



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Tero Virtala

# ELEKTRONISEN PORTIN OHJAUS RASPBERRY PI:LLÄ

Tekniikka ja liikenne  
2014

## TIIVISTELMÄ

Tekijä	Tero Virtala
Opinnäytetyön nimi	Raspberry pi:n avulla toteutettu portinavaaja
Vuosi	2014
Kieli	suomi
Sivumäärä	31
Ohjaaja	Jukka Matila

---

Tutkimuksen toimeksiantajana oli TJK Tietolaite OY. Työssä on tarkoituksena toteuttaa laite, jolla voidaan ohjata porttia. Tämän lisäksi laitteen on pystyttävä tunnistamaan, onko kävijällä oikeus kulkea portista.

Toteutuksessa käytettiin Raspberry Pi:tä keskeisenä laitteena. Raspberry Pi:n lisäksi työssä käytettiin rfid –lukijaa. Ohjelmointikielenä on Python. Koodit kirjoitettiin Raspberry Pi:llä. Tietokantaohjelmana toimi MySQL. Testaus toteutettiin erillisen virtalähteen, moottorin sekä Pi Face lisälevyn avulla.

Johtopäätöksenä tultiin siihen tulokseen, että työ oli hyvä toteuttaa Raspberry Pi:llä. Laitetta on mahdollista kehittää monella tavalla, kun testauksessa päästään eteenpäin.

## ABSTRACT

Author	Tero Virtala
Title	Gate Controller Implemented with Raspberry Pi
Year	2014
Language	Finnish
Pages	31
Name of Supervisor	Jukka Matila

---

The study was commissioned by TJK Tietolaite OY. The purpose of this work is was to provide device for controlling the gate. In addition the device must be able to identify whether the visitor is authorized to pass through the gate.

The main device used in the work was Raspberry Pi. In addition to Raspberry Pi rfid reader was also used. The programming language is Python. Codes were written with Raspberry Pi and the database program used was MySQL. The testing was carried out with a separate power supply, motor and Pi Face.

We came to the conclusion that it was good to use Raspberry Pi in this work. The device can be advancement in many ways when testing takes a step forward.

# SISÄLLYS

## TIIVISTELMÄ

## ABSTRACT

1	JOHDANTO.....	6
2	LAITTEIDEN TOIMINTA.....	7
	2.1 Raspberry Pi.....	7
	2.1.1 Käyttö ja historia.....	7
	2.1.2 Käyttöjärjestelmä.....	8
	2.1.3 Alkuvalmistelut ja tekniikka.....	10
	2.2 Raspberry Pi:n koodaus.....	12
	2.2.1 Python.....	13
	2.2.2 Koodaus.....	13
	2.3 GPIO -pinnit.....	15
	2.3.1 Pi Face.....	19
	2.4 Yhteenveto.....	20
3	KÄYTÄNTÖ.....	21
	3.1 Perustiedot.....	21
	3.1.1 Suunniteltavan laitteen tekninen kuvaus.....	21
	3.2 Toteutus.....	21
	3.2.1 Koodaus.....	23
	3.2.2 Kirjastot.....	26
4	TESTAUS.....	27
	4.1 Pi Face testauksessa.....	29
	4.2 Yhteenveto.....	29
5	JOHTOPÄÄTÖKSET.....	30
	LÄHTEET.....	31

**KUVIOLUETTELO**

<b>Kuvio 1.</b>	Raspberry Pi	s. 4
<b>Kuvio 2.</b>	Raspbian -työpöytä	s. 7
<b>Kuvio 3.</b>	Raspberry Pi, tekniikkaa ja oheistuotteita	s. 8
<b>Kuvio 4.</b>	Kuvakkeet Raspbianissa	s. 10
<b>Kuvio 5.</b>	Python Shell	s. 12
<b>Kuvio 6.</b>	IDLE Editor	s. 13
<b>Kuvio 7.</b>	GPIO -pinnien paikka piirilevyssä	s. 14
<b>Kuvio 8.</b>	GPIO -pinnien liitännät	s. 15
<b>Kuvio 9.</b>	Pi Face -lisälevy	s. 17
<b>Kuvio 10.</b>	ID-12LA -kortin lukija	s. 20
<b>Kuvio 11.</b>	Ohjelman rakenne	s. 22
<b>Kuvio 12.</b>	Käyttöliittymän rakenne	s. 24
<b>Kuvio 13.</b>	Näkymä käyttöliittymästä	s. 25
<b>Kuvio 14.</b>	Use Case -kaavio	s. 26
<b>Kuvio 15.</b>	Testausvaihe	s.28

## 1 JOHDANTO

Opinnäytetyön toimeksiantaja on Tietolaite OY. Elektroniset ovet ja portit ovat yleisiä nykyään. Tutkimuksessa on tarkoitus tehdä laite, joka helpottaa porttien ja ovien käyttöä sekä hallintaa. Opinnäytetyössä käytetään Raspberry Pi:tä, joka mahdollistaa avainkorttien lukemisen, tietojen käsittelyn ja siihen on olemassa paljon oheislaitteita, jotka mahdollistavat laitteen kehittämisen.

Opinnäytetyössä käydään läpi toteutuksessa tarvittavat laitteet. Tämän lisäksi kerrotaan niiden käyttötarkoituksista ja mahdollisuuksista. Työssä käydään myös läpi, mitä toteutuksessa on tehty ja missä vaiheessa.

## 2 LAITTEIDEN TOIMINTA

### 2.1 Raspberry Pi

Raspberry Pi on luottokortin kokoinen tietokone, joka käyttää linuxin käyttöjärjestelmää. Siinä on kaksi USB -porttia, joihin voi laittaa hiiren ja näppäimistön sekä HDMI -liitin näyttöä varten. Raspi on tunnettu sen pienestä koosta sekä halvasta hinnasta. Osaksi syynä sen halpaan hintaan on se, että siihen sisältyy kirjaimellisesti vain Raspberry Pi. Esimerkiksi suojakotelo tai virtalähde täytyy ostaa tarvittaessa erikseen (Monk 2013, s1). Kuvassa 1 on havainnollistettu, miltä Raspberry Pi näyttää.



**Kuvio 1.** Raspberry Pi (Raspberry Pi)

Raspberry Pi:stä on nopeasti tulossa maailmanlaajuinen ilmiö. Ihmiset alkavat pikkuhiljaa ymmärtämään, miten monipuolinen vain 35 dollarin arvoisen tietokone voi olla (Monk 2013, xvii).

#### 2.1.1 Käyttö ja historia

Kun Raspberry Pi ensimmäisen kerran julkaistiin, ihmiset olivat epäileväisiä. Luottokortin kokoinen tietokone 35 dollarin hintaan kuulosti liian hyvältä

(Richardson & Wallace 2012, vii). Kuitenkin siinä vaiheessa, kun Raspberry Pi tuli markkinoille helmikuun lopussa 2012, se alhaisen hinnan ansiosta myytiin loppuun ja toimittajien nettisivut kaatuivat tilauksien määrästä (Monk 2013, xvii). Tästä seurauksena oli, että jonot olivat pitkiä ja omaa raspia sai odottaa jopa kuukausia (Richardson & Wallace 2012, vii).

Yksi raspin parhaimmista ominaisuuksista on sen monipuolisuus. Raspilla voi muun muassa surffata netissä, katsella videoita, tai suorittaa erilaisia kokeiluja elektronisten laitteiden kanssa. Monenlaisia projekteja on mahdollista tehdä vain pienellä rakentelulla (Richardson & Wallace 2012, vii).

Monipuolisuuden lisäksi Raspberry Pi on halpa ja projektien suhteen ei tarvitse pelätä ongelmia. Raspia voi huoletta kokeilla erilaisiin tilanteisiin pelkäämättä muistin korruptoitumista, koska käyttöjärjestelmän asentaminen on todella yksinkertaista ja ilmaista. Tämä tekeekin Raspberry Pi:stä erinomaisen alustan oppimiselle ja erilaisille kokeiluille (Richardson & Wallace 2012, vii). Oppimisen kannalta on myös hyvä, että raspi tukee monia ohjelmointikieliä esimerkiksi C, Ruby, Java ja Perl (Richardson & Wallace 2012, viii).

Raspi on monella tapaa kuin tavallinen tietokone. Siihen voi halutessaan asentaa, vaikka tekstinkäsittelyohjelman (Richardson & Wallace 2012, viii)). Raspi on kuitenkin pienempi ja halvempi. Erona tietokoneeseen on koon ja hinnan lisäksi mahdollisuus helposti liittää elektronisia projekteja raspiin. Näistä esimerkkinä voidaan mainita muun muassa led-valot tai kytkimet (Richardson & Wallace 2012, viii).

### **2.1.2 Käyttöjärjestelmä**

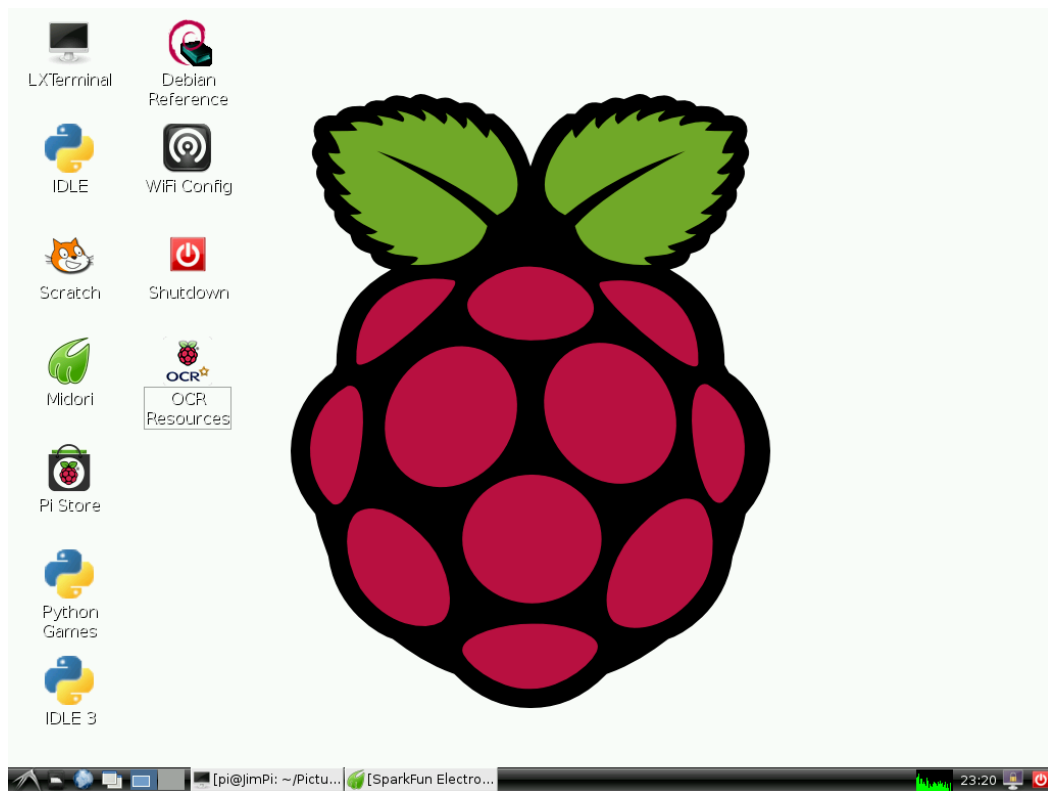
Raspberry Pi:ssä on Linuxin käyttöjärjestelmä. Linuxin jakeluversiot ovat kehittyneet vuosien varrella, yleisimpiä tietokoneissa ovat Ubuntu, Debian, Fedora ja Arch. Jokaisella on omat yhteisönsä, käyttäjänsä sekä perusohjelmansa. Pi perustuu kuitenkin mobiililaitteen piirisarjaan, joten siinä on erilaiset ohjelmistovaatimukset kuin tietokoneessa. Tästä syystä tavalliset jakeluversiot eivät käy (Richardson & Wallace 2012, s9). Monissa tietokoneissa on myös



gigabittejä RAM -muistia ja satoja gigabittejä tallennustilaa, Pi on näiltä osin rajallisempi. Erityisiä Linux distribuutioita on kuitenkin suunniteltu Pi:lle. Näistä esimerkkeinä Rasbian, Adafruit Raspberry Pi Educational Linux, Arch Linux, Xbian ja QtonPi (Richardson & Wallace 2012, s10).

Käyttöjärjestelmiä on ladattavissa useampia Rasperryn sivuilta (<http://www.raspberrypi.org/downloads>). Linux sopii hyvin raspiin. Tämä muun muassa siitä syystä, että Linux on ilmainen (Richardson & Wallace 2012, Preface ix - Preface x). Tässä työssä käytettiin Linuxin Debianiin perustuvaa Raspbiania.

Raspbianissa on LXDE (Lightweight X11 Desktop Environment) eli graafinen työpöytä-ympäristö. Kuvassa 2 näkyy Raspbianin työpöytä.

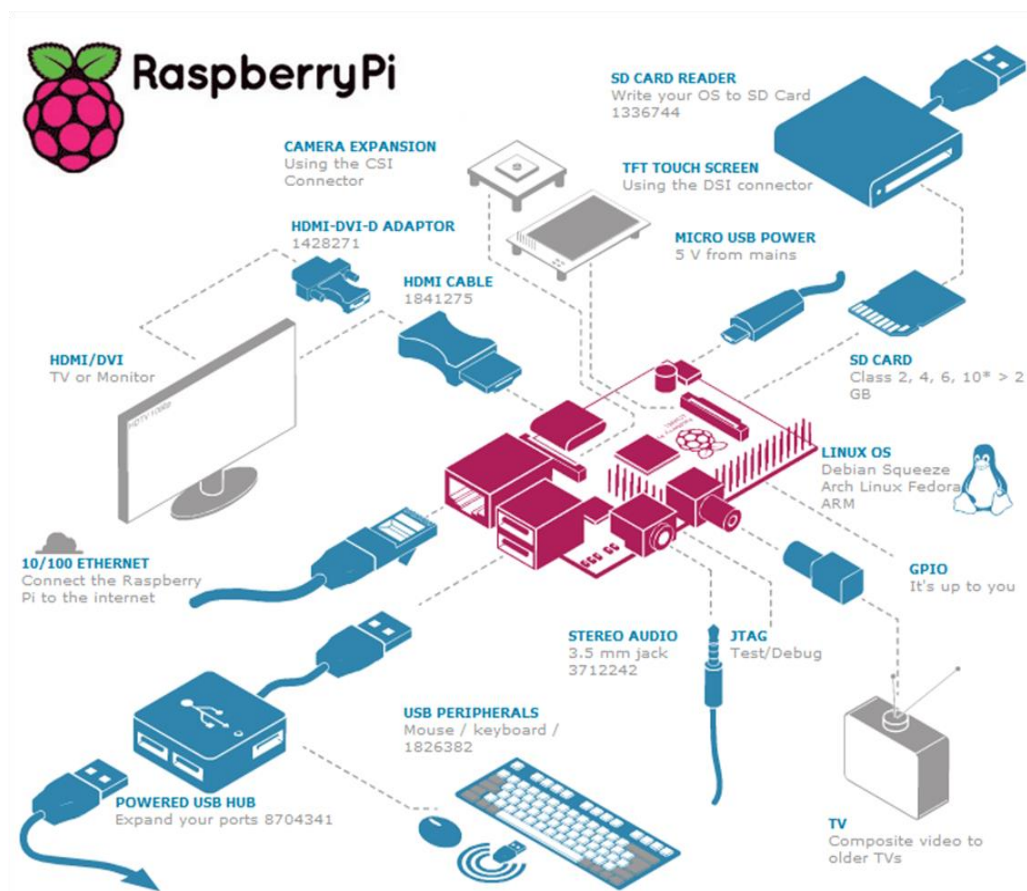


**Kuvio 2.** Raspbian työpöytä (Raspbian)

### 2.1.3 Alkuvalmistelut ja tekniikka

Raspberry Pi:n prosessori on sama kuin iPhone 3G:ssä ja Kindle 2:ssa, joten raspin tehoja voisi verrata näihin laitteisiin. Järjestelmä on 32-bittinen ja prosessorinopeus 700 MHz, joka on rakennettu ARM11 arkkitehtuurille. B-mallissa on 512 MB keskusmuistia ja kaksi USB 2.0 -porttia. Raspissa ei ole kovalevyä, vaan kaikki tieto on SD -kortilla. Siitä johtuen raspista löytyy tietysti SD -kortinlukija. Muun muassa tästä syystä on suositeltavaa hankkia oston yhteydessä kotelo, joka suojaa Pi:tä. Kotelo auttaa suojaamaan raspin itsensä lisäksi myös korttia, joka jää asennettuna sen verran ulos, että se voi vahingossa vääntyä (Richardson & Wallace 2012, 2).

Seuraavassa kuvassa 3 on esiteltynä Raspberry Pi:n tekniikkaa ja oheistuotteita. Kuvion avulla pystyy helposti hahmottamaan myös raspin mahdollisia kytkentöjä.



**Kuvio 3.** Raspberry Pi, tekniikkaa ja oheistuotteita (Raspberry Pi tekniikka)

Raspin saa internetiin suoraan RJ45 ethernet -portin avulla. Jos tätä vaihtoehtoa ei ole saatavilla, voi myös ostaa WiFi USB -sovittimen. USB -portteihin voi liittää myös ulkoisen kovalevyn tai USB -hubin, jos tuntuu, että tarvitaan ylimääräisiä portteja (Monk 2013, s3-4).

Raspissa on analoginen audio -ulostulo. Tämä on standardi 3.5 mm. Kuulokkeiden tai ilman virtalähdettä olevien kaiuttimien äänenlaatu on huono. Tämä johtuu osaksi siitä, että audio ohjainohjelmisto on vielä kehitysvaiheessa. Paremman äänenlaadun tässä tilanteessa saa HDMI:n kautta (Richardson & Wallace 2012, s4). Audio -ulostulon vieressä on standardi RCA-tyyppinen liitin, joka mahdollistaa komposiitti NTSC tai PAL -videosignaalin. Tämä on kuitenkin tarkoitettu vanhemmille televisioille ja on erittäin pieni resoluutioinen verrattuna HDMI -liitäntään, joka löytyy raspin vastakkaiselta puolelta. Tästä voidaan päätellä, että HDMI:llä varustettu televisio tai näyttö on parempi vaihtoehto. Näytön voi myös liittää tarpeen vaatiessa HDMI-DVI -muuntajalla, jos näytössä on DVI -liitin (Richardson & Wallace 2012, s4). Ethernetin ja HDMI -liitännän välissä on CSI -liitin (Camera Serial Interface), joka mahdollistaa kameramoduulin yhdistämisen suoraan levyyn (Richardson & Wallace 2012, s6).

Vähintään luokan 4 SD -kortti on suositeltavaa olla. Kortissa pitäisi olla ainakin 4 GB tilaa, eikä isommastakaan haittaa ole. Muistin tila on tärkeä, koska, kuten tutkimuksessa on jo aiemmin mainittu, kaikki tieto tulee käyttöjärjestelmän lisäksi olemaan täällä. Myös microSD -kortti adapterin kanssa on täysin käyttökelpoinen (Richardson & Wallace 2012, s6). Kortin voi ostaa valmiiksi kirjoitettuna, mutta sen voi myös tehdä itse. Tähän tarvitaan toinen tietokone, jossa on SD -kortinlukija ja muutamia työkaluja prosessin suorittamista varten riippuen tietokoneesta. Windows, Mac ja Linux koneissa prosessi eroaa hieman toisistaan, mutta kaikkiin löytyy selkeät ohjeet. Mikään ei kuitenkaan estä ostamasta muutamaa korttia, joilla voi kokeilla useita käyttöjärjestelmiä kerralla (Monk 2013, s7-8).

Pi:ssä ei ole erillistä virtanappia. Virtalähteelle löytyy microUSB -portti SD -kortin vierestä, joka on tarkoitettu vain virtalähteelle. Virtalähteen on tuotettava 5

V ja vähintään 700 mA. Kaikki kännykän laturit eivät kelpaa, vaikka niissä olisi oikea liitäntä. Osa kännykkälatureista saattaa näyttää toimivan, mutta niiden kanssa Pi voi silti käyttäytyä arvaamattomasti. MicroUSB valittiin, koska se on halpa ja USB -virtalähteitä on helppo löytää. SD -kortin vieressä on myös DSI -liitin (Display Serial Interface), johon on mahdollista kiinnittää LCD tai OLED -näyttö (Richardson & Wallace 2012, s5).

Suojakotelo on hyvä hankkia niin pian kuin mahdollista. Sen lisäksi, että kotelo suojaa raspia, on raspi vakaampi ja mukavampi käsitellä, kun siinä on kotelo ympärillä. Vaihtoehtoja on rajattomasti esimerkiksi kotelon voi ostaa valmiina tai tehdä itse. Jos on pääsy 3D -tulostimelle, internetissä on jopa osotteita, joissa voi vapaasti suunnitella oman kotelon tulostamista varten (Monk 2013, s8-9).

## 2.2 Raspberry Pi:n koodaus

Integroitu kehitysympäristö (IDE) niputettuna Pythonin ja Raspberry Pi:n kanssa on kutsumanimeltään IDLE. Alla olevassa kuvassa 4 on näkyvillä Raspbianin työpöydällä näkyvät vaihtoehdot. IDLE, integroitu kehitysympäristö Python versiolle 2, ja IDLE 3 Python versiolle 3 (Richardson & Wallace 2012, s35-36).



**Kuvio 4.** Kuvakkeet Raspbianissa (IDLE)

On yleinen käytäntö, että Pythonista on asennettuna kaksi eri versiota. Python 3 on uudempi, mutta kieleen tehdyt muutokset versioiden kaksi ja kolme välille aiheuttavat sen, että kehittyminen ei ole taaksepäin yhteensopivaa. Toisin sanoen versiot eivät ole täydellisen yhteensopivia toistensa kanssa. Vaikka Python 3 on ollut mukana jo vuosia, kesti aikaa, että siitä tuli laajalti hyväksytty. Monia käyttäjien myötävaikutteisia paketteja ei ole kehitetty Python 3:lle. Tästä johtuen täytyy olla tarkkana, että kyse on oikeasta versiosta. Esimerkkinä voidaan mainita tilanne, missä tarvitsee etsiä tietoa Python dokumentaatiosta (Richardson & Wallace 2012, s35).

### **2.2.1 Python**

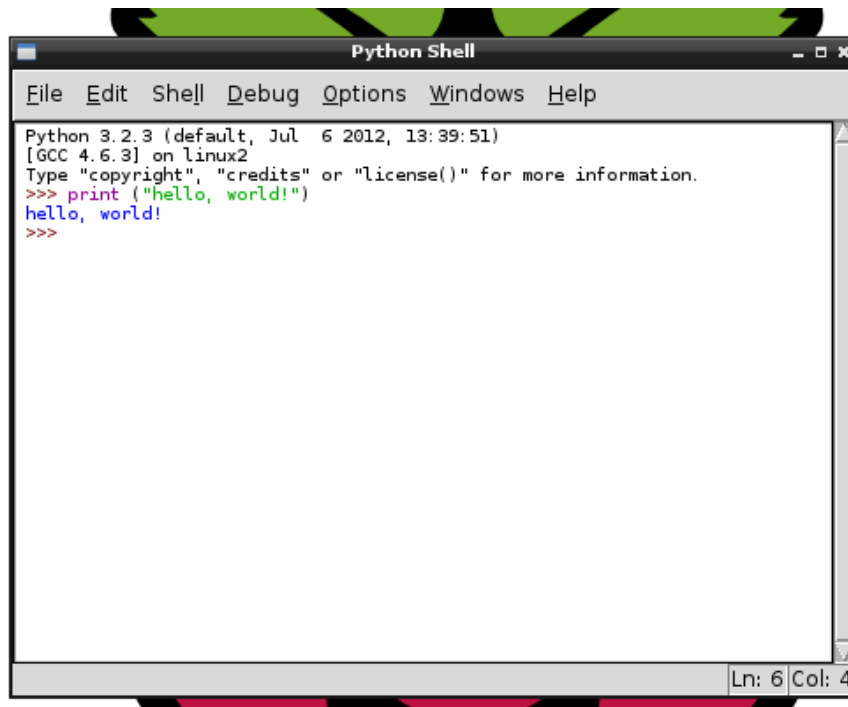
Python ohjelmointikieli on hyvä ensimmäinen kieli, koska se on selkeä ja helppokäyttöinen. On myös paljon muita ihmisiä, jotka käyttävät Pythonia, joten heidän kanssa voi jakaa koodia ja heiltä voi tarvittaessa kysyä ohjeita.

Ohjelmointikielen on kehittänyt Guido van Rossum. Vuonna 1999 van Rossum kokosi yhteen paljon luetun ehdotuksen Computer Programming for Everybody. Tämä sai aikaan kunnianhimoisen näkemyksen siitä, että ohjelmointia ruvettaisiin opettamaan Pythonia apuna käyttäen peruskouluissa, ja toisen asteen kouluissa. Raspberry Pi:n ilmestyttyä näyttää siltä, että se on oikeasti tapahtumassa (Richardson & Wallace 2012, s5).

### **2.2.2 Koodaus**

Python on tulkattava kieli, joka tarkoittaa, että ohjelman voi ajaa kääntämättä sitä ensin konekielelle. Tulkattavat kielet ovat hieman nopeampia ohjelmoida. Tietokoneelle ei tarvitse esimerkiksi erikseen kertoa, onko muuttuja numero, lista vai merkkijono, vaan tulkki selvittää sen, kun skripti ajetaan (Richardson & Wallace 2012, s35).

Python -ohjelmia voi ajaa kahdella tavalla. Interaktiivisessa ikkunassa (Python shell) voi ajaa yksittäisiä komentoja (Richardson & Wallace 2012, s35). Kuvassa 5 näkyy Python Shell -ikkuna, jolla on ajettu yksinkertainen print -komento. Komento print tulostaa sulkujen sisällä olevan tekstin.

A screenshot of a Python Shell window. The window title is "Python Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following text:

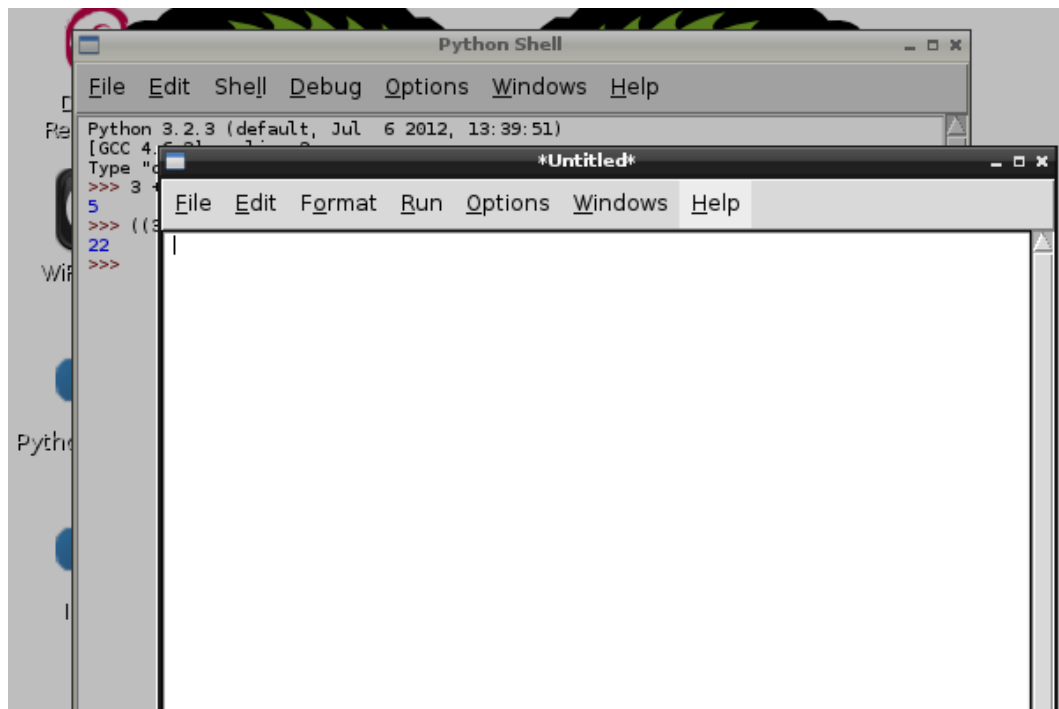
```
Python 3.2.3 (default, Jul 6 2012, 13:39:51)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> print ("hello, world!")
hello, world!
>>>
```

The status bar at the bottom right indicates "Ln: 6 Col: 4".

**Kuvio 5.** Python Shell (Python Shell)

Python Shell -ohjelmalla voi kokeilla eri Python komentoja. Komento kirjoitetaan ikkunaan, ja painetaan Enter. Ohjelma näyttää seuraavalla rivillä tuloksen (Monk 2013, s26).

Python Shell on hyvä paikka kokeiluille ja yksinkertaisille operaatioille, mutta sillä ei voi kirjoittaa ohjelmaa. Valitaan siis ikkunan ylävalikosta File, ja avataan uusi ikkuna. Uusi ikkuna avaa IDLE Editor -ohjelman, johon halutut komennot voidaan kirjoittaa. IDLE Editor ei aja komentoja heti, vaan ennen ajoa Python ohjelmat on tallennettava tiedostoihin. Tiedosto saattaa sisältää pitkän listan komentoja ohjelmointikielellä, josta ne voidaan tarvittaessa ajaa (Monk 2013, s27). Kun tiedosto halutaan ajaa, voidaan valita kuvassa 6 näkyvän Editorin ylävalikosta Run ja Run Module, jolloin ohjelman tulos tulee näkymään Python Shell -ikkunassa. Python Shell -ikkunaan kirjoitetut komennot katoavat, kun IDLE suljetaan, mutta koska ohjelma on tallennettu se voidaan ladata uudestaan (Monk 2013, s28-29).

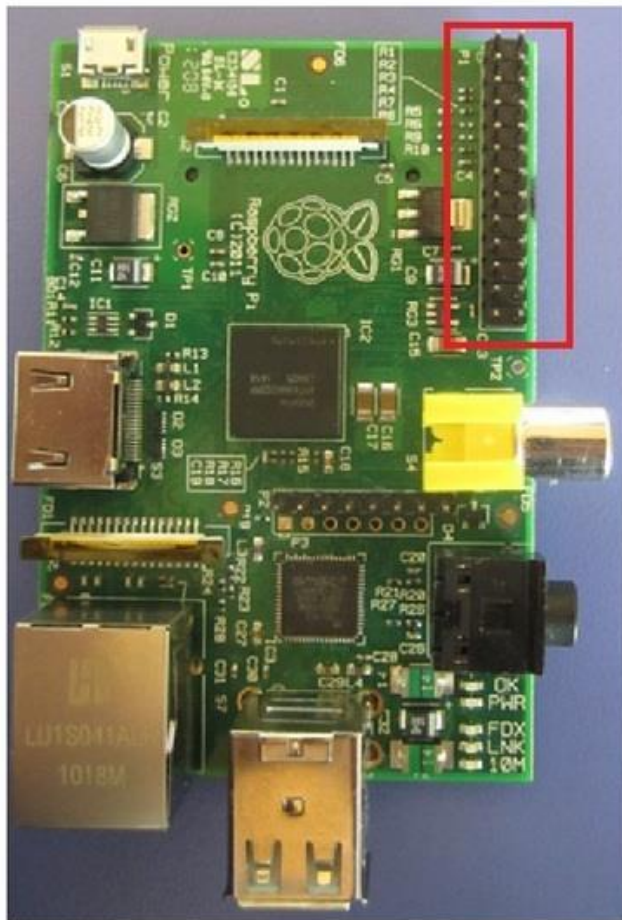


**Kuvio 6.** IDLE Editor

IDLE -ohjelmalla on helppo kirjoittaa Python ohjelmia. Toisaalta, kun kirjoitetaan IDLE -ohjelmalla koodi, joka tulostaa ikkunaan, se on erittäin hidas. Tästä syystä ohjelmia ei aina haluta ajaa suoraan, vaan käytetään LXTerminal -komentoriviohjelmaa. Terminaalissa mennään ajettavan ohjelman sijaintiin ja annetaan komento ”python” ja ”ajettava ohjelma” eli ohjelman nimi sekä perään ”.py” (Richardson & Wallace 2012, s38).

### 2.3 GPIO -pinnit

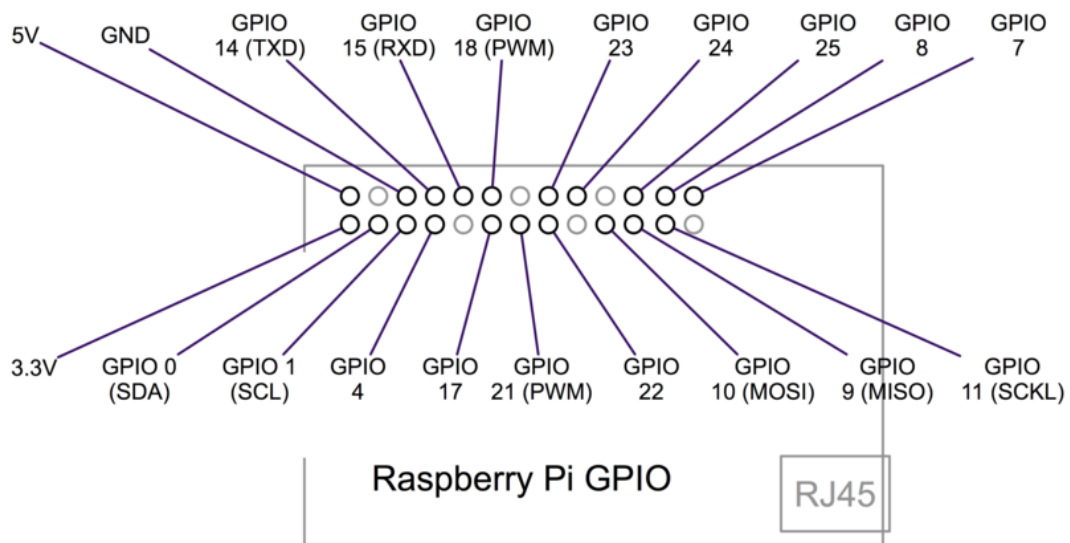
Raspberry Pi on olemukseltaan halpa Linux tietokone. Muutamat asiat kuitenkin erottavat sen näistä tavallisista kannettavista ja pöytäkoneista, joita usein käytetään tekstinkäsittelyyn, sähköpostin kirjoittamiseen tai internetin selaamiseen (Richardson & Wallace 2012, s85).



**Kuvio 7.** GPIO pinnien paikka piirilevyssä (GPIO pinnien paikka)

Yksi merkittävä ero on raspin piirilevyssä olevat GPIO (general purpose input/output) -pinnit. Nämä kuvassa 7 näkyvät pinnit mahdollistavat Raspberry Pi:n yhdistämisen suoraan elektronisiin projekteihin (Richardson & Wallace 2012, s85).





**Kuvio 8.** GPIO pinnien liitännät (GPIO pinnit)

Yllä olevassa kuvassa 8 on listattu raspin GPIO -pinnit. Pinnit, joissa lukee GPIO, on tarkoitettu yleiseen käyttöön. Toisin sanoen, mikä tahansa näistä voidaan aluksi asettaa sisäänmeno tai ulostulo pinniksi. Jos pinni asetetaan sisäänmenoksi, voidaan testata, onko sen tila 0 (alle 1.7 V) tai 1 (1.7 V tai yli). Kaikkiin GPIO -pinneihin on tarkoitettu 3.3 V. Suurempi määrä saattaa vahingoittaa raspia. Kun asetetaan ulostulo pinni, sen tila voi olla 0 V (tila 0) tai 3.3 V (tila 1) (Monk 2013, s115-116). Ulostulo pinnien avulla voidaan kontrolloida esimerkiksi LED -valoja, moottoreita tai releitä. Sisäänmeno pinnien avulla Raspberry Pi voi lukea painonappien, kytkimien tai valitsimien tilaa. Sisäänmeno pinnit mahdollistavat myös sensoreiden lukemisen, jotka mittaavat esimerkiksi lämpötilaa, valoa, liikettä tai etäisyyttä (Richardson & Wallace 2012, s85).

Osa GPIO -pinneistä on erikoista tarkoitusta varten, näiden alle on kuvassa (Kuvio 8) merkitty lyhenne sulkujen sisälle. GPIO 0 ja 1, joiden alla lukee SDA ja SCL, on tarkoitettu I2C -sarjaliikennettä varten. Se on suosittu kommunikoinnissa oheislaitteiden kanssa, kuten esimerkiksi LCD -näyttö. I2C väylää käyttää myös Pi Face, jota tässä tutkimuksessa tullaan käyttämään. Pinneihin 14 ja 15 on merkitty Rx ja Tx (Receive ja Transmit), pinnit ovat raspin sarjaporttia varten. Myös toisen tyyppinen sarjaliikenne on mahdollista GPIO -pinnien 9 – 11 (MISO, MOSI ja SCLK) kautta. Tätä sarjaliitännää kutsutaan nimellä SPI. Lopuksi pinnit

18 ja 21 joihin on merkitty PWM mahdollistavat pulssinleveysmodulaation (Monk 2013, s116).

Kun on GPIO -pinneillä varustettu tietokone, voidaan helposti luoda ohjelmia, jotka lukevat sisäänmenoja, ja ohjaavat ulostuloja useissa eri tilanteissa. Toisin kuin tyypillisissä mikrokontrollereissa joissa on ohjelmoitavat GPIO -pinnit, Raspberry Pi:ssä on muutama ylimääräinen sisäänmeno ja ulostulo, kuten näppäimistö, hiiri, näyttö ja ethernet -portti. Nämä voivat toimia sisäänmenona tai ulostulona. Kaiken lisäksi ne ovat sisäänrakennettuna, joten ylimääräisiä kytkentöjä ei tarvita.

Näppäimistö, hiiri ja näyttö eivät ole ainoita etuja Raspberry Pi:ssä kun sitä verrataan tyypillisiin mikrokontrollereihin. Raspista löytyy muutama keskeinen piirre, joka helpottaa elektronisten projektien kanssa.

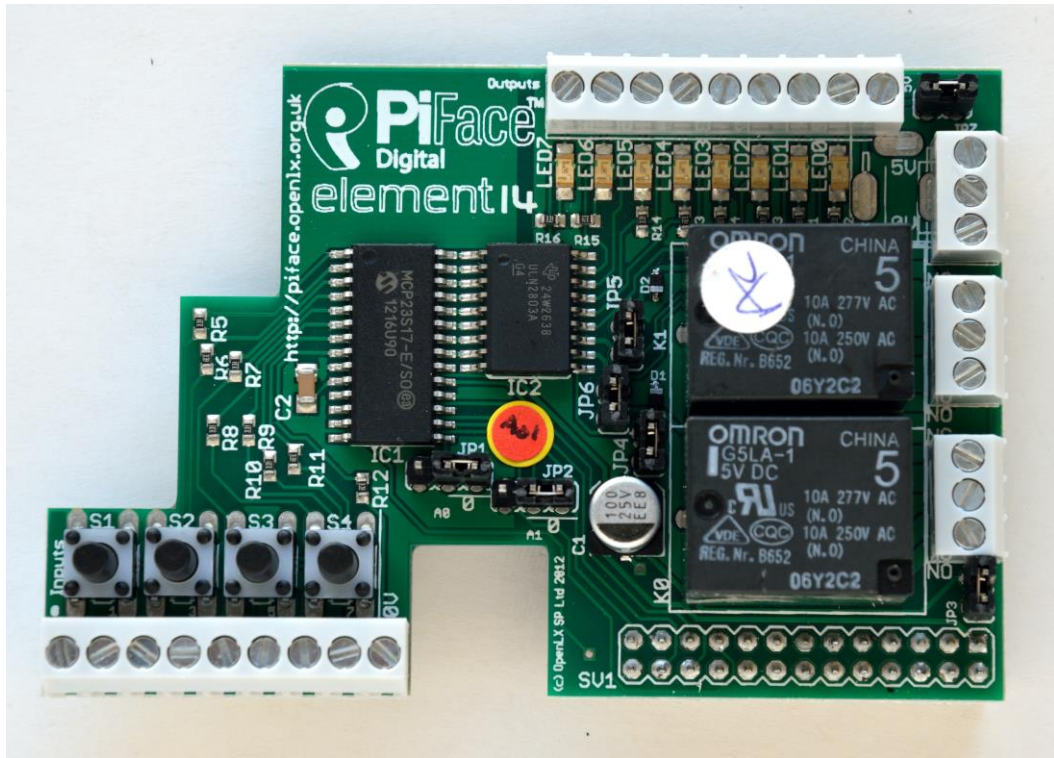
Linuxin tiedostojärjestelmä pystyy lukemaan ja kirjoittamaan dataa, joka tekee monista projekteista paljon helpompia. Voi esimerkiksi yhdistää lämpötilaa lukevan sensorin raspiin ja lukea sitä kerran sekunnissa. Jokaiset lukemat voidaan liittää lokitiedoston loppuun, josta ne on helppo ladata ja jäsentää piirto-ohjelmalle. Tulokset voidaan jopa tarpeen vaatiessa piirtää raspilla itsellään.

Raspiin pakatussa Linux -jakelussa on perus komentoriviapuohjelmat, jotka mahdollistavat tiedostojen kanssa työskentelemisen, prosessien ohjauksen ja useiden eri tehtävien automatisoinnin. Nämä tehokkaat työkalut ovat käytettävissä jokaisessa projektissa. Koska on olemassa valtava Linux -käyttäjien kommuuni, jotka ovat riippuvaisia näistä perus ohjelmista, apua löytyy usein internettiä selaamalla.

Ohjelmointikieliä on paljon ja sulautettu Linux -järjestelmä, kuten Raspberry Pi, antaa joustavuutta mieleisen kielen valinnassa. Tässä tutkimustyössä käytetään Pythonia, mutta sen tilalla voisi olla esimerkiksi C, Java, Perl tai moni muu (Richardson & Wallace 2012, s86).

### 2.3.1 Pi Face

Kuvassa 9 näkyvä Pi Face on alun perin tarkoitettu opetuskäyttöön. Se tarjoaa hyödyllisen laitteistoalustan, sekä helppokäyttöisen Python kirjaston.



**Kuvio 9.** Pi Face -lisälevy (Pi Face)

Pi Face -levyn voi asentaa suoraan raspin päälle, ja siinä on kätevät ruuviliittimet laitteiden kytkemistä varten. Pi Face ei käytä GPIO -pinnejä suoraan, vaan siinä on MCP23S17 –siru, joka kommunikoi raspin kanssa käyttäen I2C -sarjaliitäntää. Tämä tarjoaa lisälevyyn kahdeksan sisäänmenoa ja kahdeksan ulostuloa, vaikka raspin GPIO -liittimestä on käytössä vain kaksi I2C -pinniä.

Ulostulo laitteina levyllä on kaksi relettä, näitä voidaan käyttää katkaisijana virralle. Releissä on myös LED-valot, joista näkee, koska rele on aktivoitu. Levyssä on myös LED-valoja joita voidaan ohjata itsenäisesti. Neljässä sisäänmenossa on painokytkimet, jotka löytyvät niiden vierestä.

Pi Face -lisälevylle on olemassa oma Python -moduuli, joka helpottaa levyn käyttöä. Kirjastossa on LED-valoille ja releille on omat funktiot, joilla niitä ohjataan. Kirjasto täytyy ladata ja asentaa erikseen (Monk 2013, s117-119).

## **2.4 Yhteenveto**

Raspberry Pi on kokoonsa ja hintaansa nähden erittäin monipuolinen laite. Sitä on helppo soveltaa monenlaisiin projekteihin ja se soveltuu myös opetteluun. Raspberry Pi:lle on olemassa paljon oheistuotteita, joka tekee siitä entistä monipuolisemman. Internetistä löytyy tietoa ja apua, koska on paljon käyttäjiä ,jotka ovat julkaisseet omat projektinsa.

## 3 KÄYTÄNTÖ

### 3.1 Perustiedot

Tavoitteena on toteuttaa avausmekanismi porttiin, jolla voidaan hallita portin kulkua. Työ on vielä testausvaiheessa ja vaatii lisää kehittämistä, että se voitaisiin ottaa käyttöön.

#### 3.1.1 Suunniteltavan laitteen tekninen kuvaus

Työn keskeisenä laitteena toimii Raspberry Pi. Raspiin kytketään ID-12LA RFID -lukija. Ohjelmointikieli on Python ja tarvittavia tietoja säilytetään MySQL -tietokannassa.

### 3.2 Toteutus

Raspiin on saatava käyttöjärjestelmä, ennen kuin sillä voi tehdä mitään. Työ alkaa käyttöjärjestelmän kirjoittamisella SD -kortille. Tämä tehtiin erillisellä tietokoneella, jossa oli SD -kortinlukija. Raspin voi sen jälkeen käynnistää kortin kanssa. Vielä ei päästä työpöydälle, mutta alkuasetusten jälkeen Raspbianin työpöytä avautuu. Tällä hetkellä raspissa on kiinni näyttö, näppäimistö ja hiiri.

Ennen kuin raspin käyttöjärjestelmään asennetaan mitään, kannattaa varmistaa, että se on ajan tasalla. Tämä onnistuu kirjoittamalla terminaaliin komennot ”sudo apt-get update”, jonka jälkeen ”sudo apt-get upgrade” ja jakeluversion päivitys komennolla ”sudo apt-get dist-upgrade”. Sudo komennon alussa tarkoittaa, että komento ajetaan superkäyttäjän oikeuksilla. Työskentelyn helpottamiseksi päätin heti alkutekijöissä ottaa etäyhteyden raspiin. Käytin tähän tightvnc -ohjelmaa. Asensin raspiin tightvnc -serverin komennolla ”sudo apt-get install tightvncserver”. Tein tarvittavat muutokset, että sain tightvnc:n käynnistymään raspin käynnistyksen yhteydessä. Tämän jälkeen minun ei tarvinnut kytkeä raspiin kuin virtajohto ja internet -yhteys, jolloin raspin kaksi USB -porttia jäi vapaaksi mahdollista muuta käyttöä varten.

Tarvitsin laitteeseen jonkinlaisen tavan, jolla voisi hallita kulkua ja päätin käyttää kuvassa 10 näkyvää ID-12LA RFID -lukijaa. Valintani perustui lukijan yhdistämismahdollisuuksiin.



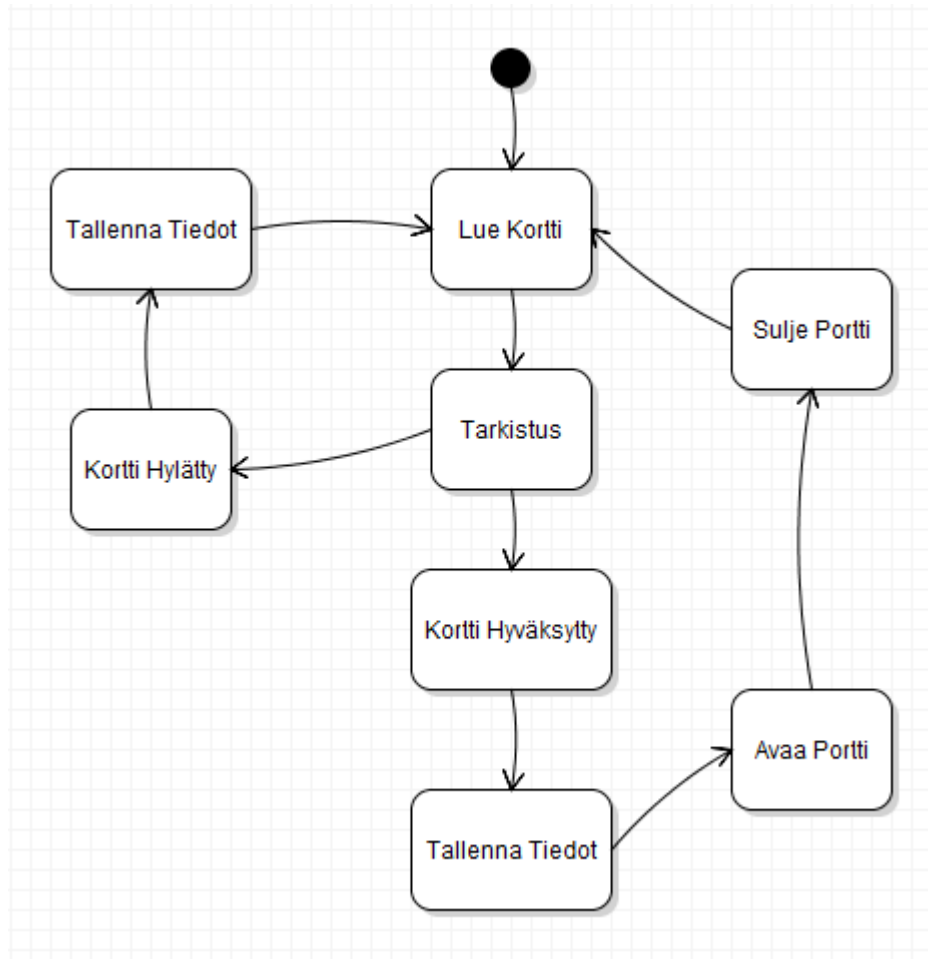
**Kuvio 10.** ID-12LA –kortinlukija (RFID –lukija)

Minulla oli kaksi USB -porttia vapaana ja työssä käytettävä RFID -kortinlukija oli mahdollista yhdistää USB -portin kautta, joten käytin vapaista porteista toisen siihen. Kortinlukijaa varten täytyi asentaa Python Serial -paketti, tämä onnistuu komennolla ”apt-get install python-serial”.

Korttien tietoja täytyy säilyttää jossain, joten asensin raspiin MySQL -tietokantaohjelman komennolla ”sudo apt-get install mysql-server python-mysqldb”. MySQL shell -ohjelman avulla pystyin luomaan Raspberry Pi:lle tietokannan, jossa on kaksi taulukkoa. Toisen taulukon avulla hallittaisiin kortteja ja toiseen tallennettaisiin lokitiedot portin käytöstä. Tein myös samat taulukot koulun omaan tietokantaan, joten tietojen tallennukseen on mahdollista pienellä muutoksella koodissa käyttää raspin omaa tietokantaa tai ulkoista.

### 3.2.1 Koodaus

Lähdin suunnittelemaan ohjelmaa, joka lukee tiedot avainkortilta, vertaa niitä tietokannassa oleviin sekä avaisi tarvittaessa portin. Kuvassa 11 näkyy ohjelman rakenne.

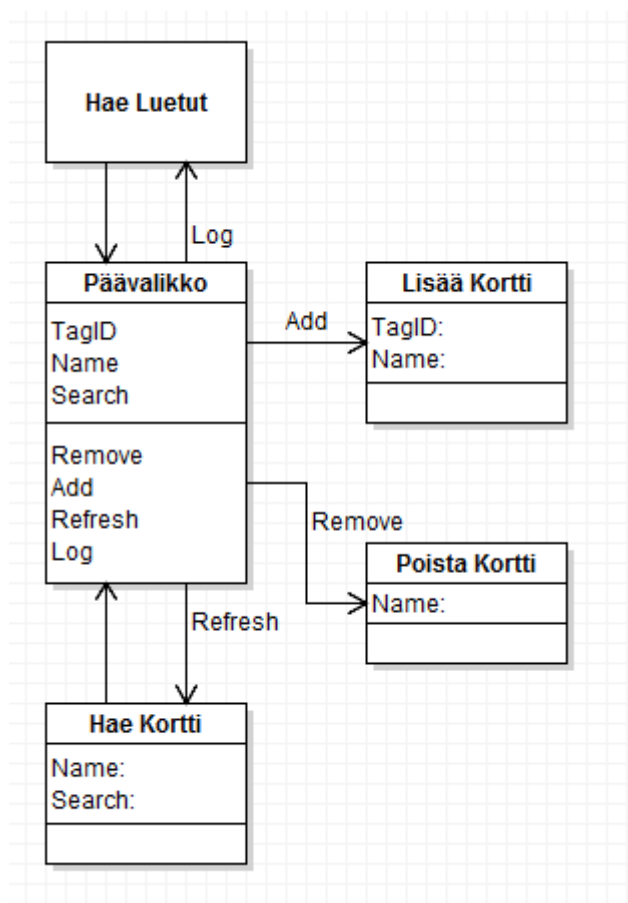


**Kuvio 11.** Ohjelman rakenne

Koodissa on while –silmukka, jonka sisällä luetaan käyttäjän avainkortti RFID -lukijan avulla. Ohjelma avaa yhteyden tietokantaan, ja tarkistaa kortin avainkoodin. Kortti hyväksytään tai hylätään, molemmissa tapauksissa ohjelma kirjaa tiedot ja ajat luetuista korteista tietokantaan sekä lokitiedostoon. Koodiin on kirjoitettu funktiot avaa ja sulje, jotka Pi Face -levyllä olevia releitä ohjaten mahdollistavat moottorin ajamisen kiinni tai auki asentoon. Jos kortin tiedot hyväksytään, ohjelma ajaa moottorin oletettuun asentoon, jossa portti olisi auki.

Tämän jälkeen ohjelma odottaa hetken, sulkee portin ja palaa while –silmukan alkuun.

Seuraavaksi tarvittaisiin ohjelma, jolla voitaisiin käsitellä tietoja. Toisin sanoen lisätä korttien tietoja tietokantaan ja poistaa niitä. Kuvassa 12 on nähtävillä käyttöliittymän rakenne. Ohjelma mahdollistaa korttien lisäämisen tietokantaan. Pääikkunaan syötetään nimi ja id –numero, jonka jälkeen ”Add” -nappia painamalla ohjelma lisää tiedot tietokantaan. Ohjelmalla voi etsiä ja poistaa haluttuja kortteja. Käyttöliittymässä on Listbox –kenttä, johon voi tuoda korttien nimiä tietokannasta hakukentän perusteella. Kun haku on suoritettu, voi kortteja poistaa Listbox –kentästä ”Remove” -napilla. Lisäksi pääikkunassa on ”Log” -nappi, jonka avulla voidaan tarkastella tietoja luetuista korteista.

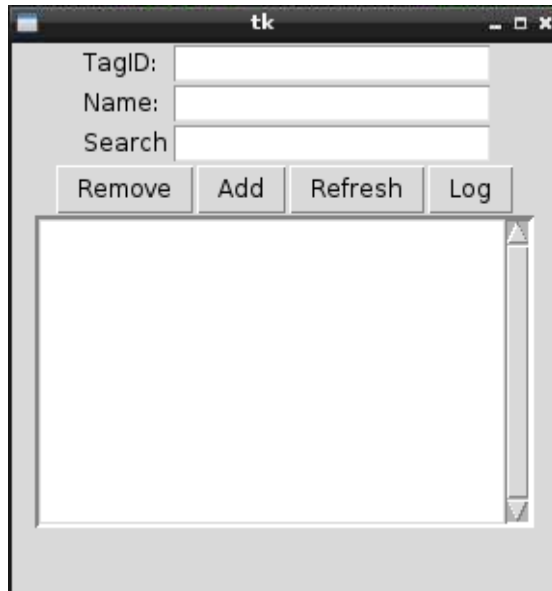


**Kuvio 12.** Käyttöliittymän rakenne

Graafisen käyttöliittymän kirjoittamiseen tarvitaan moduuli. Käyttöliittymä on toteutettu Tkinter -työkalulla. Tkinter tulee Pythonin mukana eikä vaadi

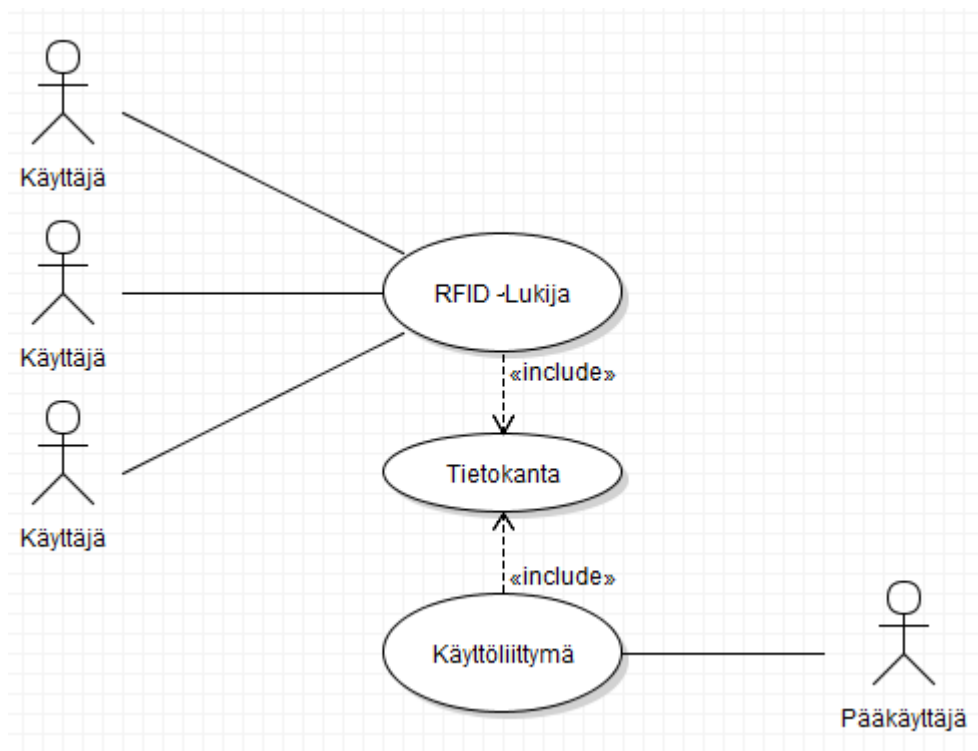


asentamista. Tkinter -moduuli helpottaa komponenttien luomisessa. Siinä on esimerkiksi painikkeita varten luokka "Button" ja tekstin esittämistä varten luokka "Label".



**Kuvio 13.** Näkymä käyttöliittymästä

Kun koodin ajaa, aukeaa ikkuna, jossa on yksinkertainen käyttöliittymä. Kuvassa 13 näkyy käyttöliittymän tekstikentät "TagID", "Name" ja "Search", joihin voi syöttää korttien tietoja. Niiden alla on napit "Remove", "Add" ja "Refresh", joilla syötettyjä tietoja voidaan käsitellä. "Log" -nappi avaa uuden ikkunan, josta voidaan tarkastella korttien käyttöhistoriaa. Sekä Listbox -kenttä, joka mahdollistaa haettujen korttien tarkastelun.



**Kuvio 14.** Use Case -kaavio

Kuten kuvasta 14 näkee, ohjelmat kommunikoivat tietokannan kautta. Tietokannassa olevien taulujen avulla saadaan käyttöliittymältä tieto avainkortista RFID –lukijalle. Samoin siis tietokannan kautta RFID –lukijalta saadaan tieto käyttöliittymälle luetuista kortteista.

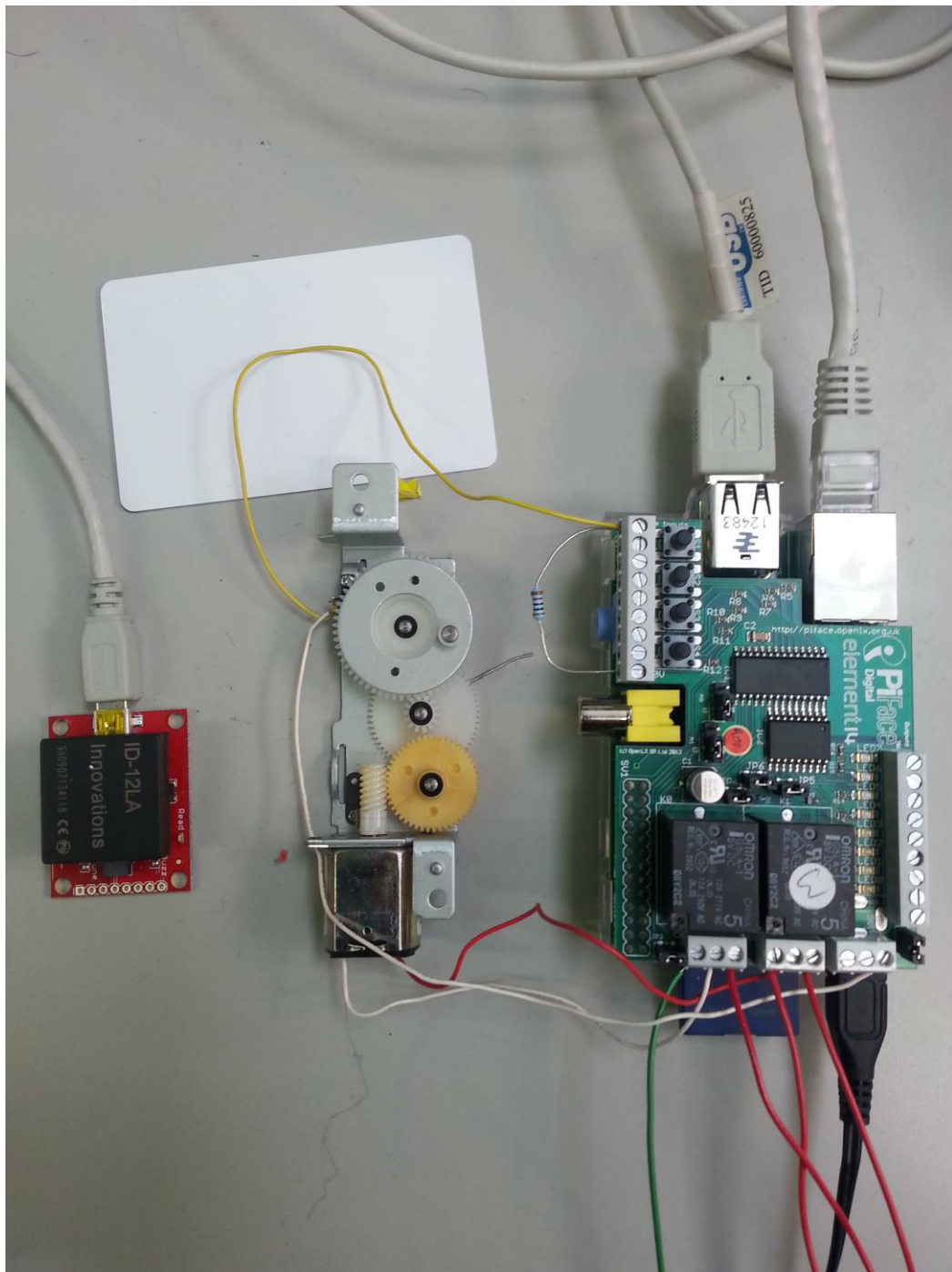
### 3.2.2 Kirjastot

Python ohjelmointikieli sisältää jo valmiiksi paljon kirjastoja, joka muodostuu kokoelmasta moduuleita eri tarkoituksiin. Kirjastomoduuleita voi käyttää lähdekoodissa kuin normaaleja funktioita, mutta ensin ne täytyy sisällyttää koodiin. Tämä onnistuu import –komennolla, koodin alussa.

Esimerkiksi tässä työssä asennettiin Python Serial paketti RFID –lukijaa varten. Kirjastomoduuli otettiin käyttöön kirjoittamalla koodin alkuun ”import serial”. MySQL -tietokantaa varten käytettiin kirjastomoduulia MySQLdb, ja Pi Face - levyä varten otettiin käyttöön pifacedigitalio.

## 4 TESTAUS

Kuvassa 15 näkyy laite testausvaiheessa. Laitteen testauksessa käytettiin Pi Face -lisälevyä, erillistä virtalähdettä sekä tietokoneen cd -aseman kelkan moottoria ja avausmekanismia. Pi Face sopii suoraan Raspberry Pi:n GPIO -pinnien päälle. Kytkenässä käytettiin Pi Face -levyn kahta relettä moottorin ohjaukseen. Releiden asentoja koodissa muuttamalla moottorin saa pyörimään eteenpäin ja taaksepäin sekä pysähtymään.



**Kuvio 15.** Testausvaihe

Cd -aseman avausmekanismissa, jota moottori pyörittää, oli vain toisella puolella kytkin. Tämä kytkin pysäyttää moottorin. Tästä syystä portin avausfunktio oli koodissa toteutettu ajastamalla.

#### 4.1 Pi Face testauksessa

Pi Face kommunikoi Raspberry Pi:n kanssa SPI –liitännän kautta. Raspbianissa on valmiiksi SPI –liitännän ajuri, mutta oletusasetuksena se ei ole käytössä. Jos halutaan SPI –ajuri pysyvästi käyttöön, täytyy käydä muokkaamassa tiedostoa ”raspi-blacklist.conf”. Tämä onnistuu kirjoittamalla komentoriville ”sudo nano etc/modprobe.d/raspi-blacklist.conf”. Tiedostossa on rivi, jossa lukee ”blacklist spi-bcm2708”, rivin eteen muokataan ”#”. Muokattu tiedosto tallennetaan. Seuraavaksi asennetaan Pi Face kirjastot ja muutetaan oikeudet SPI –liitännälle. Tähän on olemassa valmis skripti ”sudo apt-get update wget <http://pi.cs.man.ac.uk/download/install.txt> | bash”. Ohjelman asennus kestää muutaman minuutin, jonka jälkeen käynnistetään Raspberry Pi uudestaan komennolla ”sudo reboot”.

Kuten kuvassa 15 näkyy moottori on kytketty sisääntuloon. Pi Face kirjastosta löytyy funktio ”digital\_read(pin\_numero)”, jolla pystyy lukemaan sisääntulojen tilaa. Näin on mahdollista lukea koodissa, missä asennossa moottori on. Moottorin ohjaus toimii releiden avulla, joita ohjataan ulostuloilla yksi ja nolla. Koodissa ulostulojen tilaa voidaan muuttaa funktiolla ”digital\_write(pin\_numero, tila)”.

#### 4.2 Yhteenveto

Raspberry Pi lukee kortin, tarkistaa sen kelpoisuuden ja ajaa tarvittaessa testauksessa käytetyn moottorin oikeaan asentoon. Käyttöliittymällä voi lisätä, etsiä ja poistaa kortteja tietokannasta. Ohjelmalla voi myös tarvittaessa hakea tiedot korttien käytöstä. Ohjelmat toimivat, mutta laitteessa on vielä kehitettävää ennen kuin sen voi ottaa käyttöön.

## 5 JOHTOPÄÄTÖKSET

Kun aloitin, minulla ei ollut käytännön kokemusta Raspberry Pi:stä, eikä Python ohjelmointikielestä. Projektin edetessä alkoi kuitenkin selviämään, mitä kaikkea Raspberry Pi:llä on mahdollista tehdä. Työtä tehdessä tuli vastaan myös joitain ideoita, miten työtä voisi jatkossa kehittää.

Korttien lisääminen tietokantaan on tällä hetkellä hieman hidasta, joten ehkä toinen kortinlukija nopeuttaisi prosessia. Raspin SD –kortti saattaa korrumpoitua, jos sen sammuttaa väärin, joten sähkökatkon varalle olisi hyvä olla suoja.

Johtopäätöksenä voidaan hyvin sanoa, että Raspberry Pi on monipuolinen ja helppokäyttöinen apuväline erilaisiin projekteihin. Se on myös todella mielenkiintoinen, koska mahdollisuuksia Raspberry Pi:in hyödyntämiseen löytyy koko ajan lisää. Suosittelen ehdottomasti kaikille Raspberry Pi:ihin tutustumista.

## LÄHTEET

Monk, S. 2012. Programming the Raspberry Pi: Getting Started With Python. McGraw-Hill.

Richardson, M. & Wallace, S. 2012. Getting Started With Raspberry Pi. O'Reilly Media inc.

GPIO pinnien paikka Viitattu 15.2.2014.

[http://bhavyanshu.me/assets/imag/raspi/pi\\_gpio.jpg](http://bhavyanshu.me/assets/imag/raspi/pi_gpio.jpg)

GPIO pinnit Viitattu 15.2.2014.

[https://learn.adafruit.com/system/assets/assets/000/003/059/medium800/learn\\_raspberry\\_pi\\_gpio-srm.png?1396790782](https://learn.adafruit.com/system/assets/assets/000/003/059/medium800/learn_raspberry_pi_gpio-srm.png?1396790782)

IDLE Viitattu 6.2.2014. [http://www.raspberrypi-spy.co.uk/wp-content/uploads/2013/10/minecraft\\_desktop\\_icon.png](http://www.raspberrypi-spy.co.uk/wp-content/uploads/2013/10/minecraft_desktop_icon.png)

Pi Face Viitattu 21.4.2014.

<http://www.element14.com/community/servlet/JiveServlet/download/1452-113865/PiFace.JPG>

Python Shell Viitattu 6.2.2014. <http://images.jeremymorgan.com/raspberry-pi-programming-python-1.png>

Raspberry Pi Viitattu 2.2.2014.

<http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg>

Raspberry Pi tekniikka Viitattu 4.2.2014. <http://1.bp.blogspot.com/-ExbOsinGESo/UwutAw6YBUI/AAAAAAAAAFKM/-HSOoBBj27Y/s1600/Raspberry+pi+accesorios.png>

Raspbian Viitattu 3.2.2014.

<https://dlnmh9ip6v2uc.cloudfront.net/assets/f/6/4/1/4/52b0aa14757b7f95688b456d.png>

RFID -lukija Viitattu 21.4.2014. [http://www.robot-](http://www.robot-italy.com/media/catalog/product/cache/3/image/d43192dcd82ea942982b4b1d2a6e2479/1/1/11839-02.jpg)

[italy.com/media/catalog/product/cache/3/image/d43192dcd82ea942982b4b1d2a6e2479/1/1/11839-02.jpg](http://www.robot-italy.com/media/catalog/product/cache/3/image/d43192dcd82ea942982b4b1d2a6e2479/1/1/11839-02.jpg)