



TALLINNA TEHNIKAÜLIKOOL

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

**KASUTAJALIIDES RT BOX
RIISTVARASIMULAATORILE**

**USER INTERFACE FOR RT BOX HARDWARE-IN-LOOP
SIMULATOR**

BAKALAUREUSETÖÖ

Üliõpilane: Lars-Henry Vehman

Üliõpilaskood: 179397EAAB

Juhendaja: Indrek Roasto, vanemlektor

Tallinn 2022

(Tiitellehe pöördel)

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

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

"....." 20.....

Autor:

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

"....." 20.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

"....."20.....

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina _____ (autori nimi)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

(lõputöö pealkiri)

mille _____ juhendaja _____ on

(juhendaja nimi)

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

_____ (kuupäev)

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Lars-Henry Vehman

Lõputöö liik: Bakalaureusetöö

Töö pealkiri: Kasutajaliides RT box riistvarasimulaatorile

Kuupäev:
21.12.2022

45 lk (lõputöö lehekülgede arv koos lisadega)

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja(d): Indrek Roasto

Töö konsultant (konsultandid): -

Sisu kirjeldus:

Jõuelektroonika muunduritel on palju rakendusi ja seetõttu on pidevalt suurenenud ka muundurite keerukus. Kasutusele on võetud uued jõuelektroonika süsteemide arendusmeetodid, mis lihtsustavad muundurite arendust. Üheks selliseks näiteks on erinevad riistvara simulaatorid, mida kasutatakse tihti toitemuundurite kiirprototüüpimisel. Kiirprototüüpimise all mõeldakse muunduri graafilist programmeerimist st programm koostatakse blokkskeemi kujul ja kasutaja ei pea seetõttu tundma C keelt ja mikrokontrolleri siseelu. Programmi konverteerimine masinkoodi tehakse ära arvuti poolt. Lisaks on vaja reaajas jälgida ka muunduri tööparameetreid (voolud, pinged) või mõnda parameetrit ka muuta. Selleks on kaks võimalust: kasutada spetsiaalset tarkvara ja arvutit või välist füüsilist kasutajaliidest. Arvuti ja tarkvara kasutamise puuduseks on info ja sätete üleküllus ning väiksem töökindlus kuna puuduvad füüsilised nupud muunduri kiireks väljalülitamiseks. Füüsiline kasutajaliides lahendab need probleemid pakkudes kasutajale ühte piiratud suurusega ekraani, konkreetse infoga ning vabalt programmeeritavaid nuppe. Lisaks ei vaja väline kasutajaliides eritarkvara ega litsentse, mis laiendab muunduri kasutamismõimalusi. Antud lõputöö eesmärgiks ongi sellise välise kasutajaliidese väljatöötamine.

Eesmärgiks on projekteerida ja ehitada ESP32 mikrokontrolleri baasil kasutajaliides jõuelektroonika riistvarasimulaatorile RT box. Kasutajaliides peab omama nuppe ja LCD ekraani. Kasutajaliides võimaldab andmevahetust RT boxiga ja kuvada andmeid nii LCD ekraanil kui ka üle wifi arvutis. Täpsem lähteandete analüüs on tarvilik enne tehnilise teostuse planerimist.

Märksõnad: riistvarasimulaator, kasutajaliides, andmevahetus, pingemuundur

ABSTRACT

<i>Author:</i> Lars-Henry Vehman	<i>Type of the work:</i> Bachelor
<i>Title:</i> User Interface for RT Box Hardware-in-Loop Simulator	
<i>Date:</i> 21.12.2022	<i>45 pages (the number of thesis pages including appendices)</i>
<i>University:</i> Tallinn University of Technology	
<i>School:</i> School of Engineering	
<i>Department:</i> Department of Electrical Power Engineering and Mechatronics	
<i>Supervisor(s) of the thesis:</i> Indrek Roasto	
<i>Consultant(s):</i> -	
<p><i>Abstract:</i></p> <p>Power electronics converters have many applications and therefore the complexity of converters has constantly increased. Advanced development methods for power electronics systems have been introduced to facilitate the development of converters. One such example is various hardware simulators, which are often used in fast prototyping of power converters. Fast prototyping refers to graphical programming of the converter, i.e. the program is created in the form of a block diagram and the user does not have to know the C language and the inner workings of the microcontroller. The conversion of the program into machine code is carried out by the computer. In addition, it is necessary to monitor the operating parameters (currents, voltages) of the converter in real time or change some parameter. There are two options for this: use special software and a computer or an external physical user interface. The disadvantage of using a computer and software is the overload of information and settings, and lower reliability as there are no physical buttons for quick shutdown of the converter. The physical user interface solves these problems by offering the user a screen of limited size with specific information and programmable buttons. In addition, the external user interface does not require special software or licenses, which expands the possibilities of using the converter.</p> <p>The goal of this final thesis is to develop such an external user interface. The goal is to design and build a user interface for the RT box power electronics hardware simulator based on the ESP32 microcontroller. The user interface must have buttons and an LCD screen. The user interface allows data exchange with the RT box and displays data on both the LCD screen and over WiFi on a computer. A more detailed analysis of the data is necessary before planning the technical implementation.</p>	
<i>Keywords:</i> hardware-in-loop simulator, user interface, data transmission, voltage converter	

LÕPUTÖÖ ÜLESANNE

Lõputöö teema: **Kasutajaliides RT box riistvarasimulaatorile**
Lõputöö teema inglise keeles: **User Interface for RT Box Hardware-in-Loop Simulator**
Üliõpilane: **Lars-Henry Vehman, 179397EAAB**
Eriala: **Elektroenergeetika ja mehhatroonika**
Lõputöö liik: **bakalaureusetöö**
Lõputöö juhendaja: **Indrek Roasto**
Lõputöö kaasjuhendaja:
(ettevõtte, amet ja kontakt)

Lõputöö ülesande
kehtivusaeg: **kehtivusaja annab juhendaja**
2022/2023 2022/2023 Sügis
Lõputöö esitamise tähtaeg: **21.12.2022**

Üliõpilane (allkiri)

Juhendaja (allkiri)

Õppekava juht (allkiri)

Kaasjuhendaja (allkiri)

1. Teema põhjendus

Riistvarasimulaatorit kasutatakse tihti toitemuundurite kiirprototüüpimisel, mis tähendab muunduri graafilist programmeerimist. Graafilise programmeerimise korral koostatakse programm blokk skeemi kujul ja kasutaja ei pea seetõttu tundma C keelt ja mikrokontrolleri siseelu. Programmi konverteerimine masinkoodi tehakse ära arvuti poolt. Sellise programmi jooksutamiseks on vaja spetsiaalset arvutit (nn riistvarasimulaator). Tihti on aga vaja reaajas jälgida muunduri tööparameetreid (voolud, pinged) või mõnda parameetrit ka muuta. Selleks ongi vaja kasutajaliidest, millel oleks ekraan ja nupud. Kuna RT box riistvarasimulaator ei oma füüsilist kasutajaliidest (nupud, LCD) siis tuleb see ise projekteerida.

2. Töö eesmärk

Eesmärgiks on projekteerida ja ehitada ESP32 mikrokontrolleri baasil kasutajaliides jõuelektroonika riistvarasimulaatorile RT box. Kasutajaliides peab omama nuppe ja LCD ekraani. Kasutajaliides võimaldab andmevahetust RT boxiga ja kuvada andmeid nii LCD ekraanil kui ka üle wifi arvutis.

3. Lahendamisele kuuluvate küsimuste loetelu:

- Ülevaate koostamine RT box riistvarasimulaatorist ja selle juhtimisvõimalustest.
- Kasutajaliidese projekteerimine ja ehitamine.
- Kasutajaliidese testimine vastavalt lõputöö eesmärgile.

4. Lähteandmed

Riistvarasimulaatori toote koduleht [1]

Andmelehed: mikrokontroller [2], riistvarasimulaator [3]

Mikrokontrolleri programmeerimise juhend [4]

5. Uurimismeetodid

Lähteandmete analüüs ja täpse tehnilise teostuse plaanimine.

Trükkplaadi disain ja ehitamine

Juhtprogrammi koostamine ja katsetamine

6. Graafiline osa

Elektroonikaskeemid, koostude joonised. Algoritmide kirjeldused.

Töö põhiosas ja lisades.

7. Töö struktuur

Töö ülesehitus

1. Sissejuhatus

2. Riistvarasimulaatori ülevaade ja võimalused

3. Kasutajaliidese projekteerimine

1. Tehniliste parameetrite määramine

2. Andmeside liidese valimine

3. Komponentide valimine

4. Trükkplaadi disainimine

4. Programmi koostamine

1. Andmeside liidese tööpõhimõtted

2. Täisarvuliste andmete saatmine

3. Ujuvpunktarvude saatmine

4. Nupuvajutuste lugemine

5. LCD tööpõhimõte

6. Wifi kaudu andmete kuvamine

7. Testimine ja tulemuste analüüs

8. Kasutatud kirjanduse allikad

Töö koosneb võrgupõhistest allikatest, mis on toodete andmelehed ja API dokumentatsioon

9. Lõputöö konsultandid

Puuduvad

10. Töö etapid ja ajakava

kirjanduse läbitöötamine	08.10.22
täpse tehnilise teostuse plaanimine.	15.10.22
skeemi ja trükkplaadi disain	22.10.22
prototüübi koostamine	01.11.22
juhtprogrammi koostamine ja katsetamine	14.12.22
uuringu tulemuste kirjeldamine	21.12.22
järelduste kirjutamine	28.12.22
kokkuvõtte koostamine	05.12.22
töö esimene versioon valmis	12.12.22
juhendajale läbilugemiseks saatmine	12.12.22
paranduste sisseviimine	13.12.22
juhendajale teiseks läbilugemiseks saatmine	14.12.22
töö lõplik versioon valmis.	20.12.2022

SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE.....	4
ABSTRACT.....	5
LÕPUTÖÖ ÜLESANNE.....	6
EESSÕNA.....	10
Lühendite ja tähiste loetelu.....	11
SISSEJUHATUS.....	12
1. Riistvarasimulaatori ülevaade ja võimalused.....	13
1.1 RT Box riistvarasimulaatori võimalused.....	13
1.1.1 Riistvara ahelas simulatsioon(HIL).....	13
1.1.2 Kiirprototüüpimine.....	14
1.1.3 Mikrokontroller ahelas simulatsioon (PIL).....	14
1.2 Riistvarasimulaatori laiendusplaat.....	15
2. Kasutajaliidese projekteerimine.....	16
2.1 Kasutajaliidese nõuded.....	17
2.2 Tehniliste parameetrite määramine.....	17
2.2.1 ESP32-S2-Saola-1 arendusplaadi parameetrid.....	17
2.3 Andmesideliidese valimine.....	20
3. Programmi koostamine.....	21
3.1 Andmeside liidese tööpõhimõtted.....	23
3.2 Täisarvuliste andmete saatmine.....	24
3.3 Andmevahetuse protokoll üle SPI liidese.....	25
3.4 Nupuvajutuste lugemine.....	26
3.5 LCD tööpõhimõte.....	26
3.6 Programm RT Boxis.....	27
3.6.1 Muunduri käivitamise ja seiskamise skeem.....	27
3.6.2 Drosseli voolu mõõtmise skeem.....	28
3.6.3 Muunduri sisendvõimsuse mõõtmise skeem.....	29
3.6.4 Muunduri väljundvõimsuse mõõtmise skeem.....	30
4. Kasutajaliidese testimine ja analüüs.....	31
4.1 Muunduri käivitamine/seiskamine kasutajaliidese kaudu.....	32
4.2 Drosseli voolu keskväärtuse mõõtmine.....	34
4.3 Muunduri sisendvõimsuse arvutamine.....	35
4.4 Muunduri väljundpinge keskväärtuse mõõtmine.....	36
4.5 Tulemuste analüüs.....	37
KOKKUVÕTE.....	39
SUMMARY.....	40
KASUTATUD KIRJANDUSE LOETELU.....	41

LISAD.....42

EESSÕNA

Autor tänab oma juhendajat Indrek Roastot, kes on andnud autorille kõik vajaliku abi ja teadmised lõputöö tegemiseks ning olnud alati toetav ja kättesaadav.

Lühendite ja tähiste loetelu

GPIO	<i>General-Purpose Input Output</i> , ühendusviik
SPI	<i>Serial Peripheral Interface</i> , välisseadmete jadaliides
MISO	<i>Master Input, Slave Output</i> , kontrolleri seadme sisend
MOSI	<i>Master Output, Slave Input</i> , alamseadme sisend
CS	<i>Chip Select</i> , kiibi valimise sisend
LCD	<i>Liquid Crystal Display</i> , vedelkristallekraan
FPGA	<i>Field Programmable Gate Array</i> , programmeeritav loogika massiiv
TS	<i>Testitav seade</i>
HIL	<i>Hardware-In-Loop</i> , riistvara ahelas
DC	<i>Direct Current</i> , alalisvool
PWM	<i>Pulse Width Modulation</i> , pulsilaiusmodulatsioon
BNC	Bayonet Neill–Concelman connector, pisituku tüüp
DNC	<i>Do Not Care</i> , ei loeta
USB	<i>Universal Serial Bus</i> , universaal jadaliides
UART	<i>Universal asynchronous receiver-transmitter</i> , universaalne asünkroonne vastuvõtja-saatja
GND	<i>GrouND</i> , maapotentiaal
PIL	<i>Processor-In-Loop</i> , mikrokontroller ahelas testimine
SoC	<i>System on Chip</i> , süsteem kiibil
I2S	Inter-IC Sound, jadaliides heli edastuse tarbeks
LED	<i>Light Emitting Diode</i> , valgusdiode
ROM	<i>Read Only Memory</i> , ainult lugemiseks mõeldud mälu
RAM	<i>Random Access Memory</i> , muutmälu
SRAM	<i>Static Random Access Memory</i> , staatiline muutmälu

SISSEJUHATUS

Jõuelektronika e muundustehnika mängib tänapäeva maailmas võtmerolli. Tänu toitemuunduritele on saanud võimalikuks roheenergia laialdane kasutamisele võtmine. Samuti on toitemuundurid kõigis nutiseadmetes, mis on muutumas üha kompaktsemateks ja mobiilsemateks. Teiselt poolt on pidevalt suurenenud ka muundurite keerukus, et tagada seadmete töökindlus ja energiatõhusus. Kasutusele on võetud uued jõuelektronika süsteemide arendusmeetodid, mis lihtsustavad keerukate süsteemide arendust ja testimist. Üheks selliseks näiteks on erinevad riistvara simulaatorid, mida kasutatakse tihti toitemuundurite kiirprototüüpimisel. Kiirprototüüpimise all mõeldakse muunduri graafilist programmeerimist st programm koostatakse blokk skeemi kujul ja kasutaja ei pea seetõttu tundma C keelt ja mikrokontrolleri siseelu. Programmi konverteerimine masinkoodi tehakse ära arvuti poolt. Sellise programmi jooksutamiseks on vaja spetsiaalset arvutit (nn riistvarasimulaator). Tihti on aga vaja reaajas jälgida muunduri tööparameetreid (voolud, pinged) või mõnda parameetrit ka muuta. Selleks on kaks võimalust: kasutada spetsiaalset tarkvara ja arvutit või välist füüsilist kasutajaliidest. Arvuti ja tarkvara kasutamise puuduseks on info ja sätete üleküllus ning väiksem töökindlus kuna puuduvad füüsilised nupud muunduri kiireks väljalülitamiseks. Füüsiline kasutajaliides lahendab need probleemid pakkudes kasutajale ühte piiratud suurusega ekraani, konkreetse infoga ning vabalt programmeeritavaid nuppe. Lisaks ei vaja väline kasutajaliides eritarkvara ega litsentse, mis laiendab muunduri kasutamise võimalusi. Antud lõputöö eesmärgiks ongi sellise välise kasutajaliidese väljatöötamine, millel oleks LCD ekraan ja nupud.

1. Riistvarasimulaatori ülevaade ja võimalused



Joonis 1.1 Plecs RT Box tootefoto

Maailmas leidub palju erinevaid riistvarasimulaatoreid nt dSPACE [1], RTDS [2], NI Veristand [3], Opal-RT [4]. Tallinna tehnikaülikoolis on aga juba üle 4 aasta kasutatud Plexim GmbH poolt arendatavat riistvarasimulaatorit RT Box [5] koos vastava tarkvaraga PLECS. Plexim süsteemi eripäraks on asjaolu, et ta on spetsiaalselt disainitud jõuelektroonika otstarbeks. Võrreldes teiste süsteemidega on RT box ka üks kõige paindlikum riistvarasimulaator. RT box omab Xilinx Zynq Z-7030 [6], mis on kahe ARM Cortex-A9 [7] tuumaga SoC. RT Boxil on 16 16-bitilist analoogsisendit ja väljundit, mis on seadistavad -10 ... 10 V pinge vahemikus. RT Boxil on samuti 32 digitaalset sisendit ja väljundit. RT box sobib riistvara ahelas simulatsiooniks (HIL), juhtimise kiirprototüüpimiseks ja mikrokontroller ahelas simulatsiooniks (PIL). RT Boxist on mitu variant erinevate jõudluse näitajatega . Tehnikaülikoolis kasutatakse esimese põlvkonna varianti RT Box 1 (vt joonis 1.1). Töö eesmärgiks on edasi arendada ja tõhusamaks muuta juba olemas olevat RT Boxi arendussüsteemi.

1.1 RT Box riistvarasimulaatori võimalused

1.1.1 Riistvara ahelas simulatsioon(HIL)

Kui RT Boxi kasutatakse HIL-testimiseks, emuleerib see tavaliselt jõuelektroonika süsteemi jõuaste. Jõuaste võib olla lihtne DC/DC muundur, vahelduvvoolu ajamisüsteem või keerukas mitmetasandiline inverterisüsteem. Testitav seade (TS) on mikrokontroller, mis ühendatakse RT Boxiga. Sellise seadistuse korral saab kogu kontrollerit testida ilma tegeliku jõuastmeta. Kontrolleri sisenditeks on enamasti pinge- ja vooluanduritelt tulevad analoogsignaaliid. HIL-simulatsioonis edastavad need

signaalid RT boxi analoogväljundkanalid. Jõuelektronilistes rakendustes genereerib kontrolleri PWM-signaale, et juhtida pooljuhtlüliteid. PWM-signaale saab lugeda RT Boxi digitaalsete sisendite kaudu ajaeraldusvõimega alla 10 ns. Kasutades jäädvustatud sisendandmeid, simuleerib RT Box jõusastet ja esitab simulatsioonitulemused hiljem oma analoogväljunditel. Alla 10 ns I/O viivituse korral tundub kontrolleri jaoks, et juhib tõelist muundursüsteemi [5].

1.1.2 Kiirprototüüpimine

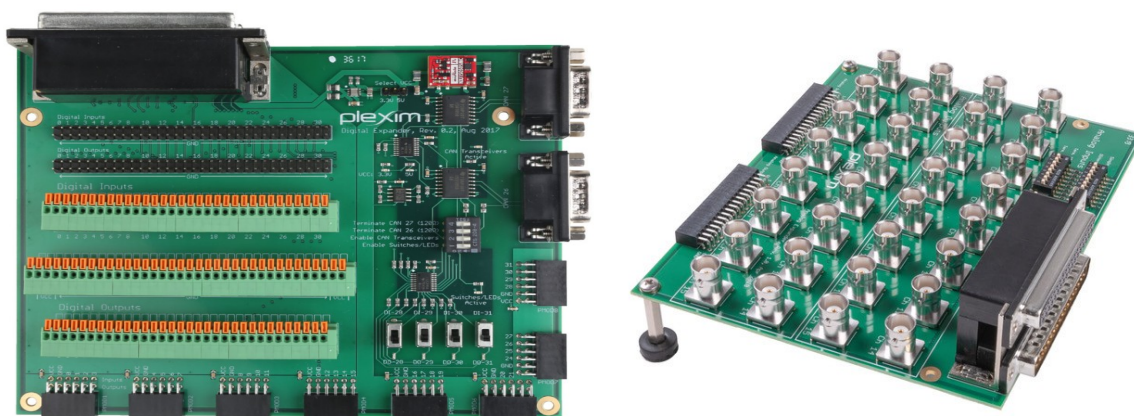
RT Boxi saab kasutada ka toiteastme kontrolleri jaoks. Sel juhul ühendatakse ping- ja vooluandurid RT Boxi analoogsisenditega. Digitaalseid väljundeid kasutatakse toitepooljuhtide jaoks kõrge täpsusega PWM-signaalide genereerimiseks. Kuna RT boxil on rohkem analoog- ja digitaalkanaleid kui enamikul mikrokontrolleritel, samuti kõrgem diskreetimissagedus, milleks on 100kHz. Kaks RT Box, saab need süsteemi täielikuks testimiseks omavahel ühendada. Üks RT Box simuleeriks jõufaasi ja teine toimiks kontrolleri jaoks [5].

1.1.3 Mikrokontroller ahelas simulatsioon (PIL)

RT Boxi saab ka kasutada mikrokontroller-ahelas-simulatsiooni (PIL) eesmärgil. Selle abil saab kinnitada juhtimissüsteemide ja muu riistavara käitumist simulatsioonikeskkonnas. PIL testimine hõlmab reaalaaja protsessori kasutamist, mis tavaliselt on ühendatud simulatsiooni platvormiga, antud juhul siis otse PLECSiga, mis kordineerib ja sünkroniseerib kontrolleri tööd. Protsessori eesmärk on täita reaalaajas juhtalgoritme.

1.2 Riistvarasimulaatori laiendusplaat

RT Boxi esipaneel koosneb neljast pistikust analoog- ja digitaalsed sisendid/väljundid. Kasutaja peab ühendama sisendid/väljundid testitava seadmega. Välise riistvaraga ühendamise lihtsustamiseks ja mugavaks RT Boxi sisendite ja väljunditega ühendamiseks pakub Plexim erinevaid RT Boxi komplekte tarvikud. Digital Breakout Board (vaata joonis 1.2) hõlbustab lihtsat juurdepääsu digitaalsele sisendile ja RT Boxi väljundkanalid klemmplokkide ja viigupäiste kaudu. See plaati kasutatakse koos Analog Breakout Boardiga, mis võimaldab juurdepääsu RT Boxi analoogsisendidele/väljunditele BNC-pistikute või ühendusjuhtme kaudu [8].



Joonis 1.2 RT Boxi digitaalne laiendusplaat (a), RT Boxi analoog laiendusplaat (b) [8]

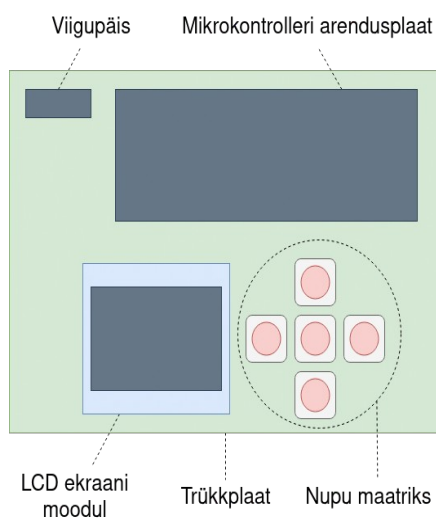
2. Kasutajaliidese projekteerimine

Kasutajaliidese peamine eesmärk on võtta vastu kasutaja poolt sisestatud info ja edastada see RT Boxi simulatsiooni kui ka reaalsele mikrokontrollerile. Kasutaja lihtsuse mõttes otsustati kasutada viite nuppu ja ekraani millega numbreid kuvada. Lõppkasutaja saab tagasisidet oma sisendist läbi ekraanipildi, ning saab läbi nupupaneeli muundurit kaugelt juhtida.

Kasutajaliides peab olema võimeline kahe-suunaliseks suhtluseks, ehk siis kasutajaliides peab olema võimeline saatma andmeid, kui ka vastu võtma.

Ekraani ja nuppude käitamiseks on tarvis kasutada mikrokontrollerit, mis interpreteerib nupuvajutusi kui sündmusi, mida kasutaja soovib teha. Veel on mikrokontroller tarvilik, et ekraanile numbreid ja vajadusel teksti joonistada. Mikrokontroller hoolitseks ka RT Boxi simulatsiooniga suhtlemise eest.

Kasutajaliidese mõtte arendamisel koostati eskiis (vaata joonis 2.1), mis pidi kõik eeltoodud omadused kokku võtma ja aitama luua parema ettekujutuse lõplikust prototüübist.



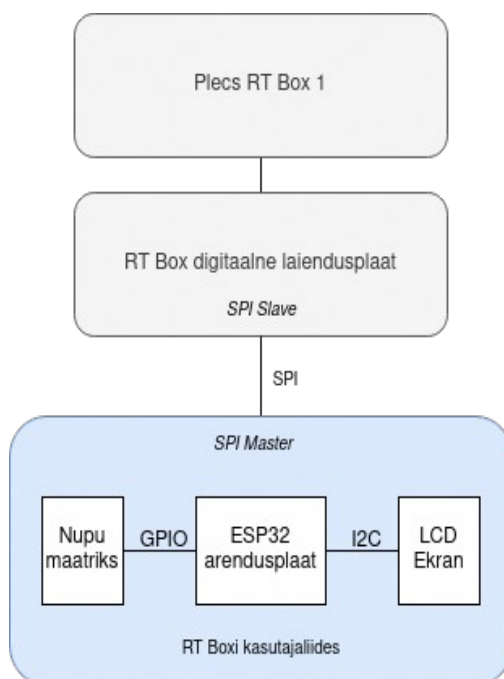
Joonis 2.1 Eskiis kasutajaliidese

2.1 Kasutajaliidese nõuded

Kasutajaliidese lähteandmete koostamisel tuli arvestada inimese tajuvuse piiridega, tuli. Oli vaja leida mikrokontroller, mis toetab samu andmesideliideseid kui RT Box. Valitav mikrokontroller pidi olema piisavalt kiire, et suhelda üle valitud andmeside kanali ilma kasutajale märgatava viivitusega. Veel oli vaja sidestuda valitud ekraaniga.

2.2 Tehniliste parameetrite määramine

Tehniliste parameetrite määramisel tuli arvesse võtta vaja minevaid erinevaid signaalide ühendusi ja ka toiteühendusi. Joonisel 2.2 on välja toodud RT Boxi ja kasutajaliidese vaheline SPI ühendus, RT Boxi ja laiendusplaadi vaheline elektriline ühendus, mikrokontrolleri ja nupumaatriksi vaheline ühendus GPIO liinil, ning mikrokontrolleri ja ekraani vaheline I2C ühendus.



Joonis 2.2 Kasutaja liidese funktsionaalne skeem

2.2.1 ESP32-S2-Saola-1 arendusplaadi parameetrid

Projektis kasutati ESP32-S2-Saola-1 arendusplaati. Arendus plaati põhilised omadus on välja toodud tabelis 2.1 .

Tabel 2.1 ESP32-S2-Saola-1 arendusplaadi eri komponendid [5]

ESP32-S2-WROVER	Vaata tabel 2.2
Viigupäised	Kõik saadaolevad GPIO-viigud (välja arvatud püsimälu ja PSRAM-i SPI-siin) on eraldatud plaadi viigupäistesse. Kasutajad saavad programmeerida ESP32-S2 kiibi, et võimaldada mitmeid funktsioone, nagu SPI, I2S, UART, I2C, puuteandurid, PWM jne.
3,3 V toite indikaator LED	Lülitub sisse, kui USB-toide on plaadiga ühendatud.
USB-UART-i sild	Üks USB-UART-sillakiip tagab edastuskiiruse kuni 3 Mbps.
Reset-nupp	
Micro-USB port	USB liides. Toiteplokk plaadile, samuti sideliides arvuti ja ESP32-S2 kiibi vahel.
Algladimisnupp	Allaladimisnupp. Kui hoiate all nuppu Boot ja seejärel vajutage nuppu Lähtesta, käivitatakse püsivara allaladimise režiim jadapordi kaudu püsivara allaladimiseks.
Kolme värviline LED	Adresseeritav RGB LED (WS2812), juhib GPIO18.

Arendusplaat on tegelikkuses ainult toetav riistvara. Oma kasulikkuse mikrokontrollerina saab ESP32-S2-Saola arendusplaat ESP32-S2-WROVER moodulist. Mooduli parameetrid on toodud tabelis 2.2.

Tabel 2.2 ESP32-S2-WROVERi parameetrid [5]

Protsessor ja mälu	ühetuumaline 32-bitine LX7 mikroprotsessor, kuni 240 MHz
	128 KB ROM
	320 KB SRAM
	16 KB SRAM RTC-s
WiFi	802,11 b/g/n
	Bitikiirus: 802.11n kuni 150 Mbps
	Töösagedus: 2412 ~ 2484 MHz
Perifeeria	GPIO, SPI, LCD, UART, I2C, I2S, kaamera liides, IR, impulssloendur, LED PWM, TWAI® (ühildub standardiga ISO 11898-1, st CAN Spetsifikatsioon 2.0), täiskiirusega USB OTG, ADC, DAC, puuteandur, temperatuuriandur

	Mooduli integreeritud komponendid
GPIO	43 tk
SPI	4 tk
I2C	2 tk
	40 MHz kristallostsillaator
	4 MB SPI-püsimälu
	2 MB PSRAM
Antenni valikud	Sisseehitatud PCB antenn
Töötingimused	Tööpinge/toiteallikas: 3,0 ~ 3,6 V
	Töökeskkonna temperatuur: -40 ~ 85 °C

Kuna ESP32-S2-WROVER moodul (ESP moodul) vajab tööks 3.0...3.6V on ESP32-S2-Saola-1 arendusplaadil (ESP arendusplaat) toitemuundur, mis teeb 5V toitepingest 3.3V. Selline muundur lubab arendusplaati toita otse arvuti USB pesast.

Meie kasutuses lubab antud muundur kasutada RT Boxi laiendusplaadi 5V väljundpinget, et toita ESP-i arendusplaadil olevat ESP-i moodulit. Väline toitepine ühendati viigu A20 külge (vaata Lisa 1).

Täielik skeem on leitav graafilises osas, lisa 1. Täielikul skeemil on viikude kaupa kõik ühendused graafiliselt näidatud.

Joonisel tähisega „U1” on ESP-i arendusplaat. Tähisega „U2” on LCD ekraanimoodul. Tähisega „J1” on viigupäis, millest lähevad edasi ühendused RT Boxi. Tähisega „SWx”, kus x on täisarv 1...5, on lülitid.

Arendusplaadi ühendused teiste seadmetega on välja toodud tabelis. Viigud, mis pole tabelis mainitud, tuleb lugeda ühendamata jäetuks.

Tabel 2.3 ESP32-S2-Saolo arendusplaadi ühendused

Arendusplaat viigu number	Väline ühendus
A2	LCD ekraani SDA viik
A6	LCD ekraani SCK viik
A7	Nupp 1
A8	Nupp 2
A9	Nupp 3
A10	Nupp 4
A11	Nupp 5
A14	RT Boxi SPI MISO viik
A15	RT Boxi SPI MOSI viik
A16	RT Boxi SPI SLCK viik

A17	RT Boxi SPI cs viik
A20	5V toide
GND	Maapotentsiaal

2.3 Andmesideliidese valimine

Andmeside liidese valikul tuli lähtuda asjaolust, et ressursivaesed mikrokontrollerid ei toeta kõiki andmeside liideseid riistvaraliselt. Valitud mikrokontrollir on riistvaraline tugi I2C, SPI, UART ja muudele liidestele (Vaata tabel 2.2) Kõrgemasemelistele liidestele riistvaraline tugi puudub, ning nende kasutamine nõuab nende interpreteerimist tarkvaraliselt.

Tänapäevastel mikrokontrolleritel on piisav jõudlus, et toetada kõrgetasemelisi andmisede liideseid nagu näiteks HTTP või XML, küll aga tarkvaraliselt. Antud juhul lubab andmete lihtsus vältida keerulisemaid andmeside liideseid.

Otsustati SPI andmeside liidese kasuks, selle lihtsuse tõttu.

3. Programmi koostamine

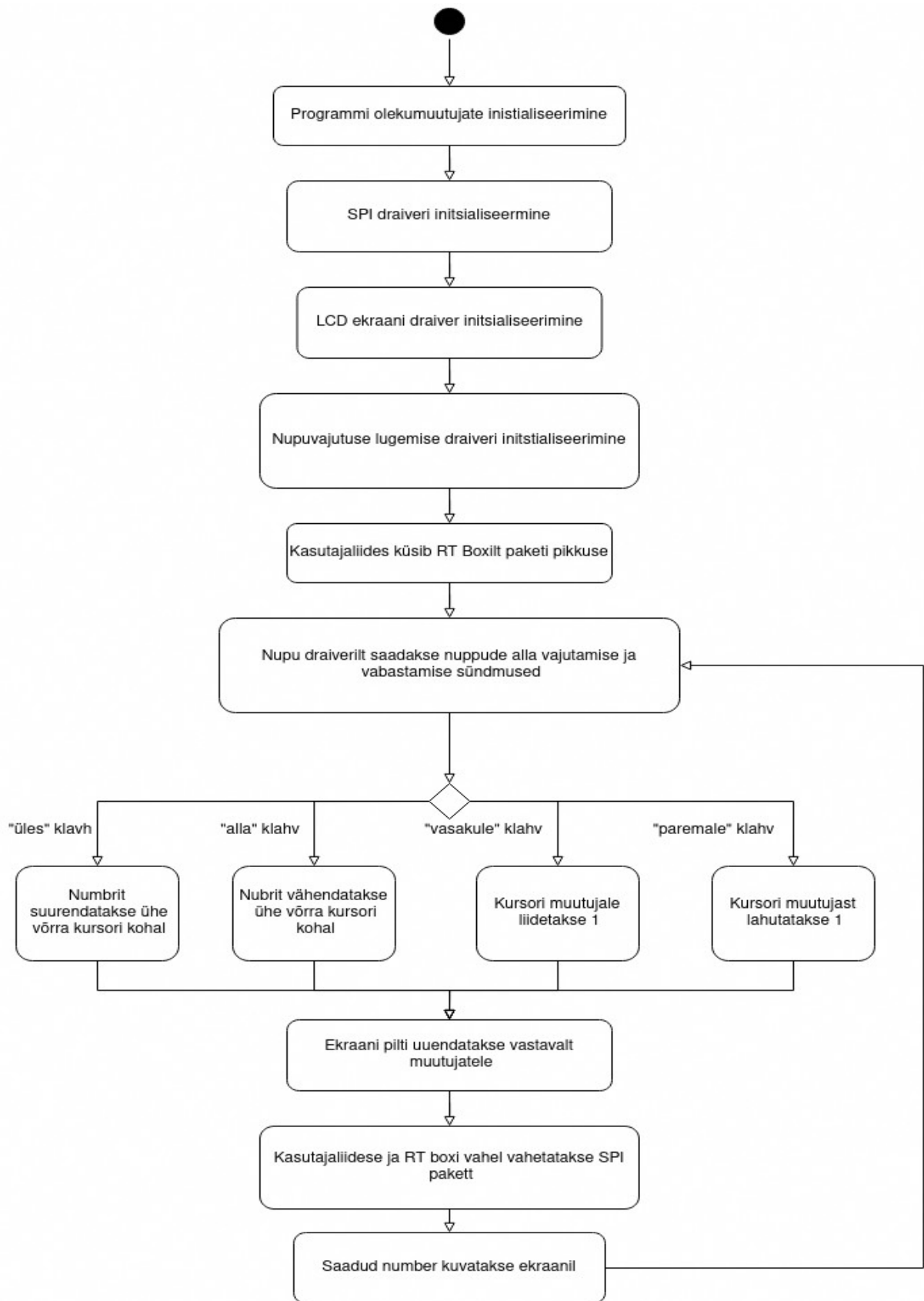
Programmi koostamine toimus nvim käsurea tekstiredaktori abil. Nvim on edasiarendus tuntud vim tekstiredaktorist.

Programm põhineb Espressifi oma tarkvaraarenduskomplekt, esp-idf, mis hõlbustas koodi kirjutamist. Esp-idf annab esp32 mikrokontrollirile FreeRTOS operatsioonisüsteemi ja palju kasulikke mooduleid, mis kiirendavad arendus protsessi. Kõikide kriitiliste ülesannete täitmiseks leidis esp-idf ja espressifi laiendatud moodulite andmebaasis moodul, et neid ülesandeid täita.

Piisas erinevate moodulite kokku koondamisest ja ei olnud tarvis kirjutada madalataseme masinkoodi, et kasutada riistvara komponente. Moodulite dokumentatsioon oli enamasti inglise keelselt kätte saadav, kuid esines ka dokumentatsiooni vigu, kus oli ainult hiinakeelne variant.

Koodi kompileerimine toimub samuti käsurealt. Esp-idf sisaldab kõiki vajalikke osi, et kompileerida kood sihtriistvarale, laadida kood mikrokontrolleri püsिमällu ja jälgida mikrokontrolleri ja arvuti vahelist suhtlust.

Programmi töö käib vastavalt algoritmile, mida on näha joonisel 3.1. Programm alustab oma käiku olekumuutujate, LCD draiveri, SPI draiveri ja nupuvajutuste lugemise draiveri initsiliseerimisega. Seejärel saab kasutaja liides RT Boxilt paketi sõnumi pikkusega. Edasi järgneb töötsükkel, kus kogutakse nuppude vajutamise sündmused, ning selle põhjal tehakse otsus, kas numbrit suurendada või vähendada. Või veel liigutada kursorit. Pärast seda uuendatakse ekraani pilt. Järgminisena toimub andmevahetus RT Boxi ja kasutaliidese vahel üle SPI ja saadud numbrid kuvatakse ekraanil. Tsükkel naaseb olekusse, kus algab uuesti nuppude lugemine.



Joonis 3.1 Programmi algoritm

3.1 Andmeside liidese tööpõhimõtted

Jadaandmesideliidese (SPI) füüsiliseks kihiks on neli juhet

SCLK – Serial Clock

MISO – Master In, Slave Out

MOSI – Master Out, Slave In

CS – Chip select

SPI liidese signaalide nimed ei ole standardiseeritud. Näiteks SCLK võib esineda andmelehtedes kui CLK ja CS kui SS. Edaspidi kasutatakse signaalide nimedeks eelmainitud tähised.

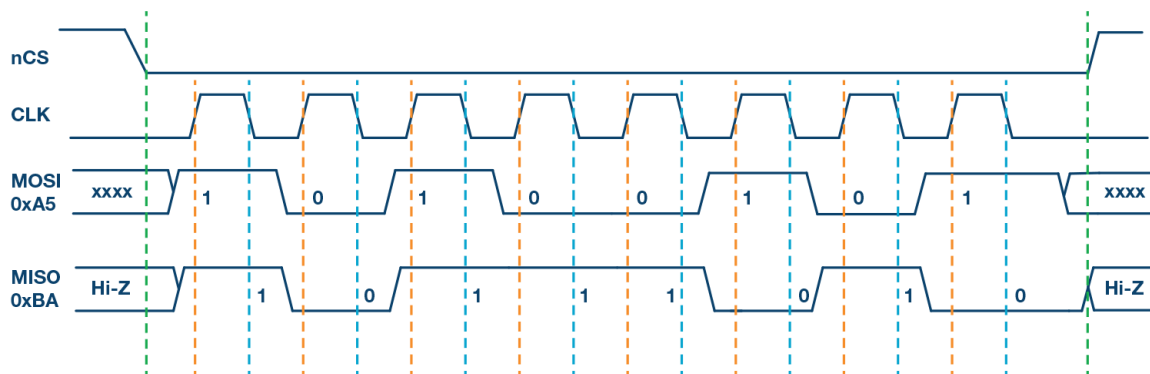
Paljud mikrokiibid toetavad ka paralleelset andmeedastust üle SPI, kaasaarvatud esp32 ja RT Box. Keerukuse vähendamiseks jäädi ühe andmeedastus kanali juurde. Antud olukorras on andmemahud võrdlemisi väikesed, ning mingisugust olulist eelist paralleelne andmeedastus ei annaks.

SPI-l on mitu võimalikku seadistust: Signaali ajastuse ja põhine Rolli põhine

SPI suhtluses on kaks rolli: master ja slave. Suhtluses võib olla mitu slave seadet, aga ainult üks master. Antud juhul on ainult kaks seadet, millega välistame olukorra, kus on mitu slave seadet. Suhtluse alustamiseks muudab master CS liini olekut. Üldjuhul annab master CS liiniline loogilise 0 väärtuse. Selle järel hakkab master liigutama SCLK liini, et anda edasi kellasignaali. SCLK liikumise ajal saavad nii slave kui master edastada andmeid, isegi samaaegselt. Sellist andmeside tüüpi nimetatakse täis-duplex suhtluseks.

Master edastab oma andmed üle MOSI liini ja slave MISO liini. Olenevalt kasutusel olevast ajastuse režiimist interpreteeritakse MISO ja MOSI liinide liikumist ühtedeks ja nullideks.

On olemas 4 erinevat ajastuse režiimi [9]. Autor valis kasutada režiimi 0,0, mis toimib järgnevalt :



Joonis 3.2 SPI režiim 0 visualiseering [9]

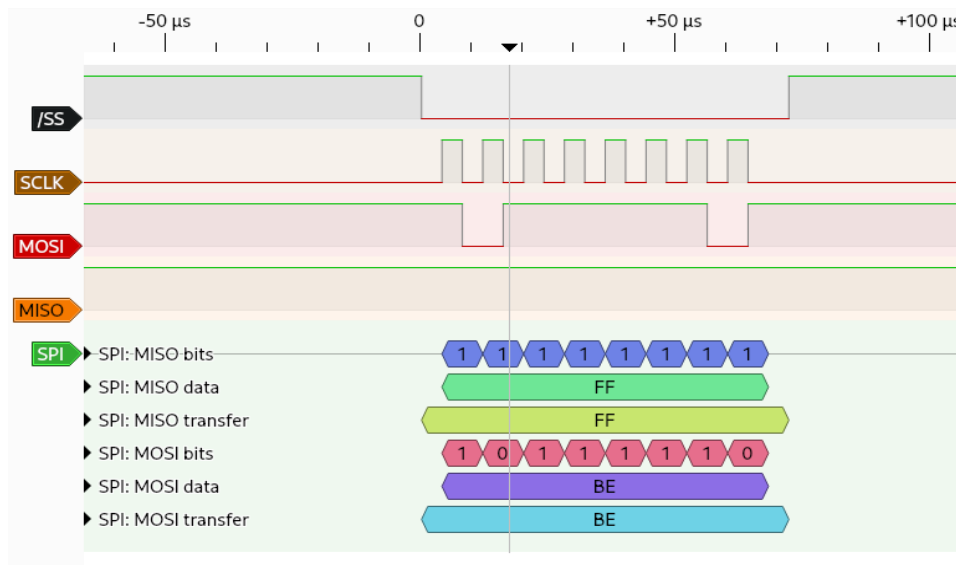
Kellasisignaali tõusmise hetkel loeb slave MOSI liinilt väärtuse, ning salvestab selle. Kellasisignaali langemise hetkel loeb master MISO liini väärtuse ja salvestab selle.

3.2 Täisarvuliste andmete saatmine

Täisarvuliste numbrite saatmine on PLECSis hästi toetatud. PLECS SPI Slave blokid toetavad „uint8_t” ja „uint16_t” saatmist ja vastuvõtmist. „uint8_t” on ühe baidi laiune positiivne täisarvu muutuja andmetüüp, millega saab esitada numbreid 0...255 ja „uint16_t” on kahe baidi laiune positiivne täisarvu muutuja andmetüüp, millega saab esitada numbreid 0...65535. Simulatsiooni sisesed uijupunkarvud konverteeritakse täisarvudeks läbi andmetüübi bloki. Bloki kirjelduses on need märgitud „Bits per word” valiku all.

SPI blokk toetab nii ühe kui ka kahe baitised sõnepikkused. Maksimaalne sõnede arv on 127, mis teeb maksimaalseks andmeedastus ühikuks 252 baiti, kui kasutada kahe baidist sõnepikkust. Antud juhul on 2 baiti piisav pikkus, mis vastab siis uin16_t.

SPI suhtluse pealt kuulamiseks kasutati Saleallogic 24MHz 8CH loogikaanalüsaatorit. Koostöös Pulseview tarkvaraga sai visualiseerida esp32 ja RT Boxi vahelist suhtlust graafiliselt joonisel 3.3. Joonisel olev /SS tähisega liin on sama mis CS.



Joonis 3.3 Kuvatõmmis Pulseview tarkvarast

Antud juhul saatis master (ESP-i moodul) kuueteistkümnendik süsteemis numbri 0xBE ja slave (RT box) saatis numbri 0xFF.

3.3 Andmevahetuse protokoll üle SPI liidese

RT boxi ja kasutajaliidese vaheline suhe toimib järgnevalt.

Esimese sammuna küsib kasutajaliides RT Boxilt ühe sõne, mis on kahe baidi laiune, milleks on tulevase andmevahetuse sõnede arv. Tähisega „DNC” on tähistatud väärtused, mida andmevahetuse protokollis ei kasutata, kuid siiski edastatakse.

Tabel 3.1 Esimene andmevahetuse pakett

	Bait 0
MOSI	DNC
MISO	Sõnede arv

Teise sammuna jätkub kahesuunaline andmevahetus, kus RT box võib saata andmeid samal ajal kui kasutajaliides saadab andmeid RT boxile. Sõnede arvuks on n.

Tabel 3.2 Edasine RT Boxi ja kasutaliidese vaheline suhtlus

	Bait 0	Bait 2	Bait 4	Bait 6	Bait 8	Bait 2n
MOSI	DNC	DNC	DNC	DNC	DNC	Kasutajaliideselt valitud kahe baidi laiune
MISO	Sõnede arv	Esimene tähesümbol	Numbriline täisarv väärtus	Teine tähesümbol	Kolmas tähesümbol	DNC

Viimane tsükkel, kus kasutajaliides saadab välja kasutaja poolt valitud väärtuse, võib juhtuda ükskõik mis hetkel, välja arvatud baidil arvuga 0. See tähendab, et sõnumite pikkus peab olema pikem kui üks sõne.

3.4 Nupuvajutuste lugemine

Nupuvajutuse lugemiseks piisab otse mikrokontrolleri GPIO viigu ühendamisest nupuga. Antud nupul on küll neli viiku, aga tegemist on ühepooluselise lülitiga. Elektriline ühendus on näidatud lisas 1.

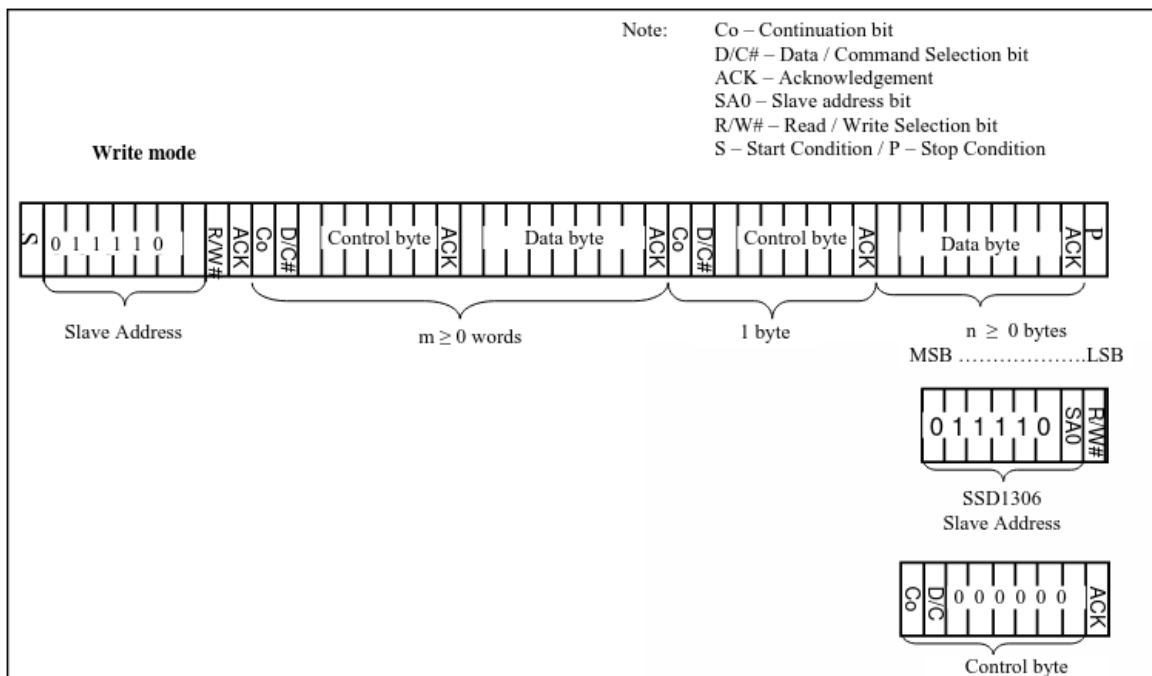
Nupuvajutuse tuvastamise skeemis tegi lihtsaks asjaolu, et enamuse ESP32 GPIO viikudel on sisemiselt olemas ülestõmbe (pull-up) takistid [10]. Selle puudmisel oleks olnud tarvis sidu sisendi viik toitepingega läbi välise takisti.

Hetkel kui mikrokontroller viigul langeb nulli lähedale, siis registreeritakse nupuvajutus. See on võimalik, sest mikrokontrolleri sisene ülestõmbe takisti ei suuda toitepinget hoida, kui viik on lühistatud maapotentiaaliga.

3.5 LCD tööpõhimõte

LCD moodul võtab vastu käsklusi üle I2C suhtlusprotokolli. Suhtlust alustab master seade, milleks siin juhul on ESP32. Pärast start biti saatmist saadab master soovitud slave-i 7-bitise aadressi. Selle sõnumi peale vastab saadetud aadressiga slave ACK bitiga. Pärast ACK-i vastuvõtmist saadab master ühe baidi, mis üldjuhul on registri valik, aga selle LCD puhul nimetatakse, kui „Control Byte“, ehk kontrollbait. Selle peale vastab slave uuesti ACK-iga. Peale ACK-i saadab master uue baidi, mis peaks

sisaldama andmeid. Seda baiti saab nimetada kui andmebaiti. Selline kontrollbaidi ja andmebaitide jada võib jätkuda veel sadu kordi, kuni terve pilt on edastatud ekraanile [11]. Sellist kirjutamist iseloomustavat andmevahetust kujutab joonis 3.4.



Joonis 3.4 Ekraani suhtlusprotokoll [11]

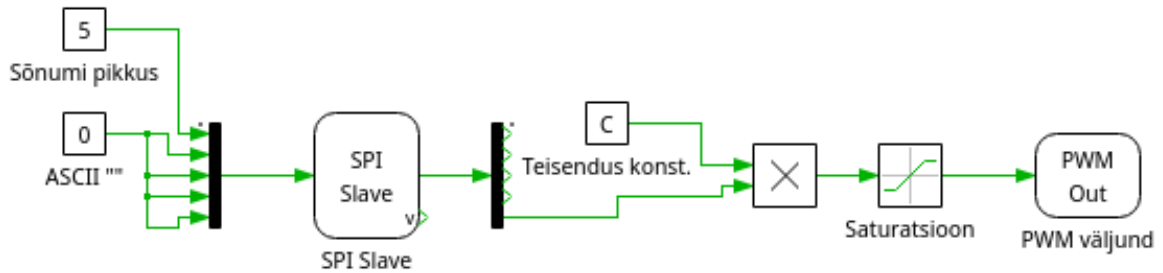
LCD ekraani I2C viigud sai ühendada otse ESP-i arendusplaadil sobivatele viikudele. Enamamus ESP32 viikudest toetavad I2C suhtlust. Toiteviigu sai ühendad otse viie voldi toitesse (vaata lisa 1).

3.6 Programm RT Boxis

Selleks, et mikrokontrolleril oleva programmiga suhelda oli vaja koostada PLECS keskkonnas programm. Skeemid oli vaja (vaata jooniseid 3.5, 3.6, 3.7 ja 3.8) luua, et sooritada erinevevaid mõõtmisi, mis demonstreeriks kasutajaliidese võimekust.

3.6.1 Muunduri käivitamise ja seiskamise skeem

Muunduri käivitamise ja seiskamise demonstreerimiseks koostati lihtsustatud PLECSi mudel, mida on näha joonisel 3.5

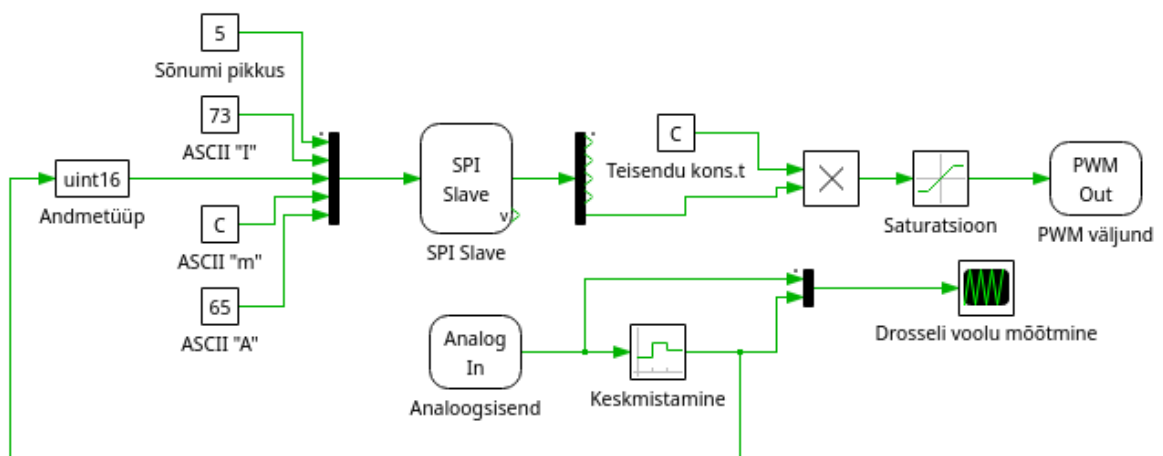


Joonis 3.5 Lihtsustatud skeem PLECSis

Skeemil on SPI slave blokk, mis kasutab RT box sisendeid 0-2 ja ühte väljundit 0. SPI Slave bloki sisend on vektor, kus on viis väärtust. Esimeseks väärtuseks on sõnumi pikkus, milleks on viis sõne. Ülejäänud neli väärtus on antud hetkel nullid, et kuvada kasutajaliidesel tühja rida. Bloki väljund on samuti vektor, milles on 5 väärtust. Esimesed neli väärtust on nullid, mis saadetakse ainult selleks, et saada õige sõnumi pikkus, milleks on 5. Viimane väärtus on kasutajaliideselt saadud täitetegu, mis on 100 kordne täitetegur. Saadud väärtus liigub edasi tehte blokki, mis on tähistatud „X”, kus see jagatakse teisenduskonstandiga, milleks on 100. Edasi liigub väärtus küllastus blokki, kus seatakse maksimaalne täitetegur. Lõplik väärtus liigub PWM väljundblokki, kus genereeritakse PWM signaal väljundis 26.

3.6.2 Drosseli voolu mõõtmise skeem

Drosseli keskvärtuse mõõtmiseks sai skeemi täiendatud (vaata joonis 3.6). SPI Slave bloki sisendvektorisse suunduvad konstandid on nüüd ASCII tähemärkide väärtustega. Teine sümbol tähistab ASCII "I" tähte, kolmas „m” ja neljas „A”. Need on vajalikud, et kuvad ekraanil voolu väärtust kujul „I xmA”, kus x on voolu keskvärtus.



Joonis 3.6 PLECS-i skeem drosseli voolu mõõtmiseks

Voolu väärtus saadakse RT boxi analoogsisendist, kuhu on ühendatud ampertangid. Ampertangidest seati 100mV/A piirkonda, ja nende pinge väärtus teisendatakse vooluks analoogsisendi bloki sees valemiga:

$$I = U \cdot 10^5 \quad (3.1)$$

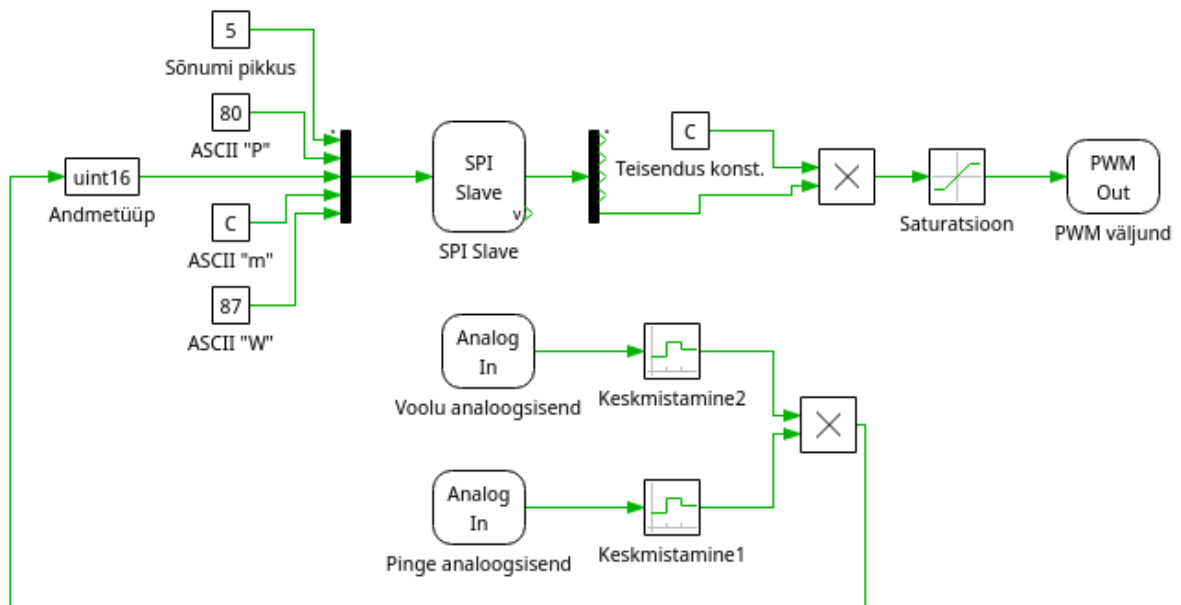
Kus I – Mõõdetav vool, mA

U – Pinge ampertangide väljundil, mV

Analoogsisendist väljuv voolu väärtus saadetakse scope blokki ja keskvaartuse arvutamise blokki, kus toimub keskvaartuse arvutamine lugemitest ühe sekundi jooksul. Nii analoogsisendi kui keskvaartuse blokki väljund suunatakse scope blokki

3.6.3 Muunduri sisendvõimsuse mõõtmise skeem

Muunduri sisendvõimsuse arvutamiseks tuli skeemi täiendada teise analoogsisendiga. PLECSi skeem koostati joonise 3.7 järgi. Ampertangid mõõtsid drosselit läbivat voolu sarnaselt eelnevale katsele.



Joonis 3.7 PLECSi skeem muunduri sisendvõimsuse mõõtmiseks

Teises analoogsisendis mõõdeti sisendpinget otse toite allika klemmidelt. Toiteallika pinget oli 10V. Mõlema mõõtmise keskvaartused liikusid edasi korrutamise blokki, kus väärtused korrutati valemi järgi:

$$P = I \cdot U \quad (3.2)$$

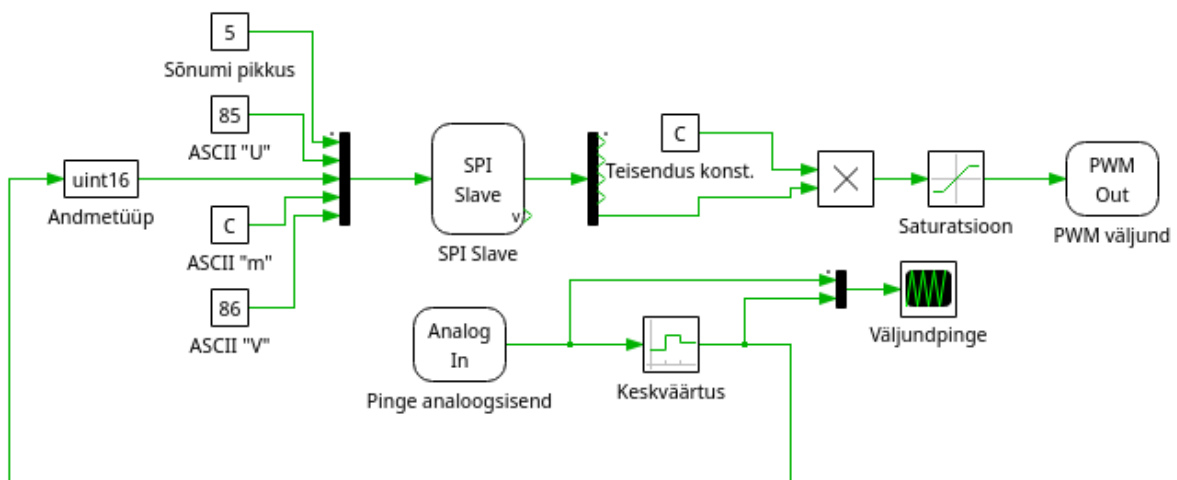
Kus P – sisendvõimus, W

I - drosselit läbiv vool, A

U -n sisendpinge, V

3.6.4 Muunduri väljundvõimsuse mõõtmise skeem

Muundur väljundpinge mõõtmiseks sai PLECSi skeemis kasutatud ainult RT Boxi analoogpinge sisendit. Skeem seati ümber vastavalt joonisele 3.8.



Joonis 3.8 PLECS-i skeem muunduri pinge keskväärtsuse mõõtmiseks

Pinge loetakse analoogisendist läbi pingejaguri, mis koosneb 20kΩ ja 2kΩ takistist. Mõõdetava pinge saab leida järgneva valemiga:

$$U_{\text{väljund}} = \frac{U_{\text{jagur}} \cdot R_1}{R_2 + R_1} \quad (3.3)$$

Kus R_1 – Takisti 1, Ω

R_2 – Takisti 2, Ω

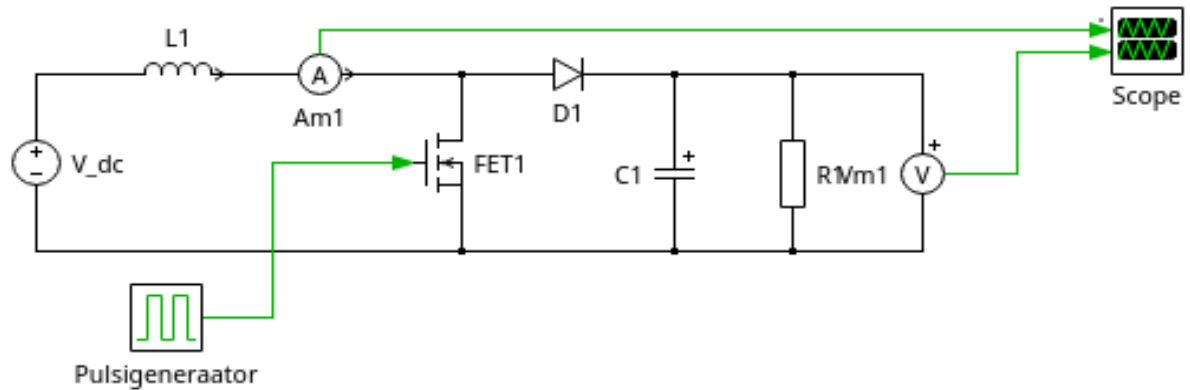
$U_{\text{väljund}}$ - Muunduri väljundpinge, V

U_{jagur} - R1 ja R2 kahe vahel oleva sõlmpunkti potentsiaal

R1 ja R2 väärtused on 20kΩ ja 2kΩ.

4. Kasutajaliidese testimine ja analüüs

Testimiseks koostas autor kõrgendav pingemuundur, et katsetada kasutajaliidese kahe-suunalist suhtlust. Muundur ehitati Tallina Tehnika Ülikooli laboris, eelnevalt uurimisrühma poolt välja arendatud makettplaadile. Muunduri skeem on näha joonisel 4.1.



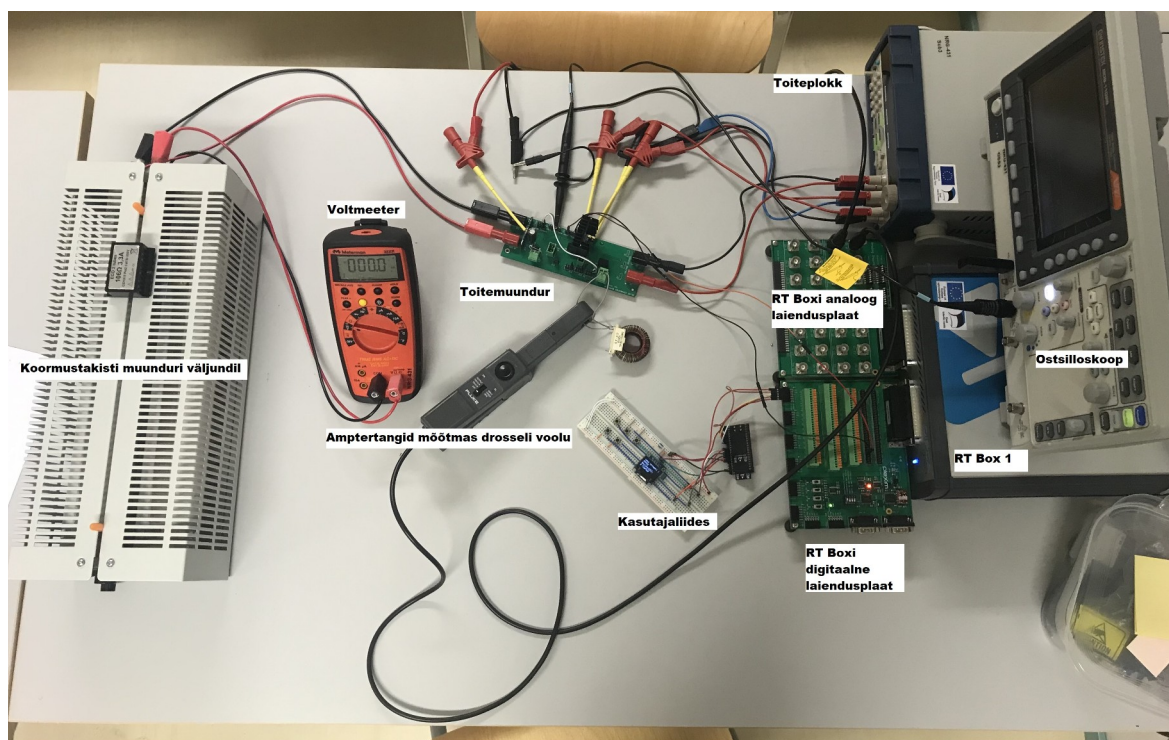
Joonis 4.1 Kõrgendava muunduri skeem

Skeem koostati komponentidest, mis on leitavad tabelist 4.1.

Tabel 4.1 Muunduri komponentide nimistu

Komponent	Parameeter	Tähis joonisel 4.1
Transistor	IRL540N	FET1
Diood	HER303G	D1
Induktor	325 μ H	L1
Kondensaator	22 μ F	C1
Koormus	26 Ω	R1

Pulsigeneraatori rolli täidab katseskeemis RT Boxi digitaalväljund. Ampermeetri ja voltmeetri rolli täidavad RT Boxi analoogsendid.

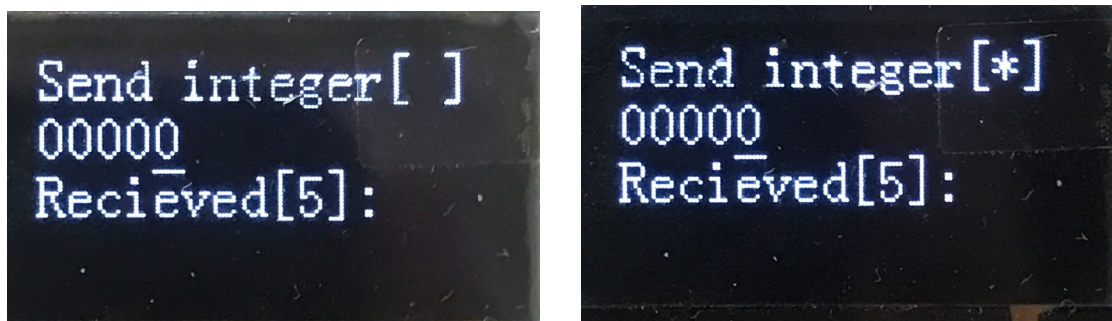


Joonis 4.2 Foto katsestendist

Katsekeem kasutas eraldi juhitavat kolme allikaga toiteplokki, ostsilloskoopi, voltmeetrit, Fluke 80i-101s ampertange, RT Box 1, RT Boxi digitaallaiendusplaati, RT Boxi analooglaiendusplaati ja ka antud töö eesmärgiks arendatud kasutajaliidest. Katsestend on näha joonisel 4.2. Kasutajaliidese katsetamiseks koostati skeem maketplaadile.

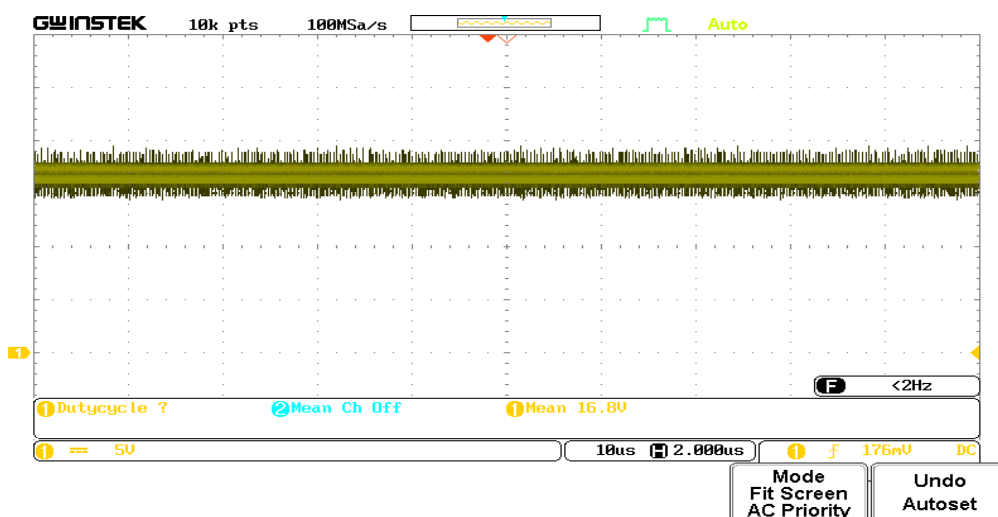
4.1 Muunduri käivitamine/seiskamine kasutajaliidese kaudu.

Muundur käivitati kasutajaliidesele nupuvajutuse teel. Kasutaja liidese ekraanil (vaata jooniseid Error: Reference source not found 4.3) on näha esimesel real, kas sõnumi saatmine toimub. Sõnumi saatmise lubamist tähistab „*” sümbol. Teisel real on saadetak number ja kursor numbri valimiseks. Kasutajaliidese kolmandal real on näha mudelist saadud sõnumi pikkus, milleks on antud olukorras viis.

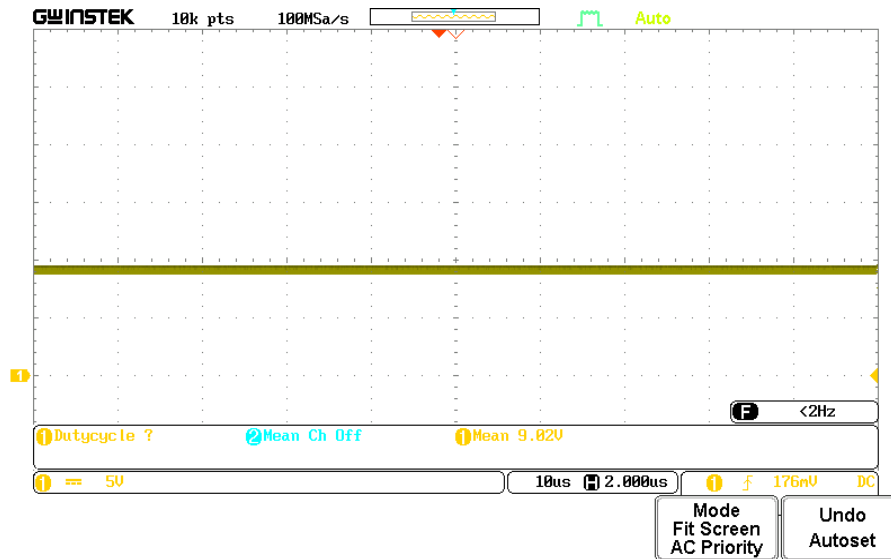


Joonis 4.3 Kasutaliides seisatud muunduri režiimis (a), kasutajaliides töörežiimis (b)

Kasutajaliidesel sai seadistatud saadetavaks numbriks 50, mis tähistab täitetegurt 0.5. Sisse ja väljalülitamisel on näha muunduri väljundpinge muutumist algsest 9 V kõrgemale 16 V (vaata jooniseid).



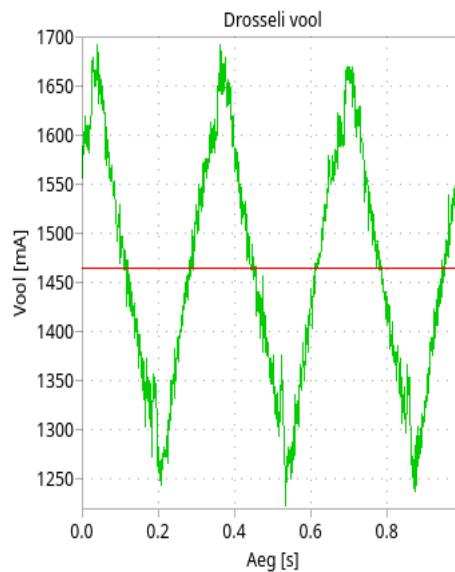
Joonis 4.4 Muunduri pinge väljund käivatatud olekus



Joonis 4.5 Muunduri väljund seisatud olekus

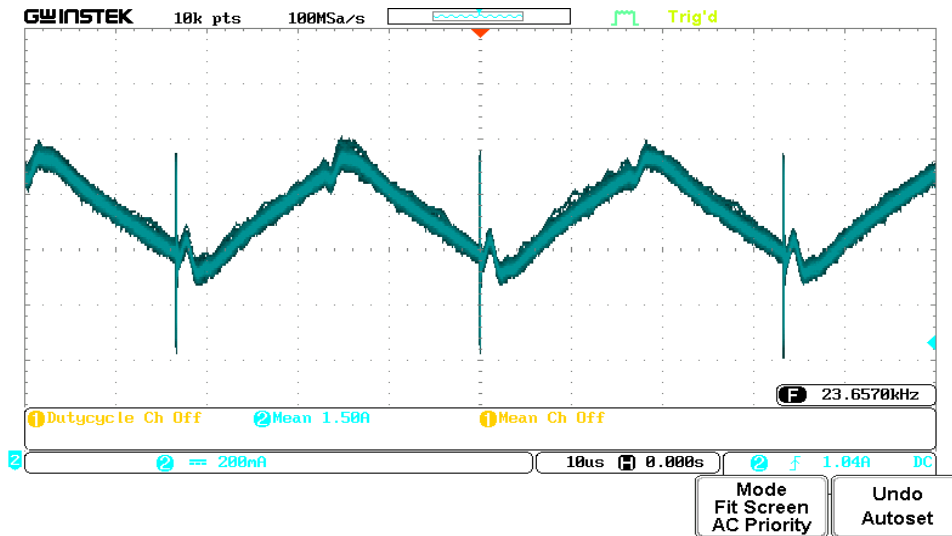
4.2 Drosseli voolu keskväärtuse mõõtmine

Drosseli voolu mõõtmise käigus, seati täitetegur väärtusele 0.5. Seejärel muundur käivitati kasutajaliidese abil. PLECSi scope blokkis saadi drosselit läbiv vool, mille visualiseerinut on näha joonisel 4.6. Keskväärtuseks saadi 1,46 A.



Joonis 4.6 Drosselit läbiv vool

Võrdluseks korraldati mõõtmist ostsilloskoobiga (vaata joonis 4.7), millega saadid tulemuseks 1,50 A.



Joonis 4.7 Drosselit läbiv vool ossilloskoobist mõõdetuna

PLECSis saadud tulemus, mida on näha joonisel 4.8, kuvati kasutaliidesel, milleks on 1460 mA.

```
Send integer[*]
00050
Recieved[5]:
I 1460mA
```

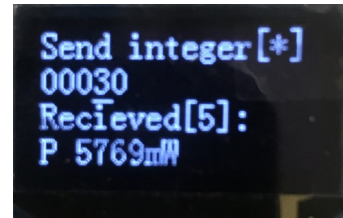
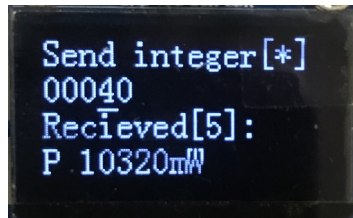
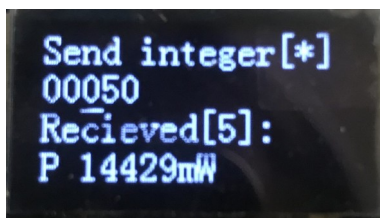
Joonis 4.8 Kuvatõmmis kasutajaliidest drosseli voolu mõõtmise hetkel

4.3 Muunduri sisendvõimsuse arvutamine

Sisendvõimsuse arvutust katsetati erinevate katsetegurite juures, mida on näha joonisel 4.9, millest saadud tulemused on tabelis 4.2. Saadud tulemused arvutati valemi 3.5.3 abil.

Tabel 4.2 Muunduri sisendvõimsus erinevate täitetegurite juures

Täitetegur	Sisendvõimsus [mW]
0.5	14429
0.4	10320
0.3	5769

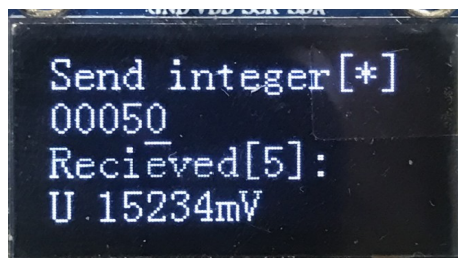


Joonis 4.9 Esimene sisendvõimsuse mõõtmine (a), teine sisendvõimsuse mõõtmine (b), kolmas sisendvõimsuse mõõtmine(c)

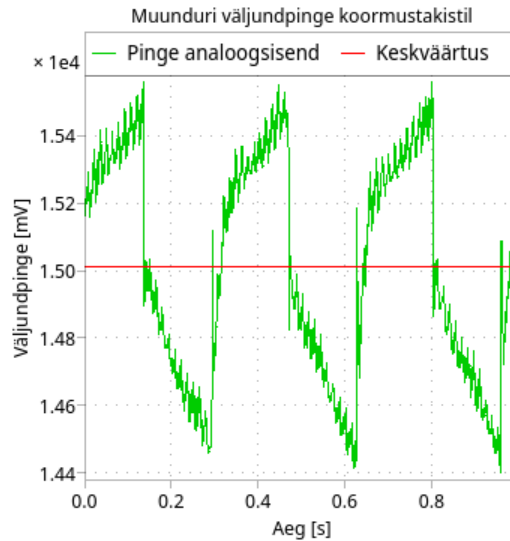
4.4 Muunduri väljundpinge keskväärtuse mõõtmine

PLECSIS oli sai näha pinge muutumist ajas ja ka pinge keskväärtust millivoltides (vaata joonis 4.11).

Sama tulemust on ka näha kasutajaliidesel (vaata joonis 4.10), kus on ka näha väljundpinge täpsem keskväärtus, milleks on 15234 mV.



Joonis 4.10 Kuvatõmmis kasutajaliidese pinge keskväärtuse kuvast



Joonis 4.11 Muunduri väljundpinge PLECS-sist mõõdetuna

Väljundpinge mõõdeti ka voltmeetriga, ning selle tulemus on välja toodud tabelis 4.3.

Tabel 4.3 Kasutajaliidese ja voltmeetri mõõtetulemused

Voltmeeter	Kasutajaliides
16,80 V	15,234 V

4.5 Tulemuste analüüs

Katsetuste tulemusena valmis esialgne katseseade, mille eesmärk oli tõestada kasutajaliidese võimekust.

Muunduri sisse ja välja lülitamise katse tulemus demonstreeris kasutajaliidese suhtluse võimekust suunal kasutajaliidese RT Boxi. Nupu vajutuse peale seiskus ja käivitus toitemuundur.

Drosseli voolu mõõtmise katse tulemus näitas täpsust mida on kasutajaliidese võimalik saavutada. Ja veel demonstreeris suhtluse võimekust RT Boxist kasutajaliidesele.

Muunduri sisendvõimsuse mõõtmise katse demonstreeris võimekust teha arvutusi PLECSi simulatsioonis ja nende arvutuste tulemuste kuvamise võimekust kasutajaliidesele.

Muunduri väljundpinge mõõtmine näitas samuti edukalt kõike eelnevat, aga andis ka võrdleva tulemuse teise mõõtevahendiga. Kuigi kasutajaliidese tulemus erines voltmeetri omast, jääb RT Boxi kui mõõteriista hindamine antud töö ulatusest välja.

Kasutajaliides täitis kõiki eeldatavaid nõudeid. Antud katse tulemust loetakse edukaks.

KOKKUVÕTE

Eesmärgiks oli projekteerida ja ehitada ESP32 mikrokontrolleri baasil kasutajaliides jõuelektroonika riistvarasimulaatorile RT box. Kasutajaliidese projekteerimine oli edukas. Mõlema seadme lähteandmed olid tootjate poolt lihtsasti kättesaadavaks tehtud, mis soodustas eesmärkide saavutamist.

Programmi koostamise lihtsustas tootjapoolt pakutava tarkvara, erinevad tarkvara moodulid ja teegid, ning hea dokumentatsioon. Programmi koostamisel sooviti kasutada võimalikult palju olemasolevaid tükke, et vähendada töö mahtu ka keskenduda probleemi lahendamisele. Testimine ja testitulemuste analüüs näitas, et seade täidab püstitatud nõudeid ja eesmärgi. Kasutajal on võimalik reaajas näha simulatsiooniparameetreid ja juhtida simulatsiooni kulgu.

Lõputöö sisse ei mahtunud veebipõhise liidese arendamine aja nappuse tõttu. Selline edasiarendus oleks antud töö järgmine samm. Riistvara sai välja valitud eeldusega, et veebipõhine liides oleks teostatav. Samuti ei mahtunud samuti ujupunkt arvude saatmine. Ujupunktarvude saatmine ei ole toetatud PLECSi tarkvara blokkides. Küll aga on võimalik implementeerida suhtlusskeem, kus edastatud numbrit käsitletakse kui ujupunktarvu. Sõnum peab koosnema viiest baidist. Üks bait, et näidata kas tegemist ujupunktarvuga või täiskohalise arvuga. Ning kui kasutada IEEE 754 standardile vastavad 32-bitilist ujupunktarvu, siis veel lisaks neli baiti. Saatmist raskendab asjaolu, et IEEE 754 standardi järgsi numbri komponendid ei lange täpselt baidi piiridesse (vaata lisa 2). Veel ei mahtunud lõputöö sisse kasutajaliidesele trükkplaadi projekteerimine.

SUMMARY

The aim of this thesis was to design and build a user interface for a RT box power electronics hardware simulator based on the ESP32 microcontroller. The design of the user interface was successful. The data of both devices was easily accessible from the manufacturers, which facilitated the achievement of the objectives.

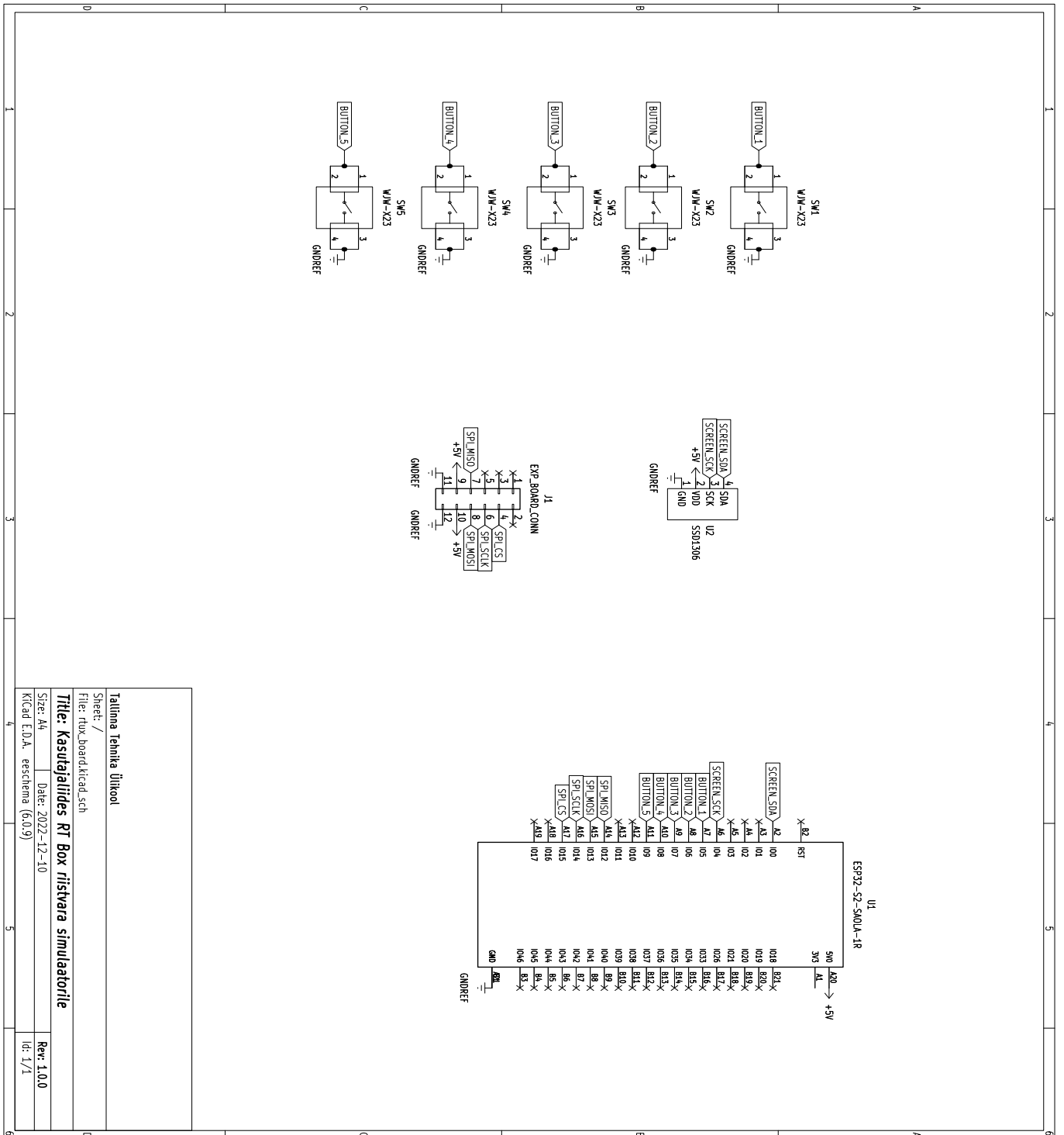
The software offered by the manufacturer, various software modules and libraries, and good documentation facilitated the development of the program. During the development of the program, it was desired to use as many existing pieces as possible in order to reduce the workload and focus on solving the problem. Testing and analysis of the test results showed that the device meets the requirements and objectives. The user can see the simulation parameters in real time and control the simulation process.

Due to time constraints, the development of a web-based interface was not included in the final thesis. Such an extension would be the next step for this work. The hardware was selected with the assumption that a web-based interface would be feasible. Also, the transmission of floating point numbers was not included in the final thesis. The transmission of floating point numbers is not supported in the PLECS software blocks. However, it is possible to implement a communication scheme where the transmitted number is treated as a floating point number. The message must consist of five bytes. One byte to show whether it is a floating point number or a full number. And if using a 32-bit floating point number according to the IEEE 754 standard, then an additional four bytes. The transmission is complicated by the fact that the components of the number following the IEEE 754 standard do not fall exactly within the limits of the bytes (see appendix 2). The design of a printer board for the user interface was also not included in the final thesis.

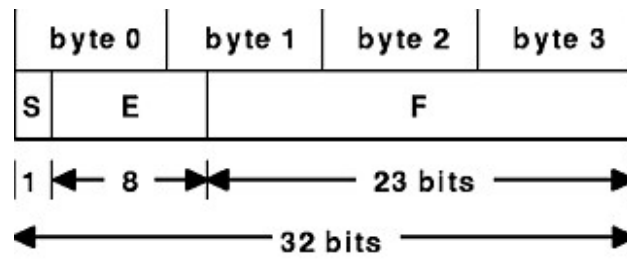
KASUTATUD KIRJANDUSE LOETELU

- [1] dSPACE GmbH, „Solutions&Products,” [Võrgumaterjal]. Saadaval: <https://www.dspace.com/en/pub/home/applicationfields.cfm>
- [2] RTDS Technologies Inc., „HIL testing for renewable, resilient, and reputable grids,” [Võrgumaterjal]. Saadaval: <https://www.rtds.com/applications/microgrids-renewable-energy/>
- [3] NATIONAL INSTRUMENTS CORP., „What is VeriStand,” 2022 [Võrgumaterjal]. Saadaval: <https://www.ni.com/en-us/shop/data-acquisition-and-control/application-software-for-data-acquisition-and-control-category/what-is-veristand.html>
- [4] OPAL-RT TECHNOLOGIES INC. , „RT-LAB,” 2022 [Võrgumaterjal]. Saadaval: <https://www.opal-rt.com/software-rt-lab/>
- [5] Plexim GmbH, „RT Box,” [Võrgumaterjal]. Saadaval: https://www.plexim.com/products/rt_box
- [6] Advanced Micro Devices, Inc, „Product brief Zynq-7000,” 2022 [Võrgumaterjal]. Saadaval: <https://www.xilinx.com/content/dam/xilinx/support/documents/product-briefs/zynq-7000-product-brief.pdf>
- [7] Arm Limited, „Cortex-A9,” 2022 [Võrgumaterjal]. Saadaval: <https://developer.arm.com/Processors/Cortex-A9>
- [8] Plexim GmbH, „RT Box Digital Breakout Board User Manual,” May 2020 [Võrgumaterjal]. Saadaval: <https://www.plexim.com/sites/default/files/digitalbreakoutmanual.pdf>
- [9] Piyu Dhaker, „Introduction to SPI Interface,” 2018 [Võrgumaterjal]. Saadaval: <https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html>
- [10] Espressif Systems, „ESP32 Series Datasheet,” [Võrgumaterjal]. Saadaval: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [11] SOLOMON SYSTECH, „SSD1306 Advance Information,” 2008 [Võrgumaterjal]. Saadaval: https://components101.com/sites/default/files/component_datasheet/SSD1306-OLED-Display-Datasheet.pdf
- [12] IBM, „Single-Precision Floating Point,” 2022 [Võrgumaterjal]. Saadaval: <https://www.ibm.com/docs/en/aix/7.2?topic=types-single-precision-floating-point>

LISAD



Lisa 1: Kasutajaliidese elektriskeem



Single-Precision Floating Point

Lisa 2: 32-bitine ujuduntarv [10]