

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Informaatikainstituut

Infosüsteemide õppetool

**Infrapunasiinaalidega juhitavate seadmete
juhtimine nutiseadmega**

Magistritöö

Üliõpilane:	Ilmar Telga
Üliõpilaskood:	122148IAPMM
Juhendaja:	Enn Õunapuu

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)

Annotatsioon

Infrapunasiinaale on koduelektroonika seadmete juhtimiseks kasutatud aastakümneid. Täna on paljud koduelektroonika seadmed juhitavad infrapunasiinallidega. Televisoorid, digiboksid, muusikakeskused, videoprojektorid, konditsioneerid, koduautomaatika süsteemid on mõned näited. Iga sellise seadme jaoks on tavaliselt vaja eraldi juhtimispuhli infrapunasaatjaga. Kui aga seadmeid ja nende kasutajaid on palju, siis võib nende kasutamine olla tülikas, näiteks võivad puldid mõnikord kaduma kippuda või puldi nupud katki minna.

Sellele probleemile on üks võimalik lahendus. Uuemad nutiseadmed on varustatud infrapunasaatjaga. Sellistele nutiseadmetele on võimalik luua mobiilirakendus, mille abil saab juhtida paljusid seadmeid, mis on juhitavad infrapunasiinallidega. Seega pole vajadust alati erinevaid seadmete juhtimispuhli kasutada. Ühe ja sama nutiseadmega on võimalik juhtida enamus infrapunasiinallidega juhitavaid seadmeid.

Käesoleva **töö eesmärgiks** on uurida kuidas infrapunasaatjaga varustatud nutiseadme ja mobiilirakenduse abil juhtida mistahes infrapunasiinallidega juhitavaid koduelektroonika seadmeid, näiteks televisoreid või muusikakeskusi, ning samuti vähemlevinud seadmeid, näiteks digibokse, mida ei õnnestu alati universaalpuldiga juhtida.

Töös uuriti esmalt kuidas luua nutiseadmele mobiilirakendus, millega saab nutiseadet kasutada infrapunasiinallidega juhitavate seadmete juhtimispuhli. Teiseks uuriti kuidas mobiilirakenduse jaoks füüsilisel tasemel infrapunasiinallidele salvestada infrapunasiinallidega juhitavate seadmete juhtimispuhli. Lõpuks loodi seadmete juhtimiseks nutiseadme mobiilirakendus kasutades salvestatud infrapunasiinallide.

Töö olulisemad tulemused on riistvaraline lahendus füüsilisel tasemel infrapunasiinallide salvestamiseks infrapunasiinallidega juhitavate seadmete juhtimispuhli ja nutiseadme mobiilirakendus salvestatud infrapunasiinallide saatmiseks, et juhtida infrapunasiinallidega juhitavaid seadmeid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 5 peatükki, 4 joonist, 2 tabelit.

Abstract

Consumer infrared devices have been in use for decades. Today still many consumer electronics devices can be remote controlled by infrared signals. Television sets, digital TV tuners, home audio systems, video projectors, air conditioners, home automation system devices are some examples. Usually for each device, there is an infrared remote controller. However, if there are many infrared remote controlled devices then using many different remote controllers are necessary to use all these devices. This can be often troublesome. For example, the remote controllers can get lost or the buttons damaged so that the remote controllers become unusable.

There is one possible solution to this problem. New smartphones are equipped with an infrared emitter. It is also possible to create mobile application software to use a smartphone as an infrared remote controller. This makes it possible to use one smartphone to remote control most of the infrared remote controlled devices.

The aim of this thesis is to study the possibilities and the means to control any infrared remote controlled consumer electronics device by using a smartphone equipped with an infrared emitter and a mobile application. For example, how to control a television set or home audio system and also less common devices like digital tuners, which cannot be always fully controlled by a universal remote controller.

The main problems dealt with in this thesis were studying how to create a mobile application to use a smartphone to control infrared remote controlled devices. Secondly, how to save infrared signals on the physical layer from infrared remote controllers and use the saved signals to control infrared remote controlled devices by the use of a smartphone and a mobile application.

The main results of this thesis are the hardware solution to capture infrared signals on the physical layer from infrared remote controllers and the smartphone application to control infrared remote controlled devices by using the captured infrared signals.

The thesis is in Estonian and contains 37 pages of text, 5 chapters, 4 figures, 2 tables.

Lühendite ja mõistete sõnastik

USB

Universal Serial Bus

Universaalne järjestiksiin arvuti välisseadmete jaoks

PIC

Peripheral Interface Controller

RISC tüüpi käsustikuga mikrokontroller, mida arendab firma Microchip Technology

TSOP

Thin Small Outline Package

Pindmontaažiks mõeldud väga õhuke neljakandiline plastmassist kiibikorpus, mille kummaski lühemas servas on rida väljaviike.

API

Application Programming Interface

Arvuti operatsioonisüsteemiga või rakendusprogrammiga määratud reeglistik, mille alusel rakendusprogramm kasutab operatsioonisüsteemi või teise rakendusprogrammi teenuseid.

LED

Light-emitting Diode

Valgusdiod ehk pooljuhtdiod, mis läbiva elektrivoolu toimetel kiirgab nähtavat valgust, samuti optilist kiirgust spektri infrapunases või ultravioletses osas.

Jooniste nimekiri

Joonis 1. Mobiilirakenduse kasutajaliides	13
Joonis 2. Tüüpiline infrapunasiin角度 füüsilisel tasemel	14
Joonis 3. Arduino kontroll器 ühendatud infrapunavastuvõõtjaga.....	17
Joonis 4. AnyMote Home.....	22

Tabelite nimekiri

Tabel 1. Infrapunasiinialide salvestamise võimaluste võrdlus.....	16
Tabel 2. Olemasolevate lahenduste võrdlus.	24

Sisukord

1. Sissejuhatus	9
1.1 Taust ja probleem	9
1.2 Ülesande püstitus	10
1.3 Metoodika	10
1.4 Ülevaade tööst	11
2. Nutiseadmega infrapunasiinjalide saatmine	12
2.1 Android API infrapunasiinjalide saatmiseks	12
2.2 Mobiilirakendus	13
3. Infrapunasiinjalide salvestamine	14
3.1 Infrapunasiinjalid	14
3.2 Võimalused infrapunasiinjalide salvestamiseks	15
3.3 Infrapunasiinjalide salvestamise võimaluste võrdlus	16
3.4 Infrapunasiinjalide salvestamine Arduino kontrolleriiga	17
4. Lahenduse kasutamise võimalused	20
5. Olemasolevad lahendused infrapunasiinjalidega juhitavate seadmete juhtimiseks	21
5.1 Panja	21
5.2 AnyMote Home	22
5.3 RedEye	23
5.4 Olemasolevate lahenduste võrdlus	24
6. Kokkuvõte	26
Summary	28
Kasutatud kirjandus	29
Lisa 1	31
Lisa 2	33

1. Sissejuhatus

1.1 Taust ja probleem

Infrapunasiinjalidega juhitavad seadmed on kodutehnikas kasutusel olnud juba mitukümmend aastat [1]. Seniajani on paljud seadmed juhitavad infrapunasiinjalide abil, näiteks televiisorid, digiboksid, videoprojektorid, muusikakeskused, DVD mängijad, kliimaseadmed ja „targa maja“ seadmed.

Iga sellise seadme jaoks on eraldi juhtimispuht infrapunasatjaga. Kui aga seadmeid on palju, siis võib nende kasutamine olla tülikas, näiteks võivad puldid mõnikord kaduma minna või nende otsimine võtab palju aega. Kui puliti kasutab palju inimesi, võib puht muutuda mõne aastaga kasutamiskõlbmatuks.

Sellele probleemile on üks võimalik lahendus. Uuemad nutiseadmed sisaldavad muuhulgas ka infrapunasatjat. Selliste nutiseadmetega tuleb tavaliselt tootja poolt kaasa ka mobiilirakendus, mille abil saab juhtida paljusid seadmeid, millel on infrapunasiinjalide vastuvõtja. Seega pole vajadust alati erinevaid seadmete juhtimispuhte kasutada. Ühe ja sama nutiseadmega on võimalik juhtida paljusid infrapunasiinjalidega juhitavaid seadmeid.

Paljude seadmetootjate infrapunasiinjalide kirjeldused ehk protokollid on avalikult internetis kättesaadavad. Samuti on võimalik nutiseadmetele alla laadida erinevaid mobiilirakendusi infrapunasiinjalidega juhitavate seadmete juhtimiseks. Lisaks on võimalik infrapunasiinjalide kirjeldusi kasutades luua nutiseadmele näiteks isiklikuks otstarbeks mobiilirakendus, mille abil saab juhtida infrapunasiinjalidega juhitavaid seadmeid.

Kuid mõned seadmetootjad ei avalda infrapunasiinjalide kirjeldusi, mida saab kasutada infrapunasiinjalidega juhitavate seadmete juhtimiseks. Samuti ei saa olemasolevate mobiilirakenduste abil mõnesid seadmeid juhtida, näiteks seadmeid, mille infrapunasiinjalide protokoll pole teada. Lisaks puudub nutiseadmetel infrapunasiinjalide vastuvõtja, millega saaks signaale juhtimispuhtidelt salvestada, et kasutada salvestatud signaale infrapunasiinjalidega juhitavate seadmete juhtimiseks.

Töö on kasulik infrapunasiinuaalidega juhitavate seadmete kasutajatele, kes soovivad juhtida nutiseadme abil mistahes seadmeid, kaasa arvatud ka neid seadmeid, mille juhtimiseks puuduvad tavakasutajale võimalused, näiteks seadme infrapunasiinuaalide protokoll pole teada või selle juhtimine universaalpuldiga pole täielikult võimalik.

Töö praktiline ja kirjalik osa valmib kodus. Mõningaid seadmeid katsetatakse töö kontoril.

Töö teema tuleneb autori hobist mobiilirakenduste ja elektroonikaga. Töö valmis ajavahemikul juuni 2014 kuni mai 2015.

1.2 Ülesande püstitus

Käesoleval tööil on kaks eesmärki:

- 1) uurida kuidas luua infrapunasaatjaga varustatud nutiseadmele mobiilirakendus infrapunasiinuaalide saatmiseks, et juhtida infrapunasiinuaalidega juhitavaid seadmeid;
- 2) uurida kuidas salvestada infrapunasiinuaale füüsilisel tasemel infrapunasiinuaalidega juhitavate seadmete juhtimispuultidelt.

Töös pakutavad hüpoteesid:

- 1) saab kasutada uuemaid nutitelefone, mis sisaldavad infrapunasiinuaalide saatjat
- 2) saab juhtida kõiki infrapunasiinuaalidega juhitavaid koduelektroonika seadmeid
- 3) selline lahendus on odav ja kättesaadav paljudele inimestele.

1.3 Metoodika

Infrapunasiinuaalide salvestamiseks tutvutakse infrapunasiinuaalidega juhitavate seadmete juhtimispuultide tööõhimõttega, uuritakse erinevaid võimalusi siinuaalide salvestamiseks ja kasutades internetis teadaolevaid õpetusi tehakse katseid puultidelt infrapunasiinuaalide vastuvõtmiseks ja salvestamiseks.

Seejärel luuakse salvestatud siinuaale kasutades mobiilirakendus nutiseadmele, millel on infrapunasaatja. Mobiilirakenduse katsetamiseks proovitakse juhtida erinevaid infrapunasiinuaalidega juhitavaid seadmeid: televiisoreid, digibokse, muusikakeskusi, videoprojektoreid ja DVD-mängijaid. Katsete käigus selgub, kas nutiseadme ja

mobiilirakenduse abil saab juhtida infrapunasiinnaalidega juhitavaid seadmeid salvestatud infrapunasiinnaale välja saates.

Seatud nõuded riistvaralise lahendusele infrapunasiinnaalide salvestamiseks:

- 1) infrapunasiinnaalide salvestamine peab olema võimalikult lihtne;
- 2) infrapunasiinnaale peab saama salvestada füüsilisel tasemel;
- 3) vahend siinnaalide salvestamiseks peab olema võimalikult odav.

1.4 Ülevaade tööst

Töö esimeses osas uuritakse, kuidas luua infrapunasaaaja varustatud nutiseadmele mobiilirakendus infrapunasiinnaalide saatmiseks, et juhtida infrapunasiinnaalidega juhitavaid seadmeid.

Töö teises osas uuritakse erinevaid riistvaralisi võimalusi, millega saab infrapunasiinnaalidega juhitavate seadmete juhtimispultidelt infrapunasiinnaale füüsilisel tasemel vastu võtta ja salvestada. Võimaluste hulgast valitakse välja üks, mis kõige paremini vastab riistvaralise lahendusele püstitatud nõuetele.

Kolmandas osas käsitletakse muid olemasolevaid lahendusi erinevate infrapunasiinnaalidega juhitavate seadmete juhtimiseks ja võrreldakse neid.

2. Nutiseadmega infrapunasiisnaalide saatmine

2.1 Android API infrapunasiisnaalide saatmiseks

Androidi operatsioonisüsteem on populaarne nutiseadmete kasutajate hulgas. Samuti on sellele platvormile lihtne olemasolevate arendusvahenditega luua mobiilirakendusi. Seepärast valiti käesolevas töös mobiilirakenduse loomisel platvormiks Androidi operatsioonisüsteem.

Mobiilirakendusega saab nutiseadet, millel on infrapunasaatja, programmeerida erinevate infrapunasiisnaalide saatmiseks [2]. Sel viisil on võimalik nutiseadmega juhtida erinevaid infrapunasiisnaalidega juhitavaid seadmeid. Alates Androidi API level 19-st on võimalik kasutada nutiseadme infrapunasaatjat erinevate infrapunasiisnaalide saatmiseks. Selleks tuleb kasutada klassi `android.hardware.ConsumerIrManager`. Selle klassi objekti on võimalik luua kasutades `android.content.Context` klassi meetodit `getSystemService(String name)` kasutades argumenti `Context.CONSUMER_IR_SERVICE`.

`ConsumerIrManager` klassil on olemas meetod ehk funktsioon `transmit(int carrierFrequency, int[] pattern)`, millega saab nutiseadme infrapunasaatjat kasutada erinevate infrapunasiisnaalide saatmiseks, et juhtida erinevaid infrapunasiisnaalidega juhitavaid seadmeid. Esimeseks meetodi argumentiks on infrapunasiisnaali kandjasagedus hertsides. Teiseks argumentiks on massiiv infrapunasiisnaali impulsside ja nendevaheliste pauside kestustest vaheldumisi. Tegemist on signaali füüsilisel tasemel kirjeldusega ja protokolle pole vaja teada. Järgnevalt näide meetodi kasutamisest:

```
ConsumerIrManager irManager;  
  
irManager = (ConsumerIrManager) getSystemService(CONSUMER_IR_SERVICE);  
  
irManager.transmit(38000, new int[] {  
342,166,22,20,23,19,23,18,23,19,23,19,23,18,23,61,21,22,22,62,23,61,23,60,23,61,23,  
,60,23,61,22,21,22,62,23,19,23,18,23,61,23,61,23,19,22,19,23,19,21,21,22,61,23,60,  
23,18,22,20,22,62,23,60,23,61,23,61,23,1509,341,82 });
```

Sellisel kujul infrapunasiisnaalide salvestamiseks tuleb luua riistvaraline lahendus, mille abil salvestada füüsilisel tasemel infrapunasiisnaale seadmete juhtimispultidelt.

Uurimisel selgus, et kõik infrapunasaatjaga nutiseadmed, näiteks Sony Xperia ZL, ei sobi mistahes infrapunasiinjalide saatmiseks. Sony IR Remote API võimaldab saata ainult kindlaid signaale, mis on Sony poolt määratud [3]. Seepärast kasutati töös nutiseadet Samsung Galaxy S4 GT-I9505. Sellega õnnestus edukalt juhtida infrapunasiinjalidega juhitavaid seadmeid.

2.2 Mobiilirakendus

Käesoleva töö käigus loodi nutiseadme jaoks lihtsa kasutajaliidesega mobiilirakendus, millega saab juhtida erinevaid infrapunasiinjalidega juhitavaid seadmeid. Selle tulemusena saab nutiseadet kasutada juhtimispuhdina erinevate seadmete juhtimiseks. Näiteks saab nutiseadmega juhtida televiisorit, digiboksi, muusikakeskust, DVD mängijat või videoprojektorit.



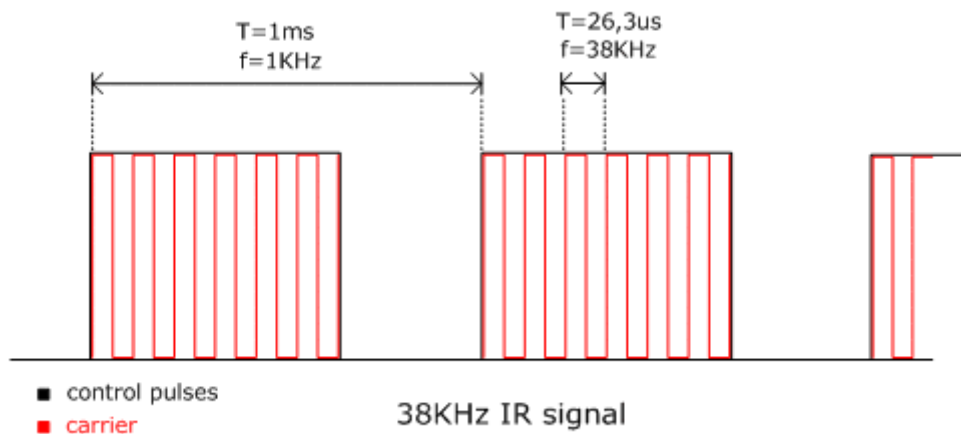
Joonis 1. Mobiilirakenduse kasutajaliides

Mobiilirakenduse kasutajaliidesel on olemas seadme sisse- ja väljalülitamise nupp, nupud numbritega 0 kuni 9 erinevate numbrite valimiseks, nupp ühekohaliste või kahekohaliste numbrite valimiseks, nupud heli valjuse reguleerimiseks, kanalite ülesse ja alla vahetamiseks, kanalite häälestamiseks, heleduse ja kontrasti muutmiseks, teleteksti vaatamiseks ning lisanupud erinevate funktsioonide jaoks. Vajadusel saab mobiilirakenduse kasutajaliidesele lisada nuppe mistahes funktsioonide jaoks, mis on seadmespetsiifilised ja pole universaalpuhdidel saadaval. Mobiilirakenduse lähtekood on toodud töö lisa (vt Lisa 2).

3. Infrapunasiinialide salvestamine

3.1 Infrapunasiinialid

Tutvume alustuseks infrapunasiinialidega juhitavate seadmete juhtimispuultide t oop ohim ottega. Selleks, et juhtida infrapunasiinialidega juhitavat seadet (n aiteks televiisorit), tuleb seadme infrapunavastuv otjasse saata samu siiniale, mida saadab seadme juhtimispuult. Puultid kasutavad enamasti umbes 38 kHz kandjasagedusega siiniali, mille  hte impulssi saadetakse 0,4 – 50 ms jooksul [4]. Seega saab  he ja sama siiniali f uusilisel tasemel m aarata impulsside ja nende vaheliste pauside kestustega.



Joonis 2. T uupiline infrapunasiinial f uusilisel tasemel

Allikas: <http://www.electronics-lab.com/projects/sensors/004/>

On mitu p ohjust, miks kasutatakse infrapuna-diodides k orke sagedusega kandjasignaali [4]. Esiteks v oimaldab see diodil jahtuda, sest infrapuna-diodid v oivad t ootavad elektrivooluga kuni 1 amper ja kui diod t ootaks pidevalt, siis see kuumeneks  le. Teiseks v oimaldab k orke sagedusega kandjasignaali kasutamine v alja filtreerida k orvalised infrapunakiirguse allikad, kuna infrapunasiinialide vastuv otjad reageerivad ainult teatud sagedusega kiirgusele.

Tavaliselt kasutavad erinevad seadmetootjad infrapunasiinialides erinevaid protokolle. K aesolevas t oos ei k asitleta siinialide protokolle, vaid ainult siiniale f uusilisel tasemel. Seega langevad  ra paljud probleemid, mis tulenevad erinevatest protokollidest.

3.2 Võimalused infrapunasiinialide salvestamiseks

Infrapunasiinialide salvestamiseks on vaja fotoelementi infrapunasiinialide vastuvõtmiseks ja mikrokontrollerit siinialide salvestamiseks. Internetis leidub erinevaid riistvaraliste lahenduste loomise juhendeid infrapunasiinialide salvestamiseks. Keerulisemad neist eeldavad häid teadmisi ja kogemusi elektroonika valdkonnas, kuid leidub ka lihtsamaid lahendusi, mille üles seadmine võtab aega mõne tunni.

Üks keerulisem võimalus riistvaralise lahenduse kokkupanekuks on kasutada erinevaid diode, PIC mikrokontrollerit, kondensaatorit ja DE9 ühendust arvutiga ühendamiseks [5]. See lahendus eeldab põhjalikemaid kogemusi ja oskusi elektroonikas ning võtab rohkem aega.

Seadmete juhtimist infrapunasiinialide abil kasutatakse muuhulgas ka robotikas. Internetis kirjeldatakse BrainLink nimelist seadet, millega on võimalik kopeerida infrapunasiiniale ja juhtida erinevaid infrapunasiinialidega juhitavaid ümberehitatud seadmeid (muuhulgas ka roboteid) kohandatud juhtimispultide abil [6].

Kasutatud on ka Atmel STK500 kontrollerit, mis on varustatud AVR ATmega8 mikroprotsessoriga ja TSOP1738 fotoelemendiga [7]. Andmete lugemiseks personaalarvutisse kasutati RS232 liidest. Infrapunasiinialide mõõtmiseks kasutati assemblerkeeles loodud tarkvara. Tarkvara jälgib fotoelemendilt tuleva sisendi olekut iga teatud ajavahemiku tagant ja registreerib sisendi oleku muutusi. Sel viisil mõõdetakse siiniali impulsside ja nende vaheliste pauside kestusi infrapunasiinialis.

Suvel 2014 tuli müügile IR Widget nimeline seade infrapunasiinialide salvestamiseks infrapunasiinialidega juhitavate seadmete juhtimispultidelt [8]. Seadmel on olemas fotoelement infrapunasiinialide vastuvõtmiseks, mikroprotsessorid siinialide mõõtmiseks ja analüüsimiseks ning miniUSB liides personaalarvutiga (näiteks sülearvuti) ühendamiseks. Infrapunasiinialidega juhitava seadme juhtimispuldilt infrapunasiiniali salvestamiseks tuleb juhtimispuldi infrapunasaatja suunata IR Widget küljes olevale fotoelemendile.

Personaalarvutisse paigaldatava IRScope nimelise tarkvara abil on võimalik tuvastada infrapunasiiniali kandjasagedus, siiniali kirjeldus füüsilisel tasemel (impulsside ja nende vaheliste pauside kestused mikrosekundites) ja protokoll, kui tegemist on sellise protokolliga, mida tarkvara suudab tuvastada. Kui tarkvara ei suuda tuvastada protokoll, siis näidatakse ainult siiniali kandjasagedust ja siiniali füüsilist kirjeldust. Tarkvara võimaldab ka siiniali

erinevatel kujudel faili salvestada, kuid siiski mitte sellisel kujul, nagu seda on vaja nutiseadme mobiilirakenduse jaoks. Mobiilirakenduse loomiseks tuleb IRScope tarkvara poolt väljastatavat signaali esitust teisele kujule teisendada.

Infrapunasiinaale saab dekodeerida ka MSP430 mikrokontrolleriga, mida valmistab Texas Instruments [9]. Selle dokumentatsioonis kirjeldatakse Timer_A3 funktsionaalsust, millega dekodeeritakse SIRC ja RC5 infrapunasiinade protokolle. Dekodeerimiseks kasutatakse assemblerkeeles kirjutatud tarkvara. Füüsilisel tasemel signaalide salvestamisest juttu pole.

Arduino mikrokontrolleri kasutajatel on võimalik ühendada kontroller fotoelemendiga ja luua kontrollerile tarkvara, millega mõõta infrapunasiinaale ja mõõtmistulemusi soovitud kujul personaalarvutisse kuvada [10]. Tarkvara loomiseks kasutatakse C keele sarnast programmeerimiskeelt Processing. Sobival kujul salvestatud infrapunasiinaale saab kasutada nutiseadme mobiilirakenduse loomiseks.

3.3 Infrapunasiinade salvestamise võimaluste võrdlus

Eelpool kirjeldatud infrapunasiinade salvestamise võimaluste võrdluseks hinnatakse nende vastavust töös püstitatud nõuetele järgnevas tabelis.

Tabel 1. Infrapunasiinade salvestamise võimaluste võrdlus

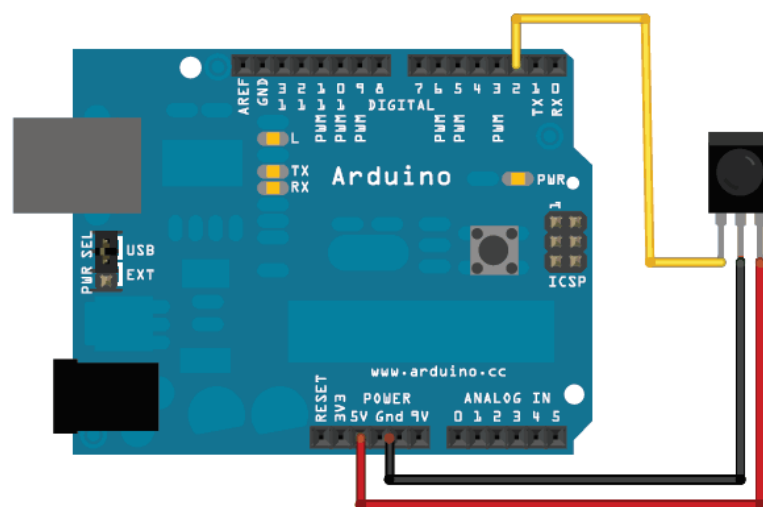
Lahendus	Lahenduse keerukus	Saab salvestada signaale füüsilisel tasemel	Hind
PIC mikrokontroller, erinevad diodid ja muud elemendid	Keeruline – tuleb palju elemente kokku panna ja kasutada assemblerkeelt programmeerimiseks.	Jah, kuid arvutiga ühendamine võib olla keeruline.	Üle 50 €
BrainLink	Valmistoode, kuid ei salvesta mobiilirakenduse jaoks vajalikul kujul signaale.	Jah.	125 €
Atmel STK500	Programmeerimiseks kasutatakse assemblerkeelt.	Jah.	80 €
IR Widget	Valmistoode, kuid ei salvesta mobiilirakenduse jaoks	Jah. Arvutiga USB	35 € Lisaks

	vajalikul kujul signaale.	ühendus.	saatmiskulud.
MSP430	Kasutatakse assemblerkeelt.	Ei.	10 €
Arduino	Fotoelemendiga ühendamine ja programmeerimine umbes 2 tundi. Kasutatakse C keele sarnast programmeerimiskeelt Processing.	Jah. Arvutiga USB ühendus.	35 €

Infrapunasiinade salvestamise erinevate võimaluste võrdlusel selgus, et enamus lahendustest ei sobi mobiilirakenduse jaoks signaalide salvestamiseks. Põhjuseks on enamasti assemblerkeele kasutamine riistvara programmeerimiseks, lahenduse keerukus ja suhteliselt suur hind.

3.4 Infrapunasiinade salvestamine Arduino kontrollriga

Käesolevas töös on valitud signaalide salvestamiseks Arduino mikrokontroller, mis on ühendatud fotoelemendiga (vaata joonis 3). Selle kasutamine ei eelda põhjalikke teadmisi ja kogemusi elektroonikas ning seda on lihtne USB liidese abil ühendada personaalarvutiga, näiteks sülearvutiga. Samuti on Arduino kontrolleri jaoks lihtne luua tarkvara, mille abil mõõta infrapunasiinade ja mõõtmistulemusi sobival kujul personaalarvutisse kuvada ja kasutada nutiseadme mobiilirakenduse loomiseks.



Joonis 3. Arduino kontrollriga ühendatud infrapunavastuvõtjaga

Allikas: <https://learn.adafruit.com/ir-sensor/using-an-ir-sensor>

Kontrolleriga ühendatav fotoelement peab olema häälestatud vastu võtma infrapunasiignaale sellise kandjasageduse vahemikuga, mille korral oleks mõõdetava signaali kandjasagedus selle vahemiku keskosas. Infrapunasiiaalidega juhitavate seadmete juhtimispuldid kasutavad enamasti umbes 38 kHz kandjasagedusega signaali. Seega võib mõõdetava signaali kandjasageduseks arvestada seda väärtust. Käesoleva töö käigus kasutati TSOP34838 fotoelementi. See on ette nähtud 38 kHz signaalide vastuvõtmiseks. Fotoelemendil on kolm viiku – toide, maa ja digitaalväljund. Infrapunakiirguse peale langemisel puudub fotoelemendi digitaalväljundis pinge. Kiirguse puudumisel on väljundpinge 5 volti.

Mikrokontrollerit on võimalik programmeerida kasutades Arduino IDE arendusvahendit ja C keele sarnast programmeerimiskeelt Processing. Kui infrapunasiignaali kandjasagedus on umbes 38 kHz, siis on signaali periood umbes 26 mikrosekundit. Fotoelemendiga ühendatud mikrokontroller programmeeriti kontrollima iga 20 mikrosekundi tagant fotoelemendi digitaalsisendi olekut ning mõõtma sel viisil impulsside ja nendevaheliste pauside kestusi infrapunasiignalis. Mikrokontrolleri programmi lähtekood on toodud käesoleva töö lisas (vaata Lisa 1). Mõõtmistulemuse saab USB pordi kaudu arvutis (näiteks sülearvuti) soovitud kujul kuvada ja salvestada ning kasutada nutiseadme mobiilirakenduse loomisel.

Infrapunasiignaali salvestamiseks juhtimispuldilt tuleb puldi infrapunasaatja (infrapuna LED) suunata fotoelemendi suunas. Infrapunasaatja ja fotoelemendi vahel peab olema otsenähtavus. Mõõtmise ajal tuleb vältida seda, et samal ajal satuks fotoelemendile teisi signaale. Infrapunasiiaalide salvestamise katsete käigus oli infrapunasaatja ja fotoelemendi vaheline kaugus 10 kuni 30 sentimeetrit.

Kui valida ajaühikuks signaali kandjasageduse periood, mis on umbes 26 mikrosekundit, siis on ühe signaali mõõtmistulemus ehk impulsside ja nendevaheliste pauside kestused vaheldumisi esitatav järgmisel kujul:

```
343,166,23,19,22,20,21,20,22,20,22,20,21,20,22,63,21,22,21,62,22,62,23,62,21,63,21,63,22,63,21,22,21,63,21,20,21,20,22,20,20,21,21,20,23,19,22,20,21,20,22,63,21,63,21,63,22,62,22,62,22,62,22,63,22,62,21,1511,340,83
```

Sellisel kujul signaali esitust saab kasutada nutiseadme mobiilirakenduse loomisel. Tegemist on füüsilisel tasemel infrapunasiignaali kirjeldusega. Kodeeringuid ja protokollitasetel signaali kirjeldust mobiilirakenduse loomisel vaja ei ole.

Kui ülaltoodud mõõtmistulemusi pisut hinnata, siis paistab, et need sisaldavad mõõtmisvigu. Siiski jäävad need vead 10% piiresse ja ei ole hiljem probleemiks mobiilirakenduse loomisel, sest infrapunasiinade vastuvõtjad arvestavad selliste võimalike vigadega vastuvõetavas (dekodeeritavas) signaalis [11].

Eeltoodud riistvaraline lahendus infrapunasiinade mõõtmiseks ja salvestamiseks on töö autori hinnangul kõige lihtsam lahendus ja vajab minimaalselt teadmisi elektroonikas. Internetis on saadaval juhendeid ka keerulisemate lahendustega, mis eeldavad paremaid teadmisi ja kogemusi elektroonika valdkonnas. Käesolevas töös on keskendunud põhimõttelisele lahendusele ehk füüsilisel tasemel signaali salvestamisele, mitte tehnilistele üksikasjadele, mis jäävad käesoleva töö skoobist välja.

4. Lahenduse kasutamise võimalused

Käesolevas töös valitud lahendus Androidi mobiilirakenduse ja Arduino kontrolleri ei sobi tavakasutajale kodus kasutamiseks. Tavakasutajal ei ole reeglina vajalikke teadmisi ja eelnevaid kogemusi käesolevas töös kasutatud riistvaralise lahenduse seadistamisel ja mobiilirakenduse loomisel. Tavakasutajale võib sellist lahendust pakkuda ühe teenusena või müügiartiklina IT teenuseid pakkavas ettevõttes.

Teenuse pakkumine võiks toimida järgmiselt. Klient toob ettevõttesse infrapunasiinjalidega juhitava seadme juhtimispuldi. Teenust pakkuv tehnik salvestab puldi infrapunasiinjalid ja kohandab mobiilirakenduse infrapunasiinjalide saatmiseks, et juhtida kliendi soovitud seadet – sisestab infrapunasiinjalid mobiilirakendusse ja lisab kliendi soovil mõne puuduva funktsiooni nupu, mida universaalpultidel pole.

Kasutajal pole vaja omada lisaseadet, kui teenust osutab spetsialist. Tehnilise toe eest tasumise võib eraldi kokku leppida. Kui kliendil puudub infrapunasaatjaga nutiseade, siis võib tehnik soovitada kliendile nutiseadmeid, millel on infrapunasaatja olemas ja millele saab luua mobiilirakenduse infrapunasiinjalidega juhitavate seadmete juhtimiseks.

Infotehnoloogia vallas kogenumad kasutajad võivad Arduino kontrolleri ka ise osta ja kodulektroonika (näiteks televiisor, digiboks või muusikakeskus) seadmete juhtimispultidelt signaalid salvestada. See on odav ja kasutamiseks suhteliselt lihtne. Materiaalsed kulud antud lahenduse loomiseks on umbes 35 € (Arduino kontrolleri, fotoelement, ühendusjuhtmed). Teostamiseks kodustel tingimustel kulub umbes paar tundi (jootmine, programmeerimine, infrapunasiinjalide salvestamine).

Fotoelemendi ühendamine kontrolleri ja infrapunasiinjalide salvestamine pultidelt ei vaja põhjalikke teadmisi ja oskusi elektroonikas ja programmeerimises. Samuti on internetis saadaval õpetused lihtsamate mobiilirakenduste loomiseks. Käesolevas töös kasutatud Androidi mobiilirakenduse arendusvahend Android Studio on vabavaraline.

5. Olemasolevad lahendused infrapunasiinagaalidega juhitavate seadmete juhtimiseks

Internetist otsitud materjali põhjal on järgnevalt välja toodud mõned olemasolevad lahendused infrapunasiinagaalidega juhitavate seadmete juhtimiseks.

5.1 Panja

1990ndate aastate lõpus hakkasid firma Panja rajajad Scott Miller ja tema abikaasa aktiivselt välja töötama ja rakendama mugava kasutajaliidesega universaalset juhtimiskompleksi erinevate seadmete juhtimiseks, näiteks audio-videoseadmed, kliimaseadmed ja teised [12]. Selle süsteemi aluseks on keskkontroller, mis integreerib kõik seadmed spetsiaalsete liideste abil. Näiteks infrapunaliidese abil võib juhtida pea kogu koduelektroonikat.

Praktiliselt kõik Panja integreeritud juhtimise süsteemi komponendid jäävad lõppkasutaja jaoks varjatuks. Keskkontroller, infrapunasatjad, juhtmed ja pistikud monteeritakse seintesse või paigaldatakse selliselt, et ruumi interjööri ei muutuks nende olemasolust. Kasutaja töötab ainult juhtpaneeliga, mis kujutab endast sensor-ekraani diagonaaliga kuni 21 tolli. Ekraan on saadaval kahes variandis: juhtmevaba, infrapuna ja raadiosagedusliku liidese, ja juhtmega statsionaarne. Juhtmevaba paneel võimaldab juhtida seadmeid praktiliselt igast ruumi punktist. Puldi graafiline kasutajaliides võib olla suvaline. Standardset lahendust siin ei ole – puldi nupud, nende parameetrid ja järjekord määratakse spetsiaalse tarkvara abil vastavalt kasutaja konkreetsetele individuaalsetele soovidele.

Integreeritud juhtimissüsteemide lahendused nõuavad suurt eeltööd: seadmete asukoha märkimine, seadmete valik vastavalt kasutusotstarbele, ruumi interjööri arvestamine jne. See aga kujundab hinda. Näiteks audio-videoseadmete juhtimissüsteemi paigaldamine ja kasutajaliidese programmeerimine maksab 3 kuni 5 tuhat dollarit. Eramaja juhtimissüsteem, mis võimaldab mistahes toast juhtida lisaks ka kliimaseadmeid, garaaži väravaid ja olmeseadmeid jne. maksab 40 tuhat dollarit.

See süsteem on küll väga tõhus ja välja töötatud firma Panja poolt, kuid see vajab suurt ja mahukat eeltööd ja on ettenähtud suurte kontorite jaoks, mitte üksiku eraisikust kliendi jaoks. Samuti on see väga palju aega nõudev ja väga kallis.

5.2 AnyMote Home

Infrapunaliidesega nutitelefoni saab muuta universaalseks puldiks spetsiaalse seadme AnyMote Home abil [13]. AnyMote kujutab endast kompaktnutitelefoni, mis muudab nutitelefoni universaalseks puldiks. Selle abil on võimalik juhtida televiisorit, arvutit, muusikakeskust, kliimaseadmeid ja palju teisi koduelektroonika seadmeid.



Joonis 4. AnyMote Home

Allikas: <https://www.kickstarter.com/projects/1635386542/anymote-home-your-phone-the-ultimate-universal-rem>

AnyMote Home sisaldab 4 võimsat infrapuna-valgusdiodi ja ühte infrapuna-vastuvõtjat. Seade suudab töötada 10 meetri raadiuses. Piisab, kui panna AnyMote Home selliselt, et juhitav seade oleks nähtav. Peale seda tuleb ühendada nutiseade AnyMote seadmega Bluetooth 4.0 LE abil. Nutiseadmel peab olema eelnevalt paigaldatud vajalik mobiilirakendus, mis laiendab võimalusi seadmeid juhtida.

Korpus, milles asub AnyMote Home, on raadiusega 10 cm ja kõrgusega 4,5 cm. Sellel on kaks AA tüüpi akupatareid mahtuvusega 2000 mAh. Keskmiselt jätkub nendest patareid terveks aastaks juhul, kui päevas täidetakse 2700 käsku.

AnyMote on kasutatav nii iOS kui ka Androidi seadmetega. AnyMote Home maksumus on 70 – 85 dollarit. Seade tuleb müügile 2015. aasta mais või juunis.

Kirjeldatud lahendus on igati positiivne ja laia kasutusvõimalustega erinevate seadmete suhtes. Kuid see vajab lisaseadet AnyMote Home, mis on kallis ja töötab akupatareidega.

5.3 RedEye

RedEye on ThinkFlood loodud juhtimissüsteem, mis võimaldab kasutada iOS või Android operatsioonisüsteemiga seadet multifunktsionaalse universaalse puldina [14]. RedEye süsteem koosneb kahest osast: iOS või Androidi mobiilirakendus ja riistvaraline seade. Mobiilirakendused iOS jaoks on tasuta ja töötavad kõikide iPhone, iPod Touch või iPad seadmetega alates iOS versioonist 3.0. Androidi rakendused on samuti tasuta ja töötavad alates versioonist Android OS 1.6 (Donut).

Nutiseade saadab Wi-Fi kaudu kontrolleri info ja RedEye muudab saadud andmed infrapunasiinjalideks ja saadab need koduelektroonika seadmetele. Wi-Fi kaudu on võimalik ühendada seade ka internetiga ja sellega saab sünkroniseerida konfiguratsioone kõikide nutiseadmetega, mis suhtlevad RedEye kontrolleri ja mida kasutatakse juhtimispuldina. Sel viisil saavad ühes kodus mitu inimest kasutada oma nutiseadmeid juhtimispuldina. Vajadusel saab RedEye abil teha varukoopia konfiguratsioonidest.

iOS ja Androidi rakendused pakuvad rohkeid võimalusi ja võimaldavad luua makrosid, mille abil saab ühe nupuvajutusega sooritada mitu mistahes kestusega tegevust. Suur interneti andmebaas üle 1600 seadmetootjaga võimaldab valida vajalikud infrapunasiinjalide koodid praktiliselt igale seadmele majas.

Puudusena võib välja tuua keerulise algse seadistamise, mis võtab palju aega, eriti kui pole palju kogemusi elektroonikaga [15].

Võimalused:

- Mistahes seadmete juhtimine, mis töötavad infrapunasiinjalidega
- Võimalus luua makrokäskude suure arvu käskudega ja ajalise viitega
- Automaatne sünkroniseerimine mitme nutiseadme vahel
- Võimalus luua varukoopiaid konfiguratsioonidest
- Suur interneti andmebaas üle 1600 seadmetootja infrapunasiinjalide koodidega
- Infrapunasiinjalide õpetamise võimalus (olemas infrapunavastuvõtja)

5.4 Olemasolevate lahenduste võrdlus

Eelpool kirjeldatud olemasolevate lahenduste võrdluseks võib välja tuua nende eelised ja puudused järgnevas tabelis.

Tabel 2. Olemasolevate lahenduste võrdlus.

Seadme nimi	Eelised	Puudused
Panja	Süsteem on väga tõhus ja mitme aasta jooksul välja töötatud.	Vaja eraldi seadet ja palju eeltööd paigaldamiseks. Ette nähtud suurte kontorite jaoks, mitte üksiku eraisikust kliendi jaoks. Kallis (3 – 5 tuhat dollarit).
AnyMote Home	Laia kasutusvõimalustega. Seade suudab töötada 10 meetri raadiuses.	Vaja eraldi seadet. Kallis (alates 85 dollarit).
RedEye	Võimaldab luua makrosid (ühe nupuvajutusega sooritada mitu tegevust). Suur andmebaas infrapunasiinjalide koodidega.	Vaja eraldi seadet. Keeruline algne seadistamine, mis võtab palju aega, eriti kui pole palju kogemusi elektroonikaga. Kallis (üle 100 dollari).
Käesolevas töös valitud lahendus Arduino kontrolloriga	Odav lahendus (kontroller, fotoelement ja ühendusjuhtmed maksid 35 €). Sobib kasutamiseks eraisikule teenuse vahendusel. Kasutajal pole vaja lisaseadet omada.	Ei ole mõeldud masstootmiseks.

Nagu võrdlustabelist näha, on olemasolevad toodetavad seadmed Panja, AnyMote Home ja RedEye tõhusad, võimaldavad juhtida palju erinevaid koduelektroonika seadmeid, nende järele on nõudlus, kuid need on suhteliselt kallid. Eelpool kirjeldatud infrapunasiinjalidega juhtimise lahendustes kasutatakse lisaseadmeid (AnyMote ja RedEye) või paigaldatakse kontrollerid, juhtmeid ja pistikud (Panja). Ajaliselt peab arvestama, et esmane paigaldus võtab aega 4 tundi (RedEye) või vajab mahukat eeltööd (Panja).

Käesolevas töös kasutatud lahendus Arduino kontrolleriiga on mõeldud eraisikule koduelektroonika (näiteks audio-videoseadmed, digiboks) seadmete juhtimiseks. See on odav ja kasutamiseks lihtne, sest selle kasutamine ei vaja põhjalikke teadmisi ja oskusi elektroonikas ja programmeerimises. Materiaalsed kulud antud lahenduse loomiseks on umbes 35 € (Arduino kontrolleri, fotoelement, ühendusjuhtmed).

Arduino platvorm on mõeldud riistvaraliste lahenduste prototüüpide loomiseks. Teostamiseks kodustel tingimustel kulub umbes paar tundi (jootmine, programmeerimine, infrapunasiinjalide salvestamine). Kasutajal pole vaja omada lisaseadet, kui teenust osutab spetsialist.

6. Kokkuvõte

Käesoleval töö on kaks eesmärki. Üheks eesmärgiks on uurida kuidas luua nutiseadme mobiilirakendus infrapunasiinjalide saatmiseks, et juhtida infrapunasiinjalidega juhitavaid seadmeid. Teiseks eesmärgiks on uurida kuidas salvestada infrapunasiinjalid füüsilisel tasemel infrapunasiinjalidega juhitavate seadmete juhtimispuulidelt.

Töö olulisemad tulemused on riistvaralise lahenduse valik füüsilisel tasemel infrapunasiinjalide salvestamiseks infrapunasiinjalidega juhitavate seadmete juhtimispuulidelt ja nutiseadme mobiilirakendus salvestatud infrapunasiinjalide saatmiseks, et juhtida infrapunasiinjalidega juhitavaid seadmeid.

Töös püstitatud hüpoteesid:

- 1) saab kasutada uuemaid nutitelefone, mis sisaldavad infrapunasiinjalide saatjat;
- 2) saab juhtida kõiki infrapunasiinjalidega juhitavaid koduelektronika seadmeid;
- 3) selline lahendus on odav ja kättesaadav paljudele inimestele.

Esimesed kaks hüpoteesi ei leidnud töös täielikku kinnitust, sest selgus, et näiteks Sony Xperia ZL nutiseadmega ei saa mistahes infrapunasiinjalid saata. Samas õnnestus see Samsung Galaxy S4 GT-I9505 nutiseadmega.

Käesolevas töös pakutav lahendus mobiilirakenduse ja Arduino kontrolleriiga ei sobi tavakasutajale kodus kasutamiseks. Tavakasutajale võib sellist lahendust pakkuda ühe teenusena IT teenuseid pakkuvast ettevõttes, kuhu klient toob infrapuna-juhtimispuuldi. Tehnik salvestab puuldi infrapunasiinjalid ja kohandab mobiilirakenduse infrapunasiinjalide saatmiseks, et juhtida kliendi soovitud seadet. Kasutajal pole vaja omada lisaseadet, kui teenust osutab spetsialist.

Infotehnoloogia vallas kogenumad kasutajad võivad Arduino kontrolleri ka ise osta ja koduelektronika (näiteks televiisor, digiboks või muusikakeskus) seadmete juhtimispuulidelt siinjalid salvestada. See on odav ja kasutamiseks suhteliselt lihtne. Materiaalsed kulud antud lahenduse loomiseks on umbes 35 € (Arduino kontrolleri, fotoelement, ühendusjuhtmed). Teostamiseks kodustel tingimustel kulub umbes paar tundi (jootmine, programmeerimine, infrapunasiinjalide salvestamine).

Fotoelemendi ühendamine kontrolloriga ja infrapunasiinnaalide salvestamine pultidelt ei vaja põhjalikke teadmisi ja oskusi elektroonikas ja programmeerimises. Samuti on internetis saadaval õpetused Androidi mobiilirakenduste loomiseks. Käesolevas töös kasutatud Androidi mobiilirakenduse arendusvahend Android Studio on vabavaraline.

Lisaks võrreldi muid olemasolevaid toodetavaid seadmeid Panja, AnyMote Home ja RedEye. Need on tõhusad, võimaldavad juhtida palju erinevaid koduelektroonika seadmeid, nende järele on nõudlus, kuid need on suhteliselt kallid.

Töös püstitatud eesmärgid on saavutatud. Valitud on riistvaraline lahendus infrapunasiinnaalide salvestamiseks infrapunasiinnaalidega juhitavate seadmete juhtimispultidelt ja loodud on lihtne nutiseadme mobiilirakendus erinevate seadmete juhtimiseks. Katsete tulemusel selgus, et valitud lahendus töötab ja mobiilirakendusega saab juhtida erinevaid infrapunasiinnaalidega juhitavaid seadmeid.

Töös seati nõuded riistvaralisele lahendusele infrapunasiinnaalide salvestamiseks:

- 1) infrapunasiinnaalide salvestamine peab olema võimalikult lihtne;
- 2) infrapunasiinnaale peab saama salvestada füüsilisel tasemel;
- 3) vahend siinnaalide salvestamiseks peab olema võimalikult odav.

Infrapunasiinnaalide salvestamise võimaluste võrdluseks hinnati nende vastavust töös püstitatud nõuetele. Töös püstitatud eesmärgid on saavutatud uurimise ja katsetamise teel. Infrapunasiinnaalide salvestamiseks tutvuti infrapunasiinnaalidega juhitavate seadmete juhtimispultide tööpõhimõttega, uuriti erinevaid võimalusi siinnaalide salvestamiseks ja tehti katseid pultidelt infrapunasiinnaalide vastuvõtmiseks ja salvestamiseks.

Salvestatud siinnaale kasutades loodi mobiilirakendus nutiseadmele, millel on infrapunasaatja. Mobiilirakenduse katsetamiseks prooviti juhtida erinevaid infrapunasiinnaalidega juhitavaid seadmeid: televiisoreid, digibokse, muusikakeskusi, videoprojektoreid ja DVD-mängijaid. Katsete tulemusena selgus, et nutiseadme ja mobiilirakenduse abil saab juhtida infrapunasiinnaalidega juhitavaid seadmeid eelnevalt salvestatud infrapunasiinnaalidega.

Võimalikud edasiarendused: infrapunasiinnaale võiks salvestada interneti. Võib olla tuleb müüki nutiseade, millel on infrapunavastuvõtja olemas ja mis suudab siinnaale salvestada. Sel juhul pole vaja kontrolleri infrapunasiinnaalide salvestamiseks.

Summary

The aim of this thesis is to study the possibilities and the means to control any infrared remote controlled device by using a smartphone and a mobile application. This thesis has two main objectives. The first objective is to study how to capture infrared signals at a physical level from different infrared remote controllers and saving those signals to a computer. The second objective is to create a mobile application for a smartphone equipped with an infrared emitter. The mobile application enables to use the captured and saved infrared signals for controlling many infrared remote controlled devices with an infrared emitter equipped smartphone.

The main problems dealt with were saving the raw infrared signals (on the physical level) from infrared remote controllers and using the saved signals to control infrared remote controlled devices by the use of a smartphone and a mobile application.

The main results of this thesis are the hardware solution to capture infrared signals from infrared remote controllers and the smartphone application to control infrared remote controlled devices using the captured infrared signals. The hardware solution for capturing signals consists of Arduino controller connected to an IR photo-receiver. The smartphone application comprises an Android application that uses the infrared emitter of the smartphone.

The hardware is cheap and easy to use. It does not require advanced knowledge and experience in electronics and it is easy to connect to a personal computer, for example a laptop. In addition, it is easy to create software for Arduino controller to capture infrared signals and save them to a computer in the necessary format for the smartphone application.

The main objectives of this thesis have been achieved. It is possible to use a smartphone equipped with an infrared emitter to control most of the infrared remote controlled consumer electronics devices – television sets, digital tuners, home audio systems, video projectors, DVD players.

Kasutatud kirjandus

1. Инфракрасное дистанционное управление. [WWW]
<http://altor1.narod.ru/Articles/IRC.pdf> (25.04.2015)
2. ConsumerIrManager, Android Developers. [WWW]
<https://developer.android.com/reference/android/hardware/ConsumerIrManager.html>
(25.04.2015)
3. Sony Add-on SDK documentation kit, Sony Developer World. [WWW]
<https://developer.sony.com/downloads/documentation/sony-add-on-sdk-documentation-kit/> (25.04.2015)
4. IR Remote Signals, Adafruit Learning System. [WWW]
<https://learn.adafruit.com/ir-sensor/ir-remote-signals> (25.04.2015)
5. IR Remote Control Signal Capture and Visualization, Instructables. [WWW]
<http://www.instructables.com/id/IR-Remote-Control-Signal-Capture-and-Visualization/>
(25.04.2015)
6. Brainlink Hardware Description, Brainlink. [WWW]
<http://www.brainlinksystem.com/brainlink-hardware-description> (25.04.2015)
7. Measuring Infrared-Remote Controls, AVR Assembler Tutorial. [WWW]
http://www.avr-asm-tutorial.net/avr_en/ir/ir_measure/ir_measure.html (25.04.2015)
8. IR Remote Control Signal Recorder from TommyTyler on Tindie. [WWW]
<https://www.tindie.com/products/TommyTyler/ir-widget-infrared-remote-control-signal-recorderanalyzer/> (25.04.2015)
9. Texas Instruments: Decode TV IR Remote Control Signals Using Timer_A3. [WWW]
<http://www.ti.com/lit/an/slaa134/slaa134.pdf> (25.04.2015)
10. Using an IR Sensor, Adafruit Learning System [WWW]
<https://learn.adafruit.com/ir-sensor/using-an-ir-sensor> (25.04.2015)

11. Infrared Remote Control Protocols, irq5.io. [WWW]
<http://irq5.io/2012/07/27/infrared-remote-control-protocols-part-1/> (25.04.2015)
12. Елена Новикова, Игорь Ишеев „Системы интегрированного управления, или как с помощью мобильного телефона включить кондиционер“. [WWW]
<http://www.install-pro.ru/archive/008/61-63.shtml> (25.04.2015)
13. AnyMote Home + Your Phone = The Ultimate Universal Remote, Kickstarter. [WWW]
<https://www.kickstarter.com/projects/1635386542/anymote-home-your-phone-the-ultimate-universal-rem> (25.04.2015)
14. RedEye – универсальный пульт дистанционного управления для вашего мобильного устройства на базе iOS или Android [WWW]
<http://www.airfon-media.ru/2012/10/17/redeye-универсальный-пульт-дистанционного-управления-для-вашего-мобильного-устройства-на-базе-ios-или-android/> (25.04.2015)
15. RedEye Universal Remote System for iPhone, iPod touch and iPad. Amazon.com. [WWW] <http://www.amazon.com/RedEye-Universal-Remote-Discontinued-Manufacturer/dp/B0020A0IPO> (25.04.2015)

Lisa 1

Programmikood Arduino kontrolleri infrapunasignaalide mõõtmiseks ja kuvamiseks.

Kasutatud allikas: <https://learn.adafruit.com/ir-sensor/using-an-ir-sensor>

```
/* Raw IR decoder sketch!
This sketch/program uses the Arduno and a PNA4602 to decode IR received. This can
be used to make a IR receiver (by looking for a particular code) or transmitter
(by pulsing an IR LED at ~38KHz for the durations detected. Code is public domain,
check out www.ladyada.net and adafruit.com for more tutorials!
*/

// We need to use the 'raw' pin reading methods because timing is very important
here and the digitalRead() procedure is slower!
//uint8_t IRpin = 2; Digital pin #2 is the same as Pin D2 see
// http://arduino.cc/en/Hacking/PinMapping168 for the 'raw' pin mapping
#define IRpin_PIN PIND
#define IRpin 2

// the maximum pulse we'll listen for - 65 milliseconds is a long time
#define MAXPULSE 65000

// what our timing resolution should be, larger is better
// as its more 'precise' - but too large and you wont get
// accurate timing
#define RESOLUTION 20

// we will store up to 100 pulse pairs (this is -a lot-)
uint16_t pulses[100][2]; // pair is high and low pulse
uint8_t currentpulse = 0; // index for pulses we're storing

void setup(void) {
  Serial.begin(9600);
  Serial.println("Ready to decode IR! Output is in microseconds divided by 26.");
}

void loop(void) {
  uint16_t highpulse, lowpulse; // temporary storage timing
  highpulse = lowpulse = 0; // start out with no pulse length

  // while (digitalRead(IRpin)) { // this is too slow!
  while (IRpin_PIN & (1 << IRpin)) {
    // pin is still HIGH

    // count off another few microseconds
    highpulse++;
    delayMicroseconds(RESOLUTION);
  }
}
```

```

// If the pulse is too long, we 'timed out' - either nothing
// was received or the code is finished, so print what
// we've grabbed so far, and then reset
if ((highpulse >= MAXPULSE) && (currentpulse != 0)) {
    printpulses();
    currentpulse=0;
    return;
}
}
// we didn't time out so lets stash the reading
pulses[currentpulse][0] = highpulse;

// same as above
while (!(IRpin_PIN & _BV(IRpin))) {
    // pin is still LOW
    lowpulse++;
    delayMicroseconds(RESOLUTION);
    if ((lowpulse >= MAXPULSE) && (currentpulse != 0)) {
        printpulses();
        currentpulse=0;
        return;
    }
}
pulses[currentpulse][1] = lowpulse;

// we read one high-low pulse successfully, continue!
currentpulse++;
}

void printpulses(void) {

for (uint8_t i = 0; i < currentpulse-1; i++) {
    Serial.print(pulses[i][1] * RESOLUTION / 26, DEC);
    Serial.print(",");
    Serial.print(pulses[i+1][0] * RESOLUTION / 26, DEC);
    Serial.print(",");
}
Serial.println("");
}

```


Lisa 2

Nutiseadme Androidi mobiilirakenduse programmikood.

```
package com.iremotecontrol.app;

import android.hardware.ConsumerIrManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

import com.iremotecontrol.device.*;

public class MainActivity extends ActionBarActivity implements
    AdapterView.OnItemClickListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        onCreateAction();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }

    private ConsumerIrManager irManager;

    private void onCreateAction() {
        irManager = (ConsumerIrManager) getSystemService(CONSUMER_IR_SERVICE);
        initSpinner();
    }

    private void setButtonListeners(AbstractDevice device) {
        setListenerForButton(R.id.buttonOnOff, device.ONOFF);
        setListenerForButton(R.id.button0, device.NR_0);
    }
}
```

```

setListenerForButton(R.id.button1, device.NR_1);
setListenerForButton(R.id.button2, device.NR_2);
setListenerForButton(R.id.button3, device.NR_3);
setListenerForButton(R.id.button4, device.NR_4);
setListenerForButton(R.id.button5, device.NR_5);
setListenerForButton(R.id.button6, device.NR_6);
setListenerForButton(R.id.button7, device.NR_7);
setListenerForButton(R.id.button8, device.NR_8);
setListenerForButton(R.id.button9, device.NR_9);
setListenerForButton(R.id.buttonNumberMode, device.NR_MODE);
setListenerForButton(R.id.buttonVolumePlus, device.VOL_PLUS);
setListenerForButton(R.id.buttonVolumeMinus, device.VOL_MIN);
setListenerForButton(R.id.buttonMute, device.MUTE);
setListenerForButton(R.id.buttonInfo, device.INFO);
setListenerForButton(R.id.buttonProgramPlus, device.PROG_PLUS);
setListenerForButton(R.id.buttonProgramMinus, device.PROG_MIN);
setListenerForButton(R.id.buttonAV, device.AV);
setListenerForButton(R.id.buttonTextTV, device.TEXT_TV);
setListenerForButton(R.id.buttonTuning, device.TUNE);
setListenerForButton(R.id.buttonContrast, device.CONTRAST);
setListenerForButton(R.id.buttonPlus, device.PLUS);
setListenerForButton(R.id.buttonMinus, device.MINUS);
setListenerForButton(R.id.buttonF1, device.F1);
setListenerForButton(R.id.buttonF2, device.F2);
setListenerForButton(R.id.buttonF3, device.F3);
}

```

```

private void setListenerForButton(int buttonId, final int[] irPattern){
    View button = findViewById(buttonId);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            new Thread(
                new Runnable() {
                    @Override
                    public void run() {
                        irManager.transmit(38000, irPattern);
                    }
                }
            ).start();
        }
    });
}

```

```

private void initSpinner() {
    Spinner spinner = (Spinner) findViewById(R.id.devices_spinner);
    // Create an ArrayAdapter using the string array and a spinner layout
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
        R.array.planets_array, R.layout.spinner_item);
    // Specify the layout to use when the list of choices appears
    adapter.setDropDownViewResource(
        android.R.layout.simple_spinner_dropdown_item);
    // Apply the adapter to the spinner
    spinner.setAdapter(adapter);
    spinner.setOnItemSelectedListener(this);
}

```

```

@Override
public void onItemClick(AdapterView<?> parent, View view, int pos, long id)
{
    // An item was selected. You can retrieve the selected item.
    switch (parent.getItemAtPosition(pos).toString()) {
        case "Toshiba":
            setButtonListeners(new ToshibaCT9463());
            break;
        case "Digibox":
            setButtonListeners(new DigiboxStarman());
            break;
        case "Projector":
            setButtonListeners(new VideoProjector());
            break;
        case "DVD player":
            setButtonListeners(new DvdPlayer());
            break;
        case "Music center":
            setButtonListeners(new SonyMusicCenter());
            break;
        default:
            break;
    }
}

@Override
public void onNothingSelected(AdapterView<?> parent) {
    // Another interface callback
}
}

```

```

package com.iemotecontrol.device;

public abstract class AbstractDevice {

    public int[] ONOFF = new int[] {};
    public int[] NR_0 = new int[] {};
    public int[] NR_1 = new int[] {};
    public int[] NR_2 = new int[] {};
    public int[] NR_3 = new int[] {};
    public int[] NR_4 = new int[] {};
    public int[] NR_5 = new int[] {};
    public int[] NR_6 = new int[] {};
    public int[] NR_7 = new int[] {};
    public int[] NR_8 = new int[] {};
    public int[] NR_9 = new int[] {};

    public int[] NR_MODE = new int[] {};
    public int[] VOL_PLUS = new int[] {};
    public int[] VOL_MIN = new int[] {};
    public int[] MUTE = new int[] {};
    public int[] INFO = new int[] {};
    public int[] PROG_PLUS = new int[] {};
    public int[] PROG_MIN = new int[] {};
    public int[] AV = new int[] {};
    public int[] TEXT_TV = new int[] {};
}

```

```

public int[] TUNE = new int[] {};
public int[] CONTRAST = new int[] {};
public int[] PLUS = new int[] {};
public int[] MINUS = new int[] {};

public int[] F1 = new int[] {};
public int[] F2 = new int[] {};
public int[] F3 = new int[] {};
}

```

```
package com.iremotecontrol.device;
```

```
public class ToshibaCT9463 extends AbstractDevice {
```

```
    public ToshibaCT9463() {
```

```

        ONOFF = new int[] {338,167,21,20,22,20,21,20,21,20,21,20,21,63,20,
20,21,63,20,63,21,62,21,63,20,63,21,63,20,20,21,62,21,20,21,62,21,20,21,63,20,
1,20,21,20,20,20,21,63,21,20,21,63,21,62,21,20,20,63,21,62,22,62,21,1496,338,83};
        NR_0 = new int[] {338,167,21,20,21,20,21,20,21,20,21,20,21,63,20,20,
21,62,21,63,21,62,21,63,21,62,21,63,21,20,21,63,21,20,21,20,21,20,20,20,21,2
0,21,20,21,20,21,62,21,62,21,63,21,62,21,63,21,62,21,63,21,62,21,1496,338,83};
        NR_1 = new int[] {338,167,23,19,20,20,21,20,21,20,21,20,21,63,23,18,
23,60,23,60,23,60,23,60,23,60,23,18,21,63,23,60,23,18,23,18,23,18,23,18,23,1
8,23,18,23,18,20,20,23,61,23,60,23,60,23,60,23,60,23,60,21,62,21,1496,338,83};
        NR_2 = new int[] {339,166,23,18,23,18,23,18,23,18,23,15,26,18,23,60,23,18,
23,60,23,60,23,60,23,60,23,60,23,18,23,60,23,18,23,60,23,18,23,19,23,18,23,1
8,23,18,23,18,23,60,23,18,23,60,23,60,23,60,23,60,23,60,23,60,23,1496,338,83};
        NR_3 = new int[] {338,167,23,18,21,20,21,20,21,20,22,20,21,20,21,62,23,18,
20,63,23,60,23,60,23,60,23,60,23,19,23,60,23,60,23,18,23,18,23,18,23,18,21,2
0,21,20,23,18,21,20,23,18,23,60,23,60,23,60,23,60,23,60,23,60,23,60,23,1496,338,84};
        NR_4 = new int[] {337,168,23,18,23,17,23,18,23,17,23,18,23,17,23,60,23,18,
23,60,23,60,23,60,23,60,23,60,23,60,23,18,23,60,23,17,23,18,23,60,23,19,23,18,23,1
8,23,18,23,18,23,60,23,60,23,18,23,60,23,60,23,60,23,60,23,60,23,60,23,1496,338,83};
        NR_5 = new int[] {338,167,23,18,23,19,23,18,23,18,23,18,23,18,23,60,23,18,
23,60,23,61,23,60,23,60,23,60,23,60,23,18,23,60,23,60,23,18,21,62,23,18,23,17,23,1
8,23,17,23,18,21,20,23,60,23,18,23,60,23,60,23,60,23,60,23,60,23,1496,339,83};
        NR_6 = new int[] {338,167,23,18,23,18,23,18,23,18,23,18,23,18,23,60,23,18,
23,60,23,60,23,60,23,60,23,60,23,18,23,60,23,17,23,60,23,60,23,18,23,19,23,1
8,23,18,23,18,23,60,23,18,23,17,23,60,23,60,23,60,23,60,23,60,23,60,23,1496,338,84};
        NR_7 = new int[] {337,168,23,18,23,18,23,18,23,18,23,18,23,18,23,60,23,17,
23,60,23,60,23,60,23,60,23,60,23,18,21,62,23,60,23,60,23,60,23,18,21,20,21,2
0,21,20,21,20,21,20,21,20,21,63,23,60,23,60,23,60,23,60,23,1496,338,83};
        NR_8 = new int[] {338,167,23,18,23,17,23,18,23,18,23,18,23,18,23,60,23,18,
23,60,23,60,23,60,23,60,23,60,23,17,23,60,23,18,23,18,23,17,23,60,23,18,23,1
8,23,17,23,18,23,60,23,60,23,60,23,17,23,60,23,60,23,60,23,60,23,60,23,1496,338,83};
        NR_9 = new int[] {338,167,23,17,21,20,21,20,20,20,21,20,21,20,21,62,23,18,
21,62,23,60,23,60,23,60,23,60,23,18,23,60,23,60,23,17,21,20,21,62,23,18,21,2
0,22,20,21,20,21,20,21,62,23,60,23,18,23,60,23,60,23,60,23,60,23,1496,338,83};

        NR_MODE = new int[] {338,167,21,20,21,20,21,20,21,20,21,20,21,63,21,
20,21,63,21,62,21,62,21,63,21,62,21,63,20,20,21,63,21,62,21,63,20,20,21,63,21,20,2
1,20,21,20,21,20,22,20,21,20,20,63,21,20,20,63,21,62,21,63,21,62,21,1496,338,83};
        VOL_PLUS = new int[] {338,167,21,20,21,20,21,20,20,20,21,20,21,20,21,62,

```

```

21,20,21,62,21,63,21,62,21,63,21,62,21,62,21,20,21,62,21,20,21,62,21,20,21,62,21,6
3,21,20,21,20,21,20,21,63,21,20,21,63,20,20,21,20,22,62,21,62,21,63,21,1496,338,83
};
    VOL_MIN = new int[] {338,167,21,20,21,20,20,20,21,20,21,20,21,20,21,62,21,
20,21,62,21,63,20,63,21,62,21,63,21,62,21,20,21,62,21,20,21,62,21,63,20,63,21,62,2
1,20,21,20,21,20,21,62,21,20,20,20,21,20,21,20,22,62,21,62,21,63,20,1496,338,83};
    MUTE = new int[] {338,167,22,20,21,20,21,20,21,20,20,20,21,20,21,62,21,20,
22,62,21,62,21,63,20,63,21,62,21,63,21,20,21,63,21,20,21,20,21,20,20,20,21,62,22,2
0,21,20,21,20,21,62,21,63,20,63,21,62,22,20,21,62,21,63,21,62,21,1496,338,84};
    INFO = new int[] {341,168,22,21,20,22,20,20,21,21,20,22,20,20,21,63,21,23,
20,63,22,62,22,63,21,63,21,63,22,63,21,23,20,63,21,22,20,63,20,63,23,20,20,62,23,2
0,20,22,20,20,21,63,20,21,21,20,20,63,22,21,20,63,22,63,21,63,21,1500,340,83};
    PROG_PLUS = new int[] {338,167,21,20,21,20,21,20,22,20,21,20,20,20,21,63,
21,20,21,63,20,63,21,62,21,63,20,63,21,63,20,20,21,62,21,63,21,62,21,20,20,63,21,6
3,20,20,21,20,21,20,21,20,21,63,21,20,21,20,20,63,21,62,21,63,21,1496,338,83
};
    PROG_MIN = new int[] {338,167,21,20,21,18,23,20,21,20,21,20,21,20,21,62,
21,20,21,62,21,63,21,62,21,62,21,63,21,62,21,20,21,62,21,63,20,63,21,62,21,63,21,6
2,21,20,21,20,21,20,21,20,21,20,21,20,21,20,21,63,20,63,21,63,20,1496,337,83
};
    AV = new int[] {338,167,21,20,21,20,21,20,21,20,21,20,21,20,21,63,21,
20,21,63,21,62,21,63,21,62,21,63,20,63,21,20,22,62,21,20,21,20,20,63,21,20,20,63,2
1,20,21,20,21,20,21,62,21,63,20,20,21,63,20,20,21,62,22,62,21,62,21,1496,338,83};
    TEXT_TV = new int[]
{340,169,21,20,21,20,22,20,21,20,21,20,23,20,20,63,23,20,23,62,21,63,21,63,22,63,2
1,63,21,62,23,20,22,63,21,63,21,63,22,63,21,20,22,63,21,20,22,20,21,20,21,20,23,19
,22,20,21,63,21,20,21,63,21,63,22,63,21,1510,341,83,21,1116,340,83};

    TUNE = new int[]
{340,169,22,20,21,20,21,20,21,20,22,20,21,20,21,63,21,23,20,63,22,63,21,62,21,63,2
3,62,21,63,21,23,20,62,23,20,20,20,22,62,21,63,22,21,20,20,21,20,22,20,21,63,22,63
,21,21,21,20,21,63,22,62,22,62,22,62,22,1510,340,84,22,1115,341,83};
    CONTRAST = new int[]
{340,169,21,20,23,20,21,20,21,20,22,20,22,20,21,63,23,20,21,63,22,63,22,62,23,60,2
2,63,21,63,23,20,20,63,23,20,21,20,21,20,23,62,21,63,23,19,21,19,23,20,21,63,22,63
,21,63,23,19,21,20,23,62,21,63,23,60,22,1510,340,84};
    PLUS = new int[]
{340,169,22,20,21,20,22,20,21,21,20,20,22,20,21,63,22,20,23,63,21,62,21,63,23,62,2
1,63,21,63,22,21,22,63,21,63,20,21,21,20,23,62,21,63,21,20,21,20,23,20,21,20,21,63
,21,63,22,20,21,21,21,62,21,63,22,63,21,1510,340,83,21,1117,340,83};
    MINUS = new int[]
{341,168,23,20,20,22,20,20,21,21,20,21,20,20,21,63,21,23,20,63,21,62,23,62,21,63,2
1,63,22,63,21,23,20,63,21,62,23,20,20,63,22,63,21,62,21,22,20,20,21,21,20,21,20,63
,21,21,20,21,21,20,20,62,23,62,22,62,21,1511,340,83,23,1115,340,85};

    F1 = new int[] {};
    F2 = new int[] {};
    F3 = new int[] {};
}
}

```