

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Karl-Heinrich Tamm 206238IACB

Videoloengute vaatamist lihtsustava seadme väljatöötamine

Bakalaureusetöö

Juhendaja: Kaiser Pärnamets

lektor

Tallinn 2023

Autorideklaratsioon

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

Autor: Karl-Heinrich Tamm

13.05.2023

Annotatsioon

Lõputöö eesmärgiks oli luua seade, mis lihtsustab videoloengute vaatamist veebibrauseris. Lahenduse kasutaja saab juhtida seadme abil veebibrauseri laiendust, et muuta video esituskiirust, kerida videot edasi ning tagasi ja kasutada video järjehoidjaid.

Lõputööga lahendatakse peamiselt kahte probleemi. Esiteks, pika videoloengu puhul võib vaatajal kuluda liigselt aega videost otsitava informatsiooni leidmiseks. Teiseks, autori hinnangul puudus piisavalt hea moodus videole järjehoidjate lisamiseks, et vaataja saaks videos mugavalt tagasi minna teda huvitava teema juurde.

Lõputöö eesmärgi saavutamiseks valmistati juhtnuppe sisaldav seade. Lisaks arendati seadmele püsivara koos arvutis töötava seadmedraiveriga, mis suhtleb loodud veebibrauseri laiendusega video taasesituse ning järjehoidjate haldamiseks. Seadmedraiveri paigaldamiseks ja laienduse seadistamiseks loodi tarkvarapaigaldaja.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 38 leheküljel, 5 peatükki, 24 joonist.

Abstract

Developing a Gadget to Facilitate Watching Recorded Lectures

The aim of this thesis was to design and develop a gadget to facilitate watching recorded lectures in a web browser. Using the gadget, a user can interact with a web browser extension in order to control a video playback speed, rewind and fast forward a video and use video bookmarks.

The thesis focuses on two problems. Firstly, a user might have to spend more time than needed for finding information from a recorded lecture that is long. Secondly, the author believes that there wasn't a sufficient way to add bookmarks to video in order to be able to easily return to the topic of interest.

To achieve the goal of the thesis, the author developed a gadget that has mechanical switches and a knob (incremental rotary encoder) for the user interface. The gadget also features an Arm Cortex-M0 based microcontroller and a USB Type-C connector for communication and power delivery. The microcontroller firmware is written in C, and it communicates with the device driver that runs on the computer and is written in Java. The driver exchanges messages between the gadget and a browser extension. The browser extension is written in TypeScript and uses React for the bookmark's graphical user interface. In addition, the author created a Qt based installer to install the driver and to configure the browser extension.

The thesis is in Estonian and contains 38 pages of text, 5 chapters, 24 figures.

Lühendite ja mõistete sõnastik

API (<i>application programming interface</i>)	Tarkvarapõhine liides tarkvara kasutamiseks
ESD-kaitse	Tehnoloogiline lahendus, et vähendada staatilise elektri mõju elektroonikale
FIFO (<i>First In, First Out</i>)	Esimesena saabunud andmeid töödeldakse esimesena
HTML (<i>HyperText Markup Language</i>)	Märgistuskeel veebilehe struktuuri loomiseks
JSON (JavaScript <i>Object Notation</i>)	Andmevahetuse vorming, mis tuleneb JavaScripti keelest
Lõim	Programmi alamprotsess, millele on eraldatud arvutusressursid
Lüliti kontakti väreelus	Nähtus, kus lüliti kontakt tekitab lüliti kasutamisel müra
Mikrokontroller	Programmeritav integraallülitus, mis koosneb protsessorist, mälust ning sisend- ja väljundliidestest
Pisipilt (<i>thumbnail</i>)	Väikesemõõtmeline kujutis pildist või videokaadrist
Seadmedraiver	Juhtprogramm, mis pakub tarkvaraliidest riistvaraga suhtlemiseks
Silumine (<i>debugging</i>)	Programmi lähtekoodis puuduste kõrvaldamine
USB (<i>Universal Serial Bus</i>)	Standard elektrooniliste seadmete ühendamiseks, võimaldades andmeedastust ja toitevoolu jagamist nende vahel
Veebibrauseri laiendus	Lisarakendus, mis installitakse veebibrauserisse, et lisada brauserile funktsioone või muuta brauseri kasutamist mugavamaks

Sisukord

1 Sissejuhatus	11
1.1 Lahendatav probleem	11
1.2 Lõputöö eesmärk	11
1.3 Lõputöö ulatus	12
1.4 Olemasolevad lahendused	12
2 Süsteemi arhitektuur	13
2.1 Riistvara	14
2.2 Püsivara	14
2.3 Seadmedraiver	15
2.4 Veebibrauseri laiendus	15
3 Kasutatavad tehnoloogiad.....	16
3.1 Altium Designer	16
3.2 Mikrokontroller	16
3.3 Java	17
3.4 TypeScript	17
3.5 React	17
3.6 Sass	18

3.7 Node.js	18
3.8 Veebibrauseri rakendusliides.....	18
3.9 Qt	19
3.10 SOLIDWORKS	19
4 Süsteemiarendus	20
4.1 Riistvara.....	20
4.1.1 Mikrokontroller	20
4.1.2 USB liides.....	22
4.1.3 Juhtnupud	24
4.1.4 Toitesüsteem.....	27
4.1.5 Trükkplaat.....	27
4.2 Püsivara	29
4.2.1 Lipud.....	29
4.2.2 Tegumihaldur	30
4.2.3 Ringpuhver	31
4.2.4 Lüliti kontakti värina summutamine	34
4.2.5 Seadmedraiveriga ühenduse kontrollimine	35
4.3 Seadmedraiver	36
4.3.1 Seadmega suhtlemine	36
4.3.2 Veebibrauseri laiendusega suhtlemine	37
4.3.3 Logback	37

4.4 Veebibrauseri laiendus	38
4.4.1 Video haldamine.....	38
4.4.2 Video järjehoidjad	39
4.4.3 Sündmuste juhtimine	42
4.4.4 Veebilehtede toetamine	42
4.4.5 Laienduse tootmine	42
4.5 Tarkvarapaigaldaja	43
4.5.1 Nõuetele vastavuse kontrollimine	43
4.5.2 Sihtkausta valimine	44
4.5.3 Veebibrauseri register.....	44
4.6 Korpus	45
4.7 Avastatud probleemid ja nende lahendused	46
5 Kokkuvõte	48
5.1 Soovitused tulevikuks.....	48
Kasutatud kirjandus	49
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	53
Lisa 2 Lõputöö repositoorium GitHubis.....	54

Jooniste loetelu

Joonis 1. Süsteemi arhitektuur.....	13
Joonis 2. Riistvara plokkdiagramm.	14
Joonis 3. Mikrokontrolleri programmeerimisliidese skeem.....	20
Joonis 4. Välise kristallosillaatori skeem.....	21
Joonis 5. Mikrokontrolleri toiteühendused.....	21
Joonis 6. Mikrokontrolleri andmeühendused.	22
Joonis 7. Seadme USB Type-C pesa skeem.....	23
Joonis 8. Seadme ESD-kaitse skeem.....	23
Joonis 9. USB Type-C liinide CC1 ja CC2 takistid.	24
Joonis 10. Pöördenkooderi skeem.	25
Joonis 11. Klahvi skeem.....	26
Joonis 12. Seadme toitesüsteemi skeem.....	27
Joonis 13. Trükkplaat pealtvaates.....	28
Joonis 14. Trükkplaat altvaates.	29
Joonis 15. Mäluaadressi pärimine ringpuhvrisse kirjutamiseks.....	32
Joonis 16. Mäluaadressi pärimine ringpuhvrist lugemiseks.....	33
Joonis 17. Juhtnupu kasutamise tuvastamine.	35
Joonis 18. Video järjehoidjate loend.	40

Joonis 19. Järjehoidja lisamise vaade.....	41
Joonis 20. Järjehoidja haldamise vaade.....	41
Joonis 21. Tarkvarapaigaldaja avakuva.....	43
Joonis 22. Tarkvarapaigaldaja pakub vanade failide kustutamist.	44
Joonis 23. Korpuse 3D mudeli lahti võetud vaade.	45
Joonis 24. Valmistatud seade.....	46

1 Sissejuhatus

Lõputöö autor vaatas ülikooli aastate jooksul YouTube'is rohkelt õppeainetega seotud videoloenguid. Pikkades videodes oli vaja tihtilugu liikuda edasi ja tagasi, et leida vajalik informatsioon. Autor soovis luua lahenduse, mis lihtsustab veebibrauseris videoloengute vaatamist, pakkudes lahenduseks mugavat viisi videos liikumiseks.

1.1 Lahendatav probleem

Selleks, et leida videoloengust ülesse otsitav teave, võib videot edasi kerida, valides hiirega videomängija edenemisribal sobiva ajahetke, aga nõnda on oht minna mööda olulisest infost, mis aitab õpitavat paremini mõista. Teine valik on suurendada video esituskiirust, aga soovitud kiiruse valimine YouTube videomängija menüüs nõuab mitu hiireklikki. Kui video vaataja tahab vajaliku info leidmiseks muuta esituskiirust tihti, siis peagi muutub kiiruse valimine tüütuks.

Samuti, kui vaataja on leidnud ülesse koha videoloengus, kus räägitakse temale olulisel teemal, siis vaataja võib tahta sinna kohta hiljem tagasi minna. Üks võimalus on kirja panna videokaadri ajatempel (ingl. k. *timestamp*) ning juurde kirjutada märksõnad, mis aitavad eristada kirjapanud ajatempleid, aga kui neid kirjapanemisi on rohkelt, siis muutub selline lähenemine tülikaks. Lisaks, video vaataja peab soovitud hetke tagasiminemiseks käsitsi videot kerima.

1.2 Lõputöö eesmärk

Lõputöö lahenduse abil peab kasutaja saama vähese vaevaga muuta veebibrauseris esitatava video esituskiirust. Lisaks, lahendus peab võimaldama hõlpsasti videot edasi ja tagasi kerida ning võimaldama videot pausile panna ja jätkata. Selleks, et kasutaja ei peaks videokaadri ajatemplit käsitsi kirja panema, et saaks videos tagasi minna soovitud ajahetke, peab saama lisada videole järjehoidjaid ja kerida videot edasi või tagasi lisatud järjehoidjate alusel.

1.3 Lõputöö ulatus

Esiteks, lõputöö käigus disainiti ja konstrueeriti seade, millel on juhtnupud video esituskiiruse määramiseks, edasi- ja tagasikerimiseks ning pausile panemiseks ja jätkamiseks. Lisaks saab seadme abil lisada videole järjehoidjaid ja kerida videot edasi ning tagasi lisatud järjehoidjate alusel. Teiseks, loodud seadmele arendati püsivara ning arvutil töötav seadmedraiver. Kolmandaks, videote esituse haldamiseks kirjutati veebibrauseri laiendus. Viimaseks, seadmedraiveri paigaldamiseks ning veebibrauseri laienduse seadistamiseks loodi tarkvarapaigaldaja.

1.4 Olemasolevad lahendused

Lõputöö autor ei ole teadlik, et leidub sellist seadet, mida saab lihtsasti kasutada HTML5 videote taasesituse juhtimiseks: esituskiiruse muutmine, video edasi- ja tagasikerimine ning järjehoidjate kasutamine. Tarkvarana on olemas lahendusi, mis pakuvad osaliselt käesoleva projekti tulemusena loodud funktsionaalsust. Näiteks, YouTube pakub kiirklahve teatud toimingute jaoks [1]. Kasutaja saab klahvidega suurendada ja vähendada video esituskiiruse kordajat 0,25 võrra vahemikus 0,25 kuni 2. Samas, lõputööna loodud lahendus võimaldab ühe vajutusega valida soovitud esituskiiruse kordaja: 1, 1,5 või 2. YouTube sisse ehitatud kiirklahvidele sarnast funktsionaalsust pakub ka Google Chrome'le saada olev laiendus Video Speed Controller [2], mida saab kasutada HTML5 video puhul. Järjehoidjate kasutamise jaoks on Google Chrome'le saadaval laiendus Smart Bookmarks for YouTube [3], aga võrreldes lõputöö käigus loodud brauseri laiendusega ei näita see laiendus järjehoidjana lisatud videokaadrite pisipilte, mis võib raskendada otsitava järjehoidja leidmist kuvatavast loetelust.

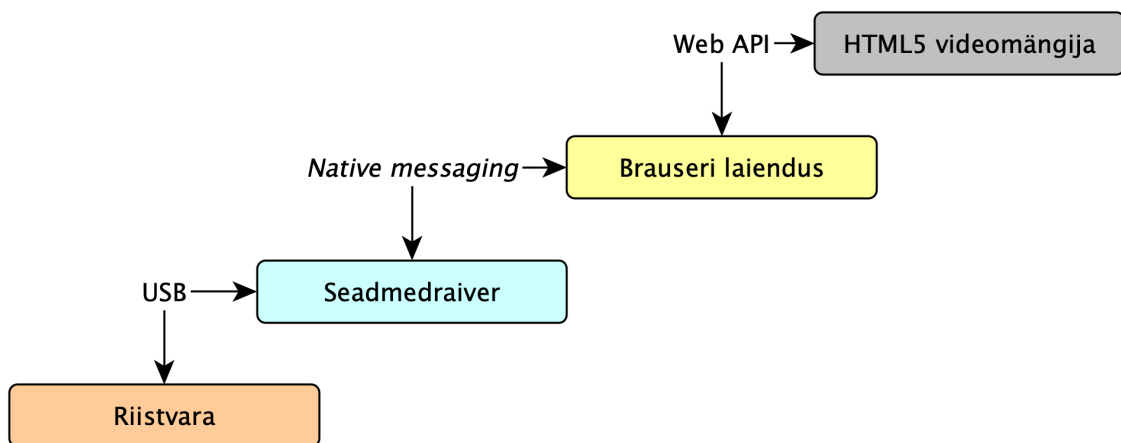
2 Süsteemi arhitektuur

Loodud süsteemi arhitektuur koosneb kolmest põhikomponendist: riistvaraseade, seadmedraiver ja veebibrauseri laiendus. Joonis 1 on kujutatud neid komponente ja nende suhtluseks kasutatavat liidest ning suhtluse suunda.

Riistvaraseade sisaldab nuppe ja pöördenkooderit video taasesituse haldamiseks ning järjehoidjate kasutamiseks. Seade on *Universal Serial Bus*'i (USB) kaudu arvutiga ühendatud ja seade suhtleb seadmedraiveriga, et teisendada kasutaja sisendid käskudeks, mida saadetakse veebibrauseri laiendusele.

Seadmedraiver töötab arvutis ja toimib liidesena seadme ja brauseri laienduse vahel. Draiver vastutab seadmelt saadud käskude tõlgendamise ja brauseri laienduse poolt teostatavateks toiminguteks tõlkimise eest. Draiver saadab ka sõnumeid seadme püsivarale, et hallata seadme funktsioone.

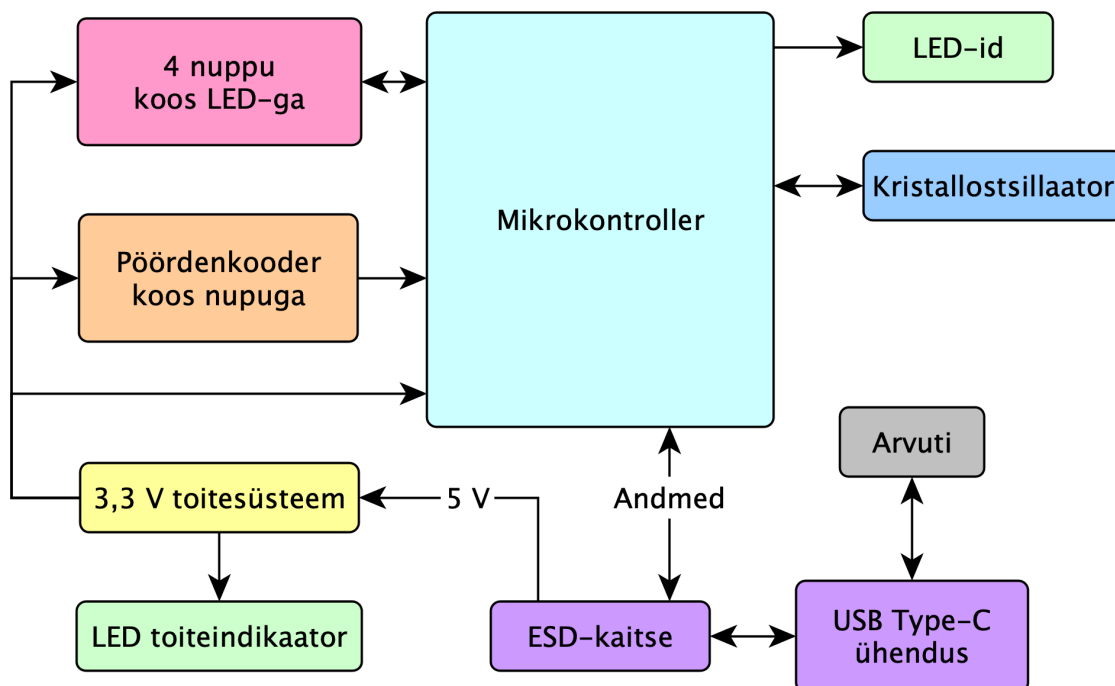
Veebibrauseri laiendus on tarkvaramoodul, mis töötab veebibrauseris ja pakub lisafunktsioone video taasesituse haldamiseks. Laiendus suhtleb seadmedraiveriga, et võtta vastu kasutaja sisendeid ja sooritada toiminguid, nagu video esituskiiruse reguleerimine, järjehoidjate lisamine ja video kerimine.



Joonis 1. Süsteemi arhitektuur.

2.1 Riistvara

Riistvara komponentide hulka kuuluvad Joonis 2 kujutatud mikrokontroller, sisend- ja väljundliidesed ning toiteskeem.



Joonis 2. Riistvara plokkdiagramm.

2.2 Püsivara

Mikrokontrolleril töötava püsivara arhitektuur põhineb modulaarsel disainil ning on kirjutatud C-keeles. Esiteks, püsivara vastutab nupuvajutuste ja pöördenkooderi kasutamise tuvastamise eest. Samuti tegeleb püsivara lülite kontakti väreluse vähendamisega (ingl. k. *switch debouncing*), et seade reageeriks kasutaja sisenditele töökindlalt. Teiseks, püsivara kasutab seadmedraiveriga suhtlemisel vastuvõetud ning saadetavate sõnumite ajutiseks hoiustamiseks ringpuhvrit, enne kui neid sõnumeid töödeldakse. Neljandaks, püsivara sisaldab lihtsat tegumihaldurit, mis jooksutab eri ülesandeid kindlate ajavahemike järel. Tegumihaldurit juhitakse lippude abil, et anda märku, kui ülesande läbiviimiseks on möödunud piisavalt aega.

2.3 Seadmedraiver

Seadmedraiver on kirjutatud Java keeles ja toimib liidesena püsivara ja veebibrauseri laienduse vahel. Draiver suhtleb laiendusega brauseri sõnumside funktsiooni abil [4] ning kasutab sõnumite saatmiseks ja vastuvõtmiseks programmi sisend- ja väljundvooge. Seadmega suhtlemiseks kasutab draiver jSerialComm [5] teeki, mis pakub jadaside jaoks platvormist sõltumatut liidest. Draiver initsialiseerib jadaühenduse seadmega, võtab seadmelt vastu andmeid ning saadab seadmele sõnumeid.

2.4 Veebibrauseri laiendus

Veebibrauseri laiendus koosneb taustskriptist [6] ja sisuskriptist [7]. Taustskript on skript, mis töötab brauseri taustal, sõltumata brauseris avatud veebilehest. Seda kasutatakse tavaliselt laiendi haldamiseks, brauseriülestest sündmuste käsitlemiseks ja ülesannete täitmiseks, mis ei ole otseselt seotud veebilehe sisuga. Sisuskript on skript, mis käivitub konkreetse veebilehe kontekstis, millele on laiendusel ligipääs. Sisuskriptid võimaldavad laiendusel muuta veebilehe sisu ja parandada või täiendada veebilehe funktsionaalsust. Sisuskriptid erinevad taustskriptidest, kuna sisuskript käivitatakse ainult siis, kui laiendusega seotud veebileht on laetud, ja sisuskriptil on piiratud juurdepääs brauseri poolt pakutavale rakendusliidesele.

Käesoleva laienduse taustskript vastutab seadmedraiveri ja sisuskripti vahelise suhtluse haldamise eest. See kasutab laienduse ja draiveri vahel sõnumite saatmiseks veebibrauseri poolt pakutavat sõnumside funktsiooni [4]. Kui draiver saadab brauseri laiendusele sõnumi, siis taustskript töötleb sõnumit ja saadab selle sisuskriptile. Taustskript käsitleb ka veebibrauseri vahekaartide sündmusi, et teavitada sisuskripti muudatustest.

Sisuskript vastutab HTML5 video taasesituse ja video järjehoidjate haldamise eest. Järjehoidjate kasutajaliidest kuvatakse üle HTML5 videomängija. Kasutajaliides võimaldab järjehoidjate lisamist, kuvamist, kasutamist ja eemaldamist. Järjehoidjad salvestatakse lokaalselt brauseri salvestussüsteemi [8].

3 Kasutatavad tehnoloogiad

Lõputöö lahendamiseks kasutatud tehnoloogiad valiti peamiselt kahe tingimuse alusel. Esiteks, valik ei seadnud piirangut lahenduse elluviimisele. Teisisõnu, valitud tehnoloogia võimaldas lahendust ellu viia eesmärgipäraselt. Teiseks, lõputöö autor oli valitud tehnoloogiat varem kasutanud, mistõttu sai keskenduda projekti lahenduse loomisele, ilma, et peaks aega kulutama konkreetse tehnoloogia tavade ja kasutusviisidega tutvumiseks.

3.1 Altium Designer

Altium Designerit [9] kasutati seadme elektroonikaskeemide ja trükkplaadi disainimiseks. Altium Designer on professionaalne elektroonilise disaini automatiseerimise programm, mida kasutatakse trükkplaatide projekteerimiseks. Programm pakub laiaulatuslikku tööriistakomplekti skeemide ja trükkplaatide disainimiseks, elektroonikakomponentide kogumike haldamiseks, trükkplaadi disainireeglite kontrollimiseks ning tootmisfailide ja dokumentatsiooni genereerimiseks. Programmi abil disainitud skeemidest ning trükkplaadist on lähemalt juttu peatükis 4.1.

3.2 Mikrokontroller

Seadme jaoks valiti mikrokontroller STM32F042C6T6 [10], mis kuulub ettevõtte STMicroelectronics tooteseeriasse STM32F0. Mikrokontroller põhineb ARM Cortex-M0 protsessoril ning on mõeldud vähese energiatarbega ja kulutundlikele rakendustele. Mikrokontrolleril on 32 väljaviiku, mida on piisavalt projektiks vajaminevate signaaliühenduste jaoks. Selle mikrokontrolleri üks peamisi eeliseid on sisseehitatud USB liides, mis võimaldab arvutiga suhtlemist ilma väliseid komponente vajamata.

Mikrokontrolleri püsivara kasutab riistvara abstraktsiooni kihti STM32Cube [11], et suhelda arvutiga USB kaudu, seadistada mikrokontrolleri taktsagedust ja liideseid ning kasutada sisendeid ja väljundeid. Püsivarast on lähemalt juttu peatükis 4.2.

3.3 Java

Projekti seadmedraiver on kirjutatud Javas [12], mis on laialdaselt kasutatav objektorienteeritud programmeerimiskeel. Java on platvormist sõltumatu keel ehk draiverit saab kasutada hõlpsasti erinevates operatsioonisüsteemides. Lisaks, Javas kirjutatud programmi puhul ei pea arendaja manuaalselt mälu haldama, sest sellega tegeleb Java programmi jooksutav keskkond [13]. Samuti tuleb arendustöö puhul kasuks, et Javal on suur ning aktiivne kogukond ja laiaulatuslik teekide ning raamistike valik. Draiverist on lähemalt juttu peatükis 4.3.

3.4 TypeScript

Veebibrauseri laienduse lähtekood on kirjutatud TypeScriptis [14], mis on JavaScripti ülemhulk ehk TypeScript vastab ECMAScripti standardile [15]. See tähendab, et olemasolevat JavaScripti koodi saab kergesti TypeScripti üle viia ning TypeScriptis saab kasutada populaarseid JavaScripti teeke ja raamistikke. TypeScripti üheks peamiseks eeliseks JavaScripti ees on staatiliste andmetüüpide võimaldamine, mis aitab tabada lähtekoodi kompileerimise ajal andmetüüpidega seotud vigu, võimaldades paremat koodi kvaliteeti ja vähem käitusvigu. Brauseri laiendusest on rohkem juttu peatükis 4.4.

3.5 React

Brauseri laienduse graafiline kasutajaliides on loodud Reacti [16] abil, mis on laialdaselt kasutatav JavaScripti teek veebirakenduste kasutajaliideste loomiseks. React pakub tõhusat viisi korduvkasutatavate kasutajaliidese komponentide loomiseks, mida saab dünaamiliselt värskendada vastavalt muutuvatele andmetele. React järgib komponendipõhist arhitektuuri, mis võimaldab luua keerulisi kasutajaliideseid, koostades väiksemaid iseseisvaid komponente. Samuti kasutab React virtuaalset veebilehe dokumendi struktuuri, et optimeerida jõudlust, vähendades reaalses struktuuris tehtavate manipulatsioonide arvu.

3.6 Sass

Brauseri laiendus kasutab kasutajaliidese kujundamiseks kaskaadlaadistiku eelprotsessorit Sass [17], mis laiendab kaskaadlaadistiku võimalusi lisafunktsioonidega, nagu muutujad, pesastatud reeglid ja funktsioonid. See võimaldab arendajal kirjutada lihtsamini hallatavat ja modulaarsemat koodi, pakkudes mitmeid võimsaid funktsioone, mis pole kaskaadlaadistikus saadaval. Sass-keeles kirjutatud lähtekood kompileeritakse veebibrauseris kasutamiseks kaskaadlaadistikku.

3.7 Node.js

Veebibrauseri laienduse arendamiseks kasutati Node.js [18], mis on JavaScripti käituskeskkond. Node.js võimaldab kasutada paketihooldurit [19], mida omakorda kasutatakse teekide, skriptide ja muude arendustööriistade haldamiseks. Paketihoolduri skriptide abil saab automatiseerida TypeScripti koodi kompileerimist ning brauseri laienduse tootmist.

Lõputöös kasutatakse lähtekoodi moodulite pakkimise süsteemi Webpack [20]. See suudab kompileerida spetsiaalse laienduse [21] abil mitmest erinevast lähtekoodi failist koosneva TypeScripti projekti ühte JavaScripti faili, mida veebibrauser vajab laienduse jooksutamiseks. Lisaks suudab Webpack kompileerida Sass keeles kirjutatud veebilehe kujunduse kaskaadlaadistikku ning sisestada selle eelpool mainitud väljundfaili. Samuti oskab Webpack sisestada väljundfaili pildifaile.

3.8 Veebibrauseri rakendusliides

Veebibrauseri laiendus kasutab brauseri rakendusliidest [22], et käsitleda vahekaartidega seotud sündmusi. Laiendus kasutab ka brauseri rakendusliidest sõnumite saatmiseks laienduse taustal töötava taustskripti ja aktiivse veebilehe kontekstis töötava sisuskripti vahel. See võimaldab laienduse erinevate osade vahelist suhtlust ja koordineerimist, võimaldades neil sujuvalt koos töötada. Samuti kasutab laiendus arvutis töötava seadmedraiveriga suhtlemiseks brauseri sõnumside (ingl. k. *native messaging* [4]). Videoloengu järjehoidjate talletamiseks salvestab laiendus andmeid võti-väärtus paaride kujul püsivalt brauseri mällu, mis võimaldab laiendusel pääseda ligi järjehoidjate andmetele kuni laienduse eemaldamiseni brauserist.

3.9 Qt

Loodud tarkvarapaigaldaja kasutab Qt'd [23], mis on populaarne platvormideülene raamistik graafiliste kasutajaliideste ja C++ programmeerimiskeelt kasutavate rakenduste loomiseks. Qt pakub laiulatuslikku tööriistade ja teekide komplekti rakenduste arendamiseks. Qt signaali- ja pesamehhanism (ingl. k. *signals & slots* [24]) võimaldab tõhusat suhtlust tarkvarapaigaldaja erinevate komponentide vahel. Qt'l on suur ja aktiivne arendajate kogukond, mis pakub ulatuslikku dokumentatsiooni, foorumeid ja tugiresse, muutes arendusprotsessi käigus hõlpsaks abi leidmise, õppimise ja probleemide lahendamise. Tarkvarapaigaldajast on lähemalt juttu peatükis 4.5.

3.10 SOLIDWORKS

Loodud seadme korpuse disainimiseks kasutati SOLIDWORKS-i [25], mis võimaldab täpselt modelleerida ja visualiseerida seadme füüsilist vormi ja mõõtmeid, aidates kaasa tootmis- ja monteerimisprotsessile. SOLIDWORKS on populaarne raalprojekteerimise tarkvara, mis võimaldab inseneridel ja disaineritel luua mehaaniliste detailide, koostude ja toodete üksikasjalikke 3D mudeleid. Seda kasutatakse laialdaselt sellistes tööstusharudes nagu autotööstus, lennundus, tootmine ning tarbekaubad [26]. Seadme korpusest on lähemalt juttu peatükis 4.6.

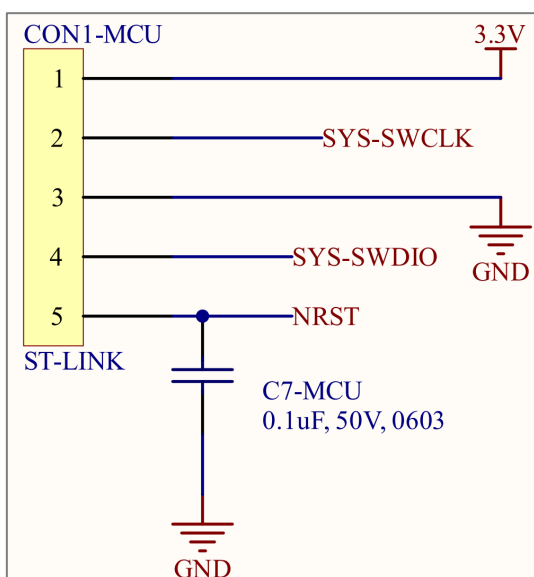
4 Süsteemiarendus

Peale lõputöö eesmärkide seadmist ning süsteemi arhitektuuri määramist, alustati seadme elektroonika disainimise ja valmistamisega. Kui trükkplaat oli antud tootmisse, siis alustati mikrokontrolleri püsivara arendamisega. Trükkplaadi kättesaamisel tegeleti elektroonikakomponentide jootmisega ning testiti trükkplaadi ühendusi ning kontrollnuppude tööd. Olles saanud riistvara tööle, tegeleti seadmedraiveri arendamisega, mille valmides sai keskenduda brauseri laienduse loomisele. Kuna draiveri ning brauseri laienduse vahelise suhtluse võimaldamine nõuab arvutis teatud seadistuste tegemist, siis otsustati luua tarkvarapaigaldaja, et automatiseerida vajalikke toiminguid. Viimaks disainiti seadme korpus ning telliti selle valmistamine.

4.1 Riistvara

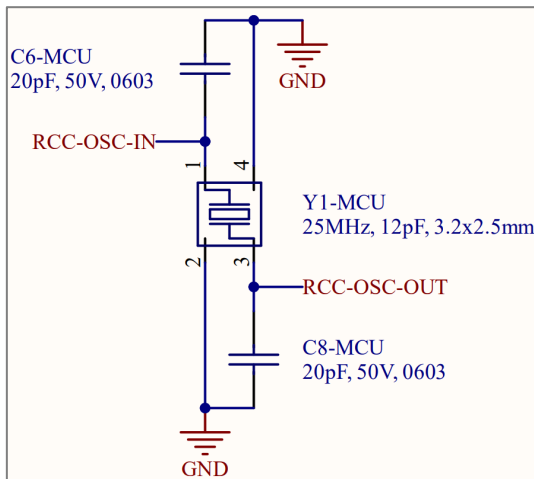
4.1.1 Mikrokontroller

Nagu eelpool mainitud, riistvara kasutab mikrokontrollerit STM32F042C6T6. Mikrokontrolleri programmeerimiseks on Arm'i *Serial Wire Debug* liides [22, lk 27], mille ühendused on kujutatud Joonis 3. Mikrokontrolleri programmeerimiseks kasutati ST-LINK programmeerijat, mis toetab seda liidest.



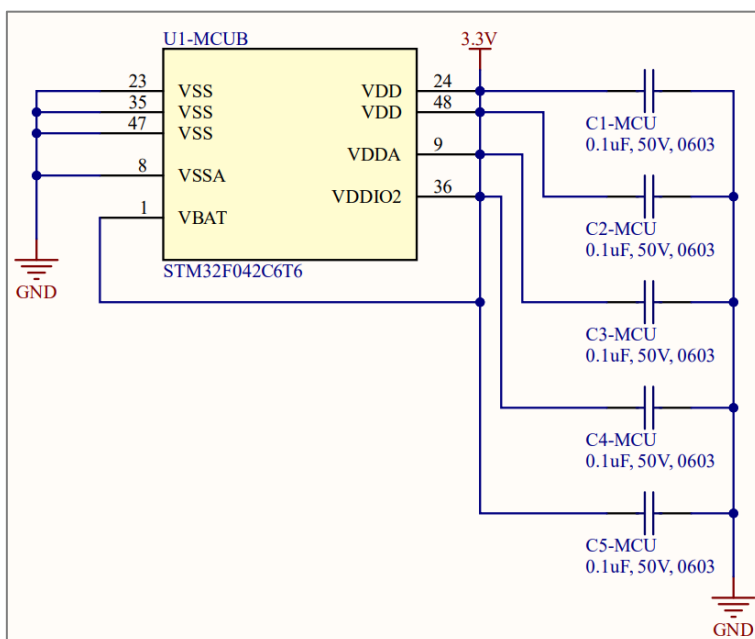
Joonis 3. Mikrokontrolleri programmeerimisliidese skeem.

Kuigi mikrokontrolleril on integreeritud USB liides, mis suudab töötada ka ilma välise otsillaatorita, sest integreeritud otsillaatorit saab sünkroniseerida USB andmepaketi alguse järgi [27, lk 27], otsustati igaks juhuks võimaldada välise kristallostsillaatori kasutamist, mille skeem on kujutatud Joonis 4. Välise otsillaatori sagedus peab olema vahemikus 4 kuni 32 MHz [27, lk 62].



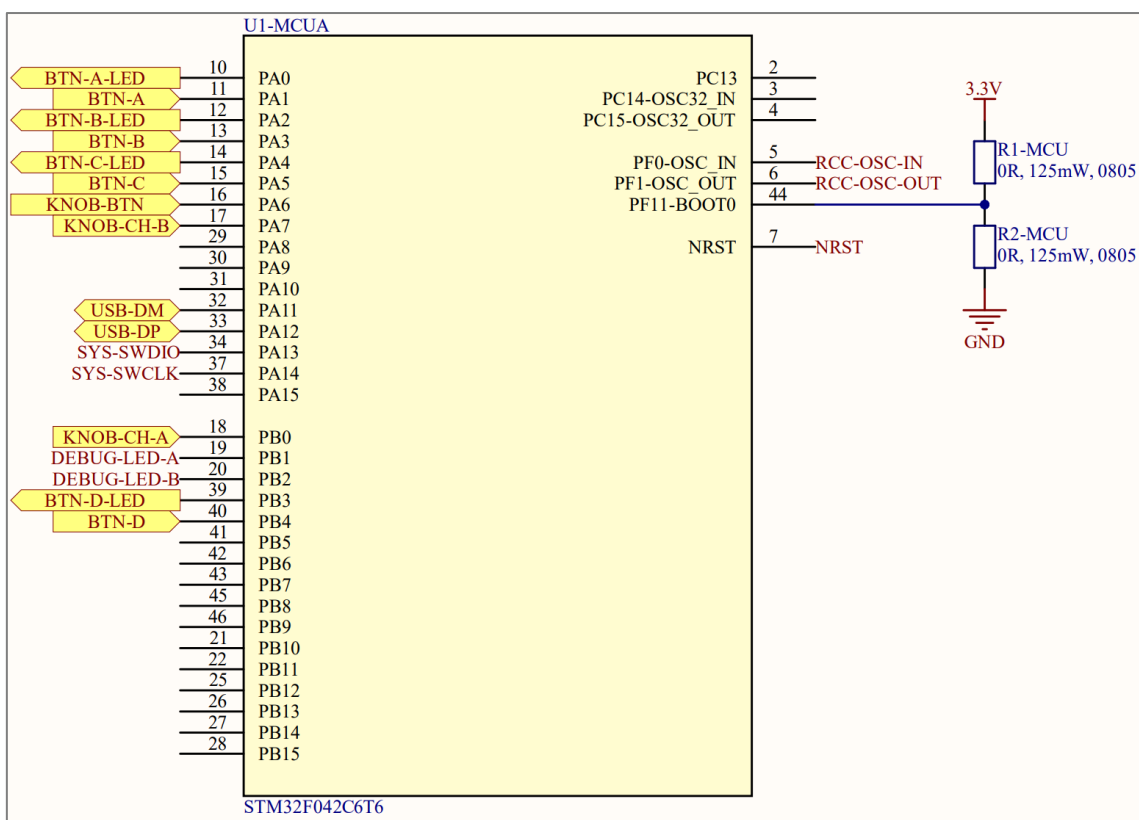
Joonis 4. Välise kristallostsillaatori skeem.

Joonis 5 kujutatud mikrokontrolleri toiteühenduste juurde on andmelehe järgi [22, lk 43] pandud 0,1 μ F suurused kondensaatorid, et stabiliseerida vajaminevat toidet ning vähendada võimalikku toitemüra.



Joonis 5. Mikrokontrolleri toiteühendused.

Mikrokontrolleri USB, juhtnuppude, programmerimisliidese ning välise ostsillaatori andmeühendused on kujutatud Joonis 6. Programmeerimisliidese, välise ostsillaatori ning USB andmeside jaoks kasutatavad väljaviigud asuvad mikrokontrolleris kindlaks määratud kohtades. Ülejäänud kasutust leidvad väljaviigud valiti nõnda, et saaks võimalikult vähese vaevaga trükkplaadile paigutada signaalirajad, mida on vaja klahvide ning pöördenkooderi ühendamiseks mikrokontrolleriga. Kuna trükkplaadil on neljast kihist kaks, mida saab kasutada signaaliradade jaoks, siis üritati paigutada need rajad nõnda, et rada kulgeks vaid mööda üht kihti ning raja pikkus oleks mõistlikult lühike. Mikrokontrolleri kasutamata väljaviigud jäeti ühendamata.

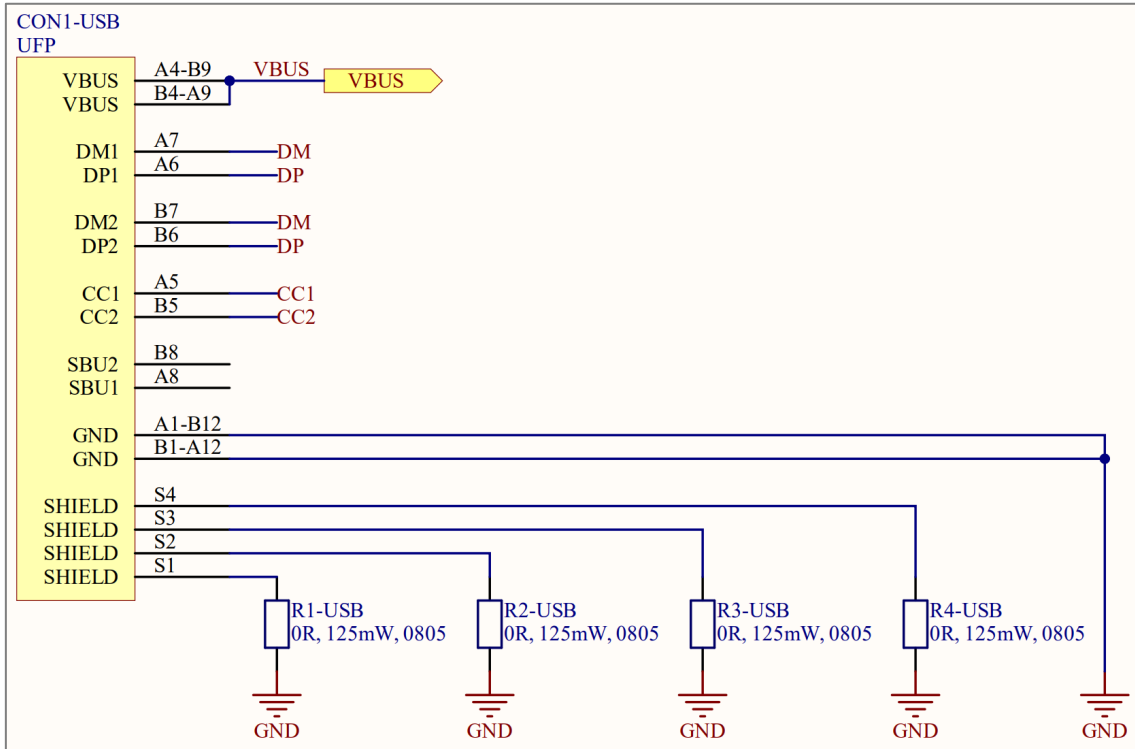


Joonis 6. Mikrokontrolleri andmeühendused.

4.1.2 USB liides

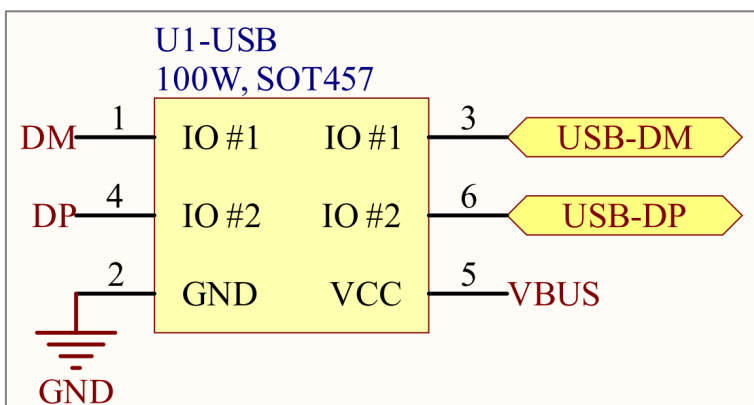
Seade kasutab USB Type-C ühendust, mille kaudu suhtleb seade arvutis töötava draiveriga ning saab toitevoolu. Type-C valiti, sest tegemist on kaasaegse ühendusega, mille pistikut saab ühendada pesa mõlemat pidi. Andmeside toimub USB 2.0 standardi kohaselt, kasutades diferentsiaalset signaaliedastust, mis hõlmab kahte komplementaarset signaalirada andmete edastamiseks. USB Type-C pesa kontaktide pöördsummeerilise paigutuse tõttu on võimalik kaablit ühendada pesa mõlemat pidi, mille tõttu tuleb

mõlemad andmeside diferentsiaalpaarid ühendada mikrokontrolleriga. Joonis 7 kujutatud skeemil moodustavad liinid DM1 ja DP1 ühe diferentsiaalpaari ning liinid DM2 ja DP2 moodustavad teise paari.



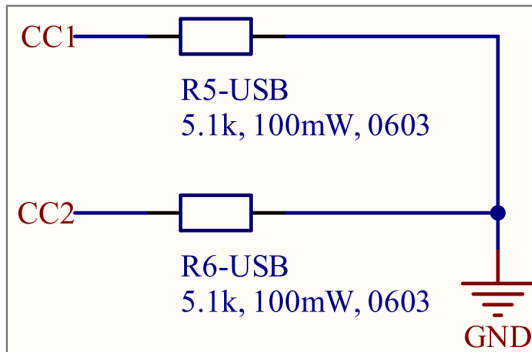
Joonis 7. Seadme USB Type-C pesa skeem.

Selleks, et kaitsta riistvara staatilise elektri eest, läbivad USB liidese kaudu ühendatud toite- ja signaaliühendused Joonis 8 kujutatud ESD-kaitset. Skeem on koostatud ESD-kaitse komponendi andmelehe põhjal [28, lk 2].



Joonis 8. Seadme ESD-kaitse skeem.

Selleks, et arvuti saaks pakkuda seadmele toitevoolu, on vaja Joonis 9 kujutatud seadme USB Type-C pesa liinidele CC1 ja CC2 ühendada 5,1 k Ω takistid, mis on ühendatud maandusega. Sel viisil võimaldab arvuti 5 V toitepinget ja kuni 3 A voolu.

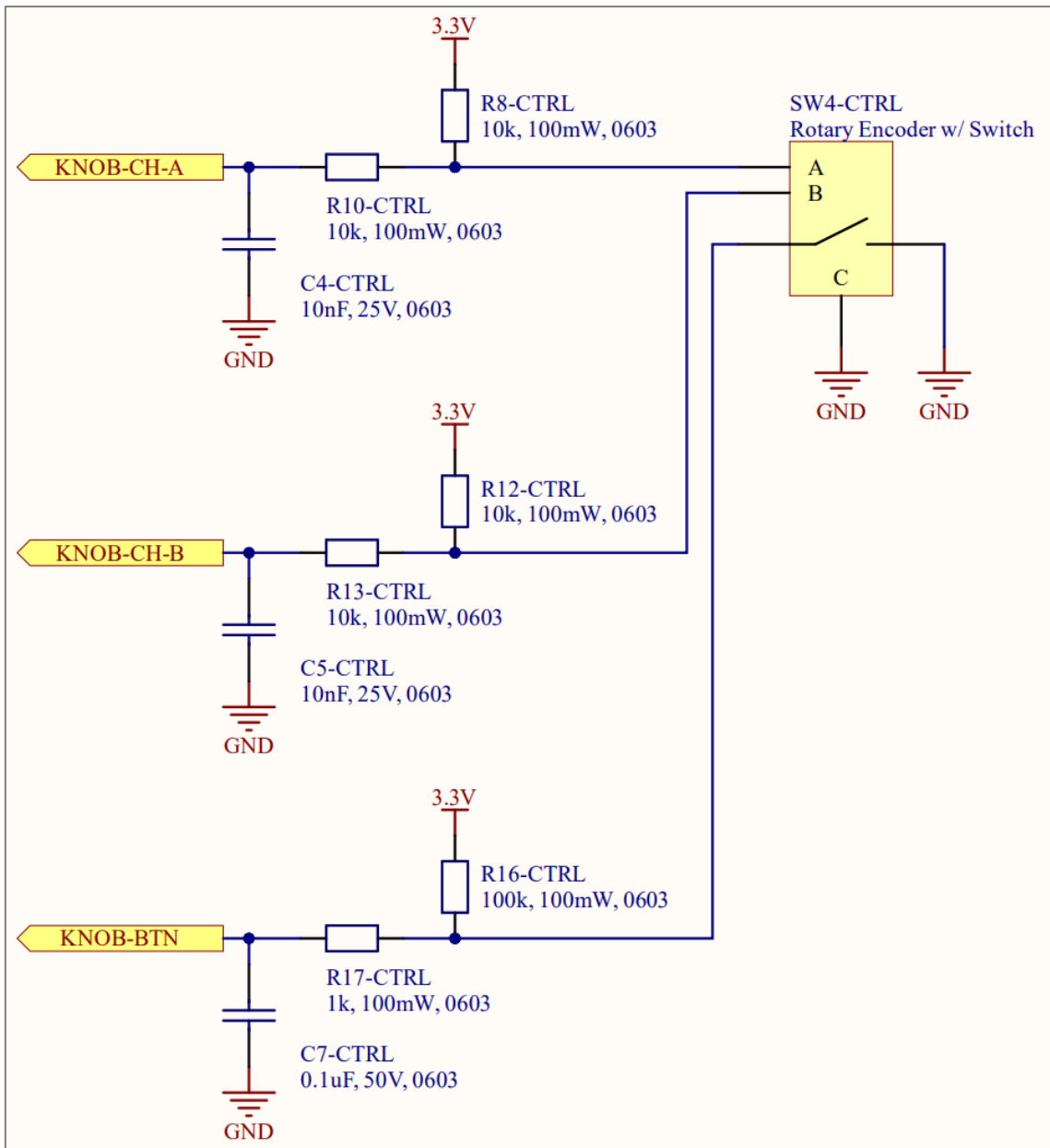


Joonis 9. USB Type-C liinide CC1 ja CC2 takistid.

4.1.3 Juhtnupud

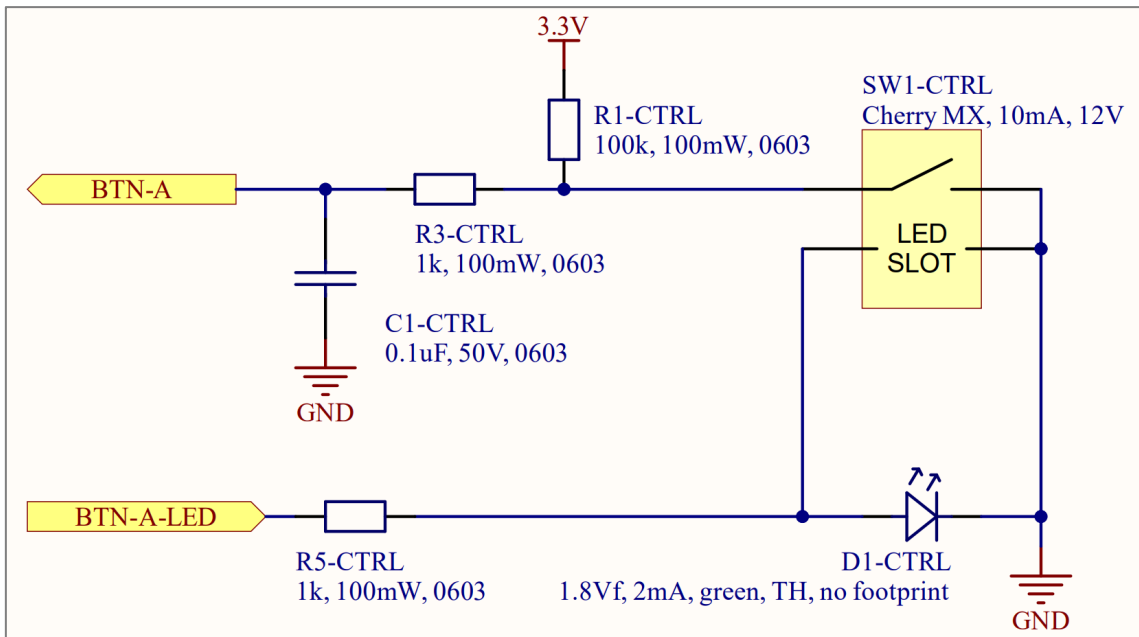
Loodud seadmel on kasutajaliidese jaoks Cherry MX mehaanilised klahvid, mida kasutatakse video esituskiiruse valimiseks ning järjehoidjate kasutamiseks. Klahvi korpuses on ava, mille kaudu on trükkplaadile lisatud eraldiseisev LED, mida mikrokontroller saab juhtida klahvi kasutamisest sõltumata. Need klahvid valiti, sest nende jaoks on lai valik eri nupukatteid. Seadmel on samuti Bourns'i astmeline pöördenkooder koos nupuga, et saaks kerida videot ning liikuda järjehoidjate menüüs. Konkreetne pöördenkooderi mudel [29] valiti, sest selle keeramiseks on vaja rohkem jõudu võrreldes tavalise pöördenkooderiga [30], mille tõttu sobib valitud mudel paremini kasutajaliidese jaoks.

Joonis 10 kujutatud pöördenkooderi skeemis on RC-filter kontakti väreluse vähendamiseks, ning kuigi takisti ja kondensaatori väärtused on võetud pöördenkooderi andmelehel [29], siis sellegi poolest on see RC-filter ebatõhus, sest isegi filtri olemasolul suudab mikrokontroller tuvastada ühte pööret mitu korda, mis on ebasoovitav olukord.



Joonis 10. Pöördenkooderi skeem.

Kuigi Joonis 11 oleval klahvi skeemil on näha RC-filtrit, et vähendada klahvi kasutamisel esinevat lüliti kontakti värelust, siis seadme kasutamisel selgus, et RC-filter on ebaefektiivne taolise klahvi kontakti väreluse vähendamisel, isegi kui RC-filtri ajakonstant on sobiv mikrokontrolleri sisendpinge kontrollimiseks.

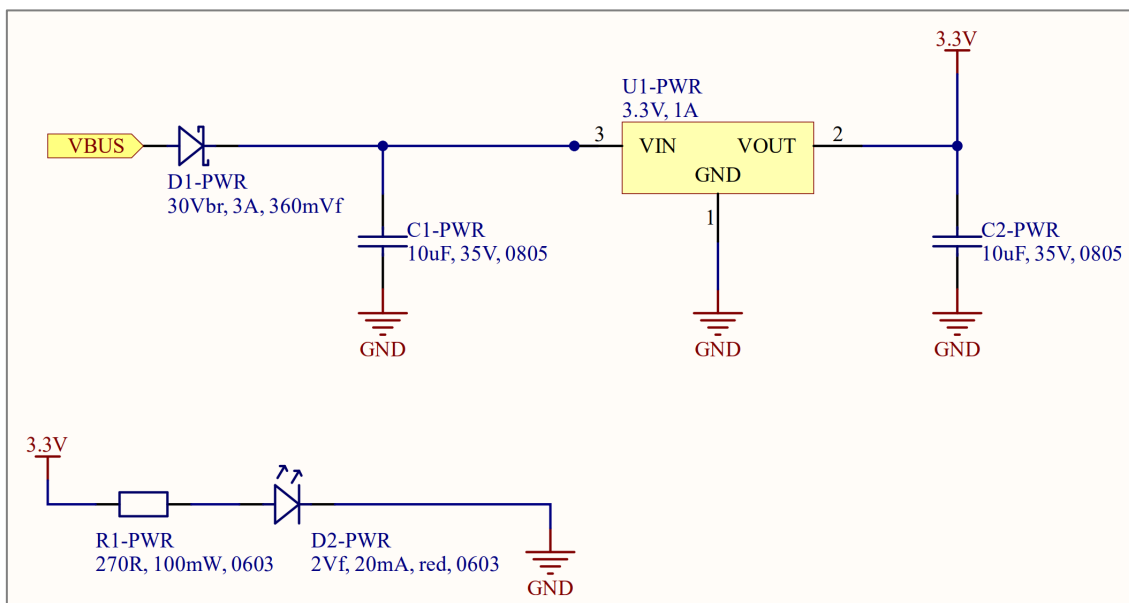


Joonis 11. Klahvi skeem.

Joonis 11 kujutatud skeemil on klahvi jaoks ülestõmbetakisti (ingl. k. *pull-up resistors*) R1-CTRL, et mikrokontrolleri sisendil olev pinge käituks ettearvatavalt. Teisisõnu, kui klahvi ehk lüliti üks klemm on ühendatud mikrokontrolleri külge ning teine maanduse või toitepinge külge ja skeemis ei ole üles- ega allatõmbetakistit, siis klahvi mittekasutamisel võib mikrokontrolleri sisend ujuda (ingl. k. *floating*), mis saab põhjustada ebakorrektselt klahvi kasutamise tuvastamist.

4.1.4 Toitesüsteem

Mikrokontrollerile sobiva toitepinge võimaldamiseks kasutatakse STMicroelectronics'i lineaarset pingestabilisaatorit LD1117AS33TR [31], mille väljundpingeks on fikseeritud 3,3 V. Pingestabilisaator võimaldab kuni 1 A väljundvoolu. Joonis 12 on kujutatud pingestabilisaatori skeemi koos LED toiteindikaatoriga. Indikaator on selleks, et saaks lihtsasti visuaalselt tuvastada, kas seadmel on toitepinge.



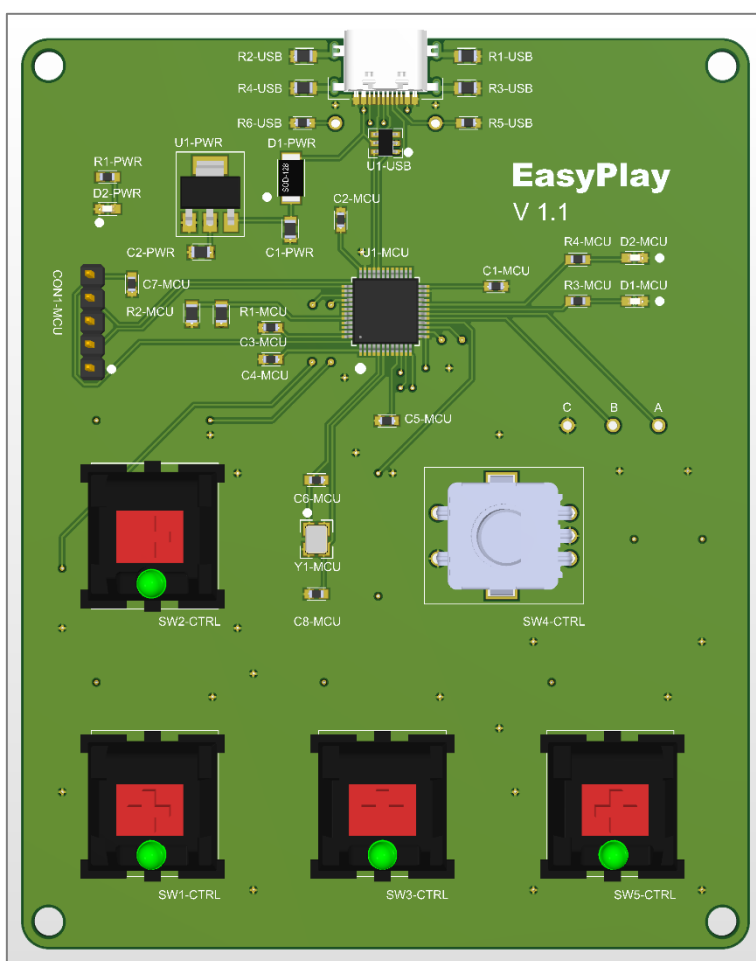
Joonis 12. Seadme toitesüsteemi skeem.

4.1.5 Trükkplaat

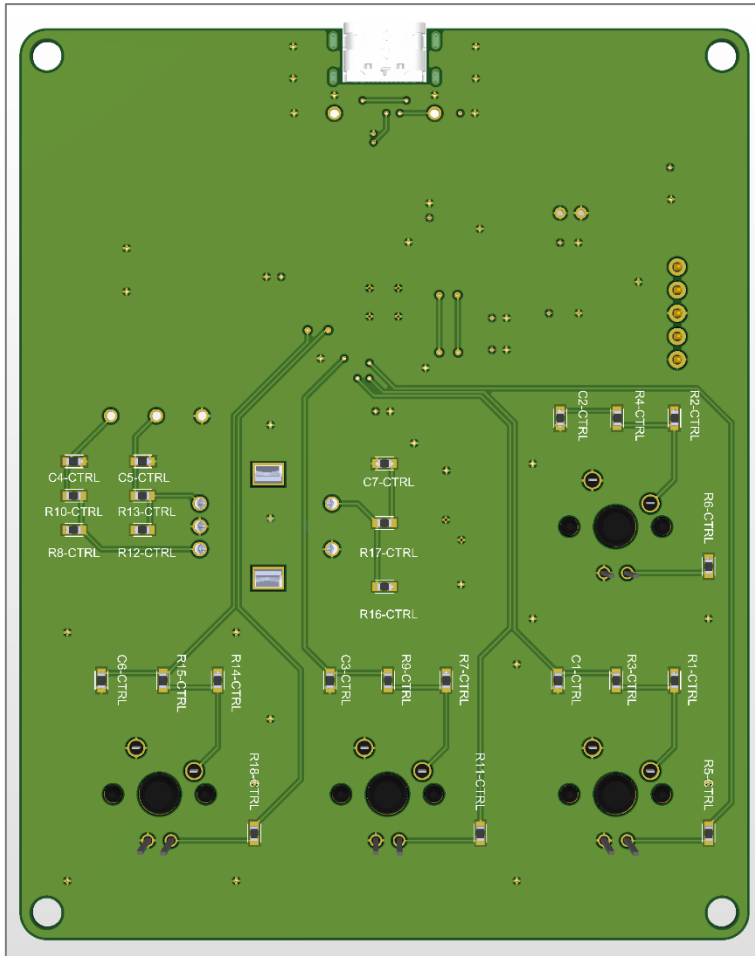
Joonis 13 kujutatud trükkplaadi komponentide paigutamisel lähtuti eelkõige kasutatavate klahvide eesmärgist ning vajalike signaaliühenduste loomisest. Nimelt, seadme allosas on tahtlikult ühes reas klahvid, millega kasutaja saab määrata video esituskiirust. Klahvide vahele on jäetud piisavalt ruumi, et klahvile saaks lisada nupukatte. Pöördenkooderi ümber on üksjagu ruumi, et selle kasutamisel ei vajutaks kasutaja kogemata ümber olevaid klahve.

Mikrokontroller, USB pesa ja ESD-kaitse on joondus, et USB komplementaarsed signaalirajad oleks sirged, sest nõnda on rajad ühepikad. Rajad peavad olema võimalikult ühepikad, et edastatavad signaalid jõuaks vastuvõtjasse samal ajal, sest nõnda on signaali müratase madalam [32]. ESD-kaitse on võimalikult lähedal USB pesale, et vähendada staatilise elektri tekkimisel negatiivset mõju trükkplaadil olevatele komponentidele.

Trükkplaadil on neli kihti ning on mõõtmetelt 80 mm pikk ja 100 mm lai. Keskmistel kihtidel on kihi täies ulatuses vaseala (ingl. k. *copper pours*) ja üks kiht on maandatud ning teine on 3,3 V toite jaoks. Sellisel viisil on trükkplaadil olevatel komponentidel lihtne luua ühendust toite ning maandusega. Peamisel kihil asub enamik komponente. Ülejäänud komponendid asuvad Joonis 14 kujutatud alumisel kihil, et nendele pääseks jootekolbiga mugavalt ligi. Teisisõnu, nõnda ei sega peamisele trükkplaadi kihile kinnitatud klahvid jootekolbiga ligi pääsemist klahvide jaoks mõeldud takistitele ja kondensaatoritele.



Joonis 13. Trükkplaat pealtvaates.



Joonis 14. Trükkplaat altvaates.

4.2 Püsivara

Mikrokontrolleri püsivara peamine eesmärk on saata seadmedraiverile sõnumeid juhtnuppude kasutamise kohta ning juhtida klahvide valgusdioode seadmedraiverilt tulevate sõnumite põhjal.

4.2.1 Lipud

Püsivara tegumihaldur kasutab lippe, et anda märku ülesande käivitamiseks. Lipusüsteem kasutab lippude ja nende olekute tähistamiseks kahendnumbreid. Igat lippu tähistatakse kahendarvuga, milles ainult üks number on võrdne ühega; võimalikud lipu väärtused on 0001_2 , 0010_2 , 0100_2 , 1000_2 ja nii edasi. Lipu olek (aktiivne või mitteaktiivne) salvestatakse lipurühma, mis on samuti kahendarv.

Lipu aktiivseks seadmiseks teeb püsivara loogilise liitmise (disjunktsioon) lipurühma ja lipu väärtuse vahel. Selle tulemusel seatakse lipurühmas lipu väärtusele vastav number üheks, mis näitab, et lipp on nüüd aktiivne.

Selleks, et kontrollida kas lipp on aktiivne, teeb püsivara loogilise korrutamise (konjunktsioon) lipurühma ja lipu väärtuse vahel. Kui tulemus on võrdne ühega, näitab see, et lipp on aktiivne.

Lipu kustutamiseks teeb püsivara loogilise korrutamise lipurühma ja lipu pöördväärtuse vahel ning määrab vastuse lipurühma väärtuseks. Selle tulemusel seatakse lipurühmas lipu väärtusele vastav number nulliks, jättes teised lipurühma numbrid muutmata.

4.2.2 Tegumihaldur

Püsivara tegeleb mitmete ülesannetega, nende hulgas USB kaudu sõnumite saatmine ning vastuvõtmine, juhtnuppude kasutamise tuvastamine ja klahvide valgusdioodide juhtimine. Tegumihalduri poolt käivitataavad tegevused on C-keele funktsioonid, mis täidavad konkreetseid toiminguid. Tegumihaldur ei jookсутa tegevusi paralleelselt. Teisisõnu, kõik funktsioonid töötavad samas lõimes ja saavad üksteist blokeerida.

Tegumihaldur kasutab mikrokontrolleri taimerit, mis käivitab iga millisekundi järel katkestusetäituri, mis omakorda tõstab lipu, kui jälgitav ajavahemik, näiteks 1 ms, 15 ms, 55 ms või 1000 ms, on möödas. Need lipud annavad märku, et aeg on käivitada nende ajavahemikega seotud tegevused.

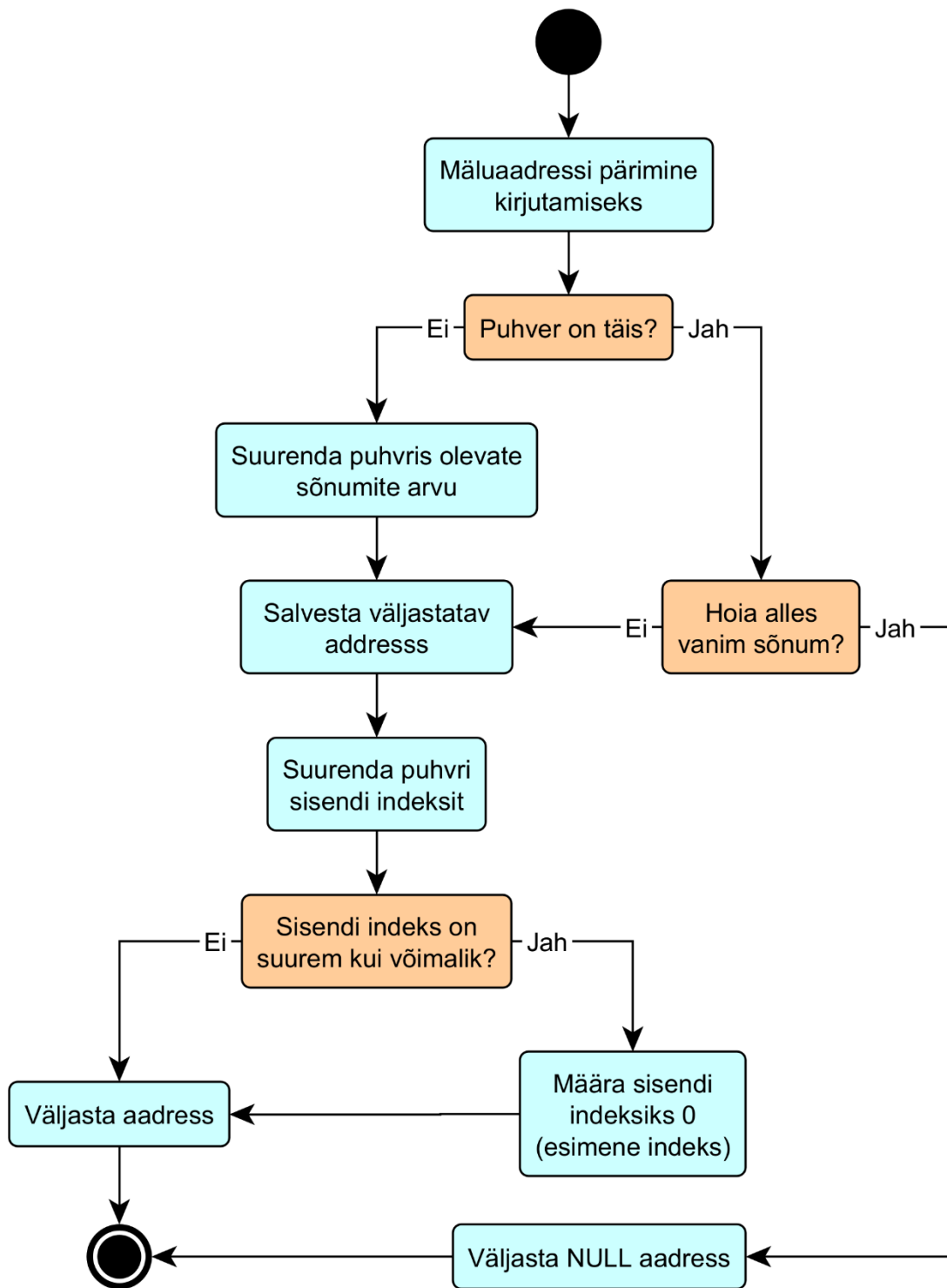
Tegumihalduri tuginemine lippudele ja riistvarapõhisele taimerile võimaldab ülesannete täpset ajastamist, tagades nende täitmise kindlate ajavahemike järel. Samas, ühes lõimes töötavad ülesanded peavad olema ajastatud ning kirjutatud nõnda, et need ei takistaks teiste ülesannete täitmist.

4.2.3 Ringpuhver

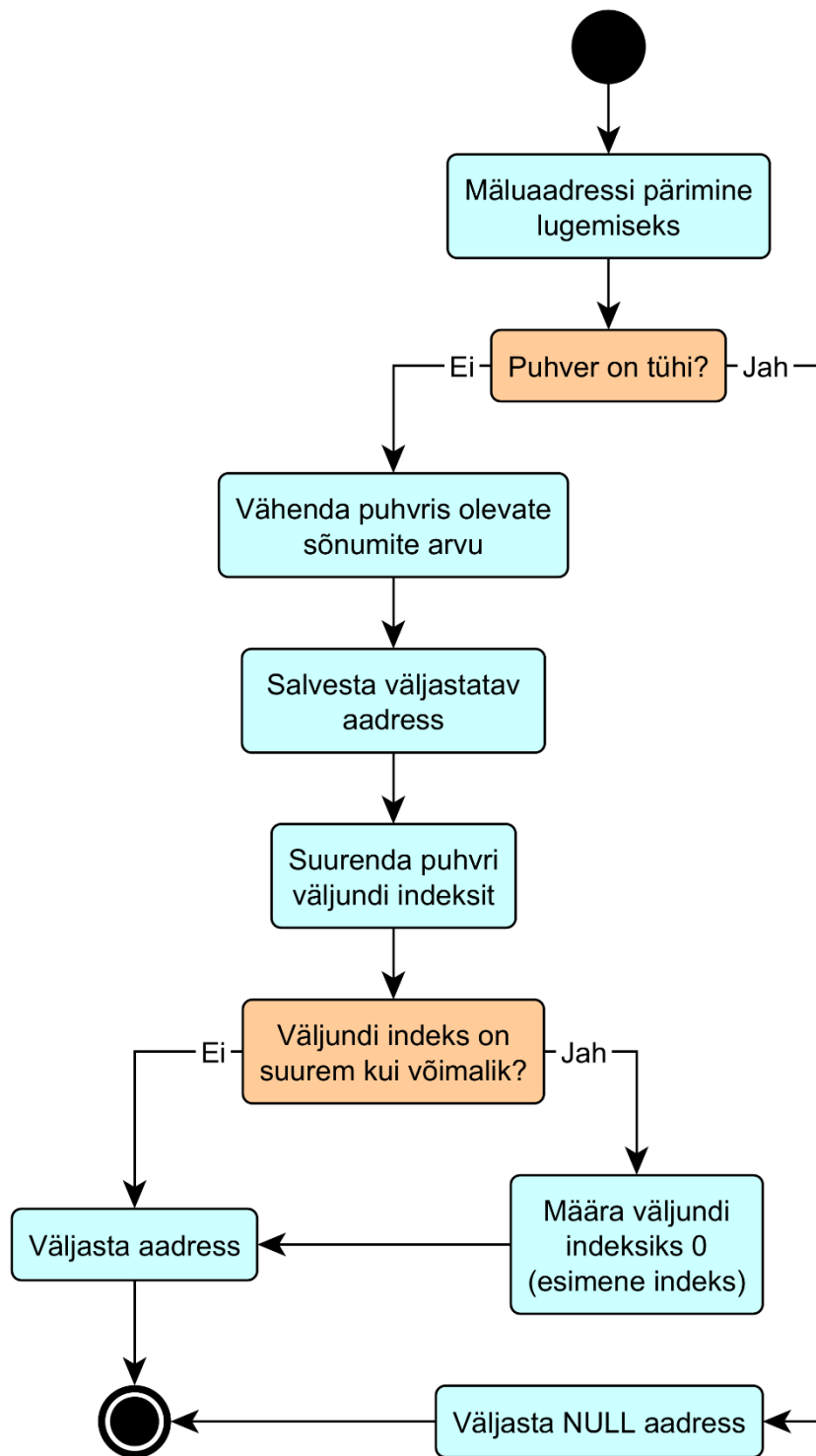
Mikrokontroller suhtleb seadmedraiveriga üle USB. Püsivara kasutab ringpuhvrit, et ajutiselt talletada draiverilt tulevaid sõnumeid ning draiverile saadetavaid sõnumeid, et neid sõnumeid saaks töödelda sobival ajal. Nii sissetulevate kui ka väljasaadetavate sõnumite jaoks on eraldi puhver. Ringpuhver on *First In, First Out* (FIFO) printsiipi järgiv andmestruktuur, millel on fikseeritud suurus andmete hoiustamiseks. Ringpuhvriil on üks mäluaadress uute andmete kirjutamiseks ning teine mäluaadress vanade andmete lugemiseks. Andmete kirjutamine ja lugemine käib puhvris ringiratast. Teisisõnu, mäluaadressi suurendatakse kuni jõutakse puhvri viimase aadressini. Seejärel kasutatakse taaskord puhvri esimest mäluaadressi.

Joonis 15 kujutab olukorda, kus püsivara soovib lisada puhvrise uue sõnumi. Püsivara küsib ringpuhvrit vaba koha mäluaadressi. Kui puhver ei ole täis, siis püsivara saab mäluaadressi, kuhu saab kirjutada uue sõnumi. Kui puhver on täis, siis tulenevalt püsivara seadistusest käitub ringpuhver kahel võimalikul viisil. Esiteks, kui vanade andmete ülekirjutamine on lubatud, siis ringpuhver väljastab kõige vanema sõnumi mäluaadressi, et sõnum kirjutataks üle. Teiseks, kui vanade andmete ülekirjutamine ei ole lubatud, siis ringpuhver väljastab tühja mäluaadressi (NULL), mis tähendab, et puhvrise ei saa lisada uut sõnumit, sest puhver on täis.

Joonis 16 kujutab olukorda, kus püsivara soovib lugeda puhvris olevat sõnumit. Püsivara küsib ringpuhvrit kõige vanema sõnumi mäluaadressi. Kui puhver on tühi, siis ringpuhver väljastab tühja (NULL) mäluaadressi.



Joonis 15. Mäluaadressi pärimine ringpuhvrisse kirjutamiseks.



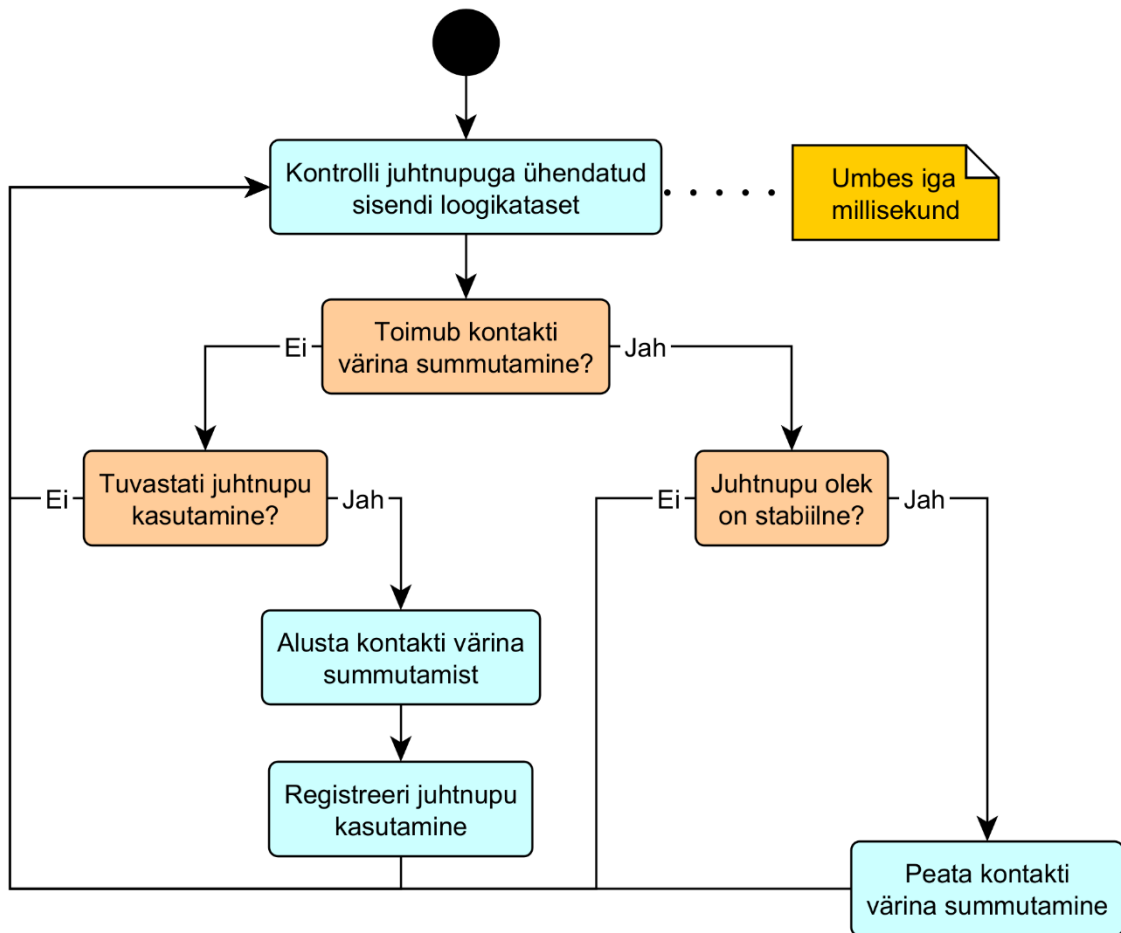
Joonis 16. Mäluaadressi pärimine ringpuhvrist lugemiseks.

4.2.4 Lüliti kontakti värina summutamine

Nagu eelpool mainitud, siis seadme klahvide ning pöördenkooderi kasutamisel esineb lülite kontakti värin, mille tõttu mikrokontroller saab klahvi vajutust või üht pöördenkooderi pööret tuvastada mitmekordselt.

Lüliti kontakti värin on nähtus, mis ilmneb mehaanilistes lülitites, kui neid kasutatakse. Kontakti värin on põhjustatud lüliti kontaktide füüsilisest ebakorrapärasusest. Kui lüliti vajutada või vabastada, võivad need ebakorrapärasused põhjustada rohkelt kontakti sulgumist ja katkemist, tekitades müra, mida elektroonika võib tõlgendada kui lüliti mitmekordset vajutust, enne kui lüliti jõuab stabiilsesse olekusse.

Joonis 17 kujutab juhtnupu kasutamise tuvastamise algoritmi. Püsivara kontrollib regulaarselt seadme juhtnupuga ühendatud mikrokontrolleri sisendi loogikataset. Kui püsivara tuvastab nupu kasutamise, siis alustatakse lüliti kontakti värina summutamist ning registreeritakse nupu kasutamine. Kui sisendi loogikatase on püsinud muutumatuna kindlaks määratud aja, siis summutamine lõpeb ning püsivara võimaldab registreerida antud juhtnupu järgmist kasutamist. Kui sisendi loogikatase muutub enne kindlaks määratud aja möödumist, siis algab ajamõõtmine uuesti.



Joonis 17. Juhtnupu kasutamise tuvastamine.

4.2.5 Seadmedraiveriga ühenduse kontrollimine

Seadmedraiver saadab seadmele regulaarselt kontrollsõnumeid, mille eesmärgiks on püsivarale märku anda, et mikrokontroller saab suhelda draiveriga. Kui püsivara ei ole saanud ühtegi kontrollsõnumit kindlaks määratud aja jooksul, siis püsivara peatab draiverile sõnumite saatmise ning seadme kõigi klahvide valgusdiodid hakkavad vilkuma. Järgmise kontrollsõnumi vastuvõtmisel jätkub seadme normaalne töö.

4.3 Seadmedraiver

Seadmedraiver on kirjutatud Javas. Draiveri peamiseks eesmärgiks on vahendada seadme ning veebibrauseri laienduse vahelist suhtlust.

4.3.1 Seadmega suhtlemine

Draiver kasutab seadmega USB kaudu suhtlemiseks teeki `javax.comm` [5], mis pakub jadaside jaoks platvormist sõltumatut liidest. Teek võimaldab sündmustepõhist tarkvara arhitektuuri. Näiteks, kui `javax.comm` tuvastab seadmelt sõnumi saamise, siis draiverit teavitatakse sellest sündmusest ning draiver lisab saadud sõnumi loendisse, et sõnum edastaks veebibrauserile.

Draiver kasutab seadmelt tulevate ning seadmele saadetavate sõnumite ajutiseks hoiustamiseks Javasse sisseehitatud andmestruktuuri `ConcurrentLinkedQueue` [33], mida kasutatakse lõimekindla (ingl. k. *thread-safe*), piiramata suurusega ja lingitud loendipõhise järjestuse (ingl. k. *linked list-based queue*) loomiseks. `ConcurrentLinkedQueue` järgib FIFO printsiipi. See andmestruktuuri valiti, sest sõnumite loendeid kasutavad mitmed eri lõimed ning antud andmestruktuur aitab vältida võistlusseisundit (ingl. k. *race condition*) ehk olukorda, kus programmi käitumine sõltub sündmuste suhtelisest ajastusest ja jagatud andmetele juurdepääsevad lõimed võivad põhjustada ootamatuid tulemusi.

Draiver kasutab seadme haldamiseks mitut regulaarselt läbiviidavat tegevust, mis kõik töötavad eri lõimedes, et vältida teiste ülesannete läbiviimise takistamist. Need tegevused põhinevad Java liidesel `ScheduledExecutorService` [34], mis on täituriteenus ülesannete ajastamiseks määratud ajavahemike järel. Üheks ülesandeks on seadmega USB ühenduse loomine, kui seda pole õnnestunud luua või ühendus on vahepeal katkenud. Teiseks ülesandeks on seadmelt saadud sõnumite edastamine brauseri laiendusele. Kolmandaks ülesandeks on veebibrauserilt saadud sõnumite edastamine seadmele. Viimaseks ülesandeks on seadmele eelpool mainitud kontrollsõnumi saatmine, mis tähistab seadme jaoks USB ühenduse olemasolu.

4.3.2 Veebibrauseri laiendusega suhtlemine

Draiver järgib brauseri laiendusega suhtlemisel brauseri sõnumside (ingl. k. *native messaging* [4]) protokoll. Sõnumite sisu on UTF-8 kodeeringus ning JavaScript *Object Notation* (JSON) vormingus.

Selleks, et saata laiendusele sõnum, kirjutab draiver oma programmi väljundisse (ingl. k. *standard output*) sõnumi päise ja sisu. Päiseks on saadetava sõnumi pikkus. Päise pikkuseks on neli baiti. Päis on pöördjärjestusega vormingus (ingl. k. *little-endian*). Päise esimeses baidis hoitakse sõnumi pikkuse arvu kõige madalama väärtusega bitte.

Brauseri laienduselt tulevate sõnumite vastuvõtmiseks loeb draiver regulaarselt programmi sisendisse (ingl. k. *standard input*) saadatud andmeid. Draiver arvutab esimese nelja saadud baidi põhjal vastuvõetud sõnumi pikkuse ning seejärel loeb välja sõnumi sisu. Seejärel võtab draiver sõnumist välja info, mis pannakse eespool mainitud loendisse, et see info siis juba edasi saata seadmele.

4.3.3 Logback

Draiver logib seadmelt ning veebibrauseri laienduselt saadud sõnumeid, USB ühenduse loomist ja sulgemist ning sõnumitega seotuid veateateid. Seadmedraiver kasutab logimise haldamiseks Java teeki Logback [35], mis pakub seadistatavat logimiskäitumist, toetab eri väljundeid ja logimistasemeid.

Kuna draiveri programmi väljundit kasutatakse brauseriga suhtlemiseks, siis logisõnumid tuleb suunata logifailidesse. Kui draiver käivitatakse päevas esimest korda, siis luuakse konkreetse kuupäevaga logifail. Kui logifaili suurus ületab 100 MB, siis luuakse uus fail ning eelmisele logifaili nimele lisatakse faili indeks. Logback on seadistatud hoidma kas kolme päeva jagu logifaile või kuni 500 MB logiandmeid. Kui 500 MB andmeid on täis, siis kustutatakse kõige vanem fail. Need andmemahud valiti nõnda, et programmis puuduse avastamisel oleks piisavalt logiandmeid silumise läbiviimiseks, aga samas ei kasutataks ülemäära salvestusseadme ruumi.

4.4 Veebibrauseri laiendus

Brauseri laienduse peamine eesmärk on hallata video taasesitust ning järjehoidjaid vastavalt seadmelt tulevatele sõnumitele. Laiendust saab hetkel kasutada vaid Firefoxis, sest lõputöö autor kasutab seda brauserit igapäevaselt. Loodud laiendust saab vajadusel ümber kirjutada Google Chrome's kasutamiseks [36].

4.4.1 Video haldamine

Kui brauseri laienduse sisuskript [7] laetakse veebilehel, siis esmalt peab laiendus leidma ülesse otsitava HTML5 video elemendi. Tulenevalt veebilehe tehnilisest ülesehitusest, võib video elemendi laadimine võtta kauem, kui laienduse laadimine. Sellest tulenevalt otsib laiendus video elementi hetkeseisuga kuni pool minutit. Video element loetakse leituks, kui video element on ekraanil visuaalselt nähtav (video mõõtmed on laiendusele kättesaadavad). Kui video elementi ei suudeta leida antud aja jooksul, siis laiendust ei saa kasutada selles olukorras. Video elemendi laadimise all ei peeta silmas kogu video sisu laadimist, vaid puhtalt video elemendi loomist veebilehel.

Video esituskiiruse määramiseks kasutatakse video elemendi sätet `playbackRate` [37], mis on numbriline väärtus. Veebibrauser korrutab video tavapärase esituskiiruse selle sätte väärtusega, et saada soovitud esituskiirus. Seadmelt on kolm klahvi video esituskiiruse kordaja valimiseks: 1, 1,5 ja 2. Laiendus ei võimalda muuta video esituskiirust aeglasemaks kui tavaline kiirus, sest see nõuaks mittesoovitud kompromisside tegemist. Kui kasutaja valib video esituskiiruse seadme abil, siis saadetakse seadmele sõnum, et põleks vaid vajutatud klahvi valgusdiod. Sel viisil saab kasutaja seadmelt tagasisidet tehtud valiku kohta.

Video edasi- ja tagasikerimiseks kasutab laiendus video elemendi sätet `currentTime` [38], mis tähistab video taasesituse ajahetke sekundites. Kui videot on vaja edasi kerida üks sekund, siis tuleb sellele sättele liita juurde üks. Tagasikerimiseks ühe sekundi jagu, tuleb lahutada sellest sättest üks. Kui sättele liita juurde piisavalt suur arv, et sätte väärtus oleks suurem kui video kestus, siis brauser määrab sätte väärtuseks video kestuse sekundites. Teisisõnu, sellises olukorras keritakse video täiesti lõppu. Kui sätte väärtusest lahutada piisvalt suur arv, et sätte väärtus oleks väiksem kui null, siis brauser määrab sätte väärtuseks nulli. Teisisõnu, video keritakse täiesti algusesse.

Kui kasutaja vajutab seadme pöördenkooderit ning järjehoidjate menüü ei ole avatud, siis laiendus peatab või paneb tööle video esituse. Selle jaoks kasutab laiendus video elemendi meetodeid `play` [39] ning `pause` [40].

4.4.2 Video järjehoidjad

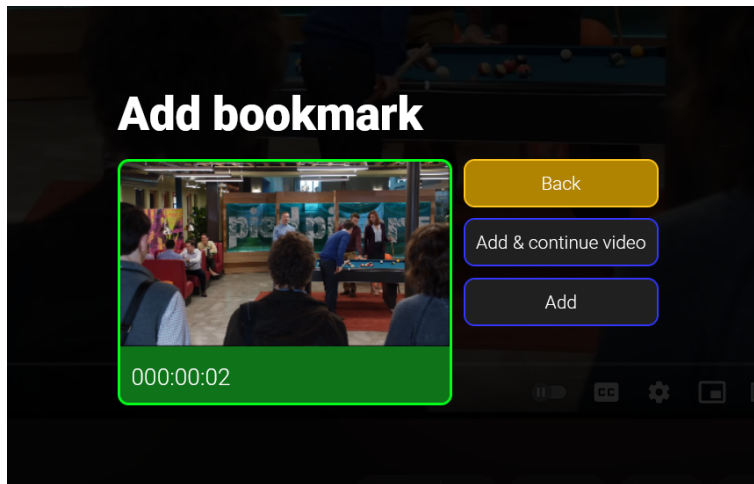
Joonis 18 on kujutatud järjehoidjate loendit. Pöördenkooderit pöörates saab liikuda erinevate valikute vahel ning pöördenkooderit vajutades saab teha valiku. Joonis 18 all vasakul nurgas on näha, et mida saab seadme klahvide abil teha.

Iga järjehoidja juures on kuvatud videokaadri pisipilt, et kasutaja leiaks lihtsamini üles otsitava järjehoidja. Pisipildi loomiseks peab brauseri laiendus üles seadma veebilehel HTML lõuendi (ingl. k. *canvas* [41]), et sinna saaks joonistada videokaadrist pildi. See lõuend on veebilehel visuaalselt peidetud. Kui kasutaja avab järjehoidjate menüü, siis ajutiselt salvestatakse mälli videokaadri pisipilt. Järjehoidja pisipildi laius on kõikide videote puhul sama, aga kuna erinevatel veebilehtedel võivad video mõõtmed erineda, siis iga video puhul arvutatakse pisipildi kõrgus vastavalt video reaalse laiuse ning pisipildi määratud laiuse suhtele, et loodav pilt oleks õiges proportsioonis. Kui pisipilt on joonistatud lõuendile, siis lõuendilt võetakse loodud pildi andmed, mis on PNG formaadis ning Base64 kodeeringus. Kui kasutaja otsustab salvestada antud kaadri enda järjehoidjate hulka, siis salvestatakse järjehoidja ajahetk ning pisipildi andmed brauseri laienduse salvestuskohta [8].

The image shows a YouTube interface with a video player and a list of related videos. The video player displays a sequence of thumbnails from the show 'Silicon Valley'. The first thumbnail is highlighted with a blue border and shows three men standing in a hallway. The second thumbnail is highlighted with a green border and shows a modern office interior with colorful beanbag chairs. The third thumbnail shows a group of people in a meeting room. The fourth thumbnail shows two people sitting at a table. The sidebar on the right lists various related videos, including SNL segments and tech-related content. A keyboard shortcuts menu is visible in the bottom left corner, listing F1 (Add bookmark), F2 (Scroll 10 up), and F3 (Scroll 10 down).

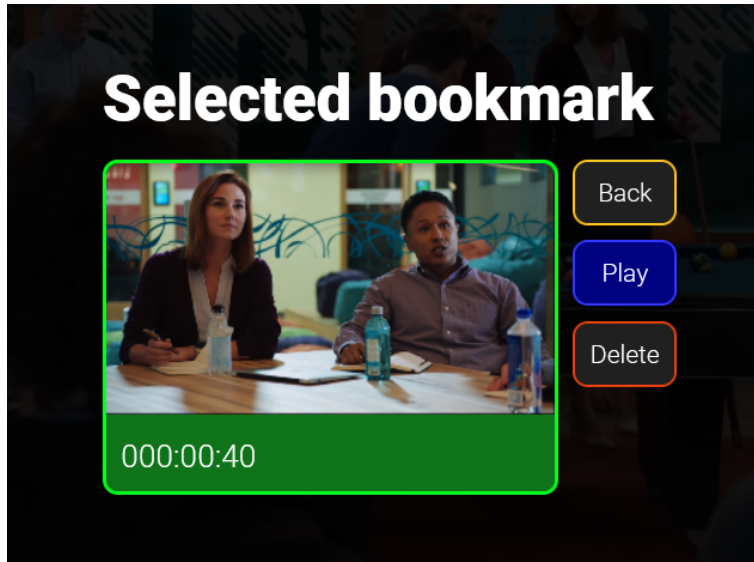
Joonis 18. Video järjehoidjate loend.

Järjehoidja lisamiseks avaneb Joonis 19 kujutatud vaade. Kasutajal on võimalus sulgeda see vaade, lisada järjehoidja ning jätkata video vaatamist või lisada järjehoidja ning automaatselt minna tagasi järjehoidjate menüüsse, et näiteks soovi korral valida sealt mingi järjehoidja.



Joonis 19. Järjehoidja lisamise vaade.

Järjehoidja kasutamiseks või kustutamiseks on Joonis 20 kujutatud vaade.



Joonis 20. Järjehoidja haldamise vaade.

4.4.3 Sündmuste juhtimine

Kui brauseri laienduse tarkvarakomponent annab teada sündmuse toimumisest, siis teised komponendid saavad sellele sündmusele reageerida ilma, et need komponendid peaksid suhtlema otse. See võimaldab kirjutada koodi paindlikumalt ning võimaldab vajadusel teha koodis lihtsamini muudatusi. Komponendid annavad sündmustest teada läbi vahendaja, millele on kõigil komponentidel koodis lihtne juurdepääs.

Laienduseülese sündmuste juhtimiseks loodi funktsionaalsus, mis kasutab brauseri EventTarget [42] liidest. Esiteks, komponendid saavad märku anda sündmustest. Teiseks, komponendid saavad jälgida sündmuseid ning määrata toimingut, mis viiakse läbi sündmuse esinemisel. Kolmandaks, komponent saab lõpetada sündmusele reageerimise.

4.4.4 Veebilehtede toetamine

Loodud brauseri laiendus suudab teoorias töötada igal veebilehel, kus esitatakse HTML5 videot. Praktikas piirab laienduse tööd veebilehe tehniline ülesehitus. Näiteks, lõputöö autor soovis võimaldada laienduse kasutamist Echo360 videoloengute jaoks. Echo360 võimaldab esitada ühe loengu raames paralleelselt mitut videot. Sellest tulenevalt haldab Echo360 neid videoid oma äranägemise järgi ning laiendus ei suuda (töökindlalt) kerida videot edasi ega tagasi ja ei suuda muuta video esituskiirust. Nende tehniliste puuduste tõttu ei toeta antud laiendus Echo360 lehel näidatavaid videoid.

Lisaks, brauseri laiendus peab küsima brauserilt luba, kui laiendus soovib veebilehtede sisu muuta või neid muul moel kasutada. Laiendus saab määrata, et see võib ligi pääseda igale veebilehele, mida kasutaja külastab, ootuses, et mõnel nendel veebilehtedel kuvatakse videoloenguid. Samas, selline lähenemine ei ole soositud privaatsuse kaalutlustel ei brauseri ega tõenäoliselt ka laienduse kasutaja poolt. Sellest tulenevalt saab laiendust kasutada hetkel YouTube's ning Coursera's.

4.4.5 Laienduse tootmine

Laiendus on kirjutatud TypeScriptis [14] ning Reacti [16] komponentide kirjapanemiseks kasutatakse JSX [43] süntaksit, et saaks lihtsamini kirjutada komponentides kasutatavat HTML koodi. Kuna veebibrauser toetab vaid JavaScripti, siis tuleb laienduse lähtekood kompileerida JavaScripti. Lisaks, laienduse lähtekood koosneb mitmest failist, aga brauseri laienduse sisuskript [7] peab olema saadaval ühe failina. Nendest asjaoludest

tulenevalt kasutatakse laienduse tootmiseks Webpack'i [20], mis suudab kompileerida TypeScripti JavaScriptiks ning panna mitme lähtefaili sisu kokku üheks failiks. Kuna laienduse kasutajaliideses kuvatakse ka ikoone, siis Webpack lisab väljundfaili ka nende ikoonide andmed nõnda, et brauser oskaks neid ikoone kuvada.

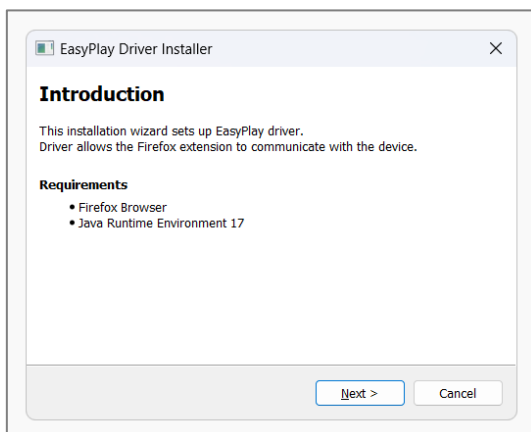
4.5 Tarkvarapaigaldaja

Lõputöö jaoks arendati tarkvarapaigaldaja, mille eesmärgiks on seadistada Firefox'i seadmedraiveriga suhtlemiseks ning kopeerida draiver kasutaja poolt valitud kausta. Paigaldaja on kirjutatud C++ ning kasutab Qt tarkvararaamistikku [23].

4.5.1 Nõuetele vastavuse kontrollimine

Tarkvarapaigaldaja kontrollib enne Joonis 21 kujutatud avakuvast edasiliikumist, et arvutisse oleks paigaldatud Firefox ning sobiv Java Runtime Environment (JRE) [44]. Firefox'i olemasolu kontrollimiseks pärib paigaldaja Windowsi registrist (Windows Registry) Firefox'i võtmete arvu kasutades QSettings klassi [45]. Kui võtmete arv on suurem kui 0, siis tarkvarapaigaldaja järeldeb, et Firefox on arvutisse paigaldatud.

Selleks, et tuvastada arvutisse paigaldatud JRE versioon, loob paigaldaja ajutiselt uue alamprotsessi kasutades klassi QProcess [46], et jooksutada käsku „java --version“. Protsessi väljundit võrreldakse regulaaravaldisega (ingl. k. *regular expression*), et saada kätte JRE versiooninumber. Kui arvutisse on paigaldatud JRE, mis võimaldab kasutada antud seadmedraiverit, siis tarkvarapaigaldaja lubab paigaldusega edasi liikuda.

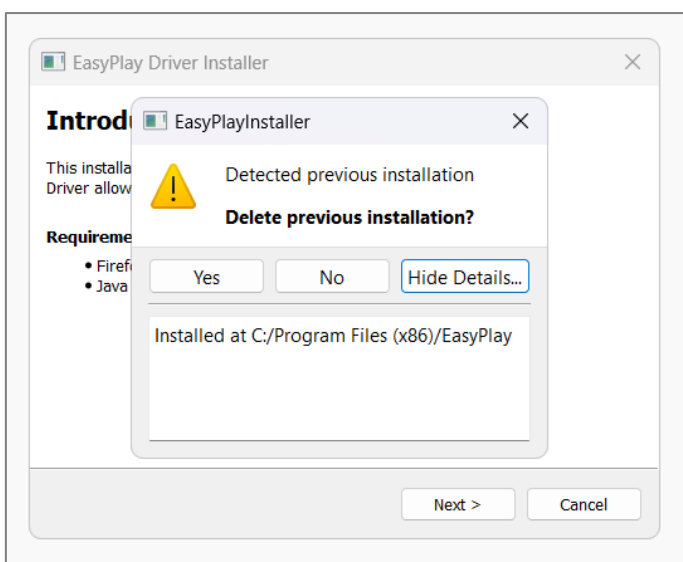


Joonis 21. Tarkvarapaigaldaja avakuva.

4.5.2 Sihткаusta valimine

Selleks, et seadmedraiverit saaks kasutaja soovi korral kopeerida näiteks Windowsi kausta *Program Files (x86)*, peab saama tarkvarapaigaldajat käivitada administraatori õigustega (ingl. k. *run as administrator*). See võimekus antakse paigaldamisprogrammile tootmise (ingl. k. *build*) käigus.

Joonis 22 on kujutatud olukorda, kus tarkvarapaigaldajat on arvutis juba varem kasutatud ning paigaldaja uuesti kasutamisel pakutakse automaatselt eelmisel korral paigaldatud failide eemaldamist. Tarkvarapaigaldaja teab kuhu on faile varem salvestatud, sest paigaldamise käigus salvestatakse Windowsi registrisse paigalduse sihtkausta aadress.



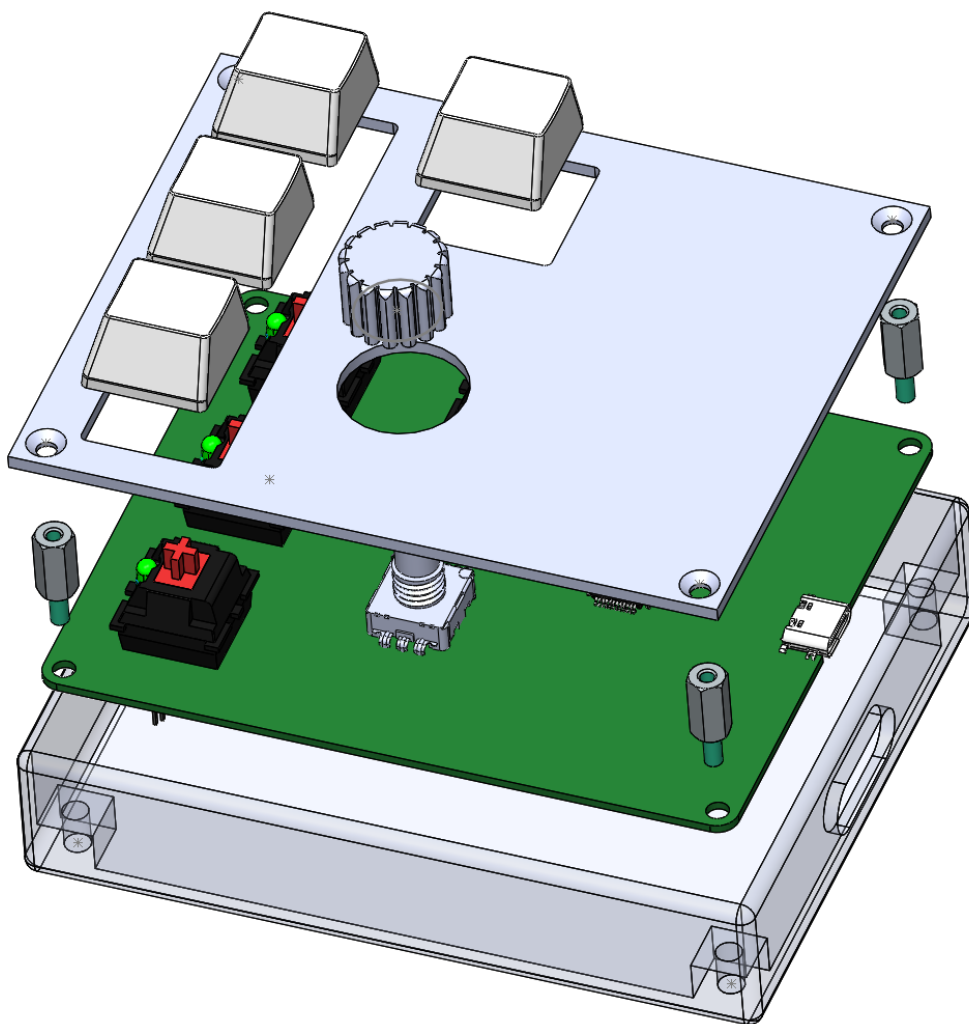
Joonis 22. Tarkvarapaigaldaja pakub vanade failide kustutamist.

4.5.3 Veebibrauseri register

Selleks, et veebibrauseri laiendus saaks suhelda seadmedraiveriga, peab brauser teadma, et kus asub draiveri programm, et brauser saaks programmi käivitada. Brauseri jaoks vajalik info on kirjas JSON vormingus olevas konfiguratsioonifailis [47]. Tarkvarapaigaldaja loob paigalduse käigus selle konfiguratsioonifaili. Seejärel salvestatakse konfiguratsioonifaili failitee (ingl. k. *file path*) Windowsi registrisse, et brauser leiaks ülesse vajamineva konfiguratsioonifaili.

4.6 Korpus

Seadme trükkplaadi jaoks disainiti SOLIDWORKS-i abil Joonis 23 kujutatud korpus, mis koosneb kahest detailist: karp ja kaas. Korpus on valmistatud 3D printeriga. Karbi põhjas on igas nurgas kuubikujuline kõrgendik, mille keskel on silindrikujuline süvend. Süvend on keermemuhvi (ingl. k. *threaded insert*) jaoks, et selle sisse saaks kinnitada M2,5 keermega distantspuksi (ingl. k. *standoff*), mille üks ots on isase keermega ning teine emasega. Kõrgendikku kasutatakse ka ruumi jätmiseks trükkplaadi ja karbi põhja vahele. Distantspuksi abil kinnitatakse trükkplaat kõrgendiku külge ning fikseeritakse kaas trükkplaadi kohale. Kaas kinnitatakse kruvidega distantspükside külge.



Joonis 23. Korpuse 3D mudeli lahti võetud vaade.

Lisaks disainiti Joonis 24 kujutatud pöördenkooderi nupukate ning lisati klahvide jaoks poolläbipaistvad katted, et näha klahvidele lisatud valgusdioode.



Joonis 24. Valmistatud seade.

4.7 Avastatud probleemid ja nende lahendused

Süsteemiarenduse käigus kontrolliti pidevalt süsteemi komponentide tööd, et tuvastada võimalikke puuduseid, mis takistavad süsteemi ootuspärast toimimist. Nii riistvara ega ka tarkvara testimiseks ei kasutatud automatiseeritud lahendusi.

Avastati, et teatud olukordades, kui anda seadmele toitevoolu, siis mikrokontroller ei hakanud tööle, kuigi alalispingemuundur suutis muuta sisendpinge 5 volti mikrokontrollerile vajaminevaks 3,3 voldiks. Kui labori toiteplokk ühendati otse trükkplaadi 3,3 voldise toiteühendusega, siis mikrokontroller hakkas ilusti tööle. Seade kasutas esimese trükkplaadi versiooni ajal pingemuundurit, mis võimaldas kuni 250 mA voolu. See pingemuundur asendati uuega, mis võimaldab kuni 1 A voolu. Uue komponendi jaoks pidi disainima ja tootma uue trükkplaadi, sest uue pingemuunduri korpus (SOT-223) erines eelmisest (SOT-23-3). Õnneks tasus uue trükkplaadi valmistamine ennast ära, sest seade töötab nüüd korralikult.

Seadme trükkplaat on disainitud selliselt, et juhtnuppude lülitite kontaktide värina summutamiseks saaks kasutada RC-filtrit. Nagu eelpool mainitud, siis avastati, et kasutatavate lülitite ja pöördenkooderi puhul osutus RC-filter kasutuks ning soovitud tulemuse saamiseks peab kasutama hoopis tarkvaralist lahendust.

Seadmedraiver kirjutas esialgu logisõnumeid programmi standardsesse väljundvoogu, mis lihtsustas draiveri testimist arendustöö käigus. Kui hakati tegelema brauseri laienduse arendamisega, siis selgus, et draiveri programmi väljundvoogu ei tohi kirjutada mitte midagi muud, kui brauseri laiendusele saadetavaid sõnumeid, sest muidu brauser peatab draiveri programmi. Selle tõttu seadistati draiver kirjutama logisõnumeid failidesse, et vajadusel saaks ikka jälgida draiveri töötamist.

Kui seadmedraiver on paigaldatud Windowsi kausta, kuhu failide lisamiseks on vaja administraatori õiguseid, siis draiver ei saa luua logifaili, sest brauser ei käivita draiveri programmi administraatori õigustega. Kui draiver ei saa luua logifaili, siis vaikimisi kirjutab Logback [35] programmi väljundisse veateate, mille tõttu brauser sulgeb programmi. Lahenduseks lisati Logbacki konfiguratsioonifaili käsk, et Logback ei kirjutaks programmi väljundisse ühtegi teadet.

Selleks, et brauseri laiendust saaks kasutaja püsivalt Firefox'i lisada, peab laiendus olema Mozilla poolt allkirjastatud. Allkirjastamiseks laetakse laienduse failid Mozilla teenusesse, et faile saaks automaatselt analüüsida tingimustele vastavaks. Kuna laienduse kasutajaliidese jaoks kasutati teeki React [16], siis laienduse faili lisati ka teegi kood. Allkirjastamise käigus tuvastati, et laienduse failis leidub sellist Reacti koodi, mis võimaldab veebilehte rünnata skriptisüstimisega (ingl. k. *cross-site scripting*). Õnneks laiendus ei vaja töötamiseks seda ebatavalist koodi ning need koodiread võib failist eemaldada. Lahenduseks kirjutati skript, mis automaatselt eemaldab laienduse tootmisel väljundfailist need koodiread.

5 Kokkuvõte

Lõputöö eesmärgiks oli luua seade, mis lihtsustab videoloengute vaatamist veebibrauseris. Lahenduse kasutaja saab juhtida seadme abil veebibrauseri laiendust, et muuta video esituskiirust, kerida videot edasi ning tagasi ja kasutada video järjehoidjaid. Lõputöö eesmärgi saavutamiseks loodi juhtnuppudega seade ning veebibrauseri laiendus. Selleks, et brauseri laiendus saaks suhelda seadmega, arendati seadmedraiveri. Lisaks valmistati seadmedraiveri ülesseadmiseks tarkvarapaigaldaja, et draiver saaks suhelda brauseriga.

Lõputöö projektile läheneti süstemaatilisest, pannes esmalt paika süsteemi arhitektuuri. Samas, süsteemi väljaarendamine nõudis loomingulisust ning ootamatute piirangutega tegelemist, kuna autor ei olnud varem tegelenud brauseri laienduse arendamisega, rääkimata laienduse ning kohandatud riistvara vahelise suhtluse ülesseadmise.

5.1 Soovitused tulevikuks

Seadmedraiveri programm on hetkel toodetud vaid Windowsi operatsioonisüsteemi jaoks. Autori soovitus edasiseks on toota draiveri programm ka Linuxi ning macOS peal, sest draiveri lähtekood on olemuselt platvormideülene. Samuti peaks sellisel juhul kompileerima tarkvarapaigaldaja nendes operatsioonisüsteemides ning tagama, et paigaldaja lähtub seadistuste tegemisel operatsioonisüsteemi omapäradest. Lisaks, Firefox'i laienduse saab teha sobivaks Google Chrome jaoks, kui kasutada teeki WebExtension API Polyfill [48] ning kohandada brauseri sõnumside kasutust [49].

Kasutatud kirjandus

- [1] „Keyboard shortcuts for YouTube“. <https://support.google.com/youtube/answer/7631406?hl=en> (vaadatud 1. mai 2023).
- [2] „Video Speed Controller“. Vaadatud: 1. mai 2023. [Online]. Available at: <https://github.com/igrigorik/videospeed>
- [3] „Smart Bookmarks for YouTube: Productivity Up“. <https://chrome.google.com/webstore/detail/smart-bookmarks-for-youtu/ichpfnleoiponjdhcbieonjdpnallmh?hl=en> (vaadatud 1. mai 2023).
- [4] Mozilla, „Native messaging“. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Native_messaging (vaadatud 19. aprill 2023).
- [5] „jSerialComm“. Fazecast, Inc. Vaadatud: 19. aprill 2023. [Online]. Available at: <https://fazecast.github.io/jSerialComm/>
- [6] Mozilla, „Background scripts“. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Background_scripts (vaadatud 19. aprill 2023).
- [7] Mozilla, „Content scripts“. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Content_scripts (vaadatud 19. aprill 2023).
- [8] Mozilla, „Extension storage“. <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/storage> (vaadatud 16. aprill 2023).
- [9] „Altium Designer“. Vaadatud: 20. aprill 2023. [Online]. Available at: <https://www.altium.com/altium-designer>
- [10] STMicroelectronics, „STM32F042C6“. <https://www.st.com/en/microcontrollers-microprocessors/stm32f042c6.html> (vaadatud 19. aprill 2023).
- [11] „STM32Cube MCU & MPU Packages“. STMicroelectronics N.V. Vaadatud: 16. aprill 2023. [Online]. Available at: <https://www.st.com/en/embedded-software/stm32cube-mcu-mpu-packages.html>
- [12] „Overview of Java“, *Java Developer’s Guide*, Vaadatud: 19. aprill 2023. [Online]. Available at: <https://docs.oracle.com/en/database/oracle/oracle-database/21/jjdev/Java-overview.html>

- [13] „Java Garbage Collection Basics“.
<https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>
(vaadatud 12. mai 2023).
- [14] Microsoft, „TypeScript“. <https://www.typescriptlang.org/> (vaadatud 19. aprill 2023).
- [15] „A Typed Superset of JavaScript“, *TypeScript for the New Programmer*.
<https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html#a-typed-superset-of-javascript> (vaadatud 19. aprill 2023).
- [16] „React“. Meta. Vaadatud: 16. aprill 2023. [Online]. Available at:
<https://react.dev/>
- [17] „Sass“. Vaadatud: 19. aprill 2023. [Online]. Available at: <https://sass-lang.com/>
- [18] „Node.js“. Vaadatud: 19. aprill 2023. [Online]. Available at:
<https://nodejs.org/en>
- [19] „NPM“. Vaadatud: 19. aprill 2023. [Online]. Available at:
<https://www.npmjs.com/>
- [20] „Webpack“. Vaadatud: 19. aprill 2023. [Online]. Available at:
<https://webpack.js.org/>
- [21] „ts-loader“. Vaadatud: 13. mai 2023. [Online]. Available at:
<https://www.npmjs.com/package/ts-loader>
- [22] Mozilla, „JavaScript APIs“. <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API> (vaadatud 20. aprill 2023).
- [23] „Qt“. Vaadatud: 20. aprill 2023. [Online]. Available at: <https://www.qt.io/>
- [24] „Signals & Slots“. <https://doc.qt.io/qt-6/signalsandslots.html> (vaadatud 20. aprill 2023).
- [25] „SOLIDWORKS“. Dassault Systèmes SE. Vaadatud: 20. aprill 2023. [Online].
Available at: <https://www.solidworks.com/>
- [26] „View Customer Stories by Industry“.
<https://www.solidworks.com/sw/successes/customer-story-industries.htm> (vaadatud 20. aprill 2023).
- [27] „STM32F042x6 andmeleht“. Vaadatud: 20. aprill 2023. [Online]. Available at:
<https://www.st.com/resource/en/datasheet/stm32f042c6.pdf>
- [28] „IP4234CZ6“. Vaadatud: 12. mai 2023. [Online]. Available at:
<https://www.mouser.ee/datasheet/2/916/IP4234CZ6-2937992.pdf>

- [29] „PEC11H Series“. Vaadatud: 20. aprill 2023. [Online]. Available at:
<https://www.bourns.com/docs/Product-Datasheets/PEC11H.pdf>
- [30] „PEC11R Series“. Vaadatud: 12. mai 2023. [Online]. Available at:
<https://www.farnell.com/datasheets/3717670.pdf>
- [31] „LD1117A“. Vaadatud: 12. mai 2023. [Online]. Available at:
<https://www.st.com/content/ccc/resource/technical/document/datasheet/a5/c3/3f/c9/2b/15/40/49/CD00002116.pdf/files/CD00002116.pdf/jcr:content/translations/en.CD00002116.pdf>
- [32] „Differential Pair Length Matching: Best Practices for Signal Integrity“. <https://resources.altium.com/p/differential-pair-length-matching-best-practices-signal-integrity> (vaadatud 12. mai 2023).
- [33] „ConcurrentLinkedQueue“. Vaadatud: 26. aprill 2023. [Online]. Available at:
<https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ConcurrentLinkedQueue.html>
- [34] „ScheduledExecutorService“. [Online]. Available at:
<https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ScheduledExecutorService.html>
- [35] „Logback“. Vaadatud: 26. aprill 2023. [Online]. Available at:
<https://logback.qos.ch/index.html>
- [36] „Porting a Google Chrome extension“. <https://extensionworkshop.com/documentation/develop/porting-a-google-chrome-extension/>
- [37] „HTMLMediaElement: playbackRate property“. <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/playbackRate> (vaadatud 27. aprill 2023).
- [38] „HTMLMediaElement: currentTime property“. <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/currentTime> (vaadatud 27. aprill 2023).
- [39] „HTMLMediaElement: play() method“. <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/play> (vaadatud 27. aprill 2023).
- [40] „HTMLMediaElement: pause() method“. <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/pause> (vaadatud 27. aprill 2023).
- [41] „Canvas API“. https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API (vaadatud 27. aprill 2023).

- [42] „EventTarget“. <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget> (vaadatud 27. aprill 2023).
- [43] „JSX“. <https://www.typescriptlang.org/docs/handbook/jsx.html> (vaadatud 27. aprill 2023).
- [44] „What is the Java Runtime Environment (JRE)?“
<https://www.ibm.com/topics/jre> (vaadatud 28. aprill 2023).
- [45] „QSettings Class“. <https://doc.qt.io/qt-6/qsettings.html> (vaadatud 28. aprill 2023).
- [46] „QProcess Class“. <https://doc.qt.io/qt-6/qprocess.html> (vaadatud 28. aprill 2023).
- [47] „Native messaging manifests“. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Native_manifests#native_messaging_manifests (vaadatud 30. aprill 2023).
- [48] „WebExtension API Polyfill“. Vaadatud: 1. mai 2023. [Online]. Available at: <https://github.com/mozilla/webextension-polyfill>
- [49] „Chrome incompatibilities: Native messaging“. https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Chrome_incompatibilities#native_messaging (vaadatud 1. mai 2023).

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Karl-Heinrich Tamm

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Videoloengute vaatamist lihtsustava seadme väljatöötamine“, mille juhendaja on Kaiser Pärnamets
 - 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.

13.05.2023

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.

Lisa 2 Lõputöö repositoorium GitHubis

Lõputöö repositoorium GitHubis asub veebiaadressil github.com/karltamm/EasyPlay.