Tallinna Ülikool

Digitehnoloogiate instituut

RASPBERRY COMPUTE MODULE-ILE EMAPLAADI LOOMINE

Seminaritöö

Autor: Joosep Jõeleht

Juhendaja: Jaagup Kippar

Tallinn 2016

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

(kuupäev) (autor)

Sisukord

Sissejuhatus

Käesoleva töö eesmärgiks on anda ülevaade sellest, kuidas saab luua Raspberry Compute Module'ile (edaspidi RPCM) emaplaati, mille põhiülesanne on suhelda IT5888 liidesplaadiga ja kasutada antud plaadilt saadud infot kontrollimaks automehhaanikat, mis on liidestuskaardiga ühendatud. Sekundaarse funktsioonina on tarvis, et plaadil oleks võime suhelda välismaailmaga läbi Etherneti pesa ja samuti pakkuda võimalust lisada välisseadmeid kahe A-tüüpi Universal Serial Bus (edaspidi USB) pesa kaudu.

Antud töö ei ole käsitletav kui juhend, kuidas sellist seadet luua, vaid pigem ülevaade enda läbitud tööprotsessist, mille raames emaplaat valmis. Töö on suunatud neile, kes soovivad alustada oma emaplaadi loomisega, kuid pole kindel, kuidas alustada. Samuti on välja toodud minu omapoolne hinnang sellest, mida peaks järgnevatel kordadel teisiti tegema ning millega peaks arvestama, kui hakata sellise projektiga tegelema. Seminaritöö käigus loon ka näidisplaadi, millel näitan natukene täpsemalt, kuidas toimub KiCadis plaadi loomise protsess.

Nagu eelnevalt mainitud, olen valinud emaplaadi loomise baasiks RPCM'i, millest tulenevalt ka seminaritöö esimene osa võtab lühidalt kokku sihtasutuse Raspberry Foundation ajaloo ning räägib erinevatest toodetest, mida nad pakuvad.

Teise osana tööst seletan kust ja kuidas ma kogusin informatsiooni selleks, et oleks võimalik hakata looma emaplaati. Toon välja ka lühitutvustused kõikide suuremate komponentide kohta, mida antud töö raames kasutan.

Kolmandaks seletan plaadi komponentide vahelise skemaatika loomise protsessi ning plaadi materiaalse kuju loomist. Teises ja kolmandas peatükis tutvustan kõige tähtsamaid põhimõtteid, mida protsessi käigus jälgida.

RPCM'i kasutamine IO-mooduli IT5888 juhtseadmena kõrvaldaks kõik jõudlusega seotud piirangud väga pikaks ajaks nii mälu kui protsessori kiiruse osas. Sobiva emaplaadi projekteerimisel saab arvesse võtta vajaliku mõõtu, pesade ja signaallampide asukohti. RPCM'i enda *general-purpose input/output* (edaspidi GPIO) kanaleid

kasutatakse peamiselt vaid valgusdioodide juhtimiseks, muu suhtlus võib käia üle *universal asynchronous receiver/transmitter*-i (edaspidi UART), USB või kohtvõrgu. Pääsemaks juhtseadmele ligi ka võrguühenduse puudumisel, on vajalik ka konsoolipordi (UART) olemasolu. (Takis, 2017)

1 Raspberry Pi

Aastal 2006 hakkas Eben Upton töötama Raspberry Pi esimese variandi kallal, olles märganud langust Cambridge Ülikooli sisseastumiskandidaatide arvutialastes oskustes. 2008. aastaks oli tiimiga liitunud David Braben, Jack Lang, Pete Lomas ja Cambridge Ülikooli arvutiteaduskonnast Alan Mycroft ning Rob Mullins. Sellel aastal loodi sama tiimi poolt Raspberry Pi sihtasutus, mille eesmärgiks on toota madala hinna ja suure jõudlusega arvuteid, mida inimesed saaksid kasutada õppimiseks, probleemide lahendamiseks ja lõbutsemiseks. (The Raspberry Foundation, 2014)

Algselt loodeti, et lapsed õpivad kodeerimist vabavaraga Scratch koos Raspberry Pi GPIO võimalustega, kontrollimaks väliseid vahendeid. Lisaks leiti, et tänu madalale voolutarbimisele on võimalik plaati kasutada patareidel töötavate robotite loomiseks, samuti igapäevaselt meediakompuuterina.

1.1 Raspberry Pi 1. generatsioon

Iga Raspberry Pi Model A / A+ või Model B / B+ (Illustratsioon 1-1) südamikuks on Broadcomi BCM2835 mikrokontroller, mis sisaldab endas ARM11 protsessorit 700MHz juures ja omab ka Videocore 4 graafikaprotsessorit, mis kasutab OpenGL ES2.0 ja OpenVG teeke, et saavutada piisavalt võimsust kuvamaks Blu-ray videoid kvaliteedil 30 kaadrit sekundis koos heliga üle HDMI. Kõikidel esimese generatsiooni Raspberry Pi-del on 512 mb põhimälu (The Raspberry Foundation, kuupäev puudub).

1.2 Raspberry Pi 2. generatsioon

Aastal 2015 asendati vana Pi Model B+ uue variandiga. Vanaga võrreldes on uuel versioonil vahetunud protsessor ning eelneva 512MB mälu asemel on nüüd 1GB r*andom-access memory*'t (edaspidi RAM). Uueks protsessoriks sai ARM Cortex-A7, mille taktisagedus on 900MHz, mis on võrreldes eelneva variandiga 28% tõus. (The Raspberry Foundation, kuupäev puudub)

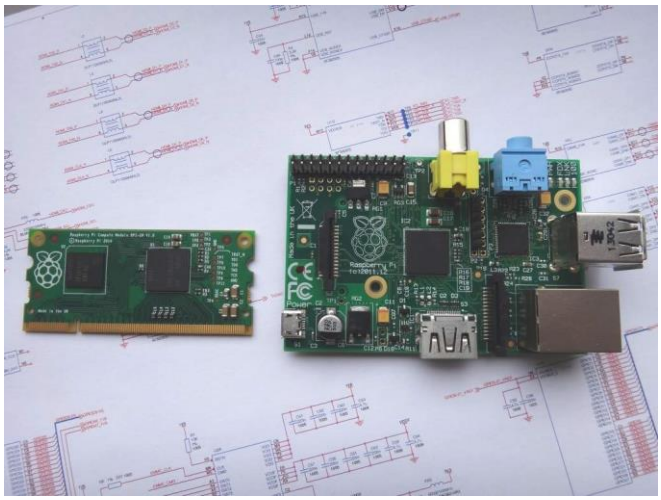
1.3 Raspberry Pi 3. generatsioon

Hetkel on Raspberry sihtasutuse viimaseks tooteks Pi3, mis lisaks protsessori uuendusele on saanud ka WiFi võimekuse ja samuti Bluetooth'i toe. (The Raspberry Foundation, kuupäev puudub) Praegu on plaadi hind Eestis umbkaudu 39€.

1.4 Raspberry Compute Module

RPCM (Illustratsioon 1-1) on erinevalt kõikidest teistest Raspberry sihtasutuse mikrokompuuteritest mõeldud kasutamiseks tööstuslikul tasandil. See on ehitatud vastavalt *double data rate small outline dual in-line random access memory module* (edaspidi SODIMM) sokli spetsifikatsioonidele. Oma olemuselt on see sarnane esimese generatsiooni Pi'le ehk selles on BCM2835 protsessor ja 512 MB RAM-i, kuid erinevalt kõikidest teistest Raspberry sihtasutuse kontrolleritest on sellel plaadil ka sisseehitatud püsimälu, mis tähendab, et selle kasutamiseks ei pea ostma eraldi välkmälukaarti.

Antud plaadil on eelnevalt ühendatud vaid protsessor ja mälukiip, seega jääb RPCM'i emaplaadi loojale kasutada kuni 45 GPIO-d ja võimalus lisada *High-Definition Multimedia Interface* ning kaamerate pesasid. Samuti on RPCM-il olemas ka USB ühenduse võimalus ning juhul, kui emaplaadil on olemas USB jaotur, siis on võimalik ka plaadile panna Etherneti pistikupesa ja mitu USB ühendust. (The Raspberry Foundation, kuupäev puudub)



Illustratsioon 1-1 Vasakul RPCM paremal Pi 1

2 Tööks vajaliku informatsiooni kogumine

Kujundasin esmase plaani (peatükk 2.2), mille baasil koostasin esialgsed skemaatikad. Töökäigus tuli plaanis ette muudatusi. Samuti valisin välja emaplaadi komponendid (peatükk 2.3). Allpool annan ka ülevaate tähtsamatest spetsifikatsioonidest, mida järgida.

2.1 Emaplaadi vajadus

Antud emaplaat sai loodud, kuna tellijal oli vaja välja vahetada vananenud Olinuxino iMX233 Max poolt sätestatud 64 MB-i suurust mälupiirangut ja protsessori kiirust. Plaadilt puudus sisseehitatud reaalajakell, mis oli eraldi liidesena korpusele teibiga külge liimitud, kuid tol ajal sai Olilinuxino siiski valituks, kuna see mahtus juba kasutusel olevasse korpusesse, lisaks olid mugavalt paigutatud USB ja Ethernet pesad. Tuleviku peale mõeldes sai antud probleemi lahenduseks RPCM'i baasil loodud emaplaat. (Takis, 2017)

2.2 Esmane plaan

Esimese etapina uurisin, kui kaugele oli firmasiseselt jõutud emaplaadi arendusega, ning sain teada, et eelnevalt on arendustegevusega alustatud, kasutades Eagle PCB Design'i. See jäi seisma, kuna antud programmi tasuta versioonis kehtestatud piirangud ei võimaldanud luua piisava võimekusega plaati. Firmalt saadud arendustöö alastest failidest leidsin USB ja Etherneti asukohad.

Pärast e-kirjade vahetust tekkis mulle nimekiri komponentidest, mis on vajalikud emaplaadi tööks. Antud infot arvestades toimib plaat järgnevalt:

- läbi neljaharulise lintjuhtme toimub suhtlus liideplaadiga ning sellelt saadakse toimimiseks vajaliku voolu
- liidesplaadilt tulev vool muundatakse 1.8, 3.3 ja 5 voldisteks pingeteks, mis on toiteallikaks kõikidele komponentidele plaadil
- emaplaadil on RPCM-i majutamiseks SODIMM pesa
- läbi USB jaoturi toimub RPCM-i suhtlus USB ja Etherneti portidega

2.3 Komponendid

Pärast esialgse idee kirjapanemist hakkasin juhendeid läbi töötlema ning järgnevalt on välja toodud kõikide suuremate komponentide lühitutvustused ning murekohad, mida peaks jälgima.

2.3.1 Sokkel

Esimene ese, mis sai välja valitud, oli RPCM-i ühenduspesa. Pesa valikul tuli jälgida mitut aspekti, millest ma ei olnud esimest versiooni välja töötades teadlik. Järgnevas tabelis (Tabel 2-1) on välja toodud põhilised erinevused DDR1 ja DDR2 SODIMM pesade vahel, mille järgi on kõige lihtsam valida enda vajadustele vastavat pesa.

	DDR1	DDR2
Elektriline erinevus (kasutatav pinge)	2,5V	1,8V
Visuaalne erinevus (sisselõige)	1,5 cm vasakult poolelt	1,6 cm vasakult poolelt

Tabel 2-1 DDR1 ja DDR2 erinevused

Kõigepealt tellisin DDR1 standardile vastava pesa, mis osutus sobimatuks minu projekti jaoks, kuna DDR1 standardil on sisselõige 1,5 cm vasakust äärest, mis ei kattu DDR2 standardiga. Eksimus siiski ei muutnud tervet esimest prototüüpi kasutuskõlbmatuks, nimelt oli võimalik laiendada RPCM-i sisselõiget. Antud juhul ei mänginud elektriline erinevus suurt rolli, kuna RPCM-i ühendusjalgade elektrilised asetused ei ole otseselt seotud DDR1 ega DDR2 standardiga. Igal jalal on oma funktsioon (Lisa 5).

2.3.2 Topelt USB

Emaplaadi tellijal oli nõudmine, et ta soovib plaadile kahte USB pesa. Kõige optimaalsem lahendus selleks oli kahekordne ehk *stacked* USB. Valisin emaplaadile TE Connectivity toote koodiga 5787617-1 (Illustratsioon 2-1). USB

Illustratsioon 2-1 Kahekordne USB

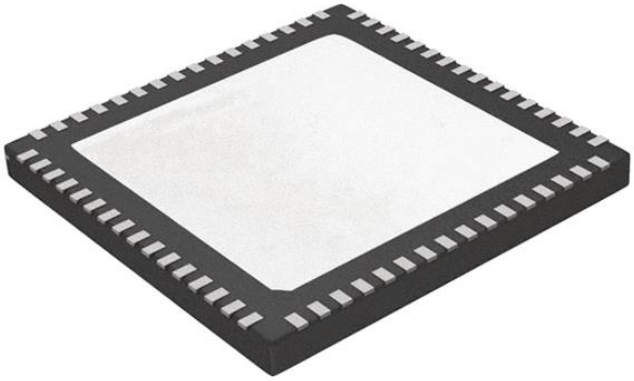
emaplaadile implementeerimisel on kaks aspekti mida tuli järgida: esiteks peavad olema USB informatsiooni rajad kuni 1% täpsusega samapikkused, teiseks peavad need olema 90 oomise lainetakistusega, mille saab välja arvutada KiCad-i sisseehitatud kalkulaatoriga.

2.3.3 Ethernet port

Emaplaadi ja välismaailma vahelise ühenduse tekitamiseks kasutasin Amphenol Commercial Products'i Etherneti konnektorit RJHSE-5384. Sellel on 12 plaati läbivat ühendusjalga, millest nelja kasutatakse kahe konnektoril oleva valgusdioodi kontrollimiseks.

2.3.4 LAN9512

LAN9512 on üks kõige tähtsamatest ja keerulisematest tükkidest plaadil. Ilma selleta ei ole võimalik panna plaadile kahekorruselist USB pistikut ega Ethernet konnektorit. Antud tükk koosneb 65 jalast (Illustratsioon 2-2), millest kaks on ülesvoolu ehk RPCM-i poole ühenduse tekitamiseks, neli on allavoolu USB-ga ja neli Etherneti pordiga ühendamiseks.



Illustratsioon 2-2 LAN9512 kiip

2.3.5 Järjestikliides

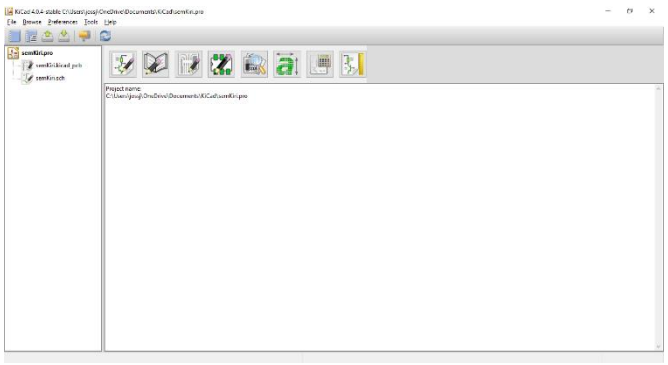
Pärast operatsioonisüsteemi installeerimist toimub kohapealne konfigureerimine üle järjestikliidese. Järjestikliidese loomiseks on kasutusel nelja ühendusjalaga UART liides.

3 Emaplaadi loomine

Järgnevas peatükis seletan, milliseid vahendeid kasutasin emaplaadi loomise protsessis, kuidas seadistasin KiCad-i vastavalt trükikoja nõuetele ning kuidas lõin ja ühendasin kõik komponendid emaplaadil.

3.1 KiCad

KiCad (Illustratsioon 3-1) on avatud lähtekoodiga tarkvara, mida kasutatakse emaplaatide loomisel, skemaatika ja trükiplaadi asetuse ning gerber ehk trükifailide genereerimisel, samuti on see ka programm, millega disainisin antud seminaritöö raames emaplaadi. Kicad sai valikuks CadSoft'i programmi Eagle asemel, kuna see on tasuta ja lubab teha kuni 16 kihilisi emaplaate, kuigi üle nelja kihi plaadil ei kasutata. KiCadi tarkvara komplekt koosneb 6 programmist, millest põhiliselt kasutame kahte: Eeschema ja Pcbnew.

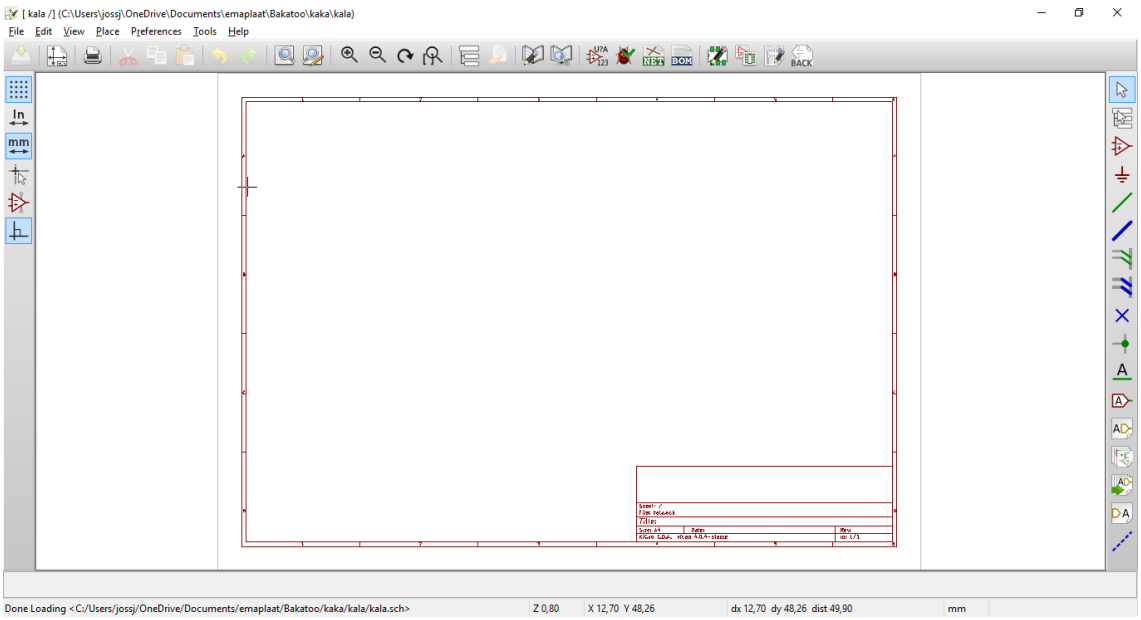


Illustratsioon 3-1 KiCad kuvatõmmis

3.2 Eeschema

Eeschema (Illustratsioon 3-2) on programm, millega saab koostada elektriskeeme, see on osa KiCadist ning on saadaval Linuxi, OS X ja Windowsi operatsioonisüsteemidel. Eeschema on integreeritud rakendus, milles joonistamine, kontrollimine, asetamine ja teekide haldamine toimub kõik ühes programmis. Eeschema on mõeldud kasutamiseks koos PcbNew programmiga, mis on KiCadi trükiplaadi kujundamise tarkvara. Eeschemal on võimeline eksportima faile, mis sisaldavad iga komponendi kohta käivat elektrilist informatsiooni. Lisaks võimaldab see kontrollida elektriliste reeglite

järgimist, eksportida prindifaile mitmes erinevas formaadis ning genereerida materjaliandmiku. (The KiCad Team, 2015) Eeschemaga tuleb installeerides kaasa üle 2800 erineva komponendi elektriskeemi, kuid erinevate trükiplaadi komponentide arvu arvestades ei ole see piisav. Seega tuli minulgi ise luua mõned elektriskeemid, näiteks sokkel, mille külge kinnitub RPCM, ja jaotur, mis tegeleb USB ja Etherneti informatsiooni edastamisega.



Illustratsioon 3-2 Eeschema kuvatõmmis

3.2.1 Näidisplaat I

Algusest peale oli plaanis lisaks põhiplaadile luua ka üks näidisplaat, et näidata, kuidas toimivad erinevad KiCadi programmid. Lõplik plaat nähtav Lisa 1.

Esimese asjana valisin vasakul olevast küljemenüüst millimeetermõõdustiku. Järgnevana tutvusin korra parempoolse küljemenüüga. Ülevalt kolmas ja neljas nupp on kaks kõige tähtsamat nuppu antud programmis, ülemine nupp on komponentide nagu näiteks valgusdioodid, takistid, kondensaatorite ja muu sellise lisamiseks; alumine on kõikide vooluvõrguga seonduva lisamiseks. Ühe valimisel tuleb vajutada suvalise koha peale valgel taustal ning siis tekib ekraanile menüü, kust saab valida vajalikke komponente. Kasutades otsimisfunktsiooni, leidsin enda plaadile kaheksa komponenti: nupp, pingemuundur, kaks 2.54mm konnektorit, vagusdiood, takisti, maandus,

kondensaator. Viimase etapina valisin antud programmis parempoolsest menüüst ülevalt neljanda nupu millega, saab luua komponentide vahelisi ühendusi, ning ühendasin kõik vajalikud jalad. Ülemisest menüüst paremalt kolmandat nuppu valides avaneb CvPcb aken, millest räägin allpool.

3.2.2 Esimene kavand

Otsustasin sokli ja jaoturi luua sama asetusega, kui nad päriselus välja näevad, kasutades selleks iga komponendi juhendit, kus on kirjas iga jala tegevus. Olin teadmatuses faktist, et skeemil olev kujutis ei mõjuta komponendi väljanägemist plaadil, seega oleks targem olnud jagada komponent mitmeks eraldi osaks, näiteks voolu, USB, Ethernetiga ning kõige muuga seonduv. Niiviisi toimides oleks tulemuseks olnud lihtsamini loetav skeem. Järgmiseks tegelesin tellija nõudmistes välja toodud kahekordse USB-ga, mille jaoks lõin samuti uue elektriskeemi. Mõne tunniga oli olemas esimene variant, kus olid ühendatud omavahel eelnevalt mainitud komponendid, lisatud oli ka vooluvõrku kontrolliv osa. Lisa 3 on välja toodud 3 põhilist komponenti ja nende ühendused.

3.2.3 Teine kavand

Pärast esimese kavandi ettenäitamist tellijale jõudsime järeldusele, et plaadile tuleks lisada veel püsimälu (tootekoodiga 93C66A-I/SN) jaoturi jaoks ja emaplaadi töökindluse tõstmiseks lisada patarei ning reaalajakell, kuid selle tulemusel avastasin, et tuleb ümber ehitada voolumuunduri ülesehitus, kuna plaadi kogutarbimine ületas eelneva voolumuunduri maksimaalse väljundi võimsuse. Lisatud osad nähtavad Lisa 4.

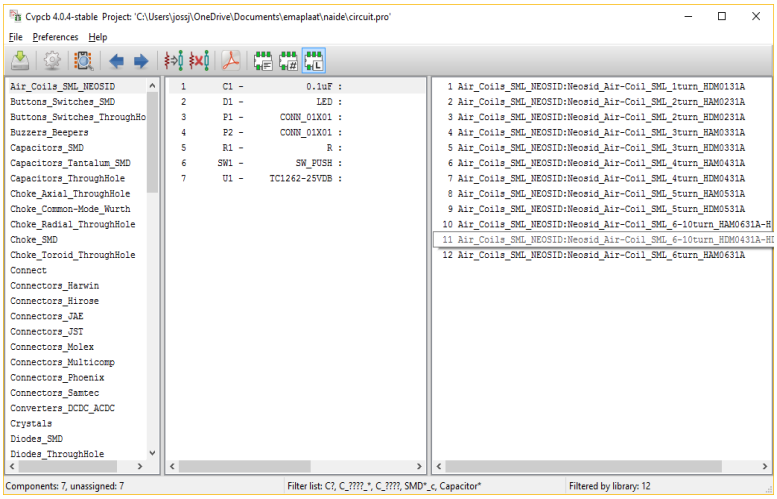
3.3 CvPcb

CvPcb on tööriist, mis aitab määrata komponentidele skeemil vastavad *footprint*-id, mida kasutatakse trükiplaadi laotuse loomisel. CvPcb annab arendajatele mugava meetodi, kuidas määrata komponentidele neile vastav *footprint*. CvPcb-le on sisse ehitatud ka võimalus sorteerida *footprint*'e erinevate näitajate järgi, samuti võimaldab see vaadelda kõiki *footprint*e 2D formaadis ning mõningaid ka 3D formaadis. (The KiCad Team, 2015)

Cvpcb kasutamine on teine etapp KiCadiga emaplaadi loomise protsessis. Selleks etapiks on tarvis täpselt välja uurida, mis korpusega on iga komponent, lisaks teada, kas vastav *footprint* on juba olemas, näiteks kõik takistite standardsuurused on koheselt programmis kaasatud. Kui mõnel komponendil puudub *footprint*, tuleb see ise luua Footprint Editori kasutades. Footprint Editori on kerge kasutada, kuid isiklikust kogemusest oskan öelda, et täpsete *footprint*'ide loomiseks peab oskama *datasheet*'e hästi lugeda. Esimest prototüüpi luues tegin vea - arvasin, et kahekordse USB *datasheet*'il tähendab sile külg komponendi äärt ja sakiline külg teist äärt. Tegelikkuses olid ääred vahetuses ja sellest tulenevalt oli probleeme prototüübile USB paigutamisega.

3.3.1 Näidisplaat II

Pärast nupu „CvPcb" vajutamist tuleb ette aken, kus tuleb vajutada nuppu „Annotate", seejärel „OK". Järgmiseks avaneb CvPcb aken (Illustratsioon 3-3), kus vasakul ääres on teekide nimistu, keskel skeemil kasutatavad komponendid ning paremal *footprint*'ide nimekiri. Teekide listist saab kergesti leida vajaminema ning seejärel igale komponendile vastav *footprint* lisada.



Illustratsioon 3-3 CvPcb kuvatõmmis

3.4 Pcbnew

Pcbnew on võimas trükiplaadi loomise tööriist, mida kasutatakse koos skeemide loomise tööriistaga Eeschema, samuti on Pcbnew saadaval nii Windowsi, OS X kui ka Linuxi platvormidel. Pcbnew põhiülesanne on hallata *footprint*'ide teeke. Iga *footprint* on joonis füüsilise komponendi jalgade, puuraukude ja piiretega. Pcbnew pakub ka tööriista disaini kontrollimiseks, mis hoiab ära siinide ja komponentide jalgade kokkusattumist kohtades, kus seda ei tohiks juhtuda, samuti ei luba see kokku erineva võrgustiku siine. (The KiCad Team, 2014)

3.4.1 Näidisplaat III

Pärast CvPcb-s kõikidele komponentidele *footprint*'ide lisamist ning seejärel Eeschemast võrgustiku faili eksportimist, mis on võimalik ülemises menüüs nupule „NET" klõpsates, tuleb avada Pcbnew, kus esiteks saab importida sisse võrgustiku faili, mille tulemusena asetatakse üksteise otsa kõik plaadil olevate komponentide *footprint*'id. Antud näitel ei ole seadistatud eraldi piiranguid siinidele, vaid jaotasin kõik komponendid üksteise kõrvale kahte ritta. Järgmiseks tuleb valida „Edge.Cuts" kiht millele lõin ümber komponentide plaadi välisääre, kasutades tööriista „Add graphic line". Lõpetuseks tuleb valida „Add tracks" tööriist, millega ühendatakse kõik komponendid omavahel ära. Pärast kõikide komponentide jalgade ühendamist on plaat tootmisesse saatmiseks valmis. Plaadi asetus on nähtaval Lisa 1 oleval pildil ning samuti on Pcbnewsse sisseehitatud võimalus vaadata loodud plaadi 3D mudelit, mida saab avada klahvikombinatsiooniga „alt + 3" (Lisa 2).

3.4.2 Seadistamine

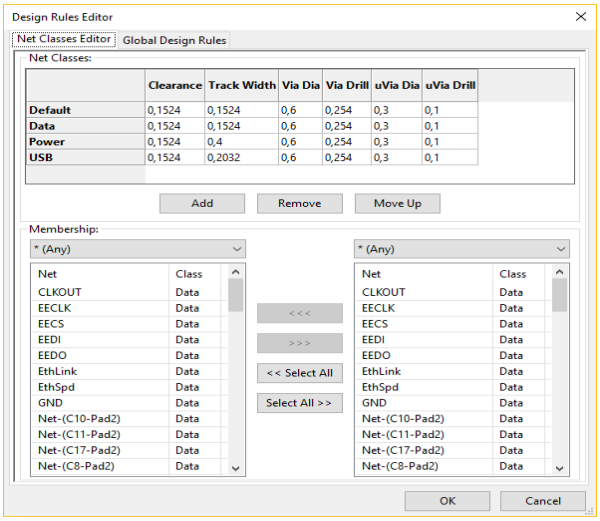
Esimese sammuna on tarvis seadistada Pcbnew, et see vastaks trükiplaadi tootja nõudmistele ja et sellel oleks olemas ka plaadi enda nõudmiste täitmiseks vajalikud seadistused. Antud emaplaadi puhul oli tarvis vähemalt neljakihilist trükiplaati, mis võimaldaks varjestada USB siine.

Pcbnew programmis on antud seadistuste määramine väga lihtsaks tehtud, valides ülemisest menüüst nupp „Design Rules" ja rippmenüüst „Layer Setup". Avanenud aknast tuleb valida rippmenüüst „Copper Layers" valik „4" ning „Board Thickness" panna 1.6 mm peale. Soovi korral võib nimetada kihid ümber, ise nimetasin neid järgnevalt: „F.Cu", „GND", „In.Cu", „GND2". Maanduse kihtidel määrasin neile tüübi „Power".

Järgmiseks etapiks oli tarvis otsida plaadi tootja kodulehelt (antud plaadi puhul oli selleks https://www.Eurocircuits.com) informatsiooni selle kohta, millise täpsusega nad suudavad plaate toota ning selle informatsiooni põhjal täita ära „Design rules" rippmenüüst samanimeline aken. Nende kodulehelt leidsin infot valiku „PCB design guidelines" alt, kus oli välja toodud, et nad ei aktsepteeri ühtegi failivormingut peale

gerber failide, kuna need on ühtlustatud ning nende baasil toimub lõplik tootmine. Samuti oli antud lehel toodud välja informatsioon selle kohta, et nad toodavad trükiplaate vastavalt klassidele (Lisa 6). Seadistusega, mis oli Pcbnew-s vaikevalikuks, leidsin, et plaadile pole võimalik mahutada kõiki komponente. Otsustasin valida tabelist „Pattern Class“ klassi „6“ ning tabelist „Drill Class“ klassi „C“ ning seadistasin akna vastavalt (Illustratsioon 3-4).

Lisaks tekitasin endale kaks klassi, millest üks oli mõeldud voolu jaoks, mille tegin suurema rajalaiusega, et olla kindel, et maksimaalselt voolu jõuaks iga komponendini ning ükski rada läbi ei põleks. Antud klassile andsin nimeks „Power“ ja sinna said lisatud peamised 1.8V, 3.3V ja 5V rajad. Lisaks läksid sinna alla ka USB-st tulevad 5V rajad „VCCA“ ja „VCCB“ ning ka lan9512 jaoks vajalikud 1.8V rajad VDD1V8_ETH, VDD1V8_USB ja VDD_1V8_CORE. Teine loodud klass, on mõeldud ainult USB informatsiooni radade jaoks kuna neil on tarvis 90 oomist rajatakistust.



Illustratsioon 3-4 Design Rules Editor kuvatõmmis

3.4.3 Laotus

Pärast seadistamist alustasin komponentide paigutamist, milleks valisin „Tools“ menüüst valiku „Netlist. Avanenud aknast vajutades „Browse“ nupule, avanes aken kust valisin eelnevalt Eeschemas loodud võrgustiku faili. Järgnevalt vajutasin „Read Current Netlist“ nupule, ning seejärel genereeriti põhivaatesse kõik komponendid. „F11“ nupule vajutades valisin OpenGL vaate, mis võimaldas hilisemalt komponentide vahelisi radasid luua. Alguses valisin kõik komponendid ning hiire paremklõpsu vajutamisega tekkinud menüüst valisin „Align/Distribute“ ja „Distribute horizontally“. Viimase ettevalmistusena seadistasin ruudustiku suuruse, mis oli vastavalt plaadi suurusele x-

teljel 103 mm ja y-teljel 58 mm, ning lõin paremalt äärest valitud „Edge Cuts“ kihile raami.

Asetasin kõige suuremad komponendid nende asukohtadesse, mille sain eelneva arendustegevuse käigus tekkinud failidest, ehk Ethernet port läks vasakust alumisest äärest 2.5 mm eemale ning USB port sellest veel 4 mm edasi. Järgnevalt tegelesin SODIMM pesaga mille asetasin pesa jalajälje allumise osa keskjoone järgi plaadi ülemisest äärest 31 mm, kuna RPCM on 30 mm kõrge, otsustasin lisaks veel ühe millimeetri panna. Kõik ülejäänud komponendid asetasin loominguliselt asukohtadesse, mis tundusid neile sobivat, jälgides seda, millega nad ühendatud olema peaksid. Viimasena tulid takistid, kondensaatorid, induktorid ja transistorid. Lisasin kondensaatorid võimalikult lähedale neile vastavale jalale.

Esialgses variandis hakkasin siine marsruutima pärast laotuse tegemist, kuid tegevuse mõttetus avaldas end peagi seoses ruumi puudusega. Pidin iga järgneva raja juures eelnevaid kas kustutama või ümber liigutama selleks, et oleks võimalik midagi teha. Sel hetkel otsustasin hakata uurima, millised Pcbnews võimalused automatiseerida marsruutimist. Ilmnes, et selleks on kasutusel programm nimega FreeRoute, mille panin tööle eraldiseisvalt, kuid on ka võimalus seda integreerida Pcbnewsse. Genereerisin dsn faili, valides ülemisest menüüst kollase nupu „Fast access to the FreeROUTE external advanced router“.

3.5 FreeRoute

FreeRoute on 2004. aastal loodud programm, mis on mõeldud trükiplaatide radade marsruutimiseks. Seda enam ei hallata, kuid projekti käigus leidsin veel toimiva variandi. Importisin sinna sisse just loodud dsn faili ning vajutasin „Start“ nupule. Jätsin programmi üleöö jooksma lootuses, et see leiab võimalikult optimaalsed rajad. Viimaks eksportisin sessioonifaili, ning lugesin selle uuesti sisse Pcbnew-s.

3.6 Parandused

Alguses võtsin ette USB informatsiooni rajad, mida polnud eelnevalt marsruutinud, ning vedasin need käsitsi neile vajalikele kohtadele. Pärast seda, marsruutisin lahti

jäänud jalgade vahelised ühendused ja hakkasin parandama „Design Rule Check“ loodud murekohti.

Pärast neljandat kontrolli Eurocircuitsi veebirakenduses ja esile tõusnud vigade parandamist läks plaat trükki. Viimase asjana koostasin http://www.farnell.ee veebipoes komponentide nimekirja, milles tõin välja iga komponendi lingi ja kulunud koguse. Kuu aega hiljem jõudis plaat Eestisse ning läks Keilas asuvasse Alktechi tehasesse, kus lisati plaadile juurde kõik nimekirjas olnud komponendid.

4 Testimine

Kaks kuud pärast trükiplaadi failide tehasesse saatmist jõudis valmis plaat minuni. Enne testimise alustamist koostasin nimekirja omadustest, mida peaksin testima:

- Ilma RPCM-ta
    - Radade jätkuvus
    - Pingemuunduri toimimine
    - 5V, 3.3V, 1.8V radade täpsus
    - Kohtvõrgu pordi tulekeste toimimine
- RPCM-ga
    - RPCM-i erinevate pingete olemasolu
    - Raspberry Pi 2 pealt trükiplaadi algseadistamine
    - Erinevate väljundite kontrollimine (valgusdioodid, järjestikliidese toimimine)

Üllatavalt ei olnud ükski rada katki ja signaal liikus testitud kohtades sujuvalt. Järgmisena ühendasin toiteploki, mille seadistasin algselt 5V peale ja hiljem 12V peale, et testida probleemide tekkimist erinevate pingetega. Kummagagi ei tekkinud ühtegi probleemi, kõik rajad olid sobiva pinge peal. Viimase ilma RPCM-ita testina ühendasin Etherneti kaabli porti ning ühenduse toimimist näitavad valgusdioodid läksid põlema.

RPCM-i lisades sain RPCM-i testpunktidest tulemused 3.3V ja 1.8V, mis peaksidki olema. Ootamatult tekkis järgneva etapiga probleem – nimelt ei võimaldanud minu valitud USB jaotur ühendust „Slave mode“ luua. „Slave mode“ oli vajalik, kuna juhul kui RPCM ei leia enda püsimälust „bootcode.bin“ faili, siis jääb see ootama ühendust „Slave mode“, et sellele fail kirjutatakse. Linuxiga on seda kõige kergem teha, selleks oli kasutusel Raspberry Pi 2. Tekkinud olukorras ei olnud mul enam võimalik projektiga edasi liikuda ning arendusprotsess lõppes seminaritöö raames. Plaanis on tegeleda projektiga edasi bakalaureusetööna.

Kokkuvõte

Kuigi seminaritöö väljund oli algselt mõeldud olema toimiv lahendus, olen ma rahul oma saavutuste ja faktiga, et osa sellest plaadist toimis ega põlenud läbi esimese kasutuskorraga.

Emaplaadi loomise protsessi käigus esines mitmeid tagasilööke, mis tihitipeale pikendasid plaadi arendustegevust. Kahjuks ei avanenud mul võimalust näidata, kuidas panna RPCM-i toimima IT5888 liidesplaadiga, mis oleks olnud selle plaadi peamine tööülesanne.

Töö käigus leidsin mitmeid uusi informatsiooniallikaid ja töömeetodeid, kuidas panna komponente koos toimima. Suurim õppetund minu jaoks oli *datahseet*'i tähtsus, mis sisaldab endas peaaegu kõike selleks, et võimaldada tervikuna toimimist: elektrilisi skeeme ja komponentidega kaasas käivat informatsiooni. Kindlasti kavatsen neid teadmist kasutada järgnevate projektide käigus. Suureks abiks oli ka KiCad tarkvarakomplektiga tutvumine.

Huvitav oli projekti puhul see, et sain kogemusi paljudest eri valdkondadest, näiteks disainimine ja projekteerimine. Isegi nii väikese seadme puhul on detailide kogus ääretult suur ning seega nõuab palju planeerimist, erinevate komponentide hankimist ja plaanide joonistamist. Kindlasti aitas projekt aru saada detailide tähtsusest ja korrektsest tööst.

Kõigile, kes soovivad midagi sarnast ette võtta, soovitan varuda palju kannatust ja järjekindlust.

Kasutatud allikad

Eurocircuits. (kuupäev puudub). *PCB design guidelines*. Kasutamise kuupäev: 15. Detsember 2016. a., allikas Eurocircuits: http://www.eurocircuits.com/PCB-design-guidelines

Takis, N. (30. 01 2017. a.). Vajadus väikesemõõtmelise CPU-plaadi järele. (J. Jõeleht, Intervjueerija)

The KiCad Team. (17. March 2014. a.). *Pcbnew*. Kasutamise kuupäev: 29. Jaanuar 2017. a., allikas KiCad Documentation: http://docs.kicad-pcb.org/stable/en/pcbnew.html

The KiCad Team. (22. May 2015. a.). *CvPcv*. Kasutamise kuupäev: 29. Jaanuar 2017. a., allikas KiCad Documentation: http://docs.kicad-pcb.org/stable/en/cvpcb.html

The KiCad Team. (30. May 2015. a.). *Eeschema*. Kasutamise kuupäev: 29. 01 2017. a., allikas KiCad Documentation: http://docs.kicad-pcb.org/stable/en/eeschema.html#_description

The Raspberry Foundation. (24. 04 2014. a.). *About Us*. Kasutamise kuupäev: 08. 02 2017. a., allikas Raspberry Pi: https://web.archive.org/web/20140627054612/http://www.raspberrypi.org/about/

The Raspberry Foundation. (kuupäev puudub). *COMPUTE MODULE IO BOARD V1*. Kasutamise kuupäev: 08. 02 2017. a., allikas Raspberry pi: https://www.raspberrypi.org/products/compute-module-io-board/

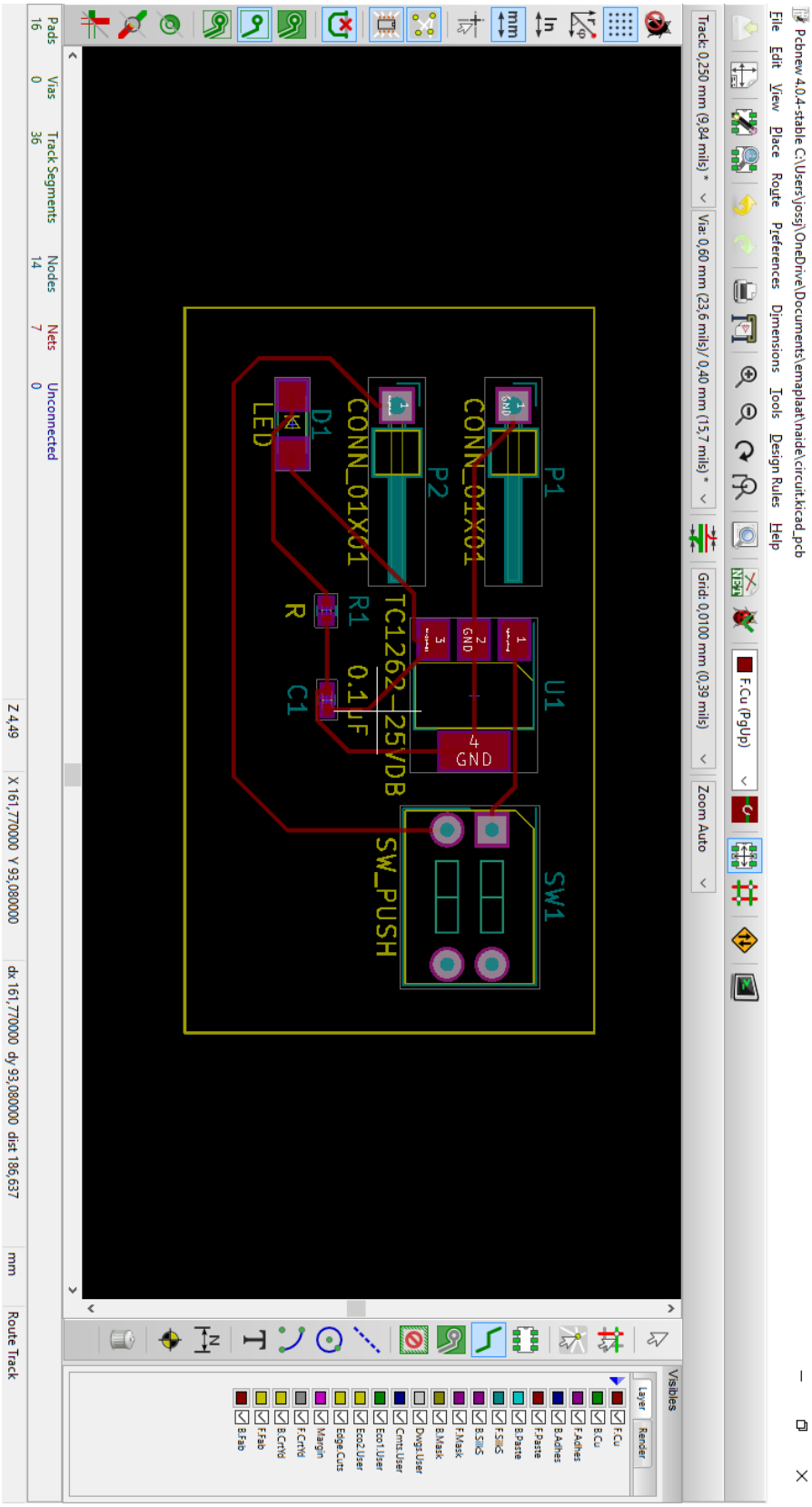
The Raspberry Foundation. (kuupäev puudub). *RASPBERRY PI 1 MODEL A+*. Kasutamise kuupäev: 8. 02 2017. a., allikas Raspberry Pi: https://www.raspberrypi.org/products/model-a-plus/

The Raspberry Foundation. (kuupäev puudub). *RASPBERRY PI 2 MODEL B*. Kasutamise kuupäev: 08. 02 2017. a., allikas Raspberry Pi: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

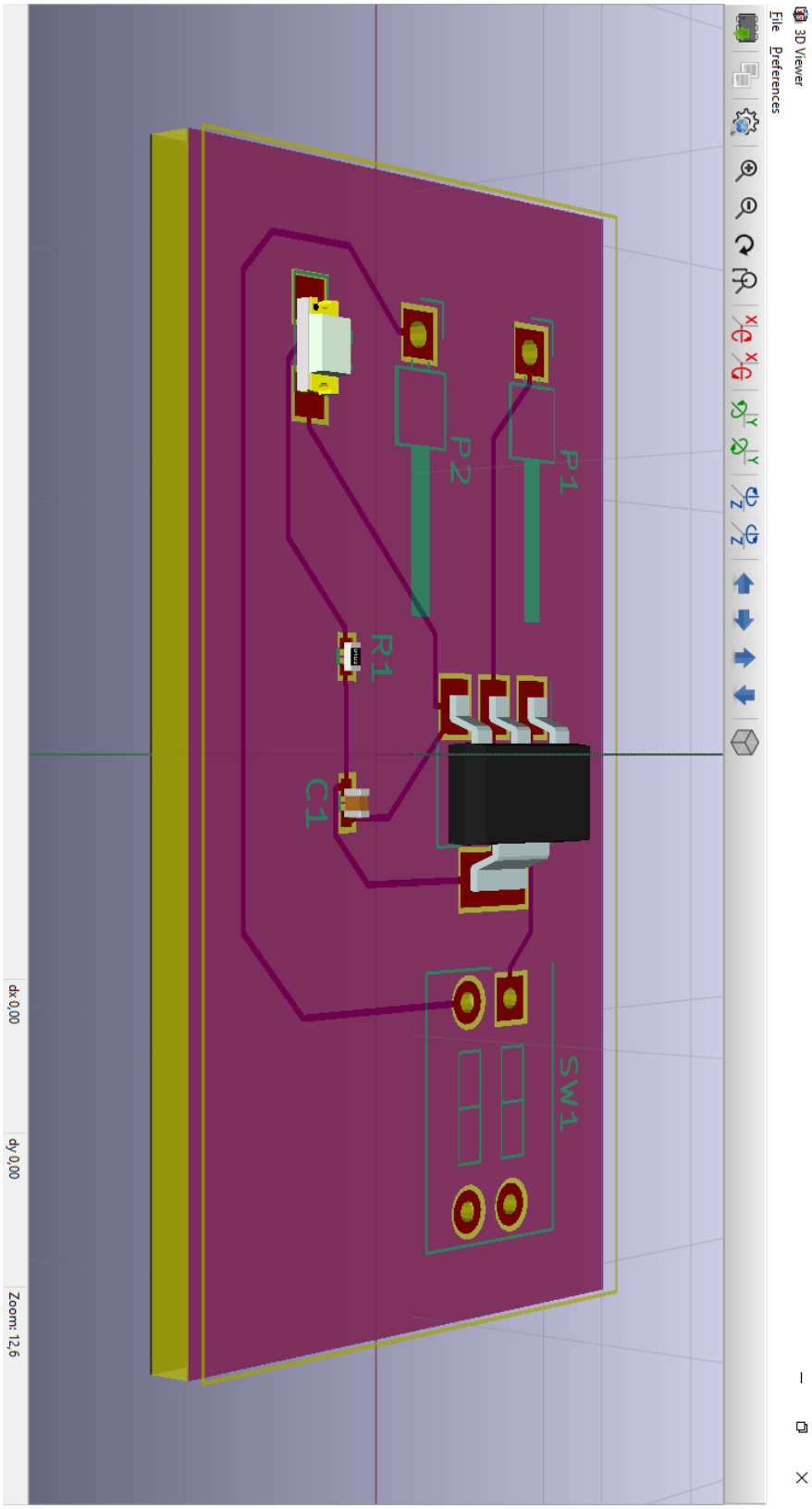
The Raspberry Foundation. (kuupäev puudub). *RASPBERRY PI 3 MODEL B*. Kasutamise kuupäev: 08. 02 2017. a., allikas Raspberry Pi: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

Lisad

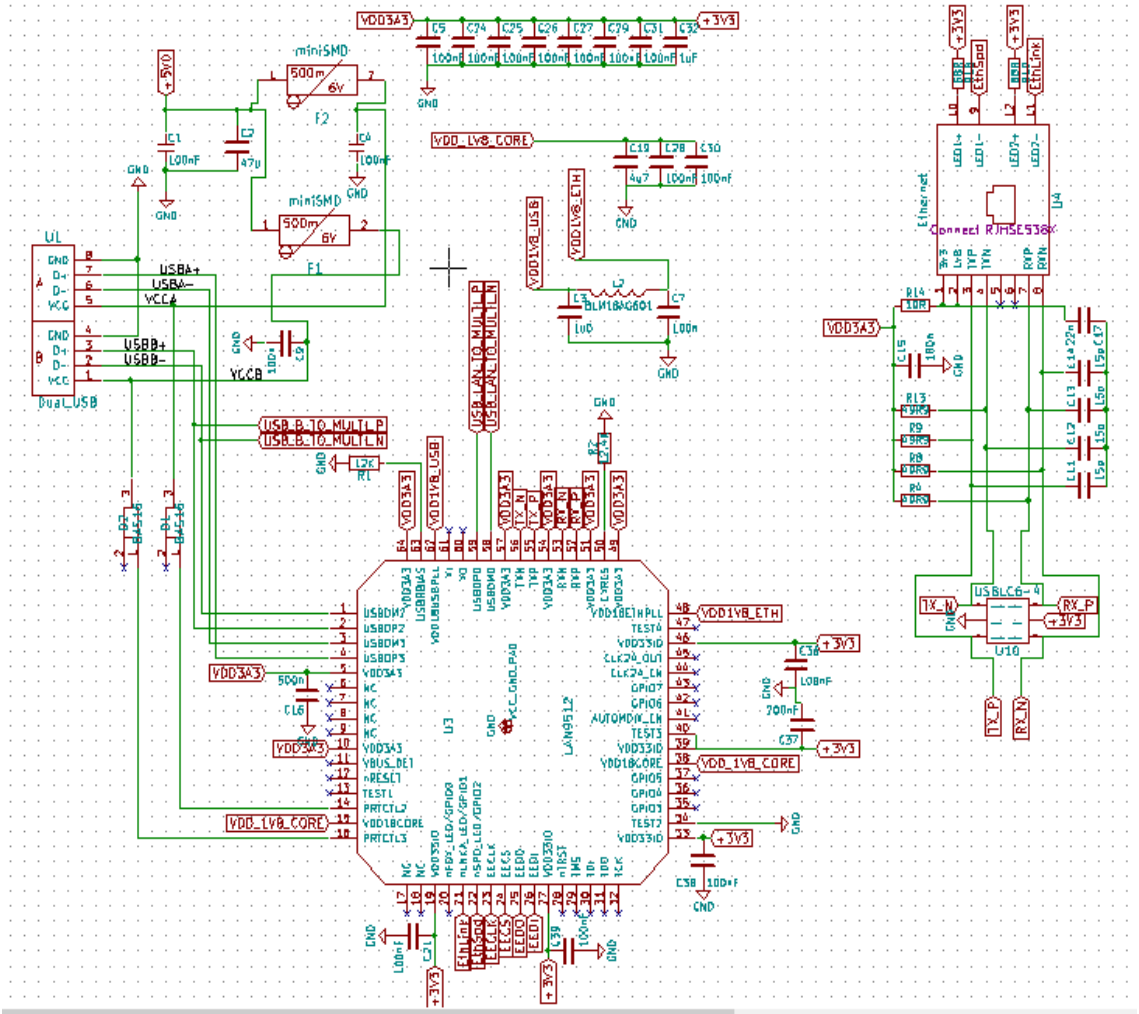
Lisa 1. Valmis näidisplaat



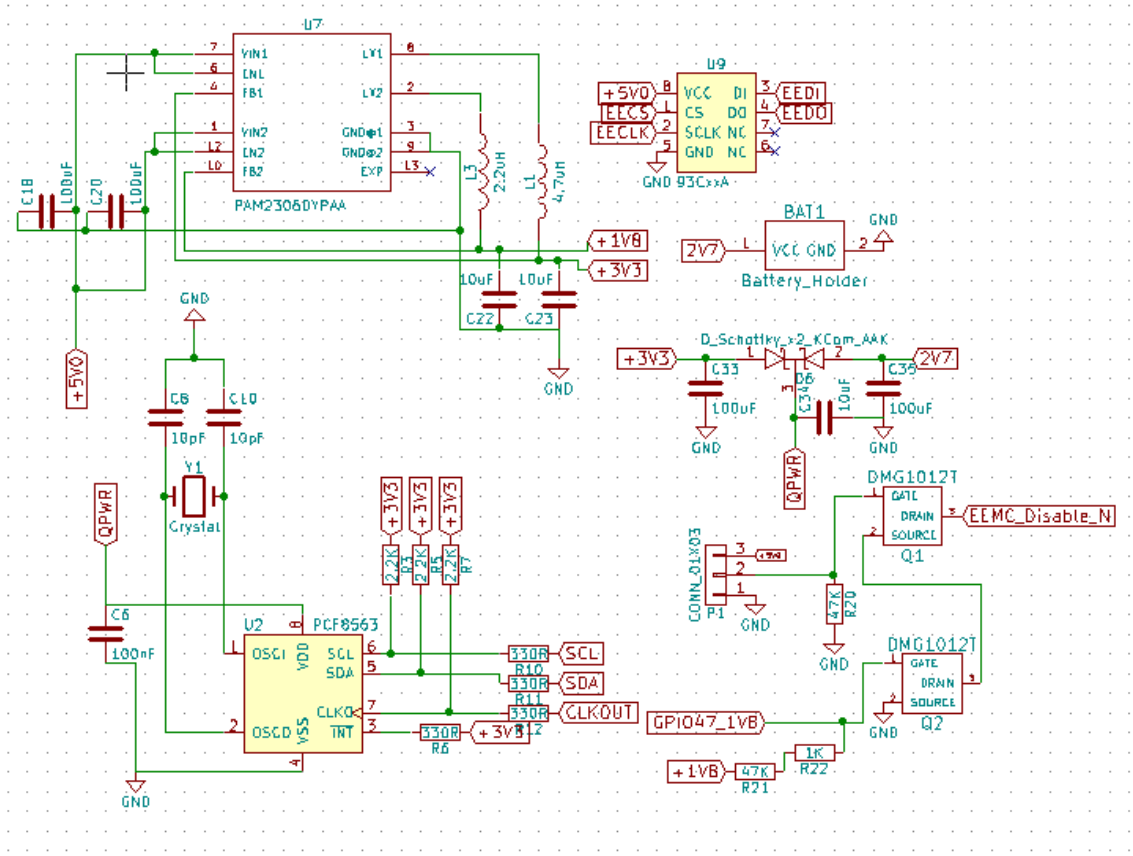
Lisa 2. Näidisplaadi 3D mudel



Lisa 3. Esimene kavand



Lisa 4. Teises kavandis lisatud osad



Lisa 5. Compute Module'i jalgade asetus

CM1	CM3-Lite	CM3	PIN		PIN	CM3	CM3-Lite	CM1
GND			1		2	EMMC_DISABLE_N		
GPIO0			3		4	NC	SDX_VDD	NC
GPIO1			5		6	NC	SDX_VDD	NC
GND			7		8	GND		NC
GPIO2			9		10	NC	SDX_CLK	NC
GPIO3			11		12	NC	SDX_CMD	NC
GND			13		14	GND		NC
GPIO4			15		16	NC	SDX_D0	NC
GPIO5			17		18	NC	SDX_D1	NC
GND			19		20	GND		NC
GPIO6			21		22	NC	SDX_D2	NC
GPIO7			23		24	NC	SDX_D3	NC
GND			25		26	GND		
GPIO8			27		28	GPIO28		
GPIO9			29		30	GPIO29		
GND			31		32	GND		
GPIO10			33		34	GPIO30		
GPIO11			35		36	GPIO31		
GND			37		38	GND		
GPIO0-27_VDD			39		40	GPIO0-27_VDD		
			KEY					
GPIO28-45_VDD			41		42	GPIO28-45_VDD		
GND			43		44	GND		
GPIO12			45		46	GPIO32		
GPIO13			47		48	GPIO33		
GND			49		50	GND		
GPIO14			51		52	GPIO34		
GPIO15			53		54	GPIO35		
GND			55		56	GND		
GPIO16			57		58	GPIO36		
GPIO17			59		60	GPIO37		
GND			61		62	GND		
GPIO18			63		64	GPIO38		
GPIO19			65		66	GPIO39		
GND			67		68	GND		
GPIO20			69		70	GPIO40		
GPIO21			71		72	GPIO41		
GND			73		74	GND		
GPIO22			75		76	GPIO42		
GPIO23			77		78	GPIO43		
GND			79		80	GND		
GPIO24			81		82	GPIO44		
GPIO25			83		84	GPIO45		
GND			85		86	GND		
GPIO26			87		88	HDMI_HPD_N_1V8		GPIO46_1V8
GPIO27			89		90	EMMC_EN_N_1V8		GPIO47_1V8
GND			91		92	GND		
DSI0_DN1			93		94	DSI1_DP0		
DSI0_DP1			95		96	DSI1_DN0		
GND			97		98	GND		
DSI0_DN0			99		100	DSI1_CP		
DSI0_DP0			101		102	DSI1_CN		

DSI0_DP0	101		102	DSI1_CN
GND	103		104	GND
DSI0_CN	105		106	DSI1_DP3
DSI0_CP	107		108	DSI1_DN3
GND	109		110	GND
HDMI_CLK_N	111		112	DSI1_DP2
HDMI_CLK_P	113		114	DSI1_DN2
GND	115		116	GND
HDMI_D0_N	117		118	DSI1_DP1
HDMI_D0_P	119		120	DSI1_DN1
GND	121		122	GND
HDMI_D1_N	123		124	NC
HDMI_D1_P	125		126	NC
GND	127		128	NC
HDMI_D2_N	129		130	NC
HDMI_D2_P	131		132	NC
GND	133		134	GND
CAM1_DP3	135		136	CAM0_DP0
CAM1_DN3	137		138	CAM0_DN0
GND	139		140	GND
CAM1_DP2	141		142	CAM0_CP
CAM1_DN2	143		144	CAM0_CN
GND	145		146	GND
CAM1_CP	147		148	CAM0_DP1
CAM1_CN	149		150	CAM0_DN1
GND	151		152	GND
CAM1_DP1	153		154	NC
CAM1_DN1	155		156	NC
GND	157		158	NC
CAM1_DP0	159		160	NC
CAM1_DN0	161		162	NC
GND	163		164	GND
USB_DP	165		166	TVDAC
USB_DM	167		168	USB_OTGID
GND	169		170	GND
HDMI_CEC	171		172	VC_TRST_N
HDMI_SDA	173		174	VC_TDI
HDMI_SCL	175		176	VC_TMS
RUN	177		178	VC_TDO
VDD_CORE (DO NOT CONNECT)	179		180	VC_TCK
GND	181		182	GND
1V8	183		184	1V8
1V8	185		186	1V8
GND	187		188	GND
VDAC	189		190	VDAC
3V3	191		192	3V3
3V3	193		194	3V3
GND	195		196	GND
VBAT	197		198	VBAT
VBAT	199		200	VBAT

Lisa 6. Eurocircuits'i kategooriate informatsioon

Eurocircuits - PCB design classification overview

Pattern Class

Service	class 3	class 4	class 5	class 6	class 7	class 8	class 9	class 10	
OTW	0.250 10	0.200 8	0.175 7	0.150 6	0.125 5	0.100 4	0.090 3.5	<0.090 <3.5	mm-mil
OTT-OTP-OPP	0.250 10	0.200 8	0.175 7	0.150 6	0.125 5	0.100 4	0.090 3.5	<0.090 <3.5	mm-mil
OAR	0.200 8	0.150 6	0.150 6	0.125 5	0.125 5	0.100 4	0.100 4	<0.100 <4	mm-mil
ITW	0.250 10	0.200 8	0.175 7	0.150 6	0.125 5	0.100 4	0.090 3.5	<0.090 <3.5	mm-mil
ITT-ITP-IPP	0.250 10	0.200 8	0.175 7	0.150 6	0.125 5	0.100 4	0.090 3.5	<0.090 <3.5	mm-mil
IAR	0.200 8	0.150 6	0.150 6	0.125 5	0.125 5	0.100 4	0.125 5	<0.125 <5	mm-mil
IPI	0.275 11	0.225 9	0.225 9	0.200 8	0.200 8	0.200 8	0.200 8	<0.200 <8	mm-mil

The smallest value (OTW, OTT-OTP-OPP, OAR, ITW, ITT-ITP-IPP, IAR, IPI) determines the **Pattern Class** of the board

Base Cu

Base Cu OL	min Pattern values OTT-OTP-OPP		OTW		
12µm ⅓oz	0.090	3.5	0.090	3.5	mm-mil
18µm ½oz	0.125	5	0.090	3.5	mm-mil
35µm 1oz	0.175	7	0.125	5	mm-mil
70µm 2oz	0.250	10	0.200	8	mm-mil
105µm 3oz	0.300	12	0.250	10	mm-mil
Base Cu IL	**ITT-ITP-IPP**		**ITW**		
12µm ⅓oz	0.090	3.5	0.090	3.5	mm-mil
18µm ½oz	0.100	4	0.090	3.5	mm-mil
35µm 1oz	0.125	5	0.125	5	mm-mil
70µm 2oz	0.250	10	0.200	8	mm-mil
105µm 3oz	0.300	12	0.250	10	mm-mil

Preceding letters O and I stand for Outer- and Inner layer
Example: OTW = Outer layer Track Width

OAR : smallest OAR (Outer layer Annular Ring = 1/2 (Outer layer pad diameter - PHD))
IAR : smallest IAR (Inner layer Annular Ring = 1/2 (Inner layer pad diameter - PHD))
IPI (Inner layer Pad Insulation) : Clearance between edge PHD of any unconnected hole(PTH/NPTH) and any nearest copper

Smallest PHD : Production Hole Diameter or tool size = finished hole size + 0.10mm/4mil for Plated Through Holes
+ 0.00mm/0mil for Non Plated Through Holes

Drill Class

Service	class A	class B	class C	class D	class E	class F	
	P+S+R+I	P+S+R	P+S+R	S+R	S+R	-	
min PHD	0.60 0.026	0.45 0.018	0.35 0.014	0.25 0.010	0.20 0.008	<0.20 <0.008	mm-inch
Corresponding finished holes sizes							
PTH	0.50 0.022	0.35 0.014	0.25 0.010		0.15 0.006	<0.10 <0.004	mm-inch
NPTH	0.60 0.026	0.45 0.018	0.35 0.014		0.25 0.010	<0.20 <0.008	mm-inch

The smallest value (PHD) determines the **Drill Class** of the PCB

Max. PCB thickness to Drill Class

3.20 0.125	3.20 0.125	2.40 0.093	2.00 0.079	1.60 0.062	mm-inch	Aspect ratio is 1 / 8

Note A: VIA holes are Plated Through Holes, default defined as <=0.45mm (18mil) for all services or <= as defined by the customer in the order details.
VIA holes have a maximum negative tolerance of 0.30mm (12mil)

Note B: This classification table can only be put into praxis on PCB designs that have a **Plating Index of 0.40 or higher**. This is calculated in the PCB Visualizer analyse and displayed in the PCB Visualizer order details.

Services index : P = PCB proto S = STANDARD pool R = RF pool I = IMS pool