

Petri Niemi

Windows 10 IoT Coren käyttö Raspberry Pi -laitteessa

Opinnäytetyö

Kevät 2016

SeAMK Tekniikka

Tietotekniikan Tutkinto-ohjelma



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikanyksikkö

Tutkinto-ohjelma: Tietotekniikka

Suuntautumisvaihtoehto: Ohjelmistotekniikka

Tekijä: Niemi Petri

Työn nimi: Windows 10 IoT Coren käyttö Raspberry Pi -laitteessa

Ohjaaja: Mäkelä Petteri

Vuosi: 2016

Sivumäärä: 40

Liitteiden lukumäärä: 4

Opinnäytetyön tavoite oli tutkia Raspberry Pi 2 Model B -laitteen toimintaa Windows 10 IoT Core -käyttöjärjestelmän kanssa. Työssä on toteutettu kodinvalvontajärjestelmä käyttäen uutta Windows 10 IoT Core -käyttöjärjestelmää Raspberry Pi 2 Model B -laitteella. Ohjelmisto koostuu Raspberry Pi 2 Model B -laitteessa olevasta kuvaa keräävästä ja lähettävästä ohjelmasta sekä palvelinohjelmasta, joka tallentaa tiedon tietokantaan ja näyttää käyttäjälle tallennetut tiedot. Palvelinohjelma toteutettiin käyttäen PHP-ohjelmointikieltä, joka toimii Windows-tietokoneella.

Avainsanat: Windows 10 IoT Core, Raspberry Pi, PHP

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Software Engineering

Author: Niemi Petri

Title of thesis: The usage of Windows 10 IoT Core on a Raspberry Pi device

Supervisor: Mäkelä Petteri

Year: 2016

Number of pages: 40

Number of appendices: 4

The goal of this thesis was to study the usage of Raspberry Pi 2 Model B device with the Windows 10 IoT Core operating system. The work implemented a home security system using the new Windows 10 IoT Core operating system on a Raspberry Pi 2 Model B device. The software consists of a Raspberry Pi 2 Model B program that takes and sends pictures and a web server that saves the received pictures in to a database and displays them to a user. The server program was implemented using the PHP programming language that runs on a Windows pc.

Keywords: Windows 10 IoT Core, Raspberry Pi, PHP

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	1
Thesis abstract.....	2
SISÄLTÖ.....	3
Kuva- ja taulukkoluetelo	5
Käytetyt termit ja lyhenteet	7
1 Johdanto.....	8
1.1 Työn tausta	8
1.2 Työn tavoitteet.....	8
1.3 Työn rakenne	8
2 Raspberry Pi	10
2.1 Raspberry Piin historia	10
2.2 Raspberry Pi laitteena.....	11
2.2.1 ARM.....	13
2.2.2 Raspberry Pi 2 Model B	14
2.2.3 Raspberry Pi 3 Model B	14
2.3 Vaihtoehtoisia laitteita Raspberry Piille	15
2.3.1 Dragonboard 410c	15
2.3.2 Arduino/Genuino UNO	16
2.4 Eri laitteiden vertailu.....	18
2.5 Raspberry Pi -käyttöjärjestelmät	19
3 Windows 10 IoT Core.....	21
3.1 Windows 10 IoT Core.....	21
3.2 Windows 10 IoT Coren ominaisuudet	21
3.3 Windows 10 Universal Windows Platform.....	21
3.4 Windows 10 IoT Coren asennus	22
3.4.1 Windowsin omilla ohjeilla	22
3.4.2 Asennus NOOB-ohjelman avulla	25
3.4.3 Visual Studion asennus ja valmiiksi laitto.....	27
4 Palvelin	29
4.1 Apache.....	29

4.2 XAMPP	29
4.3 PHP.....	29
4.3.1 PHP-kielen historia	29
4.3.2 PHP-kielen käyttötarkoitukset	30
5 Windows 10 IoT Core kodinvalvonta	31
5.1 Windows 10 IoT Core -ohjelma.....	31
5.2 PHP-palvelin ja MYSQL-palvelin.....	34
6 TYÖN TULOKSET	37
6.1 Tulokset	37
6.2 Yhteenveto.....	37
LÄHTEET	38
LIITTEET	40

Kuva- ja taulukkoluetelo

Kuva 1. Raspberry Pi 2 Model B.....	11
Kuva 2. Raspberry Pi A+, B+ ja Raspberry pi 2 GPIO-pinnit	12
Kuva 3. Raspberry Pi 3 -laitteen erot edeltäjiinsä.	15
Kuva 4. Dragonboard 410c	16
Kuva 5. Arduino UNO	17
Kuva 6. Windows 10 IoT Core Set up a new device -painike.....	23
Kuva 7. Laitteen, käyttöjärjestelmän ja aseman valinta	23
Kuva 8. My Devices -painike.....	24
Kuva 9. My devices -valikko ja Settings-painike.....	25
Kuva 10. NOOBS-käyttöjärjestelmän valinta.....	26
Kuva 11. Käyttöjärjestelmäversion valinta	27
Kuva 12. Visual Studio -versioiden tarkistus	28
Kuva 13. Windows IoT Core Project Templaten asennus.....	28
Kuva 14. Ohjelman perusnäkyä.....	32
Kuva 15. Platform target	32
Kuva 16. Ohjelmassa tarvittavat toiminnot.....	33
Kuva 17. Oman ohjelman käynnistys ja oletusohjelmaksi asetus Windows 10 IoT Core Dashboardin avulla	34
Kuva 18. Palvelimen näyttämät kuvat.....	36

Taulukko 1. Laitevertailu	18
---------------------------------	----

Käytetyt termit ja lyhenteet

API	Application programming interfrace. Ohjelmointirajapinta
UWP	Universal Windows Platform. Ohjelmointialusta, jonka avulla voi tehdä ohjelmia usealla Windows-laitteelle teke-mättä koodia uudelleen.
GPIO	General Purpose I/O. Yleinen pinni piirilevyissä. Sen kautta voi kirjoittaa ja lukea dataa.
NOOBS	New Out Of the Box Software. Helppokäyttöinen käyttöjär-jestelmän asennustyökalu Raspberry Pi -laitteille.
BLOB	Binary Large Object. Kokoelma tietokantaan tallennettua binääridataa.
POST	Tapa lähettää data selaimesta palvelimelle.
CGI	Common Gateway Interface. Standardi, jolla selain lähet-tää dataa palvelinohjelmalle.

1 Johdanto

1.1 Työn tausta

Raspberry Pi on luottokortin kokoinen yhden piirilevyn tietokone, joka mahdollista erilaisten antureiden ja laitteiden kytkemisen. Raspberry Pi on kehittyessään saanut useita mielenkiintoisia ominaisuuksia. Viimeisimpänä näistä on Windows 10 IoT Core -käyttöjärjestelmä. Windows 10 IoT Core mahdollistaa Microsoftin tekniikoiden ja Universal Windows Appien ajon Raspberry Pi -laitteella.

Tässä opinnäytetyössä keskitytään lisälaitteelta tulleen tiedon tallentamiseen Raspberry Pi -laitteelle, ja tiedon lähettämiseen palvelimelle käyttäen C#-ohjelmointikielen perustuvia tekniikoita. Dataa voidaan tarkastella mistä tahansa selaimella varustetusta laitteesta

1.2 Työn tavoitteet

Opinnäytetyön tavoitteena on rakentaa yksinkertainen kodinvalvontajärjestelmä, jossa Raspberry Pi 2 Model B -laitteeseen kytketyn kameran ottamia kuvia lähetetään internetin välityksellä palvelimelle. Palvelimella olevia kuvia pääsee katsomaan internetselaimen avulla mistä tahansa. Palvelin ylläpitää halutun määrän tuoreimpia kuvia muistissa, minkä jälkeen nämä hävitetään.

Toisena tavoitteena on testata Windows 10 IoT Coren käyttöä Raspberry Pi 2 Model B -laitteella.

1.3 Työn rakenne

Luvussa 2 käsitellään Raspberry Pi -laitteen historiaa, Raspberry Pi -laitteen käyttämää ARM-prosessoria ja uusimmat Raspberry Pi -laitteet. Luvussa myös käsitellään vaihtoehtoisia laitteita Raspberry Pi -laitteelle ja vertaillaan näiden välisiä eroja. Lisäksi luvussa käsitellään Raspberry Pi -käyttöjärjestelmiä.

Luvussa 3 käsitellään Windows 10 IoT Core -käyttöjärjestelmää, sen ominaisuuksia ja käydään läpi järjestelmän asennus Raspberry Pi -laitteelle ja kehitysympäristön valmistelu tietokoneelle.

Luvussa 4 käsitellään työssä käytettyä palvelinta lyhyesti käyden läpi käytetyt teknologiat.

Luvussa 5 käydään läpi itse kodinvalvonnan toteutus ja palvelimen toteutus.

Luvussa 6 esitellään työn tulokset ja yhteenveto.

2 Raspberry Pi

Raspberry Pi on halpa, pankkikortin kokoinen tietokone, jonka voi liittää tietokoneen näyttöön tai televisioon HDMI-liittimellä. Laite pystyy pienestä koostaan huolimatta tekemään kaikkea mitä tavallinen tietokonekin kykenee, kuten selaamaan internettiä, katsomaan HD-videoita, pelaamaan pelejä ja editoimaan tekstiä. Laiteella kaiken ikäiset voivat opetella tietojenkäsittelyä ja ohjelmointia. (Raspberry Pi Foundation [Viitattu 1.2.2016].)

2.1 Raspberry Piin historia

Raspberry Piin idea tuli vuonna 2006, kun Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft Cambridgen yliopistossa huolestuivat vuosittaisesta kadosta tietojenkäsittelyyn hakeneiden opiskelijoiden määrässä ja taidoissa. Vielä vuonna 1990 suurin osa hakeneista opiskelijoista oli harrastelijaohjelmoijia, kun taas 2000-luvulla useimmat hakijat olivat tehneet vain vähän web-suunnittelua. (Raspberry Pi Foundation [Viitattu 1.2.2016].)

Vuosina 2006–2008 he suunnittelivat useita versioita laitteesta, josta tuli Raspberry Pi. Ideana oli valmistaa laite, joka on heti valmis ohjelmoitavaksi. Esimerkkinä olivat vanhat laitteet, joilla 1990 opiskelleet oppivat ohjelmoimaan, kuten Amiga ja Commodore 64. (Raspberry Pi Foundation [Viitattu 1.2.2016].)

Vuoteen 2008 mennessä mobiililaitteille suunnitellut prosessorit olivat tulossa edullisemmiksi ja tarpeeksi tehokkaiksi tarjotakseen hyvän multimedian, se kiinnostaisi ennemmin ohjelmointia välttäneitä lapsia. Tällöin Eben, Rob, Jack ja Alan alkoivat tehdä yhteistyötä Pete Lomasin ja David Bradenin kanssa. He perustivat yhdessä Raspberry Foundationin. Kolme vuotta myöhemmin Raspberry Pi Model B saapui massatuotantoon. Kahdessa vuodessa Raspberry Pi on myynyt yli kaksi miljoonaa kappaletta Raspberry Pi -laitteita. (Raspberry Pi Foundation [Viitattu 1.2.2016].)

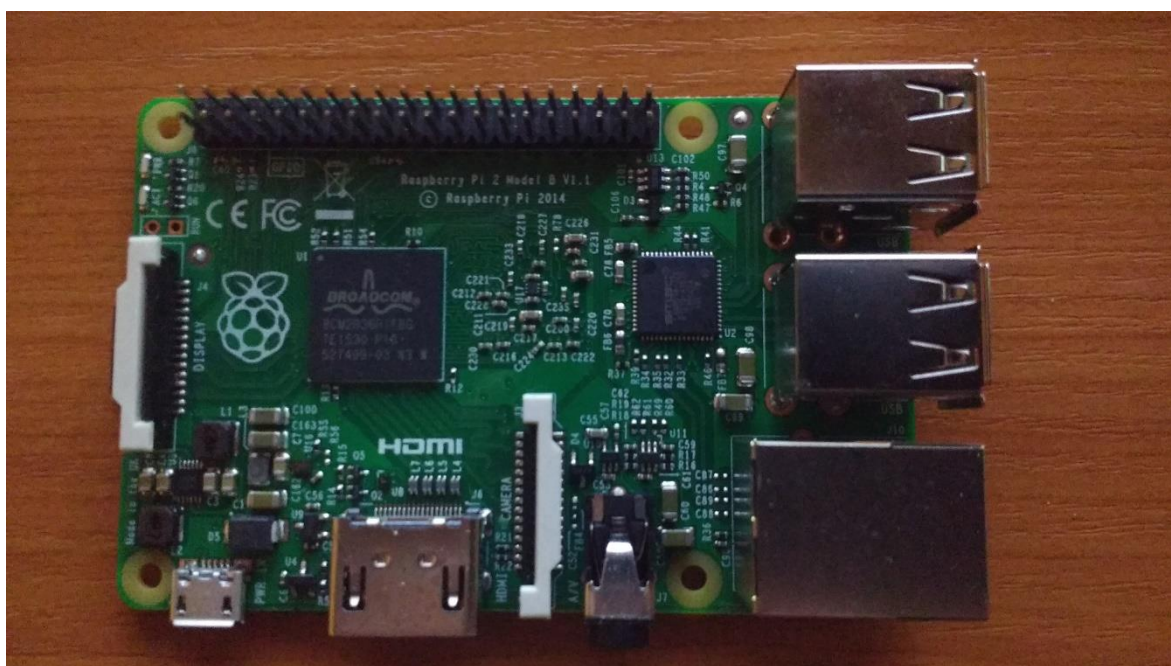
Raspberry Foundation on voittoa tavoittelematon järjestö, jonka tavoitteena on kehittää aikuisten ja lapsien tietotekniikan opiskelua (Raspberry Pi Foundation [Viitattu 1.2.2016]).

Raspberry Pi 2 Model B julkaistiin helmikuussa 2015. Se korvasi edeltäneen Raspberry Pi Model B+ version. Laitetta suositellaan käytettävän opiskeluun toisin kuin Raspberry Pi Model A+ -laitetta, joka soveltuu paremmin sulautettujen projekteihin ja projekteihin jotka vaativat vähäistä virrankulutusta. (Raspberry Pi Foundation [Viitattu 3.2.2016].)

Raspberry Pi 3 Model B julkaistiin helmikuussa 2016. Se korvasi Raspberry Pi 2 Model B laitteen. (Raspberry Pi Foundation [Viitattu 6.2.2016].)

2.2 Raspberry Pi laitteena

Työssä käytetään Raspberry Pi 2 Model B -laitetta, joka oli työn aloituksen aikaan uusien laitteista.



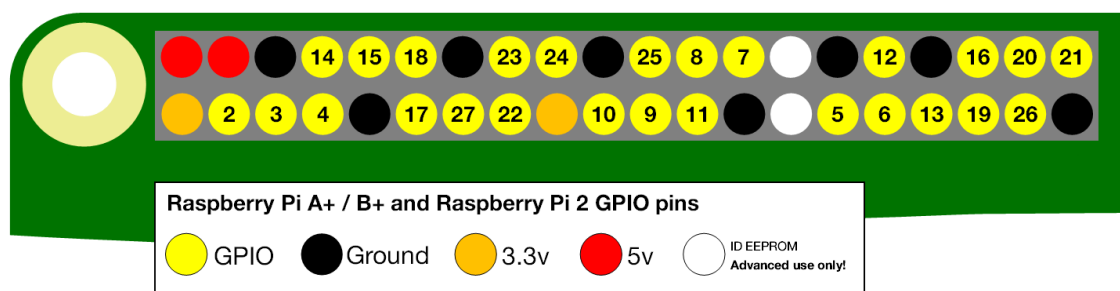
Kuva 1. Raspberry Pi 2 Model B.

Kuvassa 1 on Raspberry Pi 2 Model B. Raspberry Pi -laitteen fyysinen koko on 85,60 mm x 56 mm x 21 mm. Laitteen koko ei ole muuttunut Raspberry Pi 1 Model B+ -laitteen jälkeen, jolloin laitteeseen lisättiin enemmän GPIO-pinnejä ja kaksi uutta USB-liitintä. Laite painaa 45 g. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

Raspberry Pi vaatii toimiakseen 5V:n micro-USB-virtalähteen, joka kytketään laitteen kyljessä olevaan micro-USB-liitäntään. Laiteversiosta riippuen virtalähteen ampeeri vaatimus voi vaihdella. Virran lisäksi Raspberry Pi vaatii muistikortin, jossa on käytettävä käyttöjärjestelmä ja ajettava ohjelmakoodi. Muut lisälaitteet ovat valinnaisia laitteen käyttötarkoituksen mukaan. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

Raspberry Pi -laitteet käyttävät järjestelmäpiireinään Broadcommin piirejä, jotka sisältävät laitteen prosessorin, muistit ja näytönohjaimen. Raspberry Pi Model A-, B- ja B+ -versioissa on käytössä BCM2835-piiri. Raspberry Pi 2 -laitteen mukana tuli uusi BCM2865-järjestelmäpiiri. Raspberry Pi 3 -laitteeseen tuotiin uusi BCM2837-järjestelmäpiiri. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

Raspberry Pi Model B+ -laitteeseen lisättiin 14 GPIO-pinniä. GPIO-pinnejä on nyt 40. Pinnien määrää ei ole enään muutettu laitteisiin Raspberry Pi 2 ja Raspberry Pi 3. (Raspberry Pi Foundation [Viitattu 5.2.2016].)



Kuva 2. Raspberry Pi A+, B+ ja Raspberry pi 2 GPIO-pinnit (Raspberry Pi Foundation [Viitattu 3.2.2016].)

Kuvassa 2 on Raspberry Pi -laitteiden GPIO pinnijärjestys ja toiminnot. Kuvan 2 mukaan laitteen pinneistä on varattuna kaksi kappaletta 5 V:n virtaa varten, kaksi kappaletta 3.3 V:n virtaa varten. Kahdeksan pinneistä on maaliitäntöjä. Kaksi pinneistä on varattu muistia varten. Loput kaksikymmentäkuusi pinniä ovat GPIO-pinnejä, joita käyttäjä voi käyttää vapaasti kirjoittaakseen tai lukeakseen tietoa laitteista. Käyttöjärjestelmät voivat varata osan pinneistä muuhun tarkoitukseen ja tällöin kyseisiä pinnejä ei voi käyttäjä käyttää. (Raspberry Pi Foundation [Viitattu 3.2.2016].)

Kaikissa Raspberry Pi -laitteissa on Ethernet-liitin internetiä varten. Kaikissa laitteissa Raspberry Pi Model B+ jälkeen on myös neljä kappaletta USB-liittimiä erilaisten laitteiden, kuten näppäimistön, hiiren ja web-kameran liittämiseen. Jokaisessa laitteessa on Camera interface (CSI), joka mahdollistaa Raspberryn oman kameran liittämisen. Laitteissa on myös Display Interface (DSI). Laitteissa on myös yhdistetty 3,5 mm:n audio-liitin ja komposiittivideo. Laitteen saa kytkettyä näyttöön HDMI-liittimellä kyljessä olevan liittimen avulla. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

2.2.1 ARM

ARM Cortex-A7- ja Cortex-A8-prosessorit, joita Raspberry Pi -laitteet käyttävät, ovat ARM-yrityksen valmistamia virtaa säästäviä prosessoreja. Nämä prosessorit on suunniteltu mobiililaitteisiin ja sulautettuihin laitteisiin. ARM-prosessorit pohjautuvat 32-bittiseen prosessoritekologiaan, mutta Cortex-A8 tarjoaa myös 64-bittisen teknologian. Cortex-A8 on yhteensopiva Cortex-A7-prosessorille tehdyille ohjelmille. (Raspberry Pi Foundation [Viitattu 6.2.2016].)

ARM käyttää RISC-arkkitehtuuria, minkä ajatuksena on lähettää yksinkertaisia mutta tehokkaita käskyjä, jotka pyritään suorittamaan yhden prosessorityökierron aikana. Yksinkertaisilla komennoilla vähennetään monimutkaisuutta laitteistokäskyissä. Yksinkertaisuutta tosin ei saavuteta ilmaiseksi, sillä RISC-arkkitehtuuri aiheuttaa tavallista kovempaa kuormaa kääntäjälle. (Sloss, Symes & Wright 2004, 4.)

ARM-prosessorin suunnitteluun on vaikuttanut useita tekijöitä. Ensinnäkin ARM-prosessoreja käytetään usein laitteissa, jotka vaativat akkuvirtaa kuten puhelimet. ARM-prosessorit on suunniteltu pieniksi ja mahdollisimman virtapiheiksi, jotta akun kesto saadaan pidennettyä. Toiseksi ARM-laitteet on suunniteltu mahdollisimman pieniksi ja edullisiksi, jotta näitä kyetään käyttämään sulautetuissa piirilevyissä. (Sloss, Symes & Wright 2004, 5.)

2.2.2 Raspberry Pi 2 Model B

Raspberry Pi 2 Model B -laitetta on paranneltu uudella järjestelmäpiirillä BCM2865. BCM2865-järjestelmäpiirin 900 MHz:n nelilydin ARM Cortex-A7 prosessori on 200 MHz tehokkaampi kuin edeltäneen BCM2835-järjestelmäpiirin 700 MHz:n ARM11-prosessori. BCM2835-järjestelmäpiiriä käyttävät Raspberry Pi Model A, B, B+, Compute Model ja Raspberry Pi Zero. BCM2865-järjestelmäpiirissä on muistia 1 GB, kun edeltäneessä BCM2835 järjestelmäpiirissä muistia oli 512 MB. BCM2836 on muilta ominaisuuksiltaan identtinen edeltäneeseen BCM2835-järjestelmäpiiriin. Molemmissa toimii VideoCore IV -näytönohjain. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

Raspberry Pi 2-laitteen lisätty teho mahdollistaa uusien käyttöjärjestelmien käytön kuten Snappy Ubuntu Core ja Windows 10 IoT Core. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

2.2.3 Raspberry Pi 3 Model B

Raspberry Pi 3 -laitetta on paranneltu uudella Broadcom BCM2837-piirillä. Piirin prosessorina toimii 1,2 GHz:n 64-bit-nelilydin ARMv8-prosessori, joka on Raspberry Pi 2 Model B:n prosessoria 300 MHz nopeampi ja on toteutettu 64-bit arkkitehtuurilla. Tehokkaammasta prosessorista huolimatta BCM2837 on tehty samalla pohjarakkitehtuurilla kuin edeltäneet BCM2835 ja BCM2836. Tämä mahdollistaa vanhoille Raspberry Pi -versioille tehtyjen ohjelmien käytön Raspberry Pi 3 -laitteella. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

Raspberry Pi 3 -laitteeseen on lisätty sisäänrakennettu 803.11n Wifi ja Bluetooth 4.1, jotka mahdollistavat laitteen käytön ilman erillistä bluetooth-liitintä ja verkkojohdot (Raspberry Pi Foundation [Viitattu 5.2.2016]).

Raspberry Pi Foundation suosittelee Raspberry Pi 3 -laitteella käytettäväksi 2,5 ampeerin adapteria virtalähteessä, mikäli käyttäjä haluaa käyttää virtaa kuluttavia USB-laitteita (Raspberry Pi Foundation [Viitattu 5.2.2016]).



Kuva 3. Raspberry Pi 3 -laitteen erot edeltäjiinsä.
(Raspberry Pi Foundation [Viitattu 5.2.2016].)

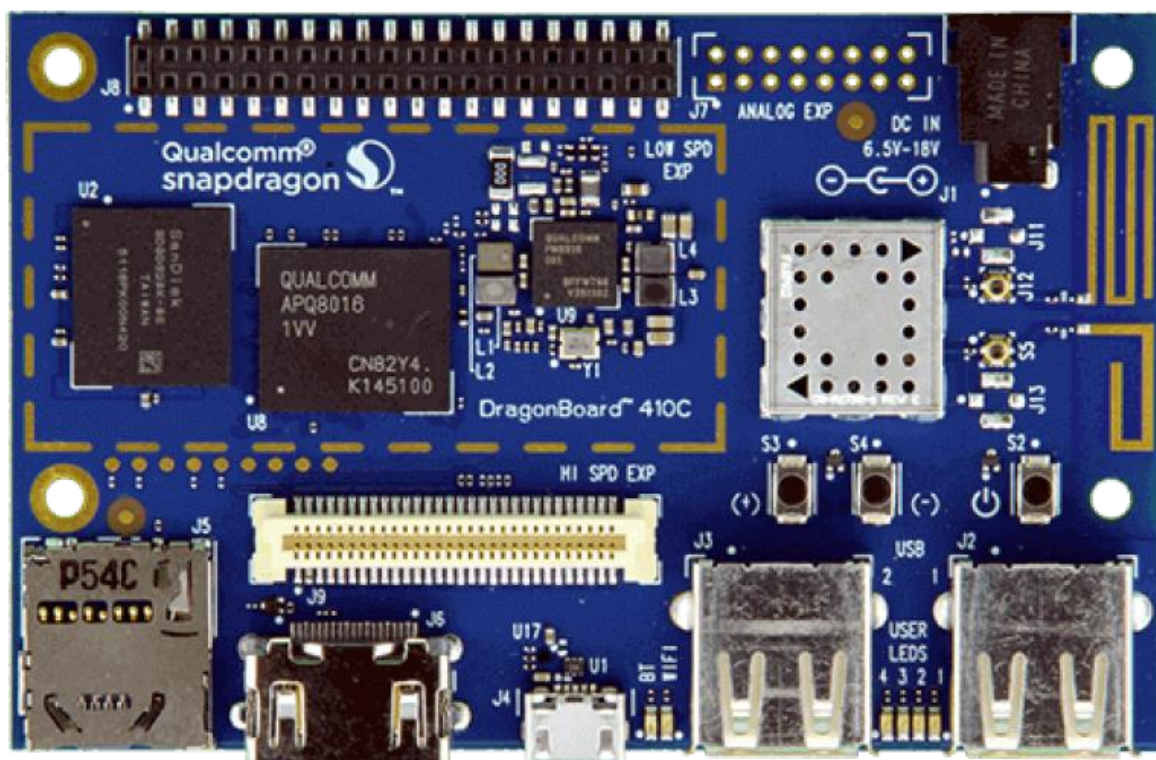
Kuvassa 3 näkyy Raspberry Pi 3 -laite. Laite on kooltaan vastaava kuin Raspberry Pi 2 -laite. Erona piirilevyssä on LED-valojen paikka, joka on siirretty laitteen toiselle puolelle. Vanhaan paikkaan on tullut Wifi-antenni. (Raspberry Pi Foundation [Viitattu 5.2.2016].)

2.3 Vaihtoehtoisia laitteita Raspberry Pille

Raspberry Pi -laitteelle on useita eri vaihtoehtoja, joista kerrotaan seuraavassa.

2.3.1 Dragonboard 410c

DragonBoard on Qualcomm Snapdragon -prosessoriin pohjautuva pankkikortin kokoinen tietokone (96boards, 2016).



Kuva 4. Dragonboard 410c
(96boards, 2016)

Kuvassa 4 on Dragonboard 410C. Laitteessa on sisäänrakennettu Wi-Fi, Bluetooth ja GPS. Prosessori on Quad-core ARM Cortex A53 1,2 GHz ja se on varustettu 32- ja 64-bittisellä tuella. Laitteessa on 1 GB LPDDR3 533 MHz muistia. Laitteessa on seuraavat liitännät: HDMI Type A -liitin, micro USB, kaksi kappaletta USB 2.0 ja micro-SD-korttipaikka. (96boards, 2016.)

Dragonboardissa on 40-pinninen low speed -laajennusliitäntä ja 60-pinninen high speed -laajennusliitäntä. Laitteessa on myös 16 lisäliitäntäpaikkaa. (96boards, 2016.)

Dragonboard tukee Android 5.1-, Debian 8.0- ja Windows 10 IoT Core -käyttöjärjestelmiä (96boards, 2016).

2.3.2 Arduino/Genuino UNO

Arduino Uno on mikrokontrollerilevy, joka pohjautuu ATmega328P-piiriin. Laitteessa on 14 digitaalista input/output-pinniä, 6 analogista input-pinniä, 16 MHz:n kide ja USB-liitin. (Arduino, 2016.)



Kuva 5. Arduino UNO
(Arduino, 2016.)

Arduino UNO on suunniteltu nopeaan laitteiden ja ohjelmien testaamiseen. Arduino UNO -laitteeseen ei voi asentaa käyttöjärjestelmää, vaan ohjelmat asennetaan suoraan mikrokontrollerille ajettavaksi tietokoneen USB-liitäntän kautta. (Arduino, 2016.)

Euroopassa laite julkaistaan Genuino UNO -nimellä (Arduino, 2016).

2.4 Eri laitteiden vertailu

Taulukko 1. Laittevertailu

Laite	Prosessori ja kellotaajuus	Muisti	Käyttöjärjestelmä	Hinta
Raspberry Pi Model B+	ARM11 700MHz	512MB	Kyllä	34,90€ Verkkokauppa.com
Raspberry Pi 2 Model B	ARM Cortex-A7 900Mhz	1GB	Kyllä	39,90€ Verkkokauppa.com
Raspberry Pi 3 Model B	ARM Cortex-A8 1.2GHz 64-bit quad-core	1GB	Kyllä	54,90€ Verkkokauppa.com
Dragonboard 410c	ARM Cortex A53 1.2GHz Quad-core	1GB LPDDR3 533MHz	Kyllä	75.00\$ www.arrow.com
Arduino UNO	ATmega328P 20MHz	32 kByte	Ei	29,90€ Verkkokauppa.com

Taulukossa 1 näkyy eri laitteiden prosessori, muisti, tuetut käyttöjärjestelmät ja hinta.

Arduino UNO jää selvästi tehossa ja muistissa muiden jälkeen. Tosin laitetta ei ole suunniteltu pyörittämään käyttöjärjestelmää, vaan se on tarkoitettu laitteiden ja koodien testaamiseen.

Uusin Raspberry Pi 3 Model B ja Dragonboard 410c ovat saman tehoisia. Suurin ero laitteiden välillä on GPIO-pinnien määrä.

Käyttöjärjestelmätuki on kaikissa paitsi Arduino-laitteissa. Jokainen Raspberry Pi -laite tukee useampaa Linux-käyttöjärjestelmää. Raspberry Pi 2 Model B ja Raspberry Pi 3 Model B tukevat molemmat Windows 10 IoT Core -käyttöjärjestelmää

(Raspberry Pi Foundation [Viitattu 4.2.2016]). Dragonboard tukee ainoastaan Android 5.1-, Debian- ja Windows 10 IoT Core -käyttöjärjestelmää (96boards 2016).

Vain Raspberry Pi 3- ja Dragonboard-laitteissa on WiFi ja Bluetooth. Dragonboard 410C sisältää myös GPS-toiminnon.

Kaikki taulukossa 1 luetellut laitteet ovat saatavilla Suomessa lukuun ottamatta Dragonboard 410C -laitetta.

2.5 Raspberry Pi -käyttöjärjestelmät

Raspberry Pi on suunniteltu toimimaan GNU/Linux-käyttöjärjestelmillä eli Linux-käyttöjärjestelmillä. Prosessori Raspberry Pi:ssä on 32-bittinen ARM-prosessori, joka vaatii toimiakseen ARM-arkkitehtuurille käännetyn käyttöjärjestelmän. (Raspberry Pi Foundation [Viitattu 6.2.2016].)

Käyttöjärjestelmiä laitteeseen on useita mahdollisia. Näistä käyttäjä voi käyttötarkoituksensa mukaan valita haluamansa. Raspbian on Raspberry Foundationin suosittu käyttöjärjestelmä Raspberrille. Raspbian on ilmainen käyttöjärjestelmä, joka on optimoitu Raspberry Pi -laitteiden käyttöön. Tämä pohjautuu Linux-pohjaiseen Debian-käyttöjärjestelmään. Muita Linux-pohjaisia Raspberry Pi Foundationin suosimia käyttöjärjestelmiä ovat Ubuntu Mate, Snappy Ubuntu Core, OSMC, OpenElec ja PiNet. Näiden lisäksi ei Linuxiin pohjautuvia käyttöjärjestelmiä ovat Windows 10 IoT Core ja RISC OS. (Raspberry Pi Foundation [Viitattu 6.2.2016].)

NOOBS on helppokäyttöinen käyttöjärjestelmän asennusohjelma Raspberry Piille. Sen tarkoituksena on helpottaa Raspberry Pi -laitteen käyttöjärjestelmän asennusta. NOOBS-lyhenne tulee sanoista New Out Of the Box. NOOBS -järjestelmän voi ostaa kaupasta suoraan muistikortille asennettuna tai ladata internetistä ilmaiseksi ja asentaa itse muistikortille. NOOBS-ohjelma kysyy ensimmäisellä käynnistyskerrallaan, minkä käyttöjärjestelmän käyttäjä haluaa asentaa. Näitä käyttöjärjestelmiä ovat Raspbian, Pidora, OpenELEC, OSMC, RISC OS ja Arch Linux. Ainoastaan Raspbian on muistikortilla valmiina. Kaikki muut käyttöjärjestelmät NOOBS lataa internetin välityksellä. (Raspberry Pi Foundation [Viitattu 2.2.2016].) Versiosta

1.6.0 alkaen Windows 10 IoT Core on ollut mukana NOOBS-paketissa (Microsoft [Viitattu 16.3.2016]).

3 Windows 10 IoT Core

3.1 Windows 10 IoT Core

Windows 10 IoT Core on Windows 10 -versio, joka on optimoitu pienemmille laitteille. Sitä voidaan käyttää ilman näyttöä tai näytön kanssa. Laitteita, joissa Windows 10 IoT Core toimii, ovat Raspberry Pi 2 ja 3, Arrow DragonBoard 410c ja MinnowBoard MAX. (Microsoft [Viitattu 17.3.2016].)

3.2 Windows 10 IoT Coren ominaisuudet

Windows 10 IoT Core -ohjelmien tekemiseen voi käyttää Visual Studio Community editionia. Visual Studio on ilmainen ja mahdollistaa Universal Windows -ohjelmien tekemisen sekä ohjelmien testaamisen ja julkaisemisen. (Microsoft [Viitattu 18.3.2016].)

Windows 10 IoT Core mahdollistaa teknologioiden käytön. Connect the Dots, joka mahdollistaa ohjelmien yhdistämisen Microsoft Azure -palveluun -pilvipalveluiden kehittämistä varten. Lisäksi se mahdollistaa helppokäyttöisen Arduino Wiring Apin, jota käytetään Arduino luonnoksissa ja kirjastoissa suoraan laitteisto käyttöön. (Microsoft [Viitattu 18.3.2016].)

3.3 Windows 10 Universal Windows Platform

UWP toimii jokaisella Windows-pohjaisella laitteella, kuten tietokoneella, puhelimella ja Raspberry Pi -laitteilla. UWP mahdollistaa sen, että yksi ohjelma voi pyöriä useilla eri Windows-laitteilla ilman ohjelman muokkausta. (Microsoft [Viitattu 16.3.2016].)

UWP-ohjelmat säätävät kokoaan automaattisesti skaalausalgoritmilla. Tämän avulla ohjelmat saadaan sopimaan käytettävälle laitteelle muokkaamatta koodia.

Tämä skaalaus säätää ohjelman fonttien ja elementtien kokoa varmistaen, että ohjelman katsominen tietokoneen ruudulta sekä puhelimen näytöltä ovat molemmat hyvin katsottavissa ja käytettävissä. (Microsoft [Viitattu 16.3.2016].)

UWP käyttää Windows Runtime APIa, mikä on rakennettuna koneen käyttöjärjestelmään ja on kehitetty C++ kielellä. UWP tukee C++-, C#-, Visual Basic- ja Javascript-ohjelmointikieliä. Microsoft Visual Studio 2015 tarjoaa jokaiselle edellä mainitulle kielelle valmiit pohjat, jotka mahdollistavat ohjelmoijaa tekemään yhden ohjelmapaketin. Tämän paketin käyttäjät voivat ladata mille tahansa Windows 10 -laitteelle. (Microsoft [Viitattu 16.3.2016].)

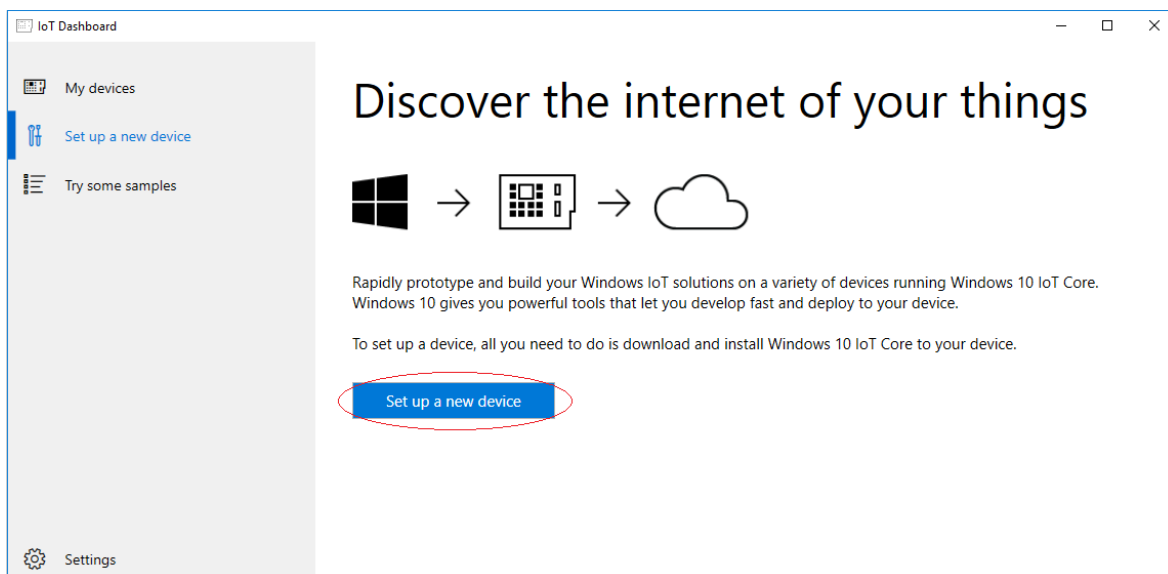
3.4 Windows 10 IoT Coren asennus

Windows 10 IoT Core voidaan asentaa laitteelle usealla eri tavalla. Ennen asennusta tulee käyttäjällä olla asennettuna Windows 10 -käyttöjärjestelmä ja Windows 10 IoT Core Dashboard. Windows 10 IoT Core Dashboard mahdollistaa Windows 10 IoT Coren asennuksen SD-kortille, käyttöjärjestelmän etäkäytön ja esimerkkiohjelmien katselun. (Microsoft [Viitattu 18.3.2016].)

Microsoft suosittelee 8 GB:n micro-SD-korttia, joka on luokkaa 10 (Microsoft [Viitattu 18.3.2016]).

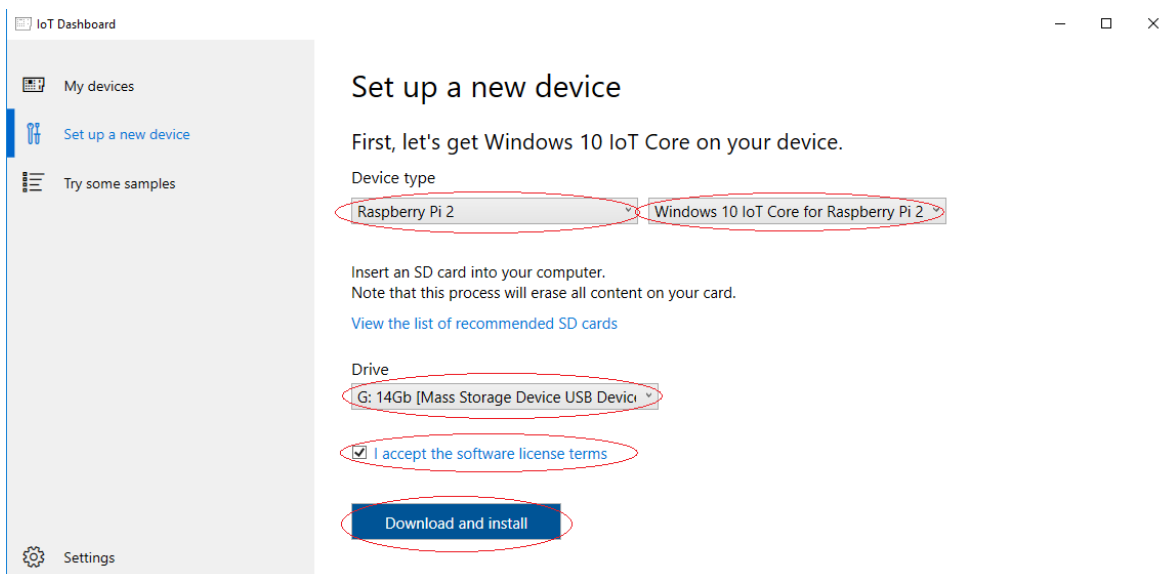
3.4.1 Windowsin omilla ohjeilla

Windowsin omalla Windows 10 IoT Core Dashboard -ohjelmalla asennettaessa tulee käyttäjällä olla oma micro-SD-muistikortti ja muistikortin lukija. Käyttäjän tulee kytkeä nämä tietokoneeseen, jossa on Windows 10 IoT Core Dashboard asennettuna. Tämän jälkeen ohjelmasta voi painaa kuvan 6 mukaisesti Set up a new device -painiketta. (Microsoft [Viitattu 18.3.2016].)



Kuva 6. Windows 10 IoT Core Set up a new device -painike.

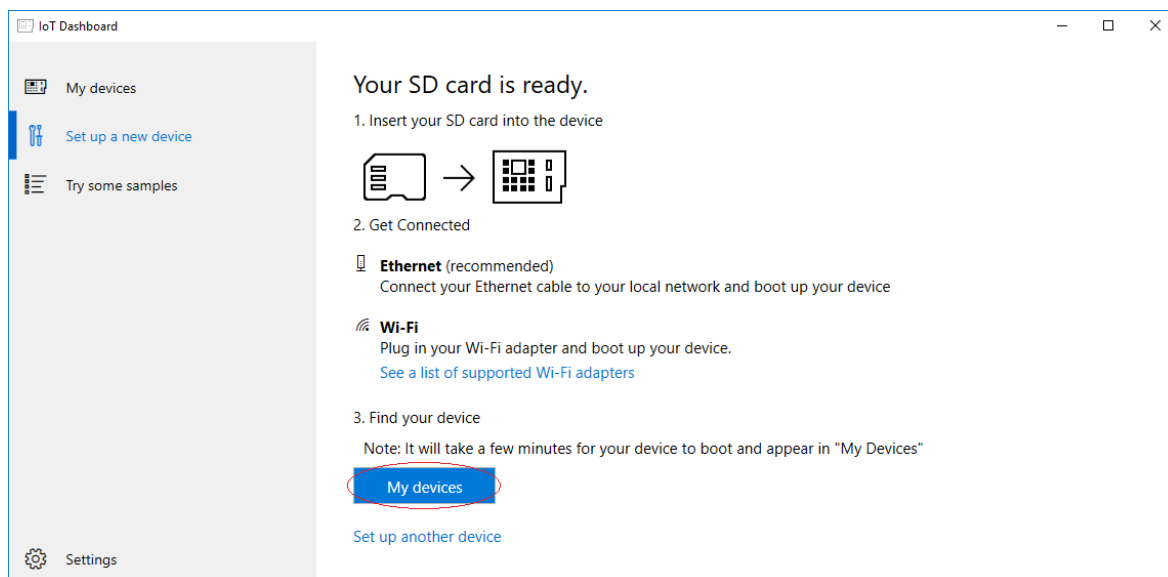
Tämän jälkeen ohjelma pyytää valitsemaan kohdelaitteen, käyttöjärjestelmän ja halutun aseman johon käyttöjärjestelmä asennetaan. Kuvassa 7 näkyy työssä käytetyt asetukset.



Kuva 7. Laitteen, käyttöjärjestelmän ja aseman valinta

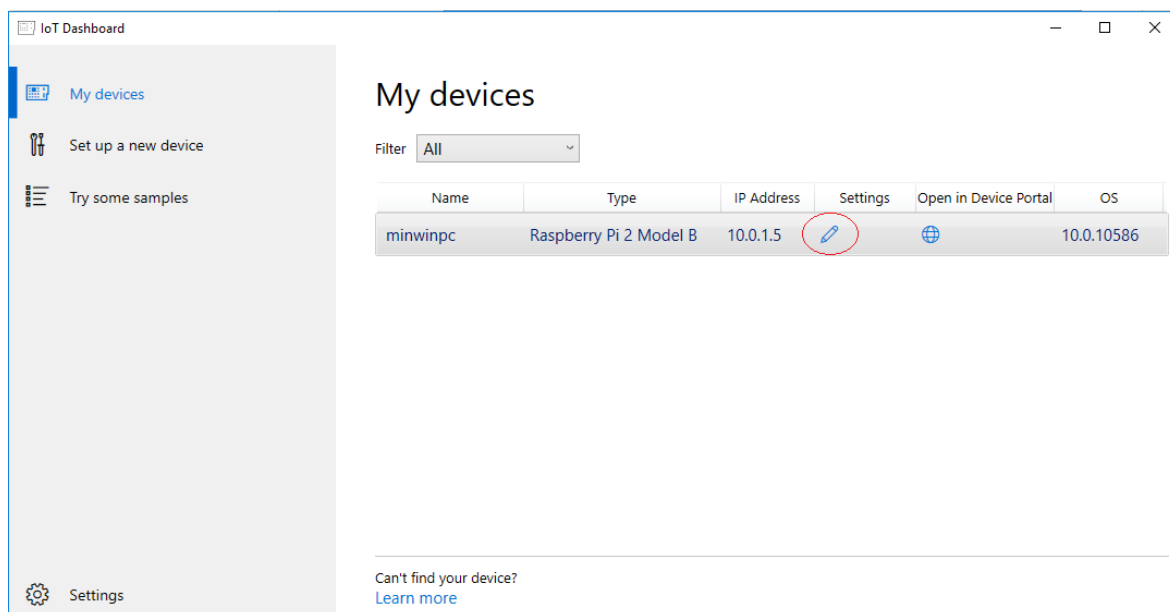
Seuraavaksi Windows 10 IoT Dashboard lataa tarvittavat tiedostot ja asentaa käyttöjärjestelmän muistikortille. Asennuksen jälkeen muistikortin voi irrottaa lukijasta ja kytkeä Raspberry Pi 2 -laitteeseen. Ensimmäinen käynnistyskerta voi kestää useamman minuutin.

Asennuksen jälkeen tulee käyttäjän kytkeä Raspberry Pi 2 -laite samaan verkkoon kuin tietokone. Kytkeä tapahtuu käyttäen joko tavallista verkkokaapelia tai erikseen ostettavaa Wi-Fi-liitintä. Microsoft suosittelee asentamista verkkokaapelin kautta. (Microsoft [Viitattu 18.3.2016].) Verkkoon kytkemisen jälkeen käyttäjä voi painaa kuvan 8 mukaisesti My devices painiketta.



Kuva 8. My Devices -painike

My devices -valikosta löytyy listaus verkossa olevista laitteista (kuva 9). On suositeltavaa, että käyttäjä käy vaihtamassa laitteen nimen ja salasanan. Tämä tapahtuu laitteen asetuksista painamalla kynäikonia Settings-kohdasta. (Microsoft [Viitattu 18.3.2016].)



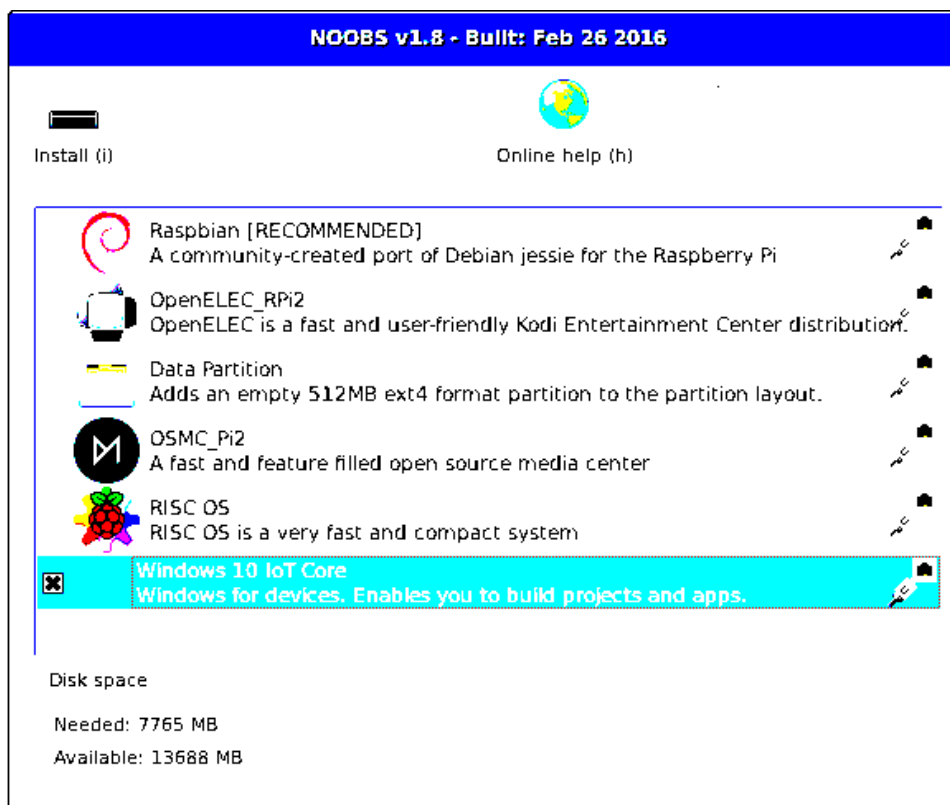
Kuva 9. My devices -valikko ja Settings-painike.

Kuvassa 9 on IoT Dashboard –sovellus. Tämän jälkeen laite on valmis käytettäväksi. Nyt käyttäjä voi tehdä omia ohjelmia tai kokeilla Dashboardista löytyviä valmiita ohjelmia.

3.4.2 Asennus NOOB-ohjelman avulla

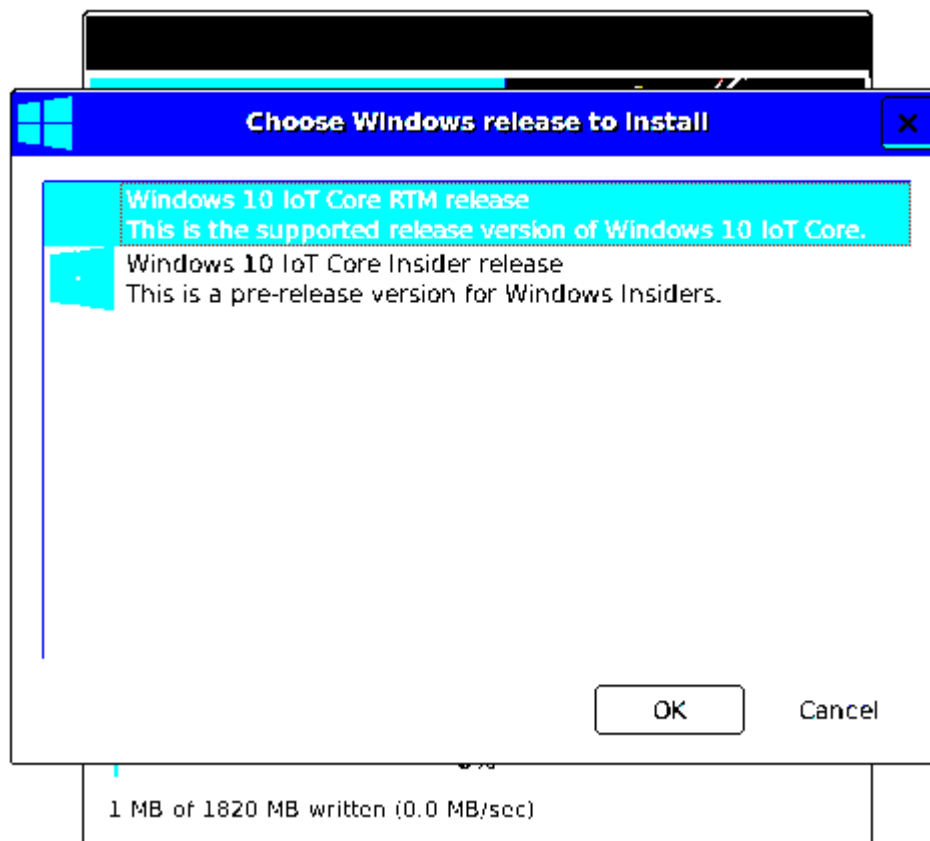
NOOBS-asennusmanageria käytettäessä tulee käyttäjällä olla muistikortti, johon on esiasennettu NOOBS. Toisena vaihtoehtona tulee ladata ja asentaa NOOBS muistikortille sivulta <https://www.raspberrypi.org/downloads/noobs/>. Muistikortti tulee alustaa ennen NOOBS-tiedostojen siirtoa sille. (Raspberry Pi Foundation [Viitattu 2.2.2016].)

NOOBS-muistikortti kytketään Raspberry Pi -laitteeseen. Laitteeseen kytketään virta, verkkokaapeli, näyttö ja hiiri. Kun kaapelit on kytketty, NOOBS-asennus alkaa. (Microsoft [Viitattu 18.3.2016].)



Kuva 10. NOOBS-käyttöjärjestelmän valinta

Laitteen käynnistyttyä näyttöön tulee kuvan 10 mukainen ikkuna. Tästä ikkunasta valitaan Windows 10 IoT Core -käyttöjärjestelmä. Seuraavaksi painetaan Install-painiketta, ja käyttöjärjestelmän asennus alkaa. Asennuksen alussa tulee valita halu-aako asentaa RTM release -version vai Insider releasen. Insider-versio on kehitys-versio käyttöjärjestelmästä ja vaatii kirjautumisen Microsoft-tiliin. Tässä työssä käytetään kuvan 11 mukaisesti RTM release -versiota. (Microsoft [Viitattu 18.3.2016].)

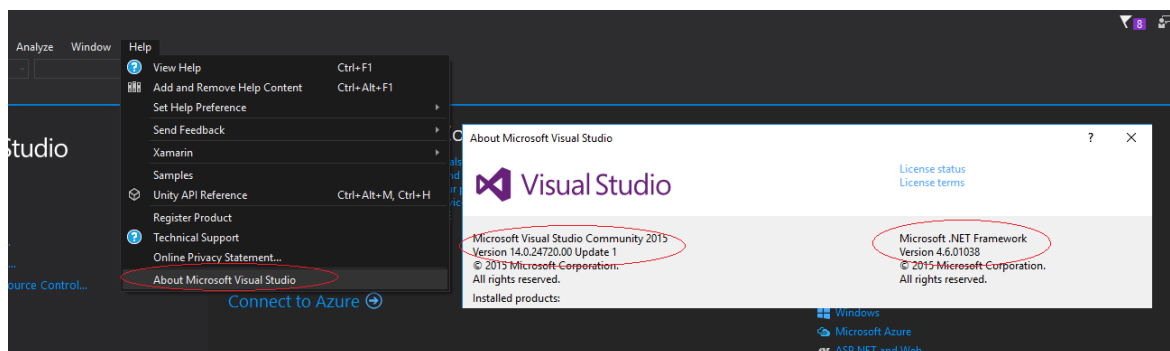


Kuva 11. Käyttöjärjestelmäversion valinta

Tämän jälkeen NOOBS lataa ja asentaa Windows 10 IoT Core -käyttöjärjestelmän muistikortille (Microsoft [Viitattu 18.3.2016]).

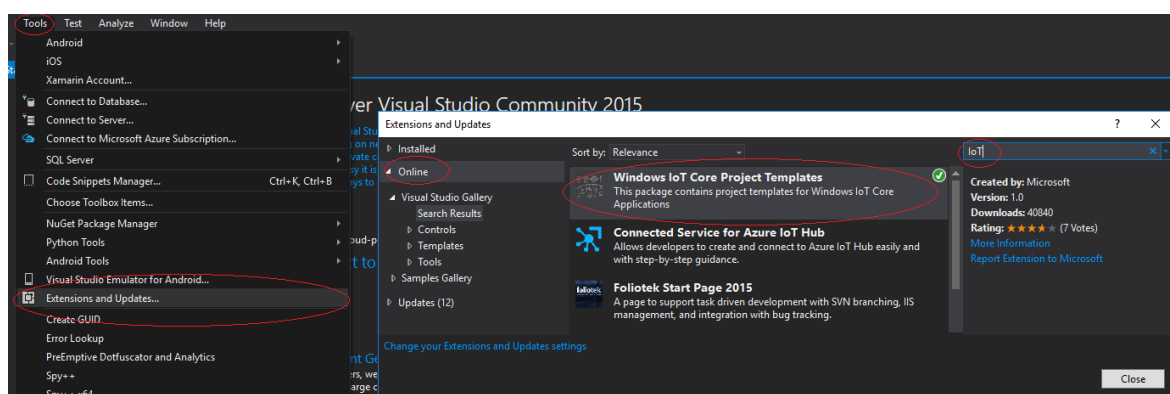
3.4.3 Visual Studion asennus ja valmiiksi laitto

Windows 10 IoT Core -ohjelmien ohjelmoimiseen vaaditaan tietokoneelta Windows 10 -käyttöjärjestelmää. Koneella tulee myös olla asennettu Visual Studio 2015. Visual Studio asentaa ensin Update 1 -päivityksen. Käyttäjä voi varmistaa version painamalla About Microsoft Visual Studio -painiketta. Version tulee olla Visual Studio 14.0.24720.00 tai uudempi. Visual Studio Tools for Universal Windows Apps -version tulee olla 14.0.24720.00 tai uudempi. (Microsoft [Viitattu 18.3.2016].)



Kuva 12. Visual Studio -versioiden tarkistus

Kuvassa 12 on About-ikkuna, josta näkee käyttöjärjestelmän. Seuraavaksi käyttäjän tulee asentaa Windows IoT Core Project Template. Tämä tapahtuu painamalla valikosta Tools Extensions and Updates ja siitä aukeavasta ikkunasta painamalla Online. Online kentän valinnan jälkeen tulee listaus ohjelmista, ja käyttäjä voi hakea hakukentän avulla tarvittavan Windows IoT Core Project Templatesin. Oikean löydettyä tulee se valita ja asentaa. (Microsoft [Viitattu 18.3.2016].)



Kuva 13. Windows IoT Core Project Templaten asennus

Kuvassa 13 näkyy Windows 10 IoT Core Project Templaten asennus. Tämän jälkeen Visual Studio on valmis Windows 10 IoT Core -ohjelmien luomiseen.

4 Palvelin

4.1 Apache

Apache-verkkopalvelin on Apache Software Foundationin ylläpitämä projekti avoimen lähdekoodin HTTP-palvelimien valmistamiseen ja ylläpitoon. Nämä palvelimet tukevat moderneja käyttöjärjestelmiä kuten UNIX ja Windows. Windows-käyttöjärjestelmälle on mahdollista ladata useampia Apache-palvelinsovelluksia, kuten ApacheHaus, WampServer ja XAMPP. (Apache [Viitattu 10.4.2016].)

4.2 XAMPP

XAMPP on Apache Friendsin tekemä Apache-palvelin, jossa on tuki MariaDB-tietokannalle sekä PHP- ja Perl-ohjelmointikielille. XAMPP on helposti asennettava Apache-palvelin. XAMPP-ohjelmisto antaa käynnistää palvelimet: Apache, MySQL, FileZilla, Mercury ja Tomcat. (XAMPP [Viitattu 2.4.2016].)

4.3 PHP

PHP on laajasti käytössä oleva avoimen lähdekoodin skriptikieli, joka soveltuu web-ohjelmointiin ja voidaan upottaa HTML-sivuun. Erona JavaScript-kieleen on se, että PHP ajetaan web-palvelimella eikä käyttäjän koneella. Palvelimella ajettu koodi estää käyttäjää näkemästä mitään, mikä on PHP-koodin sisällä. (PHP [Viitattu 10.4.2016].)

4.3.1 PHP-kielen historia

PHP sai alkunsa vuonna 1994 kun Rasmus Lerdorf kehitti Common gateway interfacen (CGI) käyttäen C-ohjelmointikieltä. Tämän avulla hän seurasi katsomiskertoja internet ansioluettelonsa. Hän nimesi tekemänsä skriptit Personal Home Page

Tools eli lyhyesti PHP. Sitten hän kirjoitti PHP-työkalut uusiksi useampaan otteeseen. Näistä muodostui lopulta moderni PHP-kieli. PHP 3 oli ensimmäinen versio, joka läheisesti muistuttaa nykyistä PHP-kieltä. PHP 3 valmistui vuonna 1998. PHP 4 julkaistiin vuonna 2000 ja PHP 5 vuonna 2004. Uusin versio on joulukuussa 2015 julkaistu PHP 7.0. (PHP [Viitattu 10.4.2016].)

4.3.2 PHP-kielen käyttötarkoitukset

PHP pohjautuu kielenä pääosin palvelinpuolen skiptaamiseen. PHP pystyy keräämään esimerkiksi form dataa, luomaan dynaamista sivun sisältöä tai lähettämään ja vastaanottamaan evästeitä. PHP-kielillä voidaan ohjelmoida palvelinsovelluksia sekä työpöytäohjelmia. (PHP [Viitattu 10.4.2016].)

Palvelinpuolen skriptaaminen on yleisin PHP-kielen käyttötarkoitus ja tähän tarvitaan kolme asiaa, jotta tämä onnistuu. Ensimmäisenä tarvitaan PHP-parseri (CGI tai serveri moduuli). Toiseksi tarvitaan web-palvelin ja viimeisenä tarvitaan web-selain. Web-palvelimeen tulee olla asennettu PHP-toiminto. Kaikki edellä mainitut toiminnot pystyy tekemään kotikoneella. (PHP [Viitattu 10.4.2016].)

Komentorivi skriptaamisessa ei tarvita palvelinta tai selainta. Tätä käytetään usein käytettyjen skriptien ajamiseen koneella. Ainoana vaatimuksena on PHP-parseri. (PHP [Viitattu 10.4.2016].)

Työpöytäsovellusten tekoon PHP ei ole paras mahdollinen vaihtoehto (PHP [Viitattu 10.4.2016]).

5 Windows 10 IoT Core kodinvalvonta

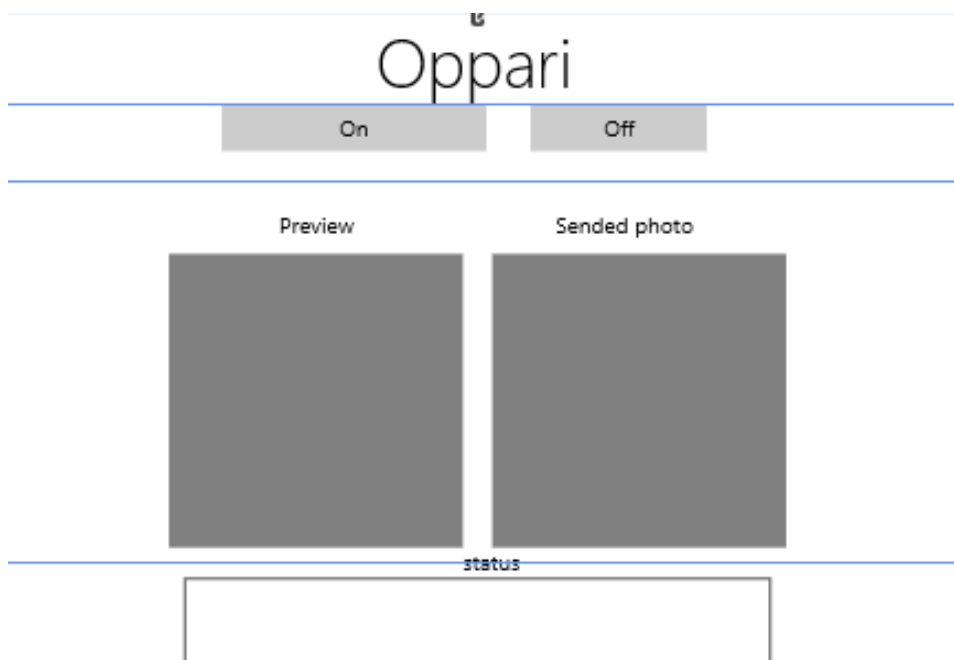
Työn yksi päätarkoituksista oli tutustua Windows 10 IoT Core -käyttöjärjestelmään. Olemassa olleen laitteiston takia työssä päädyttiin tekemään yksinkertainen kodinvalvontajärjestelmä.

Laitteistona työssä oli Raspberry Pi 2 Model B, johon oli kytkettynä Kingstonin 16 GB:n Class 4 Micro-SD-muistikortti. Microsoftin suosittelee käytettäväksi Class 10 -muistikorttia, mutta class 4 toimi moitteettomasti. Lisäksi laitteeseen oli kytketty verkkokaapeli, Logitech G400 -hiiri ja Microsoft LifeCam HD-3000 -kamera. Virran laite sai tietokoneen USB-portista. Laite oli kytkettynä näyttöpäätteeseen HDMI-kaapelilla. Huomioitavaa oli, että laite ei käynnisty mikäli näytön kytki laitteeseen käynnistyksen aikana.

Windows 10 IoT Core -käyttöjärjestelmä asennettiin Windowsin omilla ohjeilla käyttämällä Windows 10 IoT Core Dashboardia.

5.1 Windows 10 IoT Core -ohjelma

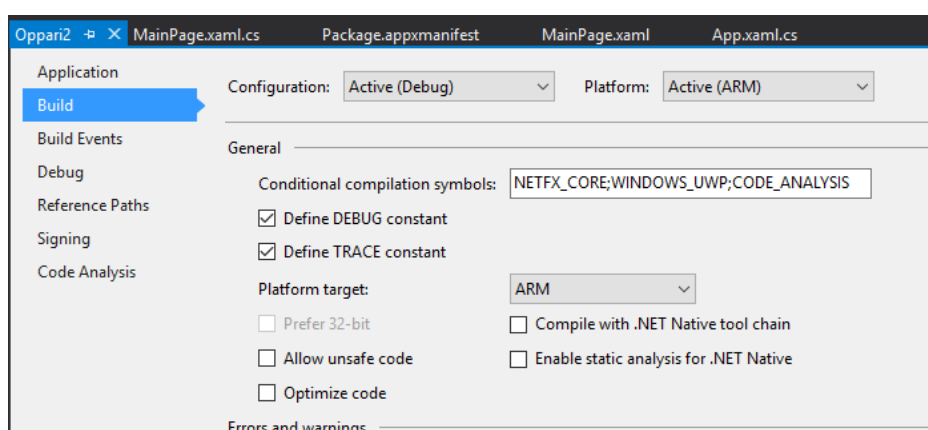
Ohjelmointi toteutettiin C#-ohjelmointikielellä. Ohjelmassa on 2 painiketta, joita painamalla käyttäjä voi kytkeä valvonnan päälle. Ohjelman käynnistyttyä Raspberry Pi 2 -laitteeseen kytketty kamera aloittaa kuvien ottamisen ja lähettämisen, kunnes käyttäjä painaa Off-painiketta.



Kuva 14. Ohjelman perusnäky

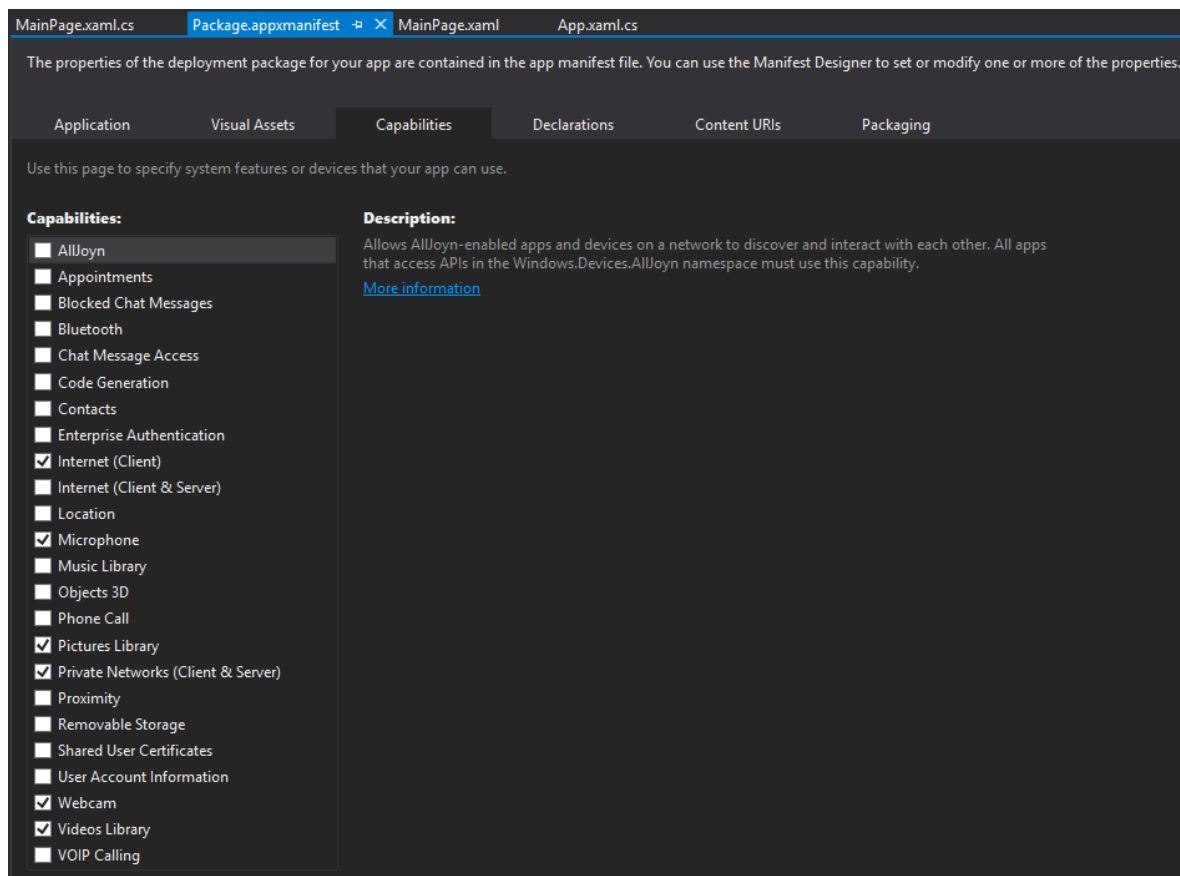
Kuvassa 14 on Windows 10 IoT Core -ohjelman perusnäky. Windows 10 IoT Coren ohjelmointiin tutustuttiin valmiiden mallien pohjalta. Ohjelma ottaa kuvan 3 sekunnin välein On-painikkeen painamisen jälkeen ja lähettää otetut kuvat web-palvelimelle.

Ohjelma aloitettiin uutena Visual C# Universal App -projektina ja ohjelman asetuksista vaihdettiin Platform target kuvan 14 mukaiseksi.



Kuva 15. Platform target

Kuvassa 15 on valittu ohjelman kohdealustaksi ARM. Windows 10 IoT Core -ohjelmoinnissa tulee määrittellä toiminnot, joita laite saa käyttää. Toimintojen määrittäminen tapahtuu Package.appxmanifest-tiedostossa Capabilities-välilehdellä. Tässä työssä tarvittavat toiminnot näkyvät kuvassa 16.



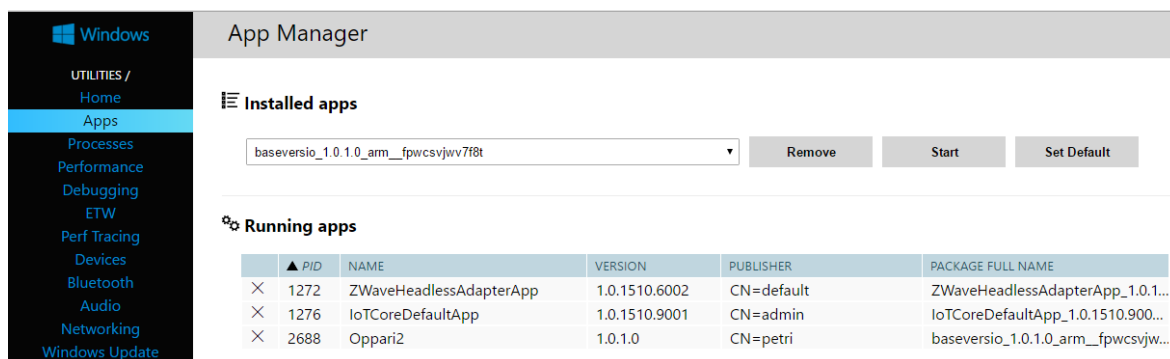
Kuva 16. Ohjelmassa tarvittavat toiminnot

Kuvassa 16 näytetään oikeuksien antaminen ohjelmassa käytetyille toiminnoille. Kuvien ottaminen ohjelmassa toteutettiin käyttämällä MediaCapture-kirjastoa. StorageFile-kirjastoa käytettiin kuvien tallentamiseen. Ohjelma tallentaa kuvan laitteen muistiin nimellä photo.jpg. Mikäli tiedosto on jo olemassa lisää, se nimen perään kasvavan numeron.

Kuvien lähetys palvelimelle toteutettiin käyttämällä http-kirjastoa, joka mahdollistaa tiedon lähettämisen web-palvelimelle. Tämä tapahtui yksinkertaisesti määrittelemällä palvelimen osoite ja mitä halutaan lähettää. Kuvat muokattiin bitti-taulukoksi lähettämistä varten. Web-palvelin huolehtii loput tiedon käsittelystä.

Ohjelman testaamista varten Raspberry Pi -laite tuli olla kytkettynä samaan verkkoon. Laitteen ip-osoite tuli olla tiedossa ja tämän näki Windows 10 IoT Core dashboard -ohjelmasta. Ohjelmaa testattiin käyttämällä Visual studion Debug -toimintoa. Tämä näyttää käyttäjälle mahdolliset virheet, ja ohjelmaa ei asenneta Raspberry Pi -laitteelle vaan ajetaan sitä siellä väliaikaisesti.

Kun ohjelma oli saatu toimimaan halutulla tavalla, se julkaistiin valmiiksi Raspberry Pi -ohjelmaksi. Ohjelma asennettiin laitteelle ja sen pystyi käynnistämään Windows 10 IoT Core Dashboardin avulla. Ohjelman pystyi laittamaan oletusohjelmaksi Raspberry Pi -laitteen käynnistyessä Dashboardin avulla. Mikäli ohjelman poisti laitteelta tämän ollessa oletuskäynnistysohjelma, niin Raspberry Pi -laitetta ei kykene käynnistämään ilman käyttöjärjestelmän uudelleen asennusta.



Kuva 17. Oman ohjelman käynnistys ja oletusohjelmaksi asetetus Windows 10 IoT Core Dashboardin avulla

Kuvassa 17 näkyy Windows 10 IoT Core Dashboardi, jossa näkyy päällä olevat sovellukset ja ohjelman käynnistys oletusohjelmaksi. Liite 1 sisältää Raspberry Pi -laitteella olevan koodin ja Liite 2 tämän tyylikoodin.

5.2 PHP-palvelin ja MYSQL-palvelin

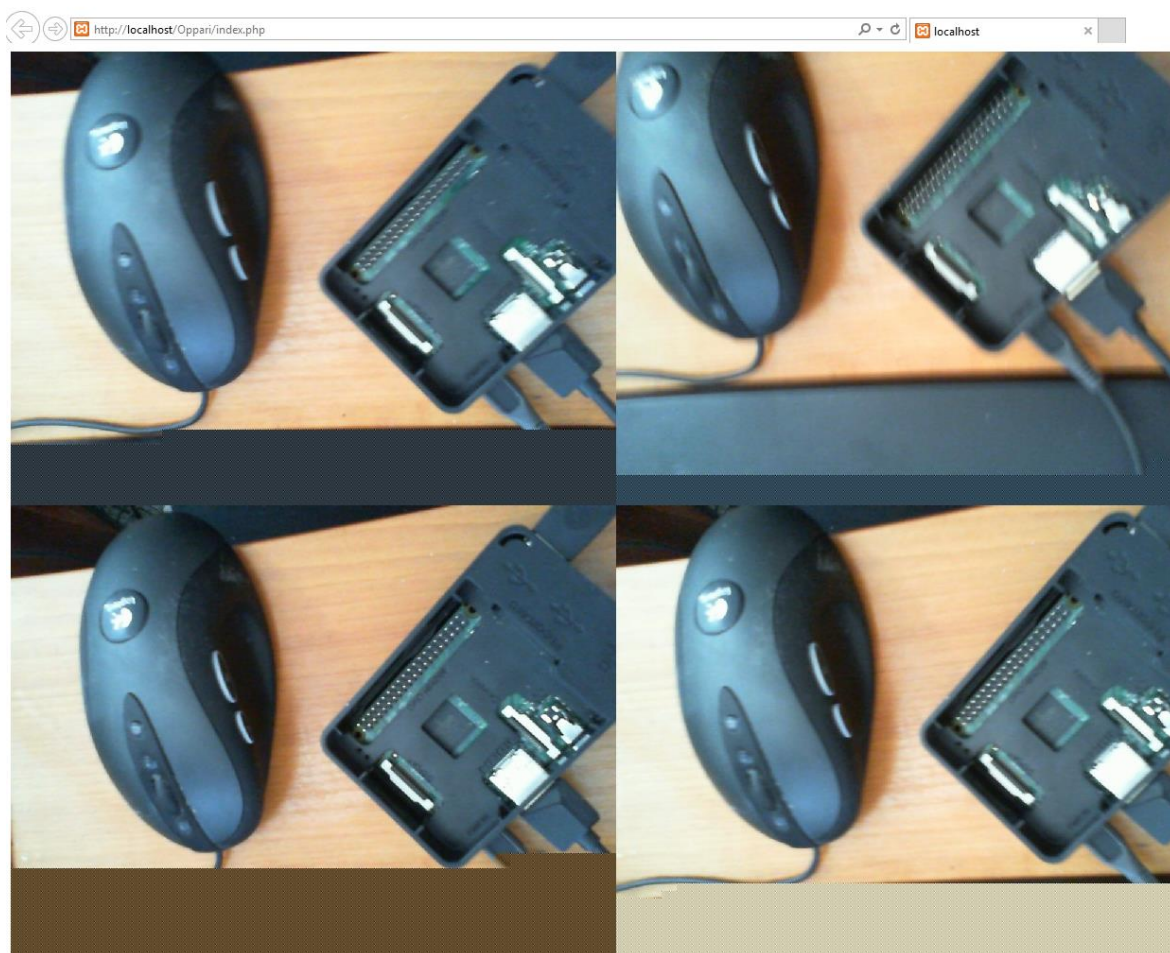
Työn toisena osiona oli toteuttaa palvelin, johon Raspberry Pi -laite lähettää kuvat. Tämä palvelin toteutettiin käyttämällä ilmaista XAMPP-ohjelmistoa. Apache-palvelin on palvelinohjelma, jolla työssä käytetty PHP-palvelin suoritettiin.

Itse ohjelman koodaamiseen käytettiin NetBeans-ohjelmaa, joka on ilmainen avoimen lähdekoodin ohjelma. Tällä voi kehittää ohjelmia useille eri kielille, kuten HTML, PHP ja Java (NetBeans [Viitattu 3.4.2016]).

Ennen ohjelman tekoa tuli Apache-palvelimelle käydä antamassa oikeudet vastaanottaa lähiverkosta saapuvat POST-komennot. Oikeudet täytyi lisätä tiedostossa `httpd-xampp.conf`, joka löytyy kansioista, johon XAMPP on asennettu.

PHP-palvelimen koodi oli jaoteltu kahteen eri osioon. Näistä ensimmäinen ohjelma vastasi tiedon vastaanotosta Raspberry Pi -laitteelta, tietojen lähettämisestä MySQL-palvelimelle ja varmistamalla, että tietokannassa on vain viisi viimeisintä kuvaa. Ohjelmakoodi odottaa, että se saa Raspberry Pi -laitteelta POST-komennon, minkä jälkeen tallennetaan POST-komennon mukana tullut base64-muotoinen kuvadata. Datan vastaanoton jälkeen muodostetaan yhteys tietokantaan käyttämällä `mysqli`-komentoja (w3schools.com [Viitattu 3.4.2016]). Kuva tallennetaan tietokantaan BLOB-muodossa ja tietokanta luo tälle oman yksikäytteisen tunnisteiden. Kuvan tallennuksen jälkeen ohjelma vielä varmistaa, että tietokannassa ei ole enempää kuin viisi viimeisintä kuvaa tallennettuna.

PHP-palvelimen toinen koodi näyttää käyttäjälle internetselaimessa viisi viimeisintä kuvaa. Ohjelma käy lukemassa tietokannasta kaikki viisi kuvaa, jotka sinne on tallennettu, ja näyttää nämä käyttäjälle.



Kuva 18. Palvelimen näyttämät kuvat.

Kuvassa 18 näkyy Raspberryltä vastaanotettu kuva. Tässä työssä nettisivuna toimi `localhost/Oppari/index.php`. Jos ohjelmaan yhdistettiin toisella laitteella netin kautta, tuli osoitinrivin `localhost` sana vaihtaa tietokoneen ip-osoitteella. Liite 3 sisältää vastaanottokoodin (PHP). LIITE 4 sisältää kuvat näyttävän koodin.

6 TYÖN TULOKSET

6.1 Tulokset

Työssä saatiin toteutettua kodinvalvontajärjestelmä, joka kykenee ottamaan kuvia Raspberry Pi 2 Model B -laitteeseen kytketyllä web-kameralla käyttäen Windows 10 IoT Core -käyttöjärjestelmää. Saadut kuvat saatiin lähetettyä palvelimelle käyttäen http POST komentoa ja tallennettua palvelimen MySQL-tietokantaan. Tietokannassa olleet kuvat saatiin näytettyä onnistuneesti halutulla laitteella.

Kuvien katsomista testattiin palvelinkoneella sekä Android-puhelimella. Android-puhelimella testaus tapahtui lähiverkossa eikä internettoteutusta ehditty testaamaan.

Windows 10 IoT Core on jatkuvasti kehittyvä alusta, johon lisätään uusia toimintoja jatkuvasti. Työssä tuli usein vastaan käyttöjärjestelmän puutteet. Nämä puutteet todennäköisesti tullaan tulevaisuudessa paikkaamaan.

Mikäli ohjelmien haluttaisiin toimivan internetin välityksellä, tulisi tietoturvaongelmat ratkaista lisäämällä kirjautuminen. Myös Apache- ja MySQL-palvelimen käyttöoikeudet tulisi muokata internettiin sopiviksi.

6.2 Yhteenveto

Yhteenvetona voidaan todeta, että Raspberry Pi on jatkuvasti kehittyvä alusta, joka mahdollistaa uusien toimintojen ja käyttöjärjestelmien käytön laitteella. Hinnaltaan laite on edullinen ja mahdollistaa helpon kehitysalustan useisiin tarkoituksiin. Lisäksi uusi Windows 10 IoT Core sallii entistä helpomman tavan käyttää Microsoftin teknologioita Raspberry Pi -laitteella.

Työ oli mielenkiintoinen, koska siinä pystyi tutustumaan uuteen ja kehittyvään Windows 10 IoT Core -käyttöjärjestelmään ja tämän soveltamiseen Raspberry Pi -laitteella. Työn ohessa tuli kerrattua PHP-ohjelmointia ja C#-ohjelmointia sekä SQL-lauseita. Lisäksi työssä tuli tutustuttua Universal Windows Appien kehitykseen ensimmäistä kertaa.

LÄHTEET

96boards. Ei Päiväystä. Dragonboard 410c Documentation, [www-dokumentti]. Linaro. [Viitattu 9.4.2016]. Saatavissa: www.96boards.org/products/ce/dragonboard410c/docs/

Apache Friends. Ei Päiväystä. XAMPP Apache + MariaDB + PHP + Perl. [www-dokumentti]. Apache Friends. [Viitattu 2.4.2016]. Saatavissa: <https://www.apachefriends.org/index.html>

Apache. Ei Päiväystä. http server project [www-dokumentti]. Apache Software Foundation. [Viitattu 10.4.2016]. Saatavissa: <https://httpd.apache.org/>

Arduino. Ei Päiväystä. Arduino UNO & Genuino UNO. [www-dokumentti]. Arduino. [Viitattu 27.3.2016]. Saatavissa: <http://www.arduino.cc/en/Main/ArduinoBoardUno>

Microsoft. Ei päiväystä. Develop UWP apps. [www-dokumentti]. Microsoft. [Viitattu 16.3.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/windows/uwp>

Microsoft. Ei päiväystä. Get started with Windows apps. [www-dokumentti]. Microsoft. [Viitattu 17.3.2016]. Saatavissa: <https://dev.windows.com/en-us/getstarted>

Microsoft. Ei päiväystä. Get Started with Windows IoT. [www-dokumentti]. Microsoft. [Viitattu 18.3.2016]. Saatavissa: <http://ms-iot.github.io/content/en-US/GetStarted.htm>

Microsoft. Ei päiväystä. Learn about Windows 10 IoT Core. [www-dokumentti]. Microsoft. [Viitattu 19.3.2016]. Saatavissa: <http://ms-iot.github.io/content/en-US/IoTCore.htm>

NetBeans. Ei Päiväystä. NetBeans IDE. [www-dokumentti]. NetBeans. [Viitattu 3.4.2016]. Saatavissa: <https://netbeans.org/>

PHP. 2016. PHP Manual. [www-dokumentti]. PHP Group. [Viitattu 10.4.2016]. Saatavissa: <http://php.net/manual/en/index.php>

Raspberry Pi foundation. Ei päiväystä. About us. [www-dokumentti]. Raspberry Pi Foundation. [Viitattu 1.2.2016]. Saatavissa: <https://www.raspberrypi.org/about/>

Raspberry Pi foundation. Ei päiväystä. Downloads. [www-dokumentti]. Raspberry Pi Foundation. [Viitattu 2.2.2016]. Saatavissa: <https://www.raspberrypi.org/downloads/>

Raspberry Pi foundation. Ei päiväystä. GPIO: Models A+, B+, Raspberry Pi 2 B and Raspberry Pi 3 B. [www-dokumentti]. Raspberry Pi Foundation. [Viitattu 3.2.2016]. Saatavissa: <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>

Raspberry Pi foundation. Ei päiväystä. Products. [www-dokumentti]. Raspberry Pi Foundation. [Viitattu 4.2.2016]. Saatavissa: <https://www.raspberrypi.org/products/>

Raspberry Pi foundation. Ei päiväystä. Raspberry Pi hardware. [www-dokumentti]. Raspberry Pi Foundation. [Viitattu 5.2.2016]. Saatavissa: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/>

Raspberry Pi foundation. Ei päiväystä. What is a Raspberry Pi?. [www-dokumentti]. Raspberry Pi Foundation. [Viitattu 6.2.2016]. Saatavissa: <https://www.raspberrypi.org/help/>

Sloss, A., Symes, D. & Wright, C. 2004. ARM System Developer's Guide. Burlington: Morgan Kaufmann

w3schools. Ei Päiväystä. PHP 5 MySQLi Functions. [www-dokumentti]. w3schools. [Viitattu 3.4.2016]. Saatavissa: http://www.w3schools.com/php/php_ref_mysqli.asp

XAMPP. Ei Päiväystä. About. [www-dokumentti]. Apache Friends. [Viitattu 10.4.2016]. Saatavissa: <https://www.apachefriends.org/about.html>

LIITTEET

Liite 1. Raspberry Pi -koodi

Liite 2. Raspberry Pi -ohjelman ulkonäkö koodi

Liite 3. PHP-datan vastaanottokoodi

Liite 4. PHP-datan näyttökoodi.

LIITE 1. Raspberry Pi koodi

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Runtime.InteropServices.WindowsRuntime;
using System.Threading.Tasks;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.Media.Capture;
using Windows.Media.MediaProperties;
using Windows.Storage;
using Windows.Storage.Streams;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Media.Imaging;
using Windows.UI.Xaml.Navigation;

namespace Oppari2
{
    public sealed partial class MainPage : Page
    {
        private MediaCapture mediaCapture;
        private StorageFile photoFile;
        private StorageFile recordStorageFile;
        private readonly string PHOTO_FILE_NAME = "photo.jpg";
        private bool isPreviewing;
        private bool isRecording;
        private DispatcherTimer timer;

        enum Action
        {
            ENABLE,
            DISABLE
        }

        private void SetInitButtonVisibility(Action action)
        {
            if (action == Action.ENABLE)
            {
                video_init.IsEnabled = true;
            }
            else
            {
                video_init.IsEnabled = false;
            }
        }

        public MainPage()
        {
            this.InitializeComponent();
        }
    }
}
```

```

        SetInitButtonVisibility(Action.ENABLE);

        isRecording = false;
        isPreviewing = false;
    }

    private async void Cleanup()
    {
        if (mediaCapture != null)
        {
            if (isPreviewing)
            {
                await mediaCapture.StopPreviewAsync();
                captureImage.Source = null;
                isPreviewing = false;
            }
            if (isRecording)
            {
                await mediaCapture.StopRecordAsync();
                isRecording = false;
            }
            mediaCapture.Dispose();
            mediaCapture = null;
        }
        SetInitButtonVisibility(Action.ENABLE);
    }

    private async void initVideo_Click(object sender, RoutedEventArgs e)
    {
        SetInitButtonVisibility(Action.DISABLE);

        try
        {
            if (mediaCapture != null)
            {
                if (isPreviewing)
                {
                    await mediaCapture.StopPreviewAsync();
                    captureImage.Source = null;
                    isPreviewing = false;
                }
                if (isRecording)
                {
                    await mediaCapture.StopRecordAsync();
                    isRecording = false;
                }
                mediaCapture.Dispose();
                mediaCapture = null;
            }
            mediaCapture = new MediaCapture();
            await mediaCapture.InitializeAsync();

            previewElement.Source = mediaCapture;
            await mediaCapture.StartPreviewAsync();
            isPreviewing = true;

            Task kuvat = new Task(async () =>
            {
                while (isPreviewing == true)

```

```

        {
            takePhoto();

            await Task.Delay(3000);
        } });
//takePhoto();

kuvat.RunSynchronously();
}
catch (Exception ex)
{
    status.Text = "Kameran käynnistys ei onnistunut: " + ex.Message;
}
}

private void cleanup_Click(object sender, RoutedEventArgs e)
{
    Cleanup();
}

private async void takePhoto()
{
    try
    {
        captureImage.Source = null;
        HttpContent stringContent = new StringContent("kuva");

        photoFile = await KnownFolders.PicturesLibrary.CreateFileAsync(
            PHOTO_FILE_NAME, CreationCollisionOption.ReplaceExisting);
        ImageEncodingProperties imageProperties = ImageEncodingPropert-
ties.CreateJpeg();
        await mediaCapture.CapturePhotoToStorageFileAsync(imageProperties,
photoFile);

        status.Text = "Kuvan otto onnistui: " + photoFile.Path;

        HttpClient client = new HttpClient();

        IRandomAccessStream photoStream = await photoFile.OpenReadAsync();
        BitmapImage bitmap = new BitmapImage();
        bitmap.SetSource(photoStream);
        captureImage.Source = bitmap;

        byte[] myPicArray = await GetBytesAsync(photoFile);
        client.BaseAddress = new Uri("http://10.0.1.6/Oppari/data.php");
        MultipartFormDataContent form = new MultipartFormDataContent();
        String base64 = System.Convert.ToBase64String(myPicArray);
        HttpContent bytesContent = new ByteArrayContent(myPicArray);
        HttpContent basecontent = new StringContent(base64);

        form.Add(bytesContent, "image");
        form.Add(basecontent, "base");

        HttpResponseMessage res = await client.PostA-
sync("http://10.0.1.6/Oppari/data.php", form);

        res.EnsureSuccessStatusCode();
    }
    catch { }
}

```

```
String content = await res.Content.ReadAsStringAsync();

if (res.IsSuccessStatusCode)
{
    status.Text = content;
}
else
{
    status.Text = "ei onnistunut";
}

}
catch (Exception ex)
{
    status.Text = ex.Message;
    Cleanup();
}
finally
{
}
}

}

public static async Task<byte[]> GetBytesAsync(StorageFile file)
{
    byte[] fileBytes = null;
    if (file == null) return null;
    using (var stream = await file.OpenReadAsync())
    {
        fileBytes = new byte[stream.Size];
        using (var reader = new DataReader(stream))
        {
            await reader.LoadAsync((uint)stream.Size);
            reader.ReadBytes(fileBytes);
        }
    }
    return fileBytes;
}

}

}
```

LIITE 2. Raspberry Pi ohjelman ulkonäkö koodi

```

<Page
  x:Class="Oppari2.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:Oppari2"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">

  <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
    <Grid.ChildrenTransitions>
      <TransitionCollection>
        <EntranceThemeTransition/>
      </TransitionCollection>
    </Grid.ChildrenTransitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="Auto"/>
      <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>

    <TextBlock Grid.Row="0" x:Name="pageTitle" Text="Oppari" Style="{StaticResource HeaderTextBlockStyle}"
      IsHitTestVisible="false" TextWrapping="NoWrap" HorizontalAlignment="Center" VerticalAlignment="Center"/>

    <StackPanel Grid.Row="1" x:Name="ContentRoot" HorizontalAlignment="Center" VerticalAlignment="Center" Orientation="Horizontal" Margin="12,0,0,0">
      <Button x:Name="video_init" Content="On" Click="initVideo_Click" HorizontalAlignment="Left" Margin="0,0,30,20" Width="180"/>
      <Button x:Name="cleanup" Content="Off" Click="cleanup_Click" Margin="0,0,30,20" Width="120"/>
    </StackPanel>

    <StackPanel Grid.Row="3" HorizontalAlignment="Center" VerticalAlignment="Center" Orientation="Horizontal" Margin="0,0,0,10">
      <StackPanel>
        <TextBlock Name="PreviewTitle" Grid.Row="0" Grid.Column="0" Text="Preview" Height="30" HorizontalAlignment="Center" Margin="0,20,0,0" Visibility="Visible"/>
        <Canvas Name="PreviewCanvas" Grid.Row="1" Grid.Column="0" Width="200" Height="200" Background="Gray" Margin="0,0,0,0" Visibility="Visible">
          <CaptureElement x:Name="previewElement" Width="200" Height="200" HorizontalAlignment="Left" Visibility="Visible"/>
        </Canvas>
      </StackPanel>
      <StackPanel>
        <TextBlock Name="ImageTitle" Grid.Row="0" Grid.Column="1" Text="Sended photo" Height="30" HorizontalAlignment="Center" Margin="0,20,0,0" Visibility="Visible"/>
        <Canvas Name="PhotoCanvas" Grid.Row="1" Grid.Column="1" Width="200" Height="200" Background="Gray" Margin="20,0,0,0" Visibility="Visible">
          <Image x:Name="captureImage" Width="200" Height="200" Visibility="Visible"/>
        </Canvas>
      </StackPanel>
    </StackPanel>
  </Grid>

```

```

        <StackPanel/>
        <StackPanel>
            <Canvas Grid.Row="1" Grid.Column="3" x:Name='playbackCanvas3'
Width='0' Height='0' Margin="0,0,0,0">
                <MediaElement x:Name='playbackElement3' Width="0" Height="0"/>
            </Canvas>
        </StackPanel>
    </StackPanel>

    <TextBlock Grid.Row="4" Text="status" HorizontalAlignment="Center" Verti-
calAlignment="Top" Margin="19,-10,0,0"/>
    <ScrollViewer Grid.Row="5" Margin="12,10,12,0" HorizontalAlignment="Center"
VerticalAlignment="Center" >
        <TextBox
                                x:Name="status" TextWrap-
ping="Wrap" IsReadOnly="True" Height="60" Width="400" HorizontalAlignment="Left"
VerticalAlignment="Top"
                                ScrollViewer.HorizontalScroll-
BarVisibility="Disabled" ScrollViewer.VerticalScrollBarVisibility="Auto"/>
    </ScrollViewer>
</Grid>
</Page>

```

LIITE 3. PHP datan vastaanotto koodi

```
<!DOCTYPE html>

<html>
<head>
  <meta charset="UTF-8" />
  <title></title>
</head>
<body>
  <?php
    $testi = "";

    if (isset($_POST))
    {
      $testi = $_POST['base'];

      $conn = new mysqli("localhost", "root","", "oppi");

      $conn->set_charset("utf8");

      $sql = "INSERT INTO kuvat (idkuvat , kuvat) VALUES(NULL, '$testi)";
      $sql1 = "SELECT * FROM kuvat";
      $result1 = $conn->query($sql1);
      $viimeinen = "";
      while ($row = mysqli_fetch_assoc($result1))
      {
        $viimeinen = $row['idkuvat'];
      }
      $viimeinen = $viimeinen - 3;
      $sql1 = "DELETE FROM kuvat WHERE idkuvat < $viimeinen";
      $result = $conn->query($sql1);

      $result1 = $conn->query($sql1);

    }
    $conn->close();

  ?>
</body>
</html>
```


LIITE 4. PHP Datan näyttö koodi.

```
<?php
```

```
$conn = new mysqli("localhost", "root", "", "oppi");
```

```
$conn->set_charset("utf8");
```

```
$id = 0;
```

```
$sql = "SELECT * FROM kuvat";
```

```
$result = $conn->query($sql);
```

```
while ($row = mysqli_fetch_assoc($result))
```

```
{
```

```
    echo '';
```

```
}
```

```
$conn->close();
```

```
?>
```