

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Ilja Obuhov 213437IACB

# **USB-HID JÕUTAGASISIDEGA TESTSEADE**

Bakalaureusetöö

Juhendaja: Andres Rähni

Lektor

Tallinn 2024

## **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: Ilja obuhov

01.05.2024

## **Annotatsioon**

Antud lõputöö keskendub USB-HID seadme arendamisele ja rakendamisele, millel on sisseehitatud võimekus jõutagasiside toetamiseks. Sellistel seadmetel on oluline roll mitmesugustes valdkondades, kus inimese ja arvuti või teiste seadmete vaheline suhtlus on oluline, nagu näiteks mängutööstus, virtuaalreaalsus, haridus, meditsiin ja tööstus.

Töö eesmärgiks on arendada USB-HID testseade, mis kuulub PID-klassi, võimaldades suhelda arvutiga USB-HID protokolliga kaudu ja tagades samal ajal võime jõutagasisidet edastada ja vastu võtta.

Tulemuseks on valminud füüsilise liidese tüüpi USB-HID seade, mida arvuti tuvastab kui mänguseadet, millel on sisseehitatud võimekus jõutagasiside toetamiseks. Testseade suudab töödelda ja esitada arvutist saadud efekte ning samuti teavitada arvutit nende efektide hetkeseisust läbi USB-HID protokolliga.

Selle töö tulemust on võimalik edasi arendada, lisades seadmele rohkem efekte, asendades mootori ja mootoridraiveri võimsamate vastu, samuti lisades täiendavaid juhtnuppe nagu pedaalid, käigukast või mõni muu haptiline liideseade, mis aitavad kasutajal paremini tunnetada virtuaalset või kaugget keskkonda, sellesse rohkem suhestuda.

Seletuskiri pakub kõike olulist hõlmava ülevaate USB-HID seadmete arendamise jaoks ja jõutagasisidet toetava testseadme arendamise praktilise näite.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 42 leheküljel, 5 peatükki, 11 joonist.

## **Abstract**

### **Implementation of a USB-HID test device with force feedback**

This thesis focuses on the development and implementation of a USB-HID device with built-in support for force feedback. Such devices play a significant role in various fields where interaction between humans and computers or other devices is essential, such as the gaming industry, virtual reality, education, medicine, and industry.

The aim of the work is to develop a USB-HID test device that belongs to the PID class, allowing communication with a computer via the USB-HID protocol and at the same time providing the ability to transmit and receive force feedback.

The result is a physical interface type USB-HID device that is recognized by a computer as a gaming device with built-in capability to support force feedback. The test device can process and play the effects received from the computer and also inform the computer about the current status of these effects through the USB-HID protocol.

It is possible to develop an result of this work further by adding more effects to the device, replacing the motor and motor driver with more powerful ones, as well as adding additional controls such as pedals, a gearbox or some other haptic interface device that helps the user to better feel the virtual or remote environment, to relate to it more.

The white paper provides an all-important overview for developing USB-HID devices and a practical example of developing a test device that supports force feedback.

The thesis is in estonian and contains 42 pages of text, 5 chapters, 11 figures.

## Lühendite ja mõistete sõnastik

I2C	<i>Inter-Integrated Circuit</i> , interintegreeritud ahel
USB	<i>Universal Serial Bus</i> , universaalne seeriaside liides
USB D	<i>Universal Serial Bus Device</i> , universaalne seeriaside liidese seade
HID	Human interface device, inimliidese seade
PID	Physical Input Device, füüsiline sisendseade
IDE	Integrated Development Environment, integreeritud arenduskeskkond
STM	Semiconductor Technologies Microelectronics, Semiconductor Technologies Mikroelektronika
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor, Metalloksiid-pooljuhtide väljatransistor
MIDI	Musical Instrument Digital Interface, muusikariistade digitaalne liides
PWM	Pulse-Width Modulation, impulsi laiuse modulatsioon
ARM	<i>Advanced RISC Machine</i> , täiustatud RISC-masin
RISC	<i>Reduced Instruction Set Computer</i> , vähendatud juhiskomplektiga arvuti
ID	Identifier, identifikaator
RAM	<i>Random Access Memory</i> , suvapöördusmälu ehk muutmälu
ROM	<i>Read-only memory</i> , ainult loetav mälu ehk püsिमälu

# Sisukord

Sissejuhatus .....	9
1 Tagasisidega USB-HID seadmete kasutusvaldkonnad.....	10
1.1 Mängutööstus.....	10
1.2 Virtuaalreaalsus .....	10
1.3 Meditsiinitehnoloogia.....	11
1.4 Tööstus- ja robotitehnilised rakendused .....	11
1.5 Haridus ja õpe.....	11
1.6 Meediarakendused ja meelelahutus .....	12
2 Testseadme komponentide ülevaade .....	13
2.1.1 Mootor .....	13
2.1.2 Toiteplokk.....	14
2.1.3 Mootori juhtahel .....	14
2.1.4 Kontroller .....	14
2.1.5 Pööramisnurga andur.....	14
2.2 Kasutatav tarkvara .....	15
2.2.1 Arenduskeskkond .....	15
2.2.2 Testimisprogramm.....	15
3 USB-HID seadme tööpõhimõte ja struktuur .....	16
3.1 Tööpõhimõte.....	16
3.1.1 HID-klassi seadme suhtlus .....	16
3.1.2 Füüsiline sisendseade (PID) .....	21
3.2 Arendatud testseadme tarkvara struktuur .....	28
4 Testseadme realiseerimine.....	30
4.1 Ettevalmistamine realiseerimiseks .....	30
4.2 Raporti deskriptor .....	31
4.3 Parameetrite ploki haldamine .....	31
4.4 <i>Out</i> - ja <i>Get</i> -raportite vastuvõtmine.....	32
4.5 USB-siini andmete vastuvõtt ja saatmine.....	32

4.6 Rooli telje asendi saatmine USB siini kaudu.....	33
4.7 Raporti haldur .....	33
4.8 Efektide haldur .....	34
5 Seadme testimine.....	36
Kokkuvõte .....	38
Kasutatud kirjandus .....	39
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	41
Lisa 2 - Testseadme lõplik vaade .....	42

## Jooniste loetelu

Joonis 1. Komponentide ühendusskeem plok skeemina.....	13
Joonis 2. Deskriptorite struktuur USB-HID seadme jaoks.....	17
Joonis 3. Raport deskriptor parseri vaatepunktist [4].....	18
Joonis 4. Raporti struktuuri näide raporti suurusega.....	19
Joonis 5. Raporti suuruse pärsimise näide.....	20
Joonis 6. Seadme struktuur plok-skeemina.....	29
Joonis 7. Get-raport haldamise meetodi algoritm.....	32
Joonis 8. Raport haldamise meetodi algoritm.....	34
Joonis 9. Efekti haldamise meetodi algoritm.....	35
Joonis 10. Testimise programm mis kuvab andmeid seadmest.....	36
Joonis 11. Testimise programm kuvab efekti parameetrid.....	37



## Sissejuhatus

USB-HID seadmed jõutagasisidega mängivad olulist rolli erinevates valdkondades, kus on vajalik inimese suhtlemine arvuti või teiste seadmetega. Jõutagasiside võimaldab kasutajatel tunda füüsilist reaktsiooni oma tegevustele, suurendades elamuslikkust ja tõhusust erinevates rakendustes, sealhulgas mängutööstuses, virtuaalses reaalsuses, hariduses, meditsiinis ja tööstuses.

Antud lõputöö keskendub USB-HID seadmete jõutagasiside kasutamisele, sealhulgas USB-HID-protokolli PID-klassi uurimisele ja USB-HID-testseadme raportideskriptori rakendamisele. Lisaks hõlmab see kontrolleri seadistamist *Custom* USB-HID-na, koodi kirjutamist vastavalt raportites kirjeldatud deskriptoritele, algoritmide arendamist erinevate tüüpide tagasiside töötlemiseks ning tagasiside implementeerimist seadmestamist.

Töö käigus viidi läbi lühike uurimus nende seadmete rakendusvaldkondade kohta, uuriti USB-HID seadmete põhimõtteid ja töömeetodeid ning arendati ja koostati jõutagasisidega rooli testseade simulaatoritele.

Eesmärkide saavutamiseks oli tehtud mitmeid tööetappe. Esiteks uuriti USB-HID seadmete jõutagasiside rakendusvaldkondi, kaaludes olemasolevaid lahendusi ja tehnoloogiaid ning arutledes peamiste rakendusvaldkondade üle. Järgnevalt arendati ja rakendati simulaatorite jaoks jõutagasisidega mängurooli testseadme, võttes aluseks saadud teadmised. Seejärel kirjutati arvutiga suhtlemiseks vajalik raportideskriptor ja andmevahetuse loogika USB-HID protokolli kasutades. Efektide töötlemise ja mootori juhtimise rakendamine oli järgmine samm, kus loodi loogika efektide töötlemiseks ja mootori juhtimiseks vastavalt neile efektidele, võimaldades jõutagasiside rakendamist. Lõpuks viidi läbi testimine, mis hõlmas jõutagasiside efektide toimimise kontrollimist. See töö aitab kaasa interaktiivsete seadmete tehnoloogiaga tutvumisel ning pakub praktilist näidet USB-HID seadme arendamisest koos jõutagasisidega.

# 1 Tagasisidega USB-HID seadmete kasutusvaldkonnad

USB-HID seadmed koos tagasisidega mängivad olulist rolli paljudes valdkondades, kus on vajalik inimese interaktsioon arvuti või muude seadmetega. Tagasiside võimaldab kasutajatel tunnetuslikku tagasisidet, mis parandab elamuslikkust ja efektiivsust erinevates rakendustes. Selles uurimuses vaatleme erinevaid USB-HID seadmete kasutusvaldkondi koos tagasisidega ja nende näiteid.

## 1.1 Mängutööstus

Mängutööstus on üks peamisi USB-HID tagasisideseadmete tarbijaid. Selles valdkonnas kasutatakse tunnetuslikku tagasisidet mängijate kaasamise suurendamiseks ja realistlikuma mängukogemuse loomiseks. Rakendusnäited [1]:

- Tagasiside jõud võimaldab mängijatel tunda teekatet, juhtida autot või lennukit arvestades vastupanu ja teisi tegureid.
- Vibratsioon ja muud jõu tagasiside vormid kasutatakse erinevate sündmuste edastamiseks mängus, nagu lasud, plahvatused või löögid.

## 1.2 Virtuaalreaalsus

Virtuaalreaalsuses muutub tagasiside veelgi olulisemaks, kuna see aitab suurendada kasutaja kohalolu tunnet virtuaalses maailmas. Rakendusnäited [2] [3]:

- USB-HID-seadmed tagasisidega on integreeritud virtuaalreaalsuse kontrolleritesse, võimaldades kasutajal tunda vastupanu, tekstuure ja muid objekte virtuaalses maailmas.
- Virtuaalsed simulaatorid, nagu lendamise või võidusõidu simulaatorid, kasutavad tagasisidega seadmeid realistlike tingimuste loomiseks ja kasutajate juhtimisoskuste õpetamiseks.

### **1.3 Meditsiinitehnoloogia**

Meditsiinitehnoloogias võib tagasiside olla kasutusel kirurgide koolitamiseks, manipulatsioonide simuleerimiseks ja protseduuride täpsuse parandamiseks. Kasutusnäide [4]:

- Tagasisidega USB-HID seadmeid kasutatakse meditsiinikoolide üliõpilaste õpetamiseks ja kogunud kirurgide treenimiseks, võimaldades neil tunda jõudu ja vastupanu virtuaalsete operatsioonide tegemisel.

### **1.4 Tööstus- ja robotitehnilised rakendused**

Tööstuses ja robotitehnoloogias võib tagasiside parandada protsesside juhtimist ja ohutust. Rakendusnäited [4]:

- Tagasisidet kasutavad USB-HID-seadmed tööstuslike robotite juhtimisel, võimaldades operaatoritel tunda vastupanu ja jõudu, mida robot ülesande täitmisel kogeb.
- Kirurgias ja teistes telerobootika valdkondades võib tagasiside aidata operaatoritel tunda kaugust ja jõudu kaugjuhitavate seadmete juhtimisel.

### **1.5 Haridus ja õpe**

Tagasisidega USB-HID seadmed mängivad olulist rolli hariduse ja õppe valdkonnas, pakkudes taktilist kogemust, mis parandab õppematerjali omastamist ja oskuste arendamist. Rakendusnäited [2]:

- Õppeprogrammides ja mängudes lastele kasutatakse tagasisidega USB-HID seadmeid interaktiivsete keskkondade loomiseks, kus nad saavad tunda füüsilist reaktsiooni oma tegevustele, mis soodustab sügavamalt materjali mõistmist.
- Erinevates valdkondades, nagu lennundus, autotööstus, meditsiin ja teised, aitab tagasiside õpilastel ja spetsialistidel treenida realistlikes tingimustes, mis parandab nende erialaseid oskusi.

## 1.6 Meediarakendused ja meelelahutus

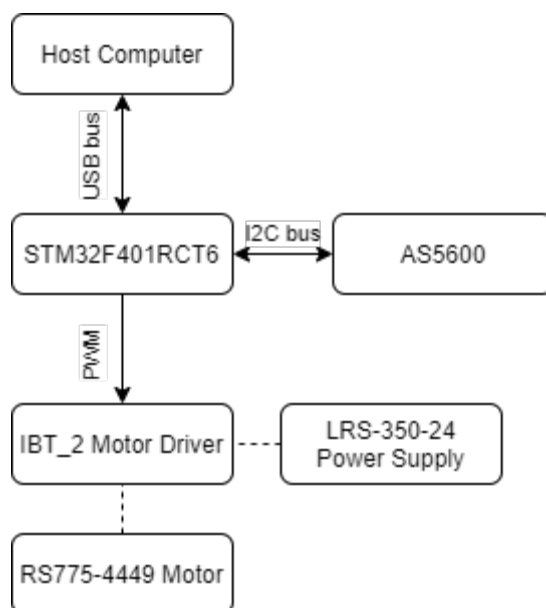
USB-HID tagasisidega seadmed lisavad meelelahutus- ja multimeediumivaldkonnale uusi väljavaateid, muutes kasutajakogemuse haaravamaks ja kaasahaaravamaks. Rakendusnäited [4]:

- Muusikariistad ja kontrollid, nagu näiteks MIDI-klaviatuurid, saavad tagasisidet kasutada muusikutele taktilise tagasiside pakkumiseks nende mängu või miksimise osas.
- Virtuaalsed keskkonnad, kus kasutajad saavad kogeda reaalsete instrumentide mängimise tunnet või lavale astumist, muutuvad tänu tagasisidele reaalsemaks ja kaasakiskuvamaks.
- USB-HID tagasisidega seadmeid saab kasutada interaktiivsetes installatsioonides ja kunstilistes meelelahutustes, kus kasutajad saavad suhelda kunstiteostega ja tunda nende mõju oma kehale.

USB-HID seadmetel tagasisidega on mitmekülgsed rakendused erinevates valdkondades, sealhulgas mängutööstuses, virtuaalreaalsuses, meditsiinitehnikas, tööstuslikes ja robotrakendustes. Need seadmed võimaldavad kasutajatel tunda vastupanu, tekstuure ja muid füüsilisi omadusi virtuaalses või kaugkeskkonnas, parandades nende suhtlust arvuti ja teiste seadmetega. Tehnoloogiate arengu ja rakendusvaldkondade laienemisega võib oodata nende seadmete jätkuvat nõudluse kasvu.

## 2 Testseadme komponentide ülevaade

Järgmises peatükis on kirjeldatud kõik komponendid ja tööriistad, mida kasutati selle töö käigus projekti arendamiseks ja elluviimiseks. Eriti kirjeldatakse tarkvara, mida kasutatakse kontrolleri programmeerimiseks ja testseadme testimiseks, samuti komponente nagu andur ja mikrokontroller, mootor, toiteplokk, mootori juhtahel. Kõikide komponentide ühendusskeemi on võimalik näha Joonisel 1, mis on toodud allpool:



Joonis 1. Komponentide ühendusskeem plokk skeemina

### 2.1.1 Mootor

Täitmismehhanismina kasutatakse 12 V alalisvoolu elektrimootorit RS775-4449. Sellel mootoril on järgmised omadused [5]:

- Toitepinge: 12-36 V
- Nominaalne kiirus: 4000-12000 p / min
- 12V koormusvool: 1,7A (algusvool 2,2A)
- Seiskumisvool: 4,7A

Selle mootori valimine on tingitud selle kompaktsusest, piisavast kiirusest ja piisavalt kõrgest voolust, mis tähendab, et mootor ei põle läbi koormuse all, mis tavaliselt rakendatakse jõu tagasisidega roolile. Lisaks suurema pöördemomendi saavutamiseks kasutati reduktorit, mille ülekanne oli 21:1.

### **2.1.2 Toiteplokk**

Toiteplokkina kasutatakse LRS-350 seeria toiteplokki, täpsemalt LRS-350-24 mudelit, sest see katab mootori toitevajadused varuga. Toiteploki omadused on järgmised [6]:

- Otsese voolu pinge: 24 V
- Nominaalne voolutugevus: 14,6 A
- Nominaalne võimsus: 350,4 W

### **2.1.3 Mootori juhtahel**

Mootori juhtimiseks kasutatakse IBT\_2 draiverit. See draiver on moodul, mis põhineb MOSFET-transistoril BTS7960. IBT\_2 draiveril on järgmised omadused [7] [8]:

- Toitepinge: 7–24 V
- Maksimaalne vool: 43 A
- PWM sagedus: kuni 25 kHz

Valitud sai see draiver, sest see on hea töökindlusega, lihtne kasutada ja sisseehitatud ülekuumenemiskaitsega.

### **2.1.4 Kontroller**

Kontrollerina kasutatakse mikrokontrollerit STM32F401RCT6. Sellel mikrokontrolleril on järgmised omadused [9]:

- Tuum: ARM Cortex-M4
- Taktsagedus: 84 MHz
- Flash-mälu maht: 256 KB
- RAM-mälu maht: 64 KB
- Liidesed: USB, I2C, SPI, UART, SDIO, ADC, DAC

Antud mikrokontrolleril on antud rakenduse jaoks piisav jõudlus, mälumaht ja vajalikud liidesed. Samuti oli autoril sellega varasem kasutajakogemus.

### **2.1.5 Pööramisnurga andur**

Rooliratta pöördenurga määramiseks kasutatakse AS5600 andurit. See andur põhineb Halli efektil ja omab järgmisi omadusi [10]:

- Mõõteulatus: 0–360°
- Resolutsioon: 12 biti
- Täpsus:  $\pm 1^\circ$
- Liides: I2C

Selle anduri valik on tingitud selle täpsusest, kompaktsusest ja ka I2C-liidest, mis lihtsustab selle ühendamist mikrokontrolleriga.

## **2.2 Kasutatav tarkvara**

### **2.2.1 Arenduskeskkond**

Arenduskeskkonnana valiti STM32CubeIDE versioon 1.9.0, kuna see pakub integreeritud arenduskeskkonda koos laia valikuga tööriistadega STM32 mikrokontrollerite programmeerimiseks. STM32CubeIDE pakub mugavat liidest tarkvara arendamiseks ja silumiseks ning integreerib CubeMX-i teekidega, et genereerida koodi valitud perifeersetes seadmetes ja mikrokontrolleri pin-konfiguratsiooni põhjal. See teeb selle heaks valikuks USB-HID-seadme arendamiseks STM32F401RCT6 mikrokontrolleril.

### **2.2.2 Testimisprogramm**

Põhiprogrammina seadme testimiseks kasutati programmi nimega iRacing Force Feedback Test 1.72. See programm on spetsiaalselt loodud jõu tagasiside testimiseks ja seadistamiseks mängusimulaatorites, pakkudes mugavat viisi seadme funktsionaalsuse ja suhtlemiskvaliteedi kontrollimiseks mängutarkvaraga. Selle programmi kasutamine võimaldas teha põhjalikku jõu tagasiside testimist ja kinnitada STM32F401RCT6 mikrokontrolleril loodud USB-HID-seadme õiget toimimist. Lisaks on see programm võimeline kuvama efektide praegust olekut, pakkudes seadme töö analüüsimise ja silumise võimalust [11].

## 3 USB-HID seadme tööpõhimõte ja struktuur

### 3.1 Tööpõhimõte

USB-seadme teave salvestatakse ROM-segmenDis. Neid segmente nimetatakse deskriptoriteks. Arvuti (edaspidi host) tuvastab seadme klassi, lugedes liidesedeskriptori sisu ja rakendab klassi jaoks spetsiaalseid draivereid, et töödelda andmeid. Arendatava seadme puhul on selleks USB-HID klass. Andmete töötlemine toimub seadmel esitatud deskriptorite lugemise kaudu.

HID-klassi seadme deskriptor tuvastab, millised muud HID-klassi deskriptorid on olemas ja näitab nende suurusi. Näiteks raportideskriptorid.

#### 3.1.1 HID-klassi seadme suhtlus

HID-klassi seade suhtleb HID-klassi juhi abil kas kasutades juhtimis (vaikimisi) toru või katkestustoru.

Juhtimistoru kasutatakse järgmiseks:

- USB-juhtimis- ja klassiandmete päringute vastuvõtmine ja nendele vastamine.
- Andmete edastamine, kui neid küsib HID-klassi juht (kasutades *Get\_Report* päringut).
- Andmete vastuvõtmine hostilt.

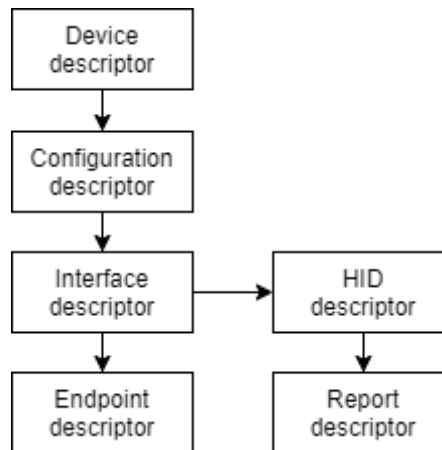
Katkestustorud on kasutusel järgmiseks:

- Asünkroonsete (mittenõutud) andmete vastuvõtmine seadmeSt.
- Madala latentsusega andmete edastamine seadmele.

Katkestustoru väljund on valikuline. Kui seade deklareerib katkestustoru väljundi, siis väljundraportid edastatakse hostile seadmele katkestustoru väljundi kaudu. Kui katkestustoru väljundit ei deklareerita, siis väljundraportid edastatakse seadmele juhtimisotsa kaudu, kasutades *Set\_Report(Output)* päringuid [12].

Üldine deskriptorite struktuur USB-HID seadmes ja nende suhe näeb välja järgmiselt (vt. Joonis 2):





Joonis 2. Deskriptorite struktuur USB-HID seadme jaoks

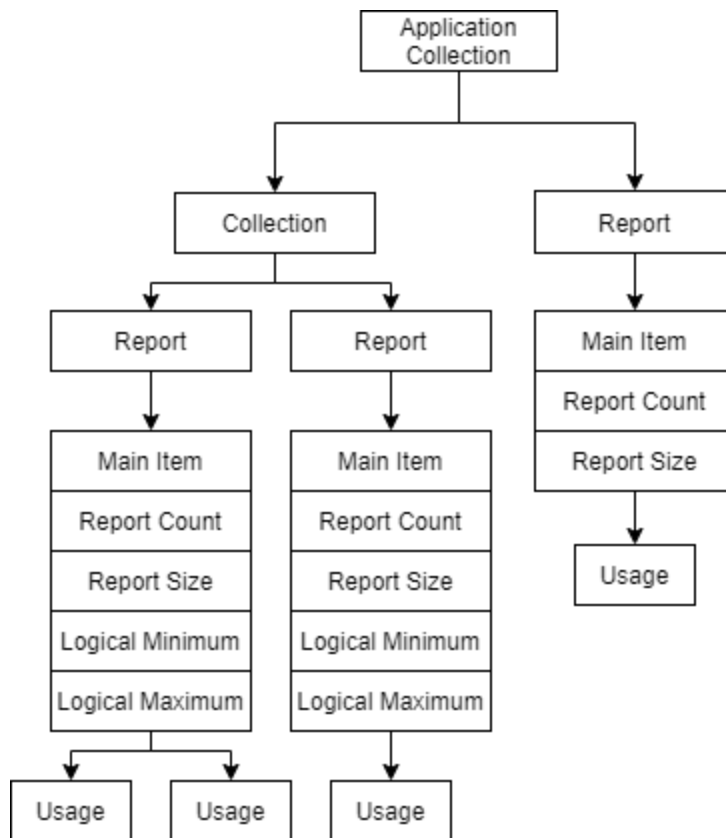
Seadme deskriptor (*Device descriptor*) kirjeldab üldist teavet USB-seadme kohta. See sisaldab teavet, mis kehtib üldiselt seadme ja kõigi seadme konfiguratsioonide kohta. USB-seadmel on ainult üks seadme deskriptor.

Konfiguratsiooni deskriptor (*Configuration descriptor*) kirjeldab konkreetse seadme konfiguratsiooni kohta teavet. Deskriptor sisaldab välja *bConfigurationValue* koos väärtusega, mis, kui seda kasutatakse parameetrina *SetConfiguration()* päringus, sunnib seadet võtma kirjeldatud konfiguratsiooni [12].

Liidese deskriptor (*Interface descriptor*) kirjeldab konkreetset liidest konfiguratsiooni sees. Konfiguratsioon pakub ühte või mitut liidest, millest igaühel on null või rohkem lõpp-punkti deskriptori, mis kirjeldavad unikaalset lõpp-punktide komplekti konfiguratsioonis.

Iga liidese jaoks kasutataval lõpp-punktis on oma deskriptor (*Endpoint descriptor*). See deskriptor sisaldab teavet, mida on vaja võrguressursi nõuete määramiseks iga lõpp-punkti jaoks. Lõpp-punktis deskriptor tagastatakse alati konfiguratsiooni teabe osana, mis tagastatakse *GetDescriptor(Configuration)* päringuga. Lõpp-punktis deskriptorist ei saa otse kasutada *GetDescriptor()* ega *SetDescriptor()* päringuga. Lõpp-punkti deskriptorist pole kunagi lõpp-punkti null jaoks [12].

HID-deskriptor (*HID descriptor*) tuvastab alamdeskriptorite pikkuse ja tüübi seadme jaoks. HID-klassi draiver sisaldab parserite süntaksianalüsaatorit, mida kasutatakse Report-deskriptoris leitud elementide analüüsimiseks. Parseri vaatepunktist näeb HID-klassi seade välja järgmise joonise kujul (vt. Joonis 3):



Joonis 3. Report deskriptor parseri vaatepunktist [4]

Parserit on vaja, et läbi analüüsida kogu raporti deskriptor ja leida kõik peamised üksused. See on vajalik, et analüüsida seadme poolt saadetud raportid.

### 3.1.1.1 Raportid

USB-terminoloogiat kasutades võib seade saata või vastu võtta tehingu iga USB-kaadriga (1 millisekund). Tehing võib koosneda mitmest paketest (märguanne, andmed, kätlus), kuid selle suurus on piiratud 8 baidiga madala kiirusega seadmete ja 64 baidiga kõrge kiirusega seadmete jaoks. Ülekanne on üks või mitu tehingut, luues seadmele tähendusliku andmehulga - näiteks sisendi-, väljundi- ja omaduste raportid.

Enamik seadmeid genereerib raporteid ehk ülekandeid, tagastades struktuuri, kus iga andmeväli on järjestikku esindatud. Siiski võivad mõned seadmed ühel ühenduspunktis omada mitut raportistruktuuri, kus igaüks esindab ainult mõnda andmevälja. Näiteks klaviatuur, millel on integreeritud näidik, võiks iseseisvalt edastada „klahvi vajutuse” andmeid ja „näidiku” andmeid samal ühenduspunktis. Raporti ID-üksused näitavad, milliseid andmevälju iga raportistruktuuris esindatakse. Raporti ID-üksuse silt määrab igale raporti ülekandele 1-baidise identifitseerimise eesliite. Kui Raporti deskriptoris pole

Raporti ID-üksuse silte, võib eeldada, et eksisteerib ainult üks sisendi-, väljundi- ja omaduste raporti struktuur ning koos nad esindavad kogu seadme andmeid [4].

Raporti deskriptor koosneb üksustest, mis annavad teavet seadme kohta. Raporti deskriptor kirjeldab iga andmeelementi, mida seade genereerib, ja millist andmeid tegelikult mõõdetakse. Üksuse esimene osa sisaldab kolme välja: üksuse tüüp, üksuse silt ja üksuse suurus. Koos määravad need väljad, millist teavet üksus pakub [13].

Näiteks määratleb Raporti deskriptor üksused, mis kirjeldavad asukohta või nupu olekut. Üksuse teave on kasutusel järgmistel eesmärkidel:

- Määrata, kuhu suunata sisend - näiteks sisend saata juhtkangile
- Luba tarkvaral määrata sisendile funktsionaalsus

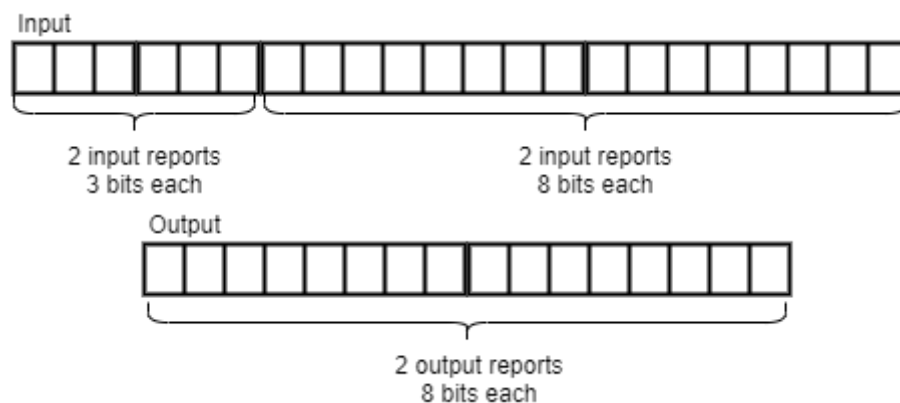
Raporti deskriptor annab iga juhtelemendi kohta seadmes andmete kirjelduse. Raporti deskriptorid koosnevad informatsioonitükkidest. Iga informatsioonitükk on nimetatud Üksuseks. Iga peamise üksuse silt (Sisend, Väljund või Omadus) määrab kindlaks konkreetse juhtelemendi tagastatud andmete suuruse ja määratleb, kas andmed on absoluutsed või suhtelised ning muud olulised andmed. Eelnevad Kohalikud ja Üldised üksused määratlevad minimaalse ja maksimaalse andmeväärtused jne. Raporti deskriptor on seadme kõigi üksuste täielik kogum. Pelgalt Raporti deskriptori vaatamisest saab rakendus teada, kuidas käsitleda sissetulevaid andmeid, samuti selle, milleks andmeid võiks kasutada [4].

Ühe või mitme juhtelemendi andmeväljad määratletakse peamise üksuse poolt ja täpsustatakse veelgi eelnevate Üld- ja Kohalike üksustega. Kohalikud üksused kirjeldavad ainult järgmise peamise üksuse määratletud andmevälju. Üldised üksused muutuvad vaikimisi atribuutideks kõikidele järgnevatele andmeväljadele selles deskriptoris. Joonisel 4 on toodud raporti struktuuri näide (üksikasjad lühiduse huvides välja jäetud):

```
Report Size (3)
Report Count (2)
Input
Report Size (8)
Input
Output
```

Joonis 4. Raporti struktuuri näide raporti suurusega

Üksuse parsimine tõlgendab ülaltoodud Raporti deskriptori üksuseid ja loob järgmised raportid (väikseim bitt on vasakul) (vt. Joonis 5):



Joonis 5. Raporti suuruse parsimise näide

Raporti deskriptor võib sisaldada mitmeid põhiüksuseid. Juhtimise andmete kirjeldamiseks peab Raporti deskriptor sisaldama iga järgmist üksust (kõik teised üksused on valikulised):

- *Input (Output or Feature)*
- *Usage*
- *Usage Page*
- *Logical Minimum*
- *Logical Maximum*
- *Report Size*
- *Report Count*

Sisend (*Input*)-, Väljund (*Output*)- ja Omadusüksused (*Feature*) luuakse andmeväljade loomiseks raportis.

- Sisendüksus kirjeldab teavet, mis puudutab ühe või mitme füüsilise kontrolli poolt esitatud andmeid. Rakendus võib kasutada seda teavet seadme poolt esitatud andmete tõlgendamiseks.
- Väljundüksust kasutatakse väljundandmevälja määratlemiseks raportis. See üksus on sarnane Sisendüksusele, välja arvatud see, et see kirjeldab andmeid, mis saadetakse seadmesse.
- Omadusüksused kirjeldavad seadme konfiguratsiooniteavet, mis võib saata seadmesse.

Kasutamised (*Usages*) on osa Raporti deskriptorist, mis annab rakenduse arendajale teavet selle kohta, mida tegelikult mõõdetakse. Lisaks näitab Kasutamise silt tootja soovitatud kasutust konkreetse juhtimise või juhtimiserühma jaoks. Kuigi Raporti deskriptorid kirjeldavad andmete vormingut - näiteks kolme 8-bitist välja - määratleb Kasutamise silt, mida andmetega teha tuleks - näiteks sisend x, y ja z. See omadus võimaldab tootjal tagada, et kasutaja näeb juhtimistele ühtlaseid funktsioonide määramisi kõikides rakendustes. Raporti deskriptoril võib olla mitu Kasutamise silti. Kasutamise ja juhtimiste vahel on üks-ühele vastavus, üks kasutusjuht on määratletud deskriptoris. Massiiv näitab, et iga Raporti deskriptori väli esindab mitut füüsilist juhtimise. Igal juhtimisel võib olla omadused, näiteks sellele määratud kasutamine. Näiteks võib nelja nupu massiivil olla iga nupu jaoks unikaalne Kasutamise silt. Kasutamine tõlgendatakse kui 32-bitine määratlemata väärtus, kus kõrgemad 16 bitid määratlevad Kasutamislehe ja madalamad 16 bitid määratlevad Kasutamise ID. Kasutamise ID-d kasutatakse üksikute Kasutamiste valimiseks Kasutamislehel [4].

Loogiline miinimum/maksimum on loogilistes ühikutes ulatusväärtused. See on väikseim/suurim väärtus, mida muutuja või massiivi element teatab.

Raporti suurus määratleb raporti väljade suuruse bittides. See võimaldab parsijal luua üksuse kaarti, mida raporti haldur kasutab.

Raporti kogus määratleb üksuse jaoks andmeväljade arvu. Teisisõnu määrab see, mitu välja on raportis selles konkreetse üksuse jaoks (ja seega, mitu bitti lisatakse raportile).

### **3.1.2 Füüsiline sisendseade (PID)**

Jõutagasiside seadmed kasutavad PID tüüpi kasutusi oma väljundite kirjeldamiseks ning teiste kasutuslehtede kasutusi sisendi kirjeldamiseks. Füüsiline sisendseade saab kasutada kahte meetodit kasutajale aistingute tekitamiseks: efekte või kohandatud efekte. Seade võib toetada ühte või mõlemat meetodit. Efektid tekitavad jõutagasisidet, mis põhineb erinevate parameetriplokkide parameetritel, mis on hoitud seadmes. Host laadib parameetriplokid ja seejärel saadab Start Effect käsu katkematu väljavoolu toru kaudu [1] [14].

Efektid koosnevad parameetriplokkidest. On määratletud 6 tüüpi parameetrite raportid: *Set Effect*, *Set Envelope*, *Set Condition*, *Set Periodic*, *Set Constant Force* ja *Set Ramp*

*Force*. Kõik efektid nõuavad efektiparameetri plokki. Sõltuvalt efektist võib vajalik olla üks või mitu täiendavat parameetriplokki. Parameetriplokke saab juurde pääseda ohjuri abil. Parameetriploki suurus konkreetse seadme jaoks sõltub seotud raporti suurusel. Iga tüüpi parameetriploki vaikimisi eraldust pakub seade.

Selles töös rakendatakse ainult püsiva jõu ja vedru efekte, mis tähendab, et tuleb kirjeldada ainult *Set Effect Report*, *Set Constant Force Report* püsiva jõu efekti jaoks ning *Set Condition Report* vedru efekti jaoks.

### 3.1.2.1 *Set Effect Report* Raport

*Set Effect Report* rakendab loogilist kogumit, mis tuvastab seotud raporti, mis on seotud PID efektiparameetrite seadistamisega. Efekti ploki indeks varieerub 1 kuni maksimaalse arvu efektideni, mis on salvestatud seadmesse.

Minimaalne *Set Effect Report* peab sisaldama *Effect Block Index*, *Effect Type*, *Duration*, *Sample Period*, *Gain*, *Trigger Button*, *Trigger Repeat Interval*, *Axis Direction* ja *Type Specific Block Offset* väärtusi.

- *Effect Type* - parameeter, mis määratleb seadme toetatud efektid.
- *Duration* - efekti kogukestus. Efekti jätkamiseks kuni selle peatamiseni Stop meetodiga seadistage *Duration INFINITE* (Null).
- *Sample Period* - periood, mille järel seade peaks efekti esitama. Väärtus null näitab, et tuleks kasutada vaikimisi esitusmäära.
- Kui seade ei suuda efekti esitada määratud määra järgi, valib see toetatud määra, mis on lähim palutud väärtusele.
- Kohandatud *Sample Periodi* seadistamine on võimalik eriefektide jaoks. Näiteks kui mängitakse signaalilainet kunstlikult suure sample perioodiga, tulemuseks on karedam tekstuur.
- *Gain* - efektile rakendatav tugevus.
- *Trigger Button* - nupu identifikaator või nihutus, mida kasutatakse selle efekti esitamiseks. *Null Trigger Button* väärtus näitab, et see efekt ei ole seotud nupuga.
- *Trigger Repeat Interval* - automaatse kordumise intervall efektide esitamiseks, mida käivitatakse nupu all hoidmisega. Intervall on aeg mängitud efekti lõppemise ja järgmise efekti alguse vahel. Kui see efekt on ühekordne efekt (automaatset kordamist ei soovita), tuleks see väärtus seada *INFINITE* (Null).

- *Axes Enable* - element, mis näitab telge, millel efekt peaks esinema.
- *Direction Enable* - lipp, mis näitab, et ainult üks olekuparameetri plokk on määratud ja suund kehtib olekuparameetri plokile kui polaarsuund. Kui see lipp on seatud, ignoreeritakse *Axes Enable* parameetrit ja kasutatakse *Direction* parameetrit.
- *Direction* - sarnaselt *Axes Enable* parameetritele. Saab kasutada telje või pöörlemissuuna määramiseks.
- *Type Specific Block Offsets* kasutatakse konkreetsete blokkide asukohtade näitamiseks, mis on seotud konkreetse jõutagasiside efekti parameetritega. See võimaldab hostile näidata, kus seadme mälus on konkreetse efekti jaoks vajalikud parameetrid. Kuid kuna seadmes kasutatakse jõutagasiside rakendamiseks ainult ühte telge, siis seda parameetrit töös ei kasutata.

Üksikute efektide identifitseerimiseks kasutatakse *Effect Type* välja sisu. Deklareeritud efekti tüübid kasutused loetlevad efektid, mida seade toetab. Võimalikud tüübid:

- *Constant Force*
- *Ramp*
- *Custom Force Data*
- *Square*
- *Sine*
- *Triangle*
- *Sawtooth Up*
- *Sawtooth Down*
- *Spring*
- *Damper*
- *Inertia*
- *Friction*

### **3.1.2.2 Set Constant Report Raport**

See efekt on üsna lihtne, kuna see määrab mootori pöörlemiskiiruse ühes suunas. Selle struktuur sisaldab järgmist: *Effect Block Index* ja element magnitute, mis määrab telje pöörlemisjõu.

### 3.1.2.3 *Set Condition Report Raport*

Erinevalt püsiva efektiga struktuurist on Set Condition Report palju keerulisem, kuna see simuleerib vedru efekti teljele ja nõuab keerukamaid arvutusi. Selle struktuur sisaldab järgmisi elemente: *Effect Block Index*, *Parameter Block Offset*, *Parameter Block Offset*, *Center Point Offset*, *Positive Coefficient*, *Negative Coefficient*, *Positive Saturation*, *Negative Saturation* ja *Dead Band*.

- *Parameter Block Offset* - seadmele, mis kasutab seadmehaldusmudelit, määrab see parameeter sihttelje indeksi.
- *Center Point Offset* - telje 0-asendist nihutamine.
- *Positive Coefficient* - positiivse külje konstantne kordaja neutraalse positsiooni suhtes.
- *Negative Coefficient* - negatiivse külje konstantne kordaja neutraalse positsiooni suhtes.
- *Positive Saturation* - maksimaalne positiivne jõu väljund.
- *Negative Saturation* - maksimaalne negatiivne jõu väljund.
- *Dead Band* - CP Offseti ümbruses olev ala, kus tingimus pole aktiivne. Teisisõnu, tingimus pole aktiivne Offset – *Dead Band* ja Offset + *Dead Band* vahel.

### 3.1.2.4 *Effect Operations Report Raport*

Seda raportit kasutatakse efektide toimingute juhtimiseks.

Efekti toimingute raport rakendab loogilist kogumit, mis tuvastab raporti, mis on seotud efekti tööparameetrite seadistamisega. Selle kogumi peab sisaldama *Effect Block Index*, *Effect Operation* ja *Loop Count* kasutusi.

- *Effect Operation* - toiming efekti teostamiseks, mis on tuvastatud *Effect ID* abil. See on nimetatud massiiv, mis sisaldab operatsiooni valijaid.
- *Effect Start* - Efekti käivitamine, mis on tuvastatud *Effect Handle* abil.
- *Effect Start Solo* - Efekti käivitamine, mis on tuvastatud *Effect Handle* abil, ja kõigi teiste efektide peatamine.
- *Effect Stop* - Efekti peatamine, mis on tuvastatud *Effect Handle* abil.
- *Loop Count* – Seadme korduste arv operatsiooni kordamiseks. *INFINITE* väärtus tähendab „kordusta lõpmatult”.



### 3.1.2.5 *Device Gain Report* Raport

See raport on kasutusel üldise seadme tugevuse juhtimiseks.

Seadme tugevuse raport on raport, mis on seotud seadme tugevuse parameetrite määramisega. See raport võib sisaldada Seadme tugevuse kasutust, mis on kõigi seadmel olevate efektide jaoks rakendatav seadme tugevus.

### 3.1.2.6 Parameetri ploki haldamine

Parameetri plokid on salvestatud PID-klassi seadmesse. Arendaja saab valida, kas lasta süsteemidraiveril hallata parameetri ploki hulka või hallata seda ise.

Kui draiver haldab parameetri ploki hulka, siis seade lihtsalt pakub mälu ploki, kuhu draiver laadib parameetri plokid. Kõik otsused selle kohta, kuhu laadida parameetri plokid, tehakse draiveri poolt. Draiver hoiab loendit kõigist parameetri plokkidest, mis on praegu plokis salvestatud, ja teostab vajadusel prügikoristust.

Kui seade haldab oma ploki, siis draiver lihtsalt edastab parameetri plokid seadmele ja saab vastuseks kas negatiivse kinnituse, kui plokile polnud ruumi, või positiivse kinnituse ja ploki käepideme, kui ruumi oli. Käepidet kasutatakse kõigis tulevastes viidetes parameetri plokile.

Kui seade haldab parameetri plokkide, siis plokkide aadressid plokis ei ole draiverile nähtavad. Selle asemel peab host saama efekti jaoks käepideme, saates seadmele päringu uue efekti loomise raportiga soovitud efekti tüübiga. Saades selle raporti, eraldab seade mälu efekti jaoks ning määrab efekti jaoks *Effect Block* Indexi.

Pärast uue efekti loomise raporti saatmist peab host saatma HID *Get\_Report* päringu. Seade vastab *PID Block Load* raportiga, mis sisaldab äsja määratud *Effect Block* Indexit. Kõiki järgnevaid viiteid efekti plokile hosti poolt kasutatakse selle indeksiga.

Pärast seda, kui host on saanud ja kontrollinud *PID Block Load* raporti sisu, saadab ta *Set Effect* raporti, mis sisaldab *Effect Block* Indexit, mis tagastati *PID Block Load* raportis koos teiste efekti määratlemiseks vajalike parameetritega. *PID Block Load* raport näitab välja mälu kättesaadavust mälu kasutamisel, jääva mälu hulka seadmes, et host saaks ette

näha, kas seadmel on piisavalt jääkraami lisaefektide või kohandatud efektide andmete jaoks.

#### **3.1.2.6.1 Create New Effect Report Raport**

Nagu nimest võib aru saada, vastutab see raport uue efekti loomise eest seadmel. Seega on selle raporti struktuur järgmine: *Effect Type* ja mälu kogus, mida selle efekti jaoks vajatakse.

#### **3.1.2.6.2 PID Pool Report Raport**

Selle raporti abil eristab host, kuidas mälu seadmel hallatakse. See raport sisaldab järgmisi andmeid:

- *RAM Pool Size* - PID RAM parameetrite ploki suurus baitides.
- *Simultaneous Effects Max* - Maksimaalne samaaegsete efektide arv, mida seade toetab. Seade deklareerib selle kasutuse, et näidata kõigi korraga käivitatavate efektide koguarvu.
- *Device Managed Pool* - Selle kasutus määratakse väärtusega 1 seadmetele, mis haldavad omaenda mälupooli, ja väärtusega 0 neile, mis toetavad draiveriga hallatavat mälu.
- *Shared Parameter Blocks* - Selle kasutus on 1 seadmetele, mis toetavad ühe parameetri ploki jagamist mitme efekti vahel. Väärtus on null seadmetele, millel on iga efekti jaoks üks parameetri plokk.

#### **3.1.2.6.3 PID Block Load Report Raport**

Nagu varem mainitud, kasutatakse PID Ploki Laadimise Raportit seadme vastusena Create New Effect Raportile ja see sisaldab teavet uue efekti loomise oleku kohta. Selle raporti parameetrid on järgmised:

- *Block Load Status* - tuvastab *Create New Effect* Raporti päringu lõpuleviimise oleku. See nimetatud massiiv sisaldab Ploki Laadimise valikuid.
- *Block Load Success* - *Create New Effect* Reporti käsk viidi edukalt lõpule.
- *Block Load Full* - *Create New Effect* Reporti käsk ebaõnnestus, kuna seadme mälupoolis ei olnud piisavalt ruumi.

- *Block Load Error - Create New Effect* Reporti käsk ebaõnnestus, kuna seadmes ilmnes viga.
- *RAM Pool Available* - baitide arv, mis on alles seadme mälu poolses.

#### **3.1.2.6.4 *PID Block Free Report* Raport**

Selle raporti saatmisel sisaldab see käsku konkreetse efekti kustutamiseks. Seega sisaldab raport ainult *Effect Block* Indexit, mis viitab efektile, mida tuleb kustutada. Kui väärtus on võrdne FF-iga, siis kustutatakse seadmest kõik efektid.

#### **3.1.2.6.5 *PID State Report* Raport**

Lisaks tagasiside seadme korrektsele toimimisele peab seade tingimata saatma teavet oma oleku kohta, mida teostatakse *PID State* raporti abil. Selles raportis märgib seade oma hetkeseisu, sisaldades järgmisi väärtusi:

- *Effect Playing* - see kasutus näitab efekti praegust esitusseisu seadmepool. Kui see on seatud, siis efekt mängitakse hetkel seadmepool. Kui see on tühjentatud, siis efekt hetkel ei mängita (see on peatatud).
- *Device Paused* - näitab, et kõik seadme efektid on peatatud. Kõik seadme efektid on peatatud praegusel ajahetkel.
- *Actuators Enabled* - näitab, et seadme aktuaatorid on lubatud.
- *Safety Switch* - kontroll näitab turvavõtme olekut seadmepool. Kui turvavõtme lipp on seatud ja seade on kasutatav.
- *Actuator Override Switch* - see on kasutaja käsutuses olev aktuaatori ülekattelüliti. Kui see lipp on seatud, lubab kasutaja seadme aktuaatorid.
- *Actuator Power* - tagasiside korral näitab see kasutus praegust aktuaatori toite olekut.

#### **3.1.2.6.6 *PID Device Control Report* Raport**

*PID Seadme Juhtimise Raport* on seotud seadme juhtimiskäskude täitmisega. Need on käsud, mis mõjutavad kogu seadet ja ei vaja parameetreid. Sellel raportil on järgmised võimalikud väärtused:

- *Enable Actuators* - lubab kõik seadme täituriid.
- *Disable Actuators* - keelab kõik seadme täituriid.

- *Stop All Effects* - peatab kõik töötavad efektid.
- *Reset* - kustutab igasuguse seadme pauseseisu, lubab kõik täiturid ja kustutab kõik mälu efektid.
- *Pause* - kõik seadme efektid on hetkel peatatud.
- *Continue* - kõik efektid, mis töötasid, kui seade oli peatatud, käivitatakse uuesti nende viimasest ajahetkest.

### 3.2 Arendatud testseadme tarkvara struktuur

Arendatud seadme aluseks on mikrokontroller STM32F401RCT6. Süsteem sisaldab järgmisi põhilisi tarkvarakomponente: raporti deskriptor, raporti halduri (*Report Handler*) ja efektide halduri (*Effect Handler*) (vt. Joonis 6).

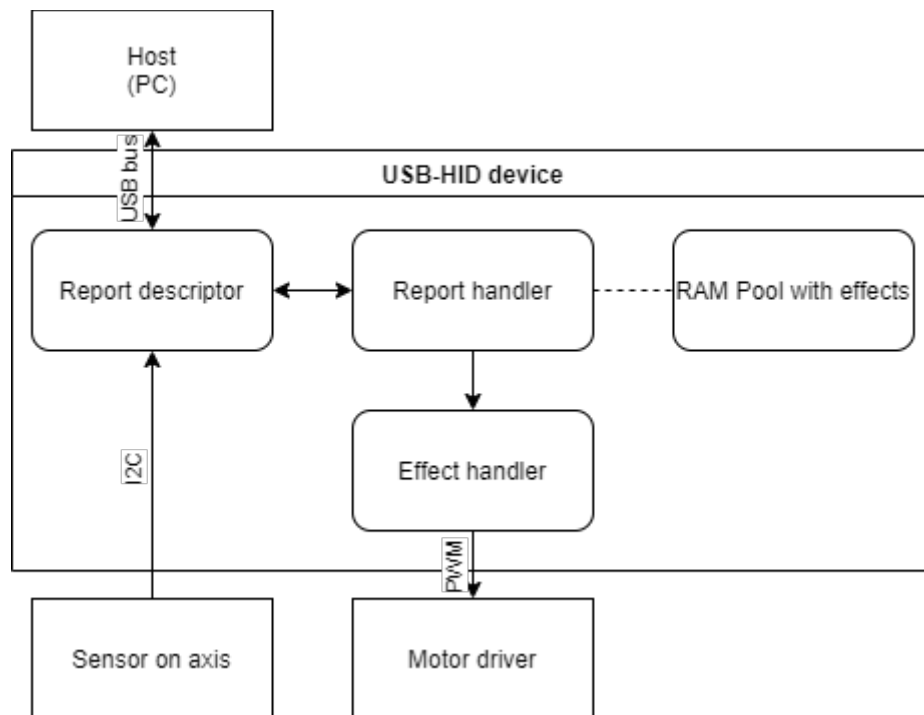
Lähtuvalt siin seadme tööpõhimõtete peatükis kirjeldatust, tuleb määrata andmestruktuur ja tüüp, millega seade ja host vahetavad andmeid. See struktuur kirjeldatakse raporti deskriptoris.

Lisaks on seadmel vaja salvestada laaditud efekte RAM-puhvrisesse, kus efekte hoitakse raporti deskriptoris kirjeldatud vormingus. Seade haldab efektide mälu ise, mis hõlbustab oluliselt tööd ja vähendab vajadust spetsiaalse draiveri kirjutamise järele sellele seadmele.

Pärast andmete saamist hostilt peab seade määrama andmete tüübi ja kuidas neid töödelda. Sellega tegeleb raporti haldur. Ta määrab raporti tüübi raporti ID põhjal, mis on saadetud andmetes ja seejärel töötleb vastavalt raporti tüübile.

Samuti pärast efekti ja selle täitmise käsu saamist peab seade efekti parameetrid tõlgendama PWM signaaliks ja seejärel mootorit selle signaali abil juhtima. Selle ülesande jaoks rakendatakse efekti haldur meetodit. See rakendatakse osana raporti haldur klassist. Seega on efekti halduril otsejuurdepääs efektidele ja nende parameetritele.

Rooli teljeandurilt saadud signaal ei nõua keerukat töötlemist, seega rooliratta asend laaditakse otse raporti deskriptoris kirjeldatud struktuuri ja saadetakse otse USB-ühenduse kaudu.



Joonis 6. Seadme struktuur plokkskeemina

## 4 Testseadme realiseerimine

### 4.1 Ettevalmistamine realiseerimiseks

Arenduskeskkonnaks valiti STM32CubeIDE, millega tööle asumiseks tuli esmalt seadistada projekt. Selleks oli vaja valida STM32F401RCT6 ning ühendada ka kõik vajalikud teegid ja funktsioonid. STM32CubeIDE-s tehakse seda tänu väga mugavale ja arusaadavale liidesele üsna lihtsalt ja kiiresti.

Samuti oli vaja teha sageduse seadistamine, kuna kõik funktsioonid ei töötanud standardsagedusel. USB-ühenduse siini sagedus määrati 48 MHz-le ja PWM signaali sagedus määrati 8 MHz-le. 8 MHz sageduse väärtus määrati empiirilisel. Mootori reaktsioon oli kõige täpsem sellise sageduse korral [15] [16].

Pärast seda oli võimalik luua projekt ja põhiosa koodist kõigi peamiste funktsioonide initsialiseerimisega, mille STM32CubeIDE iseseisvalt genereeris. Hoolimata automaatselt koodigeneratsioonist, oli vaja realiseerida ka raportite vastuvõtt (*get reports*) ja korrigeerida väljaminevate raportite töötlemist (*out reports*), kuna genereeritud kood on vaid raamistik ega ole lõplik lahendus. Järgmistes alapunktides kirjeldatakse, kuidas algne genereeritud funktsionaalsus muudeti ja loodi loogika väljaminevate ja saabuvate raportite töötlemiseks.

Selleks, et host saaks seadet tuvastada kui USB-seadet, mis toetab tagasisidet, oli vaja kirjutada raporti deskriptor, mis kirjeldab kõiki raporti struktuure, samuti seadme jõutagasisidega seotud funktsionaalsust. Alustuseks kirjeldati deskriptoris, et seade kuulub mänguseadmete klassi, omab kaheksat nuppu ja kuut telge. Selles etapis tuvastatakse seade kui mängu-USB HID-seade, kuid hetkel veel tagasiside toetuseta [17].

Tagasiside toe lisamiseks kirjeldati deskriptoris järgmised raportid: *Set Effect Report*, *Set Condition Report*, *Set Constant Report*, samuti raport, mis vastutab efektide juhtimise eest. Lisaks efektide korrektseks toimimiseks oli vajalik rakendada komplekt raporteid, mis kirjeldavad, kuidas efekte seadmes hoitakse.

Kuna iga deskriptori elemendi jaoks on hädavajalik määrata raporti ID, suurus ja konkreetsele elemendile saadetavate pakettide arv, samuti edastatava väärtuse loogilise

miinimumi ja maksimumi määramine, siis see osa jääb järgnevast raporti struktuurikirjeldusest välja.

Lisaks oli vaja rakendada loogika USB-ühenduse kaudu andmete vastuvõtmiseks ja saatmiseks, sissetulevate ja väljaminevate raportide töötlemiseks, nimelt meetod Rapordi Haldur, samuti loogika saadud efektiparameetrite PWM-signaaliks teisendamiseks mootori juhtimiseks rooli teljel, st meetod Efekti Haldur.

## 4.2 Raporti deskriptor

Järgnevalt antakse antud peatükis ülevaade raporti kirjutamise protsessist, mis on vajalik jõu tagasisidega USB-HID seadme rakendamiseks. Töö käigus on kirjeldatud kõik vajalikud USB-HID seadme PID klassi raportid (vt. p 3.1.2), nimelt: *Set Effect Report*, *Set Constant Force Report*, *Set Condition Report*, *PID Pool Report*, *Create New Effect Report*, *PID Block Load Report*, *PID Block Free Report*, *Effect Operation Report*, *PID Device Control Report*, *PID State Report*.

Set Effect Raportis on märgitud sellised efektide tüübid nagu Püsiv efekt ja Vedru efekt. Selle tulemusena oli vajalik ka kirjutada raportid, mis on spetsiaalselt seotud nende efektidega, nimelt: *Set Constant Force Report* ja *Set Condition Report*.

## 4.3 Parameetrite ploki haldamine

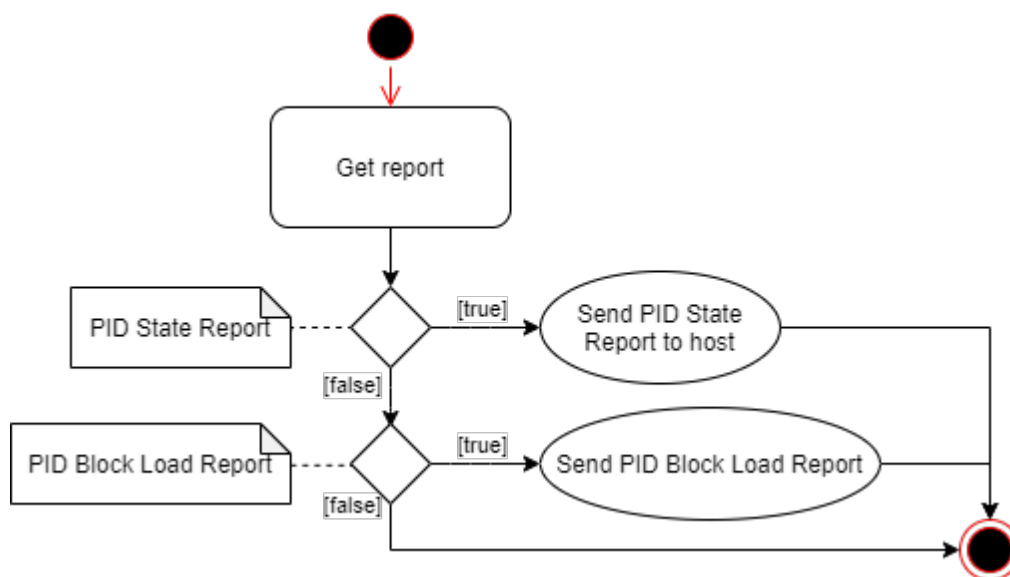
Nagu mainitud tagasisidega USB-HID seadme tööpõhimõtete peatükis, peab deskriptor sisaldama teavet selle kohta, kuidas seade haldab mälu. Kuna seade haldab mälu ressursse efektide jaoks autonoomselt, määrati *PID Pool* Raportis järgmised parameetrid: seadme poolt hallatava mälu lipp, jagatud parameetrite plokkide lipp, RAM-i puhvri suurus ja samaaegselt mängitavate efektide arv. Kuna seade haldab mälu iseseisvalt, kirjeldati järgmiste raportite struktuure: *Create New Effect Report*, *PID Block Load Report*, *PID Block Free Report*, *Effect Operation Report*, *PID Device Control Report*, *PID State Report*.

Sellega on raporti kirjutamine lõpetatud, kuna on täidetud miinimumnõuded tagasiside tagamiseks. Selle pärast on võimalik rakendada algoritmid seotud andmete töötlemisega. Edasi kirjeldatakse tööd andmetega, mida saadetakse nende raportite kaudu.

#### 4.4 Out- ja Get-raportite vastuvõtmine

Järgnevalt kirjeldatakse meetodid, mida seade kasutab *Out-* ja *Get*-raportite vastuvõtmiseks. Pärast genereerimist oli rakendatud ainult *Out*-raportite töötlemine, mis suutsid vastu võtta ainult täisarvulisi väärtusi. Kuid selle projekti jaoks oli vaja töödelda suurt hulka andmeid, seega seda meetodit laiendati nii, et see suudaks vastu võtta 8-bitise täisarvude massiivi ning samuti selles meetodis rakendati funktsionaalsust, mis takistab andmete lugemist juhul, kui eelmine andmete lugemine pole veel lõpetatud.

Samuti oli vaja rakendada meetod, mis töötleks *Get*-raporteid. *Get*-raportitega esitatakse päringud seadmest teatud andmete saamiseks, nimelt PID-puhvri ja ploki laadimise raportite saatmiseks. *Get*-raportite töötlemise loogika on järgmine: see meetod saab raportit, mis sisaldab vajaliku raportit ID-d seadmest, ja seejärel see meetod saadab nõutud andmed USB-ühenduse kaudu. Seejärel tuli see meetod integreerida meetodisse, mis tegeleb *Custom* HID raportite töötlemisega (vt. Joonis 7).



Joonis 7. Get-raport haldamise meetodi algoritm [17]

#### 4.5 USB-siini andmete vastuvõtt ja saatmine

Meetod *USBD\_CUSTOM\_HID\_Setup* koodis vastutab USB-siinil tehtud päringute töötlemise eest kasutaja HID-seadme jaoks. See töötleb päringuid tüüpi *CLASS* ja *STANDARD*. *CLASS* tüüpi korral määrab ta protokollid, saab ja seab ooteoleku ning töötleb raportite edastamist ja vastuvõtmist. *STANDARD* tüüpi korral teeb ta erinevaid toiminguid, sõltuvalt päringu tüübist, nagu oleku, deskriptori ja liideste saamine.



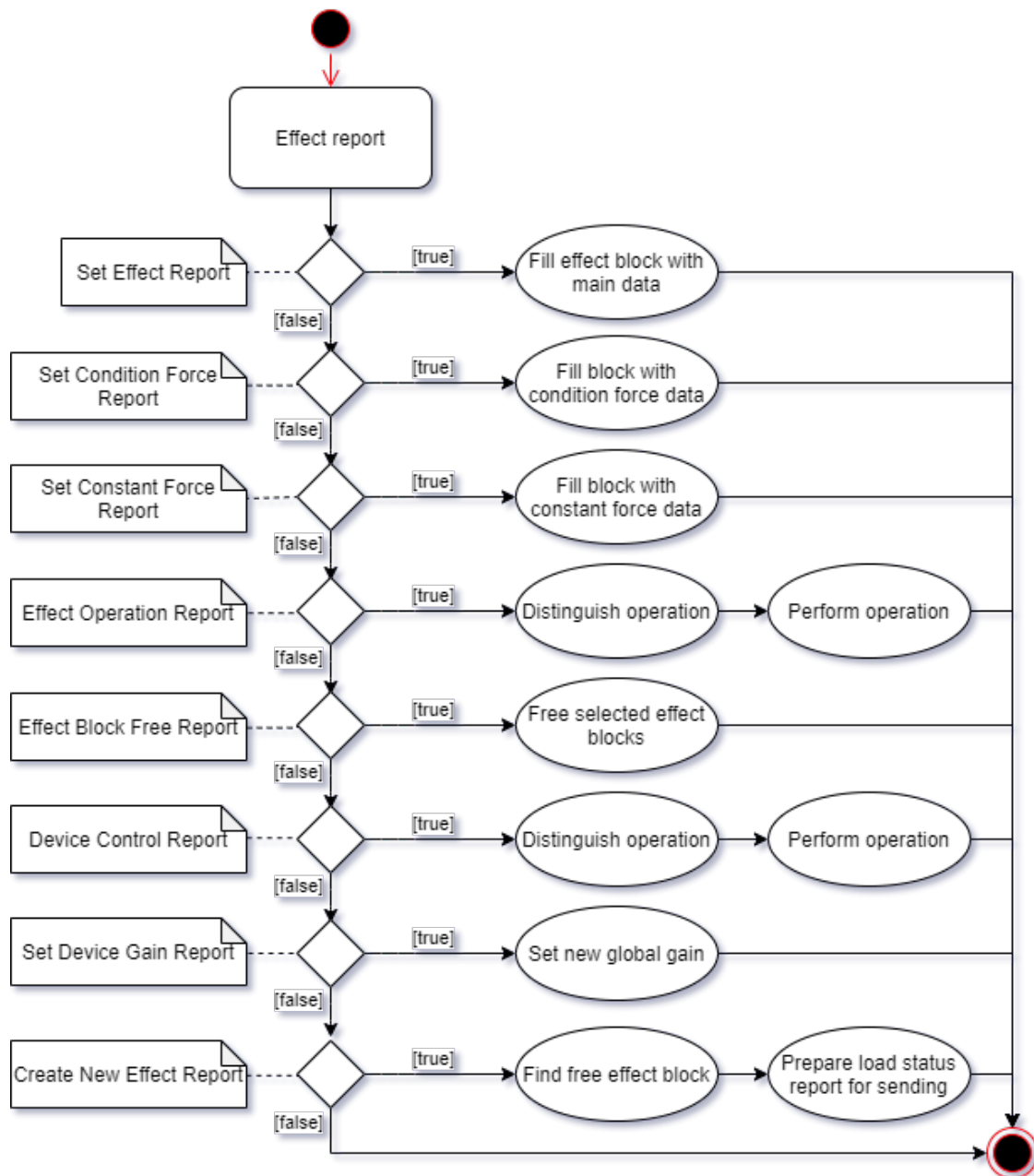
Tuleb mainida, et meetod toetab andmete edastamist ja vastuvõtmist raportite kaudu (*SET\_REPORT* ja *GET\_REPORT*). *GET\_REPORT* korral kutsub meetod tagasikutsumisfunktsiooni, et saada andmed USB kaudu saatmiseks. Lõputöö raames laiendati meetodit töötlemaks täisarvude massiive ja vältimaks andmete lugemist juba aktiivse päringu ajal. Samuti lisati meetod *GET\_REPORT* päringute töötlemiseks, mis saadavad vajalikud andmed pärast päringu saamist hostilt.

#### **4.6 Rooli telje asendi saatmine USB siini kaudu**

Kuna seade saadab ainult rooli telje asendit ja andurite andmed ei sisalda müra, muutub nende väärtuste töötlemine tarbetuks. Seega saab selle väärtuse otse USB-ühenduse kaudu edastada. Kui seade loeb väärtuse andurist, koostab see kohe vastava struktuuriga andmepaketi, sarnaselt raportis deskriptorile. Kuna seadme eesmärk on jõu tagasiside realiseerimine, on vajalik ainult tagasiside telje kasutamine, st roolitelg. See tähendab, et kõiki teisi telgede väärtusi, mis on määratletud raporti deskriptoris, ei kasutata ja need seatakse nulliks. Pärast andmete ettevalmistamist saadetakse need meetodi *USBD\_CUSTOM\_HID\_SendReport* abil, mis võtab vastu andmed saatmiseks ja saadab selle andmepaketi USB-siini kaudu.

#### **4.7 Raporti haldur**

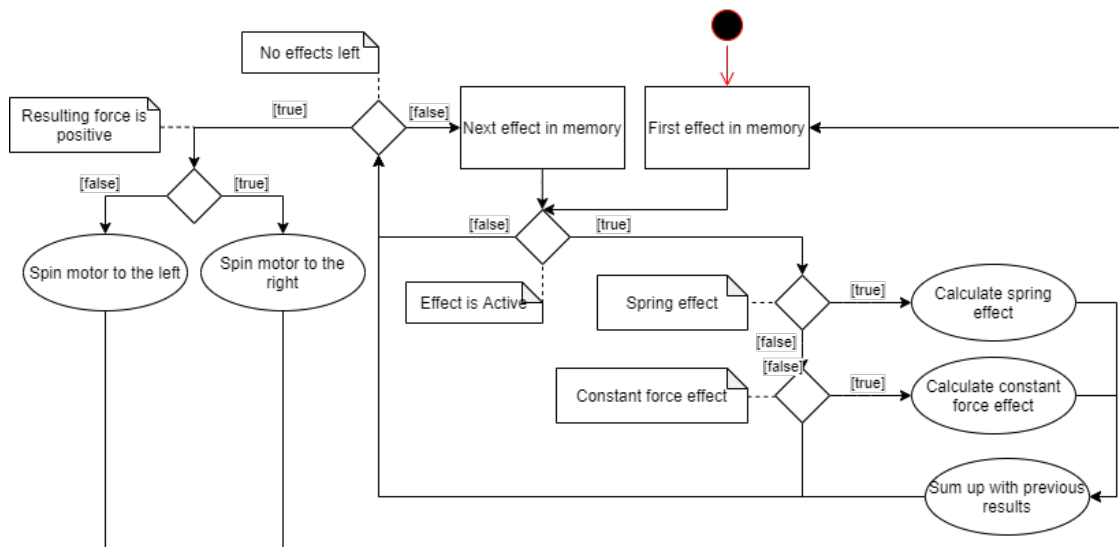
Pärast andmepaketi saamist hostilt, tuleb see töödelda ja teostada vastavad käsklused. *GET*- raportite töötlemine toimub otse raportite vastuvõtmise meetodis, kuna *GET*- raportid on seadmeandmete päringud; seega pole nende töötlemiseks keerulist loogikat vaja, ja seega saab seade kohe saata päritud andmed. Kuid *OUT*- raportite puhul on vaja palju keerulisemat töötlemist, kuna need sisaldavad mitte ainult raporti ID-d, vaid ka andmeid, mida tuleb töödelda erinevalt sõltuvalt raporti tüübist. Sellega tegeleb raporti haldur ehk haldamismeetod. Sellele edastatakse raporti ID ja sellega seotud andmed. See meetod määratleb, millist toimingut tuleb teha raporti ID abil, ja seejärel teostab selle (vt. Joonis 8).



Joonis 8. Raport haldamise meetodi algoritm [17]

## 4.8 Efektide haldur

Programmiblokk, mida nimetatakse „Efektide haldur” (*Effect Handler*), vastutab laaditud ja käivitatud efektide töötlemise eest. See blokk läbib iga laaditud ja käivitatud efekti. Iga arvutuse tulemus summeeritakse lõppväärtuseni. Seejärel, sõltuvalt selle tulemuse märgist, pöörab seade mootorit vastavas suunas, kasutades arvutuste tulemusena saadud PWM-väärtust. Pärast meetodi töö lõpetamist seatakse arvutuste tulemus nulliks ja meetod alustab tööd uuesti (vt. Joonis 9).



Joonis 9. Efekti haldamise meetodi algoritm [17]

Allpool on esitatud efektide arvutuste kirjeldused valemite kujul, lähtudes esitatud koodist:

Püsiva jõu efekti (*Constant Force*) valem (1):

$$F = \text{magnitude} \quad (1)$$

Kus  $F$  on jõud,  $\text{magnitude}$  on efekti suurus.

Vedru efekt (*Spring Effect*) valem (2):

$$F = \begin{cases} \text{negK} \times (\text{pos} - (\text{cpOffset} - \text{deadBand})) & (\text{tagasilöögitsoonis}) \\ \text{posK} \times (\text{pos} - (\text{cpOffset} + \text{deadBand})) & (\text{tagasilöögitsooni taga}) \end{cases} \quad (2)$$

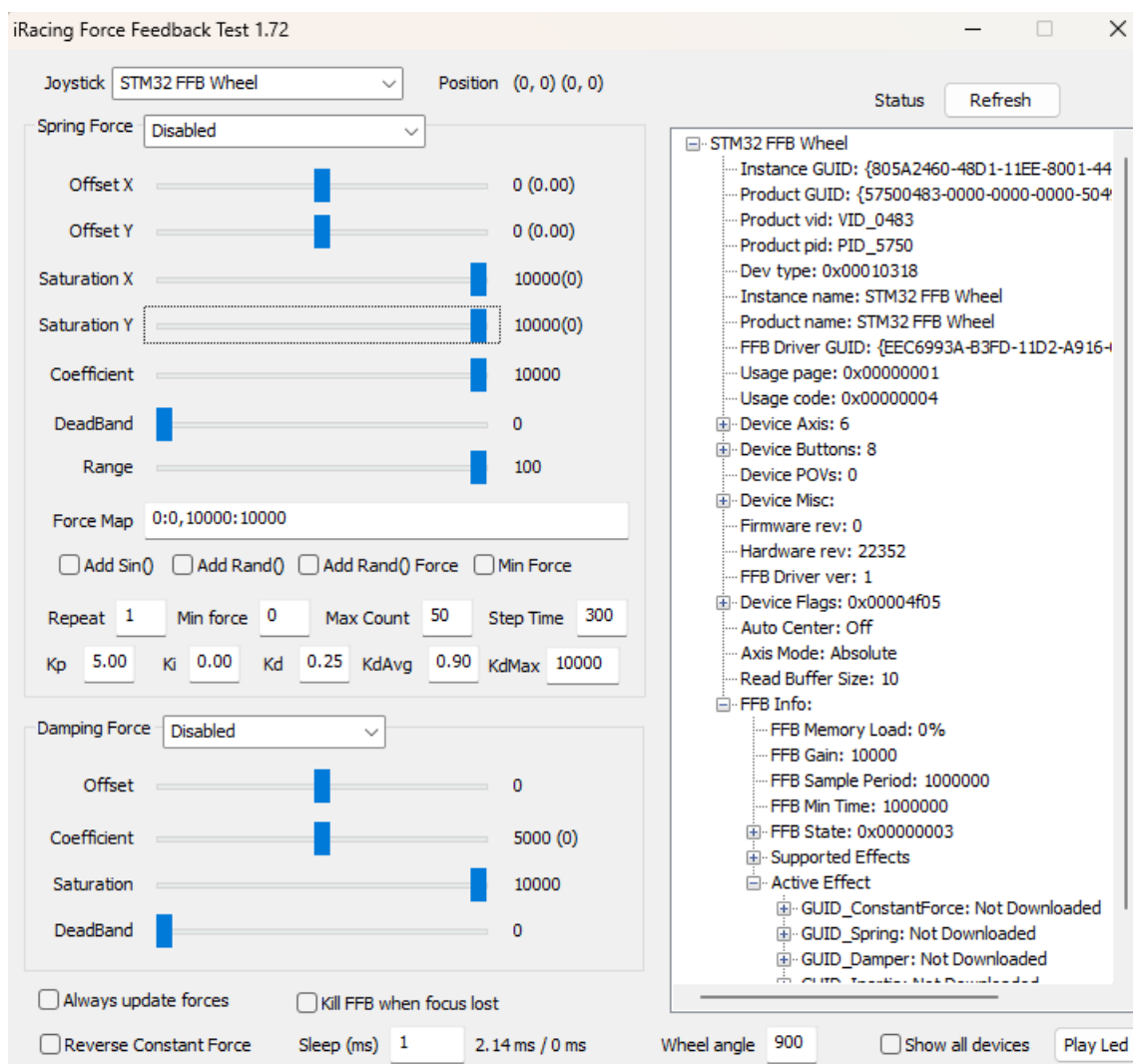
Kus  $F$  on jõud,  $\text{cpOffset}$  on keskme nihutamine,  $\text{deadBand}$  on tundetuse tsoon,  $\text{pos}$  on praegune asend,  $\text{negK}$  ja  $\text{posK}$  on jõudu reguleerivad kordajad.

Lisaks sellele, et kohandada saadud väärtusi mootoriga, on tulemusele ka lisatud võimenduskoefitsient, mis on otseselt seadme koodis määratud, et võimaldada seadmel töötada erinevate mootoritega.

## 5 Seadme testimine

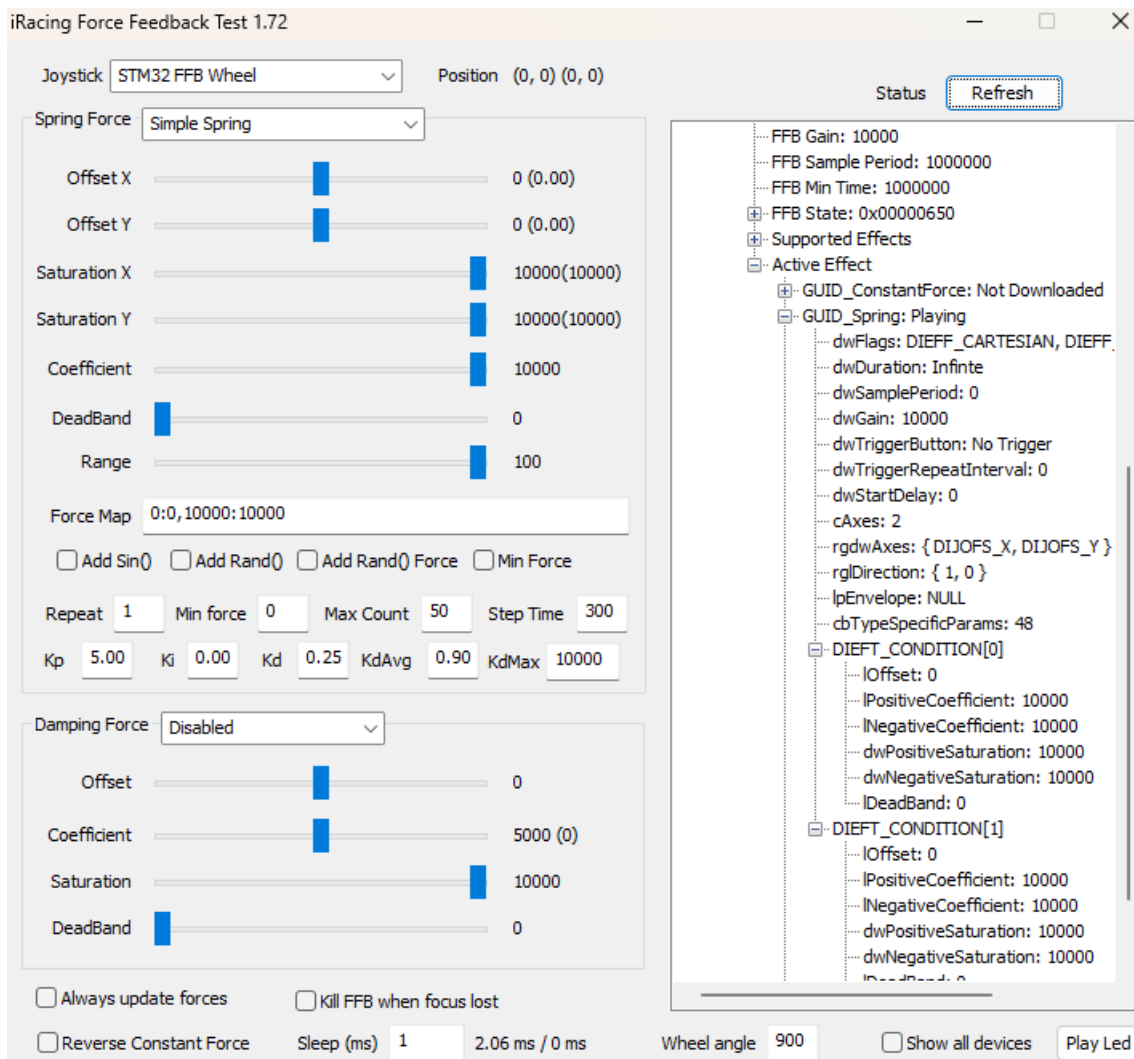
Testseadmes loodud võimaluste testimiseks kasutati programmi iRacing Force Feedback Test 1.72. Sellel programmil on võimalus lugeda seadme andmeid, toetatud efektide loendit ja lugeda ka laaditud efektide olekut.

Alloleval joonisel (vt. Joonis 10) loetakse testseadme infona välja, sellel on 6 telge, 8 nuppu ning samuti pole seadmesse laetud ühtegi efekti.



Joonis 10. Testimise programm mis kuvab andmeid seadmest

Katsetusena käivitati seadmest vedru efekt (*Spring Effect*), mille järel seade püüdis rooli nurka tagasi keskele viia, samas kui programm muutis vedruefekti olekuks “mängib” ning programm näitas ka kõik selle efekti spetsiifilised parameetrid (vt. Joonis 11).



Joonis 11. Testimise programm kuvab efekti parameetreid

Katsetati ka konstantse jõu efekti (*Constant Force Effect*) ja see töötab samuti õigesti. See efekt põhjustab telje pöörlemise etteantud kiirusega kindlas suunas.

## Kokkuvõte

Töö tulemusena on loodud PID tüüpi USB-HID testseade (vt. Lisa 2), mida tuvastatakse hosti poolt mänguseadmena, mis toetab jõu tagasisidet, nimelt selliseid efekte nagu *Constant Force* ja *Spring*. Lisaks suudab see seade töödelda ja esitada hostilt saadud efekte ning teavitada hosti efektide olekust USB-HID protokolliga abil.

Kõik selle töö eesmärgid on edukalt täidetud, kuna töö käigus on tehtud ülevaade selliste seadmete rakendusala kohta, põhjalikult selgitatud USB-HID protokolliga PID klassi, loodud on raporti deskriptor, milles toetatakse vähemalt kahte efekti. Seade on realiseeritud kui kohandatud USB-HID seade, on loodud loogika raportite vastuvõtmiseks ja saatmiseks ning saadud raportite töötlemiseks. Realiseeritud jõu tagasisidega testimisseade algoritmi loogika toetab kahe efekti teisendamist PWM signaaliks selle väärtuse edastamiseks mootori juhtimiseks.

Selle töö tulemust on võimalik edasi arendada lisades seadmele rohkem efekte, asendades mootori ja mootoridraiveri võimsamate vastu, samuti lisades täiendavaid juhtsisendid nagu pedaalid, käigukang või mõni muu haptiline liideseade, mis aitavad kasutajal paremini tunnetada virtuaalset või kaugget keskkonda, sellesse rohkem suhestuda.

Selle töö käigus on rakendatud Tallinna Tehnikaülikoolis „Riistvara Arendus ja Programmeerimine” erialal omandatud programmeerimisoskusi ning teadmisi.

## Kasutatud kirjandus

- [1] Microsoft Corporation, Immersion Corporation, Intel Corporation, „Device Class Definition for Physical Interface Devices (PID),“ 08. 09. 1999. [Võrgumaterjal]. Available: [https://www.usb.org/sites/default/files/documents/pid1\\_01.pdf](https://www.usb.org/sites/default/files/documents/pid1_01.pdf). [Kasutatud 03. 05. 2024].
- [2] Y. C. Jongyoon Lim, „Force-Feedback Haptic Device for Representation of Tugs in Virtual Reality,“ 01. 04. 2022. [Võrgumaterjal]. Available: <https://www.mdpi.com/2079-9292/11/11/1730>. [Kasutatud 03. 05. 2024].
- [3] LinkedIn, „How can you effectively use force feedback in VR?,“ [Võrgumaterjal]. Available: <https://www.linkedin.com/advice/3/how-can-you-effectively-use-force-feedback-vr-zybf>. [Kasutatud 03. 05. 2024].
- [4] Alps, Cybernet, DEC, Intel, Key Tronic Corporation, LCS/Telegraphics, Logitech, Microsoft Corporation, NCR, Sun Microsystems, ThrustMaster, „Device Class Definition for Human Interface Devices (HID),“ 27 05. 2001. [Võrgumaterjal]. Available: [https://www.usb.org/sites/default/files/hid1\\_11.pdf](https://www.usb.org/sites/default/files/hid1_11.pdf). [Kasutatud 03. 05. 2024].
- [5] „775 DC Motor 12-36V,“ [Võrgumaterjal]. Available: [https://www.aliexpress.com/item/4000293295742.html?spm=a2g0o.order\\_list.order\\_list\\_main.671.1bcc1802iLRg5u](https://www.aliexpress.com/item/4000293295742.html?spm=a2g0o.order_list.order_list_main.671.1bcc1802iLRg5u). [Kasutatud 03. 05. 2024].
- [6] Mean Well, „350W Single Output Switching Power Supply LRS-350 series,“ 04. 02. 2024. [Võrgumaterjal]. Available: <https://www.meanwell.com/Upload/PDF/LRS-350/LRS-350-SPEC.PDF>. [Kasutatud 03. 05. 2024].
- [7] MARLIN P. JONES MARLIN P. JONES & ASSOC., INC., „37225-MD High Current "H" Bridge Motor Driver,“ [Võrgumaterjal]. Available: <https://www.mpja.com/High-Current-H-Bridge-Motor-Driver-43A/productinfo/37225%20MD/>. [Kasutatud 03. 05. 2024].
- [8] „BTS 7960,“ 12. 2004. [Võrgumaterjal]. Available: <https://image.dfrobot.com/image/data/DRI0018/BTS7960.pdf>. [Kasutatud 03. 05. 2024].
- [9] STM Electronics, „High-performance access line, Arm Cortex-M4 core with DSP and FPU, 256 Kbytes of Flash memory, 84 MHz CPU, ART Accelerator,“ [Võrgumaterjal]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32f401rc.html>. [Kasutatud 03. 05. 2024].
- [10] ams OSRAM Group, „AS5600 12-Bit Programmable Contactless,“ 20. 06. 2018. [Võrgumaterjal]. Available: <https://look.ams->

osram.com/m/7059eac7531a86fd/original/AS5600-DS000365.pdf. [Kasutatud 03. 05. 2024].

- [11] „WheelCheck\_1\_72,“ 26. 03. 2016. [Võrgumaterjal]. Available: [https://www.dropbox.com/s/3zxf5yd1bvuyji5/WheelCheck\\_1\\_72.zip?e=1&dl=0](https://www.dropbox.com/s/3zxf5yd1bvuyji5/WheelCheck_1_72.zip?e=1&dl=0). [Kasutatud 03. 05. 2024].
- [12] Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips, „Universal Serial Bus Specification,“ 27. 04. 2000. [Võrgumaterjal]. Available: [http://sdpha2.ucsd.edu/Lab\\_Equip\\_Manuals/usb\\_20.pdf](http://sdpha2.ucsd.edu/Lab_Equip_Manuals/usb_20.pdf). [Kasutatud 03. 05. 2024].
- [13] Microsoft, „Introduction to Human Interface Devices (HID),“ 12. 09. 2023. [Võrgumaterjal]. Available: <https://learn.microsoft.com/en-us/windows-hardware/drivers/hid/>. [Kasutatud 03. 05. 2024].
- [14] Multiple contributors, „HID Usage Tables FOR Universal Serial Bus (USB),“ 26. 01. 2024 . [Võrgumaterjal]. Available: [https://www.usb.org/sites/default/files/hut1\\_5.pdf](https://www.usb.org/sites/default/files/hut1_5.pdf). [Kasutatud 03. 05. 2024].
- [15] N. San, „Is there an ideal PWM frequency for DC brush motors?,“ 07. 2017. [Võrgumaterjal]. Available: <https://electronics.stackexchange.com/questions/242293/is-there-an-ideal-pwm-frequency-for-dc-brush-motors>. [Kasutatud 03. 05. 2024].
- [16] „Controlling Brushed DC Motors Using PWM,“ 2019. [Võrgumaterjal]. Available: [https://www.portescap.com/-/media/project/automation-specialty/portescap/portescap/pdf/whitepapers/wp\\_controlling\\_brushed\\_DC\\_motors\\_using\\_pwm.pdf](https://www.portescap.com/-/media/project/automation-specialty/portescap/portescap/pdf/whitepapers/wp_controlling_brushed_DC_motors_using_pwm.pdf). [Kasutatud 03. 05. 2024].
- [17] I. Obuhov, „FFBWheel,“ 17. 04. 2024. [Võrgumaterjal]. Available: <https://github.com/Vielder/FFBWheel>. [Kasutatud 03. 05. 2024].
- [18] F. Zhao, „Tutorial about USB HID Report Descriptors,“ 01. 01. 2013. [Võrgumaterjal]. Available: <https://elecceleator.com/tutorial-about-usb-hid-report-descriptors/>. [Kasutatud 03. 05. 2024].
- [19] Iris Dynamics, „What is Force Feedback: Explained,“ 03. 09. 2022. [Võrgumaterjal]. Available: <https://irisdynamics.com/articles/what-is-force-feedback>. [Kasutatud 03. 05. 2024].



## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Ilja Obuhov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “USB-HID jõutagasisidega testseade“, mille juhendaja on Andres Rähni
  - 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.

01.05.2024

---

<sup>1</sup> 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 tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 - Testseadme lõplik vaade

