



TAMPEREEN AMMATTIKORKEAKOULU  
Tampere University of Applied Sciences

# **CODESYS ohjelmistoympäristö**

## **Ohjekirja Raspberry Pi:lle**

Atso Aalto

Opinnäytetyö  
Toukokuu 2017  
Sähkötekniikka  
Automaatio  
Älykkäät koneet

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Sähkötekniikka  
Automaatio / Älykkäät koneet

ATSO, AALTO:  
Codesys ohjelmistoympäristö  
Ohjekirja Raspberry Pi:lle

Opinnäytetyö 22 sivua, joista liitteitä 2 sivua  
Toukokuu 2017

---

Opinnäytetyön tarkoituksena oli tutustua Codesys-ohjelmistoympäristöön ja kirjoittaa sen käytöstä lyhyt ohjekirja käyttäen esimerkkinä Raspberry Pi:lle tehtävää esimerkkisovellusta. Ohjekirja tehtiin englanninkielisenä Seilo Consultingille harjoitusmateriaaliksi. Työssä tutustutaan Codesykseen ja Raspberryyn yleisemmin sekä myös yrityksiin niiden takana.

Kehitetyllä esimerkkisovelluksella ohjataan ja seurataan kesämökin valaistuksia, lämpötiloja ja lämminvesivaraajaa. Järjestelmää ohjataan etäältä verkkoselaimen kautta esimerkiksi kännykällä.

Laitteisto ja sovellus saatiin onnistuneesti rakennettua. Se on valmiina asennettavaksi kesämökkiin tai vastaavaan käyttöön. Laitteistoon ja ohjelmistoon on myös helposti tehtävissä tulevaisuudessa muutoksia ja/tai lisäyksiä.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Electrical engineering  
Automation / Intelligent machines

ATSO AALTO:  
Codesys software environment  
Manual for Raspberry Pi

Bachelor's thesis 22 pages, appendices 2 pages  
May 2017

---

The objective of this thesis was to get to know to Codesys software environment, and to write a short manual about its usage using example project made for Raspberry Pi as an example. Manual was made in English for Seilo Consulting as a training material. In this thesis, we are getting to know Codesys and Raspberry in general and companies behind them.

In example project we are controlling and supervising the lighting, temperatures and water heater. The system is controlled remotely via web browser for example with phone.

The project was successfully built and it is ready to be installed to summer cottage, or to some similar usage. It's also easy to add and/or modify equipment and sensors in project.

---

Key words: Raspberry Pi, Codesys, automation system, remote control

## SISÄLLYS

1JOHDANTO.....	6
2CODESYS.....	7
2.1.CODESYS OHJELMISTO .....	7
2.1.1VISUALISOINTI.....	9
3RASPBerry PI.....	11
3.1.RASPBerry PI PLUG-IN.....	13
4ESIMERKKISOVELLUS.....	15
5OHJEKIRJA.....	18
6POHDINTA.....	19
LÄHTEET.....	20
LIITTEET.....	21
Liite 1. Katkelma ohjekirjasta .....	21

## LYHENTEET JA TERMIT

PLC	Ohjelmoitava logiikka (Programmable logic)
CFC	Continuous Function Chart
FBD	Function Block Diagram
IL	Instruction List
LD	Ladder Logic Diagram
SFC	Sequential Function Chart
ST	Structured text
IO	Sisään ja ulostulo (Input/output)
I <sup>2</sup> C	Raspberryn käyttämä väyläteknikka
SPI	Tiedonvälitys-standardi tietokoneen ja oheislaitteen välille
OPC	Avoimen tiedonsiirron standardi
VPN	Virtuaalinen erillisverkko (Virtual Private Network)
GPIO	General purpose input/output
POU	Program organization units
HMI	Käyttöliittymä (Human Machine Interface)
PWM	Pulssisuhdemodulaatio (Pulse width modulation)
ADC	Analogisesta digitaaliseksi muunnin (Analog digital converter)
ESD	Elektrostaattinen purkaus (Electrostatic discharge)

# 1 JOHDANTO

Työn tutustutaan Codesys ohjelmistoympäristöön ja luoda sen käytöstä kompakti ohjekirja Seilo Consultingille. Seilo Consulting on automaatio- ja sähkötekniikan konsultointiyritys. Yrityksessä työskentelee kymmenisen työntekijää mutta se toimii maailmanlaajuisesti ja se on pääasiallisesti keskittynyt teollisuuteen ja voimantuotantoon. Seilo Consultingin kaikilla työntekijöillä on kokemusta ohjelmoitavista logiikoista, mutta harvemmillä on kokemusta Codesyksestä.

Tarkoituksena on luoda lyhyt ohjekirja Codesyksestä kahden esimerkin avulla. Jos tekee ohjekirjan tarjoamat esimerkit, ja on jo aikaisempaa kokemusta muista logiikoista, oppii helposti käytännön eroavaisuudet muihin logiikkoihin nähden.

Ohjekirjassa tulee olemaan käytännön esimerkkinä Raspberry Pi:llä toimiva yksikkö jolla voidaan netin välityksellä säätää ja monitoroida kesämökin lämpötiloja ja ohjata lämminvesivaraajaa. Ohjekirja on suunnattu ammattilaisille, joilla on jo kokemusta ohjelmoitavista logiikoista, mutta myös aloittelijoille siitä on varmasti apua.

Tässä dokumentissa käsitellään tarkemmin Codesyystä ja Raspberry Pi:tä. Kerron myös sovelluksen ja manuaalin teosta ja sisällöstä yleisellä tasolla. Manuaali itsessään ei tule olemaan yleisessä jaossa, mutta liitteenä on pieni katkelma siitä.

## **2 CODESYS**

Codesys on ohjelmistoympäristö, joka on suunniteltu täyttämään monien teollisten automaatioprojektien tarpeet. Codesys noudattaa standardia IEC 61131-3 joten se on hyvin yhteensopiva monien laitteistojen kanssa. Tänä päivänä yli 350 laitteistovalmistajaa ovat integroineet Codesyksen laitteistoihinsa. Codesyksen kehittäjä ja markkinoija on 3S-Smart Software Solutions. (Codesys, yritys, 2015)

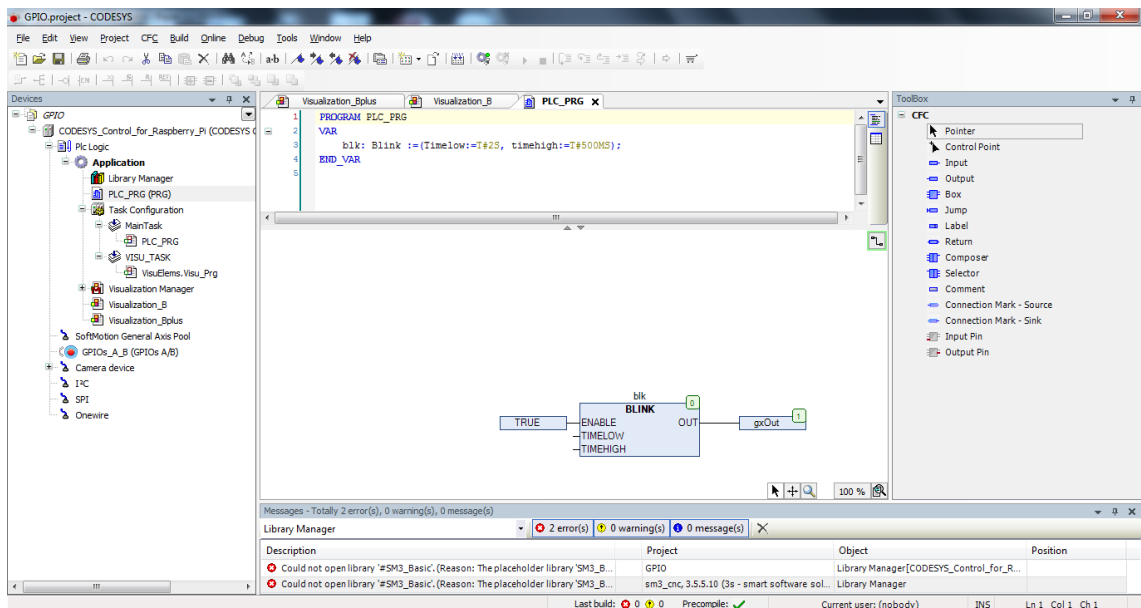
IEC (International Electrotechnical Commission) on tuloja tavoittelematon organisaatio, joka julkaisee standardeja ja hallinnoi vaatimuksenmukaisuuden arviointia. (IEC, yritys, 2017)

3S-Smart Software Solutions on vuonna 1994 perustettu saksalainen yritys. Yritystä johtavat sen perustajat Dieter Hess ja Manfred Werner. Firmalla ei tällä hetkellä ole muita tuotepereitä kuin Codesys. Nykyään reilut 100 työntekijää työskentelee 3S-Smart softwaresilla, jotka kehittävät ja marketoivat Codesystä. (Codesys, yritys, 2015)

Codesyksessä on integroituna monia toimintoja perinteisen PLC:n ohjelmointityökalujen lisäksi, kuten esim. visualisointi- ja väylätyökalut. Sopeutuvuus erinäisiin laitteistoihin on Codesyksen suurimpia vahvuuksia. Codesyksellä voi ohjata käytännössä melkein mitä tahansa systeemiä joissa käytetään tiedonsiirron standardeja kuten esim. OPC. Codesyksellä ei itsellään ole edes ole olemassa varsinaista laitteistoa, joten se luottaa kolmannen osapuolen laitteistoihin. (Codesys, järjestelmä, 2015)

### **2.1. CODESYS OHJELMISTO**

Codesyksen ohjelmointityökalua on melko helppo käyttää, jos on kokemusta muista PLC ohjelmointiohjelmista. Kuten kaikissa ohjelmissa niin myös Codesyksessä on omat yksityiskohtansa mitkä pitää opetella, mutta se on kokonaisuutena mielestäni helpompi oppia kuin esimerkiksi Siemensin Step 7. Laitteesta riippuen ohjelmiston ominaisuudet tietysti hieman muuttuvat ja tulee käyttöön erilaisia komentoja ja kirjastoja, mutta perusosat pysyvät samoina. Kuvassa 1. on näkymä esimerkkisovelluksen pääohjelmasta.



Kuva 1: I/O testi esimerkkisovellus

Codesys tukee kaikkia IEC 61131-3 standardin luettelemia ohjelmointikieliä:

- SFC Sequential Function Chart: Hajottaa ohjelman pienempiin osiin ja hallitsee kokonaisuutta. Päättää milloin muita ohjelman osia suoritetaan ja mahdollisesti ajaa niitä rinnakkain.
- FBD Function Block Diagram: Graafinen ohjelmointikieli jossa yhdistellään aritmeettisia, Boolean ja muita muuttujia yhteen Funktio blokkien kanssa. Puhtaasti Boolean muuttujia käyttävät osat pystytään yleensä kääntämään suoraan LD muotoon ja toisinpäin.
- LD Ladder Diagram: Graafinen ohjelmointikieli jossa yhdistellään Boolean muuttujia. LD näyttää relekaaviolta, jonka ansiosta sähkömiesten on varmaankin helppo päästä tähän kieleen käsiksi.
- IL Instruction List: Alatason ohjelmointikieli. Käytännössä kielessä kirjoitetaan komentoja allekkain ja kone tekee niitä yksi kerrallaan. IL on lähellä koneen ymmärtämää binäärimuotoista ”konekieltä”.
- ST Structured Text: Ylätason ohjelmointikieli. ST muistuttaa hieman C:tä ja ohjelmointia harrastaneet varmasti oppivat sen hyvin nopeasti.

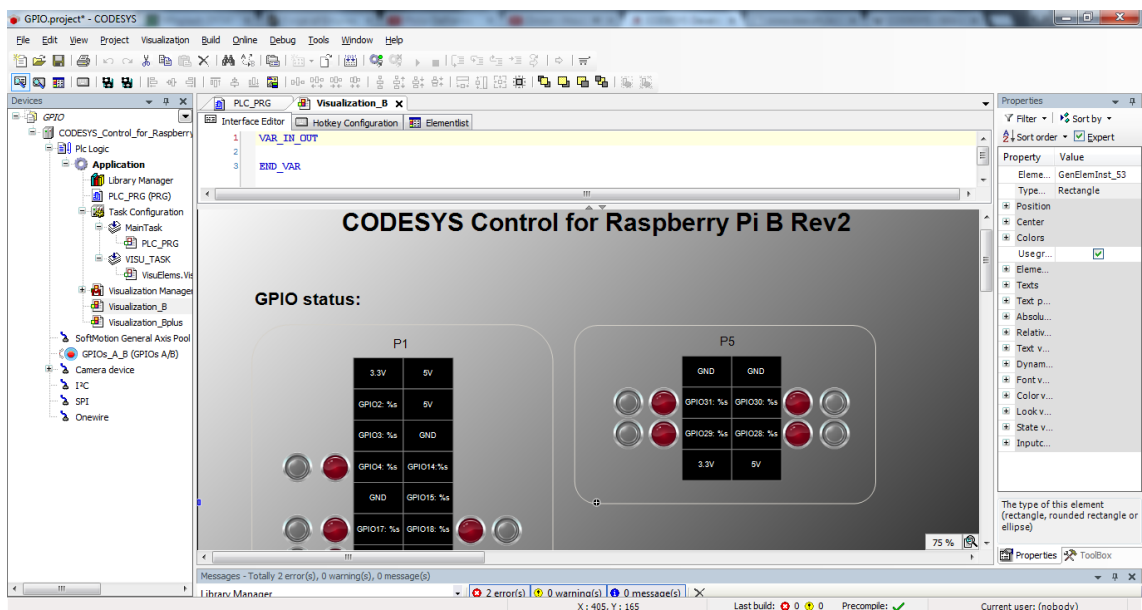
Lisäksi Codesys tukee myös yhtä kyseisessä standardissa määrittelemätöntä ohjelmointikieltä.



- CFC Continuous Function Chart: Vapaamuotoisempi FBD jossa sisääntulot ja ulosmenot eivät tule automaattisesti vaan tarvitsee asettaa manuaalisesti. Tämä mahdollistaa eri verkkojen yhdistelyä tarvitsematta käyttää sisäisiä apumuuttujia. (Codesys, kehitysjärjestelmä, 2015)

## 2.1.1 VISUALISOINTI

Codesyksen ohjelmisto tarjoaa myös ohjelmistoon integroidun työkalun visuaalisen käyttöliittymän tekoa varten. Kuvassa 2. on esimerkkisovelluksen käyttöliittymän editori.



Kuva 2: Esimerkkisovelluksen HMI

Ohjelmassa on valmiina useita erilaisia kytkimiä ja merkkivaloja sekä histogrammeja, joten graafisen asun saa halutessaan näyttämään suhteellisen hyvältä ilman suuria ponnisteluja.

Visualisointityökalun käyttö on helppoa: vedä haluttu objekti valikosta näytölle ja kerro objektille muuttuja mitä se seuraa. Erillisiä asetuksia ei tarvitse asetella. Esimerkiksi laittamalla kytkimen visualisointiin voidaan sillä ohjata suoraan jotain ulostuloa. Yhdistettäessä selaimella Raspberryn IP-osoitteeseen visualisointi-ikkuna avautuu automaattisesti, ellei erillistä salasana-kyselyä ole toiminnassa.

Yksinkertaisimmat ohjelmat joissa vain ohjataan suoraan ulostuloja ja mitataan sisääntuloja, voidaan suorittaa puhtaasti pelkällä visualisointityökalulla. Tämä varmasti

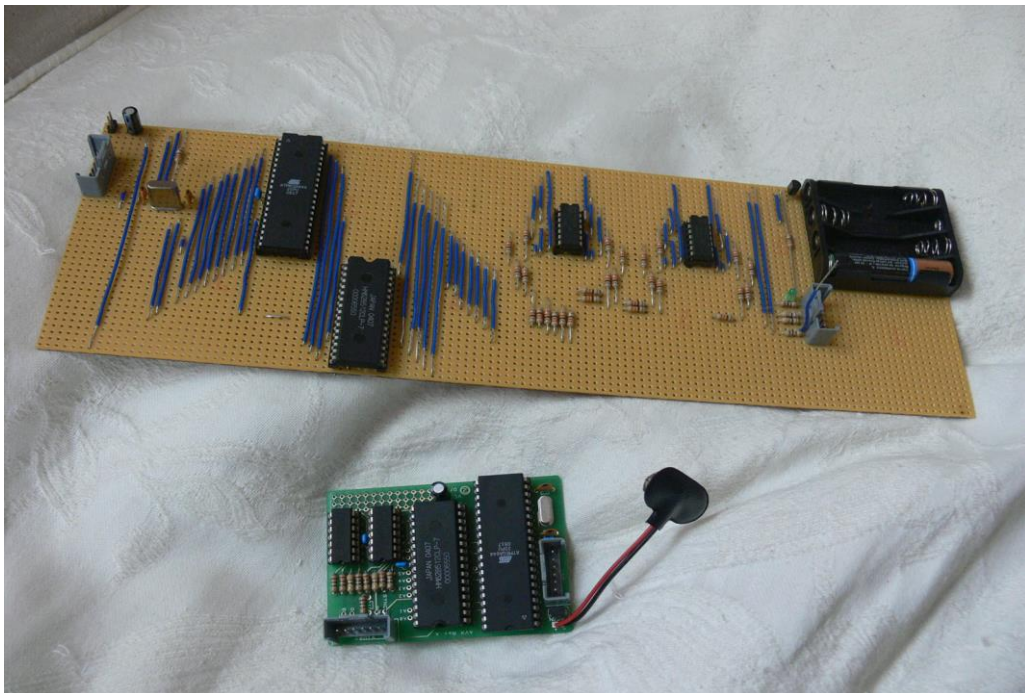
auttaa etenkin aloittelevia ohjelmoijia, sillä työkalun alottelijaystävällisyyteen on selvästi kiinnitetty paljon huomiota.

### 3 RASPBERRY PI

Raspberry Pi on käytännössä pankkikortin kokoinen tietokone. Alun perin se on luotu koulutuskäyttöön, mutta halpuutensa sekä avoimen lähdekoodinsa ansiosta se on erityisen suosittu harrastelijoiden parissa. (Raspberry Pi, Mikä on Raspberry Pi, 2015)

Säätiö Raspberryn takana on Raspberry Pi Foundation. Säätiö toimii Britanniassa ja se on rekisteröity koulutus ja hyväntekeväisyysjärjestö. Tavoitteikseen säätiö on määritellyt aikuisten ja lasten koulutuksen edistäminen, etenkin tietokoneiden ja tieteen alalla. (Raspberry Pi, Mikä on Raspberry Pi, 2015)

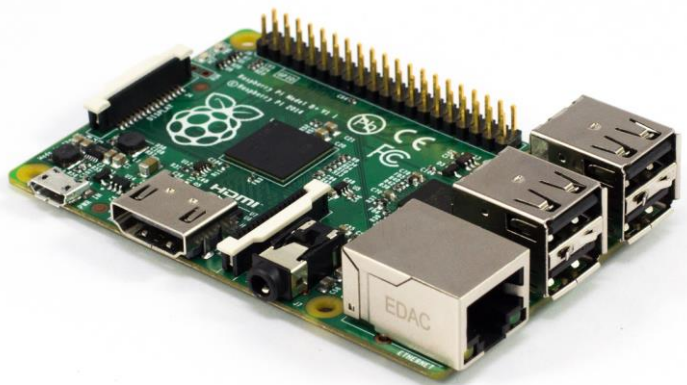
Ajatus Raspberrystä syntyi vuonna 2006 Cambridgen yliopistossa, kun Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft huolestuivat vuosittaisesta oppilaiden taitojen huononemisesta tietokoneluokassa. Ryhmä päätti tehdä laitteen joka mahdollistaisi tietokoneiden opiskelun ilman suuria sijoituksia. Vuodesta 2006 vuoteen 2008 ryhmä valmisti useampia prototyyppejä (kuva 3). (Raspberry Pi, Meistä, 2015)



*Kuva 3: Raspberry Pi:n ensimmäisiä prototyyppejä (Raspberry, prototyyppi, 2015)*

Vuonna 2008 ryhmä lyöttäytyi yhteen Pete Lomasin ja David Brebenin kanssa perustaakseen säätiön. Kolme vuotta myöhemmin Raspberry Pi:n malli b:tä alettiin massatuottaa. (Raspberry Pi, Meistä, 2015)

Raspberry toimitetaan normaalisti ilman mitään lisätarvikkeita, joten yleensä kannattaa hankkia samaan aikaan ainakin virtajohto sekä muistikortti. Nykyään Raspberryjä on useaa eri versiota ja uusia kehitetään koko ajan. Tässä työssä käytetty malli on B+ (kuvassa 4).



*Kuva 4: Sovelluksessa käytetty Raspberryn versio B+  
(Raspberry, B+ 2015)*

Mallissa B+ on 40 GPIO:ta ja neljä USB 2.0 liityntää sekä HDMI-liitin. Kyseinen Raspberryn versio kuluttaa 0,5W – 1W sähköä. Kaikki I/O:t ovat digitaalisia ja toimivat 3,3v jännitteellä. Analogisen I/O:n puuttuminen monimutkaistaa monia projekteja ja useasti vaatii ulkoisen muuntajan hankkimista, joka muuntaa analogiatiedon digitaalisignaalksi.

### **3.1. RASPBERRY PI PLUG-IN**

Raspberry Pi toimitetaan ilman käyttöjärjestelmää. Codesystä varten on tehty Raspberryille oma käyttöjärjestelmä, jonka voi ladata suoraan Raspberryn kotisivuilta 50€ hintaan. Käyttöjärjestelmä on Linux pohjaisen Rasbian käyttöjärjestelmän Codesystä varten mukautettu versio. Tuotteen pystyy myös lataamaan kokeiluversiona, joka sammuu itsestään kahden tunnin kuluttua käynnistyksestä mutta toimii muuten täysin samalla tavalla kuin kokoversio.

Käyttöjärjestelmän asentaminen Raspberryyiin on tehty äärimmäisen helpoksi. Sinun tarvitsee vain siirtää ladattu käyttöjärjestelmä micro SD kortille ja laittaa kortti kiinni Raspberryyiin. Tämän jälkeen tietokoneen Codesys ohjelma tunnistaa samassa verkossa olevan Raspberryn ja voit käyttää sitä kuin mitä tahansa ohjelmoitavaa logiikkaa.

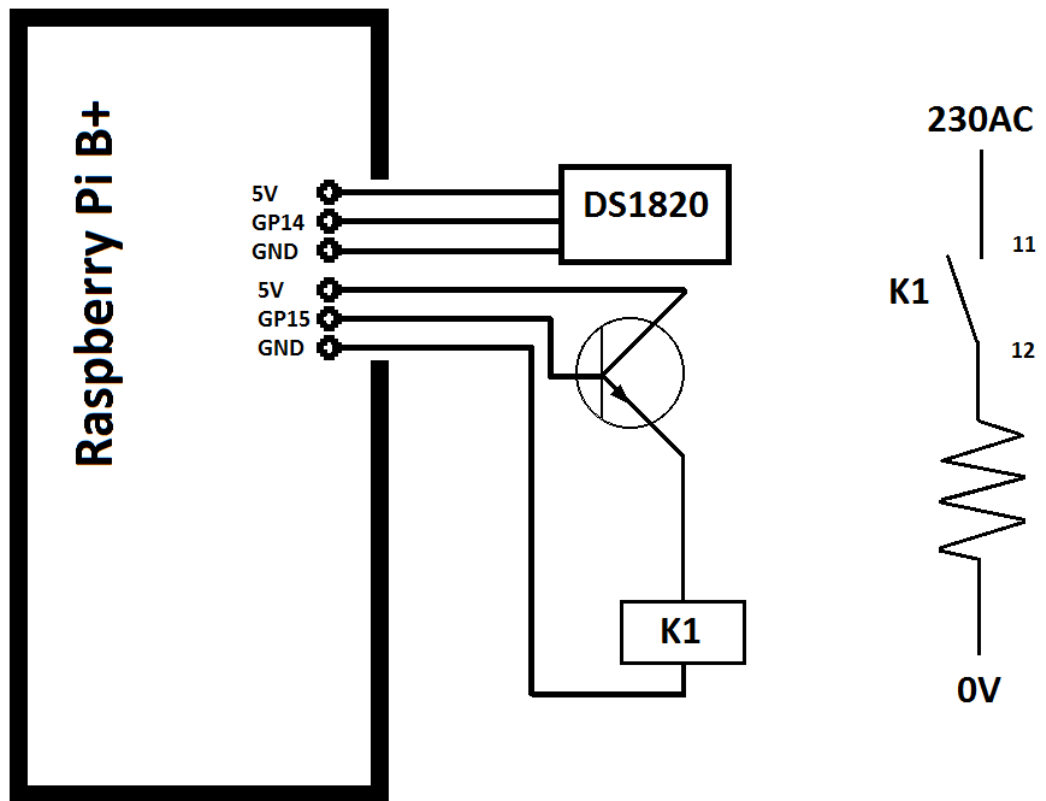
Itse Codesykseen kannattaa lisäksi asentaa Raspberryn kirjastot, jotta pystyy käyttämään kaikkia Raspberryn ominaisuuksia helpommin. Käytännössä Raspberryn kirjastot tuovat uusia komentoja Codesykseen, ja auttavat määrittämään liitetyjä laitteita helpommin. Esimerkiksi esimerkisovelluksessa käytetyn yksijohdinlämpötila-anturin määrittäminen on huomattavasti helpompaa.

Käyttöjärjestelmän mukana tulee jonkin verran kirjastoja ja netistä pystyy ilmaiseksi lataamaan useiden muiden laitteiden kirjastoja. Kirjastot tulevat seuraaville laitteille:

- Raspberry Pi
- Raspberry PiFace Digital (lisäosa jossa releitä ja ledejä ja I/O:ta)
- Raspberry PiFace Control&Display (lisäosa jossa näyttö ja nappuloita)
- Raspberry Pi Camera (Raspberryn virallinen kamera)
  - I<sup>2</sup>C laitteet
- SRF02 (ultraäänietäisyysmittari)
- Adafruit PWM (16-kanavainen 12-bittinen PWM)
- MPU6050 (liikkeenjäljittäjä)
- MPU9150 (liikkeenjäljittäjä)
- AK8975 (elektroninen kompassi)
  - SPI laitteet
- MCP3008 (8-kanavainen 10-Bittinen ADC)
- MCP23S17 (16-bittinen GPIO)
  - DS18B20 (yksijohdin digitaalilämpömittari)

## 4 ESIMERKKISOVELLUS

Sovelluksen tarkoituksena on hallita kesämökkiä etäältä netin välityksellä. Raspberryyn kiinnitetään moka, jonka avulla se on yhteydessä internetiin. Nettiselaimella ja salasanaalla pystytään yhdistämään Raspberryn käyttöliittymään, josta pystytään tarkkailemaan sekä säätämään mökin lämpötiloja sekä asettamaan lämminvesivaraaja päälle. Kuvassa 5 on esimerkkisovelluksen kytkentäkaavio.



*Kuva 5: Esimerkkisovelluksen kytkentäkaavio*

Jälkikäteen järjestelmään voidaan lisätä lisää toimintoja, esimerkiksi liiketunnistimen tai kameran lisäämisellä saadaan murtohälytin. Lisäksi valojen kauko-ohjauksen lisäys onnistuu helposti laitteistoon.

Lämpöanturina tässä sovelluksessa toimi Dallaksen ds1820 joka on yksijohdinlämpömittari (kuva 5). Valitsin tämän anturin koska se oli halvin mahdollinen 5v toimiva digitaalinen anturi jonka löysin. Analogisen anturin valinta olisi vaatinut erillisen A/D muuntajan ja turhaan monimutkaistanut kytkentää. Kyseiseen anturiin löytyy suoraan mukana tulleista kirjastoista valmiita komentoja.



*Kuva 6: Sovelluksessa käytetty yksijohdinanturi (Ds1820, yksijohdinlämpömittari 2015)*

Rasperryn ulostulot antavat 3,3v ja lämminvesivaraaja vaatii verkkovirtaa, joten kontaktorin käyttö on pakollista. Kontaktorina tässä sovelluksessa toimi 5vDC/230vAC. Kontaktori liipaistiin transistorilla, joka oli kytkettynä Rasperryn uloslähtöön.

Rasperry ja sitä ohjaava päätelaite eivät ole samassa verkossa. Tämän takia joudutaan käyttämään VPN:nä. VPN on tapa yhdistää kaksi verkkoa yhteen internetin välityksellä siten, ettei kukaan pääse kopeloimaan sitä. Ohjekirjassa tai tässä dokumentissa ei ole tarkemmin kerrottu miten kyseinen järjestely saadaan toimimaan koska jokaisella reitittimellä on omat asetuksensa mitä pitää säätää. Tavallisesti kuitenkin reitittimen ohjekirjoissa on yksityiskohtaisesti selvitetty, miten VPN saadaan toimintaan.

Sovelluksen tekeminen onnistui muutaman tunnin YouTuben katselun ja help-tiedostojen lukemisen jälkeen melko kivuttomasti. Koska Rasperry on hyvin suosittu harrastelijoiden parissa, löytyy internetistä lukemattomat määrät tietoa aiheesta. Jos kohtaa jonkin ongelman on hyvin todennäköistä, että joku muukin on saman ongelman kohdannut, ja siitä foorumeilla kirjoittelee. Lisäksi huomasi, että jos ei välttämättä halua tai osaa tehdä sovellusta aivan alusta alkaen itse, löytyy netistä myös huomattavan helposti sovelluksia joista muokkaamalla saa oman näköisensä helposti. (YouTube, Codesys oppitunti, 2015)



Lämpöanturi tuntui olevan melko herkkä ja ensimmäistä en saanut ollenkaan toimimaan. Luultavammin anturi kärehti väärin kytkentöjen takia, tai ESD purkauksen seurauksena. Koska kyseessä oli vain kokeiluluontoinen ja väliaikainen kytkentä, käytin johtimia joiden päissä oli valmiit liittimet.

Itse Raspberry Pi maksoi 30€ ja lämpöanturit maksoivat 6€. Kontaktorin hinnaksi tuli 2€ ja transistori kustansi 1€, tämän lisäksi hieman johtimia ja tinaa joiden yhteishinta noin 3€. Koska Raspberryn lisensoitu käyttöjärjestelmä kustansi 50€, kokonaishinnaksi asennukselle tuli vajaa 100€. Tietysti lisäksi tarvitsee hankkia toimiva nettiliittymä kohteeseen joka saattaa vaatia morkkulan ostamista. Pitempää käyttöä varten kannattaa tietysti myös harkita kunnollisen laatikon hankkimista tai tekemistä osien suojelun takia.

Raspberryn ostin verkkokauppa.com:ista ja käyttöjärjestelmän suoraan Codesyksen kotisivuilta. Loput osat ostin Elektorilta Tampereelta, koska se sattui olemaan lähin elektroniikkaliike. Jos hieman tarkemmin katsoisi mistä tavarat ostaa ja tekisi hintavertailuja laitteiston kokonaishintaa saisi varmaankin laskettua noin kymmenellä eurolla.

## 5 OHJEKIRJA

Ohjekirja on suunniteltu insinööreille, joilla on jo valmiiksi kokemusta ohjelmoitavista logiikoista. Sen on tarkoitus olla kompakti ja se selittää perusasiat Codesyksen ja Raspberryn käytöstä kahden esimerkkisovelluksen avulla. Tavoitteena on, että ohjekirjan lukija ymmärtää lukemisen jälkeen, miten perustoiminnot toimivat Codesyksessä. Manuaali on kirjoitettu englanniksi työnantajan toiveesta. Liitteessä 1 on katkelma ohjekirjasta.

Vaikka ohjekirja onkin lyhyt, ja siinä käydään vain perustoimintoja lävitse, se kertoo selvästi suurimmat erot käyttötavoissa ja perusterminologioissa muihin logiikkaohjelmiin verrattuna.

Käytännössä ohjekirjassa kerrotaan kädestä pitäen miten muuttujat toimivat ohjelmistossa ja miten niitä saadaan liitettyä ulostuloihin ja sisääntuloihin. Lisäksi siinä kerrotaan lyhyesti, miten visuaalista puolta ohjelmoidaan, jotta saadaan esimerkiksi kännykällä yhdistettäessä näkyviin verkkosivu, jossa on lämpötila ja päälläolotietoja valoilta ja lämmittimiltä, sekä pystytään ohjaamaan kyseisiä kohteita. Lisäksi näytetään, miten yhdistetään Raspberryn, ja miten ohjelmisto ladataan sinne.

Ohjekirjan kirjoittaminen englanniksi onnistui helpommin kuin uskoinkaan. Kun kaikki termit ovat jo valmiiksi samalla kielellä niin ei tarvitse erikseen mietiskellä mitään käännöksiä.

## 6 POHDINTA

Alku oli hankala niin kuin monessa työssä yleensäkin. Jouduin opettelemaan uuden ohjelmiston käytön ja lisäksi käyttämään sitä laitteessa jota en ollut koskaan aikaisemmin edes nähnyt.

Huomasin että helpoin tapa päästä sisään uuteen ohjelmistoon on katsoa videoita YouTubesta jotka kertovat perusasiat. Tämän jälkeen on huomattavasti helpompaa opetella itseksensä ohjelmistoa, kun perusteet on jo hallinnassa. Onnekseni myös help-tiedostot olivat perusteellisesti tehty.

Minulla on jonkin verran kokemusta ohjelmoitavista logiikoista työni puolesta, joten osaan sanoa, että Codesys on huomattavasti alottelijaystävällisempi. Näin lyhyellä käyttömäärällä on tietenkin hieman vaikeaa kertoa, pystyykö Codesyksellä tekemään kaikkia samoja asioita kuten esimerkiksi Schneiderin Unity Pro:lla.

Kokonaisuutena minusta tuntuu, että opinnäytetyö onnistui kohtuullisesti ja saavutin asetetut tavoitteet, joskin hitaasti. Suurimmat vaikeudet tuntuivat tulevan tämän dokumentin kirjoittamisen kanssa. Motivaation löytäminen kirjoittamaan turhanpäiväistä tekstiä vain saadakseni paperin ulos koulusta tuntuu hieman vanhanaikaiselta ja turhalta.

Opinnäytetyötä tehdessä olen oppinut paljon uutta etenkin Codesyksestä ja Raspberrystä, mutta myös tiedon hakemisesta. Opetellessaan itse koko ohjelmiston oppii hakemaan tietoja internetin erinäisistä kolkista kuten foorumeilta ja YouTubesta.

Tulevaisuudessa tulen varmasti itselleni ostamaan oman Raspberry Pi:n ja jatkamaan sillä näpertelyä. Olen yllättynyt, että TAMK:ssa ei ole Raspberrystä käytössä, ainakaan kursseilla joilla minä olen ollut. Voin lämpimästi sitä suositella etenkin koulutuskäyttöön.

## LÄHTEET

Codesys. Yritys. Luettu 28.7.2015. <http://www.codesys.com/company.html>

IEC. Yritys. Luettu 20.2.2017. <http://www.iec.ch/about/activities>

Codesys. Järjestelmä. Luettu 16.7.2015. <http://www.codesys.com/the-system.html>

Codesys. Kehitysjärjestelmä. Luettu 28.7.2015.

<https://www.codesys.com/products/codesys-engineering/development-system.html>

Raspberry Pi. Mikä on Raspberry Pi. Luettu 30.9.2015.

<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

Raspberry Pi. Meistä. Luettu 30.9.2015. <https://www.raspberrypi.org/about/>

Raspberry Pi. Prototyyppi. <https://www.raspberrypi.org/wp-content/uploads/2011/10/atmel1.jpg>

Raspberry Pi. B+ <https://btcstore.eu/sites/default/files/products/F0002%20-%20Raspberry%20Pi%20model%20B%2B%20512%20MB%20%2B%208GB%20NO%20OBS.jpg>

Ds1820. Yksijohdinanturi.

[https://developer.mbed.org/media/cache/components/components/18b20.jpg.200x200\\_q85.jpg](https://developer.mbed.org/media/cache/components/components/18b20.jpg.200x200_q85.jpg)

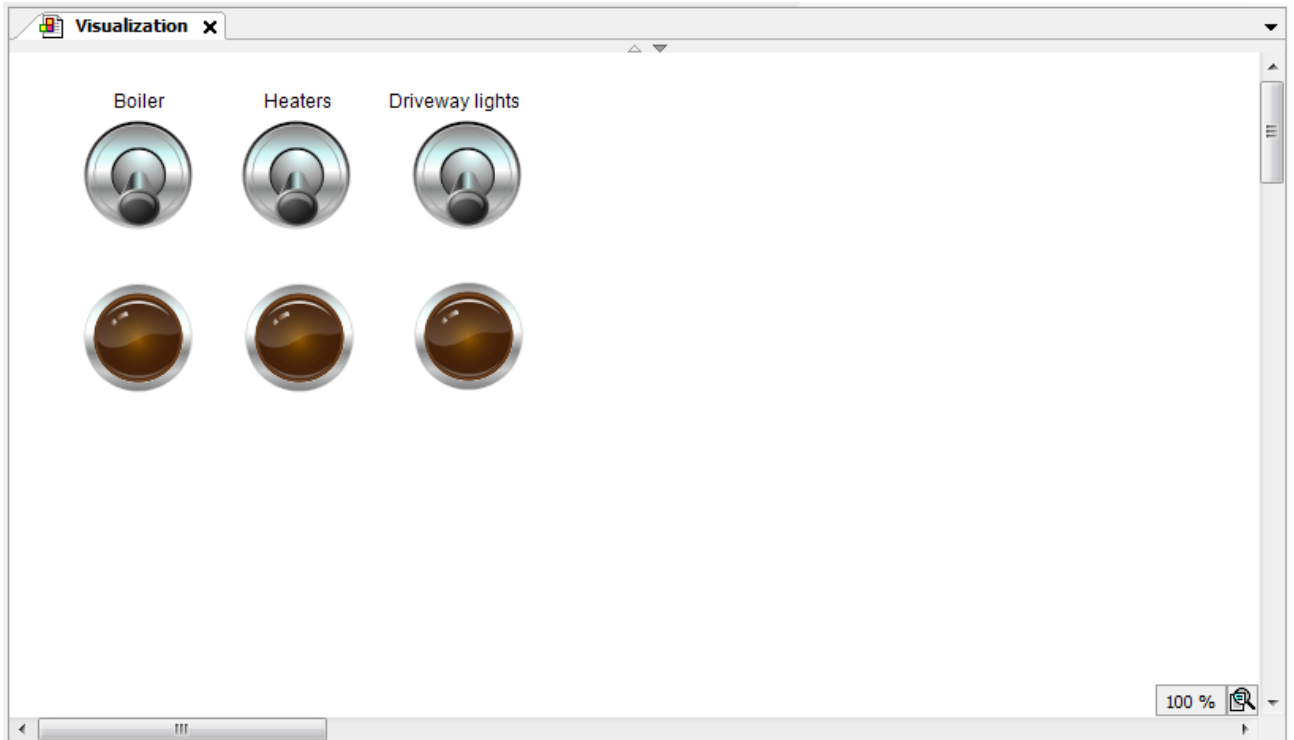
YouTube, Codesys oppitunti

[https://www.youtube.com/watch?v=2tX6gumm2zg&list=PLskTKxU53xsWIILK\\_S3qW-ttTADGDFL6J](https://www.youtube.com/watch?v=2tX6gumm2zg&list=PLskTKxU53xsWIILK_S3qW-ttTADGDFL6J)

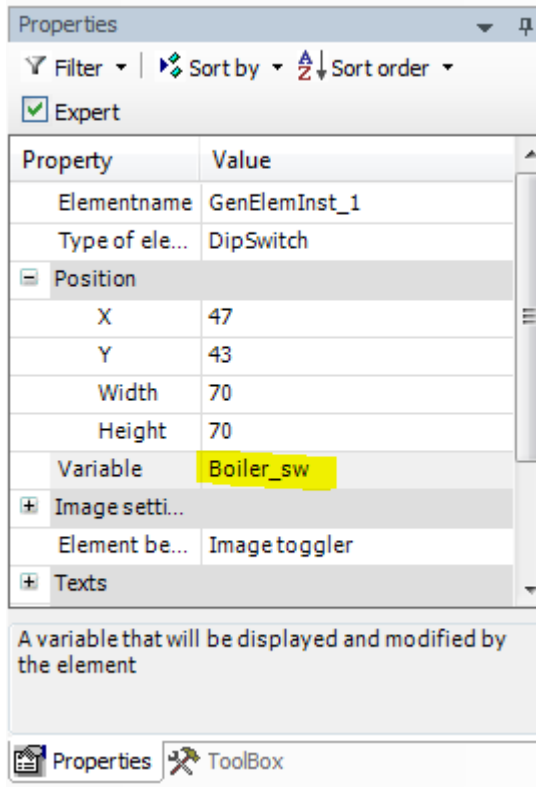
## LIITTEET

### Liite 1. Katkelma ohjekirjasta

Select Visualization and add 3 switches and tree lamps by dragging and dropping from right side window. Also, put some labels on them from same menu.



Select one of the switches and add right variable from menu on the right.



Do the same for rest of the switches and lamps.

Add these switches to inputs 4, 17, and 18 same way as in example 1. Then add lamps to inputs 22, 23, and 24.

Next, we are going to setup one wire temperature sensor ds 1820. This is not so simple because we have to first find the id of the sensor. Go to PLC\_PRG and add following code:

```
IF NOT X THEN
```

```
    X := TRUE;
```

```
    ID := Onewire_master.pasScannedDevices^[0].StringRepresentation;
```

```
END_IF;
```

If and x are simply used because we want to do this only once. Line starting with ID gets the unique id from first sensor found and saves it to variable ID.