



# IoT-tekniikan hyödyntäminen opetusalustassa

Kevin Hauger

OPINNÄYTETYÖ  
Kesäkuu 2019

Kone- ja tuotantotekniikka  
Koneautomaatio

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Kone- ja tuotantotekniikka  
Koneautomaatio

HAUGER, KEVIN:  
IoT-tekniikan hyödyntäminen opetusalustassa

Opinnäytetyö 43 sivua, joista liitteitä 26 sivua  
Kesäkuu 2019

---

Tämä opinnäytetyö toteutettiin Tampereen ammattikorkeakoulun kone- ja tuotantotekniikan insinöörikoulutusohjelman tueksi. Työn tavoitteisiin kuului opetusalustan luominen opiskelijoille esineiden internetiä hyödyntämällä. Alustan tarkoituksena on opettaa käytännön ja harjoitusten kautta IoT:sta ja sen mahdollisuuksista. Lisäksi työstä oli tarkoitus luoda käyttöohjeet harjoitustehtävien muodossa, jotta työn voisi toteuttaa uudestaan ja lisätä opetukseen.

Opetusalusta oli jo olemassa ja käytössä oppilaitoksen opetuksessa. Alustassa on ohjelmoitava liukuhihna sekä konenäkökamera, joka tarkkailee hihnaa pitkin kulkevia esineitä. Tavoitteena opinnäytetyössä oli kytkeä kokonaisuus internetiin ja sitä kautta ohjata sekä valvoa laitteita etäyhteydellä. Laitteiden operointi oli tarkoitus toteuttaa langattomasti verkon välityksellä joko älypuhelimella tai tietokoneella.

Opinnäytetyössä tutustutaan IoT-määritelmään ja sen teknologiaa kehittäviin vaikutuksiin. Työssä perehdytään IoT-tekniikkaan ja toimintaan yksityisellä ja teollisella sektorilla. Lisäksi tutustutaan kehitettävän opetusalustan rakentamiseen ja toimintaan sekä sen ohjelmointityökaluihin. Internetin lisäys alustaan sekä sen ohjelmointi ja operointi on selitetty tarkasti harjoitustehtävissä, jotta opiskelijat voisivat toteuttaa prosessin uudestaan.

Työn tuloksena on etäyhteydellä toimiva opetusalusta. Alustan operointi on mahdollista älypuhelimella tai tietokoneella. Alustan liukuhinnan pystyy käynnistämään, sammuttamaan sekä valitsemaan sen pyörimissuunnan. Lisäksi on luetavissa, onko hihna käynnissä vai sammuksissa. Laskurin avulla voi tarkkailla konenäkökameran tunnistamia liukuhihnaa pitkin kulkevia kappaleita.

Valmis opetusalusta on tarkoitus lisätä kone- ja tuotantotekniikan koulutusohjelmaan. Se lisää koulutuksen modernin opetuksen tarjontaa ja opettaa käytännön kautta IoT-tekniikasta. Oppilaitokselle on luovutettu yksityiskohtaisemmat ohjeet laitteiston toiminnasta kolmen harjoitustehtävän muodossa. Harjoitustehtävien avulla opiskelijat voivat tutustua alustan etävalvontaan ja -ohjaukseen.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Mechanical and Production Engineering  
Machine Automation

HAUGER, KEVIN:  
Implementing IoT-Technology in a Teaching Platform

Bachelor's thesis 43 pages, appendices 26 pages  
June 2019

---

This thesis was carried out for the Degree Programme in Mechanical and Production Engineering of Tampere University of Applied Sciences. The purpose of the thesis was to introduce students to IoT technology by adding Internet to a university's teaching platform and creating a number of exercises about the subject.

The teaching platform consists of a programmable conveyor belt and machine vision camera which monitor objects that pass by on the conveyor belt. The goal of adding Internet to the system was to allow remote access to the platform. The platform could then be operated with a smartphone or a computer.

The thesis deals with IoT and its impact on technology in the private and industrial sectors. It also includes the implementation of Internet to the teaching platform and how it is operated and programmed. The exercises contain detailed instructions so that everything created in the thesis can be remade by the students.

The final product is a remote accessed conveyor belt and machine vision camera. The conveyor belt can be activated, shut down and its direction of rotation switched. It is possible to also monitor the activity and state of the belt. A counter allows the following of how many objects the machine vision camera has recognized passing by on the belt.

---

Key words: iot, iiot, internet of things, industrial internet of things

## SISÄLLYS

1	JOHDANTO .....	6
2	IOT JA IIOT .....	8
	2.1 Esineiden internet .....	8
	2.2 Teollinen internet.....	9
	2.3 IoT:n toimintaperiaate .....	10
	2.4 IoT-käyttöjärjestelmä .....	12
	2.5 Esineiden- ja teollisen internetin käyttökohteet .....	13
3	OPETUSALUSTAN SUUNNITTELU.....	16
	3.1 Opetusalustan vaatimukset.....	16
	3.2 Opetusalustan rakenne .....	16
	3.3 Tarvittavat komponentit.....	18
	3.4 Ohjelmointiympäristö.....	19
	3.4.1 TIA Portal.....	19
	3.4.2 IoT Studio .....	20
	3.4.3 In-Sight Explorer.....	20
4	LIUKUHIHINAN TOIMINTA .....	22
	4.1 Liukuhinnan käyttöönotto ja sen tavoitteet .....	22
	4.2 Liukuhinnan etävalvonnan toteutus.....	22
	4.3 Liukuhinnan etäohjauksen toteutus.....	27
5	KONENÄKÖKAMERAN TOIMINTA.....	33
	5.1 Kameran käyttöönotto ja sen tavoitteet .....	33
	5.2 Kameran etävalvonnan toteutus.....	33
6	VALMIS OPETUSALUSTA JA SEN TOIMINNOT.....	38
7	OPETUSALUSTAN HARJOITUSTEHTÄVIEN LAATIMINEN .....	39
8	POHDINTA .....	40
	LÄHTEET .....	42
	LIITTEET.....	44
	Liite 1. Liukuhinnan etävalvontaharjoitus .....	44
	Liite 2. Liukuhinnan etäohjausharjoitus .....	51
	Liite 3. Konenäkökameran etävalvontaharjoitus.....	60

**LYHENTEET JA TERMIT**

flow	IoT Studiossa nodeista koostuva komentorivi
gateway-laite	kahden tietoverkon yhdistävä laite
IoT	Internet of Things eli esineiden internet
IIoT	Industrial Internet of Things eli teollinen internet
node	IoT Studiossa käytettävät komennot

## 1 JOHDANTO

Internet of Things eli esineiden internet tarkoittaa verkkoon kytkettyjä esineitä, laitteita ja ajoneuvoja. Yksinkertaisimmillaan on kyse sensoreilla kerättävästä mitaustiedosta, mutta IoT:ta käytetään myös automaatiassa sekä monimutkaisten reaaliaikaisten prosessien analysoinnissa. Esineiden internetissä fyysiset laitteet liitetään antureilla ja sensoreilla verkkoon, mikä mahdollistaa niiden ympäristön aistimisen, viestinnän sekä toimimisen käsitellyn tiedon perusteella.

Kone- ja tuotantotekniikan opiskelijoita on tarkoitus tutustuttaa IoT-tekologiaan eli esineiden internetiin. IoT:n leviäminen on ollut rajussa nousussa viimeisen vuosikymmenen aikana. Tampereen ammattikorkeakoulun kone- ja tuotantotekniikan opetuksesta on puuttunut kokonaan IoT. Opetukseen halutaan lisätä se, jotta koulutus pysyisi mahdollisimman modernina. Koulu haluaa tehdä sen modernisoimalla liukuhihna-alustaansa kytkemällä sen internetiin. Tämä mahdollistaa konenäkökameran ja liukuhinnan ohjelmoinnin mobiililaitteella tai tietokoneella langattoman verkon välityksellä.

Opinnäytetyön tarkoitus on kehittää eteenpäin Tampereen ammattikorkeakoulun vanhaa opetusalustaa. Alusta on ohjelmitava liukuhihna, joka on varustettu konenäkökameralla. Kamera voidaan ohjelmoida tunnistamaan liukuhihnaa pitkin kulkevien esineiden tiettyjä ominaisuuksia, kuten muotoa tai väriä. Opetusalusta on ollut käytössä osana kone- ja tuotantotekniikan insinöörikoulutusohjelmaa.

Tämän opinnäytetyön tavoitteena on tutustua esineiden internetiin, ottaa se käyttöön opetusalustaan sekä luoda siitä käyttöohjeet harjoitustehtävien muodossa opiskelijoille. Harjoitustehtävien avulla tulee pystyä etänä valvomaan ja ohjaamaan liukuhihnaa. Lisäksi konenäkökameran lähettämä tieto liukuhihnaa pitkin kulkevista kappaleista tulee olla luettavissa langattoman verkon välityksellä. Opinnäytetyössä tutustutaan IoT-käsitteeseen, työn suunnitteluun, toteutukseen ja lopputuloksiin. Teoriaosuudessa tutustutaan IoT:n merkitykseen, toimintaperiaatteeseen ja käyttökohteisiin. Työn suunnitteluosuudessa on eritelty vaatimukset, komponentit ja ohjelmointityökalut. Opinnäytetyön fyysinen toteutus ja

lopputulokset ovat jaettu erikseen liukuhinnan ja konenäkökameran osuuksiin.  
Lopuksi on kerrottu lyhyesti harjoitustehtävien laadinnasta.

## 2 IOT JA IIOT

### 2.1 Esineiden internet

IoT (Internet of Things) eli suomennettuna esineiden internet tarkoittaa laitteiden tai koneiden liittämistä internetverkkoon. IoT-laitetta voi valvoa tai ohjata etänä useimmiten pilvipalvelun kautta. Melkein jokaisen laitteen tai koneen pystyy nykypäivänä liittämään verkkoon. (Meola 2018.)

IoT:ssa data kulkee fyysisestä maailmasta digitaaliseen internetin kautta. Fyysiseen puoleen kuuluvat koneet ja laitteet sekä niiden anturit ja päätelaitteet. Digitaalinen osuus puolestaan koostuu laitteiden antureista siirtyvästä datasta, pilvialustoista sekä ohjelmistoista ja käyttöliittymistä. Fyysisen ja digitaalisen yhdistelmä tekee laitteista älykkäitä. (Collin & Saarelainen 2016, 15.)

IoT:n historia ulottuu vuoteen 1999, kun Kevin Ashton (2009) ehdotti termiä ”Internet of Things”. Termi esiintyi ensimmäisen kerran hänen esitelmässään työpaikallaan Procter & Gamblella, jossa hän ehdotti RFID-tekniikan liittämistä työpaikan toimitusketjuun. (Ashton 2009.)

Kevin Ashton kuvaa Smithsonian-lehden haastattelussa (2015) IoT:n ”silmien ja korvien” verkkona tietokoneille. 1900-luvulla tietokoneet toimivat ilman aisteja ja tiesivät vain mitä niille kerrottiin. 2000-luvulla esineiden internetin ansiosta tietokoneet voivat aistia asioita itse. Ashton käyttää hyvänä esimerkkinä IoT-sovelluksesta GPS-paikantimia, jotka 2000-luvun alussa saapuivat markkinoille tavallisille kuluttajille. (Gabbai 2015.)

”Internet of Things” -termin syntyessä internet oli kytkettynä vain tietokoneisiin. IoT:n aikakauden katsotaan alkaneen varsinaisesti vuosien 2008-2009 aikana. Tänä aikana internetiin kytkettyjen laitteiden määrä ylitti maailman väkiluvun. Tätä hetkeä pidetään esineiden internetin aikakauden alkuna. (Hanes, Salgueiro & Barton 2017.)



## 2.2 Teollinen internet

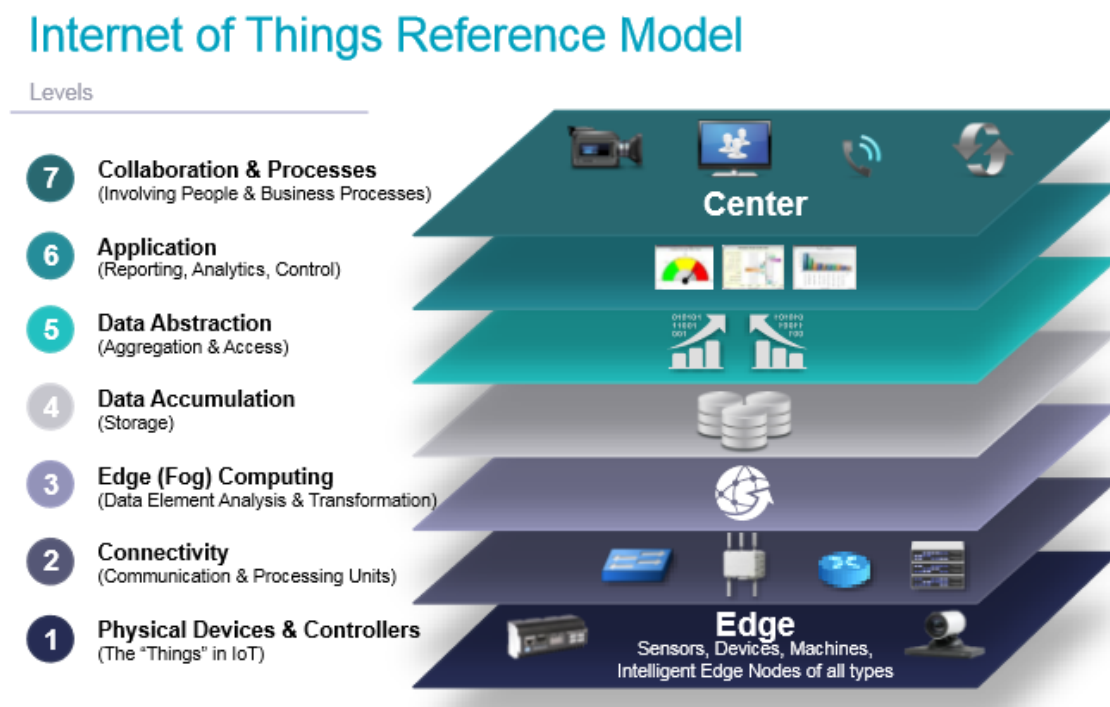
Teollinen internet, englanniksi IloT (Industrial Internet of Things), on esineiden internetiin kuuluva keskeinen osa-alue. Esineiden internetillä tarkoitetaan usein yksittäisiä laitteita, jotka on suunnattu yksityisille kuluttajille. Teollinen internet puolestaan on IoT:n alakäsite ja toimii teollisella mittakaavalla. Internetin lisääminen koneisiin ja prosesseihin digitalisoi teollisuuden ja tekee liiketoiminnasta datavetoista. Yritykset voivat luoda uusia tuotteita ja palveluita fyysisen ja digitaalisen yhdistelmän ansiosta. Digitalisaatiota käytetään aloilla ja paikoissa, joissa sitä ei ole ennen hyödynnetty. Se mahdollistaa tehtaiden laitteiden ja kokonaisten tuotantojärjestelmien helpomman ohjattavuuden ja valvonnan uusien autonomisten ratkaisujen myötä. Älykkäät koneet helpottavat ihmisten työtehtäviä ja osittain muuttavat niiden luonnetta, kun yksinkertaisiin rutiinitöihin ei tarvita enää ihmistä. Fyysiset työt väistämättä vähenevät samalla, kun tuotekehitys ja palvelutyöt lisääntyvät. (Collin & Saarelainen 2016.)

Teollisen internetin termin IloT kehitti yhdysvaltalainen monialayritys General Electric vuonna 2012 (General Electric n.d.). Yritys on määritellyt teollisen internetin jakautuvan kolmeen osa-alueeseen: älykkäät koneet, kehittynyt analytiikka ja ihmiset työssä. Älykkäitä koneita ovat fyysiset laitteet, kalustot ja laitokset, jotka on varustettu kehittyneillä antureilla, ohjaimilla ja ohjelmistoilla. Kehittynyt analytiikka yhdistää fysiikkaan perustuvan analytiikan, ennustavat algoritmit sekä toimialan asiantuntemuksen. Nämä mahdollistavat koneiden ja suurempien järjestelmien toiminnan ymmärryksen. Ihmiset hyötyvät teollisesta internetistä, kun saavat milloin ja missä vaan tukea älykkäämpiin päätöksiin, toimintoihin ja huoltotoimenpiteisiin sekä turvallisempaan työntekoon. (Evans & Annunziata 2012, 3.)

General Electric arvioi, että lähes puolet maailmantaloudesta voi lähitulevaisuudessa hyötyä teollisesta internetistä. Tämän lisäksi IloT:n tuomat edut voivat mahdollisesti laskea maapallon energian kulutusta lähes puolella sekä auttaa huomattavasti energian tuotannossa (General Electric n.d.). Yritys laski teollisen internetin lisäävän Euroopan bruttokansantuotteeseen jopa 2,8 biljoonaa dollaria vuoteen 2030 mennessä. Tämä vastaa yhtä neljäsosaa alueen nykyisestä taloudesta. (General Electric 2013, 22.)

## 2.3 IoT:n toimintaperiaate

IoT-järjestelmää suunniteltaessa on tärkeää tuntea sen eri osa-alueet ja täyttää niiden vaatimukset. IoT World Forum on luonut mallin IoT:n arkkitehtuurille vuonna 2014 (kuva 1). Mallissa on seitsemän tasoa, joilla voidaan hahmottaa IoT-järjestelmän eri toimintoja.



KUVA 1. IoT-järjestelmän rakennemalli (IoT World Forum 2014)

Tämä malli ei rajoita komponenttien sijaintia eli fyysiset laitteet ja anturit voivat sijaita yhdessä paikassa tai vaikka eri puolilla maapalloa. Malli myös sallii eri tasoilla tapahtuvien prosessien monimutkaisuuden vaihtelevan niin paljon kuin on tarpeellista. Data kulkee mallin eri tasojen läpi molempiin suuntiin. Laitteiden ohjauksessa data kulkee alaspäin tasosta seitsemän kohti tasoa yksi. Sen sijaan laitteiden antureista tuleva tieto kulkee toiseen suuntaan tasosta yksi tasoon seitsemän asti. (IoT World Forum 2014.)

IoT World Forumin IoT-järjestelmän rakennemallin ensimmäinen taso ovat fyysiset laitteet ja ohjaimet. Laitteiden anturit keräävät ympäristöstään tietoa, jota voi-

daan prosessin myöhäisemmässä vaiheessa analysoida. IoT-laitteet ovat monipuolisia, eikä niiden kokoa, sijaintia tai alkuperää ole määritetty. Toisen tason tärkein tehtävä on tiedon välitys luotettavasti. Fyysisten laitteiden data siirtyy verkkoyhteydestä toiseen ja lopulta tason 3 prosesseihin. Toisella tasolla on mahdollista toteuttaa osa protokollamuunnoksista ja tietoturvatarkistuksista. Protokollamuunnos mahdollistaa tietojen siirron ja keruun eri valmistajien laitteiden välillä. (IoT World Forum 2014.)

Fyysisistä laitteista tuleva data ei yleensä kelpaa varastoitavaksi tai korkeamman tason prosessoitavaksi. Kolmannen tason tehtävä on analysoida laitteista tulevaa runsasta datamäärää ja muuttaa se seuraavalle tasolle lukukelpoiseksi. Neljännellä tasolla data muutetaan tallennettavaan muotoon haluttuun sijaintiin. Data voidaan tallettaa esimerkiksi paikalliseen levyjärjestelmään tai suureen tietokantaan. Tallennettu data on sitten ylempien tasojen käytettävissä. (IoT World Forum 2014.)

Viidennen tason tehtävä on datan abstrahointi. Tarvittava data saattaa olla tallennettuna monessa eri sijainnissa, jos fyysisiä laitteita on monia, ne sijaitsevat eri paikoissa tai tallennustila on rajallinen. Taso 5 huolehtii datan yhdistämisestä eri lähteistä ja luo yhtenäiset näkymät, joita on helppo käsitellä ylempien tasojen sovelluksissa. Tasolla myös varmistetaan datan yhtenäisyys ja suojataan se tarvittavilla tunnistautumis- ja lupakäytännöillä. (IoT World Forum 2014.)

Kuudennella tasolla data analysoidaan sovelluksilla ja voidaan esittää yksinkertaisella tavalla visuaalisesti käyttäjälle. Sovelluksella voidaan valvoa ja ohjata IoT-laitetta tai vaikka vertailla siitä tulevaa dataa haluttuihin ohjearvoihin. Seitsemänteen tasoon kuuluvat ihmiset ja prosessit. IoT-laitteen lähettämä data hyödyttää ihmisiä tai erilaisia prosesseja. Saatua tietoa helpottaa käyttäjää työssään ja sen perusteella voi toimia tarvittavalla tavalla. (IoT World Forum 2014.)

IoT World Forumin luoma rakennemalli esineiden internetille ei ole ainut olemassa oleva malli. Malli on valittu tähän työhön sen selkeyden ja yksityiskohtaisuuden vuoksi. Mallin jokaista vaihetta ei välttämättä tarvitse seurata IoT-järjestelmää suunniteltaessa, mutta se antaa hyvän kuvan rakenteesta. IoT:lle ei ole

vielä määritetty virallista rakennetta tai terminologiaa, mutta IoT World Forumin luoma malli on hyvä askel kohti konseptin standardointia.

## 2.4 IoT-käyttöjärjestelmä

Käyttöjärjestelmä on ohjelmisto, joka liittää sovellusohjelmat ja laitteistot yhteen. Sen ansiosta sovelluksia on mahdollista käynnistää ja hallita. Käyttöjärjestelmä tarjoaa sovelluksille tarpeelliset työkalut, kuten oheislaitteiden ohjauksen ja muistinhallinnan. Järjestelmä huolehtii myös resurssien jakamisesta, kun monet sovellukset ja ohjelmat ovat samaan aikaan käynnissä. Käyttöjärjestelmät jakaantuvat tarkoituksensa mukaan yleiskäyttöön ja työasemakoneisiin. Yleiskäyttöiset käyttöjärjestelmät on suunniteltu pääasiassa palvelin- ja osituskäyttökoneille. Työasemille tarkoitetut käyttöjärjestelmät, kuten Windows-järjestelmät, on suunniteltu yhden käyttäjän koneisiin. (Haikala & Järvinen 2004, 11-12.)

IoT-käyttöjärjestelmässä yhdistyvät sovellusalustat, sulautetut järjestelmät sekä väliohjelmistot. Järjestelmässä tapahtuu IoT-verkon datan kerääminen, laitteiden integrointi sekä reaaliaikainen analysointi. Data kulkee IoT-laitteen ja käyttöjärjestelmän välillä molempiin suuntiin. Järjestelmä hoitaa datan analysoinnin, mitaamisen, suodattamisen sekä turvaamisen. Tietojen käsittelyn päätteeksi IoT-laitteista tuleva data muuttuu käyttöjärjestelmässä lukukelpoiseksi käyttäjälle. (Tutorialspoint: IoT Software n.d.)

IoT:n käyttöjärjestelmiä käytetään päätelaitteissa ja gateway-laitteissa eli IoT-reitittimissä. Päätelaitteet eivät välttämättä tarvitse käyttöjärjestelmää, mutta vuosien varrella ne ovat monimutkaistuneet muun muassa lisätyillä sensoreilla ja korkeammalla prosessointiteholla. Käyttöjärjestelmä helpottaa näiden osa-alueiden prosessointia ja muokkaamista. Gateway-laitteen ansiosta käyttöjärjestelmä on mahdollista lisätä päätelaitteisiin, joissa sellaista ei ole. Laitteen eri osa-alueiden prosessointi siirtyy silloin gateway-laitteeseen, jonka kanssa sovelluskäyttäjä pystyy kommunikoimaan. (Devopedia: IoT Operating Systems 2017.)

## 2.5 Esineiden- ja teollisen internetin käyttökohteet

Esineiden internetiä voidaan hyödyntää nykyään lähes kaikilla osa-alueilla niin yksityiselämässä kuin teollisuudessakin. Kuvassa 2 on suuntaa antavasti jaoteltu IoT:n käyttökohteita maailmalla (Bouhai & Saleh 2017). Esimerkit ovat erittäin rajalliset, mutta kuvastavat hyvin, miten internetiä voi hyödyntää monessa paikassa. Kuvassa IoT:n käyttökohteet on jaettu kolmeen osa-alueeseen: teollisuus, terveydenhuolto ja ”älykäs kaupunki”.



KUVA 2. IoT:n käyttökohteita (Bouhai & Saleh 2017)

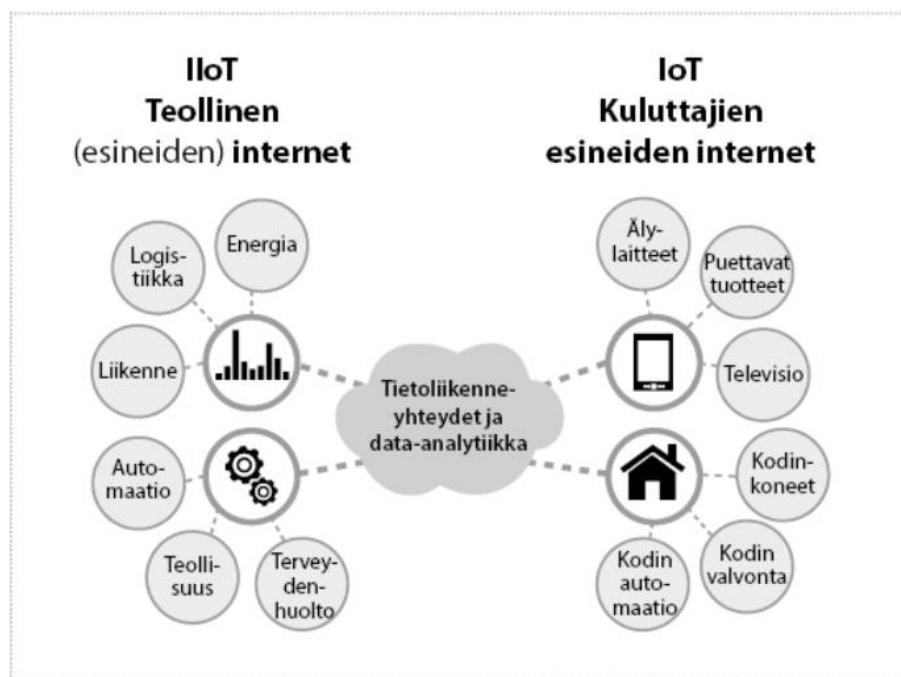
Teollinen osa-alue koostuu teollisista prosesseista, maataloudesta sekä logistisista. Hyviä esimerkkejä käyttökohteista näillä alueilla ovat tuotantolaitosten valvonta, eläinten paikannus ja seuranta sekä varastojen inventaariot.

Terveysthuolto on jaettu lääketieteeseen ja itsenäiseen elämiseen. Lääketieteeseen kuuluvat esimerkiksi potilaiden elintoimintojen kaukovalvonta ja diagnoosi. Itsenäisellä elämisellä tarkoitetaan ihmisten jokapäiväistä elämää avustavia IoT-laitteita. Iäkkäiden ja liikuntarajoitteisten ihmisten avustaminen helpottuu internetin avulla ja myös kaikki muut ihmiset kotona ja kodin ulkopuolella saavat elämää helpottavia ratkaisuja.

Älykkääseen kaupunkiin kuuluvat liikenne, älykäs verkko ja koti sekä ihmisten turvallisuus ja ympäristön valvonta. IoT:n eräs käyttökohde näille on esimerkiksi videovalvonta ympäristössä ja kotona. Energian tuotanto ja jakelu hyödyntää internetiä tehokkaasti ja liikenteessä esimerkiksi ruuhkien seuranta ja autojen pysäköinti tehostuu.

Esineiden internet eli IoT viittaa ensisijaisesti yksityisille kuluttajille suunnattuihin laitteisiin. Teollinen internet eli IIoT sen sijaan koostuu raskaammista koneista ja laitteista yrityksissä. Teknisiltä ominaisuuksiltaan IoT on selvästi yksinkertaisempi kuin IIoT. Teollisessa internetissä koneiden oletetaan kestävän jopa vuosikymmeniä, kun taas yksityisellä puolella tuotteiden elinkaari on selvästi lyhyempi. (Collin & Saarelainen 2016.)

Yksityisellä ja teollisella puolella internetiä käytetään vähän erilaisissa sovelluksissa. Kuvassa 3 on Collinin ja Saarelaisen (2016) esittämiä käyttökohteita näille kahdelle osa-alueelle. Kuvan perusteella teollinen internet koostuu teollisuudesta, automaatiosta ja terveydenhuollosta. Näiden lisäksi liikenne, logistiikka ja energiantuotanto kuuluvat siihen. Kuluttajien osuus koostuu yksinkertaisemmista IoT-laitteista, jotka ovat käytössä ihmisten arjessa. Nämä ovat kodin avustavia sovelluksia kuten automaatio, valvonta ja kodinkoneet. Puhelimet ja televisiot kuuluvat myös yksityisten laitteisiin.



KUVA 3. IoT:n ja IIoT:n käyttökohteiden jakautuminen (Collin & Saarelainen 2017)

### **3 OPETUSALUSTAN SUUNNITTELU**

#### **3.1 Opetusalustan vaatimukset**

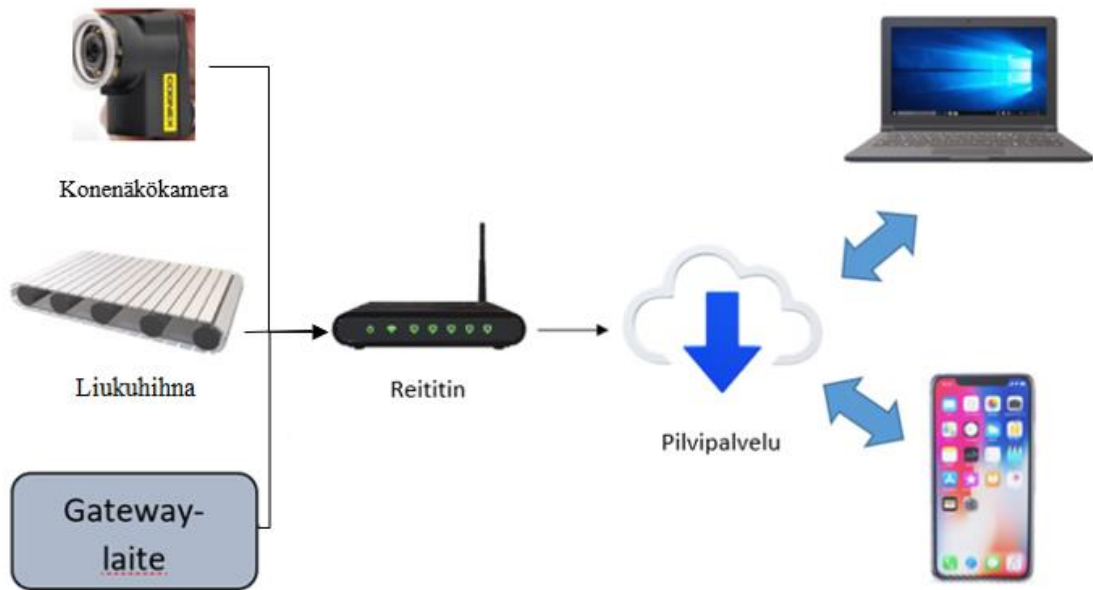
Opinnäytetyön tarkoituksena oli kehittää eteenpäin Tampereen ammattikorkeakoulun vanhaa opetusalustaa. Alusta on ohjelmoitava liukuhihna, joka on varustettu konenäkökameralla. Kamera voidaan ohjelmoida tunnistamaan liukuhihnaa pitkin kulkevien esineiden tiettyjä ominaisuuksia, kuten muotoa tai väriä.

Työn tavoitteena oli liukuhihna-alustan modernisointi lisäämällä siihen internet. Tämä mahdollistaa konenäkökameran ja liukuhihnan ohjelmoinnin mobiililaitteella tai tietokoneella langattoman verkon välityksellä. Projektin päätteeksi opetusalustan operoinnista ja ohjelmoinnista oli tarkoitus tehdä käyttöohjeet opiskelijoille. Käyttöohjeet tuli tehdä kolmen harjoitustehtävän muodossa. Niiden avulla on mahdollista toteuttaa työ uudelleen koulun opetuksessa.

#### **3.2 Opetusalustan rakenne**

Projektin tarkoituksena oli luoda langattoman verkkoyhteyden kautta operoitava opetusalusta (kuva 4). Alustana toimi koulusta löytyvä valmis liukuhihna (kuva 5), joka tunnistaa konenäkökameran (kuva 6) avulla esineitä. Tähän täytyi liittää gateway-laite eli IoT-reititin, joka mahdollistaa opetusalustan datan siirtymisen internetin kautta pilvipalveluun ja toisinpäin. Pilven kautta kulkevaa dataa voidaan lopuksi analysoida tietokoneella tai mobiililaitteella.





KUVA 4. Opetusalustan rakenne



KUVA 5. Opetusalustan liukuhihna



KUVA 6. Opetusalustan konenäkökamera

Opetusalustassa toteutuu opinnäytetyöraportin toisessa luvussa kuvattu IoT World Forumin IoT-järjestelmän rakennemalli. Mallin ensimmäisen tason fyysisinä laitteina toimivat alustassa liukuhihna ja konenäkökamera. Liukuhihnan ohjelmointi tapahtuu ja konenäkökameran keräämä data siirtyy reitittimellä pilveen analysoitavaksi rakennemallin toisella tasolla. Alustaan kytkettävä gateway-laite toteuttaa datan turvaamisen ja muuttamisen lukukelpoiseksi IoT Studiolla tasojen 3 ja 5 mukaisesti. Rakennemallin taso 4 eli datan tallennus toteutuu pilvessä IoT Studiassa. Tason 6 datan visuaalinen esitys käyttäjälle ja sen analysointi tapahtuu myös IoT Studiassa. Seitsemäs ja viimeinen taso ovat opiskelijat, jotka pääsevät analysoimaan alustasta tulevaa dataa sekä operoimaan sitä.

### 3.3 Tarvittavat komponentit

Elektroniikassa komponentilla tarkoitetaan sähkölaitteen rakenneosaa. Sähkölaitteet sisältävät yleensä useita komponentteja, joita kiinnitetään useimmiten piirilevylle tai toisiinsa johtimilla. (Karkkulainen, Karkkulainen & Kinnunen 2014.)

Projektin toteuttamiseen käytettiin Tampereen ammattikorkeakoulusta löytyviä komponentteja. Seuraavalla sivulla on lista käytetyistä komponenteista:

- liukuhihna
- ohjelmitava logiikka (Siemens S7-300)
- konenäkökamera (Cognex 1403)
- I/O-moduuli (Cognex CIO-MICRO)
- gateway-laite (Nexcom CPS 100-RE)
- reititin
- kaapeleita ja verkkokaapeleita
- tietokone/älypuhelin

### 3.4 Ohjelmointiympäristö

Ohjelmointiympäristö on yksinkertaisimmillaan tekstieditori, joka kääntää ja tulkaa käyttäjän kirjoittamat käskyt (MAOL n.d.). Visuaalinen ohjelmointiympäristö eroaa tavallisesta tekstieditorista antamalla käyttäjän ohjelmoida graafisten elementtien avulla (Dehouck 2015).

Tässä työssä käytettiin kolmea eri visuaalista ohjelmointiympäristöä opetusalan toiminnassa. Logiikkaa ja sitä kautta liukuhihnaa ohjelmoitiin TIA Portalilla. Konenäkökameran parametrien säätö ja valvontakuvan seuranta tapahtui In-Sight Explorerilla. Työn varsinainen uusi osuus oli internetin lisäys alustaan, mikä tapahtui gateway-laitteen avulla. Gatewayn ansiosta liukuhihnaa ja konenäkökameraa pääsee operoimaan verkkopohjaisella IoT Studiolla.

#### 3.4.1 TIA Portal

TIA Portal on Siemensin luoma ohjelmointityökalu automaatiosuunnitteluun. Lyhenne ”TIA” tulee sanoista Totally Integrated Automation eli suomennettuna ”täysin integroitu automaatio”. Ohjelmassa on yhdistetty logiikkojen (SIMATIC STEP 7), käyttöliittymien (SIMATIC WinCC) sekä taajuusmuuttajien (SINAMICS Start-Drive) ohjelmointi. Nämä kolme ohjelmaa yhdessä mahdollistavat suunnittelu- ja tuotantoprosessien luomisen koko tuotantoketjulle. Automaation konfigurointi, diagnostiikka sekä ylläpito onnistuu TIA:n ansiosta yhdellä ohjelmalla. (Siemens: TIA Portal n.d.)

SIMATIC STEP 7 on logiikkojen ohjelmointiin suunniteltu työkalu. Sillä voi ohjelmoida S7-300-, S7-400-, S7-1200-, S7-1500- sekä WinAC-logiikkaohjaimet. SIMATIC WinCC mahdollistaa visuaalisten käyttöliittymien luomisen. Sillä voi tehdä niin yksittäisten koneiden käyttöliittymät kuin tehtaaneläimistöjen valvontajärjestelmätkin. SINAMICS StartDrive:lla voi ohjelmoida SINAMICS-taajuusmuuttajia. Ohjelmalla toteutetaan taajuusmuuttajien liittäminen automaatioon ja järjestelmän käyttöönotto. (Siemens: Automaatio-ohjelmistot n.d.)

### **3.4.2 IoT Studio**

IoT Studio on Nexcomin verkkopohjainen konfigurointityökalu IoT-laitteille. Laitteiden asetukset muokataan valitsemalla halutut toiminnot kirjastosta ja lisäämällä ne vuokaavioon. Ohjelma pohjautuu Node.js- ja IBM Node-RED -sovelluksiin. (Nexcom 2011.)

Node.js on alustariippumaton avoimen lähdekoodin runtime-ympäristö palvelin- ja verkkosovellusten luomiseen. Node.js-ohjelmat kirjoitetaan Google Chromen JavaScript-komentokielellä ja voidaan suorittaa OS X:llä, Microsoft Windowsilla tai Linuxilla. (Tutorialspoint n.d.)

IBM Node-RED on visuaaliseen ohjelmointiin tarkoitettu työkalu, jolla voi yhdistää tietokonelaitteistoja, ohjelmointirajapintoja ja verkkopalveluita osana esineiden internetiä. Ohjelma toimii verkossa ja siinä luodaan vuokaavioilla halutut toiminnot. (Node-RED n.d.)

### **3.4.3 In-Sight Explorer**

In-Sight Explorer on Cognexin konenäkökameroiden ohjelmointiin tarkoitettu työkalu. Ohjelmalla voi säätää kameran parametrit ja tarkkailla valvontakuvaa. In-Sight Explorerin avulla kameran kuvaan ilmestyviä esineitä voi tutkia ominaisuuksiensa mukaan. Ohjelmalla voi tunnistaa ja seurata esineitä muodon tai värin perusteella. Ohjelma tunnistaa esineissä myös epämuodostumia tai vikoja. Myös

kirjaimet ja viivakoodit on mahdollista lukea konenäkökameran kautta. (Cognex: In-Sight Explorer Software n.d.)

Konenäkökamerat ja In-Sight Explorer auttavat yrityksiä parantamaan tuotteiden laatua, poistamaan tuotannon virheitä sekä alentamaan valmistuskustannuksia. Cognex on maailman johtava konenäkökameroiden, ohjelmien, antureiden sekä viivakoodien lukijoiden tarjoaja. (Cognex: The Leader in Machine Vision n.d.)

## 4 LIUKUHIHINAN TOIMINTA

### 4.1 Liukuhinnan käyttöönotto ja sen tavoitteet

Projektin tarkoituksena oli liittää liukuhinna internetiin, mikä mahdollistaa sen etäohjauksen ja -valvonnan tietokoneella tai mobiililaitteella. Liukuhinna kytkettiin Siemensin S7-300-logiikkaan, jonka ohjelmointi tapahtui TIA Portalilla. Liukuhinnan käynnistys ja pyörimissuunnan vaihto tapahtuu kahdella ulostulolla. Kytketyt ulostulot ovat käynnistykselle 136 (%Q136.0) ja pyörimissuunnan vaihdolle 136,1 (%Q136.1).

Internetin yhdistäminen liukuhintaan tapahtui liittämällä ohjelmoitava logiikka verkkokaapelilla erilliseen reitittimeen. Reitittimeen tuli yhdistää verkkokaapelilla myös CPS 100-RE -gateway-laite, jonka sisälle ohjelmoitiin IoT Studiolla liukuhinnan toiminta. IoT Studion käyttämiseksi kytkettiin tietokone tai mobiililaitte reitittimeen kaapelilla tai langattomasti. Tämä kokonaisuus mahdollistaa liukuhinnan ohjauksen ja valvonnan etäyhteydellä IoT Studion kautta. Etäyhteys on kuitenkin rajoitettu käytetyn reitittimen kantamalle.

Tarkoituksena tässä projektissa oli luoda liukuhinnalle IoT Studioon kaksi komentosarjaa, joita tässä ohjelmassa kutsutaan nimellä "flow". Ensimmäinen flow ilmoittaa, onko liukuhinna päällä vai sammutettu. Toisen flown avulla liukuhinna voidaan käynnistää ja sen pyörimissuunta vaihtaa.

### 4.2 Liukuhinnan etävalvonnan toteutus

Liukuhinnalle luotiin ohjelma, jolla voidaan valvoa, onko hihna päällä vai sammutettu. Siemensin logiikkaan tehtiin TIA Portalin avulla ohjelma liukuhinnan käynnistykselle. Käynnistys tapahtuu logiikan fyysisestä kytkimestä sisääntulolla 0 (%I0.0), joka ohjelmoitiin aktivoimaan ulostulo 136. TIA Portaliiin luotiin yksinkertainen ohjelma, jolla saadaan liukuhinna käyntiin logiikan fyysisestä kytkimestä (kuva 7). Tarkoitus on nähdä IoT Studion kautta, onko fyysinen kytkin aktivoitu.



KUVA 7. TIA Portal-ohjelma käynnistykselle

IoT Studioon luotiin flow, joka ilmoittaa logiikan käynnistyskytkimen asennosta ja sitä kautta liukuhinnan tilasta. Flow sisältää "nodeja" eli komentoja. Käytetyt nodet ovat "inject", "read S7", "function" sekä "debug" ja ne asetettiin kuvan 8 esittämällä tavalla.



KUVA 8. IoT Studion etävalvonnan flow

"Inject"-noden tehtävä on aktivoida flow antamalla sille haluttu käsky tietyin aikavälein. Käsky voi olla eri muodoissa, kuten numerona tai oikein/väärin, ja aikaväli toistolle on säädettävissä. Tässä flowssa ei ollut tarkoitus ohjata liukuhintaa, joten noden käskyksi määritettiin "timestamp", joka ilmoittaa kellonajan sekunnin välein (kuva 9).

**Edit inject node**

Cancel Done

✉ Payload

☰ Topic

🔄 Repeat

every

Inject once at start?

KUVA 9. Inject-node

”Read S7” on Siemensin S7-sarjan logiikoille luotu node. Se mahdollistaa yhteyden ja signaalien lukemisen IoT Studion ja logiikan välillä. ”Read S7” -nodeen määritettiin logiikan IP-osoite ja portti (kuva 10). Tampereen ammattikorkeakoulun Siemensin S7-300-logiikoissa on IP-osoitteena 192.168.50.20. Portin numero selvisi Siemensin internetsivulta, jossa ilmoitetaan TCP:n eli tietokoneiden välisen yhteyden luomiseksi portti 102 (Siemens support 2017). Nodeen määritettiin signaalit, joita halutaan tutkia eli tässä tapauksessa TIA Portalissa luodun ohjelman ulostulon osoite (offset) ja tyyppi (output & boolean).

read > **Edit connection node**

Delete Cancel Update

**Connection parameter**

IP-Adress:  Port:

Rack:  Slot:

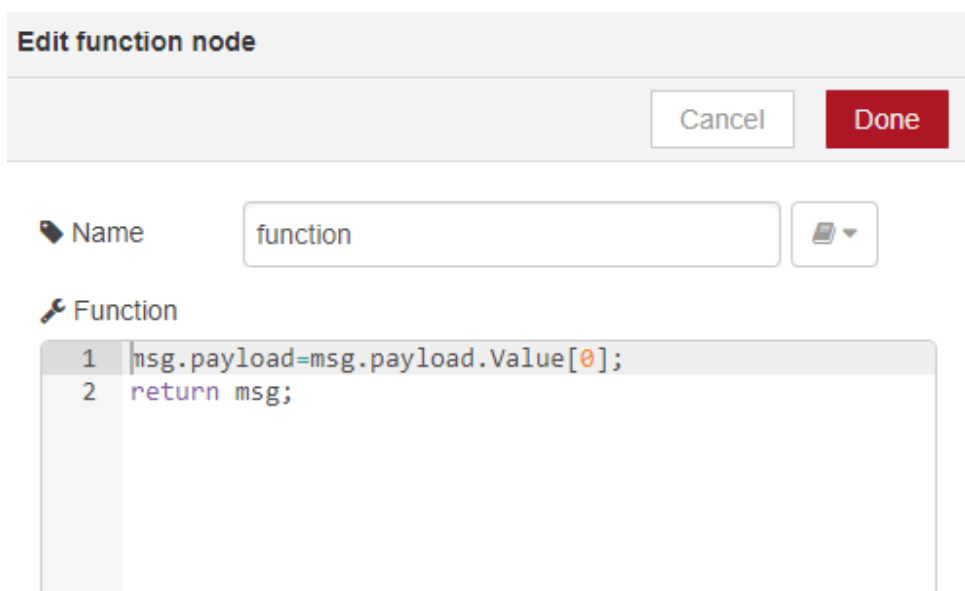
Signals:

Offset:

KUVA 10. Read S7 -node

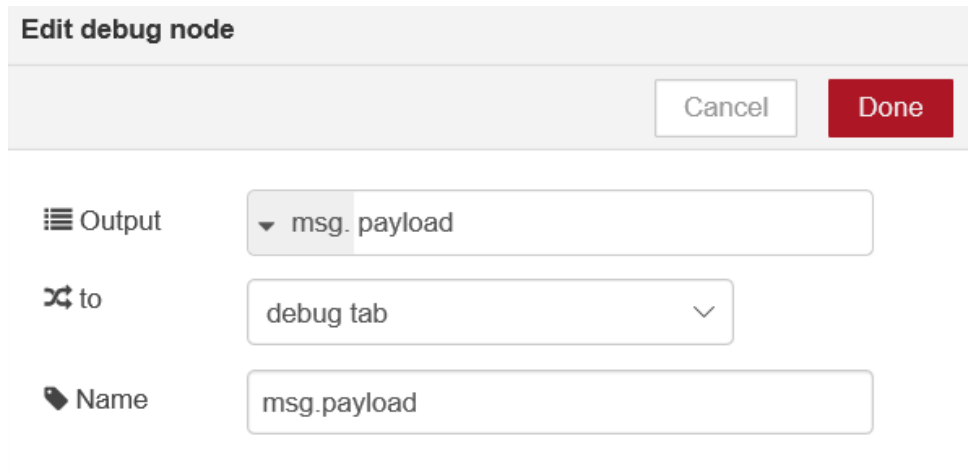


"Function"-nodea käytetään, kun ei löydy sopivaa valmista nodea haluttuun tehtävään. Node on vapaasti säädettävissä JavaScript-ohjelmointikielellä. Tässä tapauksessa kirjoitettiin ohjelma, joka poimii tiedon sisääntulon signaalista (kuva 11).



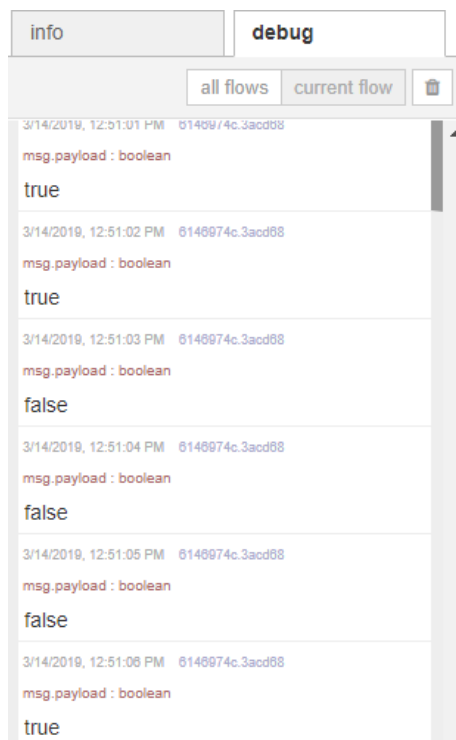
KUVA 11. Function-node

Viimeinen node, "debug", käynnistää ja sammuttaa luodun flown. Se myös julkaisee flown keräämän datan, joka tässä tapauksessa ilmoittaa sisääntulosta sanoilla "true" ja "false". Sana "true" ilmoittaa logiikan kytkimen olevan ON-asennossa eli liukuhihnan olevan käynnissä. "False" sen sijaan ilmoittaa kytkimen OFF-asennosta, jolloin hihna on pysähdyksissä. Noden alkuperäiset asetukset kelpasivat tähän tarkoitukseen (kuva 12).



KUVA 12. Debug-node

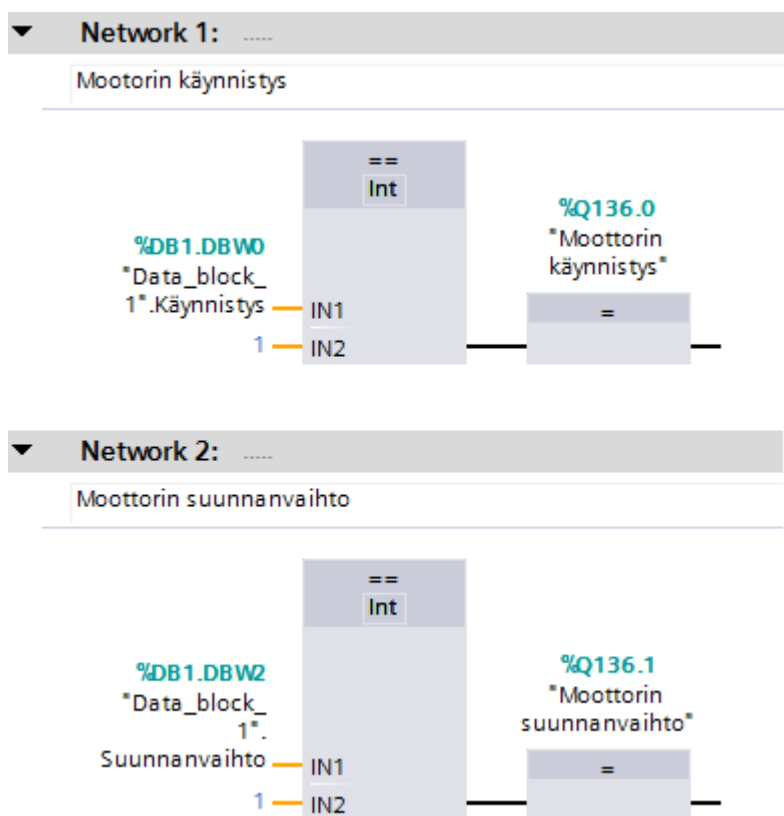
Luotu flow säädettiin julkaisemaan sekunnin välein, onko liukuhihna käynnissä. Tiedon löytää IoT Studion oikeasta ylänurkasta debug-välilehdeltä (kuva 13).



KUVA 13. Flown ilmoittama liukuhihnan tila

### 4.3 Liukuhinnan etäohjauksen toteutus

Tavoitteena oli saada etäyhteydellä liukuhihna käyntiin ja vaihtamaan suuntaa. Tämän toteuttamiseksi logiikka käyttää käynnistykselle ja suunnanvaihdolle uloslutot 136 (%Q136.0) ja 136,1 (%Q136.1). Tässä osiossa ei tarvinnut käyttää fyysisiä kytkimiä, sillä ohjaus tapahtuu IoT Studion kautta. Etäohjauksen mahdollistamiseksi luotiin TIA Portal -ohjelma, jonka avulla saadaan liukuhihna käyntiin ja vaihtamaan pyörimissuuntaa (kuva 14). Ohjelma säädettiin käynnistämään ja vaihtamaan liukuhinnan pyörimissuunta, kun IoT Studiolta tulee käskynä numero "1". Käskyllä "0" saa samat toiminnot pois päältä. IoT Studion kautta pystyy lähettämään käskyt erikseen käynnistykselle tai suunnanvaihdolle.



KUVA 14. TIA Portal -ohjelma etäohjaukselle

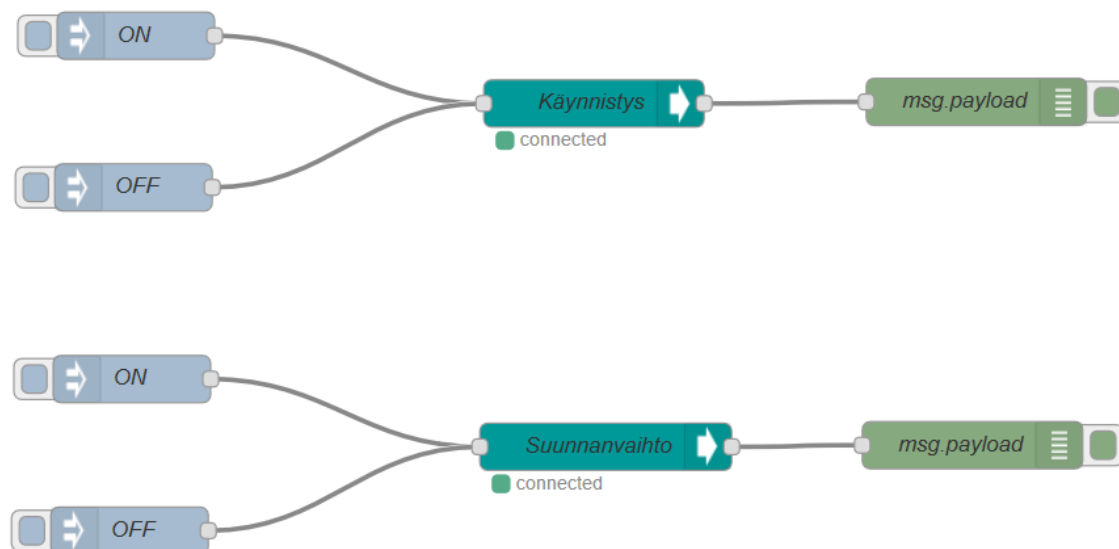
Etäohjauksen mahdollistamiseksi käytettiin fyysisten kytkinten ja sisääntulojen sijaan TIA Portal -ohjelman "datablock"-osiota. Datablock mahdollistaa kirjoittami-

sen ja signaalin vastaanottamisen IoT Studiolta. Kuvassa 14 esitetty ohjelma valvoo datablock-välilehteen tulevia signaaleja ja sen perusteella käynnistää ulostulot. Kuvassa 15 on esitetty datablockin tarvitsemat osoitteet liukuhihnan käynnistykseksi ja suunnanvaihdolle.

Data_block_1				
	Name	Data type	Offset	Start value
1	Static			
2	Käynnistys	Int	0.0	0
3	Suunnanvaihto	Int	2.0	0

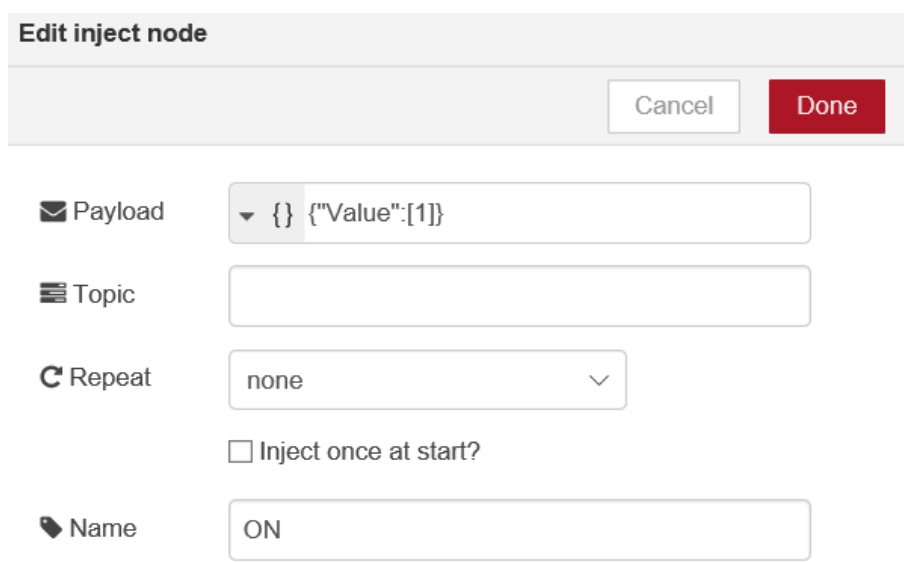
KUVA 15. Etäohjauksen datablock-lista

IoT Studioon luotiin etäohjauksen flow, joka eroaa vähän liukuhihnan etävalvonnasta. Tarvittavat nodet ovat inject, write S7 sekä debug. Flowssa tehtiin erikseen komentorivi liukuhihnan käynnistykseksi ja suunnanvaihdolle. Rivien nodet asetettiin kuvan 16 esittämässä järjestyksessä. Rivit ovat muuten samanlaiset keskenään, mutta kummatkin ohjaavat yhden logiikan ulostulon.



KUVA 16. IoT Studion etäohjauksen flow

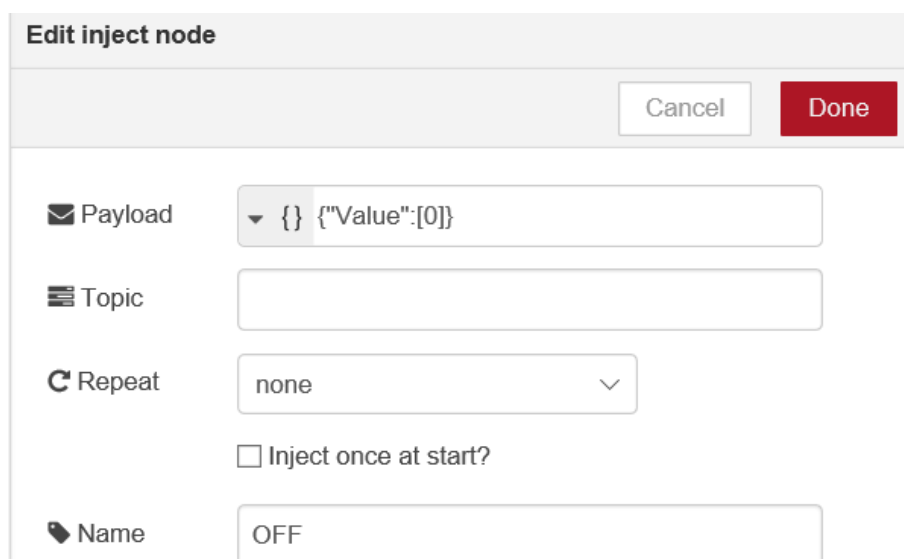
Flowssa on neljä inject nodea, joilla saa ohjattua liukuhihnaa. Nodet nimitettiin "ON"- ja "OFF"-napeiksi, joita painamalla saa liukuhinnan käynnistettyä, vaihtamaan suuntaa sekä sammutettua. Inject-nodeille määritettiin käskyksi (payload) JSON-ohjelmointikielellä komentorivi `{"Value": [X]}`. X:n tilalle kirjoitettiin ON-nappeihin "1" (kuva 17) ja OFF-nappeihin "0" (kuva 18). Kirjoitetut rivit lähettävät napin painalluksella arvon 1 tai 0. Käskyt halutaan kirjoittaa vain kerran napin painalluksesta, joten nodejen toistot kytkettiin pois päältä.



The screenshot shows the 'Edit inject node' dialog box. At the top right, there are 'Cancel' and 'Done' buttons. The dialog contains the following fields:

- Payload:** A dropdown menu showing `{ } {"Value":[1]}`.
- Topic:** An empty text input field.
- Repeat:** A dropdown menu showing 'none'.
- Inject once at start?
- Name:** A text input field containing 'ON'.

KUVA 17. Inject-noden ON-asetukset



The screenshot shows the 'Edit inject node' dialog box. At the top right, there are 'Cancel' and 'Done' buttons. The dialog contains the following fields:

- Payload:** A dropdown menu showing `{ } {"Value":[0]}`.
- Topic:** An empty text input field.
- Repeat:** A dropdown menu showing 'none'.
- Inject once at start?
- Name:** A text input field containing 'OFF'.

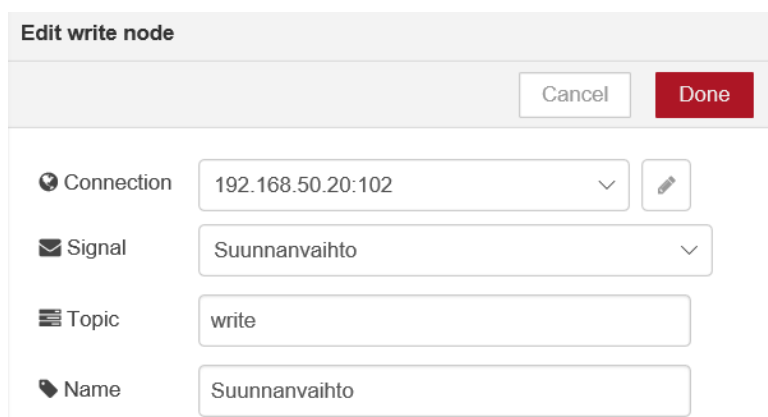
KUVA 18. Inject-noden OFF-asetukset

Tässä flowssa on write S7 -node, joka ei ole käytössä etävalvonnan flowssa. Read S7 -noden tavoin write S7 -node on suunniteltu Siemensin logiikoille. Se mahdollistaa kirjoittamisen IoT Studiolla Siemensin logiikoihin. Nodeen määritettiin logiikan IP-osoite ja portti samalla tavalla kuin etävalvonnan read S7 -nodessa. Lisäksi määritettiin TIA Portalissa luodut datablockin osoitteet, joihin käskyt lähetetään (kuva 19).

KUVA 19. Write S7 -node

Write S7-nodeja on flowssa erikseen käynnistykselle ja suunnanvaihdolle. Osoitteet pysyvät samoina kummassakin, mutta se, kumpaan signaaliin kirjoitetaan käskyt, valitaan. (kuva 20 & 21).

KUVA 20. Write S7 käynnistykselle



**Edit write node**

Cancel Done

Connection 192.168.50.20:102

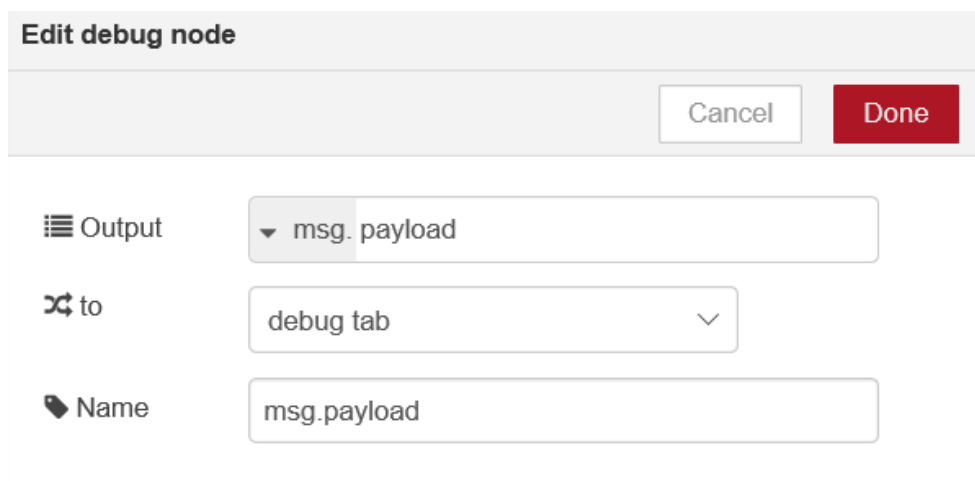
Signal Suunnanvaihto

Topic write

Name Suunnanvaihto

KUVA 21. Write S7 suunnanvaihdolle

Flown viimeinen node on debug, jolla saadaan flown data julkaistua IoT Studiassa. Noden asetuksia ei tarvinnut säätää (kuva 22). Node julkaisee debugvälilehteen ilmoituksia signaaleista, kun käskyjä lähetetään logiikalle (kuva 23).



**Edit debug node**

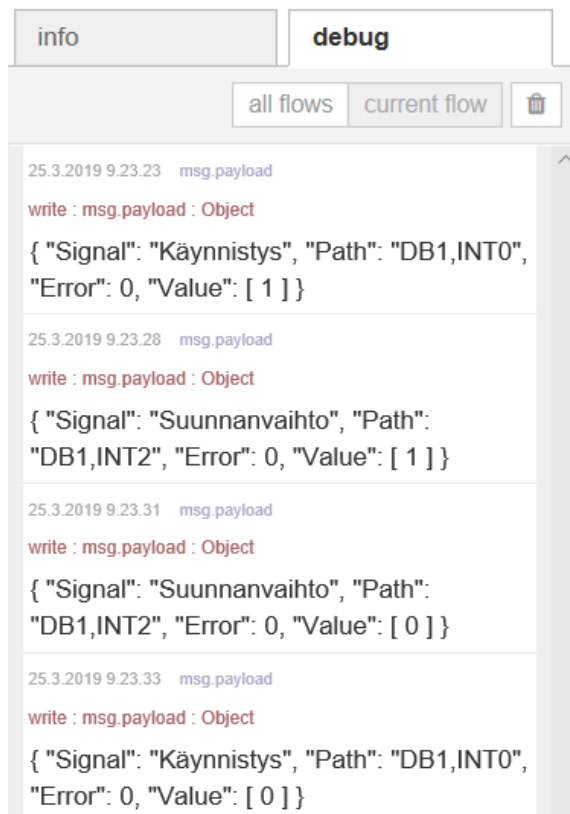
Cancel Done

Output msg.payload

to debug tab

Name msg.payload

KUVA 22. Debug-noden asetukset



KUVA 23. Flown tulostama signaalien tila

Etävalvonnan flow lähettää Siemensin logiikalle ON- ja OFF-napeista käskyt 0 tai 1, minkä perusteella logiikka reagoi ja toimii. Flown käynnistysrivistä saa liukuhihnan päälle ja suunnanvaihdosta vaihtuu sen pyörimissuunta.



## 5 KONENÄKÖKAMERAN TOIMINTA

### 5.1 Kameran käyttöönotto ja sen tavoitteet

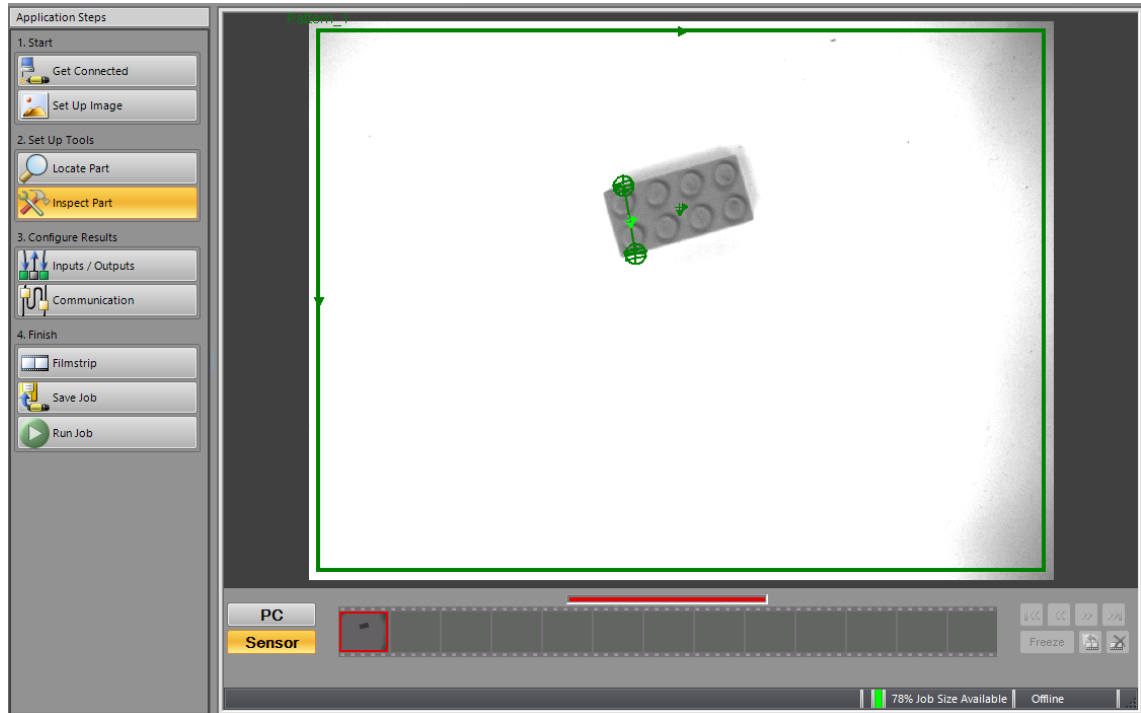
Konenäkökamera Cognex 1403 oli tarkoitus liittää internetiin ja valvoa siitä tuleva data etäyhteydellä. Kamera kytkettiin Cognexin CIO-MICRO I/O -moduuliin, josta data saadaan verkkokaapelin välityksellä tietokoneelle In-Sight Explorer -ohjelmaan. Konenäkökameran ohjelmointi toteutettiin samalla ohjelmalla. IoT Studioissa ei ole valmista nodea Cognexin kameroille, joten dataa ei voi suoraan lähettää kameralta IoT Studioon. Ongelman ratkaisemiseksi kameran data siirrettiin Siemensin ohjelmoitavaan logiikkaan, josta sen lukeminen on mahdollista IoT Studiolla.

Siemensin S7-300-logiikan sisääntulo 136 (%I136.0) kytkettiin Cognexin I/O-moduulin ulostuloon 0. Tarvittava data kulkee sitä kautta kamerasta ohjelmoitavaan logiikkaan.

Konenäkökameraan oli tarkoitus luoda yksinkertainen ohjelma, joka tunnistaa tietyn Lego-palikan. Legot kuljetetaan kameran näkökenttään liukuhihnaa pitkin, joka edellisessä kappaleessa kuvatus mukaisesti oli saatu ohjattua etäyhteydellä. Tavoitteena oli IoT Studion kautta saada selville, montako Lego-palikkaa konenäkökamera on havainnut. Tähän tarkoitukseen oli luotava IoT Studioon flow, joka lukee TIA Portalista kameran antaman tiedon Legojen määrästä. Projektin lopputuloksena voidaan konenäkökameran ja liukuhihnan yhdistelmää ohjata etäyhteydellä niin, että Legot kuljetetaan liukuhihnaa pitkin kameran näkökenttään ja kamera rekisteröi tiedon niiden määrästä.

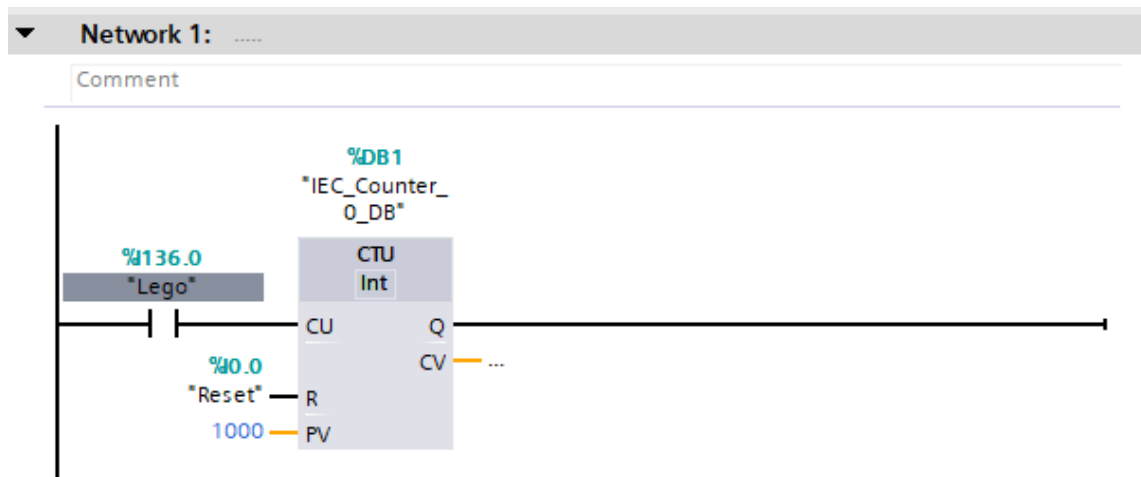
### 5.2 Kameran etävalvonnan toteutus

Cognexin konenäkökameraan luotiin yksinkertainen ohjelma Lego-palikan tunnistukselle. Ohjelma säädettiin tunnistamaan palikan muoto ja leveysmitta (kuva 24). Kamera määritettiin lähettämään signaalin I/O-moduulin ulostulon 0 kautta, kun se tunnistaa Lego-palikan leveysmitan.



KUVA 24. Lego-palikan tunnistus In-Sight Explorerissa

I/O-moduulin ulostulosta tieto Lego-palikan tunnistuksesta kulkee johtoa pitkin Siemensin logiikan sisääntuloon 136. TIA Portaliin luotiin ohjelma laskemaan, montako kertaa sisääntuloon tulee palikasta signaali (kuva 25). Laskurin nollamiseksi lisättiin fyysinen kytkin sisääntulolla 0. Kuvassa 26 on TIA Portalin luoma automaattinen datablock-välilehti laskurille.



KUVA 25. TIA Portaliin luotu laskuri

IEC_Counter_0_DB							
	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint
1	▼ Input				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	■ CU	Bool	0.0	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	■ R	Bool	0.1	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	■ PV	Int	2.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	▼ Output				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	■ Q	Bool	4.0	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	■ CV	Int	6.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	▼ Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	■ CUO	Bool	8.0	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

KUVA 26. Laskurin datablock-välilehti

Konenäkökameralle luotiin IoT Studiossa flow, joka ilmoittaa TIA Portalin laskurin arvon. Flown nodet ovat inject, read S7 sekä debug (kuva 27).



KUVA 27. Konenäkökameran flow

Flowssa luetaan Siemensin logiikalta saapuvaa dataa eli inject-noden asetuksiin valittiin timestamp ja toisto sekunnin välein (kuva 28). Sillä saa luettua saapuvan datan sekunnin välein.

**Edit inject node**

✉ Payload ▼ timestamp

☰ Topic

🔄 Repeat interval ▼

every 1 ↑ ↓ seconds ▼

Inject once at start?

KUVA 28. Konenäkökameran inject-node

Read S7 -nodeen asetettiin logiikan IP-osoite ja portti, kuten liukuhinnan ohjelmassa. Luettavaksi signaaliksi valittiin TIA Portalilla luodun datablock-välilehden CV:n osoite (kuva 29). CV:ssä on luettavissa laskurin antama Lego-palikkoiden lukumäärä.

read > **Edit connection node**

**Connection parameter**

IP-Adress:	<input style="width: 95%;" type="text" value="192.168.50.20"/>	Port:	<input style="width: 95%;" type="text" value="102"/>
Rack:	<input style="width: 95%;" type="text" value="0"/>	Slot:	<input style="width: 95%;" type="text" value="2"/>

Signals:

<input style="width: 95%;" type="text" value="Databloc ▼"/>	Num:	<input style="width: 95%;" type="text" value="1"/>	Int ▼	Offset:	<input style="width: 95%;" type="text" value="6"/>	Quantity:	<input style="width: 95%;" type="text" value="1"/>	<input style="width: 95%;" type="text" value="Laskuri"/>
---	------	--	-------	---------	--	-----------	--	--

KUVA 29. Konenäkökameran read S7 -node

Flown päättää debug-node, jolla saa julkaistua luetun datan IoT Studiassa. Nodeen alkuperäisiä asetuksia ei tarvinnut muuttaa. Aktivoitu flow tulostaa debug-välilehteen tiedon, montako Lego-palikkaa konenäkökamera tunnistaa (kuva 30).

The screenshot shows a debug console window with two tabs: 'info' and 'debug'. The 'debug' tab is active. At the top, there are buttons for 'all flows', 'current flow', and a trash icon. The main area displays a list of messages, each with a timestamp, a unique ID, and a payload object. The payload objects are JSON objects with the following structure: {"Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 0 ]}, {"Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 1 ]}, {"Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 2 ]}, {"Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 3 ]}, {"Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 4 ]}, {"Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 5 ]}. The 'Value' array represents the number of Lego bricks, which increases from 0 to 5 across the messages.

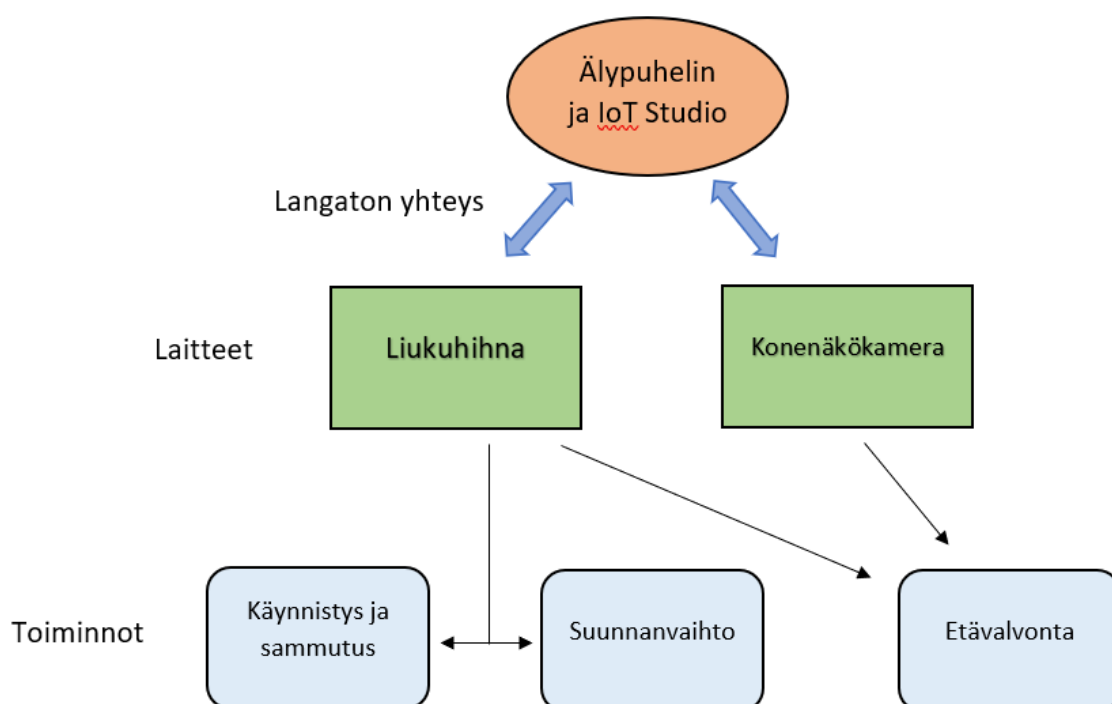
```
info | debug
all flows | current flow | trash
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 0 ] }
29.3.2019 12.42:09 9500181d.d760b8
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 0 ] }
29.3.2019 12.42:42 9500181d.d760b8
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 1 ] }
29.3.2019 12.42:52 9500181d.d760b8
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 2 ] }
29.3.2019 12.43:01 9500181d.d760b8
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 3 ] }
29.3.2019 12.43:09 9500181d.d760b8
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 4 ] }
29.3.2019 12.43:18 9500181d.d760b8
msg.payload : Object
{ "Signal": "Laskuri", "Path": "DB1,INT6",
  "Error": 0, "Value": [ 5 ] }
```

KUVA 30. Flown tulostama Lego-palikoiden lukumäärä

## 6 VALMIS OPETUSALUSTA JA SEN TOIMINNOT

Valmistunut opetusalusta toimii langattomasti etäyhteydellä ja ohjaus tapahtuu älypuhelimella tai tietokoneella (kuva 31). IoT Studioon kirjautumalla voi valvoa ja ohjata opetusalustan liukuhihnaa ja konenäkökameraa. Liukuhihnan pystyy käynnistämään ja sammuttamaan sekä sen suunnan vaihtamaan IoT Studiosta. Ohjelmasta pystyy myös valvomaan, onko liukuhihna käynnissä. Konenäkökamerasta on luettavissa tunnistettujen Lego-palikoiden lukumäärä. Näitä toimintoja yhdistämällä pystyy liukuhihnan käynnistämään, siirtämään sitä pitkin Lego-palikoita konenäkökameralle ja valvomaan palikoiden lukumäärää.

Opinnäytetyössä opetusalustaan lisättiin gateway-laite ja reititin. Liukuhihna ja konenäkökamera kytkettiin ohjelmoitavan logiikan kautta edellä mainittuihin laitteisiin. Näitä asioita lukuun ottamatta opetusalustaa ei muutettu. Uusi kokonaisuus lisäsi etäyhteyden liukuhihnalle ja konenäkökameralle. Tämä avasi uusia mahdollisuuksia alustan käytölle ja koulun opetukselle.



KUVA 31. Opetusalustan rakenne ja toiminnot

## 7 OPETUSALUSTAN HARJOITUSTEHTÄVIEN LAATIMINEN

Opinnäytetyön tavoitteisiin kuului harjoitustehtävien laatiminen Tampereen ammattikorkeakoulun kone- ja tuotantotekniikan opetukseen. Harjoitustehtävillä on tarkoitus tutustuttaa opiskelijat IoT:n käytännön puoleen. Tästä syystä tehtävistä piti tehdä yksinkertaisia ja selkeitä. Harjoitusten avulla tulee pystyä toteuttamaan tässä opinnäytetyössä luodut etävalvonnat ja -ohjaukset liukuhihnalle ja konenäkökameralle.

Opinnäytetyössä luotu opetusaluusta pilkottiin kolmeksi harjoitustehtäväksi: 1. Liukuhinnan etävalvonta, 2. Liukuhinnan etäohjaus ja 3. Konenäkökameran etävalvonta. Ohjeiden laadinnassa oli tärkeää huomioida ongelmat, joita opiskelijat voisivat kohdata. Täytyi myös olettaa, etteivät opettajat ole välttämättä tutustuneet IoT Studioon aiemmin. Näistä syistä erityisesti IoT Studion käytön ohjeistuksen piti olla mahdollisimman yksityiskohtaista.

Harjoitustehtäviä tulee käyttää opetuksessa vasta kun TIA Portal sekä liukuhinnan ja konenäkökameran ohjelmointi on opiskelijoille tuttua. Tästä syystä tehtäviä laadittaessa oletettiin opiskelijoilla olevan aiemmin mainitut perusteet hallussa. Liukuhinnan ja konenäkökameran kytkennät jätettiin joko kokonaan pois tai lisättiin vain osittain ohjeistukseen. TIA Portaliin luotavien ohjelmien perusteella tulisi opiskelijoiden pystyä kytkemään edellä mainitut laitteet viimeistään opettajan avustuksella.

Valmiit harjoitustehtävät testattiin yhden opiskelijan kanssa. Opiskelija ei tuntenut ennestään IoT Studiota ja tarkoitus oli tarkistaa, miten hyvin tehtävät oli laadittu. Palautteen perusteella harjoituksiin tehtiin muutamat muokkaukset ja lisäykset. Tehtävien pitäisi olla nyt tarpeeksi selkeät ja yksityiskohtaiset, jotta ne voidaan ottaa opetuksessa käyttöön. Harjoitustehtävät löytyvät opinnäytetyön liitteistä 1-3.

## 8 POHDINTA

Tässä opinnäytetyössä oli tarkoitus kehittää Tampereen ammattikorkeakoulun kone- ja tuotantotekniikan koulutuksessa käytettyä opetusalustaa liittämällä se internetiin. Tavoitteena oli pystyä langattomasti etäyhteydellä ohjaamaan tai valvomaan opetusalustan liukuhihnaa ja konenäkökameraa. Näihin tavoitteisiin on päästy tämän työn aikana. Työn tuloksena liukuhihna pystytään ohjaamaan etäyhteydellä ja on luettavissa, onko hihna käynnissä vai sammuksissa. Lisäksi pystytään valvomaan, montako esinettä kulkee liukuhihnaa pitkin konenäkökameran tunnistettavaksi.

Työn haastavin osuus oli aiheeseen liittyvän kirjallisuuden vähäisyys. Suomenkielistä aineistoa oli erittäin vähän ja englanniksikin oli vaikeuksia löytää vastauksia spesifiin kysymyksiin. IoT Studiosta ei löytynyt lähes ollenkaan käyttöohjeita, mikä hankaloitti ohjelman perusteiden oppimista. Gateway-laitteelle löytyi hyvät asennusohjeet, mutta Siemensin ohjelmoitavan logiikan yhdistäminen IoT Studioon osoittautui haastavaksi ilman ohjeistusta.

Opetusalustan toiminnasta on luovutettu koululle käyttöohjeet kolmen harjoitustehtävän muodossa. Harjoitustehtävät opettavat käytännön työn kautta, mitä esineiden internet on. Ne havainnollistavat, miten fyysisen laitteen voi kytkeä internetiin ja mitkä ovat sen tuomat hyödyt. Tehtävissä on tätä opinnäytetyöraporttia yksityiskohtaisemmin kerrottu, miten kytkennät ja logiikkojen ohjelmat on tehty. Harjoitustehtävät ovat tarpeeksi tarkat, jotta tässä työssä luotu kokonaisuus on mahdollista toteuttaa uudestaan opetuksessa.

Opetukseen luoduilla harjoitustehtävillä on tarkoitus tutustuttaa opiskelijat IoT:hen ja sen ohjelmointiin. Aiheen käytännön puoli on uutta Tampereen ammattikorkeakoulussa, minkä vuoksi harjoitustehtävät ovat yksinkertaisia. Tulevaisuudessa olisi mahdollista laatia vaativampia tehtäviä, jotta opiskelijat pääsisivät suunnittelemaan monimutkaisempia ohjelmia. IoT Studioon voisi perehtyä enemmän ja tutustua sen lukuisiin toimintoihin. Olisi myös mahdollista tutkia, löytyykö koulusta muita laitteita tai sensoreita, joita voisi kytkeä internetiin. Nämä toisivat



uusia mahdollisuuksia IoT Studion ohjelmointiin sekä aihealueen syvempään perehtymiseen.

## LÄHTEET

Ashton, K. 2009. That "Internet of Things" thing. RFID Journal. Luettu 14.1.2019 <https://www.rfidjournal.com/articles/view?4986>

Bouhai, N. & Saleh, I. 2017. Internet of Things. Iso-Britannia: Wiley-ISTE.

Cognex. N.d. Cognex – The Leader in Machine Vision. Luettu 29.1.2019. <https://www.cognex.com/company>

Cognex. N.d. In-Sight Explorer Software. Luettu 29.1.2019. <https://www.cognex.com/products/machine-vision/2d-machine-vision-systems/in-sight-vision-software>

Collin, J. & Saarelainen, A. 2016. Teollinen internet. Helsinki: Talentum.

Dehouck, R. 2015. The maturity of visual programming. Blogi. Luettu 4.2.2019. <https://www.craft.ai/blog/the-maturity-of-visual-programming/>

Devopedia. 2017. IoT Operating Systems. Luettu 21.3.2019. <https://devopedia.org/iot-operating-systems>

Evans, P. & Annunziata, M. 2012. Industrial Internet: Pushing the Boundaries of Minds and Machines. Verkkokirja. General Electric. Luettu 18.1.2019 [https://www.ge.com/docs/chapters/Industrial\\_Internet.pdf](https://www.ge.com/docs/chapters/Industrial_Internet.pdf)

Gabbai, A. 2015. Kevin Ashtonin haastattelu. Smithsonian Magazine. Luettu 14.1.2019. <https://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/>

General Electric. N.d. Everything you need to know about the Industrial Internet of Things. Blogi. Luettu 18.1.2019. <https://www.ge.com/digital/blog/everything-you-need-know-about-industrial-internet-things>

General Electric. 2013. Industrial Internet: A European Perspective. Luettu 18.1.2019. <https://www.genewsroom.com/sites/default/files/media/201306/5901.pdf>

Haikala, I. & Järvinen, H.M. 2004. Käyttöjärjestelmät. Toinen painos. Helsinki: Talentum.

Hanes, D., Salgueiro, G. & Barton, R. 2017. IoT Fundamentals: Networkin Technologies, Protocols, and Use Cases for the Internet of Things. Cisco Press.

IoT World Forum. 2014. The Internet of Things Reference Model. Luettu 21.1.2019. [http://cdn.iotwf.com/resources/71/IoT\\_Reference\\_Model\\_White\\_Paper\\_June\\_4\\_2014.pdf](http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf)

Karkkulainen, J., Karkkulainen, O. & Kinnunen, A. 2014. eFYYSIikka 9. e-Oppi. Luettu 1.4.2019.

<https://peda.net/oppimateriaalit/e-oppi/verkkokauppa/yläkoulu/poistuneet-tuotteet/e9k1/3srek>

MAOL ry. N.d. Ohjelmointiympäristö. Luettu 4.2.2019. <https://peda.net/yhdistykset/maol-ry/materiaalit/kpm/7-luokka/p1/yleistä/1o/1o>

Meola, A. 2018. What is the Internet of Things (IoT)? Meaning & Definition. Businessinsider. Luettu 14.1.2019. <https://www.businessinsider.com/internet-of-things-definition?r=US&IR=T&IR=T>

Nexcom. 2011. IoT Studio. Luettu 29.1.2019. <http://www.nexcom.com/Products/industrial-computing-solutions/iot-solutions/iot-studio>

Node-RED. N.d. Node-RED: Flow-based programming for the Internet of Things. Luettu 4.2.2019. <https://nodered.org/>

Siemens. N.d. TIA Portal – teollisuusautomaation ohjelmistoalusta. Luettu 29.1.2019. [http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/tia\\_portal.php](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/tia_portal.php)

Siemens. N.d. Automaatio-ohjelmistot. Luettu 29.1.2019. [http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/automaatiotekniikka/automaatio\\_ohjelmistot.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/automaatio_ohjelmistot.htm)

Siemens support. 2017. Which ports are used by the various services for data transfer? Luettu 14.3.2019. <https://support.industry.siemens.com/cs/document/8970169/which-ports-are-used-by-the-various-services-for-data-transfer-via-tcp-and-udp-and-what-should-you-watch-out-for-when-using-routers-and-firewalls-?dti=0&lc=en-WW>

Tutorialspoint. N.d. Internet of Things – Software. Luettu 5.2.2019. [https://www.tutorialspoint.com/internet\\_of\\_things/internet\\_of\\_things\\_software.htm](https://www.tutorialspoint.com/internet_of_things/internet_of_things_software.htm)

Tutorialspoint. N.d. What is Node.js. Luettu 4.2.2019. [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)

## LIITTEET

### Liite 1. Liukuhinnan etävalvontaharjoitus

#### HARJOITUS: LIUKUHIHMAN ETÄVALVONTA

Harjoituksen tavoitteena on yhdistää koulun liukuhinna internetiin ja lukea liukuhinnan data etäyhteydellä IoT Studiolla. Harjoituksen päätteeksi tulee tietokoneella tai älypuhelimella saada IoT Studion kautta tietää, onko liukuhinna käynnissä vai sammutettu.

Harjoituksen toteuttamiseksi tarvittavat:

- liukuhinnan
- PLC:n (Siemens S7-300)
- gateway-laitteen (Nexcom CPS 100-RE)
- reitittimen
- kaapeleita ja verkkokaapeleita
- tietokoneen/älypuhelimien

Liitä liukuhinna PLC:hen kiinni ja kytke kokonaisuus tietokoneeseen. Luo TIA Portaliin yksinkertainen ohjelma, jolla saa liukuhinnan moottorin käyntiin ja sammutettua PLC:n fyysisestä kytkimestä (kuva 1 & 2). Kytke liukuhinna PLC:hen kiinni TIA Portalin ohjelman ulostulon mukaisesti. Lataa valmis ohjelma PLC:hen.



KUVA 1. TIA Portal -ohjelma

Default tag table							
	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Moottori	Bool	%Q136.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Käynnistys	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	

KUVA 2. Tag-lista

Kytke seuraavaksi gateway-laitteeseen virrat (24V) ja liitä se ethernet-kaapelilla reitittimeen. Kaapeli täytyy kytkeä gatewayn 2-liittimeen kuten kuvassa 3 on esitetty.

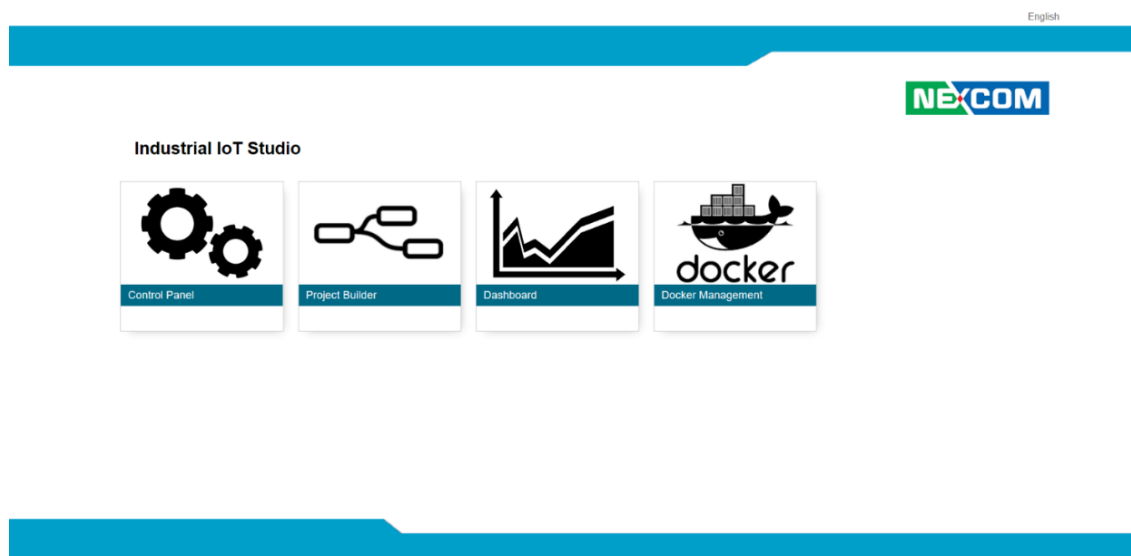
Tässä vaiheessa molemmat PLC:n ethernet-portit ovat varattuja, mutta toisen kaapeleista voi irrottaa, jos TIA Portal -ohjelma on jo ladattu PLC:hen. Yhdistä tietokone reitittimeen kiinni ethernet-kaapelilla. Kytke lopuksi PLC reitittimeen ethernet-kaapelilla. Reitittimen WLAN:iin on mahdollista kytkeytyä myös suoraan langattomasti, mutta koulun koneilla sitä ei ole mahdollista tehdä.

Myös älypuhelimella voi ohjelmoida IoT Studiota, mutta sitä ei suositella sen hankaluuden takia. Valmista työtä voi kuitenkin valvoa älypuhelimien kautta kytkeytymällä reitittimen WLAN:iin ja kirjautumalla IoT Studioon.



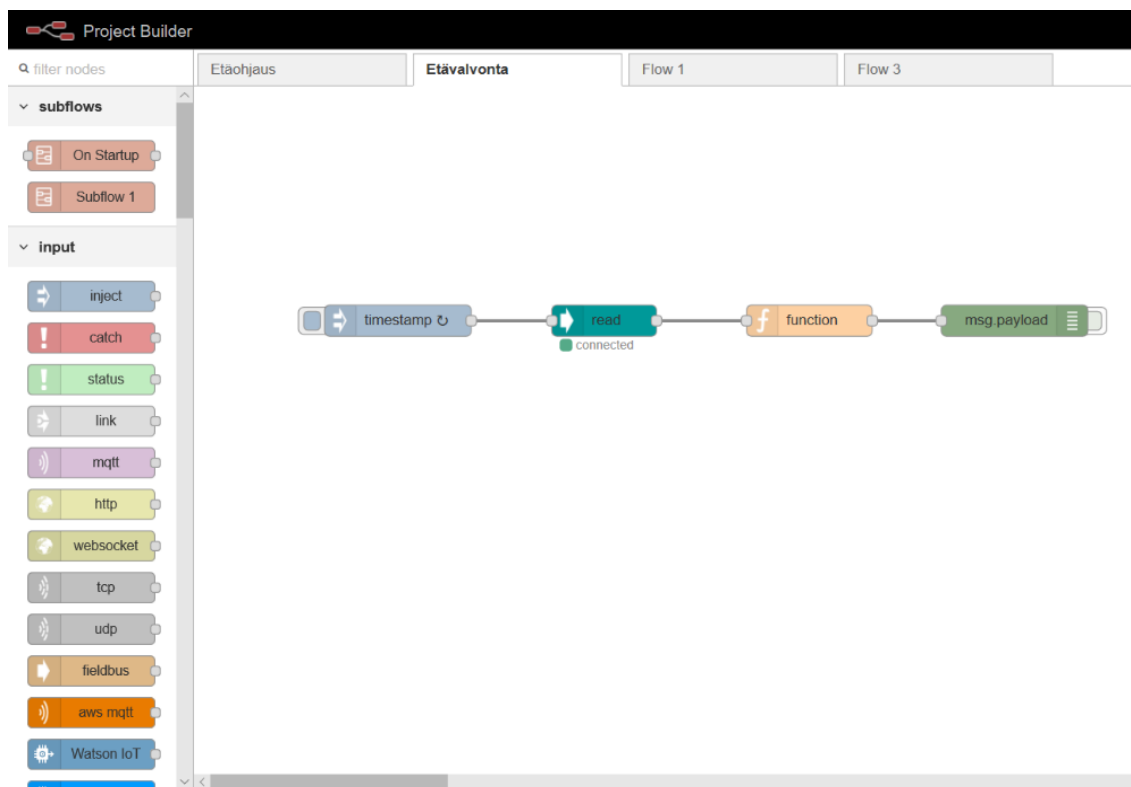
KUVA 3. Gatewayn kytkentä

IoT Studion saa avattua laittamalla tietokoneen tai puhelimen selaimen hakukenttään gatewayn IP-osoiteen. Tämän harjoituksen tekohetkellä koulun ainoan Nexcomin gatewayn IP-osoite on 192.168.50.30. Selaimeen aukeaa Nexcom Industrialin IoT Studion päävalikko (kuva 4). Gatewayn IP-osoitteen voi halutessa vaihtaa ”Control Panelin” kautta. IP-osoitteen tulee olla saman IP-lohkon alla PLC:n ja liukuhinnan kanssa eli muotoa 192.168.50.XX. Osoitteen tulee olla kuitenkin eri kuin muilla laitteilla. IoT Studio saattaa pyytää jossain vaiheessa käyttäjätunnuksen ja salasanan.



KUVA 4. IoT Studion päävalikko

Päävalikon Project Builder -napista pääsee luomaan ohjelman gateway-laitteeseen. Liukuhinnan moottorin tilan saa luettua IoT Studiolla kuvassa 5 luodulla komentorivillä eli ”flowlla”. Komennot eli ”nodet” lisätään flow-kenttään ja yhdistetään keskenään. Nodejen nimet saattavat muuttua, kun niitä lisää flow-kenttään. Nodejen asetuksia pääsee säättämään niistä kaksoisklikkaamalla. Nurkassa ylävasemmalla on hakukenttä nodeille etsinnän helpottamiseksi.



KUVA 5. Etävalvonnan flow

Ensimmäinen node on ”inject”, joka tässä tapauksessa on kuvassa 5 nimellä timestamp. Inject-noden tehtävä on syöttää flowlle haluttu käsky joko napinpainalluksella tai tietyin aikavälein. Nodeen säädetään käskyksi (payload) timestamp ja valitaan käskyn aikaväliksi yksi sekunti (kuva 6). Näillä asetuksilla node käskee sekunnin välein flowta ilmoittamaan sen tuloksesta.

The screenshot shows the 'Edit inject node' dialog box. It has a 'Cancel' button and a red 'Done' button. The 'Payload' is set to 'timestamp'. The 'Topic' field is empty. The 'Repeat' section is set to 'interval' with 'every 1 seconds'. There is an unchecked checkbox for 'Inject once at start?'. The 'Name' field is empty. A note at the bottom states: 'Note: "interval between times" and "at a specific time" will use cron. See info box for details.'

KUVA 6. Inject-noden asetukset

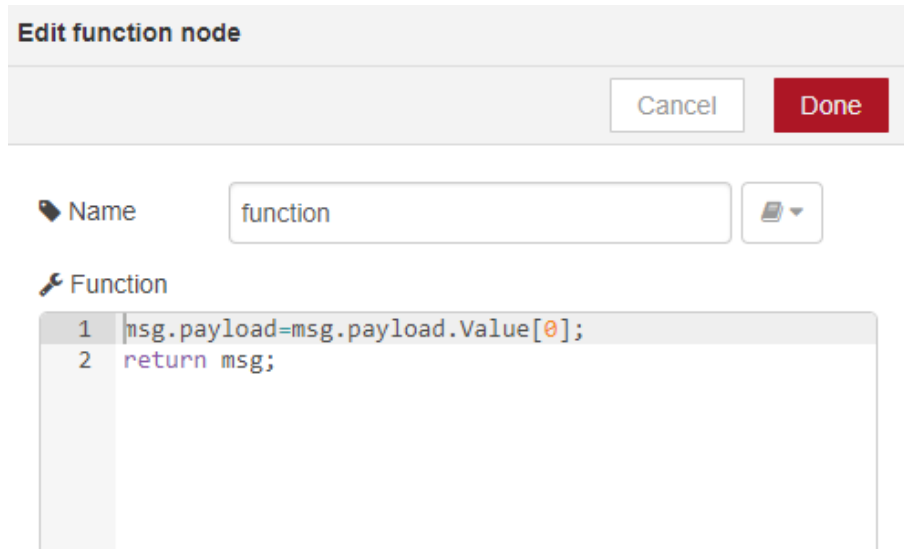
Toinen node on ”read S7”. Se on Siemensin S7-sarjan logiikoille luotu node. Se mahdollistaa yhteyden ja signaalien lukemisen IoT Studion ja logiikan välillä. Nodeen määritetään logiikan IP-osoite ja portti (kuva 7). Koulun Siemensin S7-300-logiikoissa on IP-osoitteena 192.168.50.20. Portin numeron pitäisi olla valmiiksi 102. Nodessa määritetään signaalit, joita halutaan tutkia eli tässä tapauksessa TIA Portalissa luodun ohjelman ulostulon osoite (offset) ja tyyppi (output & boolean).

The screenshot shows the 'Edit connection node' dialog box. At the top, there are three buttons: 'Delete', 'Cancel', and 'Update'. Below these is a section titled 'Connection parameter' containing four input fields: 'IP-Adress' with the value '192.168.50.20', 'Port' with '102', 'Rack' with '0', and 'Slot' with '2'. Below this is a 'Signals' section with a list of signals. Each signal entry consists of a type dropdown, a name, and an offset dropdown. The visible signals are: 'Outq' (type: Outq), 'Boo' (type: Boo), 'Offset: 136', and '0' (type: 0). The signal name 'Liukuhinnan t' is visible next to the '0' offset. There is a close button (X) and a scroll arrow on the right side of the signals list.

KUVA 7. Read S7 -noden asetukset

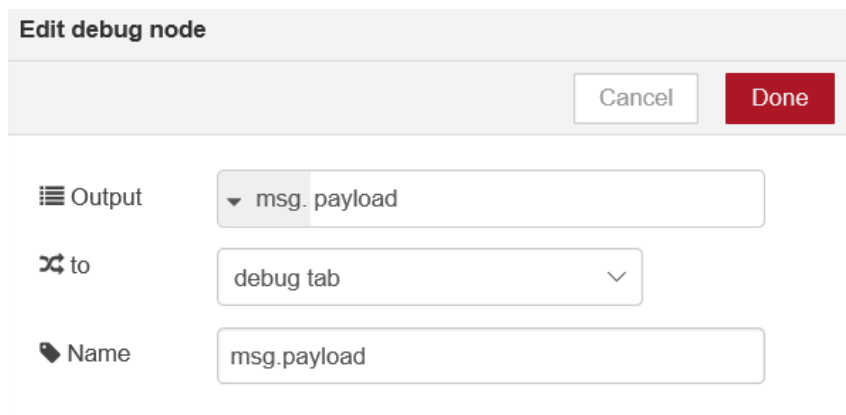
Seuraava node on nimeltään ”function”. Function-nodea käytetään, kun ei löydy haluttuun tehtävään valmista nodea. Node on vapaasti säädettävissä JavaScript-ohjelmointikielellä. Tässä tapauksessa kirjoitetaan ohjelma, joka poimii tiedon sisääntulon signaalista (kuva 8).





KUVA 8. Function-noden asetukset

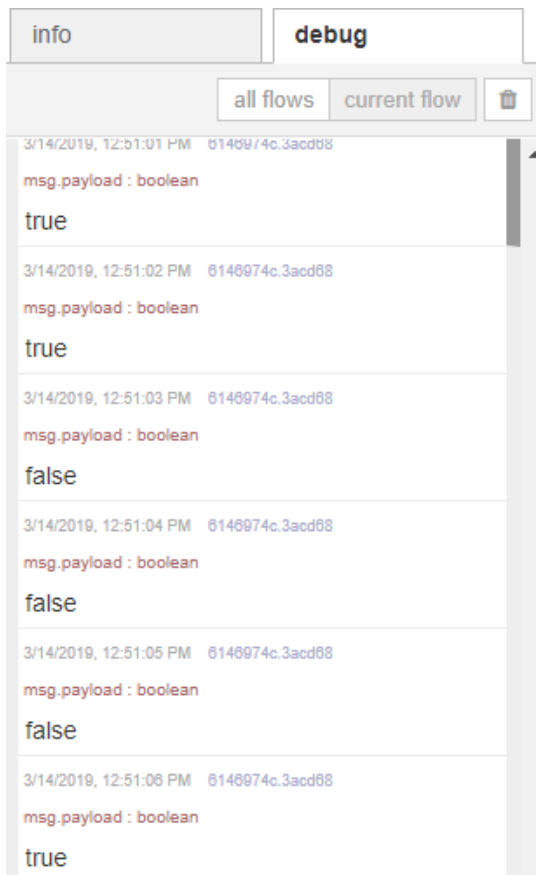
Flown viimeinen node on ”debug”. Sen nimi muuttuu automaattisesti msg.payloadiksi, kun node lisätään. Debug-node käynnistää ja sammuttaa luodun flown sekä julkaisee sen keräämän datan. Tässä tehtävässä kelpaavat noden alkuperäiset asetukset (kuva 9).



KUVA 9. Debug-noden asetukset

Klikkaa IoT Studion oikeassa yläkulmassa löytyvästä punaisesta ”deploy”-napista. Se tallentaa ja hyväksyy flowssa tehdyt muutokset. Read S7 -noden alle pitäisi syttyä vihreä valo ja ilmestyä sana ”connected”, jos sen asetukset ovat oikein. Debug-nodessa on kiinni vihreä nappi, jota painamalla saa flown tulostamaan liukuhihnan tilan. Saapuva data löytyy IoT Studion oikeasta nurkasta ”debug”-välilehdeltä. Välilehdelle tulostuvat sanat

”true” ja ”false” sekunnin välein riippuen siitä, onko liukuhihna päällä (kuva 10). True-sana tulostuu, kun PLC:n fyysinen kytkin on ”ON”-asennossa. False-sana sen sijaan tulostuu kytkimen ollessa ”OFF”-asennossa.



KUVA 10. Flown tulostama liukuhihnan tila

## Liite 2. Liukuhihnan etäohjausharjoitus

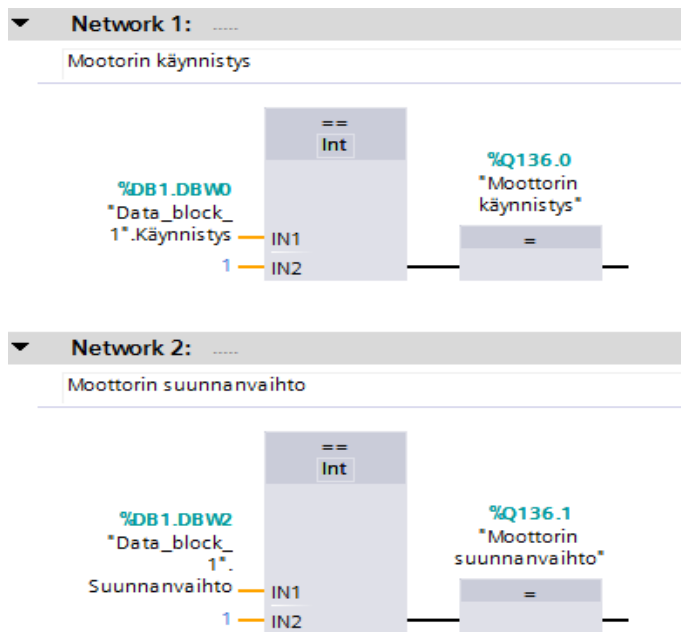
### HARJOITUS: LIUKUHIHMAN ETÄOHJAUS

Harjoituksen tavoitteena on yhdistää koulun liukuhihna internetiin ja ohjata sitä etäyhteydellä IoT Studiolla. Harjoituksen päätteeksi tulee tietokoneella tai älypuhelimella IoT Studion kautta pystyä käynnistämään ja sammuttamaan liukuhihna sekä vaihtamaan sen pyörimissuuntaa.

Harjoituksen toteuttamiseksi tarvittavat:

- liukuhihnan
- PLC:n (Siemens S7-300)
- gateway-laitteen (Nexcom CPS 100-RE)
- reitittimen
- kaapeleita ja verkkokaapeleita
- tietokoneen/älypuhelimien

Kytke liukuhihna ja PLC tietokoneeseen kiinni ja luo TIA Portaliin ohjelma, jolla liukuhihnaa saa ohjattua IoT Studiolla (kuva 1 & 2). IoT Studio lähettää datablock-välilehteen käskyt ”1” tai ”0”. Käskystä 1 liukuhihna käynnistyy tai vaihtaa suuntaansa. Käskystä 0 saa toiminnot sammutettua. Kytke liukuhihna PLC:hen kiinni ohjelman ulostulojen perusteella. Lataa valmis ohjelma PLC:hen.



KUVA 1. TIA Portal -ohjelma

Data_block_1				
	Name	Data type	Offset	Start value
1	Static			
2	Käynnistys	Int	0.0	0
3	Suunnanvaihto	Int	2.0	0

KUVA 2. Datablock-välilehti

Kytke seuraavaksi gateway-laitteeseen virrat (24V) ja liitä se ethernet-kaapelilla reitittimeen. Kaapeli täytyy kytkeä gatewayn 2-liittimeen kuten kuvassa 3 on esitetty.

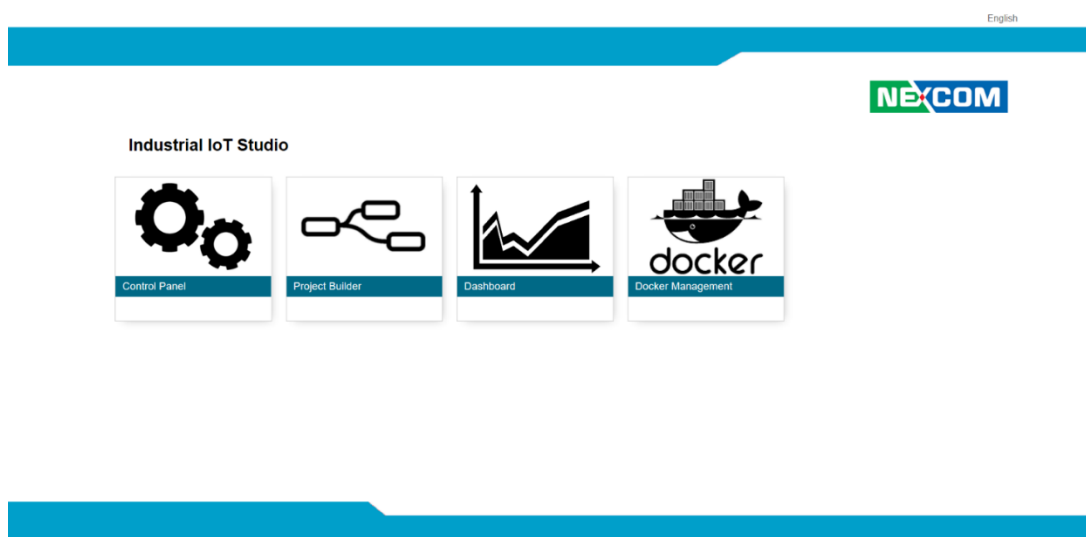
Tässä vaiheessa PLC:n molemmat ethernet-portit ovat varattuja, mutta toisen kaapeleista voi irrottaa, jos TIA Portal -ohjelma on jo ladattu PLC:hen. Yhdistä tietokone reitittimeen ethernet-kaapelilla. Kytke lopuksi PLC reitittimeen ethernet-kaapelilla. Reitittimen WLAN:iin on mahdollista kytkeytyä myös suoraan langattomasti, mutta koulun koneilla sitä ei ole mahdollista tehdä.

Myös älypuhelimella voi ohjelmoida IoT Studiota, mutta sitä ei suositella sen hankaluuden takia. Valmista työtä voi kuitenkin valvoa älypuhelimien kautta kytkeytymällä reitittimen WLAN:iin ja kirjautumalla IoT Studioon.



KUVA 3. Gatewayn kytkentä

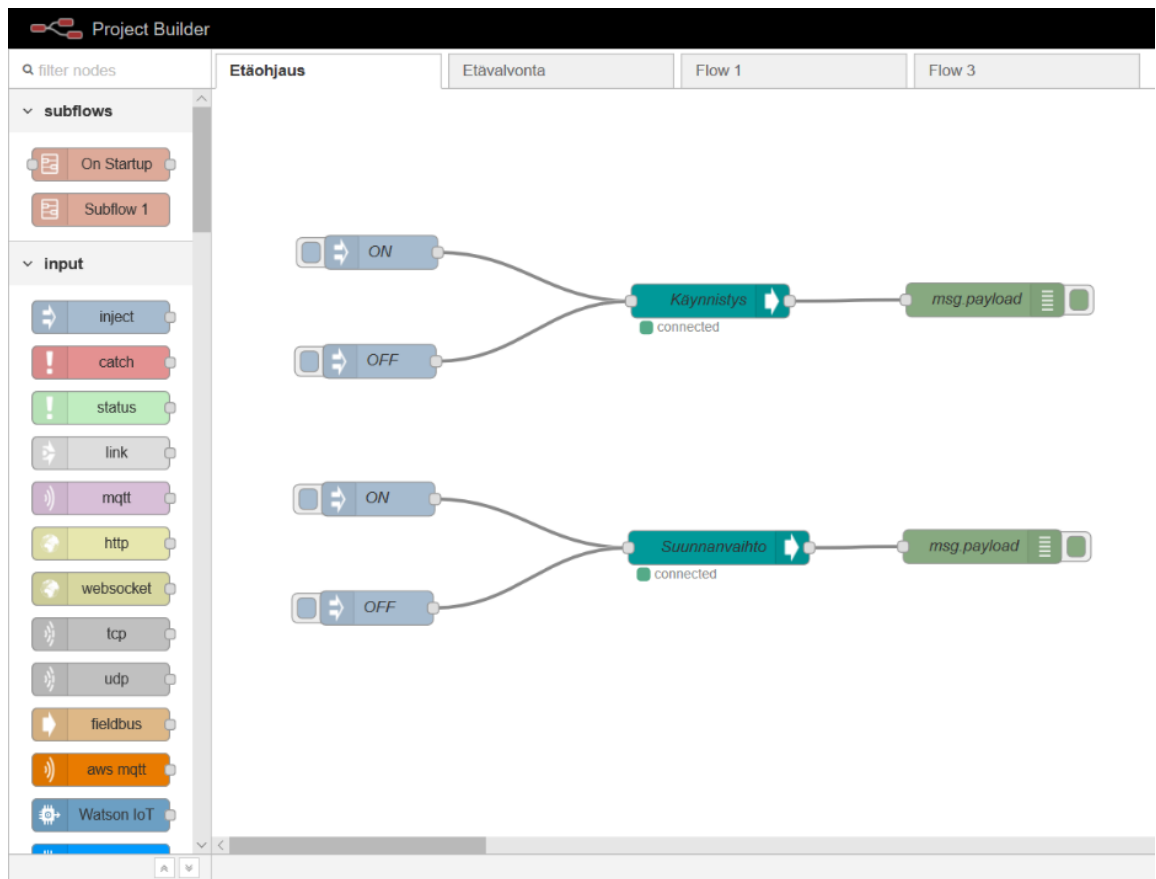
IoT Studion saa avattua laittamalla tietokoneen tai puhelimen selaimen hakukenttään gatewayn IP-osoiteen. Tämän harjoituksen tekohetkellä koulun ainoan Nexcomin gatewayn IP-osoite on 192.168.50.30. Selaimen aukeaa Nexcom Industrialin IoT Studion päävalikko (kuva 4). Gatewayn IP-osoitteen voi halutessa vaihtaa ”Control Panelin” kautta. IP-osoitteen tulee olla saman IP-lohkon alla PLC:n ja liukuhinnan kanssa eli muotoa 192.168.50.XX. Osoitteen tulee olla kuitenkin eri kuin muilla laitteilla. IoT Studio saattaa pyytää jossain vaiheessa käyttäjätunnuksen ja salasanan.



KUVA 4. IoT Studion päävalikko

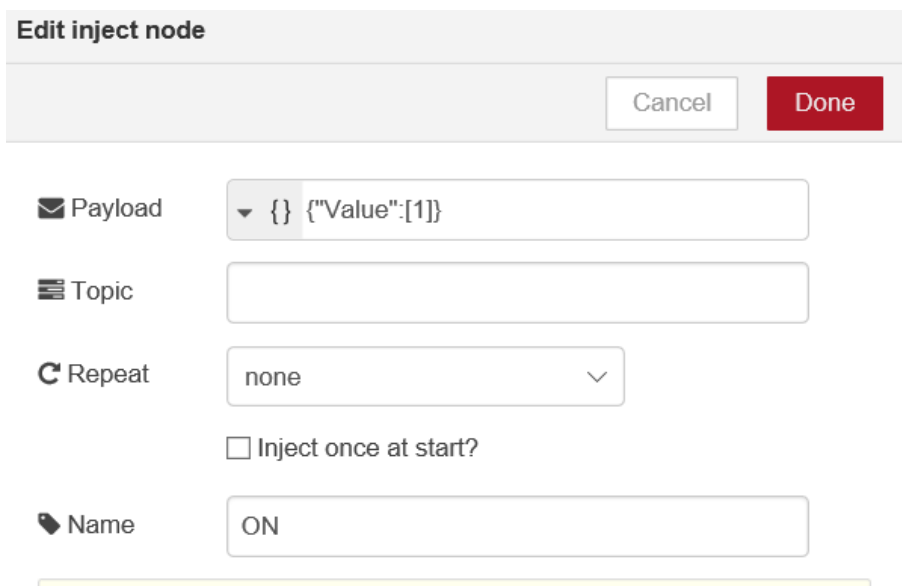
Päävalikon Project Builder -napista pääsee luomaan ohjelman gateway-laitteeseen. Komennot eli ”nodet” lisätään flow-kenttään ja yhdistetään keskenään. Nodejen nimet saattavat muuttua, kun niitä lisää flow-kenttään. Nodejen asetuksia pääsee säätämään niistä kaksoisklikkaamalla.

Liukuhinnan moottoria saa ohjattua IoT Studiolla kuvassa 5 luoduilla komentoriveillä eli ”flowlla”. Flowssa on erikseen komentorivi liukuhinnan käynnistykselle ja suunnanvaihdolle. Rivit ovat muuten täsmälleen samanlaiset keskenään, mutta kumpikin ohjaa yhden logiikan ulostuloa.



KUVA 5. Etäohjauksen flow

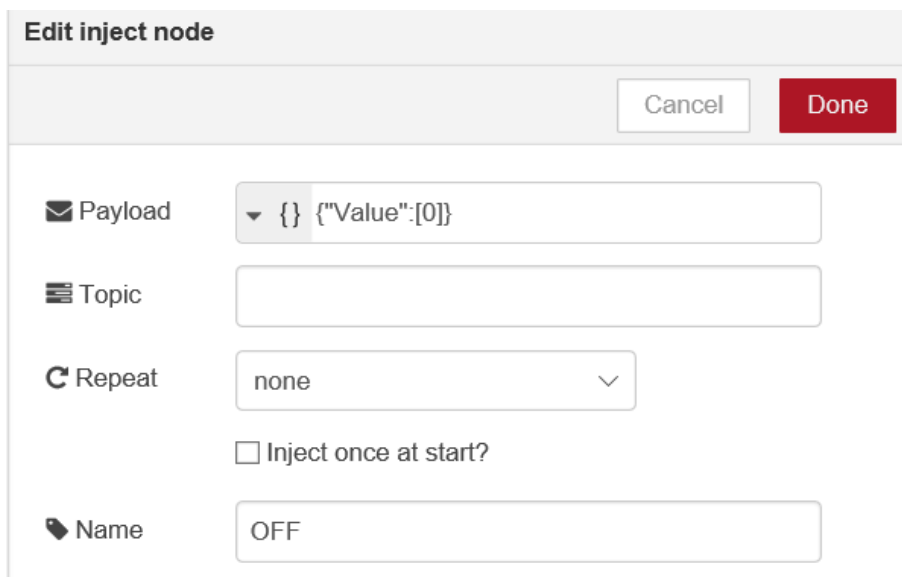
Flowssa on neljä inject-nodea, joilla saa ohjattua liukuhihnaa. Nodet on nimetty ”ON”- ja ”OFF”-napeiksi, joita painamalla saa liukuhihnan käynnistettyä, vaihtamaan suuntaa ja sammutettua. Inject-nodeille määritetään käskyksi (payload) JSON-ohjelmointikielellä komentorivi `{”Value”: [X]}`. X:n tilalle laitetaan ON-nappeihin ”1” (kuva 6) ja OFF-nappeihin ”0” (kuva 7).



The screenshot shows the 'Edit inject node' dialog box with the following settings:

- Cancel** and **Done** buttons at the top right.
- Payload:** A dropdown menu showing a selected option with a downward arrow and the text `{ } {"Value":[1]}`.
- Topic:** An empty text input field.
- Repeat:** A dropdown menu showing the selected option `none` and a downward arrow.
- Inject once at start?:** An unchecked checkbox.
- Name:** A text input field containing the text `ON`.

KUVA 6. Inject-noden "ON"-asetukset



The screenshot shows the 'Edit inject node' dialog box with the following settings:

- Cancel** and **Done** buttons at the top right.
- Payload:** A dropdown menu showing a selected option with a downward arrow and the text `{ } {"Value":[0]}`.
- Topic:** An empty text input field.
- Repeat:** A dropdown menu showing the selected option `none` and a downward arrow.
- Inject once at start?:** An unchecked checkbox.
- Name:** A text input field containing the text `OFF`.

KUVA 7. Inject-noden "OFF"-asetukset

Komentorivien seuraava node on Write S7, joka on Siemensin logiikoille suunniteltu. Se mahdollistaa kirjoittamisen Siemensin logiikoihin IoT Studiolla. Nodeen määritetään logiikan IP-osoite ja portti (kuva 8). Koulun Siemensin S7-300-logiikoissa on IP-osoitteena



192.168.50.20. Portin numeron pitäisi olla valmiiksi 102. Lisäksi määritetään TIA Portallissa luodut datablockin osoitteet (offset) ja tyyppi (datablock & Int), joihin käskyt lähetetään.

KUVA 8. Write-S7 -noden asetukset

Write S7 -nodeja on flowssa erikseen käynnistykselle ja suunnanvaihdolle. Osoitteet pysyvät kummassakin samoina, mutta se valitaan, kumpaan signaaliin käskyt kirjoitetaan (kuvat 9 & 10).

KUVA 9. Write S7 käynnistykselle

KUVA 10. Write S7 suunnanvaihdolle

Flown viimeinen node on ”debug”. Sen nimi muuttuu automaattisesti msg.payloadiksi, kun node lisätään. Debug-node käynnistää ja sammuttaa luodun flown, sekä julkaisee sen keräämän datan. Tässä tehtävässä kelpaavat noden alkuperäiset asetukset (kuva 11).

KUVA 11. Debug-noden asetukset

Klikkaa IoT Studion oikeasta yläkulmasta löytyvästä punaisesta ”deploy”-napista. Se tallentaa ja hyväksyy flowssa tehdyt muutokset. Write S7 -nodejen alle pitäisi syttyä vihreä valo ja ilmestyä sana ”connected”, jos niiden asetukset ovat oikein. Debug-nodeissa on kiinni vihreät napit, joita painamalla saa flown tulostamaan liukuhihnan tilan. Lähtevät

käskyt löytyvät IoT Studion oikeasta nurkasta ”debug”-välilehdeltä (kuva 12). Välilehdelle tulostuu ilmoituksia aina, kun käskyjä lähetetään logiikalle. Etävalvonnan flow lähettää Siemensin logiikalle ON- ja OFF-napeista käskyt ”0” tai ”1”, minkä perusteella logiikka reagoi ja toimii. Flown käynnistysrivistä saa liukuhinnan päälle ja suunnanvaihdosta vaihtuu sen pyörimissuunta.



KUVA 12. Flown tulostamat etäohjauksen käskyt

### Liite 3. Konenäkökameran etävalvontaharjoitus

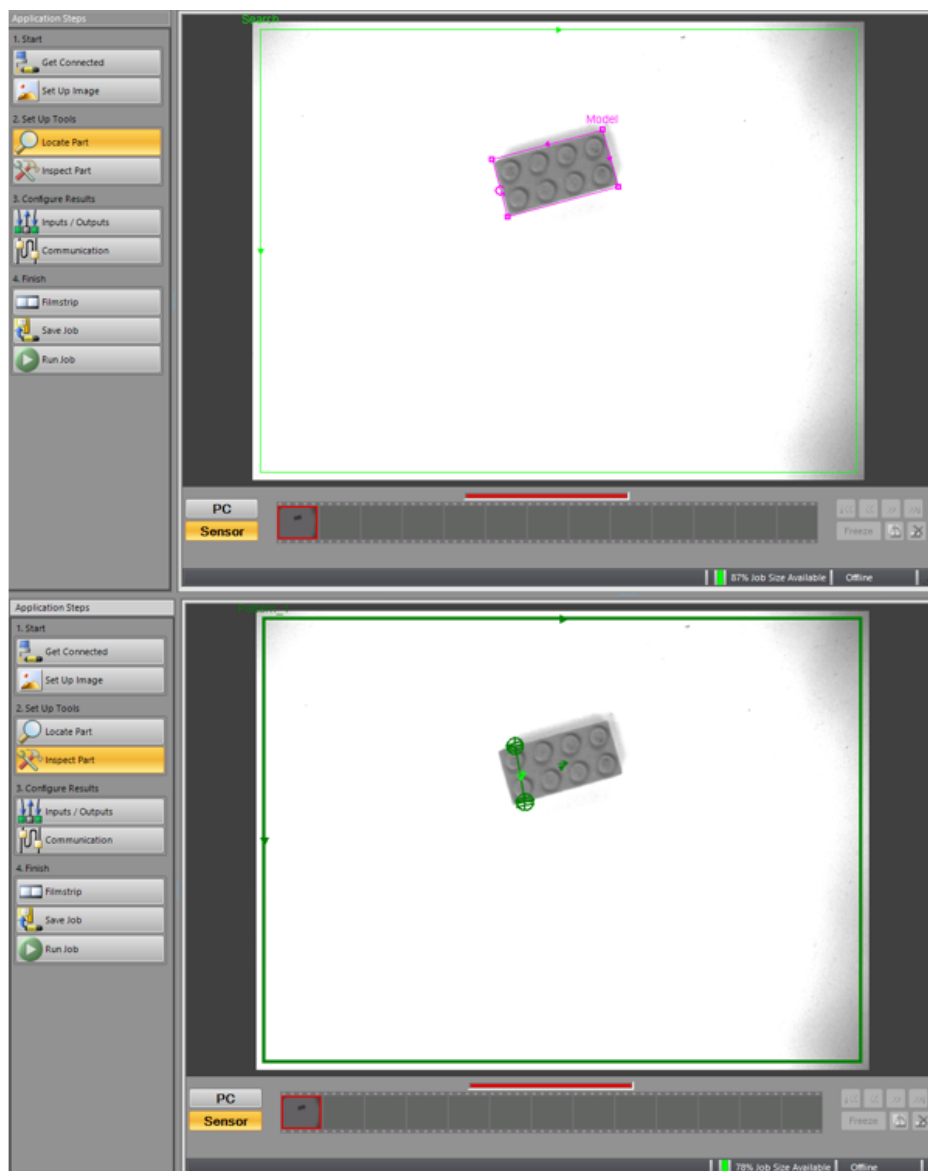
#### HARJOITUS: KONENÄKÖKAMERAN DATAN ETÄVALVONTA

Harjoituksen tavoitteena on yhdistää Cognexin konenäkökamera internetiin ja lukea sen avulla kameran data etäyhteydellä IoT Studiolla. Kamera ohjelmoidaan laskemaan, montako Lego-palikkaa kulkee sen ohitse. IoT Studiossa ei ole valmista nodea eli käskyä Cognexin kameroille, joten dataa ei voi lähettää suoraan kameralta IoT Studioon. Ongelman ratkaisemiseksi kamerasta saatava data lähetetään Siemensin ohjelmoitavaan logiikkaan, josta sen lukeminen on mahdollista IoT Studiolla. Harjoituksen päätteeksi tietokoneelle tai älypuhelimelle tulee saada IoT Studion kautta tieto, montako Legoa kamera on tunnistanut.

Harjoituksen toteuttamiseksi tarvittavat:

- konenäkökameran (Cognex 1403)
- kameran I/O-moduuli (Cognex CIO-MICRO)
- PLC:n (Siemens S7-300)
- gateway-laitteen (Nexcom CPS 100-RE)
- reitittimen
- kaapeleita ja verkkokaapeleita
- tietokoneen/älypuhelimien

Kytke kamera Cognexin CIO-MICRO I/O -moduuliin, josta data kulkeutuu verkkokaapelin välityksellä tietokoneelle In-Sight Explorer -ohjelmaan. Luo kameralle ohjelma Lego-palikan tunnistamiseksi. Luo ohjelmaan palikan muodon ja leveysmitan tunnistus (kuva 1). Säädä kamera lähettämään I/O-moduulin ulostulon 0 kautta signaali, kun se tunnistaa Legon leveysmitan (kuva 2).



KUVA 1. Legon muodon ja leveyden tunnistus

Name	Signal Type	Edge Type	Job Result
Line 0	Job Result		Distance_1Pass
Line 1	Job Result		Undefined
Line 2	Job Result		Undefined
Line 3	Job Result		Undefined
Line 4	Job Result		Undefined
Line 5	Job Result		Undefined
Line 6	Job Result		Undefined
Line 7	Job Result		Undefined

KUVA 2. Signaali ulostulolle 0 Legon leveysmitan tunnistuksessa

Kytke PLC:hen virrat ja liitä se verkkokaapelilla tietokoneeseen ja kytke johto sisääntulon 136 ja I/O-moduulin ulostulon 0 välillä. Ohjeita I/O-moduulin ulostulojen kytkennästä PLC:hen löytyy kuvasta 3, joka on otettu Cognexin sivuilta.

### Output to PLC (I/O Module Sources Current)

The PLC input is energized by a +24VDC signal from the vision system.

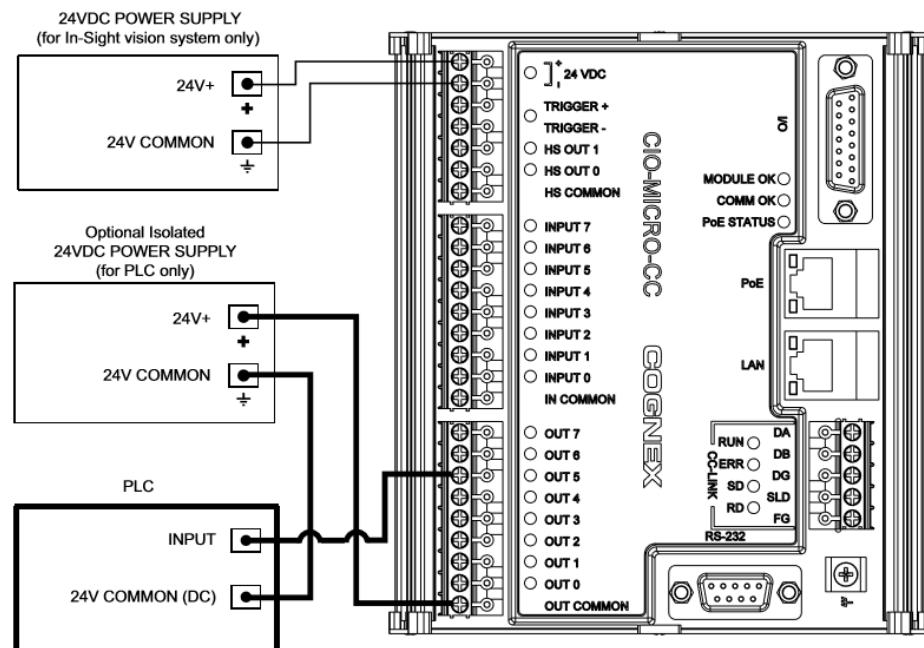
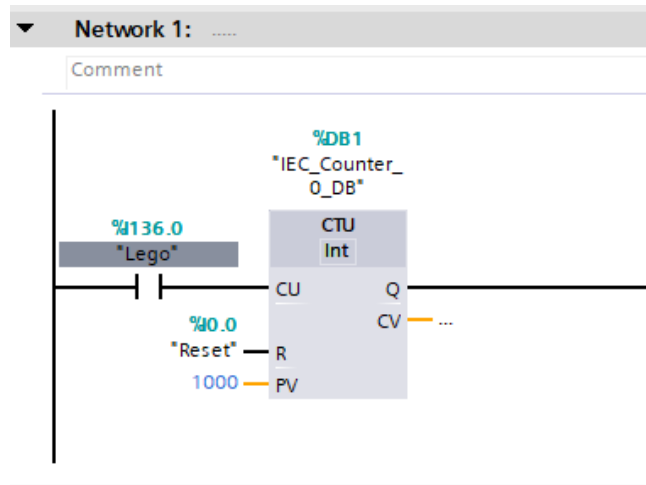


Figure A-5: Output to PLC (I/O Module Sources Current)

KUVA 3. Kytkennät I/O-moduulin ulostulojen ja PLC:n välillä

I/O-moduulin ulostulosta kulkee johtoa pitkin tieto Lego-palikan tunnistuksesta Siemen-  
sin logiikan sisääntuloon 136. Luo TIA Portaliin ohjelma laskemaan, montako kertaa si-  
sääntuloon tulee signaali Lego-palikasta (kuva 4). Lisää laskurin nollaamiseksi fyysinen  
kytkin sisääntulolla 0. Kuvassa 5 on laskurin automaattisesti luotu datablock-välilehti.



KUVA 4. TIA Portalin laskuri

IEC_Counter_0_DB							
	Name	Data type	Offset	Start value	Retain	Visible in ...	Setpoint
1	Input						
2	CU	Bool	0.0	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	R	Bool	0.1	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	PV	Int	2.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	Output						
6	Q	Bool	4.0	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	CV	Int	6.0	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	InOut						
9	Static						
10	CUO	Bool	8.0	FALSE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

KUVA 5. Laskurin datablock

Kytke seuraavaksi gateway-laitteeseen virrat (24 V) ja liitä se ethernet-kaapelilla reitittimeen. Kaapeli täytyy kytkeä gatewayn 2-liittimeen kuten kuvassa 6 on esitetty.

Tässä vaiheessa PLC:n molemmat ethernet-portit ovat varattuja, mutta toisen kaapeleista voi irrottaa, jos TIA Portal -ohjelma on jo ladattu PLC:hen. Yhdistä tietokone reitittimeen ethernet-kaapelilla. Kytke lopuksi PLC reitittimeen ethernet-kaapelilla. Reitittimen WLAN:iin on mahdollista kytkeytyä myös suoraan langattomasti, mutta koulun koneilla sitä ei ole mahdollista tehdä.

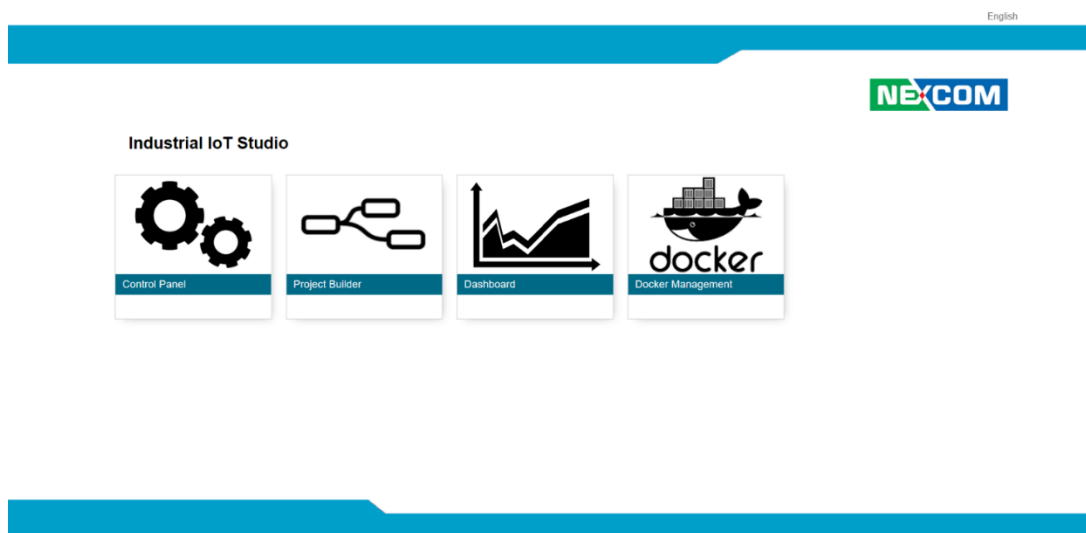
Myös älypuhelimella voi ohjelmoida IoT Studiota, mutta sitä ei suositella sen hankaluuden takia. Valmista työtä voi kuitenkin valvoa älypuhelimien kautta kytkeytymällä reitittimen WLAN:iin ja kirjautumalla IoT Studioon.



KUVA 6. Gatewayn kytkentä

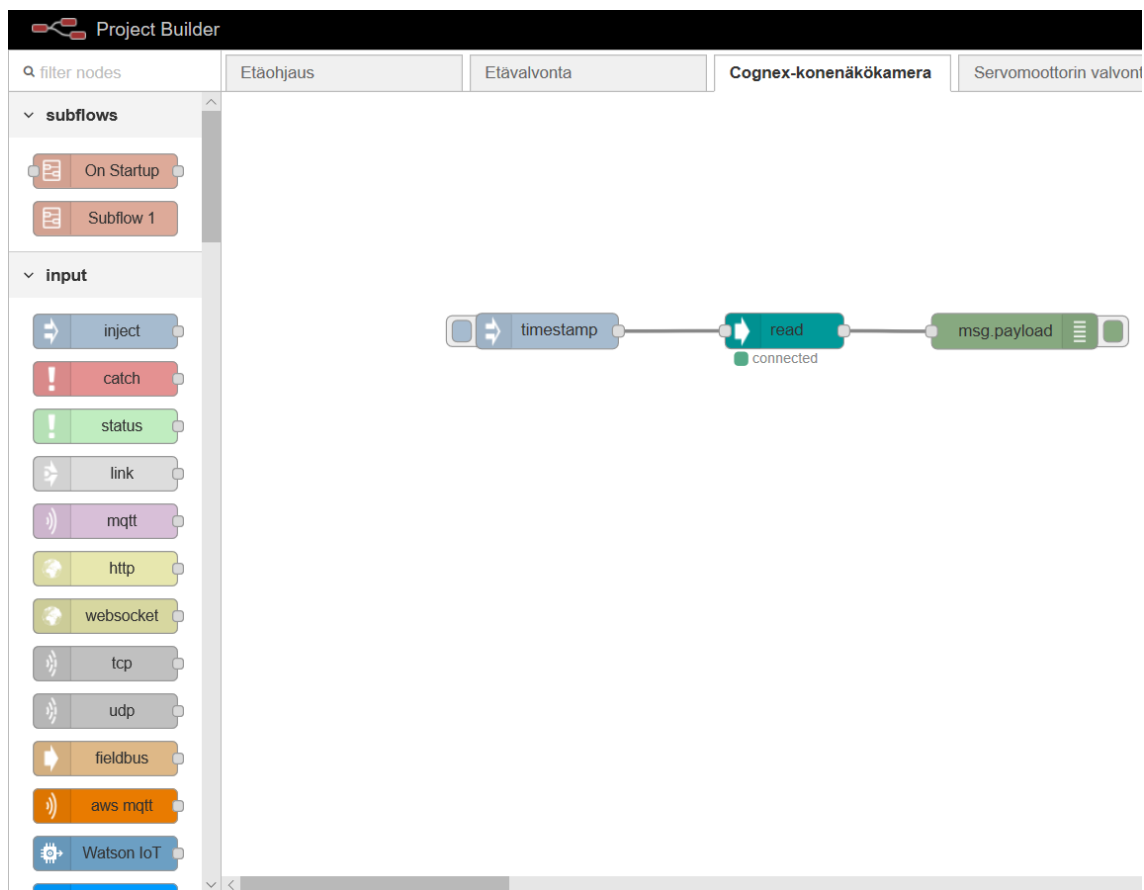
IoT Studion saa avattua laittamalla tietokoneen tai puhelimen selaimen hakukenttään gatewayn IP-osoiteen. Tämän harjoituksen tekohetkellä koulun ainoan Nexcomin gatewayn IP-osoite on 192.168.50.30. Selaimen aukeaa Nexcom Industrialin IoT Studion päävalikko (kuva 7). Gatewayn IP-osoitteen voi halutessa vaihtaa ”Control Panelin” kautta. IP-osoitteen tulee olla saman IP-lohkon alla PLC:n ja liukuhinnan kanssa eli 192.168.50.XX. Osoitteen tulee olla kuitenkin eri kuin muilla laitteilla. IoT Studio saattaa pyytää jossain vaiheessa käyttäjätunnuksen ja salasanan.





KUVA 7. IoT Studion päävalikko

Päävalikon Project Builder -kohdasta pääsee luomaan ohjelman gateway-laitteeseen. Kokenäkökameran lähettämän datan saa luettua IoT Studiolla kuvassa 8 luodulla komentorivillä eli ”flowlla”. Komennot eli ”nodet” lisätään flow kenttään ja yhdistetään keskenään. Nodejen nimet saattavat muuttua, kun niitä lisätään flow-kenttään. Nodejen asetuksia pääsee säätämään niistä kaksoisklikkaamalla.



KUVA 8. Konenäkökameran flow

Konenäkökameralle luotu flow ilmoittaa TIA Portalin laskurin arvon. Flown nodet ovat inject, read S7 sekä debug.

Flowssa luetaan Siemensin logiikalta tulevaa dataa eli inject-noden asetuksissa on timestamp sekä toisto sekunnin välein (kuva 9). Sillä saa luettua saapuvan datan sekunnin välein.

**Edit inject node**

Cancel
Done

✉ Payload ▼ timestamp

📄 Topic

🔄 Repeat interval ▼

every 1 ▲ ▼ seconds ▼

Inject once at start?

KUVA 9. Inject-noden asetukset

Toinen node on ”read S7”. Se on Siemensin S7-sarjan logiikoille luotu node. Se mahdollistaa yhteyden ja signaalien lukemisen IoT Studio ja logiikan välillä. Nodeen määritetään logiikan IP-osoite ja portti (kuva 10). Koulun Siemensin S7-300-logiikoissa on IP-osoitteena 192.168.50.20. Portin numeron pitäisi olla valmiiksi 102. Luettavaksi signaaliksi valitaan TIA Portalilla luodun datablock-välilehden CV:n osoite (offset). CV:stä on luettavissa laskurin Lego-palikkoiden lukumäärä. Lisäksi signaaliin määritetään CV:n tyyppi (datablock) ja muoto (Int).

read > **Edit connection node**

Delete

Connection parameter

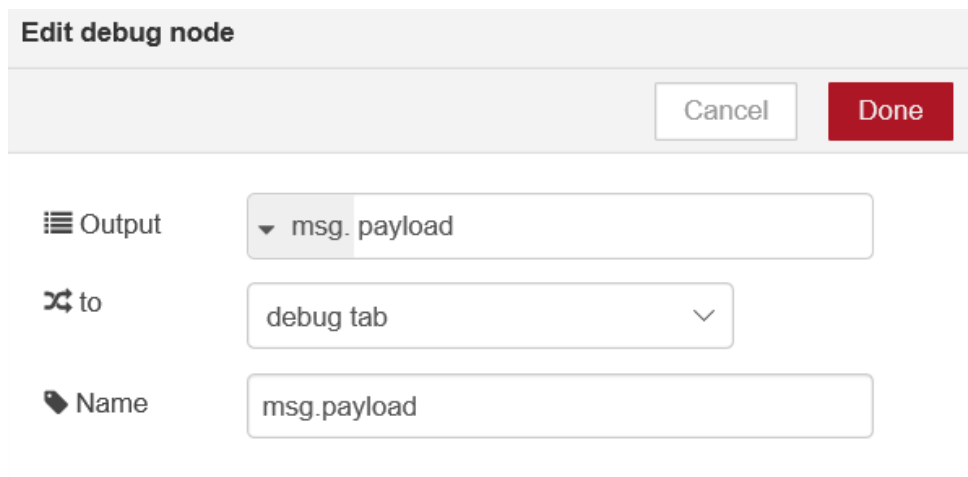
IP-Adress:	<input style="width: 95%;" type="text" value="192.168.50.20"/>	Port:	<input style="width: 95%;" type="text" value="102"/>
Rack:	<input style="width: 95%;" type="text" value="0"/>	Slot:	<input style="width: 95%;" type="text" value="2"/>

Signals:

Databloc ▼
Num: 
Int ▼
Offset: 
Quantity: 
Laskuri

KUVA 10. Read S7 -noden asetukset

Flown päättää debug-node, jolla saa read S7 -nodessa luetun signaalin data julkaistua IoT Studiossa. Sen nimi muuttuu automaattisesti msg.payloadiksi, kun node lisätään. Debug-node käynnistää ja sammuttaa luodun flown sekä julkaisee sen keräämän datan. Noden alkuperäisiä asetuksia ei tarvitse muuttaa (kuva 11).



The image shows a dialog box titled "Edit debug node". At the top right, there are two buttons: "Cancel" and "Done". Below the buttons, there are three configuration fields:

- Output:** A dropdown menu with a downward arrow, currently showing "msg.payload".
- to:** A dropdown menu with a downward arrow, currently showing "debug tab".
- Name:** A text input field containing "msg.payload".

KUVA 11. Debug-noden asetukset

Klikkaa IoT Studion oikeassa yläkulmassa löytyvästä punaisesta "deploy"-napista. Se tallentaa ja hyväksyy flowssa tehdyt muutokset. Read S7 -noden alle pitäisi syttyä vihreä valo ja ilmestyä sana "connected", jos sen asetukset ovat oikein. Debug-nodessa on kiinni vihreä nappi, jota painamalla saa flown tulostamaan liukuhihnan tilan. Saapuva data löytyy IoT Studion oikeasta nurkasta "debug"-välilehdeltä. Välilehdelle tulostuu sekunnin välein, montako Lego-palikkaa konenäkökamera on tunnistanut (kuva 12).

The screenshot shows a debug console interface with two tabs: 'info' and 'debug'. The 'debug' tab is active. Below the tabs are two buttons: 'all flows' and 'current flow', and a trash icon. The main area displays a list of messages. Each message entry consists of a timestamp, a unique ID, and a JSON payload. The messages are as follows:

- 29.3.2019 12.42.09 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 0 ] }
- 29.3.2019 12.42.42 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 0 ] }
- 29.3.2019 12.42.42 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 1 ] }
- 29.3.2019 12.42.52 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 2 ] }
- 29.3.2019 12.43.01 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 3 ] }
- 29.3.2019 12.43.09 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 4 ] }
- 29.3.2019 12.43.18 9500181d.d760b8  
msg.payload : Object  
{ "Signal": "Laskuri", "Path": "DB1,INT6", "Error": 0, "Value": [ 5 ] }

KUVA 12. Flown tulostama Lego-palikoiden lukumäärä