

TARTU ÜLIKOOL  
MATEMAATIKA – JA INFORMAATIKA TEADUSKOND  
Arvutiteaduste instituut  
Informaatika eriala

**Hans Aarne Liblik**  
**Raspberry Pi kasutamine koos LEGO Mindstorms**  
**EV3 robotiga**

Bakalaureusetöö (9 EAP)

Juhendajad: Anne Villems  
Alo Peets  
Taavi Duvin

Tartu 2015

# **Raspberry Pi kasutamine koos LEGO Mindstorms EV3 robotiga**

## **Lühikokkuvõte:**

Käesoleva bakalaureusetöö eesmärgiks on koostada eestikeelne õppematerjal õpilastele ja tavakasutajatele Raspberry Pi B+ miniarvuti ja LEGO Mindstorms EV3 roboti koos kasutamise kohta. Töö esimeses peatükis selgitatakse, mis on Raspberry Pi ja LEGO Mindstorms EV3. Teises peatükis on juhend, mis aitab seadistada mõlemat seadet. Kolmandas peatükis on näiteülesanded, mis annavad ülevaate Raspberry ja EV3 koostööst ning ülesanded kasutajatele ise lahendamiseks.

## **Võtmesõnad:**

LEGO Mindstorms EV3, Raspberry Pi, informaatika, elektroonika, robotika, juhend, ülesanded

## **Raspberry Pi and LEGO Mindstorms EV3 robot**

### **Abstract:**

The purpose of this bachelor thesis is to create learning materials in Estonian for students and other users for using Raspberry Pi minicomputer and LEGO Mindstorms EV3 robot together. The first chapter gives an overview about Raspberry Pi and LEGO Mindstorms EV3. The second chapter contains the manual for setting up both gadgets. The third chapter contains example tasks that give an overview about Raspberry and EV3 and tasks for the user to solve.

### **Keywords:**

LEGO Mindstorms EV3, Raspberry Pi, Computer science, electronics, robotics, manual, tasks

# Sisukord

Kasutatavad lühendid .....	5
Sissejuhatus .....	7
1 Kasutusjuhendit vajava riistvara kirjeldus .....	9
1.1 Raspberry Pi .....	9
1.2 Raspberry Pi tehnilised andmed .....	9
1.3 Raspberry Pi võrdlus teiste sarnaste toodetega .....	10
1.4 LEGO Mindstorms EV3.....	11
2 Töö keskkonna ettevalmistamine .....	13
2.1 Raspberry Pi seadistamine (Riistvara, NOOBS, VNC).....	13
2.1.1 Mälukaardi vormindamine .....	14
2.1.2 Tarkvara paigaldamine.....	15
2.2 Raspberry Pi kasutamine tavaarvutist .....	19
2.2.1 Vajalik riistvara .....	19
2.2.2 Raadiovõrgu konfigureerimine .....	20
2.2.3 VNC seadistamine.....	22
2.3 LEGO Mindstorms EV3 seadistamine .....	26
2.3.1 Mälukaardi ettevalmistamine .....	26
2.4 Raspberry Pi ja EV3 juhtbloki ühendamise .....	28
2.4.1 Raspberry Pi ja EV3 USB kaabliga ühendamise .....	28
2.4.2 SSH ühendus Rasperry ja EV3 vahel.....	33
2.4.3 EV3 Interneti ühenduse loomine.....	34
2.5 EV3 sensorite ja mootorite programmeerimine .....	35
2.5.1 Pythoni teegi paigaldamine .....	36
3 Python2.7 näidisprogrammid ja ülesanded .....	38

3.1	„Tere maailm!“ .....	38
3.2	Sisendid .....	40
3.2.1	Ülesanded .....	41
3.3	LED pirni vilgutamine .....	41
3.3.1	Ülesanded .....	45
3.4	Pythoniga suhtlus EV3 ja Raspberry vahel .....	46
3.4.1	Ülesanded .....	49
3.5	Puutesensori abil LED pirni vilgutamine .....	50
3.5.1	Ülesanded .....	51
3.6	EV3 juhtbloki liigutamine nooleklahvidega .....	51
3.6.1	Ülesanded .....	57
3.7	Asünkroonne suhtlus EV3 juhtbloki ja Raspberry vahel .....	57
3.7.1	Ülesanded .....	60
3.8	Edasiarendus .....	61
	Kokkuvõte .....	62
	Viited .....	63
	LISAD .....	71
	I. Litsents .....	71

## Kasutatavad lühendid

**ARM** *Advanced RISC Machine* väikese voolutarbega protsessoriahitektuur, kasutatakse enamasti mobiilsetes seadmetes nagu mobiiltelefonid ja tahvelarvutid

**Arvuti** Laua- või sülearvuti

**CDC** *Communication Device Class* andmeliides

**DVI** *Digital Visual Interface* digitaalvideoliides, millega saab ühendada nii analoog- kui digisignaali kandvaid pistikuid

**FAT32** *File Allocation Table 32-bit* 32-bitine failisüsteem

**GB** *Gigabyte* mälumahu suurus; sisaldab  $2^{30}$  baiti

**GND** *Ground*

**GPIO** *General purpose input/output* kontrollitav viik integraalskeemis

**HDMI** *High-Definition Multimedia Interface* digitaalliides audio- ja videosignaali edastamiseks

**I<sup>2</sup>C** *Inter-Integrated Circuit* järjestiksiin, mida kasutatakse integraalskeemide ühenduslülina

**IDLE** *Integrated DeveLopment Environment* Python programmeerimiskeele redaktor- ja interpretaatorikeskkond

**LED** *Light-Emitting Diode* pooljuhtdiod, mis läbiva voolu toimetel kiirgab valgust

**MB** *Megabyte* mälumahu suurus; sisaldab  $2^{20}$  baiti

**NOOBS** *New Out Of the Box Software* Raspberry Pi jaoks loodud abimees operatsioonisüsteemi paigaldamiseks

**RNDIS** *Remote Network Driver Interface Specification* Microsofti suhtlusprotokoll, mille abil saab luua virtuaalse võrguühenduse

**RPi, Raspberry** Raspberry Pi

**SD** *Secure Digital* mälukaartiformaat, mida kasutatakse mobiilsetes seadmetes, näiteks kaamerad

**SSH** *Secure Shell* protokoll, mis võimaldab turvalist suhtlust seadmete vahel

**TCP** *Transmission Control Protocol* usaldusväärne ühendusega võrguprotokoll

**UART** *Universal Asynchronous Receiver/Transmitter* mikrokiip, mis vahendab andmeid jada ja paraleel vormide vahel

**UDP** *User Datagram Protocol* ebausaldusväärne ühenduseta võrguprotokoll

**UHS** *Ultra High Speed* SD kaardi siin, mis võimaldab suuremat andmeedastus kiirust

**USB** *Universal Serial Bus* universaalne välissiini standard välisseadmete ühendamiseks

**UTF-8** *Universal Character Set Transformation Format 8-bit* märkide kodeerimismeetod

**VNC** *Virtual Network Computing* ekraani jagamise tarkvara

## Sissejuhatus

Tänapäeva digiühiskonnas on robotid ja arvutid muutunud kergesti kättesaadavaks. Enamus digivahendite funktsioone on tänapäeva noortele kergesti mõistetavad, kuid digivahendite kasutamist raskendavad teatud vajakajäämised - lisakomponente on vähe või need on kallid, vahendeid ei saa kasutada teistes valdkondades, seade on kohmakas või tarkvaras esinevad puudused.

Üks võimalusi muuta vahendid universaalsemaks on süsteem üles ehitada mikrokontrollerit või võimekamat miniarvutit kasutades. Selleks pakub väga häid võimalusi Raspberry Pi nimeline miniarvuti ning robotiks sobib tuntud LEGO Mindstorms EV3, sest taolisi ühendusi on Raspberry Pi ja LEGO Mindstorms EV3 vahel varemgi tehtud. Kahjuks puuduvad eestikeelsed juhendid, mis motiveeriksid võimalikke kasutajaid robotikaga tegelema. Lisaks on Raspberry Pi üks jõudsamaid hetkel tavakasutajale kättesaadavaatest miniarvutitest, mille külge on võimalik ühendada peaaegu ükskõik missugust elektroonilist seadet, et siis kasutada seda kui elektroonikaseadme aju. Selline väikesemõõtmeline arvuti on üleilmselt kasutusel eesmärgiga tekitada sellest näiteks lihtne piltide esitaja pildiraamis [1] või kujundada arenenud toaandur, mis suudab jälgida inimeste liikumist ruumis [2], mõõta temperatuuri, õhuniiskust [3] või midagi muud ning edastada andmed otse serverisse või e-maili teel kasutajale.

Käesoleva töö eesmärgiks on koostada juhend erinevas eas arvutikasutajale, et demonstreerida võimalusi maailmas laialdaselt kasutusel oleva LEGO Mindstorms EV3 roboti ja Raspberry Pi omavaheliseks ühendamiseks. Töö tulemusena näidatakse, kuidas on võimalik koostatud juhendit järgides panna koos tööle miniarvuti ja juhtplokk ning selgitatakse, mida on võimalik selliselt valmistatud ühendusega teha. Lisaks koostatakse ülesannete pakett, mis sialdab õppeotstarbelisi näidisülesandeid ning ülesandeid, mida kasutaja peaks olema suuteline lahendama, olles eelnevalt lahendanud näidisülesanded.

Käesolevas töös esitatud juhendid ja seadmete kombinatsioonide kirjeldused toetavad digipädevuse kui ülekantava pädevuse kujunemist. 2011-2012 läbi viidud rahvusvahelise täiskasvanute oskuste uuring PIAAC tõi ühe tulemusena esile, et Eesti elanike arvutikasutusoskuse ja –julguse ning tehnoloogiarikas keskkonnas probleemilahendusoskuse tase on oodatust madalam. Heade ja väga heade probleemilahendusoskustega on vaid iga neljas

täiskasvanu. Seetõttu on käesoleva töö raames koostatud kasutusjuhendid ja ülesanded toeks vastava pädevuskomponendi kujunemisel igas eas kasutajale, kellel võib olla huvi, kuid kes kahtleb oma robotika-alases suutlikkuses. [4]

Eeskätt toetabki käesolev töö probleemilahenduse, mis on üks viiest digipädevuse komponendist, kujunemist. Probleemilahenduse all digipädevuste kontekstis mõistetakse oskust selgitada välja tegevuseks vajalikud digivajadused ja -ressursid, teha informeeritud otsuseid kõige otstarbekamate või kõige paremini vajadustele vastavate digivahendite kohta, lahendada digivõimaluste abil kontseptuaalseid probleeme, kasutada tehnoloogiat loovalt, lahendada tehnilisi probleeme ning ajakohastada enda ja teiste pädevust. Probleemilahendus on kõigist digipädevusvaldkondadest kõige läbivam ning vastav oskus laieneb igat tüüpi tegevustele, kus kasutatakse digivahendeid ja on ühtlasi seotud ka IKT ohutu ja kestliku kasutusega. [5] [6]



# 1 Kasutusjuhendit vajava riistvara kirjeldus

Käesolevas peatükis kirjeldatakse Raspberry Pi riistvara, võrreldakse seda analoogsete seadmetega ning kirjeldatakse LEGO Mindstorms EV3 robotit.

## 1.1 Raspberry Pi

Raspberry Pi (Joonis 1) on umbes pangakaardi mõõtmetega (85,6 mm × 56,5 mm) mikrokontroller ehk miniarvuti jaehinnaga 35€ ja seda toodetakse eesmärgiga edendada ja tutvustada programmeerimist kõigile vanuserühmadele [7] [8]. Raspberry Pi'd kasutatakse keskse seadmena, et juhtida erinevaid elektroonilisi komponente, mille puhul tavalise arvuti kasutamine oleks tema suuruse tõttu kohmakas ja ebaotstarbekas.



Joonis 1. Raspberry Pi B+ miniarvuti [42].

Raspberry Pi väikesemõõtmelisusest tingituna on ka tema tehnilised näitajad madalad ning neid võib hoopis võrrelda mobiiltelefoni näitajatega.

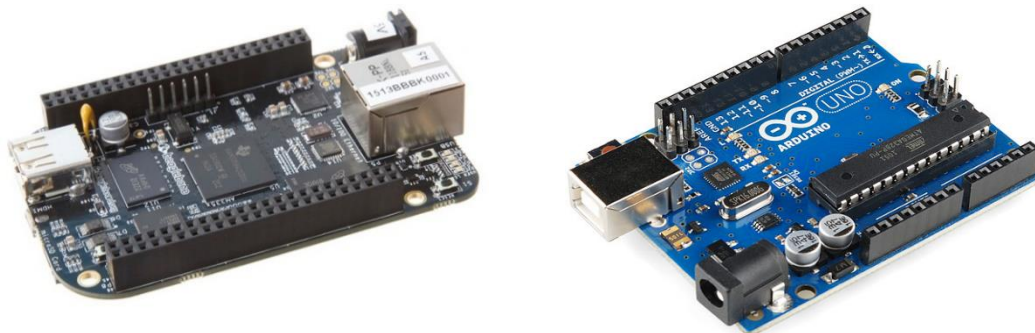
## 1.2 Raspberry Pi tehnilised andmed

Raspberry vajab töötamiseks 5-voldilist pinget. Seda saab kas 220V pistikust adapteriga, mis muundab pinget 5-voldiseks, või arvuti USB liidesest. Seadme külge saab ühendada klaviatuuri, hiire, võrgukaabli ja muid komponente nagu arvutile kohane. Raspberry tehnilised andmed on aga madalad ning neid võib võrrelda pigem mobiiltelefoni kui arvuti andmetega. Seadet kasutatakse üle maailma erinevatel eesmärkidel: selle abil saab ehitada kodukino süsteemi, teha LED lampidega valgusetendusi, kontrollida kellaajaliselt lambivalgust ja palju muud.

Raspberry Pi operatsioonisüsteemina saab kasutada mitmeid erinevaid Linuxi versioone, näiteks Raspbian ja Android. Programmeerimiskeelena soovitatakse kasutada Pythonit, kuid kasutada võib mistahes programmeerimiskeelt, mis kompileerub ARMv6 protsessoril. [7] [9]

### 1.3 Raspberry Pi võrdlus teiste sarnaste toodetega

Raspberry Pi'ga sarnaseid või analoogilisi tooteid on tänapäeval digiturul mitmeid [10]. Nendest enimtuntud on BeagleBone, Arduino (Joonis 2) ja Intel NUC seeria miniarvutid. Neist omakorda kõige väiksem on Arduino, kuid nagu tema suurus, on ka tema jõudlus väike. Seadet kasutatakse peamiselt üheotstarbelise riistvara juhtimiseks. BeagleBone ja Raspberry Pi on oma välimuselt ja liidestelt suhteliselt sarnased. Sobilikuks kasutusala on keerukamad ja mahukamad projektid kui Arduinol. BlackBone on ka peaaegu kolm korda rohkem maksev kui Raspberry Pi või Arduino. Hinna ja kvaliteedi ning oodatavate ülesannete tõttu on antud bakalaureuse töös kasutusel Raspberry Pi. Erinevate miniarvutite riistvara võrdlus on esitatud tabelis 1. [11] [12]



Joonis 2. BeagleBone Black Rev A5 (vasakul) [65] ja Arduino Uno R3 (paremal) [66].

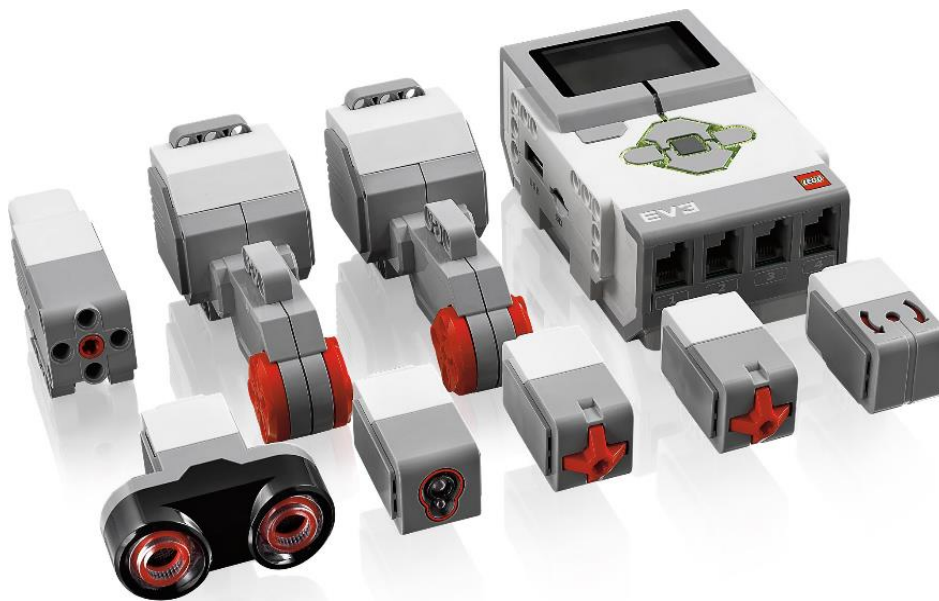
Tabel 1. Raspberry Pi B+, Arduino ja BeagleBone riistvara võrdlus [12] [13] [14].

	Raspberry Pi B+	Raspberry Pi 2 B	Arduino Uno R3	BeagleBone Rev A5
Protsessor	700MHz ARM11	900MHz ARM Cortex-A7	16MHz ATmega 328	700MHz ARM Cortex-A8
Muutmälu	512 MB	1024 MB	2 KB	512 MB
Andmekandja	microSD kaart	microSD kaart	32 KB	microSD kaart
Väljundid	4xUSB2.0, HDMI, Ethernet, 3.5mm heli	4xUSB2.0, HDMI, Ethernet, 3.5mm heli		1xUSB2.0, micro-HDMI, Ethernet, 3.5mm heli
Hind	\$35	\$60	\$30	\$89
I/O	40 GIOP	40 GPIO	14 GPIO, 6 analog	66 GPIO, 7 analog
Min. töövõimsus (energiatarve)	700mA (3.5 W)	800mA (4.0 W)	42mA (0.3 W)	170mA (0.85 W)

Seega on Raspberry Pi oma mõõtmete poolest sobiv antud töö eesmärgi saavutamiseks.

#### 1.4 LEGO Mindstorms EV3

LEGO Mindstorms EV3 (Joonis 3) on LEGO Mindstormsi robotite kolmanda põlvkonna mudel, mida hakati müüma 1. septembril 2013. LEGO Mindstorms roboteid, nagu LEGO klotse, on võimalik kasutajal vastavalt soovile kokku panna ning nende programmeerimisega saavad hakkama isegi algajad kasutajad.



Joonis 3. LEGO Mindstorms EV3 hariduslik baaskomplekt (ingl k. Education Core Set) [67].

Lisaks LEGO klotside külge ühendamisele, saab juhtbloki külge ühendada erinevaid sensoreid, mis reageerivad näiteks valgusele, helile või puudutusele. Robotit saab ühendada arvutiga USB liidese kaudu ning sealt laadida robotile kasutaja enda koostatud programme. Võimalusi roboti kasutamiseks on mitmeid: programmeerimise ja tehnika õppimine, Rubiku kuubiku lahendamine [15], Braille kirja printimine [16] jne. Roboti programmeerimiseks on mitmeid erinevaid võimalusi, kuid kasutajatele on loodud kergesti arusaadavad graafilised programmeerimiskeeled RCX Code või ROBO LAB, milles programm valmib plokkide lohistamise meetodil. [17] [18]

## 2 Töö keskkonna ettevalmistamine

Käesolevas peatükis kirjeldatakse Raspberry Pi ülesseadmist, LEGO Mindstorms EV3 põhivara muutmist ning nende kahe seadme ühendamise võimalusi.

### 2.1 Raspberry Pi seadistamine (Riistvara, NOOBS, VNC)

Raspberry Pi on oma olemuselt tavaline arvuti, millele installeeritavate operatsioonisüsteemide valik on suur – Raspbian, Pidora, Risc OS [19]. Raspberry Pi 2 operatsioonisüsteemina on võimalik kasutada isegi Windows 10, mida tulevikus saab tasuta alla laadida [20] [21]. Lisaks leidub modifitseeritud süsteeme nagu näiteks MusicBox [22], mis aitab kerge vaevaga seada üles muusikakeskuse, Kodi [23] (varem tuntud kui XBMC), mis on mõeldud filmide ja muude videote voogedastuseks, ja mitmeid teisi.

Käesolevas töös kasutatakse Raspberry Pi operatsioonisüsteemina Raspbiani, mille saame paketi NOOBS lisana. Raspbian on saadaval ka eraldi, kuid lihtsam on see installeerida koos NOOBS paketiga. NOOBS pakett on graafiline abimees operatsioonisüsteemi lihtsamaks paigaldamiseks. Kasutajal on võimalus paigaldada endale mitu süsteemi, et ei peaks vajadusel mälukaarte vahetama. NOOBS pakett sisaldab endas ka taastusliidest, mille abil on võimalik riknenud tarkvara uuesti paigaldada, muuta operatsioonisüsteemi konfiguratsiooni faili ning veebibrauseriga on võimalik hätta jäädes Internetist abi otsida. NOOBS pakette on saadaval kahte tüüpi – tavaline ja kerge. Tavaline sisaldab kõike vajaminevat ja ka lisasid. Kerge pakett sisaldab ainult NOOBS abimeest ning paigaldamise käigus laetakse vajalikud andmed Internetist. Tavaline pakett on üle 30 korra mahukam (737MB vs 22MB), aga kerge paketiga paigaldamiseks on vajalik Interneti ühendus. [24]

Enne töö alustamist on Raspberry andmekandja vaja vormindada FAT32 formaati. Andmekandjaks sobivad peaaegu kõik [25] vähemalt 8GB mahuga microSD mälukaardid ning soovitatavaks kaardi kiiruseks on vähemalt Class 10 (Joonis 4), kus number tähistab mälukaardi kirjutamiskiirust MB/s, või isegi UHS-1 tüüpi mälukaardid, mille kirjutamiskiirus võib küündida kuni 50MB/s. Mälukaardi formaatimiseks arvutis läheb vaja adapterit, mis muudab microSD kaardi SD kaardiks (Joonis 4). Formaatumiseks on mõttekas kasutada Internetist tasuta allalaetavat SDFormatter [26] tarkvara.



Joonis 4. microSD kaart ja SD adapter.

Kui Raspberry on valmis, on vaja sisend- ja väljundseadmeid. Kuna Raspberry külge saab ühendada USB seadmeid, siis on kohe alguses otstarbekas hankida sobilik klaviatuur ja hiir. Väljundseadmeks on HDMI liidesega monitor ja sel juhul on vaja ka HDMI kaablit. Sobib ka DVI liidesega monitor, kuid sel juhul on vaja HDMI-DVI juhet. Kui kõik vajalikud komponendid on olemas, tuleb Raspberry Pi kodulehelt allalaadida NOOBS tarkvara. Allalaetav fail on kokkupakitud kujul ja kasutamiseks tuleb see lahti pakkida.

### 2.1.1 Mälukaardi vormindamine

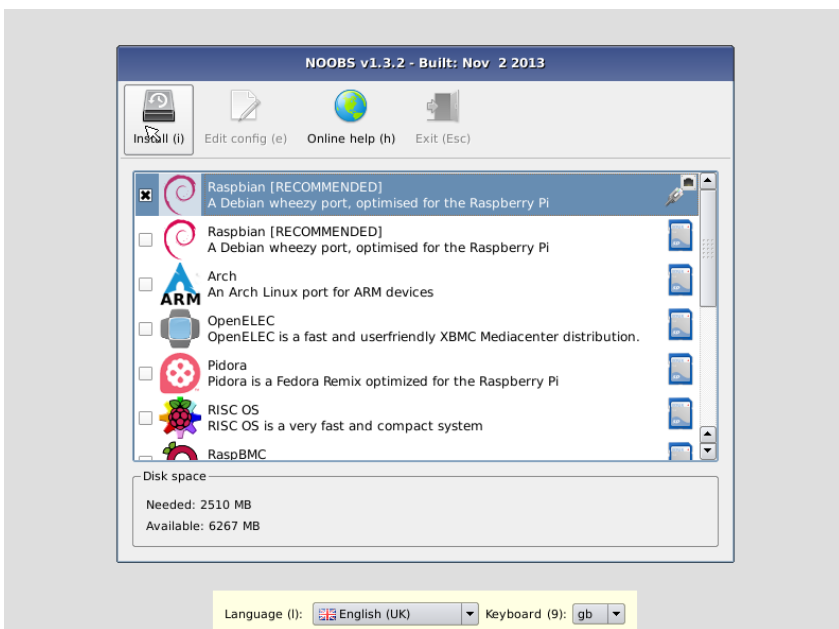
Mälukaart on vaja enne kasutamist vormindada FAT32 süsteemi. FAT32 failisüsteemi saab formaatida andmekandjaid mahuga 512MB - 2TB. Sellesse vahemikku jääb enamus tänapäeva mälukaarte. Lisaks saab selles failisüsteemis olla maksimaalseks faili suuruseks 4GB.

Formaatimiseks on soovitatav kasutada SDFormatter [26] tarkvara, mille saab tasuta allalaadida SD Association veebileheküljelt ([https://www.sdcard.org/downloads/formatter\\_4/](https://www.sdcard.org/downloads/formatter_4/)), aga võib kasutada ka Windowsi enda vormimistarkvara. Kindlasti tuleb järgida ekraanil kuvatavaid juhiseid tarkvara paigaldamiseks. Koos adapteriga on vaja sisestada mälukaart arvuti vastavasse pesse, SD Formatter tarkvaras tuleb valida õige ketta tähis ja vajutada „Format“. Kui vormimine on lõpetatud, siis tuleb eelnevalt lahtipakitud failid lohistada või kopeerida mälukaardile ning kopeerimise lõppedes eemaldada turvaliselt mälukaart arvutist ning sisestada Raspberry Pi mälukaardi pesse.

## 2.1.2 Tarkvara paigaldamine.

Järgnev tekst tugineb Raspberry Pi kodulehel olevale ingliskeelsele NOOBS paketi paigaldamise õpetusele [27] ning raamatule „Raspberry Pi for DUMMIES“ [28], milles tuuakse välja, missuguseid seadistusi Raspberry Pi süsteemis teha ja miks.

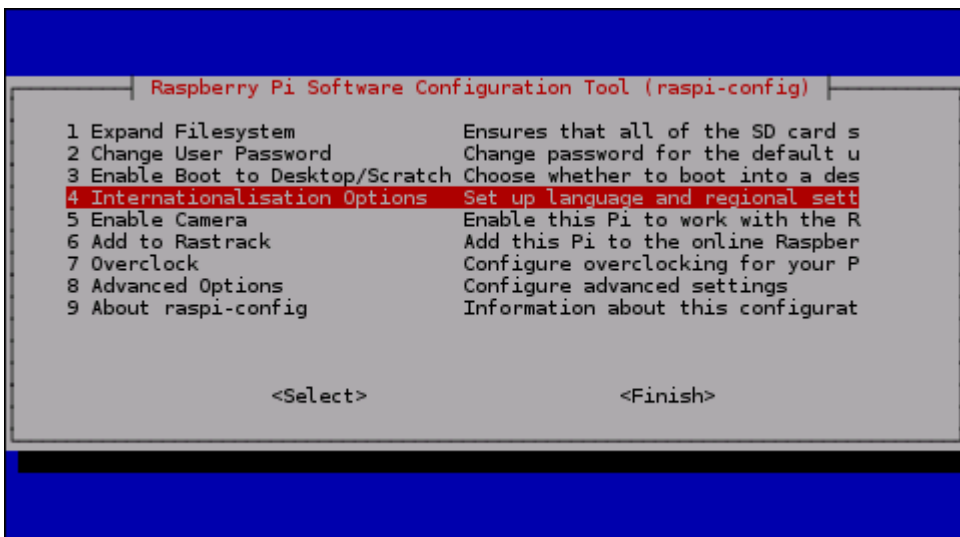
Pärast vormimist tuleb Raspberry külge ühendada klaviatuur, hiir ja monitor ning seejärel toitejuhe. Klaviatuuri ja hiire hilisemal ühendamisel ei pruugi nad korralikult tööle hakata. Ekraani hilisemal ühendamisel on resolutsioon väga väike ning see võib olla edaspidises töös häiriv. Seade hakkab kohe peale toite ühendamist ise tööle ja ekraanile ilmub loetelu erinevate võimalike installeeritavate operatsioonisüsteemidega. Soovitav on valida Raspbian ja vajutada nupule „Install“ (Joonis 5).



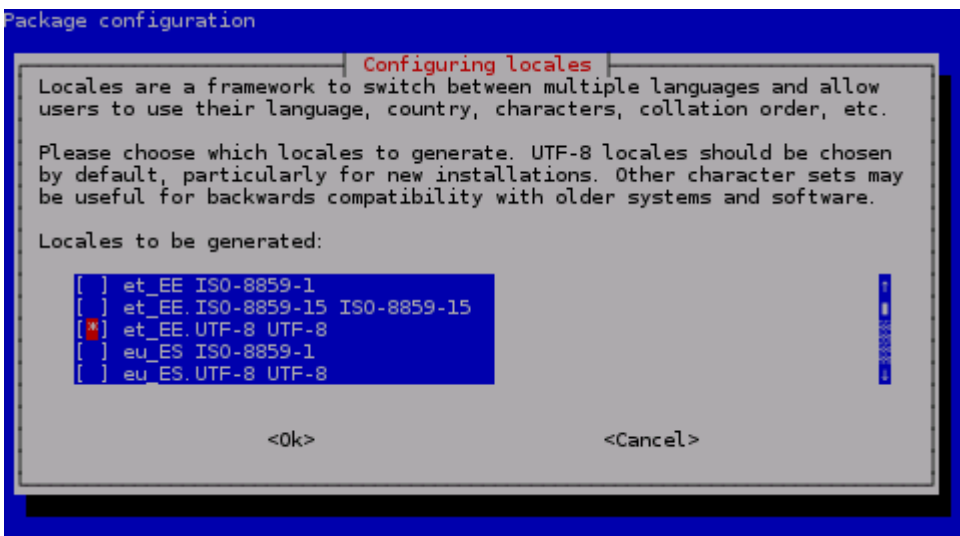
Joonis 5. NOOBS paketiga Raspbian süsteemi paigaldamine [29].

Järgmise tegevusena hakkab Raspbian end ise paigaldama ning peab arvestama, et näiteks Class 10 kiirusega mälukaardi puhul kulub selleks umbes 25 minutit. Peale paigaldamist Raspberry taaskäivitub ning seejärel kuvatakse konfiguratsiooni menüü ehk 'raspi-config', kus on võimalik muuta kuupäeva ja kellaaja seadeid, lisada kasutajaid, tõsta protsessori taktsagedust ja palju muud. Soovitav on kellaeg ja kuupäev õigeks muuta, valida sobivad keele ja klaviatuuri seaded, lülitada sisse SSH ja töölauale bootimine ning muuta mälu korraldust ehk määrata, kui palju vahemälu antakse graafikaprotsessori kasutusse.

Keele seadete muutmiseks tuleb avanenud konfiguratsioonimenüüst valida '4 Internationalisation Options'(Joonis 6). Järgmiseks valida 'I1 Change Locale' ning oodata, kuni järgmine menüü laeb. Järgmises menüüs tuleb valida kasutajale sobiv süsteemi keel. Vaikimisi on selleks keeleks inglise keel (en\_GB, UTF-8, UTF-8), aga võimalik on valida ka eesti keel. Sel juhul tuleb ära märkida 'et\_EE.UTF-8 UTF-8' liikudes sobiva variandi peale ja vajutades tühikut (Joonis 7). Seejärel liikuda 'TAB' klahvi vajutades '<Ok>' nupu juurde ja vajutada Enter klahvile. Peale seda hakkab Raspberry oma konfiguratsiooni seadeid muutma.



Joonis 6. Raspi-config peamenüü vaade.



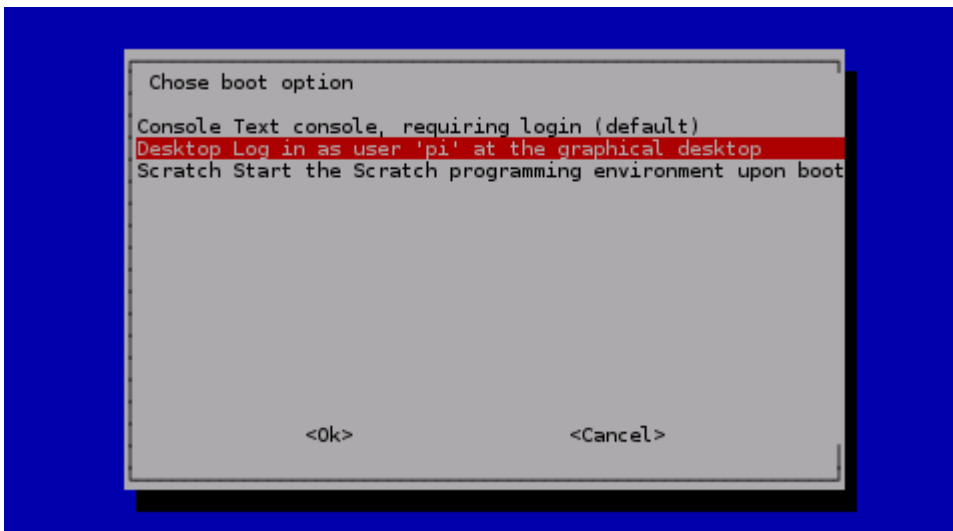
Joonis 7. Raspi-config Internationalisation alammenüü vaade.



Kellaaja ja kuupäeva seadete muutmiseks tuleb uuesti valida '4 Internationalisation Options', aga järgmisest menüüst juba 'I2 Change Timezone'. Peale laadimist kuvatakse valik kõikide geograafiliste aladega. Eesti valimiseks tuleb valida 'Europe' ning järgmisest menüüs 'Tallinn'.

Klaviatuuri seadete muutmiseks tuleb taas suunduda '4 Internationalisation Options' valiku juurde ja sealt edasi valida 'I3 Change Keyboard layout'. Sellega muudab Raspberry oma klaviatuuri seadistuse ühtseks keele seadetega.

Järgmiseks tuleb sisse lülitada töölauale bootimine. Selleks tuleb põhimenüüs valida '3 Enable Boot to Desktop/Scratch' ning järgmisest menüüs 'Desktop Log in as user 'pi' at graphical desktop' (Joonis 8).



Joonis 8. Raspi-config boot menüü vaade.

SSH ehk krüpteeritud tekstipõhise liicluse lubamiseks tuleb see kõigepealt sisse lülitada. Selle jaoks tuleb valida raspi-config menüüst '8 Advanced Options' ning sealt 'A4 SSH' ja siis valida '<Enable>'

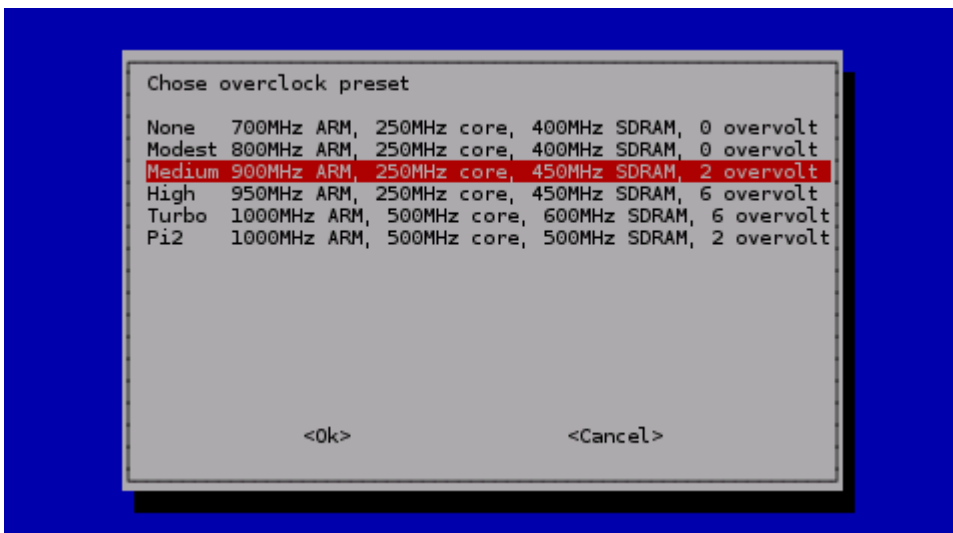
Kui Raspberry kasutuselaks ei ole mõeldud väga graafiline töö, siis on otstarbekas muuta natuke seadme mälu kasutust. Selle käigus saab valida, kui mitu MB vahemälu antakse graafika protsessori kasutusse. Mälu korralduse muutmiseks tuleb valida põhimenüüst uuesti '8 Advanced Options' ning järgmisest menüüst 'A3 Memory Split'. Avanenud aknas saab sisestada kogusteks kas 16, 32, 64, 128 või 256MB. Kui ei tegutseda väga graafilistes keskkondades, siis on soovitatav valida 16.

Turvalisuse kaalutlustel on soovituslik muuta vaikekasutaja 'pi' parool. Kasutaja vaikeparooliks on algselt seatud 'raspberry'. Parooli muutmiseks tuleb valida '2 Change User Password', millele järgneb kinnitus, milles ühtlasi teavitatakse, et järgmiseks tuleb sisestada kasutaja 'pi' uus parool 2 korda. Pärast seda avaneb Terminalilaadne aken, millesse tuleb sisestada uus parool 2 korda. Kirjutades ei paista ekraanil midagi juurde tekkivat, kuid see on Linux-i-põhiste operatsioonisüsteemide turvaprotokoll. Kui kõik oli edukas, siis kuvatakse aken, mis ütleb, et parool on edukalt muudetud. Parooli saab ka hiljem käsitsi muuta, sisestades käsu

- `passwd`

kuid sel juhul küsitakse kasutajalt ka kehtivat parooli ning pärast seda alles kaks korda uut parooli.

Seadet on võimalik ka ülekiirendada [30] (ingl. k. Overclock), kuid tuleb meeles pidada, et ülekiirendus võib lühendada seadme eluiga. Ülekiirendamine on toiming, mille käigus sunnitakse mingit komponenti töötama kiiremini, kui ettenähtud. Lisaks võidakse komponendile anda rohkem voolu, et see püsiks stabiilne. Selle toiminguga käigus proovitakse saada seadmelt paremat jõudlust. Ülekiirendamise sisselülitamiseks tuleb valida peamenüüst '7 Overclock'. Järgmiseks kuvatakse eespoolnimetatud hoiatus. Seejärel kuvatakse menüü, milles pakutakse mitu erinevat ülekiirendamise valikut. Veidi jõudluse parandamiseks, kuid süsteemi stabiilsuse säilitamiseks on soovituslik valida 'Medium' (Joonis 9).



Joonis 9. Raspi-config ülekiirendamise menüü vaade.

Lõpetuseks tuleb vajutada tabulatsiooni klahvi (ingl k. *Tab*), et liikuda Finish nupule, tuleb vajutada Enter klahvi ning tekkinud reale kirjutada

- `sudo reboot`

mis taaskäivitab seadme, et muudatused jõustuksid ja kuvataks graafiline kasutajaliides. Raspiconfig menüüsse on alati võimalik tagasi minna, kirjutades käsureale ehk Terminali aknasse käsu

- `sudo raspi-config`

Kui Raspberry on ülesseatud, siis on mõttekas luua juhtmevaba ühendus Raspberry ja arvuti vahel, mille abil on võimalik arvutist kontrollida Raspberry ekraani nii, nagu oleks seadmega ühendatud klaviatuur, hiir ja monitor.

## 2.2 Raspberry Pi kasutamine tavaarvutist

Kuigi Raspberryt on võimalik kasutada eraldi klaviatuuri, hiire ja monitoriga, on see tülikas ja ruuminõudev, kui laual on lisaks arvuti lisadele veel vähemalt üks paar komponente või kui Raspberry asub raskest ligipääsetavas kohas. Veelgi tülikam on lisasid Raspberry ja arvuti vahel pidevalt lahti ja kinni ühendada. Üks võimalik lahendus sellele probleemile on neid seadmeid Raspberry külge ühendada, kasutades USB jaoturit (ingl. k. *hub*), kuid ühendamine võib tekitada probleeme Raspberry võimsuses – ühendus muutub nõrgaks või lausa puudub [31]. Kindlaim lahendus sellele probleemile on kaugtöölaua ühendus ehk VNC, mis annab kasutajale võimaluse juhtida mitmeid erinevaid seadeldisi ühest seadmest.

### 2.2.1 Vajalik riistvara

VNC kasutamiseks peavad kõik ühendamist vajavad seadmed olema ühendatud, kas lokaalse või globaalse arvutivõrguga. Nagu ka tavaarvutit, saab Raspberryt Interneti ühendada võrgukaabliga või üle raadiovõrgu ehk Wi-Fi. Raspberry Pi traadita raadiovõrku ühendamiseks peab olema seadme külge ühendatud USBga Wi-Fi tongel (Joonis 10).



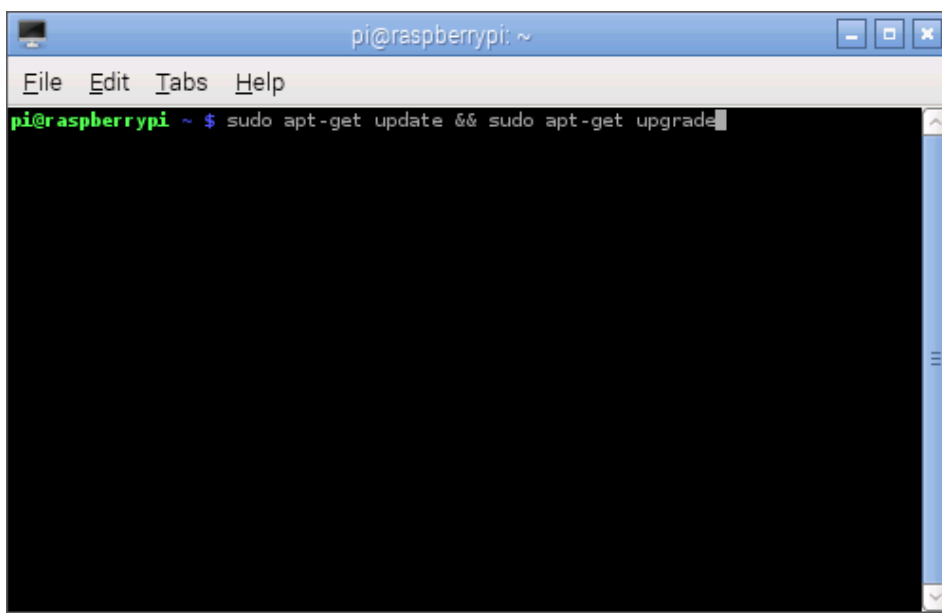
Joonis 10. Edimax EW-7811Un USB Wi-Fi tongel [32].

Paljud poelettidel müüdavatest seadmetest sobivad edukalt ning hakkavad ühendamisel automaatselt tööle [33]. Vaja on vaid konfigurereida seadme ühendus raadiovõrguga. Kui Wi-Fi tongel ühendades tööle ei hakka, on mõttekas proovida seadmete ühendamist enne Raspberry käivitamist. Kui ka siis ühenduse loomine nurjub, tuleb pöörduda ingliskeelsete õpetuste poole [33].

### 2.2.2 Raadiovõrgu konfigureerimine

Raspberry Pi raadiovõrku ühendamiseks on mitu moodust. Lihtsaim nendest on kasutada graafilist võrguhalduri programmi, mida on võimalik allalaadida Raspberry repositooriumist. Võrguhalduri graafilise programmi abil on võimalik automaatselt otsida lähedal asuvaid raadiovõrke ning sobivatega kergesti ühenduda. Repositooriumist vajaliku allalaadimiseks peab Raspberry olema eelnevalt ühendatud võrguga, mis tähendab, et Raspberry külge peab olema ühendatud võrgukaabel, sest raadiovõrku ühendamiseks on vaja võrguhaldurit. Lisaks tuleb uuendada Raspberry andmed sellesse paigaldatud tarkvara kohta ning nendest vajaminevad uuendada ja kasutatud programmid eemaldada. Selle jaoks tuleb avada Terminali aken ning käivitada seal käsk

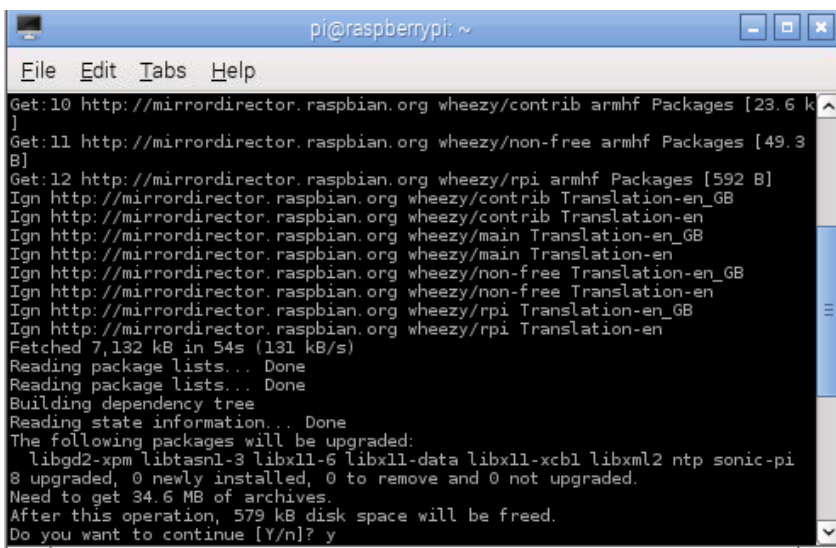
- `sudo apt-get update && sudo apt-get upgrade` (Joonis 11).



Joonis 11. Raspberry Terminali vaade süsteemi uuendamise käskust.

Selle käsu läbiviimine võib nõuda palju aega sõltuvalt Interneti ühenduse ning mälukaardi kiirusest. Kui käsk küsib parooli, siis tuleb sisestada kasutaja 'pi' parool. Kui seda ennist ei

muudetud, on selleks vaikeparool 'raspberry'. Kui küsitakse nõusolekut millegi uuendamiseks või eemaldamiseks, siis tuleb kirjutada 'y' (Joonis 12).



```
pi@raspberrypi: ~  
File Edit Tabs Help  
Get:10 http://mirrordirector.raspbian.org wheezy/contrib armhf Packages [23.6 kB]  
Get:11 http://mirrordirector.raspbian.org wheezy/non-free armhf Packages [49.3 B]  
Get:12 http://mirrordirector.raspbian.org wheezy/rpi armhf Packages [592 B]  
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB  
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en  
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB  
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en  
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB  
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en  
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB  
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en  
Fetched 7,132 kB in 54s (131 kB/s)  
Reading package lists... Done  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages will be upgraded:  
  libgd2-xpm libtasn1-3 libx11-6 libx11-data libx11-xcbl libxml2 ntp sonic-pi  
8 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
Need to get 34.6 MB of archives.  
After this operation, 579 kB disk space will be freed.  
Do you want to continue [Y/n]? y
```

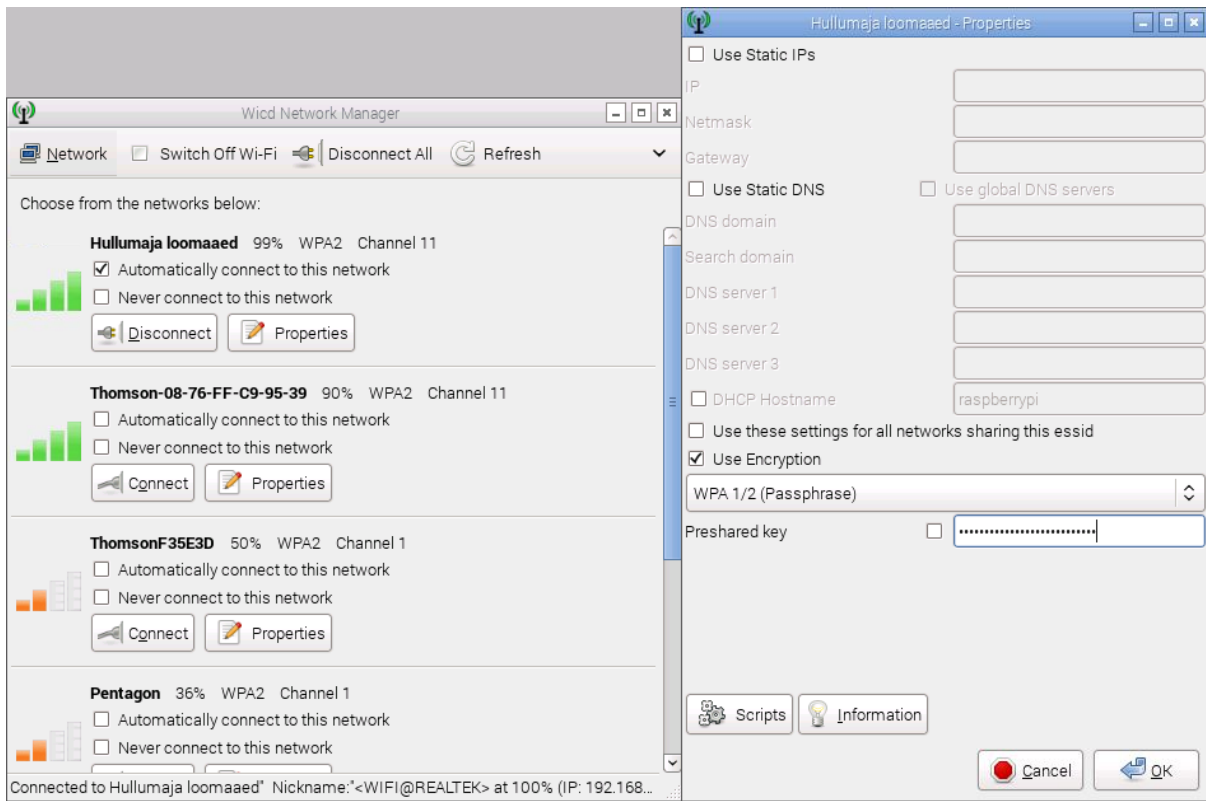
Joonis 12. Raspberry Terminali vaade süsteemi uuendamise kinnitusest.

Kui Raspberry süsteem on hoidlaga sünkrooni viidud, siis tuleb allalaadida 'Wicd Network Manager' või mõni muu võrguhaldur. Wicd programmi saab allalaadida, käivitades käsu

- `sudo apt-get install wicd`

Selle käsuga otsitakse hoidlast 'wicd' nimeline pakett ning kui see leidub ja seda veel paigaldatud ei ole, siis laetakse see alla ja installeeritakse. Paigaldamise käigus küsitakse kasutajalt kinnitust ja kinnituseks tuleb vastata 'y'.

Pärast võrguhalduri paigaldamist tuleb see avada. Selleks tuleb menüüst (ingl. k. Menu) valida 'Internet' ning sealt 'Wicd Network Manager'. Järgmiseks avaneb aken ning juhul, kui Wi-Fi seade on edukalt leitud, hakkab Wicd otsima raadiovõrke. Leitud võrkudest tuleb valida endale teadaolev võrk ning märkida sellejuures 'Automatically connect to this network'. Kui võrgul on peal parool, siis tuleb enne ühendamist see sisestada. Selle jaoks tuleb vajutada nupule 'Properties' (eesti k. omadused) ning märkida linnuke parameetri 'Use Encryption' ette, seejärel sisestada vastav parool lahtrisse 'Key'. (Joonis 13)



Joonis 13. Raspberry WICD võrguhalduriga raadiovõrku ühendamine.

Parooli eduka sisestamise järel tuleb vajutada nupule 'OK' ning seejärel nupule 'Connect' (eesti k. ühenda). Valitud raadiovõrgu parooli puudumisel tuleb lihtsalt valida 'Connect'. Nüüd on Raspberry seatud automaatselt ühenduma valitud raadiovõrguga, kui see on leitav ning seadme küljest võib eemaldada võrgujuhtme.

### 2.2.3 VNC seadistamine

Järgnev tekst tugineb veebipõhisel elektroonika õppimissaidi Adafruit ingliskeelsel õpetusel, kuidas paigaldada ja kasutada VNC tarkvara Raspberry Pi-ga [34].

Arvuti ja Raspberry vahelise VNC ühenduse loomiseks on vaja Raspberry kettale allalaadida vastav serveri tarkvara ning arvutisse VNC klient. Selle jaoks on vajalik Raspberry Pi's avada käsuviip ja käivitada seal käsk

- `sudo apt-get install tightvncserver`

See käsk laeb Raspberry Pi kettale hoidlast alla VNC serveri tarkvara. Paigaldamise käigus küsitakse kinnitust ning sinna tuleb vastata 'y'.

Järgmiseks tuleb tekitada automaatkäivitus fail VNC-le. Selle jaoks tuleb käsuraal käivitada käsk

- `cd /.config`

mis muudab jooksva kataloogi asukohta. Terminali käivitamisel on jooksvaks kataloogiks kasutaja kodukataloog, milleks on enamasti `/home/kasutajanimi`. Kodukataloogi tuleb luua alamkaust `autostart`

- `mkdir autostart`

ning tekitada sinna fail, mis automaatselt koos Raspberryga käivitab VNC serveri. Selle jaoks tuleb luua uus fail

- `sudo nano tightvnc.desktop`

ja lisada sinna kood

```
[Desktop Entry]
Type=Application
Name=TightVNC
Exec=vncserver :1 -geometry 1600x900 -depth 16 -pixelformat rgb565
StartupNotify=false
```

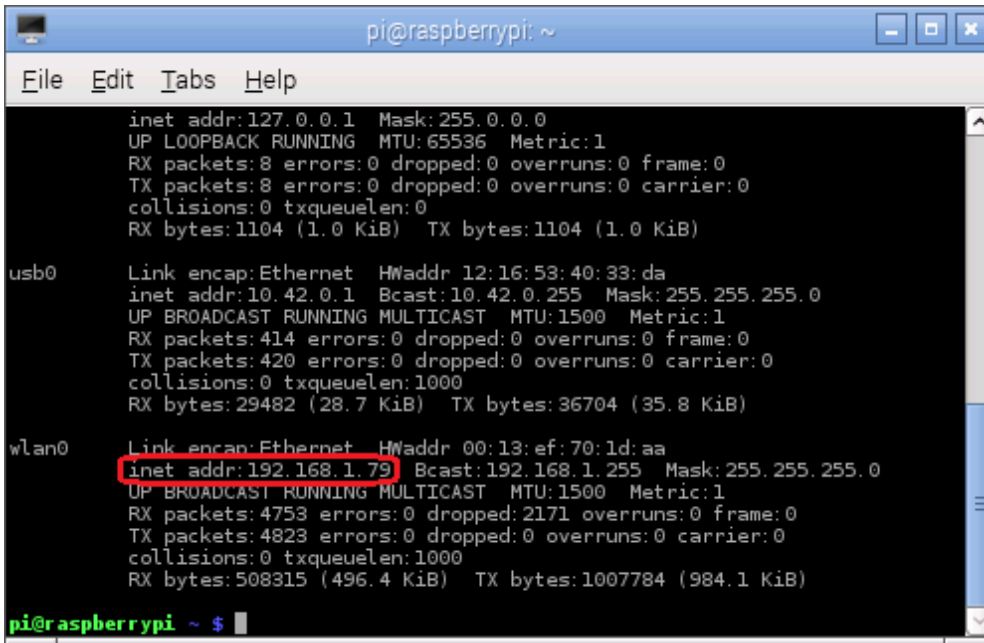
See kood tekitab sissekande nimega TightVNC ning käivitab VNC serveri pordi peal 1. VNC resolutsiooniks on 1600 x 900 pikslit, värvi sügavuseks on 16, mis kiirendab ühenduse kiirust ning pixelformat rgb 565 märgib ära, et 16biti seas on 5 rohelist, 6 sinist ja 5 punast bitti vastava värvi jaoks. Soovi korral võib VNC resolutsiooni muuta, kuid kuna Raspberry ise on aeglane, siis viimased 2 parameetrit on ühenduse kiirendamiseks tähtsad.

Faili sulgemiseks tuleb vajutada CTRL + X, seejärel sisestada 'y', mis kinnitab faili tehtud muudatused ja lõpetuseks vajutada Enter klahvile, mis salvestab faili kettale.

Järgmiseks on vaja kindlaks teha, missugune IP-aadress on määratud Raspberry võrgukaardile. Selle jaoks tuleb sisestada Terminali aknasse käsk

- `ifconfig`

mis kuvab ekraanile kõikide olemasolevate võrguliideste andmed. Pärast seda tuleb otsida välja `'wlan0'` ning selle juurest atribuut `'inet addr:'`, milles pärast semikoolonit on neli arvu eraldatud kolme punktiga (Joonis 14). Terve see blokk on Raspberry IP-aadress ning seda on vaja hilisemaks VNC-ga ühendumiseks. Seetõttu on aadress vaja meelde jätta või turvaliselt üles kirjutada.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
inet addr:127.0.0.1 Mask:255.0.0.0  
UP LOOPBACK RUNNING MTU:65536 Metric:1  
RX packets:8 errors:0 dropped:0 overruns:0 frame:0  
TX packets:8 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:1104 (1.0 KiB) TX bytes:1104 (1.0 KiB)  
  
usb0 Link encap:Ethernet HWaddr 12:16:53:40:33:da  
inet addr:10.42.0.1 Bcast:10.42.0.255 Mask:255.255.255.0  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:414 errors:0 dropped:0 overruns:0 frame:0  
TX packets:420 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:29482 (28.7 KiB) TX bytes:36704 (35.8 KiB)  
  
wlan0 Link encap:Ethernet HWaddr 00:13:ef:70:1d:aa  
inet addr:192.168.1.79 Bcast:192.168.1.255 Mask:255.255.255.0  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:4753 errors:0 dropped:2171 overruns:0 frame:0  
TX packets:4823 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:508315 (496.4 KiB) TX bytes:1007784 (984.1 KiB)  
  
pi@raspberrypi ~ $
```

Joonis 14. Punase tähistusega ümbritsetud Raspberry Pi lokaalne Wi-Fi IP-aadress.

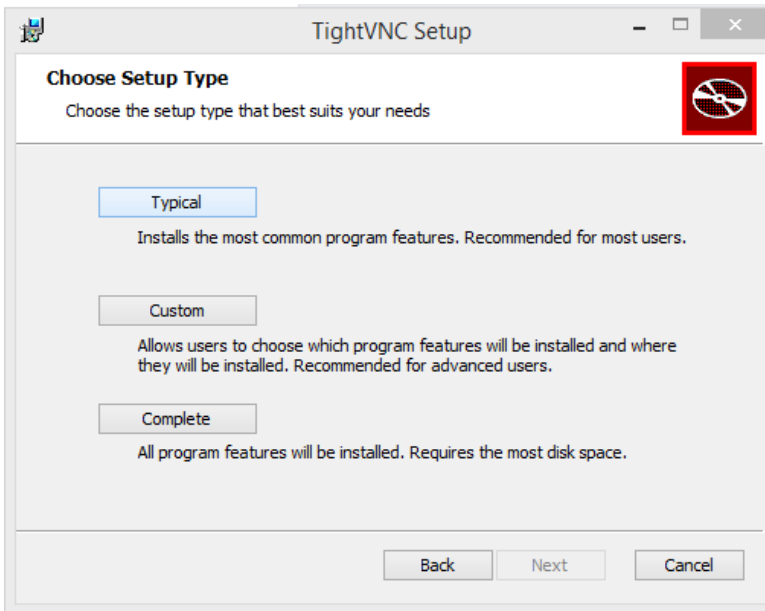
Sellega on VNC server seatud käivituma algaadimisel. Muudatuste salvestamiseks tuleb Raspberry taaskäivitada. Selle jaoks on käsk

- `sudo reboot`

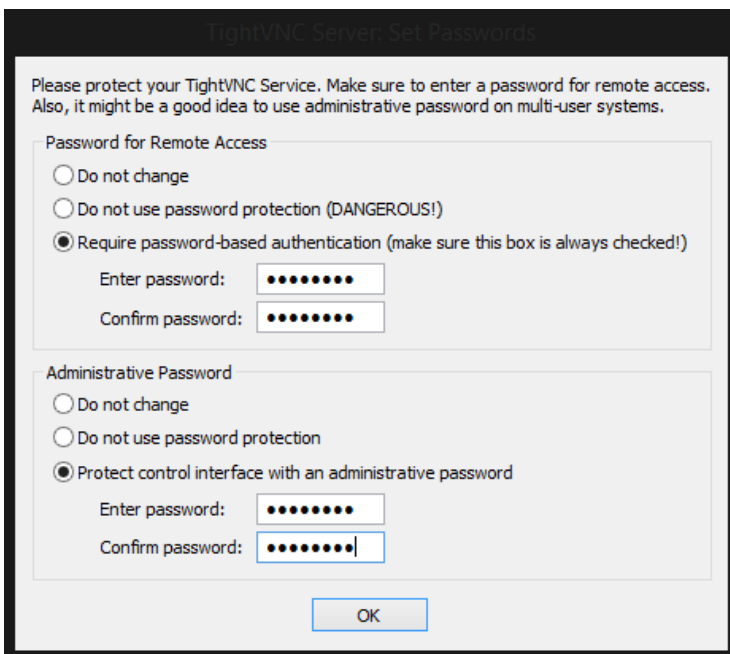
Pärast Raspberry üles seadmist tuleb arvutisse paigaldada VNC klient. Selle jaoks sobib Windowsi operatsioonisüsteemiga arvutitel TightVNC [35], mis hõlmab nii klienti kui ka serverit. Teiste tuntud operatsioonisüsteemidega on soovitatav kasutada RealVNC [36] tarkvara, mis ühildub ka Windowsi operatsioonisüsteemiga.

Kui vastav tarkvara on allalaetud, siis tuleb see käivitada ja paigaldada. Paigaldamise käigus küsitakse kasutajalt, millist tüüpi paigaldust sooritada. Mugavaim viis on valida 'Typical' (Joonis 15), mis paigaldab kõik põhilised programmi funktsioonid. Lisaks küsitakse kasutajalt, kuidas soovitakse serverit turvata. Soovitatav on jätta need seaded muutmata ning lahtritesse sisestada meelde jäävad kuni 8 tähemärki pikad paroolid (Joonis 15).





Joonis 15. VNC paigaldamine arvutisse (1).



Joonis 16. VNC paigaldamine arvutisse (2).

Pärast paigaldamist tuleb avada 'TightVNC Viewer' ehk VNC vaatur ning esimesse lahtrisse sisestada Raspberry aadress kujul IP:PORT, kus IP aadressiks on eelnevalt Raspberry seadetest välja uuritud aadress ning pordiks on 1, sest VNC seati töötama pordil 1. Tegelikult on VNC pordiks 5901, kuid kuna 5900 on vaike port, ei pea esimest osa sisestama. Ühendamisel küsitakse ka parooli, milleks tuleb sisestada Raspberry kasutaja 'pi' parooli. Kui parooli eelnevalt ei muudetud, siis on selleks 'raspberrypi'.

Korrektsete tegevuste tulemusena avaneb kasutajale Raspberry ekraani vaade. See vaade peaks olema identne vaatega, mis avanes siis, kui Raspberry külge oli ühendatud monitor. Järgmiseks on vaja ülesseada EV3 robot.

## 2.3 LEGO Mindstorms EV3 seadistamine

Järgnev tekst põhineb ev3dev püsivara veebilehel oleval ingliskeelsel õpetusel [37].

Nii nagu Raspberry operatsioonisüsteemide valik oli suur, on ka EV3'le võimalik paigaldada mitmeid erinevaid süsteeme, asendamaks tema originaalset Linuxil põhinevat tarkvara, näiteks ev3dev [38], leJOS [39], MonoBrick [40] jpt. EV3 puhul nimetatakse neid süsteeme püsivaradeks (ingl. k. firmware). Selline püsivara muutmine annab kasutajale rohkem võimalusi - kasutada endale sobivat programmeerimiskeelt, kasutada Linuxile iseloomulikke programme jne. Kuna püsivara laetakse EV3 juhtblokki mälukaardilt, siis jääb algne süsteem puutumata ning soovi korral võib mälukaardi eemaldada ja sel juhul laetakse käivitumisel algne süsteem. Käesolevas bakalaureusetöös on kasutusel ev3dev püsivara, tänu millele on võimalik EV3 robotile programme kirjutada peaaegu kõikides programmeerimiskeeltes, mida on võimalik kasutada Linuxiga, näiteks Python ja C.

Püsivara muutmiseks on vajalik [37]

1. kuni 32GB microSD mälukaart,
2. adapter, mis muudab microSD kaardi SD kaartiks,
3. arvuti, millega mälukaardile kopeerida uus süsteem,
4. win32DiskImager tarkvara, mis kopeerib andmed mälukaardile,
5. püsivara, mille saab allalaadida ev3dev GitHubi lehelt,
6. USB kaabel, et arvuti ja EV3 vahel suhelda.

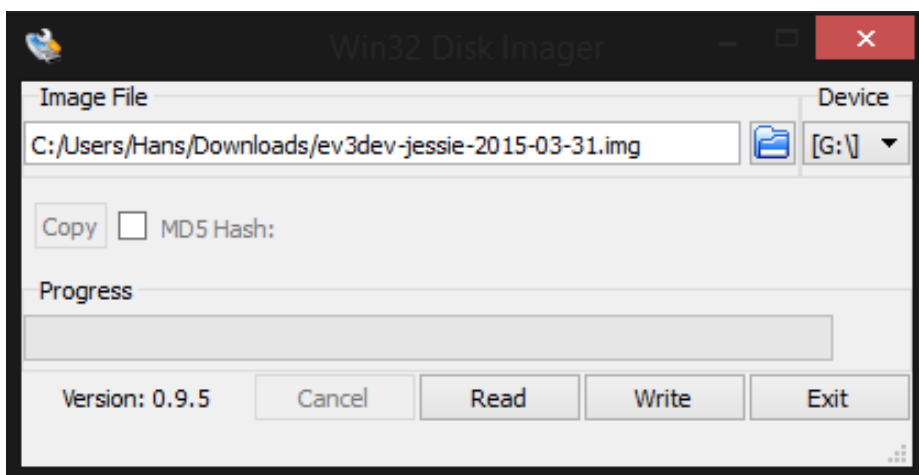
Püsivara paigaldamiseks tuleb kõigepealt valmis seada mälukaart, millele paigaldatakse püsivara tarkvara.

### 2.3.1 Mälukaardi ettevalmistamine

Nii nagu Raspberry mälukaart tuli vormida, tuleb ka EV3 mälukaardile paigaldada korrektne süsteem ning sellesse kopeerida uus püsivara. Raspbian süsteemi asemel tuleb mälukaardile

kopeerida hoopis ev3dev tarkvara. Selle saab ev3dev GitHubi lehelt (<https://github.com/ev3dev/ev3dev/releases>) kokkupakitud kujul allalaadida ning arvutis lahti pakkida. Kui kasutatava arvuti operatsioonisüsteem on Windows või Mac, siis tuleb allalaadida '.zip' laiendiga fail. Kui kasutatava arvuti operatsioonisüsteem on Linux, on otstarbekas allalaadida '.xz' laiendiga faili, sest see on väiksem. Ka Windowsi või Maci puhul võib kasutada '.xz' laiendiga faili, kuid seljuhul tuleb eelnevalt paigaldada lisa tarkvara selle lahtipakkimiseks. Seejärel tuleb allalaadida ning installeerida Win32DiskImager tarkvara (<http://sourceforge.net/projects/win32diskimager/>), mis vormindab mälukaardi ja kopeerib sellele uue tarkvara.

Pärast tarkvara paigaldamist tuleb see avada. Avanenud aknas tuleb vajutada sinisele nupule ning otsida üles lahtipakitud fail. See fail on laiendiga '.img'. Kui fail on leitud, siis tuleb programmis valida õige seade. Kui arvutis on ainult üks mälukaart sisestatud, siis peaks suutma Win32DiskImager selle kohe ise üles leida ja valida ning sel juhul ei saa ka ühtegi teist seadet valida. (Joonis 17)



Joonis 17. Win32 Disk Imager tarkvaraga mälukaardile süsteemi paigaldamine.

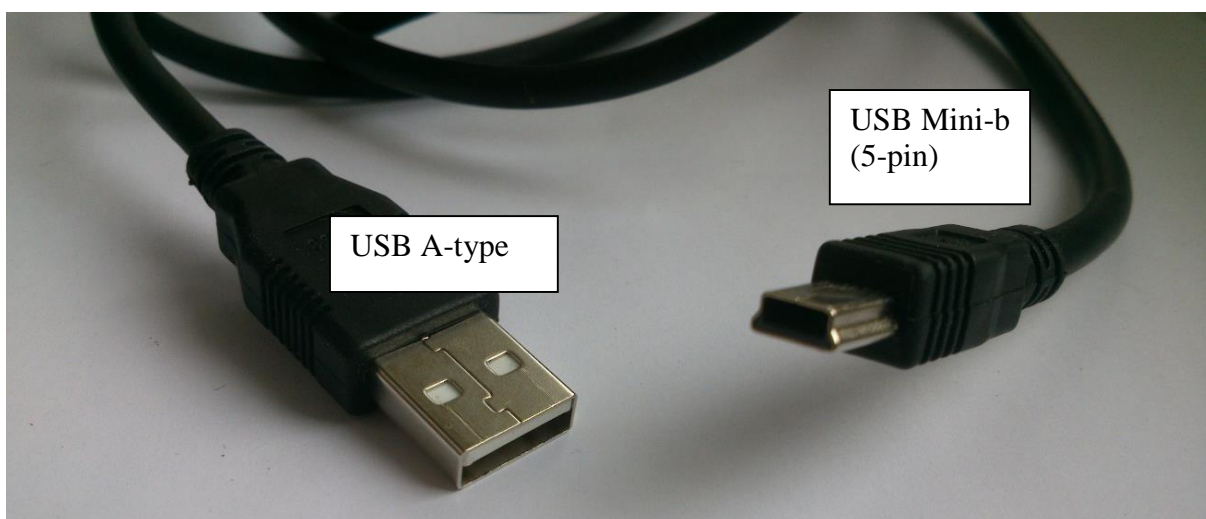
Seejärel tuleb vajutada nupule 'Write' ehk kirjuta, mis alustab andmete kirjutamist kaardile. Kasutajalt küsitakse ka kinnitust selle kohta, kas kasutaja on teadlik sellest, et füüsilisele seadmele kirjutades on risk seda kahjustada. Sellele hoiatusele tuleb vastata jaatavalt.

Kui programm on töö lõpetanud, siis teavitatakse kasutajate sellest ekraanile tekkiva hüpikakna kaudu. Seejärel võib Win32DiskImager programmi sulgeda, mälukaardi arvutist eemaldada ja sisestada EV3 juhtbloki SD kaardi pesa niimoodi, et kaardi klemmid jääksid allapoole ning

käivitada EV3 juhtbloki. Juhtbloki käivitamisel kuvatakse ekraanil LEGO Mindstorms logo ning seejärel ev3dev põhivara laadimise logo. Kui ev3dev põhivara on ära laadinud, hakkab laadima brickman tarkvara ehk ev3dev põhivara jaoks koostatud LEGO Mindstorms EV3 kasutajaliides. Peale brickman tarkvara laadimist on EV3 juhtblokk kasutamiseks valmis.

## 2.4 Raspberry Pi ja EV3 juhtbloki ühendamise

Võimalusi Raspberry Pi ja EV3 vahel ühendust luua on mitmeid – bluetooth, USB kaabel, Wi-Fi. Nendest lihtsaim on luua võrguühendus, kasutades USB kaablit, mille üks ots on A-tüüpi ning teine ots on mini-b tüüpi 5 jalaga (Joonis 18).



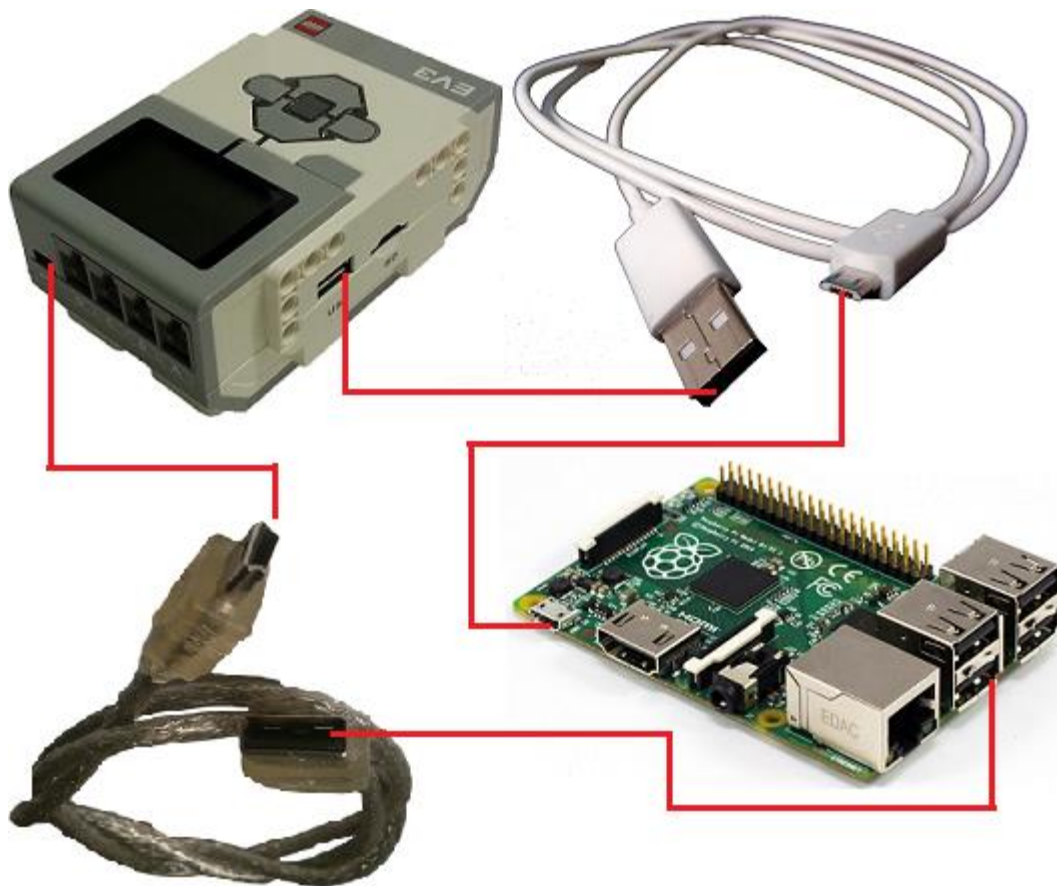
Joonis 18. USB kaabel Raspberry ja EV3 ühendamiseks.

Ühenduse loomiseks on vaja ühendada A-tüüpi ots Raspberry Pi USB kontakti ning Mini-b tüüpi ots kontakti, mis asub EV3 ekraani juures. Lisaks on vaja veel teha mõningad muudatused mõlema seadme seadistustes.

### 2.4.1 Raspberry Pi ja EV3 USB kaabliga ühendamise

Järgmine tekst tugineb ev3dev veebilehel asuvale Linuxi süsteemiga seadme ja ev3dev püsivaraga EV3 juhtbloki ühendamise ingliskeelsele õpetusele [41].

Esimese asjana tuleb lahti ühendada Raspberry küljest kõik juhtmed, välja arvatud toitejuhe, mille teine ots tuleb sisestada EV3 juhtbloki USB pessa, ning seejärel ühendada EV3 ja Raspberry Pi omavahel USB kaabliga (Joonis 19). Seejärel tuleb sisse lülitada EV3, vajutades juhtblokil keskmist nuppu. Kui EV3 tööle panna, siis hakkab ka Raspberry ise tööle.



Joonis 19. Ühendused EV3 juhtbloki ja Raspberry Pi vahel [42].

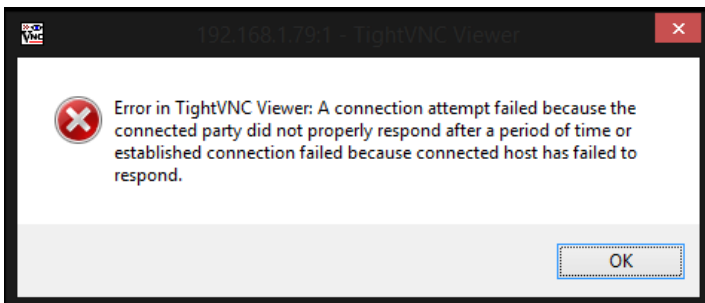
Kui EV3 on tööle hakanud, siis tuleb kontrollida üle, kas CDC draiver on aktiveeritud, sest Raspberry, mille külge robot ühendatakse, on Linux-i-põhise süsteemiga. Kasutades Windowsi operatsioonisüsteemiga arvutit, peab RNDIS olema sisselülitatud.

CDC kontrollimiseks tuleb liikuda alumise nupuga ekraanil valikule 'Wireless and Networks' ning sealt edasi valikule 'USB'. Järgmiseks tuleb vaadata, kas CDC kõrval on kirjas '(active)' või '(inactive)' (Joonis 20). Kui kirjas on '(inactive)', siis tuleb alumise nupuga liikuda CDC valiku peale ning see sisse lülitada, vajutades keskmist nuppu. Kui CDC kõrval olev tekst on '(active)', siis tuleb vajutada tagasi nuppu ning avanenud menüüs valida 'All Network Connections' ning seejärel 'Wired'.



Joonis 20. EV3 roboti võrgudraiveri valimise ekraani vaade [43].

Järgmiseks tuleb Raspberry ja arvuti vahel luua VNC ühendus. Kuna Raspberry juba töötab, tuleb arvutis avada VNC vaatur, mis sai eelnevalt installeeritud, ning selle kaudu Raspberry süsteemi sisselogida. Kui kuvatakse veateadet (Joonis 21), siis ei ole samal ajal võimalik Raspberryga ühendust luua. See võib tähendada, et Raspberry laadimine pole veel lõpule viidud või halvemal juhul on tekkinud mingi probleem. Enne probleemi lahendamist on mõttekas natuke aega oodata ja uuesti proovida. Probleemi esinemisel tuleb Raspberry külge uuesti ühendada klaviatuur, hiir ja monitor ning teha kindlaks, kas IP aadress sai õigesti sisestatud ja kas Raspberry on ikka raadiovõrguga ühenduses. Kui ühendus on võimalik luua, siis küsitakse kasutajalt parooli. Selleks parooliks on Raspberry kasutaja 'pi' parool. Kui seda ei ole muudetud, siis on selleks 'raspberry'.



Joonis 21. VNC kliendi ja serveri vahelise ühenduse loomise ebaõnnestumise veateade.

VNC ühenduse loomise järgselt tuleb avada Raspberry Terminal ning käivitada käsud

- `sudo echo 1 > /proc/sys/net/ipv4/ip_forward`
- `sudo nano /etc/sysctl.conf`

eemaldada kommentaari märk ehk '#' rea 'net.ipv4.ip\_forward=1' eest ning seejärel fail sulgeda. Nendest kahest esimene käsk lubab ajutiselt pakettide marsruutimise, teine käsk avab

muutmiseks faili 'sysctl.conf', kus lubatakse permanentsest pakettide marsruutimine. Lisaks on vaja suunata kõik sissetulevad paketid liidesesse wlan0. Seda saab teha käskudega [44]

- `sudo iptables -P FORWARD ACCEPT`
- `sudo iptables -t nat -I POSTROUTING -o wlan0 -j MASQUERADE`

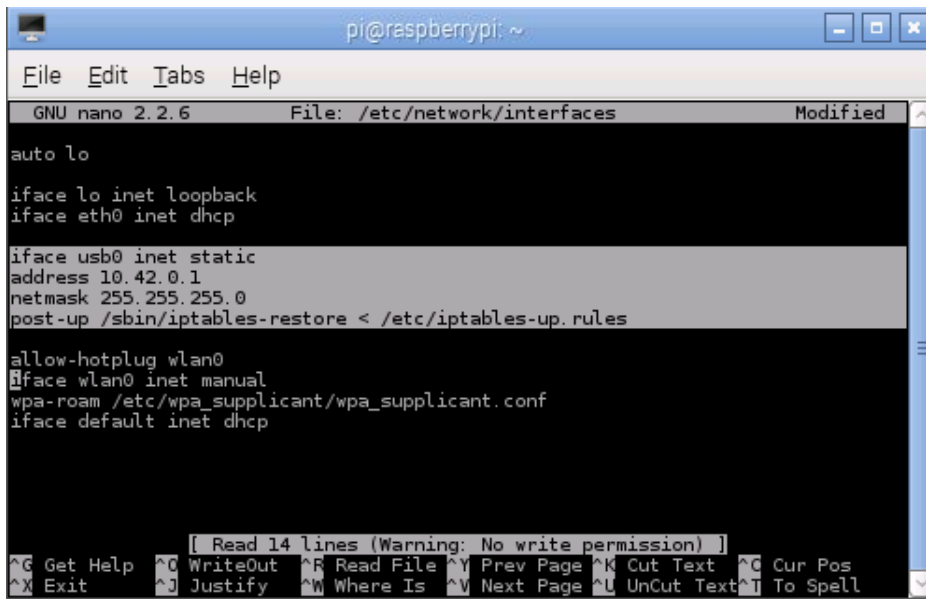
Järgmiseks on vaja muudatused salvestada faili ja need alati uuesti sisse lugeda, sest vastasel juhul need Raspberry taaskäivitamisel kaovad. Selle jaoks tuleb käivitada käsud

- `sudo bash -c „iptables-save >/etc/iptables-up.rules“`
- `sudo nano /etc/network/interfaces`

ning avanenud ekraanile lisada 'iface eth0 inet dhcp' alla järgmised 4 rida:

```
iface usb0 inet static
address 10.42.0.1
netmask 255.255.255.0
post-up /sbin/iptables-restore < /etc/iptables-up.rules
```

Käskudest teine avab administraatori õigustes muutmiseks faili 'interfaces', mis asub kaustas '/etc/network'. Esimeses reas kirjeldatakse, et liidesele usb0 lisatakse andmed staatiliselt. Järgmine rida määrab liidese aadressi ja kolmas määrab liidese võrgumaski. Neljas rida käsib 'iptables' faili lugeda sisse muudatused, mis eelnevalt salvestati. Need muudatused käivitatakse Raspberry laadimisel automaatselt. (Joonis 22)



```
pi@raspberrypi: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/network/interfaces Modified
auto lo
iface lo inet loopback
iface eth0 inet dhcp

iface usb0 inet static
address 10.42.0.1
netmask 255.255.255.0
post-up /sbin/iptables-restore < /etc/iptables-up.rules

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp

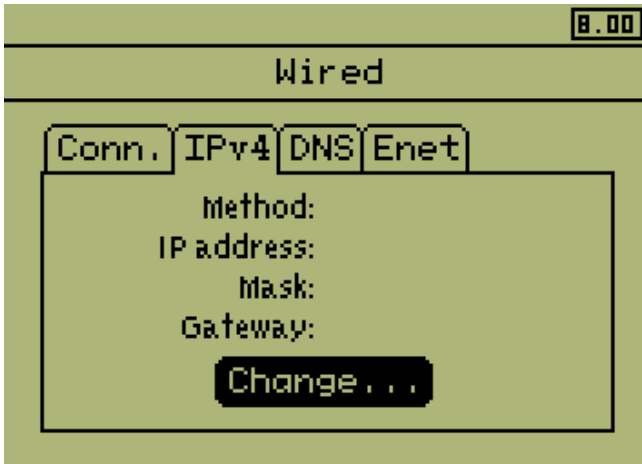
[ Read 14 lines (Warning: No write permission) ]
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^G Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Joonis 22. Raspberry USB0 võrguseadme konfigureerimine.

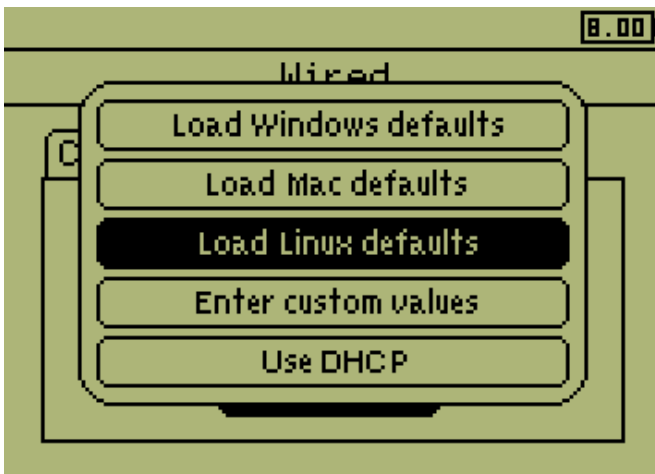
Muudatuste sisseviimiseks tuleb Raspberry taaskäivitada käsuga

- `sudo reboot`

Seejärel tuleb EV3 ekraanil liikuda paremale ning avada vaheleht 'IPv4', sealt liikuda alla ja valida 'Change...' (Joonis 23), seejärel 'Load Linux defaults' (Joonis 24). Ühendades Windowsi operatsioonisüsteemiga arvutiga, tuleb valida 'Load Windows defaults' ning Macintosh süsteemiga arvutiga ühendades tuleb valida 'Load Mac defaults'.



Joonis 23. EV3 roboti Interneti seadete muutmise ekraani vaade (1) [45].



Joonis 24. EV3 roboti Interneti seadete muutmise ekraani vaade (2) [46].

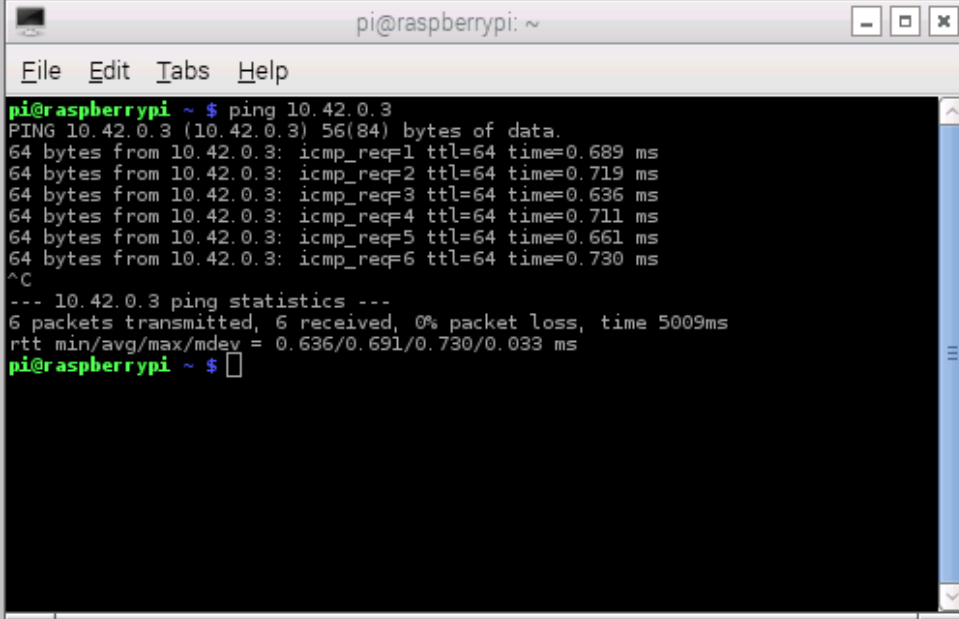
Ühenduse loomiseks EV3 ja Raspberry vahel tuleb lõpuks EV3 ühendus lubada. Selle jaoks avada 'Conn.' vaheleht ning liikuda alla, et märgistada valik 'Connect automatically' ning valida 'Connect'.

Ühenduse testimiseks võib Raspberry Terminalis käivitada käsu

- `ping 10.42.0.3`



mis saadab aadressile '10.42.0.3' kontrollpakette (Joonis 25). See aadress on ev3dev püsivaraga EV3-e vaikeadress. Käsk jääb tööle igavesti või seniks, kuni ta peatatakse. Käsu saab peatada, kui sulgeda Terminali aken või vajutada klahvikombinatsiooni CTRL + C.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ ping 10.42.0.3  
PING 10.42.0.3 (10.42.0.3) 56(84) bytes of data.  
64 bytes from 10.42.0.3: icmp_req=1 ttl=64 time=0.689 ms  
64 bytes from 10.42.0.3: icmp_req=2 ttl=64 time=0.719 ms  
64 bytes from 10.42.0.3: icmp_req=3 ttl=64 time=0.636 ms  
64 bytes from 10.42.0.3: icmp_req=4 ttl=64 time=0.711 ms  
64 bytes from 10.42.0.3: icmp_req=5 ttl=64 time=0.661 ms  
64 bytes from 10.42.0.3: icmp_req=6 ttl=64 time=0.730 ms  
^C  
--- 10.42.0.3 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5009ms  
rtt min/avg/max/mdev = 0.636/0.691/0.730/0.033 ms  
pi@raspberrypi ~ $
```

Joonis 25. Raspberry ja EV3 roboti vahelise ühenduse testimine.

Pärast seadmete vahelise ühenduse loomist on kõigepealt vaja ühenduda EV3-ga, et seal käivitada programme.

## 2.4.2 SSH ühendus Raspberry ja EV3 vahel

Kuna graafiline liides on EV3 roboti jaoks liiga ressursinõudlik, on kõige otstarbekam luua suhtlus nende kahe seadme vahel üle SSH. Selle jaoks tuleb Raspberry Terminali aknas käivitada käsk

- `ssh root@10.42.0.3`

See käsk loob SSH ühenduse aadressiga '10.42.0.3', kasutades sihtpunkti kasutajana juurkasutajat 'root'. Esmakordsel ühendumisel küsitakse kasutajalt, kas seda sihtpunkti võib usaldada. Sellele küsimusele tuleb vastata 'yes'. Seejärel küsitakse EV3 kasutaja 'root' parooli, milleks on 'r00tme'. (Joonis 26)



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ ssh root@10.42.0.3  
The authenticity of host '10.42.0.3 (10.42.0.3)' can't be established.  
ECDSA key fingerprint is 88:5a:41:d3:de:b7:ed:c2:af:35:6d:79:2e:35:df:ad.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '10.42.0.3' (ECDSA) to the list of known hosts.  
root@10.42.0.3's password:  
  
ev3dev  
  
Debian GNU/Linux jessie on LEGO MINDSTORMS EV3!  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sat Jan 1 00:11:33 2000 from 10.42.0.1  
root@ev3dev: ~#
```

Joonis 26. Turvalise ühenduse loomine Raspberry ja EV3 vahel.

Kohe peale ühendamist on soovitatv ära muuta ev3dev püsivara juurkasutaja 'root' parool. Seda saab teha, käivitades avatud aknas käsu

- `passwd`

ning 2 korda sisestada uut parooli.

### 2.4.3 EV3 Interneti ühenduse loomine

Proovides külastada näiteks 'www.google.com' veebilehte, siis EV3 ei saa aru, mida teha. Kui aga kasutada Google IP aadressi ehk '173.194.32.209', siis on seadmel sihtaadress teada. See tähendab, et juhtblokil puudub domeeninimede lahendamise võime.

Kuna EV3 kasutab võrgu haldamiseks 'connman' võrguhaldurit, siis ei saa nimeserveri aadresse lihtsalt faili kirjutada, vaid seda tuleb teha läbi võrguhalduri. Selle jaoks tuleb EV3 juhtblokis kõigepealt käivitada vastav keskkond käsuga

- `connmanctl`

ning selles keskkonnas teha järgmised sammud

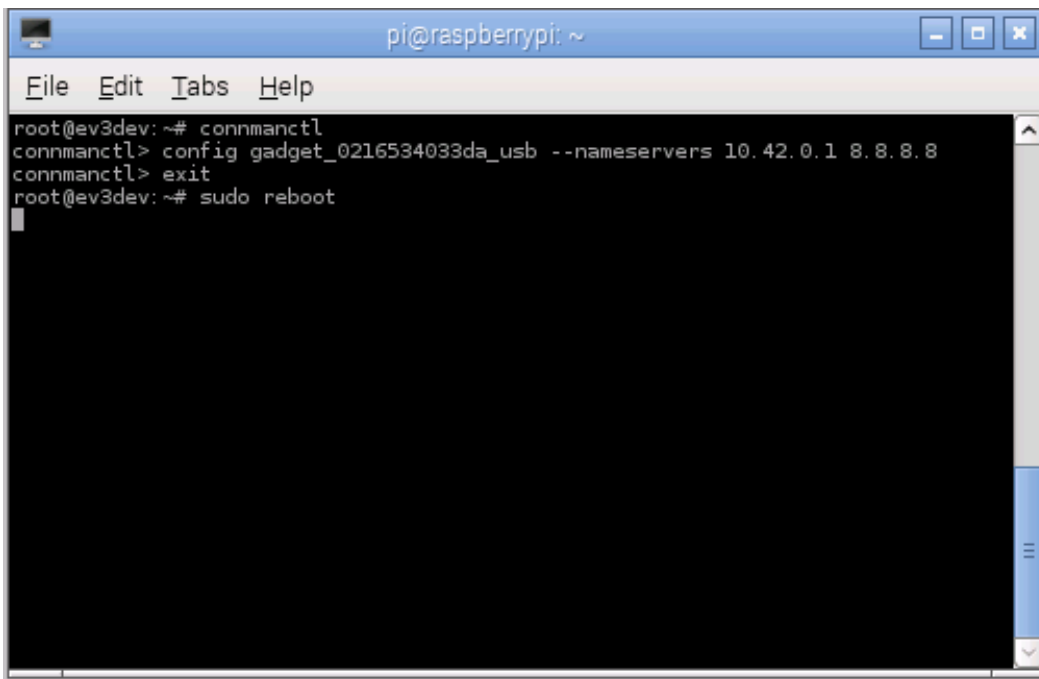
1. kirjutada 'config gadg'
2. vajutada tabulatsiooni klahvi, et süsteem leiaks ise sobiva vaste
3. kirjutada '--nameservers 10.42.0.1 8.8.8.8'

4. vajutada Enter klahvi
5. käivitada käsk 'exit'.

Nende sammudega võetakse kasutusele kaks nimeserverit. Esimene on Raspberry ning teine on Google'i nimeserver. Lõpetuseks tuleb seade taaskäivitada käsuga

- `sudo reboot`

mis lisaks robotile taaskäivitab ka Raspberry.(Joonis 27)



```
pi@raspberrypi: ~
File Edit Tabs Help
root@ev3dev:~# connmanctl
connmanctl> config gadget_0216534033da_usb --nameservers 10.42.0.1 8.8.8.8
connmanctl> exit
root@ev3dev:~# sudo reboot
```

Joonis 27. EV3 roboti nimeserverite lisamine ning taaskäivitamine.

Kui nimeserveriga on ühendus loodud, siis on võimalik robotisse repositooriumist allalaadida vajalikke teke.

## 2.5 EV3 sensorite ja mootorite programmeerimine

Tänu ev3dev püsivara olemasolule saab programmeerida EV3 jaoks programme mitmes erinevas keeles [47]. Programmeerimiskeele kasutamiseks peab EV3 robotisse olema laetud vastava keele teek (ingl. k. library). Mõningatele programmeerimiskeeltele teegid leiduvad, kuid mitte kõikidele. Kui vastavale keelele teeki veel ei ole, siis on programmeerimine keeruline ning aeganõudev. Teegi olemasolul tuleb see ainult alla laadida ning saab programmeerimist alustada.

### 2.5.1 Pythoni teegi paigaldamine

Järgnev tekst tugineb allikale GitHubi kasutaja 'topikachu' loodud teegi ingliskeelsele paigaldamis õpetusele [48].

Käesolevas juhendis olevad näited on esitatud programmeerimiskeeles Python2.7, mille selgeksõppimine on lihtne - Internetis leidub palju kursusi ja õpetusi [49] [50], ülikoolides on vastavaid kursusi ja õppematerjale [51]. Kasutusel on Pythoni vanem versioon 2.7, mitte 3.4, sest vanemale versioonile leidub rohkem lisateeke. GitHubi kasutaja 'topikachu' (Gong Yi) on kokku pannud projekti [48], millega saab Pythonis programmeerida EV3 sensoreid ja mootoreid.

Esmalt tuleb käima panna EV3 robot, vajutades klahvistiku keskmisele nupule. Käivitumisel saab toite ka Raspberry ning hakkab tööle. Alglaadimine võib võtta aega mõned minutid. Seejärel tuleb luua VNC ühendus arvuti ja Raspberry vahel ehk arvutis käivitada VNC vaatur ning luua ühendus Raspberry IP aadressiga.

Kui VNC ühendus on loodud, siis tuleb avada Raspberry Terminal ning luua SSH ühendus EV3 robotiga. Seda saab teha käsuga

- `ssh root@10.42.0.3`

ning sisestada 'root' kasutaja parool. Kui seda eelnevalt pole ära muudetud, on selleks 'r00tme'.

Järgmiseks on vaja uuendada EV3 püsivara paketid nagu seda tehti varem Raspberry pakettidega. Pakettide uuendamiseks tuleb EV3 käsureal käivitada käsk

- `sudo apt-get upgrade && sudo apt-get update`

mille sooritamine võib võtta tükk aega.

Seejärel tuleb alla laadida virtuaalse keskkonna failid ning Pythoni lisad. Seda kõike saab teha ühe käsuga

- `apt-get install virtualenv virtualenvwrapper python-setuptools python-smbus python-pil`

Pythonit programmeeritakse eraldi isoleeritud keskkonnas, kuhu saab installeerida erinevaid teeke, mis võivad segada teisi teeke või süsteemi faile.

Järgmiseks tuleb käivitada virtuaalkeskkonna pakendaja ja luua vastav keskkond. Selle jaoks on käsud

- `source /etc/bash_completion.d/virtualenvwrapper`
- `mkvirtualenv ev3_py27 --python=/usr/bin/python2.7 --system-site-packages`

Kui keskkond on üles seatud, tuleb see käivitada käsuga

- `workon ev3_py27`

ning siis sinna allalaadida ev3dev Pythoni teek

- `easy_install python-ev3`

Keskkonnast saab väljuda käsuga

- `deactivate`

ning alati, kui on soovi käivitada Pythoni programme, mis kasutavad EV3 mooduleid, tuleb EV3 robotis kõigepealt käivitada keskkond käsuga

- `workon ev3_py27`

Pärast Raspberry Pi ja EV3 juhtbloki seadistamist on võimalik alustada programmide kirjutamisega. Järgmises peatükis on toodud välja 7 järjestikust huvitavat ülesannet kasvava raskusastmega.

## 3 Python2.7 näidisprogrammid ja ülesanded

Järgnevad näidisprogrammid ning ülesanded annavad hea ülevaate Pythoni ning LEGO Mindstorms EV3 sensorite ja mootorite kasutusest. Ülesanded on toodud raskusastme kasvades ehk lihtsamad ülesanded on eespool ning keerulisemad lõpus. Iga järgnev ülesanne on eelmisest sõltuv ehk eelpool seletatud asju kasutatakse ka hilisemates programmides. Ülesannetes kasutatavate mootorite ja sensorite kohta saab abi otsida ev3dev Python moodulist (<https://github.com/topikachu/python-ev3>).

### 3.1 „Tere maailm!“

Iga programmeerimiskeele õppimine algab programmikoodi „Tere Maailm“ kirjutamisega, mis on üks lihtsamaid programme enamikes programmeerimiskeeltes ja algajatele sobiv. See programm väljastab käivitamisel ekraanile teksti „Tere Maailm“ või kasutaja valikul mingi muu teksti.

Esmalt tuleb Raspberry süsteemis avada Python 2 programmeerimiskeskond ehk IDLE. IDLE avamisel kuvatakse ekraanile valge aken - Pythoni kest ehk Shell -, mis toimib nagu Pythoni terminal. Programme võib kirjutada nii spetsiaalse programmi kui ka tekstiredaktori abil. Mõistlik on siiski programmeerida Pythoni enda tekstiredaktoris, mis rühmitab nähtavad koodi osad värvilistena, esitades need visuaalselt arusaadavamalt. Selleks vajutada klahvikombinatsiooni

- CTRL + N

või valida tegumirealt

- File -> New Window

Avanenud aknasse on võimalik koodi kirjutada samamoodi, nagu tekstiredaktorisse teksti. „Tere Maailm“ programmi jaoks tuleb aknasse kirjutada kood

```
print ("Hello World"),
```

salvestada fail klahvikombinatsiooniga

- CTRL + S

või valida tegumirealt

- File -> Save As ...

Seejärel tuleb valida faili salvestuskoht ja anda salvestatavale failile nimi. Lihtsaim viis on salvestada fail töölauale ning nimeks valida midagi asjakohast, näiteks 'TereMaailm.py', milles faililaiend '.py' tähistab, et tegu on Pythoni koodifailiga.

Programmi käivitamiseks tuleb avada Raspberry Terminali aken ehk käsirida ning suunduda salvestatud faili asukohta või sisestada käivitatava faili nimi koos täieliku asukohaga. Kui fail salvestati töölauale, siis on terminalis jooksva kataloogi muutmiseks käsk

- `cd Desktop`

Antud käsus 'cd' tähendab 'change directory' ehk 'muuda asukohta' ning 'Desktop' on inglise keeles 'töölaud'. Kui fail salvestati kuhugi mujale või mingisse alamkausta, siis on algselt võimalik kirjutada

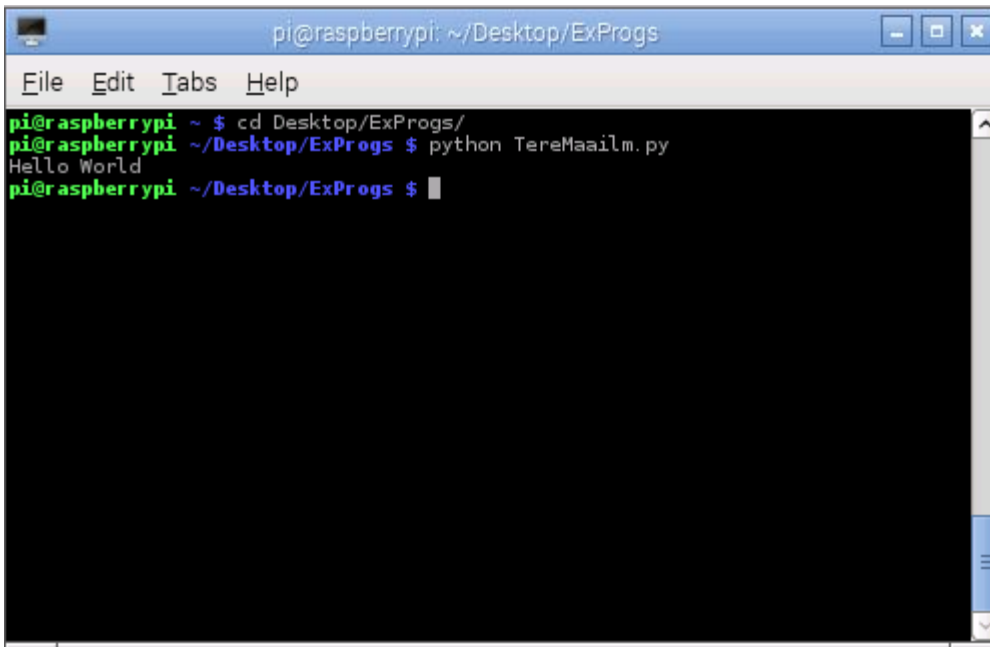
- `cd Desktop/alamkaust`

Lisaks suudab Terminal pakkuda otsitava objekti nime, millest on kasu näiteks juhul, kui faili nimi on keeruline või täpne nimi ununenud. Selleks tuleb Terminali sisestada objekti nimes esimesi tähti niikaua, kuni ilmub unikaalne tähekombinatsioon, millele sarnast antud asukohas ei esine ning seejärel vajutada klaviatuuril tabulatsiooni klahvi. Kui avatud asukohas on mitu sarnase nimega faili, siis kahekordsel tabulatsiooni klahvi vajutusel kuvatakse kõik sisestatud tekstiga sobivad vasted.

Olles Terminaliga õiges kaustas, tuleb seal käivitada käsk

- `python TereMaailm.py`

mis käivitab programmi 'python' ja annab talle argumendiks 'TereMaailm.py'. Seejärel käivitab Python antud koodifaili ning väljastab ekraanile teksti „Hello World“.(Joonis 28)



```
pi@raspberrypi ~ $ cd Desktop/ExProgs/
pi@raspberrypi ~/Desktop/ExProgs $ python TereMaailm.py
Hello World
pi@raspberrypi ~/Desktop/ExProgs $
```

Joonis 28. Linux terminalis asukoha vahetamine ja Pythoni programmi käivitamine.

Eespool esitatud kood `print ("Hello World")` oli näide selle kohta, kuidas väljastada teksti ekraanile. Paraku ei saa Python aru täpitähtedest: väljastades näiteks tekst 'mängumaja', annab Python veateate. Selle probleemi lahendamiseks tuleb koodifaili esimesel real defineerida faili kodeering ehk sinna kirjutada

```
# -*- coding: utf-8 -*-
```

mis määrab faili kodeeringuks UTF-8 ning siis saab lisaks ingliskeelse tähestiku tähtedele kasutada ka muid tähti, näiteks 'ä' ja '兩'.

## 3.2 Sisendid

Sisendite töötlemise ülevaateks sobib programm, mis küsib kasutajalt tema nime ning seejärel tervitab teda nimepidi. Selleks tuleb koodifaili sisestada kood

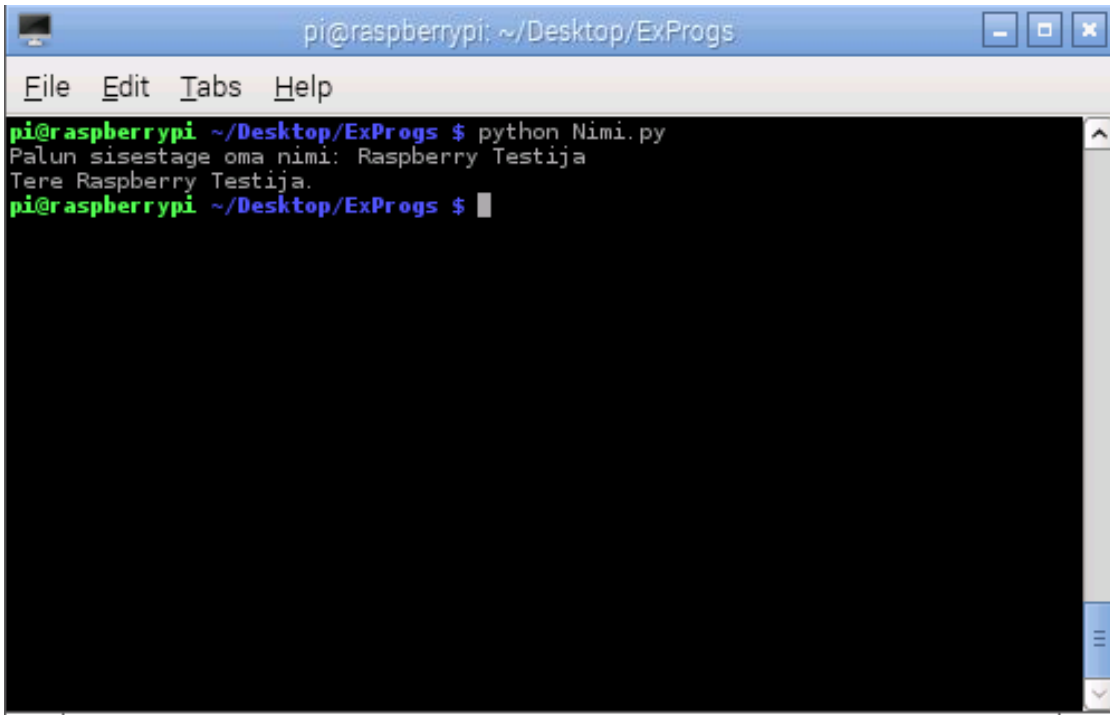
```
nimi = raw_input("Palun sisestage oma nimi: ")
print("Tere " + nimi + "!")
```

ning seejärel fail salvestada, suunduda Terminali aknas vastavasse kausta ja käivitada programm käsuga

- `python failinimi.py`



Seejärel kuvatakse ekraanile tekst, mis palub sisestada kasutaja nimi. Nime sisestamise kinnitamiseks on vaja vajutada Enter klahvile. Seejärel kuvatakse ekraanile tekst, kus tervitatakse sisestatud nime. (Joonis 29)



```
pi@raspberrypi: ~/Desktop/ExProgs
File Edit Tabs Help
pi@raspberrypi ~/Desktop/ExProgs $ python Nimi.py
Palun sisestage oma nimi: Raspberry Testija
Tere Raspberry Testija.
pi@raspberrypi ~/Desktop/ExProgs $
```

Joonis 29. Linux Terminalis Pythoni programmi käivitamine ja töö.

Rohkem lisainfot algajale ja lisa ülesandeid on võimalik leida näiteks Tartu Ülikooli Arvutiteaduste instituudi kursuste leheküljel olevast Pythoni õpikust ([https://courses.cs.ut.ee/MTAT.03.100/2012\\_fall/uploads/opik/01\\_sissejuhatus.html](https://courses.cs.ut.ee/MTAT.03.100/2012_fall/uploads/opik/01_sissejuhatus.html)).

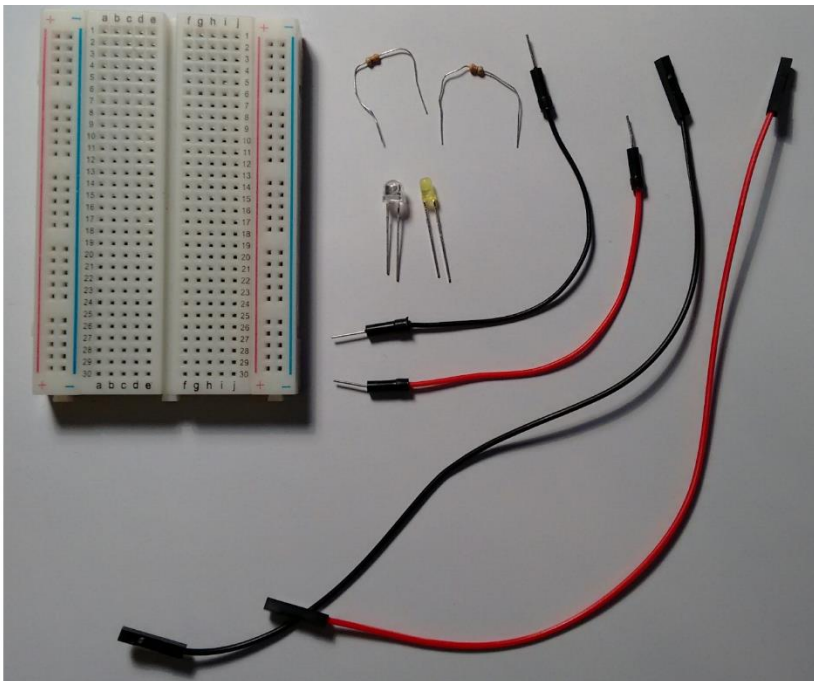
### 3.2.1 Ülesanded

- Muuta programmi nii, et ta küsiks kasutajalt eraldi sisenditena nime, vanust, elukohta ning siis tervitaks teda

## 3.3 LED pirni vilgutamine

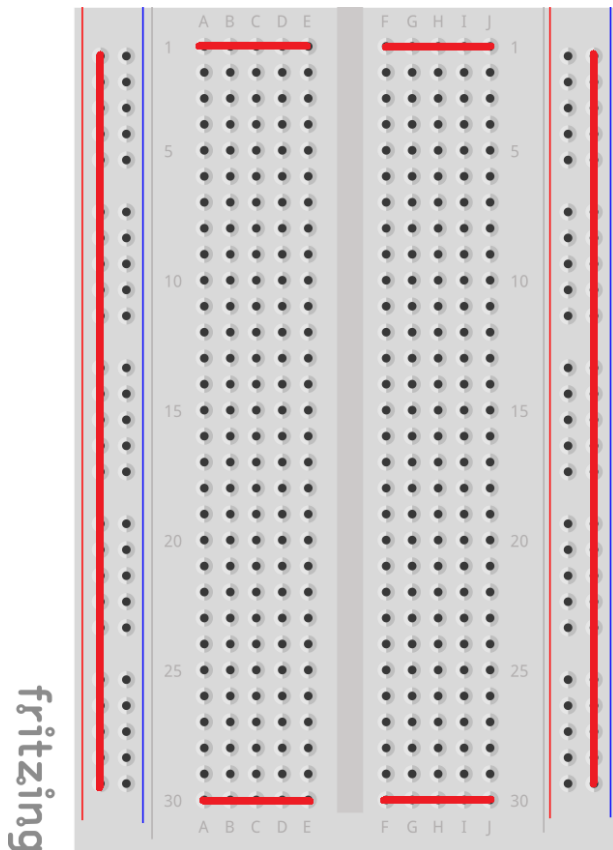
Eelmised ülesanded olid sissejuhatuseks Pythoni programmeerimiskeele kasutamisel ning töötasid igas seadmes, millesse oli Python paigaldatud, kuid järgnevad ülesanded on mõeldud spetsiaalselt Raspberry Pi peal kasutamiseks.

LED ehk valgusdiod on pooljuhtdiod, mis läbiva voolu toimel kiirgab valgust. Raspberry GPIO viiksid kasutades saab näiteks valgusdiodi sisse-välja lülitada. Selle ülesande sooritamiseks on vaja valgusdiodi, diodile vastavat takistit, juhtmeid ning makettlauda (Joonis 30).



*Joonis 30. Raspberry ja LED piri ühendamiseks vajalikud komponendid.*

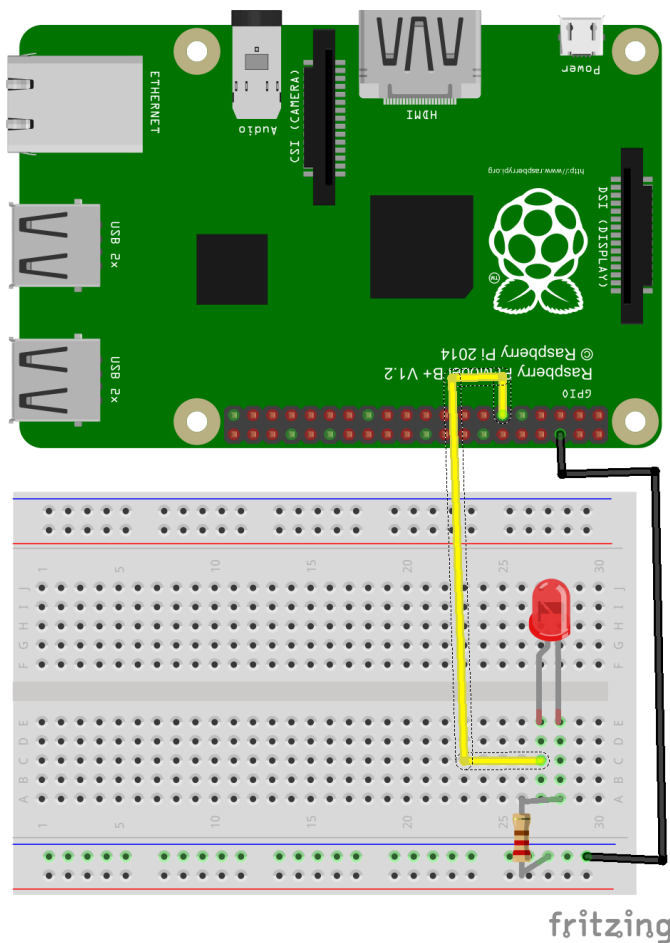
Vajalikud komponendid - juhtmed, diod ja takisti - tuleb omavahel ühendada. Ühendamiseks kasutatakse makettlauda. Makettlauda ühendused jooksevad pikuti mööda külgi. See tähendab, et külgedel jooksevad ühendused piki lauda ja keskel jooksevad ühendused küljelt küljele (Joonis 31).



Joonis 31. Maketlauru ühendatud augud märgitud punasega.

Juhtmed, diodid ja takisti tuleb maketlauru ning Raspberry külge ühendada järgmiselt: (Joonis 32)

- LED pihv maketlaurule
- 1. juhe GND viigust #6 maketlauru sinise joonega küljele
- 2. juhe GPIO17 viigust #11 maketlauru LED lambi pikema jalaga samasse ritta
- takisti maketlauru LED lambi lühikese jala juurest sinise joonega küljele.



Joonis 32. Raspberry ja LED piri vahelise ühenduse visualisatsioon.

Pärast komponentide ühendamist tuleb Raspberry IDLE aknasse kirjutada kood

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)

def blink(pin):
    GPIO.output(pin, GPIO.HIGH)
    time.sleep(0.1)
    GPIO.output(pin, GPIO.LOW)
    time.sleep(0.1)

try:
    blink(17)
finally:
    GPIO.cleanup()
```

See kood laeb sisse mooduli 'RPi.GPIO' [52] ning määrab talle nimeks 'GPIO'. Lisaks laetakse sisse modul 'time'. Järgmiseks antakse Raspberry'le korraldus kasutada BCM tähiseid.

Raspberry GPIO oskab kasutada kahte moodi tähiseid BCM ja BOARD. Kui kasutatakse BCM tähistust, siis on viigu numbriks tema nime number, näiteks GPIO17 #11, GPIO18 #12, GPIO27 #13 tähisteks on vastavalt 17, 18 ja 27. Kui kasutusel on aga BOARD tähistus, siis on viigu numbriks tema asukoht Raspberry küljes ehk eelmises näites vastavalt 11, 12 ja 13.

Peale tähistuse seadmist määratakse viik numbriga 17 ehk GPIO17 väljundiks. See tähendab, et selle viigu pinget saab lülitada sisse ja välja ehk kõrgeks ja madalaks, kus kõrge seadistus tähendab pinget 3,3V ja madal 0V. Viik saab olla ka sisend ning oskab sel juhul talle saadetavaid andmeid lugeda. Kuna Raspberry GPIO viigud on digitaalsed, siis andmed on kahendsüsteemis ehk kas 1 või 0.

Kui viik on seatud väljundiks, siis defineeritakse meetod nimega 'blink', mille argumendiks on element 'pin'. Meetodi sees seatakse viigu väljund kõrgeks, mis tähendab, et viik annab voolu välja. Seejärel lastakse programmil oodata 0,1 sekundit ehk 100 millisekundit ning lülitatakse viigu pinge välja ning lastakse programmil taas oodata 0,1 sekundit. Kuna programm töötab väga kiiresti, siis tagab paus selle, et on näha LED pirni põlemist või mittepõlemist.

Järgmiseks moodustatakse 'try-finally' blokk, milles proovitakse välja kutsuda meetod 'blink', andes talle argumendiks 17. Peale meetodit taastatakse kõikide antud programmis seatud viikude seaded algseadeteks ehk viigud muudetakse sisendiks [53].Pärast koodi kirjutamist tuleb fail salvestada ning käivitada. Kuna antud juhul muudetakse Raspberry riistvara, siis tuleb see programm käivitada 'root' õigustes, sisestades Terminali aknasse käsu

- `sudo python failinimi`

Pärast programmi käivitamist peaks LED vilkuma 1 korra.

Lisaks Pythoni materjalidele leidub Internetis hulgaliselt muid materjale, mille abil saab õppida Raspberry kasutamist [54] [55]. Enamus nendest on inglise keeles, kuid tänu visualiseerimisele on need siiski arusaadavad ka mittekeeleoskajaile.

### 3.3.1 Ülesanded

- LED panna põlema 3ks sekundiks
- LED panna vilkuma 10 korda
- Lisada veel üks LED ja vilgutada vaheldumisi

### 3.4 Pythoniga suhtlus EV3 ja Raspberry vahel

Võimalusi EV3 roboti ja Raspberry vahelise suhtluse korraldamiseks on mitmeid: I<sup>2</sup>C, UART, UDP jpt. I<sup>2</sup>C ja UART on riistvaralised jadaühenduse tüübid, mille realiseerimine on Raspberry Pi peal tehniliselt keerukas. UDP on võrguprotokoll, mille abil töötavad mitmed tänapäeva arvuti rakendused, näiteks heli ja video edastamine võrgus, nimelahendus, VPN ning mõningad mängud. Seega on UDP protokoll kasutamine lihtsam ning ei vaja eraldi komponentide manuaalset ühendamist.

UDP suhtluse loomiseks [56] peab eksisteerima vähemalt üks server ehk vastuvõtja ning vähemalt üks klient ehk saatja. Antud juhul on EV3 serveri rollis ning Raspberry kliendi rollis.

EV3 juhtbloki haldamiseks tuleb esmalt luua VNC ühendus arvuti ja Raspberry vahel ning seejärel SSH ühendus Raspberry ja EV3 vahel

- `ssh root@10.42.0.3`

ning seejärel käivitada Pythoni keskkond

- `workon ev3_py27` (Joonis 33)



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ ssh root@10.42.0.3  
root@10.42.0.3's password:  
ev3dev  
Debian GNU/Linux jessie on LEGO MINDSTORMS EV3!  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun May 3 14:27:30 2015 from 10.42.0.1  
root@ev3dev: ~# workon ev3_py27  
(ev3_py27) root@ev3dev: ~#
```

Joonis 33. SSH ühendus Raspberry ja EV3 roboti vahel ning virtuaalse keskkonna käivitamine.

Kuna EV3-l puudub graafiline liides, tuleb programm kirjutada Terminali tekstiredaktoris, näiteks nano, käsuga

- `nano UDPServer.py`

Kui selle nimega fail juba eksisteerib jooksvas kaustas, siis 'nano' käsk avab olemasoleva faili, mitte ei loo uut. Sellesse faili tuleb serveri loomiseks sisestada kood

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
listen_addr = ("", 9972)
sock.bind(listen_addr)
try:
    while True:
        data, addr = sock.recvfrom(1024)
        print(data.strip())
        sock.sendto("Tekst: " + data)
finally:
    sock.close()
```

Selles koodis laetakse sisse kõigepealt 'socket' moodul [57]. Seejärel luuakse uus sokkel, määratakse ära aadress ja port, mida kuulatakse ning seotakse see sokliga. Kuulataav aadress on tühisõne ehk kuulatakse kõiki saatjaid ning vastuvõtvaks pordiks on 9972. Pordi numbri võib kasutaja vabalt valida, kuid soovitatav on, et see jääks vahemikku 1024 – 65535.

Pärast sokli moodustamist luuakse varasemast tuttav 'try-finally' blokk, kus moodustatakse while-tsükkel. Tsükli sees võetakse vastu saatja aadress ning andmed maksimaalse suurusega 1024 baiti. Seda suurust nimetatakse puhvriks ning see peab olema kahe aste. Järgmiseks väljastatakse saadud andmed, mille lõpust on eemaldatud uue rea tähis. Siis saadetakse tagasi vastus. Kui tsükli sees tekib viga või programmi töö lõpetatakse, siis suundutakse 'finally' blokki ja sokkel suletakse.

Serveri koodi salvestamise järgselt tuleb Raspberry süsteemis käivitada UDP klient. Selleks tuleb luua uus Pythoni fail ning sinna sisestada kood

```
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
addr = ("10.42.0.3", 9972)
try:
    while True:
        msg = raw_input("Sisestage saadetav tekst: ")
        sock.sendto(msg, addr)
        recData, recAddr = sock.recvfrom(1024)
        print("Saadeti vastus: " + recData)
finally:
    sock.close()
```

Selles koodis laetakse samuti sisse 'socket' moodul ning luuakse sarnaselt serveriga sokkel. Aadressiks määratakse seekord serveri aadress ning port on sama, mida server kuulas ehk 9972. Järgmises 'try-finally' blokis olevas while-tsüklis küsitakse kasutajalt sisendit, mis saadetakse serverisse. Serverist vastuse saamiseks võib kuluda aega ning saadud tulemus väljastatakse Terminali aknasse (Joonis 34). Programmi sulgemisel või vea tekkimisel sokkel suletakse.

Kuna näites jääb klient vastust ootama, siis praeguses näites kliendiprogramm lukustub. See tähendab, et rea `recData, recAddr = sock.recvfrom(1024)` tõttu ootab klient serverilt andmeid ning programmi töö ei liigu edasi ning edasisi programmi osi ei käivitata enne, kui on saadud vastus. Antud juhul on see välja toodud ühenduse testimiseks ning tegelikkuses, kui vastust ei ole vaja, ei pea seda ka ootama. Kui server siiski saadab vastuse nii, et klient seda ei oota, siis seda ei võeta vastu ja see kaob ära.



```
pi@raspberrypi: ~/Desktop/ExProgs
File Edit Tabs Help
pi@raspberrypi ~/Desktop/ExProgs $ python UDPClient.py
Sisestage saadetav tekst: Tere!
Saadeti vastus: Tekst: Tere!
Sisestage saadetav tekst: Minu nimi on Henry
Saadeti vastus: Tekst: Minu nimi on Henry
Sisestage saadetav tekst:

(ev3_py27) root@ev3dev: ~# python UDPServer.py
Tere!
Minu nimi on Henry

```

Joonis 34. Klient (ülemine aken) küsib kasutajalt sisendit ja edastab selle serverisse (alumine aken). Server väljastab sisendi ja saadab vastuse kliendile.

### 3.4.1 Ülesanded

- Koostada programm, mille abil klient tervitab serverit ning seejärel küsib server kliendilt eraldi sisenditena nime, vanust ja elukohta ja väljastab need serveris.
- Koostada programm, kus klient, saades serverist vastuse, paneb põlema LED piri kaheks sekundiks

### 3.5 Puutesensori abil LED pirni vilgutamine

LEGO puutesensori abil Raspberry valgusdiodi vilgutamise ülesanne annab hea ülevaate, kuidas panna Raspberry ja EV3 juhtblokk omavahel suhtlema. Selle ülesande jaoks on lisaks vaja EV3 sensori kaablit ning puutesensorit. Puutesensor tuleb kaabliga ühendada EV3 juhtbloki vabalt valitud numbriga tähistatud porti. Seejärel tuleb Raspberry külge ühendada LED pirn (Joonis 32).

Kui komponendid on ühendatud, siis tuleb EV3 juhtblokkis käivitada programm, mis loeb, kas sensor on alla vajutatud ning kui see on tõene, edastab mingi teate Raspberry peal töötavasse programmi. Raspberry peal töötav programm võtab teate vastu ning lülitab LED pirni vastavalt sisse või välja, muutes külgehaagitud GPIO viigu pinget kas kõrgeks või madalaks. Antud näites on Raspberry serveri rollis ning EV3 on klient.

EV3 juhtblokkis töötavaks programmiks sobib järgnev kood

```
from ev3.lego import TouchSensor
import socket
import time

ts = TouchSensor(port=4)
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
addr = ("10.42.0.1", 9972)
try:
    while True:
        if ts.is_pushed:
            sock.sendto("True", addr)
            time.sleep(0.3)
finally:
    sock.close()
```

Selles koodis peaks kõik olema juba tuttav, välja arvatud puutesensori osa. Selleks laetakse sisse moodul 'TouchSensor' [58], mis annab kasutajale kontrolli ühendatud puutesensorite üle. Järgmiseks väärtustatakse muutuja 'ts' väärtuseks puutesensor, mis on ühendatud pordi number neli külge. Tsüklis kontrollitakse, kas puutesensor on alla vajutatud. Selleks kontrollitakse, kas 'ts.is\_pushed' väärtus on True ehk tõene või False ehk väär. Kui väärtus on tõene, siis saadetakse UDP sokli kaudu sõnum Raspberry programmi.

Raspberry osa selles ülesandes on võtta vastu sõnumeid ning muuta viigu pinget. Selle kontrollimiseks sobib järgnev kood

```
import RPi.GPIO as GPIO
import socket
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
listen_addr = ("", 9972)
sock.bind(listen_addr)
pressed = False
try:
    while True:
        data, addr = sock.recvfrom(1024)
        if (data == "True"):
            pressed = not pressed
        if (pressed):
            GPIO.output(17, GPIO.HIGH)
        else:
            GPIO.output(17, GPIO.LOW)
finally:
    sock.close()
    GPIO.cleanup()
```

Ka siin koodis peaks kõik olema tuttav. Laetakse sisse sokli ja viikude moodulid, seatakse üles nii sokkel kui viigud. Tsükli sees oodatakse andmeid ning kui andmed on sobivad, siis lülitatakse viigu pinge kas kõrgeks või madalaks. Programmi töö lõppedes suletakse sokkel ning viikude seaded taastatakse algväärtusteks.

### 3.5.1 Ülesanded

- Lisada veel puutesensoreid, mis kontrollivad sama LED pirni
- Muuta üks sensor leedi sisselülitajaks ning teine väljalülitajaks
- Lisada LED pirne nii, et erinevad sensorid kontrolliks erinevaid LED pirne

## 3.6 EV3 juhtbloki liigutamine nooleklahvidega

Juhtbloki liigutamiseks on esmalt vaja ehitada ratastega robot. Selleks tuleb EV3 juhtbloki külge kinnitada kaks mootorit ning mõlema mootori külge ratas. Rattad võiksid asuda roboti esiosas ning tagumisse otsa tuleks seljuhul paigaldada LEGO omniratas (Joonis 35). Mootori külge paigaldatud rataste abil saab robot liikuda ning end pöörata. Klahvivajutused loetakse

sisse Raspberry Terminali aknas ning saadetakse UDP sokliga EV3 juhtbloki, mis kontrollib rattaid.



Joonis 35. LEGO ommiratas, mida kasutatakse toetusrattana.

Kui robot on valmis meisterdatud, siis on vaja Raspberry süsteemis käivitada kood, mis loeb kasutaja sisendeid ning edastab need sobival kujul EV3 juhtblokki. Juhtblokk peab oskama saadetud sõnumeid vastu võtta ning vastavalt andmetele panna tööle või peatada vastavaid mootoreid. Selleks on vaja Raspberry andmekandjale tekitada uus koodifail ning sinna kirjutada kood

```
import curses
import socket

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
def main(screen):
    global sock
    addr = ("10.42.0.3", 9972)
    stdscr = curses.initscr()
    while True:
        d = stdscr.getch()
        if (d == curses.KEY_UP):
            sock.sendto("UP", addr)
        elif (d == curses.KEY_DOWN):
            sock.sendto("DOWN", addr)
        elif (d == curses.KEY_LEFT):
            sock.sendto("LEFT", addr)
        elif (d == curses.KEY_RIGHT):
            sock.sendto("RIGHT", addr)
        else:
            try:
                sock.sendto(chr(d), addr)
```

```

except:
    pass
if __name__ == '__main__':
    curses.wrapper(main)
    sock.close()

```

Selles koodis loetakse sisse kõigepealt kaks moodulit – 'curses' [59] ja 'socket'. 'Curses' moodul lubab Terminali aknal lugeda ning töödelda klahvi- ning hiire vajutusi. Järgmiseks luuakse uus sokkel, mis hakkab suhtlema EV3 juhtblokiga. Seejärel defineeritakse meetod 'main', mille argumendi nimeks on 'screen'. Seda varianti on vaja kasutada selleks, et programmi töö lõppedes taastuks Terminali akna esialgne vaade. Meetodis defineeritakse globaalne muutuja 'sock'. See tähendab, et ennist loodud sokli andmeid on võimalik muuta 'main' meetodis ning ka teised meetodid näevad neid muudatusi. Lisaks väärtustatakse 'addr' muutuja sokli ühenduse jaoks vajalike andmetega. Enne andmete lugemise tsükli käivitatakse 'curses' moodul.

Kui kõik vajalik on valmis seatud, siis hakkab programm tsükli sees kuulama kasutaja klahvivajutusi. Kõik klahvivajutused registreeritakse numbriliste väärtustega ning igale numbrile vastab mingi klahv. Kui klahvivajutus on 'curses.KEY\_UP' ehk klaviatuuril vajutati ülesnoolt, siis edastab programm UDP sokli kaudu aadressile '10.42.0.3' üle pordi '9972' sõnumi „UP“. Kui aga klahvivajutus on 'curses.KEY\_DOWN' ehk klaviatuuril vajutati allanoolt, siis edastatakse sõnum „DOWN“. Need valikud on programmis eeldefineeritud ning saadetavaid sõnumeid võib kasutaja vastavalt soovile kohandada. Kui aga klahvivajutus ei ole ükski noolte klahvidest, siis proovitakse klahvivajutuse numbrist moodustada täht või sümbol ning siis edastatakse see serverisse. Kui ka see ei õnnestu, siis jätkatakse programmi tööd midagi edastamata.

Programmikoodi viimases blokis märgitakse ära, et kui programm tahab käivitada 'main' meetodit, siis käivitatakse ta curses pakendi sees. Tänu sellele säilib Terminali algolek ning ei esine viperusi. Peale 'main' meetodi tööd suletakse ennist avatud sokkel.

Kui koodifail salvestatud, siis tuleb EV3 juhtblokis luua serveri koodifail, mis suudaks saadetud andmeid töödelda. Selleks tuleb tekitada uus fail järgneva sisuga:

```

import socket
from ev3.ev3dev import Motor

```

```

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
listen_addr = ("", 9972)
sock.bind(listen_addr)
a = Motor(port=Motor.PORT.A)
d = Motor(port=Motor.PORT.D)
try:
    while True:
        data, addr = sock.recvfrom(1024)
        print("Nupp: " + data)
        if data == 'UP':
            a.run_forever(100, regulation_mode=False,
stop_mode=Motor.STOP_MODE.BRAKE)
            d.run_forever(100, regulation_mode=False,
stop_mode=Motor.STOP_MODE.BRAKE)
        elif data == 'DOWN':
            a.run_forever(-70, regulation_mode=False,
stop_mode=Motor.STOP_MODE.BRAKE)
            d.run_forever(-70, regulation_mode=False,
stop_mode=Motor.STOP_MODE.BRAKE)
        elif data == 'LEFT':
            a.run_forever(50, regulation_mode=False,
stop_mode=Motor.STOP_MODE.BRAKE)
            d.stop()
        elif data == 'RIGHT':
            a.stop()
            d.run_forever(50, regulation_mode=False,
stop_mode=Motor.STOP_MODE.BRAKE)
        else:
            a.stop()
            d.stop()
finally:
    a.stop()
    d.stop()
    sock.close()

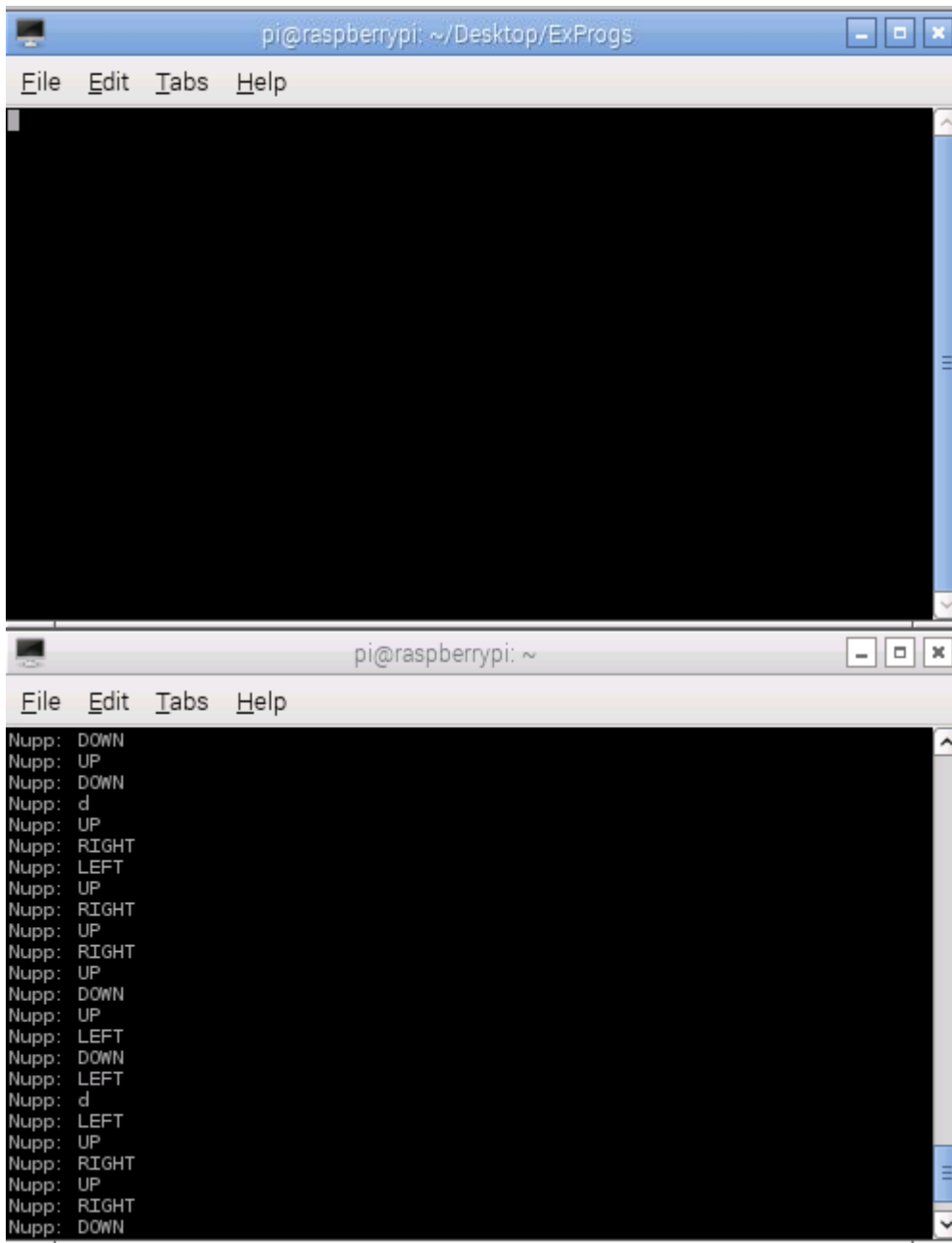
```

Selles koodis laetakse kõigepealt sisse moodul 'socket' ning seejärel moodul 'Motor' [60] paketist 'ev3.ev3dev'. Mooduliga 'Motor' on võimalik kontrollida LEGO Mindstorms EV3 mootoreid. Järgmiseks luuakse sokkel ning määratakse see kuulama kõiki aadresse pordil '9972'. Seejärel väärtustatakse muutujale 'a' väärtuseks mootor, mis on ühendatud EV3 juhtbloki porti tähisega A. Sama tehakse muutujaga 'd', kuid siis on mootor ühendatud porti tähisega D.

Peale eelseadistusi proovib programm läbida tsüklit, kus esmalt oodatakse UDP sokli kaudu andmeid ning seejärel need väljastatakse. Kui saadud andmeteks on sõnum „UP“, siis

käivitatakse mõlemad mootorid 100% töövõimega ning mootorite peatamiseks kasutatakse pidurdamist, mitte võimsuse mahakeeramist, et peatumine toimuks koheselt. Kui sõnumiks on „DOWN“, siis käivitatakse mootorid töövõimsusega -70 ehk pannakse liikuma tagurpidi võimsusega 70%. Kui aga sõnumiks on „LEFT“ või „RIGHT“ ehk vasak või parem, siis peatatakse vajalik mootor ning teise võimsuseks määratakse 50%. Kui kasutaja on vajutanud mingile sümboliga klahvile, siis peatatakse mõlemad mootorid. Erindi tekkimisel või programmi sulgemisel peatatakse mõlemad mootorid ning sokkel suletakse.

Kui mõlemad failid salvestatud, siis tuleb nad käivitada. Otstarbekas on käivitada programmid eraldi Terminali akendes, aktiivseks teha aken, milles käivitati klient ning vajutada klaviatuuril mõnele nooleklahvile (Joonis 36).



*Joonis 36. Aktiivses kliendi aknas (ülemine aken) püütakse kinni klahvi vajutused ning saadetakse EV3 serverisse (alumine aken) kus saadetud sõnumid väljastatakse.*

Kui kõik töötab korrektselt, siis peaks robot hakkama liikuma. Roboti liikuma hakkamisel võib klahvi vabastada ja robot liigub edasi. Peatamiseks tuleb programm sulgeda või vajutada suvalisele klahvile, mis ei ole nooleklahv.



### 3.6.1 Ülesanded

- Lisada 3 LEDi ja panna üks põlema, kui robot sõidab otse, teine, kui robot pöörab vasakule ja kolmas, kui robot pöörab paremale.
- Lisada piisavalt LED pirne ning moodustada nendest esi-, tagu- ja suunatud
- Peatada robot ainult 'q' klahvi vajutusega või programmi sulgedes

### 3.7 Asünkroonne suhtlus EV3 juhtbloki ja Raspberry vahel

Andmete vahetamiseks kahe seadme vahel ei ole sünkroonne suhtlus otstarbekas: kui saadetakse andmeid, ilma et teine neid ootaks, võivad nad kaduma minna, samas kui andmeid on vaja saata, aga programm hoopis ootab neid, lukustub programm ega liigu edasi. Selliste olukordade vältimiseks on vaja suhtlus teha asünkroonselt [61] ehk mitmelõimeliselt. See tähendab, et kui muidu töötab programm ainult ühe lõimega, siis asünkroonse programmi puhul on lõimi rohkem ning nad ei sega üksteise tööd. Antud juhul töötab andmeid ootav UDP sokkel tagataustal ning kuulab liiklust ning esiplaanil töötab programm. Soklisse andmete saabumisest teavitatakse sellest teist lõime ning saadetakse talle andmeid.

Selle implementeerimiseks tuleb olemasolevat programmikoodi veidi ümber teha. Vaja on lisada asünkroonne kuulaja [62]. Näide selle lisamisest on järgmine

```
import curses
import socket
import SocketServer
import threading

data = ""
stdscr = ""

class ThreadedUDPRequestHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        global data
        global stdscr
        data = self.request[0].strip()
        stdscr.addstr("Saabunud andmed: "+data+"\n")
        stdscr.refresh()

class ThreadedUDPServer(SocketServer.ThreadingMixIn,
SocketServer.UDPServer):
    pass
```

```

def main(screen):
    global data
    global stdscr
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    addr = ("10.42.0.3", 9972)
    stdscr = curses.initscr()
    curses.echo()
    try:
        while True:
            d = stdscr.getch()
            if (d == curses.KEY_UP):
                sock.sendto("UP", addr)
            elif (d == curses.KEY_DOWN):
                sock.sendto("DOWN", addr)
            elif (d == curses.KEY_LEFT):
                sock.sendto("LEFT", addr)
            elif (d == curses.KEY_RIGHT):
                sock.sendto("RIGHT", addr)
            else:
                try:
                    sock.sendto(chr(d), addr)
                except:
                    pass
    finally:
        sock.close()

if __name__ == '__main__':
    server = ThreadedUDPServer(("", 9972),
ThreadedUDPRequestHandler)
    server_thread = threading.Thread(target=server.serve_forever)
    server_thread.daemon = True
    server_thread.start()
    curses.wrapper(main)
    server.shutdown()

```

See kood on asünkroonseks UDP suhtluseks muudetud juhtbloki liigutamise ülesande Raspberry kood. Programmi alguses loetakse lisaks sisse 'SocketServer' [63] ja 'threading' [64] moodulid. 'Threading' moodul aitab kasutajal luua uusi lõimi ning neid kontrollida. 'SocketServer' moodul on mõeldud sokli serveri loomiseks. Seejärel väärtustatakse kaks muutujat, et nendesse hiljem andmeid salvestada ja mitmes meetodis globaalselt kasutada. Järgmisena defineeritakse meetod 'ThreadedUDPRequestHandler', mis on antud juhul asünkroonne andmete vastuvõtja. See meetod võtab andmed vastu ning salvestab need globaalsesse muutujasse 'data'. Kuna kasutusel on 'curses' moodul, siis muutuja 'stdscr' abil väljastatakse saadud andmed Terminali aknasse. Meetod 'ThreadedUDPServer' loob UDP serveri. 'Main' meetod on sarnane eelmise ülesande 'Main' meetodiga. Andmete saatmiseks loodav sokkel ei lukusta lõime, sest tegu on UDP suhtlusega mitte TCP ühendusega.

Koodi viimases osas luuakse UDP server ning 'threading' mooduli abil luuakse serverile eraldi lõim. Lõim muudetakse 'daemoniks' ehk tagaplanil jooksvaks programmiks ning käivitatakse. Seejärel pannakse tööle mooduli 'curses' pakendis 'main' meetod. Kui 'main' meetod on töö lõpetanud, siis suletakse server ning koos sellega ka lõim.

Samamoodi tuleb ka EV3 juhtbloki programm muuta asünkroonseks. Näidiseks sobib järgnev kood

```
import SocketServer
import threading
import socket
from ev3.ev3dev import Motor
from ev3.lego import UltrasonicSensor

data = ""
addr = ("10.42.0.1", 9972)

class ThreadedUDPRequestHandler(SocketServer.BaseRequestHandler):
    def handle(self):
        global data
        data = self.request[0].strip()

class ThreadedUDPServer(SocketServer.ThreadingMixIn,
SocketServer.UDPServer):
    pass

def robot():
    global data
    global addr
    a = Motor(port=Motor.PORT.A)
    d = Motor(port=Motor.PORT.D)
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        while True:
            if(data != ""):
                print("Nupp: " + data)
            if data == 'UP':
                a.run_forever(100,
stop_mode=Motor.STOP_MODE.BRAKE)
                d.run_forever(100,
stop_mode=Motor.STOP_MODE.BRAKE)
            elif data == 'DOWN':
                a.run_forever(-70,
stop_mode=Motor.STOP_MODE.BRAKE)
```

```

        d.run_forever(-70,
stop_mode=Motor.STOP_MODE.BRAKE)
        elif data == 'LEFT':
            a.run_forever(50,
stop_mode=Motor.STOP_MODE.BRAKE)
            d.stop()
        elif data == 'RIGHT':
            a.stop()
            d.run_forever(50,
stop_mode=Motor.STOP_MODE.BRAKE)
        elif data == 'q':
            a.stop()
            d.stop()
            data = ""
    finally:
        a.stop()
        d.stop()
        sock.close()

if __name__ == "__main__":
    server = ThreadedUDPServer(("", 9972),
ThreadedUDPRequestHandler)
    server_thread =
threading.Thread(target=server.serve_forever)
    server_thread.daemon = True
    server_thread.start()
    robot()
    server.shutdown()

```

See kood on muudetud samal viisil, nagu on muudetud Raspberry koodi: faili alguses on asünkroonne kuulaja, robot-meetodis luuakse sokkel andmete saatmiseks ning viimases osas luuakse lõim ning käivitatakse see. Robot-meetodis on toimunud aga lisaks mõned väiksemad muutused – enne 'data' muutuja sisu väljastamist kontrollitakse, et see ei oleks tühi ja mootorid peatab ainult 'q' klahvile vajutus.

### 3.7.1 Ülesanded

- Lisada ultrahelisensor ning meetodi 'dist\_cm' abil peatada robot, kui eesolev eseme on lähemal kui 10cm.
- Lisada valgussensor ning tagastada ees oleva eseme värvus kasutades meetodit 'color'. Tagastatav värvus on vahemikus [0-7], kus 0:värvitu, 1:must, 2:sinine, 3:roheline, 4:kollane, 5:punane, 6:valge ja 7:pruun.

### **3.8 Edasiarendus**

Pärast asünkroonse suhtluse loomist ning ev3dev Pythoni teegiga tutvumist võib olemasolevat robotit edasi arendada arenenud toaanduriks, millele saab külge lisada veebikaamera, temperatuurisensori ja valgussensori. Kuna robotit saab juhtida üle Interneti, siis tänu veebikaamerale saab näha kuhu robot liigub. Lisaks on võimalik koostada roboti juhtimiseks mobiilsetele seadmetele vastav rakendus.

Lisaks toaandurile on võimalik muuta robot turvasüsteemiks, lisades robotile veebikaamera ning juhtmevabad kõlarid. Liikumist tuvastades väljastatakse kõlarite kaudu häiresignaal ning saadetakse kasutaja e-maili kontole vastav teade.

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli anda ülevaade Raspberry Pi miniarvutist ja LEGO Mindstorms EV3 robotist, koostada seadmete jaoks eestikeelne kasutusõpetus ja ülesanded. Eesmärgi saavutamiseks tutvus autor mõlema seadme võimalustega, ühendas seadmed ning koostas vastavad eestikeelsed juhendid ja ülesanded.

Töö esimeses osas anti ülevaade Raspberry Pi miniarvutist ja LEGO Mindstorms EV3 robotist ning võrreldi miniarvutit teiste sarnaste seadmetega. Töö teises osas koostati eestikeelne juhend nende kahe seadme seadistamiseks, riistvaraliseks ühendamiseks ja suhtluse loomise ettevalmistuseks. Välja toodi vajatavad lisa komponendid ning kuidas neid on vaja eelnevalt ettevalmistada. Kolmandas osas koostati programmeerimiskeeles Python huvitavad näidisülesanded, mis tutvustab mõlema seadme võimalusi. Lisaks näidisülesannetele koostati kasutajale iseseisvaks lahendamiseks kasvava keerukusega ülesanded, mille lahendamine on eelnevaid näidiseid kasutades edasiarendav. Ülesannete lahendamisel tuleb lahendada probleeme ning tutvuda vastava Pythoni teegiga süvenenumalt.

Töö käigus valminud juhendi abil on kasutajal võimalik ehitada robot, mida on võimalik klaviatuuri abil kaugjuhtida, erinevate andurite abil peatuda või suunda muuta ning vastavaid valgusdioode vilgutada. Iseseisva arenduse abil on võimalik robotile lisada USB kaamera, erinevaid sensoreid ja palju muud ning juhtida roboti, ise samal ajal eemal viibides.

Töö tulemusena valminud eestikeelne juhend on sobilik igas eas kasutajale, kel on soovi teha tutvust robotika ning elektroonikaga ning isegi kogenenud insenerile, kes plaanib oma kätega muuta oma kodust keskkonda turvalisemaks ja lõbusamaks.

## Viited

- [1] C. Wiebe, „TheSwitchLinux,“ Blogspot, 4 12 2012. [Võrgumaterjal]. Available: <http://theswitchtolinux.blogspot.com/2012/12/raspberry-pi-daily-deviations-picture.html>. [Kasutatud 8 Mai 2015].
- [2] Adrian, „Mod my Pi,“ 25 Märts 2015. [Võrgumaterjal]. Available: <http://www.modmypi.com/blog/raspberry-pi-gpio-sensing-motion-detection>. [Kasutatud 8 Mai 2015].
- [3] wpnsmith, „Instructables,“ [Võrgumaterjal]. Available: <http://www.instructables.com/id/Raspberry-Pi-Temperature-Humidity-Network-Monitor/>. [Kasutatud 8 Mai 2015].
- [4] V. Halapuu ja A. Valk, „Haridus- ja Teadusministeerium,“ 2013. [Võrgumaterjal]. Available: [http://www.hm.ee/sites/default/files/piaac\\_esmased\\_tulemused.pdf](http://www.hm.ee/sites/default/files/piaac_esmased_tulemused.pdf). [Kasutatud 8 Mai 2015].
- [5] A. Ferrari, „Institute for Prospective Technological Studies,“ August 2013. [Võrgumaterjal]. Available: <http://ftp.jrc.es/EURdoc/JRC83167.pdf>. [Kasutatud 8 Mai 2015].
- [6] A. Ferrari, „Haridus- ja Teadusministeerium,“ 2013. [Võrgumaterjal]. Available: [https://www.hm.ee/sites/default/files/digipadevuse\\_ensehindamise\\_raamistik\\_0.pdf](https://www.hm.ee/sites/default/files/digipadevuse_ensehindamise_raamistik_0.pdf). [Kasutatud 8 Mai 2015].
- [7] „Raspberry Pi Foundation,“ [Võrgumaterjal]. Available: <https://www.raspberrypi.org/>. [Kasutatud 8 Mai 2015].
- [8] What is Raspberry Pi? - Raspberry Pi,“ [Võrgumaterjal]. Available: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. [Kasutatud 8 Mai 2015].

- [9] A. Cat, „Raspberry Pi - Wikipedia, the free encyclopedia,“ 7 Mai 2015. [Võrgumaterjal]. Available: [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi). [Kasutatud 8 Mai 2015].
- [10] T. Relph-Knight, „Ten Raspberry Pi 2 alternatives | ZDNet,“ 17 Veebruar 2015. [Võrgumaterjal]. Available: <http://www.zdnet.com/pictures/ten-raspberry-pi-2-alternatives/>. [Kasutatud 8 Mai 2015].
- [11] A. Allan, „Arduino Uno vs BeagleBone vs Raspberry Pi | Make:,“ 15 Aprill 2013. [Võrgumaterjal]. Available: <http://makezine.com/2013/04/15/arduino-uno-vs-beaglebone-vs-raspberry-pi/>. [Kasutatud 8 Mai 2015].
- [12] T. Klosowski, „How to Pick the Right Electronics Board for Your DIY Project,“ 11 November 2013. [Võrgumaterjal]. Available: <http://lifelife.com/how-to-pick-the-right-electronics-board-for-your-diy-pr-742869540>. [Kasutatud 8 Mai 2015].
- [13] J. Brodtkin, „Raspberry Pi 2 arrives with quad-core CPU, 1GB RAM, same \$35 price | Ars Technica,“ 2 Veebruar 2015. [Võrgumaterjal]. Available: <http://arstechnica.com/information-technology/2015/02/raspberry-pi-2-arrives-with-quad-core-cpu-1gb-ram-same-35-price/>. [Kasutatud 8 Mai 2015].
- [14] „Raspberry Pi 2 Model B - Raspberry Pi,“ Veebruar 2015. [Võrgumaterjal]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>. [Kasutatud 8 Mai 2015].
- [15] Banedon, „Cubestormer 3 - Wikipedia, the free encyclopedia,“ 27 Märts 2015. [Võrgumaterjal]. Available: [http://en.wikipedia.org/wiki/Cubestormer\\_3](http://en.wikipedia.org/wiki/Cubestormer_3). [Kasutatud 8 Mai 2015].
- [16] B7T, „Braigo - Wikipedia, the free encyclopedia,“ 4 Mai 2015. [Võrgumaterjal]. Available: <http://en.wikipedia.org/wiki/Braigo>. [Kasutatud 8 Mai 2015].



- [17] „Lego Mindstorms EV3 - Wikipedia, the free encyclopedia,“ 5 April 2015. [Võrgumaterjal]. Available: [http://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_EV3](http://en.wikipedia.org/wiki/Lego_Mindstorms_EV3). [Kasutatud 8 Mai 2015].
- [18] „About EV3 - LEGO.com,“ [Võrgumaterjal]. Available: <http://www.lego.com/en-gb/mindstorms/about-ev3>. [Kasutatud 8 Mai 2015].
- [19] M. Gibbs, „Ten operating systems for the Raspberry Pi | Network World,“ 3 November 2014. [Võrgumaterjal]. Available: <http://www.networkworld.com/article/2842678/ten-operating-systems-for-the-raspberry-pi.html>. [Kasutatud 8 Mai 2015].
- [20] G. Clarke, „Microsoft eyes slice of Raspberry Pi with free Windows 10 sprinkled on top | The Register,“ 2 Veebruar 2015. [Võrgumaterjal]. Available: [http://www.theregister.co.uk/2015/02/02/microsoft\\_eyes\\_slice\\_of\\_raspberry\\_pi\\_with\\_free\\_windows\\_10/](http://www.theregister.co.uk/2015/02/02/microsoft_eyes_slice_of_raspberry_pi_with_free_windows_10/). [Kasutatud 8 Mai 2015].
- [21] „Develop Windows IoT apps – Windows IoT development,“ [Võrgumaterjal]. Available: <https://dev.windows.com/en-US/iot>. [Kasutatud 8 Mai 2015].
- [22] „Pi MusicBox - A Spotify, SoundCloud, Google Music player for the Raspberry Pi, with remote control,“ [Võrgumaterjal]. Available: <http://www.woutervanwijk.nl/pimusicbox/>. [Kasutatud 8 Mai 2015].
- [23] „About | Kodi,“ [Võrgumaterjal]. Available: <http://kodi.tv/about/>. [Kasutatud 8 Mai 2015].
- [24] „Introducing the New Out Of Box Software (NOOBS) - Raspberry Pi,“ [Võrgumaterjal]. Available: <https://www.raspberrypi.org/introducing-noobs/>. [Kasutatud 8 Mai 2015].

- [25] Korovamilk, „RPi SD cards - eLinux.org,“ 24 Aprill 2015. [Võrgumaterjal]. Available: [http://elinux.org/RPi\\_SD\\_cards#Working\\_2F\\_Non-working\\_SD\\_cards](http://elinux.org/RPi_SD_cards#Working_2F_Non-working_SD_cards). [Kasutatud 11 Mai 2015].
- [26] „Home - SD Association,“ [Võrgumaterjal]. Available: <http://www.sdcard.org/>. [Kasutatud 8 Mai 2015].
- [27] „Raspberry Pi Documentation,“ [Võrgumaterjal]. Available: <https://www.raspberrypi.org/documentation/installation/noobs.md>. [Kasutatud 8 Mai 2015].
- [28] M. C. Sean McManus, „How to Use Raspi-config to Set Up Your Raspberry Pi - For Dummies,“ Aprill 2013. [Võrgumaterjal]. Available: <http://www.dummies.com/how-to/content/how-to-use-raspiconfig-to-set-up-your-raspberry-pi.html>. [Kasutatud 8 Mai 2015].
- [29] [Võrgumaterjal]. Available: <https://www.raspberrypi.org/documentation/installation/images/noobs.png>. [Kasutatud 8 Mai 2015].
- [30] Dsimic, „Overclocking - Wikipedia, the free encyclopedia,“ 2 Aprill 2015. [Võrgumaterjal]. Available: <http://en.wikipedia.org/wiki/Overclocking>. [Kasutatud 8 Mai 2015].
- [31] „RPi Powered USB Hubs - eLinux.org,“ 3 Mai 2015. [Võrgumaterjal]. Available: [http://elinux.org/RPi\\_Powered\\_USB\\_Hubs](http://elinux.org/RPi_Powered_USB_Hubs). [Kasutatud 8 Mai 2015].
- [32] Savageautomate, 29 Detsember 2012. [Võrgumaterjal]. Available: <http://elinux.org/File:Edimax-ew-7811un.jpg>. [Kasutatud 13 Mai 2015].
- [33] Tociiek, „RPi USB Wi-Fi Adapters - eLinux.org,“ 29 Aprill 2015. [Võrgumaterjal]. Available: [http://elinux.org/RPi\\_USB\\_Wi-Fi\\_Adapters](http://elinux.org/RPi_USB_Wi-Fi_Adapters). [Kasutatud 8 Mai 2015].

- [34] „Installing VNC | Adafruit's Raspberry Pi Lesson 7. Remote Control with VNC | Adafruit Learning System,“ 4 Mai 2015. [Võrgumaterjal]. Available: <https://learn.adafruit.com/adafruit-raspberry-pi-lesson-7-remote-control-with-vnc>. [Kasutatud 8 Mai 2015].
- [35] „TightVNC: VNC-Compatible Free Remote Control / Remote Desktop Software,“ [Võrgumaterjal]. Available: <http://www.tightvnc.com/>. [Kasutatud 8 Mai 2015].
- [36] „RealVNC remote access & control software for desktop and mobile,“ [Võrgumaterjal]. Available: <http://www.realvnc.com/>. [Kasutatud 8 Mai 2015].
- [37] D. L. Ralph Hempel, „Getting Started with ev3dev,“ [Võrgumaterjal]. Available: <http://www.ev3dev.org/docs/getting-started/>. [Kasutatud 8 Mai 2015].
- [38] D. L. Ralph Hempel, „ev3dev,“ [Võrgumaterjal]. Available: <http://www.ev3dev.org/>. [Kasutatud 8 Mai 2015].
- [39] „LeJOS, Java for Lego Mindstorms,“ [Võrgumaterjal]. Available: <http://www.lejos.org/>. [Kasutatud 8 Mai 2015].
- [40] „MonoBrick.DK | Home of MonoBrick,“ [Võrgumaterjal]. Available: <http://www.monobrick.dk/>. [Kasutatud 8 Mai 2015].
- [41] „Connecting to the Internet via USB,“ [Võrgumaterjal]. Available: <http://www.ev3dev.org/docs/tutorials/connecting-to-the-internet-via-usb/>. [Kasutatud 8 Mai 2015].
- [42] „Shopify,“ [Võrgumaterjal]. Available: [http://cdn.shopify.com/s/files/1/0188/6794/products/B\\_Pi\\_1\\_of\\_4\\_1024x1024.JPG](http://cdn.shopify.com/s/files/1/0188/6794/products/B_Pi_1_of_4_1024x1024.JPG). [Kasutatud 8 Mai 2015].
- [43] [Võrgumaterjal]. Available: <http://www.ev3dev.org/images/brickman/usb-cdc-active.png>. [Kasutatud 8 Mai 2015].

- [44] „Operatsioonisüsteemid - Kursused - Arvutiteaduse instituut,“ [Võrgumaterjal]. Available: <https://courses.cs.ut.ee/2013/os/fall/Main/Praktikum12>. [Kasutatud 8 Mai 2015].
- [45] [Võrgumaterjal]. Available: <http://www.ev3dev.org/images/brickman/networking-ipv4-tab-change-selected.png>. [Kasutatud 8 Mai 2015].
- [46] [Võrgumaterjal]. Available: <http://www.ev3dev.org/images/brickman/networking-load-linux-defaults.png>. [Kasutatud 8 Mai 2015].
- [47] „Getting Started with ev3dev,“ [Võrgumaterjal]. Available: <http://www.ev3dev.org/docs/getting-started/#writing-programs>. [Kasutatud 8 Mai 2015].
- [48] GongYi, „topikachu/python-ev3 · GitHub,“ 24 Jaanuar 2015. [Võrgumaterjal]. Available: <https://github.com/topikachu/python-ev3>. [Kasutatud 8 Mai 2015].
- [49] „Python | Codecademy,“ [Võrgumaterjal]. Available: <http://www.codecademy.com/tracks/python>. [Kasutatud 8 Mai 2015].
- [50] „Learn Python - Free Interactive Python tutorial,“ [Võrgumaterjal]. Available: <http://www.learnpython.org/>. [Kasutatud 8 Mai 2015].
- [51] „1. Sissejuhatus - Programmeerimise algkursus,“ 2011. [Võrgumaterjal]. Available: [https://courses.cs.ut.ee/MTAT.03.100/2012\\_fall/uploads/opik/01\\_sissejuhatus.html](https://courses.cs.ut.ee/MTAT.03.100/2012_fall/uploads/opik/01_sissejuhatus.html). [Kasutatud 8 Mai 2015].
- [52] „Raspberry Pi Documentation,“ [Võrgumaterjal]. Available: <https://www.raspberrypi.org/documentation/usage/gpio/>. [Kasutatud 5 Mai 2015].
- [53] alex, „RPi.GPIO basics 3 - How to Exit GPIO programs cleanly, avoid warnings and protect your Pi » RasPi.TV,“ 13 Juuli 2013. [Võrgumaterjal]. Available:

<http://raspi.tv/2013/rpi-gpio-basics-3-how-to-exit-gpio-programs-cleanly-avoid-warnings-and-protect-your-pi>. [Kasutatud 8 Mai 2015].

- [54] „RaspberryPiIVBeginners - YouTube,“ [Võrgumaterjal]. Available: [https://www.youtube.com/channel/UCRAvo5cQWyfog8nRzlf\\_jWg](https://www.youtube.com/channel/UCRAvo5cQWyfog8nRzlf_jWg). [Kasutatud 8 Mai 2015].
- [55] „Gaven MacDonald - YouTube,“ [Võrgumaterjal]. Available: <https://www.youtube.com/channel/UCfMxbH6WR35780HqCw7eDiA>. [Kasutatud 8 Mai 2015].
- [56] S. Moon, „Programming udp sockets in python,“ 16 Oktoober 2012. [Võrgumaterjal]. Available: <http://www.binarytides.com/programming-udp-sockets-in-python/>. [Kasutatud 8 Mai 2015].
- [57] „17.2. socket — Low-level networking interface &mdash; Python 2.7.10rc0 documentation,“ [Võrgumaterjal]. Available: <https://docs.python.org/2/library/socket.html#module-socket>. [Kasutatud 8 Mai 2015].
- [58] J. Friis, „python-ev3/lego.py at master · topikachu/python-ev3 · GitHub,“ 20 Jaanuar 2015. [Võrgumaterjal]. Available: <https://github.com/topikachu/python-ev3/blob/master/ev3/lego.py>. [Kasutatud 8 Mai 2015].
- [59] „15.11. curses — Terminal handling for character-cell displays &mdash; Python 2.7.10rc0 documentation,“ [Võrgumaterjal]. Available: <https://docs.python.org/2/library/curses.html>. [Kasutatud 8 Mai 2015].
- [60] GongYi, „python-ev3/ev3dev.py at master · topikachu/python-ev3 · GitHub,“ 22 Jaanuar 2015. [Võrgumaterjal]. Available: <https://github.com/topikachu/python-ev3/blob/master/ev3/ev3dev.py>. [Kasutatud 8 Mai 2015].

- [61] Jandalhandler, „Asynchronous communication - Wikipedia, the free encyclopedia,“ 3 September 2014. [Võrgumaterjal]. Available: [http://en.wikipedia.org/wiki/Asynchronous\\_communication](http://en.wikipedia.org/wiki/Asynchronous_communication). [Kasutatud 8 Mai 2015].
- [62] „Tutorial :: Python :: Python Udp Server Python Implementation Tutorial,“ [Võrgumaterjal]. Available: <http://www.lampdev.org/programming/python/python-udp-server-python-implementation-tutorial.html>. [Kasutatud 8 Mai 2014].
- [63] „20.17. SocketServer — A framework for network servers &mdash; Python 2.7.10rc0 documentation,“ [Võrgumaterjal]. Available: <https://docs.python.org/2/library/socketserver.html#module-SocketServer>. [Kasutatud 8 Mai 2015].
- [64] „16.2. threading — Higher-level threading interface &mdash; Python 2.7.10rc0 documentation,“ [Võrgumaterjal]. Available: <https://docs.python.org/2/library/threading.html>. [Kasutatud 8 Mai 2015].
- [65] „Alphao[.]org | Why not share your knowledge ?,“ [Võrgumaterjal]. Available: [http://wordpress.alphao.org/wp-content/uploads/2015/01/REV\\_A5A.jpg](http://wordpress.alphao.org/wp-content/uploads/2015/01/REV_A5A.jpg). [Kasutatud 8 Mai 2015].
- [66] „Australian Robotics,“ [Võrgumaterjal]. Available: [http://www.australianrobotics.com.au/sites/default/files/imagecache/product\\_full/11021-01a.jpg](http://www.australianrobotics.com.au/sites/default/files/imagecache/product_full/11021-01a.jpg). [Kasutatud 8 Mai 2015].
- [67] „imgur: the simple image sharer,“ [Võrgumaterjal]. Available: <http://i.stack.imgur.com/ChpU5.jpg>. [Kasutatud 8 Mai 2015].

# LISAD

## I. Litsents

Mina **Hans Aarne Liblik** (sünnikuupäev: 15.05.1993)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsents) enda loodud teose  
**Raspberry Pi kasutamine koos LEGO Mindstorms EV3 robotiga,**  
mille juhendajad on Anne Villems, Alo Peets ja Taavi Duvin,
  - 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **14.5.2015**