



Osaamista  
ja oivallusta  
tulevaisuuden  
tekemiseen

Teemu Voutilainen

# Prosessidatan hyödyntäminen kunnossapidon suunnittelussa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Konetekniikan tutkinto-ohjelma

Insinööriytyö

29.4.2019

Tekijä Otsikko	Teemu Voutilainen Prosessidatan hyödyntäminen kunnossapidon suunnittelussa
Sivumäärä Aika	35 sivua + 1 liite 29.4.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Konetekniikka
Ammatillinen pääaine	Koneautomaatio
Ohjaajat	Kunnossapitopäällikkö Aleksi Kyrö Lehtori Pekka Salonen
<p>Tämän insinööriyön tavoitteena oli selvittää, miten prosessidataa voisi hyödyntää kunnonvalvontakeinona, erityisesti Fazer Makeiset Oy:n Universal-linjalla.</p> <p>Työ aloitettiin tutustumalla alan kirjallisuuteen ja Universal-linjan toimintaan sekä haastatteleamalla Fazer Makeiset Oy:n kunnossapitopäällikköä linjan ennakkohuoltosuunnitelmista sekä Siemensin SIMANTIC PCS7:n ja MindSpheren tuotepäälliköitä, järjestelmien ominaisuuksien osalta.</p> <p>Kerätyn tiedon pohjalta pohdittiin, mistä Universal-linjan prosessin vaiheista saatavan datan kerääminen ja analysoiminen olisi mahdollista ja kunnonvalvonnan kannalta hyödyllistä. Lopuksi tehtiin pilotti prosessidatan hyödyntämisestä Universal-linjan muottiketjun pituuden muutoksen osalta ja teoriaa testattiin käytännössä.</p> <p>Pilotilla todistettiin, että teollisen internetin tuomat hyödyt ovat todellisia, eikä aina tarvitse asentaa erillisiä kunnonvalvonta-antureita.</p> <p>Työn lopputuloksena oli se, että Universal-linjalta tulevan prosessidatan perusteella olisi mahdollista tehdä kunnonvalvontaa jäähdytystunneleiden vesi-glykolikiertoon sekä muottiketjuun, sen pituuden muutoksen osalta.</p>	
Avainsanat	Kunnonvalvonta, kunnossapidon suunnittelu, IIoT

Author Title Number of Pages Date	Teemu Voutilainen Using Process Data as a Condition Monitoring Tool in Maintenance Planning 35 pages + 1 appendix 29 April 2019
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Professional Major	Machine Automation
Instructors	Aleksi Kyrö, Maintenance Manager Pekka Salonen, Senior Lecturer
<p>The objective of this Bachelor's thesis was to find out how process data could be used as a condition monitoring tool, especially at Fazer Confectionery Oy's Universal-production line.</p> <p>Working on the thesis was started with exploring topic-related literature and examining how the production line process works. Information was also gathered from interviews with Fazer Confectionery Oy's maintenance manager. Product managers from Siemens SIMANTIC PCS7 and MindSphere were interviewed as well.</p> <p>Based on the information gathered, the author examined in which phases of the Universal-production line's process it would be possible to gather data and analyze it with condition monitoring in mind. A pilot project was made for testing how to measure the stretch on the conveyor belt which moves the chocolate moulds through the production line.</p> <p>With the pilot project, it was possible to show that the benefits provided by the industrial Internet of Things are real. It also showed that you do not need additional sensors to carry out condition monitoring on machines.</p> <p>As a result, it was discovered that the process data from the Universal-production line can be used as a method to monitor the condition of the water-glycol circulation line for the cooling phase. In addition, the changes in length of the chocolate mould conveyor belt can be monitored.</p>	
Keywords	Condition monitoring, maintenance planning, IIoT

## Sisällys

### Lyhenteet

1	Johdanto	1
2	Kunnossapito	2
2.1	Kunnossapidon määritelmä	2
2.2	Kunnossapitolajit	2
2.2.1	Suunniteltu kunnossapito	3
2.2.2	Häiriökorjaukset	4
2.2.3	Kunnonvalvonta	4
2.3	Vika ja vikaantuminen	14
3	Prosessidatan hyödyntäminen yleisesti	16
3.1	Siemens SIMATIC PCS7 -automaatiojärjestelmä	16
3.1.1	PumpMon Block	17
3.1.2	VlvMon Block	20
3.2	MindSphere	23
4	Prosessidatan hyödyntäminen Universal-linjalla	25
4.1	Universal-linjan toiminta ja osat	25
4.1.1	Toiminta	25
4.1.2	Moottorit	27
4.1.3	Jäähdytystunnelit	28
4.1.4	Pumput	28
4.2	Prosessidatan hyödyntämismahdollisuudet Universal-linjalla	30
4.2.1	Valukoneen moottorit	30
4.2.2	Valukoneen pumput	30
4.2.3	Suklaaputkiston venttiilit	31
4.2.4	Jäähdytystunneli	31
4.2.5	Valukoneen muottiketju	32
4.3	Pilotti muottiketjun pituuden muutoksen mittaamisesta	33
5	Yhteenveto	35

Liite. Pilotin logiikkaohjelma

## Lyhenteet

CML	<i>Simatic PCS7 Condition Monitoring Library</i> , kunnonvalvontakirjasto, jota käytetään PCS7-ympäristössä.
IIoT	<i>Industrial Internet of Things</i> , teollinen esineiden internet.
IoT	<i>Internet of Things</i> , esineiden internet.
MTBF	<i>Mean Time Between Failures</i> , keskimääräinen aika laitteen vikaantumiseen sen edellisestä korjauksesta.
PLC	<i>Programmable Logic Controller</i> , ohjelmoitava logiikka.
rpm	<i>Rounds per minute</i> , kierrosta minuutissa.
RTF	<i>Run To Failure</i> , ajetaan vikaantumiseen asti ilman huoltoa.

## 1 Johdanto

Tämän insinööriyön aiheena on prosessidatan hyödyntäminen kunnossapidon suunnittelussa. Perinteisesti tuotantolaitoksissa kunnossapito on toteutettu häiriökorjauksia suorittamalla, panostamatta suuremmin kunnonvalvontaan tai ennakkohuoltoihin. Tämä on toimiva ratkaisu useimmissa kohteissa, joissa ennakkohuoltojen ja kunnonvalvonnan suorittaminen maksaa enemmän kuin häiriön aiheuttama vika ja sen korjaus. Kohteissa, joissa tuotantoseisokit aiheuttavat suuria tappioita, hyödytään kuitenkin panostamalla enemmän kunnonvalvontaan ja ennakoiwaan huoltoon. Näin voidaan vähentää suunnittelemtomia tuotantoseisokkeja ja lisätä tuotannon käyntivarmuutta.

Kunnonvalvontaa voidaan toteuttaa joko aistinvaraisesti tai mittaamiseen perustuen. Aisteihin perustuvan kunnonvalvonnan heikkoutena ovat erityisesti tulosten riippuminen ympäristöstä ja kunnonvalvonnan suorittajan aistiherkkyydestä. Mittaaminen taas on tarkempaa kuin aistinvarainen kunnonvalvonta. Se kuitenkin vaatii huomattavia resursseja. Mittaavaa kunnonvalvontaa voi kuitenkin tehdä myös ilman siihen erikseen tarkoitettujen anturien asentamista ja käyttämistä. Tuotantolaitteet sisältävät paljon antureita ja mitauspisteitä, joilla ohjataan prosessia. Prosessista itsestään on mahdollista löytää hyödyllistä dataa suoraan automaatiojärjestelmästä, logiikoilta tai taajuusmuuttajista. Insinööriyön tavoitteena on selvittää, miten näistä antureista saatavaa dataa voitaisiin hyödyntää muuhunkin käyttöön kuin prosessinohjaukseen, esimerkiksi löytämällä syy-yhteyksiä eri toimintojen välillä, jolloin voitaisiin seurata laitteen kuntoa reaaliajassa. Tätä dataa voisi käyttää apuna päätöksenteossa, kun ajoitetaan huoltoja tai kunnonvalvontakierroksia.

Työ on tehty Fazer Makeiset Oy:lle, ja sen tavoitteena on selvittää erityisesti, voiko teollisen internetin tuomia etuja hyödyntää yrityksen Universal-linjan kunnossapidon suunnittelussa. Universal on Fazer Makeiset Oy:lle 1980-luvulla tuotu suklaan valulinja. Linjan kunnonvalvonnassa ei tällä hetkellä hyödynnetä prosessidataa, mutta menettelystä voisi olla huomattavia hyötyjä linjan toiminnalle muun muassa tuotantoseisokkien vähenemisenä.

Työssä käydään ensin läpi teoriaa kunnossapidosta, kunnonvalvonnasta ja vikaantumisesta ja kerrotaan, miksi kunnonvalvontaa tarvitaan ja miten sitä voi tehdä. Tämän jälkeen kolmannessa luvussa esitellään prosessidatan yleisiä hyödyntämismahdollisuuksia, keskittyen erityisesti Siemens SIMATIC PCS7-automaatiojärjestelmään, joka on prosessinohjausjärjestelmien markkinajohtaja. Neljännessä luvussa esitellään toimeksianton kohteena oleva Universal-linja ja pohditaan, miten prosessidataa olisi mahdollista hyödyntää linjan eri komponenttien kunnonvalvonnassa. Lopussa esitellään aiheesta tehty pilottityö, jossa prosessidatan hyödyntämistä muottiketjun mittauksessa testattiin erillisellä pilottialustalla.

## 2 Kunnossapito

### 2.1 Kunnossapidon määritelmä

Kunnossapito on erittäin laaja käsite. PSK 6201 standardi (11) määrittelee sen seuraavasti:

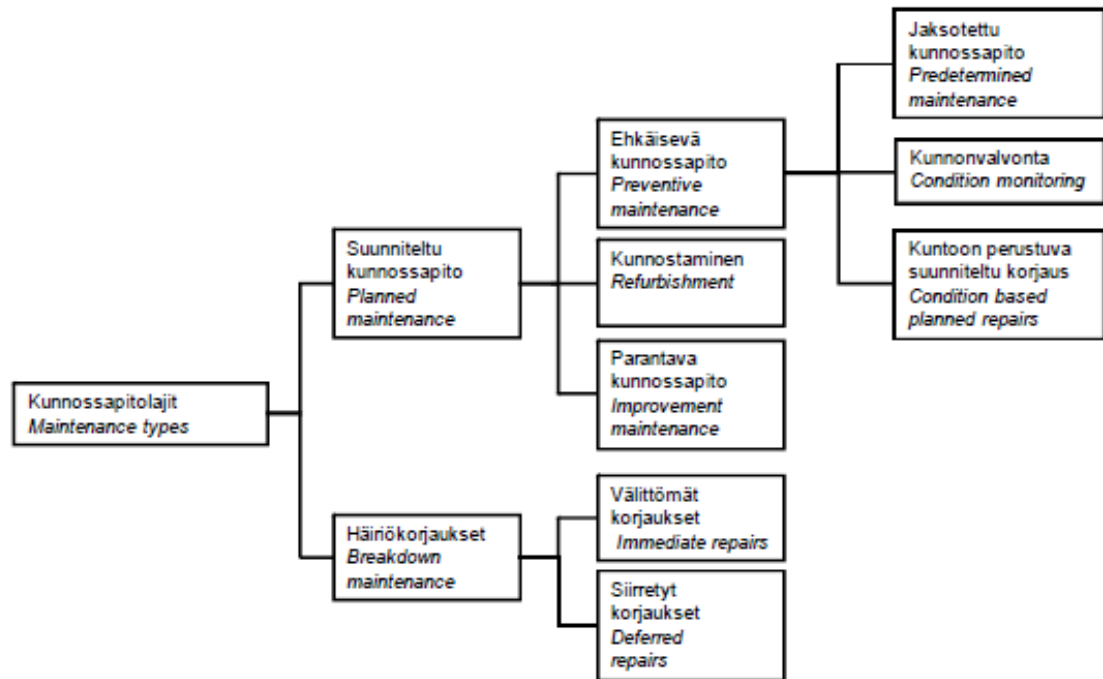
Kunnossapito on kaikkien niiden teknisten, hallinnollisten ja johtamiseen liittyvien toimenpiteiden kokonaisuus, joiden tarkoituksena on säilyttää kohde tilassa tai palauttaa se tilaan, jossa se pystyy suorittamaan vaaditun toiminnon sen koko elinjakson aikana.

Kunnossapidon tavoitteena on ylläpitää ja huoltaa tuotanto-omaisuutta. Tarkoituksena on ehkäistä suunnittelemattomat tuotantoseisakit. Niillä tarkoitetaan tilannetta, jossa tuotanto-omaisuus on pysähtynyt johtuen odottamattomasta viasta tai vauriosta.

### 2.2 Kunnossapitolajit

Kuvassa 1 on standardissa PSK 7501 (1) esitetty kaavio eri kunnossapitolajien suhteista toisiinsa. Sen mukaan kunnossapito jaetaan kahteen osaan, eli suunniteltuun kunnossapitoon ja häiriökorjaukseen. (4, s. 96.)





Kuva 1. Kunnossapitolajit (1)

### 2.2.1 Suunniteltu kunnossapito

Suunniteltu kunnossapito on tuotanto-omaisuuden ylläpitämistä suunnitellusti joko suorittamalla sille ennakkohuoltoja tai, kunnostamalla tai parantamalla sitä. Suunnitellun kunnossapidon edut ja hyödyt tulevat kontrolloiduista seisakiajoista ja häiriökorjausten vähentämisestä. Tuotantoseisakkeja saadaan kontrolloitua, kun viat korjataan ennen kuin ne pysäyttävät linjan tai aiheuttavat vaurioita, joita pitää häiriökorjata.

Suunniteltu kunnossapito pitää sisällään käsitteet ehkäisevä kunnossapito, kunnostaminen ja parantava kunnossapito. Ehkäisevän kunnossapidon voi jakaa jaksotettuun kunnossapitoon, kunnonvalvontaan ja kuntoon perustuvaan kunnossapitoon. Jaksotetulla tarkoitetaan tapaa, missä huollot tehdään esim. kalenteriin, käyttötunteihin tai tuotantomääriin perustuen. Kunnonvalvonnalla pyritään määrittelemään laitteen nykyinen toimintakunto ja löytämään mahdolliset piiloviat. Kuntoon perustuvassa kunnossapidossa huollot suoritetaan kunnonvalvonnasta saatujen tulosten perusteella. Parantavassa kunnossapidossa pyritään parantamaan laitteen luotettavuutta ja kunnossapidettävyyttä, ilman että muutetaan kohteen toimintaa. Tämä voidaan toteuttaa vaihtamalla vanhentuneita

osia uudempiin, kuten tasavirtamoottoreita oikosulkumoottoreiksi tai vahvistamalla koneen rakenteita. (11.)

### 2.2.2 Häiriökorjaukset

Häiriökorjauksessa vikaantunut laite palautetaan takaisin toimintakuntoon. Siinä tarvitaan paljon asentajia, jotta saadaan reagointiajat lyhyiksi. Yksi asentaja pystyy korjaamaan yhtä laitetta kerralla. Häiriökorjauksessa ei myöskään sidota pääomaa kalliisiin kunnonvalvontalaitteisiin.

Häiriökorjausta on kahden tyyppistä: välitöntä ja siirrettyä häiriökorjausta. Välitön häiriökorjaus on nimensä mukaisesti välittömästi tehtävää korjausta. Tuotanto pysäytetään korjauksen ajaksi, mistä syntyy suunnittelemattomia seisakkeja ja tappioita mahdollisista tuotehävikeistä ja työntekijöiden työtunneista. Siirretyllä häiriökorjauksella tarkoitetaan tilannetta, jossa todetaan laitteessa olevan vikaa mutta tämä vika ei vielä pysäytä laitetta taikka prosessia, vaan vika voidaan korjata esimerkiksi meneillään olevan tuotanto-ajon jälkeen taikka seuraavassa vuorossa, kun on enemmän asentajia tai varaosia saatavilla. (11.)

### 2.2.3 Kunnonvalvonta

Kunnonvalvonnan tavoitteena on löytää mahdolliset viat ja piiloviat, jotka voisivat joko pysäyttää laitteen toiminnan tai jotka estävät laitteen optimaalista toimintaa. Kunnonvalvonta vaatii ylimääräisiä resursseja työtunteina ja pääomana mittausvälineisiin, mutta näillä uhrauksilla on mahdollisuus saada aikaan huomattavia säästöjä tuotantolaitoksen eri kulumuodoissa. Toimivalla kunnonvalvonnalla voidaan vähentää muun muassa turhia koneiden avaamisia ja pienentää varaosavarastoa. (3.)

Kunnonvalvonnan avulla voidaan suunnitellut seisokkiajat pienentää, koska on jo havaittu, mitä vikoja lähdetään korjaamaan. Tällöin voidaan vikoihin liittyvät esitoimenpiteet tehdä tuotantoajalla, esimerkiksi tilata varaosat, valmistella työkalut, tehdä työohjeet ja tarvittaessa rakentaa telineet taikka valmistella lisäsuojausta tulitöitä varten. Kunnossapitoaikaa voidaan myös lyhentää, koska vika on jo havaittu ennen kuin se on pysäyttänyt tuotannon tai päässyt kehittymään vaurioksi. Kunnonvalvonnan avulla voidaan siis

kasvattaa tuotantoaikaa, koska mahdolliset viat eivät niin usein kehity vaurioiksi. Kunnonvalvonta on myös äärimmäisen hyvä keino minimoida kunnossapidon turhia kustannuksia, joita saattaa syntyä jaksotetusta kunnossapidosta. On turhaa korjata konetta kalenterin mukaan, jos siinä ei ole vikaa.

#### Aisteihin perustuva kunnonvalvonta

Kunnonvalvontaa voidaan suorittaa esimerkiksi aistinvaraisesti, jolloin tutkitaan, tuleeko laitteesta ylimääräisiä ääniä, valuuko nesteitä, täriseekö se tai ovatko lämpötilat normaalia poikkeavia. Laakereita voidaan kuunnella esimerkiksi ruuvimeisselin välityksellä tai stetoskoopilla. Suurin osa tuotantolaitteista on aistinvaraisen kunnonvalvonnan piirissä sen edullisuuden vuoksi. Aistinvarainen kunnonvalvonta on toimiva tapa, kunhan toiminta on suunnitelmallista ja toistuvaa.

Aistinvaraisen kunnonvalvonnan etuina on, että se ei vaadi kalliita laitteita. Valvontakierroksia voi tehdä nopeasti päivärutiinin omaisena, ja näiden kierroksien tuloksien perusteella on helppo lisätä mittaavaa kunnonvalvontaa tai suunniteltuja korjauksia tarpeen vaatiessa.

Aistinvaraisen valvonnan suorittajan tulee ymmärtää koneiden vikaantumislmiöt. Suorittajan tulisi osata yhdistää havainnot ja oireet, tehdä kunnonvalvontaa systemaattisesti ja tehdä raportoinnit havainnoista. Aistinvaraisessa kunnonvalvonnassa täytyy ottaa myös huomioon, että eri suorittajien aistien herkkyys ja vireystila vaikuttavat tuloksiin, kuten myös ympäristö. (4, s. 418, s. 421.)

Aistinvaraisessa valvonnassa hyödynnetään yleensä näköaistia, kuuloaistia ja tuntoaistia. Jokaiselle aistille on omat apuvälineensä helpottamaan päätöksen tekoa. Taulukossa 1 on listattuna näitä apuvälineitä.

Taulukko 1. Kunnonvalvonnassa käytettäviä aistien apuvälineitä (4. s. 422)

<b>Aisti</b>	<b>Apuväline</b>
Näköaisti	Valo Vesi, öljy tai muu neste Suurennuslasi Jauhe Valkoinen paperi Stroboskooppi Peili Endoskooppi Kamerat ja suurnopeuskamerat
Kuuloaisti	Keppi Stetoskooppi Akustinen koetin Ultraäänikoetin Nauhuri
Tuntoaisti	Lämpötarra Lämpöliitu Vesi Laatta, ”kolikko”

#### Mittaamiseen perustuva kunnonvalvonta

Kunnonvalvontaa voidaan toteuttaa myös mittaamalla. Mittaaminen on tarkempaa kuin aistinvarainen kunnonvalvonta. Mittaustulokset eivät ole riippuvaisia mittaajasta ja tämän aistien herkkyydestä, joten samalla laitteella saadaan samat tulokset eri mittaajien toteuttamina. Mittaustulokset ovat myös verrattavissa edellisiin tuloksiin.

Mittaamiseen perustuva kunnonvalvonta vaatii suunnittelua. Mittauskierrokset on oltava tarkoin harkittuja, koska mittaaminen vaatii resursseja ja niille halutaan vastinetta. Mittaamalla saadaan tarkkaa tietoa laitteen kunnosta, joten mittaustuloksia voidaan käyttää työn suunnittelun päätöksenteossa.

Yleisimmät mitattavat asiat ovat laitteiden värähtelyt ja lämpötilat sekä voiteluaineiden koostumus.

Värähtelymittauksella tarkoitetaan menetelmää, jossa moottoreista, vaihteistoista tai pumpuista mitataan tietyistä mittapisteistä värähtäntäsuureita. Mittausvälineet voivat olla kiinteästi asennettuja värähtelyantureita (kuva 2) tai kannettavia mittalaitteita (kuva 3). Käytettäviä suureita ovat siirtymä, nopeus, kiihtyvyys, taajuus, pyörimisnopeus ja jakso, näin ollen soveltuvia anturityyppejä ovat siirtymä-, nopeus-, kiihtyvyys- ja iskusysäysanturi. (10.)

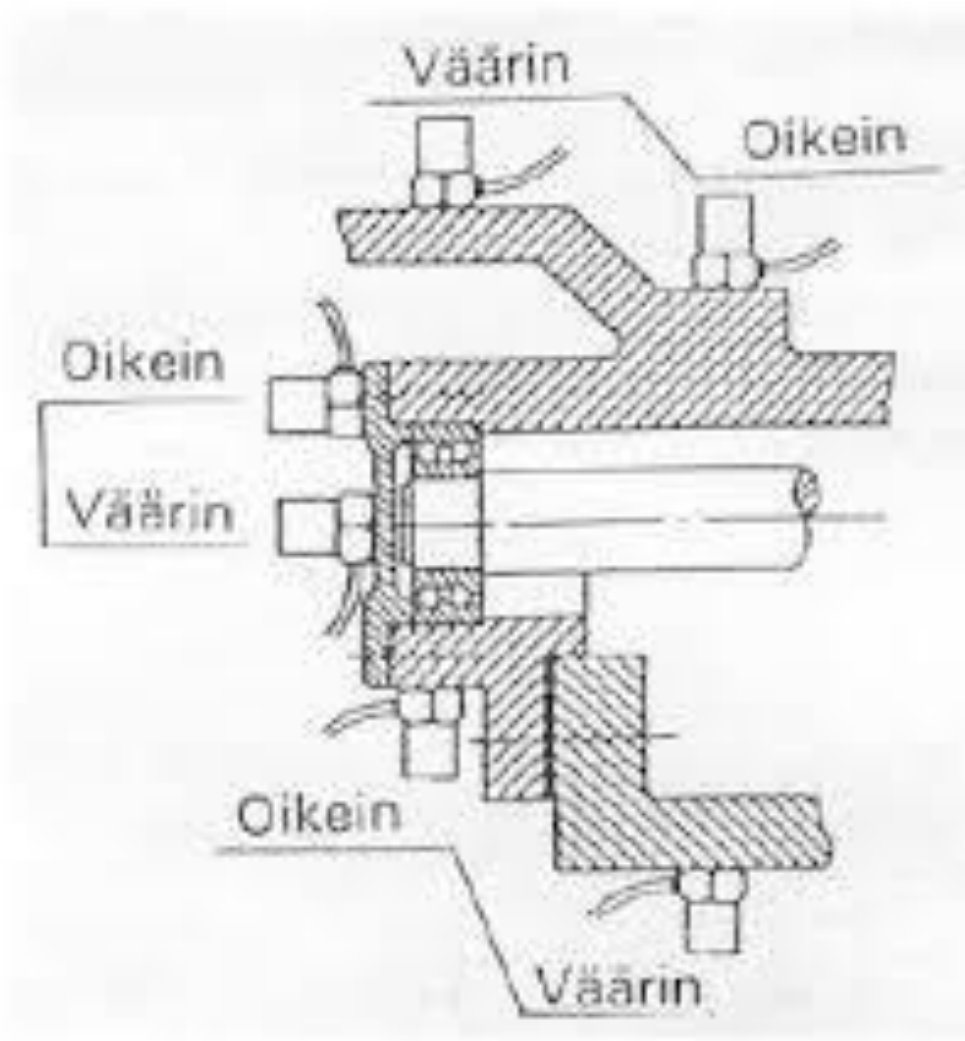


Kuva 2. IFM VTV122-tärinälähetin



Kuva 3. Emerson CSI 2140 kannettava värähtelyanalysaattori

Mittauspisteen valinta on tärkeä osa mittausta, mittauspisteen on oltava mahdollisimman lähellä värähtelyn lähdettä. Korkeataajuinen värähtelyn voimakkuus pienenee sen myötä mitä kauempaa sitä mitataan, tästä syystä on mittauspisteen oltava lähellä lähdettä. Mittaus suoritetaan yleensä säteissuunnassa, mutta on mahdollista mitata myös aksiaalisuunnassa. (Kuva 4.)



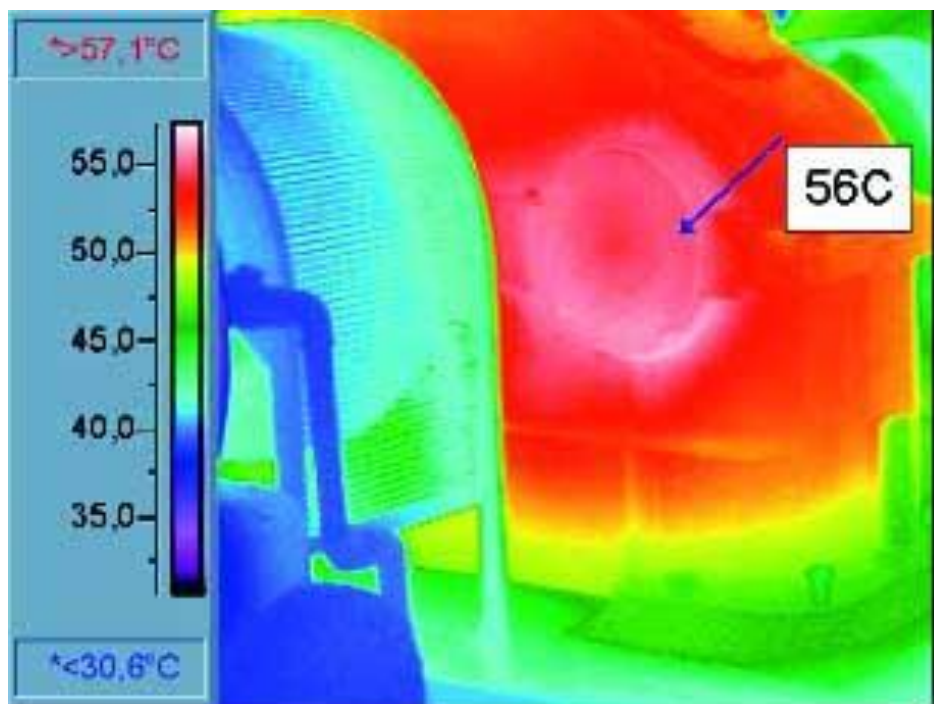
Kuva 4. Värähtelymittauksen mittauspisteen valinta, (10)

Lämpötilan mittaus on myös yleinen kunnonvalvonta tapa. Varsinkin moottorien tai pumppujen kohonneet lämpötilat voivat paljastaa laakerivian, linjausvirheet tai yleisen kulumisen. Moottori lämpenee, jos se joutuu käyttämään normaalia enemmän virtaa. Näin tapahtuu, kun laitteet vastustavat esimerkiksi laakerivian takia. Lämpötila on myös hyvä hälytys- tai suojamenetelmä, esimerkiksi moottoreissa ylikuumentumissuoja on yleinen tapa suojata moottoria rikkoontumiselta. (4, s. 439.)

Yksi tapa hyödyntää lämpötilan mittaamista kunnonvalvonnassa on käyttää lämpökameraa. Kaikkialle ei ole järkevää asentaa lämpöantureita ja tehdä logiikkaa niiden ympärille, koska lämpökameralla voidaan paljastaa esimerkiksi laakerivikoja tai hydraulisylinterin

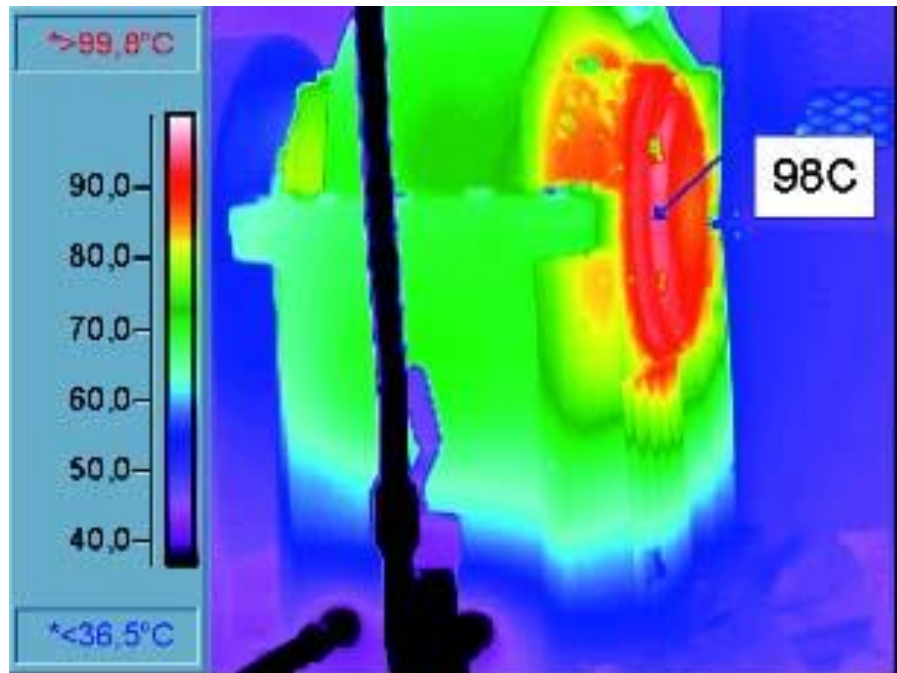
vuotoja. Lämpökameralla tehtävät kierrokset vaativat vähemmän resursseja kuin antureiden ja logiikoiden asentaminen. (6.)

Kuvissa 5 ja 6 näkyy sama laakeri ehjänä ja vaurioituneena. Kuvat toimivat helppoina kunnonvalvontakeinoina. Kuvien tulkitseminen vaatii ammattitaitoa, koska molemmissa kuvissa laakerit ovat punaisella, mutta skaalaa lukemalla voi nähdä, että niissä on 40 °C:n ero. Lämpökuvien analysointi ei ole ihan yksiselitteistä, kuvassa voi olla useampi komponentti ja skaala voi näyttää, että yksi niistä on lämpimämpi kuin muut ja värjätä sen punaiseksi. Tämä ei automaattisesti tarkoita, että komponentissa on vikaa, se on vain lämpimämpi kuin muut. Pitää siis tietää, milloin se on liian kuuma ja kriittisellä rajalla. Tämän vuoksi henkilöllä, joka suorittaa kuvien analysoimisen on hyvä olla ymmärrystä termodynamiikasta, fysiikasta ja mekaniikasta. (7.)



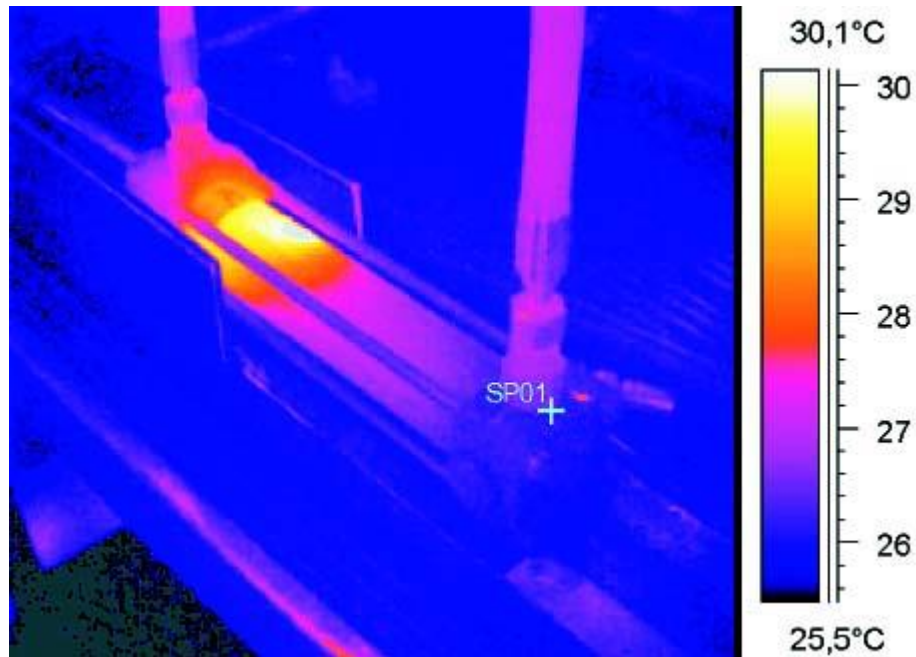
Kuva 5. Toimiva laakeri (6)





Kuva 6. Vaurioitunut laakeri (6)

Kuvassa 7 on hydraulisylinteri. Sylinterissä ei ulkoisesti näy mitään vikaa, mutta sisällä oleva männän vuoto paljastuu lämpökuvaamalla. Hydrauliöljy pääsee virtaamaan männän läpi ja lämpenee tämän vuoksi. Vikaa olisi erittäin vaikea havaita ilman lämpökuvausta.



Kuva 7. Vaurioitunut hydraulisyylinteri (6)

Sähkökeskuksia on hyvä lämpökuvata ja seurata komponenttien lämpötrendejä. Ylikuumeneminen voi heikentää kaapeleiden eristeitä. Heikentyneen kaapelieristeen läpi voi päästä valokaari, joka saattaa vaurioittaa koko laitteen tai aiheuttaa jopa tulipalon. (7.)

Voiteluaineanalyysit ovat myös tärkeä osa kunnonvalvontaa. Voiteluainetta voidaan pitää yhtenä koneen osana. Voiteluaineanalyysillä saadaan paljon tietoa laitteen kunnosta, itse voiteluaineesta taikka prosessista. Voiteluaineesta voidaan mitata epäpuhtauksien määrä ja kokojakauma, metallipitoisuudet ja voiteluominaisuudet. Laakerien ja hammasrattaiden kuluessa niistä irtoaa metallihiukkasia ja nämä metallihiukkaset päätyvät voiteluaineen sekaan. Suunnitelman mukainen voiteluaineen kunnonvalvonta voi paljastaa esim. laakereiden normaalia nopeamman kulumisen, voiteluaineeseen päässeeseen veteen tai voiteluaineen sopimattomuuden käyttökohteeseen. (4, s. 428.)

Voiteluaineanalyysit jaotellaan analyysityypeittäin (4, s. 429):

- Perusanalyysit, jotka keskittyvät itse voiteluaineen kuntoon.
- Hiukkasanalyysi, jolla seurataan voiteluaineeseen päässeitä epäpuhtauksia, niiden määrää ja kokoa.

- Kulumametallianalyysit, jolla seurataan voiteluaineessa olevan metallin määrää ja sen mahdollista lisääntymistä.
- Vesipitoisuusanalyysit.

Laitteen kunnonvalvonta kulumametallianalyysillä perustuu trendiseurantaan. Analyysillä seurataan metallihiukkasten lisääntymisen nopeutta ja/tai kokojakaamaa. Kohonneet metallihiukkasmäärät voivat olla seuraus esimerkiksi laakeriviasta tai hammasrat-  
taiden kulumisesta. (4, s. 435.)

## 2.3 Vika ja vikaantuminen

Vika on tila, jossa laite ei toimi vaaditulla tavalla. Vaaditulla tavalla tarkoitetaan tilannetta, jossa joko koko toiminta puuttuu tai se ei ole tarpeeksi tehokasta tai turvallista käyttöä. (5, s. 71.)

Esimerkki vikaantuneesta laitteesta joka silti toimii:

Valmistusprosessi tarvitsee kemikaalia 300 l/min. Kemikaali pumpataan säiliöstä teholla 350l/min. Vaadittu toiminto on syöttää prosessiin 300 l/min. Pumpun todellinen pumppaus teho on siis jonkin verran suurempi, joten pumpun teho saa laskea huomattavasti (350->300 l/min), ennen kuin vaaditun toiminnon suorittaminen vaarantuu. Kun pumppausteho on alle 300 l/min, pumppu on vikatilassa, vaikka se muuten toimii. (5, s. 71.)

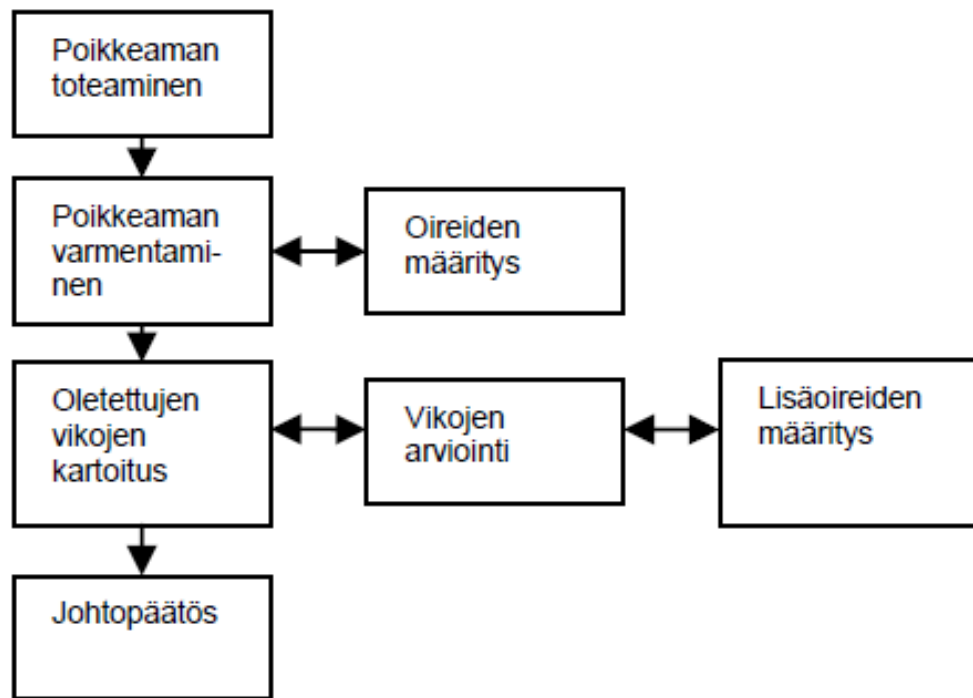
Laitteen vikaantumisesta seuraa vika, joka voi olla joko häiriö taikka vaurio. Häiriössä laite ei ole rikki, mutta vaatii korjausta ja aiheuttaa tuotannon menetyksiä. Häiriö voidaan korjata joko puhdistamalla, säätämällä tai yksinkertaisimmillaan uudelleen käynnistämällä. Laitteen tai komponentin vikaantumisväli voidaan määrittää häiriöiden perusteella.

Vauriossa laite on rikki. Sen seuraukset ovat samat kuin häiriössä. Vaurioitunut laite tai sen osa vaatii perusteellisempaa korjausta. Laitteen tai komponentin vikavälin lisäksi sen elinikä voidaan määrittää vaurioiden perusteella. (5, s. 71.)

Vian oireiden selvittäminen on usein vaikeaa. Kunnossapitostrategia voi olla painottunut korjausten tekemiseen, tällöin vikojen oireiden selvittämiseen ei ole uhrattu aikaa. Tarkastaminen voi olla yleisluontoista. Tähän voi vaikuttaa esimerkiksi tarkastuspisteiden vaikea luokse päästävyys tai likainen ympäristö. Lian alla saattaa piillä alkava vuoto tai särö, joka jää helposti huomaamatta. (5, s. 85.)

Usein vian oireita ei huomata tai niitä pidetään normaalina vanhenemiseen liittyvinä oireina. Oire saattaa myös olla niin pieni, ettei siitä raportoida tai sitä ei pidetä vakavana. Laitteen tuotantonopeuden heikkeneminen on vaikea havaita, mutta omalla tavallaan se on vika. (5, s. 86.)

Vianmäärityksen laukaisee poikkeama mittaustuloksissa. Varsinainen diagnoosi viasta on usean vaiheen lopputulos, suljetaan pois epätodennäköisiä vaihtoehtoja ja varmistetaan todennäköisiä syitä. Kuvassa 8 on havainnollistettu vianmäärityksen kulku. PSK 5707 -standardi suosittelee käyttämään vianmäärityksen raportoinnissa PSK 5705 -standardin mukaista johtopäätösraporttia. (10.)



Kuva 8. Vianmäärityksen kulku (15)

### 3 Prosessidatan hyödyntäminen yleisesti

Mittaavaa kunnonvalvontaa voi tehdä myös ilman erikseen siihen tarkoitettujen anturien asentamista ja käyttämistä. Prosessista itsestään on mahdollista löytää hyödyllistä dataa suoraan automaatiojärjestelmästä, logiikoilta tai taajuusmuuttajista.

Tärkeintä on valita sopiva data, jota hyödyntää. Usein tarvitaan useamman kohteen tiedot, joita verrata keskenään. Näiden syy-yhteyksien löytäminen ja hyödyntäminen on teollisen internetin luoma mahdollisuus.

Esimerkkinä prosessidatan hyödyntämisestä toimii nestesäiliö, josta pumpataan pumpulla toiseen säiliöön nestettä. Jos molemmat säiliöt on varustettu joko vaa'alla tai pinnan korkeuden mittaamisella, voidaan pumpun tuottoa mitata. Seuraamalla pumpun moottoria ohjaavan taajuusmuuttajan virran syöttöä ja pumpun tuottoa, voidaan saada hyvä käsitys pumpun kunnosta. Jos tuotto pysyy samana ja pumpattava tuote on sama, mutta virran tarve kasvaa, pumpussa tai sitä ohjaavassa moottorissa on jotain vikaa.

Siemensin kehittämä SIMATIC PCS7 -automaatiojärjestelmä pitää sisällään omia kunnonvalvontaohjelmia. Niiden hyödyntäminen vaatii tietyt lähtötiedot, jotka nykyaikaisesta prosessista yleisesti löytyvät.

Siemens on kehittänyt myös pilvipalvelun, jossa voi tehdä datalle analytiikkaa tai tehdä visuaalisia kuvaajia sen pohjalta. Tätä pilvipalvelua kutsutaan nimellä MindSphere. MindSpheren ideana on, että yksittäisiltä logiikoilta voidaan lähettää muuttujia pilveen ja tehdä niiden perusteella syy-yhteyksiä, joista voi piirtää trendejä tai laskea milloin mahdollisia vikoja ilmaantuu.

#### 3.1 Siemens SIMATIC PCS7 -automaatiojärjestelmä

PCS7 sisältää oman Condition Monitoring Libraryn (CML), jonka avulla voidaan seurata mekaanisten laitteiden kuten pumppujen ja venttiilien toimintakuntoa ja analysoida niiden toimintaa. CML sisältää visuaalisia näkymiä, joiden avulla voidaan helposti yhdellä silmäyksellä nähdä poikkeamia. CML:n avulla on myös mahdollista luoda hälytyksiä joko

järjestelmään itseensä tai automaattisina hälytyksinä push-viesteinä tai sähköpostin kautta operaattoreille tai kunnossapidon suunnittelijalle. (12.)

CML on kirjasto, joka on luotu PCS7:n sisään. Se on ilmainen lisäosa, joka on ladattavissa Siemensin kautta. CML on yhteensopiva yhden version taaksepäin (PCS7 on nyt versiossa 9, joten versio 8 on vielä tuettu). Kirjasto pitää sisällään viisi blockia: PumpMon, VlvMon, PrDrpMon, SteadyState ja PST. Jokaisella blockilla on oma käyttökohteensa, jossa valvotaan toimilaitteiden toimintaa (13):

- PumpMonilla seurataan keskipakopumppuja.
- VlvMonilla seurataan takaisin kytkettyjä ohjausventtiilejä.
- PrDrpMonilla seurataan painehäviöitä tai paineen muutoksia esim suodattimissa.
- SteadyStatella valvotaan signaaleja, eli onko signaali vakaa vai heitteleekö se.
- PST:llä voi testata venttiilien toimintaa ajon aikana.

### 3.1.1 PumpMon Block

PumpMon Block on lyhenne termistä Pump Monitoring Block. Se luotiin keskipakopumppujen yksinkertaiseksi ja kustannustehokkaaksi kunnonvalvontakeinoksi. Sillä on mahdollista valvoa myös muita pumpputyyppejä keskipakopumppujen lisäksi. Blockia käytetään havaitsemaan pumpussa olevia mahdollisia vikoja, jotka paljastuvat normaalin käytön yhteydessä, sekä pumpun toiminnan optimoimiseen sen käytön reaaliaikaisen analysoinnin avulla.

PumpMon Block on puhtaasti diagnostiikkatyökalu. Sillä ei voi vaikuttaa suoraan prosessin toimintaan.

Työkalu tarvitsee seuraavat signaalit toimiakseen: virtaus, paine ennen ja jälkeen pumpun, sähköteho, pumpattavan tuotteen lämpötila ja moottorin pyörimisnopeus.

PumpMon Block sisältää seuraavat näkymät:

- pumpun nostokorkeuskuvaaja: nostokorkeuden ja tilavuusvirran välinen funktio
- teho- ja hyötysuhdekuvaaja: pumpun teho ja hyötysuhde tilavuusvirtaan suhteutettuna
- NPSH-käyrä: vähimmäispaine kavitaation estämiseksi.

Nostokorkeuskuvaajassa (kuva 9) Y-akselilla näemme pumpun tuottaman nosteen ja X-akselilla tilavuusvirran. Sininen käyrä osoittaa pumpun optimaalisen toiminta-alueen ja sininen pallo pumpun tämänhetkisen tuoton. Kulunut pumpu ei pysty tuottamaan optimaalista painetta ja taajuusmuuttaja joutuu nostamaan pumpun pyörimisnopeutta, jotta tavoitepaine saadaan saavutettua, eli pumpu ottaa enemmän virtaa tuottaakseen tarvittavan paineen.

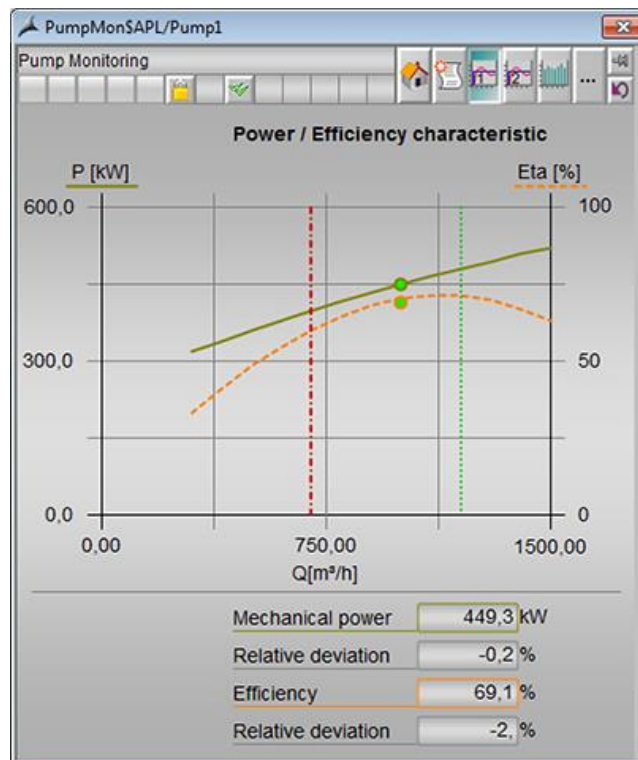


Kuva 9. SIMATIC PCS7 PumpMon Blockin perusnäkö

Teho ja hyötysuhteenäkymässä (kuva10) Y-akselilla vasemmalla näemme tehon, oikealla hyötysuhteen ja X-akselilla tilavuusvirran. Punainen viiva osoittaa optimaalisen

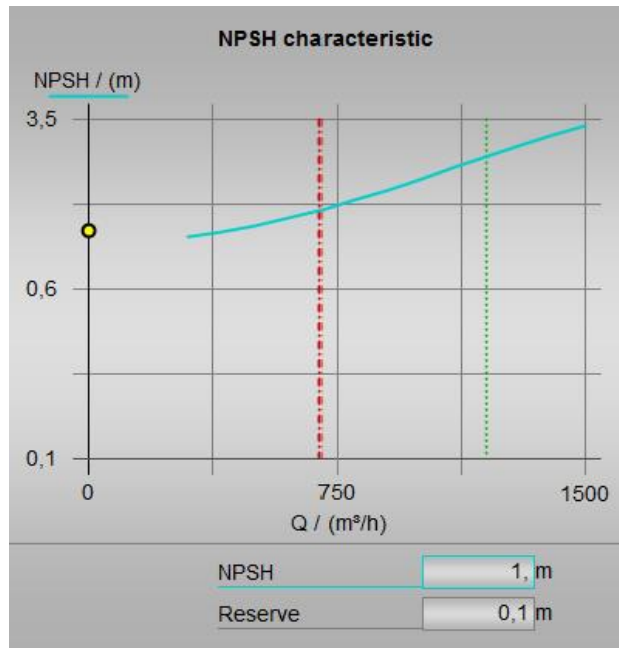


toimintakäyrän ja katkoviiva todellisen tilanteen, ja kun todellinen tuotto eriiä optimaalisesta käyrästä yli toleranssin, syntyy automaattinen hälytys.



Kuva 10. Tehokkuus- ja voimanäkymä

NPSH-kuvaajassa (kuva 11) näkyy Y-akselilla minimipaine ja X-akselilla tilavuusvirta. Sinisellä näkyy turvallinen raja, jossa pumppu ei ala kavitoimaan. Vihreä pallo kertoo, että ollaan turvallisella alueella, ja keltainen ilmoittaa riskialueesta.



Kuva 11. NPSH-kuvaaja

### 3.1.2 VlvMon Block

VlvMon Block on lyhenne termistä Valve Monitoring Block. Sillä pystytään seuraamaan venttiilin käyttökertoja, käyttöaikaa, virtauksen ominaiskäyrää ja liikeaikoja.

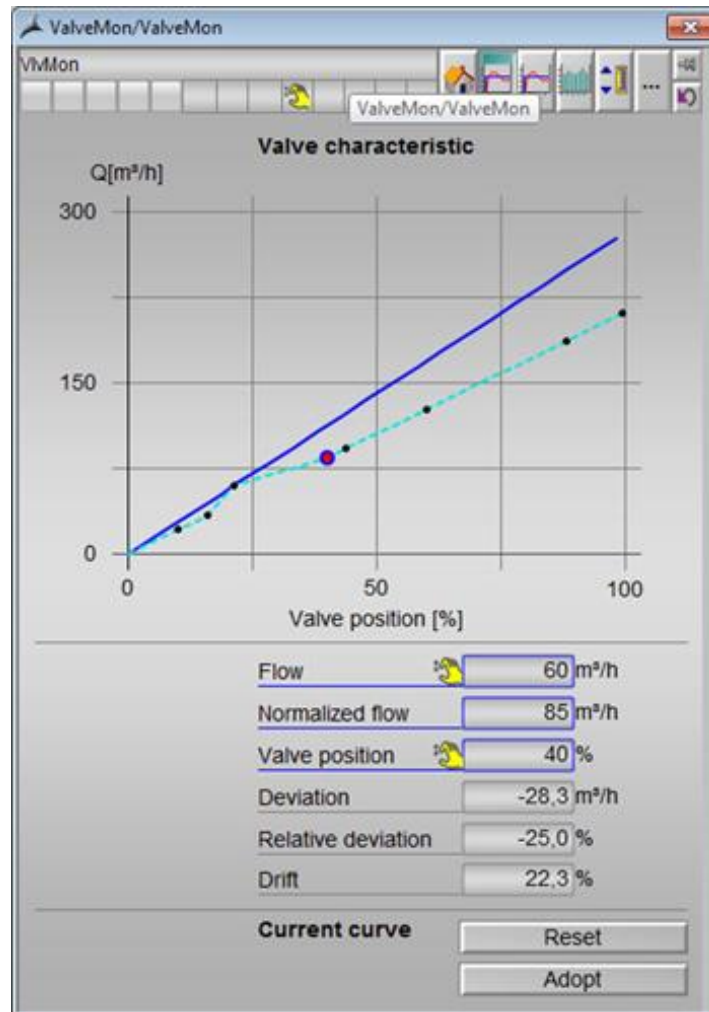
Venttiilin käyttökertoja ja käyttöaikaa voidaan käyttää huoltovälin ajoittamiseen. Virtauksen ominaiskäyrän ja liikeaikojen avulla voidaan paljastaa piilovikoja tai vikoja.

Perusnäkymässä (kuva 12) nähdään täysien auki-kiinni -liikkeiden lukumäärä, suunnan vaihdot auki-kiinni tai kiinni-auki ja käyttöajat.

	Mode	<b>On</b>	...
	<b>Operational data</b>		
	No. of full strokes	5	
	Wear reserve	-25 %	
			Reset
	Direction changes	8	
	Wear reserve	-14 %	
			Reset
	<b>Operating time</b>		
	Continuous move.	0 min	
	No movement	86 min	

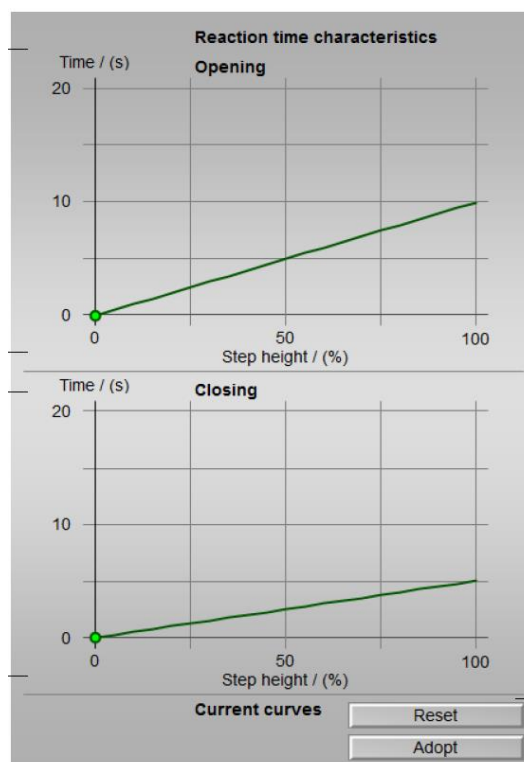
Kuva 12. Venttiilin perusnäky

Venttiilin virtauksen ominaiskäyränäkymän (kuva 13) avulla voidaan seurata takaisinkytketyn venttiilin toimintakuntoa. Y-akselilla on tilavuusvirta ja X-akselilla on venttiilin auki-asento prosentteina. Sininen käyrä taas osoittaa venttiilin optimaalista toiminta-alueita ja sininen pallo kuvaa venttiilin nykyistä asentoa. Kulunut venttiili päästää enemmän virtausta läpi ja likaantuneen venttiilin pitää olla enemmän auki päästääkseen saman määrän läpi.



Kuva 13. Venttiilin virtauksen ominaiskäyränäkymä

Venttiilin reaktioaikaominaiskäyränäkymässä (kuva 14) voidaan seurata aikaa joka venttiilillä kestää päästä asetusarvoon. Poikkeamat voivat johtua toimilaitteen vioista, venttiilin vioista tai virtaavan tuotteen muutoksista.



Kuva 14. Venttiilin toiminta-ajat aukeamis- ja sulkeutumistilanteissa

### 3.2 MindSphere

Siemens MindSphere tarjoaa keinon viedä anturidatan logiikalta pilveen, jossa on mahdollista suorittaa analytiikkaa datalle. MindSphere tarjoaa myös keinoja visualisoida dataa ja analytiikkaa esimerkiksi luomalla trendejä moottorin virrantarpeesta tai tuotant nopeudesta. Siinä on mahdollista luoda hälytysrajoja datan perusteella. MindSpheressä on mahdollista luoda tai ostaa erilaisia sovelluksia MindSpheren omasta sovelluskau-pasta.

Logiikan kytkeminen MindSphereen vaatii gateway-laitteen tai OPC:n yhteyden muodos-tamiseen. Siemens tarjoaa kaksi erilaista gateway-laitetta, Siemens IoT2040 (kuva 15) ja Siemens MindConnect Nano (kuva 16). Laitteeseen voidaan kytkeä kahdeksan logiik-kaa. Laite lähettää mittauspisteiltä tulevan datan yksisuuntaisesti pilveen. Data on kryp-tattua, ja datan lähettäjä omistaa datan.



Kuva 15. Simatic IoT2040



Kuva 16. MindConnect Nano

Tästä esimerkkinä voisi käyttää mielikuvitus linjaston päämoottoria, joka on oikosulkumoottori. Logiikalta voidaan viedä MindSphereen pyörimisnopeus, tehontarve, virrantarve ja vääntömomentti. Suureista tehdään trendit ja mittarit. Kunnossapidon suunnittelija pystyy seuraamaan reaaliajassa tilannetta ja hän voi myös asettaa hälytysrajat suureille. MindSphere ilmoittaa suunnittelijalle push-viestillä, että vääntömomentti on toistuvasti ylittänyt raja-arvot. Suunnittelija voi analysoida mistä piikit trendissä johtuu ja tarvittaessa lisätä esimerkiksi kunnonvalvontakierroksen alueelle ja kierrokselta saadun tiedon avulla löydetään mahdollinen vika ja sen perusteella voidaan suunnitella huolto seuraavalle tuotantokatkokseksi. (9.)

## 4 Prosessidatan hyödyntäminen Universal-linjalla

### 4.1 Universal-linjan toiminta ja osat

#### 4.1.1 Toiminta

Universal on 80-luvulla Vaaralaan tuotu suklaan valulinja. Se on Aastedin valmistama. Valulinjan toimintaperiaate perustuu muottivaluun. Suklaa valetaan lämmitettyyn muottiin, muotti siirtyy jäähdytystunneliin, jäähdytyksen jälkeen muotista irrotetaan suklaa hihnakuljettimelle, josta suklaa siirretään tarjottimilla varastointikaseteille ja kasetit kuljetaan varastoon.

Valulinjan toiminnan voi jakaa seuraaviin toimintoihin:

- suklaan varastointi
- suklaan temperointi
- muotin sisään syöttö/ vaihto
- muotin lämmitys
- kuoren valu
- täytevalu
- pohjan valu
- jäähdytys
- uloslyönti
- suklaan asettelu varastointilevyille
- varastointilevyjen kasetointi
- kasettien kuljetus varastoon.

Suklaan valu tapahtuu valukoneessa, jossa tapahtuvat muotin lämmitys, valut, muotin jäähdytys ja uloslyönti.

Muotit on kiinnitetty muottiketjuun, joka kiertää valukoneen sisällä lenkkiä. Muottiketjussa on 1293 muottia kiinni. Jos valetaan samaa tuotetta, ei muotteja vaihdeta valun aikana, eli samalle muotille voi valaa useamman kerran saman laadun suklaata.

Linjalla ei valmisteta suklaata vaan, se tuodaan linjan varastosäiliöihin tehtaan suklaan valmistusosastolta. Varastosäiliöitä on 9 kpl, joissa säilytetään eri suklaalaatuja. Linjalla voidaan valaa erikokoisia suklaalevyjä, konvehteja tai patukoita. Jokaiselle tuotteelle on omanlaisensa muottisarja. Muotit ovat samankokoisia. Tuotteet voivat olla joko yhtä suklaalaatua, kuori ja pohja eri laatua tai kuori, täyte ja pohja kaikki eri laatuja.

Universalin ohjaus on toteutettu PCS7 versio 8.1:n avulla, logiikkoina toimii 2 kpl Siemens S7-400, joissa on Siemens ET200M-etäyksiköt. Toimintaa ohjataan ja valvotaan kolmelta PC-valvomolta, joissa on WinCC-ohjelmisto. Linjastolla on myös ohjauspaneelita.

Logiikat ja valvomot on yhdistetty toisiinsa Ethernet-väylän välityksellä. Ohjauspaneelit ja etäyksiköt on kytketty logiikan Profibus-väylään. CML:ää ei ole otettu käyttöön.



#### 4.1.2 Moottorit

Valukoneen kaikki moottorit ovat tasavirtamoottoreita, joten niitä ei ohjata taajuusmuuttajilla. Tämän takia kaikista moottoreista ei saada ulos kovin laajasti tietoa, verrattuna siihen, jos ne olisivat oikosulkumoottoreita. Logiikalle tulee tieto, pyöriikö moottori vai ei, sekä millä nopeudella moottori on asetettu pyörimään, mutta takaisinkytkentää ei ole.

Päämoottoria ohjataan ABB:n DCS550 vakiotasavirtakäytöllä (kuva 17) josta on mahdollista saada ulos seuraavat parametrit:

- moottorin nopeus (rpm)
- moottorin virrankäyttö (%)
- todellinen virta (A)
- moottorin lämpötila (°C)
- vääntö (%)
- vääntö (Nm).



Kuva 17. Päämoottorin vakiotasavirtakäyttö

#### 4.1.3 Jäähdytystunnelit

Jäähdytystunneleiden toiminta perustuu siihen, että pumppu pyörittää vesi-glykoli-yhdistelmää suljetussa kierrossa jäähdytyspatterin läpi ja puhallin puhaltaa ilmaa patterin läpi jäähdyttäen tunnelin.

Jäähdytyskiertoa ei ohjata pumpun nopeutta säätämällä vaan linjasäätöventtiiliä ohjaamalla. Linjasäätöventtiilissä on kiinni toimilaite, jota ohjataan logiikalta. Toimilaitteelta on takaisinkytkentä, siitä saadaan ulos venttiilin asennon oloarvo.

#### 4.1.4 Pumput

Suklaan pumppaukseen käytetään Rotan CHD51-hammasrataspumppuja ja FIP65 051-pyöröhammaspumppuja.

Pumppuja ohjataan Danfoss VLT2800-taajuusmuuttajilla, joista on saatavilla seuraavat parametrit:

- moottorin nimellinopeus
- käyttötunnit (h)
- virta
- taajuus
- vääntö
- teho.

Pumpuilla pumpataan säiliöistä suklaata valupadoille tai temperointikoneille. Linjoissa ei ole virtausmittareita, eikä vaakoja. Linjalla oleviin säiliöihin on suunniteltu pinnankorkeuden mittaamista.

Linjalta löytyy myös yksi uusi Rotan-pumppu, jolla pumpataan valkoista suklaata säiliöstä linjalle. Pumpun malli on HD81ECHD. Pumpun eteen on magneettisihti ja pumpun jälkeen on painemittari. Pumpattavasta säiliöstä löytyy pinnankorkeuden mittaus. Tätä pumpua ohjataan ABB ACS880-taajuusmuuttajalla (kuva 18), josta on saatavilla seuraavat parametrit:

- moottorin nimellisa nopeus
- virta
- momentti
- lähtöteho
- käyttötunnit



Kuva 18. ABB ACS880-taajuusmuuttaja

## 4.2 Prosessidatan hyödyntämismahdollisuudet Universal-linjalla

### 4.2.1 Valukoneen moottorit

Valukoneen kaikki moottorit ovat tasavirtamoottoreita, joita ohjataan suoraan joko päälle tai pois, ainoastaan päämoottorilla on takaisinkytkentä, jota voisi hyödyntää.

Päämoottorin tasavirtakäytöltä saatavia parametrejä voisi hyödyntää karkeana kunnonvalvontakeinona. Tämä vaatii, että jokaiselta tuotteelta tehdään omat trendit. Trendit eivät ole vertailukelpoisia, jos tuotantonopeus on eriävä, tai jos siinä on eri tuotantovaiheet (kuori-, täyte- ja pohjavalu). Saman tuotteen trendiä voi verrata keskenään, jos on huomioitu, että on sama tuotantonopeus.

Piirtämällä trendiä päämoottorin virrantarpeesta saadaan selkeää kuvaajaa, joka paljastaa laakerikulumia linjalla. Tämä trendi toimii apuna, kun tehdään päätöksiä lisäkunnonvalvontatarpeesta. Tarkkana kunnonvalvontakeinona sitä ei voi käyttää, koska päämoottori pyörittää koko linjaa valta-akselin avulla ja linjalla on niin monta laakeria ja paikkaa, jotka voivat lisätä virrantarvetta.

### 4.2.2 Valukoneen pumput

Linjalla on suklaan pumppaukseen käytetty kahta hammaspyöröpumppua, sekä Rotan HD51- ja Rotan HD81ECHD-hammasrataspumppuja. Linjalla olevissa päiväsäiliöissä ei ole pinnankorkeuden mittausta, ainoastaan täyttö- ja tilausrajat, jotka on toteutettu optisilla antureilla. Tämän vuoksi pumppuihin ei voi hyödyntää PCS7:n tarjoamaa CML-mahdollisuutta, koska tarvittavat signaalit puuttuvat (virtaus ja paine ennen ja jälkeen pumppun).

Säiliöiden täyttymisestä ei voida piirtää myöskään aikatrendiä, koska säiliöitä ei pumpata täyteen. Käyttäjät pumppaavat säiliöihin tarpeen mukaan tuotetta, ja säiliö täytetään harvoin tilausrajasta täysirajaan.

Säiliöihin ollaan lisäämässä pinnankorkeuden mittausta, tällöin voidaan hyödyntää ”pumpun tuottomoottorin virrankäyttö” syy-yhteyttä. Tästä voisi piirtää trendiä MindSpheren välityksellä.

#### 4.2.3 Suklaaputkiston venttiilit

Linjalla on paljon putkistoa, jossa kuljetetaan suklaata. Putkiston toimintaa ohjataan etänä ja venttiileihin on liitetty paineilmakäyttöiset toimilaitteet. Venttiileiltä ei kuitenkaan ole takaisinkytkentää, jotta voisi hyödyntää PCS7:n tarjoamaa Valve Monitoring Blockia.

#### 4.2.4 Jäähdytystunneli

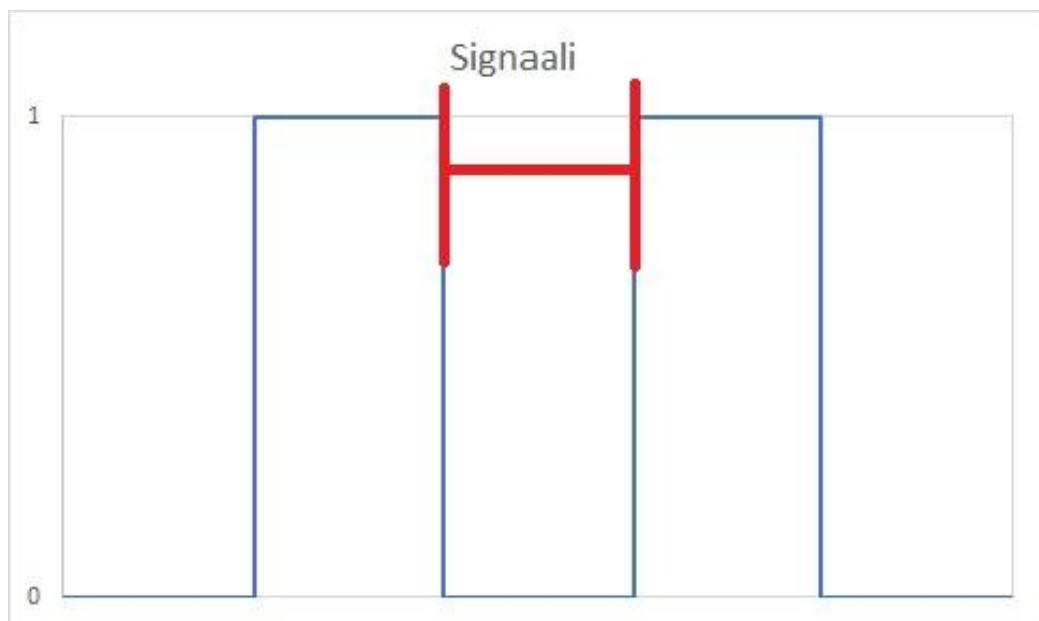
Jäähdytystunnelin vesi-glykolikierron kuntoa olisi mahdollista valvoa, koska pumppua ohjataan taajuusmuuttajalla ja ohjausventtiililtä on takaisinkytkentä. Syy-yhteytenä voisi käyttää: moottorin virran tarve, ohjausventtiilin asennon ja jäähdytystunnelin lämpötila. Näistä tiedoista piirrettyä trendiä voisi käyttää karkeana kunnonvalvontakeinona, jota voisi hyödyntää huoltosuunnittelussa ja lisäkunnonvalvontakierrosten suunnittelussa.

#### 4.2.5 Valukoneen muottiketju

Universal-linjan muottiketjun venymää ei ole mitattu, ketjuun kohdistuu arvaukseen perustuva ennakkohuoltosuunnitelma (14).

Muotit ovat suunnilleen samankokoisia ja ne ovat kiinteästi kiinni ketjussa. Muotteja on 1 293 kappaletta kerrallaan koneessa. Koneita ajetaan vakionopeudella, joka vaihtelee tuotteittain, mutta nopeus on käyttäjän asettama.

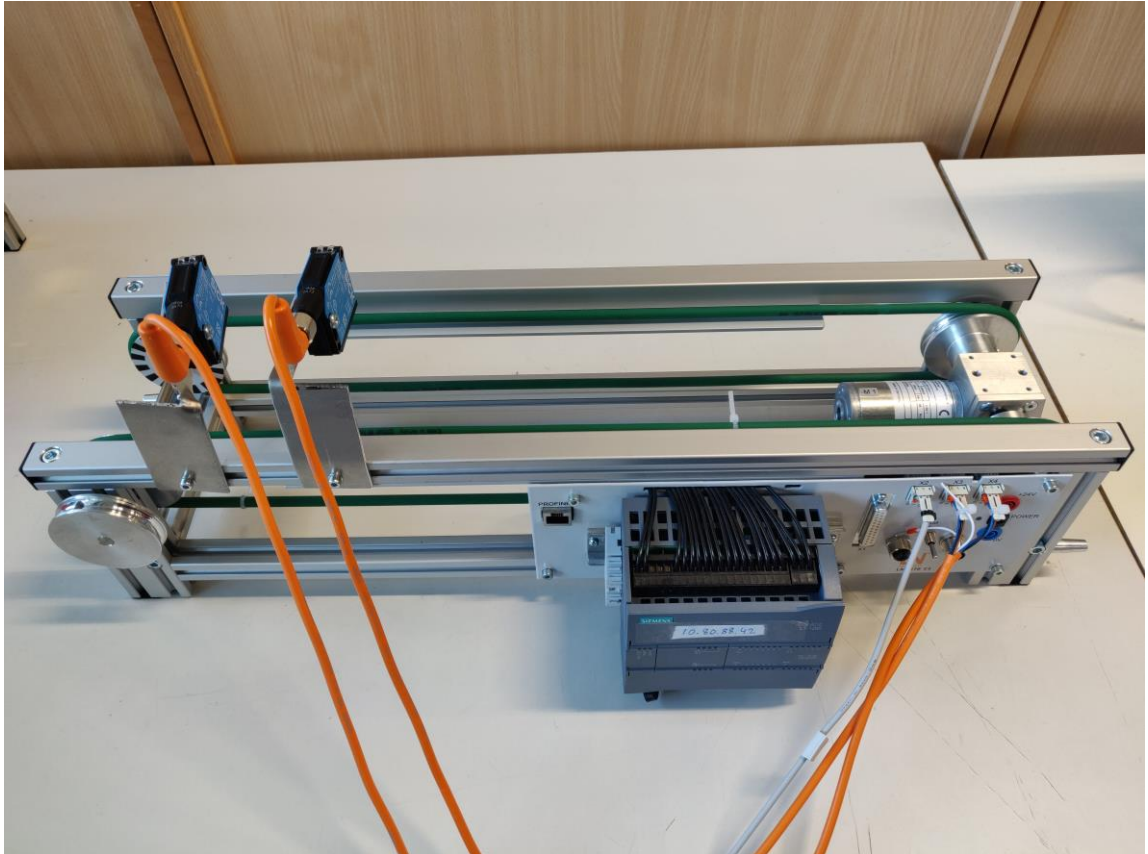
Ketjun pituuden muutosta voisi arvioida prosessiparametrien avulla, jossa hyödynnetään täytevalupadan valun ajoitukseen käytettävää optista anturia. Anturi tunnistaa muotin etureunan, jolloin logiikka saa high-signaalin niin kauan, kun muotti on valosilmän edessä. Tästä kahden muotin välisestä low-ajasta olisi mahdollista laskea muottien etäisyys (kuva 19). Jos muottien välinen etäisyys muuttuu, tämä tarkoittaa, että ketju on venynyt. Yhdellä kierroksella saisi 1 293 mittaustulosta, joista otetaan keskiarvo, ja tätä keskiarvoa seuraamalla voidaan arvioida, onko muottien välinen matka kasvanut.



Kuva 19. Sininen käyrä on signaali, punainen on mitattava väli.

#### 4.3 Pilotti muottiketjun pituuden muutoksen mittaamisesta

Muottiketjun mittauksesta tehtiin pilotti erillisellä alustalla. Pilotilla todistetaan, että ketjun pituuden muutoksen arviointi on mahdollista. (Kuva 20.)

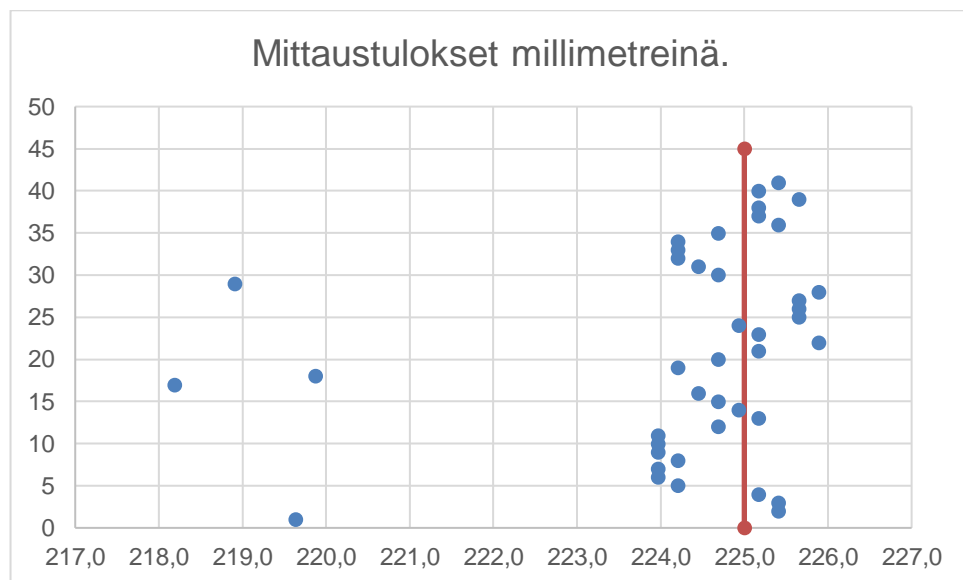


Kuva 20. Pilottialusta

Erillinen alusta on Lucas-Nuellen valmistama pyöröhihnakuljetin. Kuljetin on osa LN:n valmistamaa Smart Factory-koulutussettiä, joka koostuu useista kuljettimista ja robotiikyksiköistä. Alustaan on kiinnitetty Siemens S7-1200 PLC, 24V DC-moottori ja kaksi SICK WTB11-2P2461-optista anturia. Alusta pyörittää kahta pyöröhihnaa. Toiseen pyöröhihnaan kiinnitettiin nippusiteitä, jotka optiset anturit tunnistavat. Anturien ja nippusiteiden välimatkat on mitattu.

Pyöröhihnan nopeus lasketaan ajasta, jonka nippuside kulkee molempien antureiden läpi ja mitattu välimatka jaetaan läpikulkuajalla. Tätä nopeutta hyödynnetään nippusiteiden välisen etäisyyden laskemiseen. Nippusiteiden välinen etäisyys lasketaan niin että, kun ensimmäinen nippuside on ohittanut anturin, se aloittaa ajan laskemisen, ja kun anturi tunnistaa seuraavan nippusiteen, se lopettaa ajan laskemisen. Nopeus kerrotaan signaalien välillä kuluneella ajalla ja tästä saadaan matka.

Todellinen väli, jota mitattiin, oli rullamitalla mitattuna 225mm. Mittaustulokset olivat muutamaa poikkeamaa lukuun ottamatta kahden millimetrin sisällä. Mittaustulosten tarkkuuteen vaikuttaa kuinka tasaisella nopeudella pyöröhihna pyörii. Kuvassa 21 nähdään tulosten hajauma, pystyviiva kuvaa todellista matkaa, jota mitattiin. Tulokset pysyvät alle prosentin sisällä todellisuudesta.



Kuva 21. Mittaustulokset millimetreinä. Pystyakselilla mittaustuloksen numero ja vaaka-akselilla pituus.

Siemens S7-1200-logiikalla pystytään mittaamaan aikaa millisekunnin tarkkuudella. Matkan mittauksessa täytyy ottaa huomioon antureiden tarkkuus varsinkin, kun on kyseessä optinen anturi, jonka valokeila on 6 mm:n halkaisijalla. Tarkemmalla anturilla päästään tarkempaan mittaustulokseen.



## 5 Yhteenveto

Tämän insinööriyön tavoitteena oli selvittää, miten prosessidataa voisi hyödyntää kunnonvalvonnassa, erityisesti Fazer Makeiset Oy:n Universal-linjalla. Työ aloitettiin tutustumalla alan kirjallisuuteen ja linjan toimintaan sekä haastatteleamalla yrityksen kunnossapitopäällikköä Aleksi Kyröä ja Siemensin tuotepäälliköitä Elisa Vanhataloa ja Arja Heikkistä. Tämän jälkeen keräämäni tiedon pohjalta pohdin, missä Universal-linjan prosessista saatavan datan kerääminen ja analysoiminen olisi mahdollista ja kunnonvalvonnan kannalta hyödyllistä. Lopuksi prosessidatan hyödyntämisestä Universal-linjan muottiketjun pituuden muutoksen osalta tehtiin pilotti ja teoriaa testattiin käytännössä.

Työn toteutuksen suurimpana haasteena oli tarjolla olevan tiedon vähäisyys. Tämä johtuu osittain siitä, että kunnonvalvonta prosessidataa hyödyntämällä on monelle yritykselle myytävä tuote ja täten salassa pidettävää tietoa. Siemensiltä sain kuitenkin apua kahden tuotepäällikön kautta, jotka esittelivät PCS7:n ja MindSpheren ja miten niitä on hyödynnetty kunnonvalvonnassa. MindSpheren hyötyjen tutkiminen jäi kokonaan teoriatasolle, johtuen siitä, että MindSpheren käyttö vaatii lisenssin. Kummallakaan, Metropolialla tai Fazer Makeisilla, ei ole lisenssiä, jota kautta olisi voinut tutkia MindSpheren tarjoamia mahdollisuuksia.

Työn lopputuloksena oli se, että Universal-linjalta tulevan prosessidatan perusteella olisi mahdollista tehdä kunnonvalvontaa jäähdytystunneleiden vesi-glykolikiertoon sekä muottiketjuun, sen pituuden muutoksen osalta. Linjalle suunnitellun säiliöiden pinnankorkeuden mittauksen lisäämistä pystyisi käyttämään apuna pumppujen kunnonvalvonnassa. Pilotilla taas voitiin todistaa että IloT:n tuomat hyödyt ovat todellisia, aina ei tarvitse asentaa erillisiä kunnonvalvonta-antureita suorittaakseen kunnonvalvontaa. Syy-yhteyksien löytäminen vaatii perehtymistä prosessiin ja mitattaviin suureisiin, mutta hyödyt ovat todellisia.

Insinöörityössä tehtyä selvitystyötä voisi hyödyntää tai jatkaa tehtaan muilla tuotantolinjoilla, lisäten tehtaan yleistä kunnonvalvontaa ja ennakoivaa kunnossapitoa. Kunnonvalvontatuloksia pystyisi käyttämään apuna ennakoivan kunnossapidon päätöksenteossa. Työsuunnittelija voi lisätä tarkempia kunnonvalvontakierroksia tuloksien perusteella ja tilata huoltoon tarvittavia varaosia jo ennen kuin vika on aiheuttanut tuotantomenetyksiä hidastuneen tuotantonopeuden tai laiterikon takia.

## Lähteet

- 1 PSK 7501. Prosessiteollisuuden kunnossapidon tunnusluvut. 2010. Helsinki: PSK Standardisointi.
- 2 SFS-ISO 17359, Koneiden kunnonvalvonta ja diagnostiikka. Yleiset periaatteet. 2011. Helsinki: Suomen Standardisoimisliitto.
- 3 Alastalo Reijo; Bärling Matti; Hirvonen Mauri; Hyppönen Heikki; Issakainen Onni; Packalén Erkki; Saarinen Lars & Väyrynen Pertti. Mekaniikka - Johdanto kunnonvalvontaan. Verkkoaineisto. Opetushallitus. <[http://www03.edu.fi/oppimateriaalit/kunnossapito/mekaniikka\\_k1\\_johdanto\\_kunnonvalvontaan.html](http://www03.edu.fi/oppimateriaalit/kunnossapito/mekaniikka_k1_johdanto_kunnonvalvontaan.html)>. Luettu 13.3.2019.
- 4 Mikkonen, Henry. 2009. Kuntoon perustuva kunnossapito. 1., painos. Helsinki: KP-media oy.
- 5 Järviö, Jorma & Lehtiö, Taina. 2017. Kunnossapito: tuotanto-omaisuuden hoitaminen. 5., uudistettu painos. Helsinki: KP-media oy.
- 6 Alastalo, Reijo; Bärling, Matti; Hirvonen, Mauri; Hyppönen, Heikki; Issakainen, Onni; Packalén, Erkki; Saarinen, Lars & Väyrynen, Pertti. Mekaniikka - Lämpökamera. Verkkoaineisto. Opetushallitus. <[http://www03.edu.fi/oppimateriaalit/kunnossapito/mekaniikka\\_k5\\_lampokamera.html](http://www03.edu.fi/oppimateriaalit/kunnossapito/mekaniikka_k5_lampokamera.html)>. Luettu 17.3.2019.
- 7 Rousku, Henrik. 2014. Sähkölaitteiston lämpökuvaukseen voi säästää tulipalolta. Verkkoaineisto. Promaint. <<https://promaintlehti.fi/Kunnonvalvonta-ja-kayttovaruuks/Sahkolaitteiston-lampokuvaus-voi-saastaa-tulipalolta>>. Luettu 20.3.2019
- 8 Ahonen Tero. 2011. Monitoring of centrifugal pump operation by a frequency converter. Diplomityö. Lappeenrannan teknillinen yliopisto.
- 9 Vanhatalo, Elisa. 2019. Tuotepäällikkö, Siemens, Vantaa. Keskustelu 30.1.2019.
- 10 PSK-käsikirja 3 - Kunnonvalvonnan värähtelymittaus. 2019. 19., painos. Helsinki: KP-media oy.
- 11 PSK 6201 - Kunnossapito. Käsitteet ja määritelmät. 3. painos. 30 s. 2011
- 12 Heikkinen, Arja. 2019. Tuotepäällikkö, Siemens, Vantaa. Keskustelu 18.2.2019.

- 13 Siemens. 2019. <<https://support.industry.siemens.com/cs/document/109738815/simatic-pcs-7-condition-monitoring-library-v8-2-ready-for-delivery?dti=0&lc=en-WW>>. Luettu 23.3.2019.
- 14 Kyrö, Aleksi 2019. Kunnossapitopäällikkö, Fazer Makeiset Oy, Vantaa. Keskustelu 20.1.2019.
- 15 PSK 5707. Kunnanvalvonnan värähtelymittaus. Vianmääritys. 2019. 6. painos. Helsinki: PSK- Standardisointi.

## Pilotin logiikkaohjelma

## Program blocks

### Main [OB1]

Main Properties							
General							
Name	Main	Number	1	Type	OB	Language	FBD
Numbering	automatic						
Information							
Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

### Network 1: Moottorin ohjaus

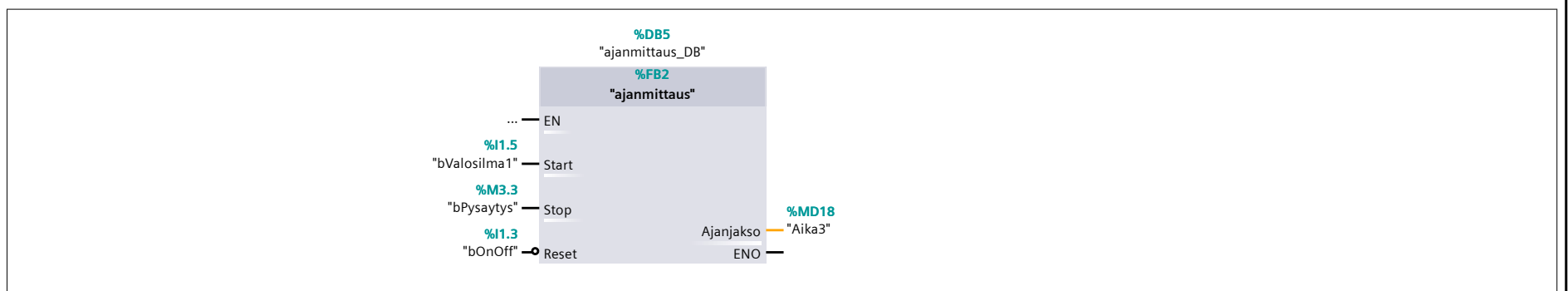
Ohjataan moottori päälle ja pois.



Symbol	Address	Type	Comment
"bMoottori"	%Q1.1	Bool	
"bOnOff"	%I1.3	Bool	

### Network 2: Kahden nousevan signaalin välisen ajan mittaus

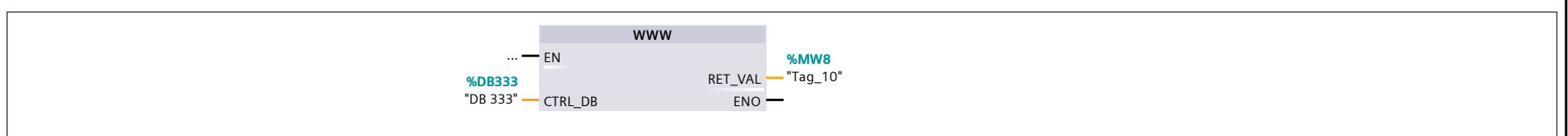
Hyödynnetään "ajanmittaus"-funktionblockia ajan mittaamiseen. Tuloksena "Aika3"-muuttuja.



Symbol	Address	Type	Comment
"Aika3"	%MD18	Time	
"bOnOff"	%I1.3	Bool	
"bPysaytys"	%M3.3	Bool	
"bValosilma1"	%I1.5	Bool	

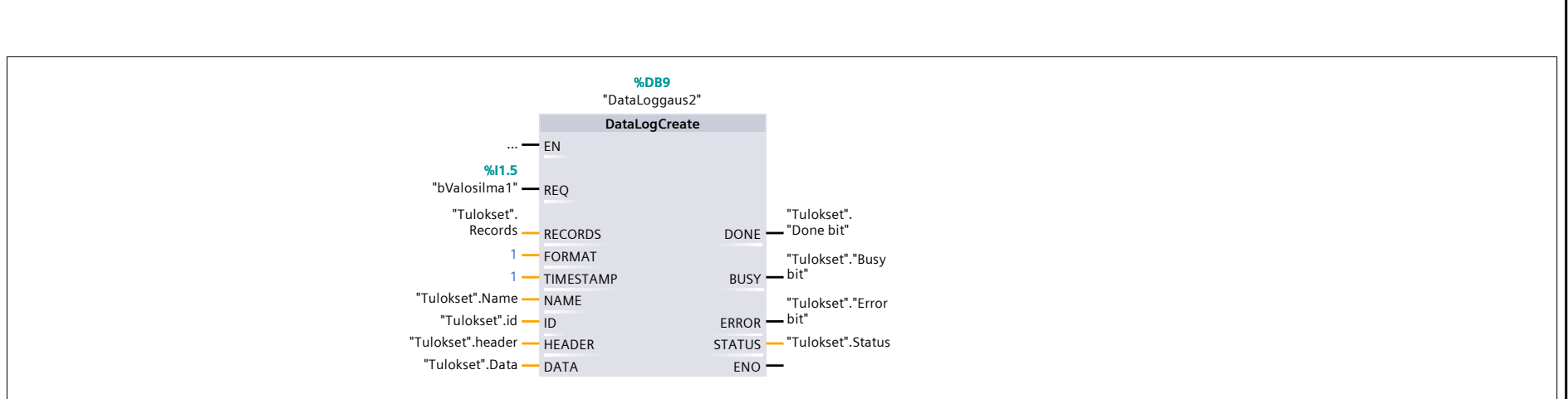
### Network 3: web-serverin luominen

Luodaan web-serveri jotta tietokannat voidaan ladata selaimen kautta.



Symbol	Address	Type	Comment
"Tag_10"	%MW8	Int	

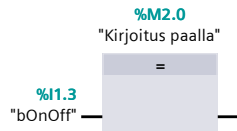
### Network 4: Tulokset-tietokannan luonti



Symbol	Address	Type	Comment
"bValosilma1"	%I1.5	Bool	
"Tulokset"."Busy bit"		Bool	
"Tulokset"."Done bit"		Bool	

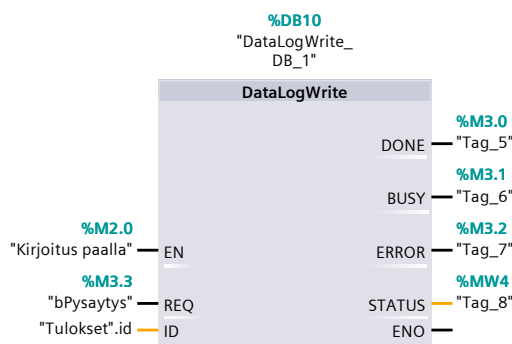
Symbol	Address	Type	Comment
"Tulokset"."Error bit"		Bool	
"Tulokset".Data		Struct	
"Tulokset".header		String	
"Tulokset".id		DWord	
"Tulokset".Name		String	
"Tulokset".Records		UDInt	
"Tulokset".Status		Int	

### Network 5: Kirjoituksen käynnistys



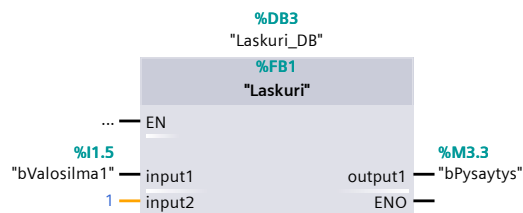
Symbol	Address	Type	Comment
"bOnOff"	%I1.3	Bool	
"Kirjoitus paalla"	%M2.0	Bool	

### Network 6: Tietokantaan kirjoittaminen



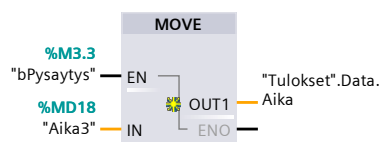
Symbol	Address	Type	Comment
"bPysaytys"	%M3.3	Bool	
"Kirjoitus paalla"	%M2.0	Bool	
"Tag_5"	%M3.0	Bool	
"Tag_6"	%M3.1	Bool	
"Tag_7"	%M3.2	Bool	
"Tag_8"	%MW4	Word	
"Tulokset".id		DWord	

### Network 7: Laskuri



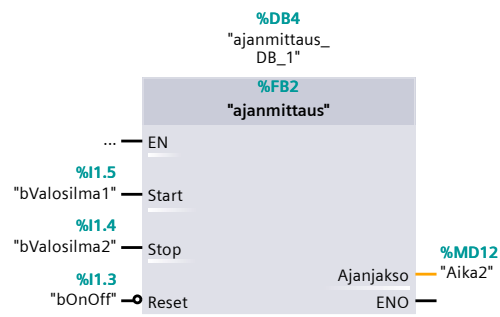
Symbol	Address	Type	Comment
"bPysaytys"	%M3.3	Bool	
"bValosilma1"	%I1.5	Bool	

### Network 8: Ajan vienti Tulokset-tietokantaan



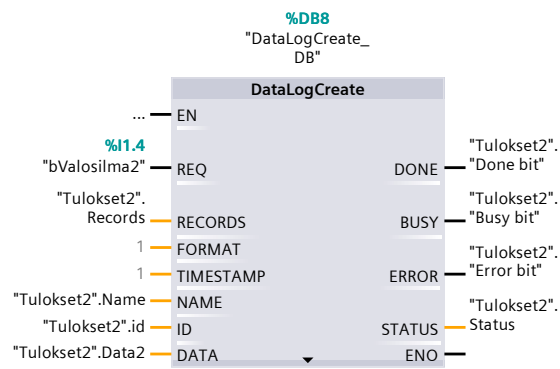
Symbol	Address	Type	Comment
"Aika3"	%MD18	Time	
"bPysaytys"	%M3.3	Bool	
"Tulokset".Data.Aika		Time	

### Network 9: Ajanmittaus



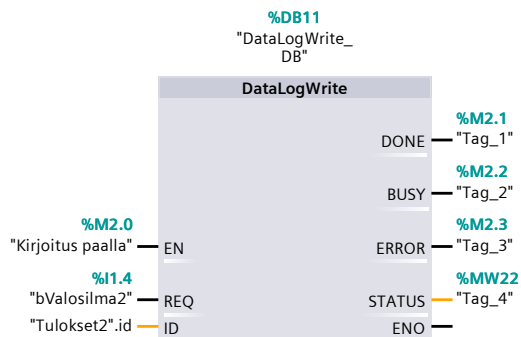
Symbol	Address	Type	Comment
"Aika2"	%MD12	Time	
"bOnOff"	%I1.3	Bool	
"bValosilma1"	%I1.5	Bool	
"bValosilma2"	%I1.4	Bool	

### Network 10: Tulokset2-tietokannan luonti



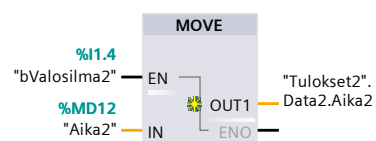
Symbol	Address	Type	Comment
"bValosilma2"	%I1.4	Bool	
"Tulokset2".Busy bit		Bool	
"Tulokset2".Done bit		Bool	
"Tulokset2".Error bit		Bool	
"Tulokset2".Data2		Struct	
"Tulokset2".id		DWord	
"Tulokset2".Name		String	
"Tulokset2".Records		UDInt	
"Tulokset2".Status		Int	

### Network 11: Tulokset2-tietokantaan kirjoitus



Symbol	Address	Type	Comment
"bValosilma2"	%I1.4	Bool	
"Kirjoitus paalla"	%M2.0	Bool	
"Tag_1"	%M2.1	Bool	
"Tag_2"	%M2.2	Bool	
"Tag_3"	%M2.3	Bool	
"Tag_4"	%MW22	Word	
"Tulokset2".id		DWord	

### Network 12: Ajan vieminen Tulokset2-tietokantaan



Symbol	Address	Type	Comment
"Aika2"	%MD12	Time	
"bValosilma2"	%I1.4	Bool	
"Tulokset2".Data2.Aika2		Time	



## Program blocks

### ajanmittaus [FB2]

#### ajanmittaus Properties

##### General

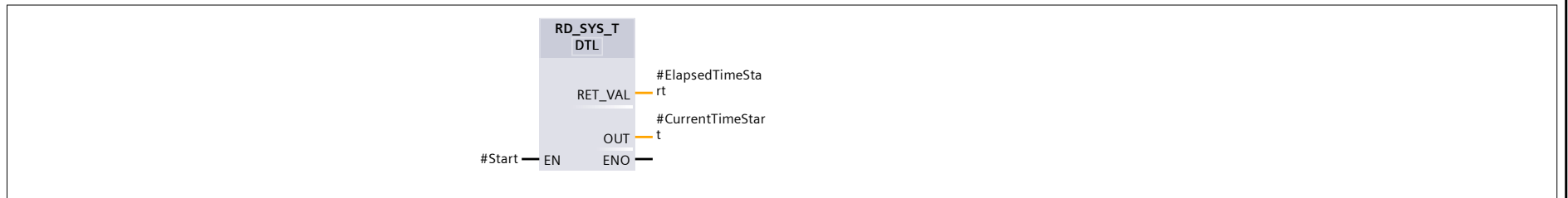
Name	ajanmittaus	Number	2	Type	FB	Language	FBD
Numbering	automatic						

##### Information

Title		Author		Comment		Family	
Version	0.1	User-defined ID					

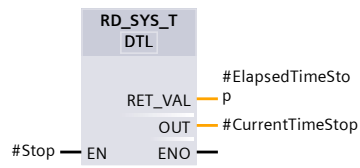
Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Start	Bool	false	Non-retain	True	True	False	
Stop	Bool	false	Non-retain	True	True	False	
Reset	Bool	false	Non-retain	True	True	False	
▼ Output							
Ajanjakso	Time	T#0ms	Non-retain	True	True	False	
InOut							
▼ Static							
▼ CurrentTime	DTL	DTL#1970-01-01-00:00:00	Non-retain	True	True	False	
YEAR	UInt	1970	Non-retain	True	True	False	
MONTH	USInt	1	Non-retain	True	True	False	
DAY	USInt	1	Non-retain	True	True	False	
WEEKDAY	USInt	5	Non-retain	True	True	False	
HOUR	USInt	0	Non-retain	True	True	False	
MINUTE	USInt	0	Non-retain	True	True	False	
SECOND	USInt	0	Non-retain	True	True	False	
NANOSECOND	UDInt	0	Non-retain	True	True	False	
ElapsedTimeStart	Int	0	Non-retain	False	False	False	
ElapsedTimeStop	Int	0	Non-retain	False	False	False	
▼ CurrentTimeStart	DTL	DTL#1970-01-01-00:00:00	Non-retain	False	False	False	
YEAR	UInt	1970	Non-retain	False	False	False	
MONTH	USInt	1	Non-retain	False	False	False	
DAY	USInt	1	Non-retain	False	False	False	
WEEKDAY	USInt	5	Non-retain	False	False	False	
HOUR	USInt	0	Non-retain	False	False	False	
MINUTE	USInt	0	Non-retain	False	False	False	
SECOND	USInt	0	Non-retain	False	False	False	
NANOSECOND	UDInt	0	Non-retain	False	False	False	
▼ CurrentTimeStop	DTL	DTL#1970-01-01-00:00:00	Non-retain	False	False	False	
YEAR	UInt	1970	Non-retain	False	False	False	
MONTH	USInt	1	Non-retain	False	False	False	
DAY	USInt	1	Non-retain	False	False	False	
WEEKDAY	USInt	5	Non-retain	False	False	False	
HOUR	USInt	0	Non-retain	False	False	False	
MINUTE	USInt	0	Non-retain	False	False	False	
SECOND	USInt	0	Non-retain	False	False	False	
NANOSECOND	UDInt	0	Non-retain	False	False	False	
ElapsedTimeStartConversion	Time_Of_Day	TOD#00:00:00	Non-retain	True	True	False	
ElapsedTimeStopConversion	Time_Of_Day	TOD#00:00:00	Non-retain	True	True	False	
Virhekoodi	Word	16#0	Non-retain	True	True	False	
KirjoitusKesken	Bool	false	Non-retain	True	True	False	
hyppy	Int	0	Non-retain	True	True	False	
hyppy2	String	"	Non-retain	True	True	False	
Temp							
Constant							

#### Network 1:



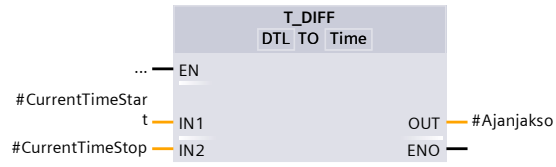
Symbol	Address	Type	Comment
#CurrentTimeStart		DTL	
#ElapsedTimeStart		Int	
#Start		Bool	

#### Network 2:



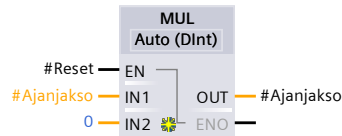
Symbol	Address	Type	Comment
#CurrentTimeStop		DTL	
#ElapsedTimeStop		Int	
#Stop		Bool	

**Network 3:**



Symbol	Address	Type	Comment
#Ajanjakso		Time	
#CurrentTimeStart		DTL	
#CurrentTimeStop		DTL	

**Network 4:**



Symbol	Address	Type	Comment
#Ajanjakso		Time	
#Reset		Bool	

## Program blocks

### ajanmittaus\_DB [DB5]

#### ajanmittaus\_DB Properties

##### General

<b>Name</b>	ajanmittaus_DB	<b>Number</b>	5	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Start	Bool	false	False	True	True	False	
Stop	Bool	false	False	True	True	False	
Reset	Bool	false	False	True	True	False	
▼ Output							
Ajanjakso	Time	T#0ms	False	True	True	False	
InOut							
▼ Static							
▼ CurrentTime	DTL	DTL#1970-01-01-00:00:00	False	True	True	False	
YEAR	UInt	1970	False	True	True	False	
MONTH	USInt	1	False	True	True	False	
DAY	USInt	1	False	True	True	False	
WEEKDAY	USInt	5	False	True	True	False	
HOUR	USInt	0	False	True	True	False	
MINUTE	USInt	0	False	True	True	False	
SECOND	USInt	0	False	True	True	False	
NANOSECOND	UDInt	0	False	True	True	False	
ElapsedTimeStart	Int	0	False	False	False	False	
ElapsedTimeStop	Int	0	False	False	False	False	
▼ CurrentTimeStart	DTL	DTL#1970-01-01-00:00:00	False	False	False	False	
YEAR	UInt	1970	False	False	False	False	
MONTH	USInt	1	False	False	False	False	
DAY	USInt	1	False	False	False	False	
WEEKDAY	USInt	5	False	False	False	False	
HOUR	USInt	0	False	False	False	False	
MINUTE	USInt	0	False	False	False	False	
SECOND	USInt	0	False	False	False	False	
NANOSECOND	UDInt	0	False	False	False	False	
▼ CurrentTimeStop	DTL	DTL#1970-01-01-00:00:00	False	False	False	False	
YEAR	UInt	1970	False	False	False	False	
MONTH	USInt	1	False	False	False	False	
DAY	USInt	1	False	False	False	False	
WEEKDAY	USInt	5	False	False	False	False	
HOUR	USInt	0	False	False	False	False	
MINUTE	USInt	0	False	False	False	False	
SECOND	USInt	0	False	False	False	False	
NANOSECOND	UDInt	0	False	False	False	False	
ElapsedTimeStartConversion	Time_Of_Day	TOD#00:00:00	False	True	True	False	
ElapsedTimeStopConversion	Time_Of_Day	TOD#00:00:00	False	True	True	False	
Virhekoodi	Word	16#0	False	True	True	False	
KirjoitusKesken	Bool	false	False	True	True	False	
hyppy	Int	0	False	True	True	False	
hyppy2	String	"	False	True	True	False	

## Program blocks

### Tulokset [DB6]

#### Tulokset Properties

##### General

<b>Name</b>	Tulokset	<b>Number</b>	6	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
Records	UDInt	500	False	True	True	False	
Name	String	'Data'	False	True	True	False	
id	DWord	16#0	False	True	True	False	
header	String	'aika'	False	True	True	False	
▼ Data	Struct		False	True	True	False	
Aika2	Time	T#0ms	False	True	True	False	
Aika	Time	T#0MS	False	True	True	False	
Done bit	Bool	false	False	True	True	False	
Busy bit	Bool	false	False	True	True	False	
Error bit	Bool	false	False	True	True	False	
Status	Int	0	False	True	True	False	

## Program blocks

### Laskuri [FB1]

#### Laskuri Properties

##### General

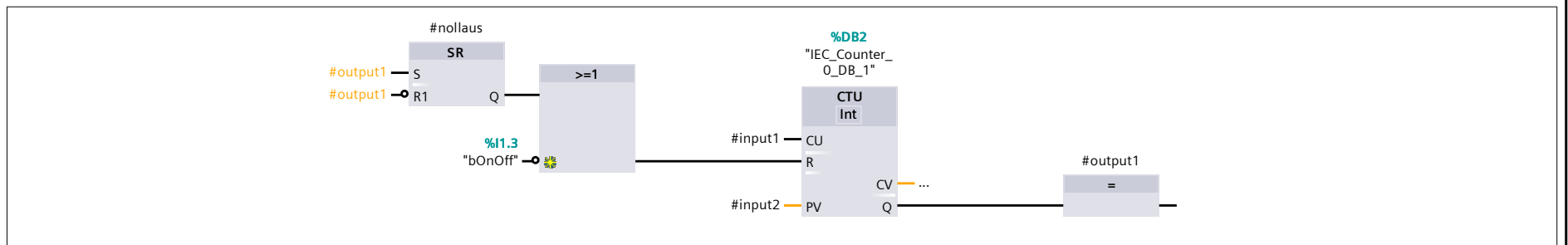
<b>Name</b>	Laskuri	<b>Number</b>	1	<b>Type</b>	FB	<b>Language</b>	FBD
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Default value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
input1	Bool	false	Non-retain	True	True	False	
input2	Int	0	Non-retain	True	True	False	
▼ Output							
output1	Bool	false	Non-retain	True	True	False	
InOut							
Static							
▼ Temp							
nollaus	Bool						
Constant							

#### Network 1:



Symbol	Address	Type	Comment
"bOnOff"	%I1.3	Bool	
#input1		Bool	
#input2		Int	
#nollaus		Bool	
#output1		Bool	

## Program blocks

### Laskuri\_DB [DB3]

#### Laskuri\_DB Properties

##### General

<b>Name</b>	Laskuri_DB	<b>Number</b>	3	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
input1	Bool	false	False	True	True	False	
input2	Int	0	False	True	True	False	
▼ Output							
output1	Bool	false	False	True	True	False	
InOut							
Static							

## Program blocks

### Laskuri\_DB\_1 [DB1]

#### Laskuri\_DB\_1 Properties

##### General

<b>Name</b>	Laskuri_DB_1	<b>Number</b>	1	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
input1	Bool	false	False	True	True	False	
input2	Int	0	False	True	True	False	
▼ Output							
output1	Bool	false	False	True	True	False	
InOut							
Static							

## Program blocks

### ajanmittaus\_DB\_1 [DB4]

#### ajanmittaus\_DB\_1 Properties

##### General

<b>Name</b>	ajanmittaus_DB_1	<b>Number</b>	4	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
Start	Bool	false	False	True	True	False	
Stop	Bool	false	False	True	True	False	
Reset	Bool	false	False	True	True	False	
▼ Output							
Ajanjakso	Time	T#0ms	False	True	True	False	
InOut							
▼ Static							
▼ CurrentTime	DTL	DTL#1970-01-01-00:00:00	False	True	True	False	
YEAR	UInt	1970	False	True	True	False	
MONTH	USInt	1	False	True	True	False	
DAY	USInt	1	False	True	True	False	
WEEKDAY	USInt	5	False	True	True	False	
HOUR	USInt	0	False	True	True	False	
MINUTE	USInt	0	False	True	True	False	
SECOND	USInt	0	False	True	True	False	
NANOSECOND	UDInt	0	False	True	True	False	
ElapsedTimeStart	Int	0	False	False	False	False	
ElapsedTimeStop	Int	0	False	False	False	False	
▼ CurrentTimeStart	DTL	DTL#1970-01-01-00:00:00	False	False	False	False	
YEAR	UInt	1970	False	False	False	False	
MONTH	USInt	1	False	False	False	False	
DAY	USInt	1	False	False	False	False	
WEEKDAY	USInt	5	False	False	False	False	
HOUR	USInt	0	False	False	False	False	
MINUTE	USInt	0	False	False	False	False	
SECOND	USInt	0	False	False	False	False	
NANOSECOND	UDInt	0	False	False	False	False	
▼ CurrentTimeStop	DTL	DTL#1970-01-01-00:00:00	False	False	False	False	
YEAR	UInt	1970	False	False	False	False	
MONTH	USInt	1	False	False	False	False	
DAY	USInt	1	False	False	False	False	
WEEKDAY	USInt	5	False	False	False	False	
HOUR	USInt	0	False	False	False	False	
MINUTE	USInt	0	False	False	False	False	
SECOND	USInt	0	False	False	False	False	
NANOSECOND	UDInt	0	False	False	False	False	
ElapsedTimeStartConversion	Time_Of_Day	TOD#00:00:00	False	True	True	False	
ElapsedTimeStopConversion	Time_Of_Day	TOD#00:00:00	False	True	True	False	
Virhekoodi	Word	16#0	False	True	True	False	
KirjoitusKesken	Bool	false	False	True	True	False	
hyppy	Int	0	False	True	True	False	
hyppy2	String	"	False	True	True	False	



## Program blocks

### Tulokset2 [DB7]

#### Tulokset2 Properties

##### General

<b>Name</b>	Tulokset2	<b>Number</b>	7	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>		<b>Comment</b>		<b>Family</b>	
<b>Version</b>	0.1	<b>User-defined ID</b>					

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
Records	UDInt	500	False	True	True	False	
Name	String	'Data2'	False	True	True	False	
id	DWord	16#0	False	True	True	False	
header	String	'aika2'	False	True	True	False	
▼ Data2	Struct		False	True	True	False	
Aika2	Time	T#0ms	False	True	True	False	
Done bit	Bool	false	False	True	True	False	
Busy bit	Bool	false	False	True	True	False	
Error bit	Bool	false	False	True	True	False	
Status	Int	0	False	True	True	False	

## Program blocks / System blocks / Program resources

### DataLoggaus2 [DB9]

#### DataLoggaus2 Properties

##### General

<b>Name</b>	DataLoggaus2	<b>Number</b>	9	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

##### Information

<b>Title</b>		<b>Author</b>	SIMATIC	<b>Comment</b>		<b>Family</b>	DataLog
<b>Version</b>	1.0	<b>User-defined ID</b>	DL_Creat				

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
REQ	Bool	false	False	True	True	False	
Records	UDInt	1	False	True	True	False	
Format	UInt	1	False	True	True	False	
Timestamp	UInt	1	False	True	True	False	
Name	Variant		False	False	False	False	
▼ Output							
DONE	Bool	false	False	True	True	False	
BUSY	Bool	false	False	True	True	False	
ERROR	Bool	false	False	True	True	False	
STATUS	Word	0	False	True	True	False	
▼ InOut							
ID	DWord	0	False	True	True	False	
Header	Variant		False	False	False	False	
Data	Variant		False	False	False	False	
Static							

Program blocks / System blocks / Program resources

DataLogWrite\_DB\_1 [DB10]

DataLogWrite\_DB\_1 Properties

General

<b>Name</b>	DataLogWrite_DB_1	<b>Number</b>	10	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

Information

<b>Title</b>		<b>Author</b>	SIMATIC	<b>Comment</b>		<b>Family</b>	DataLog
<b>Version</b>	1.0	<b>User-defined ID</b>	DL_Write				

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
REQ	Bool	false	False	True	True	False	
▼ Output							
DONE	Bool	false	False	True	True	False	
BUSY	Bool	false	False	True	True	False	
ERROR	Bool	false	False	True	True	False	
STATUS	Word	0	False	True	True	False	
▼ InOut							
ID	DWord	0	False	True	True	False	
Static							

Program blocks / System blocks / Program resources

IEC\_Counter\_0\_DB\_1 [DB2]

IEC\_Counter\_0\_DB\_1 Properties

General

<b>Name</b>	IEC_Counter_0_DB_1	<b>Number</b>	2	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

Information

<b>Title</b>		<b>Author</b>	Simatic	<b>Comment</b>		<b>Family</b>	IEC
<b>Version</b>	1.0	<b>User-defined ID</b>	CNTR				

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static							
CU	Bool	false	True	True	True	False	
CD	Bool	false	True	True	True	False	
R	Bool	false	True	True	True	False	
LD	Bool	false	True	True	True	False	
QU	Bool	false	True	True	True	False	
QD	Bool	false	True	True	True	False	
PV	Int	0	True	True	True	False	
CV	Int	0	True	True	True	False	

Program blocks / System blocks / Program resources

DataLogCreate\_DB [DB8]

DataLogCreate\_DB Properties

General

<b>Name</b>	DataLogCreate_DB	<b>Number</b>	8	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

Information

<b>Title</b>		<b>Author</b>	SIMATIC	<b>Comment</b>		<b>Family</b>	DataLog
<b>Version</b>	1.0	<b>User-defined ID</b>	DL_Creat				

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
REQ	Bool	false	False	True	True	False	
Records	UDInt	1	False	True	True	False	
Format	UInt	1	False	True	True	False	
Timestamp	UInt	1	False	True	True	False	
Name	Variant		False	False	False	False	
▼ Output							
DONE	Bool	false	False	True	True	False	
BUSY	Bool	false	False	True	True	False	
ERROR	Bool	false	False	True	True	False	
STATUS	Word	0	False	True	True	False	
▼ InOut							
ID	DWord	0	False	True	True	False	
Header	Variant		False	False	False	False	
Data	Variant		False	False	False	False	
Static							

Program blocks / System blocks / Program resources

DataLogWrite\_DB [DB11]

DataLogWrite\_DB Properties

General

<b>Name</b>	DataLogWrite_DB	<b>Number</b>	11	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	automatic						

Information

<b>Title</b>		<b>Author</b>	SIMATIC	<b>Comment</b>		<b>Family</b>	DataLog
<b>Version</b>	1.0	<b>User-defined ID</b>	DL_Write				

Name	Data type	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Input							
REQ	Bool	false	False	True	True	False	
▼ Output							
DONE	Bool	false	False	True	True	False	
BUSY	Bool	false	False	True	True	False	
ERROR	Bool	false	False	True	True	False	
STATUS	Word	0	False	True	True	False	
▼ InOut							
ID	DWord	0	False	True	True	False	
Static							

## Program blocks / System blocks / Web server

### DB 333 [DB333]

#### DB 333 Properties

##### General

<b>Name</b>	DB 333	<b>Number</b>	333	<b>Type</b>	DB	<b>Language</b>	DB
<b>Numbering</b>	manual						

##### Information

<b>Title</b>		<b>Author</b>	AWP_S7P	<b>Comment</b>		<b>Family</b>	WEB
<b>Version</b>	1.0	<b>User-defined ID</b>					

Name	Data type	Offset	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
▼ Static								
magic	DWord	0.0	DW#16#41575043	False	True	True	False	
consistency_tag	DWord	4.0	DW#16#05104075	False	True	True	False	
db_version	Word	8.0	W#16#0001	False	True	True	False	
length	UInt	10.0	110	False	True	True	False	
pagetab_of	UInt	12.0	92	False	True	True	False	
pagetab_count	UInt	14.0	0	False	True	True	False	
excludetab_of	UInt	16.0	92	False	True	True	False	
excludetab_count	UInt	18.0	0	False	True	True	False	
fragmentlist_of	UInt	20.0	92	False	True	True	False	
fragmentlist_count	UInt	22.0	0	False	True	True	False	
fragmenttab_of	UInt	24.0	98	False	True	True	False	
fragmenttab_count	UInt	26.0	0	False	True	True	False	
datatab_of	UInt	28.0	98	False	True	True	False	
datatab_count	UInt	30.0	0	False	True	True	False	
usenetab_of	UInt	32.0	98	False	True	True	False	
usenetab_count	UInt	34.0	0	False	True	True	False	
enumreftab_of	UInt	36.0	98	False	True	True	False	
enumreftab_count	UInt	38.0	0	False	True	True	False	
enumtab_of	UInt	40.0	98	False	True	True	False	
enumtab_count	UInt	42.0	0	False	True	True	False	
textlist_of	UInt	44.0	98	False	True	True	False	
textlist_count	UInt	46.0	12	False	True	True	False	
language_frag_tab_of	UInt	48.0	92	False	True	True	False	
language_frag_tab_count	UInt	50.0	6	False	True	True	False	
application_name	UInt	52.0	1	False	True	True	False	
application_url	UInt	54.0	2	False	True	True	False	
application_desc	UInt	56.0	12	False	True	True	False	
enum_defs_fragment_start	UInt	58.0	0	False	True	True	False	
enum_defs_fragment_count	UInt	60.0	0	False	True	True	False	
▼ commandstate	Struct	62.0		False	True	True	False	
last_error	Int	0.0	0	False	True	True	False	
debug_mode	Int	2.0	0	False	True	True	False	
init	Bool	4.0	True	False	True	True	False	
deactivate	Bool	4.1	False	False	True	True	False	
initializing	Bool	4.2	False	False	True	True	False	
error	Bool	4.3	False	False	True	True	False	
deactivating	Bool	4.4	False	False	True	True	False	
deactivated	Bool	4.5	False	False	True	True	False	
initialized	Bool	4.6	False	False	True	True	False	
reserved1	Bool	4.7	False	False	True	True	False	
reserved2	Byte	5.0	B#16#00	False	True	True	False	
▼ requesttab	Array[1..4] of Struct	68.0		False	True	True	False	
▼ requesttab[1]	Struct	0.0		False	True	True	False	
page_index	UInt	0.0	0	False	True	True	False	
fragment_index	UInt	2.0	0	False	True	True	False	
continue	Bool	4.0	False	False	True	True	False	
repeat	Bool	4.1	False	False	True	True	False	
abort	Bool	4.2	False	False	True	True	False	
finish	Bool	4.3	False	False	True	True	False	
idle	Bool	4.4	False	False	True	True	False	
waiting	Bool	4.5	False	False	True	True	False	
sending	Bool	4.6	False	False	True	True	False	
aborting	Bool	4.7	False	False	True	True	False	
reserved3	Byte	5.0	B#16#00	False	True	True	False	
▼ requesttab[2]	Struct	6.0		False	True	True	False	
page_index	UInt	0.0	0	False	True	True	False	
fragment_index	UInt	2.0	0	False	True	True	False	
continue	Bool	4.0	False	False	True	True	False	
repeat	Bool	4.1	False	False	True	True	False	
abort	Bool	4.2	False	False	True	True	False	
finish	Bool	4.3	False	False	True	True	False	
idle	Bool	4.4	False	False	True	True	False	
waiting	Bool	4.5	False	False	True	True	False	
sending	Bool	4.6	False	False	True	True	False	
aborting	Bool	4.7	False	False	True	True	False	

Name	Data type	Offset	Start value	Retain	Accessible from HMI	Visible in HMI	Setpoint	Comment
reserved3	Byte	5.0	B#16#00	False	True	True	False	
▼ requesttab[3]	Struct	12.0		False	True	True	False	
page_index	UInt	0.0	0	False	True	True	False	
fragment_index	UInt	2.0	0	False	True	True	False	
continue	Bool	4.0	False	False	True	True	False	
repeat	Bool	4.1	False	False	True	True	False	
abort	Bool	4.2	False	False	True	True	False	
finish	Bool	4.3	False	False	True	True	False	
idle	Bool	4.4	False	False	True	True	False	
waiting	Bool	4.5	False	False	True	True	False	
sending	Bool	4.6	False	False	True	True	False	
aborting	Bool	4.7	False	False	True	True	False	
reserved3	Byte	5.0	B#16#00	False	True	True	False	
▼ requesttab[4]	Struct	18.0		False	True	True	False	
page_index	UInt	0.0	0	False	True	True	False	
fragment_index	UInt	2.0	0	False	True	True	False	
continue	Bool	4.0	False	False	True	True	False	
repeat	Bool	4.1	False	False	True	True	False	
abort	Bool	4.2	False	False	True	True	False	
finish	Bool	4.3	False	False	True	True	False	
idle	Bool	4.4	False	False	True	True	False	
waiting	Bool	4.5	False	False	True	True	False	
sending	Bool	4.6	False	False	True	True	False	
aborting	Bool	4.7	False	False	True	True	False	
reserved3	Byte	5.0	B#16#00	False	True	True	False	
▼ language_frag_tab	Struct	92.0		False	True	True	False	
max_language	UInt	0.0	0	False	True	True	False	
fragment_count	UInt	2.0	0	False	True	True	False	
nolang_offset	UInt	4.0	0	False	True	True	False	
▼ textlist	Array[1..12] of Byte	98.0		False	True	True	False	
textlist[1]	Byte	0.0	16#0	False	True	True	False	
textlist[2]	Byte	1.0	16#69	False	True	True	False	
textlist[3]	Byte	2.0	16#6E	False	True	True	False	
textlist[4]	Byte	3.0	16#64	False	True	True	False	
textlist[5]	Byte	4.0	16#65	False	True	True	False	
textlist[6]	Byte	5.0	16#78	False	True	True	False	
textlist[7]	Byte	6.0	16#2E	False	True	True	False	
textlist[8]	Byte	7.0	16#68	False	True	True	False	
textlist[9]	Byte	8.0	16#74	False	True	True	False	
textlist[10]	Byte	9.0	16#6D	False	True	True	False	
textlist[11]	Byte	10.0	16#0	False	True	True	False	
textlist[12]	Byte	11.0	16#0	False	True	True	False	