

# **Raspberry Pi IoT-laitteena**



Ammattikorkeakoulututkinnon opinnäytetyö

Visamäki, tietojenkäsittelyn koulutusohjelma

Syksy, 2017

Alexi Juhola

Tietojenkäsittelyn koulutusohjelma  
Visamäki

---

<b>Tekijä</b>	Aleksi Juhola	<b>Vuosi</b> 2017
<b>Työn nimi</b>	Raspberry Pi IoT-laitteena	
<b>Työn ohjaaja/t</b>	Tommi Lahti, Lauri Salminen	

---

## TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli luoda Raspberry Pi -tietokoneesta ja USB-vaa'asta laite, jolla mitataan esineiden painoa ja lähetetään tieto Linux-palvelimelle. Toimeksiantaja oli Funity Oy.

Teoriaosuudessa esitellään Raspberry Pi -tietokoneen toimintaperiaatteita ja tietokoneen historiaa, ja käydään läpi myös erilaisia saatavilla olevia käyttöjärjestelmiä. Samalla selvitetään Raspberry Pin potentiaalia erilaisissa projekteissa ja IoT (Internet of Things) -ympäristöissä.

Työn käytännön osuudessa jatkokehitettiin Usbscale-nimistä C-kielellä ohjelmoitua avoimen lähdekoodin ohjelmaa, ja tuotettiin ohjelma sopimaan toimeksiantajan tarpeisiin. Työssä selvitettiin myös, mitkä ohjelmat ovat käytännöllisiä kehittämiseen IoT-ympäristössä.

**Avainsanat** Raspberry Pi, IoT, esineiden internet, avoin lähdekoodi

**Sivut** 22 sivua

Degree Programme in Business Information Technology  
Visamäki

---

<b>Author</b>	Aleksi Juhola	<b>Year</b> 2017
<b>Subject</b>	Raspberry Pi as IoT-device	
<b>Supervisors</b>	Tommi Lahti, Lauri Salminen	

---

#### ABSTRACT

The goal of this Bachelor's thesis was to create a device using Raspberry Pi and USB scale to measure weight from an item and sending the data to Linux server. Thesis was commissioned by Funity Oy.

The following theoretical section examines the operating principles and history of the Raspberry Pi computer. Different available operating systems will also be examined. We will also be looking at the potential of the Raspberry Pi in different projects and IoT (Internet of Things) environments.

The practical part of the thesis focuses on further developing of the open source program Usbscale, and using it as the basis to create a program that suits the client's needs. The thesis also delves into different programs that are useful for developing in IoT environments.

**Keywords** Raspberry Pi, IoT, Internet of Things, Open Source

**Pages** 22 pages

# SISÄLLYS

1	JOHDANTO.....	1
2	RASPBERRY PI YLEISESTI .....	2
2.1	Mikä on Raspberry Pi .....	2
2.2	Käyttöjärjestelmät.....	2
2.2.1	Raspberry NOOBS .....	3
2.2.2	Raspbian .....	3
2.2.3	RetroPie .....	3
2.2.4	Windows 10 IoT Core .....	5
2.3	Rauta .....	5
3	RASPBERRY PIN HISTORIA.....	8
3.1	Alkuperä .....	8
3.2	Kehityshistoria.....	8
4	IOT JA RASPBERRY PI .....	10
4.1	Tietokoneesta IoT-laitteeksi.....	10
4.2	Usb-vaaka .....	10
4.3	Raspberry ja modaaminen .....	11
4.3.1	TeaPi .....	12
4.3.2	Raspberry sääasemana.....	12
5	KEHITTÄMINEN RASPBERRY PILLÄ.....	14
5.1	Etäohjaus ja -hallinta .....	14
5.2	Bash-skriptaus ja automaattinen ajaminen .....	15
5.3	C-kieli ja Geany.....	16
5.4	C-kielisten ohjelmien ajaminen Raspberryillä.....	18
5.5	Usbscale ohjelma .....	20
6	YHTEENVETO .....	22
	LÄHTEET .....	23

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on luoda Raspberry Pi tietokoneesta ja USB vaa’asta laite, jolla voidaan mitata esineiden painoa. Mittaustulokset lähetetään Linux-palvelimelle mistä sitä voidaan tutkia vaivattomasti esimerkiksi puhelimella. Pääasiallinen loppukäyttö tulee olemaan asioiden mittaamista jääkaapissa, kuten esimerkiksi maitopurkin painon mittaaminen.

Opinnäytetyön tilaajana toimii Funity Oy ja tuote tulee tilaajan omaan käyttöön. Opinnäytetyön tarkoituksena on luoda siis käytännönläheinen laite, jota on helppo ja vaivaton ylläpitää.

Opinnäytetyön ideana on myös tutkia tietokoneiden ja IoT-laitteiden suhdetta, tuleeko tietokoneesta jossain vaiheessa IoT-laite vai onko se aina vain tietokone sen käyttötavasta riippumatta.

## 2 RASPBERRY PI YLEISESTI

### 2.1 Mikä on Raspberry Pi

Raspberry Pi on yhden piirilevyn tietokone ja kooltaan se on vain noin pankkikortin kokoinen. Tästä huolimatta se on kuitenkin kokonainen tietokone ja sisältää täyden version Linuxista. Raspberryyllä voi tehdä samoja asioita kuin isoillakin tietokoneilla, kuten selaamaan internetiä tai vaikkapa katselemaan videoita. (Raspberrypi Foundation n.d.a)

Raspberry Pitä käytetään kuitenkin yleensä eri tavoin kuin tavallisia tietokoneita. Sen koon ja suhteellisen tehokkaiden komponenttienstakia sitä käytetään yleensä erilaisten projektien perustana. Esimerkiksi Raspberry Pin omilla sivuilla kerrotaan vain mielikuvituksen olevan rajana Raspberryn käyttämisessä: osa luo oman sääaseman, kun taas toiset voivat luoda vaikkapa omanlaisensa robotin.

### 2.2 Käyttöjärjestelmät

Raspberry Pin yksi tärkeimmistä ominaisuuksista on sen käyttöjärjestelmien muokattavuus. Raspberryn käyttöjärjestelmä asennetaan Micro SD-muistikortille, ja käyttöjärjestelmän vaihtaminen tapahtuu vain muistikorttia vaihtamalla. Suurin osa näistä käyttöjärjestelmistä on maksuttomia ja perustuu Linux-käyttöjärjestelmään. Koska Linux itsessään on avoimen lähdekoodin käyttöjärjestelmä, voi käyttäjä muokata käyttöjärjestelmäänsä niin kuin itse haluaa tai vaikkapa luoda itse oman versionsa käyttöjärjestelmästä. Raspberry Pihin kuitenkin saa myös Windows-käyttöjärjestelmän, jos kehittäjä näin itse haluaa.

Linux on tärkeä osa Raspberry Pin olemassaoloa. On vaikea kuvitella, että Raspberryn tapaista laitetta olisi edes alun perin ollut mahdollista toteuttaa ilman Linuxin kaltaista ilmaista, suosittua ja avoimen lähdekoodin käyttöjärjestelmää. Linuxin muokattavuus ja sen pohja Android-laitteissa on luonut Raspberry Pille erinomaiset mahdollisuudet onnistua aivan uusilla markkinoilla.

### 2.2.1 Raspberry NOOBS

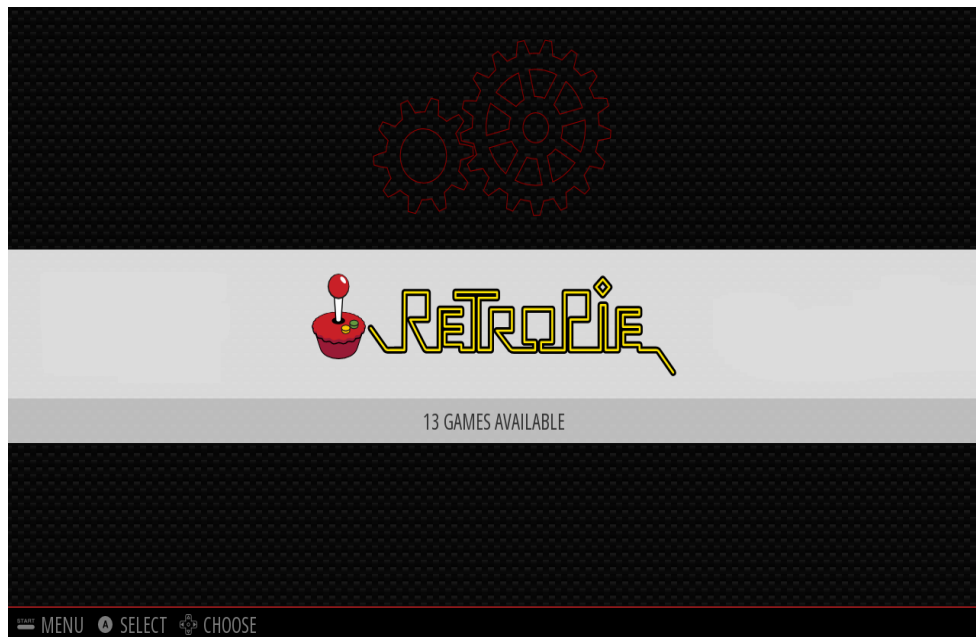
Raspberry NOOBS (New Out Of Box Software) on aloittelijoille suunnattu ohjelmisto, käytännössä kokoelma erilaisia käyttöjärjestelmiä Raspberry Pille. Raspberry NOOBS:in mukana tulee esiasennettuna Raspbian-käyttöjärjestelmä, mutta asennusvaiheessa on mahdollista valita listasta myös muita käyttöjärjestelmiä, kuten esimerkiksi Pidora, LibreELEC, OSMC, RISC OS ja Arch Linux (näiden asennus vaatii internetyhteyden, sillä ne ladataan valittaessa).

### 2.2.2 Raspbian

Raspbian on Raspberry Pi Foundationin omilla sivuillaan suosittelema käyttöjärjestelmä Raspberrylle. Se pohjautuu Debian-nimiseen Linux-käyttöjärjestelmään ja sen voi ladata myös heidän sivuiltaan. Raspbian tarjoaa monia ohjelmistoja niin kehittäjille kuin aloittelijoillekin. Siinä on monia erilaisia ohjelmistokehitysympäristöjä valmiiksi asennettuna, kuten esimerkiksi BlueJ Java IDE, Geany, MonoDevelop, Node-RED, Python 2 ja Python 3. Lisäksi siihen on asennettuna Chromium-selain ja avoimen lähdekoodin LibreOffice, joka on käytännössä ilmainen toimisto-ohjelmisto.

### 2.2.3 RetroPie

RetroPie on esimerkki siitä kuinka monipuolisia käyttöjärjestelmät voivat olla, kun alusta ja käyttöjärjestelmä luodaan avoimella lähdekoodilla. RetroPie on käyttöjärjestelmä, johon on koottu monia erilaisia emulaattoreita vanhoista (ja vähän uudemmistakin) pelikonsoleista. Emulaattorit vaativat luonnollisesti BIOS-tiedoston, jolla voidaan todistaa, että emulaattorin käyttäjä todella omistaa kyseisen laitteen, jolloin kyseessä ei ole piratismi. Piratismi kuitenkin on valitettavan yleistä RetroPien käyttäjien keskuudessa, koska BIOS-tiedostoja on helppo ladata internetistä.



Kuva 1. RetroPien näkymä asennuksen jälkeen.



#### 2.2.4 Windows 10 IoT Core

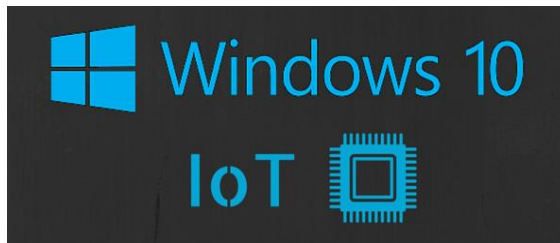
Microsoft on luonut ilmaisen käyttöjärjestelmän Raspberry Pille, joka on nimeltään Windows 10 IoT Core. Kyseinen käyttöjärjestelmä on luotu pienille laitteille, näissä laitteissa voi olla näyttö, mutta Windows 10 IoT Core toimii myös ilman näyttöä olevassa laitteessa. Tuettuja laitteita ovat Raspberry Pi 2 ja 3, Arrow DragonBoard 410c ja MinnowBoard MAX.

Tämän Windows 10 IoT Coren tärkeimpinä ominaisuuksina ovat UWP (Universal Windows Platform), Visual Studio käyttömahdollisuus ja Connect the Dots ohjelmiston käyttömahdollisuus. (Microsoft, n.d.)

UWP:n vuoksi Windows 10 IoT Corella luodut ohjelmat toimivat siis kaikilla Windows käyttöjärjestelmillä, joka tarkoittaa sitä, että UWP-laitteilla luodut ohjelmat toimivat myös Windows puhelimissa ja tietokoneissa ilman sen suurempia ohjelmiston muokkauksia.

Visual Studio käyttömahdollisuus on myös monelle varmasti iso plussa, ja esimerkiksi Raspberry Pille kehittäessä on mahdollista kirjoittaa itse koodi Windows-tietokoneella ja ajaa se vain Raspberry Pillä.

Connect the Dots on Microsoftin luoma avoimen lähdekoodin ohjelmisto, jolla on tarkoitus auttaa pieniä laitteita yhdistymään Microsoft Azure IoT:iin, jolla voidaan seurata ja tutkia kaikkia kyseiseen verkkoon liitettyjä laitteita.



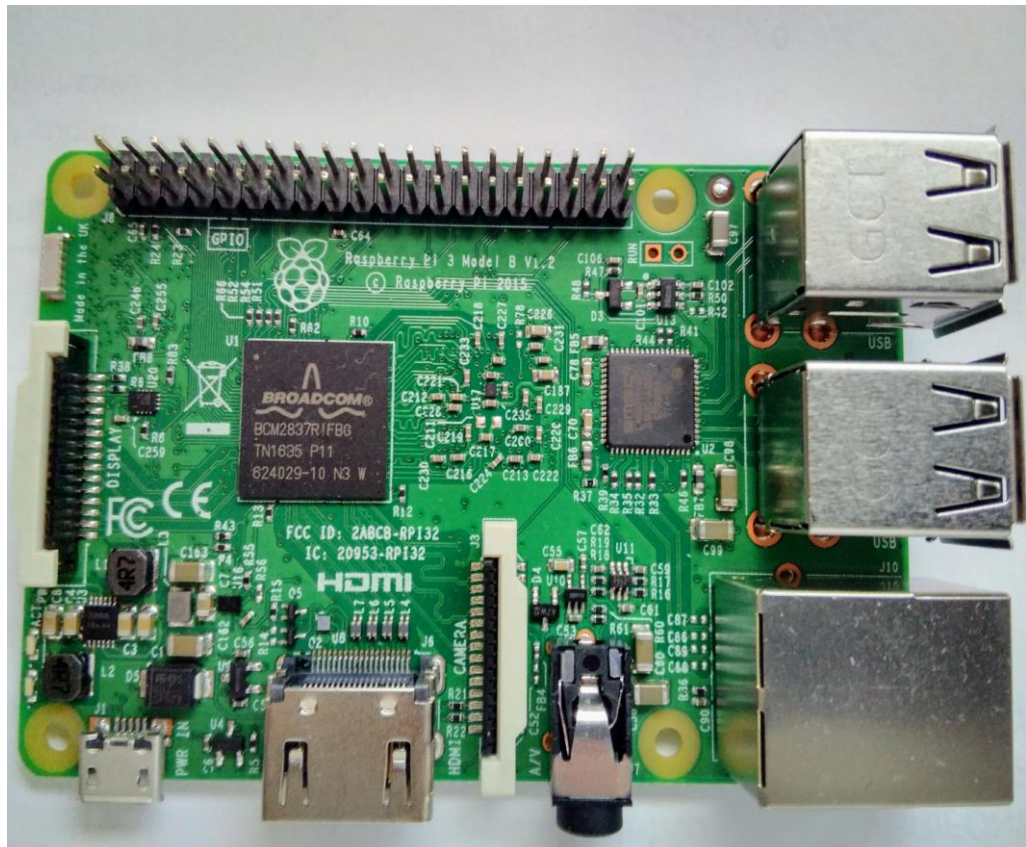
Kuva 2. Windowsin luoman IoT Core-hankkeen logo.

#### 2.3 Rauta

Tässä opinnäytetyössä käytetään Raspberry Pi 3 B -mallia, joka on kirjoitushetkellä uusin ja tehokkain Raspberry Pi. Edeltäjiinsä verrattuna siinä on nopeampi prosessori (A 1.2GHz 64-bittinen neliydin ARMv8 prosessori), sisäänrakennettu WLAN-piiri ja Bluetooth 4.1. Raspberry Pi 3 on kuitenkin täysin samankokoinen ja täysin yhteensopiva Raspberry PI 1 ja 2 kanssa.

Raspberry Pi 3 B mallissa on myös muun muassa 4 USB porttia, täysi HDMI portti ja 40 GPIO pinniä. (Raspberry Pi Foundation n.d.c)

GPIO (eli General Purpose I/O) pinnit voi ohjelmoida halutessaan vastaanottamaan tai lähettämään erilaisia signaaleita. Näiden avulla yleensä hoidetaan kaikki datan lähetykset ja vastaanotot, mutta tässä opinnäytetyössä käytetään ainoastaan USB-yhteyttä.



Kuva 3. Kuvassa Raspberry Pi 3 B -malli, jota käytetään tässä opinnäytetyössä.

Raspberry Pi:ssä oleva ARMv8-prosessori on sen suurin ero isoihin pöytätietokoneisiin tai läppäriin. ARM-prosessoreja käytetään yleensä vain puhelimissa sen alhaisen virrankulutuksen, suhteellisen tehokkaan arkkitehtuurinsa takia ja sen pienen fyysisen koon takia.

ARM-arkkitehtuuri on RISC suunnittelufilosofiaa, kun taas tietokoneissa normaalisti käytettävä x86-arkkitehtuuri (Intelin valmistama) tai AMD64 (AMD:n kehittämä 64-bittinen suoritinarkkitehtuuri) perustuu CISC suunnittelufilosofiaan. RISC tulee sanoista Reduced Instruction Set Computer ja se tarkoittaa sitä, että kyseinen arkkitehtuuri on suunniteltu pitämään prosessorille tulevat käskyt mahdollisimman yksinkertaisina ja lyhyinä. Tästä syystä kyseinen arkkitehtuuri tarvitsee vähemmän virtaa, mutta myös laskentateho on pienempi.

CISC, joka tulee sanoista Complex Instruction Set Computer, kykenee taas vastaanottamaan monimutkaisiakin käskyjä, mutta usein nykyään monet suorittimet purkavat monimutkaisia käskyjonoja pienempiin RSIC-tyylisiin osiin, jotta ne voidaan suorittaa nopeasti.

### 3 RASPBERRY PIN HISTORIA

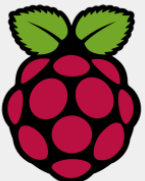
#### 3.1 Alkuperä

Raspberry Pin on kehittänyt Britannialainen yritys Raspberry Pi Foundation. Raspberry Pi Foundationin on tarkoituksena olla voittoa tavoittelematon yritys, joka tarjoaa ihmisille ympäri maailmaa hyvät työkalut tietokoneiden maailmaan.

Ensimmäinen Raspberry Pi tuli myyntiin helmikuussa 2012 ja sen ensimmäinen erä myytiin loppuun vain minuuteissa, mikä kaatoi sivustot joilla sitä myytiin. (The Guardian 2012)

Raspberry Pin tarkoituksena on olla nuorille halpa ja helppo tapa oppia, miten tietokoneet toimivat ja mitä kaikkea niillä voi tehdä. Raspberry Foundation kertoo sivuillaan, että 1980-luvulla ihmisten oli opittava ohjelmoimaan tietokoneillaan, jotta he voisivat käyttää niitä ja tästä syystä useat näistä ihmisistä ovat juurikin nyt myös ohjelmoijia. Koska tietokoneiden peruskäyttäminen on nykyään niin helppoa, ei ihmisten tarvitse opetella ohjelmoimaan ollenkaan käyttääkseen tietokoneita, vaikka nyt tarvittaisiin yhä enemmän kehittäjiä juuri siitä syystä, että tietokoneita on niin paljon. Tämän ongelman poistamiseksi luotiin Raspberry Pi. (Raspberry Pi Foundation n.d.b)

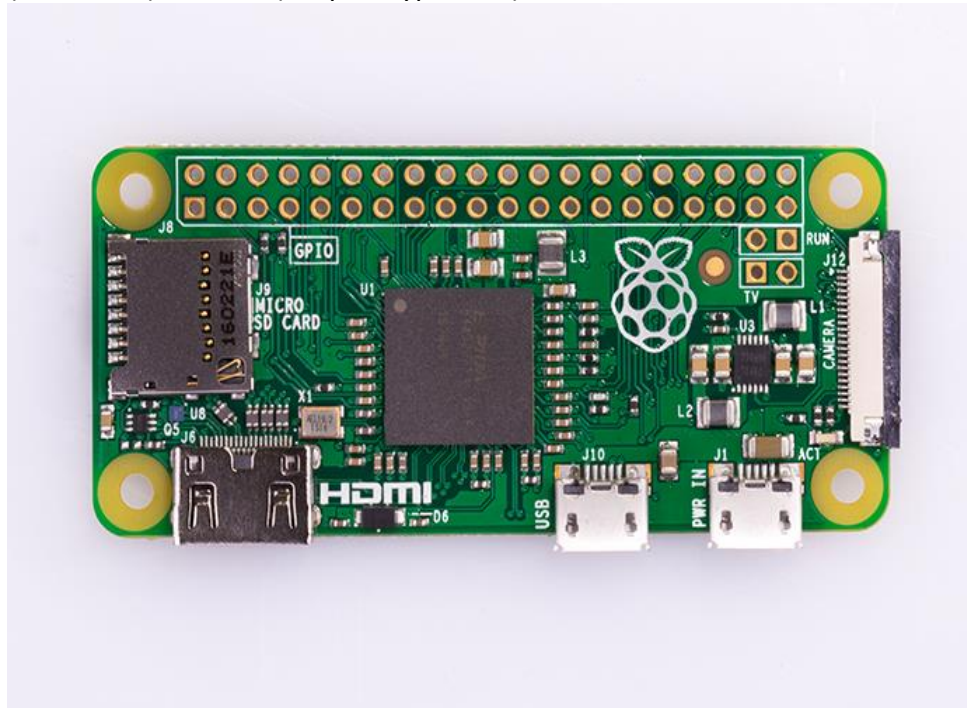
#### 3.2 Kehityshistoria

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	\$35	\$5	\$35	\$35

Kuva 4. Suurimmat erot eri Raspberry Pi versioiden välillä. (Hackaday 2016.)

Ensimmäinen myyntiin valmistettu Raspberry Pi oli Model B, ja pian tämän jälkeen valmistettiin Model A, joka oli halvempi ja tehottomampi versio alkuperäisestä Raspberrystä. Raspberry Pi Zeroa lukuun ottamatta Raspberry Pi Foundation on käyttänyt samankaltaista julkaisukaavaa, ensiksi tehdään versio B ja tämän jälkeen hiukan tehottomampi, mutta halvempi A-versio.

Raspberry Pi Zeron eroaa muista Raspberrystä myös kokonsa puolesta. Se on kooltaan melkein puolet pienempi kuin alkuperäinen Raspberry Pi ja silti sitä tehokkaampi (tosin selkeästi tehottomampi kuin uudet Raspberryt). Raspberry Pi Zero on halvin Raspberry Pi, sillä sen suositushinta on vain viisi dollaria ja sitä annettiin jopa MagPi-lehden (numero 40) kylkiäisenä. Zero on tarkoitettu pienempiin projekteihin, jotka vaativat vain vähän virtaa, ja se olisi ollut tähänkin opinnäytetyöhön varmasti sopiva versio, mutta valitettavasti sitä ei suomalaisilta jälleenmyyjiltä saanut (11.1.2017) tilattua. (Raspberrypi 2015)



Kuva 5. Raspberry Pi Zero.

## 4 IOT JA RASPBERRY PI

### 4.1 Tietokoneesta IoT-laitteeksi

IoT, eli Internet of Things (esineiden internet), tarkoittaa erilaisten laitteiden laajentumista myös internetiin. Näitä laitteita voi perinteisesti sensroida, tutkia ja/tai ohjata internetin ylitse. Esimerkiksi kännykät ovat IoT-laitteita, mutta niin myös uudet kahvinkeitin, jos ne on liitetty internetiin.

Koska internetiin kytkettävien laitteiden hinta on koko ajan alentumassa, on IoT laitteita yhä enemmän, Gartner arvioi, että vuonna 2020 IoT laitteita on jo yli 26 miljardia. (Forbes 2014)

Tietokoneet perinteisesti eivät ole IoT-laitteita, mutta Raspberry Pin kanssa raja IoT-laitteena hämärtyy. Vaikka Raspberry Pi onkin itsessään täysi tietokone, on se niin pieni ja yleisesti sitä käytetään nimenomaan erilaisissa projekteissa niin, että siitä muovautuu IoT-laite. Tässä opinnäytetyössä siis luodaan eräänlainen IoT-laite, koska tarkoituksena on nimenomaan luoda laite, jota voidaan mitata ja tutkia verkon yli.

### 4.2 Usb-vaaka

Tässä opinnäytetyössä käytettävä Usb-vaaka on Dymo:n valmistama M10-mallinen kirjevaaka. Kyseinen M10-malli on tarkoitettu maksimissaan 4,5 kilogrammaa painavien tuotteiden mittaamiseen ja se saa tarvittavan virran kokonaan Usb-yhteytensä avulla. Kyseistä vaakaa on myös mahdollista käyttää pattereilla, mutta tässä opinnäytetyössä on Usb-yhteys koko ajan päällä, joten pattereiden käyttö on turhaa.

Kyseinen vaaka on valittu siitä syystä, että taustatyötä tehdessäni huomasin monien eri kehittäjien ehdottavan sellaista ja että se toimii Raspberry Pin kanssa lähestulkoon samantien.

Vaaka oli kohtuullisen hintava ja yhteensä se maksoi (toimituskulut mukaan lukien) 62,48 euroa Amazonista, mutta itse vaaka on todella laadukas ja painon mittaustapahtuu todella nopeasti.



Kuva 6. Dymo:n Usb-kirjevaaka ja Raspberry Pi 3 B.

#### 4.3 Raspberry ja modaaminen

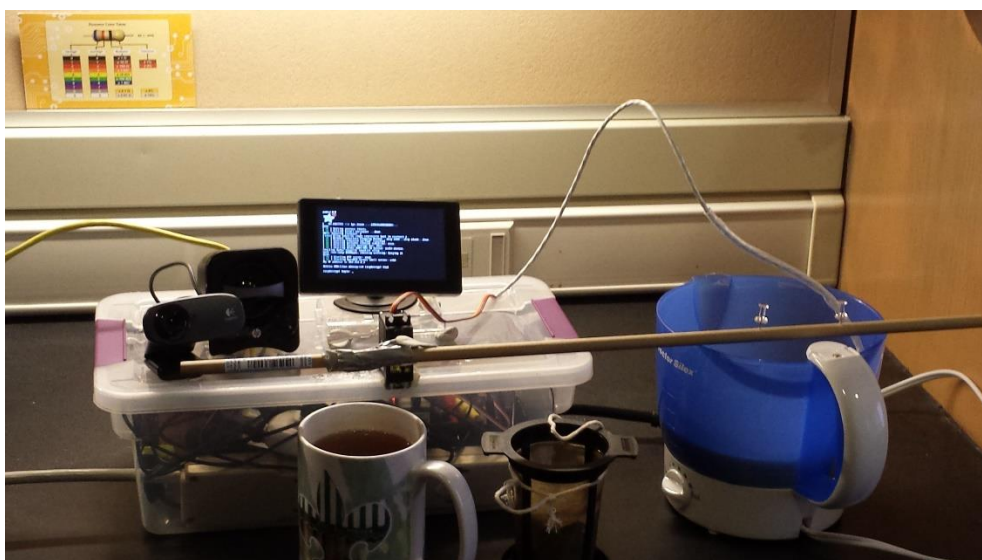
Raspberryllä on luotu monenlaisia mielenkiintoisia projekteja, niin suuria kuin pieniäkin. Tässä kappaleessa esitellään muutamia kekseliäitä projekteja jotka näyttävät, että Raspberry Pin kanssa vain mielikuvitus on rajana.



#### 4.3.1 TeaPi

TeaPi on James Pavurin kehittämä laite, jolla on tarkoitus keittää teetä mahdollisimman optimaalisilla lämpötiloilla ja käyttäen äänikomentoja. Laitteella on mahdollista ajoittaa teen valmistus käyttäen joko tekstiviesti-, internet- tai äänikomentoja. Projektin perimmäisenä ideana on luoda halpa ratkaisu laitteilla jotka saattavat löytyä jokaisen kodista. (Pavurin, James 2014)

Samankaltaisia teen valmistuslaitteita on myytävissä, mutta ne maksavat yleensä huomattavasti enemmän. Esimerkiksi James Pavur vertaa TeaPitä blogissaan BTM800XL-laitteeseen joka kirjoitushetkellä maksaa uutena yli 200 dollaria Amazonissa. Jos ottaa huomioon kaikki kustannukset jotka tulevat TeaPita tehdessä, tulee hinnaksi vain noin 70 dollaria. Tämä James Pavurin projekti pääsi semifinaaleihin vuoden 2014 Hackaday-kisoissa, joissa oli monenlaisia Raspberry Pillä tehtyjä projekteita.



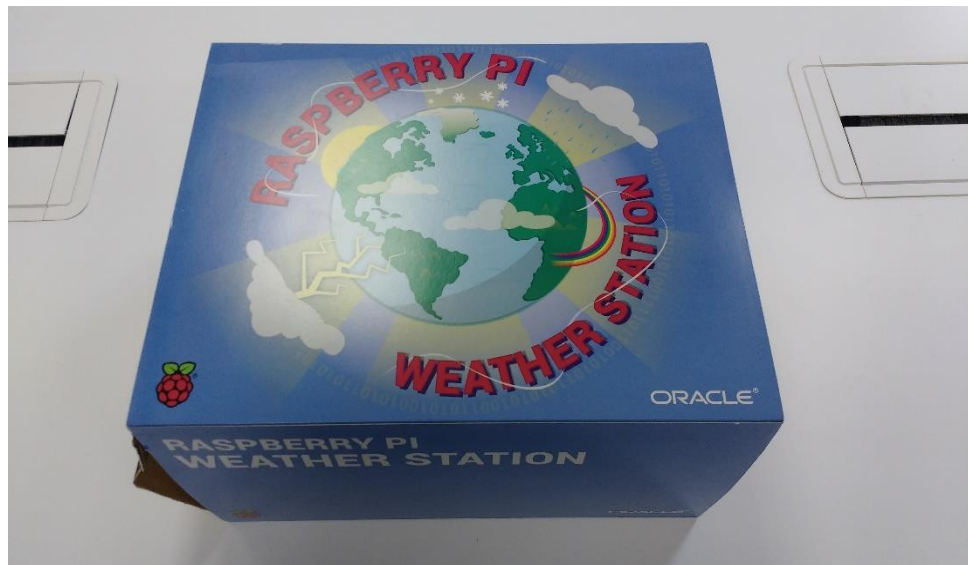
Kuva 7. James Pavurin TeaPi.

#### 4.3.2 Raspberry sääasemana

Raspberry Pistä voi myös luoda oman henkilökohtaisen sääaseman, jos niin haluaa. Raspberry Pi Foundation on myös luonut sääasemaksi optimoidun Raspberry Pin. Tätä Raspberry Pi Weather Station-nimistä pakettia ei voi vielä ostaa erikseen ja se on luotu lähinnä kouluille, mutta Raspberry Pi Foundation kertoo sivuillaan, että suunnitelmissa on tulevaisuudessa levit-



tää pakettia myös kaupallisesti. Tämä virallinen Raspberry Pi Weather Station sisältää kaikki tarvittavat sensorit sekä sään kestävän boksen johon Raspberry Pi liitetään.



Kuva 8. Virallinen Raspberry Pi Weather Station.

On myös mahdollista luoda täysin oma sääasemansa ilman virallisia sensoreita ja pakkauksia. Tätä varten on luotu [raspberrypiweather.com](http://raspberrypiweather.com)-niminen nettisivusto, joka sisältää yksityiskohtaisia ohjeita oman sääaseman luomiseen. Sivusto auttaa alusta loppuun prosessin luomisessa ja valmiin sääaseman tuloksia voi lopulta lukea WordPressillä tehdyiltä sivustoilta. Sivustolla on listattuna kaikki tarvittavat laitteet ja lopulta materiaalien hinnaksi tulee noin 100 dollaria (mukaan lukien Raspberry Pi).

## 5 KEHITTÄMINEN RASPBERRY PILLÄ

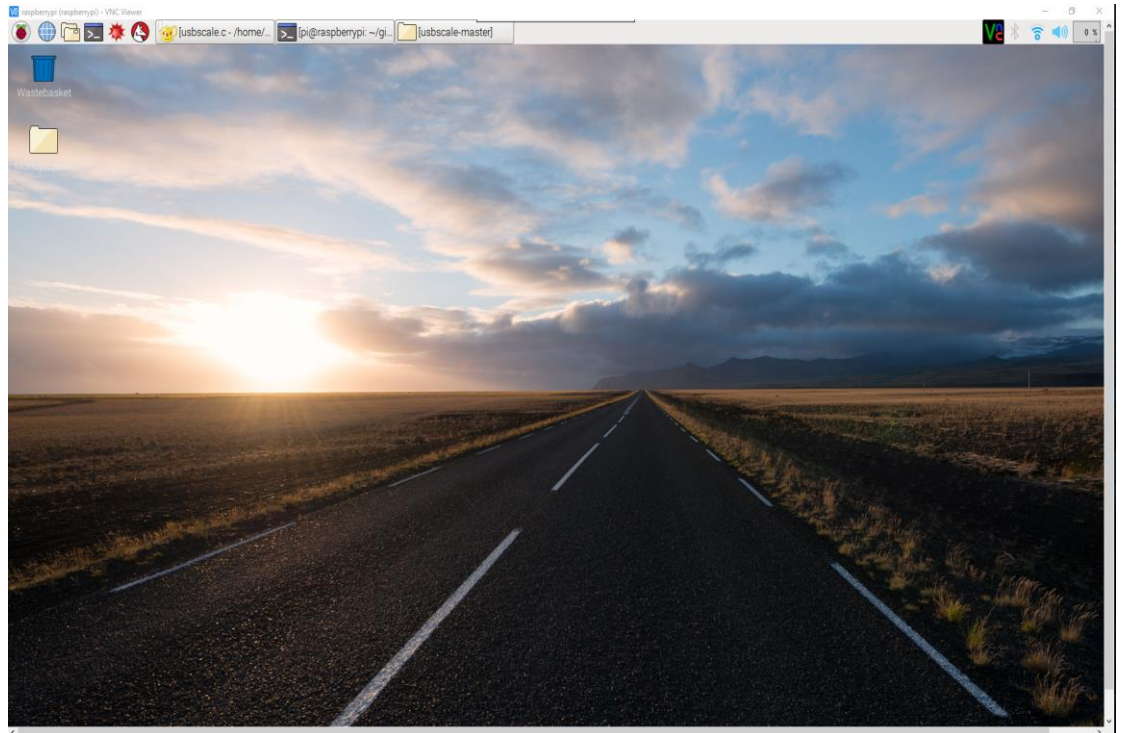
Ennen varsinaista kehittämistä on Raspberry Pihin asennettava jonkinlainen käyttöjärjestelmä. Käyttöjärjestelmän valintaan vaikuttavat monet eri asiat, kuten esimerkiksi millä kielellä käyttäjä haluaa kehittää ja tai vaikkapa haluaako käyttäjä käyttää Windows- vai Linux-pohjaista käyttöjärjestelmää.

Tässä opinnäytetyössä käytetään käyttöjärjestelmänä Raspberry Pi Foundationin sivuilta ladattua Rasbiania, joka pohjautuu Debian-nimiseen Linux-käyttöjärjestelmään. Kyseinen käyttöjärjestelmä on hyvin samantyylinen kuin muutkin Linux-käyttöjärjestelmät ja sen käyttäminen ei eroa ”normaaleista” Linuxeista millään tavoin.

### 5.1 Etäohjaus ja -hallinta

Koska lopullisen työn kannalta on mielekästä, että Raspberry Pi toimii lähestulkoon itsenäisesti, niin on tärkeää jo kehitysvaiheessa selvittää miten Raspberryä voi käyttää ilman ylimääräisiä piuhoja. Tähän ongelmaan on luotu VNC-niminen tekniikka. VNC (eli Virtual Network Computing) on protokolla, jolla voidaan käyttää tietokoneen graafista käyttöliittymää etäyhteyden avulla. Tämän tekniikan avulla Raspberryä voi käyttää toisella tietokoneella. Raspberry Pin tulee olla vain päällä ja kytkettynä lähiverkkoon. Itse Rasbianin mukana tulee RealVNC-niminen ohjelma, jota myös käytettiin tässä opinnäytetyössä. Kyseinen ohjelma ei ole valmiiksi kuitenkaan laitettu käyttövalmiiksi, vaan se tulee laittaa päälle erikseen. Tämä johtuu siitä, että VNC-tekniikka antaa käyttäjälle kaikki valtuudet tehdä tietokoneella mitä haluaa ja on täten kohtuullinen tietoturvariski, jos sitä käytävällä ei ole tarvittavia tietotaitoja tietoturva-aukkojen täyttämiseen.

RealVNC suojaa kylläkin yhteyden automaattisesti ja vaatii jonkinlaisen salasana ja käyttäjänimen ennen yhteyden luomista. Tässä opinnäytetyössä on jätetty käyttäjänimet ja salasanat täysin vakioiksi siitäkin syystä, että Raspberryn oma toimialue on varsin itsenäinen ja kehitysvaiheen loputtua VNC-yhteyttä ei ole tarkoituskaan pitää enää päällä.



Kuva 9. VNC-viewerin näkymä Raspberry Pin Rasbian-käyttöjärjestelmän työpöydästä.

## 5.2 Bash-skriptaus ja automaattinen ajaminen

Bash-skriptaus on tapa antaa tietokoneelle erilaisia käskyjä, kuten esimerkiksi: "Aja tämä ohjelma". Tällaisia skriptejä käytetään hyvin yleisesti Linux-maailmassa helpottamaan erilaisia työtehtäviä. Linuxia käytettäessä ollaan usein tekemisissä komentorivin kanssa, ja kaikki komentoriville kirjoitettavat käskyt voidaan antaa myös skriptin kautta. Tällä tavoin voi säästää aikaa ja vaivaa verrattuna siihen, että joutuisi kirjoittamaan jokaisen käskyn yksitellen komentoriville. Bash-skriptaaminen itsessään on hyvin lähellä oikeata koodaamista, sillä on oma syntaksi ja sillä voi tehdä useita samankaltaisia asioita kuin koodaamalla. (Chadwick, Ryan (2017))

Tässä opinnäytetyössä käytetään Bash-skriptiä ajamaan Usbscale-ohjelmaa loputtomiin tietyn väliajoin. Varsinainen skripti on hyvin yksinkertainen, mutta samalla myös tehokas.

```

pi@raspberrypi: ~/bin
File Edit Tabs Help
GNU nano 2.2.6 File: script_auto_run Modified
#!/bin/bash
# Talla skriptillä aloitetaan ohjelma ja sita ajetaan loputtomasti
# tietyin valiajoin.
echo "Aloitetaan automaattista ajoa..."
while true
do
sudo /home/pi/git/USB/usbscale-master/usbscale &
sleep 5
done
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Kuva 10. Opinnäytetyössä käytettävä Bash-skripti.

Skriptin ensimmäisellä rivillä oleva `#!/bin/bash` kertoo, että skripti suoritetaan käyttäen bashia riippumatta siitä, mitä komentotulkkia käyttäjä käyttää ajaessaan skriptin. Vaikka kyseinen rivi alkaakin #-merkillä niin se ei ole kommentti. Myöhemmin käytettäessä #-merkkiä nämä rivit sivuutetaan ja niihin voi kommentoida skriptin toimintaa.

Neljännellä rivillä oleva `echo` tulostaa konsoliin rivin, joka on kirjoitettu lainausmerkkien sisään. Tämän jälkeen käytettävä `while true` kertoo bashille että tätä suoritetaan niin kauan kuin ohjelma on true, eli käytännössä loputtomiin, koska kyseistä määrettä ei muuteta skriptin aikana ollenkaan. Tämän jälkeen käsketään bashia vain suorittamaan pääkäyttäjän oikeuksilla kuvanmukaisessa polussa oleva ohjelma lisäten perään &-merkki. Tämä antaa Raspberyllle tiedon siitä, että tämä ohjelma voidaan suorittaa omassa prosessissaan, joilloin ei tule ongelmia esimerkiksi silloin kun Raspberry käynnistetään.

```

@reboot cd /home/pi/bin/ && sh script_auto_run
pi@raspberrypi:~ $

```

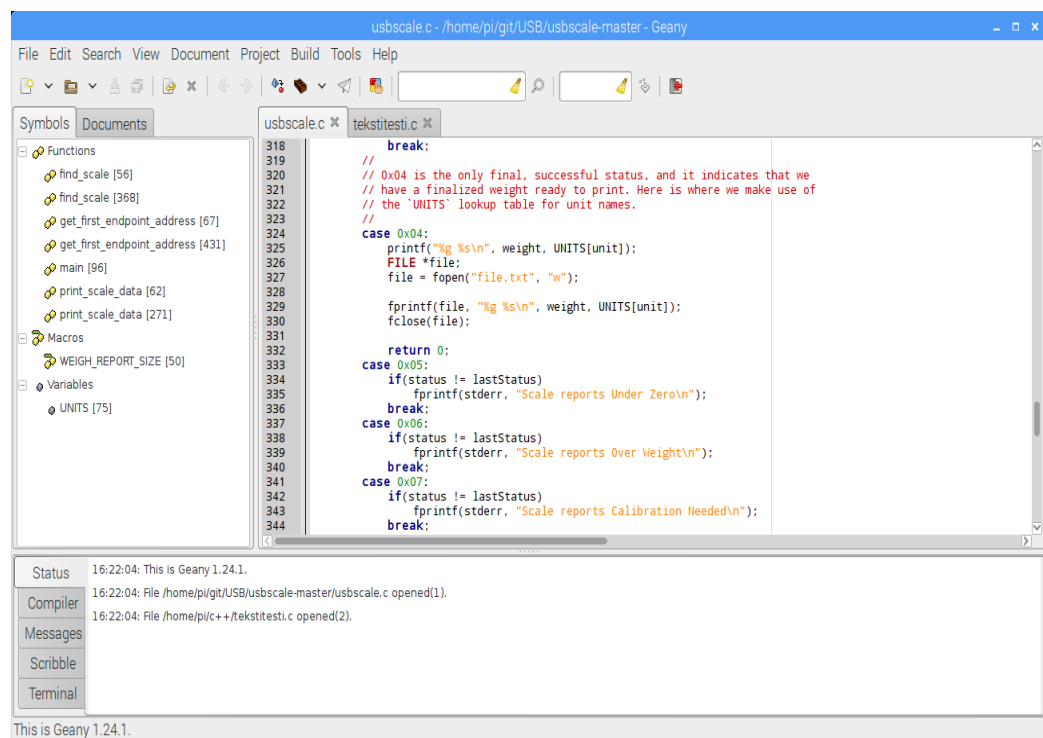
### 5.3 C-kieli ja Geany

Tässä opinnäytetyössä kehitettävä ohjelma on koodattu alun perin C-kielellä, ja vaikka alun perin oli tarkoitus kirjoittaa ohjelma C#-kielellä, niin päätin vain jatkokehittää tätä GPLv3-lisenssin (joka tarkoittaa sitä että, ohjelmaa saa käyttää, muuttaa ja jakaa kunhan lopullinen koodi on myös avoimena ja saatavilla kaikille) omaavaa Usbscale-nimistä ohjelmaa.

Kielenä C on jo vanha, alun perin se luotiin jo 1970-luvulla pääosin UNIX-käyttöjärjestelmiä varten, mutta sen verrattaisen yksinkertaisuuden vuoksi se on ollut suosittu kieli jo monien vuosien ajan.

C-kielen vahvuuksina on määritelty yleensä sen siirrettävyys ja laiteläheisyys. Siirrettävyydellä tarkoitetaan sitä, että C-kielellä kirjoitetut ohjelmat toimivat usein koodia paljoa muuttamatta (jos ollenkaan) myös toisissa laitteissa. Tämä on saavutettu sillä tavalla, että C:n koneriippuvat piirteet on eristetty kirjastofunktoihin ja nämä kirjastofunktiot ovat C-järjestelmään kuuluvia valmiita aliohjelmia, jotka toteutetaan kussakin järjestelmässä laitteiston edellyttämällä tavalla, mutta ovat ohjelmoijan näkökulmasta aina samantapaisia. (Simo Silander 1999)

Geany on kevytrakenteinen ohjelmointiympäristö (IDE, Integrated Development Environment), joka toimii varsin hyvin Raspberry Pillä. Se tukee monia erilaisia kieliä, mutta pääosin se on tarkoitettu C-kielen koodaamiseen. Geany poistaa tarpeen Raspberry Pissä kääntää C-kielen ohjelmat erikseen komentorivillä, joka tekee siitä varsin hyvän ohjelmointiympäristön. Geany tulee valmiiksi asennettuna normaalin Rasbianin yhteydessä.



Kuva 11. Geany:n kehittäisympäristö.

## 5.4 C-kielisten ohjelmien ajaminen Raspberyllä

C-kielillä ajettavat ohjelmat voidaan ajaa Raspberry Pillä parilla eri tapaa. Ennen varsinaista ajoa on kuitenkin koodi käännettävä (compile) ja tämä yleensä suoritetaan makefile-nimisen (tee, luo) tiedoston välityksellä. Makefile-tiedosto kääntää ohjelman lähdekoodin, automatisoiden prosessin niin, ettei sitä tarvitse ohjelmoijan itse enää yksitellen tehdä. Nämä makefile-tiedostot ajetaan Make-nimisellä työkalulla joka sitten kääntää lähdekoodin, eikä käyttäjän itse tarvitse tuntea tätä prosessia kovin tarkoin. (Linux 2016)

Kääntämisen jälkeen ohjelmat voi ajaa samasta kansioista missä ne on käännetty, Raspberry Pillä C-kielillä ohjelmoidut ohjelmat ajetaan komentorivillä alla olevan kuvan tavoin.

```

pi@raspberrypi: ~/git/USB/usbscale-master
File Edit Tabs Help
pi@raspberrypi:~/git/USB/usbscale-master $ ls -li
total 88
139165 -rwxrwxrwx 1 pi pi 711 Feb 19 18:19 50-usb-scale.rules
129213 -rw-rw-rw- 1 pi pi 35147 Feb 19 18:19 COPYING
140930 -rw-r--r-- 1 pi pi 5 Mar 5 16:42 file.txt
140807 -rw-rw-rw- 1 pi pi 1654 Feb 19 18:19 lsusb.c
140808 -rw-rw-rw- 1 pi pi 220 Feb 19 18:19 Makefile
140809 -rw-rw-rw- 1 pi pi 2249 Feb 19 18:19 README.md
140810 -rw-rw-rw- 1 pi pi 1103 Feb 19 18:19 scales.h
140909 -rwxr-xr-x 1 pi pi 9864 Feb 22 16:43 usbscale
140932 -rw-rw-rw- 1 pi pi 13737 Feb 22 16:43 usbscale.c
pi@raspberrypi:~/git/USB/usbscale-master $ ./usbscale

```

Kuva 12. C-kielisen ohjelman ajaminen Raspberry Pillä.

Kuten kuvassa näkyy, aluksi on otettu kansion tiedostoista selvää Linuxille perinteisellä `ls -li` komennolla, joka listaa kansiossa kaikki olevat tiedostot. Tiedostokansiossa on ajettu `Makefile`-niminen tiedosto `Make`-työkalun kanssa läpi, kääntäen tämän ja tämä on luonut kansioon `usbscale`-nimisen tiedoston joka näkyy myös listauksessa vihreällä. Lopuksi ohjelman ajaminen tapahtuukin vain kirjoittamalla `./usbscale`, jolloin ohjelma käynnistyy ja se ajetaan.

Ohjelmia voi ajaa myös suoraan Geanyssä, jolloin ei tarvitse käyttää komentoriviä ollenkaan ja tämä tapahtuu joko painamalla `F5`-näppäintä tai menemällä valikosta `Build`-valikkoon ja sieltä painamalla `Execute` painiketta. On kuitenkin huomioitava se, että ohjelmat on käännettävä ennen kuin niitä voi ajaa Geanyyn kautta, eli `Make`-komento on käytävä kuitenkin läpi ennen kuin ohjelmaa voi ajaa. Tämä on myös tehtävä joka kerta kun koodia muutellaan, muuten (komentorivillä) ohjelma ei päivity ja se ajaa vain vanhan version ja jostain syystä Geany ei pysty tätä kääntämistä (ainakaan tässä opinnäytetyössä käytettyä ohjelmaa) suorittamaan vaan antaa aina jonkinlaista virhekoodia niin kauan, kunnes `Make`-komento on annettu uudestaan (tämän jälkeen ohjelmaa voi myös ajaa normaalisti Geanyssä).

## 5.5 Usbscale ohjelma

Usbscale on tämän opinnäytetyön ohjelma jota jatkokehitettiin tämän opinnäytetyön tarkoituksia vastaamaan. Se käyttää hyväkseen libusb-nimestä avoimen lähdekoodin ohjelmaa, joka antaa kehittäjälle mahdollisuuden käyttää Raspberry Pin USB-laitteita. Usbscale-ohjelman on alun perin luonut Eric Jiang ja tämä ohjelma on myös avoimen lähdekoodin ohjelma ja se on lisensoitu GPLv3-lisenssillä, joka tarkoittaa sitä, että jos alkupe-  
räistä koodia muuttaa, niin senkin täytyy olla myös avointa lähdekoodia.

Koodissa on käytetty hyväksi paljon switch- ja case-rakenteita ja näitä rakenteita käytetään ottamalla USB-vaa'asta erilaisia status-koodeja. Esi-  
merkiksi lopullinen tulostus tapahtuu alla olevan koodiesimerkin tavoin.

```
switch(status) {
    case 0x01:
        fprintf(stderr, "Scale reports Fault\n");
        return -1;
    case 0x02:
        if(status != lastStatus)
            fprintf(stderr, "Scale is zero'd...\n");
        break;
    case 0x03:
        if(status != lastStatus)
            fprintf(stderr, "Weighing...\n");
        break;
    //
    // 0x04 is the only final, successful status, and it indicates that we
    // have a finalized weight ready to print. Here is where we make use of
    // the `UNITS` lookup table for unit names.
    //
    case 0x04:
        printf("%g %s\n", weight, UNITS[unit]);
        FILE *file;
        file = fopen("file.txt", "w");

        fprintf(file, "%g %s\n", weight, UNITS[unit]);
        fclose(file);

        return 0;
```

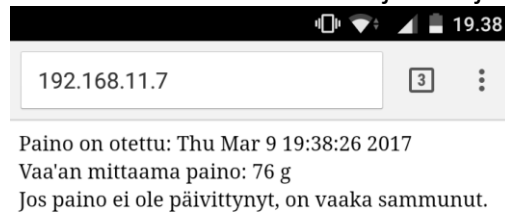
Kuva 13. Koodiesimerkki switch- ja case-rakenteesta jolla tulostuskin on tehty.

Kuten Eric Jiang on koodiin kommentoinut, paino tulostetaan ainoastaan, jos vaaka palauttaa statuskoodin 0x04 ja tämän jälkeen vaa'alta saatu arvo tulostetaan konsoliin. Sama arvo merkitään myös tekstitiedostoon, jotta tämän tekstitiedoston saa sitten edelleen jaettua Apachen avulla niin, että sen voi lukea.

Eric Jiangin kommentissa on myös mainittu UNITS-niminen lista, joka todellakin on vain lista, missä on lueteltuna 13 erilaista painoyksikköä, ja tästä listasta taas saadaan tulostuksessa tarvittava painon mittayksikkö.



Lopullinen jatkokehitetty ohjelma sisältää myös mobiililla luettavan käyttöliittymän, jossa saa selville monia relevantteja tietoja.



Kuva 14. Lopullisen ohjelman käyttöliittymä mobiililla.

## 6 YHTEENVETO

Ennen tätä opinnäytetyötä en ollut henkilökohtaisesti koodannut C:llä mitään, joten opeteltavaa oli paljon, mutta itse koodia ei onneksi tarvinnut hirveästi tuottaa, sillä pohjana toiminut usbscale-ohjelma toimi erinomaisen hyvin. Pitkin opinnäytetyöprosessia C-kielestäkin tuli tuttu ja vaikka kyseinen kieli on jo vanha, niin sitä käytetään silti paljon ja sen osaaminen ei varmasti ole haitaksi.

Itse ohjelmistopuolella tehtävää ei ollutkaan liian paljon, vaan opinnäytetyössä käytetty Usbscale-ohjelma pienellä jatkokehityksellä osoittautui erittäin käytännölliseksi.

Itse vaaka on kuitenkin loppujen lopuksi mielestäni liian iso ollakseen kiitettävä lopputuloksen kannalta. Laitteen on tarkoitus olla jääkaapissa mittaamassa (esimerkiksi) maidon painoa, mutta tähän tarkoitukseen laite on liian iso (ellei loppukäyttäjällä ole isokokoista jääkaappia). Jälkeenpäin ajateltuna olisi siis ollut järkevämpää joko käyttää GPIO-pinnejä ja painosensoreita painon mittaamiseen, jolloin olisi saanut lopputuloksesta pienemmän laitteen, tai sitten hankkia jokin pienempi USB-vaaka (tosin valikoima ei valitettavasti ole kovin iso ja toimivuus Raspberry Pin kanssa olisi myös ollut iso kysymysmerkki).

## LÄHTEET

Chadwick Ryan (2017) What is a Bash Script? Viitattu 7.3.2017 osoitteesta <http://ryanstutorials.net/bash-scripting-tutorial/bash-script.php>

Forbes (2014). A Simple Explanation Of 'The Internet Of Things'. Viitattu 26.1.2017 osoitteesta <http://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#2a891e706828>

The Guardian (2012). Demand for Raspberry Pi, the British £22 computer, crashes website. Viitattu 22.1.2017 osoitteesta <https://www.theguardian.com/technology/2012/feb/29/raspberry-pi-computer-sale-british>

Hackaday (2016). Introducing the Raspberry Pi 3. Viitattu 5.2.2017 osoitteesta <http://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/>

Linux (2016). Make. Viitattu 30.2.2017 osoitteesta <https://www.linux.fi/wiki/Make>

Microsoft (n.d). Learn about Windows 10 IoT Core. Viitattu 5.2.2017 osoitteesta <https://developer.microsoft.com/fi-fi/windows/iot/Explore/IoTCore>

Pavurin, James (2014). Tea-Pi. Viitattu 29.1.2017 osoitteesta <https://hackaday.io/project/156-tea-pi>

Raspberry Pi Foundation (n.d.a). About us. Viitattu 22.1.2017 osoitteesta <https://www.raspberrypi.org/about/>

Raspberry Pi Foundation (n.d.b). Getting Started with your Raspberry Pi. Viitattu 19.1.2017 osoitteesta <https://www.raspberrypi.org/help/videos/#getting-started-with-raspberry-pi>

Raspberry Pi Foundation (n.d.c). Raspberry Pi 3 Model B. Viitattu 19.1.2017 osoitteesta <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Raspberrypi (2015). Issue 40. Viitattu 22.2.2017 osoitteesta <https://www.raspberrypi.org/magpi/issues/40/>

Silander, Simo (1999). Ohjelmoinnin perusteet ja C-kieli. Viitattu 8.2.2017 osoitteesta [http://cs.stadia.fi/~silas/ohjelmointi/c\\_opas](http://cs.stadia.fi/~silas/ohjelmointi/c_opas)

