

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutitehnika instituut

IAY40LT

Jan Toodre 123990IASB

**RASPBERRY PI-L PÕHINEV
LIIKUMISTUVASTUSEGA
VIDEOJÄLGIMISSÜSTEEM**

Bakalaureusetöö

Mairo Leier
MSc
Nooremteadur

Tallinn 2015

Autorideklaratsioon

Olen koostanud antud töö iseseisvalt. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud. Käesolevat tööd ei ole varem esitatud kaitsmisele kuskil mujal.

Autor: Jan Toodre

Allkiri:

Kuupäev:

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks on luua Raspberry Pi-1 põhinev liikumistuvastusega videojälgimissüsteem ja uurida selle autonoomse toitmise võimalusi. Antud töös on kirjeldatud riistvara ja tarkvara arendust ning töötavat süsteemi tervikuna. Välja on toodud ka alternatiivseid lahendusi autori poolt arendatavale süsteemile. Kuna antud süsteem on tsentraliseeritud ja seda on esimeses versioonis võimalik kasutada ainult kohtvõrgus, siis käsitletakse ka süsteemi ülesseadmist ning sellega tekkinud probleeme. Töö lõpus on võimalik näha arendatud kasutajaliidest ning kirjeldust selle kohta, mida süsteem on võimeline tegema. Tööga on kaasas arendatud lähtekood CD plaadil.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 45 leheküljel, 4 peatükki, 17 joonist, 10 tabelit.

Abstract

Raspberry Pi based surveillance system with motion detection

The goal of this thesis is to develop a Raspberry Pi based surveillance system with motion detection. Nowadays the popularity of off-grid systems is rising hence the possibility of powering Raspberry Pi with solar panel is being studied.

The thesis will present several hardware and software solutions. Zoneminder and iSpy will be briefly covered on the software side. Raspberry Pi models A and B and Cubieboard 2 will be compared to each other in order to determine which one is best suited for this application.

Following best practices the software was designed before development commenced. Figures indicate how system components communicate with each other and how the directory tree looks like while briefly explaining the content of those folders.

The implementation of this software project is briefly explained. At first, how hardware and software systems were initialized. Ways of making this system off-grid are studied and whether or not it is reasonable to implement this functionality. The most time consuming processes in developing the system were initializing the software and hardware and programming. The outcome of this work is a video surveillance system with motion detection. It has an user friendly web interface and the whole project is open source. The thesis will briefly cover problems encountered and how they were solved. This thesis includes a CD with the source code of this software project.

The thesis is written in Estonian and contains 45 pages of text, 4 chapters, 17 figures, 10 tables.

Lühendite ja mõistete sõnastik

1080p	Komplekt HDTV kõrglahutusga videorežiime, mida iseloomustab 1080-realine pilt ja täiskaadrilaotus
720p	Komplekt HDTV kõrglahutusga videorezime, mida iseloomustab 720-realine pilt ja täiskaadrilaotus
Apache2	Vabavaraline veebiserver
bash	<i>Bourne Again Shell</i> – Linux-i käsuprotsessor
Bootstrap	Veebileidete arendamiseks mõeldud raamistik
C	Bell Labs-i poolt välja töötatud programmeerimiskeel
Chart.js	Javascripti teek, millega on võimalik moodustada veebilehel graafikuid
CPU	<i>Central Processing Unit</i> – protsessor
CSS	<i>Cascading Style Sheets</i> – määrab HTML dokumendi kujunduse
daemon	Taustal töötav protsess
DC	<i>Direct Current</i> – alalisvool
dd	Tööriist, mille peamiseks ülesandeks on failide kopeerimine ja konverteerimine Linux tüüpi operatsioonisüsteemides
DMI	<i>Desktop Management Interface</i> – standardne raamistik arvuti komponentide haldamiseks
dmidecode	Tööriist, millega viiakse DMI tabeli sisu inimesele loetavale kujule
Ethernet	Kohtvõrkudes kasutatav standardne OSI mudeli füüsilise kihi ja lülikihi tehnoloogia
FFmpeg	Tööriist, millega on võimalik salvestada, konvertida ja voogedastada audiot ja videot
fps	<i>Frames per second</i> – kaadrit sekundis
GPIO	<i>General Purpose Input/Output</i> – üldotstarbelised sisend- ja väljundviigud, mille tegevust saab kasutaja kontrollida programmi käitusajal
Gpio	Raspberry Pi-le mõeldud programm GPIO haldamiseks
GPU	<i>Graphics Processing Unit</i> – graafika protsessor
H.264	Populaarne videokodeki standard
Handbrake	Tööriist, millega on võimalik videosid transkodeerida

HDMI	<i>High-Definition Multimedia Interface</i> – kõrglahutusega multimeediumiliides. Digitaalliides audio- ja videosignaali edastamiseks
HTML	<i>HyperText Markup Language</i> – hüpertekst-märgistuskeel on tekstivorming veebidokumentide loomiseks
I2C	<i>Inter-IC bus</i> – kahe-suunaline järjestiksiin, mida kasutatakse integraalskeemidevahelise ühenduslülina
I2S	<i>Inter-IC Sound</i> – järjestiksiin, mida kasutatakse digitaalaudioseadmete ühendamiseks
IP-aadress	Internetiaadress
IR	<i>Infrared</i> – infrapunakiirgus
javascript	Objektorienteeritud programmeerimiskeel, mida kasutatakse peamiselt veebilehtede skriptimiseks
LED	<i>Light Emitting Diode</i> – valgusdiod
lshw	Tööriist, millega on võimalik kuvada detailset informatsiooni arvuti riistvarast
M-JPEG	<i>Motion JPEG</i> – videovorming, mis kasutab jpeg pilditihendust iga kaadri jaoks
MMC	<i>MultiMediaCard</i> – multimeedia kaart, kasutatakse mobiilseadmetes
MP4	Multimeedia konteinervormingu standard
MP4Box	Tööriist, millega on võimalik sooritada erinevaid multimeedia failidega seotud operatsioone
MySQL	Avatud lähtekoodiga andmebaasihaldur
NAS	<i>Network-Attached Storage</i> – võrgumälu, mis on ühendatud arvutivõrku
NFS	<i>Network File System</i> – võrgu-failisüsteem
nimega toru	<i>Named pipe</i> – laiendus torule Unix tüüpi operatsioonisüsteemides
Notepad++	Vabavaraline lähtekoodi redaktor Windows-i keskkonnas
ogg	Avatud lähtekoodiga, kadudega audiotihendamise tehnoloogia
PHP	<i>Hypertext Preprocessor</i> – skriptimiskeel, mida kasutatakse peamiselt serveripoolsetes lahendustes dünaamiliste veebilehtede loomisel
PIR sensor	<i>Passive Infrared sensor</i> – elektrooniline sensor, mis mõõdab infrapunakiirgust kiirgavaid objekte oma vaateväljas
PoE	<i>Power over Ethernet</i> – elektritoide üle võrgu
PuTTY	Vabavaraline telneti ja SSH klientprogramm
RAM disk	Muutmälu paiknev andmeruum, mis käitub kui kõvaketas

raspimjpeg	Avatud lähtekoodiga programm, millega saab voogedastada ning salvestada videosid ja pilte
raspistill	Raspberry Pi-le mõeldud käsurea programm, salvestamiseks pilte kaameramooduliga
raspivid	Raspberry Pi-le mõeldud käsurea programm, salvestamiseks videot kaameramooduliga
RCA connector	RCA konnektor – pistik, mida kasutatakse audio- ja videoseadmete omavaheliseks ühendamiseks
RJ-45	<i>Registered Jack - 45</i> – RJ-45 konnektor on mõeldud andmesideks ethernet võrkudes
Samba	Tarkvara, mis võimaldab Unix-i serveril töötada failiserverina Windows-i klientide jaoks
SD	<i>Secure Digital</i> – väikmälukaart
SoC	<i>System on a Chip</i> – elektroonikas ühte kiipi mahutatud terviklik toode ehk süsteemikiip
SOC	<i>State of Charge</i> – aku täituvusprotsent
SPI	<i>Serial Peripheral Interface bus</i> – sünkroonse jadatühendusega kommunikatsiooni liides
SQL	<i>Structured Query Language</i> - struktuuripäringukeel
SSH	<i>Secure Shell</i> – mitmeplatvormiline turvalist krüpteeritud kaugpöördust võimaldav protokollisari
sudo	Programm Unix tüüpi operatsiooni süsteemidele, mis võimaldab programme käivitada teise kasutaja õigustega
toru	<i>Pipe</i> – protsessi üldotstarbeline andmevahetusvahend Unixi-laadsetes opsüsteemides
tree	Tööriist kataloogipuu kuvamiseks Unix tüüpi operatsioonisüsteemides
UART	<i>Universal Asynchronous Receiver/Transmitter</i> – universaalne asünkroontransiiver
UBEC	<i>Universal Battery Elimination Circuit</i> – pingemuundur, mis teeb kõrgemast pingest madalama pinge
Ubuntu LTS	Debian-i baasil Linux operatsioonisüsteem, millel on pikaajalisem kasutajatugi kui tavaversioonil (Long Term Service – LTS)
USB 2.0	<i>Universal Serial Bus 2.0</i> – universaalne jadaliides 2.0
vi	Ekraanile orienteeritud tekstiredaktor
VLC	VideoLAN organisatsiooni poolt välja töötatud meediamängija
WebM	<i>Webmedia</i> – multimeedia konteinervormingu standard

Webmin	Veebiliidesega administratiivne tööriist Unix tüüpi operatsioonisüsteemidele
WD	<i>Western Digital</i> – kõvakettaid tootev firma
Wi-Fi	<i>Wireless-Fidelity</i> – Etherneti kasutatav kohtvõrk, kus kaablid on asendatud raadiolinkidega
Wi-Fi <i>dongle</i>	Järjestik- või USB porti ühendatav riistvara seade
wiringPi	Teek, mille abil hõlbustatakse GPIO ligipääsu. Mõeldud kasutamiseks C programmeerimiskeeles

Sisukord

1. Sissejuhatus	12
2. Süsteemi ülevaade	13
2.1. Nõuded süsteemile	13
2.2. Komponentide valik	13
2.2.1. Tarkvara	13
2.2.2. Riistvara	16
2.3. Alternatiivsed lahendused	20
2.3.1. Tarkvara	20
2.3.2. Riistvara	20
2.3.3. Terviksüsteemid	22
3. Süsteemi struktuur	24
3.1. Tarkvara	24
3.1.1. Kaustade struktuur	27
3.2. Riistvara	28
4. Teostus	29
4.1. Süsteemi lähtestamine	29
4.2. Riistvara seadistus	29
4.2.1. Reaalne süsteem	29
4.2.2. Süsteemi autonoomne toide	30
4.2.3. Andmeedastus Wi-Fi kaudu	33
4.3. Tarkvaraarendus	35
4.3.1. Arendatud tarkvara	35
4.3.2. Probleemid	36
4.3.3. Töö käik	37
4.4. Rakendus	38
5. Edasiarenduse võimalused	42
6. Kokkuvõte	43
Kasutatud kirjandus	44

Jooniste nimekiri

Joonis 1. Raspberry Pi mudel B [11].	17
Joonis 2. GPIO viigud.	17
Joonis 3. Raspberry Pi kaameramoodul.	18
Joonis 4. Süsteemi struktuur.	25
Joonis 5. Kasutajajuhtude diagramm.	26
Joonis 6. Serveri /export kataloogipuu.	27
Joonis 7. Süsteemi topoloogia.	28
Joonis 8. PoE teostus.	30
Joonis 9. PIR sensori ühendamine Raspberry Pi-ga.	30
Joonis 10. Wi-Fi <i>dongle</i> 'i võrgu testimise skript.	33
Joonis 11. Veebiliides – Pealeht.	38
Joonis 12. Veebiliides – Otseedastus.	39
Joonis 13. Veebiliides – Failihaldur.	39
Joonis 14. Veebiliides – Salvestatud pildid.	40
Joonis 15. Veebiliides – Arhiveeritud video.	40
Joonis 16. Veebiliides – Logid.	41
Joonis 17. Veebiliides – Päevapõhine liikumistuvastuste statistika.	41

Tabelite nimekiri

Tabel 1. Video transkodeerimistestide tulemused.....	15
Tabel 2. Video transkodeerimisparameetrite tähendused.....	16
Tabel 3. Raspberry Pi kaameramooduli tehnilised parameetrid.....	18
Tabel 4. PIR sensori tehnilised parameetrid.....	19
Tabel 5. Serveri tehnilised parameetrid.....	19
Tabel 6. Ühest trükkplaadist koosnevate arvutite võrdlus.....	21
Tabel 7. Raspberry Pi riistvara komponentide volutarve.....	31
Tabel 8. Raspberry Pi autonoomse toite arvutused.....	32
Tabel 9. Wi-Fi dongle'i võrgu testide tulemused.....	34
Tabel 10. Arendusel kasutatud programmeerimis- ja skriptimiskeeled.....	35

1. Sissejuhatus

Maakodust eemal olles tahaks siiski koduõuel silma peal hoida. Mõne õnnetuse või varguse korral oleks hea omada pildimaterjali sellest, kes või mis kahju tekitas. Nii tekkiski käesoleva töö autoril vajadus videovalvesüsteemi loomise järele.

Saadavalolevaid videovalvesüsteeme uurides on ostjal kaks valikut: kas osta odav või kallis süsteem. Odava eeliseks on see, et esmane väljaminek on väiksem, kuid süsteem ei pruugi rahuldada kasutaja nõudeid. Kalli süsteemi ostmisel on tootja pakutud tarkvara ja riistvara tavaliselt paremad, kuid mõlema variandi puhul on edasiarendusvõimalused piiratud. On siiski ka kolmas võimalus – valvesüsteem ise võimalikult soodsalt luua.

Odava ja paindliku valvesüsteemi arendamiseks on võrdlemisi lihtne kasutada Raspberry Pi-l põhinevat süsteemi. Praeguseks hetkeks on müüdud [1] üle 5 miljoni Raspberry Pi [2]. See on krediitkaardi suurune arvuti, mida on võimalik kasutada nii personaalse arvutina kui ka automaatikaprojektide elluviimiseks.

Raspberry Pi baasil tehtud turvasüsteemi põhiliseks eeliseks on see, et süsteemi on hiljem võimalik liidestada teiste süsteemidega ning sellele saab lisada teisi funktsioone, näiteks temperatuuri mõõtmine, õhuniiskuse jälgimine jm.

Töö teema valik tuleneb vajadusest ja huvist uurida koduautomaatika süsteemi arendust Raspberry Pi baasil. Eesmärk on luua süsteem, millega on võimalik tuvastada liikumist, salvestada videosid ja pilte, vaadata salvestatud videosid ning jälgida kaamerate pilti reaalajas. Lisaks testiti Raspberry Pi-ga juhtmevaba andmeside kasutamise võimalusi. Veel uuriti Raspberry Pi autonoomset toitmise võimalust päikesepaneeliga. Loodava süsteemi kasutamine peaks olema lihtne ja intuitiivne.

2. Süsteemi ülevaade

2.1. Nõuded süsteemile

Eesmärgiks on luua süsteem, millega on võimalik:

- salvestada pilte ja videosid;
- tuvastada liikumist;
- vaadata kaamerapilti reaalajas;
- vaadata salvestatud pilte, videosid.

Süsteemil peab olema graafiline kasutajaliides, mida on võimalik väikese vaevaga kasutama õppida. Süsteemile peab saama funktsioone juurde lisada. Süsteem olgu tsentraliseeritud, kus juhtüksuseks on server ja alamüksusteks Raspberry Pi-d. Kasutajaliidesele peab olema ligipääs kohtvõrgus. Süsteemi esimeses versioonis puudub kasutaja autentimise võimalus, mistõttu on laivõrgust ligipääs keelatud.

2.2. Komponentide valik

2.2.1. Tarkvara

Raspberry Pi

Raspberry Pi operatsioonisüsteemiks valiti Debian Wheezy [3].

Selleks et süsteemi nõudeid täita, valiti videote ja piltide voogedastuseks ning salvestamiseks *raspimjpeg*-i [4]. Et tegemist on programmiga, millel on avatud lähtekood, siis on võimalik seda vastavalt oma vajadustele kohandada.

Server

Serveri operatsioonisüsteemiks valiti Ubuntu Server 12.04 LTS. Otsus eelnimetatu kasuks on põhjendatud sellega, et Ubuntu on üks laialt levinumaid Linux-i distributsioone, sellel on hea kasutajatugi ja dokumentatsioon.

Videote arhiveerimiseks ettevalmistamisel kasutati FFmpeg [5] tarkvara, millega konverteeritakse H.264 formaadis video ümber WebM formaati [6]. Konverteerimise vajadus tuleneb sellest, et HTML5 videomärgendi abil saab kuvada vaid mp4, webm ja ogg tüüpi videosid. Videote konverteerimiseks prooviti kolme eri tarkvara paketti: FFmpeg, Handbrake [7] ja MP4Box [8]. Leidmaks sobivaimat tööriista antud töö jaoks, tuli neid testida. Handbrake langes valikust välja oma keerukuse tõttu. Dokumentatsiooni järgides sisestati kõige lihtsam konverteerimise käsk, mille tulemusel tagastas programm konverteerimist takistavad veateated. Hiljem katsetati ka MP4Box-i mis oli küll kasutajasõbralik, kuid mille otstarbekusest puudu jäi. Nimelt on MP4Box-il mõningad piirangud, sealhulgas video transkodeerimise toe puudumine, mis tähendab, et toetatud sisend- ja väljundfailide formaate on loetud arv. Seega osutus valituks FFmpeg, mis on väga paindlik oma suure hulga konfigureerimisvõimaluste poolest.

Erinevad katsetatud konfigureerimisvõimalused ja sätete tulemused on välja toodud tabelis 1. Võrdluseks on tabeli lõppu lisatud MP4Box-i konverteerimise tulemus. Testimisel kasutati ühe minuti pikkust, 22,3MB suurust H.264 formaadis sisendfaili. Tabelis 1 on välja toodud programmi käsuparameetrid, transkodeerimisele kulunud aeg, video väljundfaili maht, autori subjektiivne kvaliteedihinnang ja kommentaarid ning transkodeerimise tulemit saadud väljundfaili mahu ja programmi tööaja suhted võrreldes sisendfaili pikkusega. Märkida tuleb, et neljas katse sooritati soovitatud näitajate [9] järgi. Tabelis 2 on välja toodud testimiseks kasutatud parameetrid ning nende tähendused. Selleks et ruumi säästa, esitatakse ainult programmi parameetrid ning sisend- ja väljundfailide nimed on *in* ja *out* koos faililaienditega. Näidiskäsk, mis konverteerib FFmpeg programmiga H.264 tüüpi sisend videofaili MP4-ks, näeb välja nii:

```
./ffmpeg -i ./1min_testvid.h264 ./defaultparamvid.mp4
```

Tabel 1. Video transkodeerimistestide tulemused.

#	Käsk	Trans- kodeeri- miseks kulunud aeg	Väljundfaili suurus	Kvaliteedi- hinnang	Trans- kodeeri- mise aeg võrreldes sisendfaili pikkusega (kordades)	Väljund- faili suurus võrreldes sisendfaili suurusega (kordades)
1	-r 25 -i in.h264 - vcodec copy out.mp4	1 min 30 s	22,3 MB	Suurepärane	1,5x	1x
2	-i in.h264 -c:v libx264 -b:v 1000k -preset ultrafast out.mp4	1 min 32 s	7,44 MB	Hea	1,53x	0,33x
3	-r 25 -i in.h264 - c:v libx264 -b:v 200k -preset ultrafast out.mp4	1 min 32 s	1,5 MB	Kehv	1,53x	0,07x
4	-i in.h264 -vcodec libvpx -g 120 - rc_lookahead 16 - level 216 -qmax 42 -qmin 10 - maxrate 24M - minrate 100k [9]	2 min 38 s	1,5 MB	Väga hea	2,63x	0,07x
5	-i in.h264 -s 864x468 -vcodec libvpx -r 3 -cpu- used 0 -deadline good out.webm	28 s	1,5MB	Rahuldav	0,47x	0,07x
6	-i in.h264 -vcodec libvpx -cpu used 0 -deadline good out.webm	3 min 6 s	1,5MB	Väga hea	3,1x	0,07x
7	MP4Box -fps 25 - add in.h264 out.mp4	4 min 51 s	22,9MB	Väga hea	4,85x	1,03x

Tabel 2. Video transkodeerimisparameetrite tähendused.

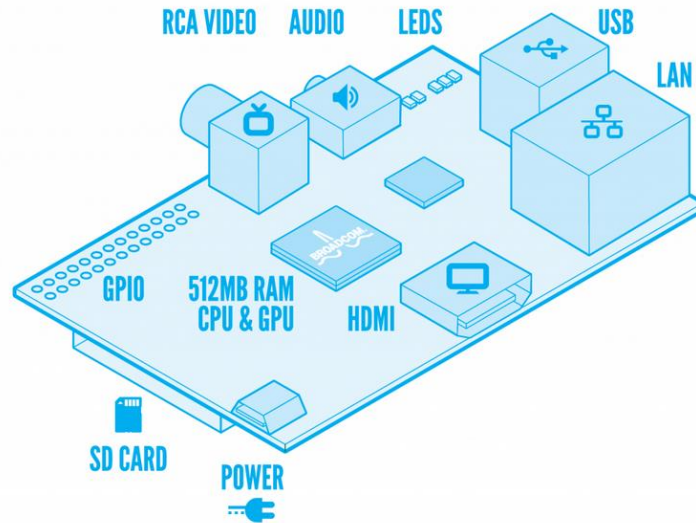
Parameeter	Tähendus
-r	Video kaadrisagedus, fps (käib selle video aadressi ette, millele rakendatakse)
-i	Sisendfaili aadress
-vcodec	Video koodek (libvpx – WebM, libx264 – MP4)
-c:v	Sisuliselt sama mis –vcodec
-b:v	Video bitisagedus
-preset	Eelnevalt valmis tehtud sätete kogum
-g	Keelab pildi liikumise ennustamise ära
-rc_lookahead	Eelvaadatavate kaadrite arv
-level	Dokumentatsioon puudub, tehtud soovitatud parameetrite järgi [9]
-qmax	Maksimaalne kvantimistase
-qmin	Minimaalne kvantimistase
-maxrate	Maksimaalne video bitikiirus
-minrate	Minimaalne video bitikiirus
-s	Video resolutsioon
-cpu-used	Kodeerimiskiirus (suuremate väärtustega kaasnevad kvaliteedi kaod)
-deadline	Kodeerimisele kulutatav aeg

Pidades silmas eelkõige mahtu ning kvaliteeti, võib testi tulemustest järeldada, et neljanda testiga saavutati parim tulemus ning edaspidi kasutataksegi videote arhiveerimiseks just neid parameetreid.

2.2.2. Riistvara

Raspberry Pi

Raspberry Pi B [10] mudeli tehnilised parameetrid on toodud välja tabelis 6. Raspberry Pi põhilisi komponente on võimalik näha joonisel 1. Lõplikku seadme valikut selgitatakse täpsemalt antud töö punktis 2.3.



Joonis 1. Raspberry Pi mudel B [11].

Kasutades wiringPi [12] gpio programmi on võimalik näha GPIO viikude asetust ning programmeerija jaoks vajalikke viikude numbreid. Neid numbreid on vaja PIR sensori programmeerimiseks ja Raspberry Pi-ga ühendamiseks. Süsteemis kasutatud viigud on välja toodud joonisel 2 ning nende füüsilised ühendused joonisel 9.

```

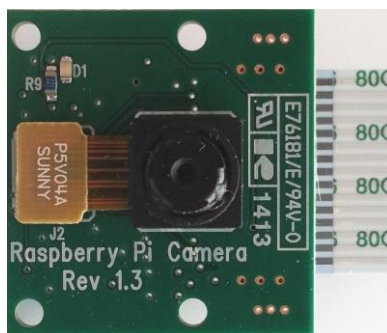
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 8 | SDA.1 | IN | 0 | 3 | 4 | | | 5v | | |
| | | 0v | | | 9 | 6 | | | 0v | | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Joonis 2. GPIO viigud.

Selleks et videot salvestada, on vaja Raspberry Pi-le lisada kaamera. Kasutada otsustati kaameramoodulit [13] (Joonis 3), mille tehnilised andmed on välja toodud tabelis 3. Kaameramooduli valik on põhjendatud sellega, et tavalised veebikaamerad ei ole võimelised edastama samaväärselt või parema kvaliteediga pilti ning professionaalsed kaamerad ei ole mõeldud pidevaks video ja pildi salvestamiseks. Raspberry Pi kaameramooduli hinna ning kvaliteedi suhe tundub autorile kõige optimaalsem. Kui tarkvaralist aspekti arvesse võtta, siis on kaameramoodulil juba olemas tarkvara, millega erinevaid video- ja pildiparameetreid muuta. Kui kasutada mõnda veebikaamerat või

profikaamerat, siis võib tekkida vajadus arendada ka vastav Raspberry Pi-le mõeldud tarkvara.



Joonis 3. Raspberry Pi kaameramoodul.

Tabel 3. Raspberry Pi kaameramooduli tehnilised parameetrid.

Sensori tüüp	OmniVision OV5647
Sensori suurus	3,67 x 2,74 mm
Pikslite arv	2592 x 1944 (5 megapikslit)
Vaatenurk	54 x 41 kraadi
Vaateväli	2,0 x 1,33 m (kahe meetri kaugusel)
Video salvestus	1080p, 30 fps. Koodek H.264 (AVC)
Mooduli suurus	25 x 24 mm (ilma ühenduskaablita)

Lisaks video salvestamisele on vaja, et süsteem oleks võimeline liikumist tuvastama. Selleks jaoks on valitud PIR sensor Jablotron JS-20 Largo [14], mille tehnilised parameetrid on välja toodud tabelis 4. Sensori valikul on silmas peetud eelkõige seda, et riistvara ressursse võiks võimalikult palju säästa. See tähendab, et liikumistuvastuseks on võimalik kasutada ka ainult tarkvaralist lahendust, mis töötleb pilte ning pildil tuvastatud muutuseid, kuid paraku suurendab see protsessori koormust. Seepärast otsustatigi just PIR sensori kasuks.

Tabel 4. PIR sensori tehnilised parameetrid.

Toiteallikas	12V DC \pm 25%
Voolu tarbimine (LED väljas)	max. 10 mA
Voolu tarbimine (LED sees)	max. 35 mA
Paigaldamise kõrgus	2,5 m (maast)
Initsialiseerimise aeg	ca 1 min
Alarmi väljund	normaalselt suletud, max. 60 V / 50 mA
Liikumise tuvastus	120 ° / 12 m (tavalise läätsega)
Töö temperatuur	-10 kuni +40 °C

Server

Serveri otstarve ilmnes varakult, kuna süsteemi üks nõuetest on, et videomaterjali on vaja talletada vähemalt seitse päeva. Iga Raspberry Pi-le kõvaketta lisamine on välistatud, sest kõvakettad pole mõeldud välitingimustes kasutamiseks ning süsteemi maksumus tõuseks iga kõvaketta ostuga oluliselt. Oli vaja välja selgitada, kui palju kõvaketta ruumi kulub seitsme päeva videomaterjali talletamiseks. Mahu suuruse arvutamiseks kasutati hea kvaliteediga 720p video soovitatud väikese varuga bitikiirust [15], milleks on 6 Mbit/s. See korrutati seitsme päeva sekundite arvuga, saades tulemuseks 453,6 GB kõvaketta mahtu ühe Raspberry Pi kohta. Serveri tehnilised parameetrid on välja toodud tabelis 5, kasutades Linux-i programme dmidecode ja lshw.

Tabel 5. Serveri tehnilised parameetrid.

Emaplaat	Gigabyte Z87N-WIFI
Protsessor	Intel(R) Core(TM) i3-4330T CPU @ 3.00GHz
Muutmälu	Micron 4GB DDR3 1600MHz (2tk)
Välismälu	120GB INTEL SSDSC2BW12 3TB WDC WD30PURX-64
Võrk	Ethernet Connection I217-V AR8161 Gigabit Ethernet Wireless 7260

Emaplaadi valikul lähtuti sellest, et hilisemas arendusjärgus võib tekkida olukord, kus on vaja kasutada nii kahte eraldi Ethernet võrguliidest kui ka traadita võrguliidest.

Kõvaketas valiti eelnevaid WD kõvaketaste kogemusi arvestades. Silmas peeti ka seda, et Purple seeria on mõeldud valvesüsteemide andmete salvestamiseks – kõvaketas töötab ja kirjutab andmeid ööpäev läbi.

2.3. Alternatiivsed lahendused

2.3.1. Tarkvara

Tarkvaralise lahendusena on võimalik valida erinevate variantide vahel, millest järgnevalt tuuakse välja need, mida töö autor ise proovis.

Video salvestamiseks on Raspberry Pi-le välja töötatud programmid raspistill ja raspivid, millest esimesega on võimalik teha pilti ning teisega videot salvestada. Nende kahe programmi suurimaks miinuseks on see, et neid ei saa korraga kasutada. Teine puudus nende programmide kasutamisel tekib videoedastusega. Videoedastuseks tuleb suunata raspivid-i väljund toru kaudu mõne teise programmi sisendisse, mis seda infot töötleb. Eelnimetatud programmiks võib olla näiteks VLC meediamängija või FFmpeg, millega on võimalik sisendvideot voogedastada ja salvestada. Lahenduse teeb keerukaks see, et kasutajale tuleb videoedastus kuidagi kättesaadavaks teha. Selleks on võimalik kasutada näiteks VLC meediamängijat, kus tuleb sisestada voogedastuse IP-aadress, port ja teised sätted. See aga on aeganõudev ning keeruline. Teine võimalus on teha voogedastus kättesaadavaks veebiliidese kaudu, mis on oluliselt kasutajasõbralikum, kuid mille ülesseadmine võib jällegi keeruline olla. Eelmainitud puuduste ja keerukuse pärast otsustati valida *raspimjpeg*-i tarkvara, milles on ühendatud video ja pildi salvestamise ning edastamise võimalused. Lisaks on sellega voogedastuse kättesaadavaks tegemine üsna lihtne.

2.3.2. Riistvara

Süsteemi arendama hakates oli vaja leida seade, mis oleks võimeline videosid salvestama. Lähtudes sellest, et seade peab olema konfigureeritav, sellele peab olema võimalik

erinevaid komponente juurde lisada ning arvestades ka koduautomaatika funktsioone, uuriti Eesti turul olevaid ühele trükkplaadile paigutatud arvuteid. Valimisse jäi kolm erinevat arvutit: Raspberry Pi A ja B mudelid ning Cubieboard 2 [17]. Põhilised valikuparameetrid olid võrguliidese olemasolu, mälu maht ja odav hind. Võrguliides pidi olema Ethernet, mida toetasid Raspberry Pi mudel B ja Cubieboard 2. Mälu oli kõige rohkem Cubieboard-il, kuid hind oli pea kaks korda suurem, sellepärast osutus valituks just Raspberry Pi B mudel.

Tabel 6. Ühest trükkplaadist koosnevate arvutite võrdlus.

	Raspberry Pi (A)	Raspberry Pi (B)	Cubieboard 2
SoC	Broadcom BCM2835	Broadcom BCM2835	Allwinner A20
CPU	700 MHz ARM11 ARM1176JZF-S	700 MHz ARM11 ARM1176JZF-S	1 GHz ARM Cortex-A7
GPU	Broadcom VideoCore IV	Broadcom VideoCore IV	Mali-400MP2
Mälu	256 MiB	512 MiB	1 GB + 4GB Flash
USB	1x USB 2.0	2x USB2.0 (läbi integreeritud USB jaoturi)	2x USB2.0
Video väljund	Komposiitvideo (RCA), HDMI	Komposiitvideo (RCA), HDMI	HDMI
Audio väljund	3.5mm, HDMI	3.5mm, HDMI	3.5mm, HDMI
Mäluseade	SD / MMC mälukaart	SD / MMC mälukaart	microSD mälukaart, SATA 2.0 kõvaketas
Võrk	–	10/100 juhtmega Ethernet RJ-45	10/100 juhtmega Ethernet RJ-45
Madala taseme väliseaded	26 GPIO viiku (SPI, I2C, I2S, UART)	26 GPIO viiku (SPI, I2C, I2S, UART)	96 GPIO viiku (SPI, I2C, ADC, ...)[16]

Tabel 6. Ühest trükkplaadist koosnevate arvutite võrdlus.

Voolutarve	300 mA (1.5W)	700 mA (3.5W)	<800 mA (<4W)
Toite allikas	5V DC läbi B tüüpi Micro USB või GPIO	5V DC läbi B tüüpi Micro USB või GPIO	5V DC
Mõõtmed	85.0 mm x 56.0 mm x 15 mm	85.0 mm x 56.0 mm x 17.0 mm	100 mm x 60 mm x 23 mm
Kaal	31 g	40 g	45 g
Hind [18]	33,6 €	42 €	78 €

2.3.3. Terviksüsteemid

Selles punktis tutvustatakse lühidalt kahte süsteemi, millega saab turvakaamerate süsteemi hallata.

iSpy [19], mille lööklause *iSpy turns your PC into a full security and surveillance system* tähendab seda, et kasutaja personaarvuti on kõikide paigaldatud kaamerate haldusvahend. Tarkvara on avaliku lähtekoodiga vabavara. Antud lahendusel on väga palju funktsioone, millest võiks välja tuua järgmise: liikumis- ja helituvastus, videote salvestamine ning üleslaadimine YouTube'i jpm. Süsteem lubab endaga ühendada ükskõik kui palju kaameraid ja mikrofone. Süsteemi suurimaks puuduseks on tõsiasi, et selle haldamiseks kasutatakse personaarvutit. See tähendab seda, et kasutaja arvuti peab kogu aeg töötama, selleks et vastavalt konfiguratsioonile videot ja pilti salvestada. Ka toetab antud süsteem paraku ainult Windows-i operatsioonisüsteeme, mis kõrvaldab võimaluse süsteemi Linux-is kasutada. Süsteemi suureks plussiks on suur vahendite valik ning võimalus võrdlemisi lihtsalt erinevaid mooduleid süsteemi integreerida.

Zoneminder [20] on samuti videojälgimissüsteemi jaoks mõeldud vabavara. Antud tarkvaralahendus on oma vahendite valiku poolest küll väiksem, kuid seda on võimalik installeerida Linux-it jooksutavale serverimasinale. Zoneminder hõlmab mitme kaamera süsteemi integreerimist, liikumistuvastust, veebiliidest jpm. Kasutades Zoneminder-it koos raspivid ja VLC meediamängijaga on võimalik rahuldada tavakasutaja

videojälgiimissüsteemi vajadusi: liikumistuvastus, videote salvestamine jms. Selle ja eelmise tarkvaralahenduse üheks suurimaks puuduseks on kaamera pildisätete muutmise – pärast video voogedastuse ülesseadmist kaamera sätteid enam muuta ei saa.

Lähtudes vajadusest kaamera sätteid ja videote talletamise süsteemi konfigureerida, otsustati kasutusele võtta *raspimjpeg*-i. Selle ümber saab ehitada käepärase videojälgiimissüsteemi, mille kasutajaliidesele on edaspidi võimalik teisigi süsteeme integreerida

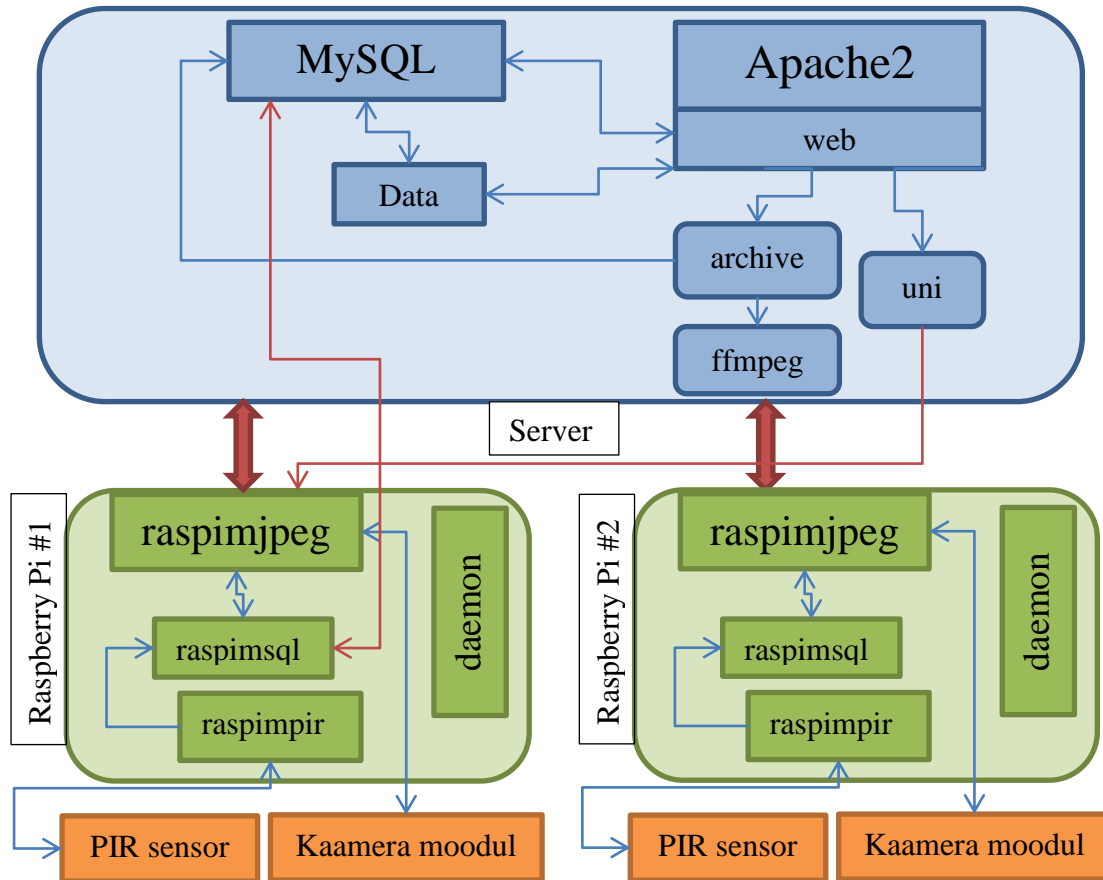
3. Süsteemi struktuur

3.1. Tarkvara

Arendatud tarkvara struktuur ja komponentide omavaheline suhtlus on välja toodud joonisel 4. Serveris on üles seatud MySQL andmebaas ja Apache2 server, mille abil toimib veebiliides (web). Veebiliides suhtleb otse andmebaasiga ning teatud tüüpi andmete jaoks on loodud eraldi Data andmemoodul. Selle ülesandeks on teha vastavalt identifikaatorile soovitud andmete kohta päring andmebaasi ning viia saadud tulemus õigesse formaati selleks, et seda kasutajale kuvada. Andmemooduli üheks päringunäiteks on kasutajale liikumisstatistika kuvamine. Andmebaasist küsitakse liikumistuvastuste arvu, keskmist liikumise kestvust ning aega, millal liikumine tuvastati. Saadud andmed viiakse sellisele kujule, et Chart.js [21] suudaks kasutajale liikumiste graafiku joonistada. Veebiliidese kaudu on kasutajal võimalik valitud video arhiveerida. Nupuvajutusel käivitatakse arhiveerimise skript (*archive*), mis teisaldab algvideo *perm* kataloogi, vältimaks seda, et poole arhiveerimise pealt vanad videod automaatselt kustutatakse. Seejärel alustatakse FFmpeg programmiga video konverteerimisega WebM formaati, mille lõppedes tehakse andmebaasi kirje, et video on arhiveeritud. Algne videofail tõstetakse tagasi *temp* kausta, et hiljem oleks võimalik video ära kustutada. Kui kasutaja soovib voogedastuse sätteid muuta või pilti teha, siis käivitatakse selleks universaalne skript (*uni*), millega saadetakse valitud Raspberry Pi-le käsk koos parameetritega.

Raspberry Pi koosneb kahest riistvara komponendist: sensor ja kaamera, mis on joonisel 4 märgistatud oranžilt. Peamine tarkvara komponent on *raspimjpeg*, mis kontrollib kogu piltide ja videote voogedastus- ja salvestustegevust. *Raspimjpeg*-i külge on integreeritud *raspimpir* ja *raspimsql* komponendid, kus *raspimsql* täidab kogu *raspimjpeg* ja MySQL serveri vahelist suhtlust. PIR sensor lähtestatakse *raspimpir*-is ning seejärel kasutatakse sensorist tulevaid signaale liikumise tuvastuseks ning liikumise kestvuse arvutamiseks, misjärel edastatakse saadud andmed *raspimsql*-i andmebaasi salvestamiseks. *Raspimjpeg* lähtestab kaamera ning muudab hiljem kaamera pildi- ja kvaliteedisätteid. Daemon komponent on selleks, et *raspimjpeg* kogu aeg töötaks. Kui mingil põhjusel on *raspimjpeg*

oma töö lõpetanud, siis taaskäivitatakse see. Raspberry Pi-de arv süsteemis ei ole piiratud, kuid kui neid juurde lisada. Tuleb arvestada sellega, et videote salvestamiseks jääb vähem ruumi ja seega tuleb muuta vanade videote automaatse kustutamise sätteid.



Joonis 4. Süsteemi struktuur.

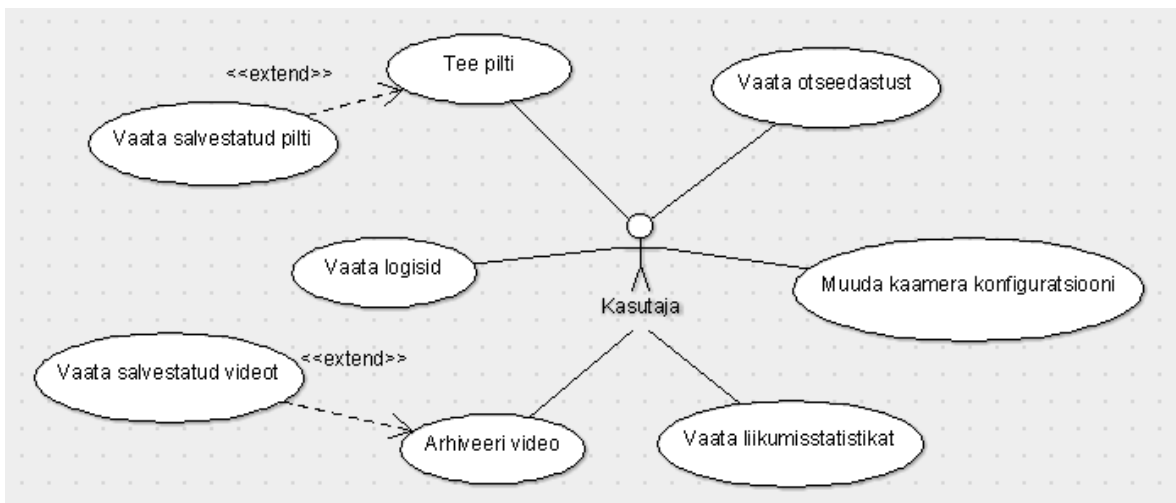
Järgnevalt on kirjeldatud seda, mis toimub videoga selle salvestamisest alates ning mida saab kasutaja salvestatud videoga teha:

- *Raspimjpeg* alustab uue video salvestamisega. Videost salvestatakse esimene kaader pildina, et oleks olemas eelvaade ning *raspimsql* teeb andmebaasi päringu uuest videost.
- Toimub liikumistuvastus, mõõdetakse tuvastatud liikumise aega (sekundites) ning liikumise lõppedes tehakse andmebaasi päring ning salvestatakse liikumise andmed.
- 15 minutit pärast seda, kui videot salvestama hakati, lõpetatakse antud video salvestamine ning alustatakse uuega. Uue video kohta tehakse andmebaasi päring

ning uuendatakse infot varem salvestatud video jooksul toimunud liikumiste arvu ja kestvuse kohta.

- Kui salvestatud H.264 tüüpi videofail on vanem kui 7 päeva, siis video kustutatakse ning tehakse päring andmebaasi.
- Kasutaja suundub veebiliidest kasutades failihalduri lehele, nagu näidatud joonisel 13, ning vajutab video juures olevat „Archive“ nuppu. Olgu ära mainitud, et failihalduris on näha ainult kustutamata H.264 tüüpi videosid ning arhiveerida saab ainult neid, mille salvestamine on lõpetatud.
- Süsteem käivitab *archive* skripti, mille ülesannet on selles peatükis juba kirjeldatud.
- Kui skript on töö lõpetanud, tekib pealehele teavitust, et video on arhiveeritud. Nüüd saab kasutaja liikuda salvestatud videote lehele (Joonis 15), kust ta arhiveeritud video leida võib.

Süsteemi esimese versiooni kõik kasutajajuhud on toodud joonisel 5. Antud töö punktis 4.4 on võimalik tutvuda valminud rakendusega, kus kasutaja saab joonisel välja toodud tegevusi teha .



Joonis 5. Kasutajajuhtude diagramm.

3.1.1. Kaustade struktuur

Server

Andmaks lugejale selgemat ülevaadet kaustade asukohast ning nende sisust on süsteemi kaustade struktuur välja toodud tree programmi kasutades. Käsk „tree -d /export“ väljundiks (Joonis 6) on /export kausta kõik alamkaustad ja sümbolnimed. *Perm* ja *temp* kaustades talletatakse videosid ja pilte vastavalt sellele, kas neid varundatakse ajutiselt või pikaajaliselt. Kaust *scripts* sisaldab kõiki süsteemis kasutatavaid skripte ning need, mida veebiliidesel vaja on, on paigutatud *cmd* alamkausta. Kogu veebiliides ja sellega seonduv on paigutatud *www* kausta. Sümbolnimesid on kasutatud selleks, et programmeerija saaks lihtsamalt ligi kasutadele ja failidele, mis on kõrgemal kataloogide hierarhiatasemel.

```
/export
├─ ramdisk
└─ rpi
    └─ system
        ├─ perm
        │   └─ img
        │   └─ vid
        ├─ scripts
        │   └─ cmd
        ├─ temp
        │   └─ img
        │   └─ vid
        └─ www
            ├─ css
            ├─ fonts
            ├─ img
            │   └─ img -> /export/rpi/system/temp/img/
            ├─ js
            ├─ scripts -> /export/rpi/system/scripts/cmd
            ├─ vid -> /export/rpi/system/temp/vid/
            └─ vidarch -> /export/rpi/system/perm/vid/
```

Joonis 6. Serveri /export kataloogipuu.

Lühidalt kokku võttes hoiab *export* kaust endas kõiki kaustasid, kuhu on võimalik Raspberry Pi-l ligi saada. Kaustas *ramdisk* hoitakse kõige uuemat voogedastuse pilti.

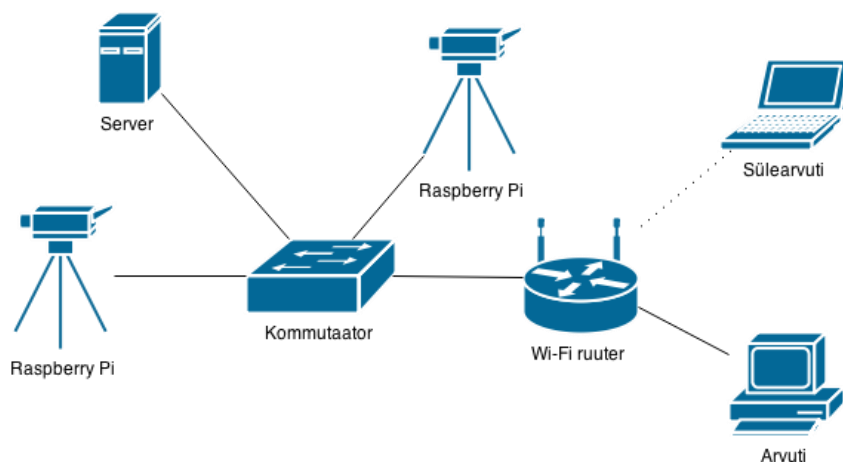
Raspberry Pi

Raspberry Pi kasutab eelmainitud *export* kausta selleks, et sinna videod ja pildid salvestada. Lisaks *export* kaustale ja selle sisule on Raspberry Pi-l veel olulisi katalooge, kus asuvad süsteemi toimimiseks vajalikud programmid ja konfiguratsioonifailid.

Raspimjpeg asub */opt/vc/bin* kaustas, mille sümbolnimi viitab */usr/bin* kaustale. Sümbolnime */usr/bin* kausta on vaja selleks, et programmi oleks võimalik käivitada ükskõik millises kaustas viibides. *Raspimjpeg* kasutab algset konfiguratsioonifaili nimega *raspimjpeg*, mis asub */etc* kataloogis.

3.2. Riistvara

Tarkvara realiseerides on eelduseks see, et riistvara on füüsiliselt omavahel ühendatud. Arendatava süsteemi riistvaraline topoloogia on kujutatud joonisel 7. Raspberry Pi-d ja server on ühendatud kommutaatoriga, mis on omakorda ühendatud Wi-Fi ruuteriga kasutades Ethernet võrku. Kasutajad saavad veebiliidesele ligi, kui ühenduvad kohtvõrku kasutades Etherneti või ruuteri toetatud Wi-Fi tehnoloogiat.



Joonis 7. Süsteemi topoloogia.

4. Teostus

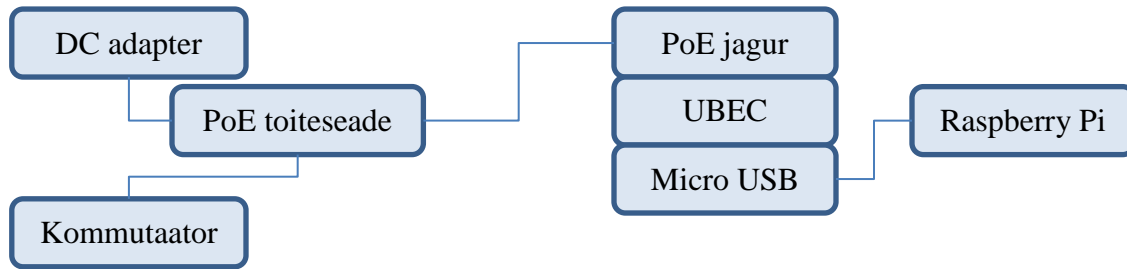
4.1. Süsteemi lähtestamine

Enne tarkvaraarenduse alustamist on vaja süsteem lähtestada. Alustati Raspberry Pi-le ja serverile operatsioonisüsteemide installeerimisega. Seejärel installeeriti Webmin [22] veebiliideselega süsteemi administreerimistööriist. Selle tööriista kasulikkus ilmneb selles, et kui kasutaja pole eelnevalt Linux-iga kokku puutunud, on tal sellega võimalik väikese vaevaga vajalikke sätteid leida ning muuta. Pärast tarkvara installeerimist sätestati SSH abil seadmetesse ühendamine ning alustati võrgu ülesseadmisega. Võrgu IP-aadress on 192.168.1.0, millest serverile eraldati IP-aadressi lõpuga 30 ning Raspberry Pi-dele IP-aadressid lõpuga 32 ja 33. Staatilise IP-aadress ülesseadmine on hädavajalik selleks, et toimiks NFS. Seejärel sätestati Apache2 veebiserver ja MySQL andmebaas, et saaks alustada veebiliidese arendusega. Kuna osa arendust toimub Windows-i keskkonnas, siis otsutas autor silumisele kaasa aitamiseks kasutada Samba serverit, millega on võimalik hallata serveri kõvakettal olevate kaustade sisu. Pärast NFS ja Samba ülesseadmist loodi sümbolnimed, et veebiliidese kaudu saaks ligi erinevatele kaustadele ilma, et peaks kasutama absoluutset faili või kausta asukohta.

4.2. Riistvara seadistus

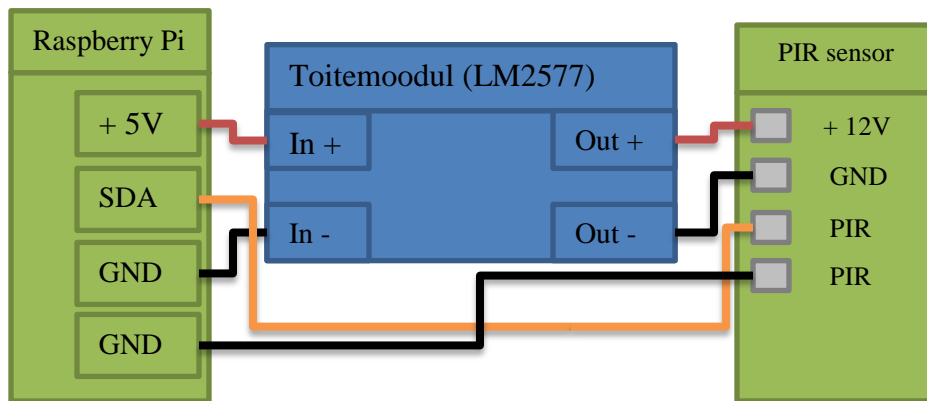
4.2.1. Reaalne süsteem

Selleks, et toita Raspberry Pi-d PoE tehnoloogiat kasutades, tuleb soetada UBEC, DC-DC konverter, PoE toitesead ja jagur, DC adapter ning Ethernet kaabel. Nende komponentide ühendamist illustreerib joonis 8 [23]. Kuna Raspberry Pi-l puudub otsene PoE tugi, siis tuleb ehitada PoE abil toitmiseks vastav süsteem. DC adapteriga suunatakse seinakontaktist vool Ethernet kaablisse kasutades PoE toitesöötjat. Toitesöötja teine ots on ühendatud kommutaatoriga, et luua andmeside. Sihtpunktis eraldatakse toide ning andmeside PoE jaguriga, kus toide läheb UBEC-isse, mis teeb kõrgemast pingest Raspberry Pi-le vajaliku 5 V pinge ning mis on ühendatud Micro USB kaabliga Raspberry Pi külge.



Joonis 8. PoE teostus.

Kui on soov süsteemiga Jablotron JS-20 Largo liikumisandurit kasutada, siis tuleb soetada sensor ning selle toitmiseks vajalik toitemoodul LM2577, mis teeb 5 V toitepinge PIR sensorile vajalikuks 12 V pingeks. Komponentide omavahelisi ühendusi on näha joonisel 9, Raspberry Pi GPIO viikude asetuses veendumiseks vaadata joonist 2.



Joonis 9. PIR sensori ühendamine Raspberry Pi-ga.

4.2.2. Süsteemi autonoomne toide

Autonoomsele süsteemile seati järgmised nõuded ja kitsendused:

- süsteem peab olema võimeline töötama laadimiseta vähemalt 7 päeva järjest
- süsteemil on soovitatav akutäituvus minimaalselt 40%
- energiakaod on 20 %
- kasutatakse 12 V akut

- arvutusteks kasutatakse SolarWorld Sunmodule Plus SW 250 Vario poly [24] päikesepaneeli
- süsteem kasutab kohtvõrgus andmevahetuseks juhtmevaba ühendust (Wi-Fi *dongle*)

Raspberry Pi riistvarakomponentide voolutarbimist mõõdeti Rhode&Schwarz HMP2030 programmeeritava toiteallikaga. Mõõtetulemused on välja toodud tabelis 7. Mõõtmiste eesmärgiks on teada saada komponentide voolutarve ning hiljem vastavad arvutused teha, et selgeks teha, kas süsteemi on võimalik autonoomselt toita ning kas see ennast ära tasub.

Tabel 7. Raspberry Pi riistvara komponentide voolutarve.

Komponendid	Voolutarve (mA)
Raspberry Pi	314
Wi-Fi <i>dongle</i>	102
PIR sensor (LED sees)	43
PIR sensor, liikumistuvastus (LED sees)	114
PIR sensor (LED väljas)	44
PIR sensor, liikumistuvastus (LED väljas)	36

Kasutades saadud mõõtetulemusi ning süsteemi nõudeid, arvutatakse välja kui suure mahutavusega akut on süsteemile vaja ning kas välja pakutud päikesepaneel on piisav, et rahuldada süsteemi energiavajadus Eesti kõige päikesevaesemal kuul, detsembris. Arvutustel kasutame SolarGIS [25] pakutavaid andmeid Tallinna kohta, kasutades 250-vatist päikesepaneeli. Andmete saamiseks on kasutatud taastuvenergia kodulehe materjali [26]. Arvutusi tehes tuleb arvestada seda, et süsteemil on energiakaod. Suurimad kaod tekivad pingete konverteerimisel ning energia akusse salvestamisel (laadimise efektiivsus). Sellest tulenevalt arvestatakse maha energiakaod, mille eeldatavaks suurusjärguks on 20%.

Tabel 8. Raspberry Pi autonoomse toite arvutused.

Nimetus	Tähistus	Valem	Tulemus	Ühik
Raspberry Pi toitepinge	V_S	–	5	V
Aku toitepinge	V_{Aku}	–	12	V
Energia kaod	C_{Kaod}	–	20	%
Minimaalne aku täituvus	C_{SOC}	–	40	%
Mõõdetud voolutarve	I_M	–	0,531	A
Kaamera mooduli voolutarve [27]	I_{Cam}	–	0,250	A
Raspberry Pi võimsustarve	P_{RPi}	$\frac{(I_M + I_{Cam}) * V_S}{C_{Kaod}}$	4,88	W
Raspberry Pi ööpäevane võimsustarve	$P_{ÖÖP}$	$P_{RPi} * 24h$	117,12	Wh
Raspberry Pi voolutarve	I_{RPi}	$\frac{P_{RPi}}{V_{Aku}}$	0,407	A
Raspberry Pi ööpäevane voolutarve	$I_{ÖÖP}$	$I_{RPi} * 24h$	9,77	Ah
Minimaalne aku maht	–	$\frac{I_{ÖÖP} * 7}{0.6}$	113,98	Ah
Päikese paneeli maksimaalne võimsus	P_{MPP}	–	250	W
Minimaalne võimsustootlikkus päeva kohta (detsembrikuus) [25]	–	$\frac{80 Wh}{P_{ÖÖP}}$	68,3	%

Tabelis 8 toodud arvutustest võib järeldada, et antud paneel katab detsembrikuus vaid 68,3% Raspberry Pi energiavajadusest. Selleks et katta kogu energiavajadus, on võimalik lisada veel üks päikesepaneel, mis peaks katma energiavajadused umbes 36% varuga. Olgugi et energiavajadus saab sellega rahuldatud, leiab autor, et ei ole mõttekas kasutada

ligi 3m² päikesepaneeli pinda, et toita ära üks Raspberry Pi ja selle riistvara komponendid. Selleks et teha autonoomse toitega süsteem, tuleb juba süsteemi planeerides see nõudeks seada ning vastavalt riistvara komponendid valida. Kindlasti tuleb ka arvestada sellega, et väiksema energiavajadusega riistvara ei ole tavaliselt nii võimas, mis tähendab, et riistvara ei pruugi hakkama saada talle seatud ülesannetega.

4.2.3. Andmeedastus Wi-Fi kaudu

Kui süsteemi toide on autonoomne, siis on mõistlik kasutada andmesideks juhtmevaba lahendust. Järgnevalt uuritakse Wi-Fi *dongle*'i sobivust võrguliideseks. Võrgu testimiseks kasutatakse skripti, mis mõõdab pingi, lingi kvaliteeti ja faili edastamiseks kulunud aega. Enne testimist loodi 100 MB suurune testfail kasutades dd tööriista. Testimiseks kasutatud skript on välja toodud joonisel 10 ning testi tulemused tabelis 9.

```
#!/bin/bash
#Author: Jan Toodre
#Created on: 19-04-2015
#param1 = test number ,param2 is description
echo "Test: $1" >> test.txt
echo "$2" >> test.txt
iwconfig wlan0 | grep Quality >> test.txt
iwconfig wlan0 | grep Rate >> test.txt
PING=$(ping -c10 192.168.1.30)
echo "$PING" >> test.txt
DATE=$(date +%m/%d/%Y %H:%M:%S.%N')
echo "$DATE" >> test.txt
sudo cp ./output.dat /export/rpi/system/temp/test.dat
DATE2=$(date +%m/%d/%Y %H:%M:%S.%N')
echo "$DATE2" >> test.txt
iwconfig wlan0 | grep excessive >> test.txt
sudo rm /export/rpi/system/temp/test.dat
echo "-----" >> test.txt
echo "$1 Test done"
```

Joonis 10. Wi-Fi dongle'i võrgu testimise skript.

Tabel 9. Wi-Fi dongle'i võrgu testide tulemused.

#	Raspberry Pi kaugus marsruuterist	Link (kvaliteet, signaali tase, bitikiirus)	Ping (avg,loss)	Segajad	Testi Kestvus	Bitikiirus (Mbit/s)
1	1 m	Ethernet (eth0)	–	–	1 min 35 s	8,84
2	1,5 m	45/70, -65 dBm, 48 Mbit/s	1,346 ms; 0%	Töötav pesumasin, betoonsein	1 min 34 s	8,94
3	2 m	53/70, -57 dBm, 54 Mbit/s	1,351 ms; 0%	–	1 min 35 s	8,84
4	5 m	41/70, -69 dBm, 48 Mbit/s	1,450 ms; 0%	Tahvelarvuti Wi-Fi võrgus, uks	1 min 35 s	8,84
5	8 m	21/70, -89 dBm, 5,5 Mbit/s	1,852 ms; 20%	Betoon sein, Tahvelarvuti ja Laptop Wi-Fi võrgus, mitme Wi-Fi võrgu signaal, Bluetooth	26 min 47 s	0,52

Võrgu mõõtetulemused näitavad, et segavate faktoritega langeb Wi-Fi *dongle*'iga andmeedastuskiirus 8,84 Mbit/s pealt 0,52 Mbit/s peale. Lühikeste vahemaadega ja segajateta on testitud *dongle* abil saadav andmeedastuskiirus piisav hea kvaliteediga video edastamiseks, kuid realses süsteemis on vaja sidevahendit, mis on töökindlam ning mille tööraadius on suurem.

4.3. Tarkvaraarendus

4.3.1. Arendatud tarkvara

Tarkvaraarenduseks kasutati vabavaralisi programme Notepad++ ning vi. Notepad ++ on kasutusel Windows-i keskkonnas ning vi Linux-i keskkonnas. Vi valiti seetõttu, et sellel on mitmeid võimalusi tööd hõlbustavate moodulite lisamiseks ning tavaliselt on see programm juba Linux-isse installeeritud.

Selleks et serveri ja Raspberry Pi kallal töötada, loodi seadmetega SSH ühendus PuTTY programmi kasutades.

Erinevate funktsioonide arendusel kasutati erinevaid programmeerimiskeeli, mis on välja toodud tabelis 10.

Tabel 10. Arendusel kasutatud programmeerimis- ja skriptimiskeeled.

Keel	Kasutus
bash	Erinevad skriptid (arhiveerimine, veebi ja <i>raspimjpeg</i> vaheline suhtlus jne)
C	<i>Raspimjpeg</i> arendus
SQL	Päringute kirjutamine andmebaasist andmete saamiseks
PHP	Veebis andmete kuvamine ja ettevalmistamine
javascript	Veebis andmete töötlus ja sündmuste haldamine
CSS	Veebikujundus

Arendatud tarkvara hõlmab endas järgmisi programme, skripte ja faile:

raspimjpeg – programm, mis salvestab videot ja pilte ning voogedastab reaalajas kaamerapilti. Juurde on arendatud PIR sensori signaalitöötlus ning MySQL andmebaasi päringute saatmine. Programm on kirjutatud C-keeles, mõni süsteemi kutse jaoks bashis. Lisaks sellele on arendatud ka daemon, kontrollimaks *raspimjpeg*-i tööd.

Süsteemi tööks vajalikud skriptid, millega juhitakse serveri või Raspberry Pi tööd:

uni – skript, mis valib välja Raspberry Pi mooduli ning saadab *raspimjpeg*-ile käsu koos parameetritega.

archive – skript, millega arhiveeritakse valitud toores videoformaadis fail ning konverteeritakse ümber WebM tüüpi videofailiks. Ühtlasi kannab skript hoolt selle eest, et arhiveerimisest jääks märges ka andmebaasi.

Veebiliidese arendamiseks on kasutatud Bootstrap [28] raamistikku, mis koosneb kirjatüüpide, ikoonide ning javascript-i ja css failidest. Bootstrap-iga veebi arendamine on lihtsustatud, kuna visuaalne osa on suuresti ette tehtud – arendaja peab vaid välja valima komponendid, mida kasutada. Bootstrap-i raamistiku abil on arendatud palju lisamooduleid, mida on võimalik väikese vaevaga oma süsteemi integreerida. Antud süsteemis on seda raamistikku kasutatud näiteks andmebaaside päringutest tabelite moodustamiseks – selleks kasutati Bootstrap Table’it[29]. Kasutaja jaoks tabeli koostamiseks on vaja vormindamise parameetreid ning andmebaasi tabeli veerupealkirja, ülejäänud töö teeb bootstrap-table. Selleks et kuvada liikumiste statistikat, kasutati Chart.js teeki. Graafikute joonistamiseks on esmalt vaja andmed sellisele kujule viia, et Chart.js neid lugeda oskaks. Selle ülesandega tegeleb data.php fail. Kuna soov oli teha võimalikult dünaamiline süsteem, siis tegi autor eri funktsiooniga andmetötlusmoodulid (nt data.php ja ajax.php) ning veebiliidese, mis koosneb kaasatavatest moodulitest (nt navigatsiooni ja kaamerakonfiguratsiooni menüü).

4.3.2. Probleemid

Selle süsteemi arendusel tekkis autoril üks suur probleem, mis omakorda teisi probleeme tekitas. Kuna süsteem on tsentraliseeritud, siis on vaja Raspberry Pi-dele serverist käske anda. *Raspimjpeg* kasutab käskude vastuvõtmiseks nimega toru, mis töötab nagu FIFO andmepuhver. Kõigepealt toodi mõlema Raspberry Pi FIFO-d serveri ramdisk-i kettale, kuid programmid ei suutnud käske vastu võtta. Põhjus oli selles, et nimega toru töötab ainult lokaalselt, st ühes masinas. Sellest tulenevalt oli vaja otsustada, kas luua uus süsteem või moodus programmile käskude saatmiseks, või uurida, kuidas saaks FIFO-t Raspberry Pi-l hoida ja sinna käske kirjutada. Probleemi lahendamiseks otsustati teist varianti kasutada. Käskude saatmiseks *raspimjpeg*-ile loodi SSH ühendus serveri ja Raspberry Pi

vahel ning käsk kirjutati FIFO-sse. Lahendus töötab – *raspimjpeg* on võimeline kätte saama saadetud käsku.

Uus probleem ilmnis siis, kui käsku oli vaja veebiliidese kaudu käivitada. Lähema uurimise käigus selgus, et SSH ühendust üritab luua mitte see kasutaja, millega kasutaja oli eelnevalt testides serverisse sisse loginud, vaid hoopis *www-data*. Turvalisuse nõuetest tulenevalt on juba alguses keelatud sellel kasutajal kesta kasutada ning sisse logida, seega oli vaja leida moodus, kuidas *www-data* saaks luua SSH ühenduse Raspberry Pi-ga ja nii käsk edasi anda. Selleks loodi kasutajale *www-data* SSH võtmepaar ning lisati Raspberry Pi-sse *www-data* kui tuntud *host*. Nii sai lahendatud probleem, kuidas veebiliidestest *raspimjpeg*-ile käske saata.

4.3.3. Töö käik

Süsteemi arendus sarnanes algselt waterfall-meetodiga. Kõigepealt soetati süsteemi komponendid ning seejärel uuriti täpsemalt, kuidas või mida oleks võinud kasutada. Edasine arendus toimus agiilseid metoodeid kasutades. Pärast iga uut tarkvara osa lasti süsteemi proovida autori lähedastel, kellelt ka tagasisidet küsiti. Vastavalt tagasisidele muudeti programmi ja lisati juurde funktsioone (näiteks toore videofaili esimene kaader on salvestatud selleks, et saaks eelvaadet kuvada).

Arenduse esimeseks etapiks oli süsteemi lähtestamine, mida on kirjeldatud punktis 4.1. Edasine arendus toimus nii, et valiti välja funktsioon ning seejärel arendati ja testiti selle tööd. Arendustöös võiks välja tuua kolm kõige mahukamat osa:

- võrgu ülesseadmine
- *raspimjpeg* ja sellega seonduva arendus
- veebiliidese arendus

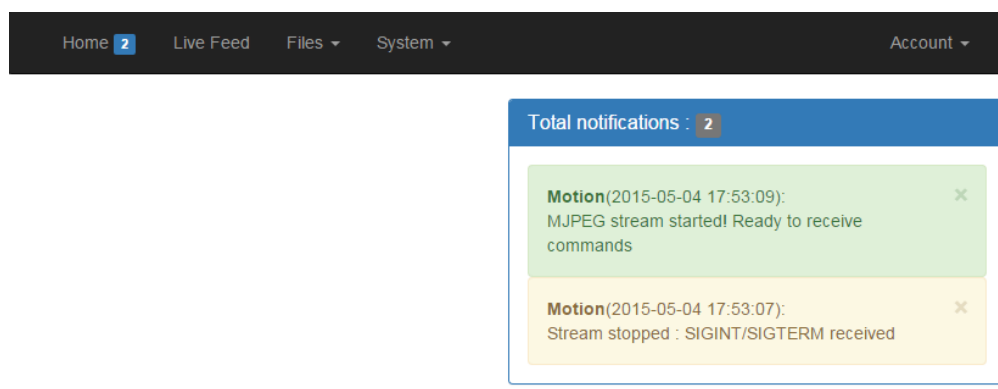
Kui võtta kokku kogu arenduskäik, siis võib öelda, et lõpptulemus vastab ootustele – süsteem töötab süsteemi nõuetele vastavalt ning veebiliides on kasutajasõbralik. Arendusmetoodikaks on kindlasti soovituslik kasutada agiilseid meetodeid ning eelnevalt

teha rohkem uurimistööd olemasolevate süsteemide kohta ning selle kohta, mida lõppkasutaja saada soovib.

4.4. Rakendus

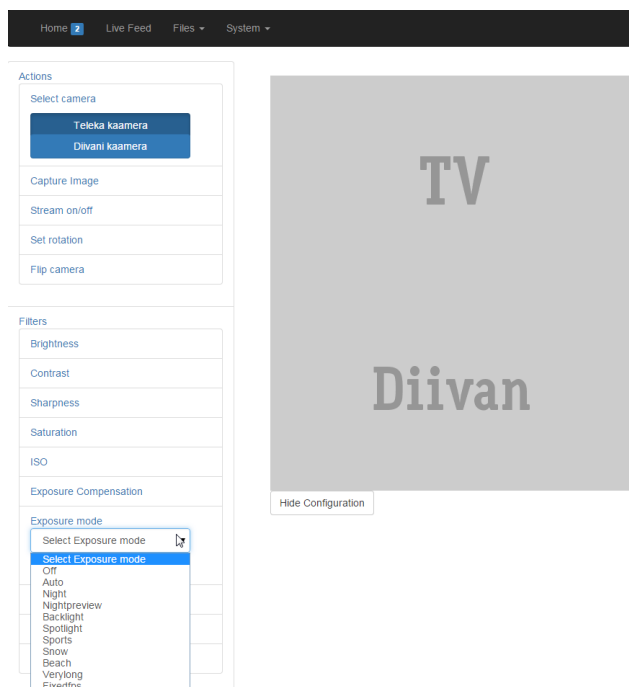
Järgnevalt on kirjeldatud veebiliidese funktsioone.

Home – Pealehel (Joonis 11) on võimalik näha serverilt ja Raspberry Pi-lt tulevaid teavitusi. Hiljem on plaanis sinna lisada õpetus selle kohta, kuidas süsteemi kasutada. Pealehe nupu juures on teadete arvu näitav sinine kast, mis on nähtav igal veebiliidese leheküljel viibides.



Joonis 11. Veebiliides – Pealeht.

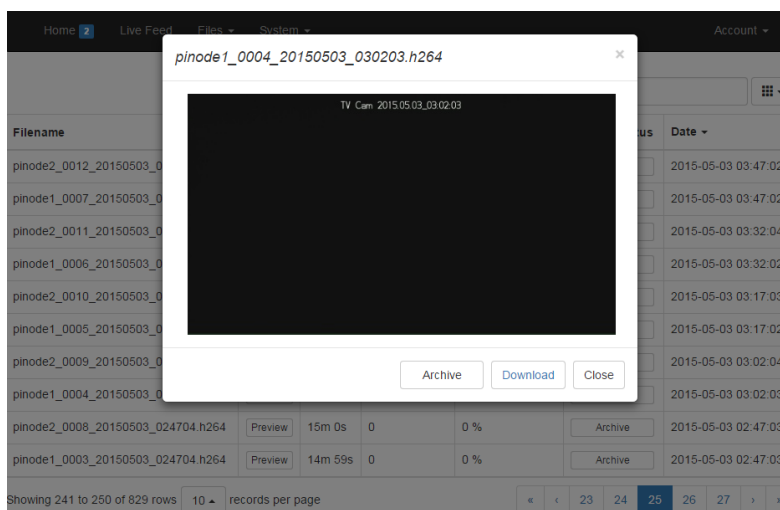
Live Feed – Otseedastuse lehel (Joonis 12) on võimalik vaadata reaajas kaamerapilti, teha pilti ning muuta kaamera sätteid vastavalt vajadusele. Vajutades nupule “Hide Configuration” kaob kaamera konfiguratsiooni menüü vasakusse ekraani serva, seda saab uuesti välja tuua “Show Configuration” nuppu vajutades.



Joonis 12. Veebiliides – Otseedastus.

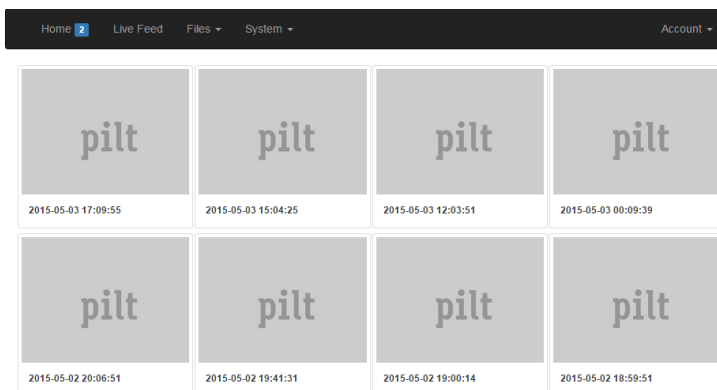
Files – Failihalduse alammenüüst on võimalik valida kolme lehe vahel:

Manage Files – Failihalduri lehel (Joonis 13) on toorete videofailide haldur. Seal on võimalik otsida ja sorteerida videosid. “Preview” nuppu vajutades on võimalik kuvada video esimest kaadrit, laadida alla toores videofail ning see fail arhiveerida. Video konverteeritakse peatükis 2.2.1 leitud video konverteerimisparameetreid kasutades.



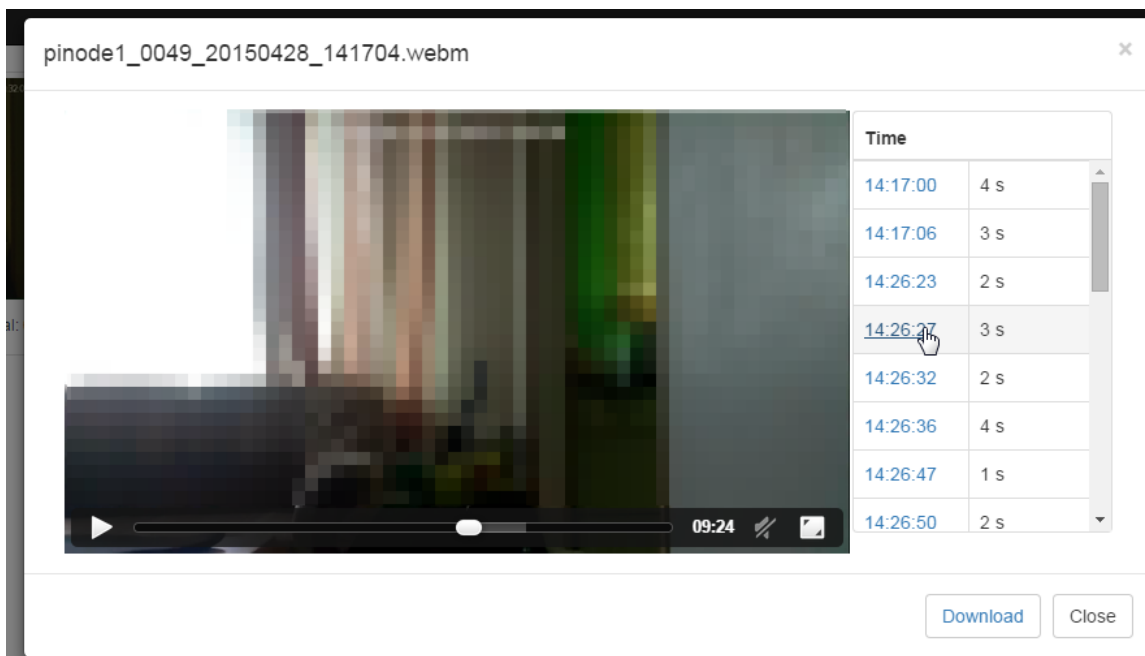
Joonis 13. Veebiliides – Failihaldur.

Saved Images – Salvestatud piltide lehel (Joonis 14) on pildigalerii, kus pildile klikkides see kasutajale suuremalt kuvatakse.



Joonis 14. Veebiliides – Salvestatud pildid.

Saved Videos – Salvestatud videote lehel on arhiveeritud videofailide haldur. Videopildil klikkides kuvatakse kasutajale video aken, mis on näidatud joonisel 15. Avanenud aknas videopildi kõrval on liikumistuvastused koos liikumiste pikkustega. Kui kasutaja klikib liikumistuvastuse ajale, keritakse video kohta, kus liikumine tuvastati. Video aknast on võimalik alla laadida arhiveeritud, kompaktsem videofail. Arhiveeritud videofaili mahu suhet võrreldes toore videofailiga on võimalik näha tabeli 1 neljanda testi viimasest tulbast.



Joonis 15. Veebiliides – Arhiveeritud video.

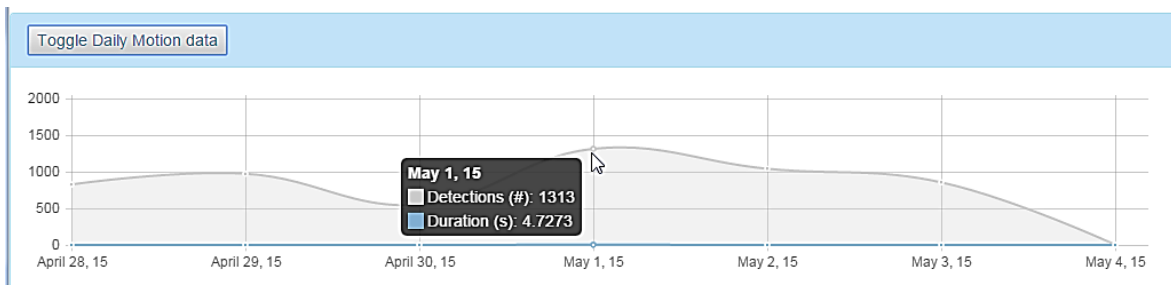
System – Süsteemi alammenüüst on võimalik valida kahe lehe vahel:

Logs – Logide lehel (Joonis 16) kuvatakse kõiki teateid (neid, mida on võimalik näha pealehelt ja neid, mis on suletud). Teateid on võimalik otsida ja sorteerida.

Message	Date
Error: Invalid exposure mode	2015-05-03 00:56:22
Error: Could not disable video port	2015-04-23 19:35:42
Error: Invalid image effect	2015-04-22 21:51:40
Error: Invalid exposure mode	2015-04-22 21:10:58
Error: Invalid exposure mode	2015-04-21 23:30:12
Error: Invalid exposure mode	2015-04-21 23:29:53
Error: Invalid exposure mode	2015-04-21 23:25:45
Error: Invalid exposure mode	2015-04-21 23:21:06
Error: Invalid exposure mode	2015-04-21 23:19:49
Error: Invalid exposure mode	2015-04-21 23:02:02

Joonis 16. Veebiliides – Logid.

Statistics – Statistika lehel kuvatakse liikumiste statistika graafikuid, mis on jaotatud kolme gruppi: minutipõhine, tunnipõhine ja päevapõhine (Joonis 17), kusjuures *Detections* on liikumistuvastuste arv ning *Duration* on keskmine liikumise kestvus.



Joonis 17. Veebiliides – Päevapõhine liikumistuvastuste statistika.

Ülejäänud menüüd ja lehed, mida on võimalik navigatsioonimenüüs näha, ei ole veel valminud või neil puudub funktsioon.

5. Edasiarenduse võimalused

Antud süsteemi on võimalik samas suunas edasi arendada ning teiste süsteemidega liidestada.

Süsteemi veebiliidesesse tuleks juurde lisada autentimisega sisselogimise võimalus. Selle funktsiooni eesmärk on teha veebiliidese kasutamine turvalisemaks, nii et võõrad isikud, kellel ei tohiks olla ligipääsu süsteemile, ei pääseks süsteemile ligi. Kindlasti tuleb eemaldada autentimata ligipääs veebilehtedele. See tähendab, et lihtsalt veebilehitseja aadressiribale veebilehe aadressi trükkides ei saaks lehele ligi pääseda.

Süsteemi paigaldamiseks tuleb leida kogu riistvarale sobiv ilmastikukindel korpus.

Rangelt soovituslik on hiljem tavalised kaameramoodulid NoIR kaameramoodulite [5] vastu välja vahetada. Selleks et NoIR kaameramoodulitega oleks võimalik täiesti pimedas ümbruskonda jäädvustada on vaja installeerida ka IR valgustid, mille valgus on inimsilmale nähtamatu. Hetkel on pimedas keskkonnas kaamerapilt täiesti must. Isegi kui IR valgustit kasutada, ei oleks pildilt võimalik midagi tuvastada, sest tavaline kaameramoodul ei ole võimeline tuvastama infrapuna lainepikkust.

Lisaks eelmainitud funktsioonide lisamisele on võimalik lisada printimisserver ja NAS. Suurematest integreeritavatest süsteemidest tuleks ära mainida välisvalgustite juhtimissüsteem veebiliidese kaudu. Samuti on autoril huvi arendada kasvuhoone juhtimissüsteem, mille abil on võimalik kontrollida taimede kastmist, avada ja sulgeda ukse-aknaid, mõõta temperatuuri jne.

6. Kokkuvõte

Antud bakalaureusetöö peamiseks eesmärgiks oli luua Raspberry Pi-1 põhinev liikumistuvastusega videojälgimissüsteem. Bakalaureusetöö raames uuriti alternatiivseid tarkvaralisi ja riistvaralisi lahendusi. Terviklikest tarkvaralistest lahendustest tutvustati lühidalt iSpyd ning Zoneminderit, mida olemasolevate turvakaamerate haldamiseks kasutatakse. Soovist luua paindlik edasiarendamisvõimalustega videojälgimissüsteem võrreldi erinevaid ühel trükkplaadil olevaid arvuteid, millest osutus sobivaks Raspberry Pi mudel B ja seda hinna ning funktsioonide pärast. Süsteemi arendus algas tarkvara ja riistvara installeerimise ning lähtestamisega. Uuriti süsteemi autonoomseks tegemise võimalusi, milleks tehti voolutarbimiste mõõtmised, Wi-Fi *dongle*'i andmeedastutestid ning arvutused Raspberry Pi päikesepaneeliga toitmise kohta. Süsteemi video ja pildi salvestamiseks ning edastamiseks võeti kasutusele avatud lähtekoodiga *raspimjpeg*. Seda programmi arendati edasi nii, et saaks liikumistuvastuse jaoks PIR sensori liidestada ning programmi poolt koostatud andmeid andmebaasi kirjutada. Andmebaasi salvestatud andmeid kasutatakse ja kuvatakse veebiliidesega. Arendamisel tekkinud probleemidega pöördus autor asjatundjate poole ning lahendamata ei jäänud neist ükski.

Võib öelda, et autor saavutas seatud eesmärgi. Sellest et teisedki isikud on süsteemi rakendamise suhtes huvi üles näidanud, võib järeldada, et saadud tulem on ootuspärane.

Kasutatud kirjandus

- [1] Upton, L.(2015). Five million sold! – Raspberry Pi blog [WWW] <https://www.raspberrypi.org/five-million-sold/> (25.04.2015)
- [2] What is Raspberry Pi? – Raspberry Pi [WWW] <https://www.raspberrypi.org/help/what-is-a-raspberrypi-pi/> (25.04.2015)
- [3] Downloads – raspberrypi.org [WWW] <https://www.raspberrypi.org/downloads/> (12.04.2014)
- [4] Rpi Cam Web Interface – elinux.org [WWW] <http://elinux.org/RPi-Cam-Web-Interface> (10.11.2015)
- [5] FFmpeg [WWW] <https://www.ffmpeg.org/> (26.04.2015)
- [6] WebM: an open web media project – webmproject.org [WWW] <http://www.webmproject.org/> (26.04.2015)
- [7] Handbrake [WWW] <https://launchpad.net/~stebbins/+archive/ubuntu/handbrake-releases> (28.03.2015)
- [8] MP4Box [WWW] <http://gpac.wp.mines-telecom.fr/mp4box/> (26.04.2015)
- [9] Encoding using ffmpeg [WWW] <https://encrypted.pcode.nl/blog/2010/10/17/encoding-webm-using-ffmpeg/> (03.05.2015)
- [10] Specifications – elinux.org [WWW] http://elinux.org/RPi_Hardware (26.04.2015)
- [11] Raspberry Pi mudel B joonis [WWW] <http://www.electronicblog.org/wp-content/uploads/2013/04/RaspiModelB-1024x902.png> (20.05.2015)
- [12] Wiring Pi [WWW] <http://wiringpi.com/> (11.11.2014)
- [13] Technical Parameters – elinux.org [WWW] http://elinux.org/Rpi_Camera_Module (26.04.2015)
- [14] JS-20 „Largo“ P.I.R detector [WWW] http://www.oomipood.ee/pdf/jablotron/JS-20_EN_MGD51205.pdf (11.11.2014)
- [15] YouTube Help [WWW] <https://support.google.com/youtube/answer/1722171?hl=en> (13.05.2015)
- [16] Cubieboard 2 [WWW] <http://docs.cubieboard.org/products/start> (03.05.2015)
- [17] Expansion Ports [WWW] <http://linux-sunxi.org/Cubieboard/ExpansionPorts> (27.04.2015)
- [18] ITT Group e-pood [WWW] <http://www.ittgroup.ee/et/e-shop> (26.11.2013)
- [19] iSpy – ISPYCONNECT [WWW] <http://www.ispyconnect.com/> (03.05.2015)

- [20] Zoneminder [WWW] <http://www.zoneminder.com/> (29.03.2015)
- [21] Chart.js [WWW] <http://www.chartjs.org/> (21.04.2015)
- [22] Webmin [WWW] <http://www.webmin.com/> (18.11.2014)
- [23] PoE lahendus Raspberry Pi-le [WWW] <http://imgur.com/a/MLCbN> (21.08.2014)
- [24] Päikese paneel [WWW] <http://www.taastuenergia.ee/pdf/polukristall-paikesepaneel-solarworld.pdf> (16.05.2015)
- [25] PVGIS [WWW] <http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php> (20.04.2015)
- [26] Päikesepaneelide tootlikkuse arvutamine [WWW] <http://www.taastuenergia.ee/paikesepaneelide-tootlikkuse-arvutamine-pvgis.html> (20.04.2015)
- [27] FAQ – raspberrypi.org [WWW] <https://www.raspberrypi.org/help/faqs/#cameraPower> (27.04.2015)
- [28] Bootstrap [WWW] <http://getbootstrap.com/> (19.01.2015)
- [29] Bootstrap Table [WWW] <http://bootstrap-table.wenzhixin.net.cn/> (07.04.2015)