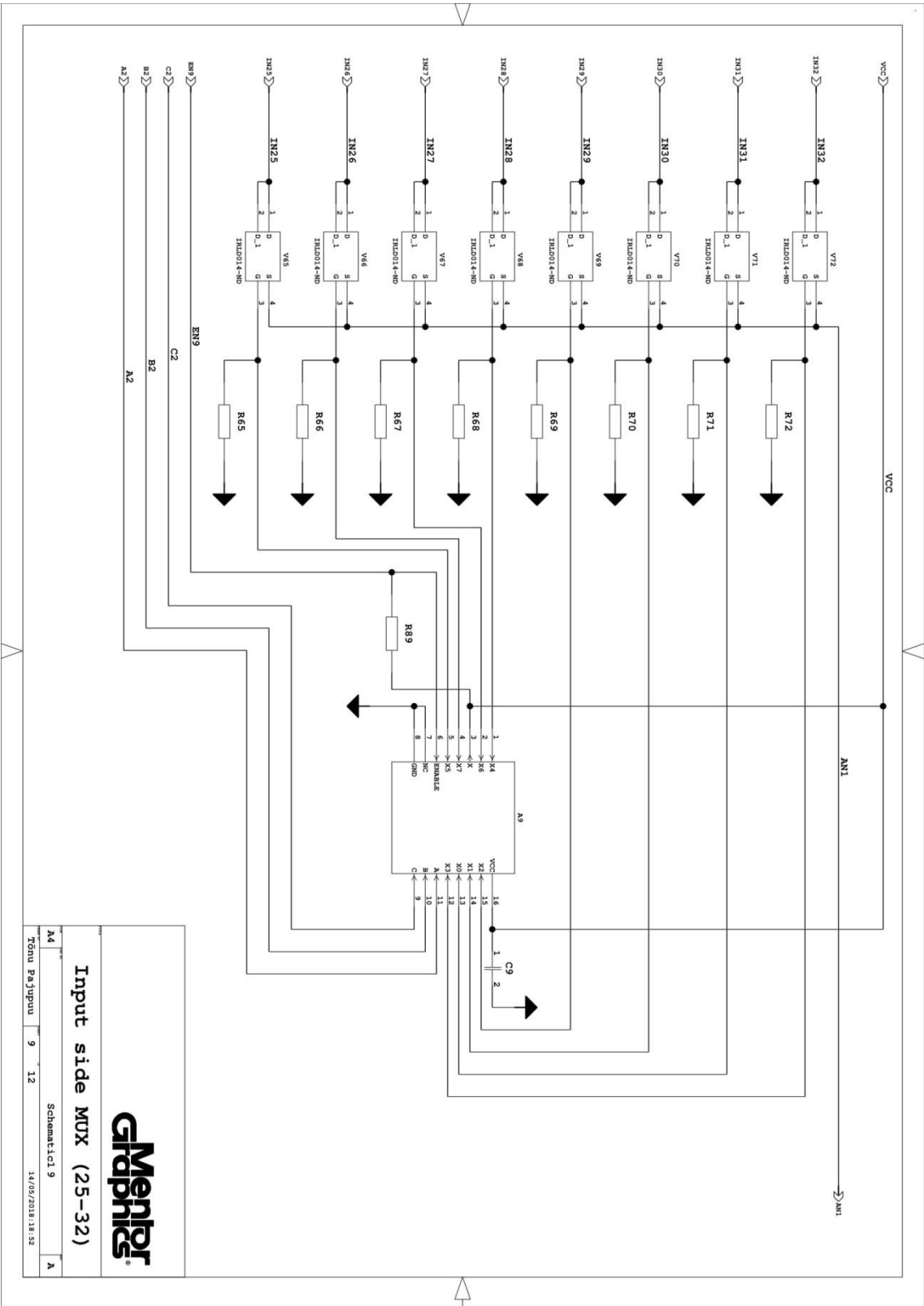
3d
Tonu PaJuppu 7 12
Schematic1 7
14/05/2018 18:58
A



Mentor
Graphics

Input side MUX (17-24)

Schematic: 1 8
 14/03/2018 18:52
 A



Mentor
Graphics

Input side MUX (25-32)

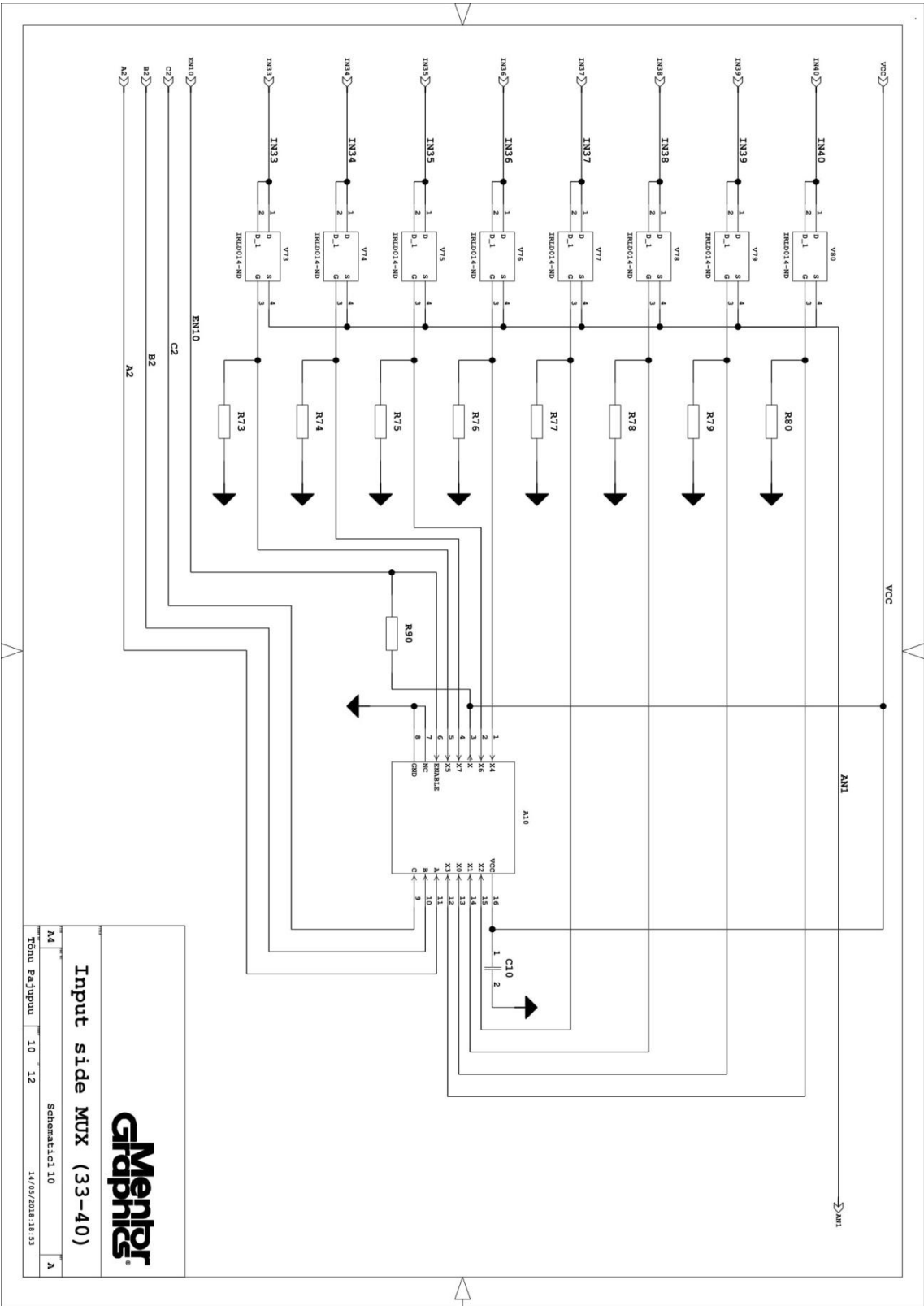
Schematic: CI 9

14/03/2018 18:58

14

Tonu Pa'juppu 9 12

A

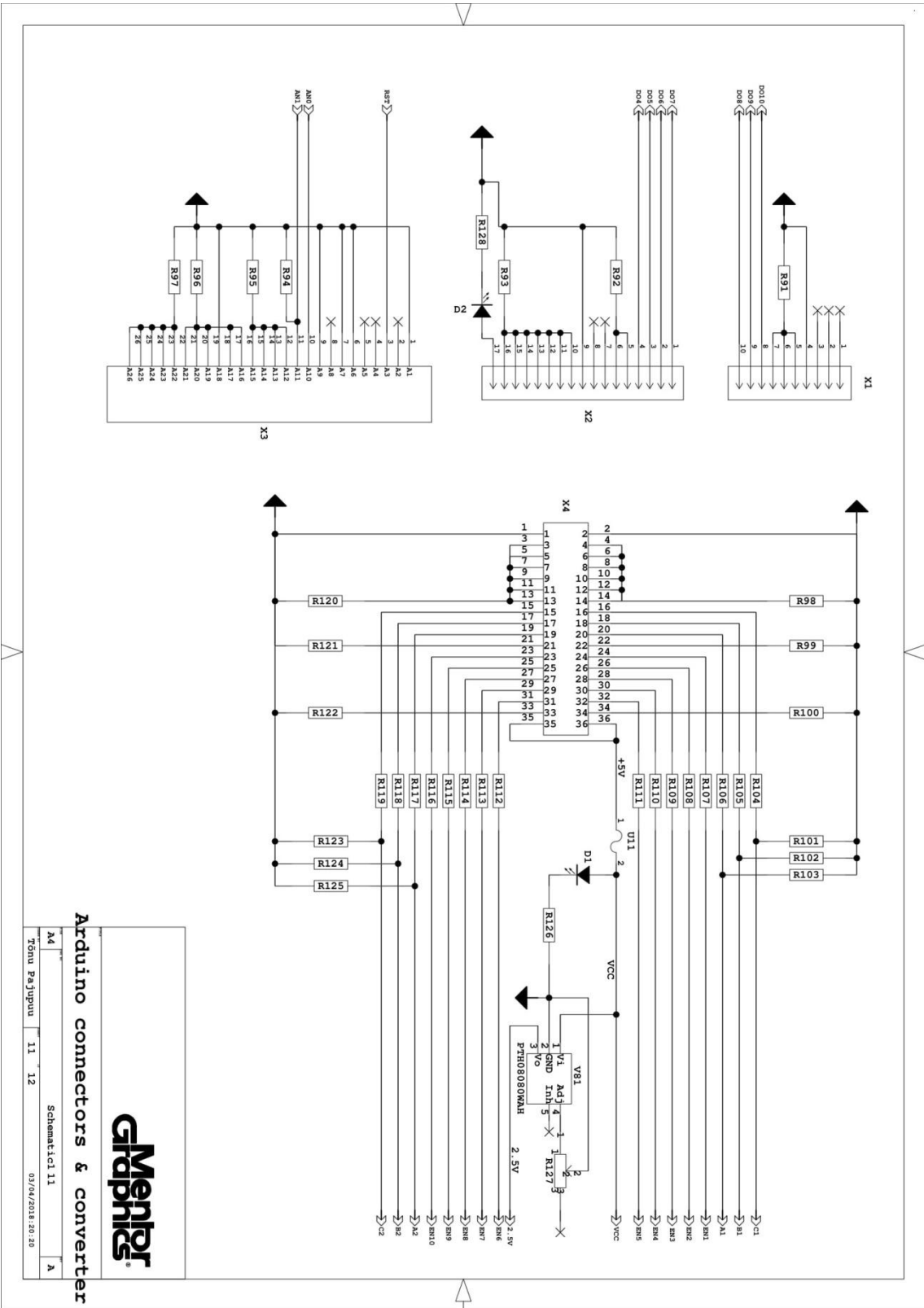


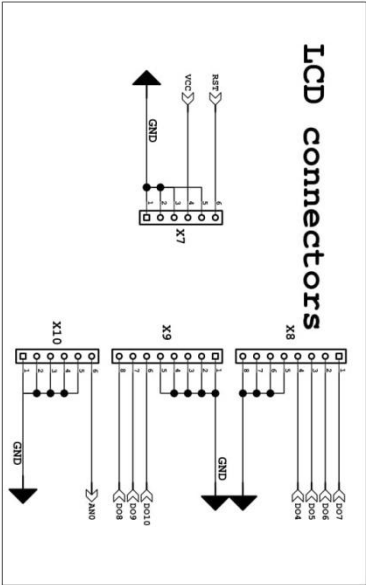
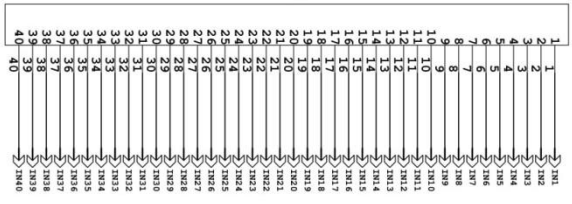
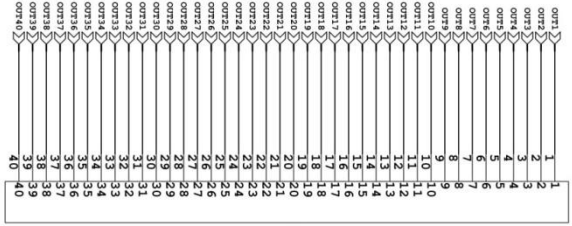
Mentor
Graphics

Input side MUX (33-40)

Schematic:10
14/03/2018 18:53

A4
Tonu PaJuppu 10 12 A





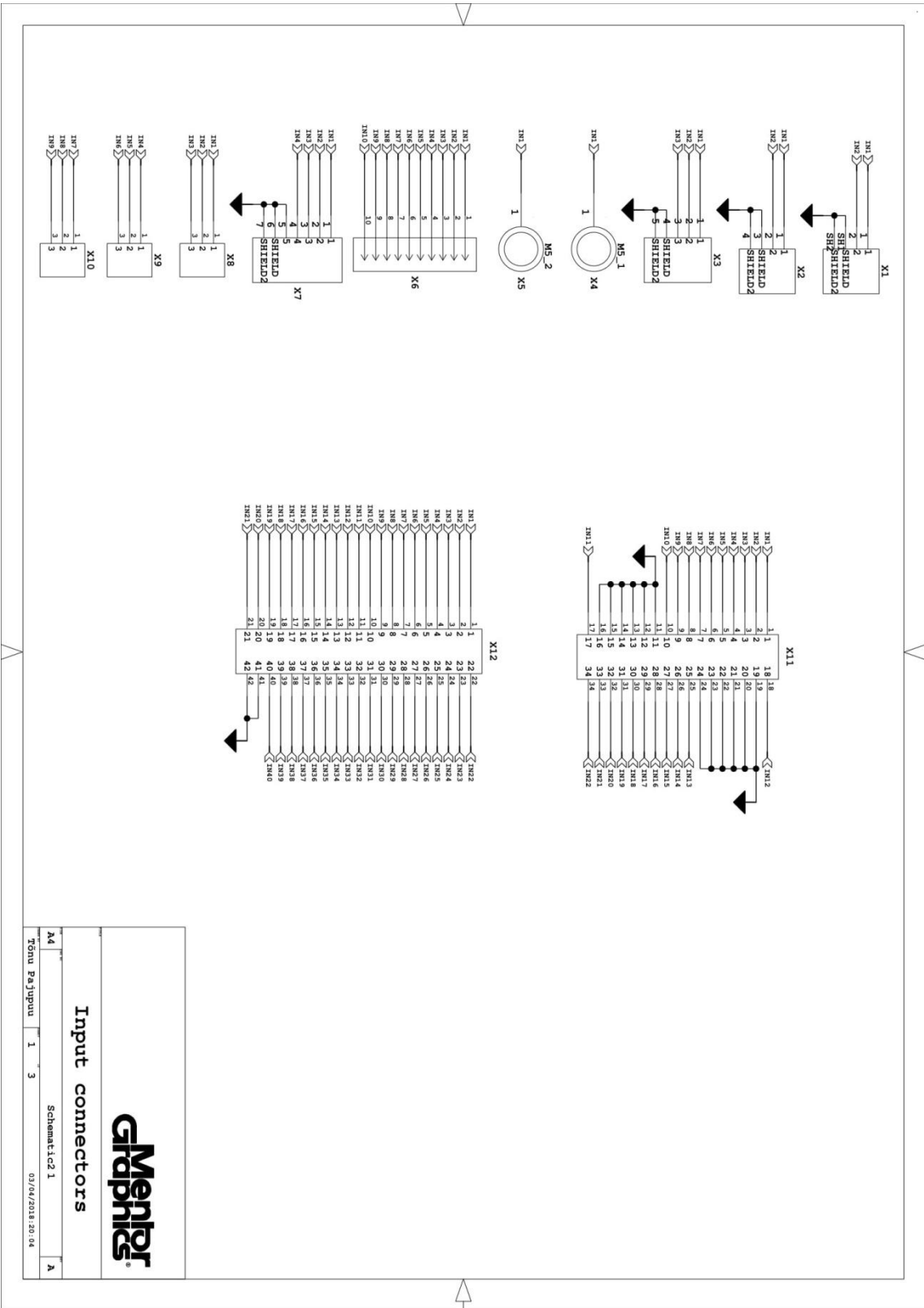
Mentor
Graphics

LCD & Connector board headers

Part Number: **34** Revision: **12** Date: **12**

Customer: **Tosnu Pa-Juppu** Project: **12** Date: **12**

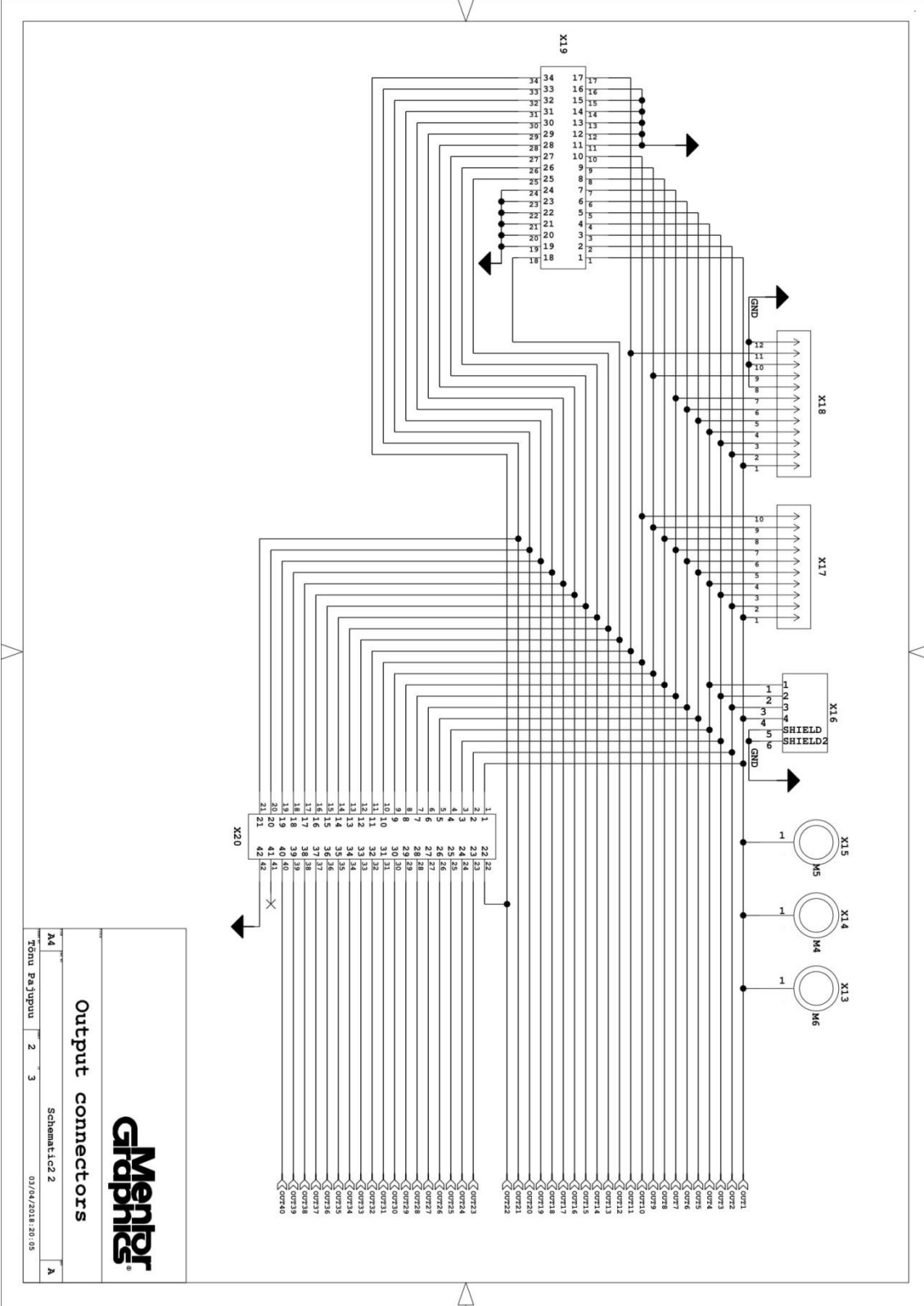
14/05/2018 18:54 A



Input connectors



Schematic2 1
 Tonu PaJuppu 1 3
 03/04/2018 20:04



OUT1	1
OUT2	2
OUT3	3
OUT4	4
OUT5	5
OUT6	6
OUT7	7
OUT8	8
OUT9	9
OUT10	10
OUT11	11
OUT12	12
OUT13	13
OUT14	14
OUT15	15
OUT16	16
OUT17	17
OUT18	18
OUT19	19
OUT20	20
OUT21	21
OUT22	22
OUT23	23
OUT24	24
OUT25	25
OUT26	26
OUT27	27
OUT28	28
OUT29	29
OUT30	30
OUT31	31
OUT32	32
OUT33	33
OUT34	34
OUT35	35
OUT36	36
OUT37	37
OUT38	38
OUT39	39
OUT40	40

X21

IN1	1
IN2	2
IN3	3
IN4	4
IN5	5
IN6	6
IN7	7
IN8	8
IN9	9
IN10	10
IN11	11
IN12	12
IN13	13
IN14	14
IN15	15
IN16	16
IN17	17
IN18	18
IN19	19
IN20	20
IN21	21
IN22	22
IN23	23
IN24	24
IN25	25
IN26	26
IN27	27
IN28	28
IN29	29
IN30	30
IN31	31
IN32	32
IN33	33
IN34	34
IN35	35
IN36	36
IN37	37
IN38	38
IN39	39
IN40	40

X22

Mentor
Graphics

Main board connections

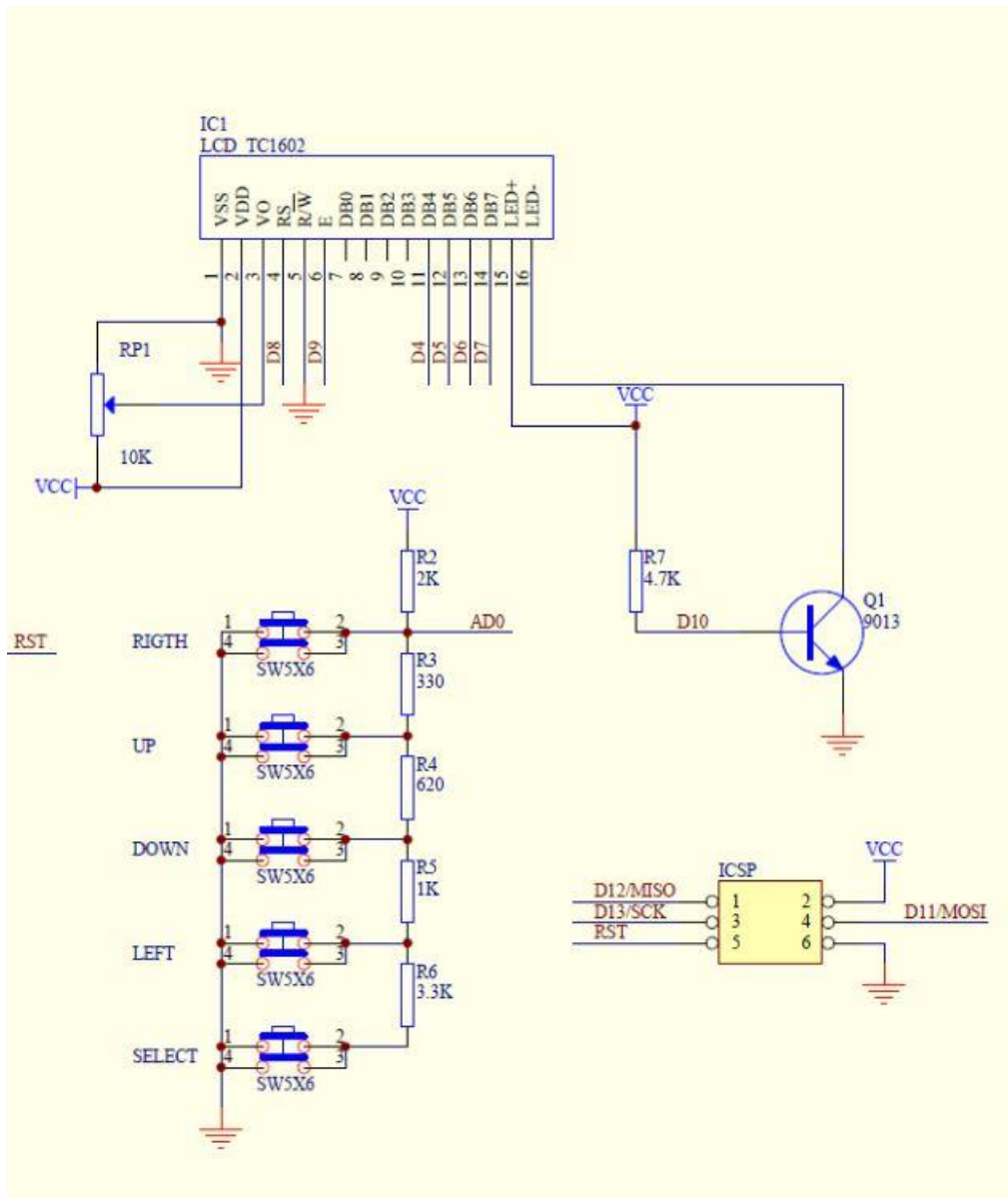
Ad 3 3

Schematic2 3

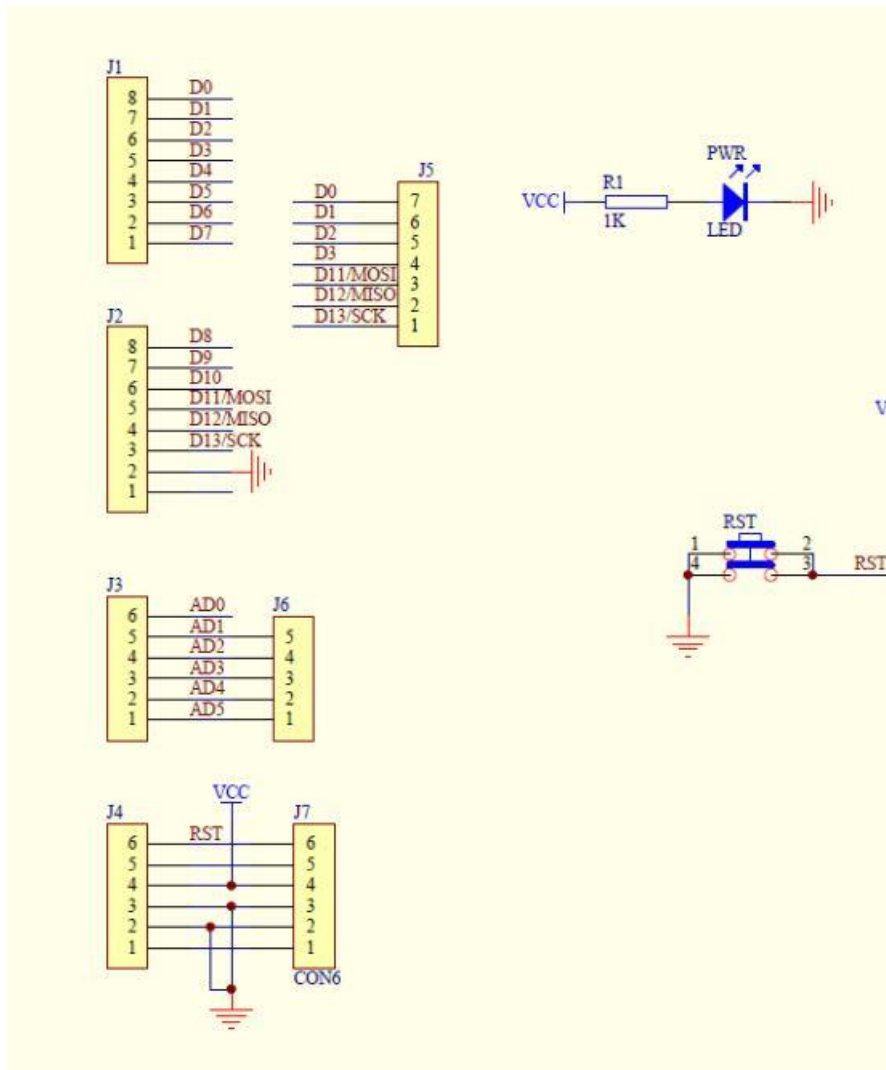
03/04/2018 20:07

A

Lisa 2 – LCD mooduli elektriskeem

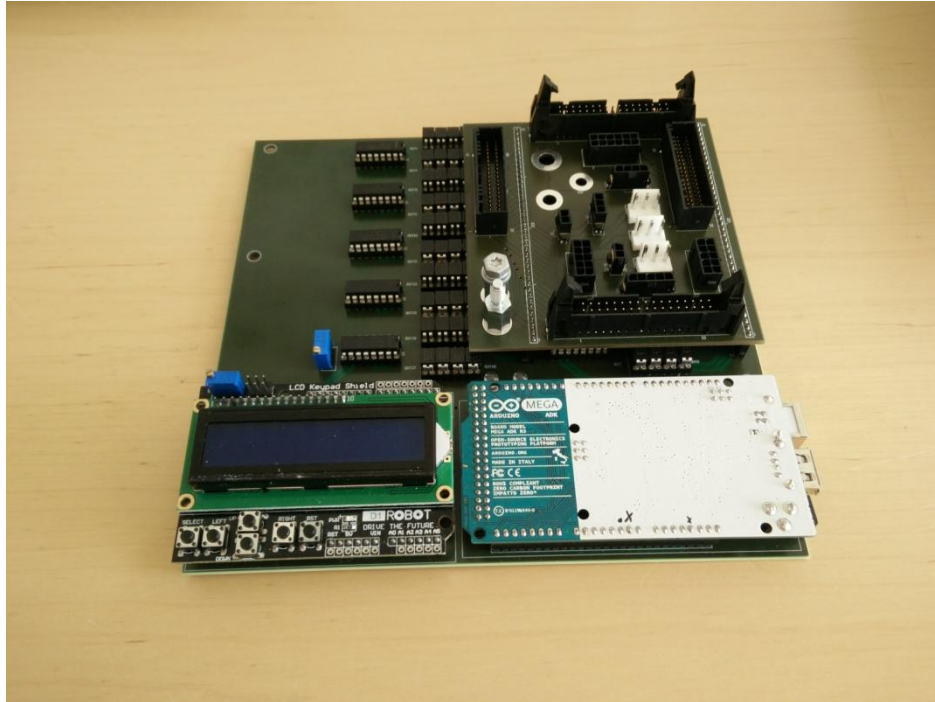


Joonis 17. LCD mooduli ICSP ühendus, klahvistik ja LCD ekraani ühendused [13].



Joonis 18. LCD mooduli LED indikaator, „Reset“ nupp ja piikriba ühendused [13].

Lisa 3 – Testri foto



Joonis 19. Kokku pandud tester.

Lisa 4 – Programmi kood

Programmi kood on tööle kaasa lisatud mälukaardi peal.