

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond  
Informaatika instituut  
Infosüsteemide õppetool

Joonas Künstler

# **Valguse sujuva muutmise tagamine valgustussüsteemis**

**Bakalaureusetöö**

Juhendaja: Raul Liivrand  
lektor

Tallinn 2015

## **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.

.....  
(kuupäev)

.....  
(allkiri)

## Lõputöö ülesanne

- Lõputöö teema: LED RGBW prožektorite valguse muutmise rakendus nutiseadmes. Kuidas tagada, et kõikide lampide valguse muutumine oleks ühtlane ja sujuv?  
Application inside smartphone for changing LED RGBW lights light. How to make lights change so that changes appear to be simultaneous and smooth?
- Teema päritolu: Toote prototüübi arendamise käigus tekkinud probleem vajab lahendamist.
- Lõputöö eesmärgid: Probleemi on vaja analüüsida ning leida lahendus olukorra parandamiseks. Leida süsteemi osa, mis põhjustab kõige rohkem probleeme.
- Oodatavad tulemused: Lahendused, mille rakendamisel süsteem töötab stabiilsemalt ning valguse juhtimine toimub sujuvalt ning kasutaja tunnetab, et valgus muutub reaalselt.
- Lähtetingimused: Valgustussüsteemi prototüüp ja juhtimise rakendus on olemas. Urida tuleb olemasolevat süsteemi.
- Lahendatavad küsimused: Valguse sujuvaks muutmiseks tuleb lahendada küsimused:  
Mida tuleks muuta valguse juhtimise tarkvaras?  
Mida tuleks muuta kasutatavate protokollide osas?  
Mida tuleks muuta Wi-Fi infrastruktuuris?  
Mida tuleks muuta valgustussüsteemis?
- Täiendavad nõuded: Juhtimiseks kasutatava nutiseadme puhul peab kasutama traadita ühendust. Valgustussüsteemi võrgumoodulite puhul peab kasutama traadita ühendust.

Kuupäev:

Üliõpilane: Joonas Künstler

-----

(allkiri)

Juhendaja: Raul Liivrand

-----

(allkiri)

Kinnitaja:

-----

(allkiri)

## **Annotatsioon**

Valgustussüsteemi prototüübi ja nutiseadme rakenduse testimisel ilmnedid probleemid Wi-Fi ühenduse stabiilsuse ja rakendusest valgustussüsteemile käskude edastamisega, mille tulemusena ei olnud valguse muutmise sujuv.

Kuna tegemist on süsteemiga, mis koosneb nii riistvarast kui tarkvarast, mis on ainult osaliselt muudetavad, tuli analüüsida kogu süsteemi, et leida kõige problemaatilisemad kohad, mille muutmise annab kõige suurema efekti. Kui tegemist on seadmete või tarkvaraga, mida ei ole võimalik muuta, siis on vaja leida parim viis, kuidas seadistamise või kasutamisega muuta valguse muutmise võimalikult sujuvaks.

Valguse sujuvat muutmist saab muuta suurusjärgu võrra paremaks, kui valgustussüsteemi Wi-Fi moodulis võtta kasutusele voo reguleerimine ning valgustussüsteemi toiteplokkis kasutada valguse muutuse silumist.

Valgustussüsteemis valguse muutmise käskude edastamiseks valitud protokollidele alternatiivi ei ole.

Selleks, et Wi-Fi infrastruktuur toetaks võimalikult sujuvat valguse muutmist, tuleb kasutada testitud *premium*-klassi seadmeid ja jätkata võimalikult hea Wi-Fi manussüsteemi mooduli otsinguid.

Nutiseadme rakenduses aitaks valguse muutmist sujuvamaks teha lõpliku olekumasina kasutamine toiteplokkiga andmetete vahetamisel, et käskude saatmine toimuks sujuvalt ja kogu andmevahetuse haldamise koormus jääks nutiseadme rakendusse.

Analüüsi tulemuste rakendamise korral saab valgust muuta märgatavalt sujuvamalt, mis tagab kasutajate rahulolu.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 57 leheküljel, 8 peatükki, 17 joonist, 2 tabelit.

## **Abstract**

Lighting system prototype and smartphone application communication testing over Wi-Fi network revealed problems with connection stability and command transfer to lighting system. As a result light change is not smooth.

To find which part of the system has biggest impact to light smoothness whole system needs to be analyzed. System consist of hardware and software. Some parts of the system can be changed others cannot be changed. Analyzing should reveal parts with biggest impact. Changing these parts will be most effective for guaranteeing smooth light changing. If part of the system cannot be changed there might be a way to improve light changing smoothness by hardware selection and configuration.

Light changing smoothness can be drastically improved by implementing flow control in lighting system Wi-Fi module and by using light change smoothing in lighting system power supply.

Current communications protocols are best options for smooth light changing compared to alternative options.

Wi-Fi infrastructure cannot be changed. Only solution is to provide users with recommended premium routers and focus on finding best embedded Wi-Fi modules for lighting system Wi-Fi module.

Implementing final state machine in smartphone application for communication management would make light changing smoother by guaranteeing connection management and leaving connection management responsibility on smartphone application.

Light change smoothing and user experience can be significantly improved by fixing problems found during lighting system analyzing.

The thesis is in Estonian language and contains 57 pages of text, 8 chapters, 17 figures, 2 tables.



## Lühendite ja mõistete sõnastik

<b>VS</b>	<b><i>Lighting system</i></b> valgustussüsteem
<b>VSTP</b>	<b><i>Lighting system power supply</i></b> valgustussüsteemi toiteplokk
<b>HMI</b>	<b>Hydrargyrum medium-arc iodide</b>  filmi- ja meelelahutustööstuse jaoks toodetud metallhaliidlamp
<b>DMX512</b>	<b><i>(Digital Multiplex) is a standard for digital communication networks that are commonly used to control stage lighting and effects.</i></b> andmevahetuse standard, mida kasutatakse tavaliselt valguse ja valgusefektide juhtimiseks
<b>IRQ</b>	<b><i>Interrupt Request</i></b>  katkestusnõue - protsessorile antakse teada, et katkestunõue vajab töötlemist
<b>RIT</b>	<b><i>Repetitive Interrupt Timer</i></b>  taimer, mis tekitab katkestusi, et töödelda andmeid iga taimeri sammu ajal
<b>DAC</b>	<b><i>Digital-to-Analog Converter</i></b>  digitaal-analoogmuundur – seade, mis muundab numbrilisel (digitaalsel) kujul esitatud andmed analoogsignaaliks
<b>MTU</b>	<b><i>Maximum Transmission Unit</i></b>  maksimum-ülekandeühik – paketi maksimaalne suurus baitides, mida sideprotokolli mingi kiht võimaldab edasi saata
<b>UART</b>	<b><i>Universal Asynchronous Receiver/Transmitter</i></b>  universaalne asünkroontransiiver – teisendab baidid järjestikuseks bitivooks
<b>DUP ACK</b>	<b><i>Duplicate acknowledgement</i></b>

topelt jaatus – kasutatakse andmeedastuseks TCP protokollil abil

**TCP** *Transmission Control Protocol*

edastusohje protokoll

**UDP** *User Datagram Protocol*

kasutajadatagrammi protokoll

**Ethernet** *Ethernet*

kohtvõrgu standard IEEE 802.3

**Wi-Fi** *wireless local area network*

Etherneti protokollil kasutatav kohtvõrk, kus kaablid on asendatud raadiolinkidega

**DAC** *Digital-to-Analog Converter*

digitaal-analoogmuundur

**SSID** *Service Set Identifier*

mestiident – traadita kohtvõrgu (Wi-Fi võrgu) nimi

**FSM** *Finite State Machine*

lõplik olekumasin

## Jooniste nimekiri

Joonis 1: Süsteemi plokk skeem.....	21
Joonis 2: Kõige lihtsam süsteem.....	23
Joonis 3: Viide näputõstmise ja LEDi pinge muutuse vahel.....	24
Joonis 4: Viide käsupaketi alguse ja LEDi pinge muutuse vahel.....	25
Joonis 5: Viite jaotumine süsteemis.....	25
Joonis 6: Mõõdetud ja tunnetatud valguse intensiivsus.....	35
Joonis 7: Valgustussüsteemi Wi-Fi mooduli plokk skeem.....	41
Joonis 8: Valguse muutuse silumise protsess.....	45
Joonis 9: Üleminek DAC väärtuselt 0 väärtusele 10000.....	46
Joonis 10: Viide näputõstmise ja LEDi pinge muutuse vahel.....	58
Joonis 11: Viide käsupaketi alguse ja pinge LEDi pinge muutuse vahel.....	59
Joonis 12: TCP/IP mooduli tekitatud paus.....	60
Joonis 13: Valgustussüsteemi Wi-Fi mooduli plokk skeem.....	60
Joonis 14: Viited andmepakettide vahel enne ja pärast pausi.....	62
Joonis 15: TCP/IP moodul saadab andmeid ilma vahedeta.....	62
Joonis 16: Andmekadu pakettide edastamisel.....	63
Joonis 17: Normaalne andmete edastamine.....	63

## **Tabelite nimekiri**

Tabel 1: Jadasiini parameetrid.....	33
Tabel 2: Käskude arv sekundis lähtuvalt käsu andmepakettide suurusest.....	33

## Sisukord

1. Sissejuhatus.....	15
1.1 Taust ja probleem.....	15
1.2 Ülesande püstitus.....	18
1.3 Metoodika.....	19
1.4 Ülevaade tööst.....	20
2. Süsteemi ülevaade.....	21
3. Süsteemis tekkiva viite analüüs.....	23
3.1 Rakendusest käsu saatmise ja nähtava valguse muutuse vahelise aja mõõtmine.....	24
3.2 Süsteemis tekkiva viite analüüsi kokkuvõte.....	26
4. Etherneti võrgu transpordikihi protokollide analüüs.....	27
4.1 TCP.....	27
4.2 UDP.....	30
4.3 Mõne muu protokollide kasutamine.....	31
4.4 Etherneti võrgu transpordikihi protokollide analüüsi kokkuvõte.....	31
5. Etherneti võrgu rakenduskihiki protokollide analüüs.....	32
5.1 Valgustussüsteemi jaoks loodud protokoll.....	32
5.1.1 Pakettide suuruse mõju lähtuvalt andmete edastamisest Wi-Fi võrgus.....	33
5.1.2 Pakettide suuruse mõju lähtuvalt valgustussüsteemi programmi tööst.....	33
5.1.3 Pakettide saatmistiheduse mõju lähtuvalt andmete edastamisest Wi-Fi võrgus.....	33
5.1.4 Pakettide saatmistiheduse mõju lähtuvalt valgustussüsteemi programmi tööst.....	34
5.1.5 Pakettide saatmistiheduse mõju lähtuvalt kasutaja tunnetusest, kas valgus muutub sujuvalt.....	34
5.2 Art-Net.....	36
5.3 Etherneti võrgu rakenduskihiki protokollide analüüsi kokkuvõte.....	37
6. Traadita võrgu seadmete analüüs.....	38
6.1 Üldised probleemid Wi-Fi seadmete kasutamisel.....	38
6.2 Infrastruktuuri piirangute kaardistamine.....	39
6.3 Valgustussüsteemi Wi-Fi moodul.....	41
6.4 Traadita võrgu seadmete analüüsi kokkuvõte.....	42
7. Valgustusüsteemis käskudele reageerimise analüüs.....	44

7.1 Jadasiinile käsu saatmise ja LEDi pinge muutuse vahelise aja mõõtmine.....	44
7.2 Valgustussüsteemis diskreetsete üleminekute silumise analüüs.....	45
7.3 Valgustussüsteemis käskudele reageerimise analüüsi kokkuvõte.....	47
8. Nutiseadme rakendusest käskude saatmise analüüs.....	48
8.1 Kasutajaliidese blokeerimine juhul, kui valguse muutmise käske ei saa saata.....	49
8.2 Rakendusele oma olekute süsteemi loomine.....	49
8.3 Lõpliku olekumasina genereerimine.....	50
8.4 Nutiseadme rakendusest käskude saatmise analüüs.....	50
Kokkuvõte.....	52
Summary.....	54
Kasutatud kirjandus.....	57
Lisad.....	58
Lisa 1. Viide rakendusest antud käsu ja pinge muutuse vahel.....	58
Lisa 2. Viide jadasiinil info vastuvõtu alguse ja LEDi jalal pinge muutmise lõpu vahel....	59
Lisa 3. Pakettide edastamine Wi-Fi moodulis.....	60
Lisa 4. iOS rakenduses TCP keepalive seadistamine.....	64

# 1. Sissejuhatus

## 1.1 Taust ja probleem

Teatris, filmitööstuses, fotograafias jm. kasutatakse valgusteid ja nende süsteeme, et edastada vaatajale kunstniku nägemust. Tänapäevani on valgustitena laialdaselt kasutusel hõõglambi baasil tehtud edasiarendused või HMI lambid, mis mõõtmel suured, eraldavad palju soojust ja tarbivad palju elektrit.

Kuna sellise süsteemi ülespanek nõuab palju füüsilist jõudu, siis kasutatakse valgustajaid, kes panevad süsteemi füüsiliselt üles. See teeb valgustite kasutamise kalliks ning aeganõudvaks, kuna raskete valgustite kinnitamiseks on vaja tugevaid ja raskeid poome ning muid aluseid, mille ülesseadmine võtab kaua aega.

Kuna valgustid tarbivad palju elektrit, tuleb kasutada suuri generaatoreid. Need võtavad palju kütust ning elektri kaablite vedamine valgustiteni on raskendatud, kuna kaablid on suure koormuse tõttu väga jämedad.

Sellises süsteemis käib valguse värvi muutmine filterkilede või geelide asetamisega lambi ette. Ka siin saavad tööd valgustajad, kes vahetavad või kinnitavad filtreid valgustite külge. Iga selline filtri kinnitamine või eemaldamine vms. takistab tööd võtteplatsil.

Viivituste tõttu tuleb personalile rohkem maksta ning värvimuutmise edukus sõltub sellest, kui hästi saab valgustaja aru operaatori soovist. Seega võib tekkida olukord, kus pärast filtri lisamist tuleb see kohe uuesti asendada jne., et proovida senikaua, kuni saavutatakse sobiv valgus.

Valgustite juhtimiseks kasutatakse üldiselt DMX512 võrku ja DMX pulte. Süsteemis on üks juhtplokk (DMX pult) ja kõigi seadmete ühendamiseks kasutatakse pärgühendust. Igale valgustile saab määrata ühe kanali viiesaja kaheteistkümnest. Kui puldist soovitakse valgustit

kontrollida, siis tuleb puldi üks juhtnupp siduda selle kanaliga ja siis saab selle nupu abil valgust muuta.

Värviliste valgustite puhul on iga lambi värvi jaoks vaja oma kanalit, lisaks võib olla vaja veel kanaleid valgustisse sisse ehitatud efektide juhtimiseks vms. Ülesseadmise lihtsustamiseks teevad DMX pultide tootjad valgustite profile, mis lihtsustavad puldi seadistamist.

Ülaltoodust on näha, et DMXi kasutamine võimaldab täielikku kontrolli süsteemi üle, kuid süsteemi ülesseadmine on keerukas. Kasutaja peab saama aru kanalitest, oskama neid määrata, valgusteid seadistada ja DMX pulti seadistada. Kasutaja peab koostama valgustite kaardi, et aru saada, kus konkreetsed valgustid asuvad. Kõik see tähendab, et valgustite ülesseadmine tuleb tellida spetsialistidelt ning neid on vaja ka süsteemi seadistamisel, kuna iga DMXi pult on erinev ning selle kasutamine nõuab eelteadmisi. See tähendab kulusid spetsialistide palkamiseks ja kasutajal puudub loominguvabadus, kuna ta sõltub DMX puldi seadistaja oskustest ja sellest, kui hästi puldi seadistaja saab aru, kuidas kasutaja soovib valgust muuta.

Valgusdiiodide areng on toonud kaasa ka muutusi valgustite ehituses. Valgusdiiodid võimaldavad teha väiksema kaalu ja võimsusega valgusteid, mis annavad sama palju valgust, kui traditsioonilised valgustid. Kasutajale tähendab see seda, et valgustite ülespanek on lihtsam ja kiirem ning valgustite jaoks ei ole vaja nii palju elektrit ehk kasutada saab väiksemaid generaatoreid. Nii väheneb tööjõukulu ja elektri tagamine valgustitele on palju lihtsam, kuna kaablid ei ole enam nii massiivsed ja kulud kütusele vähenevad märgatavalt.

Kui valgustites kasutatakse värvilisi valgusdiode, saab valgusti valguse värvi muuta ilma valgustile filtreid lisamata. See tähendab, et valgust saab muuta kaugjuhtimise teel ja nii ei teki enam viivitusi filtrite paigaldamise tõttu ja paraneb loominguuline vabadus, kuna valguse valik ei ole piiratud olemasolevate filtritega ning valguse muutmine on lihtne ja kiire. Kaovad ka filtri kasutamisest tingitud kaod valgusvoo tugevuses. Selliste valgustite kasutamine võimaldab hoida kokku kogu personali tööaega ning suurendab loominguulist vabadust.



Valgusdioodel baseeruvate valgustite juhtimisel saab kasutada DMX512 võrku. Kuna sellise võrgu seadistamine on keeruline, oleks vaja lihtsamat süsteemi, mille kasutamisega saab hakkama tavakasutaja ehk süsteemi ülespanekuks ei ole vaja palgata spetsialisti või kulutada aega DMX puldi õppimisele. Lihtsam süsteem on eriti vajalik väikeste projektide puhul, kui ei ole raha spetsialistide palkamiseks.

Kõige lihtsam tavakasutaja jaoks on kasutada graafilisi juhtnuppe, mis eristuvad värvi või väljanägemise poolest.

Nutiseadmete tehnoloogiline areng ning nende massiline levik muudavad need heaks vahendiks, mida saab kasutada kasutajaliidese platvormina. Kui lähtuda kasutajaliidese kujundamisel väljakujunenud tavadest, on võimalik luua kasutajaliides, mis on arusaadav ilma lisakoolituseta ning kasutaja saab kohe valgustussüsteemi juhtima asuda.

Nutiseadmete puuduseks on see, et nende liideste arv on väike võrreldes lauarvutitega ning mobiilsele nutiseadmele kaabliga lisamooduli ühendamine läheb vastuollu nutiseadme olemuse ja kasutamisharjumustega. Nii tuleks kasutada nutiseadmesse sisseehitatud traadita ühenduse loomise viise. Üheks üldlevinud traadita ühenduseks on Wi-Fi, mis on olemas enamikus nutiseadmetes ning Wi-Fi seadmed ning nende võrkuühendamine on tavakasutajatele tuttav.

Valgusdioodidel põhineva valgustussüsteemi tootearenduse käigus tekkis vajadus luua nutiseadme rakendus valgustussüsteemi juhtimiseks. Nutiseadmega valgustussüsteemi juhtimine muudab juhtimise tavakasutaja jaoks lihtsamaks ning on valgustussüsteemi müügiargumendiks. Wi-Fi võrgu ja seadmete kasutamine võimaldab loobuda spetsiaalsete juhtimisseadmete ja kaabelduse ostmisest ning kasutada nutiseadet ilma lisamooduliteta.

Valgustussüsteemi prototüübi ja nutiseadme rakenduse testimisel ilmnisid probleemid Wi-Fi ühenduse stabiilsuse ja rakendusest valgustussüsteemile käskude edastamisega.

Nende eesmärkide saavutamiseks tekkis vajadus süsteemi analüüsida, et leida viis, kuidas tagada valguse sujuv muutumine.

## 1.2 Ülesande püstitus

Laboritingimustes töötab süsteem korrektselt ning kasutaja tunnetab, et nutiseadmest valguse juhtimisel muutub valgus sujuvalt ning reaalselt. Kui süsteemi kasutada väljaspool laborit, tekib probleeme ühenduse loomisega, ühenduse stabiilsusega ja valguse sujuva muutmise. Äärmusliku juhtumina ühendus ei katke, kuid süsteem ei reageeri valguse muutmise käskudele.

Kasutajale tundub valguse muutmine sujuv siis, kui nutiseadmest valgust muutes muutub valgus sama sujuvalt, kui on harjutud hõõglampide valgusregulaatorite puhul. Valgus peab muutuma kohe, kui nutiseadmest virtuaalset juhtnuppu liigutatakse.

Kuna Wi-Fi sagedused on laialdaselt kasutusel, on eeter tihti ülekoormatud. Seda nii linnakeskkonnas kui ka võtteplatsidel, kus sama sagedust kasutavad helisüsteemid, videokaamerad või muud seadmed. Süsteem koosneb mitmest osast. Igal osal on mõju süsteemi töökindlusele ja valguse sujuvale muutmisele.

Kirjeldatud probleemid ei võimalda kasutada nutiseadme rakendust valgustussüsteemi juhtimiseks, kuna nutiseadmest valguse muutmine ei toimi sujuvalt ning tõrgeteta.

Kuna süsteem koosneb mitmest osast, on terve süsteemi tervikuna analüüsimine ebaotstarbekas. Süsteemi tükeldamine osadeks võimaldab analüüsida iga osa mõju valguse sujuvale muutmisele ning aitab leida sujuvale muutmisele kõige suuremat mõju avaldavad osad, mille arendamine aitab olukorda märgatavalt parandada.

Mõni süsteemi osa, näiteks Wi-Fi ruuter, ei ole valgustussüsteemi tootja kontrolli all.

Kui Wi-Fi ruuteril on süsteemis kõige suurem mõju valguse sujuvale muutmisele, siis valgustussüsteemi tootja saab ainult soovitada, milliseid seadmeid tuleks kasutada, kuid ei saa ise toodet parandada.

Kui süsteemi osa puhul on tootjal võimalik toodet muuta, siis tuleb kaaluda, kas probleemi lahendamine on majanduslikult otstarbekas või tuleks valida mingi teistsugune lahendus.

Kui kõikide süsteemi osade analüüsist selgub, et sellises süsteemis põhimõtteliselt ei ole võimalik või otstarbekas realiseerida valguse sujuvat muutmist nutiseadme rakenduse abil, siis saab asuda otsima muid lahendusi, lähtudes saadud tulemustest.

Põhieesmärk on analüüsida süsteemi, et saada ülevaade, mis põhjustab tavakasutuses viite nutiseadme rakendusest valguse juhtimise käsu andmise ja valguse muutumise vahel ning miks ei muutu valgus sujuvalt.

Kui on leitud põhjus(ed) ja sellega seotud süsteemi osad, siis järgmine eesmärk on leida iga osa probleemile lahendus, et tagada süsteemis sujuv valguse muutumine.

Lahenduste puhul tuleb arvestada, et tehtavad muudatused peaksid lähtuma järgnevatest piirangutes:

- Kasutada tuleb nutiseadme sisseehitatud traadita ühendust (Wi-Fi). Nutiseadmega ühendatav lisaosamoodul oleks lisakulu ja muudaks kasutamise ebamugavaks.
- Kasutatavad traadita ühenduse moodulid peavad jääma majanduslikult otstarbekasse hinnaklassi ning suuruse poolest olema sellised, et ei kannataks toote visuaalne disain ja vastupidavus.
- Valgustussüsteemi ennast ei saa põhimõtteliselt muuta. Jadaosast ei saa loobuda.

### **1.3 Metoodika**

Selleks, et leida, milline süsteemi osa probleeme põhjustab, tuleb süsteem jaotada väiksemateks tükideks ning mõõta käsu edastamise viide iga osa kohta eraldi.

Analüüsida käskude edastamise viite mõju olenevalt sellest, kas Wi-Fi võrgus kasutatakse käskude edastamiseks TCP või UDP protokollid.

Mõõta erinevate võrguseadmete mõju valguse muutmise käskude edastamisele. Tulemuste alusel hinnata võrguseadmete rolli viite tekkimisel.

Analüüsida protokollide käsipaketi suurust ning selle mõju andmete ülekandele ning vastuvõtmisel valgustussüsteemis.

Mõõta ja analüüsida, kui kiiresti valgustussüsteem suudab vastuvõetud käsu alusel valgust muuta ja kui tihti valgustussüsteem suudab käske vastu võtta.

## **1.4 Ülevaade tööst**

Peatükid

Süsteemis tekkiva viite analüüs

Etherneti võrgu transpordikihi protokollide analüüs

Etherneti võrgu rakenduskihi protokollide analüüs

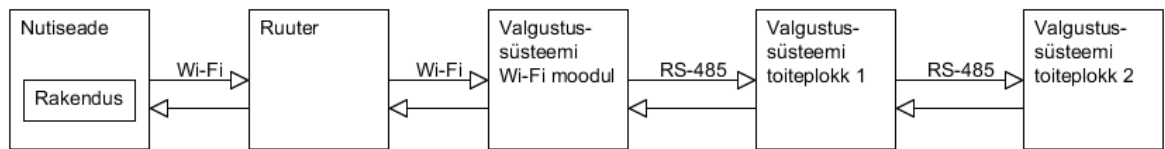
Traadita võrgu seadmete analüüs

Valgustussüsteemis käskudele reageerimise analüüs

Nutiseadme rakendusest käskude saatmise analüüs

## 2. Süsteemi ülevaade

Süsteemis on mitu osa.



Joonis 1: Süsteemi plokk skeem

1. Rakendus nutiseadmes, mis saadab pakette värviinfoga, kasutades nutiseadme Wi-Fi ühendust.
2. Nutiseade on ühendatud otse või läbi ruuteri valgustussüsteemi Wi-Fi mooduliga.
3. Valgustussüsteemi Wi-Fi moodul on ühendatud valgusti toiteplokkidega jadasini kaudu.

Kui kasutaja muudab nutiseadme rakenduse abil valguse värvi, siis saadetakse kohe pakett valgustussüsteemi poole teele.

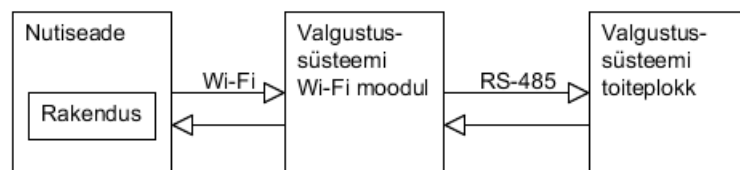
Kuna Wi-Fi ühendus sõltub ümbritsevast keskkonnast ja valgustussüsteemi Wi-Fi moodul on n-ö. manussüsteemi (*embedded*) moodul ning signaal tuleb teisendada jadasiinile, siis tekib probleeme reaajas sujuva värvimuutuse edastamisega. Lisaks edastamise probleemile on vaja ka siluda värvi muutumist, kuna diskreetne hüpe on silmale harjumatu ehk värvipaketi kättesaamisel jadasiini kaudu toimub veel sujuva muutmise rakendamine valgusti toiteplokkis.

Reaajas valguse muutmise tunnetus tekib siis, kui jutimisrakendusest antud käsk tekitab valguse muutuse kiiremini, kui kasutaja suudab uue käsu anda. Samuti on tähtis see, et kui kasutaja lõpetab käskude saatmise, peaks lõppema valguse muutus kohe ja viimases käsus toodud väärtustel. Tagantjärele toimivad muudatused või vilksatused tekitavad kasutajas ebakindlust ning segadust.

Mida kiiremini valguse muutmise käsk realiseerub valguse muutumisena, seda parem. Kuna iga süsteemi osa tekitab viivituse, on vaja mõõta iga süsteemi osa tekitatud viivitus ning hinnata selle osa terves süsteemis tekkivas viites. Selle alusel saab leida kõige suurema viivituse põhjustavad süsteemi osad.

### 3. Süsteemis tekkiva viite analüüs

Esmalt teeme mõõtmised laboritingimustes, mille puhul tundub kasutajale, et valguse muutumine on sujuv ja toimub reaajas. Süsteemist on välja jäetud ruuter ehk nutiseade ühendub otse valgustussüsteemi Wi-Fi mooduli külge ja kasutusel on ainult üks valgustussüsteemi toiteplokk. See on kõige lihtsam süsteem, kus on kasutusel nii vähe süsteemi osi kui võimalik.

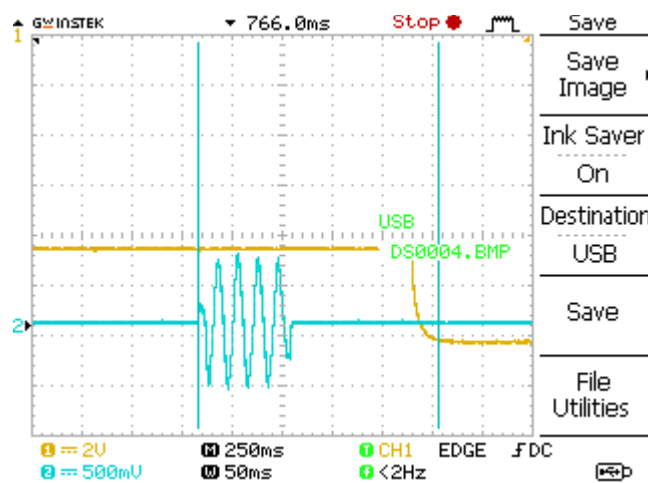


*Joonis 2: Kõige lihtsam süsteem*

Nutiseade on ühendatud valgustussüsteemi (edaspidi VS) Wi-Fi mooduli poolt loodud Wi-Fi võrku. Nutiseadme rakendus loob käivitumisel TCP ühenduse valgustussüsteemi Wi-Fi mooduliga ja saab info valgustussüsteemi ühendatud toiteplokkide (edaspidi VSTP) ja lampide kohta.

### 3.1 Rakendusest käsu saatmise ja nähtava valguse muutuse vahelise aja mõõtmine

Nutiseadme puutetundlikul ekraanil on nupp, millele vajutades (näppu ekraanilt tõstes) koostab rakendus valguse muutmise paketi ja saadab selle VSi jadasiinile. VSTP loeb jadasiinilt käsu ja muudab selle alusel LEDide voolu.

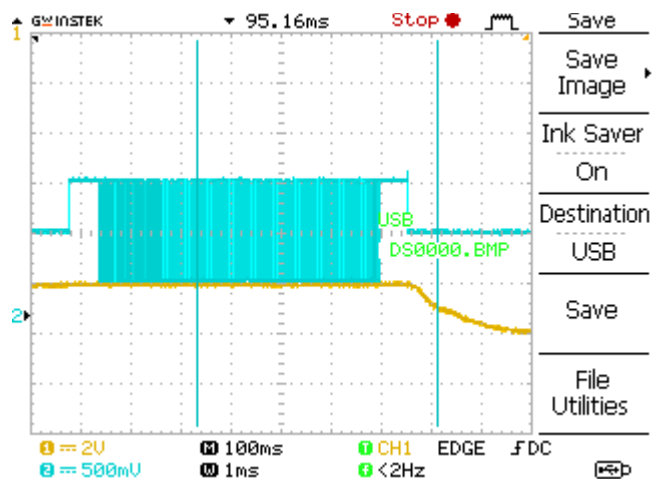


Joonis 3: Viide näputõstmise ja LEDi pingemuutuse vahel

Katse „Lisa 1. Viide rakendusest antud käsu ja pingemuutuse vahel”, lk 58 tulemuse alusel on viide näpu ekraanilt tõstmise ning pingemuutumise alguse vaheline viide 120 ms. Seega on ideaaltingimustes kogu süsteemi viide 120 ms.

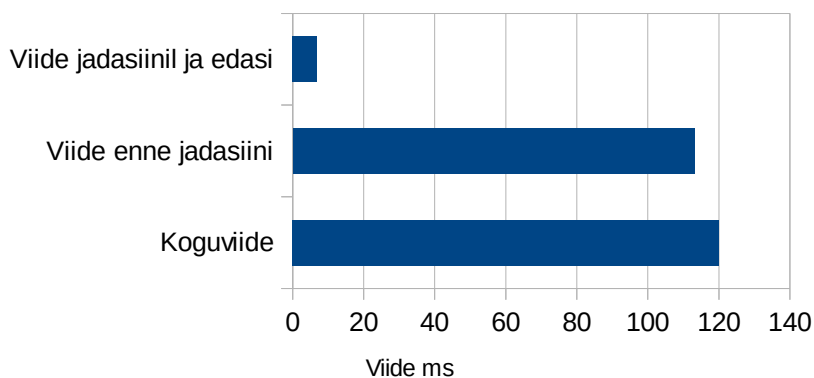


Selleks et aru saada, kui suur osa viitest jääb Wi-Fi ühenduse ja rakenduse poole, saab süsteemi jaotada kaheks. Kõik enne jadasiini ja jadasiin koos LEDi pingemuutumisega.



Joonis 4: Viide käsipaketi alguse ja LEDi pingemuutuse vahel

Katse „Lisa 2. Viide jadasiinil info vastuvõtu alguse ja LEDi jalal pingemuutmise lõpu vahel”, lk 59 tulemuse alusel on jadasiinil viide paketi vastuvõtu alguse ja pingemuutuse alguse vahel 6,8 ms, kusjuures paketi vastuvõtmise ajaga võrreldes on VSTPs paketi töötlemise aeg väga väike.



Joonis 5: Viite jaotumine süsteemis

Nende katsete tulemuste alusel on selgelt näha, et põhiline osa viivitusest jääb Wi-Fi ühenduse ja rakenduse poole.

### **3.2 Süsteemis tekkiva viite analüüsi kokkuvõte**

Katsete tulemusel selgus, et kõige suurem viide käsu andmise ja valguse muutuse vahel on tingitud andmete edastamisest Wi-Fi võrgus. Kui kasutatakse ruuterit, siis suureneb viide veelgi.

Kui viide läheb veel suuremaks, siis paneb kasutaja seda tähele. Kasutaja jaoks ei muutu valgus kohe, kui kasutaja annab käsu nutiseadme rakendusest. Suur viide tekitab tunde, et valgustussüsteem ei reageeri käskudele sujuvalt ja kohe.

Järgmises etapis tuleks mõõta viidet, mis tekib Wi-Fi moodulis ja enne jadasiinile jõudmist, kuna Wi-Fi moodulit on võimalik muuta. Wi-Fi infrastruktuuri ei ole võimalik muuta.

## **4. Etherneti võrgu transpordikihi protokollide analüüs**

VSis saadetakse käsud Wi-Fi moodulile TCP protokolliga abil. TCP protokolliga tugevuseks on see, et andmete ülekandel on lihtsam realiseerida seda osa, mis puudutab VSTP andmete vahetamist, kui ei saadeta pakette ainult ühes suunas, vaid andmevahetus on mõlemas suunas.

Valguse juhtimise käskude saatmiseks, mis toimub ühesuunaliselt, oleks parem valik UDP protokoll, mis oleks kõige kiirem viis pakettide edastamiseks ning süsteemi ei mõjutaks pakettide uuesti saatmisest tekitatud viivitused.

Mõlema korraga kasutamine oleks ka variant, kuid siin võib saada piiravaks manussüsteemi Wi-Fi moodulite võimekus. Moodulid toetavad tavaliselt UDP või TCP ühendust, kuid ei toeta nende kasutamist korraga.

### **4.1 TCP**

TCP protokolliga ehk edastusohje protokolliga olemus dikteerib selle, et kõik andmed tuleb edastada, ja kui osa läheb saatmise käigus kaduma, siis tuleb need uuesti saata.

Kui andmete edastamisel lähevad paketid kaduma või manussüsteemi Wi-Fi moodul ei suuda rakendust andmete kättesaamisest teavitada, siis valguse sujuv muutmine lakkab, kuna üritatakse rakenduse jaoks kaduma läinud pakette uuesti saata.

Nutiseadme võrguühenduse logis on selgelt näha manussüsteemi Wi-Fi mooduli saadetud DUP ACK paketid ja nutiseadme rakenduse reageerimine neile FAST Retransmission paketiga. Selline käitumine on FTP võrgus täiesti normaalne, kuid suur DUP ACK pakettide arv viitab pakettide kaole võrgus.

Selleks et andmeid TCP ühenduse kaudu edastada, peab rakendus esmalt looma TCP ühenduse ja saama aru, kas ühendus on aktiivne või tuleb mitteaktiivne ühendus katkestada ja uuesti ühenduda. Kuna vaikeolekus ei liigu TCP ühenduse kaudu andmeid, siis on rakendusel võimatu teada, kas ühenduse teine pool on ühendatud ja võimeline käsked vastu võtma. Selleks et kontrollida, kas teine pool on ühendatud, tuleks rakenduses implementeerida ühenduse

kontroll, kus vaikeolekus saadetakse ühenduse kaudu pakette ja selle alusel hinnatakse ühenduse olekut, või kasutada TCP protokolliga *keepalive* funktsionaalsust. ([1] peatükk 17).

See võimaldab jätta rakenduses realiseerimata TCP ühenduse kontrolli. Nimelt võib manussüsteemi Wi-Fi moodul võrgust kaduda halva levi või elektrikatkestuse korral ja ilma TCP *keepalive* funktsionaalsuseta ei saaks rakendus sellest teada. *Keepalive* kasutamine võimaldab avastada moodulite kadumist nii, et koormus võrgule on minimaalne ning protsess ei sega käskude edastamist. Samuti väldib see ühenduse katkemist (ruuteri taimaut), siis kui rakendus töötab, kuid ei saada käske.

Kui *keepalive* funktsionaalsuse asemel tuleks realiseerida kontroll rakenduses, oleks põhiliseks probleemiks see, et VSTPd peaksid vastama rakendusest saadetud päringule olukorras, kus nende ressursid on piiratud ja vastused võivad tulla viitega. See aeglustaks andmete sükroniseerimist rakendusega ning põhjustaks täiendava koormuse, kuna vastuse kaootilise viite tõttu võib rakendus saata liiga palju päringuid, sest rakendus oletab, et päring ei jõudnud võrgu probleemide tõttu kohale.

Rakenduses TCP *keepalive* aktiveerimine sõltub nutiseadme operatsioonisüsteemist. Funktsionaalsus on realiseeritud madalamal tasemel ja rakenduses toimub ainult funktsionaalsuse aktiveerimine ning puudub funktsionaalsuse realiseerimise vajadus rakenduses.

Aktiviseerimisel tuleb valida parameetrid nii, et need tagaksid ühenduse kadumise avastamise mõistliku aja jooksul, kuid väldiksid ühenduse katkemise kohe, kui mõnele *keepalive* paketile kohe vastust ei saada, kuna tegemist võib olla paketikao, mitte ühenduse katkemisega.

Rakendusele esitatud nõuete ja testimise kaudu selguvad optimaalsed väärtused, mille alusel otsustatakse, et TCP ühendus on katkenud. („Lisa 4. iOS rakenduses TCP *keepalive* seadistamine” lk. 64)

Kui ühendus on olemas, siis mõjutab valguse muutumise sujuvust käskude edastamise viis. TCP protokolliga kasutamist saab seadistada lähtuvalt sellest, milliseid andmeid ja kuidas on vaja edastada. Seadistus on erinev lähtuvalt sellest, kas tegemist on interaktiivse andmeedastusega ehk oodatakse reaalajas vastust või soovitakse edastada näiteks suurt andmefaili, kus ei ole tähtis saada interaktiivset tagasisidet, vaid edastada fail nii, et võrku ressursi kasutataks minimaalselt.

VSi on tähtis, et igale käsule järgneks kohe valguse muutumine. Interaktiivsus seab esikohale andmete edastamise kiiruse.

TCP ühenduse andmevoo seadistamist kirjeldab [1] peatükk 15. Kuna esikohale on seatud kiirus, siis ei ole tähtis, kui suure osa edastatavatest andmetest moodustab TCP protokollis päis. Kuna andmete edastamine toimub kohtvõrgus, ei ole probleemi ka võrgu ressursiga, kuna edastatavad paketid on väikesed ja võrgu läbilaskevõime on suur.

Selliste pakettide edastamise viisi reguleerib Nagle algoritmi [2] kasutamine. Kui Nagle algoritmi kasutamine on sisse lülitatud, siis ei saadeta andmeid enne, kui eelmiste andmete kättesaamise kinnitus on kätte saadud. Sisuliselt tähendab see seda, et kuni kinnituse saamiseni kogutakse väikesed paketid ühte segmenti kokku (kuni paketid ei ületa TCP segmenti suurust) ja saadetakse siis korraga. Selline oota ja saada lähenemine ei sobi VSi juhtimiseks. Nagle algoritmi väljalülitamise nõue on kirjeldatud RFC nõuetes [3].

Nagle algoritmi kasutamise väljalülitamine sõltub kasutatavast operatsioonisüsteemist. Rakenduse puhul nimetatakse seda parameetrit `TCP_NODELAY` vms., manussüsteemimoodulis on selle parameetri muutmiseks moodulikohane käsk.

Interaktiivse suhtluse tagamiseks tuleb Nagle algoritmi kasutamine välja lülitada nii rakenduses kui ka manussüsteemi Wi-Fi moodulites. [4]

Tähelepanu tuleks pöörata ka pakettide uuesti saatmise protsessi seadistamisele. Valgustuse juhtimise käskude tagantjärele saabumine põhjustab pigem segadust ning mõjub käskude puhverdamisena. Kuna käske soovitakse edastada reaajas, siis kõik hiljem uuesti saadetud paketid ei oma valguse juhtimise mõttes tähtsust, v.a. viimane valguse juhtimise pakett, mis peab tagama selle, et valgus vastab viimasele juhtkäsule (lõppseis).

Seega tuleks pakette uuesti saata pigem varem ja kiiremini, et taastuks normaalne andmeedastus. Uuesti saatmisel kasutatakse *fast retransmit* algoritmi ([1] 14.5). Tavaliselt saadetakse andmed uuesti, kui vastuvõtja saadab kolm DUP ACK paketti ([5] peatükk 3.2). Rakenduses tuleks TCP ühendus seadistada nii, et DUP ACK ide arv (parameeter *duplicate ACK threshold* või *dupthresh*) oleks kolm või isegi väiksem, kuna erinevates operatsioonisüsteemides võib seadistus erineda või automaatselt muutuda. Pakettide uuesti

saatmiseks on olemas veel uuemaid ja paremaid algoritme, kuid nende puhul võib probleemiks osutuda see, et manussüsteemi Wi-Fi moodulid ei toeta neid algoritme.

## 4.2 UDP

UDP ehk kasutajadatagrammi protokoll [6] erineb jadaohje protokollist põhiliselt selle tõttu, et protokoll ei loo pakettide edastamiseks ühendust ja ei taga pakettide kohalejõudmist ning järjekorda. Protokoll tagab minimaalse funktsionaalsuse mis tähendab seda, et rakendusel on suurem kontroll selle üle mida ja kuidas saadetakse. See omakorda tähendab seda, et rakendus peab vajadusel implementeerima kättesaamise kontrolli ja pakettide järjestuse kontrolli. Lisaks ei ole kasutajadiagrammi protokollil ka topelt pakettide eemaldamist, andmevoo kontrolli ja võrgu ülekoormatuse kontrolli.

VS jaoks on UDP ainus hea omadus see, et valguse muutmise käskude saatmisel rakendusest seadmele ei teki viivitust, kui mõni pakett ei jõua kohale, kuna UDP protokoll ei taga pakettide kohalejõudmist erinevalt TCP protokollist, mis palub kaduma läinud paketti uuesti saata.

Manussüsteemi Wi-Fi mooduli vastuvõtu puhvri ületäitumist ei tohiks tekkida, kuna rakendus saadab valguse muutmise käske jättes käskude vahele viite. Viide jäetakse selleks, VSTP jõuaks käsku töödelda ning vajadusel vastata.

Viidete kasutamine, väikesed paketid ning kohaliku võrgu suur läbilaskevõime peaks vältima võrgu ummistumist.

UDP protokoll kasutamise ei ole otstarbekas kuna rakendus vajab lisaks valguse muutmise käskude edastamisele info sünkroniseerimist rakenduse ja VSTP vahel. Kui kasutada ainult UDP protokoll, siis tuleks implementeerida andmete kättesaamise kontroll ja uuesti saatmine. VSTP protsessori võimsus ja aeg on piiratud, seega võib sünkroniseerimine venida pikaks, kuna vastuse genereerimine võtab VSTP aega ning tuleks kasutada viiteid enne kui rakendus infot uuesti küsiks, et vältida uut päringut ajal, kui VSTP vastust koostab.

### **4.3 Mõne muu protokollide kasutamine**

Manussüsteemi Wi-Fi moodulid toetavad tavaliselt transpordi kihis ainult TCP ja UDP protokolle. Selle tõttu ei ole teiste transpordikihi protokollide analüüsimisel mõtet.

### **4.4 Etherneti võrgu transpordikihi protokollide analüüsi kokkuvõte**

Valguse sujuvaks muutmiseks sobib kõike paremini TCP protokollide kasutamine. Kuigi pakettide uuestisaatmine võib tekitada pause, on selle mõju palju väiksem kui UDP protokollide kasutamisel vajadus saada VSTPlt tagasisidet selle kohta kas käsk on kätte saadud.

TCP protokollide kasutamine muudab ka süsteemi implementeerimise lihtsamaks, kuna jadasiinil tekkida võiva paketikaole lisaks ei ole vaja lahendada paketikadu Etherneti võrgus. Kui TCP ühendus on õigesti seadistatud, võimaldab see avastada ühenduse katkemist ilma valgustussüsteemi riistvara ressursse kasutamata (TCP *keepalive*).

## **5. Etherneti võrgu rakenduskihi protokollide analüüs**

### **5.1 Valgustussüsteemi jaoks loodud protokoll**

Valgustussüsteemis on valguse muutmise käskude edastamiseks kasutusel süsteemi oma protokoll. Esimeses etapis võeti kasutusele protokollid, kus juhtimine toimus toiteploki põhiselt ehk kogu paketti koondati kõikide toiteplokkide jaoks eraldi käskud. Selline lähenemine on kõige paindlikum, kuna siis saab kõiki toiteplokkide juhtida eraldi kasutades toiteploki unikaalset identifikaatorit. Selle kaudu saab toiteplokk aru, et tegemist on temale mõeldud teatega. Sellise lahenduse puuduseks on see, et pakettide suurus muutub vastavalt sellele, mitu toiteploki ja lampi valgustussüsteemis on.

Protokollid lisandusid pakettidele, mis võimaldavad ühe jadasiiniga ühendatud toiteplokkide juhtida koos. Ühe jadasiiniga ühendatud toiteplokkide lampe saab määrata kanalitesse, igal kanalil valgust saab juhtida eraldi. Lähtuvalt sellest ei sõltu valguse juhtimise käsu paketi suurus süsteemi suuruselt vaid kanalite arvust. Kolme kanalil juhtimise paketi suurus on 128 baiti.

Mõlemaid pakette on võimalik süsteemis korraga kasutada. Pakettide suurus on minimaalselt 128 baiti ja maksimaalselt 1024 baiti. Nende vahele jäävad suurused on täisarv korda 128 baiti nagu 256, 512 ja 768 baiti.

Kui värve saab juhtida kanalite järgi, siis näiteks kanalite määramise pakettide suurus sõltub ikka süsteemi suuruselt, kuna kanalite määramine peab käima iga toiteploki kohta eraldi. Samas ei oma siin süsteemi suurus nii suurt tähtsust, kuna iga toiteploki kohta on vähem infot (võrreldes värvide infoga), seega jääb ka kanalite määramise paketi suurus üldjuhul 128 baiti suuruseks.

Suurema paketi edastamisel on vaja paremat võrku ja rohkem aega paketi töötlemiseks (viide pakettide vahel). Üheks piiranguks saab ka jadasiini kiirus.

Selleks, et teha protokollis muudatusi, on vaja hinnata iga süsteemi osa lähtuvalt edastatavate pakettide suuruselt.



### 5.1.1 Pakettide suuruse mõju lähtuvalt andmete edastamisest Wi-Fi võrgus

Kuna süsteemis ei kasutata pakette, mis ei ole suuremad, kui 1024baiti, siis ei ole ohtu, et pakett ei mahu võrgu maksimum-ülekandeühikusse(MTU).

### 5.1.2 Pakettide suuruse mõju lähtuvalt valgustussüsteemi programmi tööst

Valgustussüsteemis piirab pakettide suurust jadasiini kiirus. Jadasiini kiirus on 250 000 bit/s.

Kuna kasutusel on 1 start bit, 8 andmebiti ja 2 stoppbitti, siis teoreetiliselt võiks andmeid edastada kiirusel  $250\,000 * (8/(1+8+2)) = 181\,818$  bit/s. Praktikas on mõistlik arvestada poolega sellest kiirusest 90909 bit/s, kuna peale käsu vastuvõtmist on vaja aega käsu töötlemiseks, vastasel juhul iga uue paketi saabumine katkestab käsu töötlemise ja täitmise.

Tabel 1: Jadasiini parameetrid

Jadasiini kiirus	250000	bitti/s
Ühe baidi andmete edastamiseks	11	bitti
Praktiline edastuskiirus	90909	bitti/s

Tabel 2: Käskude arv sekundis lähtuvalt käsu andmepakettide suurusest

Andmepaketid	baiti	bitti	ülekandeaeg ms	käsku sekundis
128 baiti	128	1024	11,26	88,78
256 baiti	256	2048	22,53	44,39
512 baiti	512	4096	45,06	22,19
1024 baiti	1024	8192	90,11	11,1

Kui võrrelda käskude edastamise kiirust DMX512 protokolliga, mille korral on võimalik sekundis edastada 44 korda käsku 512 kanali jaoks (käsk 5700 bitti), siis 128 baidiste käskude puhul peaks saama edastada kaks korda rohkem käske.

### 5.1.3 Pakettide saatmistiheduse mõju lähtuvalt andmete edastamisest Wi-Fi võrgus

Kuna edastatavad andmemahud on väikesed ja nutiseadme rakendus peab jätma pakettide vahele viiteid, et anda VSTPle aega andmete töötlemiseks, siis Wi-Fi võrgu läbilaskevõime ei saa pakettide edastamisel takistuseks.

#### **5.1.4 Pakettide saatmistiheduse mõju lähtuvalt valgustussüsteemi programmi tööst**

VSTPle lisaks tuleb siin arvestada ka Wi-Fi mooduli tööajaga. Kui nutiseadme rakendus saadab valguse juhtimise käsu üle Etherneti võrgu Wi-Fi moodulisse, siis Wi-Fi moodul töötleb vastuvõetud käsku ja saadab selle edasi jadasiinile sobiva kiirusega. Kui Etherneti võrgus läheb mõni pakett kaduma, siis uuesti saatmise (TCP) tõttu lähevad kaduma nutiseadme rakenduse tekitatud viited pakettide vahel. Nüüd peab Wi-Fi moodul pakette puhverdama ja saatma neid jadasiinile edasi, lisades pakettide vahele viiteid, mis võimaldavad VSTPdel andmeid töödelda.

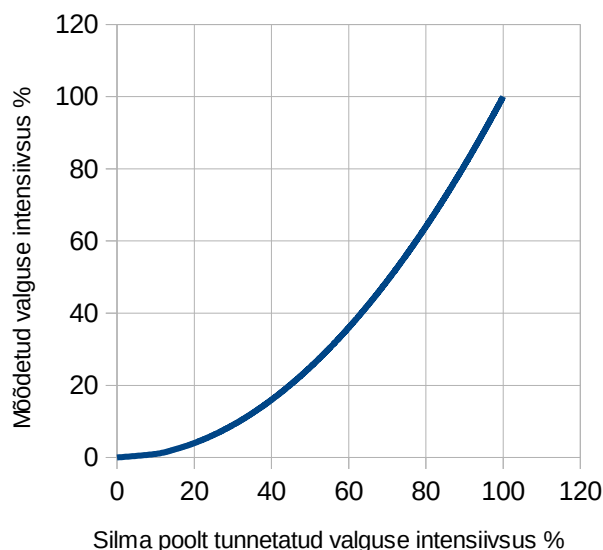
Selline puhverdamine Wi-Fi moodulis võib viia selleni, et kui kasutaja valguse muutmise lõpetab, siis tulevad muudatused veel tagantjärele. Selle probleemi leevendamiseks võiks Wi-Fi moodul võrrelda viimati saadetud paketti uue paketiga. Kui paketi sisu on sama, siis ei ole vaja seda edasi saata ja paketi võib vahele jätta. See võimaldaks puhvrit kiiremi tühjendada. Sellise lähenemise probleemiks on see, et kui mitu korda saadetud sama paketti ei saadeta edasi ja esimene pakett läheb jadasiinil kaduma, siis ei ole võimalik näiteks saata perioodiliselt sama paketti, et tagada, et kõik jadasiiniga ühendatud VSTPd saaksid paketi kätte. Pakettide võrdlemine kasutab Wi-Fi mooduli protsessori ressursi ja muudab Wi-Fi mooduli programmi keerulisemaks.

#### **5.1.5 Pakettide saatmistiheduse mõju lähtuvalt kasutaja tunnetusest, kas valgus muutub sujuvalt**

Süsteem on projekteeritud selliselt, et lampide valgus ei vilgu. Kasutajale võib vilkumisena tunduda kiire valguse intensiivsuse muutumine. Kui valguse muutmise käsku saadetakse harva, siis valguse sujuv suurendamine nutiseadme rakendusest võib silma jaoks tunduda diskreetsete hüpetena ehk kasutaja tunnetab, et valgus ei muutu sujuvalt.

Tajutava valguse muutuse võib jagada kaheks. Valguse intensiivsuse muutuseks ja valguse värvi muutuseks.

Inimsilm ei tunneta valguse intensiivsust vastavalt mõõdetud intensiivsusele [7].



Joonis 6: Mõõdetud ja tunnetatud valguse intensiivsus

$$\text{silмага tunnetatud intensiivsus} = 100 * \sqrt{\frac{\text{mõõdetud intensiivsus}}{100}}$$

Sellest tulenevalt peaks väikese intensiivsuse korral saatma käske võimalikult tihti, kuna silm on väga tundlik ja paari protsendine muudatus tundub kasutaja silma jaoks diskreetse hüppena. Suure intensiivsuse korral on vastupidi, kasutaja ei pruugi tähele panna isegi 10% intensiivsuse muutumist. Väikese intensiivsuse korral ei piisa käskude tihti saatmisest, kui selles piirkonnas ei ole piisavalt palju DAC samme valguse sujuvaks muutmiseks (kui ka üks DAC samm tundub kasutaja jaoks diskreetse hüppena).

Valguse värvi muutus võib kasutajale tunduda hüppeline olukorras, kus valguse värvi muudetakse ja liigutakse sujuvalt soojemalt temperatuurilt külmemema poole. Mingil hetkel süttivad sinised LEDid. See süttimise hetk on silmale selgelt näha, kuna valgusesse lisandub uus lainepikkus. Kuna valgus segatakse kokku erinevat värvi LEDide valgusest, siis on tegemist sellise lähenemisviisi omadusega, mida ei saa elimineerida.

Värvi muutuse tundlikkus sõltub inimsilma ja kaamera tundlikkusest erinevatel valguse lainepikkustel, selle tõttu tajuvad kasutajad värvi muutust erinevalt.

## 5.2 Art-Net

Art-Net [8] on protokoll, mis võimaldab edastada DMX512 protokollis käsked Etherneti võrgus. Edastamiseks kasutatakse UDP protokollis. Kui valgustussüsteemi oma protokollis saab arendada ja optimeerida, siis DMX512 protokollis kasutades seda teha ei saa. Kuigi DMX512 protokollis on kahepoolset suhtlust võimaldav lisa RDM (Remote Device Management), on see efektiivseks kahepoolseks suhtlemiseks kohmakas ja aeglane. RDM käsked edastatakse DMX pakettide vahel. DMX512 pakette saadab süsteemis üks konsool ja kõik seadmed võtavad käsked vastu. Süsteem on üles ehitatud sellele, et konsool saadab pakette pidevalt. Kui mõni seade ei saa esimest paketti kätte, siis tuleb kohe järgmine jne. ehk see peaks tagama, et kõik seadmed saavad info kätte olenemata paketikadudest või võrguprobleemidest.

Selle tõttu on konsool see, mis küsib iga seadme käest eraldi infot ja tekitab saatmisel pausi, et seade saaks vastata. Isegi siis võivad paketid kaduma minna.

Valgustussüsteemi oma protokollis asendamine Art-Net ehk sisuliselt DMX512 protokolliga läheks vastuollu olemasoleva arhitektuuriga. Kui DMX512 arhitektuuri puhul kõiki andmeid haldab ja edastab üks konsool, siis analüüsitava valgustussüsteemi puhul VSTPd teavad oma andmeid ja suudavad töötada autonoomselt. Nutiseadme rakendus sünkroniseerib ühendumisel andmed VSTPst ja siis saab kasutaja hakata neid muutma. Selline lähenemine võimaldab kasutajal muuta VSTP seadeid käsitsi ja jätkata hiljem nutiseadme rakendusest ilma, et käsitsi tehtud muudatused kaduma läheksid.

DMX512 protokollis puhul ei ole head lahendust olukorrale, kui juhitalval seadmel on vaja teavitada ootamatult tekkinud veast. Puudub ka võimalus seadmete oleku kontrollimiseks ehk puudub info kas seade on ühendatud või mitte.

Analüüsitava süsteemi eesmärk on anda kasutajatele reaajas visuaalne ülevaade ühendatud seadmetest ning süsteemis toimuvast. DMX512 protokollis kasutamine piiraks täiendavate lisafunktsioonide arendamist ning süsteemile pandud eesmärkide täitmist.

Süsteem võiks toetada Art-Net protokoll, kui seda juhitakse klassikalisel viisil spetsiaalse konsooli kaudu, kuid Art-Net protokoll ei sobi suhtluseks nutiseadme rakendusega, kuna selle abil ei ole võimalik teostada seatud eesmärke.

### **5.3 Etherneti võrgu rakenduskihi protokollide analüüsi kokkuvõte**

Analüüsitava valgustusüsteemi ülesehitus on selline, et valguse parameetrid salvestatakse VSTPdesse erinevalt varasemate lahendustega, kus valguse parameetreid teadis vaid üks konsool, mis juhtis kogu süsteemi. Sellest tulenevalt ei saa kasutada varasemalt kasutuses olnud protokolle vaid tuleb kasutada süsteemi jaoks loodud protokoll.

Uus protokoll võimaldab edastada käske kiiremi ning toetab kiiret andmete sünkroniseerimist ning veateadete saatmist. Selle tõttu sobib see kõige paremini nutiseadme rakendusega suhtlemiseks.

## **6. Traadita võrgu seadmete analüüs**

Wi-Fi võrgu loomiseks kasutatavad seadmed võiksid olla võimalikult universaalsed ja odavad ning võimaldaks kasutada juba olemasolevaid võrguseadmeid. Spetsiaalseadmete hind on kallis, ning nende tellimine võtab aega. Lisaks ei sobi üks seade kasutamiseks globaalselt, kuna Wi-Fi sagedused ja võimsused on riigiti erinevad. Spetsiaalseadmete arendamine iga turu jaoks teeks olukorra veel keerulisemaks.

Isegi laboritingimustes on ruuterite erinevused selgelt näha, kuigi parameetrite järgi on tegemist samaväärsete seadmetega.

### **6.1 Üldised probleemid Wi-Fi seadmete kasutamisel**

Kuna Wi-Fi kasutamine on maailmas väga laialt levinud on Wi-Fi seadmeid ja mudeleid väga palju. Selle tõttu on väga keeruline koostada juhendit, kuidas tuleks Wi-Fi võrguseadmeid häälestada, et valguse muutmine oleks sujuv. Sellise juhendi puudumine eeldab, et kasutajad oskavad võrguseadmeid häälestada ja võrgus tekkivaid probleeme lahendada. Valgustuse juhtimisel on tänaseni kasutusel DMX512 protokoll kasutavad konsolid ja nende kasutamine ei nõua teadmisi Etherneti võrgust ja selle seadistamisest. Selle tõttu on raske kasutajaid juhendada seadistamisel või probleemide lahendamisel. Lisaks on kasutajad harjunud olemasoleva süsteemiga ning suhtuvad muudatustesse vastumeelsusega.

Probleeme tekitab ka see, et Wi-Fi seadmete puhul eeldatakse, et võrku kasutatakse arvuti või nutiseadmega veebilehtede ja videote vaatamiseks. Sellest tulenevalt üritab Wi-Fi seade automaatselt vahetada sagedust vastavalt signaali tugevusele ja pakkuda rändlust jms. Valgustussüsteemi juhtimisel tekitavad sellised automaatsed muudatused probleeme. Valgustussüsteemi Wi-Fi manussüsteemi moodul ei pruugi toetada kõigi sagedusalasid ning võrkude vahetamine ning võrgu vigade lahendamine ei ole nii arenenud kui arvutites ja nutiseadmetes.

Seega peaks kasutaja seadistama võrgu nii, et kõikidel sagedustel oleks unikaalne SSID (võrgunimi), et vältida automaatset võrguvahetust.

Kuna tegemist on traadita ühendusega, siis kasutaja peaks olema kursis sellega, kuidas materjalid mõjutavad raadiolainete mõju ning kuidas kaardistada sagedusala kasutamist. Kui Wi-Fi võrku kasutatakse tihedalt asustatud piirkonnas, siis on väga suur tõenäosus, et seal on palju teisi saatjaid. Kasutaja peaks leidma vaba sageduse või vältima sattumist samale sagedusele teise lähedalasuva (tugeva signaaliga) Wi-Fi seadmega.

Kui tegemist on suletud alaga, siis valgustussüsteemi ülesseadmise ajal võib kogu sagedusala vaba olla, kuid võteteks tuuakse seadmeid juurde, mis hakkavad sama sagedusala kasutama. Selle tõttu võib tekkida olukord, kus pärast valgustussüsteemi ülespanekut süsteem töötab korralikult, kuid võttepäeval ei saa enam valgustussüsteemi üle Wi-Fi juhtida, kuna lisandunud Wi-Fi saatjad ja muud sama sagedust kasutavad seadmed segavad valgustussüsteemi juhtimist. Kuna nende seadmete väljalülitamine ei ole võimalik, siis peaks kasutaja suutma olukorrale reageerida, kuna kaugelt nõu anda on väga raske.

## **6.2 Infrastruktuuri piirangute kaardistamine**

Wi-Fi võrgu sagedused ning võimsused ja selle kaudu ka leviulatus on standardiseeritud. Siiski on erinevusi riikide vahel ([9] lisa B). Saatja väljundvõimsus ja osaliselt ka sagedusala on riigiti erinev. Sellest tulenevalt võib keegi võttel kasutada illegaalselt võimsamat Wi-Fi saatjat, mis segab või takistab teiste seadmete tööd.

Seadmete soovitamine on raske, kuna uusi mudeleid tuleb turule väga palju ning neid ei jõua testida, eriti siis, kui valgustussüsteemi müüakse globaalselt.

Mõnel Wi-Fi ruuteril on sisseehitatud aku, mis teeb ruuteri töökindlamaks olukorras, kus elektrivarustust tagavad generaatorid. Kui valgustussüsteemi seadmed lülituvad välja, siis sisselülitudes üritavad nad võrguühendust ruuteriga taastada. Kui ruuter ei ole veel käima läinud, siis võib tekkida probleeme ühenduse taastamisega või võtab see kauem aega. Seega oleks väga hea, kui ruuteril oleks autonoomne toide.

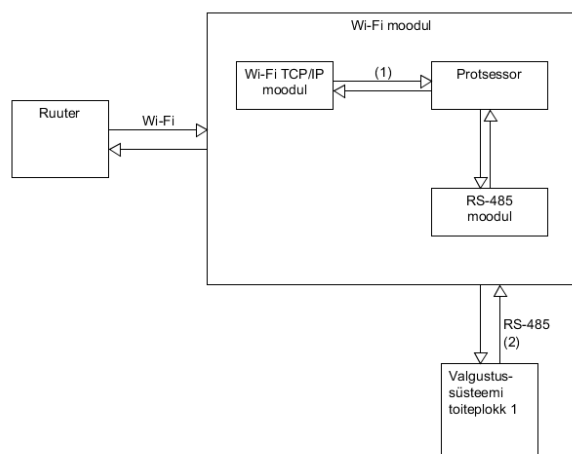
Ruuteritel on ka n.ö. peidetud omadusi ja vigu, mis hakkavad tööd segama kui pannakse kokku suurem valgustussüsteem. Näiteks ruuter teeb restardi, kui ruuteriga ühendub mingi kindel arv valgustussüsteemi Wi-Fi mooduleid.

Seoses sellega tuleks valida välja paar *premium*-klassi ruuterit mida on maailmas igal pool saada ning soovitada nende kasutamist.



### 6.3 Valgustussüsteemi Wi-Fi moodul

Valgustussüsteemi Wi-Fi moodul suhtleb nutiseadme rakendusega üle Wi-Fi võrgu ja edastab andmed RS-485 jadasiinile.



*Joonis 7: Valgustussüsteemi Wi-Fi mooduli plokskeem*

Wi-Fi mooduli sees TCP/IP moodul edastab andmed universaalse asünkroontransiiveri (UART) kaudu protsessorile, mis lisab andmepakettidele ühe stopbiti ja teisendab andmeedastuse kiiruse RS-485 siinile sobivaks. RS-485 moodul tekitab RS-485 signaali.

Selleks, et aru saada, kuidas toimub andmeedastus Wi-Fi mooduli sees valisin kaks mõõtepunkti ostsillograafia signaali mõõtmiseks. Üks mõõtepunkt oli Wi-Fi TCP/IP mooduli ja protsessori vahel(joonis 7 (1)) ning teine Wi-Fi mooduli väljundi ja valgustussüsteemi toiteploki vahel(joonis 7 (2)). Saadud tulemuste („Lisa 3. Pakettide edastamine Wi-Fi moodulis” lk. 60 ) alusel selgus, et protsessor ei kasuta voo reguleerimist ning selle tulemusena saadab TCP/IP moodul protsessorile rohkem andmeid, kui see suutab töödelda. Nii läheb osa pakettidest kaduma. Kuna pakette läheb kaduma täiesti juhuslikult ning nutiseadme rakendus ei saa sellest kuidagi teada (üle TCP jõudsid andmed edukalt kohale) ei saa nutiseadme rakenduses kuidagi olukorda parandada.

Esimese asjana tuleks muuta protsessori programmi nii, et võetakse kasutusele voo reguleerimine ning tagatakse, et kõik TCP/IP mooduli edastatud info jõuaks RS-485 siinile. Kui see on tehtud tuleks täpsemalt uurida, mis põhjustab viidet andmete edastamisel

protessorile, olukorras, kus rakendus saadab pakette järjest võrdsete vahenditega. Tegemist võib olla IP võrgu probleemidega või Wi-Fi TCP/IP manussüsteemi mooduli probleemi või omadusega. Testimisel tuleks kasutada pakette, kus sisu oleks võimalikult unikaalne, et saada aru, kas ja kuidas pakettide järjekord muutub või kas mõni pakett läheb kaduma.

Valgustussüsteemi Wi-Fi moodulis on kasutusel manussüsteemi Wi-Fi moodul. Selliseid mooduleid on turul palju, kuid need on väga erinevad. Moodulitel on erinev väljundvõimsus, mis võib jääda alla maksimaalselt lubatud väljundvõimsusele, mille tõttu kannatab leviulatus ja ühenduse kvaliteet, eriti siis, kui läheduses on kasutusel palju Wi-Fi võrke.

Kõik moodulid toetavad reeglina töötamist kliendina ehk moodul suudab ühenduda ruuteri külge. Samas ei toeta paljud töötamist ruuterina. Valgustussüsteemi puhul on vaja, et Wi-Fi moodul suudaks töötada ka ruuterina, et kasutaja ei peaks väikese süsteemi kasutamisel kasutama Wi-Fi ruuterit.

Ideaalis peaks valgustussüsteemi Wi-Fi moodulis kasutama manussüsteemi moodulit, mille väljundvõimsus oleks nii suur, kui riigis on maksimaalselt lubatav. Parim viis oleks see, kui väljundvõimsust saaks muuta tarkvarast ja kui see ei ole võimalik, siis võiks olla võimalik Wi-Fi manussüsteemi mooduli vahetamine vastavalt riigile, kus valgustussüsteemi kasutatakse.

Manussüsteemi moodul peaks toetama ka ühenduse krüpteerimist. Krüpteerimise peamine eesmärk on see, et teised kasutajad ei saaks ühenduda valgustussüsteemi juhtimise võrku ja segada valguse juhtimist.

Uusi mooduleid tuleb pidevalt turule. Sellega seoses tuleb uusi mooduleid testida ja analüüsida nende sobivust valgustussüsteemis kasutamiseks, et leida mooduleid, mille abil paranda valgustussüsteemi juhtimist üle Wi-Fi võrgu.

## **6.4 Traadita võrgu seadmete analüüsi kokkuvõte**

Wi-Fi võrgu kanalid ja võimsused on riigiti erinevad ja seoses Wi-Fi väga laialdase levikuga on Wi-Fi ruutereid ja muid võrguseadmeid väga palju. Sellest tulenevalt tuleks välja valida paar *premium*-klassi ruuterit (võimalusel autonoomse toitega) ja soovitada nende kasutamist.

Nii on kõige lihtsam juhendada kasutajaid Wi-Fi võrgus tekkivate probleemide lahendamisel, kuna kasutajatel puudub kogemus ja teadmised Wi-Fi võrgu seadistamisel, kuna varasemad süsteemid seda ei nõudnud.

Valgustussüsteemi Wi-Fi moodulis tuleb võtta kasutusele voo reguleerimine ning tagada, et kõik TCP/IP mooduli edastatud info jõuaks RS-485 siinile.

Lisaks tuleks analüüsida manussüsteemi Wi-Fi mooduleid, et leida neist parim, kuna võrguprobleemid mõjutavad valguse sujuvat muutmist.

## **7. Valgustusüsteemis käskudele reageerimise analüüs**

Kui rakenduse poolt saadetud käsk on läbinud Wi-Fi võrgu ja on jõudnud jadasiinile peavad VSi toiteplokkid jadasiinilt käsu vastu võtma, seda töötleva ning seejärel LEDe juhtima. Toiteplokkide protsessori võime andmeid töödelda on piiratud, seda tuleb arvestada käskude saatmise tiheduse juures. Iga uus käsk tekitab astmelise valguse muutmise. Selle kõrvaldamiseks peab toiteplokk veel täiendavalt realiseerima sujuva LEDide juhtimise üleminekul uue käsuga ette nähtud olekusse ehk tekitada uue käsuga defineeritud valguse nii, et üleminek tundub kasutaja jaoks sujuv.

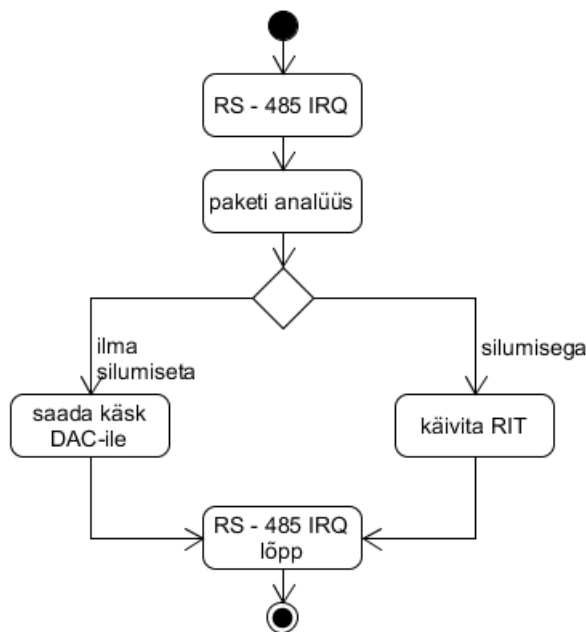
### **7.1 Jadasiinile käsu saatmise ja LEDi pinge muutuse vahelise aja mõõtmine**

„Lisa 2. Viide jadasiinil info vastuvõtu alguse ja LEDi jalal pinge muutmise lõpu vahel” lk 59 pildilt on näha, et jadasiinil saadetud paketi lõpu ja LEDi pinge muutuse vahel on väga väike viide, mis jääb alla  $200\mu\text{s}$ , seega viide käsu kättesaamise ja LEDide juhtimise alguse vahel on väga väike võrreldes näiteks paketi vastuvõtmise ajaga (6,8 ms). Seega paketi töötlemine toiteplokkis ja digitaal-analoogmuundurile esimese käsu edastamine on väga kiire võrreldes käsupakettide edastamise ajaga jadasiinil. Sellest lähtuvalt peaksid käsupaketid olema võimalikult väikesed ehk mida väiksem pakett, seda kiiremini suudab toiteplokk selle jadasiinilt vastu võtta.

## 7.2 Valgustussüsteemis diskreetsete üleminekute silumise analüüs

Kasutaja võib muuta valgust väga väga suures ulatuses väga lühikese aja jooksul. Kõige suurem muutus oleks valgusti lülitamine kahe äärmuse vahel (valgusti ei ole sisse lülitatud või töötab täisvõimsusel). Tavaliselt on muutused siiski väiksemate sammudega, kuid need sammud on selgelt nähtavad, eriti siis, kui lambid töötavad madalal väljundvõimsusel, mille juures inimese silm suudab selgelt eristada valguse intensiivsuse muutumist. Kuna käskude vahel on viide, siis kasutaja näeb valguse intensiivsuse muutumist astmeliselt. Looduses valgus ei muutu astmeliselt, seega mõjub astmeline valguse muutus inimese jaoks vilkumisena ning ebaloomulikuna.

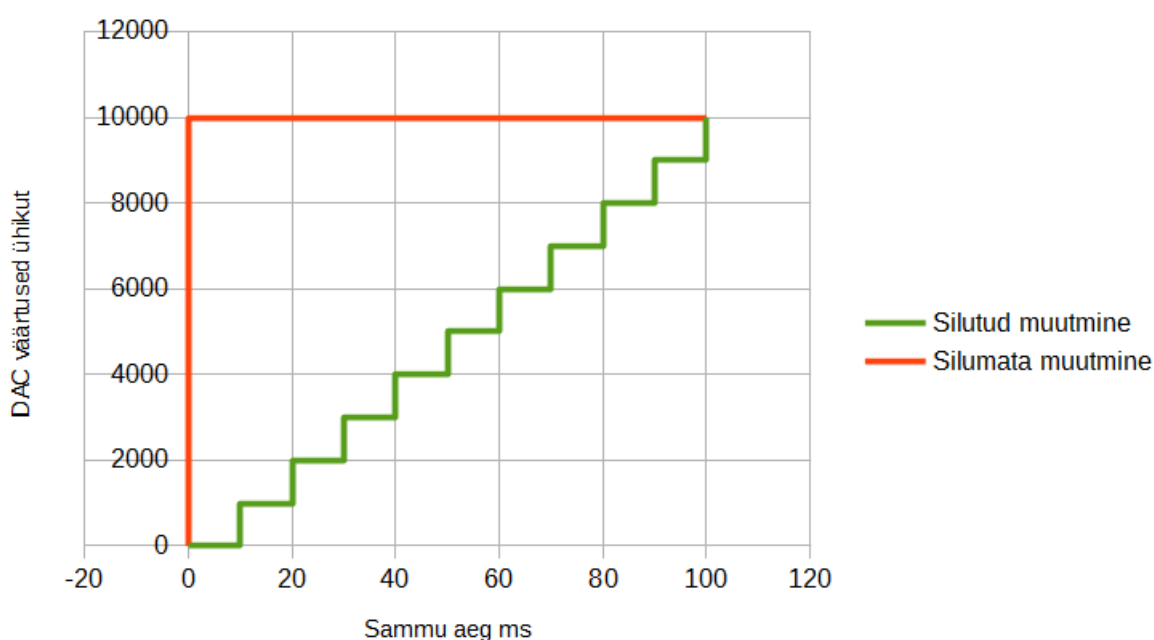
Selleks, et valguse muutus oleks sujuvam tuleb valguse üleminekut ühelt väärtuselt teisele siluda VSi toiteploki, kuna rakendus ei saa käske kiiremini saata. Praktikas tähendab see seda, et kahe väärtuse vaheline ala tuleb jagada sammudeks, mille jooksul valguse intensiivsust muudetakse igal sammul, kuni saavutatakse lõplik intensiivsus. Kuna tegemist on siiski sammudega ei ole valguse intensiivsuse muutus täiesti sujuv, kuid kasutaja jaoks tundub valguse intensiivsuse muutmine palju sujuvamana.



Joonis 8: Valguse muutuse silumise protsess

Toiteploki on võimalik saata valguse muutmise käske nii silumise nõudega kui ka ilma. Kui käsk saadetakse ilma silumise nõudeta, siis uus olek saadetakse digitaal-analoogmuundurile ja valguse intensiivsus läheb kohe üle uude olekusse.

Kui käsk saadetakse silumise nõudega, siis käivitatakse taimer (RIT), mis genereerib iga valguse juhtimise kanali (värvi) jaoks näiteks kümme sammu, mis saadakse hetke intensiivsuse ja uue valguse intensiivsuse vahe jagamisel võrdseteks tükkideks. Taimer käivitub kümme korda iga 10ms tagant ja saadab 1/10 oleku muutuse väärtusest digitaal-analoogmuundurile tekitades nii sujuvama ülemineku ühest olekust teise.



Joonis 9: Üleminek DAC väärtuselt 0 väärtusele 10000

Kui rakendus saadab käske kiiremini kui 100ms vahedega, siis silumisprotsess katkeb sellel hetkel, kui uus käsk saabub ja edasi toimub valguse muutmise juba uue käsu alusel.

100ms silumise aeg sai valitud katsetamise tulemusel, kuna see tundus kasutaja jaoks kõige sujuvam.

Hetke lahendus ei võta arvesse hetke intensiivsust ja intensiivsuse muudatuse amplituudi lähtuvalt inimesilma tundlikusest. Kuna valguse intensiivsuse kasvades silm ei ole nii tundlik valguse intensiivsuse muutuse suhtes, siis tuleks keskenduda just valguse intensiivsuse sujuvale muutmisele vahemikus, kus silm on kõige tundlikum. Samas ei saa unustada, et näiteks filmitööstuses on lisaks silma tundlikusele tähtis ka see, kuidas valguse muutus salvestub filmile.

Inimsilma jaoks ülemineku vaheväärtuste arvutamisel tuleks arvestada inimsilma tundlikust vastavalt valguse intensiivsusele. Näiteks, kui võrdsel jagamisel jääksid kolm viimast valguse intensiivsuse muutust alasse, kus silm on muudatuste suhtes tundetum ([10] lõik 5.3), siis oleks kasulikum kaks sammu nendest kasutada üleminekute genereerimisel madalama intensiivsusega osas ehk 100ms üleminekud oleksid jagatud eksponentsiaalselt kasvavalt. Kui lisaks intensiivsuse tundlikusele arvestada ka silma värvitundlikust, siis selline lähenemine teeks silumise implementeerimise toiteplokis oluliselt keerulisemaks ning ressursinõudlikumaks.

Nutiseadme rakenduses saaks hinnata kahe käsu vahelist erinevust ning kui see ei ole silmale tajutav, siis käsku mitte saata, et anda toiteplokile aega sujuva ülemineku lõpetamiseks(100ms). Kuna nutiseadme rakendusest saadetavate käskude vahel on viide võib tekkida olukord, kus kasutaja väga kiire suure amplituudiga valguse intensiivsuse muudatus nutiseadme rakenduses ei kajastu üldse reaalse valguse muutusena, kuna kasutaja tegi muudatuse ajal, kui saatmise viide oli aktiivne ja peale viite lõppu oli kasutaja jõudnud valguse intensiivsuse muutmisega tagasi sinna, kus ta oli enne viite algust. Sellise probleemi vältimiseks peaks vähendama käskudevahelise viite aega. Viite aeg sõltub põhiliselt VSi toiteploki riistvarast ja VSi Wi-Fi mooduli riistvarast.

### **7.3 Valgustussüsteemis käskudele reageerimise analüüsi kokkuvõte**

Valguse sujuva muutmise tagamiseks tuleks alati kasutada valguse muutuse silumist VSTPs. Eriti tähtis on see väiksemal valguse intensiivsusel, kuna siis suudab silm eristada ka väikeseid valguse intensiivsuse muutuseid. Edaspidi tuleks uurida võimalusi, kuidas parandada valguse sujuvat muutumist väikese intensiivsuse juures nii, et muutus tunduks veel sujuvam nii silma kui kaamera jaoks.

Kui vähendada viiteid käskude vahel, siis valguse ei muutu nii sujuvalt, kuid valgustussüsteem jätkaks vähem muudatusi saatmata selle tõttu, et käskude vahelise viite ajal ei saa käske saata. Tuleks analüüsida, kas viidete vähendamine on praktikas vajalik ning kas parem on sujuvam valguse muutus või kiirem reageerimine kasutaja tehtud muudatustele.

## **8. Nutiseadme rakendusest käskude saatmise analüüs**

Nutiseade peab käskude saatmisel arvestama sellega kui kiiresti VSTP suudab käske vastu võtta. Kui käske saata liiga kiiresti, siis iga uus käsk katkestab eelmise käsu sisselugemise ja nii jõuab kohale ainult viimane käsk. VSTPs käskude töötlemise kiirust saaks suurendada kiirema protsessori lisamisega, kuid seda ei saa vastavalt eeltingimustele teha. Seega tuleb käskude saatmise kiirust piirata nutiseadme rakenduses.

Selleks, et valguse muutumine oleks võimalikult ühtlane oleks vaja valguse muutmise käske saata võimalikult lühikeste ajavahemike tagant. Tähtis on ka see, et käske ei saadeta ajal, kui toimub sünkroniseerimine VSTPga või VSTP saadab nutiseadme rakendusele veateateid vms. Kuna VSTPd on omavahel ühendatud jadasiiniga ei saa samal siinil korraga andmeid saata ja käske vastu võtta, peab nutiseadme rakendus üritama andmevahetust hallata.

Kui nutiseadme rakendus saadab käsu VSTPle, et see saadaks tagasi VSTPde seaded, tuleb vastus mitmes osas. Vastamine võtab kauem kui mõnele teisele käsule vastamine. Sellest lähtuvalt peab nutiseadme rakendus ootama vastust lähtuvalt saadetud käsus. Vastuse ootamise aeg sõltub paketist ja VSTP tööst. VSTP tarkvara analüüsimise ja katsete tulemuste alusel saab leida igale käsule vastamiseks kuluva maksimaalse aja. Kui selle aja jooksul ei ole VSTP käsule vastanud saab eeldada, et kas VSTP ei saanud käsku kätte või vastus läks kaduma. Sellisel juhul peaks nutiseadme rakendus käsu uuesti saatma.

Sünkroniseerimise käskudele vastuse ootamine on tähtsam kui valguse muutmise käskude saatmine, kuna enne käskude saatmist on vaja valgustussüsteem ja nutiseadme rakendus sünkroniseerida. Seega peaks sünkroniseerimise ajal värvi muutmise käske nutiseadme rakenduses puhverdama ja saatma need edasi siis, kui sünkroniseerimine on lõppenud. Kuna sünkroniseerimine võib suure valgustussüsteemi korral võtta sekundeid, siis selle aja jooksul tagantjärele saadetavad käsud tekitavad efekti, et süsteem reageerib kasutaja käskudele viivitusega, kuna valgus muutub tagantjärele, kuigi kasutaja on valguse muutmise lõpetanud. Seega tuleks puhverdada ja saata kõige viimane käsk. Sellisel juhul tekib kasutajal tunne, et süsteem ei reageeri kõikidele käskudele, kuid see on parem variant kui muudatused tulevad tagantjärele.



Nutiseadme rakendus peaks koheselt reageerima saadud vastusele, mitte ootama alati seni, kuni maksimaalne vastamise aeg täis saab, kuna see teeb protsessi liiga aeglaseks. Väiksema süsteemi ja muude asjaolude koosmõjul võib vastamine võtta aega ainult kümnendiku maksimaalsest ajast või veelgi vähem.

### **8.1 Kasutajaliidese blokeerimine juhul, kui valguse muutmise käske ei saa saata**

Kui sünkroniseerimine või mõnele teisele käsule vastuse ootamine on pooleli, siis kõige lihtsam viis valguse muutmise käskude saatmise takistamiseks oleks nutiseadme rakenduses kasutajaliidese blokeerimine. Kasutaja ei saaks juhtnuppe kasutades valgust muuta ja seega ei saadetak ka valguse muutmise käske. Selle lähenemise puuduseks on see, et nutiseadme rakendusest saab korraga juhtida mitut valgustusüsteemi. Kõikide valgustusüsteemide juhtimise blokeerimine selle tõttu, et üks süsteem ei ole vastust saatnud, on liiga agressiivne sekkumine. Nii võib üks süsteem halvata teiste süsteemide juhtimise.

Kasutajaliidese blokeerimine problemaatilise süsteemi puhul on vajalik, kuid ka siin tuleb probleemi lahenedes kasutajaliidese kasutamine taastada. Mõnel juhul ei peakski kasutajaliidest blokeerima vaid valgus muutmise viimane käsk tuleks puhverdada ja saata siis, kui probleem laheneb või vastus on kätte saadud. Sellisel juhul tunnetaks kasutaja, et süsteem ei reageerinud käskudele ainult lühikese aja vältel ning viimane kasutaja poolt antud käsk saaks nii alati valgustusüsteemile edastatud. Kui kasutajaliides blokeerida, siis kasutaja näeb, et juhtnupud ei allu tema kontrollile või kasutaja peab uuesti juhtnupust kinni võtma. See on egamugav ning tekitab negatiivse kasutajakogemuse.

### **8.2 Rakendusele oma olekute süsteemi loomine**

Selleks, et hallata vastuste ootamist ning puhverdamist tuleks nutiseadme rakenduses luua olekute süsteem. Saadetud käsud ja vastuvõetud andmed peaksid muutma valgustusüsteemiga suhtlemise olekut. Oleku muutmise käigus saaks käivitada vajalikke viiteid tekitavaid taimereid ning olekust lähtuvalt puhverdada käske ning blokeerida vajadusel kasutajaliidese juhtnuppe jms.

Olekute süsteemi teeb keeruliseks asjaolu, et olekuid on palju nende haldamine arusaadaval viisil muutub kiiresti väga raskeks. Kui olekuid on palju ja oleku muutusele peavad järgnema olekust lähtuvad sammud võiks kasutada olekumasina genereerimise vahendeid.

UML vahendeid kasutades dokumenteeritud olekudiagrammide realiseerimine ilma koodi genereerimata on töömahukas ettevõtmine. Kui hiljem tekib vajadus olekudiagramme muuta, siis muudatuste realiseerimine nõuab uuesti olemasoleva lahendusega tutvumist ning iga tehtud muudatus võib põhjustada vigu programmi töös, kuna terviku haldamine on visuaalselt väga raske, kuna klasse ja koodi võib olla väga palju [11],[12] peatükk 2.7).

Objektorienteeritud lähenemise puhul saab igast olekust klass ja üleminekutest klassi funktsioonid ([13] peatükk 5.1). Kui olekuid on palju, siis tuleb käsitsi luua palju klasse.

Sellest tulenevalt tuleks kasutada olekumasina genereerimise vahendeid.

### **8.3 Lõpliku olekumasina genereerimine**

Lõpliku olekumasina (finite-state machine (FSM)) genereerimine on mõttekas, sest see võimaldab UML olekudiagrammide alusel koodi genereerida (IBM Rational Rhapsody [11]) või olekudiagramm kirjeldatakse ühes failis (SMC - The State Machine Compiler [14],[15]), mille alusel genereeritakse kood.

Selline lahendus võimaldab hoida olekutest ja üleminekutest terviklikku ülevaadet. Olekute lisandumisel või olemasolevate muutmisel ei ole vaja kõiki klasse käsitsi muuta vaid need saab automaatselt genereerida.

### **8.4 Nutiseadme rakendusest käskude saatmise analüüs**

Kuna nutiseadme rakendusest saadetud käsud jõuavad lõpuks jadasiinile, kus ei saa korraga andmeid saata ja vastu võtta, siis nutiseadme rakendus peab suutma hallata suhtlust VSTPga nii, et ei oleks vaja täiendavat suhtlust VSTPga, mis suurendaks andmete kadu jadasiinil.

Nutiseadme rakenduses tuleks võtta kasutusele lõplik olekutemasin, mille abil saab reguleerida suhtlust VSTPga. Lõplik olekutemasin tuleks genereerida olekudiagrammide alusel.

Lõpliku olekumasina korrektne realiseerimine teeb valguse muutmise sujuvamaks, kuna vähendab pause, mis võivad tekkida sünkroniseerimise või veateadete saatmise tõttu. Samuti lahendab lõplik olekumasin käskude saatmistiheduse haldamise ja võimaldab rakendada käskude analüüsi enne saatmist, et vajadusel jätta vahele käske, mille tulemusena tekkiv valguse muutus ei ole kasutajale nähtav andes nii aega VSTPdele sujuva ülemineku lõpuleviimiseks.

## Kokkuvõte

Valgustusüsteemi prototüübi ja nutiseadme rakenduse testimisel ilmnisid probleemid Wi-Fi ühenduse stabiilsuse ja rakendusest valgustusüsteemile käskude edastamisega, mille tulemusena ei olnud valguse muutmine sujuv.

Kuna tegemist on süsteemiga, mis koosneb riistvarast ja tarkvarast, mis on ainult osaliselt muudetavad, tuli analüüsida kogu süsteemi, et leida kõige problemaatilisemad kohad, mille muutmine annab kõige suurema efekti. Kui tegemist on seadmete või tarkvaraga, mida ei ole võimalik muuta, siis on vaja leida parim viis, kuidas teha seadistamise või kasutamise valguse muutmine võimalikult sujuvaks.

Süsteemi analüüs andis ülevaate süsteemi probleemidest ja kaardistas olemasoleva lahenduse positiivsed ning negatiivsed külgi ning võimalikke alternatiive.

Kõige suuremaks probleemiks osutus käsupakettide kadu valgustusüsteemi Wi-Fi moodulis. Selle tõttu ei olnud valguse muutmine sujuv, vaid astmeline ja kaootiline, kuna paketikadu oli juhuslik. Wi-Fi moodulis tuleb kasutusele võtta voo reguleerimine, et tagada, et kõik TCP/IP võrgu kaudu kohale jõudnud käsud jõuksid jadasiinile. Voo reguleerimise rakendamine parandab valguse sujuvat muutmist oluliselt.

Selleks et valgus muutuks sujuvalt, on kindlasti vaja rakendada valguse silumist, kuna suure valguse muudatuse korral ei saa nutiseadme rakendusest saata käske nii väikeste vahedega, et siluda üleminekut nii, et see tunduks kasutajale sujuv. Valgustusüsteemis toimuv silumine parandab oluliselt valguse muutmise sujuvust. Samas säilib ka võimalus saata käske ilma silumiseta, mis võimaldab tekitada valgusefekte, kus on vaja valguse järske muutusi.

Wi-Fi infrastruktuuri kasutamise suurim probleem on erinevate võrguseadmete rohkus ja sellest tulenevalt ka erinevate seadistusviiside rohkus. Lisaks sellele, et Wi-Fi sagedused ja võimsused on riikides erinevad, on kõikide ruuterite seadistamine erinev. Kasutajate jaoks, kes ei ole varem Wi-Fi võrgu seadistamise ja probleemide lahendamise kokku puutunud,

võib võrguprobleemide lahendamine olla väga raske. Samadel põhjustel on keeruline ka kasutajatoe pakkumine.

Kuna Wi-Fi infrastruktuurist loobuda ei saa, tuleb kasutajatele soovitada testitud *premium*-klassi ruuterid ja kasutada valgustussüsteemi Wi-Fi moodulis võimalikult häid manussüsteemi mooduleid, mille väljundvõimsus on võimalikult lähedal maksimaalsele lubatud võimsusele.

TCP protokoll valik oli õige valik. Pakettide uuesti saatmine võib tekitada pause valguse muutmisesse, kuid rakenduse üldise toimimise seisukohast on alternatiivide kasutamine ebaotstarbekas.

Rakenduskihis käskude edastamiseks sobib kõige paremini valgustussüsteemi jaoks loodud protokoll, kuna alternatiivid on mõeldud teistsuguse ülesehitusega süsteemi juhtimiseks. Valgustussüsteemi jaoks loodud protokoll võimaldab kõige kiiremat andmete sükroniseerimist ja veateadete saatmist VSTP ja nutiseadme rakenduse vahel.

Valguse sujuvat muutmist saab parandada nutiseadme rakenduses lõpliku olekumasina kasutuselevõtmisega, mille abil saab hallata suhtlust valgustussüsteemiga ilma valgustussüsteemi täiendavalt koormamata. Kui välditakse andmepakettide põrget jadasiinil, on valguse muutmine sujuvam ja veateadetest või sükroniseerimisest tingitud pausid võimalikult lühikesed.

Kui Wi-Fi moodulis on kasutusele võetud voo reguleerimine, VSTP kasutab valguse muutuse silumist ja nutiseadme rakenduses on lõplik olekumasin kasutusele võetud, siis tuleks süsteemi uuesti analüüsida, et hinnata tehtud muudatuste mõju.

Kuna käsu edastamisel tekib põhiline viide Wi-Fi võrgus, tuleks analüüsida võimalusi viite vähendamiseks.

Täiendavalt vajaks uurimist käskude vahele jäetava viite mõju kasutatavuse seisukohalt. Kaaluda võiks, kas jätta käskude vahele pikemad viited ja eelistada sujuvat üleminekut või üritada käske saata võimalikult tihedalt, et muuta valgust võimalikult kiirelt.

## Summary

Lighting system prototype and smartphone application communication testing over Wi-Fi network revealed problems with connection stability and command transfer to lighting system. As a result light change is not smooth.

To find which part of the system has biggest impact to light smoothness whole system needs to be analyzed. System consist of hardware and software. Some parts of the system can be changed others cannot be changed. Analyzing should reveal parts with biggest impact. Changing these parts will be most effective for guaranteeing smooth light changing. If part of the system cannot be changed there might be a way to improve light changing smoothness by hardware selection and configuration.

System analyzing mapped system problems and highlighted solution strengths and weaknesses and viewed possible alternatives.

Biggest impact to smooth light changing comes from lighting system Wi-Fi module because of data loss. Light changing commands that arrived over TCP/IP connection were randomly dropped in Wi-Fi module before they reached serial bus. To fix the data loss Wi-Fi module has to be programmed to use flow control that allows all commands reach the serial bus. Implementing flow control drastically improves light smooth changing.

Another thing that has big impact is light changing smoothing in lighting system power supply. It should be used to get smooth light changes because smartphone application cannot send commands so fast that it could replace the need for additional smoothing in lighting system power supply. It is possible to control light without smoothing when it is needed for special effects when quick light changes are needed.

Light change smoothing in lighting system power supply noticeably improves light smooth changing.

Wi-Fi infrastructure main problem is a lot of devices with different configuration options. Additionally all countries have their own frequency and power output regulations. Users without previous Wi-Fi network management experience struggle with Wi-Fi network setup and troubleshooting. Providing support for unknown devices is also very complicated when user has no experience in the subject.

Only solution is to provide users with recommended premium routers and focus on finding best embedded Wi-Fi modules with maximum allowed radio power output for lighting system Wi-Fi module.

TCP protocol was a right choice. TCP packet retransmissions can produce small pauses to light changing but alternatives are not better.

Best protocol for application layer is lighting system own communication protocol. Alternatives are made for other type of systems and cannot be used. Lighting system own protocol enables fast synchronization and error message sending between lighting system power supply and smartphone application.

Implementing final state machine in smartphone application for communication management would make light changing smoother by minimizing pauses caused by synchronization and error message sending. It also allows to avoid collisions on serial bus allowing all commands to get through.

System needs to be analyzed again after following changes:

- Lighting system Wi-Fi module flow control is implemented
- Lighting system uses light change smoothing
- Final state machine is implemented in smartphone application

Most of the delay between issuing commands and light change is caused by Wi-Fi network. It is one area that should be analyzed in the future.

Current smartphone application leaves delays between packets to provide processing time for lighting system. Smaller delays would make system more responsive and longer delays would be better for smooth light changing. This subject needs more analyzing to decide how long is optimal delay that is best for system usability.



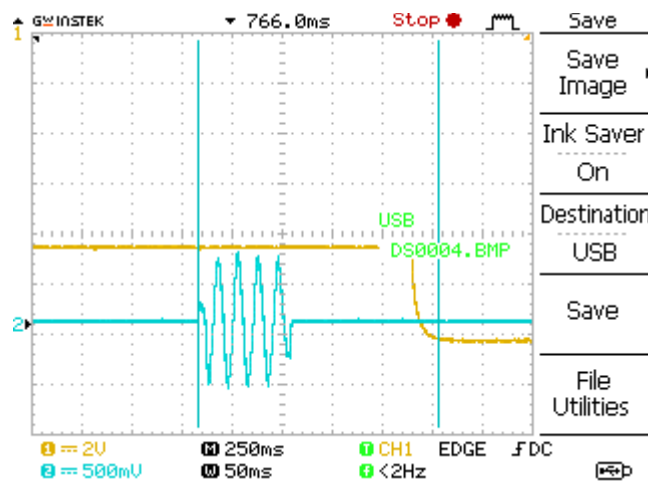
## Kasutatud kirjandus

- [1] Kevin Fall, W. Stevens, TCP/IP Illustrated, Volume 1 Second edition, 2012
- [2] John Nagle, Congestion Control in IP/TCP Internetworks, 1984,  
<https://tools.ietf.org/html/rfc896>
- [3] R. Braden, Requirements for Internet Hosts -- Communication Layers, 1989,  
<https://tools.ietf.org/html/rfc1122>
- [4] Microsoft Support, Design issues - Sending small data segments over TCP with Winsock , 2014, <https://support.microsoft.com/et-ee/kb/214397>
- [5] M. Allman, V. Paxson, E. Blanton, TCP Congestion Control, 2009,  
<https://tools.ietf.org/html/rfc5681#section-3.2>
- [6] J. Postel, User Datagram Protocol, 1980, <https://tools.ietf.org/html/rfc768>
- [7] Lutron, Measured light vs. perceived light, 2000,  
[http://www.lutron.com/TechnicalDocumentLibrary/Measured\\_vs\\_Perceived.pdf](http://www.lutron.com/TechnicalDocumentLibrary/Measured_vs_Perceived.pdf)
- [8] Artistic Licence Holdings Ltd, Specification for the Art-Net 3 Ethernet Communication Protocol, 2015, <http://www.artisticlicence.com/WebSiteMaster/User%20Guides/art-net.pdf>
- [9] Cisco Systems, Inc., Cisco Wireless Control System Configuration Guide, 2006,  
<http://www.cisco.com/c/en/us/td/docs/wireless/wcs/3-2/configuration/guide/wcscfg32/wscod.html>
- [10] Ethan Biery, Thomas Shearer, Roland Ledyard, Dan Perkins, Manny Feris, Controlling LEDs, 2014, [http://www.lutron.com/TechnicalDocumentLibrary/367-2035\\_LED\\_white\\_paper.pdf](http://www.lutron.com/TechnicalDocumentLibrary/367-2035_LED_white_paper.pdf)
- [11] Stefan Rührup, Network Protocol Design and Evaluation , 2009,  
[http://archive.cone.informatik.uni-freiburg.de/teaching/lecture/protocol-design-s09/slides/04-Protocol\\_Specification\\_1.pdf](http://archive.cone.informatik.uni-freiburg.de/teaching/lecture/protocol-design-s09/slides/04-Protocol_Specification_1.pdf)
- [12] Bob Nystrom, Game Programming Patterns, 2014,  
<http://gameprogrammingpatterns.com/>
- [13] Jauhar Ali, Jiro Tanaka, Implementing the Dynamic Behavior Represented as Multiple State Diagrams and Activity Diagrams , 1998, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.7578&rep=rep1&type=pdf>
- [14] Charles W. Rapp, SMC Programmer's Manual, 2009,  
<http://smc.sourceforge.net/SmcManual.htm>
- [15] Raúl Uranga, How to Integrate the Finite State Machine Compiler With Cocos2D, 2012,  
<http://labs.grupow.com/blog/2012/03/27/how-to-integrate-the-finite-state-machine-compiler-with-cocos2d>

## Lisad

### Lisa 1. Viide rakendusest antud käsu ja pinge muutuse vahel

Viide puudetundliku ekraani vajutusest kuni LEDi jalal pinge muutumiseni

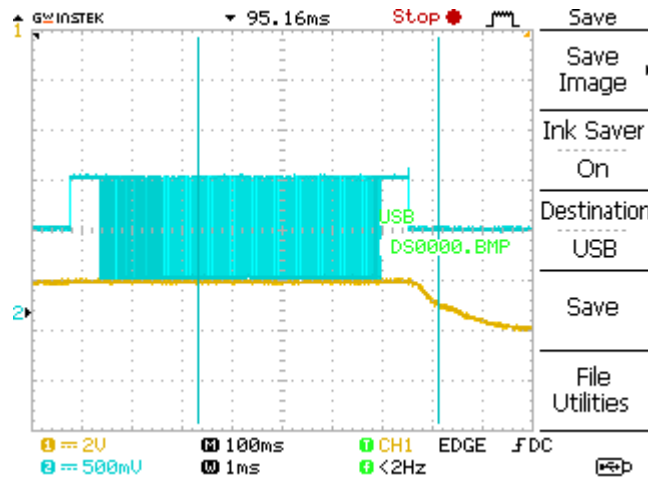


Joonis 10: Viide näputõstmise ja LEDi pinge muutuse vahel

Sinisega märgitud puudutus. Kollasega märgitud pinge LEDi jalal.

Puudutuse lõpust LEDi jalal pinge muutumise alguseni viide 120 ms.

## Lisa 2. Viide jadasiinil info vastuvõtu alguse ja LEDi jalal pinge muutmise lõpu vahel

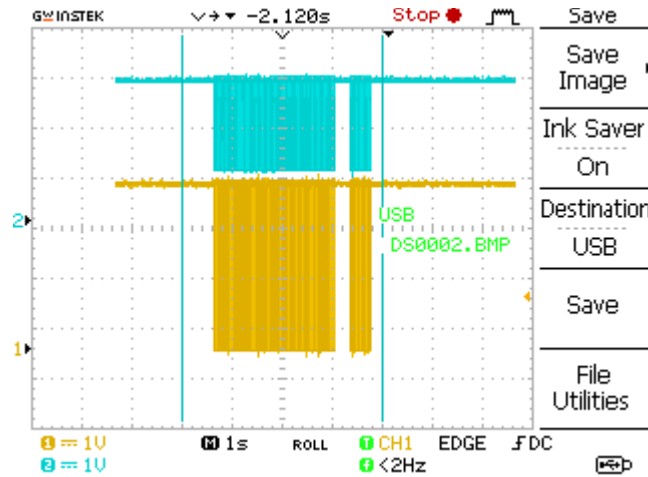


Joonis 11: Viide käsupaketi alguse ja pinge LEDi pinge muutuse vahel

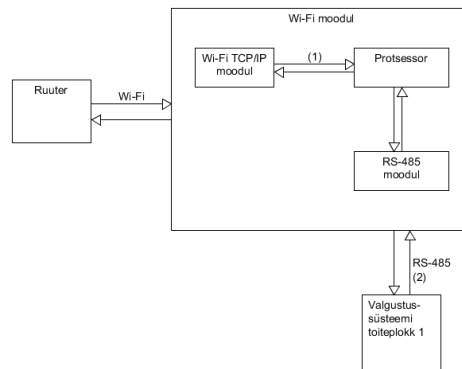
Sinisega märgitud signaal jadasiinil. Kollasega märgitud pinge LEDi jalal.

Viide jadasiinil info vastuvõtu algusest kuni pinge muutumise lõpuni LEDi jalal 8,9 ms.

### Lisa 3. Pakettide edastamine Wi-Fi moodulis



Joonis 12: TCP/IP mooduli tekitatud paus



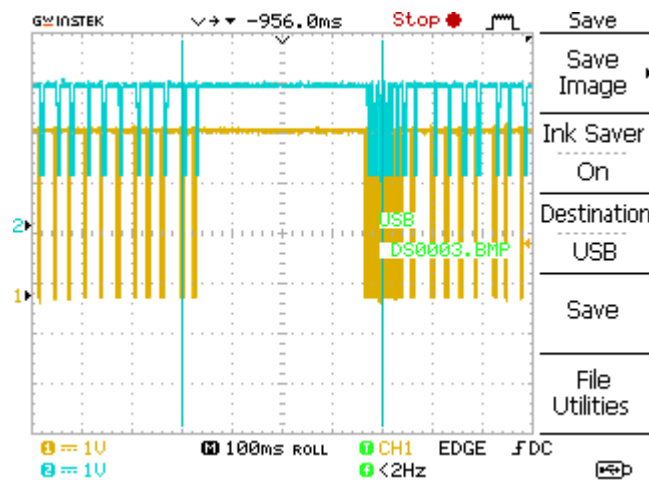
Joonis 13: Valgustussüsteemi Wi-Fi mooduli plokk skeem

Kollane tähistab admevoogu Wi-Fi mooduli sees TCP/IP mooduli ja protsessori vahel (UART (1)).

Sinine tähistab andmevoogu Wi-Fi mooduli väljundi ja esimese valgustussüsteemi toiteploki vahel (RS-485 (2)).

Rakendus saatis Wi-Fi ühenduse kaudu valgustussüsteemi Wi-Fi moodulile 128 baidi suuruseid pakette. Iga korraga 100 paketti 25 ms vahega. Pildidel vaatleme ühte sellist 100 pakettist koosnevat üksust.

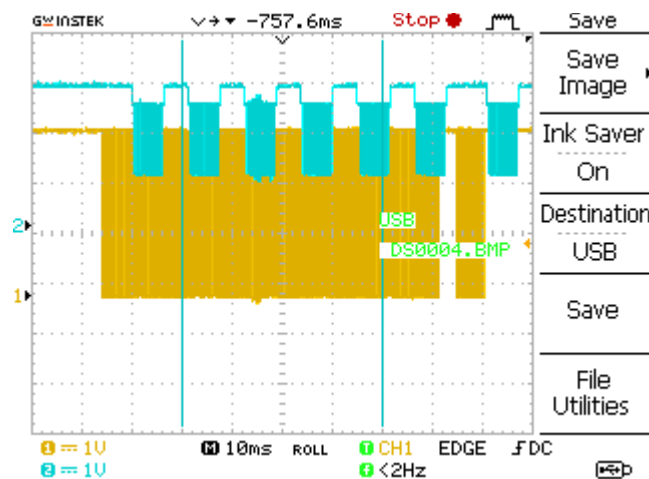
Pildilt on näha, et TCP/IP moodul teeb andmete edastamisel pausi ja siis andmete edastamine jätkub.



Joonis 14: Viited andmepakettide vahel enne ja pärast pausi

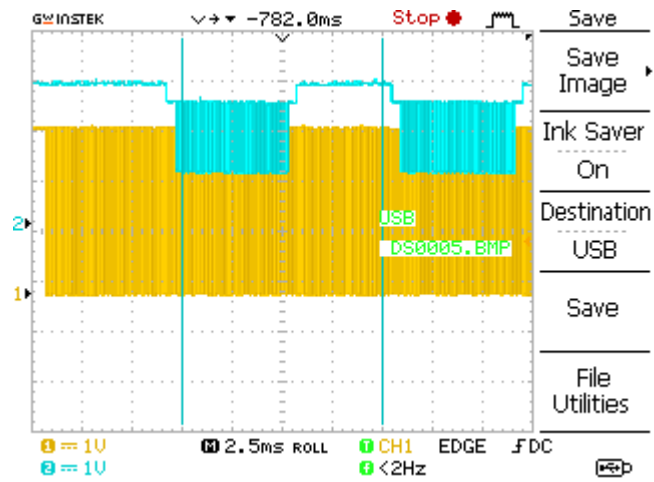
Lähemal vaatlusel on näha, et enne pausi on TCP/IP moodulist väljuvate andmepakettide vahel on ligikaudu 25ms viide ehk nii, nagu pakette rakendusest saadeti. RS-485 jadasiinile jõuavad paketid samuti ligikaudu 25 ms vahedega ja järgmise andmepaketi vastuvõtmiseni on turvaline ajapuhver.

Pärast pausi pilt muutub, kuna TCP/IP moodul edastab andmeid järjest.



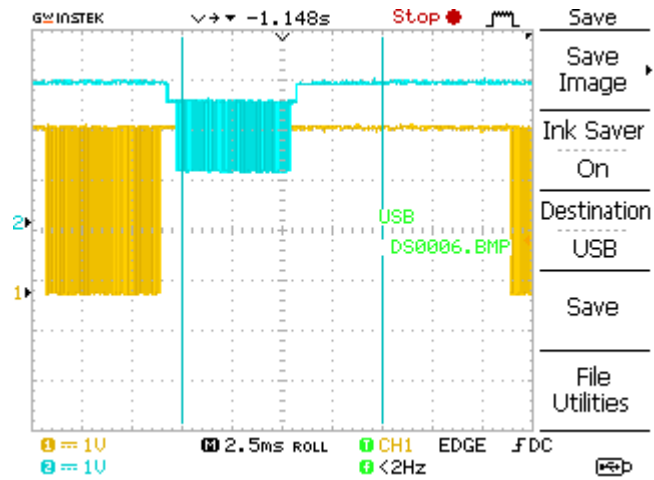
Joonis 15: TCP/IP moodul saadab andmeid ilma vahedeta

Kuna protsessor paketi töötlemise ajal uut paketti vastu ei võta, siis tekib andmekadu ehk osa pakettidest läheb kaotsi. Kui TCP/IP mooduli edastatavas andmevoos tekivad pausid, siis normaalne töö taastub.



Joonis 16: Andmekadu pakettide edastamisel

Andmekadu pakettide edastamisel.



Joonis 17: Normaalne andmete edastamine

Normaalne andmete edastamine. Protsessoril on piisavalt aega andmete töötlemiseks.

## Lisa 4. iOS rakenduses TCP *keepalive* seadistamine

Seadistus

vastavalt

operatsioonisüsteemi

juhendile

([https://developer.apple.com/library/ios/documentation/System/Conceptual/ManPages\\_iPhoneOS/man2/setsockopt.2.html](https://developer.apple.com/library/ios/documentation/System/Conceptual/ManPages_iPhoneOS/man2/setsockopt.2.html))

```
NSData *data = (NSData *)[theStream valueForKey:(__bridge NSString *)kCFStreamPropertySocketNativeHandle];
if(data)
{
    CFSocketNativeHandle socket_handle = *(CFSocketNativeHandle *)[data bytes];

    //SO_KEEPALIVE option to activate
    int option = 1;
    //TCP_NODELAY option to activate
    int option2 = 1;
    //Idle time used when SO_KEEPALIVE is enabled. Sets how long connection must be idle before keepalive is sent
    int keepalivedle = 10;
    //Interval between keepalives when there is no reply. Not same as idle time
    int keepaliveIntvl = 2;
    //Number of keepalives before close (including first keepalive packet)
    int keepaliveCount = 4;
    //Time after which tcp packet retransmissions will be stopped and the connection will be dropped.Stream is closed
    int retransmissionTimeout = 5;

    if (setsockopt(socket_handle, SOL_SOCKET, SO_KEEPALIVE, &option, sizeof(int)) == -1)
    {
        NSLog(@"setsockopt SO_KEEPALIVE failed: %s %@", strerror(errno),self);
    }

    if (setsockopt(socket_handle, IPPROTO_TCP, TCP_KEEPCNT, &keepaliveCount, sizeof(int)) == -1)
    {
        NSLog(@"setsockopt TCP_KEEPCNT failed: %s %@", strerror(errno),self);
    }

    if (setsockopt(socket_handle, IPPROTO_TCP, TCP_KEEPALIVE, &keepalivedle, sizeof(int)) == -1)
    {
        NSLog(@"setsockopt TCP_KEEPALIVE failed: %s %@", strerror(errno),self);
    }

    if (setsockopt(socket_handle, IPPROTO_TCP, TCP_KEEPINTVL, &keepaliveIntvl, sizeof(int)) == -1)
    {
        NSLog(@"setsockopt TCP_KEEPINTVL failed: %s %@", strerror(errno),self);
    }

    if (setsockopt(socket_handle, IPPROTO_TCP, TCP_RXT_CONNDROPTIME, &retransmissionTimeout, sizeof(int)) == -1)
    {
        NSLog(@"setsockopt TCP_RXT_CONNDROPTIME failed: %s %@", strerror(errno),self);
    }

    if (setsockopt(socket_handle, IPPROTO_TCP, TCP_NODELAY, &option2, sizeof(int)) == -1)
    {
        NSLog(@"setsockopt TCP_NODELAY failed: %s %@", strerror(errno),self);
    }
}
```