TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
Elektroenergeetika ja mehhatroonika instituut

ANDMEHÕIVE JA ENERGIA KOGUMISE ELEKTROONIKASÜSTEEMI ARENDAMINE EKSPERIMENTAALSELE TERMO-ELEKTRILISI ELEMENTE RAKENDAVALE PÄIKSEKOLLEKTORILE

BAKALAUREUSETÖÖ

MEHHATROONIKA ÕPPEKAVA

Juhendaja/õppejõud:	Lauri Kütt
Üliõpilane:	Kristian Kajak
Üliõpilaskood:	142913MAHB

Tallinn 2017

AUTORIDEKLARATSIOON

Deklareerin, et käesolev lõputöö on minu iseseisva töö tulemus.

Esitatud materjalide põhjal ei ole varem akadeemilist kraadi taotletud.

Lõputöös kasutatud kõik teiste autorite tööd ja seisukohad ning materjalid on varustatud vastavate viidetega.

Töö valmis Lauri Kütt juhendamisel

“20” mai 2017.a.

Töö autor: Kristian Kajak

/allkiri/

Töö vastab lõputööle esitatavatele nõuetele

“.....”.....201….a.

Juhendaja: Lauri Kütt

/allkiri/

Lubatud kaitsmisele

“.....”.....201… a.

..... õppekava lõputööde kaitsmiskomisjoni esimees: .....

/allkiri/

TTÜ Inseneriteaduskond

BSc LÕPUTÖÖ ÜLESANNE

2017 aasta kevadsemester

Üliõpilane: Kristian Kajak, 142913MAHB

Õppekava: MAHB 02/13

Eriala: BSc Mehhatroonika

Juhendaja: professor Lauri Kütt

Lõputöö teema:

Andmehõive ja energia kogumise süsteemi arendamine eksperimentaalsele termoelektrilisi elemente rakendavale päiksekollektorile

Development of a data acquisition and energy collecting system for an experimental thermoelectric solar collector

Töös lahendatavad ülesanded ja nende täitmise ajakava:

Nr	Ülesande kirjeldus	Täitmise tähtaeg
1.	Tutvumine päiksekollektori eesmärgiga ning eelnevalt valmistatud elektriskeemidega	Oktoober 2016
2.	Olemasolevate trükkplaatidele komponentide jootmine, projekteerimisvigade parandamine. Tarkvara arendamine	Detsember 2016
3.	Trükkplaatide paranduste sisse viimine, uute trükkplaatide projekteerimine, trükkplaadi tellimuste vormistamine	Veebruar 2017
4.	Uute valminud trükkplaatide jootmine ja vigade tuvastamine ning parandamine. Tarkvara arendamine	Mai 2017
5.	Valmis lõputöö koos juhendaja nõusolekuga	20. Mai 2017

Lahendatavad insenertehnlised ja majanduslikud probleemid:

Trükkplaatide projekteerimisvigade eemaldamine, uute trükkplaatide projekteerimine, tarkvara arendamine andmehõive ja energia kogumise süsteemi arendamiseks.

Täiendavad märkused ja nõuded: –

Töö Keel: Eesti keel

Kaitsmistaotlus esitada hiljemalt: 15.05.2017

Töö esitamise tähtaeg: 25.05.2017

Üliõpilane /allkiri/ kuupäev

Juhendaja /allkiri/ kuupäev

SISUKORD

EESSÕNA 7
LÜHENDITE JA TÄHISTE LOETELU 8
SISSEJUHATUS 9
1 PÄIKSEKOLLEKTOR 10
1.1 Päiksekollektorist üldiselt 10
1.2 Elektroonikasüsteemide klassifitseerimine 11
1.3 Energia kogumise ja andmehõive elektroonikasüsteemi ülesehitus 12
2 SPI PROTOKOLL 13
2.1 SPI protokollist üldiselt 13
2.2 Ülem-alam topoloogia 13
2.3 Andmevahetuse kirjeldus 13
2.3.1 SCLK 14
2.3.2 MOSI 14
2.3.3 MISO 15
2.3.4 SS 15
2.4 SPI seadistamine 16
2.5 SPI muutmine häiringukindlamaks 16
3 TRÜKKPLAATIDE KUJUNDAMINE 18
3.1 Trükkplaatidest üldiselt 18
3.2 Elektriskeemide koostamine 18
3.3 Tootekavandi koostamine 19
3.3.1 Tootekavandi joonestiku suuruse valimine 19
3.3.2 Radade asetus 19
3.3.3 Radade laiused 20
3.3.4 Vasealad ja läbiviigud 21
4 TOODETUD TRÜKKPLAADID 23
4.1 Trükkplaatide tootmisest üldiselt 23
4.2 Parandatud trükkplaadid 23
4.2.1 Informatsiooni kogumise trükkplaat 23
4.2.2 Informatsiooni kogumise trükkplaadile tehtud parandused 24
4.2.3 Analoog-digitaalmuunduri trükkplaat 25
4.2.4 Analoog-digitaalmuunduri trükkplaadile tehtud parandused 26
4.3 Projekteeritud trükkplaadid 27
4.3.1 Võimendi ja pingejagurite trükkplaat 27
4.3.2 Optoisolatsiooni trükkplaat 28
4.3.3 GPS mooduli ja SD-kaardi mooduli trükkplaat 30
5 TARKVARA 32
5.1 Tarkvarast üldiselt 32
5.2 Informatsiooni kogumise trükkplaadi tarkvara 32

5.2.1 Informatsioonikogumise trükkplaadi tarkvarast üldiselt ... 32
5.2.2 Multipleksor trükkplaadi valimise alamfunktsioon ... 33
5.2.3 MCP23S17 registrite konfigureerimise alamfunktsioon ... 33
5.2.4 MAX31855 temperatuuri lugemise alamfunktsioonid ... 35
5.2.5 MCP4912 digitaal-analoogmuunduri alamfunktsioonid ... 35
5.2.6 LTC1861 analoog-digitaalmuunduri alamfunktsioonid ... 36
5.3 Analoog-digitaalmuunduri trükkplaadi tarkvara ... 37
5.3.1 Analoog-digitaalmuunduri trükkplaadi tarkvarast üldiselt ... 37
5.3.2 ADS1248 registrite konfigureerimise alamfunktsioonid ... 37
5.3.3 ADS1248 registrite väärtuse lugemise alamfunktsioon ... 38
5.3.4 ADS1248 sisemise temperatuuri lugemise alamfunktsioon ... 38
5.3.5 ADS1248 sisendpinge väärtuse lugemise alamfunktsioon ... 38
5.3.6 Muud ADS1248 alamfunktsioonid ... 39
6 ELEKTROONIKAKEST JA TRÜKKPLAATIDE KINNITUSED ... 40
KOKKUVÕTE ... 41
SUMMARY ... 42
7 KASUTATUD KIRJANDUS ... 43
LISA 1 TERVLIKLIK ENERGIA KOGUMISE JA ANDMEHÕIVE ELEKTROONIKASÜSTEEMI ÜLESEHITUS ... 46
LISA 2 INFORMATSIOONI KOGUMISE TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND ... 47
LISA 3 ANALOOG-DIGITAALMUUNDURI TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND ... 53
LISA 4 VÕIMENDI JA PINGEJAGURITE TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND ... 57
LISA 5 CPC1832 SENSORITE TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND ... 60
LISA 6 OPTOISOLATSIOONI TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND ... 63
LISA 7 GPS JA SD-KAARDI MOODULITEGA TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND ... 67
LISA 8 ALAMA ADRESSEERIMISE ALGORITM ... 71
LISA 9 MULTIPLEKSORI TRÜKKPLAADI VALIMISE ALAMFUNKTSIOON ... 72
LISA 10 MCP23S17 REGISTRITE KONFIGUREERIMISE ALAMFUNKTSIOON ... 73
LISA 11 MAX31855 TEMPERATUURI LUGEMISE ALAMFUNKTSIOONID ... 74
LISA 12 MCP4912 DIGITAAL-ANALOOGMUUNDURI ALAMFUNKTSIOONID ... 75
LISA 13 LTC1861 ANALOOG-DIGITAALMUUNDURI ALAMFUNKTSIOONID ... 76
LISA 14 ANALOOG-DIGITAALMUUNDURI TRÜKKPLAADI TARKVARA MULTIPLEKSOR TRÜKKPLAADI VALIMISE, MCP23S17 REGISTRITRE KONFIGUREERIMISE JA MAX31855 TEMPERATUURI LUGEMISE ALAMFUNKTSIOONID ... 78
LISA 15 ADS1248 REGISTRITE KONFIGUREERIMISE ALAMFUNKTSIOONID ... 81
LISA 16 ADS1248 REGISTRITE LUGEMISE ALAMFUNKTSIOONID ... 82

LISA 17 ADS1248 SISEMISE TEMPERATUURI LUGEMISE ALAMFUNKTSIOON 83
LISA 18 ADS1248 SISENDPINGE LUGEMISE ALAMFUNKTSIOON 84
LISA 19 MUUD ADS1248 ALAMFUNKTSIOONID 85
LISA 20 ELEKTROONIKAKESTA SOLIDWORKS MUDEL 86

EESSÕNA

Käesolev bakalaureuse lõputöö teema pakuti välja TTÜ Elektroenergeetika ja mehhatroonika instituudi professor Lauri Kütt'i poolt. Lõputöös kirjeldatud päiksekollektori elektroonikaosale määras tema nõuded ja vajadusel abistas elektroonikaskeemide koostamisel ja trükkplaatide programmeerimisel. Lõputöö koostamine toimus Elektroenergeetika ja mehhatroonika instituudi tööruumides. Avaldan juhendajale tänu meeldiva koostöö eest.

LÜHENDITE JA TÄHISTE LOETELU

ADC	Analoog-digitaalmuundur, i.k. *Analog-To-Digital Converter*
DAC	Digitaal-analoogmuundur, i.k. *Digital-To-Analog Converter*
LED	Valgusdiood, i.k. *Light Emitting Diode*
LSB	Vähima kaaluga bitt, i.k. *Least Significant Bit*
LVDS	Madala pingega signaalide diferentseerimine, i.k. *Low Voltage Differential Signaling*
MISO	SPI ülem-sisse-alam-välja liin, i.k. *Master-In-Slave-Out*
MOSFET	Isoleeritud paisuga väljatransistor, i.k. *Metal Oxide Semiconductor Field-Effect Transistor*
MOSI	SPI ülem-välja-alam-sisse liin, i.k. *Master-Out-Slave-In*
MSB	Suurima kaaluga bitt, i.k. *Most Significant Bit*
PCB	Prinditud trükkplaat, i.k. *Printed Circuit Board*
SCLK	SPI kellatakti liin, i.k. *Serial Clock*
SPI	Jadamisi välisseadmete vaheline andmevahetuse standard, i.k. *Serial Peripheral Interface*
SS	SPI alama valimise liin, i.k.*Slave Select*
GPIO	Programmeeritav sisend/väljund, i.k. *General Purpose Input/Output*

SISSEJUHATUS

TTÜs on arendamisel eksperimentaalne termoelektriline päikesekollektor, mis rakendab fokuseeritud päikesekiirgust kõrge temperatuuri saavutamiseks. Antud päiksekollektor kasutab fokuseeritud päiksekiirgust termoelektriliste generaator-elementide kuumutamiseks, mille tagajärjel hakkavad termoelektrilised elemendid elektrienergiat tootma. Päikesekollektori poolt toodetava energia hindamiseks on vaja arendada energiakogumise süsteem, andmehõive süsteem ning päikse liikumise trajektoori järgimise süsteem, mis oleks on antud süsteemi käivitamiseks vajalik.

Antud lõputöö raames on energia kogumise ja andmehõive elektroonikasüsteemi arendamiseks vaja projekteerida ja toota hulgaliselt trükkplaate, arendada nende juhtimiseks välja tarkvara ning konstrueerida antud elektroonikasüsteemi trükkplaatide hoiustamiseks vajalik kest. Teooria osas käsitletakse lähemalt SPI protokolli ja trükkplaatide projekteerimise tavasid ja reegleid. Praktilises osas kirjeldatakse projekteeritud ja parandatud trükkplaatide tööpõhimõtteid ja elektriskeeme, tarkvara arendamisel kasutatud alamfunktsioone ning elektroonikakesta mudeli ülesehitust.

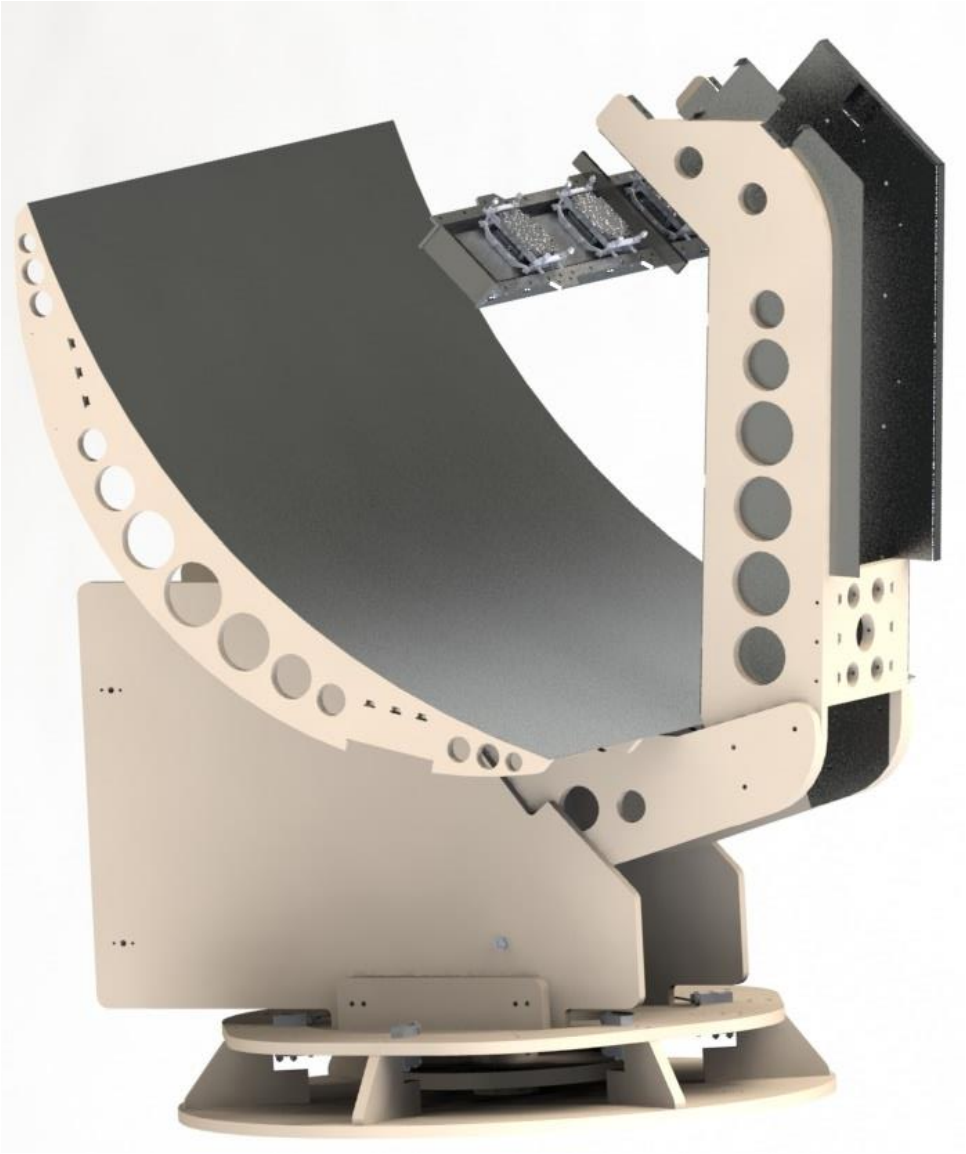
Antud lõputöö eesmärgi saavutamiseks on vaja olemasoleva päiksekollektori andmehõive ja energia kogumise süsteemi arendamiseks koostada Diptrace programmis trükkplaatidele juhtimist võimaldavad elektroonikaskeemid, luua sobiv signaaliradade ja komponentide asetus tootmise jaoks, komplekteerida, testida ning vajadusel parandada trükkplaatidel olevaid vigu.

Arduino IDE 1.8.2 programmi ja Arduino funktsiooniteekide põhjal kirjutati informatsiooni kogumise ja analoog-digitaalmuunduri trükkplaatide juhtimiseks tarkvara. Antud sardsüsteemi juhib Arduino Mega 2560. Energia kogumise ja andmehõive süsteemi jaoks vajalikud trükkplaadid paigutati nende jaoks projekteeritud elektroonikakarpi.

1 PÄIKSEKOLLEKTOR

1.1 Päiksekollektorist üldiselt

Antud lõputöö raames arendati termoelektrilisi elemente rakendavale päiksekollektorile energia kogumise ja andmehõive elektroonilist süsteemi. Lihtsustatult öeldes koosneb päiksekollektor päikese kiirgust koondavast paraboolikujulisest nõgusast peeglist, soojuse kogumise konstruktsioonist ning pöörlevast/kallutatavast alusest (sele 1.1). Päiksekollektori töö jaoks on vaja ka veel näiteks jahutusagregaati, erinevaid juhtloogika alamosasid, akude haldamise süsteemi jpm aga neid teemasid käesolevas lõputöös ei kajastata.



Sele 1.1 Päiksekollektori üldvaade

Selleks, et termoelektrilisi elemente rakendata, on vaja projekteerida elektrooniline süsteem, mille funktsioonideks on

- päikese asukoha tuvastamine,
- päiksekollektori pööramine/kallutamine päikse poole (automaat ja manuaalrežiimis),
- termoelektrilisi elemente läbiva voolutugevuse ja arendatava pinge mõõtmine ning võimsuse edasikandmine akudesse,
- päiksekollektori pööramine ohutusse kohta (vea korral),
- andmete logimine SD-kaardile.

Kuna projekteeritud päiksekollektor peaks arvutuste kohaselt kuumutama termoelektriliste elementide kuumad pooled kuni 300 °C-ni, tuleb elektroonikasüsteemi projekteerimisel pidada silmas, et rikke ilmnemisel võib suure tõenäosusega tekkida tuleohtlik olukord koondatud valguskiirte tõttu. Seetõttu tuleb projekteerida antud elektroonikasüsteem võimalikult häiringukindlaks ning vajadusel kasutada mitut sama funktsiooni omavat trükkplaati.

1.2 Elektroonikasüsteemide klassifitseerimine

Suures plaanis võib tervikliku päiksekollektori kasutamiseks vajaliku elektroonikasüsteemi jagada neljaks:

- energia kogumise ja andmehõive süsteem,
- mootorite juhtimise süsteem,
- aku haldamise ja tervikliku päiksekollektori juhtimise süsteem,
- jahutussüsteem.

Energia kogumise ja andmehõive süsteem vastutab termoelektrilistelt elementidelt tuleva võimsuse mõõtmise ja edasikandmise eest, päikse asukoha tuvastamise eest ja andmete logimise eest. Suurem osa energiakogumise ja andmehõive süsteemist asub soojuse kogumise konstruktsiooni taga olevas elektroonikakestas. Ainukese erandina ei asu GPS-mooduli ja SD-kaardi mooduliga trükkplaat päiksekollektori peal.

Mootorite juhtimise süsteem vastutab kahe samm-mootori juhtimise eest. Ühe samm-mootori abil pööratakse päiksekollektorit aluse peal ning teise samm-mootori abil muudetakse paraboolpeegli nurga horisontaali suhtes. Mootorite juhtimise süsteem asub päiksekollektori aluse küljes.

Aku haldamise ja tervikliku päiksekollektori juhtimise süsteem vastutab akude laadimise ohutuse ja päiksekollektori tervikliku funktsioneerimise eest. Antud süsteemis kasutatav Arduino Mega 2560 platvorm on kõikide teiste süsteemis olevate sardüksustega ühenduses ning erinevad elektroonikaüksused ei asu päiksekollektori peal.

Jahutussüsteem vastutab termoelektriliste elementide külmemate poolte jahutamise eest. Jahutussüsteemi agregaadis asub ka näiteks veepump, radiaator ning nende juhtimise süsteem, mis reguleerib voolu kiirust vastavalt termoelektriliste elementide temperatuurile. Sarnaselt aku haldamise süsteemile, asub jahutussüsteem päiksekollektorist eemal.

1.3 Energia kogumise ja andmehõive elektroonikasüsteemi ülesehitus

Energia kogumise ja andmehõive elektroonikasüsteem koosneb järgnevatest trükkplaatidest:

- Informatsioonikogumise trükkplaat
- Analoog-digitaalmuunduri trükkplaat
- Multipleksori trükkplaat
- Võimendi ja pingejagurite trükkplaat
- GPS mooduli ja SD-kaardi mooduli trükkplaat
- Optoisolatsiooni trükkplaat
- CPC1832 päiksesensori trükkplaat

Antud lõputöö üheks ülesandeks oli nimekirjas toodud trükkplaatidest esimesele kahele teostada parandused, kolmandal trükkplaadil vigu ei esinenud ning neli viimast trükkplaati oli vaja projekteerida ja toota. Terve süsteemi juhtimiseks kasutatakse Arduino Mega 2560 platvormi ning Arduino IDE 1.8.2 programmis loodud tarkvara.

Kõik nimetatud trükkplaadid välja arvatud GPS mooduli ja SD-kaardi mooduli trükkplaat asuvad soojuse kogumise konstruktsiooni küljes. Analoog-digitaalmuunduri, multipleksor, optoisolatsiooni ning Arduino trükkplaatide jaoks oli vaja projekteerida elektroonikakest. Informatsioonikogumise ja võimendi trükkplaadid hoiustatakse ostutoodetest karpides.

CPC1832 päiksesensoritega trükkplaadid asuvad „vihmavarju“ konstruktsiooni peal erinevate kallete all (sele 1.2). Seoses päikse ja CPC1832 sensori vahelise nurga muutmisega, on võimalik päiksesensorilt tuleneva pinge suurusi võrreldes tuvastada päikse asukoht, mida kasutatakse päiksekollektori pööramiseks päikse poole. Antud konstruktsioone on päiksekollektoril kaks, et vältida segadusi näiteks varjude langemisel sensoritele.

Sele 1.2 CPC1832 sensorite „vihmavarju“ konstruktsioon

Terviklik energia kogumise ja andmehõive elektroonikasüsteemi ülesehitus on kujutatud lisas 1.

2 SPI PROTOKOLL

2.1 SPI protokollist üldiselt

SPI *(Serial Peripheral Interface*) on sünkroniseeritud kommunikatiooniprotokoll, millega on võimalik vahetada digitaalsel kujul informatsiooni lühikeste andmeliinide kasutamisel. Seetõttu on SPI-siin laialdaselt kasutuses sardsüsteemide koostamisel, andes arendajatele võimaluse lihtsalt enda süsteemi liita näiteks vedelkristallikuvareid, SD kaardi lugejaid/kirjutajaid, analoog-digitaalmuundureid jpm. [1]

2.2 Ülem-alam topoloogia

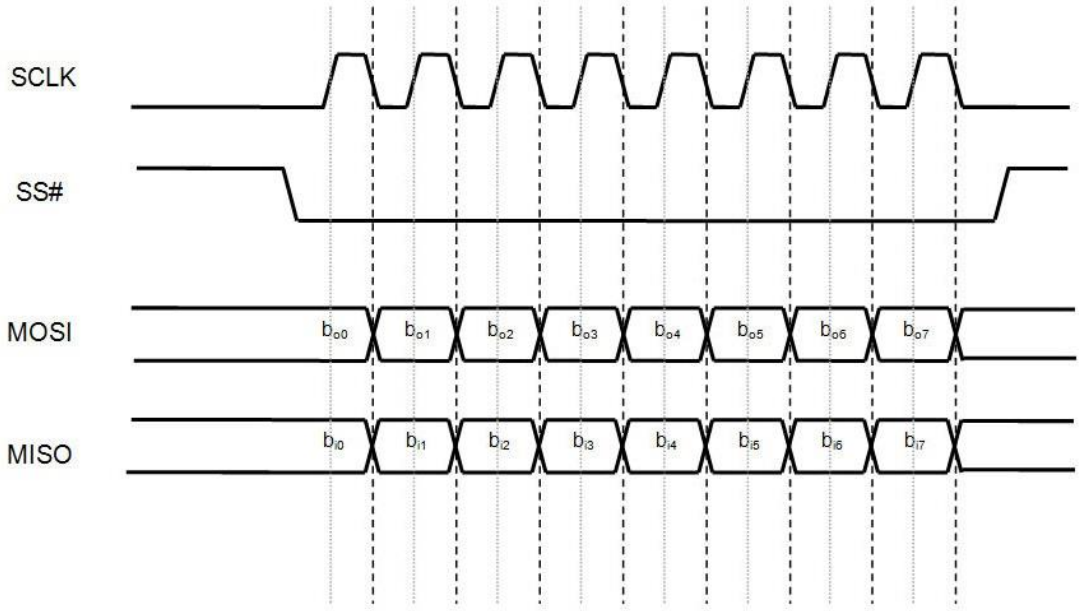
SPI andmeside protseduurides kasutakse ülem-alam topoloogiat, kus ülem on üldiselt mingisugune võimekam mikrokontroller, mis reguleerib info kirjutamist ja lugemist. Alam on mikrolülitus, mis kasutab loeb/töötleb/rakendab ülema poolt saadud informatsiooni ning saadab vajalikud andmed ülemale tagasi.

Näiteks on võimalik SPI protokolli kasutada ATmega mikrokontrolleri ja analoog-digitaalmuunduri vaheliseks suhtluseks, kus ATmega mikrokontroller on ülem ning analoog-digitaalmuundur on alam. ATmega mikrokontroller saadab analoog-digitaalmuundurile käsu lugeda väliselt toiteahelalt tuleva pinge väärtust ning transformeerida see digitaalseks informatsiooniks. Seejärel on järgmise informatsioonilausega võimalik ATmega mikrokontrolleril lugeda analoog-digitaalmuunduri poolt saadetud digitaalne informatsioon.

2.3 Andmevahetuse kirjeldus

SPI protokoll võimaldab kahesuunalist andmevahetust samaaegselt, kasutades tavaliselt nelja andmeliini (sele 2.1):

- SCLK *(Serial Clock*) – taktsignaali liin,
- MOSI (*Master Out Slave In*) – ülema poolt saadetud informatsioon,
- MISO (*Master In Slave Out*) – alama poolt saadetud informatsioon,
- SS (*Slave Select*) – liin, mis valib andmevahetuse jaoks õige alama.



Sele 2.1 SPI kommunikatsioon [4]

2.3.1 SCLK

SCLK *(Serial clock)* on kellatakti liin, mis määrab ära ajahetked, millal informatsioonijada ülemast või alamast välja saadetakse või sisse loetakse. Kellatakt on perioodi esimeses pooles kõrgel nivool ning perioodi teises pooles madalal nivool ning kellatakti perioodi kestvust on võimalik reguleerida programmikoodis.

SPI protokolli kasutades tuleb ka ära märkida kellatakti polaarsus ning faas. Kellatakti polaarsus ning faas määravad ära, kas informatsioon MISO/MOSI liinile edastatakse kellasignaali tõusval või langeval frondil. Samuti määratakse ära, kummal frondil MISO/MOSI liinil olev informatsioon loetakse.

Kuna käesolevas lõputöös kasutatakse programmeerimisel Arduino IDE 1.8.2 tarkvara, on võimalik kellatakti polaarsus ning faas valida *SPI_MODE0...SPI_MODE3* konfiguratsioone kasutades (tabel 2.1). [2]

Tabel 2.1. Kellatakti polaarsuse ja faasi konfigureerimine Arduino IDE 1.8.2 tarkvaras

Konfiguratsioon	Polaarsus	Faas	Info edastamise front	Info lugemise front
SPI_MODE0	0	0	Langev	Tõusev
SPI_MODE1	0	1	Tõusev	Langev
SPI_MODE2	1	0	Tõusev	Langev
SPI_MODE3	1	1	Langev	Tõusev

2.3.2 MOSI

MOSI *(Master Out Slave In)* andmeliinil kantakse ülemast välja tulevat informatsiooni erinevate alamate vahel edasi. Alam, kellega ülem suhelda soovib, valitakse *Slave Select* liini abil. Ülem-alam vaheline suhtlus võib toimida samaaegselt.

2.3.3 MISO

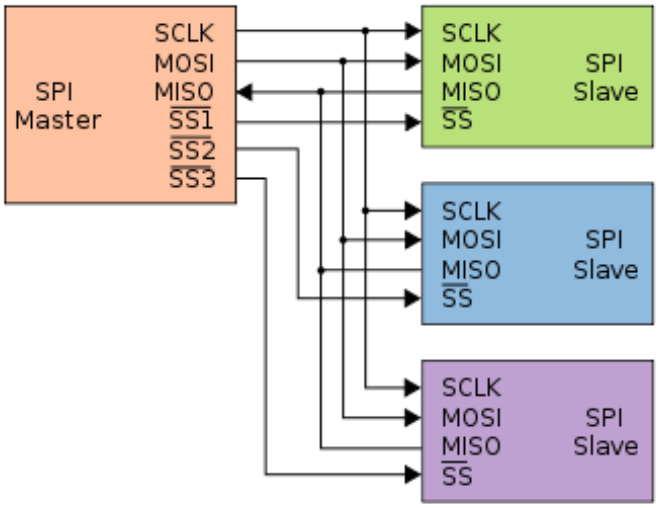
MISO *(Master In Slave Out)* andmeliinil kantakse alamast välja tulev informatsioon ülemasse. Ka MISO kanaliga suhtlemise jaoks peab olema *Slave Select* valinud õige alama, muidu ei tuvasta ülem saadetud informatsiooni.

2.3.4 SS

SS *(Slave Select)* on liin, mis määrab ära millise alamaga ülem parajasti suhtleb. SS muutub tähtsaks kui andmevahetussüsteemis kasutatakse mitut alamat ning ühe alama kasutamise korral võib SS olla konstantselt madal nivool. Tüüpilistes SPI protokolli kasutavates andmevahetus süsteemides kasutatakse *active low* loogikat, mis tähendab, et alam valitakse juhul, kui selle alama SS on madalal nivool. Enne andmevahetuse alustamist, sätitakse valitud alama SS madalale nivoole, mis aktiveerib alama. Pärast andmevahetuse lõppu sätitakse alama SS kõrgele nivoole, mis lõpetab edasise andmevahetuse valitud alamaga. [3]

Alamaid on võimalik tüüpiliselt adresseerida kahte moodi. Kõige tüüpilisem viis on kasutada iga alama jaoks eraldi SS liini (sele 2.2), mille puhul on alamate adresseerimine lihtne. Andmevahetussüsteemides, kus kasutatakse suur arv alamaid, võib tekkida probleeme, sest paljudel mikrokontrolleritel ei ole suurte süsteemide korral piisavalt vabu liine, mida alamate adresseerimiseks kasutada. Siiski on võimalik probleemi lahendada kasutades näiteks 3→8 dekoodreid, mis võimaldavad SS väljundite hulga kordistamist. [3]

Keerulisem on kasutada *Daisy Chain* adresseerimise viisi, mille korral on alamad ühendatud jadamisi ning neile on võimalik saata käsklusi ilma aadressideta kõikidele alamatele korraga. *Daisy Chain* adresseerimise puhul on siiski MISO liini kasutamine raskendatud ning seetõttu päiksekollektori elektrisüsteemi arendamisel seda ei kasutatud. [3]



Sele 2.2 Alamate adresseerimine mitmete SS liinide abil [1]

2.4 SPI seadistamine

Eelnevas peatükis oli juttu kellatakti faasi ja polaarsuse määramisest. Enne SPI protokolli kasutamist tuleb veel mõned seadistused teha. SPI protokolli kasutades tuleb selgeks teha kasutatava mikrokontrolleri kellatakti sagedus – liiga suure SPI andmevahetuse kiiruse valimine tähendab, et kontroller ei suuda sissetulevat informatsiooni lugeda ega väljaminevat informatsiooni õigeks kellatakti frondiks valmis seada. Liiga kiire andmevahetuse sageduse valimisel võib tekkida ka probleeme infoedastamise sünkronisatsiooniga pikemate andmeliinide kaudu. Seetõttu tuleks 1 m või pikema andmeliini puhul oluliselt langetada andmeedastamise kiirust või kasutada spetsiaalseid draivereid [3].

SPI protokolli kasutades tuleb määrata ära, kas andmeedastust alustatakse vähima kaaluga bitist *(LSB)* või suurema kaaluga bitist *(MSB)*. Arduino IDE 1.8.2 ja selle funktsioone kasutades saab seda teha käskudega *MSBFIRST* või *LSBFIRST*. SPI protokolli seadistamine Arduino IDE 1.8.2 tarkvaras näeks tavapäraselt välja järgmiselt:

SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPIMODE_0));

kus *160000* – andmeedastamise sagedus hertsides, Hz,

MSBFIRST – määrab ära andmeedastamise bittide järjekorra,

SPIMODE_0 – määrab ära kellatakti polaarsuse ja faasi.

2.5 SPI muutmine häiringukindlamaks

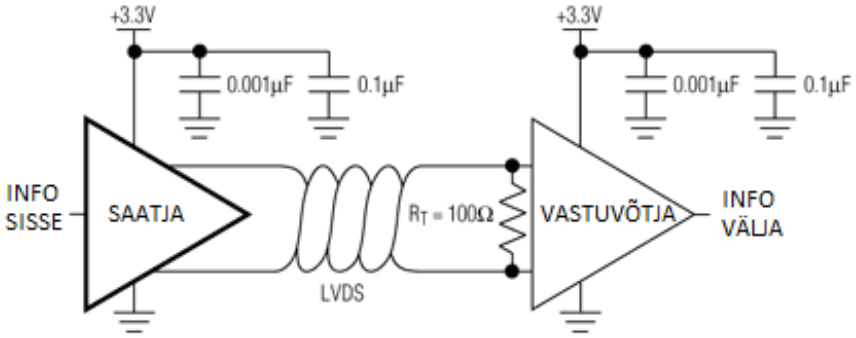
SPI protokolli korral ei ole andmeliinid balansseeritud, st et nii madalale nivoole kui kõrgele nivoole on seatud kindlad pinge väärtused maanduse suhtes. Tavapäraselt on lihtsates süsteemides madal nivoo 0 V maanduse suhtes ning kõrge nivoo 5 V või 3,3 V maanduse suhtes.

Balansseerimata andmevahetussüsteemide korral võib info saatjatel (ehk draiveritel) ja vastuvõtjatel tekkida probleeme informatsioonijada vahendamisega pikkade infoliinide korral. Pikad juhid on rohkem vastuvõtlikumad välistele magnetvälja mõjutustele ning seetõttu tekivad nendes juhtides sagedamini pingenihked. Selliste pingenihete eemaldamiseks sünkroniseeritud, mõlemasuunalise suhtlusega protokollides nagu on SPI protokoll, võib kasutada näiteks RS-422 või RS-644 standardi toetusega diferentsiaaldraivereid koos keerdpaarjuhtmetega. [5, lk 17]

Kuna keerdpaarjuhtmete korral on mõlemad juhid üksteisele võimalikult lähedal, on välismüra indutseeritud mõlemad juhis võrdselt ning andmevahetuse draiverid on võimelised selle tuvastama pingenihkena. Selliste diferentsiaaldraiverite implementeerimine süsteemi koos keerdpaardjuhtmetega ei kaitse andmevahetuse diferentsiaalsignaale üksteise vaheliste pingemuutuste eest, ent siiski vähendab oluliselt välismüra mõju juhtidele ning võimaldab eirata probleeme, mis võivad tekkida pingenihete tekkimisega maanduse suhtes. [5, lk 17]

Selleks et vältida andmeedastamise süsteemis juhtide käitumist antennina, kasutatakse näiteks RS-422 või RS-644 standardeid, mille korral koormatakse diferentseeritud signaalid omavahel 100 Ω takistiga (sele 2.3), mis on samaväärne juhtide lainetakistusega. Diferentsiaalsignaale koormava takisti puudumisel võib andmeliin hakata käituma antennina, mis võib salvestada juhi sisse laenguid, mis hakkavad segama andmevahetust. [5, lk 19]

Käesolevas andmeedastus süsteemis kasutatakse RS-644 standardit, sest RS-644 ehk LVDS standard võimaldab RS-422 standardiga võrreldes kiiremat andmevahetust ning madalamat energiatarvet. RS-644 standard kasutab signaalide diferentseerimisel 3,3 V toitepinget ning diferentsiaalsignaalide omavaheline pinge jääb vahemikku 247 mV – 454 mV 100 Ω takistuse korral. RS-644 standardi signaali vastuvõtjad suudavad tuvastada signaale juba alates ±100 mV juures, kui signaalid on maandusega ±1 V nihkes. [6, lk 1]



Sele 2.3 RS-644 standardi kasutamine signaalide diferentseerimisel koos keerdpaarkaabliga [7]

3 TRÜKKPLAATIDE KUJUNDAMINE

3.1 Trükkplaatidest üldiselt

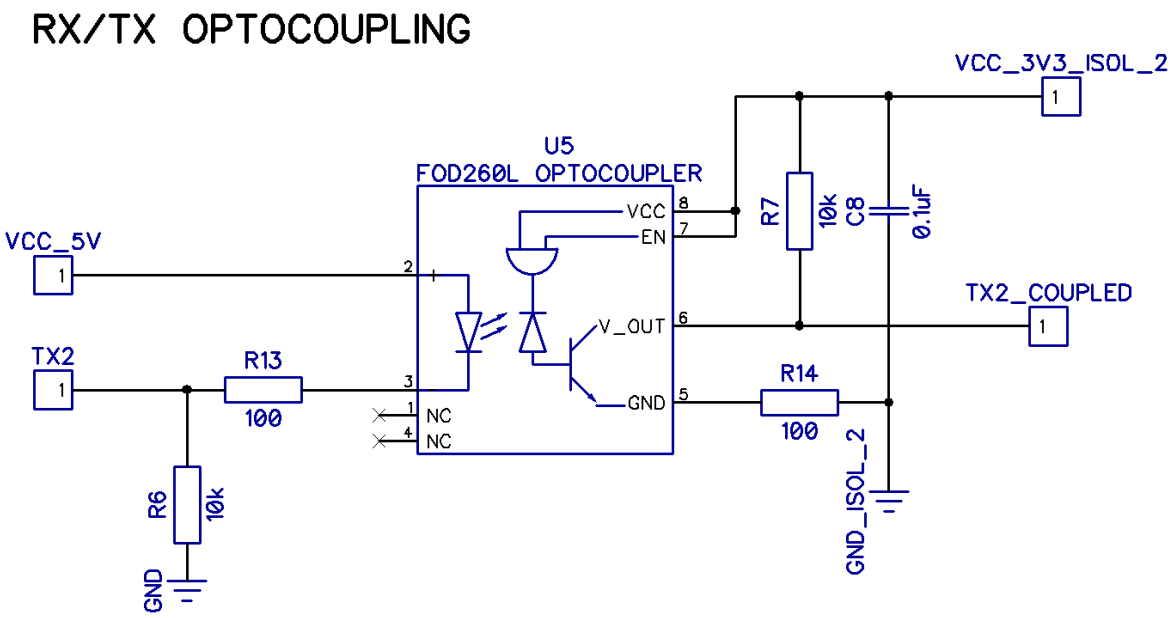
Trükkplaat (ingl.k. *Printed circuit board* – PCB) ühendab elektriskeemi osad nii mehaaniliselt kui ka elektriliselt, kasutades selleks juhtivaid radasid ning vasealasid. Elektrisüsteemi komponendid joodetakse trükkplaadil asetsevate vasealade külge, luues tervikliku ning funktsionaalse elektriskeemi. [8]

Trükkplaadi ühte juhtivaid osi sisaldavat tasandit nimetatakse kihtideks ning trükkplaate võib olla nii ühekihilisi, kahe- või enamakihilisi (trükkplaadi kihtide arv võib ulatuda näiteks 64-ni). Eestis projekteeritakse ABB AS Eesti kontsernis näiteks mh 16-kihilisi trükkplaate, mis võimaldab suuremat komponentide asetuse tihedust. Lihtsamate elektriskeemide realiseerimiseks üldiselt nii suurte kihtide arvuga trükkplaate vaja ei lähe. Suur osa lihtsamaid skeeme saab realiseerida juba ühe, kahe või nelja kihiga. Trükkplaatide kihte ühendatakse läbiviikudega *(via)* , mille abil on võimalik luua erinevatel kihtidel olevate signaali-, toite- ning maandusradade vahel ühendusi.

3.2 Elektriskeemide koostamine

Enne trükkplaatide projekteerimise alustamist, peab olema arusaam trükkplaadi funktsioonist ning selle jaoks peab olema ka terviklik ning täpne elektriskeem, mille alusel trükkplaat valmib. Kui skeem on ebatäpne või valesti koostatud, ilmuvad samad vead ka füüsilisele trükkplaadile ning hiljem tuleb teostada tõrkeotsinguid, mis võivad võtta kordades rohkem aega kui esialgse vea parandamine juba eos.

Seetõttu on soovituslik koostada elektriskeemid võimalikult selgelt, loogilise järjestusega ja mõistliku paigutusega, sest järgmises trükkplaadi tootmise etapis, tootekavandi koostamisel, on keerulisem jälgida, kuidas erinevad juhtivad rajad funktsionaalsuse tagamiseks kasutuses on. Hea tava on näiteks asetada sisendid elektriskeemi vasakule ja väljundid paremale servale ning pinget siluvad kondensaatorid kohe komponendi kõrvale (sele 3.1). Skeemiosade ning ka signaali- ja toiteradade juurde tuleb märkida nende üldine eesmärk või pealkiri, mis annab juba skeemile peale vaadates aimu, millega tegu on [9, lk 5].



Sele 3.1 FOD260L komponendil põhinev optoisolatsiooni ahel, kus sisendid on vasakul ja väljundid on paremal skeemiosas

3.3 Tootekavandi koostamine

Pärast elektriskeemi valmimist tuleb hakata selle alusel looma tootmiskavandit. Tootmiskavandis on kõik vajalik informatsioon olemas trükkplaadi tootmise jaoks, sh signaali- ja toiteradade kulgemine, komponentide asetus, komponentide mõõtmed ning ka komponentide järjekorranumbrid.

3.3.1 Tootekavandi joonestiku suuruse valimine

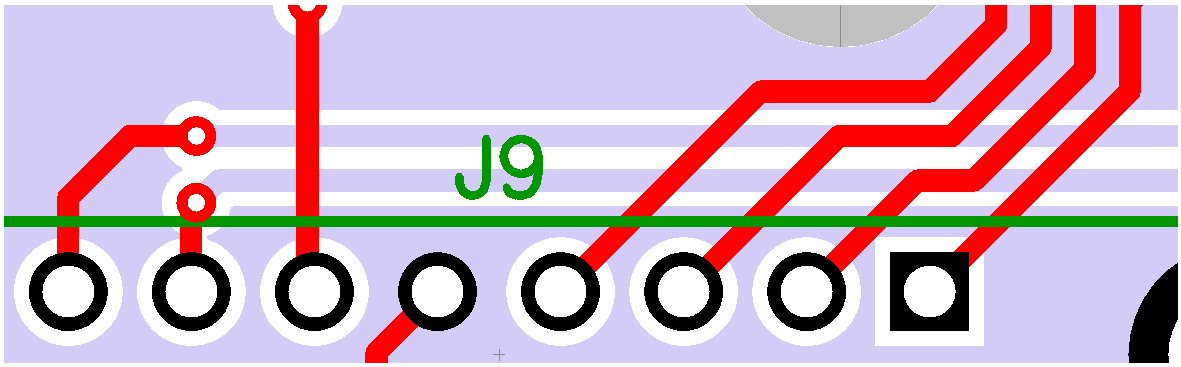
Tootmiskavandit koostades tuleks esialgu valida sobilik fikseeritud joonestiku suurus. Kui tootekavandi koostamise käigus teha joonestiku suuruse valikus muudatusi, hakkab komponentide asetus ebaesteetiliselt segaseks ning asümmeetriliseks muutuma. Lisaks muudab ühtse joonestiku suuruse kasutamine komponentide ning radade redigeerimise mugavaks.

Kuna pindmonteeritavatel komponentidel kasutatakse tihti standardseid suuruseid nii korpuse mõõtmetel kui ka jalgade mõõtmetel, tuleks lähtuda joonestiku valimisel nendest suurustest. Väga paljud pindmonteeritavate elektrikomponentide tootjad kasutavad jalga asetuse määramisel tollimõõdustikku ning seetõttu on nende komponentide jalgade vahemaad murdosad ühest tollist. Näiteks on standardse SOIC korpuse korral kahe jala vaheline vahemaa ½ tollist ning MSOP või TSSOP korpuse korral ¼ tollist. Neid mõõtmeid tuleks joonestiku suuruse valimisel arvesse võtta, kui soovitakse esteetilist lahendust [9, lk 6, 10].

3.3.2 Radade asetus

Trükkplaatide projekteerimisel on soovituslik asetada pealmisel trükkplaadi kihil olevad signaali-, toite- ja maanduse rajad horisontaalselt ning alumisel trükkplaadi kihil vertikaalselt või vastupidi (sele 3.2). Rajal

võib mõlemal kihil asetada ka diagonaalselt, ent põhiline on see, et kummalgil kihil on rajad omavahel 90-kraadise nurga all. Sellise radade asetusega projekteerimine aitab vältida radade omavahelist põimumist, mis takistab radade viimist komponentide juurde.



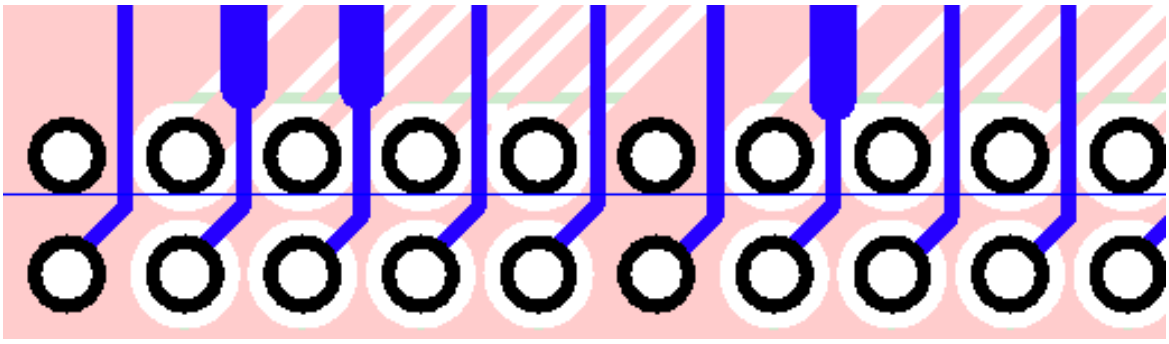
Sele 3.2 Trükkplaatide rajad on pealmisel kihil vertikaalsed ning alumisel kihil horisontaalsed

3.3.3 Radade laiused

Radade laiuste valimisel kindlaid reegleid pole. Pigem lähtutakse radade laiuste valimisel konkreetse projekteeritava trükkplaadi elektrilistest nõuetest, trükkplaadi suurusest ning nõutud radadevahelistest kaugustest. Siiski on projekteerimisel tuntud reegel, et laiemad rajad on paremad. Laiematel radadel on väiksem sisetakistus, väiksem induktiivsus, neid on lihtsam inspekteerida ning suure tõenäosusega on neil ka madalam tootmiskulu. [9, lk 7].

Kuna lõpptulemusena toodetakse trükkplaat tootmiskavandi järgi, tuleb kontrollida ka trükkplaadi tootmise ettevõtte täpsusnumbreid, sest ebatäpsemate tootmispinkide tolerantsid ei pruugi kattuda trükkplaadi projekteerija nõuetega. Eesti ettevõte Kamitra OÜ, kus toodeti käesoleva lõputöö trükkplaadid, nõuab raja ja raja vaheliseks distantsiks vähemalt 0,2 mm ning soovitab 0,25 mm üldise raja vähimaks laiuseks [11]. Tegelikkuses on trükkplaatide tootmisel siiski tolerantsiga seotud probleemide oht peenikestel ning väikese vahekaugusega radadel. Seetõttu on mõistlik valida minimaalsetest mõõtmetest suuremad ning võimalusel tuleb koostada rajad võimalikult laiad, kui trükkplaadi suurus ja tehnilised tingimused seda võimaldavad.

Mitmed käesoleva lõputöö raames rakendatavad trükkplaadid on projekteeritud ühenduma Arduino Mega 2560 platvormi piikühendustega. Arduino Mega 2560 platvormil on 40 piigiga ühenduse juures teatud kitsendused signaaliradade läbiviimisega teiste piikide vahelt, sest trükkplaadi projekteerimisel on nõutud teatud tühja vahemaad (isolatsiooni) radade ja näiteks läbiviigu aukude vahel. Selle lahendamiseks kasutatakse radade kitsendamist nendes kohtades (sele 3.3). Lühikese kitsendusega raja elektrilised parameetrid selle tõttu oluliselt ei muutu, temperatuur hajub rajas paremini ning rajal on endiselt madal näivtaksitus koos võimalusega minna kitsastest kohtadest läbi. [9, lk 7]

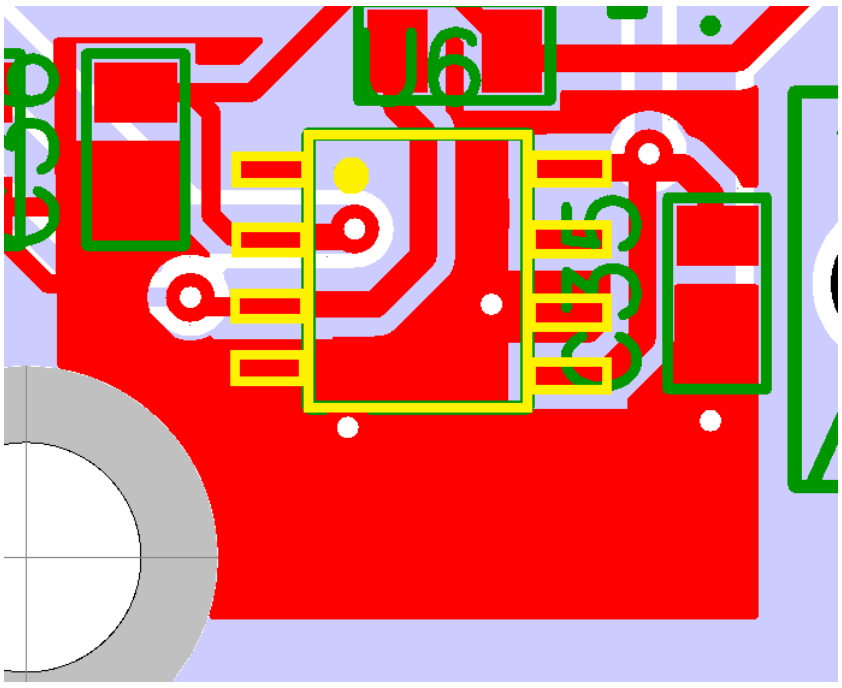


Sele 3.3 Toiteradade kitsendamine 2,54 mm sammuga piikühenduse juures

Juhendaja soovitusel on trükkplaatide projekteerimisel rakendatud rusikareeglit, mis ütleb, et 1 A voolutugevuse läbi juhtimiseks peab olema raja laius 1 mm. Seetõttu on suurem osa radasid projekteeritud laiusega 0,35 mm või 0,5 mm.

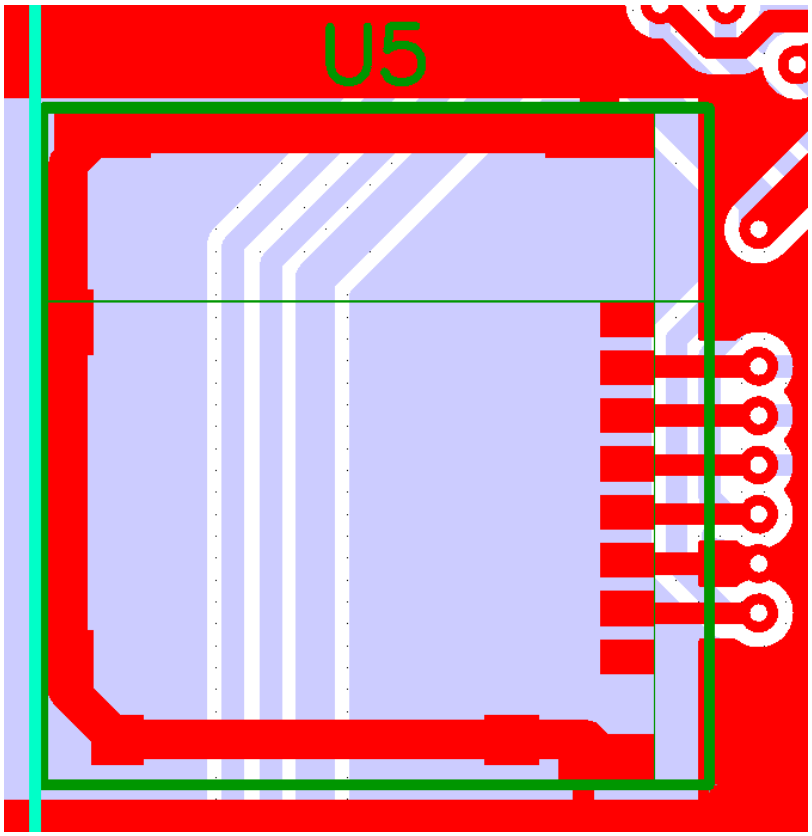
3.3.4 Vasealad ja läbiviigud

Vasealasid kasutatakse pindmontaažkomponentide ühendamiseks signaaliradadega ning suure osa trükkplaadil tekkivast soojusest eraldub ka nende samade vasealade kaudu. Seetõttu on projekteerimisel kasutatud näiteks toiteregulaatorite või suuremate lülituste juures suurendatud vasealasid, mis aitavad hoida komponente jahedatena (sele 3.4). Käesoleva lõputööga seotud trükkplaadid on kõik kahekihilised ning alumine trükkplaadi kiht alati kaetud vasealaga, mis on maandatud. Suurendatud vasealad ühendatakse reeglina läbiviikude abil alumise kihi maandusega kokku, mis aitab vältida maanduse potentsiaali nihkumist erinevates trükkplaadi osades. Suuri vasealasid kasutades tuleks eemaldada ebavajalikud „saared“, mis võivad hakata antennina käituma ning häirima ülejäänud süsteemi tööd.



Sele 3.4 Suurendatud vaseala LT3021-ADJ pingeregulaatori jahutamiseks

Mitmete komponentide vasealad on leitud Diptrace komponentide teegist. Nende komponentide hulka kuuluvad näiteks SOIC-8 mikrolülitused ning mitmed kaitsedioodid. 0805 tüüpi takistite ja kondensaatorite vasealad on siiski redigeeritud kitsamaks, et kahe vaseala vahelt oleks võimalik tõmmata 0,35 mm laiune rada. Samuti on vähem kasutatavate komponentide (näiteks GPS moodul, SD-kaardi lugeja/kirjutaja) vasealad projekteeritud trükkplaadi arendaja poolt (sele 3.5).



Sele 3.5 Molex 503182-1853 tüüpi Micro-SD kaardi lugeja/kirjutaja vasealade kavand

Läbiviikude suurusi trükkplaatide projekteerimisel muudetud ei ole ning Diptrace programmis on läbiviikude diameeter vaikimisi 0,7 mm. Kuna üldjuhul on trükkplaadi alumine külg kaetud maandatud vasekihiga, on mõne mikrolülituse või pealmise kihi vaseala juures lisatud mitmeid läbiviike, et trükkplaat oleks ühtlaselt maandatud, maandusradade takistus oleks võimalikult väike ning süsteem oleks töökindlam.

4 TOODETUD TRÜKKPLAADID

4.1 Trükkplaatide tootmisest üldiselt

Käesoleva lõputöö raames teostati tõrkeotsinguid kahele eelnevalt valminud trükkplaadile ning projekteeriti energia kogumise ja andmehõive süsteemi jaoks kuus trükkplaati, millest kahele oli vaja ainult sisse viia tõrkeotsingul ilmunud vigade parandused.

Kõikide trükkplaatide analüüsiks ning projekteerimiseks kasutati vabavaralist programmi Diptrace 3.0. Diptrace programm on võimeline koostama elektriskeeme ning tootmiskavandeid, ent kahjuks simulatsioonidega trükkplaatide tööd kontrollida võimalik ei ole.

Trükkplaatide koostamisel on kasutatud mõningaid Diptrace komponentide teegis olevaid komponentide Euroopa tingmärke (takistid, kondensaatorid, MOSFET-id jpm) ning keerulisemate ja vähem tuntumate komponentide (optoisolaator, LVDS draiver) kujutamisel elektriskeemis on projekteerija ise kujutanud tingmärkide abil komponendi tööpõhimõtet (sele 3.1).

Kuna kahte trükkplaati oli vaja vaid parandada ning nelja trükkplaati tuli algusest peale projekteerida, on seda silmas pidades jaotatud käesolev peatükk parandatud trükkplaatide analüüsimiseks ning uute projekteeritud trükkplaatide analüüsimiseks.

Trükkplaatide juhtimiseks on kirjutatud ka tarkvara, mis võimaldab andmeid lugeda erinevate trükkplaatide komponentide abil. Trükkplaatide omavaheliseks kommunikatsiooniks kasutatakse SPI protokolli ning trükkplaatide ühenduvust teiste trükkplaatidega näeb lisas 1.

4.2 Parandatud trükkplaadid

4.2.1 Informatsiooni kogumise trükkplaat

Informatsioonikogumise trükkplaadid asuvad päiksekollektori soojuse kogumise konstruktsiooni (soojuslatt, termoelektrilised elemendid, jahutustorud) taga, kus nende jaoks on karbid informatsioonikogumise trükkplaadi kinnitamiseks. Seega informatsiooni kogumise trükkplaadid on soojusvahetuse protsessile kõige lähedamal asuval plaadid.

Informatsiooni kogumise trükkplaat on võimeline lugema ning pinge ning voolutugevuse väärtust, mis tähendab, et just nende plaatide abil saadud informatsiooni on võimalik arvutada termoelektrilistest elementidest tulevat võimsust. Parema andmehõivesüsteemi arendamiseks ning täpsemaks andmete analüüsiks on lisatud trükkplaadile ka termopaaride ühendamise võimalused koos termopaari väärtuste teisendajatega (MAX31855 nimetusega mikrolülitus).

Lisavõimalusena on informatsiooni kogumise trükkplaadil ka veel digitaal-analoogmuundur, mis võimaldab kindlal väärtusel pinget väljastada ning kommunikatsiooni jaoks on implementeeritud ka LVDS süsteem, mis muudab plaadiga suhtlemise SPI protokolli alusel töökindlamaks.

Informatsiooni kogumise trükkplaadi täieliku elektriskeemi koos tootmiskavandiga leiab lisas 2.

4.2.2 Informatsiooni kogumise trükkplaadile tehtud parandused

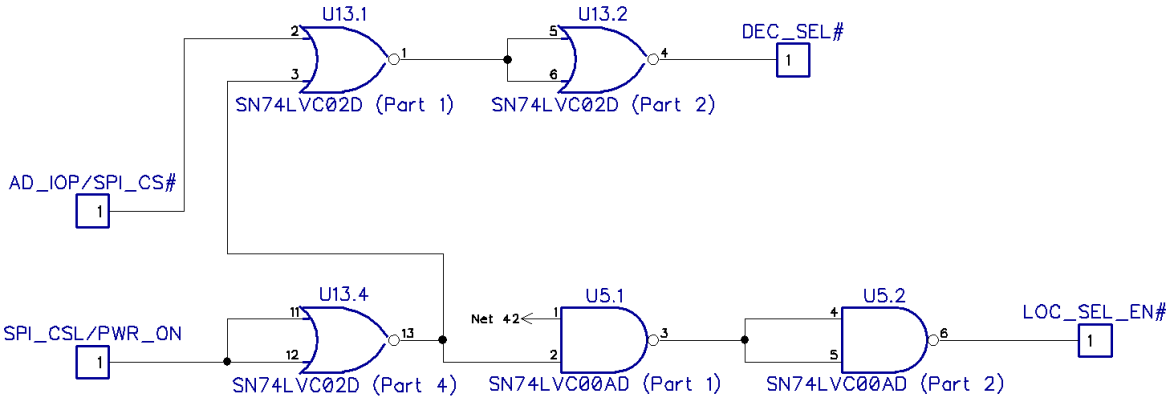
Informatsioonikogumise trükkplaadil on mitmeid komponente, millega Arduino Mega 2560 platvorm suhtleb ning selle jaoks on vaja ka õige alama valimiseks kasutada SPI protokolli SS signaali (peatükk 2.3.4). Kuna antud trükkplaadil on alamaseadmeid palju, on trükkplaadil U12 tähisega SPI liidesega Microchip MCP23S17 GPIO *(General purpose input/output)* laiendaja, mis võimaldab Arduinol Mega 2560-l suhtlemiseks valida konkreetse alama informatsioonikogumise trükkplaadil.

Eelnevalt projekteeritud trükkplaadi versioonil oli MCP23S17 mikrolülituse $\overline{\text{RESET}}$ jalg ühendatud maandusega kokku. Kuna $\overline{\text{RESET}}$ on madalal nivool aktiivne, tähendab see seda, et mikrolülitusel keelati SPI signaalidega suhtlemine ära ning see ei töötanud korrektselt. Parandatud versioonis on $\overline{\text{RESET}}$ jalg ühendatud 3,3 V ahelaga kokku.

Teine viga projekteerimisviga esines SN74LVC00AD tähistusega U5 ja SN74LVC02D tähistusega U13 lülituste töös. U5 koosneb neljast NING-EI lülitusest ning U13 koosneb neljast VÕI-EI lülitusest. Mitmete loogikalülituste tulemusel on trükkplaadil võimalik kahe sisendi ($\overline{\text{AD\_IOP/SPI\_CS}}$ ja $\overline{\text{SPI\_CSL/PWR\_ON}}$) abil luua väljundid ($\overline{\text{DEC\_SEL}}$ ja $\overline{\text{LOC\_SEL\_EN}}$), mis kontrollivad koostöös eelnevalt välja toodud MCP23S17 lülituse abil õige alama valimist SPI kommunikatsiooni jaoks. Eelnev projekteerija oli teinud loogikalülituste ühendamisel vea. Õiged lülituse sisendite ja väljundite nivood on kujutatud tabelis 4.1 ning loogikalülituste asetus on kujutatud selel 4.1.

Tabel 4.1 U5 ja U13 lülituste abil koostatud loogikalülituste sisendite ja väljundite tabel informatsioonikogumise trükkplaadil

Sisendid		**Väljundid**	
$\overline{\text{AD\_IOP/SPI\_CS}}$	$\overline{\text{SPI\_CSL/PWR\_ON}}$	$\overline{\text{DEC\_SEL}}$	$\overline{\text{LOC\_SEL\_EN}}$
0	0	1	1
0	1	0	0
1	0	1	0
1	1	1	0



Sele 4.1 U5 ja U13 komponentide abil koostatud loogikalülituste jada

Kolmas projekteerimisviga esines AD8417, tähisega U6 skeemiosas. AD8417 on šundiga voolumõõtmise juures kasutatav võimendi ning seda kasutatakse termoelektrilistest elementidest tuleva voolutugevuse mõõtmiseks. Voolutugevuse väärtuse suurenedes annab U6 väljundisse ka suurema pinge väärtuse, mida on võimalik analoog-digitaalmuunduri abiga lugeda ning Arduinole saata.

U6 mikrolülitus kasutab kahte referentspinget (VREF1, VREF2) ning esimene nendest peab olema ühendatud kõrgele nivoole (3,3 V) ning teine peab olema ühendatud madalale nivoole (0 V). Esialgsel versioonil olid mõlemad referentspinged ühendatud madalale nivoole.

Neljas projekteerimisviga esines MCP4912, tähisega U9 skeemiosas. MCP4912 on SPI liidesega digitaal-analoogmuundur, millel tuli $\overline{\text{SHDN}}$ jalg ühendada $\overline{\text{CS\_LDAC}}$ ahelaga ning $\overline{\text{LDAC}}$ jalg ühendada maandusesse.

Viies projekteerimisviga esines MAX31855, tähisega U1...U4 skeemiosades. MAX31855 on SPI liidesega mikrolülitus, mis on võimeline töötlema termopaarilt saadud pingeväärtusi ning SPI liinide kaudu saatma informatsiooni Arduino Mega 2560-le. Eelnevalt koostatud skeemis oli toitepinge jalg ühendatud maandusesse ning maanduse jalg toitepingesse. Parandustega plaadil vahetati jalgadele ühenduvad ahelad korrektseteks.

4.2.3 Analoog-digitaalmuunduri trükkplaat

Analoog-digitaalmuunduri plaadil asub Texas Instruments'i ADS1248 SPI liidesega mikrolülitus, tähisega U1, mis on 24-bitine, 2 kSPS kiirusega keerukas ent samas võimekas ja täpne analoog-digitaalmuundur. SPI protokolli kasutades on võimalik Arduino Mega 2560-l konfigureerida selle ADC seadeid ning lugeda selle abil mitmete sisendite pingeid.

Sarnaselt informatsioonikogumise trükkplaadile, on ka ADC plaadil mitmeid komponente peal, mis näiteks võimaldavad plaadiga SPI LVDS liidestust koos SS liini kordistusega. Lisaks on ADC trükkplaadil ka eraldi toiteregulaatorid 3,3 V analoogkomponentide jaoks ning 3,3 V digitaalkomponentide jaoks, termopaaride sisendid koos termopaari pingeväärtuste teisendajatega.

ADC trükkplaadi täieliku elektriskeemi koos tootmiskavandiga leiab lisas 3.

4.2.4 Analoog-digitaalmuunduri trükkplaadile tehtud parandused

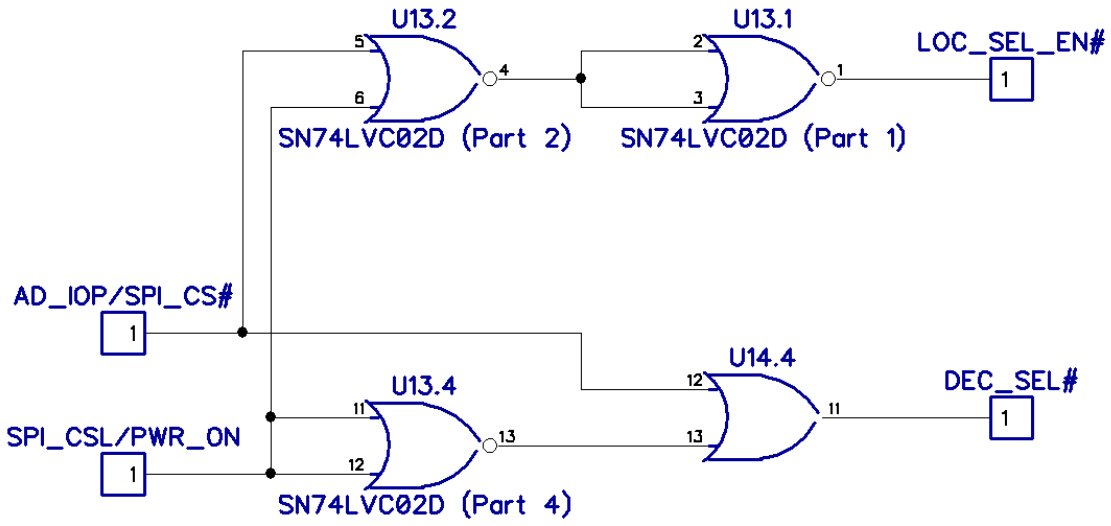
Sarnaselt informatsioonikogumise trükkplaadile, kasutatakse SS signaali kordistamiseks MCP23S17 GPIO laiendajat, tähisega U8. ADC trükkplaadil on tehtud sarnaseid vigasid nagu informatsiooni kogumise trükkplaadil ning ka ADC trükkplaadil asuvad GPIO laiendaja $\overline{\text{RESET}}$ jalg on ühendatud esialguses trükkplaadi versioonis maandusesse. Parandatud trükkplaadi skeemil on $\overline{\text{RESET}}$ jalg ühendatud mikrolülitus toitesse.

Teise projekteerimisveana on ka ADC plaadil realiseeritud loogikalülituste jada SN74LVC02D VÕI-EI lülituse, tähisega U13 ja SN74LVC32D VÕI lülituse, tähisega U14 abil. Tegelike sisendite ja väljundite suhet iseloomustab tabel 4.2.

Tabel 4.2 U13 ja U14 lülituste abil koostatud loogikalülituste sisendite ja väljundite tabel ADC trükkplaadil.

Sisendid		**Väljundid**	
$\overline{\text{AD\_IOP/SPI\_CS}}$	$\overline{\text{SPI\_CSL/PWR\_ON}}$	$\overline{\text{DEC\_SEL}}$	$\overline{\text{LOC\_SEL\_EN}}$
0	0	1	0
0	1	0	1
1	0	1	1
1	1	1	1

Vastava tabeli sisendite ja väljundite väärtused on koostatud selel 4.2 kujutatud loogikalülituste jada abil.



Sele 4.2 U13 ja U14 komponentide abil koostatud kombinatsioonloogika

Kolmandat muudatust ADC trükkplaadil ei oleks õiglane nimetada projekteerimisveaks. ADS1248 plaadi programmeerimisel tekkisid tõrked ning Arduino poolt saadetud ja vastuvõetud informatsioon ei olnud korrapärane ja töökindel. Tõrkeotsingu tulemustest lähtudes lisati ADC SCLK liini ja maanduse vahele 47 kΩ takisti ning 33 pF kondensaator paralleelselt, mis jäljendavad ostsilloskoobi sondi elektriparameetreid.

Neljanda projekteerimisveana on BAT54S Shottky dioodid, tähistega D1...D4 asetatud trükkplaadi skeemile valepidiselt. Uuendatud trükkplaadi skeemil on dioodide jalgade külge ühendatud õiged elektriahelad.

Viienda projekteerimisveana on sarnaselt informatsioonikogumise trükkplaadile ka ADC trükkplaadil oleva MAX31855 komponendi toiteahel ning maanduse ahel omavahel valesti.

Kuuenda muutusena lisati ADC trükkplaadile 3,3 V → 1,2 V LT3021ADJ toiteregulaator, tähisega U6. ADS1248 tehniliselt andmelehelt selgub, et tugipingeks ei sobi antud ADC-le 0 V ning seetõttu kavandati plaadile lisaks 1.2 V toitepinge, mida kasutatakse uue referentspingena ADC-l.

4.3 Projekteeritud trükkplaadid

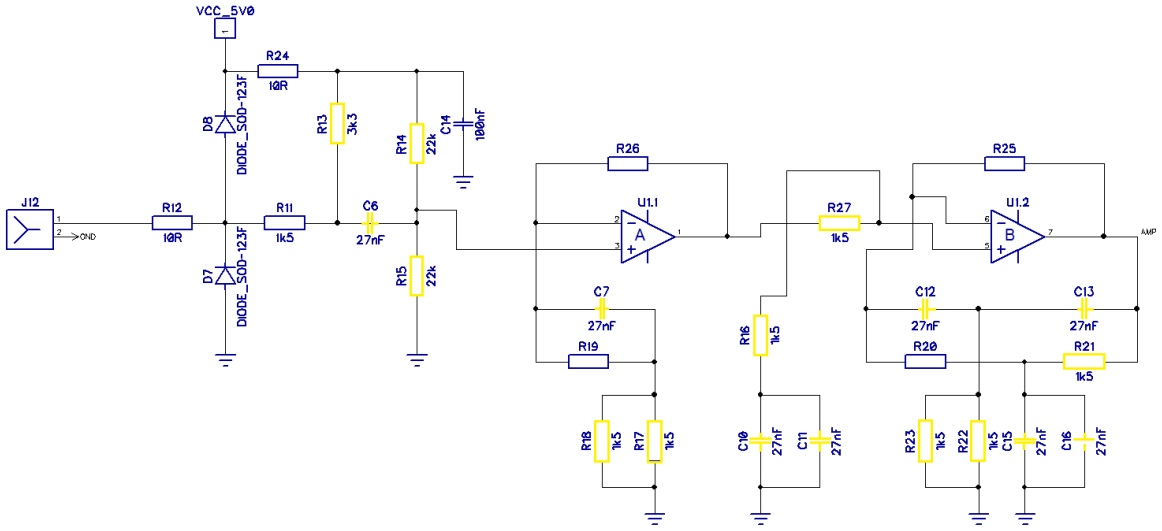
4.3.1 Võimendi ja pingejagurite trükkplaat

Võimendi ja pingejaguri trükkplaadil on kaks eesmärki: võimendada Fraunhoferi kalibreerimistunnistusega ISET päikesekiirgussensori väljundpinget ning jagada pingejagurite abil CPC1832 päiksepaneelide väljundpinget. Päikesekiirgussensor aitab leida päikse poolt tulenevat elektromagnetilist kiirgusvõimsustihedust väga täpselt ning CPC1832 sensoreid kasutatakse päikse asukoha tuvastamiseks „vihmavarju“ konstruktsiooni kaudu, mida kirjelduse leiab peatükist 1.3.

Võimendi ja pingejagurite trükkplaadi täielik elektriskeem ja tootmiskavand on esitatud lisas 4.

Võimendiga võimendatakse ISET päikesekiirgussensorist tulevat pingesignaali. ISET päikesekiirgussensori tehniliselt andmelehelt on võimalik lugeda, et ligikaudu 1000 W/m^2 päikselt lähtuva tüüpilise spektriga elektromagnetilise kiirgusvõimsustiheduse korral on pingeväljund 30 mV. Fraunhoferi kalibreerimistunnistus kinnitab täpsemalt, et 1001,798 W/m^2 elektromagnetilise radiatsiooni korral on väljundpinge väärtus 27,533 mV ±3% mõõtemääramatusega. ADC jaoks on sellised pingeväärtused liiga madalad, et piisava täpsusega mõõtetulemus teisendada ning seejärel edastada.

Selle probleemi lahendamiseks kasutataksegi suure täpsusega võimendit OP262GS, tähisega U1, mis on operatsioonivõimendite baasil ehitatud võimendi (sele 4.3). OP262GS tehniline andmeleht kinnitab, et maksimaalne viga pingete võimendamisel võib olla 0,325 µV, mis on sobilik meile vajaliku mõõtmistäpsusega. OP262GS võimaldab kaheastmelist võimendust, st esimese operatsioonivõimendi abil teostatakse esimene võimendus, mis seejärel suundub teise operatsioonivõimenduse juurde, kus teostatakse teine võimendus. Esimese astme ning teise astme võimenduse väärtused korrutades saab teada terve võimendusahela võimenduse.



Sele 4.3 OP262GS põhjal koostatud võimendi skeemiosa, kus kollaselt märgitud komponente ei monteerita

Võimendustegur mõlemale võimendusahelale valitakse välistakistite abil pingejagurite näol. Esialgu on võimendustegur esimeses astmes 6 ning teises astmes 10, ehk terve võimendusahel peaks võimendama esialgset signaali 60 kordselt. 60 kordse võimendusteguri korral on näiteks 2000 W/m^2 elektromagnetilise kiirgusvõimsustiheduse korral väljundis 3,6 V, mis on analoog-digitaalmuunduri trükkplaadil asuva ADS1248 analoog-digitaalmuunduri jaoks sobivam pingeväärtus, mida täpselt töödelda.

Pingejagureid kasutatakse SOIC korpuses olevate CPC1832 päiksepaneelide pingeväärtuste piiramiseks. Tugeva päikesekiirguse korral võivad CPC1832 mikrolülitused väljundisse anda 8 V pinge väärtuse, mis on taaskord ADC teisenduste jaoks liiga kõrge. Kahe määratud takisti abil koostatakse pingejagur, mis jagab sissetuleva pingeväärtuse kolmega.

CPC1832 päikese suuna tuvastamise andurite trükkplaadid, mis asuvad „vihmavarju“ konstruktsioonil, leiab lisas 5.

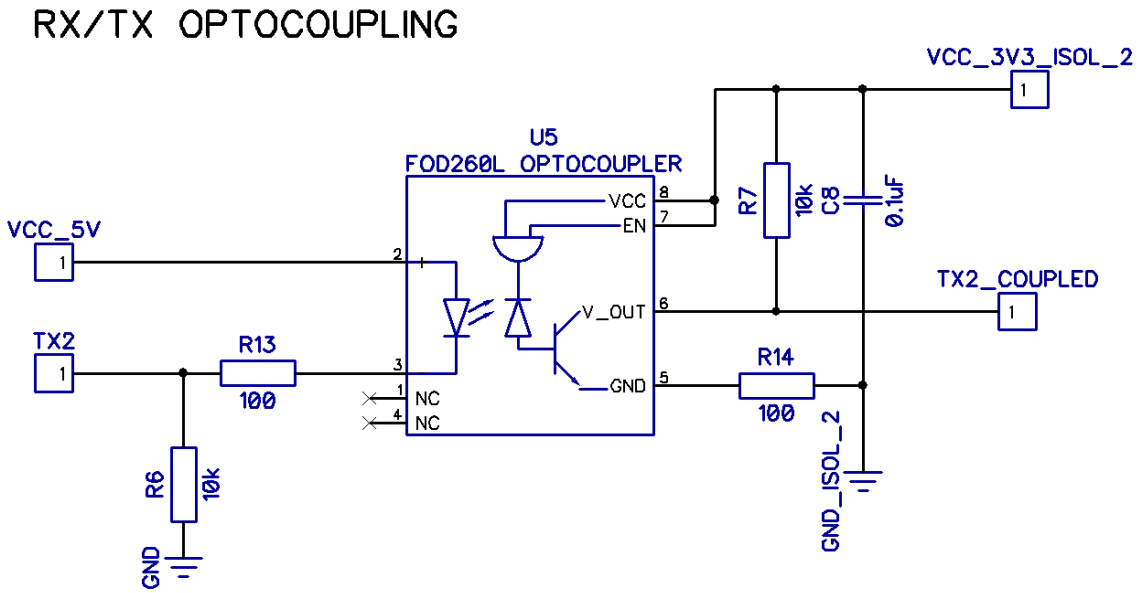
4.3.2 Optoisolatsiooni trükkplaat

Optoisolaator on komponent, mis suudab ühelt juhilt tulevat signaali edasi kanda teisele juhile, kasutades optilist eraldamist. See tähendab, et juhid ei ole omavahel elektriliselt ühendatud ning signaali edasi kandmiseks kasutatakse hoopis valgust. Optoisolatsiooni kasutatakse käesoleva lõputöös erinevate elektrisüsteemide eraldamiseks. Juhul kui ühes andmehõive või energia kogumise süsteemiosas peaks tekkima tugev pingehüpe, on optiliselt isoleeritud süsteemi osas oleva komponendid ja mikrokontrollerid kaitstud.

Optilise isolatsiooni trükkplaadi täieliku elektriskeemi ja tootmiskavandit näeb lisas 6.

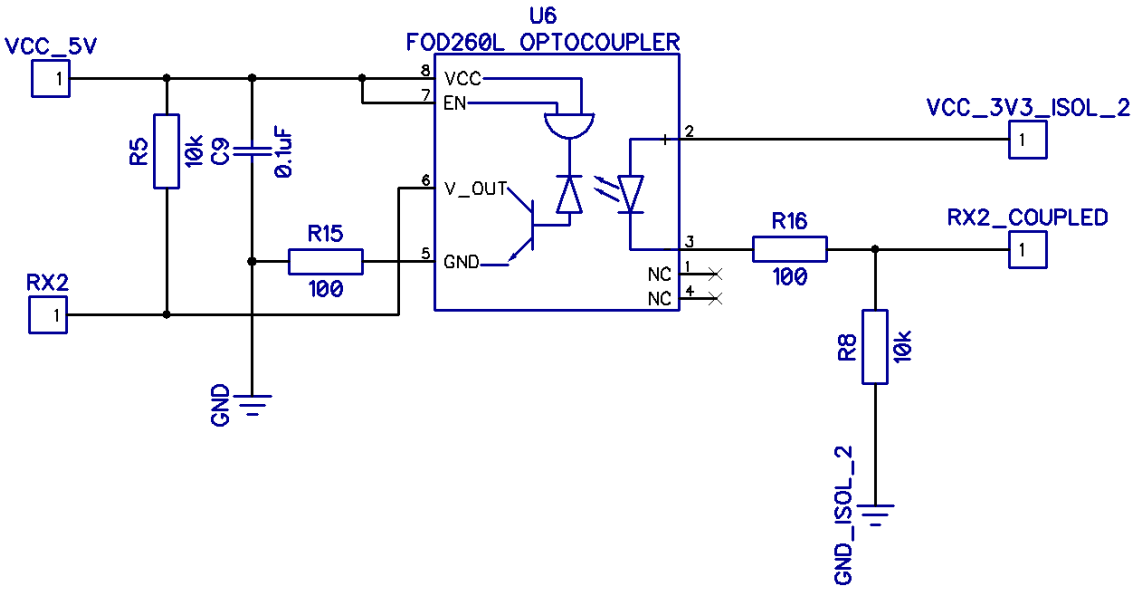
Optoisolatsiooni võimaldavad FOD260L komponendid, tähistega U1, U2, U5, U6, U8, U9. Optiliselt isoleeritud elektrisignaali saatmiseks on FOD260L komponendil sisenditeks 5 V pingel olev toide ning Arduino plaadilt tulev TX liin, mille abil on võimalik andmesignaal saata.

Optoisolatsiooni elektriskeemilt (sele 4.4) on näha, et juhul kui TX liin on madalal nivool, on 5 V toite ja TX liini vahel olev potentsiaalide vahe suur ning FOD260L mikrolülituse sees olev LED töötab. Kui TX liin on kõrgel nivool, siis on 5 V toite ja TX liini vahel madal potentsiaalide vahe, mis tähendab et LED ei tööta. Seega LED süttib ja kustub TX liinil olevate nivoode vaheldumisel. LED-i põlemise ja kustumise tuvastab ära teisel pool isolatsiooniahelat olev andur, mis kannab signaali isoleeritud juhis edasi.



Sele 4.4 FOD260L põhjal koostatud TX optoisolatsiooniahel, kus vasakul pool on Arduino Mega 2560 platvormile ühenduv pool ja paremal on andmevahetuskaabli pool

RX liinil toimub andmete vastuvõtmine, ent FOD260L komponent on vastassuunaliselt ühendatud. RX signaalide lugemise protsess on siiski TX kirjutamise protsessile väga sarnane (sele 4.5).



Sele 4.5 FOD260L põhjal koostatud RX optoisolatsiooniahel, kus vasakul pool on Arduino Mega 2560 platvormile ühenduv pool ja paremal on andmevahetuskaabli pool

Isoleeritud toiteahel kaitseb elektrisüsteeme pingenihete ja –hüpete eest ning reguleerib pingeid FOD260L komponentide jaoks. Selle jaoks kasutatakse optoisolatsiooni trükkplaadil IE0503S isoleeritud

toiteregulaatoreid, tähistega U3, U11, U12. Antud toiteregulaator reguleerib 5 V sisendpinge isoleeritud 3,3 V väljundpingele. See tähendab, et sisendi toiteahela ja maanduse ahela vahel on alati 5 V ning väljundis on toiteahela ja isoleeritud maanduse vahel alati 3,3 V. Antud olukorras võivad mängu tulla suured pingete nihkumised, mis IE0503S komponendile mõju ei avalda, sest sisendi ja väljundi maandused ei ole omavahel ühendatud. Näiteks võib sisendis olla toiteahel 5 V peal maanduse suhtes ning väljundis võib olla toiteahel 25,3 V peal tavalise maanduse suhtes. Isoleeritud maanduse ja tavalise maanduse potentsiaalide vahe on seega 22 V. Optoisolatsiooni trükkplaadil on 3 sellist toiteregulaatorit ja igaüks neist toidab ühte FOD260L RX ning TX ahela paari.

Diferentsiaalsignaalide muundajatena kasutatakse antud trükkplaadil SN65LVDM179D komponente, tähistega U4, U7, U10. Sarnaselt SN65LVDS050 komponendile, kasutab see komponent signaalide diferentseerimisel RS-644 standardit (vt peatükk 2.5) aga omab ainult kahte diferentseerimisahelat, sest diferentseerida on vaja ainult RX ja TX signaale. Ka neid komponente toidetakse isoleeritud toiteahela kaudu.

4.3.3 GPS mooduli ja SD-kaardi mooduli trükkplaat

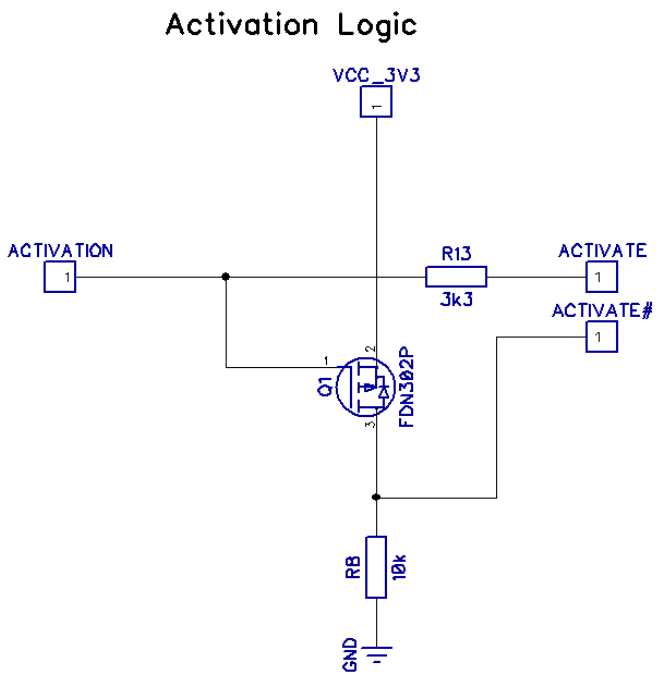
Andmehõive ja energia kogumise süsteemis on implementeeritud ka GPS moodul ja SD-kaardi lugemise/kirjutamise moodul.

GPS moodulit kasutatakse kellaaja saamiseks ning see on sellepärast vajalik, et juhul kui päikese tuvastamise ja järgmise süsteem päikese asukohta tuvastada ei suuda või üldse kokku jookseb, on kellaaja järgi võimalik päikesekollektor keerata turvalisse asendisse, kus päikesevalgust kumerpeeglile ei lange. GPS moodulina kasutatakse GY-NEO6MV2 komponenti, tähisega U2. See GPS moodul kasutab Arduino Mega 2560-ga suhtlemiseks RX ja TX liine.

SD-kaardi lugejana kasutatakse SPI liidesega Molex 503182 komponenti, tähisega U5. SD-kaardile lugemine ja kirjutamine annab võimaluse näiteks informatsiooni kogumise trükkplaadilt ja analoog-digitaalmuunduri trükkplaadilt saadud informatsiooni logimiseks.

Antud trükkplaadile on projekteeritud ka aktiveerimis-kombinatoorloogika (sele 4.6), mis juhib kahte ümberlülituse mikrolülitust ning LVDS diferentsiaalsignaalide draiverit. Näiteks on SD-kaardi lugeja/kirjutaja mooduli ees ümberlülitus, mis lülitab SPI protokolli signaale. Kui selle ümberlülituse jaoks pole aktiveerimisloogika õigel nivool, SD-kaardile lugemist ega kirjutamist toimuda ei saa. Kuna antud trükkplaadil pole väga palju alamaid, millega oleks ülemal vaja suhelda, pole vaja ka nii keerukat alamate adresseerimise viisi kasutada nagu oli näiteks informatsioonikogumise trükkplaadil, kus kasutati MCP23S17 GPIO laiendajat.

GPS ja SD-kaardi moodulitega trükkplaadi elektriskeem ja tootmiskavand on kujutatud lisas 7.



Sele 4.6 GPS-mooduli ja SD-kaardi mooduli trükkplaadil olev aktiveerimise loogika

5 TARKVARA

5.1 Tarkvarast üldiselt

Arduino Mega 2560 on Atmega2560 mikroprotsessori põhjal ehitatud mikrokontrolleri baasplatvorm. Mega 2560-l on 54 digitaalset sisendit/väljundit (15 väljundit sobivad pulsilaiusmodulatsiooni jaoks), 16 analoog sisendit, 4 UART *(Universal asynchronous receiver/transmitter)* porti, 16 MHz kellatakti sagedus, USB pistik, toitepistik, ICSP *(In-circuit serial programming)* pistik ning RESET-nupp. [12]

Käesolevas lõputöös kasutatakse mitmeid Arduino Mega 2560 abil juhitud süsteeme, mis on omavahel galvaaniliselt isoleeritud optoisolatsiooni trükkplaatide abil. Seega on ühes alamsüsteemis toimuva lühise korral teised elektrisüsteemi komponendid (sh ka Arduino) kaitstud ning turvalisuse säilitamise meetmed on endiselt töövõimelised.

Arduino Mega 2560 konfigureerimiseks ja programmeerimiseks kasutati Arduino IDE 1.8.2 tarkvara. Antud projekti arendamiseks kasutatakse just Arduino trükkplaate, sest eelnevalt koostatud trükkplaadid on projekteeritud sobituma Arduino Mega 2560 piikühendustega. Arduinode programmeerimiseks kasutatakse ka Arduino IDE tarkvara, sest Arduino valmisfunktsioonide (näiteks *SPI.transfer* jpm) teekide abil on tarkvaraarendajate jaoks elu lihtsaks tehtud. Arduino trükkplaadid ja tarkvara ei ole üldiselt professionaalsete tarkvaraarendajate seas levinud, ent lihtsamate ning vähem spetsiifiliste süsteemide programmeerimiseks sobib Arduino väga hästi.

Arduino IDE 1.8.2 programmis kasutatakse C ja C++ programmeerimiskeeli, ent tänu Arduino rohketele teekidele ja ettekirjutatud funktsioonidele, ei pea ise hakkama erinevate teekide sügavustes kompima hakkama ning neid funktsioone redigeerima hakkama.

Käesolevas lõputöö raames kirjutati Arduino IDE programmis tarkvara informatsioonikogumise trükkplaadile ja analoog-digitaalmuunduri trükkplaadile.

5.2 Informatsiooni kogumise trükkplaadi tarkvara

5.2.1 Informatsioonikogumise trükkplaadi tarkvarast üldiselt

Nagu peatükis 4.2.1 on mainitud, informatsiooni kogumise plaat suudab lugeda termoelektriliselt elemendilt tulevat pinget ja voolutugevust ning suudab ka teisendada MAX31855 lülituse abil termopaaride väärtusi digitaalinformatsiooniks.

Kuna terve programmikood on küllaltki mahukas, on välja toodud järgmised funktsioonid ja nende funktsioonide kirjeldused, mis moodustavad informatsiooni kogumise trükkplaadi tarkvarast põhiosa.

5.2.2 Multipleksor trükkplaadi valimise alamfunktsioon

Multipleksor trükkplaadi on terve energiakogumise ja andmehõive süsteemiosa, mis vastab õige trükkplaadi adresseerimise eest. Multipleksor trükkplaadid kinnituvad trükkplaatide jaoks koostatud elektroonikakesta sisse ning elektrilises süsteemis asub multipleksori trükkplaat Arduino Mega 2560 ja informatsiooni kogumise trükkplaadi vahel.

Kuna iga informatsiooni kogumise trükkplaadi jaoks on üks multipleksor plaat, tuleb enne informatsiooni vahetamist aktiveerida õige informatsiooni kogumise trükkplaadi küljes olev multipleksori trükkplaat. Õige informatsioonikogumise trükkplaadi adresseerimiseks kasutatakse nii riistvara kui ka tarkvara. Riistvaraosas kasutatakse adresseerimiseks näiteks 3→8 dekoodreid, invertereid, JA loogikaelemente ning tarkvaraosas kasutatakse väljundite nivoode muutmist õige trükkplaadi valimiseks.

Multipleksor plaadi abil aadressi valimiseks tuleb esialgu määrata M_EN liin kõrgele nivoole, mis lubab edasist informatsiooni vahetamist. Multipleksor trükkplaadi valimise alamfunktsiooni tuleb saata kolm täisarvulist muutujat, mis 3→8 dekoodri abil aktiveerivad õige multipleksori trükkplaadi. Plaadil oleva kollast värvi LED indikaatori põlemisel on trükkplaat parajasti aktiveeritud. Terviklik alama valimise algoritm on kujutatud lisas 8.

Lisaks on antud alamfunktsioonil ka kontrollid, mis keelavad suvaliste täisarvuliste muutujate sisestuse ning lisaks väljastab alamfunktsioon läbi Arduino *Serial Monitori* kasutajale valitud multipleksorplaadi numbri.

Multipleksor trükkplaadi valimise alamfunktsioon on kujutatud lisas 9.

5.2.3 MCP23S17 registrite konfigureerimise alamfunktsioon

Kuna informatsioonikogumise trükkplaadil olevaid alamaid on palju, on nende adresseerimiseks kasutatud SPI liidesega MCP23S17 komponenti, mis on GPIO laiendaja. MCP23S17 tehnilises andmelehes [13] on välja toodud hulgaliselt erinevaid registreid, mida kasutatakse selle mikrolülituse konfigureerimiseks ning programmikoodi lihtsustamise eesmärgil oli vaja koostada alamprogramm, mis neid registreid edukalt konfigureerida suudab.

MCP23S17 kasutab kahte erinevat GPIO porti, GPIO porti A ja GPIO porti B ning kumbki port on võimalik ühendada kaheksa alamaga. Kuigi tehnilise andmelehe järgi on MCP23S17 komponendil 20 erinevat registrit, mis on võimalik konfigureerida, on andmehõive süsteemi arendamiseks vaja konfigureerida nendest ainult kuute registrit, milleks on IODIRA, IODIRB, IOCONA, IOCONB, GPIOA, GPIOB. Ülejäänud registrite vaikimisi väärtused on meie jaoks juba sobivad ning neid konfigureerima ei pea.

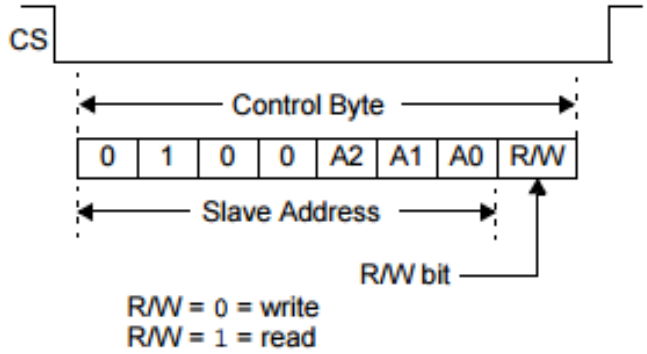
IODIRA ja IODIRB – registrid vastavalt portidele A ja B, mis valivad, millised GPIO liinide külge ühendatud komponendid on sisendid või väljundid. Kuna meile süsteemis kasutatakse MCP23S17 mikrolülitust alamate valimiseks, peavad kõik GPIO-d olema konfigureeritud väljunditeks.

IOCONA ja IOCONB – IOCON registrid võimaldavad seadistada näiteks registritesse kirjutamise järjekorda, porte konfigureerida, muuta liinide polaarsuse jpm. Täpsemat informatsiooni saab lugeda Microchip Technology MCP23S17 komponendi tehniliselt andmelehelt [13, lk 18]

GPIOA ja GPIOB – GPIO register võimaldab sättida GPIO väljundite väärtusi. Kuna antud lõputöö andmehõive süsteemis on GPIO väljunditele ühendatud alamate valimiseks SS signaalid, on vaja tarkvaras meil ainult määrata üks GPIO signaalidest madalale nivoole ning teised signaalid jätta kõrgele nivoole. Nõnda on võimalik adresseerida ühte konkreetset alamat ning sellega informatsiooni vahetada.

Registrite muutmise jaoks tuleb alamfunktsiooni kaasa saata registri nimi ning uus väärtus, mida soovitakse registrisse kirjutada. Igale saadetud registri nimele on alamfunktsioonis vastab registri aadress, mille abil programm teab, millisele registrile väärtust seada. Kuna MCP23S17 on SPI liidese mikrolülitus, kasutatakse MCP23S17 ja Arduino vaheliseks suhtluseks SPI funktsioone (vt peatükk 2).

Registri väärtuse määramiseks saadab Arduino 24-bitise infojada, kus esimesed 8 bitti määravad ära, millise MCP23S17 mikrolülitusega parajasti suheldakse. Kuna igal informatsioonilugemise trükkplaadil on vaid üks MCP23S17 lülitus peal, tuleb selle jaoks elektriskeemist järgi vaadata, millise ahela osaga antud komponendi A0...A2 jalad ühendatud on. Antud elektriskeemis on antud jalad ühendatud toitesse, seega tuleb informatsioonilugemise trükkplaadil oleva MCP23S17 ühenduse loomiseks saada bitid „01001110“, kus neli esimest bitti peavad alati samad olema, kolm järgmist bitti on A2...A0 nivoode väärtused ning viimane bitt näitab, kas MCP23S17 mikrolülitusele kirjutatakse või soovitakse sealt informatsiooni lugeda (sele 5.1).



Sele 5.1 MCP23S17 poole pöördumiseks vajalik bitijada

Järgmised saadetavad 8 bitti määravad ära, millise registri väärtus üle kirjutatakse. Registri nime abil on alamfunktsioon võimeline saatma õige registri aadressi. Viimase 8 bitina saadetakse uus registri väärtus, mida kasutaja on soovinud sättida.

Taaskord on alamfunktsioonis kasutatud kontrolle, et ei juhtuks mingisuguseid tahtmatuid olukordi. Kontrollitakse, kas registri nimi on üldse nimistus olemas ning kas soovitav väärtus oleks üldse 8 biti sisse mahtuv suurus.

Registrite konfigureerimise alamfunktsioon on kujutatud lisas 10.

5.2.4 MAX31855 temperatuuri lugemise alamfunktsioonid

Informatsioonikogumise trükkplaatidel on neli MAX31855 komponenti, tähistega U1...U4. MAX31855 on SPI liidesega mikrolülitus, mis on võimeline töötlema termopaarilt saadud pingeväärtusi ning SPI liinide kaudu saatma informatsiooni näiteks Arduino Mega 2560-le. Temperatuuri lugemiseks kasutatakse kahte alamfunktsiooni, *readMax31855Temp* ning *readMax31855TempConversion*. Esimene alamfunktsioon vastutab õige MAX31855 mikrolülituse adresseerimise eest ning teine alamfunktsioon sisaldab kõiki temperatuuri lugemise ja andmetöötluse jaoks vajalike koodiridasid. Temperatuuri lugemise alamfunktsioonis loetakse MAX31855 mikrolülitusel oleva temperatuurianduri väärtust, sest tarkvara arendamise etapi ajal polnud veel termopaarid MAX31855 mikrolülituse liinidele ühendatud. Temperatuuri lugemise põhimõte on mõlemal juhusel väga sarnane.

Esimesele alamfunktsioonile saadetakse täisarvulise muutujana kaasa number 0..3, mis annab informatsiooni selle kohta, millise MAX31855 mikrolülituse käest temperatuuri lugeda soovitakse. Adresseerimise alamfunktsiooni sees kasutatakse ka omakorda MCP23S17 registrite konfigureerimise alamfunktsiooni (vt peatükk 5.2.2), mis valib GPIO liinide küljes oleva õige MAX31855 komponendi SS signaali madalale nivoole. Antud alamfunktsioonis kasutatakse taaskord kontrolle, mis väldivad ebamääraseid olekuid alamfunktsioonides.

Teise alamfunktsiooni jaoks luuakse neli täisarvulist muutujat *data1*...*data4*, kuhu salvestatakse SPI informatsioonivahetuse käigus loetud informatsioon. Andmed loetakse täisarvulistesse muutujatesse sisse korraga ning andmejadas on informatsiooni 32 bitti. 32-bitisesse informatsioonijadast on võimalik lugeda näiteks termopaari temperatuuri, mikrolülitusel oleva temperatuurianduri temperatuuriväärtust ning antud informatsioonijadas on ka välja toodud veatuvastamise bitid.

Antud programmikoodis kasutatav Arduino funktsioon *SPI.transfer()* on võimeline lugema SPI liinil olevat informatsiooni ainult 8 biti kaupa. Seega lõpptulemuse saamiseks tuleb koondada neljas muutujas olevad väärtused üheks suureks väärtuseks. Antud informatsioonijada salvestamiseks kasutatakse *long* tüüpi muutujat, sest *int* muutujasse nii suured väärtused ei mahu. 32-bitise informatsioonijada koostamiseks kasutatakse bittide korrutamist 256-ga, mis nihutab andmed täpselt 8 biti võrra edasi.

Kuna antud lõputöös loetakse MAX31855 mikrolülitusel oleva anduri temperatuuriväärtust, tuleb 32-bitisest informatsioonijadast tehnilise andmelehe järgi filtreerida välja bitid numbriga 15...4 ning filtreeritud informatsioon tuleb temperatuuri näidu saamiseks jagada 16-ga.

Temperatuuri lugemise alamfunktsioonid on kujutatud lisas 11.

5.2.5 MCP4912 digitaal-analoogmuunduri alamfunktsioonid

Kui temperatuuri lugemiseks kasutatakse kahte alamfunktsiooni, siis DAC-i kasutamiseks kasutatakse samuti kahte alamfunktsiooni. Esimene alamfunktsioon *setOutputVoltage* määrab taaskord DAC liinil oleva SS signaali madalale nivoole MCP23S17 kaudu ning määrab ära, millisele DAC kanalile pinge väljastatakse.

Teine alamfunktsioon *applyVoltageWriting* seob vajaliku informatsiooni *SPI.tranfer()* funktsiooni jaoks kokku ning edastab selle DAC-ile. Pinge väärtuse sättimiseks tuleb alamfunktsioonide käivitamiseks anda kaasa ka kaks täisarvulist muutujat, mis nimetavad väljastamise pordi numbri ning väljundpinge väärtuse millivoltides.

Esimene alamfunktsioon kasutab taaskord registrite konfigureerimise alamfunktsiooni ning sellele lisaks valitakse seal funktsioonis pinge väljastamise port. MCP4912 DAC-il on pinge väljastamiseks port A ja port B, seega on võimalik korraga väljastada antud DAC-i kasutades kaks erinevat pinge väärtust. Antud alamfunktsioonis kasutatakse kontrolle, et vältida ebamääraste olekute tekkimist alamfunktsioonides.

Teine alamfunktsioon seob informatsiooni SPI informatsioonivahetuse jaoks kokku. Tehnilise andmelehe [14, lk 24] järgi tuleb DAC-ile saata 4 konfigureerimise bitti, 10 väljundpinge bitti ning 2 ebaolulist bitti. Antud alamfunktsioon seob antud informatsiooni 16-bitiseks informatsioonijadaks ning seejärel jagab selle kaheks 8-bitiseks informatsioonijadaks, mis on vajalik *SPI.transfer()* funktsiooni kasutamiseks.

DAC kasutamise alamfunktsioonid on kujutatud lisas 12.

5.2.6 LTC1861 analoog-digitaalmuunduri alamfunktsioonid

Sarnaselt temperatuuri lugemise ja DAC kasutamise funktsioonidega, on ka ADC kasutamiseks kaks alamfunktsiooni. Esimene alamfunktsioon *readInputVoltage* määrab ADC liinil oleva SS signaali MCP23S17 kaudu madalale nivoole ning määrab ära, millisele ADC kananlile pinge väljastatakse. Teine alamfunktsioon *applyVoltageReading* loeb vajaliku informatsiooni *SPI.tranfer()* funktsiooni kaudu ning seob informatsiooni üheks informatsioonijadaks.

Esimene alamfunktsioon kasutab registrite konfigureerimise alamfunktsiooni ning sellele lisaks valitakse seal funktsioonis pinge lugemise port. Sarnaselt MCP4912 komponendile, on LTC1861 ADC-l pinge lugemiseks port A ja port B, seega on võimalik lugeda mõlemad pordil olevat pinge väärtust. Antud alamfunktsioonis kasutatakse kontrolle, et vältida ebamääraste olekute tekkimist alamfunktsioonides.

Teine alamfunktsioon kasutab alguses SS liini nivoode muutmist, et äratada ADC ooterežiimist üles. Seejärel saadetakse 16-bitine informatsioonijada ADCle, mis sisaldab pordi numbrit. Pärast sellise käsu saamist hakkab ADC sisendite liinidel oleva pinge teisendamist, mistõttu enne informatsiooni lugemist kasutatakse *delay()* funktsiooni. Järgmise *SPI.transfer()* funktsiooniga loetakse kahe baidiga ADC pinge väärtus, mille hulgast tuleb taaskord välja eraldada 12-bitine info pinge suuruste kohta ning see jagada 2-ga.

ADC kasutamise alamfunktsioonid on kujutatud lisas 13.

5.3 Analoog-digitaalmuunduri trükkplaadi tarkvara

5.3.1 Analoog-digitaalmuunduri trükkplaadi tarkvarast üldiselt

Peatükis 4.2.3 oli mainitud, et analoog-digitaalmuunduri plaadil asub Texas Instruments'i ADS1248 SPI liidesega mikrolülitus, tähisega U1, mis on 24-bitine, 2 kSPS kiirusega keerukas, ent samas võimekas analoog-digitaalmuundur. ADC trükkplaadile arendatud tarkvara sisaldab mõningaid informatsioonikogumise trükkplaadile kirjutatud tarkvara alamfunktsioone, mille kirjeldused leiab peatükist 5.2. Sarnaselt informatsioonikogumise trükkplaadi tarkvarale, kasutab ka ADC trükkplaadi tarkvara multipleksor trükkplaadi valimise alamfunktsioon, MCP23S17 registrite konfigureerimise alamfunktsiooni ja MAX31855 temperatuuri lugemise alamfunktsiooni.

Eelnevalt nimetatud funktsioonid ei ole siiski informatsioonikogumise trükkplaadile kirjutatud tarkvaraga täielikult samane. Seoses signaaliliinide erineva füüsilise asetusega trükkplaatidel, on ADC trükkplaadile kirjutatud tarkvara kohendatud, ent alamfunktsioonide funktsionaalsus ja tööpõhimõtted on samad. Antud kohendatud funktsioone on võimalik leida lisas 14.

Alamfunktsioonid, mida informatsioonikogumise trükkplaadi tarkvaras kasutatud polnud, on näiteks ADS1248 registrite konfigureerimise alamfunktsioonid, ADS1248 registrite väärtuste lugemise alamfunktsioonid, ADS1248 mikrolülituse temperatuuri lugemise alamfunktsioon, ADS1248 sisendpinge väärtuse lugemise alamfunktsioon.

ADS1248 tehnilise andmelehe SPI funktsioonide tabelis [15, lk 45] on välja toodud ADS1248 seadistamise ja kasutamise jaoks vajalikud funktsioonid. Antud tabeli funktsioonidest kasutatakse näiteks *WAKEUP, SLEEP, RESET, RDATA, RDATAC, RREG* ja *WREG* ka antud tarkvara rakendamisel ning suur osa arendatud alamfunktsioonidest kasutavad antud funktsioone.

5.3.2 ADS1248 registrite konfigureerimise alamfunktsioonid

Sarnaselt MCP23S17 registrite konfigureerimisega, on vaja ka ADS1248 ADC registreid enne mõõtmisoperatsioonide teostamist seadistada. ADS1248 tehnilisel andmelehe tabelis 9.6.3 [15, lk 55] on kujutatud registrid, mida on võimalik kasutada ADC konfigureerimiseks. Käesoleva tarkvara arendamiseks on konfigureeritud *MUX0, MUX1, SYS0* ja *IDAC0* registreid. Ülejäänud registrid on tarkvara jaoks vaikimisi olekus. Registrite seadistamiseks kasutatakse *writeADS1248RegisterForSolarDir* ja *ExecuteWriteADS1248RegisterForSolarDir* alamfunktsioone.

Esimene alamfunktsioon funktsioon määrab ära, millise registri väärtust muudetakse ning samas teostab kontrolli, et programm ei satuks ebamäärasesse olekusse. ADS1248 registrite konfigureerimise alamfunktsiooni tuleb saata kaasa registri nimi ning täisarvuline registri tahetud väärtust kujutav muutuja. Antud alamfunktsioon saadab teisele alamfunktsioonile edasi 8-bitise informatsiooni, mis annab esimese 4 bitiga teada, et kasutatakse *WREG* käsku ning annab viimase 4 bitiga teada, millisesse registrisse kirjutada soovitakse.

Teine alamfunktsioon kasutab registri väärtuse muutmiseks kolme *SPI.transfer()* funktsiooni järjestikuliselt. Esimese 8-bitise informatsioonijadaga saadetakse registri aadress, mille väärtust soovitakse muuta. Teise 8-bitise informatsioonijadaga antakse edasi informatsiooni sellega kohta, mitmes registris soovitakse väärtusi muuta. Kuna antud alamfunktsiooni kasutatakse ainult ühe registri väärtuse muutmiseks, on selle infojada väärtus 0. Viimase *SPI.transfer()* funktsiooni abil saadetakse registri tahetud väärtus.

Kui soovitakse korraga mitut registrit muuta, võib selle jaoks kasutada *setADS1248RegisterValueForSolarDir* funktsiooni, mis on tarkvara arendaja jaoks loodud funktsioon, mille abil on kerge vaevaga võimalik kõiki nelja registrit ühe alamfunktsiooniga muuta.

Registrite väärtuste kirjutamise alamfunktsioonid on kujutatud lisas 15.

5.3.3 ADS1248 registrite väärtuse lugemise alamfunktsioon

ADS1248 registri väärtuse lugemiseks kasutatakse väga sarnase ülesehitusega programmikoodi. Registrite väärtuste lugemise ja kirjutamise alamfunktsioonid on peaaegu samasugused, ent registri väärtuse lugemisel kasutatakse *RREG* käsku.
Registrite väärtuste lugemise alamfunktsioonid on kujutatud lisas 16.

5.3.4 ADS1248 sisemise temperatuuri lugemise alamfunktsioon

ADS1248 mikrolülituse temperatuuri lugemiseks tuleb esialgu konfigureerida registrid õigetele väärtustele, millega näiteks määratakse ära, et soovitakse lugeda mikrolülituse sisest temperatuuri. Järgmisena käivitatakse ADC, valitakse MCP23S17 liinil olev ADS1248 SS liin madalale nivoole ning saadetakse *RDATA* käsk, mille vastuseks saadab ADC 24-bitise informatsioonijada.

Kolmest baidist koosnev informatsioon tuleb ühendada korrutamise tehet kasutades 24-bitiseks informatsioonijadaks. Informatsioonijadal olev informatsioon on töötlemata ning seetõttu tuleb vähima kaaluga biti väärtusele vastava pinge väärtuse ja koefitsiendi abil ümbruskonna temperatuur välja arvutada. Antud koefitsiendi väärtus on kujutatud ADS1248 tehnilisel andmelehel [15, lk 35].

ADS1248 temperatuurilugemise alamfunktsioon on kujutatud lisas 17.

5.3.5 ADS1248 sisendpinge väärtuse lugemise alamfunktsioon

Sisendpinge väärtuse lugemine on taaskord sarnane ümbruskonna temperatuuri lugemise alamfunktsioonile aga lihtsalt registrite väärtused tuleb teistmoodi seadistada. Taaskord kasutatakse *RDATA* käsku ning informatsioonijada teisendatakse koefitsientide abil kasutajale loetavale kujule. Pinge väärtus kuvatakse kasutajale millivoltides.

ADS1248 sisendpinge lugemise alamfunktsioon on kujutatud lisas 18.

5.3.6 Muud ADS1248 alamfunktsioonid

ADS1248 alamfunktsioonide koostamisel kasutati hulgaliselt lihtsamaid alamfunktsioone. Antud funktsioonide hulka kuuluvad *resetADS1248(), startADS1248Conversion(), wakeUpADS1248()* ja *stopADS1248().*

resetADS1248() määrab kõik registrite väärtuse vaikimisi olekusse, *startADS1248Conversion()* alustab ADC informatsiooni kogumise protsessi, mis alustab näiteks temperatuuri lugemist või sisendpinge lugemist. *wakeUpADS1248()* äratab ADC unerežiimist ning *stopADS1248()* saadab ADC unerežiimi.

Antud funktsioonid on kujutatud lisas 19.

6 ELEKTROONIKAKEST JA TRÜKKPLAATIDE KINNITUSED

Energia kogumise ja andmehõive elektroonikasüsteemis olevad suur osa trükkplaate hoiustatakse elektroonikakestas, mis asub päiksekollektori soojuse kogumise konstruktsiooni taga. Projekteeritud elektroonikakesta sees oleva trükkplaadid on Arduino Mega 2560, Arduino Mega 2560 mooduli trükkplaat, Arduino 2560 analoog-kontrolleri mooduli trükkplaat, viis multipleksor trükkplaati ja kaks ADC trükkplaati. Antud elektroonikakesta sisse paigaldatakse ka LCD-kuvar.

Eelnevalt oli nendele trükkplaatidele ehitatud raamistik L-profiiliga liistudest, ent ühtegi dokumentatsiooni antud elektroonikakesta kohta ei olnud. Seetõttu joonestati antud elektroonikakest DS Solidworks 2016 programmi kasutades virtuaalruumi, et oleks võimalik paika panna trükkplaatide ja LCD-kuvari asetus ning kinnitamise viisid.

Vasakus pooles elektroonikakestas asuvad kaks ADC trükkplaati ja viis multipleksori trükkplaati paralleelses ühenduses. Trükkplaadid on omavahel ühendatud 40-piigise ühendusega, mis annab võimaluse luua täiendavate juhtmeteta ühendusi ning säästab ruumi. Nimetatud trükkplaatide kohal asub ka 3,2 tollise diagonaaliga puutetundlik LCD-kuvar, kus on päiksekollektori operaatoril võimalik näha erinevatelt informatsioonikogumise trükkplaatidelt tulevad voolutugevust ning pinget.

Paremas pooles elektroonikakestas asuvad üksteisele peale asetatuna järjestikku Arduino Mega 2560, Arduino Mega mooduli trükkplaat, optoisolatsiooni trükkplaat ning Arduino Mega 2560 analoognuppudega kontroller. Analoognuppudega kontrolleri abil on päiksekollektori operaatoril võimalik juhtida päiksekollektori pöördenurka ja kallutusnurka manuaalselt.

Ilma seinapaneelideta elektroonikakest koos selle sees asuvate trükkplaatide on kujutatud lisas 20. Trükkplaadid on CAD programmis modelleeritud lihtsustatud kujul, st et paigutamisel on kasutatud tehniliselt andmelehtedelt saadud mõõtmetega kujundeid ning joonisel trükkplaate detailselt kujutatud ei ole. Arduino Mega 2560 trükkplaadi joonis on tõmmatud GrabCAD kodulehelt, et antud trükkplaadi kinnitamiseks oleks kõige täpsemad mõõtmed.

KOKKUVÕTE

Tallinna Tehnikaülikooli Elektroenergeetika ja mehhatroonika instituudis on arendamisel termoelektrilisi elemente ehk Peltier'i elemente rakendav päiksekollektor. Päiksekollektor koosneb üldpildis kolmest suuremast osast: paraboolikujulisest nõguspeeglist, soojuskiirguse kogumise konstruktsioonist ja alusest, mille abil on konstruktsioonil võimalik päikse suunas keerata. Päiksekollektor kasutab termoelektrilisi elemente, mis suudavad kahe pinna temperatuuri erinevuse tekkimisel toota elektrienergiat. Päiksekollektori poolt toodetava energia hindamiseks ja kasuteguri arvutamiseks arendati energia kogumise süsteemi, andmehõive süsteemi ja päikse liikumise järgimise süsteem. Töö eesmärgi saavutamiseks koostati päiksekollektori juhtimiseks riistvara ja tarkvara ning riistvara hoiustamise jaoks elektroonikakest.

Käesoleva bakalaureuse töö teooria osas käsitleti üksikasjalikumalt SPI protokolli ja trükkplaatide projekteerimise tavasid ja reegleid. Teooria osas on kirjutatud SPI protokollist üldiselt, andmevahetusest, signaaliliinidest ning viisidest, mille abil on võimalik SPI protokolli kasutamine muuta ümbruskonnast tulenevate häiringute suhtes töökindlamaks. Trükkplaatide teooria osas on kirjutatud trükkplaatide projekteerimisest üldiselt, on kirjutatud elektriskeemide ja tootmiskavandite koostamisest kuni väiksemate detailideni (nt läbiviigud, radade paksused jpm).

Praktika osas käsitletakse projekteeritud ja parandatud trükkplaatide tööpõhimõtteid ja elektriskeeme, tarkvara arendamisel kasutatud alamfunktsioone ja elektroonikakesta ülesehitust. Eelnevalt oli päiksekollektorile projekteeritud informatsiooni kogumise ja analoog-digitaalmuunduri trükkplaadid, ent nendel esines esialgsel elektriskeemil vigu. Käesoleva lõputöö raames koostati nimetatud trükkplaadid, teostati tõrkeotsinguid ja parandati kõik projekteerimise jooksul tekkinud vead. Lisaks paranduste tegemisele projekteeriti Diptrace programmi kasutades energia kogumise ja andmehõive süsteemi arendamiseks veel neli trükkplaati: võimendi ja pingejagurite trükkplaat, CPC1832 sensori trükkplaat, signaalide optoisoleerimise trükkplaat ning GPS ja SD-kaardi moodulitega logimise trükkplaat.

Informatsiooni kogumise ja analoog-digitaalmuunduri trükkplaatidele kirjutati ka juhtimise tarkvara. Informatsiooni kogumise trükkplaadile loodud tarkvara abil on võimalik lugeda näiteks termoelektriliste elementide ja trükkplaadi töötemperatuure ja termoelektriliste elementide poolt väljastatud pingeid. Analoog-digitaalmuunduri trükkplaadile koostatud tarkvara osa hõlmab ADS1248 ADC registrite konfigureerimise ja lugemise alamfunktsioone ning termopaaride ja ADS1248 mikrolülituse temperatuuri väärtuste lugemise alamfunktsioone. Lisaks on energia salvestuse ja andmehõive lihtsustamiseks koostatud digitaal-analoogmuundurit juhtimise, multipleksor trükkplaati juhtimise ja MCP23S17 adresseerimiskomponendi konfigureerimise tarkvara osa. Tarkvara arendamiseks kasutati Arduino IDE 1.8.2 programmi ning süsteemi juhtimiseks kasutati Arduino Mega 2560 trükkplaati.

Projekteeritud ja parandatud riistvara jaoks projekteeriti olemasolev elektroonikakesta raamistik DS Solidworks programmi kasutades virtuaalruumi, mis oli vajalik kinnituste kuju ja trükkplaatide positsioonide määramiseks.

SUMMARY

The Department of Electrical Power Engineering and Mechatronics of Tallinn University of Technology is developing a solar collector utilizing thermoelectric elements, also known as Peltier elements. The solar collector is composed of three subsystems: a parabolic concave mirror, a heat collector, and a base that can turn the assembly towards the Sun. The thermoelectric modules are able to convert heat into electric energy if the opposing faces of the modules are experiencing a temperature difference. An energy storage system, a data management system, and a solar tracking system were additionally developed in order to characterize the performance and efficiency of the solar collector. In addition to the constructed hardware, the necessary software was also programmed.

The theoretical section of the thesis describes in detail the SPI hardware communication protocol as well as the guidelines and rules of designing printed circuit boards. The specifically treated issues related to SPI include a general introduction and discussions on data exchange, signal lines, and methods to improve the robustness of the protocol to external disturbances from the operating environment. The section on PCBs gives a general introduction to the design of PCBs, as well as describing the specifics down to the finer details, such as signal line widths.

The practical section of the thesis describes the working principles of the PCBs and electrical circuits designed for the solar collector. Furthermore, the subfunctions used in the developed software and the design of the housing of the electronics are presented. A previous design iteration of the solar collector included data collecting and ADC PCBs, but these had some design faults. The scope of the present thesis includes the development and testing of the new PCBs, including design iterations made in order to eliminate initial design faults. What is more, four additional PCBs were developed for the energy storage and data logging systems: an amplifier and voltage divider PCB, a PCB for the CPC1832 sensor, an optoisolating PCB for the signals, and a PCB for the GPS and SD-card modules.

Software was developed for the data collecting and ADC PCBs. This software provides the ability to, for instance, monitor the operating temperatures and the voltages on the terminals of the thermoelectric elements. The ADC PCB software includes ADS1248 register configuration and reading subfunctions as well as reading subfunctions for reading the temperatures of the thermocouples and ADS1248 crystal. In addition, software was developed to manage the operation of the ADC, the multiplexor PCB, and the configuration of the MCP23S17 input/output expander. The Arduino IDE 1.8.2 software was used to program the software and an Arduino Mega 2560 PCB was used to control the system.

The housing for the electronics was designed using DS Solidworks in order to spatially integrate the components.

7 KASUTATUD KIRJANDUS

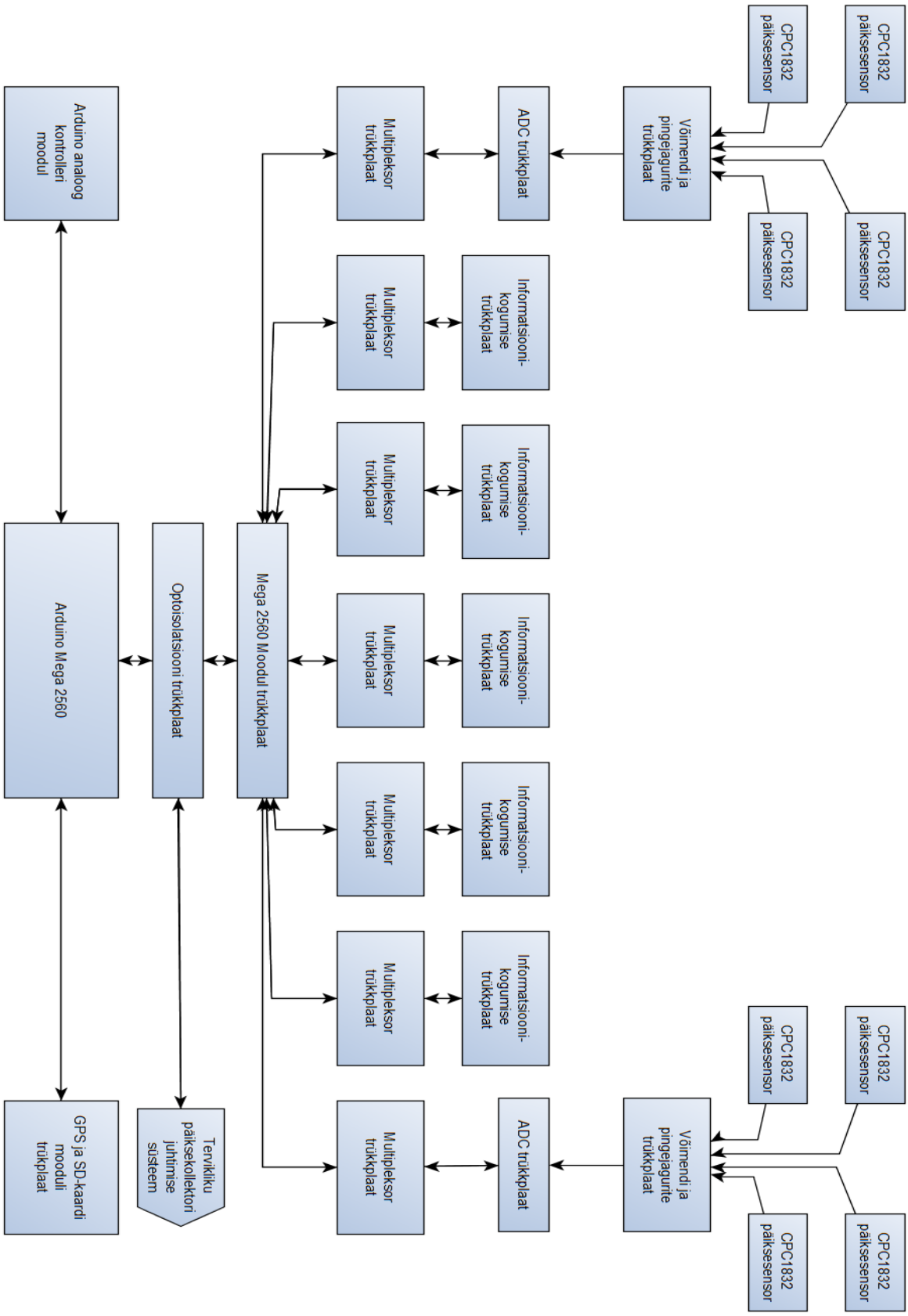
1. *Serial Peripheral Interface Bus*
   https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
   (02.05.2017, 03.05.2017)
2. *SPI library*
   https://www.arduino.cc/en/reference/SPI
   (02.05.2017, 03.05.2017)
3. *Serial Peripheral Interface (SPI)*
   https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi
   (02.05.2017, 03.05.2017)
4. *Introduction to I^2C and SPI protocols*
   http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/
   (03.05.2017)
5. Thomas Kugelstadt – *Extending the SPI bus for long-distance communication*
   http://www.ti.com/lit/an/slyt441/slyt441.pdf
   (03.05.2017)
6. *Interface Circuits for TIA/EIA-644 (LVDS), Design Notes*
   http://www.ti.com/lit/an/slla038b/slla038b.pdf
   (03.05.2017)
7. *Single/Dual LVDS Line Driver with Ultra-Low Differential Skew in SOT23*
   https://www.maximintegrated.com/en/products/interface/high-speed-signaling/MAX9110.html
   (03.05.2017)
8. *Printed Circuit Board*
   https://en.wikipedia.org/wiki/Printed_circuit_board
   (03.05.2017, 04.05.2017)
9. David L. Jones – *PCB Design Tutorial*
   https://server.ibfriedrich.com/wiki/ibfwikien/images/d/da/PCB_Layout_Tutorial_e.pdf
   (04.05.2017)
10. *Package Dimensions*
    https://www.ichaus.de/upload/pdf/Package%20dimensions%20MSOP,%20SSOP,%20TSSOP-A2.pdf
    (04.04.2017)
11. *Nõuanded*
    http://www.kamitra.ee/nouanded/
    (04.05.2017)
12. *Arduino MEGA 2560 & Genuino MEGA 2560*
    https://www.arduino.cc/en/Main/arduinoBoardMega2560
    (05.05.2017)
13. *Microchip MCP23017/MCP23S17 16-bit I/O Expander with Serial Interface*

http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf
(05.05.2017, 06.05.2017)

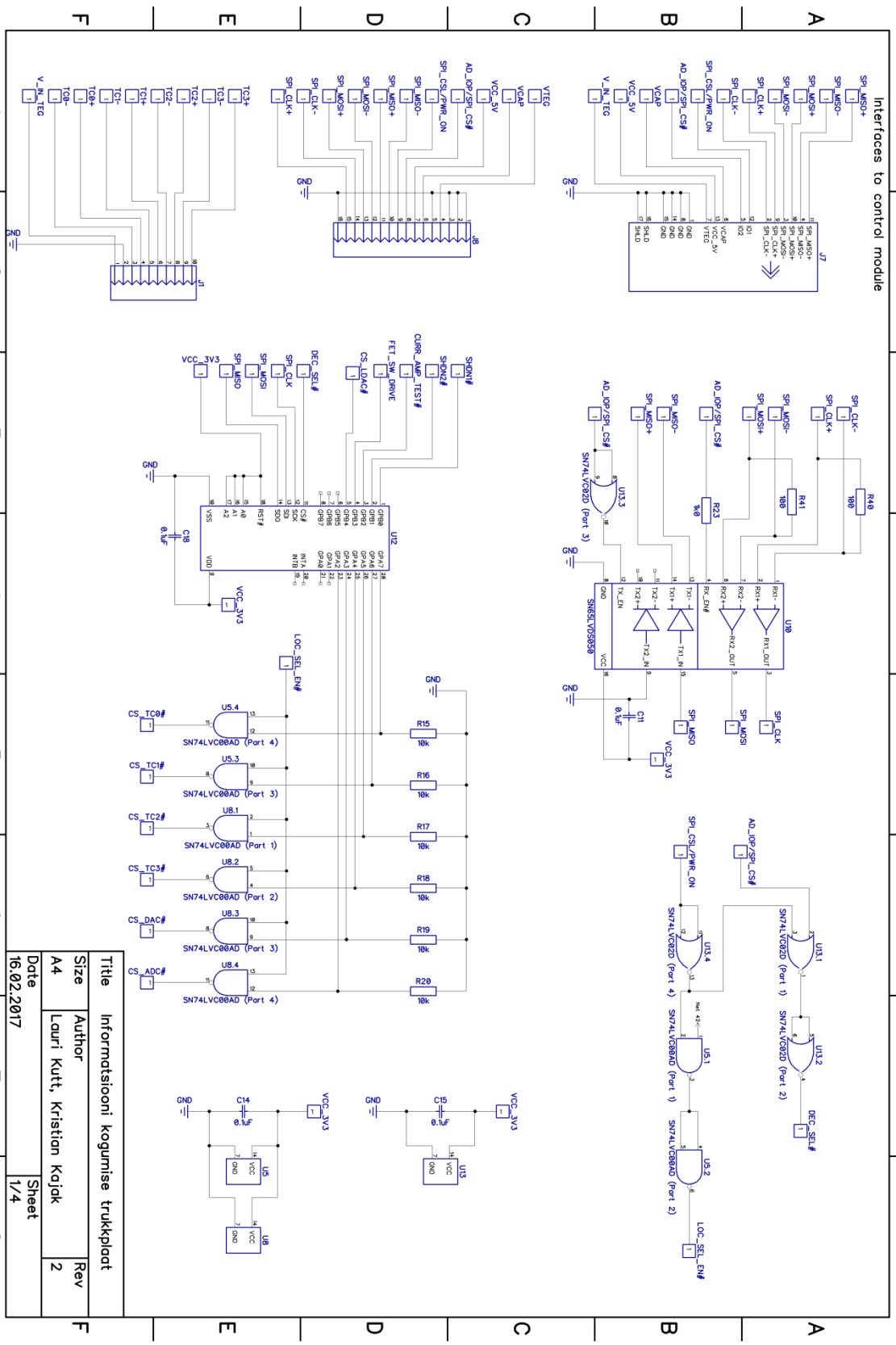
14. *Microchip MCP4902/4912/4922 8/10/12-Bit Dual Voltage output Digital-to-Analog Converter with SPI Interface*
http://ww1.microchip.com/downloads/en/DeviceDoc/22250A.pdf
(05.05.2017, 06.05.2017, 07.05.2017)
15. *ADS124x 24-Bit, 2-kSPS, Analog-to-Digital Converters With Programmable gain Amplifier (PGA) For Sensor Measurement*
http://www.ti.com/lit/ds/symlink/ads1248.pdf
(06.05.2017, 07.05.2017, 9.05.2017)

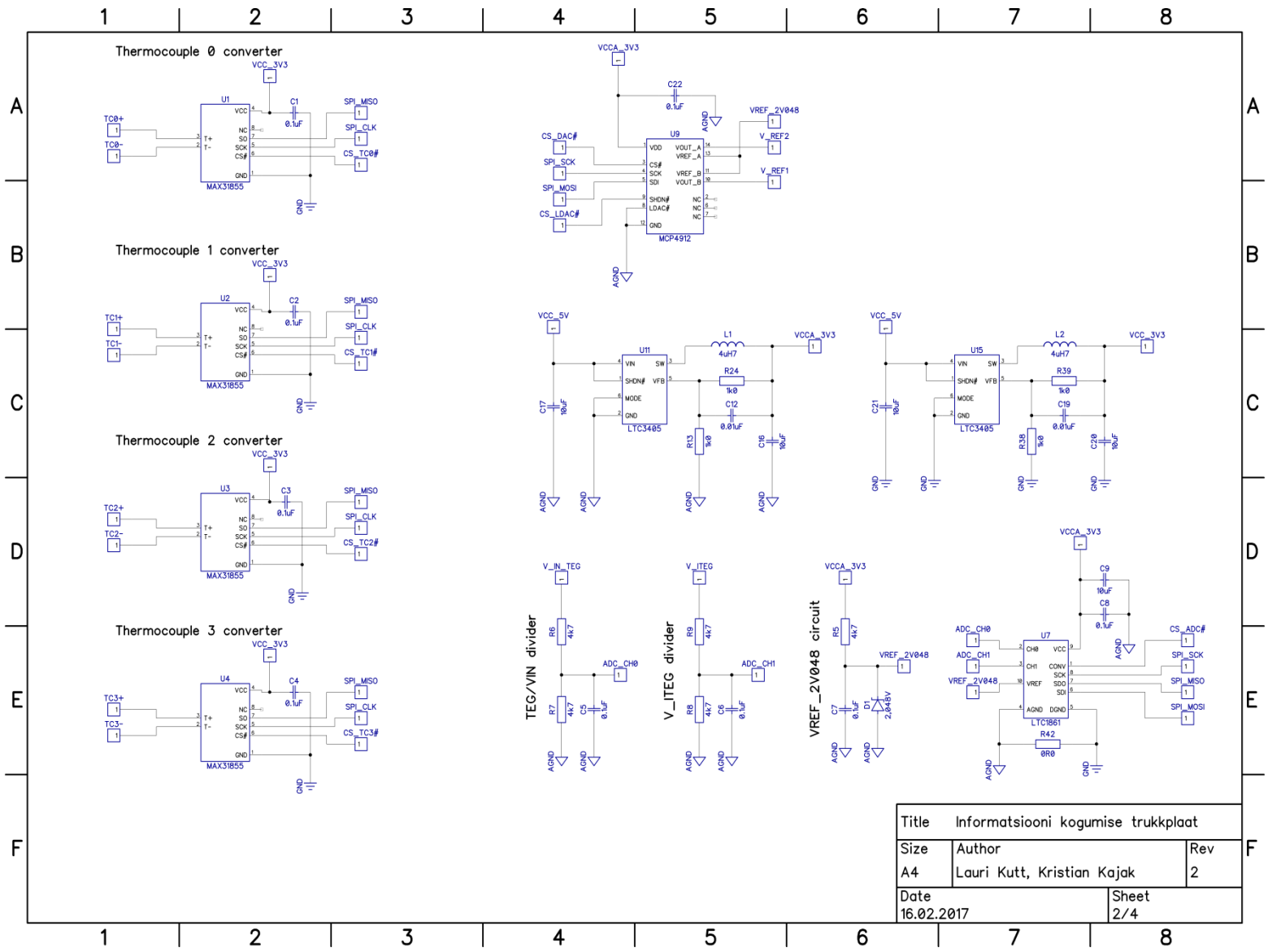
LISAD

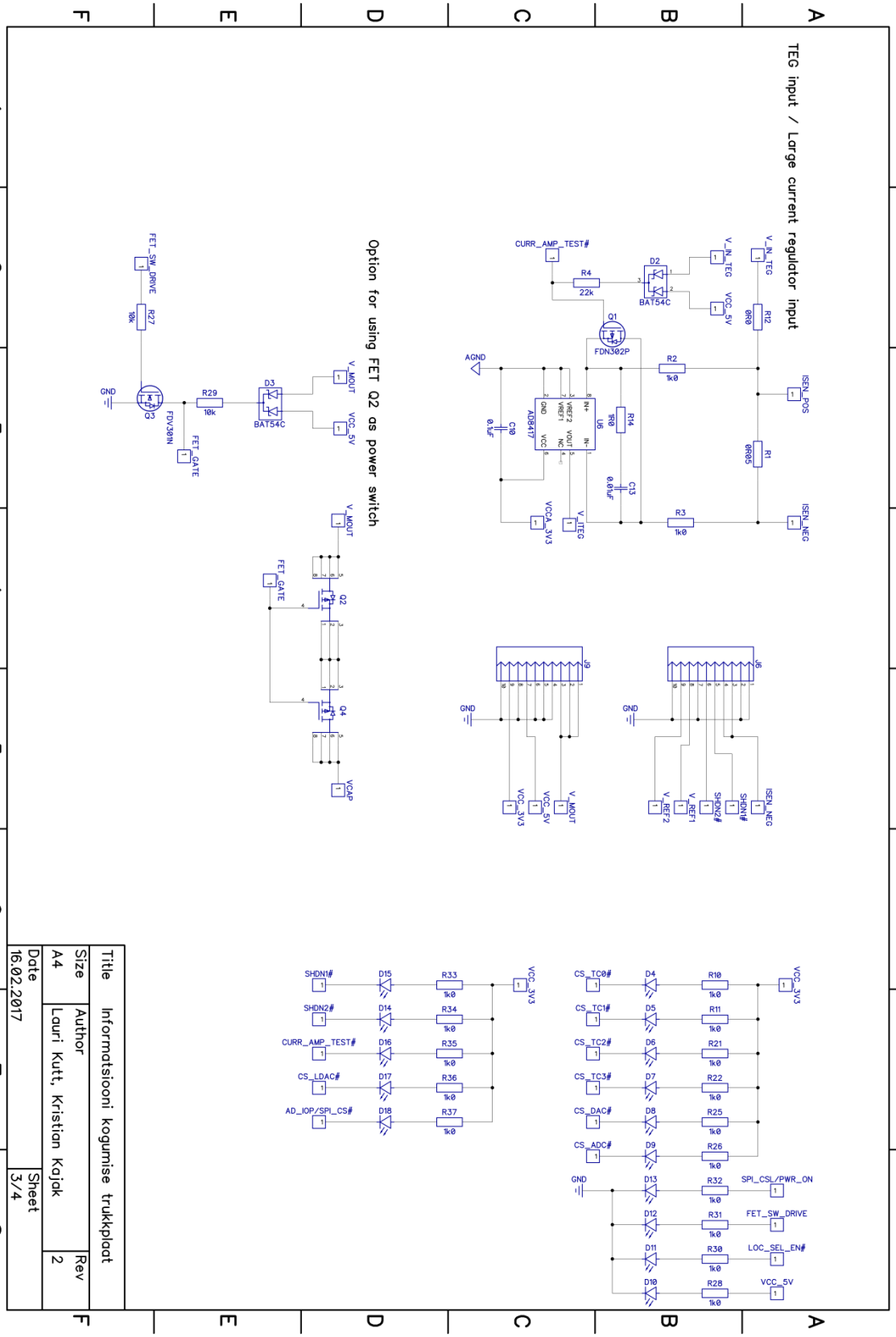
LISA 1 TERVLIKLIK ENERGIA KOGUMISE JA ANDMEHÕIVE ELEKTROONIKASÜSTEEMI ÜLESEHITUS

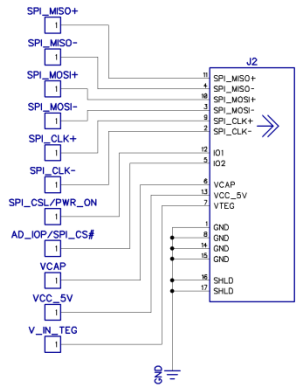


LISA 2 INFORMATSIOONI KOGUMISE TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND

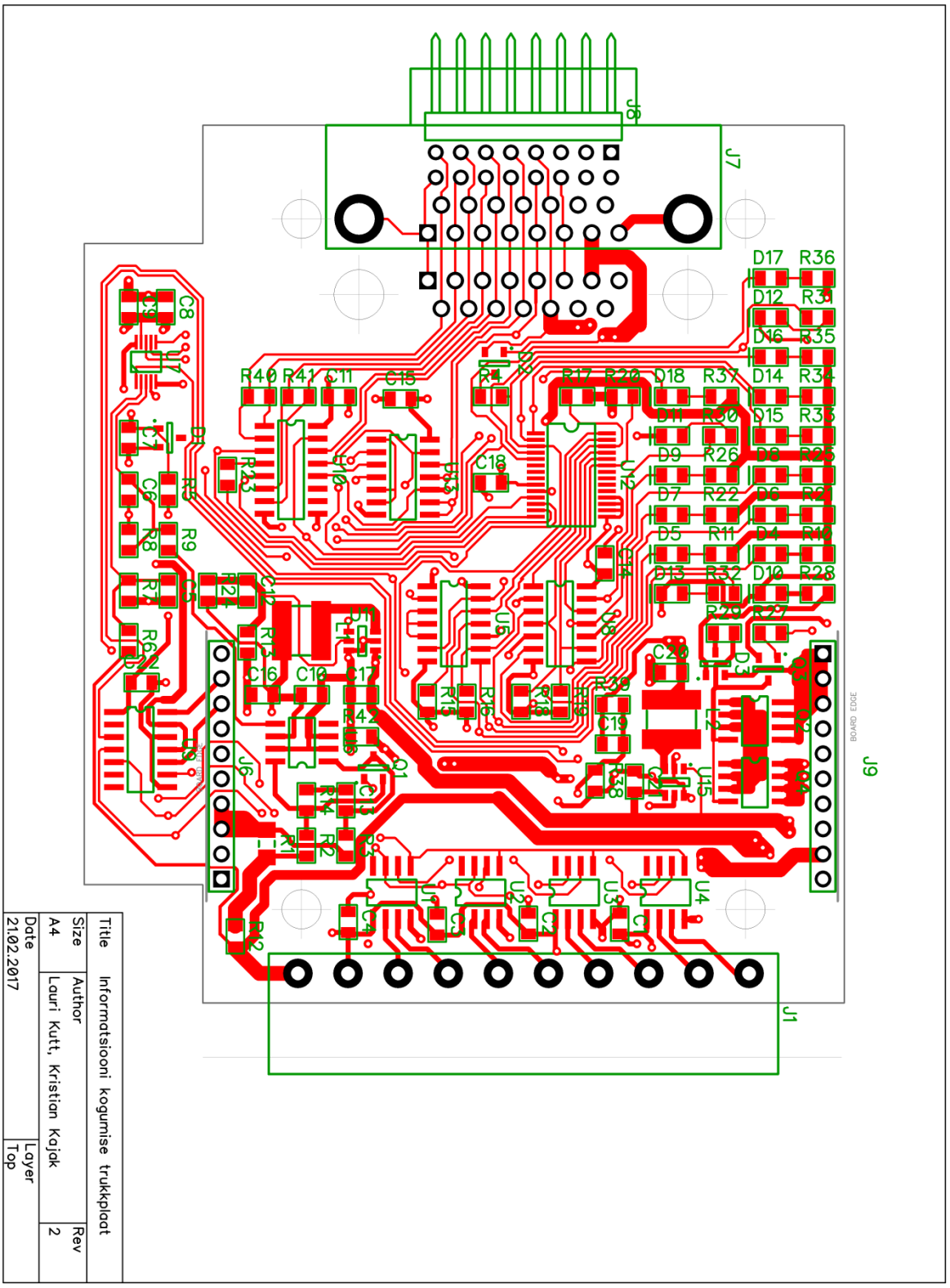


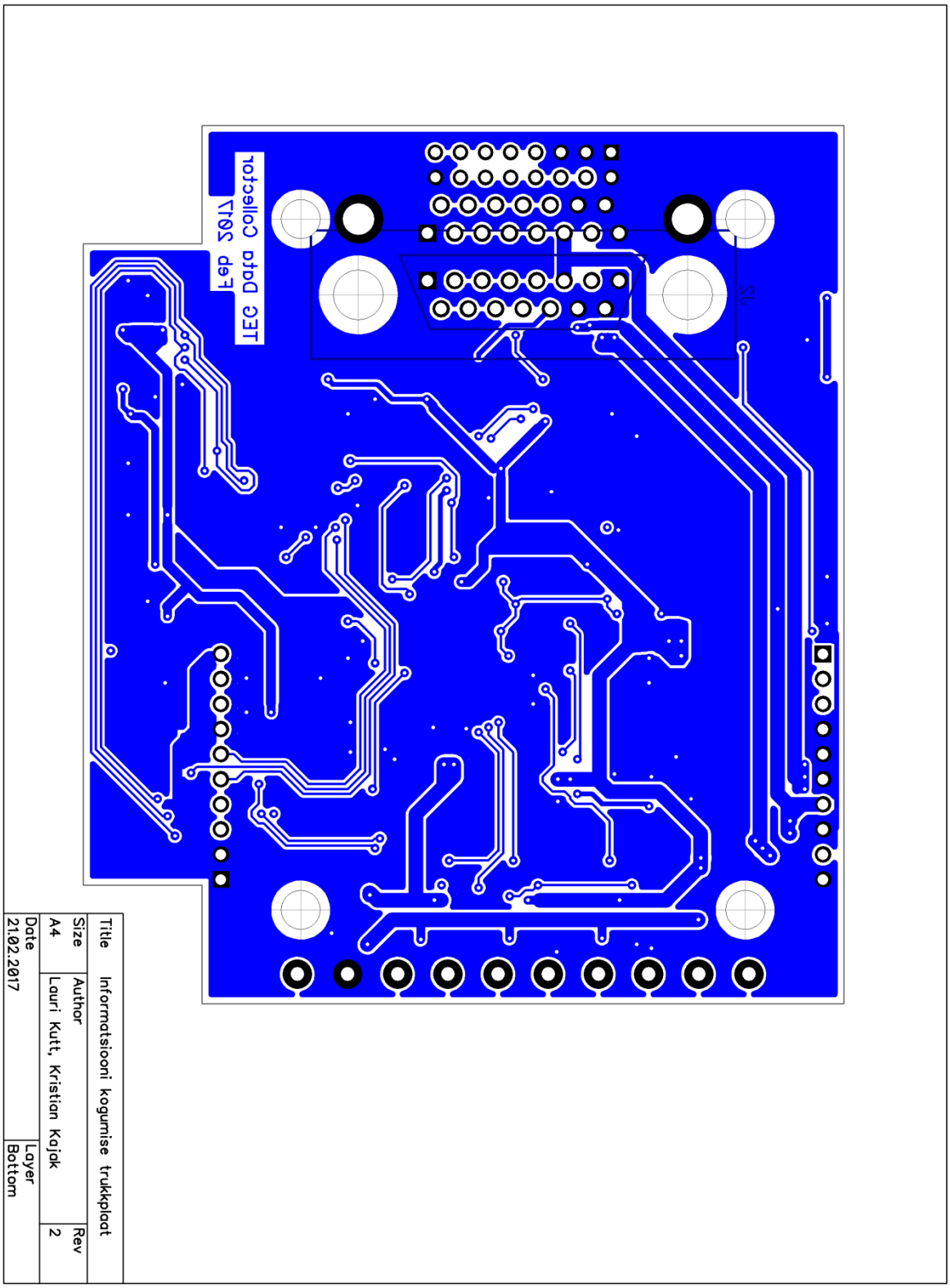

Thermocouple 0 converter
Thermocouple 1 converter
Thermocouple 2 converter
Thermocouple 3 converter
MAX31855
MCP4912
LTC3405
LTC1861
TEG/VIN divider
V_ITEG divider
VREF_2V048 circuit
Title Informatsiooni kogumise trukkplaat
Size A4
Author Lauri Kutt, Kristian Kajak
Rev 2
Date 16.02.2017
Sheet 2/4


TEG input / Large current regulator input
Option for using FET Q2 as power switch
Title Informatsiooni kogumise trukkplaat
Size A4
Author Lauri Kutt, Kristian Kajak
Rev 2
Date 16.02.2017
Sheet 3/4

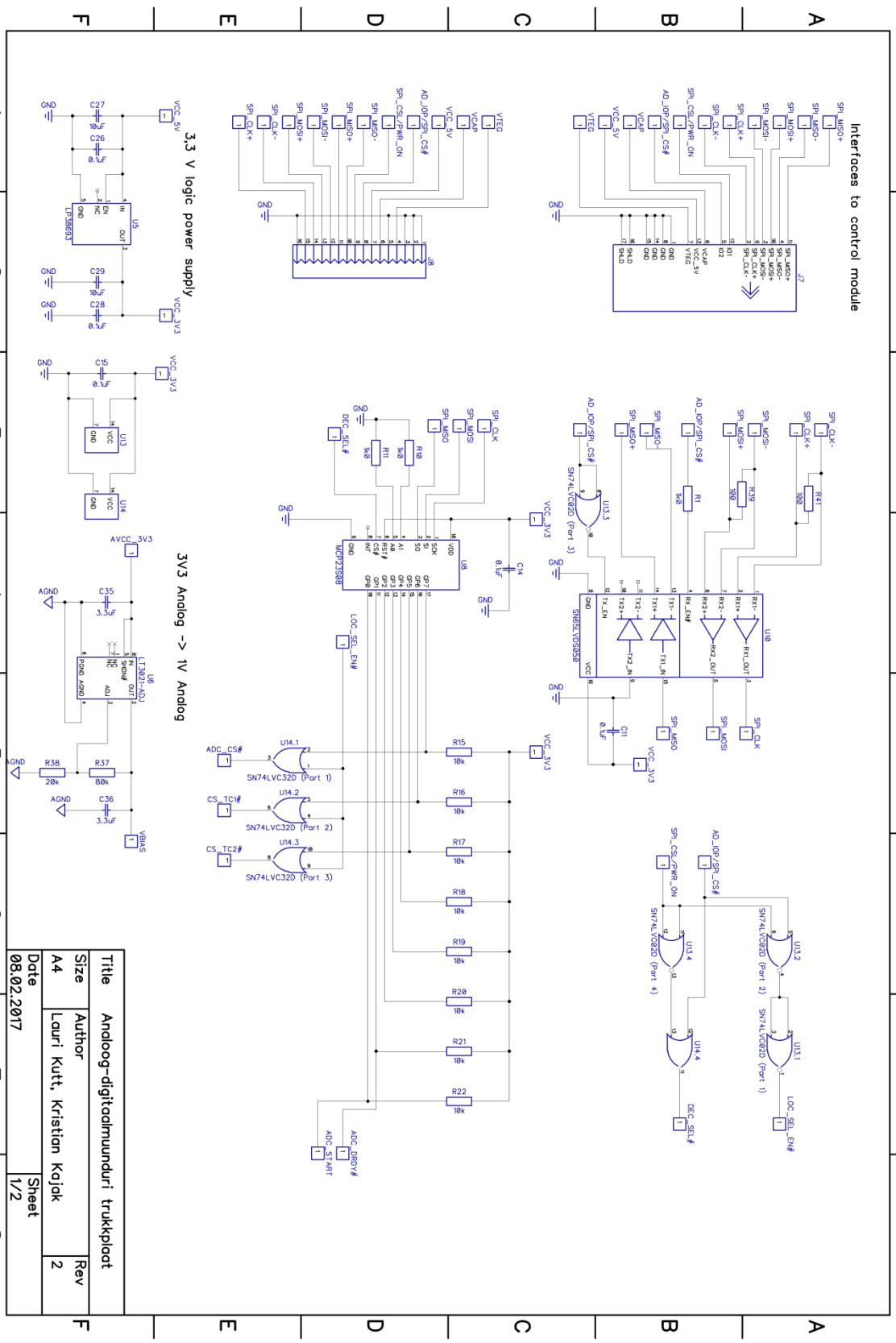


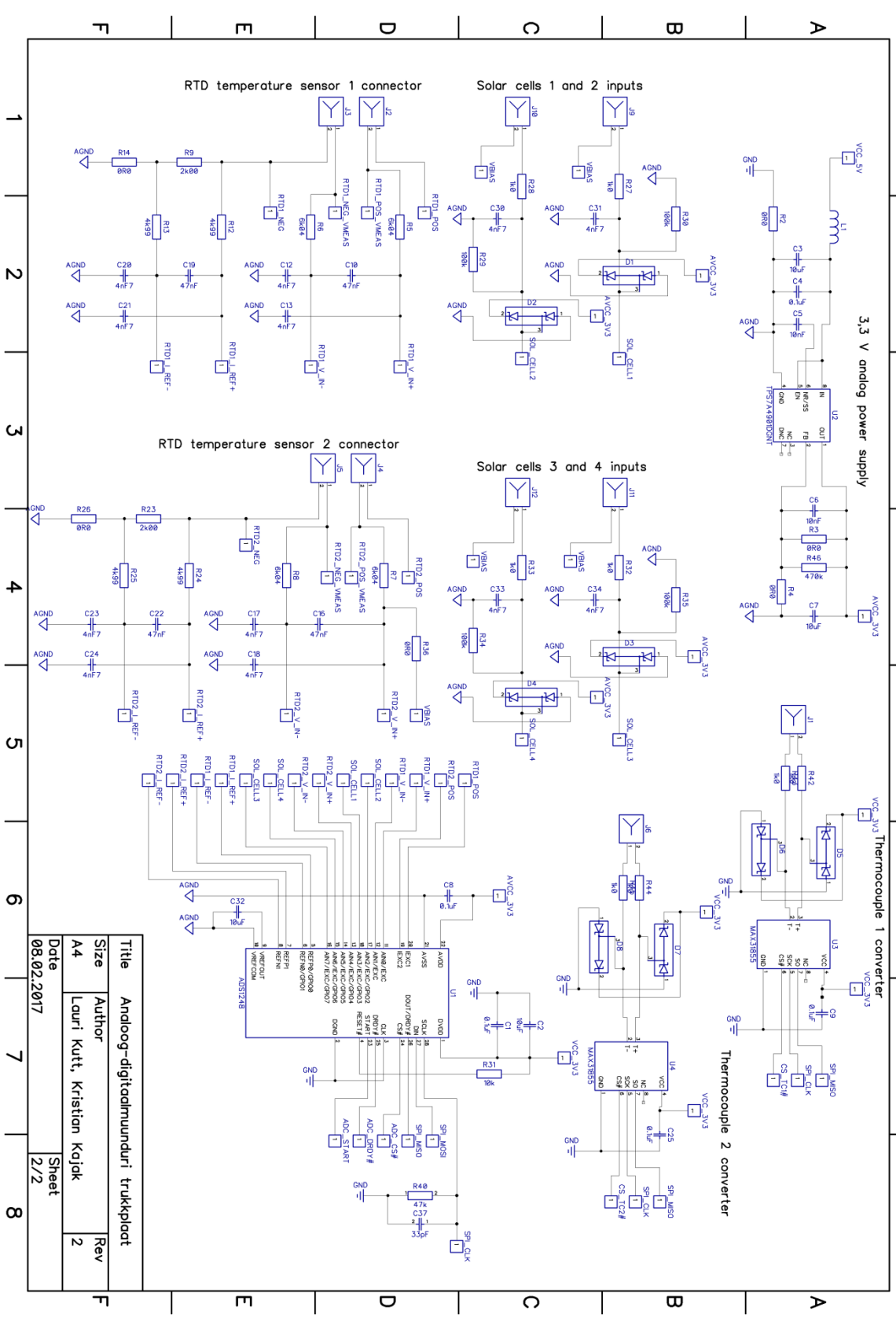
Title	Informatsiooni kogumise trukkplaat	
Size: A4	Author: Lauri Kutt, Kristian Kajak	Rev: 2
Date: 16.02.2017	Sheet: 4/4	

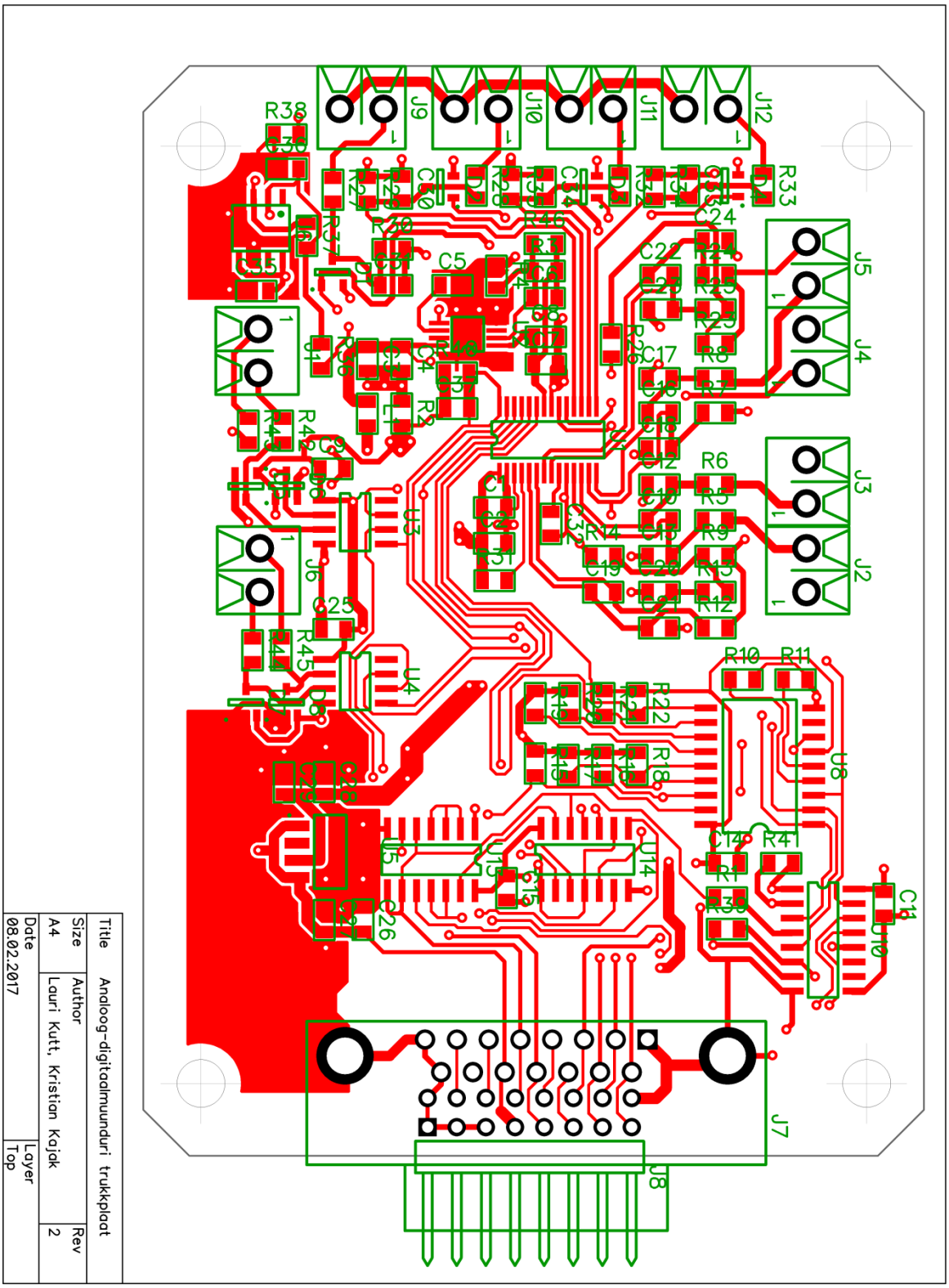

Title Informatsiooni kogumise trukkplaat
Size A4
Author Lauri Kutt, Kristian Kajak
Rev 2
Date 21.02.2017
Layer Top
J1
J6
J7
J8
J9
BOARD EDGE

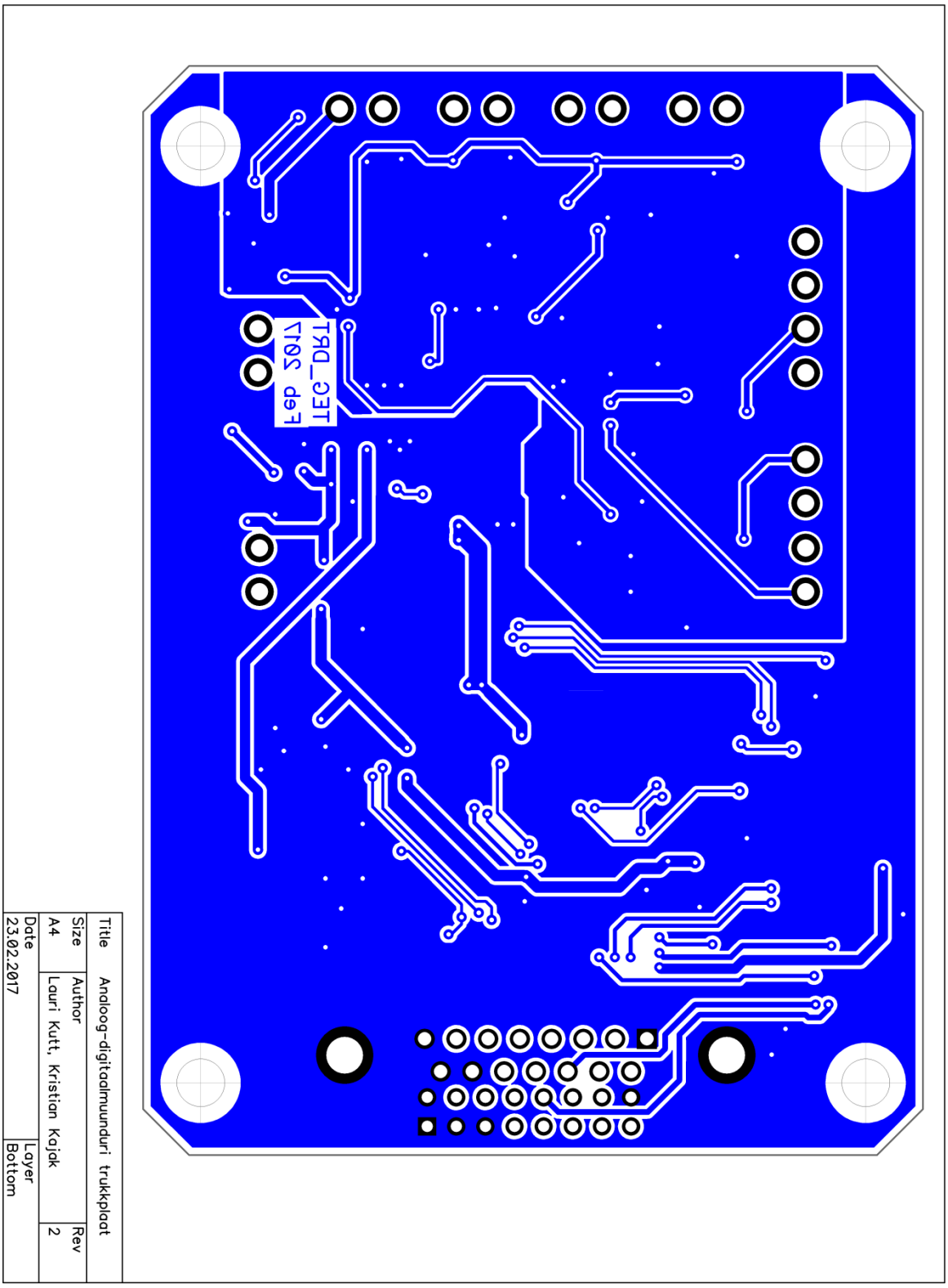

TEG Data Collector
Feb 2017
Title Informatsiooni kogumise trukkplaat
Size A4
Author Lauri Kutt, Kristian Kajak
Rev 2
Date 21.02.2017
Layer Bottom

LISA 3 ANALOOG-DIGITAALMUUNDURI TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND

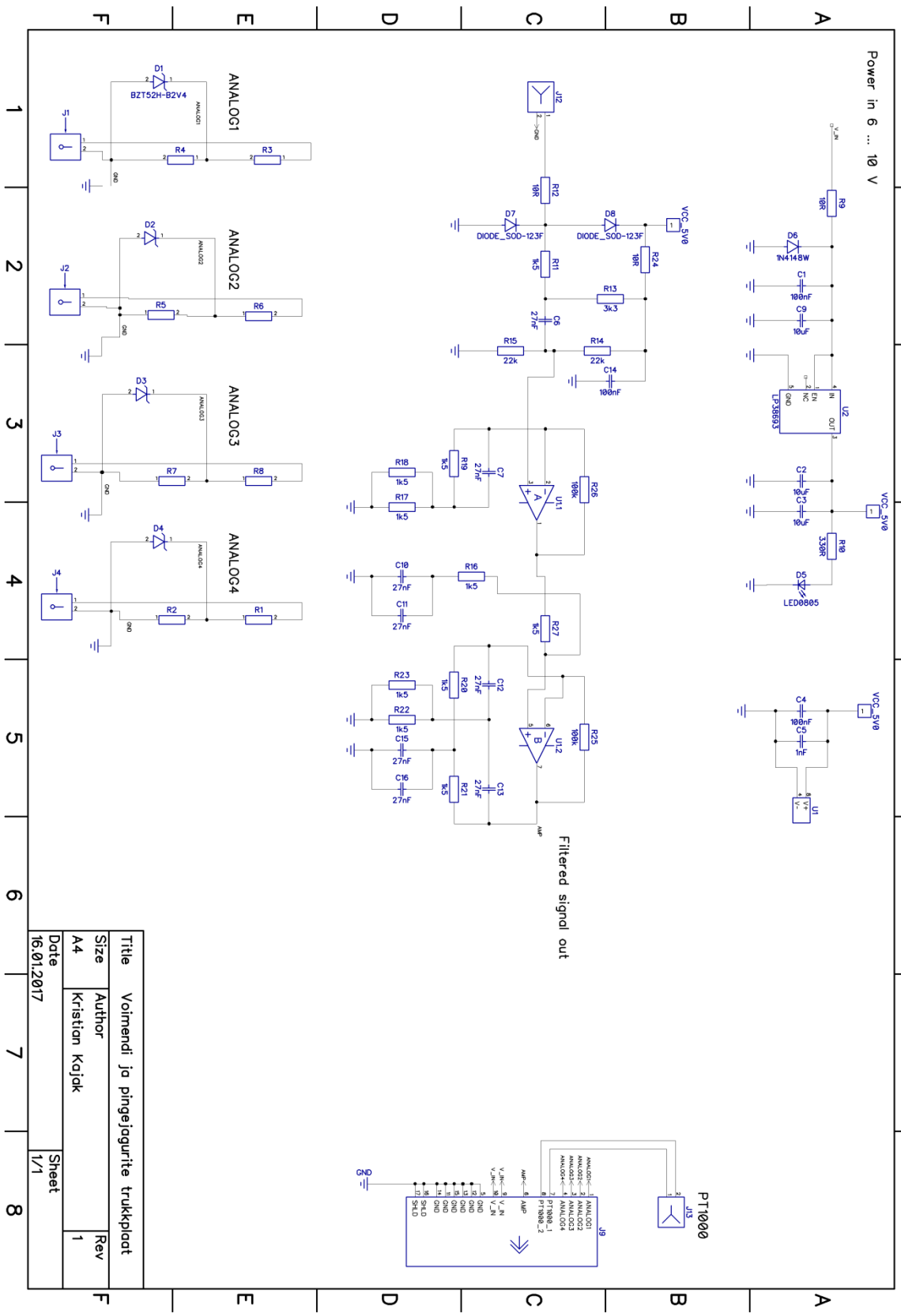


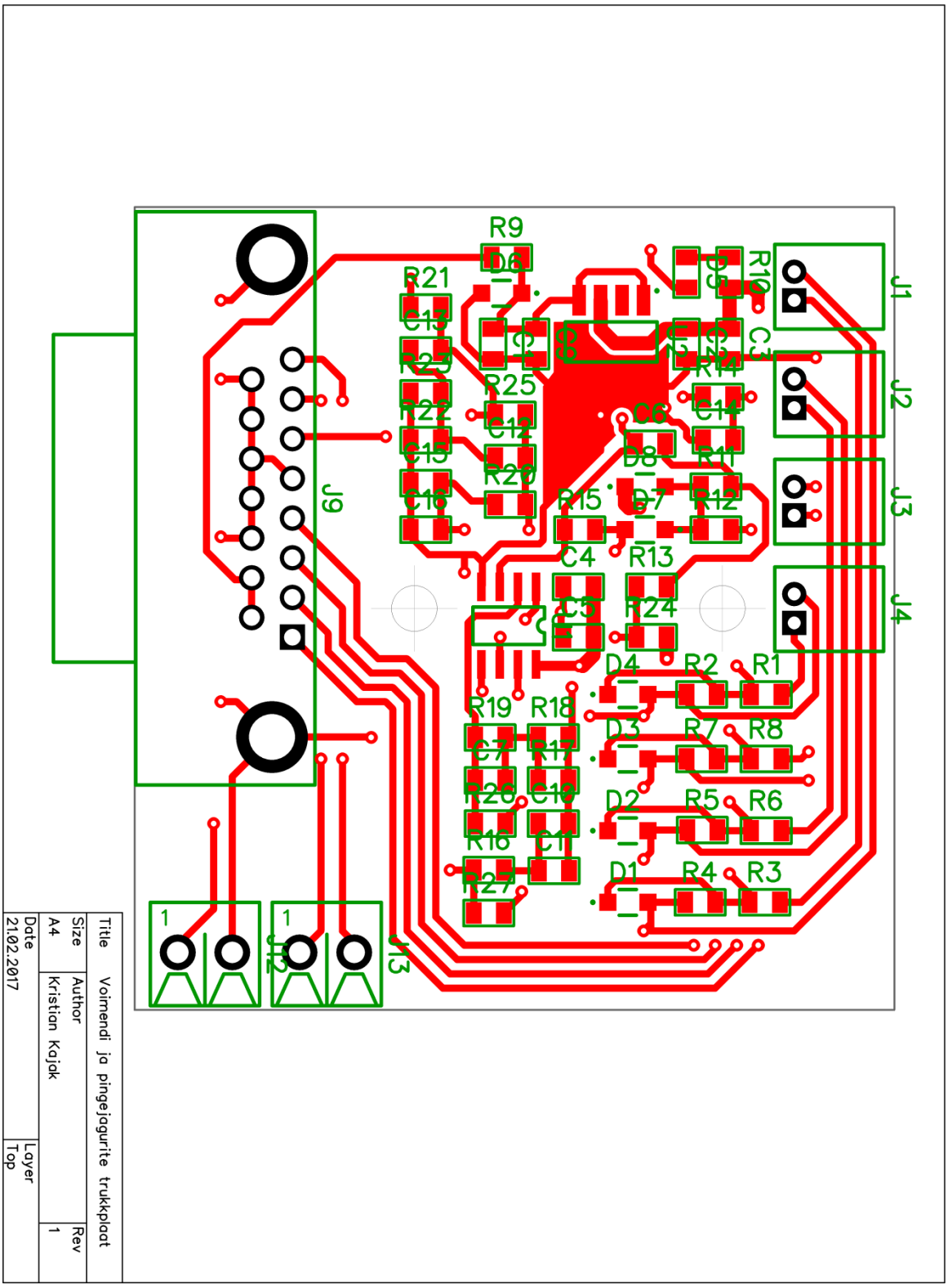

RTD temperature sensor 1 connector
Solar cells 1 and 2 inputs
3,3 V analog power supply
RTD temperature sensor 2 connector
Solar cells 3 and 4 inputs
Thermocouple 1 converter
Thermocouple 2 converter
Title Analoog-digitaalmuunduri trukkplaat
Size A4
Author Lauri Kutt, Kristian Kajak
Rev 2
Date 08.02.2017
Sheet 2/2

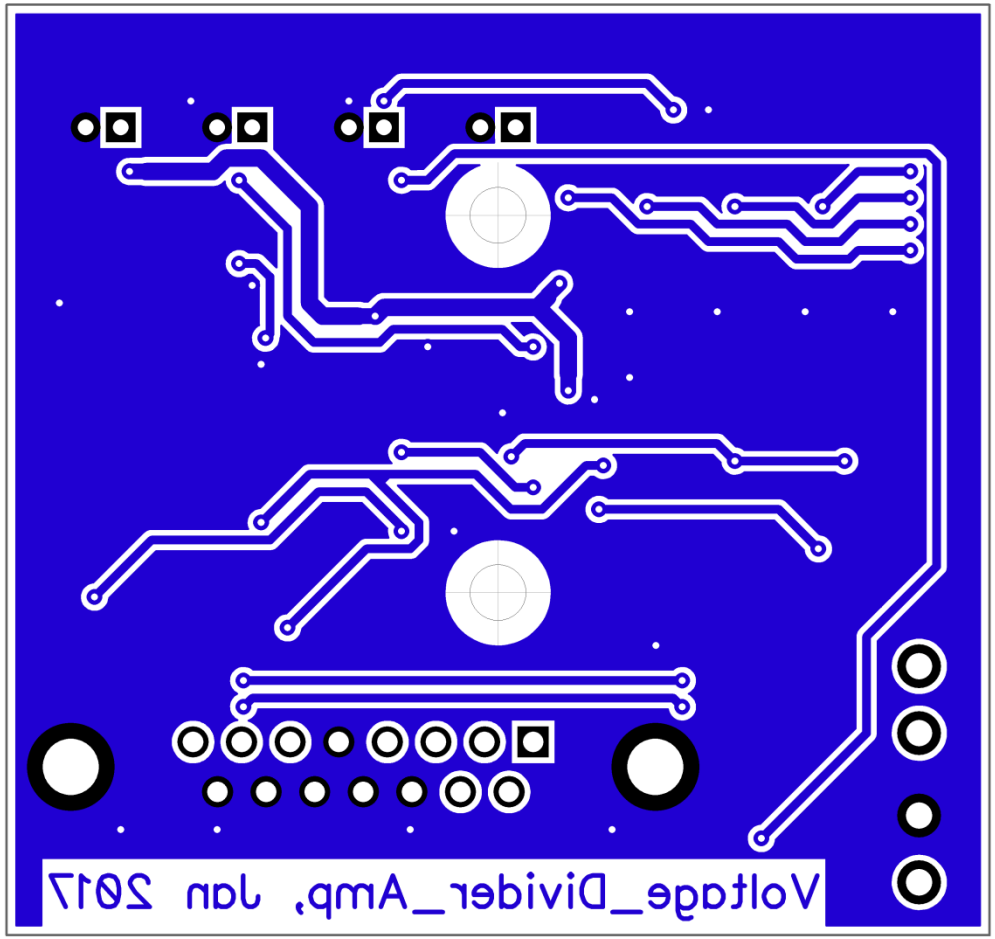

Title
Analoog-digitaalmuunduri trukkplaat
Size
A4
Author
Lauri Kutt, Kristian Kajak
Rev
2
Date
08.02.2017
Layer
Top


Title Analoog-digitaalmuunduri trukkplaat
Size A4
Author Lauri Kutt, Kristian Kajak
Rev 2
Date 23.02.2017
Layer Bottom

LISA 4 VÕIMENDI JA PINGEJAGURITE TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND

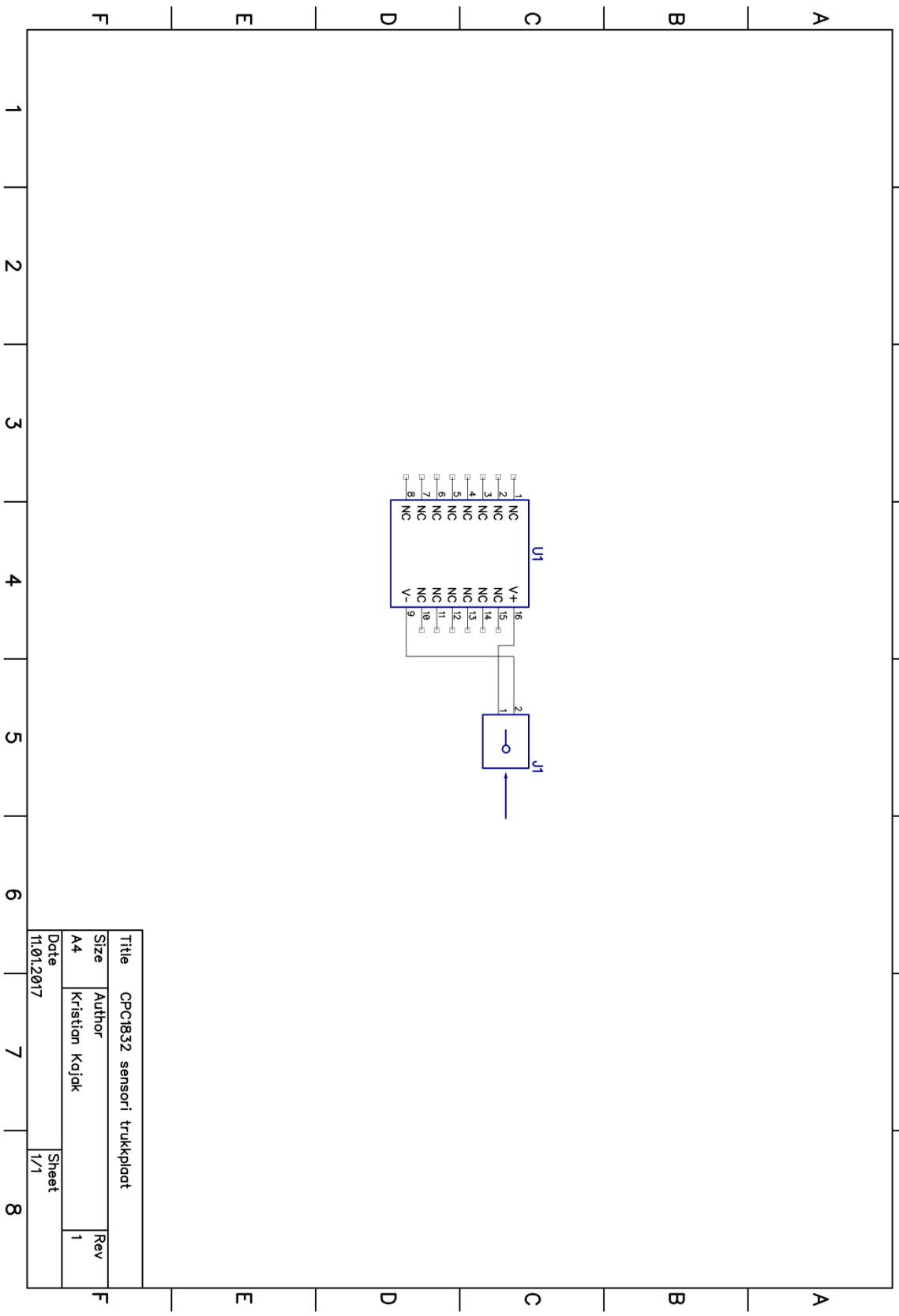


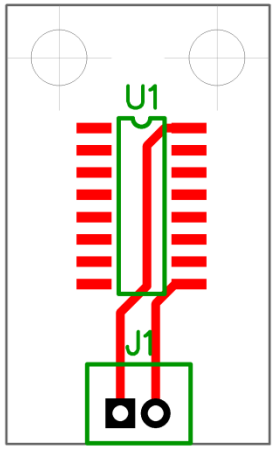

R9
D6
R21
C13
R23
R22
C15
C16
R25
C12
R20
C6
D8
R15
D7
R12
R11
C14
C4
R13
C5
R24
D4
R2
R1
R19
R18
D3
R7
R8
C7
R17
R26
C10
D2
R5
R6
R16
C11
R27
D1
R4
R3
J1
J2
J3
J4
J9
J12
J13
1
1
Title Voimendi ja pingejaguri trukkplaat
Size A4
Author Kristian Kajak
Rev 1
Date 21.02.2017
Layer Top



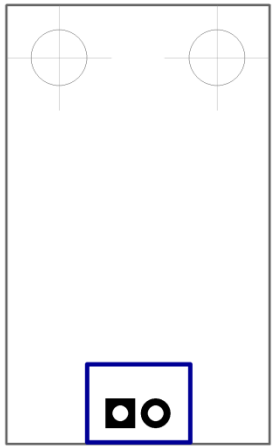
Title	Voimendi ja pingejagurite trukkplaat	
Size A4	Author Kristian Kajak	Rev 1
Date 21.02.2017	Layer Bottom	

LISA 5 CPC1832 SENSORITE TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND



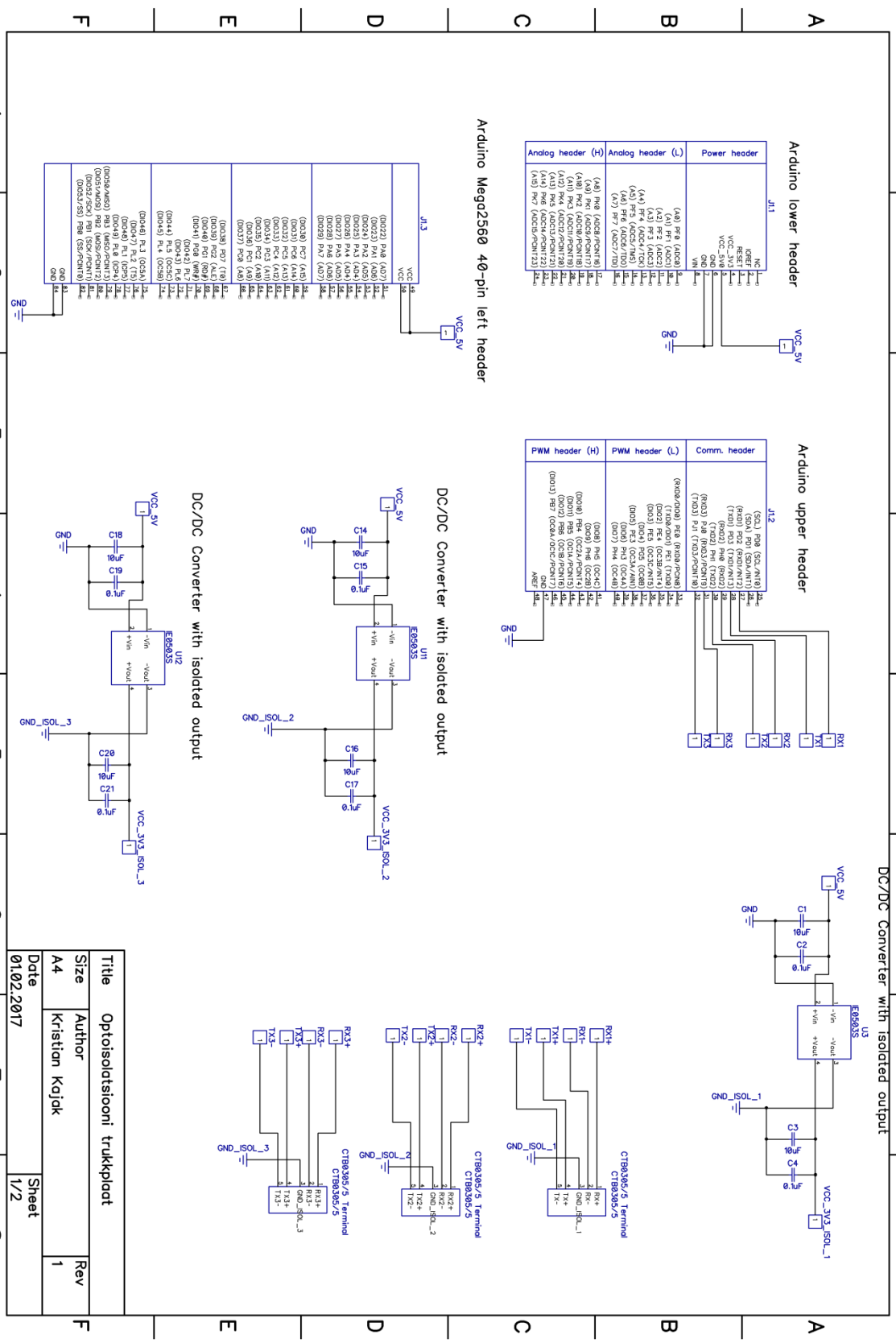


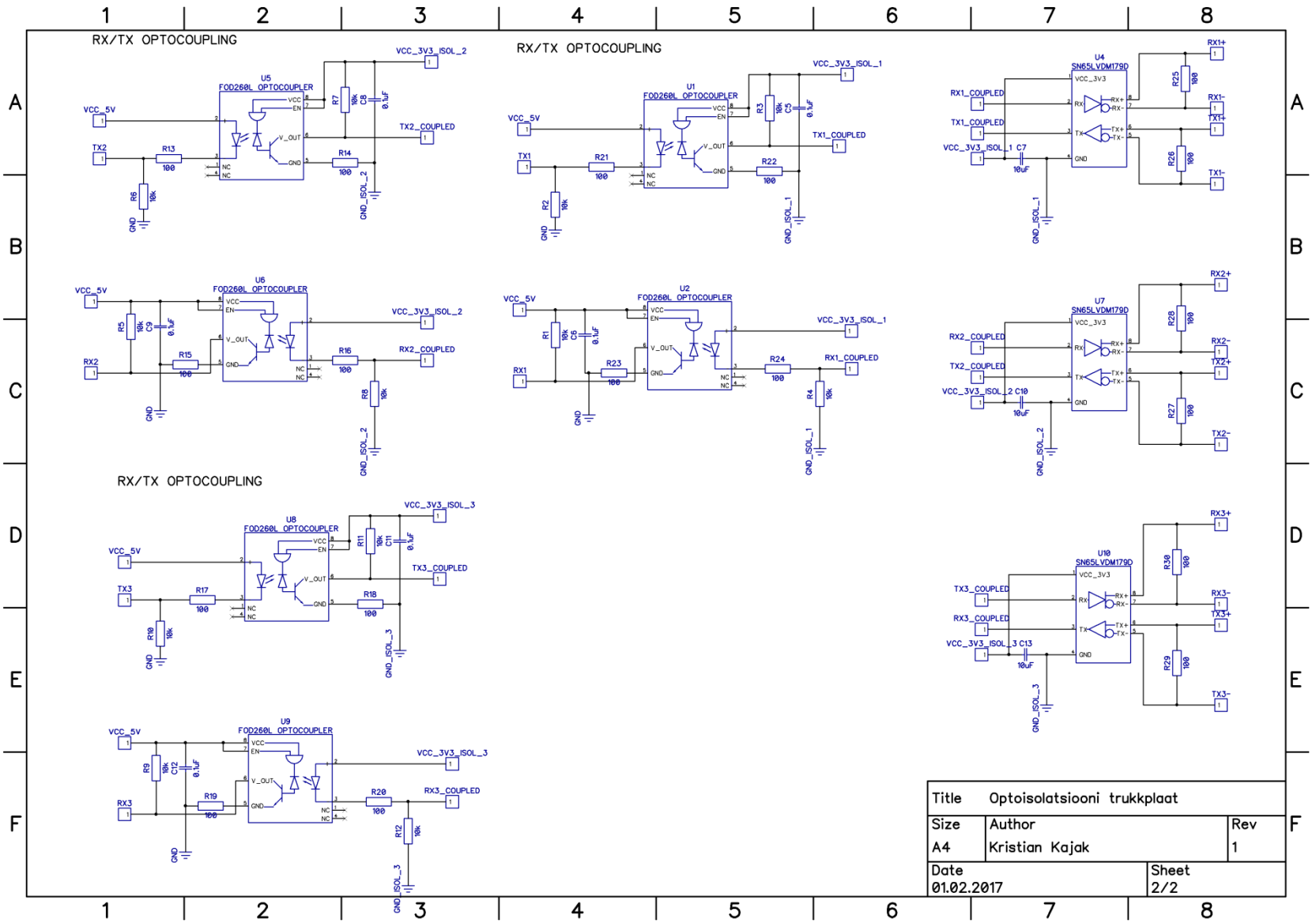
Title	CPC1832 sensori trukkplaat	
Size A4	Author Kristian Kajak	Rev 1
Date 16.02.2017	Layer Top	

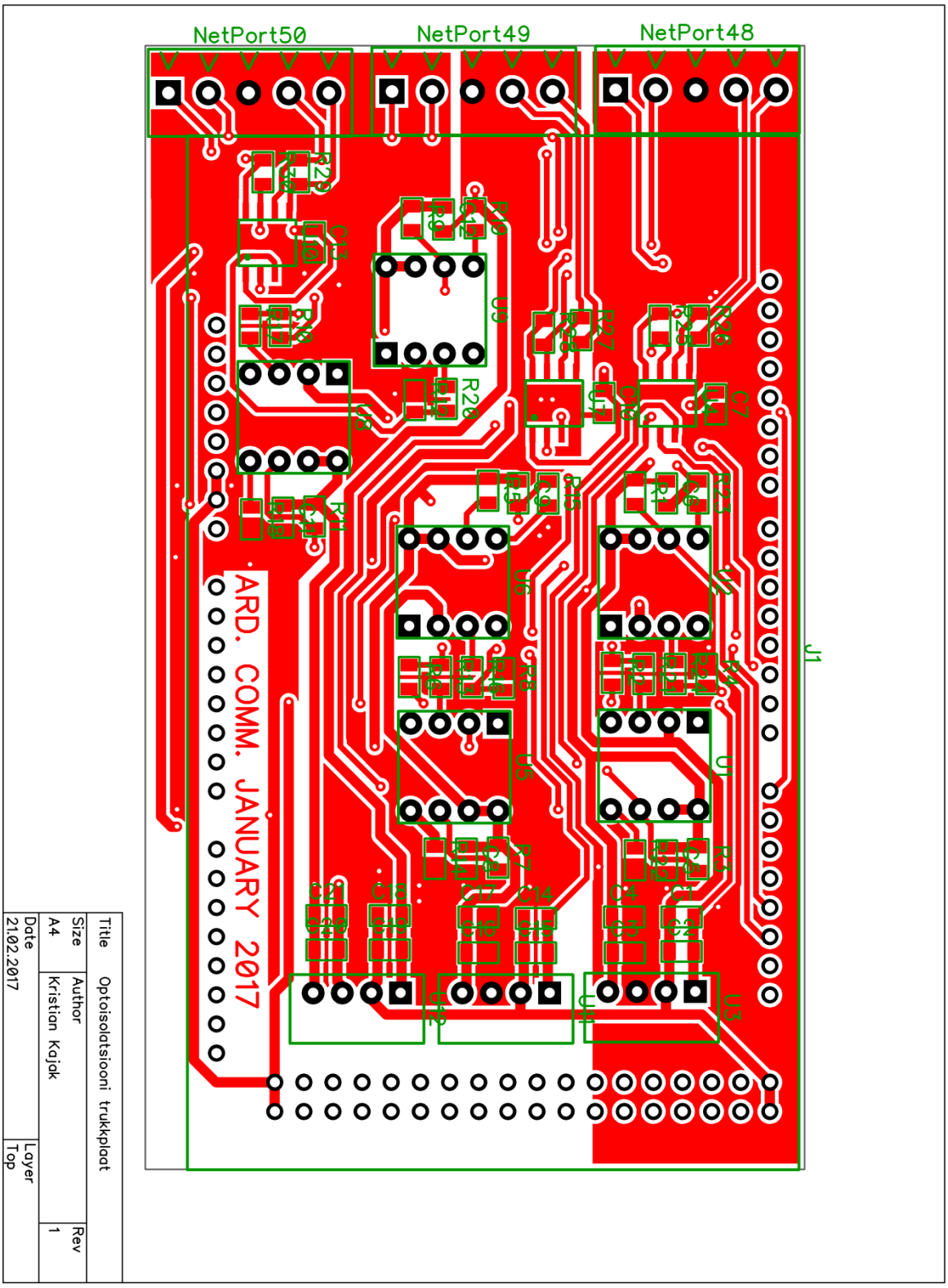


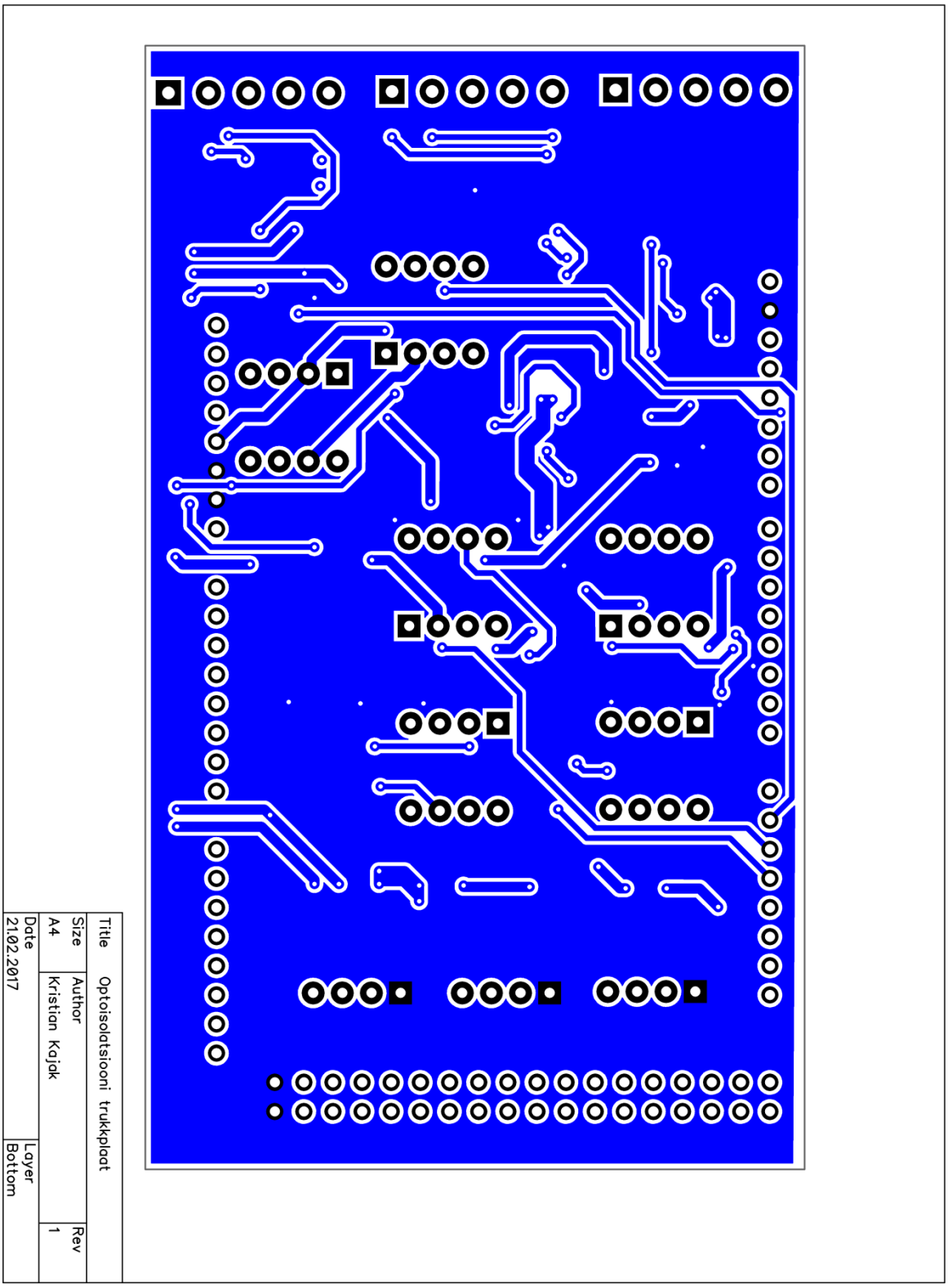
Title	CPC1832 sensori trukkplaat	
Size A4	Author Kristian Kajak	Rev 1
Date 16.02.2017		Layer Bottom

LISA 6 OPTOISOLATSIOONI TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND

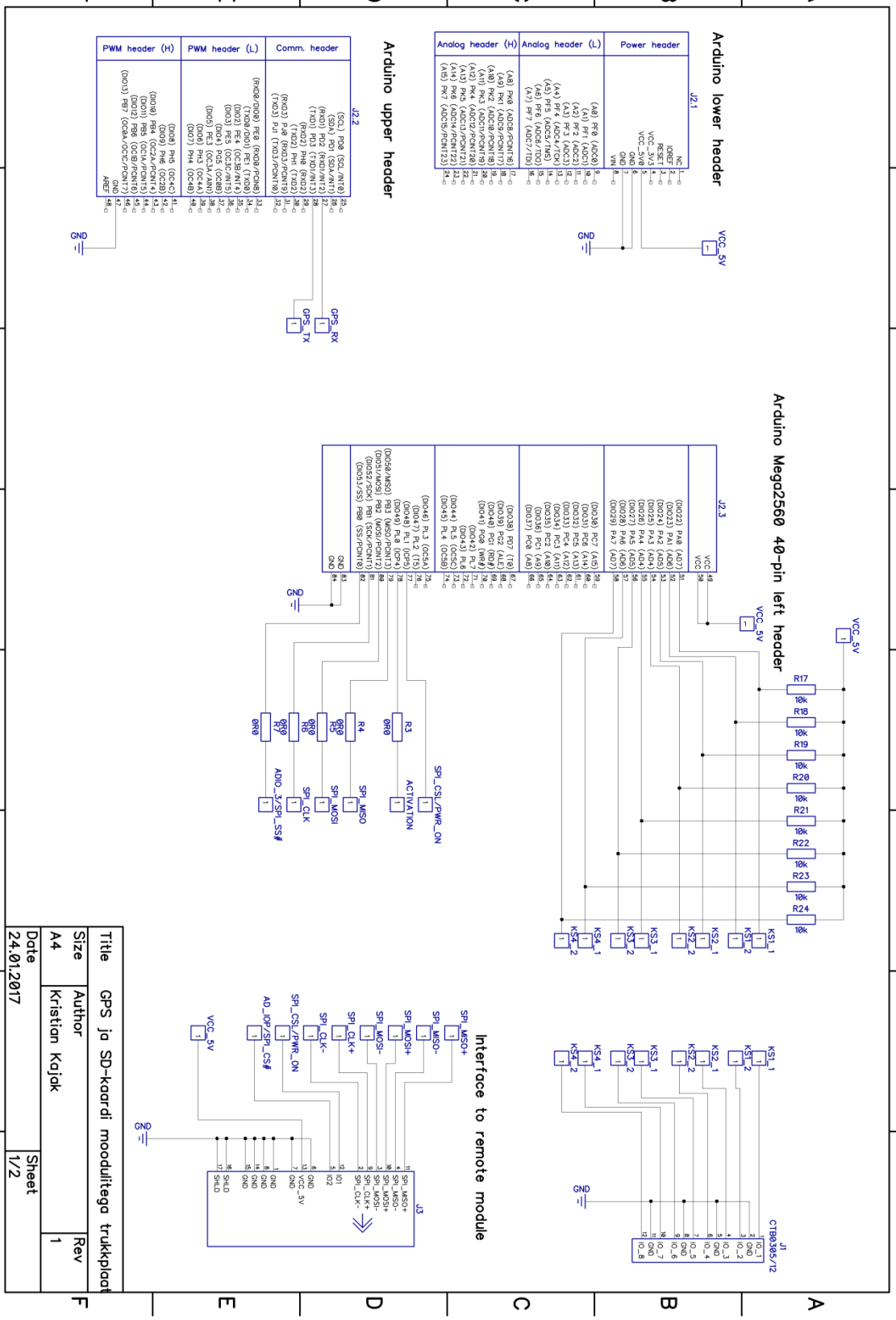


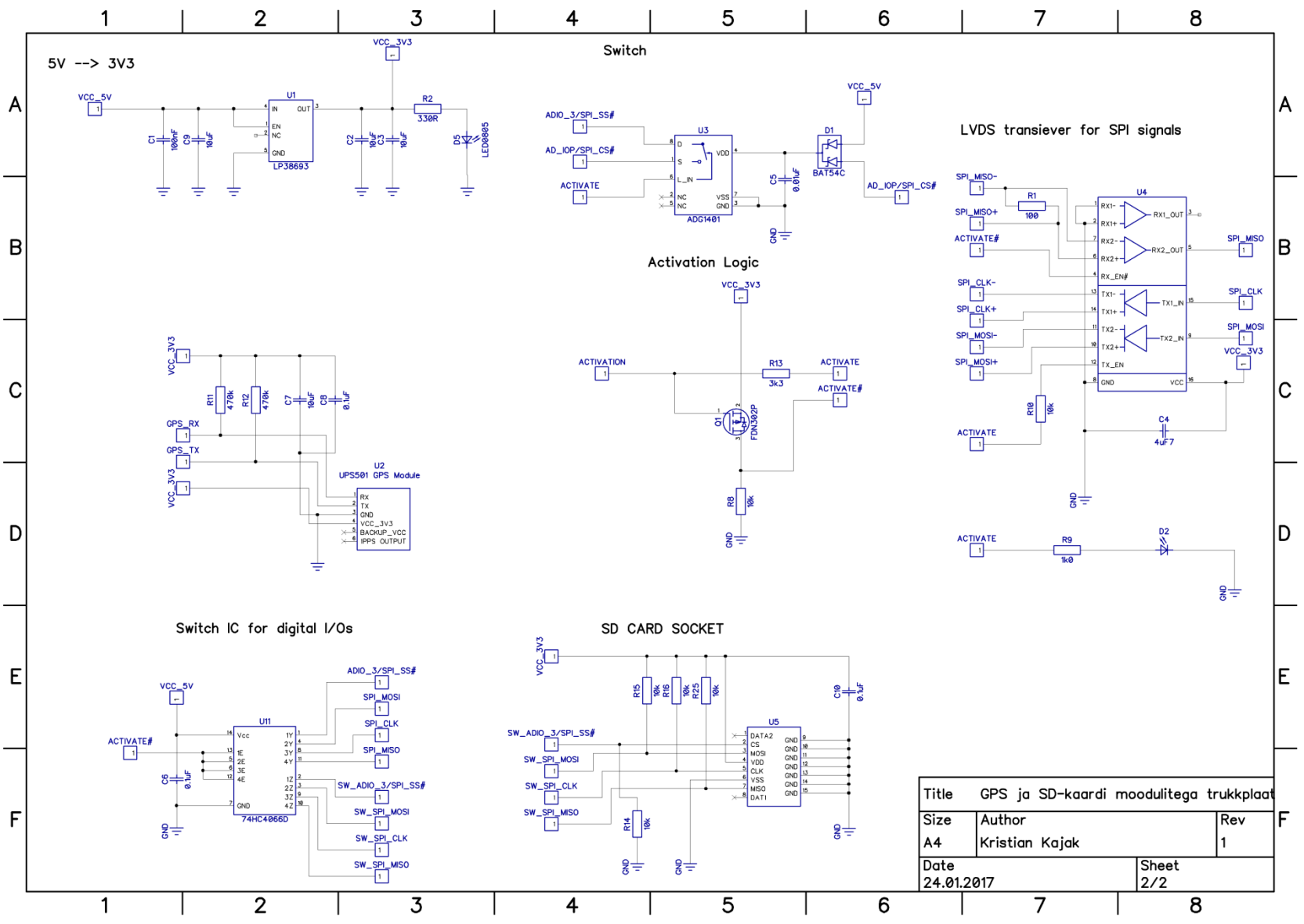

RX/TX OPTOCOUPLING
RX/TX OPTOCOUPLING
RX/TX OPTOCOUPLING
U5
FOD260L OPTOCOUPLER
U1
FOD260L OPTOCOUPLER
U4
SN65LVDM179D
U6
FOD260L OPTOCOUPLER
U2
FOD260L OPTOCOUPLER
U7
SN65LVDM179D
U8
FOD260L OPTOCOUPLER
U9
FOD260L OPTOCOUPLER
U10
SN65LVDM179D
VCC_5V
VCC_3V3_ISOL_2
VCC_3V3_ISOL_1
VCC_3V3_ISOL_3
TX2
TX1
TX3
RX2
RX1
RX3
TX2_COUPLED
TX1_COUPLED
TX3_COUPLED
RX2_COUPLED
RX1_COUPLED
RX3_COUPLED
RX1+
RX1-
TX1+
TX1-
RX2+
RX2-
TX2+
TX2-
RX3+
RX3-
TX3+
TX3-
GND
GND_ISOL_1
GND_ISOL_2
GND_ISOL_3
Title Optoisolatsiooni trukkplaat
Size A4
Author Kristian Kajak
Rev 1
Date 01.02.2017
Sheet 2/2

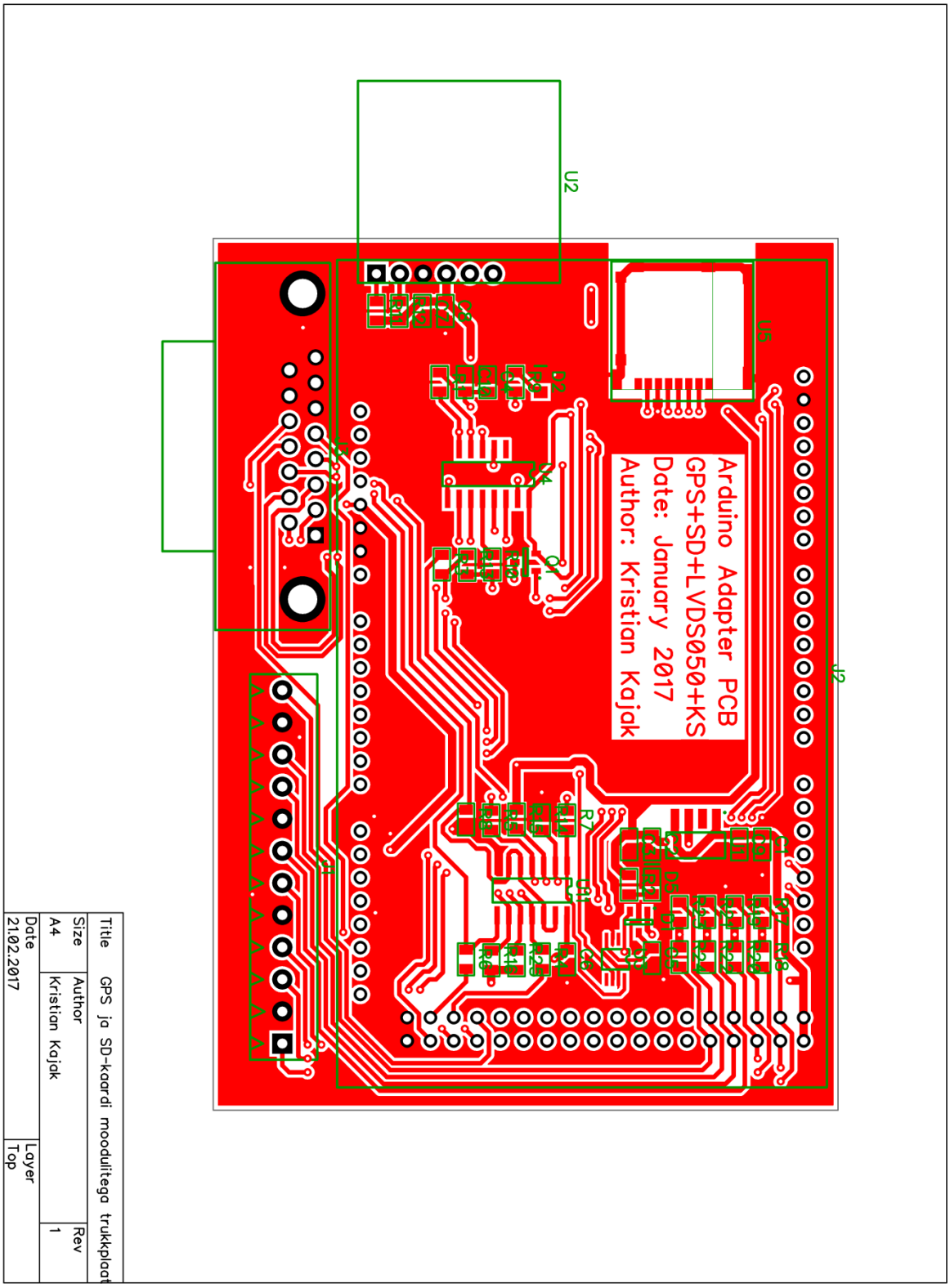

NetPort50
NetPort49
NetPort48
ARD. COMM. JANUARY 2017
J1
U1
U2
U3
U4
U5
U6
U7
U8
U9
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
R27
R28
R29
R30
Title Optoisolatsiooni trukkplaat
Size A4
Author Kristian Kajak
Rev 1
Date 21.02.2017
Layer Top

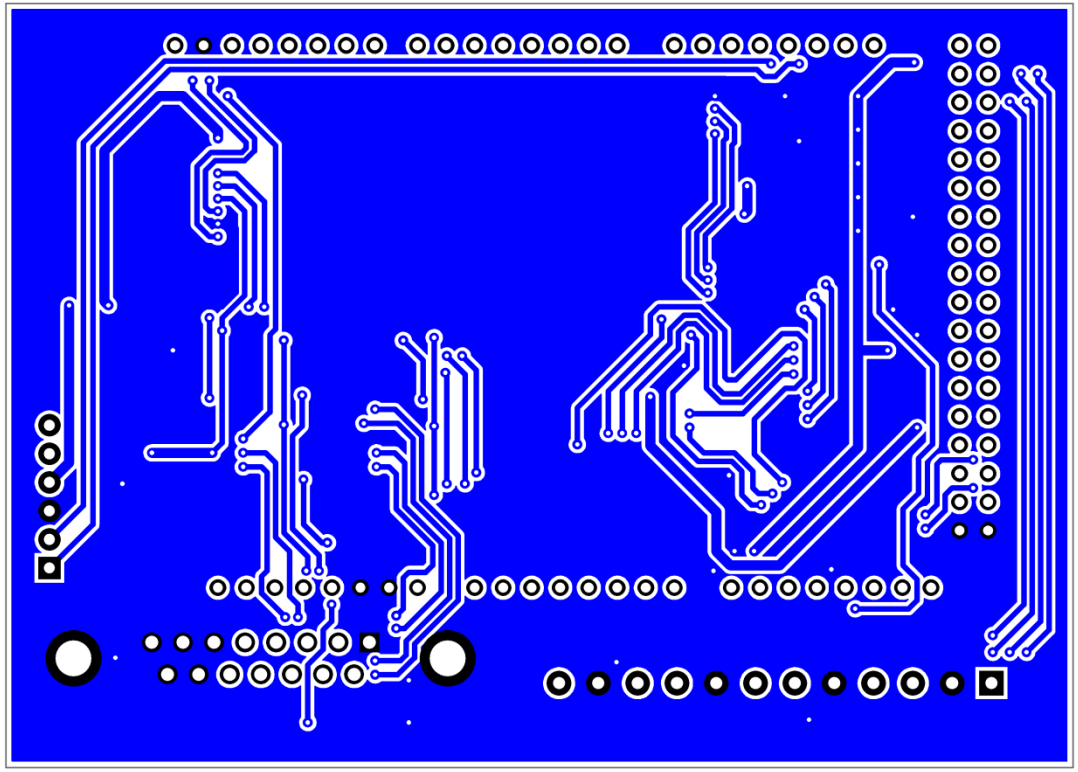

Title Optoisolatsiooni trukkplaat
Size A4
Author Kristian Kajak
Rev 1
Date 21.02.2017
Layer Bottom

LISA 7 GPS JA SD-KAARDI MOODULITEGA TRÜKKPLAADI ELEKTRISKEEM JA TOOTMISKAVAND



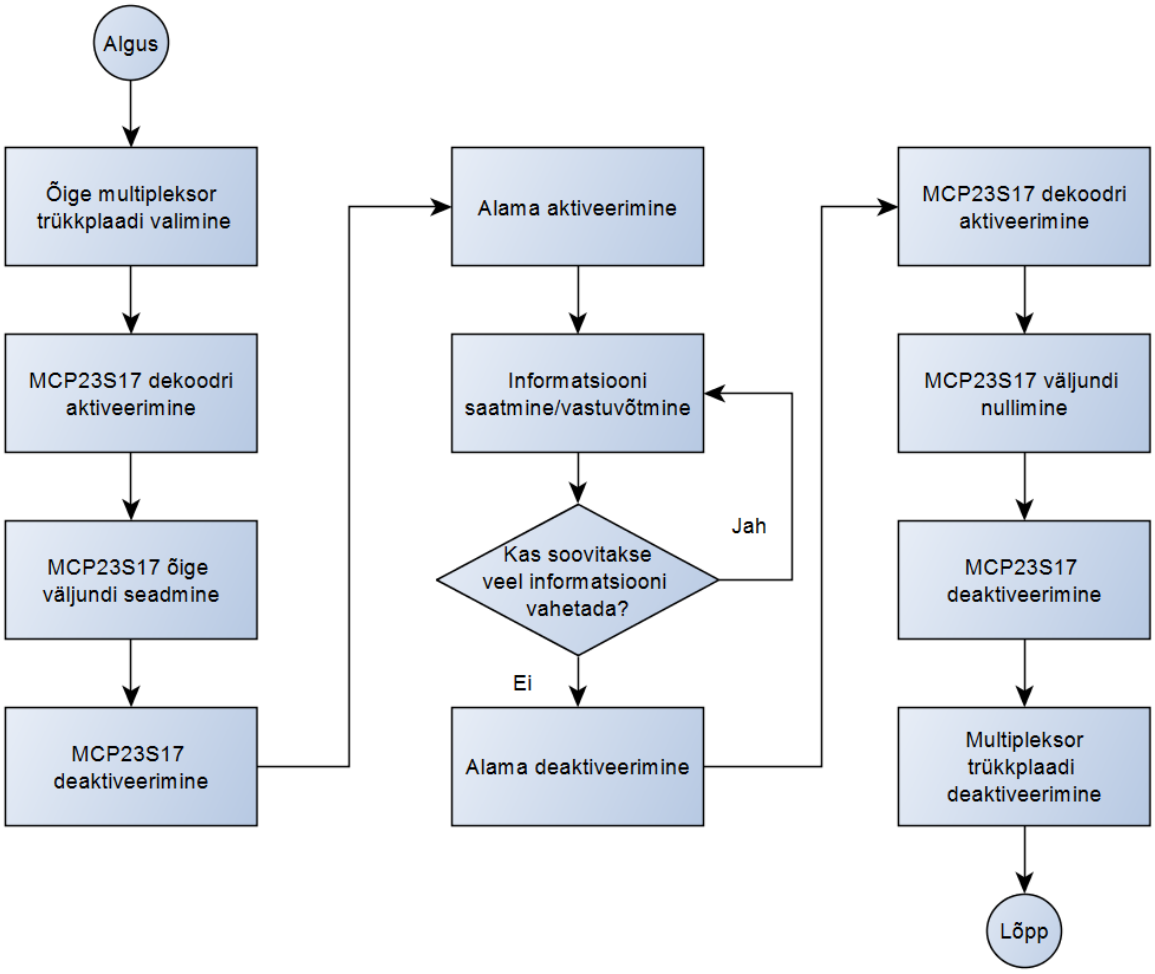

5V --> 3V3
VCC_5V
VCC_3V3
U1
LP38693
C1
100nF
C9
10uF
C2
10uF
C3
10uF
R2
330R
D5
LED0805
Switch
ADIO_3/SPI_SS#
AD_IOP/SPI_CS#
ACTIVATE
U3
ADG1401
D1
BAT54C
C5
0.01uF
AD_IOP/SPI_CS#
LVDS transiever for SPI signals
SPI_MISO-
SPI_MISO+
ACTIVATE#
R1
100
U4
RX1-
RX1+
RX1_OUT
RX2-
RX2+
RX2_OUT
RX_EN#
TX1-
TX1+
TX1_IN
TX2-
TX2+
TX2_IN
TX_EN
GND
VCC
SPI_MISO
SPI_CLK-
SPI_CLK+
SPI_CLK
SPI_MOSI-
SPI_MOSI+
SPI_MOSI
VCC_3V3
R10
10k
C4
4uF7
ACTIVATE
GND
Activation Logic
VCC_3V3
ACTIVATION
R13
3k3
ACTIVATE
ACTIVATE#
Q1
FDN302P
R8
10k
GND
R11
470k
R12
470k
C7
10uF
C8
0.1uF
GPS_RX
GPS_TX
VCC_3V3
U2
UPS501 GPS Module
RX
TX
GND
VCC_3V3
BACKUP_VCC
1PPS OUTPUT
ACTIVATE
R9
1k0
D2
GND
Switch IC for digital I/Os
VCC_5V
ACTIVATE#
U11
74HC4066D
C6
0.1uF
GND
ADIO_3/SPI_SS#
SPI_MOSI
SPI_CLK
SPI_MISO
SW_ADIO_3/SPI_SS#
SW_SPI_MOSI
SW_SPI_CLK
SW_SPI_MISO
SD CARD SOCKET
VCC_3V3
R15
10k
R16
10k
R25
10k
C10
0.1uF
U5
DATA2
CS
MOSI
VDD
CLK
VSS
MISO
DAT1
GND
SW_ADIO_3/SPI_SS#
SW_SPI_MOSI
SW_SPI_CLK
SW_SPI_MISO
R14
10k
GND
Title GPS ja SD-kaardi moodulitega trukkplaat
Size A4
Author Kristian Kajak
Rev 1
Date 24.01.2017
Sheet 2/2


U2
U5
J2
Arduino Adapter PCB
GPS+SD+LVDS050+KS
Date: January 2017
Author: Kristian Kajak
Title GPS ja SD-kaardi moodulitega trukkplaat
Size A4
Author Kristian Kajak
Rev 1
Date 21.02.2017
Layer Top



Title	GPS ja SD-kaardi moodulitega trukkplaat	
Size A4	Author Kristian Kajak	Rev 1
Date 21.02.2017	Layer Bottom	

LISA 8 ALAMA ADRESSEERIMISE ALGORITM



LISA 9 MULTIPLEKSORI TRÜKKPLAADI VALIMISE ALAMFUNKTSIOON

```
70 void selectMuxAddress(int MA2muxAddress, int MA1muxAddress, int MA0muxAddress)
71 {
72   //Enabling the sending
73   digitalWrite(M_En, HIGH); //M_EN  HIGH=ENABLED
74 
75   //Selecting the correct address of the MUX board
76   if(MA2muxAddress==1)
77     digitalWrite(MA2, LOW);  //MA1, INVERSED
78   else if(MA2muxAddress==0)
79     digitalWrite(MA2, HIGH);  //MA1, INVERSED
80   else
81     Serial.println("MA2muxAddress is invalid!");
82   if(MA1muxAddress==1)
83     digitalWrite(MA1, LOW);  //MA1, INVERSED
84   else if(MA1muxAddress==0)
85     digitalWrite(MA1, HIGH);  //MA1, INVERSED
86   else
87     Serial.println("MA1muxAddress is invalid!");
88   if(MA0muxAddress==1)
89     digitalWrite(MA0, LOW);  //MA2, INVERSED
90   else if(MA0muxAddress==0)
91     digitalWrite(MA0, HIGH);  //MA2, INVERSED
92   else
93     Serial.println("MA0muxAddress is invalid!");
94 
95   //Printing the board number
96   Serial.print("\nMUX board no. ");
97   Serial.print(MA2muxAddress);
98   Serial.print(MA1muxAddress);
99   Serial.print(MA0muxAddress);
100  Serial.println(" is selected.");
101 
102  //Disabling the sending
103  digitalWrite(M_En, LOW);  //M_EN
104 }
```

LISA 10 MCP23S17 REGISTRITE KONFIGUREERIMISE ALAMFUNKTSIOON

```
106 void setRegisterValue(String registerName, int registerValue)
107 {
108   //Setting LOC_SEL_EN# enabled
109   digitalWrite(SPI_CSL_PWR_ON, HIGH );  //By setting SPI_CSL_PWR_ON HIGH, you CS MCP23S17
110
111   //Setting the Register Address
112   int registerAddress=300; //random number for troubleshooting
113   if(registerName=="IOCONA")
114     registerAddress=0x0A;
115   if(registerName=="IOCONB")
116     registerAddress=0x0B;
117   if(registerName=="IODIRA")
118     registerAddress=0x00;
119   if(registerName=="IODIRB")
120     registerAddress=0x01;
121   if(registerName=="GPIOA")
122     registerAddress=0x12;
123   if(registerName=="GPIOB")
124     registerAddress=0x13;
125
126   //Troubleshooting
127   if(registerAddress==300)
128     Serial.println("registerName is invalid!");
129   if(registerValue >= 256)
130     Serial.println("RegisterValue is invalid!");
131
132   //Sending SPI signals to MCP23S17
133   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
134     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
135       SPI.transfer(0b01001110); // Device Opcode
136       SPI.transfer(registerAddress);       //Register address
137       SPI.transfer(registerValue); // Value
138     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
139   SPI.endTransaction();
140
141   //Setting LOC_SEL_EN# disabled
142   digitalWrite(SPI_CSL_PWR_ON, LOW  );  //By setting SPI_CSL_PWR_ON LOW, you deselect MCP23S17
143 }
```

LISA 11 MAX31855 TEMPERATUURI LUGEMISE ALAMFUNKTSIOONID

```
int readMax31855Temp(unsigned int MAX_No)
{
  unsigned long int data;
  setRegisterValue("GPIOB", 0b00011000);

  //set the correct MCP23S17 register value to select the right MAX31855 chip
  if(MAX_No==0)
    setRegisterValue("GPIOA", 0b10000000);
    data=readMax31855TempConversion(MAX_No);
  if(MAX_No==1)
    setRegisterValue("GPIOA", 0b01000000);
    data=readMax31855TempConversion(MAX_No);
  if(MAX_No==2)
    setRegisterValue("GPIOA", 0b00100000);
    data=readMax31855TempConversion(MAX_No);
  if(MAX_No==3)
    setRegisterValue("GPIOA", 0b00010000);
    data=readMax31855TempConversion(MAX_No);

  //troubleshooting
  if(MAX_No<0||MAX_No>=4)
  {
    Serial.println("Selected MAX_No is invalid!");
    return 0;
  }
  Serial.print("\nMAX31855 No ");
  Serial.print(MAX_No);
  Serial.print(" internal temperature is: ");
  Serial.println(data);

  //Deselect all chips
  setRegisterValue("GPIOA", 0b00000000);
  setRegisterValue("GPIOB", 0b00011100);
  return data;
}
```

```
int readMax31855TempConversion(unsigned int MAX_No)
{
    unsigned int data1, data2, data3, data4;
  SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
    digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
      data1=SPI.transfer(0);        //Read data from the selected MAX31855
      data2=SPI.transfer(0);
      data3=SPI.transfer(0);
      data4=SPI.transfer(0);
    digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
  SPI.endTransaction();

  //creating a 32bit data string
  unsigned long data = data1;
  data = data * 256 + data2;
  data = data * 256 + data3;
  data = data * 256 + data4;

  //Filtering out unnecessary information
  data = data>>4;
  data = data & 0b00000000000000000111111111111;
  data = data/16;
  return data;
}
```

LISA 12 MCP4912 DIGITAAL-ANALOOGMUUNDURI ALAMFUNKTSIOONID

```
206 void setOutputVoltage(unsigned int channelNumber, unsigned int outputVoltage)
207 {
208     //Select the DAC
209     setRegisterValue("GPIOA", 0b00001000);
210     setRegisterValue("GPIOB", 0b00011100);
211     int channel;
212 
213     //Choose the channel to apply the voltage to
214     if(channelNumber==0){
215       applyVoltageWriting(channel=0b0001, outputVoltage, channelNumber);
216       }
217     else if(channelNumber==1){
218       applyVoltageWriting(channel=0b1001, outputVoltage, channelNumber);
219       }
220     else{
221       Serial.println("Invalid channel number");
222     }
223 
224     //Deselect the DAC
225     setRegisterValue("GPIOA", 0b00000000);
226     delay(1);
227 }
```

```
229 void applyVoltageWriting(int channel, unsigned int outputVoltage, unsigned int channelNumber)
230 {
231     //add channel and output voltage together into 16-bit data
232     unsigned int combined, data;
233     combined = channel;
234     combined = combined<<12;
235     combined |= outputVoltage;
236 
237     //cut 16-bit data into 2 8-bit datas
238     uint8_t datalow = combined & 0xff;
239     uint8_t datahigh = (combined >> 8);
240 
241     //Send the data to DAC
242     SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
243       digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
244         SPI.transfer(datahigh);
245         SPI.transfer(datalow);
246       digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
247     SPI.endTransaction();
248 
249     Serial.print("\nDAC channel ");
250     Serial.print(channelNumber);
251     Serial.print(" selected \nOutput voltage value: ");
252     Serial.println(outputVoltage);
253 }
```

LISA 13 LTC1861 ANALOOG-DIGITAALMUUNDURI ALAMFUNKTSIOONID

```
255 unsigned int readInputVoltage(unsigned int channelNumber)
256 {
257   unsigned int data, channel;
258 
259   //Select the ADC
260   setRegisterValue("GPIOA", 0b00000100);
261   setRegisterValue("GPIOB", 0b00011100);
262 
263   //Select the channel
264   if(channelNumber==0){
265       //channel=0b10000000;
266       data=applyVoltageReading(channel=0b10000000, channelNumber);
267       return data;
268   }
269   else if(channelNumber==1){
270       //channel=0b11000000;
271       data=applyVoltageReading(channel=0b11000000, channelNumber);
272       return data;
273   }
274   else{
275       Serial.println("Invalid channel number");
276       return 0;
277   }
278 
279   //Deselect the ADC
280   setRegisterValue("GPIOA", 0b00000000);
281 }
```

```
unsigned int applyVoltageReading(unsigned int channel, unsigned int channelNumber)
{
  unsigned int data, data1, data2;

  //Wake up the ADC
  SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
    digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
    delay(1);
    digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
    delay(1);

    /Send the command to the ADC
    digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
      SPI.transfer(channel);
      SPI.transfer(0);
    digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW

    //Read from the ADC
    delay(1);
    digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
      data1=SPI.transfer(0);
      data2=SPI.transfer(0);
    digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
  SPI.endTransaction();
  delay(1);

  //Add 2 separate SPI transfers into 12bit data
  data = data1;
    data = data<<8;
    data |= data2;
    data = data>>4;
    data = data/2;
    Serial.print("\nADC channel ");
    Serial.print(channelNumber);
    Serial.print(" selected \nInput voltage value: ");
    Serial.println(data);
  return data;
}
```

LISA 14 ANALOOG-DIGITAALMUUNDURI TRÜKKPLAADI TARKVARA MULTIPLEKSOR TRÜKKPLAADI VALIMISE, MCP23S17 REGISTRITRE KONFIGUREERIMISE JA MAX31855 TEMPERATUURI LUGEMISE ALAMFUNKTSIOONID

```
77  void selectMuxAddress(int MA2muxAddress, int MA1muxAddress, int MA0muxAddress)
78  {
79    //Enabling the sending
80    digitalWrite(M_En, HIGH); //M_EN  HIGH=ENABLED
81
82    //Selecting the correct address of the MUX board
83    if(MA2muxAddress==1)
84      digitalWrite(MA2, LOW);  //MA1, INVERSED
85    else if(MA2muxAddress==0)
86      digitalWrite(MA2, HIGH);  //MA1, INVERSED
87    else
88      Serial.println("MA2muxAddress is invalid!");
89    if(MA1muxAddress==1)
90      digitalWrite(MA1, LOW);  //MA1, INVERSED
91    else if(MA1muxAddress==0)
92      digitalWrite(MA1, HIGH);  //MA1, INVERSED
93    else
94      Serial.println("MA1muxAddress is invalid!");
95    if(MA0muxAddress==1)
96      digitalWrite(MA0, LOW);  //MA2, INVERSED
97    else if(MA0muxAddress==0)
98      digitalWrite(MA0, HIGH);  //MA2, INVERSED
99    else
100     Serial.println("MA0muxAddress is invalid!");
101
102   //Printing the board number
103   Serial.print("\nMUX board no. ");
104   Serial.print(MA2muxAddress);
105   Serial.print(MA1muxAddress);
106   Serial.print(MA0muxAddress);
107   Serial.println(" is selected.");
108
109   //Disabling the sending
110   digitalWrite(M_En, LOW);  //M_EN
111 }
```

```
119 void setRegisterValueForSolarDir(String registerName, int registerValue)
120 {
121   //Setting LOC_SEL_EN# enabled
122   digitalWrite(SPI_CSL_PWR_ON, HIGH );  //By setting SPI_CSL_PWR_ON HIGH, you CS MCP23S17
123   int registerAddress=300;
124   if(registerName=="IODIR")
125     registerAddress=0x00;
126   if(registerName=="GPIO")
127     registerAddress=0x09;
128 
129   //Troubleshooting
130   if(registerAddress==300)
131     Serial.println("registerName is invalid!");
132   if(registerValue >= 256)
133     Serial.println("RegisterValue is invalid!");
134 
135   //Sending SPI signals to MCP23S17
136   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
137     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
138       SPI.transfer(0b01000000); // Device Opcode
139       SPI.transfer(registerAddress);       //Register address
140       SPI.transfer(registerValue); // Value
141     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
142   SPI.endTransaction();
143 
144   //Setting LOC_SEL_EN# disabled
145   digitalWrite(SPI_CSL_PWR_ON, LOW  );  //By setting SPI_CSL_PWR_ON LOW, you deselect MCP23S17
146 }
```

```
148 int readMax31855TempForSolarDir(unsigned int MAX_No)
149 {
150   unsigned long int data;
151 
152   //Select the correct MAX31855 chip
153   setRegisterValueForSolarDir("GPIO", 0b11111110);
154   if(MAX_No==0)
155     setRegisterValueForSolarDir("GPIO", 0b10111110);
156     data=readMax31855TempConversion(MAX_No);
157   if(MAX_No==1)
158     setRegisterValueForSolarDir("GPIO", 0b11011110);
159     data=readMax31855TempConversion(MAX_No);
160   if(MAX_No<0||MAX_No>=2)
161   {
162     Serial.println("Selected MAX_No is invalid!");
163     return 0;
164   }
165   Serial.print("MAX31855 No ");
166   Serial.print(MAX_No);
167   Serial.print(" internal temperature is: ");
168   Serial.println(data);
169 
170   //Deselect MAX31855
171   setRegisterValueForSolarDir("GPIO", 0b11111110);
172   return data;
173 }
```

```
175 int readMax31855TempConversion(unsigned int MAX_No)
176 {
177   unsigned int data1, data2, data3, data4;
178 
179   //Read the data from MAX31855
180   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
181     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
182       data1=SPI.transfer(0);
183       data2=SPI.transfer(0);
184       data3=SPI.transfer(0);
185       data4=SPI.transfer(0);
186     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
187   SPI.endTransaction();
188 
189   //creating a 32bit data string
190   unsigned long data = data1;
191   data = data * 256 + data2;
192   data = data * 256 + data3;
193   data = data * 256 + data4;
194 
195   //Filtering out unnecessary information
196   data = data>>4;
197   data = data & 0b00000000000000000011111111111;
198   data = data/16;
199   return data;
200 }
```

LISA 15 ADS1248 REGISTRITE KONFIGUREERIMISE ALAMFUNKTSIOONID

```
202 void writeADS1248RegisterForSolarDir(String registerName, unsigned int registerValue)
203 {
204   unsigned int registerAddress=300;
205   //Choose the correct register address
206   if(registerName=="MUX0")
207   {
208     registerAddress=0x40;
209     ExecuteWriteADS1248RegisterForSolarDir(registerAddress, registerValue);
210   }
211   if(registerName=="MUX1")
212   {
213     registerAddress=0x42;
214     ExecuteWriteADS1248RegisterForSolarDir(registerAddress, registerValue);
215   }
216   if(registerName=="SYS0")
217   {
218     registerAddress=0x43;
219     ExecuteWriteADS1248RegisterForSolarDir(registerAddress, registerValue);
220   }
221   if(registerName=="IDAC0")
222   {
223     registerAddress=0x4A;
224     ExecuteWriteADS1248RegisterForSolarDir(registerAddress, registerValue);
225   }
226   //Troubleshooting
227   if(registerAddress==300)
228     Serial.println("registerName is invalid!");
229   if(registerValue >= 256)
230     Serial.println("RegisterValue is invalid!");
231 }
232
233 void ExecuteWriteADS1248RegisterForSolarDir(unsigned int registerAddress, unsigned int registerValue)
234 {
235   setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
236   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE1));  //Select SPI settings
237     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
238       delay(1);
239       SPI.transfer(registerAddress);//Which register to write to
240       SPI.transfer(0x00);//Number of registers
241       SPI.transfer(registerValue);//Register value
242       delay(1);
243     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
244   SPI.endTransaction();
245   setRegisterValueForSolarDir("GPIO", 0b11111111); //Set ADS1248 CS pin high
246   delay(2);
247 }
```

LISA 16 ADS1248 REGISTRITE LUGEMISE ALAMFUNKTSIOONID

```
void readADS1248RegisterForSolarDir(String registerName)
{
  unsigned int registerAddress=300;

  //Choose the correct register address
  if(registerName=="MUX0")
  {
    registerAddress=0x20;
    Serial.print("MUX0 value is:");
    ExecuteReadADS1248RegisterForSolarDir(registerAddress);
  }
  if(registerName=="MUX1")
  {
    registerAddress=0x22;
    Serial.print("MUX1 value is:");
    ExecuteReadADS1248RegisterForSolarDir(registerAddress);
  }
  if(registerName=="SYS0")
  {
    registerAddress=0x23;
    Serial.print("SYS0 value is:");
    ExecuteReadADS1248RegisterForSolarDir(registerAddress);
  }
  if(registerName=="IDAC0")
  {
    registerAddress=0x2A;
    Serial.print("IDAC0 value is:");
    ExecuteReadADS1248RegisterForSolarDir(registerAddress);
  }

  //Troubleshooting
  if(registerAddress==300)
    Serial.println("registerName is invalid!");
}


void ExecuteReadADS1248RegisterForSolarDir(unsigned int registerAddress)
{
  unsigned int value1, value2;
  //Serial.println(registerAddress, HEX);
  setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
  SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE1));  //Select SPI settings
    delay(1);
    digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
    delay(1);
      SPI.transfer(registerAddress);//register name
      SPI.transfer(0x00);//Number of registers
      digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
      SPI.endTransaction();
      SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE1));  //Select SPI settings
      digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
      value1 = SPI.transfer(0xFF);//Register value
    digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
    delay(1);
  SPI.endTransaction();
  setRegisterValueForSolarDir("GPIO", 0b11111111); //Set ADS1248 CS pin high
  Serial.println(value1, HEX);
  delay(2);
}
```

LISA 17 ADS1248 SISEMISE TEMPERATUURI LUGEMISE ALAMFUNKTSIOON

```
340 float readADS1248InternalTempForSolarDir()
341 {
342   setADS1248RegisterValueForSolarDir(0x37, 0x73, 0x08, 0x08);
343 
344   //RDATA
345   unsigned int data1, data2, data3;
346   startADS1248Conversion();
347   setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
348   digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
349   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE1));  //Select SPI settings
350     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
351       delay(1);
352       SPI.transfer(0x12);//RDATA
353       data1=SPI.transfer(0xFF);
354       data2=SPI.transfer(0xFF);
355       data3=SPI.transfer(0xFF);
356       delay(1);
357     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
358   SPI.endTransaction();
359   setRegisterValueForSolarDir("GPIO", 0b11111111); //Set ADS1248 CS pin high
360   Serial.print("data:");
361    Serial.println(data1,HEX);
362    Serial.println(data2,HEX);
363    Serial.println(data3,HEX);
364   delay(2);
365 
366   float tempChange, temp;
367   float koefData = 0.000603;   //1*LSB/405mcV*;  LSB=2.048 * (1/2^23)
368   long data_0 = 483328;  //data in the case on 0.118mV
369 
370   //Adding all the data bytes together
371   long data = data1;
372   data = data * 256 + data2;
373   data = data * 256 + data3;
374 
375   tempChange = (data - data_0);
376   long proov = data - data_0;
377   tempChange = tempChange*koefData;
378   temp = 25 + tempChange;
379 
380   Serial.print("ADS1248 internal temperature is: ");
381   Serial.println(temp,1);
382   return temp;
383 }
```

LISA 18 ADS1248 SISENDPINGE LUGEMISE ALAMFUNKTSIOON

```
385 float readADS1248InputVoltageForSolarDir()
386 {
387   setADS1248RegisterValueForSolarDir(0x37, 0x70, 0x08, 0x08);
388 
389   //RDATA
390   unsigned int data1, data2, data3;
391   long inputVoltage;
392   startADS1248Conversion();
393   setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
394   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE1));  //Select SPI settings
395     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
396       delay(1);
397       SPI.transfer(0x12);//RDATA
398       data1=SPI.transfer(0xFF);
399       data2=SPI.transfer(0xFF);
400       data3=SPI.transfer(0xFF);
401       delay(1);
402     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
403   SPI.endTransaction();
404   setRegisterValueForSolarDir("GPIO", 0b11111111); //Set ADS1248 CS pin high
405   delay(2);
406 
407   //Adding all the data bytes together
408   unsigned long data = data1;
409   data = data * 256 + data2;
410   data = data * 256 + data3;
411   float LSBvalue = 1000*2.048/8388608;
412   inputVoltage = data*LSBvalue;
413 
414   Serial.print("ADS1248input voltage(mV) is: ");
415   Serial.println(inputVoltage);
416   return inputVoltage;
417 }
```

LISA 19 MUUD ADS1248 ALAMFUNKTSIOONID

```
427 void resetADS1248()
428 {
429   setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
430   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
431     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
432       delay(1);
433       SPI.transfer(0x06);
434       delay(1);
435     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
436   SPI.endTransaction();
437   setRegisterValueForSolarDir("GPIO", 0b11111111); //Set ADS1248 CS pin high
438   delay(20);
439 }
440
441 void startADS1248Conversion()
442 {
443   setRegisterValueForSolarDir("GPIO", 0b11111111);
444   delay(2);
445 }
446
447 void wakeUpADS1248()
448 {
449   setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
450   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
451     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
452       delay(1);
453       SPI.transfer(0x00);
454       delay(1);
455     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
456   SPI.endTransaction();
457   setRegisterValueForSolarDir("GPIO", 0b11111111); //Set ADS1248 CS pin high
458   delay(20);
459 }
460
461 void stopADS1248()
462 {
463   setRegisterValueForSolarDir("GPIO", 0b01111111); //Set ADS1248 CS pin low
464   SPI.beginTransaction(SPISettings(160000, MSBFIRST, SPI_MODE0));  //Select SPI settings
465     digitalWrite(slaveSelect, LOW);  //ADIO_3/SPI_SS# Slave is selected when LOW
466       delay(1);
467       SPI.transfer(0x02);
468       delay(1);
469     digitalWrite(slaveSelect, HIGH);  //ADIO_3/SPI_SS# Slave is selected when LOW
470   SPI.endTransaction();
471   setRegisterValueForSolarDir("GPIO", 0b11111110); //Set ADS1248 CS pin high
472 }
```

LISA 20 ELEKTROONIKAKESTA SOLIDWORKS MUDEL