

TARTU ÜLIKOOL
Loodus- ja tehnoloogiateaduskond
Tehnoloogiainstituut

Taavi Karelson

**NUTISEADMEST KAUGJUHITAVA ROBOTI
EHITAMINE RASPBERRY PI NÄITEL**

Bakalaureusetöö

Juhendaja: Alo Peets

Tartu 2015

SISUKORD

| | |
|--|----|
| Sissejuhatus..... | 3 |
| 1. Robotite kasutamise võimalused koolides..... | 4 |
| 1.1. Programmeerimiskeeled..... | 4 |
| 1.2. Koolides kasutatavad robotid ja arendusplaadid..... | 6 |
| 2. Roboti loomine | 10 |
| 3. Tulemused ja edasiarendus | 13 |
| 3.1. Juhised roboti loomiseks ja kasutamiseks..... | 13 |
| 3.2. Roboti kasulikkus õppetöös ja kasutusvõimalused ainetundides..... | 24 |
| 3.3. Roboti edasiarendamise võimalused | 24 |
| Kokkuvõte..... | 27 |
| Kasutatud kirjandus | 28 |
| Summary | 30 |

SISSEJUHATUS

Olles olnud ligemale kolm aastat arvutiõpetuse ja robotika õpetaja, olen näinud, millised probleemid on robotika õpetamisel koolides. Nähes, kui väike on kasutusel olev robotika riistvara, eriti just põhikooli tasemel, siis sellest ajendatuna sai lõputöö teemaks valitud nutiseadme abil juhitava roboti loomine, mis oleks võimalikult lihtne, kuid samas annaks ülevaate robotika võimalustest ja programmeerimise alustest. Roboti eesmärk oleks pakkuda vaheldust hetkel Eesti koolides laialt levinud LEGO robotitele. Loodav robot peaks olema alusplatvorm edasiarendusteks ja uuteks projektideks, mida on võimalike ehitada Raspberry Pi abil. Riistvaraline lahendus sai tehtud Eestis kättesaadavate seadmete baasil, hoides sellejuures kulud võimalikult madalad.

Käesoleva töö eesmärgiks on roboti ehitamise ja selle programmeerimise juhiste loomine gümnaasiumi õppekavas kasutusele võtmiseks, vabaks kasutamiseks huviringides ning kodus iseseisvalt ehitamiseks. Selle eesmärgi täitmiseks loodi ehitus- ja kasutusjuhised õpetajatele ja noortele seadme loomiseks ja kasutamiseks. Lisaks on vaja luua teoreetilise taust robotite kasutamise kohta ainetundides ja kirjeldada kasutatavat riistvara ja vajaminevaid teadmisi. Täiendavalt analüüsitakse roboti kasutusvõimalusi ainetundides ja tuuakse välja roboti edasiarendamise võimalused.

1. ROBOTITE KASUTAMISE VÕIMALUSED KOOLIDES

Robotite kasutamine üldhariduskoolides on suurendanud huvi reaalainete vastu ning uuringud on näidanud, et õpilane, kes tunneb juba põhikoolis suuremat huvi infotehnoloogia ja arvutitehnika vastu, valib suurema tõenäosusega selle valdkonna ka edasiõppimisel ja karjääri valikul [1]. Seetõttu on oluline õpilaste kire suurendamine arvutitehnika, infotehnoloogia ja robotika vastu.

Õpilaste oskuste arendamiseks on loodud programm ProgeTiiger, mille eesmärgiks on õppijate tehnoloogilise kirjasokuse ja digipädevuse tõstmine aastaks 2020 [2].

1.1. Programmeerimiskeeled

Eesti koolides on lisaks infotehnoloogia tundidele ka programmeerimise tunnid, kus õpetatakse õpilastele erinevaid programmeerimiskeeli, milles enda projekte ja programme kirjutada. Üldiselt on selliste keelte vahel väga keeruline orienteeruda, igapähe neist on omad positiivsed ja negatiivsed küljed. Gümnaasiumides õpetatakse peamiselt neist nelja suurimat: Python, Java, C ja HTML. Järgnevalt tutvustatakse iga eelpool mainitud programmeerimiskeelt lähemalt ning lisaks on toodud iga keele kohta näidiskood, mis väljastab teksti „Tere, maailm!“.[3]

HTML

HTML ehk HyperText Markup Language, eesti keeles hüperteksti märgistuskeel mida kasutatakse peamiselt veebilehtede loomiseks [4]. Antud keel loodi 1990. aastal Tim Berners-Lee poolt, kelle 1994. aastal loodud W3C tegeleb HTML'i standardite väljatöötamisega ja õpetuste kirjapanemisega [5]. Hetkel on viimasena avaldatud HTML5, mis täiendab eelnevaid HTML'e võimalusega koguda nutiseadmete anduritelt näite. Keel ise on väga lihtne, kus noolsulgude vahele kirjutatakse HTML-märgendid. Iga märgend peab omama algust ja lõppu. Näiteks <p> alustab lõiku, mille järele kirjutatakse sisu ning lõigu lõpetab </p>, ehk siis HTML-märgendi vastandit algsest märgendist eristab kaldkriips. Sellises keeles on õpilastel lihtne kirjutada enda oma veebilehtesid ja neid hallata. Selline keel on väga sobilik juba algklassidest alates, sest tulemus on õpilastele kohe nähtav ja vigu on antud keele korral lihtne leida, sest on näha kui palju koodi töötab ja kust kohast kood katki läheb. Antud keele kirjutamiseks pole vaja ka mingit täiendavat tarkvara alla laadida, sest koodi saab kirjutada ka kõige lihtsama teksti redigeerimise programmiga, kus tuleb salvestamisel lihtsalt lisada

faililaiend .html. Järgnevalt toome ära lihtsa koodi "Tere, maailm!" näite HTML keeles.

```
1. <html>
2. <title>Tervitus</title>
3. <body>
4. <p>Tere, maailm!</p>
5. </body>
6. </html>
```

C programmeerimiskeel

C programmeerimise keel loodi 1972. aastal Dennis Ritchie poolt. See keel töötati välja eesmärgiga programmeerida operatsioonisüsteemi UNIX. Tänapäeval saab seda keelt kasutada peaaegu igas operatsioonisüsteemis. Sellega luuakse peamiselt süsteemi- ja rakendustarkvarasid ning see on üks populaarsemaid programmeerimiskeeli üldse. Antud keelest on välja kujunenud suur hulk uusi keeli: C++, C#, JAVA, Python, PHP jne. Antud keel on aga algajate jaoks üpris keeruline, sest selle viitade, mäluhalduse ja stringide töötlemise omandamine on keerukam kui teistes keeltes. [6]

```
1. #include <stdio.h>
2. int main(void)
3. {
4.     printf("Tere, maailm!\n");
5.     return 0;
6. }
```

Java

Java programmeerimise keel, mis töötati välja 1995. aastal James Goslingu, Mike Sheridan ja Patrick Naughtoni eestvedamisel, sai enda nime Jaava kohvi järgi, mida loojad suurtes kogustes olid arendustööd tehes tarbinud [7]. Esialgelt plaaniti Java keelt kasutada interaktiivse televisiooni tarbeks, kuid see idee oli veel liiga oma ajast ees [8]. Antud keele omapära on see, et see töötab sõltumata platvormist hoopis Java virtuaalmasinal. Java lähtekood kompileeritakse baitkoodi, mida oskab käivitada iga virtuaalmasin. Kõik seadmed millel on tahtmist Java koodi kasutada peavad sisaldama Java käivituskeskkonda. Üks suurimaid eeliseid antud keele kasutamisel on see, et seda kasutab väga suur hulk arendajaid ja sellest tulenevalt on kogu keel väga hästi dokumenteeritud. Java on kõige populaarsem arendusplatvorm maailmas ning ka kõige populaarsem programmeerimiskeel maailmas. Enamus mobiiltelefonid (3 miljardit) jooksutavad Javat. [9] Sellest on välja kujunenud veel terve hulk keeli, mis kasutavad Java virtuaalmasinat. Antud keelel on hulk plusse ja miinuseid. Lisaks eelmainitud suurele kasutajaskonnale, heale dokumentatsioonile ja

sõltumatuses platvormist on sellele automaatsed mäluhaldus ja kirjutatud koodi elektrooniline dokumenteerimine ning selle arendamiseks on välja töötatud väga mugav arendustarkvara.

```
1. public class HelloWorld {
2.     public static void main(String args[]) {
3.         System.out.println("Tere, maailm!");
4.     }
5. }
```

Python

Pythoni programmeerimiskeele eelis on see, et see on üldotstarbeline interpreteeritav programmeerimiskeel, mis on funktsionaalne, imperatiivne, reflektiivne ja objektorienteeritud. Keel nägi ilmavalgust 1991. aastal tänu Guido van Rossumile. Üldiselt peetakse Pythonit üheks lihtsamini õpitavaks programmeerimiskeeleks ja sellest tulenevalt õpetatakse seda ka tihti koolide õpilastele. Lisaks sellele on see kergelt kättesaadav vabavara, mis on enamustes UNIX masinates juba vaikimisi installeeritud. Üks eeliseid selle keele juures on see, et pole vajadust määratleda muutujate tüüpe, ehk see on dünaamiliste andmetüüpidega keel. Antud keeles saab väga kiiresti programme arendada ja just selle tõttu kasutatakse seda tihti prototüüpimisel, mille järel kood mõnda kõrgemas keelde ümber kirjutatakse. [10]

```
1. print "Tere, maailm!" #2.x
1. print ("Tere, maailma!") #3.x
```

1.2. Koolides kasutatavad robotid ja arendusplaadid

Eestil on väga hea maine just infotehnoloogia valdkonnas ja selle kinnituseks on meie koolides kasutusel erinevaid lisaseadmeid just programmeerimise õpetamiseks. Maailmas on loodud selliseid seadmeid väga palju ja nende ülesandeks on võimaldada õpilastel rakendada programmeerimise tunnis omandatud, et juhtida näiteks mootoreid või lampe. Populaarsemad neist Eesti koolides on näiteks LEGO Mindstorms NXT ja EV3 seeriad, mille klotsidest ehitatavaid roboteid võib leida kasutamas noori alates põhikoolist kuni ülikoolini välja. Lisaks leidub ka erinevaid arendusplaate, millega saab koostada erinevaid skeeme ja projekte, näiteks Arduino ja Raspberry.

LEGO Mindstorms

LEGO Mindstorms seeria robotid on eesti koolides kõige populaarsemad robotid. Need on väga head õppevahendid, sest kuna tegu on LEGO komplektiga, siis sellest saab ehitada praktiliselt kõike. Just see on kõige suurem pluss, et kui isegi jääb umbes 550 osalisest

komplektist juppe väheseks, siis neid saab komplektidena juurde osta, või kaasõpilaste komplektidest juurde laenata. Antud töös keskendun peamiselt uuemale mudelile EV3, mille eelkäijaks oli NXT robot.

EV3 komplekt on varustatud suure hulga erinevate andurite (güroskoop-, värvi-, ultraheli-, puute- ja infrapunaandur) ja 3 mootoriga, kuid kõige olulisem on kogu robotit juhtiv programmeeritav juhtplokk. Sellel on 6 nuppu, ekraan, 4 sisend- ja 4 väljundkaablite pesa. Seadmel on sisseehitatud sinihamba ühendus, kõlar, 2 USB-pesa ja SD-kaardi pesa. [11]

Roboti programmeerimiseks on mitmeid võimalusi. Kõige lihtsam on seda teha otse juhtpaneeli peal kasutades LEGO poolt sinna salvestatud loogikaplokke. Kuna selline programmeerimine on piiratud aga 16 plokiga, siis pole see just kõige mõistlikum lahendus. Tarkvaraliseks programmeerimiseks on seadmel olemas vabavarana National Instruments LabVIEW baasil loodud graafiline programmeerimiskeskond „LEGO „Mindstorms“ EV3 Home Edition“. Selles on programmeerimine väga lihtne. Õpilane lohistab omavahel kokku juhtkastid, annab neile väärtused ja selline kastide kogu moodustabki programmi. Kuna tegu on avatud lähtekoodiga tarkvaraga, siis on juba välja töötatud ka võimalused, kuidas seadet juhtida näiteks Java keelt kasutades. Seda tänu sellele, et EV3 kasutab protsessorina ARM9 kiipi, mis on laialt levinud muude nutiseadmete sees ja sellest tulenevalt toetab seda ka Java virtuaalmasin. Samuti saab seda programmeerida ka näiteks varem mainitud C keeles. [12] Üldiselt kasutatakse selle programmeerimiseks koolides ikkagi graafilist programmeerimist.

Sensorite ja andurite suhtes on aga LEGO Mindstorms seeria üpris piiratud, sest kui vajalikku andurit LEGO poolt pole välja töötatud, siis seda ise ehitada on kooliõpilasel liiga keeruline, sest see on nõuab kindlaid ühendusporte ja koodi kirjutamise uute andurite tööle saamiseks. Lisaks eelnevale on antud komplekt 350€ hinnaga ka üpris kallis.

Arduino

Arduino on avatud lähtekoodiga ühe plaadi mikrokontroller. Selle lõi 2005. aastal Massimo Banzi ja David Cuartielles ja selle nimi tähendab itaalia keeles "tugevat sõpra". Meeste eesmärgiks oli luua seade, millega õpilased saaksid luua odavalt ja lihtsalt elektroonikaprojekte. Sellel on erinevaid versioone, mille suurused on alates 43mm x 18mm kuni 101,6mm x 53,3mm. [13] Kontrolleril olevate viikude arv ulatub 14 kuni 54ni ning need on varustatud Atmeli AVR protsessoritega. Ja siin ongi nende üks nõrkusi, sest need on üpris aeglased 32MHz ja 8-bitis loogikaga. [14] Vanemaid mudeleid tuli programmeerida läbi RS-

232 järjestikliides programmeatori. Uuemaid verisoone saab programmeerida ka juba läbi USB-liidese või sinihamba ühenduse.

Sensorite ja andurite lisamiseks on plaadil välja toodud digitaalsed sisend-väljundviigud, millest osad võimaldavad kasutada näiteks pulsilaiuse modulatsiooni. Samuti on olemas ka analoog-sisendviigud. Nende külge saab otse ühendada erinevaid andureid ja siis nendel olekuid lugeda ja välja anda signaale, millega näiteks mootoreid juhtida või LEDe põlema panna. Lisaks on olemas lisamoodulid, mis käivad viikude otsa. Müügil on näiteks 37 sensorist/andurist koosnevaid moodulite komplekte, kuhu kuuluvad näiteks moodulid LEDidega, releega, lülitiga, kõlariga, valgusanduri, soojusanduri, niiskusanduri, katseskeemi alusega jne. Antud komplekti hind on suurusjärgus 50€ [15] ja kui sellele lisada ka Arduino enda hind, siis kokku tuleb see umbes 65€.

Arduino programmeerimiskeskond on kirjutatud Javas ja see on tuletatud *Processing* programmeerimiskeelest. Antud programmeerimiskeskonnaga tuleb kaasa ka C/C++ teek nimega "Wiring", mis teeb sisend-väljundoperatsioonid väga palju lihtsamaks. See on ehitatud nii, et programmeerimisega saaksid hakkama ka vähekokogenud kasutajad. Kasutaja töö lihtsustamiseks on süntaks esile toodud ja sulgude vahed on eraldi märgistatud, et oleks aru saada, kus mingi funktsioon algab ja kus lõpeb. Programmeerimine toimubki veidi modifitseeritud C++ keeles.

Raspberry Pi

Raspberry Pi pole enam lihtsalt plaat millel on mikrokontroller, vaid tehniliselt on tegu miniatuurse arvutiga. Seade töötati välja Eben Uptoni eestvedamisel koos õpetajate, akadeemikute ja arvutientusiastidega, eesmärgiga luua arvuti, mida õpilased saaksid odavalt osta ja kasutada projektide loomiseks.

Antud töös keskendun peamiselt kirjutamise hetkel viimasena turule tulnud Raspberry Pi 2 mudelile B. Seadme võimsuse taga on neljatuumaline ARM Coretex-A7 mille normaalne töökiirus on 900MHz, aga seda võib ka ülekiirendada, kuid siis on mõistlik kivi eraldi jahutada radiaatoriga. Lisaks on seadmél muutmälu 1GB, mis on küll jagatud protsessori ja graafika (VideoCoreIV) vahel, kuid see on piisav, et seadmél jooksutada erinevaid Linuxi operatsioonisüsteeme ning ka uut Windows 10 operatsioonisüsteemi. Pi 2 on varustatud 4 USB-pesa, HDMI-pesa, kõrvaklapi pesa, RJ45 internetikaabli pesa ja mini-USB pesaga, millest viimane töötab seadme vooluallika pesana. Arvutil pole kõvaketast kui sellist, aga

andmeid ja operatsioonisüsteemi saab hoida micro-SD kaardi peal. Lisaks on seadmel ekraani DSI (Display Serial Interface) pesa ja kaamera CSI (Camera Serial Interface) pesa, kuhu saab siis vastavalt ühendada LCD ekraani või kaamera. [16]

Välise seadmete ja sensorite jaoks on lisaks USB-pesadele ka 40 viiku. USB-pesadesse saab ühendada lisaks hiirele ja klaviatuurile ka WiFi ja sinihamba mooduleid ning veebikaamera. 40 viigu seas on kaks 5V ja kaks 3,3V ühendust ja need annavad välja muretult 50mA [17], mis on piisav, et panna põlema LEDe ja liigutada väikeseid servo mootoreid. Lisaks on seadmel 8 maanduse ühendust ja 26 sisend-väljundviiku, mille väljund on maksimaalselt 3,3V. Lisaks on veel kaks viiku, mis on mõeldud elektriliselt eemaldatava programmeeritava lugemiskaitstud mälu ühendamiseks, kuid need on tegelikult lihtsalt viigud 00 ja 01.

Selle väikese arvuti hind on suurusjärgus 40€[17], mis teeb sellest väga odava ja ülimalt multifunktsionaalse seadme, millega on võimalik valmistada praktiliselt kõik projekte, mis pähe tuleb. Alates LEDi vilgutajast kuni sadadest Raspberrydest koosnevast superarvutini. Loomulikult ei tähenda see seda, et tegu oleks maailma kõige parema graafikaga arvutiga, vaid pigem saaks nendest luua väga suure arvutusvõimsusega arvuti, millega teadust teha ja seda üpris madal hinna eest, kusjuures enda madala energia tarbimise poolest (tavalise kasutuse juures umbes 4W) võiks oleks tegu palju säästlikuma arvutiga, kui samaväärne server-lauaarvuti (umbes 100W).

Raspberry Pi'd saab programmeerida praktiliselt kõikides programmeermiskeeltes. See teeb sellest väga hea õppetöös kasutatava seadme. Seda saaksid õpilased mugavalt ka koju kaasa võtta ja selle peal näiteks ka kodutöösid teha, erinevalt tavalisest lauarvutist, sest Pi on oma mõõtmetelt väga väike, kõigest 85mm x 56mm [18].

2. ROBOTI LOOMINE

Käesoleva töö eesmärk on luua nutitelefoniga ühendatav robot, mille hind oleks umbes 85€, olles sellega umbes neli korda odavam kui LEGO Mindstorms EV3 robot. Lisaks peaks see olema kõigest alusplatvorm, millest saaks edasi arendada veel keerulisemaid roboteid ja projekte. Sellest tulenevalt oli selge, et programmeerimise kohapealt oleks mõistlik keskenduda just Raspberry Pi'le, sest see on mitmekülgne ning selle hind on üpris odav võrreldes teiste sarnaste süsteemidega.

Olles ise olnud ligemale 3 aastat arvutiõpetuse, robotika ja programmeerimise õpetaja Puhja Gümnaasiumis, siis olin näinud ka mis tugevad ja nõrgad kohad esinesid robotika ja programmeerimise ainetes. Programmeerimine oli tihti õpilaste jaoks liiga teoreetiline. Rida rea järel kohati arusaamatuks jäävat teksti, mille tulemusena kuvatakse ekraanile mingeid tekste ja arve. See pole üldse kaasahaarav just eriti alg- ja põhikooli jaoks. Robotika tund seevastu oli paljude lemmik, kus sai ehitada seadmeid, mis olid kiiremad ja osavamad kui teiste omad. Samas oma kõrge hinna tõttu oli robotite arv piiratud neljaga ja rühmadena tööd tehes ei saanud kõik enda oskusi ja ideid näidata. Samas oleks võinud programmeerimises õpitud Javat ära kasutada roboti seadistamiseks. Kahjuks eelmisel aastal veel sellist head võimalust koolidel polnud. Seega sai mulle selgeks, et õpilastele tegelikult meeldiks programmeerida, kui nad näeksid ka mingisugust reaalselt tulemust. Selleks sobiksid kõiksugused väljundid: tuled, heli, liikumine jne. Kahjuks sai kool koostöös Haridus Infotehnoloogia Sihtasutuse ja vallaga soetada vaid neli LEGO Mindstorms EV3 robotit, mida jäi aga terve klassi peale liiga väheseks. Kool sai endale lubada 1440 euro eest vaid neli LEGO Mindstorms EV3 robotit. Kui arvutiklassis on 16 kohta ja soovida tagada igale õpilasele ja õpetajale üks komplekt, siis tuleb ühe roboti hinnaks maksimaalselt 84 €. Koolil ei kasvaks kulud ning samal ajal saaksid kõik õpilased ise oma seadet ehitada ja programmeerida.

Esmalt oli vaja leida sobivaim riistavara, mille abil seade tööle panna. Roboti juures on loogika plaadina kasutuses Raspberry Pi 2 mudel B, sest see võimaldab oma madala hinna juures koostada väga keeruliste lahendustega projekte. Üldine kontseptsioon on järgmine – roboti kere moodustab plastmasskarp, mille sees on Raspberry Pi, akupank, WiFi moodul. Karbi külgedel on kaks servo mootorit, mis panevad keerlema 2 CD plaati, ehk siis robot on karp kahe rattaga ja seda saab juhtida nutitelefoni abil. Nutiseadet eraldi poleks suure tõenäosusega vaja osta, sest nagunii praktiliselt igal õpilasel on olemas oma nutitelefoni ja kui

isegi neil ei ole seda, siis saab seda laenata kaasõpilastelt testimise eesmärgil või siis juhtida seda läbi arvuti. Hetkel keskendusin siiski Androidi güroskoobiga juhitavale robotile, kuid edasiarendusena võib ka kuvada nuppe näiteks veebilehel, mille abil seadet liigutada. Antud töös pole see aga eesmärk.

Töö algne eesmärk oli teha Androidi rakendus, mis loeb telefoni güroskoobi näitu, mis omakorda loetakse faili, kust see omakorda loetakse veebiserverisse ja sealt Raspberrysse ja seal töödeldakse see ümber signaaliks, mis paneb servod vastavalt vajadusele liikuma.

Esimene etapp roboti ehitamisel on Raspberryga servo liigutamine. Mootoriteks said valitud väikesed 9g servo mootorid, sest need kasutavad piisavalt vähe voolu, seega saab nende jaoks voolu otse sisend-väljundviikudelt võtta. Sobivate nõuetega ja piisavalt odavad väikesed servod liiguvad ainult 180 kraadi ja kasutavad selle jaoks potentsiomeetrit, mis on muuttakisti, mille järgi servo enda positsiooni määrab. Esimene idee oli lihtsalt kasutada otse servo sees olevaid mootoreid. Kahjuks osutus see idee ebasobivaks, sest niimoodi saab küll mootori käima, kuid see töötab kas ühtepidi või seisab ilma signaalita. Selliselt probleemist ülesaamiseks oleks vaja olnud kasutada lisa kiipi L293D, mis vajaks omakorda rohkem ruumi ja lisa vooluallikat. Sellele lisaks on selle hind 4,5€ [19], mis moodustaks üle 5% kogu hinnast. Seega tuli algne plaan ümber mõelda. Teine lahendus oli ühendada lahti potentsiomeeter ja servo ühenduses kaablid, mis on ühenduses anduriga, ühendada omavahel takistitega. Kahjuks oli mootori sees nii vähe ruumi, et sinna oleks olnud praktiliselt võimatu täiendavaid asju mahutada. Seega sai mindud kõige lihtsamat teed ja fikseeritud potentsiomeeter kindlasse asendisse liimiga. See teeb küll programmeerimise keerulisemaks, kuid samas roboti ehituse märksa kiiremaks ja lihtsamaks.

Nüüd kui servo valmis, võis seda liikuma panna Raspberry plaadiga. Selleks tuli otsustada, millist programmeerimiskeelt kasutada. Otsustasin kasutada Pythonit, sest see on kergesti õpitav ja vaikimisi juba Raspberryle installeeritud koos NOOBS operatsioonisüsteemiga Raspbian. Järgnevalt on tabelis 5 toodud näidis lihtsast servo mootori juhtimise koodist. Täpsemalt saab kood lahti seletatud roboti programmeerimise peatükis.

```
1. import RPi.GPIO as GPIO
2. import time
3. GPIO.setmode(GPIO.BOARD)
4. GPIO.setup(11, GPIO.OUT)
5.
6. frequencyHertz = 50
7. pwm1 = GPIO.PWM(11, frequencyHertz)
8.
9. pwm1.start(1)
```

```
10. print ("Start")
11. time.sleep(1)
12. pwm1.stop()
13. print ("Stop")
14.
15. GPIO.cleanup()
```

Edasi sai alla laetud Android Studio ja SDK Tools. Esimene probleem oli see, et kuigi antud tarkvarad polnud liiga suured (vastavalt 0,9 GB ja 1GB), siis SDK Toolsi uuendamine võttis ligemale 12 tundi (5M interneti puhul). Sellest tulenevalt kulus ka väga suur hulk kõvaketta ruumi, umbes 21 GB ning seega oli selge, et see pole väga hea lahendus. Samas androidi seadmelt oli vaja güroskoobi andmeid lugeda ja selle jaoks oli see tarkvara vajalik. Güroskoobi andmete lugemise näidiskood on lihtsalt kättesaadav internetist. Järgmisena oli tarvis seada ülesse veebiserver, mis vahendaks kogutud koordinaate nutiseadmelt Raspberryile. Sellel hetkel sai konsulteeritud Tartu Ülikooli, Füüsika Instituudi, Laserspektroskoopia laborandi Taavi Repäniga, kes pakkus välja lahenduse, et güroskoobi andmeid ei pea lugema läbi rakenduse, vaid seda saab teha lihtsalt läbi HTML5. Lähemal uurimisel selgus, et HTML5 võimaldab tõesti lugeda nutiseadme andurite olekuid. Tänu sellele saab ära jätta väga aeganõudva ja suure mahuga androidi arengukeskkonna installeerimise ja kasutamise.

Tulenevalt eelmises peatükis väljatoodud ideele, tuli kogu seadme tarkvaraline osa ära muuta. Uus kontseptsioon näeb välja nii, et kasutades HTML5 koguda nutiseadme güroskoobi andmeid ja edastada need veebiserverisse, kust omakorda Raspberryile, kus töödelda andmed sobivateks, et juhtida nendega mootoreid.

9. mail 2015. a võtsin osa juhendaja Alo Peetsi korraldatud Tartu Ülikooli arvutiteaduse instituudi huvilaagrist, kus aitasin läbi viia töötuba "Miniarvuti Raspberry Pi". Nimetatud töötuba aitas välja selgitada õpilaste jaoks Raspberry kasutamise ja programmeerimisega tekkivad probleemkohad. Töötoas abistamise tulemusena leidsin uue lahenduse, et teha Pi peale veebiserver, mis kogub infot brauseri (HTML5) poolt kogutud andmetest. Ehk kokkuvõttes tähendab see, et kogu vajalik programmeerimine toimub läbi Raspberry Pi enda. Tehniliselt peaks see süsteem töötama seega iga seadmega, millelt HTML5 suudab andurite olekuid mõõta, ehk siis miinimum nõueteks on androidi puhul versioon 4.0 ehk Ice Cream Sandwich ning Apple toodete puhul piisab iPhone 3Gst, millel on iOS 4.2 või uuem.

Sellega sai selgeks lõplik riistavaraline ja tarkvaraline seadistus ja kuidas asi omavahel töötama peaks. Lõplik riistvara ja ehitus on detailsemalt kirjeldatud juba järgmises peatükis.

3. TULEMUSED JA EDASIARENDUS

3.1. Juhised roboti loomiseks ja kasutamiseks

Roboti loomiseks vajalikud seadmed

Roboti loomiseks vajalikud tarvikud, tööriistad ja õpilaste töövahendid on toodud tabelites 1 ja 2.

Tabel 1. Roboti loomiseks vajalikud roboti osad.

| Nr | Vajaminevad tarvikud | Hind (eurodes) | Hinnapakkumise allikas |
|--------------------|--|----------------|---|
| ROBOTI OSAD | | | |
| 1. | Raspberry Pi 2 mudel B | ~41,00 | http://www.ittgroup.ee/en/raspberry-pi/542-raspberry-pi-2-model-b.html |
| 2. | Akupank JSP PB-100 | 8,34 | http://www.ordi.ee/EPood/Product.aspx?MC=TARVIKUD&IC=6500&ItemID=6500-1953 |
| 3. | 2 x Micro servo mootor 3.5-6V 1.2kg/cm(4.8V) 1.6kg/cm(6V) | 13,12 | http://www.oomipood.ee/product/sg90/micro-servo-mootor-3-5-6v-1-2kg-cm-4-8v-1-6kg-cm-6v&ac=1 |
| 4. | Karp, 55*95*160mm | 4,75 | http://www.oomipood.ee/product/wcah2851/plastkarp-55-95-160mm-wcah2851 |
| 5. | MicroSD kaart | 8,50 | http://www.ordi.ee/EPood/Product.aspx?MC=TARVIKUD&IC=4500&ItemID=4500-1519 |
| 7. | WiFi moodul (Eestist ostes on selle hind väga kõrge ja kuna kõik moodulid Raspberry't ei toeta, siis sai see tellitud välismaalt) | ~5,00 | http://www.ebay.com/itm/Pro-Mini-USB-2-0-Wireless-Wifi-802-11n-USB-LAN-Adapter-Dongle-for-Raspberry-Pi-/251691080838?pt=LH_DefaultDomain_3&hash=item3a99f51c86 |
| 8. | 4 x CD plaat | 0,80 | http://www.la.ee/burootehnika/toorikud/acme_cdr_52x_700mb_10p_shrink |
| 9. | 10 x kaabel (emane-emane) | ~1,00 | http://www.oomipood.ee/product/farnell_2396146/jumper-wire-assortment-65pcs-mcbbj65 |
| | KOKKU | ~83,51 | |

Allikas: autori koostatud.

Tabel 2. Roboti loomiseks vajalikud tööriistad ja õpilase töövahendid

| Nr | Vajaminevad tarvikud |
|----------------------------|--|
| TÖÖRIISTAD | |
| 1. | Akutrell koos puuridega 4mm ja 12mm |
| 2. | Kruikeeraja PH1, PH0 ja PH00 otsaga |
| 3. | Kahepoolne teip |
| 4. | Kivistuv liim |
| 5. | Vineeri saag või muu väike saag, mille tera annab keerata erinevate nurkade alla |
| 6. | Näpitsad |
| ÕPILASE TÖÖVAHENDID | |
| 7. | Arvuti |
| 8. | Klaviatuur ja hiir |
| 9. | Nutitelefon (android 4.0+ või iOS 4.2+) |
| 10. | Nutitelefoni laadija |
| 11. | HDMI kaabel vastavalt vajaminevale videoväljundile, mis on olemas monitoridel |
| MUUD VAHENDID | |
| 12. | WiFi ruuter |
| 13. | Mälukaardi adapter |

Allikas: autori koostatud.

MicroSD kaardi ettevalmistamine ja tarkvara installeerimine

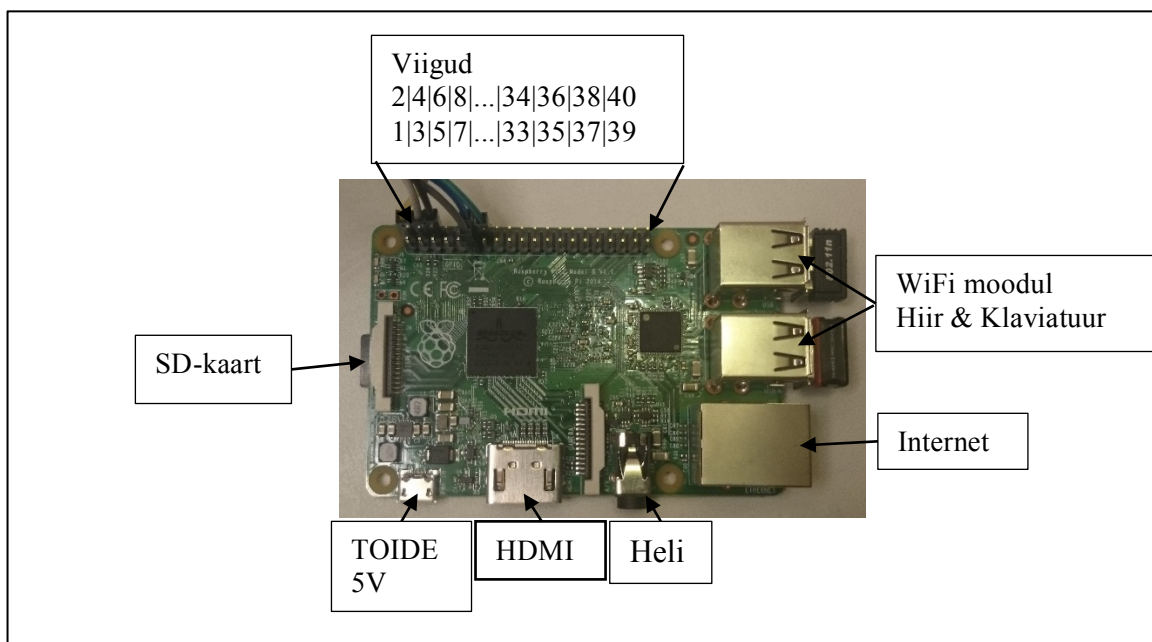
Esmalt on vaja alla laadida tarkvara SD Formatter 4.0 (https://www.sdcard.org/downloads/formatter_4/), millega tuleb mälukaart vormindada. Antud tarkvara kasutamine on selle pärast oluline, et see toetab kõiki SD/SDHC/SDXC mälukaarte ja see vormindab mälukaarte parimas võimalikus seadistuses, erinevalt tavaliste operatsioonisüsteemidega kaasasolevatele vorminduse programmidele. Alla laadimiseks tuleb programmi veebilehel valida sobiv tarkvara versioon ning järgneval lehel nõustuda tingimustega. Allalaetud kaust tuleb lahti pakkida ja seadistus failist installeerima panna. Antud programmiga tuleb mälukaart vormindada. Kõik seaded jäävad vaike seadeteks. Ettevaatlik tuleb olla õige seadme valimisel. Kontrollige, et valitud oleks ikka õige mälukaart (see kaart, mis läheb Raspberry sisse), sest antud programm kustutab kogu info, mis andmekandjal on.

Järgmisena on vaja alla laadida Raspberry operatsioonisüsteem Raspbian, millest meie kasutame NOOBS paketti (<https://www.raspberrypi.org/downloads/>). Kuna tegu on ~800MB failiga, siis on mõistlik see varakult laadima panna või eelnevalt paigaldada arvutitesse või

mälupulkadele, kust seda saaks kiiremini suurtes kogustes kopeerida. Antud kaust tuleb taaskord lahti pakkida ja selle sisu kopeerida mälukaardile.

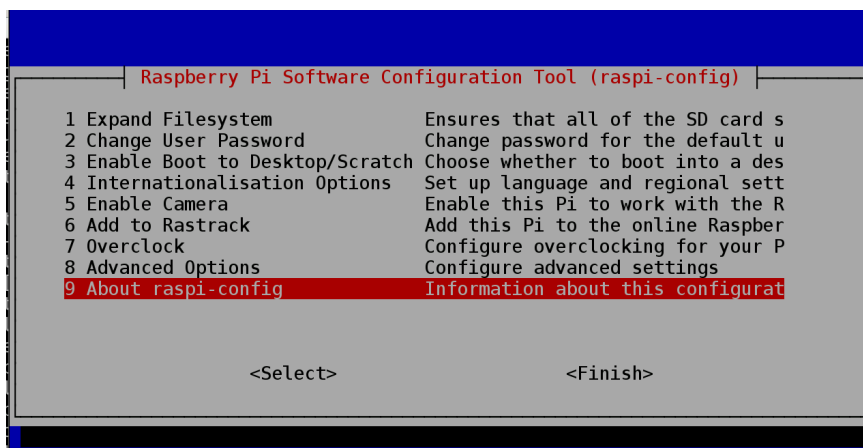
Raspberry ülesse seadmine

Kui mälukaart on valmis, siis see tuleb panna Raspberry tagaküljel olevasse pessa. Lisaks on vaja ühendada klaviatuur, hiir ja WiFi moodul ning HDMI kaabel, nagu näidatud joonisel 1. Kui kõik on õigesti ühendatud, siis võib Raspberry käima panna. Mõistlik oleks alguses seda teha kasutades nutitelefoni adapterit.



Joonis 1. Raspberry Pi sisendid ja väljundid (autori koostatud).

Raspberry laeb ennast ülesse. Avanenud menüüs tuleb märgistada Raspbian ja Data Partition ning seejärel vajutada "Install". Kui tarkvara on installeeritud, siis tuleb teade "OS Installed" vajutada "OK", mille järel teeb seade taaskäivituse ja avaneb seadete menüü, mida näeb joonisel 2. Siin oleks mõistlik muuta seadme parooli, selleks tuleb nooltega liikuda teisele reale „Change User Password“ ja vajutada Enter klahvi. Vaikimisi on kasutajanimi "pi" ja parool "raspberry".



Joonis 2. Raspberry Pi seadistuste menüü (autori koostatud).

Kolmandal seadete realt tuleb valida „Enable Boot to Desktop“ ehk seadme käivitamine graafilisel kujul koos kasutajaliidesega. Lisaks tuleb ära muuta klaviatuuri asetus Eesti (EE) omaks, seda saab teha *Internationalisation Options*’ite alt. Kui seaded paigas, siis kaks korda paremale noolt, niimoodi olete jõudnud "Finish" nupule ja Enter. Seade taaskäivitab end taas. Kui midagi jäi alguses tegemata, siis seadetes saab tagasi läbi terminali käsuga "sudo raspi-config". Kui Raspberry on tööle saanud, siis tuleb seade ühendada WiFi võrku. Selleks tuleb minna "Menu" ->"Preferences" -> "WiFi Configuration". Valida tuleb see võrk kuhu saab hiljem ühendatud ka nutiseadme ja ühendada ennast sellesse. Probleem on lihtsalt selles, et kuna igal ühel ei ole olemas oma veebiserverit, mis hoiaks staatilist ip-aadressi, siis lihtsuse mõttes on kasulikum kasutada just ühte võrku, sest nii saab nutiseadme veebibrauserist avada sisevõrgus Raspberry peal jooksvat veebilehte avada. Kui interneti ühendus on saavutatud, siis tuleb süsteemi paketid uuendada käsuga "sudo apt-get update". Järgmisena soovitan laadida graafilise tekstiredigeerimisprogrammi gedit. Selleks tuleb sisestada käsk "sudo apt-get install gedit", kui seade küsib nõusolekut, siis tuleb sinna sisestada "Y" ja vajutada sisestusklahvi.

Servo ettevalmistamine

Servo pealt tuleb kleepsud eemaldada ja kruvid lahti keerata. Kui seade on lahti, siis tuleb eemaldada hammasrataste poolne kate. Oluline on meelde jätta hammasrataste järjekord, et seda oleks hiljem võimalik tagasi kokku panna, selleks võib algseisust teha näiteks pildi nutiseadmega. Peale hammasrataste eemaldamist tuleb kõige suuremal (kõige pealmine) ära lõigata piiraja, see on väikene pulk alumisel küljel. Seejärel tuleb sama hammasratta auku, mis on alumisel küljel, puurida või viilida seest ümmarguseks. Kõige lihtsaim moodus on

seada lihtsalt selle kinnituspulga otsas edasi tagasi keerutada, kuni see vabalt ringi käib. Ja viimasena on vaja potentsiomeeter lukustada, selleks tuleb see võimalikult keskele keerata ja siis mõlemale poole metallplaadi ja plastiku vahele lükata näiteks poolik tikk ja siis peale lasta tugevat liimi nagu näiteks Super Attack, mis peab sinna vahele ära kivistuma. Tuleb olla ettevaatlik, et enne kui hammasrattaid külge tagasi panna, siis oleks ikka liim ära kuivanud (salvrätikuga katsudes), sest muidu ei sa mootorit enam kasutada. Niimoodi on servo modifitseeritud lõputult pöörlema. Seejärel tuleb üleliigsed tikku osad ära lõigata ja mootor tagasi kokku panna. Enne selle mootori ühendamist Raspberry külge võiks proovida seda käsitsi keerutada, kui see niimoodi keerleb, siis on mootor valmis.

Karbi ettevalmistamine ja rataste lisamine

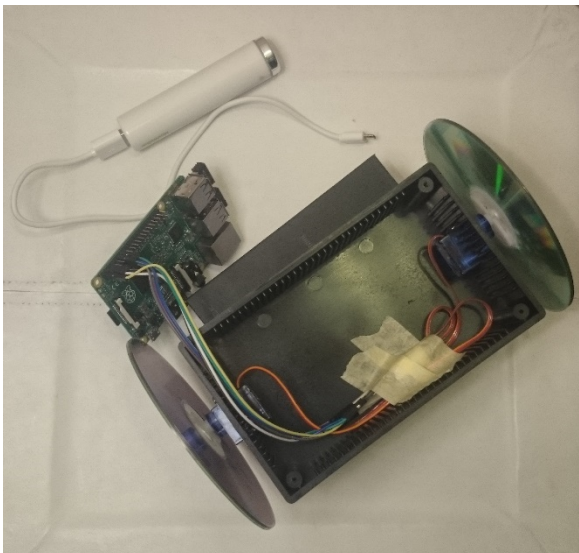
Karbi väikseima pindalaga külgedele tuleb joonistada servo mootori suurused kastid ning need välja lõigata. Selleks on mõistlik esmalt puurida 4mm puuriga sinna kaks auku nii, et kui 12mm puuriga puurima hakata ei väljuks see soovitud alast. Kui suured augud olemas, siis saab saega saagida kaks auku kokku üheks ilusaks ja sobiliku suurusega kastiks, nagu näidatud joonisel 3. Orienteeruvalt tuleb arvestada, et servo mootori väline hammasrattas võiks jääda umbes külje keskele ja kõrguseks peab olema piisavalt maad, et karp ei hakkaks vastu maad lohisema või et rattad ikka ulatuksid maani, samas tuleb arvestada sellega, et karbi raskuse hoiab ainsana ära selle, et karp seisaks ühes asendis ja rattad liiguksid, mitte vastupidi.



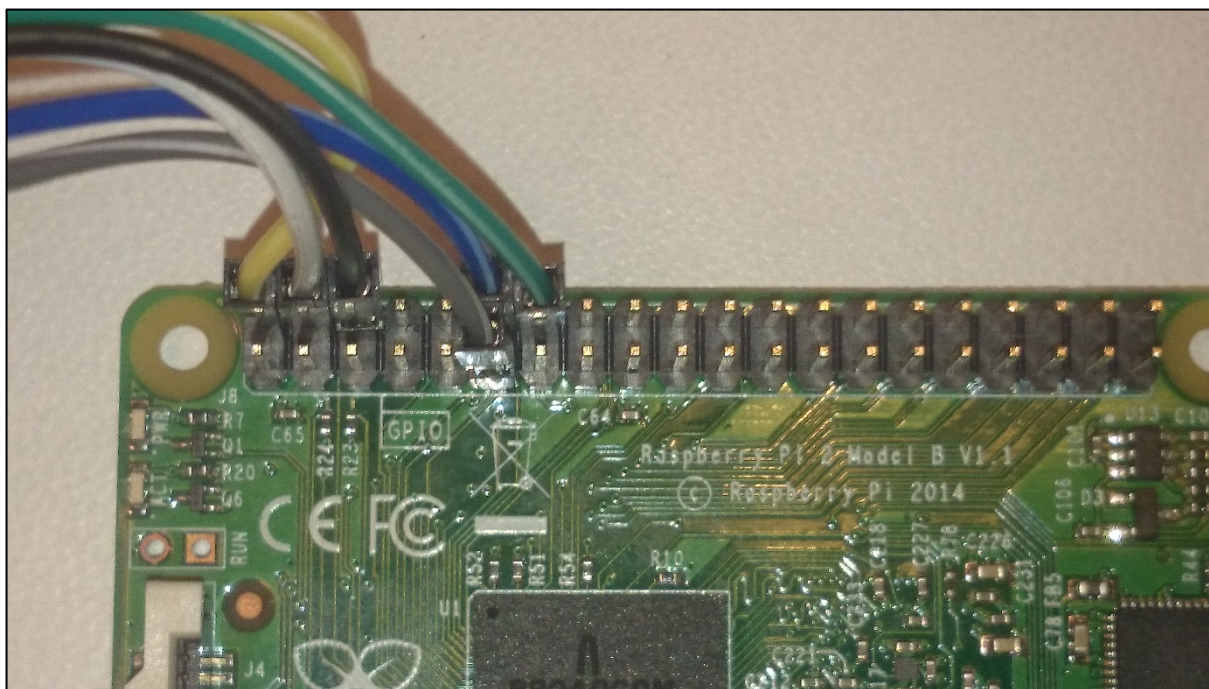
Joonis 3. Karbi sisse lõigatav auk (autori koostatud).

Servo mootorid võib nüüd kinnitada karbi külge. Selleks tuleb kasutada kaasas olevaid kruve, sest niimoodi saab mootoreid ka ümber tõsta, kui on tahtmist ehitada hiljem muid seadmeid. Rataste ehitamiseks tuleb kaks CD-toorikut omavahel kokku liimida ja omakorda ühele küljele liimida või kaasasolevate kruvidega kinnitada ristikujuuline üleminek, mis on servo mootoriga kaasas. See peab asuma täpselt CD-plaadi keskel, sest muidu hakkab robot sõitmisel ennast üles-alla viskama. Rattad saab mootori külge kinnitada kaasas oleva kruviga. Sellega hetkel roboti ehitus piirdub, sest kuni programmeerimise lõpuni ei ole mõistlik Raspberryt mõistlik lisada, kuna siis on väga ebamugav ekraani ja klaviatuuri sellele külge ühendada, selline seis on kujutatud joonisel 4. Servo mootorite tööle panemiseks kasutame viike vastavalt joonisele 5:

- 5V viikudelt (#2 ja #4) saame servo mootori töö voolu (punane kaabel)
- Maanduse viikudelt (#6 ja #14) saame servo mootori maanduse (pruun kaabel)
- Sisend-väljundviikudelt (#11 ja #12) saame servo käivitamise signaali (kollane kaabel)



Joonis 4. Karp koos mootorite ja ratastega (autori koostatud).



Joonis 5. Viikude ühendamine (kollane kaabel #2, valge kaabel #4 jne) (autori koostatud).

Roboti programmeerimine

Roboti programmeerimiseks avage gedit või endale meeldiv teksti redigeerimisprogramm. Selleks kirjutage terminali „gedit index.html“ või „nano index.html“.

Esmalt on vaja luua HTML5 veebileht, mis oskab lugeda nutiseadme güroskoobi andmeid ja neid koguda. Et kasutaja näeks ka midagi enda seadme ekraanil, siis sinna on lisatud x ja y koordinaatide näidud.

```
1. <html>
2. <head>
3.
4. <script>
5. //Kui seadmelt ei saa andmeid lugeda, väljastab veateate
6. if (window.DeviceMotionEvent == undefined) {
7.     alert("No gyro");
8. }
9. //Kui seadmelt saab andmeid lugeda, siis hakatakse sellest andmeid lugema
10. else {
11.     window.addEventListener("devicemotion", accelerometerUpdates, true);
12. }
13.
14. //Looje muutuja ja funktsiooni näitude lugemiseks
15. var lastUpdate = 0;
16. function accelerometerUpdates(e) {
17.     //Sekunditeks teisendamine
18.     var seconds = new Date().getTime() / 1000;
19.
20.     //Looje muutujad, millele hakkame vastava koordinaadi väärtusi määrama
21.     var aX = e.accelerationIncludingGravity.x;
22.     var aY = e.accelerationIncludingGravity.y;
23.
24.     //Näitude küsimine ja muutujate määramine
```

```

25.     document.getElementById("x").innerHTML = aX;
26.     document.getElementById("y").innerHTML = aY;
27.
28.     //1 korra sekundis näidu uuendamine
29.     var RATE = 1;
30.     if (lastUpdate + 1.0 / RATE >= seconds)
31.         return;
32.
33.     // Näitude edastamine
34.     var xhr = new XMLHttpRequest();
35.     xhr.open("GET", "/update_servo?x=" + aX + "&y=" + aY, true);
36.     xhr.send();
37.     lastUpdate = seconds;
38. }
39.
40. </script>
41.
42. </head>
43.
44. <body>
45.     <!-- Kuvatakse veebilehel roboti nimi (ROBI) ning x ja y koordinaadid -->
46.     ROBI x axis <b id="x"></b><br>
47.     ROBI y axis <b id="y"></b><br>
48. </body>
49. </html>

```

Kui veebileht loodud, siis on vaja veel ülesse seadistada Raspberry peal veebiserver ja kogutud näidud teisendada sobivale kujule ja need mootorile signaalidena ette sööta. Selleks tuleb luua uus fail nimega ServoServer.py.

```

1. import RPi.GPIO as GPIO #Viikude paketi lisamine
2. from http.server import * #HTTP seadete paketi lisamine
3. import re #Lihtsustatud väljendite lisamine
4.
5. GPIO.setwarnings(False) #Hoiatuste väljalülitamine
6. GPIO.setmode(GPIO.BOARD) #Viikude nummerduse valimine järjekorra alusel
7. GPIO.setup(11, GPIO.OUT) #Viikude 11 ja 12 väljundiks seadmine
8. GPIO.setup(12, GPIO.OUT)
9.
10. frequencyHertz = 50 #Sageduse määramine 50Hz
11.
12. pwm1 = GPIO.PWM(11, frequencyHertz) #Mootorite pulsilaiusmodulatsiooni määramine
13. pwm2 = GPIO.PWM(12, frequencyHertz)
14.
15. def liigutame_robotit(pwm1, x, pwm2, y): #Mootorite liikumise määramine
16.     if y >= 2: #vastavalt näidule
17.         if x >= 2:
18.             pwm1.ChangeDutyCycle(7) #PWM laine üleval olemise aja
19.             pwm2.ChangeDutyCycle(2) #määramine
20.         elif x >=0:
21.             pwm1.ChangeDutyCycle(0) #Parempoolne mootor
22.             pwm2.ChangeDutyCycle(4) #Vasakpoolne mootor
23.         elif x >=-2:
24.             pwm1.ChangeDutyCycle(0)
25.             pwm2.ChangeDutyCycle(6)
26.         else:
27.             pwm1.ChangeDutyCycle(4)
28.             pwm2.ChangeDutyCycle(7)
29.     elif y >= 1:
30.         if x >= 2:
31.             pwm1.ChangeDutyCycle(0)
32.             pwm2.ChangeDutyCycle(4)
33.         elif x >=-2:
34.             pwm1.ChangeDutyCycle(0)

```

```

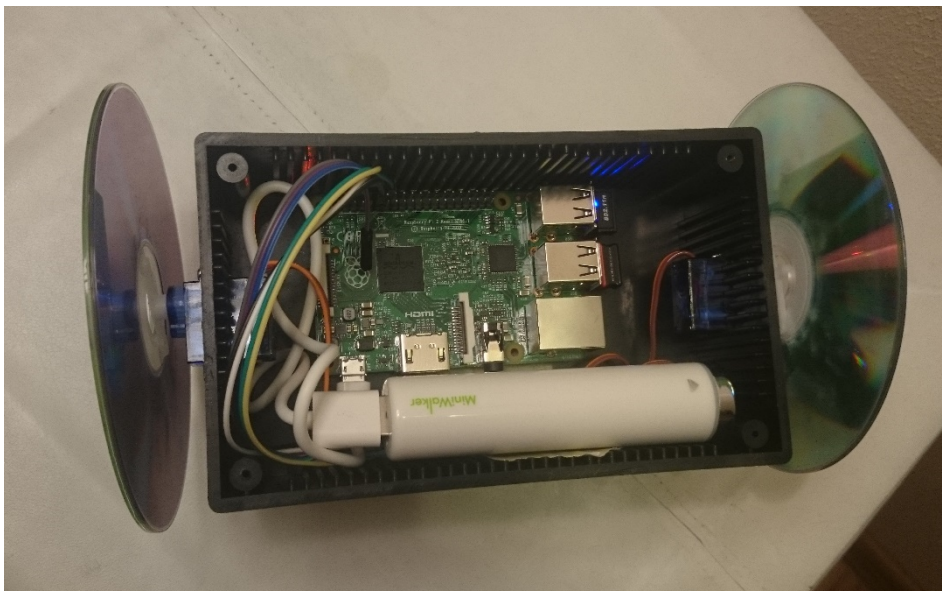
35.         pwm2.ChangeDutyCycle(0)
36.     else:
37.         pwm1.ChangeDutyCycle(0)
38.         pwm2.ChangeDutyCycle(6)
39. elif y >= -1:
40.     if x >= 2:
41.         pwm1.ChangeDutyCycle(7)
42.         pwm2.ChangeDutyCycle(2)
43.     elif x >=-2:
44.         pwm1.ChangeDutyCycle(0)
45.         pwm2.ChangeDutyCycle(0)
46.     else:
47.         pwm1.ChangeDutyCycle(2)
48.         pwm2.ChangeDutyCycle(7)
49. elif y >= -2:
50.     if x >= 2:
51.         pwm1.ChangeDutyCycle(6)
52.         pwm2.ChangeDutyCycle(0)
53.     elif x >=-2:
54.         pwm1.ChangeDutyCycle(0)
55.         pwm2.ChangeDutyCycle(0)
56.     else:
57.         pwm1.ChangeDutyCycle(4)
58.         pwm2.ChangeDutyCycle(0)
59. elif y < -2:
60.     if x >= 2:
61.         pwm1.ChangeDutyCycle(7)
62.         pwm2.ChangeDutyCycle(2)
63.     elif x >=0:
64.         pwm1.ChangeDutyCycle(6)
65.         pwm2.ChangeDutyCycle(0)
66.     elif x >=-2:
67.         pwm1.ChangeDutyCycle(4)
68.         pwm2.ChangeDutyCycle(0)
69.     else:
70.         pwm1.ChangeDutyCycle(2)
71.         pwm2.ChangeDutyCycle(7)
72. #Selles klassis käsitletakse HTTP päringuid, mis saabuvad serverisse
73. class MyHTTPHandler(BaseHTTPRequestHandler):
74.     def do_GET(self):
75.         self.send_response(200) #Lisatakse vastuse päis
76.         self.send_header("Content-type", "text/html") #Päis puhvrise
77.         self.end_headers() #Kirjutatakse puhvrise olev info väljundisse
78.         #Väljundite kirjutamise vormindamine
79.         if self.path.startswith("/update_servo"):
80.             #Muutujate määramine
81.             m = re.search("/update_servo\\?x=([^&]*)&y=(.*)", self.path)
82.             x = float(m.group(1))
83.             y = float(m.group(2))
84.             #Loodud mootorite juhtimise käivitamine
85.             liigutame_robotit(pwm1, x, pwm2, y)
86.
87.         else: #index faili kirjutamine/lugemine
88.             f = open("index.html", "rb")
89.             self.wfile.write(f.read())
90.             f.close()
91. #Loome serveri, mis kuulab kindlat porti, antud juhul :8080
92. def run(server_class=HTTPServer, handler_class=MyHTTPHandler):
93.     server_address = ('', 8080)
94.
95.     httpd = server_class(server_address, handler_class)
96.     httpd.serve_forever()
97.
98.
99. pwm1.start(0) #Mootori null oleku määramine alguses
100. pwm1.ChangeFrequency(50) #Võimalus muuta jooksvalt PWM sagedust

```

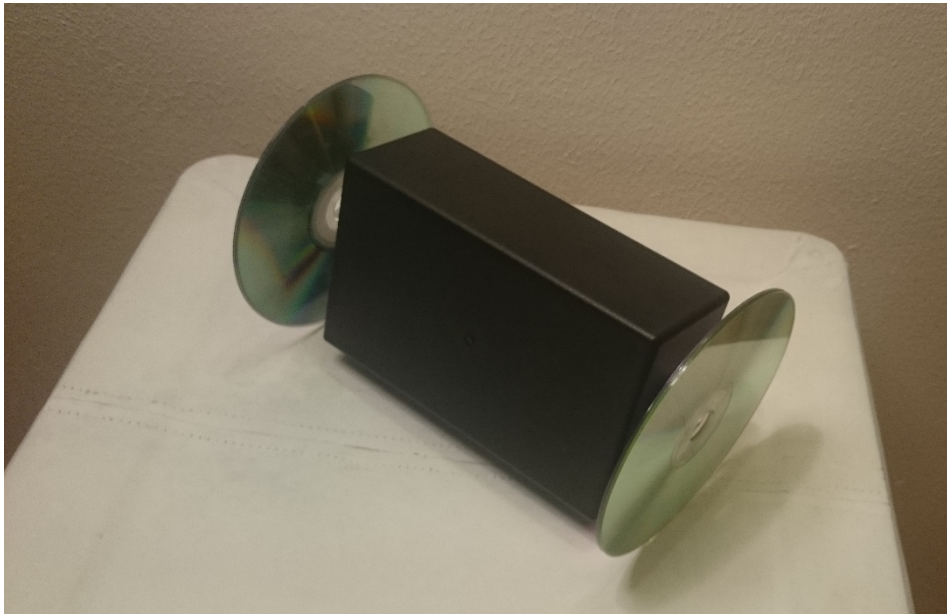
```
101.pwm2.start(0)#Mootori null oleku määramine alguses
102.pwm2.ChangeFrequency(50) #Võimalus muuta jooksvalt PWM sagedust
103.run() #Käivita kood
```

Kui veebiserver valmis, siis on vaja järgi vaadata seadme ip-aadress, et nutiseadmega õiget lehte avada. Ip-aadressi näeb terminali käsuga "ifconfig", sellelt tuleb ülesse otsida WiFi võrk, mille nimi on tõenäoliselt WLAN0 ja sealt vaadata aadress, mis on midagi sarnast 192.168.1.25. WiFi automaatselt sisselülitamiseks sisestame terminalis käsu „sudo nano /etc/network/interfaces“ ja kuhu lisame algusesse rea „auto wlan0“. Salvestame faili vajutades Ctrl+X ja seejärel klahvi Y. Seejärel tuleb käivitada veebiserver, milleks tuleb kõik redigeeritavad failid salvestada ja terminalis anda käsk "sudo python3 ServoServer.py". Kui ühtegi veakoodi ei kuvata, siis on süsteem valmis tööks. Nüüd tuleb nutiseadme veebilehitsejasse sisestada leitud ip-aadress ja selle lõpu kirjutada port 8080, ehk aadress peaks olema midagi sarnast nagu 192.168.1.25:8080. Nüüd kalluta enda telefoni ettepoole ja robot hakkab liikuma.

Kui kõik on töökorras, siis tuleb seade välja lülitada, eemaldada klaviatuur, hiir ja ekraan ning Raspberry paigutada karpi. Pi tuleb kleepida kahepoolse teibiga karbi põhja. Lisaks tuleb lisada akupank ja ka see kinnitada seadmesse. Lõplik robot on nähtav joonistel 6 ja 7.

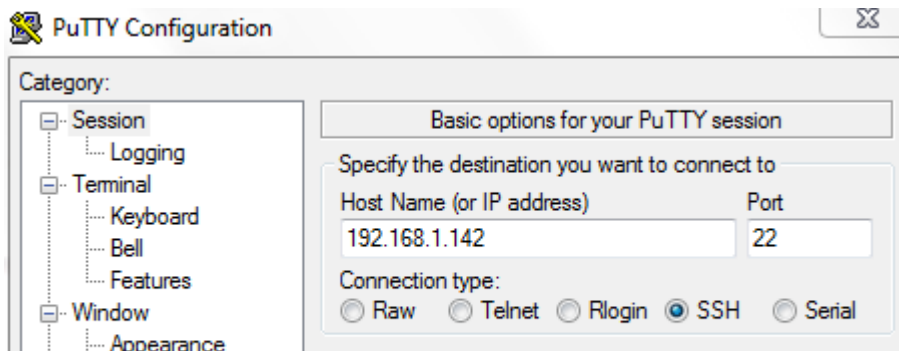


Joonis 6. Töötava roboti sisemine paigutus (autori koostatud).



Joonis 7. Valmis robot (autori koostatud).

Kuna valmis seadmele on väga ebamugav ühendada pidevalt külge nii HDMI, kui ka klaviatuuri ja hiirt, siis kasutame edaspidi selle programmeerimiseks ja uuendamiseks SSH ühendust, mis võimaldab ühendada laua arvutist end Raspberry käsureale. Selleks tuleb alla laadida näiteks PuTTY aadressilt <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. See võimaldab ühendada läbi SSH ühenduse meid robotisse. Selleks tuleb „IP address“ väljale sisestada roboti ip-aadress, nagu näha joonisel 8.



Joonis 8. Putty seaded (autori koostatud).

3.2. Roboti kasulikkus õppetöös ja kasutusvõimalused ainetundides

Hetkel kehtiva gümnaasiumi õppekava järgi on üheks õppe- ja kasvatusesmärgiks digipädevuse kasvatamine, mis näeb ette, et õpilased suudaksid kasutada uuenevaid digitehnoloogiaid ja just selle pärast ongi vaja üha enam kasutusele võtta uusi seadmeid ja lahendusi. Seda toetab ka õppekorralduse kohapealt punkt, mille järgi on koolile kohustuseks määratud tehnoloogia ja innovatsiooni teema, millega taotletakse õpilase kujunemist uuendusaltiks ja nüüdisaegseid tehnoloogiaid eesmärgipäraselt kasutada oskavaks inimeseks, kes tuleb toime kiiresti muutuvast tehnoloogilises elu-, õpi- ja töökeskkonnas. [20] Teatavasti on aga uued tehnoloogiad kallid ja nendega kaasas käimine kulukas. Raspberry Pi on aga oma hinna ja võimekuse suhtes kindlasti üks parimaid uusi seadmeid, mida koolid endale osta võiksid, sest kõigest 50€ (Raspberry + mälukaart) eest saab seadme, mida saab kasutada arvutina või projekti juhtloogikana.

Lisaks eelnevale on väga vajalik ainete omavaheline integreerimine. Antud roboti ehitamisel ja kasutamisel on võimalik ühendada väga paljude ainete oskuseid ja teadmisi. Esmalt ehituse kohapeal võiks seadme kasti ehitada näiteks tööõpetuse tunnis. Sellise kasti ehitamine, kus on ka augud servo mootorite jaoks võtaks umbes 45 minutit, kus tuleb joonistada vineerile sobivate mõõtmetega kastid, siis need välja lõigata ja kõige lõpus kokku kruvida ning viimistleda. Matemaatika tunnis saaks lahendada ülesandeid, näiteks kiiruse, läbitud maa või radiaanide arvutamine. Informaatika tunnist saab kasutada info kogumise ja programmeerimise oskusi. Füüsika tunnis saaks lisada skeemi veel LED tulesid, et luua erinevaid lisavõimalusi. Robotika aines oleksid LED tulukesed väga suureks abiks näiteks seadme vea leidmisel, sest nii saab väga ilusasti näha, millise tule juures kood katki läheb.

3.3. Roboti edasiarendamise võimalused

Käesoleva töö raames loodud robot ei ole lõpp-produkt, vaid kõigest alusplatvorm. Selline lihtne robot oleks alguseks suurematele projektidele. Sellele saaks lisada erinevaid andureid. Näiteks saaks teha targa roboti, kes vastu seinat ei sõidaks, selle jaoks oleks vaja lisada juurde ultraheli andur, millega oleks võimalik mõõta kui kaugel on takistused ja siis enne neid anda signaal, et pöörata näiteks vasakule 45 kraadi ja uuesti mõõta, kas ees on takistus või mitte.

Suurema projektina saaks näiteks lisada skeemi laseranduri ja tõmmata CD plaadile must kriips peale ning selle järgi arvutada seadme kiirus. Sellest saaks edasi juba arendada projekte, kus seade oskab ratast keerata teatud hulk kraade. Ehk põhimõtteliselt saaks ehitada kilplasprojekti, kus modifitseeritud servost saab tagasi potentsiomeetriga mootorid, aga seda tarkvaralise lahendusega.

Tegelikult saaks Raspberry Pi'ga ehitada praktiliselt kõiki pähe tulevaid projekte ja minu loodud robot võiks olla aluseks sellistele projektidele, kus on vaja kasutada kahte või enam mootorit. Kui aga mootorite ja andurite arv läheb väga palju suuremaks, siis tuleb arvestada ka sellega, et Raspberry enda pealt pole mõtet voolu võtta, sest selle viikudel pole piisavalt palju voolu, et mitmeid seadmeid jooksutada. Selle probleemi saab lahendada eraldiseisva vooluallikaga lisamisega mootoritele, milleks võib olla lihtsalt täiendav akupank.

Ühe roboti kohta tehtav investeering on kokku umbes 84€, ilma tööriistade ja lisa seadmeteta. Reaalsus oleks aga see, et ostes suurema hulga seadmeid korraga ja tellides juppe näiteks internetist, siis oleks võimalik hoida kokku peamiselt just mootorite, Raspberryde ja ka akupankade pealt.

Tabel 3. Seadmete võimalikud ostuhinnad kokkuhoiu arvestamiseks.

| Tarvik | Lõputöös kasutatud seadme hind, eurodes | Tellides internetist seadme hind, eurodes |
|------------------------|---|--|
| Raspberry Pi 2 mudel B | 40,98 | 33,33 [http://www.aliexpress.com/item/In-stock-made-in-United-Kingdom-2015-New-Original-Raspberry-Pi-2-Model-B-Broadcom-BCM2836/32301186032.html] |
| Akupank | 8,34 | 2,54 [http://www.aliexpress.com/item/2600mAh-mini-External-Portable-Power-Bank-Easy-Battery-USB-Charger-Powerbank-Mobile-Power-For-iPhone-samsung/32325613551.html] |
| Mootorid | 13,12 | 9,82 [http://www.aliexpress.com/item/360-degree-Continuous-Rotation-Servos-DC-Gear-Motor-Smart-Car-Robot-Torsion-5-5kg-cm-DC/612076270.html] |

Allikas: autori koostatud.

Et tutvustada õpilastele ka rakenduste loomist androidile, siis võiks luua ka projekte App Inventoriga, mis on võimaldab kasutades plokkide, ehitada nutiseadmele programme. Käesoleva töö kontekstis võiks luua rakenduse, mis mõõdab androidi seadme güroskoobi andmeid ja edastaks need veebiserverisse. Sellele lisaks saaks veel App Inventorit kasutades

luua lihtsalt ja kiirelt rakendusi, mis saadavad infot seadme GPS koordinaatide kohta või infot muudelt anduritelt. [21]

Käesoleva töö üheks aluseks olevat HTML5, võiks kasutada ka arendusprojektides. See võimaldab luua näiteks geolokeerimisega projekte, kasutades GPS satelliitide või levi mastide järgi lokeerimist.[22] Sarnaselt käesolevale tööle saab luua lahendusi seadme asendi määramiseks güroskoop-, kiirendus- ja kompassandurite abil.[23] Lisaks väljundina nutiseadme andmete kasutamisele, saab HTML5 abil luua ka projekte, mis kasutavad seadme sisendeid. Näiteks saaks kasutada telefoni kaamerat, et jäädvustada keskkonda, kus robot liigub.[24] Tegu on väga võimeka lahedusega, mis aitab muuta programmeerimist märksa lihtsamaks, sest võimaldab jätta vahele nutiseadme programmeerimise.

KOKKUVÕTE

Eestit peetakse üheks juhtivaks riigiks infotehnoloogia valdkonnas ja selle positiivse kuvandi hoidmiseks peame igapäevaselt kaasas käima tehnoloogiliste uuendustega. Kuid veel rohkem on meil selle jaoks vaja noori, kes oskavad ja tahavad kaasas käia tehnoloogia arenguga. Selleks on vaja õpetada noortele üha uusi ja huvitavamaid võimalusi, kuidas enda informaatika ja elektroonika teadmisi rakendada. Antud töö sisaldab juhiseid õpilaste ja õpetajatele, kuidas kasutada Raspberry Pi miniarvutit, et luua ratastega robotit, mida saab juhtida nutiseadmega. Püstitatud eesmärgiks oli luua selline seade neljandiku LEGO Mindstorms EV3 hinna eest, mis ka saavutati.

Lähemalt tutvustatakse ka Eesti koolides õpetatavaid nelja põhilist programmeerimiskeelt ja riistvarasid, millega õpilastele õpetatakse elektroonikat (Arduino ja Raspberry) ja robotikat (LEGO Mindstorms).

Loodi riistvaraline lahendus kuidas minimaalsete vahenditega luua liikuv robot Raspberry baasil. Tarkvaraline lahendus töötab puhtalt Raspberry Pi peal, millel jookseb nii veebiserver, kui ka mootorite kood. Vajalikud nutiseadme güroskoobi andmed edastatakse läbi loodud kodulehe otse veebiserverisse. Selline lahendus lihtsustab süsteemi nii palju, et nutiseadmele pole eraldi tarkvara vaja luua ja selle suurimaks kasuteguriks on see, et kõik nutiseadmed, mis HTML5 puhul güroskoobi andmeid edastavad, töötavad antud koodiga probleemideta.

Välja pakuti ideesid, kuidas sellist projekti võiks kasutada erinevate ainetundide raames õppetöö integreerimiseks ning mida võiks antud projekti aluseks võttes veel edasi arendada, nagu näiteks targa või maad mõõtvat roboti ehitamine.

KASUTATUD KIRJANDUS

1. M. Groome, L. M. Rodríguez, „How to Build a Robot. Collaborating to Strengthen STEM Programming in a Citywide System,“ in *Afterschool Matters*, 19, 1-9 (2014), <http://files.eric.ed.gov/fulltext/EJ1022073.pdf>.
2. IT Hariduse Arenduskeskus, „Programm ProgeTiiger 2015–2020,“ (2014), http://files.voog.com/0000/0034/3577/files/ProgeTiiger_kontseptsioon_VEEB_toim%202015-2020.pdf.
3. W. Rösler, „Hello World Collection,“ <http://helloworldcollection.de/> (2015).
4. C. R. Lester, D. A. Robinson, N. M. Hamilton, „Using Java and Dynamic HTML To Develop Collaborative, Computer Assisted Learning,“ presented at the *WebNet 98 World Conference of the WWW*, Orlando, 7-12. Nov. 1998. <http://files.eric.ed.gov/fulltext/ED427715.pdf>.
5. J. Quittner, „Tim Berners-Lee“ in *Time*, 153, 12 (1999), <http://eds.b.ebscohost.com.ezproxy.utlib.ee/eds/detail/detail?sid=524f0057-237c-4da0-b7bb-38b63dc62a6f%40sessionmgr114&vid=0&hid=122&bdata=JnNpdGU9ZWRzLWxpdmU%3d#db=a9h&AN=1658766>.
6. P. K. Lawlis, „Guidelines for Choosing a Computer Language: Support for the Visionary Organization,“ <http://archive.adaic.com/docs/reports/lawlis/k.htm> (1997).
7. J. Byous, „Java Technology: The Early Years,“ in *Internet archive waybackmachine*, <http://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>.
8. J. Gosling, H. McGilton, *The Java™ Language Environment* (Sun Microsystems, California, 1996).
9. „Learn About Java Technology,“ <http://www.java.com/en/about/>.
10. „Why is it called Python?,“ in *General Python FAQ*, <https://docs.python.org/2/faq/general.html#why-is-it-called-python>.
11. „LEGO® MINDSTORMS® EV3,“ in LEGO Shop, <http://shop.LEGO.com/en-FI/LEGO-MINDSTORMS-EV3-31313?fromListing=listing>.
12. L. Valk, *The LEGO MINDSTORMS EV3 Discovery Book: A Beginner's Guide to Building and Programming Robots* (No Starch Press, 2014).
13. „Frequently Asked Questions,“ in Arduino, <http://www.arduino.cc/en/Main/FAQ>.
14. „Compare board specs,“ in Arduino, <http://www.arduino.cc/en/Products.Compare>.

15. „37-in-1 Sensor Module Kit for Arduino,“ in DealExtreme, <http://www.dx.com/p/arduno-37-in-1-sensor-module-kit-black-142834#.VVzwgPntlrN>.
16. „RASPBERRY PI 2 MODEL B,“ in Raspberry Pi, <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.
17. „Raspberry Pi 2 (Mudel B),“ ITT Group, <http://www.ittgroup.ee/et/raspberry-pi/542-raspberry-pi-2-mudel-b.html>.
18. „FAQS,“ in Raspberry Pi, <https://www.raspberrypi.org/help/faqs/>.
19. „L293D-MBR 4 chan.mtr.drive+diode,“ Oomipood, <http://www.oomipood.ee/product/l293d-mbr/l293d-mbr-4-chan-mtr-drive-diode&ac=1>.
20. „Gümnaasiumi riiklik õppekava,“ Vabariigi Valitsus, määrus nr. 2, 1. sept. 2009, <https://www.riigiteataja.ee/akt/129082014021>.
21. „About Us,“ in MIT App Inventor, <http://appinventor.mit.edu/explore/about-us.html>.
22. „HTML5 for the Mobile Web: Device Orientation Events,“ in mobiForge, <http://mobiforge.com/design-development/html5-mobile-web-device-orientation-events>.
23. „Sense and sensor-bility: access mobile device sensors with JavaScript,“ in mobiForge, <http://mobiforge.com/design-development/sense-and-sensor-bility-access-mobile-device-sensors-with-javascript>
24. E. Bidelman, *Capturing Audio & Video in HTML5*, (2013), <http://www.html5rocks.com/en/tutorials/getusermedia/intro/>

SUMMARY

Building a smart device controlled robot using Raspberry Pi

Estonia is considered to be one of the leading countries in the field of information technology. In order to retain that reputation, we need to keep up with technological advancements but even more important are young people who have the know-how and the wish to follow technological developments. We need to teach these young people new and exciting ways of applying their knowledge of informatics and electronics. This paper contains instructions for students and teachers on how to use Raspberry Pi mini-computers to create a robot with wheels that can be controlled with a smart device. The goal was to create such a device for fourth of the price of Lego Mindstorms EV3, which was also achieved.

This paper also offers a thorough introduction of the four main programming languages and types of hardware that are used in Estonian schools to teach students about electronics (Arduino and Raspberry) and robotics (Lego Mindstorms).

A hardware solution was developed on creating a moving robot with minimal resources that is controlled by Raspberry. The software solution works purely on Pi that also runs the web server and the code for the servo motors. The necessary gyroscope data from the smart device is forwarded to the web server via the created webpage. That kind of a solution simplifies the process to the extent that there is no need to create separate software and also offers the advantage that all smart devices that are forwarding the gyroscope data using HTML5 encounter no operational problems with this code.

In addition, several ideas are described on the possible use of this project in different subjects for integrated learning and what could be developed further on the basis of this project, e.g. constructing smart robots or robots that measure land area.

Mina, TAAVI KARELSON

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Nutiseadmest kaugjuhitava roboti ehitamine Raspberry Pi näitel“,

mille juhendaja on Alo Peets,

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, 22.05.2015