

# metsoDNA CR

---

## Automaatiokieli

Secured Life Cycle  
Collection 2009 rev. 3  
G2047\_FI\_03

---

Metso Automation Oy varaa oikeuden tehdä muutoksia tämän julkaisun sisältämiin määrittelyihin ja muihin tietoihin ilman etukäteisilmoitusta, ja lukijan tulee kaikissa tapauksissa ottaa yhteyttä Metso Automation Oy:öön selvittääkseen, onko sellaisia muutoksia tehty. Tämä julkaisu on tarkoitettu Metso Automation Oy:n asiakkaan yksinomaiseen käyttöön, eikä sitä tule kopioida tai luovuttaa kolmansille osapuolille.

Metso Automation Oy:n valmistamien/toimittamien laitteisto- ja ohjelmistotuotteiden lisensioinnin ja käytön ehdot käyvät ilmi yksinomaisesti toimittavan yksikön ja asiakkaan välisessä kirjallisessa sopimuksessa. Tämän julkaisun sisältämät tiedot tai lausumat esim. tuotteen kapasiteetista, suorituskyvystä tai tuotteen sopivuudesta tiettyyn käyttötarkoitukseen eivät muodosta takuuta eivätkä sopimuksen osaa, eivätkä ne velvoita Metso Automation Oy:tä tai toimittavaa yksikköä.

Metso Automation Oy ja toimittava yksikkö eivät ole vastuussa vahingoista, jotka aiheutuvat tästä julkaisusta tai ovat yhteydessä sen sisältämiin tietoihin (mukaanlukien välilliset vahingot) edes siinä tapauksessa, että Metso Automation Oy:tä tai toimittavaa yksikköä olisi informoitu, tai että Metso Automation Oy:llä tai toimittavalla yksiköllä oli tai olisi pitänyt olla tiedossa sellaisen vahingon mahdollisuus.

© Metso Automation Oy, 1988–2009. Kaikki oikeudet varattu. Painettu Suomessa.



Metso Automation Oy  
PL 237, 33101 Tampere  
Puh. 020 483 170, Telefax 020 483 171

## DOKUMENTIN HISTORIA

PÄIVÄYS	REVISIO	KOMMENTIT
2007-08-21	1	metsoDNA CR Collection 2007 Tämä dokumentti perustuu metsoDNA-dokumenttiin Automaatiokieli, Collection 2006 rev. 7 (G001007)
2008-05-21	2	metsoDNA CR Collection 2008 Ei sisältömuutoksia.
2009-06-02	3	metsoDNA CR Collection 2009 Ei sisältömuutoksia.



# SISÄLLYS

<b>1</b>	<b>JOHDANTO</b> .....	<b>1</b>
1.1	LIITTYVÄT DOKUMENTIT .....	1
<b>2</b>	<b>AUTOMAATIOSOVELLUS</b> .....	<b>2</b>
<b>3</b>	<b>AUTOMAATIOKIELEN KÄSITTEET</b> .....	<b>4</b>
3.1	MODUULIT JA AUTOMAATIOKOHTEET .....	4
3.1.1	Automaatiomoduulit .....	4
3.2	DOKUMENTTIMODUULIT .....	6
3.3	AUTOMAATIOKOHTEET .....	6
3.4	KONFIGUROINTIMODUULIT .....	7
3.5	TYYPPIMODUULIT ELI TYYPIT .....	7
3.5.1	Toimilohkotyytit eli toimilohkot .....	12
3.5.2	Nipputyytit .....	12
<b>4</b>	<b>SOVELLUSOHJELMAN PERUSOSAT</b> .....	<b>13</b>
4.1	TIETOPISTEET .....	13
4.1.1	Paikalliset tietopisteet .....	13
4.1.2	Ulkoiset tietopisteet .....	14
4.2	PORTIT .....	15
4.2.1	Suorasaantiportti .....	15
4.2.2	Rajapintaportti .....	17
4.3	TOIMILOHKOT .....	17
4.3.1	Konfiguroitavat tiedot .....	19
4.3.2	Kytettävät tiedot .....	19

<b>5</b>	<b>AUTOMAATIOKIELINEN SOVELLUSOHJELMA</b> .....	<b>22</b>
5.1	AUTOMAATIOMODUULIN RAKENNE .....	22
5.2	KONFIGUROIDIMODUULIN RAKENNE .....	24
5.2.1	Hallintaosa .....	24
5.2.2	Esittelyosa .....	26
5.2.3	Toimintaosa .....	35
5.3	MODUULIN SISÄISET KYTKENNÄT .....	42
5.4	MODUULIEN VÄLINEN KOMMUNIKOINTI .....	46
5.4.1	Suorasaantiportin käyttö kommunikoinnissa .....	47
5.4.2	Rajapintaportin käyttö kommunikoinnissa .....	49
5.4.3	Näkökulman käyttö kommunikoinnissa .....	50
5.5	ESIMERKKI KONFIGUROIDIMODUULIN RAKENTEESTA, KYTKENNÖISTÄ JA KOMMUNIKOINNISTA .....	51
<b>6</b>	<b>AUTOMAATIOKIELEN NIMEYSKÄYTÄNTÖ</b> .....	<b>55</b>
6.1	YLEISTÄ .....	55
6.2	MODUULIN NIMEN RAKENNE JA PITUUS .....	55
6.3	MODUULIEN NIMISSÄ KÄYTETTÄVÄT MERKIT .....	55
6.4	NIMEN HAKEMISTOTUNNUS .....	56
6.4.1	Automaatiomodulit .....	56
6.4.2	Automaatiokohteet .....	56
6.4.3	Konfiguroidimoduulit .....	57
6.5	MODUULIN NIMEN VALVOMOTUNNUS .....	58
6.6	MODUULIN NIMEN POSITIO-OSA .....	58
6.7	KUVAMODUULIEN NIMEYS .....	60
6.8	JÄRJESTELMÄMODUULIEN NIMEYS .....	60
6.9	OPEROINTIPALVELIMEN TUOTTAMIEN MODUULIEN NIMEYS .....	61
6.9.1	Historiamoduulien nimeys .....	61
6.9.2	Piirturien skaalausmoduulien nimeys .....	61
6.9.3	Systeemimoduulien nimeys .....	61
6.9.4	Pohjamoduulien nimeys .....	62
6.10	MODUULIT TYÖKALUITTAIN .....	62
6.11	MODUULIN SIJAINNITIEETO .....	64
6.11.1	Sijaintitiedon rakenne .....	64

<b>7</b>	<b>VIKABITTIKÄYTÄNTÖ</b>	<b>66</b>
7.1	VIKABITTIEN MERKITYKSET	66
7.2	MITÄ ERI VIKABITIT INDIKOIVAT	66
7.2.1	ext – ulkoinen vika	67
7.2.2	ovf – ylitys/alitus	67
7.2.3	dis – ei ohjattavissa	67
7.2.4	inv – kelvoton tieto	67
7.2.5	old – uudistumaton tieto	68
7.2.6	der – johdettu vika	68
7.2.7	sex – poikkeuksellinen tietolähde	68
7.3	VIKABITEISTÄ SIGNAALIHÄLYTYKSIÄ	68
7.4	VIKABITTIEN KÄYTÖSSÄ HUOMIOITAVIA SEIKKOJA	68
7.5	VIKABITTIKÄYTÄNNÖN SOVELTAMISESTA	69
7.5.1	Tyypien alkuarvot	69
7.5.2	Alkuarvot moduuleissa	69
7.5.3	Kytettävät tiedot	69

**LIITE 1 ALKEISTYYPIT**

**LIITE 2 YLEISET RAKENTEISET TYYPIT**

**LIITE 3 GRAFIKKAKIELEN KOMENNOT**





# 1 JOHDANTO

Tämä dokumentti on tarkoitettu metsoDNA CR:n sovellussuunnittelijalle.

Dokumentti kuvaa automaatiokielen peruskäsitteet ja automaatiokielen sovellusohjelman rakenteen. Lisäksi tarkastellaan automaatiokielen käyttäjää tukevia piirteitä sekä automaatiokielisten ohjelmamoduulien erilaisia muotoja metsoDNA CR:ssä.

Automaatiokielellä määritellään metsoDNA CR:n toiminta eli sillä kuvataan sen sovellusohjelma. Kielen perustana on kytkentätyyppinen toimilohkoihin ja monipuolisiin tyypeihin tukeutuva kieli, jota voidaan laajentaa esimerkiksi lausekemuotoisella esitystavalla laskennalle, logiikalle ja vertailuille sekä grafiikan lausekielisellä esitystavalla.

Automaatiokielen tavoitteena on luoda selkeä ja looginen malli metsoDNA CR:stä ja sen konfigurointitiedoista sovellussuunnittelijalle. Koska kielen sovellusalue on rajoitettu, se kykenee tukemaan kielen käyttäjää merkittävästi. Sillä voidaan saavuttaa *automaatiosuunnittelussa* huomattavasti suurempi tehokkuus kuin käyttämällä yleisiä ohjelmointikieliä (esim. C, FORTRAN, PASCAL jne.).

Sovellussuunnittelun käyttäjäliityntään on tehty suunnitteluohjelmistot, jotka tukevat sovelluksen tekijää mm. tarjoamalla mahdollisuuden graafiseen suunnitteluun.

## 1.1 LIITTYVÄT DOKUMENTIT

### Suunnittelumallinnus

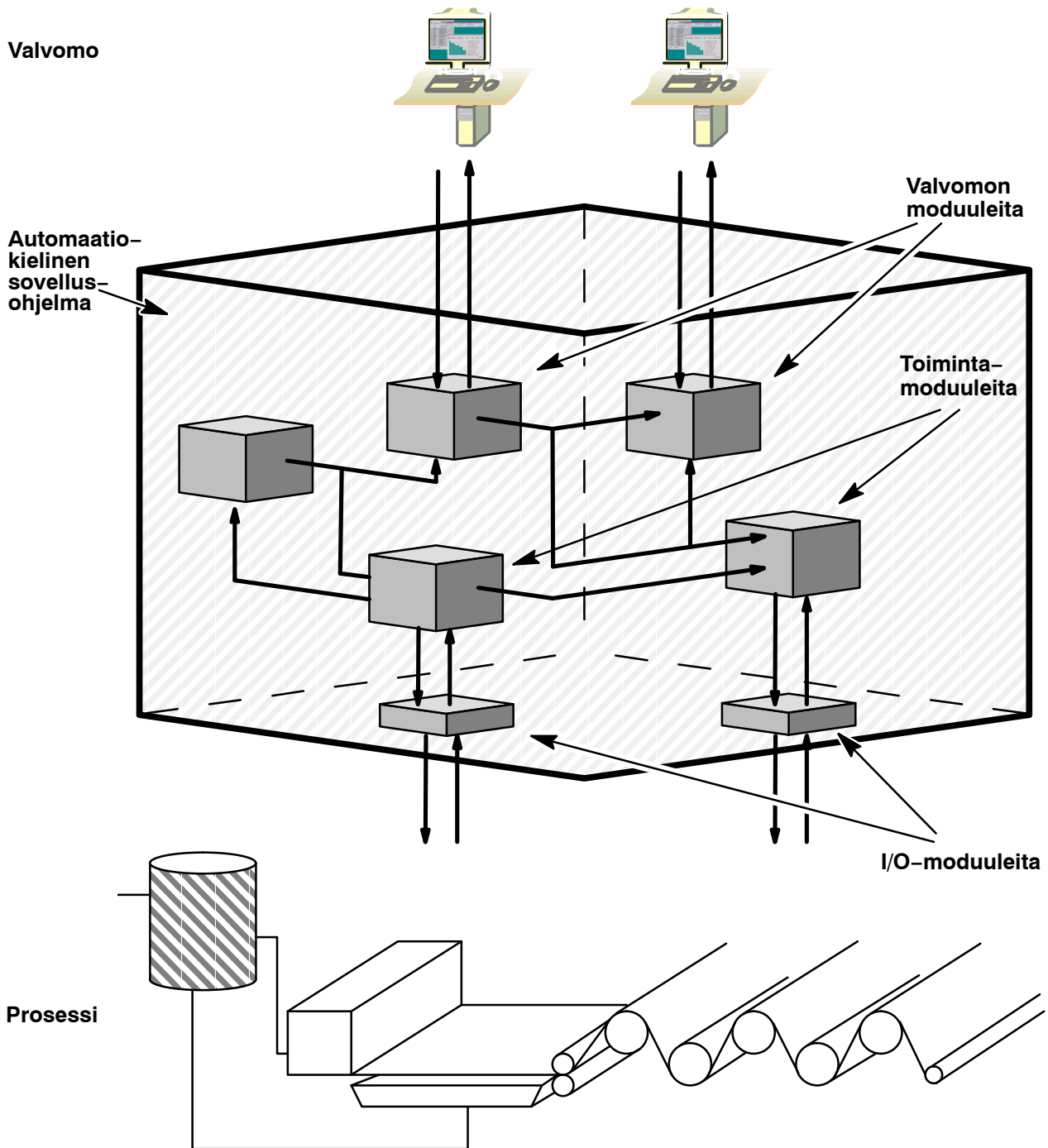
Suunnittelumallinnus-dokumentissa on esitelty metsoDNA CR:n suunnittelumallinnuksessa esiintyvät uudet ja muuttuneet käsitteet ja termit Damatic XDi:n automaatiokieleen verrattuna. Myös useat muuttumattomina pysyneet käsitteet on esitelty lyhyesti kokonaisuuden hahmottamiseksi.

metsoDNA CR:n suunnittelumallinnuksessa uutta ilmaisuvoimaa antavia uusia käsitteitä on vain yksi, prosessialue. Uusien työkalujen, varsinkin DNAexplorerin, vaatimana on vanhoja käsitteitä kuitenkin uudistettu ja jäsennelty uuteen ryhmittelyyn. Joitakin uusia, lähinnä ryhmittelyyn liittyviä apukäsitteitä on myös lisätty.

Suunnittelumallinnus-dokumentissa esiteltävät uudet käsitteet ja termit näkyvät suunnittelujärjestelmässä lähinnä vain uusien työkalujen, kuten DNAexplorerin, käyttöliittymissä.

## 2 AUTOMAATIOSOVELLUS

Automaatiosovellus eli automaatio-ohjelmisto koostuu automaatiokielen moduuleista (Kuva 1).

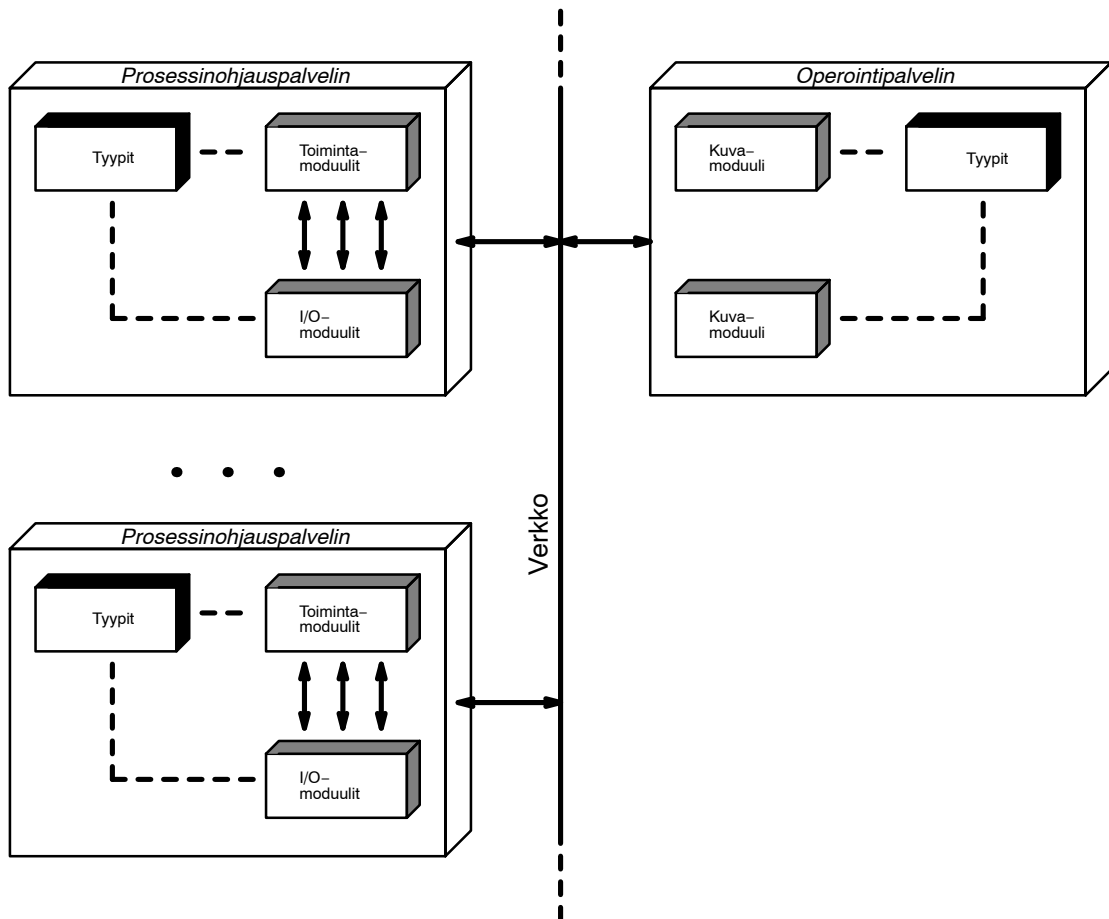


Kuva 1 Automaatiokielineen sovellusohjelma

Moduulit ovat suunnittelijan ja prosessinohjauksen kannalta mielekkäitä kokonaisuuksia. Ne ovat pienimpiä yksittäisiä ohjelmalohkoja, joita voidaan kerralla ladata aktiviteettien sovelluspalvelimille.

Moduuli voi olla esim. yksittäinen säätö- tai mittauspiiri, kokonainen sekvenssiohjelma, moottorien ryhmäohjaus tai monitorilla näkyvä kuva.

Automaatiokielen käsitteet luovat metsoDNA CR:stä ja sen konfiguroinnista suunnittelijalle seuraavan mukaisen mallin (Kuva 2).



Kuva 2 Suunnittelijan näkökulma metsoDNA CR:ään

Suunnittelija näkee metsoDNA CR:ssä aktiviteetteja ja komponentteja, joille hän sijoittaa tekemänsä moduulit. Moduulit rakentuvat erilaisista osista, jotka määritellään tyyppien avulla. Sovelluspalvelin-tyyppiset komponentit sisältävät aina vakiokirjaston tyyppimoduuleita. Moduulit kommunikoivat keskenään kopioimalla tarvittavia tietoja ja muodostavat näin automaatiosovelluksen.

Automaatiokielen moduulirakenteisuus tukee suunnittelijaa antamalla mahdollisuuden oletusarvojen sekä aiemmin tehdyn työn hyväksikäyttöön. Oletusarvotoiminnot tapahtuvat kuitenkin suunnittelutyöasemalla; kun tietoa siirretään sovelluspalvelimille, tiedosta ei voi enää päätellä, onko kyse oletusarvosta vai suunnittelijan syöttämästä arvosta.

## 3 AUTOMAATIOKIELEN KÄSITTEET

### 3.1 MODUULIT JA AUTOMAATIOKOHTEET

Automaatiokielessä käsitellään hyvin erityyppisiä tietoja, joten eri tiedoille on luotu eri käsitteet. Automaatiokielen käsitteisiin kuuluvat olennaisesti erilaiset moduulit ja automaatiokohteet. Moduulit jakaantuvat lisäksi neljään pääryhmään.

Automaatiokielen pääkäsitteitä ovat:

- automaatiomoduulit
- dokumenttimoduulit
- konfigurointimoduulit
- tyyppimoduulit eli tyypit
- automaatiokohteet.

#### 3.1.1 Automaatiomoduulit

Sovelluksen suunnittelun käyttäjäliityntään on suunniteltu graafiset ohjelmistot, joilla havainnollistetaan suunnittelua. Näiden graafisten suunnittelutyökaluohjelmien tuottamista kuvista muodostetaan automaatiokielineen ohjelma eli konfigurointimoduulit, joka on ladattavissa metsoDNA CR:n sovelluspalvelimille. Graafisten suunnittelutyökaluohjelmien tuotteita nimitetään *automaatiomuoduleiksi*.

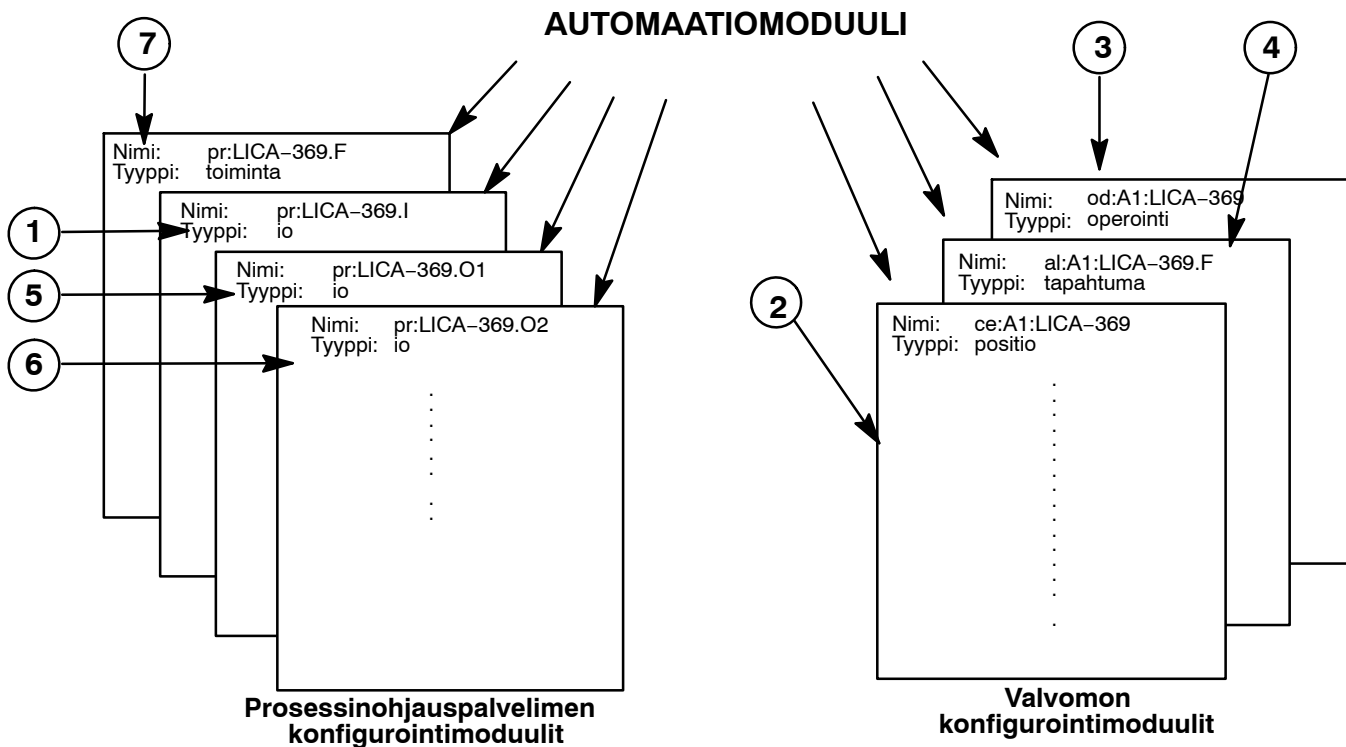
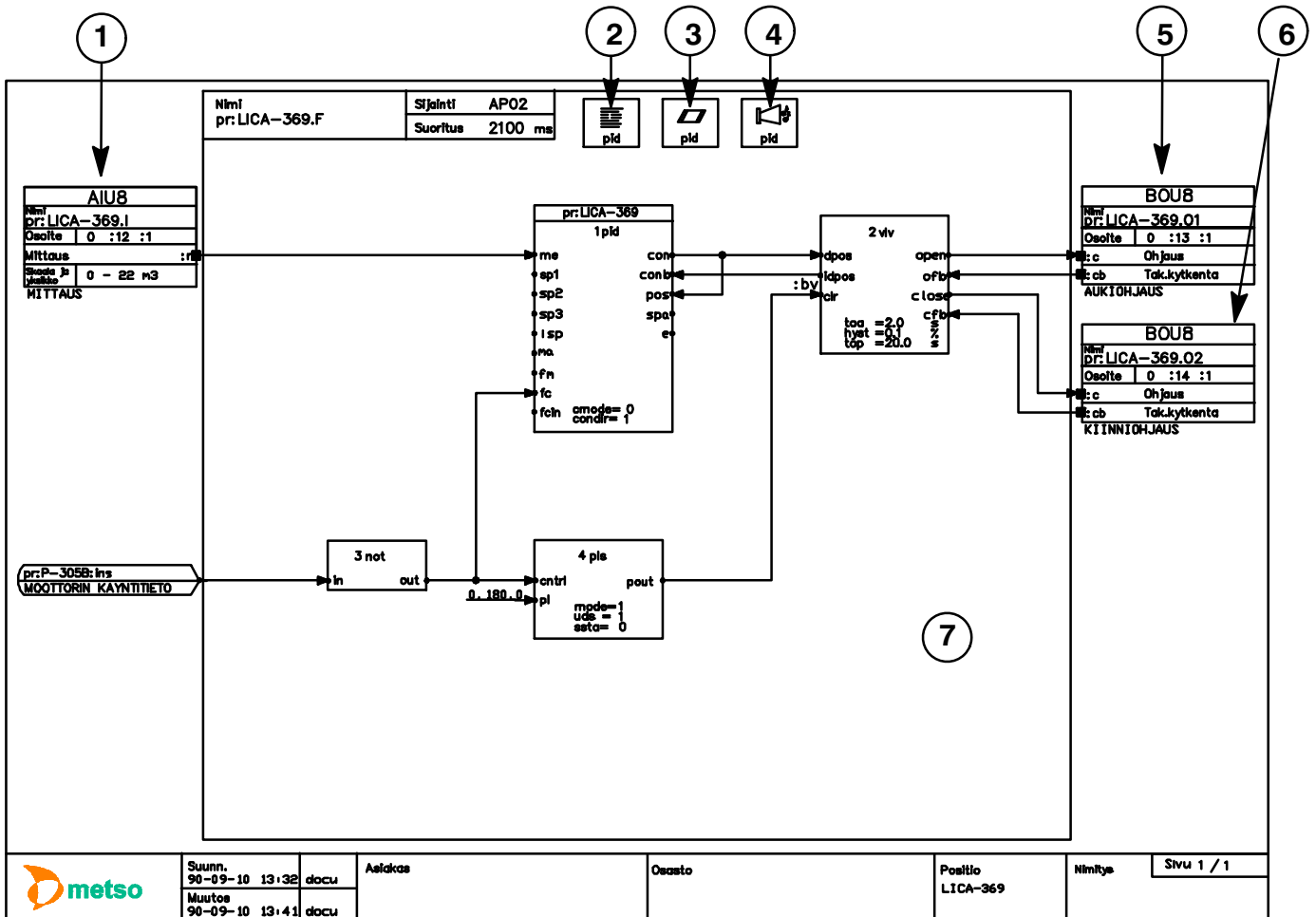
Automaatiomoduuli on graafinen kuvaus sovellusohjelman osasta. Graafinen moduulin kuvaus on palautettavissa listamuotoiseksi (automaatiokieliseksi), mutta toiminta toisin päin ei ole mahdollista, eli listamuotoisesta kuvauksesta ei saada graafista automaatiomoduulia.

Automaatiomoduulit voivat sisältää useita konfigurointimoduleita. Lähes kaikki yhteen piiriin liittyvät moduulit voidaan integroida yhteen graafiseen automaatiomoduuliin.

Automaatiomoduulin suunnittelussa on käytettävissä monipuoliset kuvan suunnittelu- ja muokkaustoiminnot sekä kirjastot, joista saadaan automaatiomuoduleihin tarvittavia symboleita.

Seuraavassa kuvassa (Kuva 3) on esimerkki eräästä automaatiomoduulista. Esimerkin automaatiomoduulista muodostetaan seuraavat prosessiohjauspalvelimen ja valvomon konfigurointimoduulit:

- prosessiohjauspalvelimen toimintamoduuli
- prosessiohjauspalvelimen tulo- ja lähtömoduulit
- valvomon positiomoduuli
- valvomon tapahtumamoduuli
- valvomon operointimoduuli.



Kuva 3 Automaatiomoduuli ja siihen liittyvät konfigurointimoduulit

### 3.2 DOKUMENTTIMODUULIT

Dokumenttimoduulit ovat nimensäkin mukaisesti pelkästään sovellusta tai verkon rakennetta kuvaavia dokumentteja.

Dokumenttimoduulit eroavat automaatiomoduuleista sillä, että dokumenttimoduuleista ei generoida automaatiokielistä ohjelmaa eli konfigurointimoduuleita.

Dokumenttimoduuleita on seuraavia eri tyyppiä:

- piirikaaviodokumentit
- säätökaaviodokumentit
- logiikkakaavio
- laitteistokaavio
- kenttäpiirustusdokumentit.

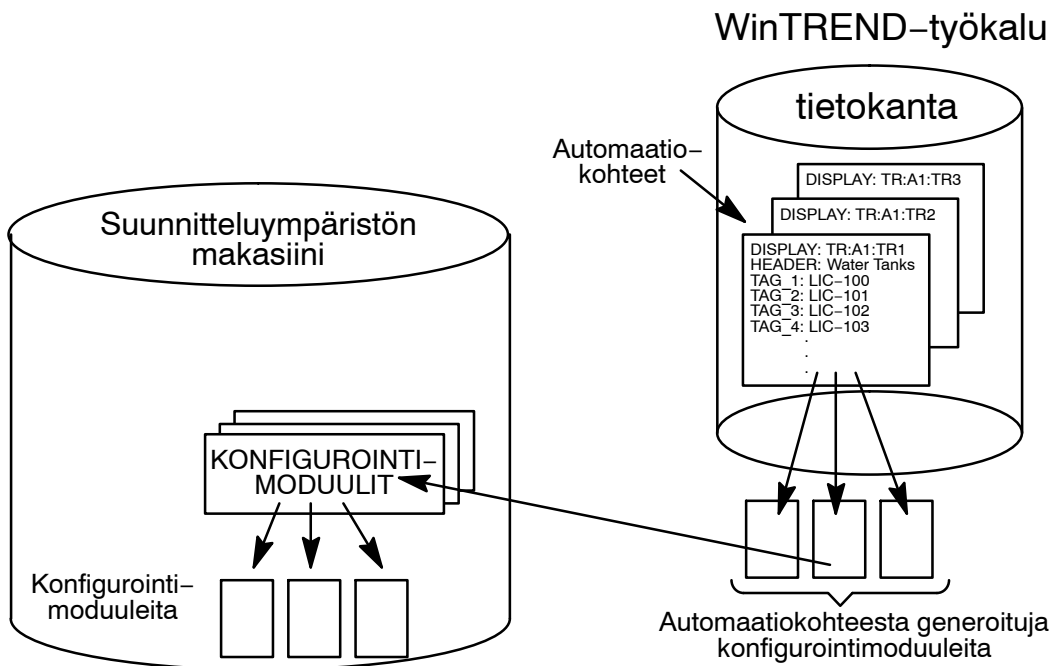
### 3.3 AUTOMAATIOKOHTEET

Automaatiokohde on tietokantapohjaisessa suunnittelutyökalussa yksi käsiteltävä itsenäinen kokonaisuus, joka muodostuu joukosta tietokannan taulukon tietoja. Automaatiokohde on olemassa vain tietokantatyökalun tietokannassa, eikä sitä voi käsitellä moduuleiden tavoin suunnittelutyökalun työtiloissa eikä makasiinissa.

Automaatiokohteita ovat:

- piirturikuvien ja -kytkentöjen tiedot
- trendien kuva- ja kytkentöjen tiedot.

Automaatiokohteista generoidaan konfigurointimoduuleita. Automaatiokohteiden ja siitä generoitavien konfigurointimoduuleiden yhteyksiä WinTREND-työkalulla (trenditietojen suunnittelutyökalu) on esitetty seuraavassa kuvassa.



### 3.4 KONFIGUROINTIMODUULIT

metsoDNA CR:n toiminnan määrittelee varsinaisesti konfigurointimoduuleista koostuva sovellusohjelma. Ohjelman toimintoja varten määritellään erilaisia moduuleita: toimintamoduuleita, I/O-moduuleita, kuvamoduuleita jne.

Konfigurointimoduulit ovat toiminnallisia kokonaisuuksia ja automaatiokielen perusyksiköitä, joita yhteen kytkemällä muodostetaan automaatiokielineen sovellusohjelma. Sovellussuunnittelija voi käsitellä moduuleita suunnittelutyöasemalla ja ladata niitä sovelluspalvelimille sovelluksen toiminnan häiriintymättä.

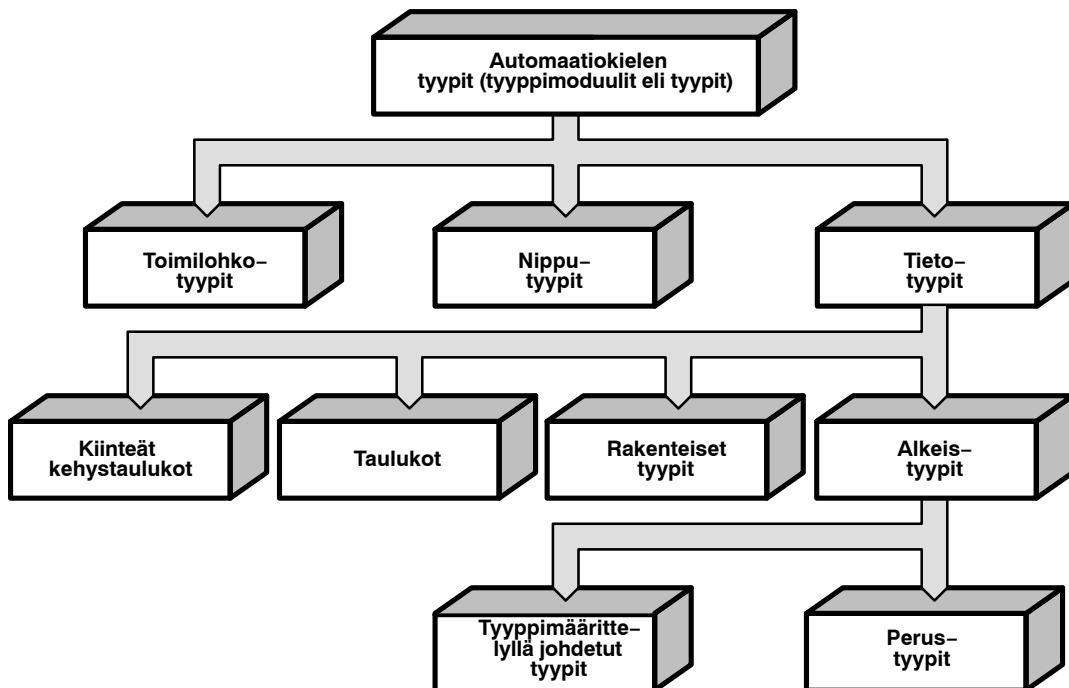
Konfigurointimoduulit koostuvat toimilohkoista, porteista ja tietopisteistä. Niihin voidaan lisätä myös laskentaa, logiikkaa ja vertailuja muodostamalla lausekielisiä osuuksia, jotka kuitenkin muotoillaan toimilohkojen tavoin. Kytkenällä muodostunutta konfigurointimoduulien kokonaisuutta voidaan nimittää myös metsoDNA CR:n sovellusohjelmistoksi.

### 3.5 TYYPPIMODUULIT ELI TYYPIT

*Tyypimoduulit eli tyypit* määrittelevät automaatiokielessä käytettävät tyypit: *toimilohkotyypit*, *tietotyypit* ja *nipputyypit*. metsoDNA CR:n sovelluspalvelimilla on aina valikoima kirjastoon kuuluvia kiinteitä tyyppisiä, sovellussuunnittelija ei voi itse suunnitella uusia tyyppisiä.

Automaatiokielen tyypit ovat rajattu osa yleisistä tyyppikäsitteistä. Ne jaetaan roolinsa perusteella kolmeen ryhmään:

- tietotyypit
- toimilohkotyypit eli toimilohkot
- nipputyypit



Kuva 4 Automaatiokielen tyyppihierarkia

## Tietotyypit

Automaatiokielen tietotyyppi kuvaa yhtenäisen tietoaueen. Automaatiokielen tietotyyppejä ovat:

- alkeistyytit
- rakenteiset tyypit
- taulukot
- kiinteät kehystaulukot

**ALKEISTYYPPEJÄ** ovat metsoDNA CR:n *kiinteät tyypit* eli *perustyytit* ja niistä yksinkertaisilla tyypimäärittelyillä *johdetut tyypit*.

*Perustyytpejä* ovat esimerkiksi:

- uns16 etumerkitön kokonaisluku, koko 2 tavua, lukualue (0...65535)
- float yksinkertaisen tarkkuuden liukuluku, koko 4 tavua, lukualue  $\pm(10^{-18} \dots 10^{18})$

*Johdettu alkeistyyppi* on esimerkiksi kiinteästä perustyytistä, etumerkitön kokonaisluku (uns16), johdettu alkeistyyppi fails. Se määritellään seuraavasti:

```
fails          TYYPPI uns16
```

Fails omaa samat ominaisuudet kuin uns16, joten johdettu alkeistyyppi on muodostettu itse asiassa perustyytin uudelleen nimeämisellä.

Automaatiokielen alkeistyytit (sekä perustyytit että johdetut tyypit) on esitetty liitteessä 1.

## Rakenteiset tyypit

Rakenteiset tyypit määritellään koostuvaksi muiden tyyppien avulla useista jäsenistä. Jäsenet voivat olla mitä tahansa alkeistyyppiä ja rakenteista tyyppiä. Kaikki jäsenet palautuvat kuitenkin tarvittaessa alkeistyypeiksi.

Esimerkki:

Rakenteinen tyyppi ana koostuu vikabittitiedosta ja signaalista.

Sen ensimmäinen jäsen on vikabittikenttä f ja toinen jäsen analogiasignaalin varsinainen arvo a SI-järjestelmän yksiköissä. Jäsen f on johdettua alkeistyyppiä fails ja a perustyyppiä float.

Ana:n rakenteen määrittely on siten seuraava:

```
ana
JÄSENET
  f TYYPPI fails
  a TYYPPI float
```

Esim. ana-tyyppisessä signaalissa (2,7.72)

2	f	vikatieto (syöttöhäiriö tai linjavika)
7.72	a	signaalin arvo SI-yksiköissä

Automaatiokielen yleiset rakenteiset tyypit on esitetty liitteessä 2.



## Taulukko

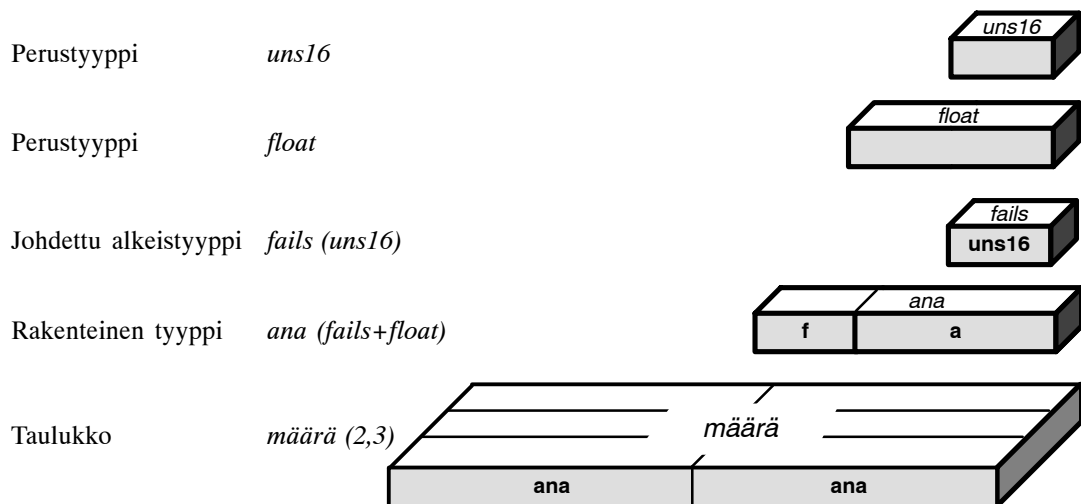
Taulukko on tietorakenne, jonka alkiot ovat samaa alkeistyyppiä tai rakenteista tyyppiä. Sen alkioihin voidaan viitata mielivaltaisessa järjestyksessä indeksejä käyttämällä. metsoDNA CR:n automaatiokielessä voidaan käyttää 1-, 2- tai 3-indeksisiä eli 1-, 2- tai 3-ulotteisia taulukoita. Kullekin indeksille sallittujen arvojen rajat ilmoitetaan taulukkoa esiteltäessä.

Esimerkki: Esitellään 2-ulotteinen taulukko määrä seuraavasti:

määrä (2, 3) TYPPI ana

Ensimmäisen indeksin raja on 2 ja toisen raja on 3. Taulukon alkiot ovat rakenteista tyyppiä ana.

Edellä mainitut esimerkit yhdistyvät (Kuva 5):



Kuva 5 Esimerkki tietotyyppien keskinäisestä hierarkiasta

## Kiinteä kehystaulukko

Kiinteä kehystaulukko on rakenne, jossa kiinteässä, tyyppin määräämässä kehyksessä on konfiguroitava ja ajon aikana muutettavissa oleva taulukko.

Kiinteän kehystaulukon kehyksen dimensiot määritellään taulukon tyyppissä. Tyyppissä määritellään myös kehyksen ja sen sisällä olevan taulukon alkioiden tyyppi. Niitä ei konfigurointivaiheessa tai suorituksen aikana voi muuttaa. Jokainen tarvittava alkiotyyppi ja dimensioiden yhdistelmä määritellään omana erillisenä tyyppinään. Kunkin metsoDNA CR -julkaisun tyyppiluettelosta käy ilmi, mitkä taulukkotyypit ko. koonnassa ovat käytettävissä.

Taulukkotyyppin nimestä käyvät ilmi sekä alkioiden tyyppi että kehyksen koko. Tyyppin nimi on muotoa typ\_ijk, esim. ana\_9 ja cha\_536.

Taulukkotyyppin nimessä (typ\_ijk) alkuosa (typ) on alkioiden tyyppin lyhenne. Lyhenteet ovat seuraavat (typ ja sitä vastaava alkion tyyppi):

### Perustyytit

cha = char	u32 = uns32
i8 = int8	b8 = bool8
i16 = int16	b16 = bool
i32 = int32	flo = float
u8 = uns8	fls = fails
u16 = uns16	–

### Yhteiset tyypit

ana = ana	inl = intl
bin = bin	bne = binev
ins = ints	ise = intsev

### Sovellustyytit

bo = bo
---------

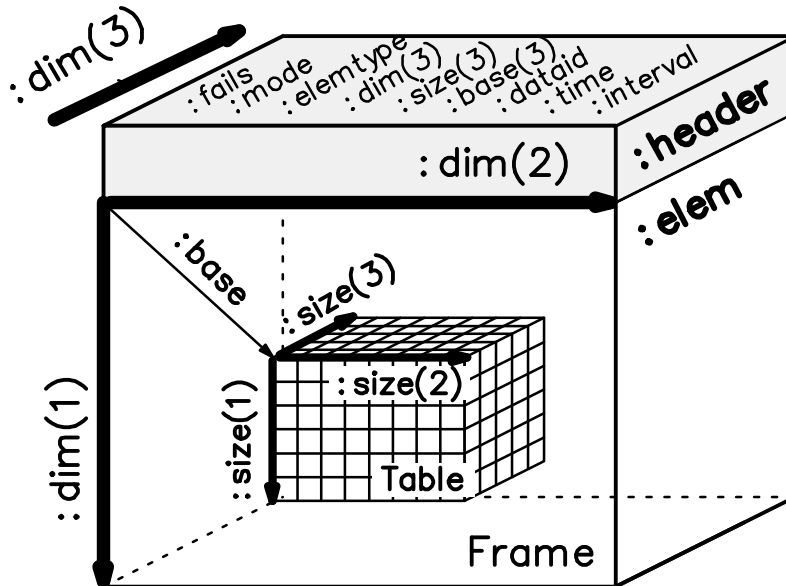
Taulukkotyyppin nimessä (typ\_ijk) loppuosa (ijk) ilmoittaa kehyksen eri dimensioiden suuruuden, kunkin yhdellä merkillä. Kehystaulukko voi olla korkeintaan 3–dimensioinen. i, j ja k ovat koodattuja kahden potenssin eksponentteja. Eksponenttien koodit ja niitä vastaavat dimensioiden suuruudet ovat seuraavat (koodi ja dimensio):

0 = 1	8 = 256
1 = 2	9 = 512
2 = 4	a = 1024
3 = 8	b = 2048
4 = 16	c = 4096
5 = 32	d = 8192
6 = 64	e = 16384
7 = 128	–

Esimerkiksi ana\_9 on 1–dimensioinen taulukko, jonka alkiot ovat tyyppiä ana ja jonka kehyksen koko on 512. cha\_536 taas on 3–dimensioinen taulukko, jonka alkiot ovat tyyppiä char ja kehyksen koko 32 x 8 x 64.

## Kehystaulukon rakenne

Tietorakenteena kehystaulukko on kaksiosainen. Siinä on hallintaosa (jäsen `:header`) ja alkio-osa (jäsen `:elem`). Kaikissa eri kokoisissa ja eri tyyppisiä alkioita sisältävissä kehystaulukoissa on samaa tyyppiä (`table`) oleva hallintaosa. Eri kokoiset ja eri tyyppisiä alkioita sisältävät kehystaulukot saadaan aikaan erilaisilla alkio-osilla ja hallintaosan erilaisella alkuarvotuksella.



Kuva 6 Kehystaulukon rakenne

- Hallintaosa

Kehystaulukon hallintaosa (`:header`) on tyyppiä `table`. Se sisältää kehyksen rakennetiedot (`:elemtype` ja `:dim`), taulukon rakennetiedot (`:mode`, `:size` ja `:base`) ja taulukkoa kuvaavia yksittäisiä skalaaritietoja (`:fails`, `:dataid`, `:time` ja `:interval`).

Hallintaosan tietoja voidaan lukea ja kirjoittaa kytkeytymällä asianomaiseen jäsenen, esim. `pr:PM6:BW:header:size(3)` tai `pr:PM6:BW:header:dataid`, tai kytkeytymällä koko kehystaulukkoon, esim. `pr:PM6:BW`.
- Alkio-osa

Kehystaulukon alkio-osa (`:elem`) on taulukon kehys. Se määrittelee kehyksen alkioiden tyypin ja kehyksen koon. Samalla se määrittelee taulukon alkioiden tyypin ja taulukon maksimikoon.

Alkio-osan tietoja voidaan lukea ja kirjoittaa kytkeytymällä haluttuun alkioon, esim. `pr:PM6:BW:elem(3)` tai `pr:QVT:elem(2,6,3)`, tai kytkeytymällä koko kehystaulukkoon, esim. `pr:PM6:BW`.

Kehystaulukon alkio-osassa on sisäkkäin kaksi erillistä osaa. Ulompi on tyypin määräämä kiinteä kehys (`frame`) ja sisempi kehyksen määräämissä puitteissa (maksimikoko ja alkioiden tyyppi) konfiguroitava ja ajon aikana muuttuva taulukko (`table`).

### 3.5.1 Toimilohkotyytit eli toimilohkot

Moduulit koostuvat pääosin toimilohkoista. Toimilohkot liittyvät muihin moduulin osiin kytketyymällä. Esimerkiksi kahden eri toimilohkon jäsenten välinen tiedonsiirto tapahtuu kytketyymisen avulla. Toimilohkotyytin kuvauksessa esitellään toimilohkon jäsenet, joista muodostuu toimilohkon kytkentärakenne.

Toimilohkojen jäsenet ovat jotain tyyppimäärittelyissä määriteltyä tyyppiä (esim. ana, bin, bo, float jne.).

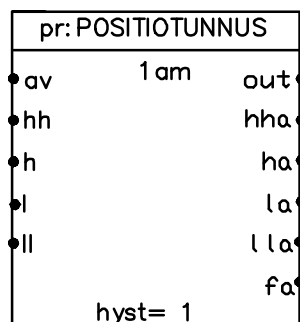
Toimilohkon jäsenet voidaan jakaa niiden toiminnan perusteella:

- tuloihin
- lähtöihin
- konfigurointiparametreihin.

Tuloista toimilohko lukee tietoa ja lähtöihin se kirjoittaa tietoa. Kytkemällä eri toimilohkojen jäseniä (tuloja ja lähtöjä) yhteen muodostetaan toimilohkojen välinen kytkentä. Kytkemisessä periaatteena on, että vain saman tyyppisiä jäseniä voidaan kytkeä toisiinsa, esim. ana–tyyppinen signaali voidaan kytkeä vain ana–tyyppiseen signaaliin, bin–tyyppinen signaali vain bin–tyyppiseen signaaliin jne.

Kytkevien jäsenten lisäksi toimilohkotyyppiin voi kuulua joukko konfiguroitavia parametreja. Niitä ei voi sovellusohjelman toiminnan aikana muuttaa, vaan niiden arvot määritellään sovellusta konfiguroidessa.

Toimilohkot kuvataan graafisessa suunnittelussa seuraaventyypisillä symboleilla, joissa symbolin toisella puolella on toimilohkon tulot (av, hh, ... ll) ja toisella puolella toimilohkon lähdöt (out, hha, ... fa). Symbolin keskiosassa voidaan esittää toimilohkon konfigurointiparametreja ja niiden arvoja (hyst=1).



### 3.5.2 Nipputyytit

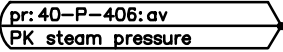


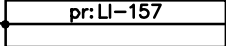
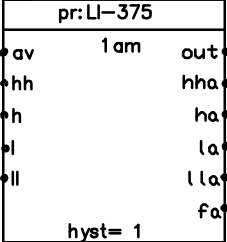
Moduulien välisessä tiedonsiirrossa käytetään hyväksi portteja. Suora toimilohkojen välinen tiedonsiirto ei ole mahdollista eri moduuleiden välillä. Moduulilla voi olla useita portteja, jotka ovat sen tiedonsiirtokanavia muille moduuleille. Muut moduulit käyttävät näitä portteja hyväksi kommunikoidessaan moduulin kanssa. Moduulin portti voi olla nipputyyppiä tai tietotyyppiä.

*Nipputyyppisen* portin jäsenten avulla voi moduulin sisältä eri kohdista kerätä tietoa yhdeksi kokonaisuudeksi, nipuksi. Nipputyyppisen portin tiedot voivat olla toisistaan riippumattomia, itsenäisiä moduulin sisällä olevia kytkettäviä tietoja. Nipun tiedot ovat käytettävissä muille moduuleille portin välityksellä.

Nipputyyppettä käytetään esim. prosessinohjauspalvelimen ja valvomon välisessä kommunikoinnissa. Esimerkiksi prosessinohjauspalvelimen tyytit cntb ja am.opb eivät sisällä mitään (toimilohko)–toimintaa, mutta niihin voidaan viitata valvomon moduuleista samoin kuin vastaaviin toimilohkoihin cnt ja am.

## 4 SOVELLUSOHJELMAN PERUSOSAT

Automaatio- ja konfigurointimoduulit koostuvat automaatiokielen perusosista (olioista), jotka jaetaan kolmeen ryhmään:

- tietopisteisiin
  -  ulkoinen tietopiste
  -  paikallinen tietopiste
- portteihin
  -  rajapintaportti
  -  suorasaaantiportti
  -  toimilohkoihin

Tietopiste määritellään tietotyypin, portti nipputyyppin tai tietotyypin ja toimilohko toimilohkotyyppin avulla.

Automaatiokielen olioiden välillä vallitsee moduulin sisällä vahva tyyppitys; eli vain samantyyppiset oliot voivat operoida keskenään. Moduulien välisessä kommunikoinnissa vahvaa tyyppitystä on hallitusti heikennetty. Tyyppille on määriteltävä ne tyytit, joiden kanssa se pystyy kommunikoimaan. Sille on määriteltävä myös erilaisia näkökulmia, jolloin oliosta tietoa kysyttäessä poimitaan vain kysytyn näkökulman mukaiset tiedot. Näin saadaan välitettyä vain kulloinkin tarvittavat tiedot. Näitä tyyppityksiä ja näkökulmia ei suunnittelija voi määrittellä eikä muuttaa.

### 4.1 TIETOPISTEET

Tietopisteet voivat olla paikallisia tai ulkoisia.

#### 4.1.1 Paikalliset tietopisteet

Paikalliset tietopisteet ovat moduulin sisällä sen data-alueella yksikäsitteisesti nimettyjä tietoalueita. Näihin kytkeytymällä siirtävät toimilohkot ja portit tietoa toisilleen moduulien sisällä. Paikallinen tietopiste on tunnettu ainoastaan moduulin sisällä, ei muualla. Paikallista tietopistettä tarvitaan tietyissä moduulin sisäisissä kytkennöissä esim. calc- (laskentatoimilohko), logic- (logiikkatoimilohko) ja cmp- (vertailutoimilohko) toimilohkojen kytkemisissä toisiinsa.

Graafisessa suunnittelussa paikallinen tietopiste esitetään seuraavalla symbolilla (Kuva 7):



Kuva 7 Paikallisen tietopisteen symboli

## 4.1.2 Ulkoiset tietopisteet

Ulkoiset tietopisteet sijaitsevat samalla data-alueella kuin paikalliset tietopisteet. metsoDNA CR:n kommunikointirutiini kopioi tietoa näihin pisteisiin muiden moduulien tietopisteistä porttien välityksellä. Ulkoisen tietopisteen nimi viittaa nimeen, porttiin, joka on tunnettu metsoDNA CR:ssä.

Moduuli esittelee kaikki ulkopuoleltaan tulevat tiedot ulkoisina tietopisteinä. Kommunikointi hakee tiedon lähdemoduulin rajapinta- tai suorasaantiportista kohdemoduulin ulkoisiin tietopisteisiin tai päinvastoin ulkoisista tietopisteistä toisen moduulin portteihin.

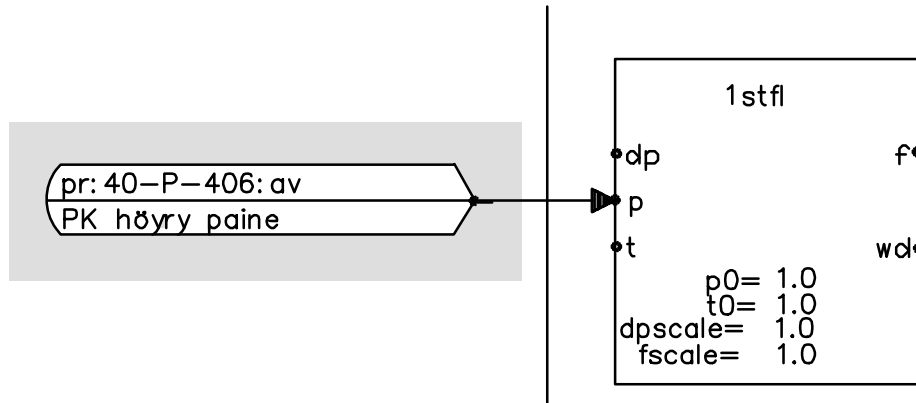
Ulkoinen tietopiste voi olla seuraavaa tyyppiä:

- tulo  
Tulo-tyyppinen ulkoinen tietopiste lukee tietoa muista moduuleista ja sen tiedot voidaan kytkeä moduulin sisällä toimilohkojen tuloihin.
- lähtö  
Lähtö-tyyppinen ulkoinen tietopiste kirjoittaa tietoa muihin moduuleihin porttien välityksellä ja se voidaan kytkeä moduulin sisällä toimilohkon lähtöön.
- tulolähtö  
Tulolähtö-tyyppinen ulkoinen tietopiste voi sekä lukea että kirjoittaa tietoa toimilohkoon ja porttien kautta toisiin moduuleihin. Tämän tyyppistä tiedonsiirtoa käytetään mm. reseptiohjauksessa.

Ulkoinen tietopiste voi kommunikoida porttien välityksellä muihin moduuleihin:

- jatkuvalla kommunikoinnilla  
Jatkovaa kommunikointia käytetään normaalisti eli tietopistettä konfiguroitaessa määrätään kommunikointiaajuus (sen tulee olla sama kuin siihen kytketyn moduulin suoritustaajuus).
- ehdollisella kommunikoinnilla  
Ehdollista kommunikointia käytetään, jos ei välttämättä jotain tietoa tarvitse päivittää jatkuvasti (esim. resepteissä ja ehdollisten kopiointitoimilohkojen yhteydessä).

Graafisessa suunnittelussa ulkoinen tietopiste esitetään seuraavan tyyppisellä symbolilla (Kuva 8 varjostettu alue).



Kuva 8 Ulkoisen tulon kytkeminen toimilohkoon

Kuvassa oleva jatkuva ulkoinen tulo pr:40-P-406:av on kytketty 1stfl-toimilohkon tuloon p. Toimilohko, johon ulkoinen tietopiste viittaa, on määriteltävä ko. moduulissa suorasaantiportiksi. Esimerkin ulkoinen tulo viittaa moduulissa pr:40-P-406.F olevan pr:40-P-406-nimisen suorasaantiportin kautta ko. porttiin kytketyn toimilohkon (am) jäsenen av.

Molemmille tietopistetyypeille (sekä ulkoiselle että paikalliselle) voidaan antaa alkuarvo ja molempiin voidaan kytkeytyä moduulin sisällä.

Ulkoiset tietopisteet nimetään sopivasti, esimerkiksi johtamalla nimet positiotunnuksista. Paikallisten tietopisteiden nimeämisessä voidaan käyttää jotain yksinkertaisempaa nimeämistä. Esim. graafisessa suunnittelussa käytetään paikallisen tietopisteen nimenä P-kirjainta ja tietopisteen identifioimiseen juoksevaa numerointia: esim. P1, P2, P3 jne.

Alkeistyyppiä olevat tietopisteet ovat jakamattomia; rakenteiset tietopisteet ja taulukkotietopisteet jakautuvat osiin, joihin pääsee käsiksi tarkentamalla tietopisteen tunnusta rakenteisen tyyppin jäsenellä tai indeksillä.

## 4.2 PORTIT

Portteja käytetään moduulin siirtopintojen luomiseen. Niiden välityksellä sovellusverkon eri moduulit kommunikoivat keskenään. Portit voivat olla *suorasaantiportteja* tai *rajapinta-portteja*, jotka nimetään sopivalla tunnuksella.

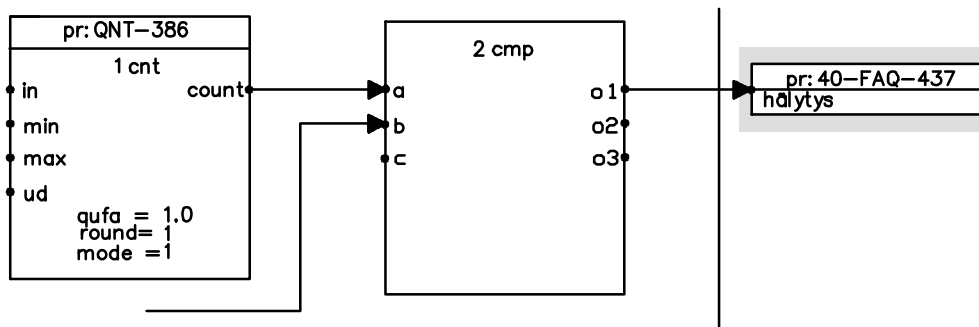
### 4.2.1 Suorasaantiportti

Suorasaantiportti on tunnettu koko sovellusverkossa eli sen nimi on yksikäsitteinen koko metsoDNA CR:n piirissä. Siihen kytkettyä tietoa voidaan kirjoittaa/lukea portin nimellä koko metsoDNA CR:ssä. Koska suorasaantiportin nimi on yksikäsitteinen koko sovellusverkossa, ei samaa nimeä saa esiintyä missään muualla sovellusverkossa.

Suorasaantiporttiin voidaan kytkeä joko yksittäinen tietopiste tai koko toimilohko. Yksittäiseen tietopisteeseen kytkeydyttäessä saadaan portin kautta vain ko. tietopisteen tieto moduulin ulkopuolelle. Koko toimilohkoon kytkettävää suorasaantiporttia käytettäessä saadaan kaikki ko. porttiin kytketyn toimilohkon jäsenet muiden moduulien saataville.

Seuraavassa kuvassa on esimerkki siitä, kuinka suorasaantiporttiin kytketään yksittäinen signaali (Kuva 9). Esimerkissä on suorasaantiportin (pr:40-FAQ-437) symboli tummenneulla alueella.

Esimerkissä olevasta kytkennästä saadaan toimilohkon 2cmp lähdön o1 tila tuotua portin välityksellä moduulin ulkopuolelle, jolloin sen tietoa voidaan käyttää missä muualla sovellusverkossa tahansa.

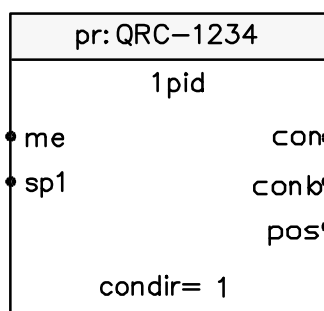


Kuva 9 Suorasaantiportin kytkeminen yksittäiseen signaaliin

Toinen tapa suorasaantiportin käyttöön on kytkeä (nimetä) kokonainen toimilohko suorasaantiporttiin. Näin saadaan kaikki toimilohkon jäsenten arvot muiden moduulien saataville (tulot/lähdöt/parametrit) pelkällä suorasaantiportin nimellä. Ensisijaisesti valvomo kysyy näyttötietoja prosessinohjauspalvelimelta suorasaantiporttia käyttäen. Lisäksi prosessinohjauspalvelimilla on muutamia toimilohkoja, joille lähes aina määritellään suorasaantiportti. Näitä ovat mm:

<b>am</b>	(analogiamittaus)
<b>pid</b>	(PID-säädin)
<b>mtr, mtre</b>	(moottorin ohjaus)
<b>mgv, mgve</b>	(magneettiventtiilin ohjaus)
<b>grp</b>	(ryhmäohjaus)
<b>qcnt</b>	(määrälaskuri)
<b>cnt</b>	(laskuri).

Seuraavassa kuvassa (Kuva 10) on koko toimilohkoon kytketyn suorasaantiportin graafinen symboli. Suorasaantiporttia kuvaa toimilohkon yläreunassa oleva suorakaide (varjostettu alue) ja sen nimeä suorakaiteen keskellä oleva nimi (pr:QRC-1234).



Kuva 10 Toimilohkoon liittyvän suorasaantiportin merkitseminen

Toimilohkon jäseniä voidaan esimerkin tapauksessa kysellä muista moduuleista nimillä esim. seuraavasti: pr:QRC-1234:me, pr:QRC-1234:con jne.

Automaatiokielen moduulirakenteisuus tukee suunnittelijaa antamalla mahdollisuuden oletusarvojen sekä aiemmin tehdyn työn hyväksikäyttöön. Oletusarvotoiminnot tapahtuvat kuitenkin suunnitteluympäristössä; kun tietoa siirretään sovelluspalvelimille, tiedosta ei voi enää päätellä, onko kyse oletusarvosta vai suunnittelijan syöttämästä arvosta.

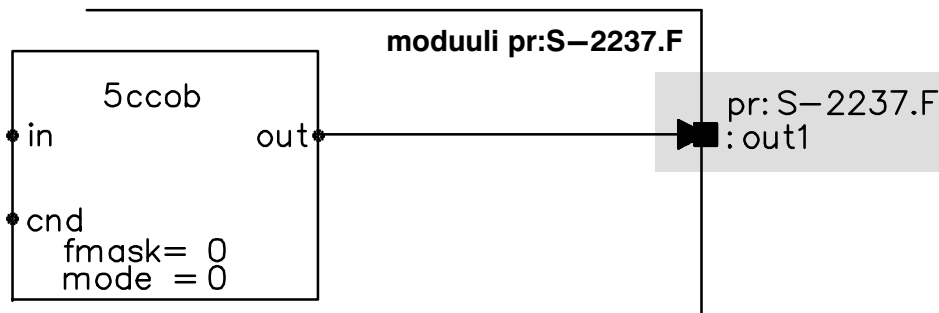


## 4.2.2 Rajapintaportti

Toimilohkon yksittäinen tieto voidaan kytkeä moduulin rajapintaporttiin. Tällä tavoin siirretään yksittäistä tietoa moduulista toiseen tai sovelluspalvelimelta toiselle.

Rajapintaporttiin kytkettyä tietoa voidaan kirjoittaa/lukea nimellä, joka muodostuu konfigurointimoduulin nimestä täydennettynä varsinaisella rajapintaportin nimellä.

Seuraavassa on esimerkki rajapintaportista (Kuva 11), jossa rajapintaportti on kytketty moduulin sisältä toimilohkon 5ccob lähtöön out. Tämän toimilohkon lähtö out on nyt saatavilla missä tahansa metsoDNA CR:ssä käyttämällä moduulin nimeä ja portin nimeä yhdessä seuraavasti pr:S-2237.F:out1. Tässä esimerkissä moduulin nimi on pr:S-2237.F ja portin nimi out1.



Kuva 11 Rajapintaportin esittäminen graafisesti

Portit kytketään moduulin sisällä, jolloin niiden arvot voidaan määrittää ja lukea porttien kautta. Toiminnallisesti suorasaanti- ja rajapintaportit ovat samanlaisia.

## 4.3 TOIMILOHKOT

Toimilohkot toteuttavat tietyn toiminnan, kuten esimerkiksi säätöalgoritmin ja liittyvät ympäristöönsä *kytkentäpisteiden* kautta. Toimilohkon kytkentäpisteitä ja konfigurointiparametreja nimitetään toimilohkon jäseniksi. Jäsenet ovat muiden automaatio-objektien tavoin tyypitettyjä.

Jos toimilohko kirjoittaa kytkentäpisteeseensä tietoa, on kyseessä lähtö. Jos se lukee tietoa tai sekä lukee että kirjoittaa, nimitään kytkentäpiste tuloksi.

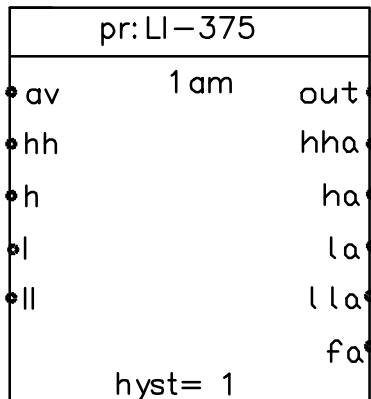
Toimilohkon toiminta määritellään konfigurointiparametreilla. Parametreille voi antaa vakioarvoja, mutta niitä ei voi kytkeä tietopisteisiin. Toimilohko sijoittaa sisäiset tilatietonsa omiin sisäisiin jäseniinsä. Niitä ei näytetä toimilohkon käyttäjälle eivätkä ne ole kytkettävissä. Toimilohkolla on tyyppinsä mukaiset ohjeet, joiden ohjaamana se käsittelee kytkentäpisteisiinsä kytkettyä tietoa.

Sekä tulot että lähdöt samoin kuin konfigurointiparametrit on määritelty tietyn tyyppiseksi, jolloin niihin voidaan kytkeä vain tietyn tyyppistä tietoa (tulot ja lähdöt) tai antaa vain tietyn tyyppisiä arvoja (konfigurointiparametrit).

Suunnittelijan tulee yksilöidä toimilohkot antamalla niille tunnus. Tunnus muodostetaan yksilönumerosta ja heti perään liitetystä toimilohkon tyyppitunnuksesta. Mahdollisia tunnuksia ovat esimerkiksi 1pid ja 99hys, joista ensimmäisen tunnusnumero on 1 ja tyyppi pid sekä toisen tunnusnumero 99 ja tyyppi hys.

Konfigurointimoduulin sisällä toimilohkot suoritetaan numerojärjestyksessä pienimmästä suurimpaan.

Seuraavassa kuvassa (Kuva 12) on eräs prosessinohjauspalvelimen toimilohko, jota käytetään analogiamittauksissa.



Kuva 12 Analogiamittaustoimilohkon symboli

Am-toimilohko sisältää:

- kaksi konfigurointiparametria, joista yksi parametri (hyst) näkyy symbolissa
- viisi tuloa
- kuusi lähtöä.

Konfigurointiparametrit merkitään graafisessa suunnittelussa toimilohkon sisään ja niillä määritellään yleensä toimilohkon toimintatapoja. am-toimilohkon konfigurointiparametritilla

- **hyst**  
määritellään toimilohkon hystereesin suuruus. Tämä jäsen on tyyppiä float.
- **un**  
insinööriyksikkö: kommenttitieto, joka ei vaikuta toimilohkon toimintaan

Toimilohko sisältää konfigurointiparametrien lisäksi mm. seuraavat tulot:

- **av**  
Tutkittavan analogiaviestin arvo
- **hh**  
Ylemmän ylähälytysrajan arvo

Molempiin tuloihin voidaan kytkeä jokin signaali tai vakioarvo. av-tulo on tyyppiä ana ja hh-tulo on tyyppiä float.

Lisäksi toimilohko sisältää mm. seuraavat lähdöt:

- **out**  
Operoitava tieto, joka viedään kentälle. out on tyyppiä ana.
- **hha**  
Ylempi ylärajahälytys, joka on tyyppiä bin.

Kaikilla tämän toimilohkon jäsenillä on alkuarvot, jotka ne saavat ennen kuin jokin tieto on päivittänyt niitä.

Am-toimilohko voidaan nyt kuvata seuraavalla tavalla (vastaavasti on toimilohkokirjoissa kuvattu kaikki metsoDNA CR:n toimilohkot.) Tässä esitysmuodossa ovat kaikki toimilohkon jäsenet, niiden tyytit sekä oletusarvot sekä kuvaus jokaisen jäsenen toiminnasta tai merkityksestä.

### 4.3.1 Konfiguroitavat tiedot

#### **hyst**

**Tyyppi:** float  
**Oletus:** 0.0  
**Kuvaus:** Hystereesi

Hälytys poistuu, kun analogiaviesti on palannut hälytysrajan sallitulle puolelle ja eroaa hälytysrajasta hystereesin verran.

#### **un**

**Tyyppi:** uns16  
**Oletus:** 0  
**Kuvaus:** Kommentti

Insinööriyksikkö; kommenttitieto, joka ei vaikuta toimilohkon toimintaan.

### 4.3.2 Kytkevät tiedot

#### **Tulot**

##### **av**

**Tyyppi:** ana  
**Oletus:** 0 0.0  
**Kuvaus:** Arvo

Tutkittavan analogiaviestin arvo.

##### **hh**

**Tyyppi:** float  
**Oletus:** 0.0  
**Kuvaus:** Ylempi yläraja (higher high limit)

Ylemmän ylähälytysrajan arvo. Mikäli hälytysrajat asetetaan ristiin (esim. hh < h), toimilohko palauttaa rajat oikeaan järjestykseen (h:n arvoksi kopioidaan hh).

##### **h**

**Tyyppi:** float  
**Oletus:** 0.0  
**Kuvaus:** Yläraja (high limit)

Ylähälytysrajan arvo. Rajojen tarkistus ks. jäsen hh.

##### **l**

**Tyyppi:** float  
**Oletus:** 0.0  
**Kuvaus:** Alaraja (low limit)

Alahälytysrajan arvo. Rajojen tarkistus ks. jäsen hh.

##### **ll**

**Tyyppi:** float  
**Oletus:** 0.0  
**Kuvaus:** Alempi alaraja (lower low limit)

Alemman alahälytysrajan arvo. Rajojen tarkistus ks. jäsen hh.

## Lähdöt

### out

**Tyyppi:** ana  
**Oletus:** 48 0.0  
**Kuvaus:** Lähtö (output)

Operoitava tieto, joka viedään kentälle. Saa initialisoitaessa alkuarvon out:a = av:a, jos tuloviestin vikabitit eivät ole asettuneet.

### hha

**Tyyppi:** bin  
**Oletus:** 48  
**Kuvaus:** Ylempi ylärajahälytys (higher high limit alarm)

### ha

**Tyyppi:** bin  
**Oletus:** 48  
**Kuvaus:** Ylärajahälytys (high limit alarm)

### la

**Tyyppi:** bin  
**Oletus:** 48  
**Kuvaus:** Alarajahälytys (low limit alarm)

### lla

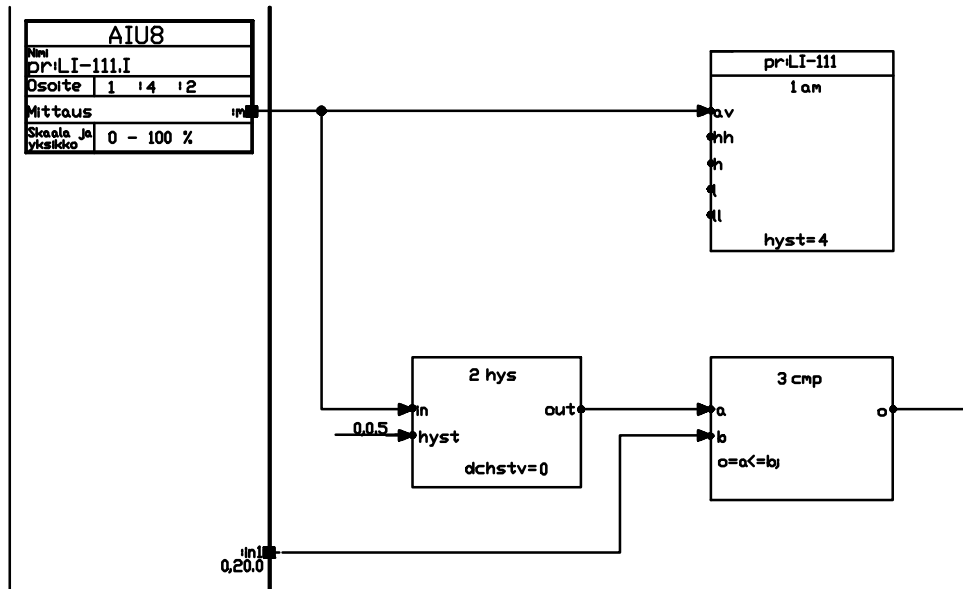
**Tyyppi:** bin  
**Oletus:** 48  
**Kuvaus:** Alempi alarajahälytys (lower low limit alarm)

### fa

**Tyyppi:** bin  
**Oletus:** 48  
**Kuvaus:** Tulosignaalihälytys (input alarm)

Tulosignaalin vikabitit ovat asettuneet.

Seuraavassa kuvassa (Kuva 13) on esimerkki am-toimilohkon kytkemisestä muihin toimilohkoihin.



Kuva 13 Esimerkki toimilohkoista ja niiden kytketyistä

Toimilohkon graafisen esityksen jälkeen on vastaava kytkentä am-toimilohkon osalta listamuodossa. (Esimerkissä on esitelty listaus vain am-toimilohkon osalta.)

```

1am ON pr:LI-111
hyst=4
un= -
av< pr:LI-111.I:m
hh< -
h< -
l< -
ll< -
out> -
hha> -
ha> -
la> -
lla> -
fa> -
;
    
```

## **5 AUTOMAATIOKIELINEN SOVELLUSOHJELMA**

Automaatiokielineen sovellusohjelma koostuu sovellussuunnittelun ja prosessinohjauksen kannalta mielekkäistä kokonaisuuksista, konfigurointimoduuleista. Moduulit koostuvat tietopisteistä, porteista, toimilohkoista sekä algoritmisistä lausekelajennuksista laskennalle, logiikalle ja vertailuille. Konfigurointimoduulit kommunikoivat keskenään porteille annettujen metsoDNA CR:n tiedonhallinnan tuntemien nimien perusteella.

Automaatiomoduulit ovat sovellusohjelmiston graafisia esityksiä. Ne voivat koostua useista konfigurointimoduuleista. Automaatiomoduuleihin voidaan liittää tärkeimmät yhteen säätöpiiriin liittyvät konfigurointimoduulit mm:

- prosessinohjauspalvelimen toimintamoduuli ja I/O-moduulit
- kaaviolamppujen ohjauksessa käytettävät moduulit
- valvomon positio-, operointi- ja tapahtumamoduulit.

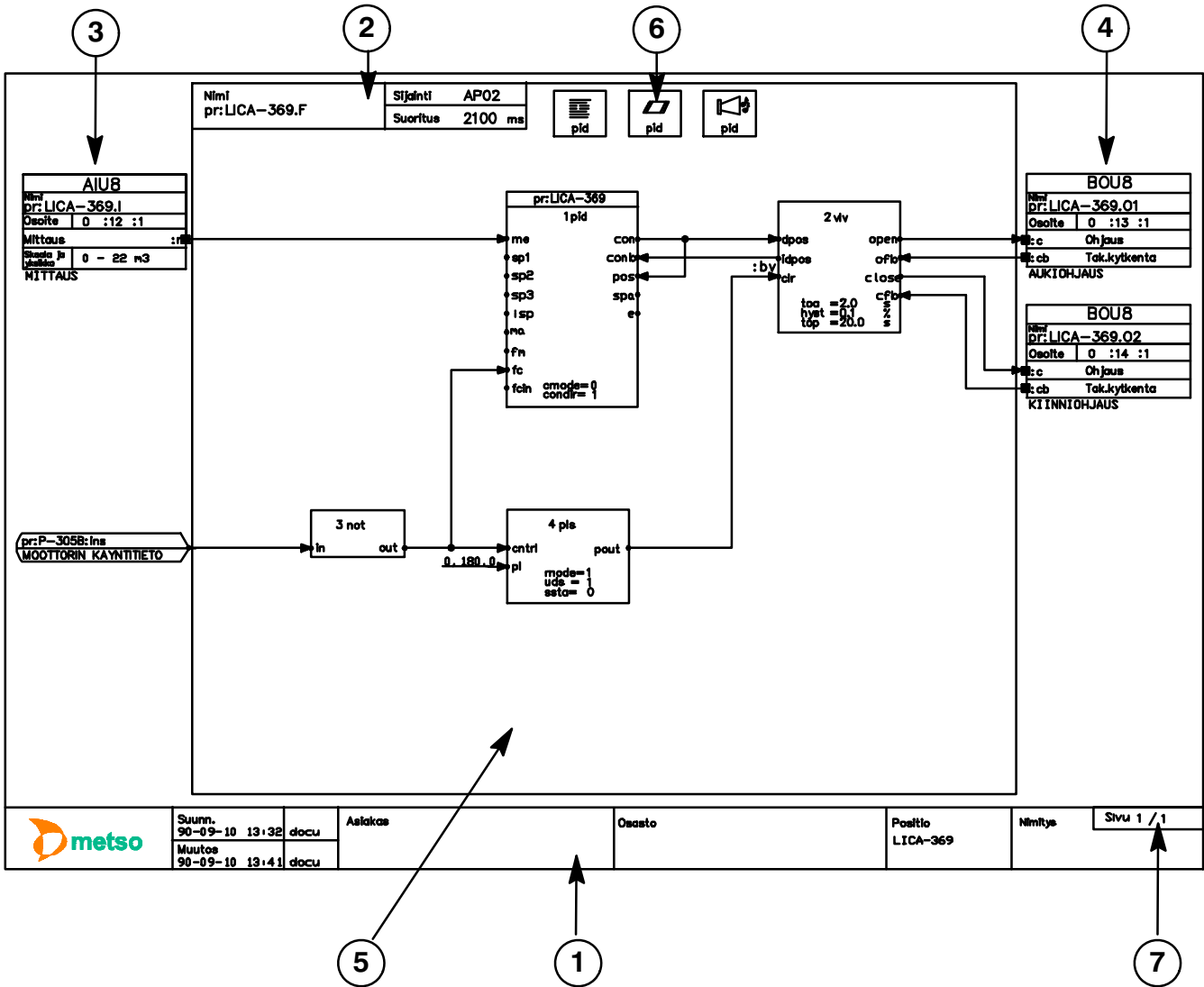
Automaatiomoduuleissa on konfigurointimoduulien osille annettu havainnolliset symbolit sekä lisäksi eri signaalityypit on koodattu eri väreillä helpottamaan kytkentöjen tekemistä.

### **5.1 AUTOMAATIOMODUULIN RAKENNE**

Automaatiomoduuli koostuu seuraavista osista.

- automaatiomoduulin hallintaosasta (1)
- toimintamoduulin hallintaosasta (2)
- ulkoisten tulojen ja tulomoduulien kytkentäkentästä (3)
- ulkoisten lähtöjen ja lähtömoduulien kytkentäkentästä (4)
- toimilohkojen kytkentäkentästä (5)
- automaatiomoduuliin liittyvien valvomo- ja piirturinohjausmoduulien osasta (6)
- sivu (7)

Seuraavassa kuvassa (Kuva 14) on esitetty eräs automaatiomoduli ja siihen on merkitty numeroin edellisen listan mukaiset moduulin osat.



Kuva 14 Automaatiomoduli ja sen osat

## 5.2 KONFIGUROINTIMODUULIN RAKENNE

Automaatiokielen konfigurointimoduuli sisältää lausekielisessä muodossa kolme osaa:

- hallintaosa
- esittelyosa
- toimintaosa.

### 5.2.1 Hallintaosa

Moduulin hallintaosassa on moduulin ja tietokannan hallinnassa käytettäviä tietoja.

Seuraavassa on esimerkki moduulin hallintaosasta listamuotoisessa esityksessä:

```
HALLINTAOSA
NIMI:      pr:89LIC-2305.F
TYYPPI:    toiminta
TILA:      kesken
LUOJA:     timo
           LUOTU:      89-08-23 15:34
MUUTTAJA: timo
           MUUTETTU:  89-08-23 16:14
SIJAINTI:  AP02
SUORITUS:  400
MONESKO:   3
KVAUS:
```

Konfigurointimoduulin hallintaosa esitetään kaavakkeena, jossa on seuraavat tiedot:

#### NIMI

- moduulin yksikäsitteinen nimi koostuu komponenteista, jotka erotellaan toisistaan kaksoispisteillä.  
Moduulien nimeyskäytäntö on esitetty luvussa 6 ”Automaatiokielen nimeyskäytäntö”.

#### TYYPPI

- moduulin tyyppi on jokin seuraavista:
  - **toiminta**  
toimintamoduulit
  - **io**  
I/O-liityntämoduulit
  - **positio**  
positiotekstien määrittelyihin käytettävät moduulit
  - **yliö**  
monitorin yliön tekstien ja kellonajan muodostamiseen käytettävät moduulit
  - **operointi**  
operointi-ikkunan muodostamiseen käytettävät moduulit
  - **aktiiviteetti**  
tapahtumakirjoittimen ja monitorien määrittelymoduulit
  - **tapahtuma**  
tapahtumalokin ja aluehälytystunnusten muodostamiseen käytettävät moduulit
  - **rekisteröinti**  
hälytyslistan, tapahtumalokin sekä yliön aluetunnusten muodostamiseen sekä hälytystorven ohjaukseen ja tapahtumakirjoittimen ohjaukseen käytettävät moduulit



- **näppäimistö**  
suoravalintapainikkeiden määrittelyssä käytettävät moduulit
- **kello**  
kesä- ja talviaikaan siirtymisen määrittelevät moduulit
- **dio**  
Damatic-liityntäpalvelimen I/O-liitynnän määrittelyssä käytettävät moduulit
- **kuva**  
monitorin kuvien muodostamiseen käytettävät moduulit
- **hierarkia**  
kuvahierarkian muodostamisessa käytettävät moduulit
- **polku**  
hierarkiapolun muodostamisessa käytettävät moduulit
- **paletti**  
monitoreilla ja kuvakopioinnissa käytettävien värien määrittelymoduulit
- **valikko**  
hierarkiavalikon operointi-ikkunaan muodostavat moduulit
- **sekvenssi**  
sekvenssiohjelmien muodostamiseen käytettävät moduulit
- **kortti**  
PLU- ja MCP-moduuli

#### **TILA**

- moduulin tila voi olla jokin seuraavista:
  - kesken
  - valmis
  - testattu

#### **LUOJA**

- henkilö, joka on luonut moduulin

#### **LUOTU**

- luontipäivämäärä ja kellonaika muodossa yy-mm-dd hh:ss

#### **MUUTTAJA**

- henkilö, joka on tehnyt viimeisimmän muutoksen moduuliin

#### **MUUTETTU**

- viimeisimmän muutoksen muutospäivämäärä ja kellonaika muodossa yy-mm-dd hh:ss

LUOJA-, LUOTU-, MUUTTAJA- ja MUUTETTU-kentät päivittyvät moduuliin automaattisesti sitä luotaessa ja muutettaessa.

#### **SIJAINTI**

- moduulin paketti  
Tarkemmat tiedot kts. luku 6 "Automaatiokielen nimeyskäytäntö".

#### **SUORITUS**

- moduulin suoritusjakso millisekunteina välillä 200 ms...64000 ms, 100 ms:n askelein

## **MONESKO**

- etumerkitön kokonaisluku, jolla määritellään moduulin suoritusjärjestys samalla aikavälillä suoritettavien moduulien ryhmästä. Moduulit suoritetaan kontrollitehtävässä MONESKO–kentän perusteella nousevassa järjestyksessä, eli sellaisten moduulien jälkeen, joissa MONESKO–kentän arvo on 0.

Jos useammalla kuin yhdellä moduulilla on sama MONESKO–kentän arvo, esim. 0, nämä moduulit suoritetaan moduulinimen perusteella aakkosellisessa järjestyksessä.

Toisin sanoen moduulien suoritusjärjestys on seuraava:

Ensin suoritetaan samalla suoritusvälillä olevat moduulit, joiden MONESKO–kentän arvo on nolla. Jos sellaisia moduuleita on useita, ne suoritetaan aakkosjärjestyksessä. Seuraavaksi suoritetaan sellaiset samalla suoritusvälillä olevat moduulit, joiden MONESKO–kentän arvo on 1 jne...

## **KUVAUS**

- suunnittelijan kuvaus moduulista. Kuvaustieto on laitettava konfigurointimoduuleissa lainausmerkkeihin (" "). Automaatiomoduuleissa KUVAUS–tietoa ei laiteta lainausmerkkeihin. Kentän voi jättää myös tyhjäksi.

Moduulin tila-, luonti- ja muutostietoja käytetään moduulin versionhallinnassa ja projektin-seurannassa. Kuvaustiedoilla kommentoidaan moduulin sijoitusta ja toimintaa.

## **5.2.2 Esittelyosa**

Konfigurointimoduulin esittelyosassa määritellään seuraavat kokonaisuudet:

- Ulkoiset tietopisteet
  - moduuliin kytkettävät ulkoiset tiedot
- Paikalliset tietopisteet
  - moduulin nimetyt paikalliset eli sisäiset tiedot
- Grafiikan määrittely mm. kuvamoduuleille
  - Grafiikan lausekielinen muoto kytketään kuvamoduuliin GRAFIIKKA–määrittelyiden jälkeen.
- Suorasaantiportit
  - Moduulin suorasaantiportit kytkeytyvät yksittäisiin tietopisteisiin tai viittaavat johonkin toimilohkoon kokonaisuudessaan.
  - Muut moduulit saavat tietoja moduulista suorasaantiportin kautta.
- Rajapintaportit
  - Moduulin rajapintaportit kytkeytyvät yksittäisiin tietopisteisiin.
  - Muut moduulit saavat tietoja moduulista rajapintaportin kautta.

Seuraavaan on koottu yhtenäinen konfigurointimoduulin esittelyosa seuraavien esimerkkien pohjalta:

## ESITTELYOSA

## ULKOISET

```
pr:SC-265.I:m TYYPPI ana SIIRTO 192,10,0,0 ;
pr:SC-265.oI:m TYYPPI ana SIIRTO 192,10,0,0 "aukiraja";
pr:SC-265.cI:m TYYPPI ana SIIRTO 192,10,0,0 "kiinniraja";
pr:SC-265.oO:c TYYPPI bo SIIRTO 65,10,0,0 "kiinniohjaus";
pr:SC-265.oO:cb TYYPPI bin SIIRTO 192,10,0,0 ;
pr:SC-265.cO:c TYYPPI bo SIIRTO 65,10,0,0 "aukiohjaus" ;
pr:SC-265.cO:cb TYYPPI bin SIIRTO 192,10,0,0 ;
```

## PAIKALLISET

```
P1 TYYPPI ana ;
```

## SUORASAANTI

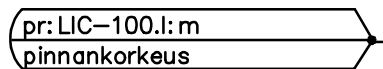
```
LOHKO pr:SC-265 ;
```

## RAJAPINTA

```
MODSTAT TYYPPI ktstat "moduulin tilatiedot" < (1,1,0,1,1) ;
```

## Ulkoiset tietopisteet

Graafisessa suunnittelussa ulkoisesta tietopisteestä käytetään mm. seuraavaa symbolia:



### Käyttö:

Jos moduuli haluaa toisesta moduulista signaalin, se esittelee ulkoisen tulon, joka on ko. signaalin tyyppiä. Yleensä ulkoisen tulon kommunikointitapa on jatkuva (EXT IN CONTINUOUS).

Erityyppisille ulkoisille tietopisteille käytetään graafisessa suunnittelussa erilaisia symboleita. Symbolin väri määrää ulkoisen tietopisteen tyyppin ja sen muoto kommunikointitavan.

Ulkoiset tietopisteet esitellään konfigurointimoduulin hallintaosassa avainsanan ULKOISET jälkeen.

Ulkoisen tietopisteen esittelyyn liittyy:

- **ulkoisen tietopisteen nimi**

Ulkoisen tietopisteen nimi eli nimi, jolla tietoa haetaan tai viedään määräytyy sen moduulin portin mukaan, mistä tieto haetaan. Nimi toimii sellaisenaan ohjeena kommunikoinnille halutun tiedon löytämiseksi.

Seuraavassa on esimerkkinä ulkoisen tulon nimi, joka muodostuu tulomoduulin rajapintaportin mukaan. Esimerkissä moduulin nimi on pr:LIC-100.I ja sen rajapintaportin tarkennin on "m". Nimen komponentit erotellaan automaatiokielen tapaan kaksoispisteillä eli ulkoisen tietopiste on silloin

```
pr:LIC-100.I:m TYYPPI ana SIIRTO 192,4,0,0 "pinnankorkeus";
```

- **tiedon tyyppi**

Ulkoisen tietopisteen varaa tyyppinsä mukaisen tietoalueen; esimerkiksi analogia (ana) –tyyppisen alueen.

Graafisessa suunnittelussa tietopisteen väri määrää tiedon tyyppin.

Listamuodossa tyyppi merkitään tietopisteen määrittelyyn avainsanan TYYPPI jälkeen. Seuraavassa on esimerkki ana-tyyppisen ulkoisen tietopisteen tyyppin merkitsemisestä:

```
pr:LIC-100.I:m TYYPPI ana SIIRTO 192,4,0,0 "pinnankorkeus";
```

- **mahdollinen tiedon alkuarvo**

Tietopiste voidaan alkuarvottaa, jolloin sillä on jokin arvo, vaikka kommunikointi ei olisi sitä vielä päivittänyt.

Alkuarvottaminen suoritetaan kirjoittamalla tyyppin koodin jälkeen ”=”-merkki ja antamalla tyyppille soveliaat alkuarvot suluisa. Esimerkiksi:

```
pr: LIC-100.l: m TYYPPI ana=(0,0.5) SIIRTO 192,4,0,0 "pinnankorkeus";
```

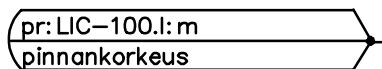
Graafisessa suunnittelussa ei sulkuja käytetä alkuarvotettaessa tietopistettä.

Jos alkuarvoa ei anneta, tietopiste saa tyyppinsä mukaisen oletusarvon moduulin siirtyessä sovelluspalvelimelle. Alkuarvo pysyy vakiona, ellei mikään muuta sitä kytkennän tai kommunikoinnin kautta.

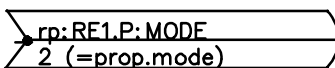
- **tiedonsiirtotavan määrittäminen**

Eri kommunikointitarpeeseen on määriteltävissä eri tyyppisiä ulkoisia tietopisteitä. Tiedonsiirtotavalla tarkoitetaan sitä, onko ulkoinen tietopiste tulo vai lähtö vai molempia ja sitä, kuinka se kommunikoi muiden kanssa, jatkuvasti vai vain silloin, kun tieto on muuttunut.

**Jatkuva ulkoinen tulo**

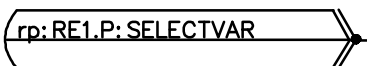


**Jatkuva ulkoinen lähtö**

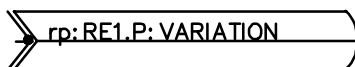


Yleisin kommunikointitapa on jatkuva kommunikointi. Tällöin tietoa siirretään jatkuvasti tietopisteiden välillä. Jos siirrettävää tietoa on paljon ja tiedon päivitystä ei tarvita jatkuvasti, voidaan käyttää myös ehdollista kommunikointia.

**Ehdollinen ulkoinen tulo**



**Ehdollinen ulkoinen lähtö**



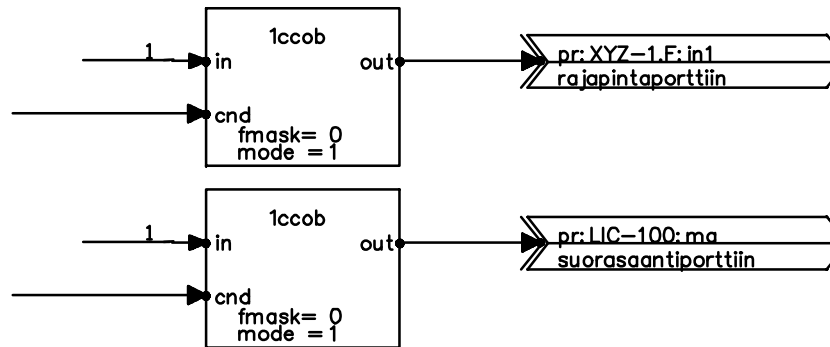
Ehdollisen ulkoisen lähdön käyttö on mahdollista vain ehdollisten kopiointitoimilohkojen (ccoX-toimilohkojen) kanssa.

Ehdollinen lähtö suoritetaan aina, kun siihen kytketty ccoX-toimilohkon ehto on siinä tilassa, että se kirjoittaa jotain lähtöönsä.

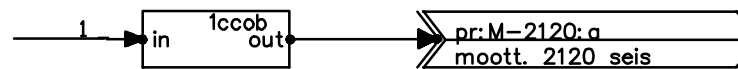
Ehdollista lähtöä käytetään varsinkin sekvensseissä ja resepteissä, sillä näissä ei yleensä ole tarvetta jatkuvaan kommunikointiin, vaan toiminta saadaan aikaan vähemmän verkkoa kuormittavalla ehdollisella kommunikoinnilla.

**HUOM!**

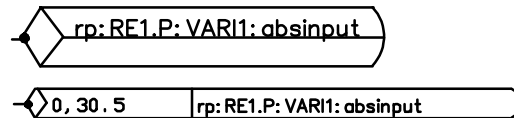
Ehdollisen ulkoisen lähdön käyttö on mahdollista vain ccoX-toimilohkojen kanssa.

**HUOM!**

Ehdollista ulkoista lähtöä ei saa käyttää jatkuvan lähdön tapaan, jos moduuli on jatkuvaasti suoritusessa (esim. sekvenssin askel), sillä se kuormittaa liikennettä liikaa.

**Jatkuva tulo ja lähtö**

Jos kommunikointitapana on jatkuva tulo ja lähtö, on kommunikointi mahdollista molempiin suuntiin tietopisteiden välillä. Tällaista kommunikointitapaa käytetään mm. resepteissä muunnelmamoduuleiden ja parametrimoduuleiden välillä.



Tiedonsiirto ilmaistaan lausekielisessä suunnittelussa neljän luvun ryhmällä, joka koodataan seuraavasti:

Ensimmäinen luku osoittaa tiedonsiirron suunnan

- 128 = haetaan tieto (tulo)
- 1 = viedään tieto (lähtö)

ja tiedonsiirron luonteen

- +64 = jatkuva
- +32 = ehdollinen
- +16 = tapahtuma
- +2 = suora osoitus

Esimerkiksi jos tietoa haetaan jatkuvasti, ensimmäinen luku muodostetaan seuraavasti:  $128 + 64 = 192$ .

Toinen luku määrittää tiedonsiirtovälin satoina millisekunteina. Luku voi saada arvoja 0...255. Luvun ollessa 0 käytetään kertalähetystä. Luvun ollessa 1 on päivitysväli 100 ms ja luvun maksimiarvolla 255 tiedonsiirtoväli on siis 25.5 sekuntia.

Kolmas luku on nolla.

Neljäs luku ilmoittaa tiedonsiirron päivitystavan

- 7 = tapahtuma 1 -> 0 tai 0 -> 1
- 6 = tapahtuma 0 -> 1
- 5 = tapahtuma 1 -> 0

Jos tietoa siirretään jatkuvasti, neljännen luvun arvolla ei ole merkitystä; se voi olla esimerkiksi nolla.

Seuraavassa on edelläkin ollutta signaalia laajennettu päivitystavalla:

pr:LIC-100.l:m TYYPPI ana **SIIRTO 192,4,0,0** "pinnankorkeus";

Graafisessa suunnittelussa käytetty symboli määrää tiedonsiirtotavan.

- **mahdollinen kommentti**

Kommenttia ei tarvitse välttämättä antaa. Kommentti kirjoitetaan ""-merkkien väliin listamuotoisessa suunnittelussa. Graafisessa suunnittelussa ei ""-merkkejä käytetä.

pr:LIC-100.l:m TYYPPI ana SIIRTO 192,4,0,0 **"pinnankorkeus"**;

- **tietopisteen määrittelyn päättää puolipiste (;)**

## Paikalliset tietopisteet

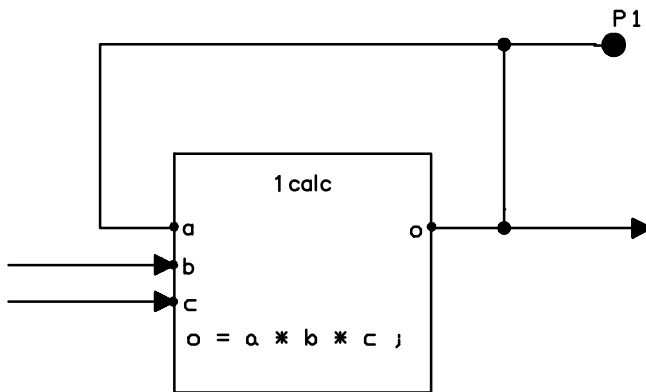
Graafisessa suunnittelussa paikallisesta tietopisteestä käytetään seuraavaa symbolia:



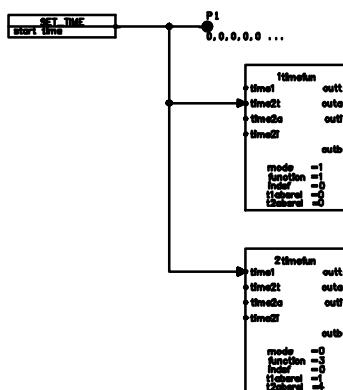
### Käyttö:

Paikallisen tietopisteen käyttötarvetta on melko harvoin graafisessa suunnittelussa. CAD-työkalu tekee useimmissa tapauksissa itse paikalliset tietopisteet kuvasta generoitavaan konfigurointimoduuliin. (Paikalliset tietopisteet eivät näy graafisessa kuvassa, vaan vain konfigurointimoduulissa.)

Paikallista tietopistettä voidaan kuitenkin käyttää esim. signaalin ”lenkityksessä”



ja alkuarvotuksessa:



Paikalliset tietopisteet esitellään konfigurointimoduulin ESITTELYOSAssa avainsanan PAIKALLISET jälkeen.

Paikallisen tietopisteen esittelyyn liittyy:

- **nimi**  
Paikallisen tietopisteen nimen on alettava kirjaimella. Nimen jatko voi sisältää sekä kirjaimia, numeroita, pisteen (.) että alaviivan (\_). Esimerkki paikallisen tietopisteen nimestä: **P1**
- **mahdollinen dimensio**  
Jos paikallinen tietopiste on taulukkotyyppiä, sille määritellään myös ulotteisuus ja käytettävien indeksien ääriarvot.  
Dimensio määritellään kaarisulkujen väliin annettavilla kahdella kokonaisluvulla. Esimerkissä määrä on 2-ulotteinen taulukko, jonka indeksien ääriarvot olivat 2 ja 3. Paikallinen tietopiste varaa tyyppinsä ja ulotteisuutensa mukaisen tietoalueen moduulin data-alueelta. Esimerkki paikallisen taulukko-tyyppisen tietopisteen nimestä:  
määrä **(2,3)** TYYPPI float=(0,2.1,0,5.0,1,40.86) "2-ulotteinen taulukko";  
Graafisessa suunnittelussa dimensio annetaan paljaana lukuna ilman kaarisulkuja.
- **tiedon tyyppi**  
Paikallinen tietopiste varaa tyyppinsä mukaisen tietoalueen; esimerkiksi float-tyyppisen alueen. Tyyppi merkitään tietopisteen määrittelyyn avainsanan TYYPPI jälkeen. Seuraavassa on esimerkki float-tyyppisen ulkoisen tietopisteen tyyppin merkitsemisestä:  
P1 TYYPPI ana=(0,2.1) "paikallinen tietopiste";  
Graafisessa suunnittelussa tietopisteen väri määrää tietopisteen tyyppin.
- **mahdollinen tiedon alkuarvo**  
Tietopiste voidaan alkuarvottaa, jolloin sillä on jokin arvo, vaikka kommunikointi ei olisi sitä vielä päivittänyt. Alkuarvottaminen suoritetaan kirjoittamalla tyyppin koodin jälkeen "="-merkki ja antamalla tyyppille soveliaat alkuarvot suluissa. Esimerkiksi:  
P1 TYYPPI ana=**(0,2.1)** "paikallinen tietopiste";  
Graafisessa suunnittelussa alkuarvo annetaan paljaana lukuna ilman sulkuja ja "="-merkkiä.  
Alkuarvoa ei tarvitse välttämättä antaa. Jos alkuarvoja ei syötetä, tietopiste saa tyyppinsä mukaiset oletusarvot moduulin siirtyessä sovelluspalvelimelle. Alkuarvot pysyvät vakiona, ellei mikään muuta niitä kytkennän tai kommunikoinnin kautta.
- **mahdollinen kommentti**  
Kommenttia ei tarvitse välttämättä antaa. Kommentti kirjoitetaan ""-merkkien väliin seuraavasti:  
P1 TYYPPI ana=(0,2.1) "**paikallinen tietopiste**";  
Graafisessa suunnittelussa ei kommenttia laiteta lainausmerkkeihin.
- **tietopisteen määrittelyn päättää puolipiste (;)**

## Grafiikan liittäminen moduuliin

Graafisessa suunnittelussa ei käyttäjän tarvitse määritellä kuvaa listamuodossa, vaan generointiohjelmat luovat listauksen kuvasta automaattisesti.

Lausekielisessä grafiikan esityksessä esitetään piirtokomennot GRAFIIKKA-sanan jälkeen tulevassa osuudessa. Ennen piirtokomentoja annetaan määriteltävän kuvan nimi KUVAILE KUVA -tekstin jälkeen.

Esittelyosassa esitelty kuvatausta kytketään moduuliin toimilohkon kautta esittelyosan nimen avulla.

Seuraavassa on esimerkki lausekielisen grafiikan liittämistä kuvamoduuliin:

```

ESITTELYOSA
  ULKOISET
  .
  .
  .
  GRAFIIKKA
    KUVAILE KUVA kuva1
      VIIVAPAKSUUS 1 1
      VÄRI 0
      LIIKU A 0 638
      VIIVA S 300 20
      .
      .
    LOPETA KUVA
  ;

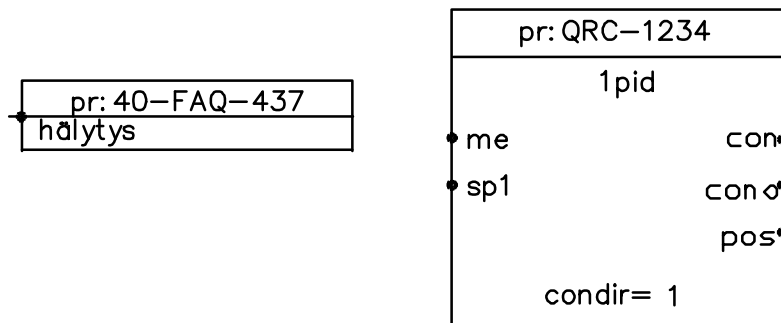
TOIMINTAOSA
  .
  .
  1draw
    nap < kuva1
  ;
  .
  .
LOPPU

```

Automaatiokielen lausemuotoisen grafiikan komennoista ja sen käytöstä on lisää tietoa liitteessä 3.

## Suorasaantiportit

Graafisessa suunnittelussa suorasaantiporteista käytetään seuraavia symboleita:



yhteen pisteeseen  
kytkeytyvä suora-  
saantiportti

kokonaiseen toimilohkoon  
kytkeytyvä suorasaantiportti



Suorasaantiportit esitellään konfigurointimoduulin ESITTELYOSAssa avainsanan SUORASAANTI jälkeen.

Suorasaantiportin esittelyyn liittyy:

- **portin nimi**  
Portin nimenä käytetään yleensä moduulin nimeen viittaavaa nimeä kuitenkin ilman moduulin tyyppitarkenninta.  
Esimerkki:  
`pr:XI-101 TYYPPI ana "kommentti" < (0,100.0);`
- **portin tyyppi**  
Portin tyyppiksi annetaan porttiin kytkettävän signaalin tyyppi. Koko toimilohkoon kytkettävälle portille ei määritellä tyyppiä. Tällöin portin määrittely alkaa sanalla LOHKO, jonka jälkeen tulee pelkkä portin nimi.  
Esimerkki:  
`pr:XI-101 TYYPPI ana "kommentti" < (0,100.0);`
- **mahdollinen kommentti**  
Porttiin on mahdollista liittää kommenttitekstiä, joka on erotettu muusta portin määrittelystä lainausmerkeillä (""").  
Graafisessa suunnittelussa ei kommenttitekstissä käytetä lainausmerkkejä.  
Esimerkki:  
`pr:XI-101 TYYPPI ana "kommentti" < (0,100.0);`
- **mahdollinen portin kytkentä** joko paikalliseen tai ulkoiseen tietopisteeseen tai portin viittaus toimilohkon jäseneseen  
Jos portti kytketään joko paikalliseen- tai ulkoiseen tietopisteeseen tai toimilohkon jäseneseen, merkitään portin kytkentä sen loppuun. Kytkentä määritellään "<"-merkillä.  
Esimerkki:  
`pr:XI-101 TYYPPI ana "kommentti" < (0,100.0);`  
Koko toimilohkoon kytkettävälle portille määrittely alkaa sanalla LOHKO, jonka jälkeen tulee pelkkä portin nimi.
- **suorasaantiportin määrittelyn päättää puolipiste (;)**

Seuraavassa on suorasaantiportin esittely kokonaisuudessaan:

```

ESITTELYOSA
.
.
SUORASAANTI
  pr:XI-101 TYYPPI ana < (0,100.0);
  LOHKO pr:LI-100;
.
.
TOIMINTAOSA
  lpid ON pr:LI-100
  pu= -

```

Esimerkeissä ensimmäinen suorasaantiportti pr:XI-101 on kytketty 2-ulotteisen taulukon määrä ensimmäiseen alkioon. Merkki "<" osoittaa kytkentää.

Suorasaantiportti voidaan esitellä myös lyhyemmin:

```
pr:XI-101 TYYPPI ana< - ;
```

Tällöin porttiin kytkettävä signaali on määritelty kytkettävässä signaalissa.

Toinen suorasaantiportti pr:LI-100 on esitelty koko toimilohkoon viittaavaksi portiksi. Toimintaosassa (lpid ON pr:LI-100) on määritelty, mihin toimilohkoon portti viittaa. Kaikki lpid-toimilohkon jäsenet on tuotu näin moduulin suorasaantiportin kautta muille moduuleille näkyviksi.

**Käyttö:**

Koko toimilohkoon kytkettävä suorasaantiportti voidaan kytkeä seuraaviin toimilohkoihin:

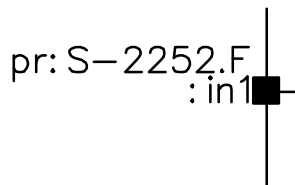
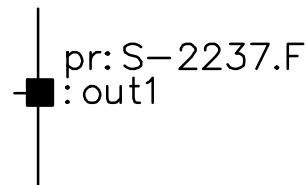
- am
- pid
- qcnt
- cnt
- mtr
- mtre
- mgv
- mgve
- grp.

Yleensä suorasaantiportiin kytketään kaikki ne piirit/signaalit, jotka näytetään valvomossa.

Tyypillisesti ”lähdemoduulissa” on suorasaantiportti ja kohdemoduulit esittelevät ulkoisen tulon, jossa pyydetään ao. portista haluttu signaali.

**Rajapintaportit**

Graafisessa suunnittelussa rajapintaportista käytetään seuraavaa symbolia:

**tulo-rajapintaportti****lähtö-rajapintaportti****Käyttö:**

Signaalit, joita halutaan siirtää muihin moduuleihin määritellään rajapintaporteiksi. Tiedon tarvitsija esittelee ulkoisen tulon, jossa pyydetään ko. signaali (pr:NIMI.F:rajapintaportin tarkennin).

Sovellusohjelman listauksessa (sen ESITTELYOSAssa) rajapintaportit esitellään avainsanan RAJAPINTA jälkeen.

Rajapintaportin esittelyyn liittyy:

- **portin nimi**  
Portin nimeyksessä on normaalin automaatiokielen rajoitukset. Yleensä porttia nimeydessä käytetään toimintamoduulien lähtöporteille out- nimiä, jotka on identifioitu numeroilla, esim. out1, out2 jne. Toimintamoduulien tuloporteille käytetään in- nimeä ja sen perässä numeroa. esim. in1, in2 jne. (Graafisessa suunnittelussa portin nimen eteen laitetaan kaksoispiste.)

Rajapintaporttien nimeyksessä voidaan käyttää myös moduulien nimeyksistä johdettuja nimiä.

**out1** TYPPI bin ”kommentti” < P1;

Porttien nimeyksissä suositellaan seuraavia nimiä:

Tulomoduuleissa:

m = mittaustieto

Lähtömoduuleissa:

c = ohjaus

cb = takaisinkytkentä

Esimerkkejä tarkentimien merkitsemisistä:

```
pr:LIC-100.I:m (mittaustieto)
pr:50-HS-115:c (ohjaus)
pr:50-P-125:cb (takaisinkytkentä)
```

Graafisessa suunnittelussa porttien nimeys on ”automaattista”.

- **portin tyyppi**

Portin tyyppiä annetaan porttiin kytkettävän signaalin tyyppi.

Graafisessa suunnittelussa symbolin väri määrää portin tyyppin.

```
out1 TYYPPI bin ”kommentti” < P1;
```

- **mahdollinen kommentti**

Porttiin on mahdollista liittää kommenttitekstiä, joka on erotettu muusta portin määrittelystä lainausmerkeillä (”).

Graafisessa suunnittelussa ei kommenttitekstissä käytetä lainausmerkkejä.

```
out1 TYYPPI bin ”kommentti” < P1;
```

- **mahdollinen portin kytkentä** joko paikalliseen tai ulkoiseen tietopisteeseen tai toimilohkoon tai vakioon

```
out1 TYYPPI bin ”kommentti” < P1;
```

- **rajapintaportin määrittelyn päättää puolipiste (;)**

```
out1 TYYPPI bin ”kommentti” < 5ccob:out;
```

Edellä esitellyissä esimerkeissä rajapintaportti out1 on kytketty moduulissa olevan 5ccob-toimilohkon lähtöön out.

Esittely voidaan tehdä myös lyhyemmin:

```
out1 TYYPPI bin < -;
```

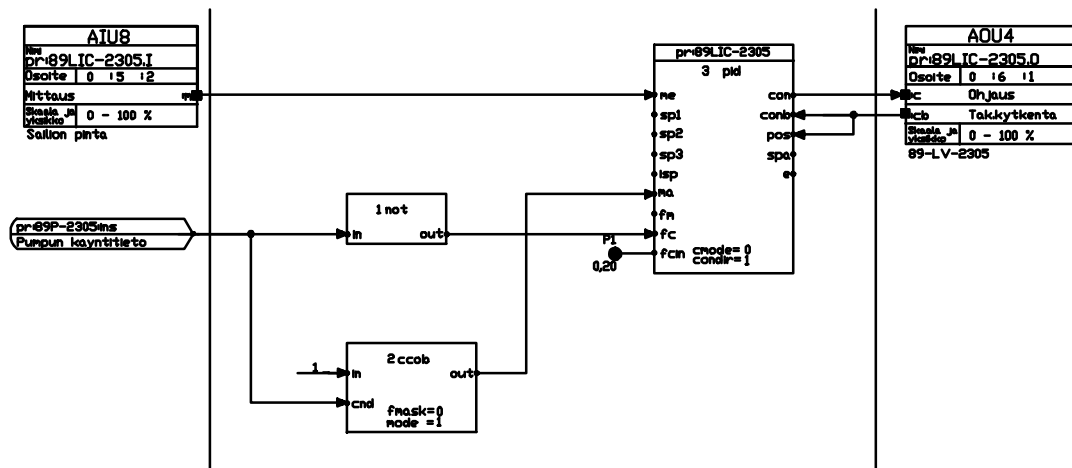
Tällöin porttiin kytkettävä signaali on määritelty kytkettävässä signaalissa.

### 5.2.3 Toimintaosa

Moduulin toimintaosa koostuu toimilohkoista. Näitä yhdistelemällä ja niihin ESITTELY-OSAssa määritellyjä tietopisteitä ja portteja yhdistelemällä saadaan muodostettua haluttu moduulin toiminta.

Moduulin toimintaosa koostuu toimilohkoista, joita voidaan parametroida ja kytkeä keskenään. Kuvamoduulin toimilohkot ovat yleensä itsenäisiä, joten ne ainoastaan parametroidaan. Sen sijaan toimintamoduulissa voi olla runsaastikin kytkentöjä.

Seuraavassa kuvassa on osa automaatiomodulia ja sen jälkeen vastaava konfigurointimoduulin toimintaosa listamuodossa.



## TOIMINTAOSA

```

1not
in< pr:89P-2305:ins
out> -
;

2ccob
mode= ( 1 )
fmask= ( 0 )
cnd< pr:89P-2305:ins
in< ( 1 )
out> -
;

3pid ON pr:89LIC-2305
pu= -
squ= -
iu= -
deu= -
ffu= -
parx= -
cmode= ( 0 )
condir= ( 1 )
aftfm= ( 0 )
aftfc= ( 1 )
fbact= ( 2 )
fbmask= ( 12 )
memi= ( 0.0 )
mema= ( 100.0 )
ffmi= -
ffma= -
comi= ( 0.0 )
coma= ( 100.0 )
bias= -
conch= ( 1.0 )
slow= -
sp2u= ( 0 )
sp3u= ( 0 )

track1= ( 0 )
track2= ( 0 )
track3= ( 1 )
chal= -
cha2= -
cha3= -
mau= -
eau= ( 0 )
coau= ( 0 )
ahys= -
kp< ( 0,0.75 )
ti< ( 0,30.0 )
td< -
tdf< -
kff< -
me< pr:89LIC-2305.I:m
mff< -
sp1< -
sp2< -
sp3< -
conb< pr:LIC-2305.0:cb
colmi< ( 0,0.0 )
colma< ( 0,100.0 )
isp< -
ma< 2ccob:out
parch< -
amc< ( 1 )
mac< ( 1 )
ion< -
fm< -
fc< 1not:out
fcin< P1
pos< pr:89LIC-2305.0:cb
mehh< ( 100.0 )
meh< ( 100.0 )
mel< ( 0.0 )
mell< ( 0.0 )
;

```

Toimintaosassa on kukin toimilohko esitetty omassa kokonaisuudessaan. Toimilohkot numeroidaan esim. 1not, 2ccob ja 3pid. Toimilohkon toimintaa voidaan listamuodossa selvittää kirjoittamalla toimilohkon tunnuksen ja nimen jälkeen kommentin lainausmerkkeihin, esim. 3pid ”Säädin”.

Esimerkin toimilohkon 2ccob toimintaa voidaan muuttaa konfigurointiparametreilla mode ja fmask, joille on listamuodossa annettu arvo ”=”-merkin jälkeen. Tässä toimilohkossa on alkuarvotettu vain mode, jolloin fmask saa ccob-toimilohkon kuvauksessa määritellyn oletusarvon.

Toimilohkon kytkennät määritellään ”<” (tulo)- ja ”>” (lähtö) -merkeillä osoitettuihin jäseniin. Edellä olevassa esimerkissä ulkoinen tulo pr:89LIC-2305:ins on kytketty 1not-toimilohkon in-tuloon (in< pr:89P-2305:ins).

Samana toimilohkon lähtö out on jätetty kytkemättä, jolloin siihen kirjoitetaan ”-”-merkki. (Käytettäessä toimilohkopohjaa listausta tehtäessä on jäsenissä valmiiksi ”-”-merkit.) Signaali voidaan jättää määrittelemättä, sillä siihen kytkettävä signaali määritellään signaalin toisessa ”päässä” eli 3pid:n fc:ssä (fc< 1not:out).

2ccob-toimilohkon tuloon in on kytketty vakio ”1”, joka esitetään listamuodossa seuraavasti in< ( 1 ).

Graafisessa suunnittelussa toimilohkon parametrit kysellään selväkielisessä muodossa käyttäjältä ja signaalien kytkennät tehdään yhdistämällä kytkettävät pisteet viivoilla toisiinsa. Tästä graafisesta kuvasta muodostetaan generointiohjelmilla listamuotoiset konfigurointimoduulit, jolloin edellä esitetyt signaalien kytkentää kuvaavat rakenteet muodostuvat automaattisesti.

Listamuodossa moduuleita konfiguroidessa saadaan avustuksia mm. editorin komentoista ja käytettävistä tyypeistä (toimilohkoista). Nämä toimilohkoavustukset (toimilohkopohjat) saadaan poimittua avustuksista suunniteltavaan moduuliin, jolloin ko. toimilohkon kaikki jäsenet tulevat näkyviin. Toimilohkoista tarvitsee alkuarvottaa vain ne jäsenet, joiden oletusarvoisia alkuarvoja halutaan muuttaa. Samoin vain ne jäsenet tarvitsee kytkeä, jotka ovat tarpeellisia ko. moduulin toiminnalle.

Toimilohkomäärittely loppuu puolipisteeseen.

Toimilohkoa määriteltäessä on huomattava:

- Konfigurointiparametreja ei voi kytkeä; niille voi vain antaa lukuarvoja. Sitä kuvaa merkki ”=” parametrien yhteydessä. Tällöin on huomattava, että esimerkiksi pid-säätimen viritysparametrit eivät ole pid-toimilohkon kannalta konfigurointiparametreja vaan tuloja. Niitä voidaan siis ohjata esimerkiksi toisella toimilohkolla.
- Vain ne konfigurointiparametrit (=), tulot (<) ja lähdöt (>), joihin halutaan syöttää tietoa, on esitettävä. Muille käytetään toimilohkomäärittelyn mukaisia oletusarvoja.
- Tuloihin ja lähtöihin voi kytkeä myös vakioarvon, joka annetaan kaarisulkeiden sisällä. Sulkeet tarvitaan, jotta esimerkiksi vakio (578.051) voidaan erottaa ulkoisesta tiedosta 578.051. Molemmathan voidaan kytkeä toimilohkoon. (Graafisessa suunnittelussa ei sulkeita tarvita vakion ympärille.)

(Laskenta-, logiikka- ja vertailutoimilohkojen käyttö eroaa muiden toimilohkojen käytöstä hieman.)

## Laskenta

Konfiguroija määrittelee laskennan algoritmisesti kirjoittamalla matemaattiset lausekkeet suoraan normaalimuodossaan. Hän voi käyttää hyväkseen operaattoreiden prioriteetteja, sulkumerkkejä jne.

Laskentaa voidaan suorittaa seuraaventyypisillä muuttujilla:

- **ana** analogiamuuttuja
- **ints** 16 bitin kokonaisluku vikabitteineen
- **intl** 32 bitin kokonaisluku vikabitteineen.

Siinä voidaan käyttää hyväksi seuraavia operaattoreita (laskevassa prioriteettijärjestyksessä):

- **\***, **/** = kerto- ja jakolasku
- **+**, **-** = yhteen- ja vähennyslasku

Laskenta voidaan ryhmitellä myös sulkumerkeillä, joilla on suurin prioriteetti.

Funktioita ovat:

- **SIN** = sinifunktio
- **EXP** = e-kantainen eksponentti
- **LN** = e-kantainen logaritmi
- **SQRT** = neliöjuuri
- **ABS** = itseisarvo

Laskenta määritellään osana konfigurointimoduulin toimilohkolistaa, kuten seuraava esimerkki osoittaa.

```

ESITTELYOSA

    ULKOISET
        pr:F-129.I:m TYYPPI ana = (0,0.5) SIIRTO 192,4,0,0 ;

    PAIKALLISET
        P1 TYYPPI ana = (0,0.6000) ;

    SUORASAANTI
        pr:FF-128 TYYPPI ana < - ;

    RAJAPINTA
        out1 TYYPPI ana < - ;
        MODSTAT TYYPPI ktstat < (1,1,0,1,1);

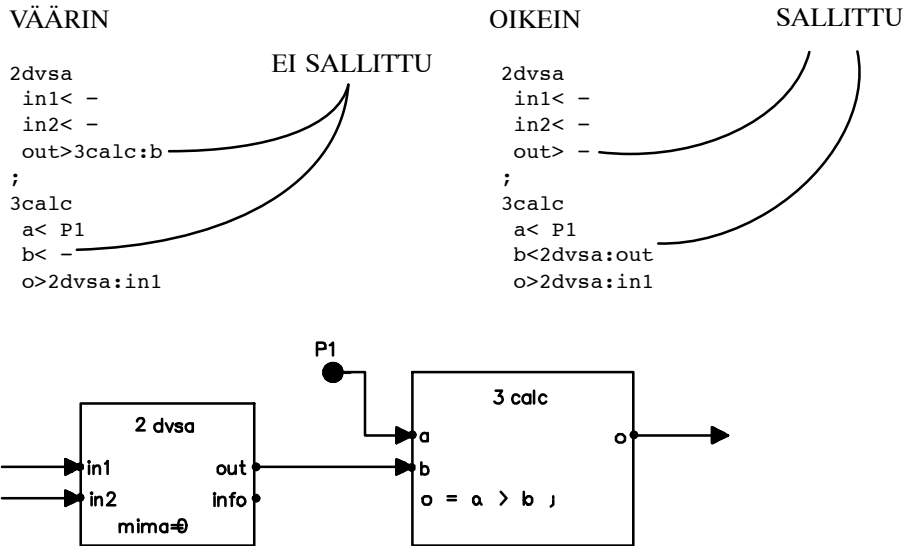
    TOIMINTAOSA
        LASKE lcalc
            KYTKE
                a TYYPPI ana < pr:FF-128 ;
                b TYYPPI ana < pr:F-129.I:m ;
                c TYYPPI ana < P1 ;
                o TYYPPI ana > out1 ;
            KAAVAT
                o=a*b/c;
        LOPETA lcalc
    .
    .
LOPPU

```

Siis laskennan syöttö on hyvin samantapaista kuin toimilohkojenkin, suunnittelijan on vain syötettävä liityntöjen tunnukset ja annettava niiden tyypit, koska valmista tyyppimäärittelyä ei ole kuten toimilohkoille yleensä.

Laskenta samoin kuin logiikka ja vertailutoimilohkojen kytkeminen on hieman rajoitetumpaa kuin ns. tavallisten toimilohkojen: Niihin ei voida kytkeytyä ulkoa päin, eli niiden jäseniin ei voi antaa ”-”-merkkiä ja määrittellä toisissa toimilohkoissa kytkentä. Näiden toimilohkojen kytkennät on määriteltävä toimilohkoissa tai käytettävä paikallisia tietopisteitä kytketymissä.

Seuraava esimerkki selventää väärää ja oikeaa kytkentätapaa lausekielessä:

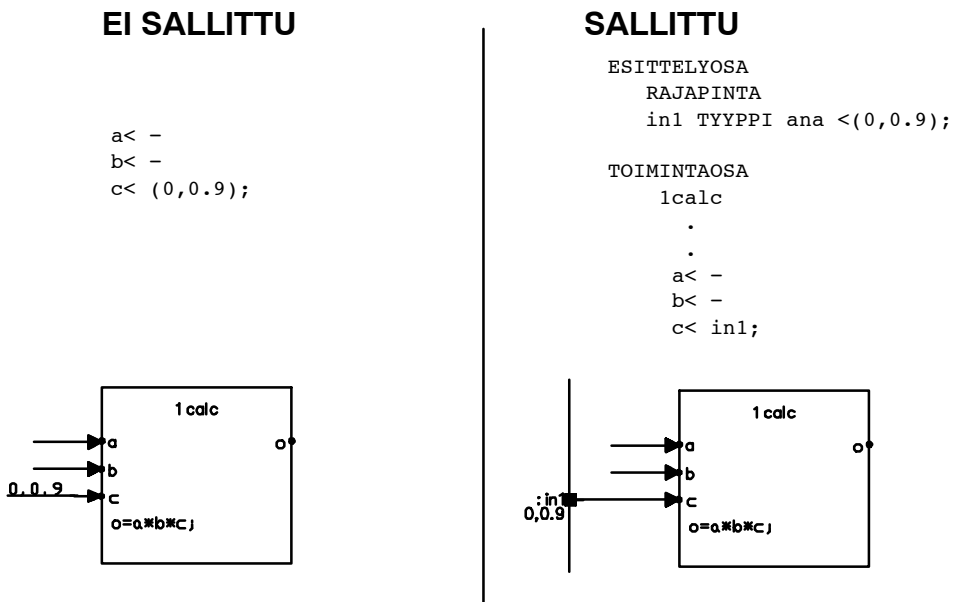


Graafisessa suunnittelussa käännösohjelmat muodostavat automaattisesti sallitut kytkennät.

Toinen rajoitus laskenta-, logiikka- ja vertailutoimilohkoissa on se, että niiden jäseniin ei voi suoraan kytkeä vakioita, vaan niiden kytkeminen on suoritettava joko paikallisen tietopisteen tai rajapinta-/suorasaantiportin kautta. Portille/tietopisteelle annetaan alkuarvoksi se arvo, joka halutaan kytkeä toimilohkoon ja portti kytketään toimilohkoon.

Suoraan kaavaan voi syöttää vakioarvon (esim.  $o = a * b * 100.0$ ). Tällöin ei kuitenkaan voida esim. debuggerilla muuttaa ko. vakion arvoa.

Jos vakion arvoa halutaan asettaa debuggerilla, kannattaa se kytkeä rajapinta- tai suorasaantiportiin ja alkuarvottaa halutulla vakiolla tämä portti.



Kolmas rajoitus laskenta-, logiikka- ja vertailutoimilohkoissa on se, että niitä ei voi kytkeä suoraan toisiinsa. Jos näitä toimilohkoja tarvitsee kytkeä toisiinsa, käytetään kytkentöjen tekemiseen paikallisia tietopisteitä.

Graafisessa suunnittelussa ko. paikallinen tietopiste generoituu konfigurointimoduuliin automaattisesti.

## EI SALLITTU

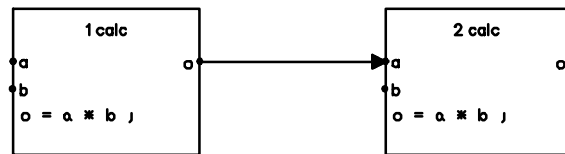
```
TOIMINTAOSA
1calc
o> 2calc:c ;
```

## SALLITTU

```
ESITTELYOSA
PAIKALLISET
P1 TYYPPI ana =(0,0.9);
```

```
TOIMINTAOSA
1calc
o> P1;

2calc
c<P1;
```



## Vertailu

Vertailu määritellään hyvin samantapaisesti kuin laskentakin, mutta tulot voivat olla tyyppiä ana, ints tai intl ja lähtö on tyyppiä bin.

Vertailua muodostettaessa voidaan käyttää seuraavia vertailuoperaattoreita:

- >= suurempi tai yhtäsuuri kuin
- <= pienempi tai yhtäsuuri kuin
- == yhtäsuuri kuin
- != erisuuri kuin
- > suurempi kuin
- < pienempi kuin

Lisäksi on käytettävissä seuraavat loogiset operaattorit cmp-toimilohkon sisällä muodostettujen bin-signaalien kesken:

- **OR** looginen tai
- **XOR** looginen ehdoton tai
- **AND** looginen ja
- **NOT** negaatio

Vertailua voi ryhmitellä sulkumerkeillä.



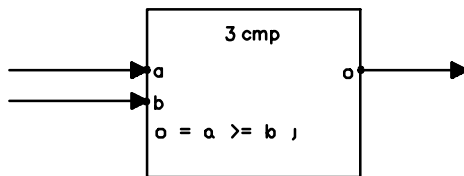
Vertailu määritellään laskennan tavoin osana konfigurointimoduulin toimilohkolistaa.

```

TOIMINTAOSA
.
.
VERTAILU 3cmp
  KYTKE
    a TYYPPI ana < 1pid:con;
    b TYYPPI ana < 2pid:con;
    o TYYPPI bin > out2;
  KAAVAT o = a >= b;
  LOPETA 3cmp
.
.
LOPPU

```

Vastaava esimerkki graafisessa muodossa:



## Logiikka

Logiikka määritellään hyvin samantapaisesti kuin laskenta ja vertailu, mutta tulot ja lähdöt ovat tyyppiä bin.

Logiikkaa muodostettaessa on käytettävissä seuraavat loogiset operaattorit:

- **OR** looginen tai
- **XOR** looginen ehdoton tai
- **AND** looginen ja
- **NOT** negaatio
- **SR(s,r,i)** ja **RS(r,s,i)** SR- ja RS-kiikut

Logiikkaa voi ryhmitellä sulkumerkeillä.

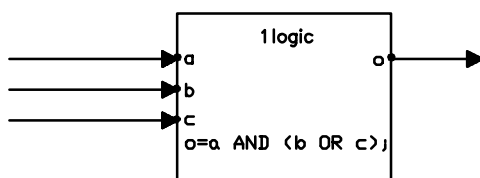
Logiikkaa määritellään laskennan ja vertailun tavoin osana konfigurointimoduulin toimilohkolistaa.

```

TOIMINTAOSA
.
.
LOGIIKKA 1logic
  KYTKE
    a TYYPPI bin <pr:507.102.I1:m;
    b TYYPPI bin <2pid:ma;
    c TYYPPI bin <in1;
    o TYYPPI bin >out1;
  KAAVAT o = a AND (b OR c);
  LOPETA 1logic
.
.
LOPPU

```

Vastaava esimerkki graafisessa muodossa:



### 5.3 MODUULIN SISÄISET KYTKENNÄT

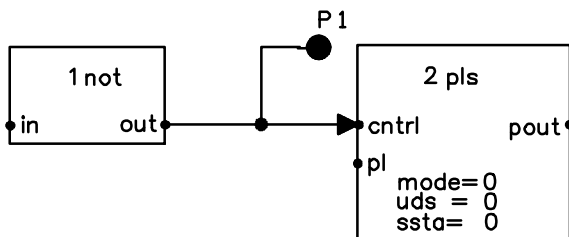
Moduulin sisällä tiedonsiirto perustuu kytkentään. Kytkenä syntyy kahden tai useamman kytkentäpisteen osoittaessa samaan tietopisteeseen. Kytkenäpisteen ja siihen kytketyn tietopisteen tai indeksillä tai tarkentimella erotetun tietopisteen osan on oltava samaa tyyppiä.

**Toimilohkojen välinen kytkentä** voidaan tehdä

- määrittelemällä paikallinen tietopiste ja kytkemällä siihen toimilohkojen kytkentäpisteitä tai
- suoraan jonkin toimilohkon kytkentäpisteen kohdalla kirjoittamalla siihen kytkettävä toimilohko ja sen kytkentäpiste. Tällöin kytkentä tehdään automaattisesti muodostuvan käyttäjälle näkymättömän tietopisteen kautta. Jos kytkentään syötetään arvo, muodostuu kytkentä tietopisteeseen, joka on alustettu kyseisellä arvolla.

Näitä menetelmiä havainnollistavat seuraavat kaksi esimerkkiä.

*Esimerkki 1: Kytkentä erikseen määritellyn paikallisen tietopisteen kautta*

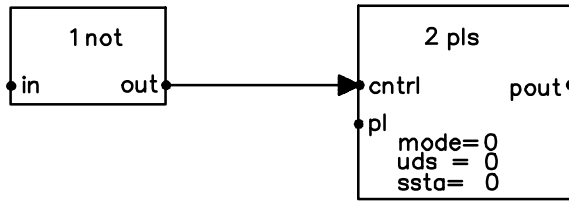


```

ESITTELYOSA
.
.
    PAIKALLISET
        P1 TYYPPI ana;
.
.
TOIMINTAOSA
    1not
        in< -
        out> P1
    ;
    2pls
        cntrl< P1
        pout> -
    ;
.
.

```

Esimerkki 2: Kytkeä suoraan toimilohkojen liityntäpisteiden välille



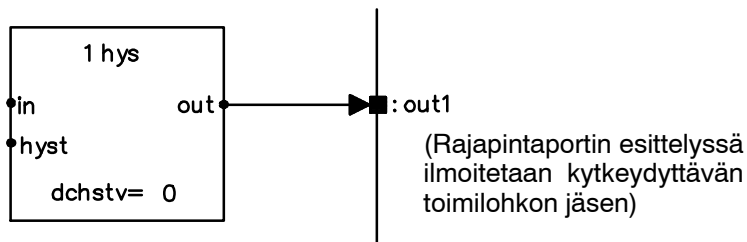
```

TOIMINTAOSA
  1not
  in< -
  out> -
  ;
  2pls
  cntrl< 1not:out
  pout> -
  ;
  .
  .
  .
  
```

Esimerkit koskevat lähinnä vain listamuotoista suunnittelua. Graafisessa suunnittelussa ei käytetä kuin esimerkin 2 esittämää tapaa.

**Toimilohkojen ja porttien välisiä kytkentöjä** havainnollistavat vastaavasti seuraavat esimerkit:

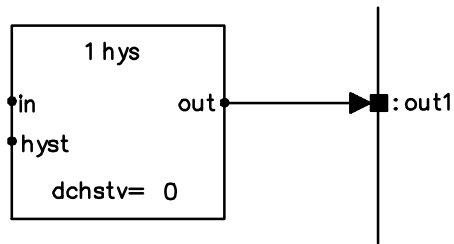
Esimerkki 3: Kytkeä tehdään suoraan portin ja toimilohkon välille. Rajapintaportin esittelyssä ilmoitetaan kytkettävä toimilohko ja sen liityntäpiste.



```

ESITTELYOSA
  .
  .
  RAJAPINTA
    out1 TYYPPI ana< 1hys:out;
  .
  .
  TOIMINTAOSA
    1hys
    in< -
    out> -
    ;
  
```

Esimerkki 4: Kytettä tehdään suoraan portin ja toimilohkon välille. Toimilohkon kytkentäpisteelle ilmoitetaan kytkettävä rajapintaportti.



(Toimilohkon kytkentäpisteessä määritellään kytkettävä rajapintaportti)

```

ESITTELYOSA
.
.
.
RAJAPINTA
    out1 TYYPPI ana< -;
.
.
TOIMINTAOSA
    lhys
    in< -
    out> out1
;

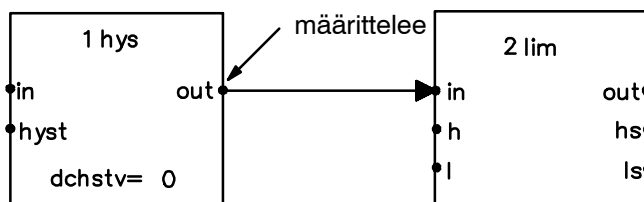
```

Konfigurointiohjelmiston käännöstyökalu muuttaa kytkennät tavan 4 mukaiseksi.

Kun halutaan kytkeytyä sellaiseen kytkentäpisteeseen, johon toinen kytkentäpiste on kytketty, voidaan käyttää polkua kyseessä olevaan toiseen kytkentäpisteeseen. Polku muodostuu toimilohkon tunnuksesta ja sitä tarkentavasta kytkentäpisteen tunnuksesta. Tunnuksen avulla voidaan kytkeytyä sillä hetkellä vielä määrittelemättömiin toimilohkojen kytkentäpisteisiin; varsinainen kytkentä muodostuu vasta toimilohkoa määriteltäessä.

Seuraavassa on tehty sama kytkentä kolmella eri tavalla tunnuspolkua käyttäen.

#### Tapa 1

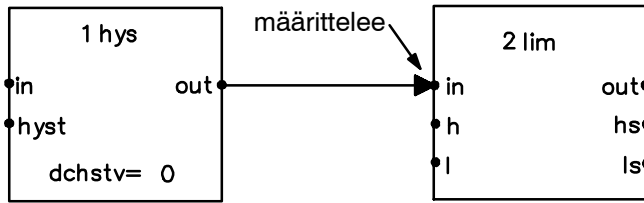


```

.
.
    lhys
    out> 2lim:in
;
    2lim
    in< -
;
.
.

```

## Tapa 2



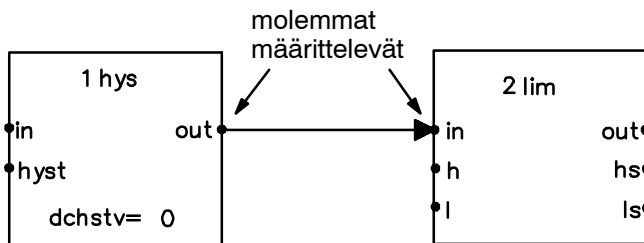
```

.
  1hys
    out> -
  ;
  2lim
    in< 1hys:out
  ;
.

```

Konfigurointiohjelmiston käännöstyökalu muuttaa kytkennät tavan 2 mukaiseksi (ei calc-, cmp- eikä logic-toimilohkojen tapauksissa, koska niiden jäseniin ei voi kytkeytyä ulkopuolelta).

## Tapa 3



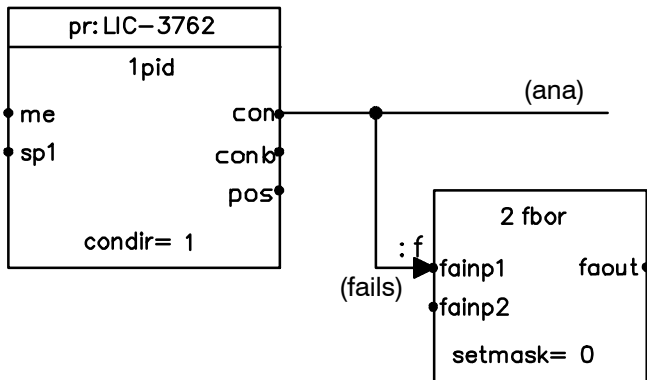
```

.
.
  1hys
    out> 2lim:in
  ;
  2lim
    in< 1hys:out
  ;
.
.

```

Rakenteisesta signaalista voidaan poimia koko signaalin lisäksi jokin sen jäsen. Tämä voidaan suorittaa tarkentamalla kytkettävää signaalia ko. tarkentimella.

Seuraavassa on esimerkki tietopisteen tarkentamisesta haluttaessa poimia ana-tyyppisestä signaalista pelkät vikabittitiedot:



Kuva 15 Esimerkki kytkentäpisteen tarkentamisesta

PID-säätimen 1pid lähtösignaali on ana-tyyppiä, jolloin se sisältää tyyppinsä mukaisesti vikabittitiedon f ja itse signaalin arvon a. Jos halutaan poimia pelkästään signaalissa olevat vikabitit saadaan ne poimittua koko ana-tyyppisestä signaalista tarkentamalla signaalia vikabittitiedon tunnuksella f seuraavasti: 1pid:con:f.

Vastaava listamuotoinen kohta moduulista:

```

TOIMINTAOSA
.
.
.
    2 fbor
    fainp1< 1pid:con:f
.
.
.

```

Signaalin tarkennus tehdään yleensä aina signaalia käyttävän toimilohkon ao. tuloon.

## 5.4 MODUULIEN VÄLINEN KOMMUNIKOINTI

Moduulien välinen kommunikointi tapahtuu portin ja ulkoisen tietopisteen muodostaman parin avulla. Kun moduuli haluaa tietoa toisesta moduulista, se esittelee ulkoisen tietopisteen, johon tieto toisesta moduulista kopioidaan sen portin kautta. Ulkoisen tietopisteen tunnuksesta selviää, mistä toisesta moduulista, minkä portin kautta ja mitä tietoa siirretään.

Jos kaikki tietyn toimilohkon kytkentäpisteiden tiedot halutaan siirtää moduulin porttiin, portti voidaan määritellä tiettyyn toimilohkoon sidotuksi lohkoksi. Tällöin kaikkiin kyseisen toimilohkon kytkentäpisteisiin päästään käsiksi määritellyn lohkon kautta.

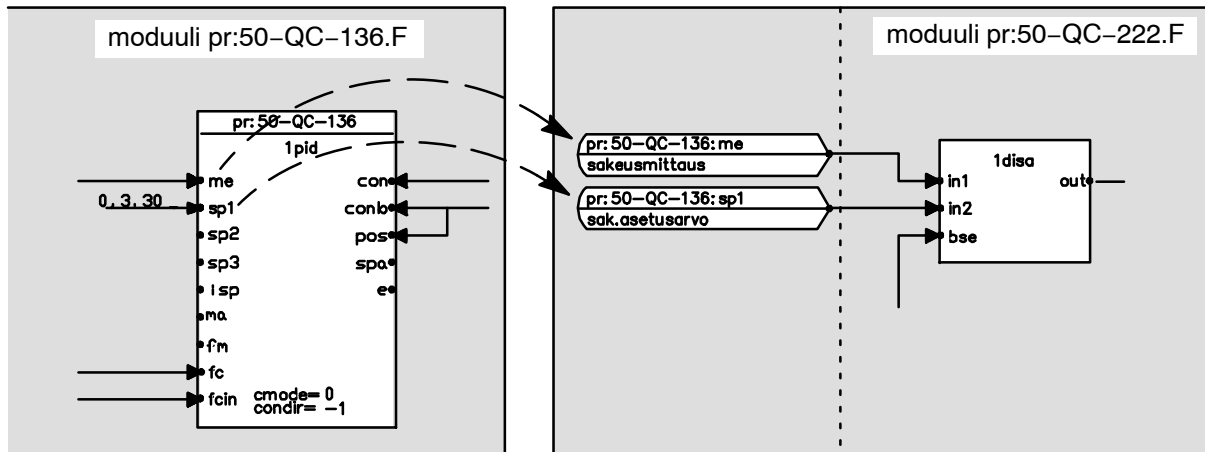
Moduulien välinen kommunikointi tapahtuu siis kopioimalla tietoa. Mikäli moduulista pyydettyä tietoa on rakenteista ja siten koostuu useista jäsenistä, voivat muut moduulit tästä rakenteisesta tiedosta pyytää:

1. Koko tiedon eli kaikki jäsenet
2. Nimetyt jäsenet
3. Joukon valittuja jäseniä tietyllä **näkökulmalla** (valvomo)

### 5.4.1 Suorasaantiportin käyttö kommunikoinnissa

Seuraavassa kuvassa (Kuva 16) on esitetty, kuinka kaksi moduulia voivat kommunikoida keskenään käyttäen suorasaantiporttia ja ulkoisia tietopisteitä.

Moduulin pr:50-QC-136.F koko PID-säädin on kytketty suorasaantiporttiin pr:50-QC-136. Tällöin kaikki säätimen jäsenet on saatavilla muille moduuleille niiden omien ulkoisten tietopisteiden välityksellä (esim. moduulissa pr:50-QC-222.F ulkoiset tietopisteet pr:50-QC-136:me ja pr:50-QC-136:sp1).



Kuva 16 Moduulien kommunikointi suorasaantiportista ulkoiseen tietopisteeseen

Lausekielisessä moduulissa pr:50-QC-136.F on oltava seuraavat määrittelyt, jotta kommunikointi on mahdollista:

```

ESITTELYOSA
  SUORASAANTI
    LOHKO pr:50-QC-136

TOIMINTAOSA
  1pid ON pr:50-QC-136
  .
  .
  .

```

- Esittelyosassa määritellään pr:50-QC-136-niminen suorasaantiportti, joka viittaa toimilohkoon.
- Toimintaosassa määritellään 1pid-säädin suorasaantiportiksi pr:50-QC-136.

Lausekielisessä moduulissa pr:50-QC-222.F on oltava seuraavat määrittelyt tiedonsiirron onnistumiseksi:

```

ESITTELYOSA
  ULKOISET
    pr:50-QC-136:me TYYPPI ana SIIRTO 192,10,0,0;
    pr:50-QC-136:sp1 TYYPPI ana SIIRTO 192,10,0,0;

TOIMINTAOSA
  1disa
    .
    .
    in1< pr:50-QC-136:me
    in2< pr:50-QC-136:sp1
    .
    .

```

- Näillä ulkoisten tietopisteiden esittelyillä saadaan moduulin ulkopuolista tietoa siirrettyä moduulin sisälle. Ensimmäisessä ulkoisessa tietopisteessä (pr:50-QC-136:me) on pid-toimilohkosta otettu sen me-jäsen, jonka tieto kytketään 1disa-toimilohkon in1-tuloon. Toisessa ulkoisessa tietopisteessä (pr:50-QC-136:sp1) on otettu pid-toimilohkon sp1-jäsen ja kytketty se 1disa-toimilohkon in2-jäseneseen.

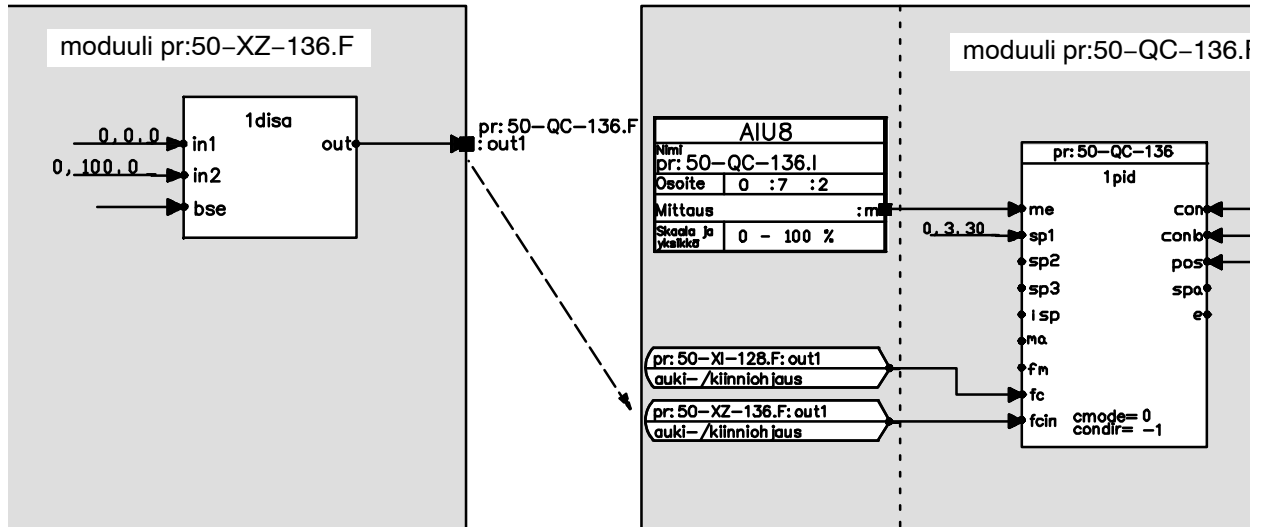
Ulkoisesta tietopisteestä voidaan siirtää tietoa myös toiseen suuntaan porteille. Tiedonsiirron suunta ja muut siirtotiedot annetaan ulkoisen tietopisteen siirtomäärittelyllä. Siirto voi tapahtua muutoksesta, jaksolla tai ehdollisesti.

Näin menetellään yleensä vain sekvensseissä (toimenpiteiden yhteydessä) ja kirjoitettaessa tietoa lähtökorteille.



## 5.4.2 Rajapintaportin käyttö kommunikoinnissa

Seuraava esimerkki selventää moduulien välistä kommunikointia rajapintaporttien ja ulkoisten tietopisteiden välillä.



Kuva 17 Moduulien kommunikointi rajapintaportista ulkoiseen tietopisteeseen

Jotta kommunikointi olisi mahdollista, on lausekielisessä moduulissa pr:50-XZ-136.F oltava rajapintaportti seuraavasti:

```
ESITTELYOSA
RAJAPINTA
    out1 TYYPPI ana <1disa:out;
MODSTAT TYYPPI ktstat < (1,1,0,1,1);
```

- Esittelyosassa on esitelty ana-tyyppinen rajapintaportti out1 ja sen kytkentä 1disa-toimilohkon out-jäseneseen. Tällä tavalla on saatu toimilohkon 1disa jäsen out tuotua moduulin sisältä sen rajapintaan ja muiden moduulien saataville.

Moduulissa pr:50-QC-136.F on lausekielisessä muodossa oltava ulkoinen tietopiste, jotta toisen moduulin tietoa voitaisiin käyttää hyväksi.

```
ESITTELYOSA
ULKOISET
    pr:50-XZ-136.F:out1 TYYPPI ana SIIRTO 192,10,0,0;
```

- Ulkoisen tietopisteen nimi muodostuu lähdekonfigurointimoduulin nimestä pr:50-XZ-136.F ja sen loppuun on lisätty rajapintaportin nimi out1 kaksoispisteellä eroteltuna. Tiedonsiirtomäärittelyllä ulkoinen tietopiste on määritetty jatkuvaksi tuloksi.

Ulkoisten tietojen käyttäminen moduulin sisällä on mahdollista seuraavasti:

```
TOIMINTAOSA
1pid
.
.
.
fcin< pr:50-XZ-136.F:out1
.
.
.
```

### 5.4.3 Näkökulman käyttö kommunikoinnissa

Moduulit voivat pyytää rakenteista tietoa muista moduuleista myös näkökulmilla. Näkökulmalla pyydetty tieto muodostuu vain tietyistä jäsenistä, jotka kyseinen näkökulma valitsee. Näkökulmalla pyydetty tieto siirretään näkökulmatyyppin kanssa tyypiltään yhteensopivaan toisen moduulin ulkoiseen tietopisteeseen. Käyttämällä näkökulmia saadaan vain kulloiseenkin tilanteeseen tarpeellinen määrä tietoa. Valvomo on tyypillinen näkökulmien käyttäjä.

Seuraavassa on esimerkki, kuinka tietoa voidaan siirtää prosessinohjauspalvelimelta.

*Esimerkki:* prosessinohjauspalvelimen toimintamoduuli pr:LI-700.F:

```

HALLINTAOSA
  NIMI:          pr:LI-700.F
  TYYPPI:        toiminta
  TILA:          kesken
  LUOJA:         simo
  LUOTU:         89-06-13 09:32
  MUUTTAJA:     simo
  MUUTETTU:     89-06-20 11:16
  SIJAINTI:     AP01
  SUORITUS:     400
  MONESKO:      3
  KUVAUS:       "testimoduuli"

ESITTELYOSA
  ULKOISET
    pr:LI-700.I:m TYYPPI ana SIIRTO 192,4,0,0;
  SUORASAANTI
    LOHKO pr:LI-700
  RAJAPINTA
    MODSTAT TYYPPI ktstat < (1,1,0,1,1);

TOIMINTAOSA
  lam ON pr:LI-700
  hyst= ( 4 )
  un= -
  av< pr:LI-700.I:m
  hh< -
  h< -
  l< -
  ll< -
  out> -
  hha> -
  ha> -
  la> -
  lla> -
  fa> -
  ;

LOPPU

```

Moduuliin on määritelty suorasaantiportti pr:LI-700, joka viittaa moduulin ainoaan toimilohkoon 1am. Muut metsoDNA CR:n moduulit voivat pyytää moduulista pr:LI-700.F tietoa ulkoiseen tietopisteeseensä kolmella eri tavalla:

*Tapa 1:*

Pyydetään suorasaantiportin tietoja eli toimilohkon jäseniä käyttämällä *nippuporttia* (esimerkissä on am.opb-tyyppinen nippuportti, jolloin saadaan seuraavat jäsenet av, hh, h, l, ll, out).

```

ESITTELYOSA
  ULKOISET
    pr:LI-700 TYYPPI am.opb SIIRTO 192,4,0,0;
    :
    :

```

Tapa 2:

Pyydetään suorasaantiportin **nimettyä jäsentä** out.

```

ESITTELYOSA
  ULKOISET
    pr:LI-700:out TYYPPI ana SIIRTO 192,4,0,0;
    :
    :
  
```

Tapa 3:

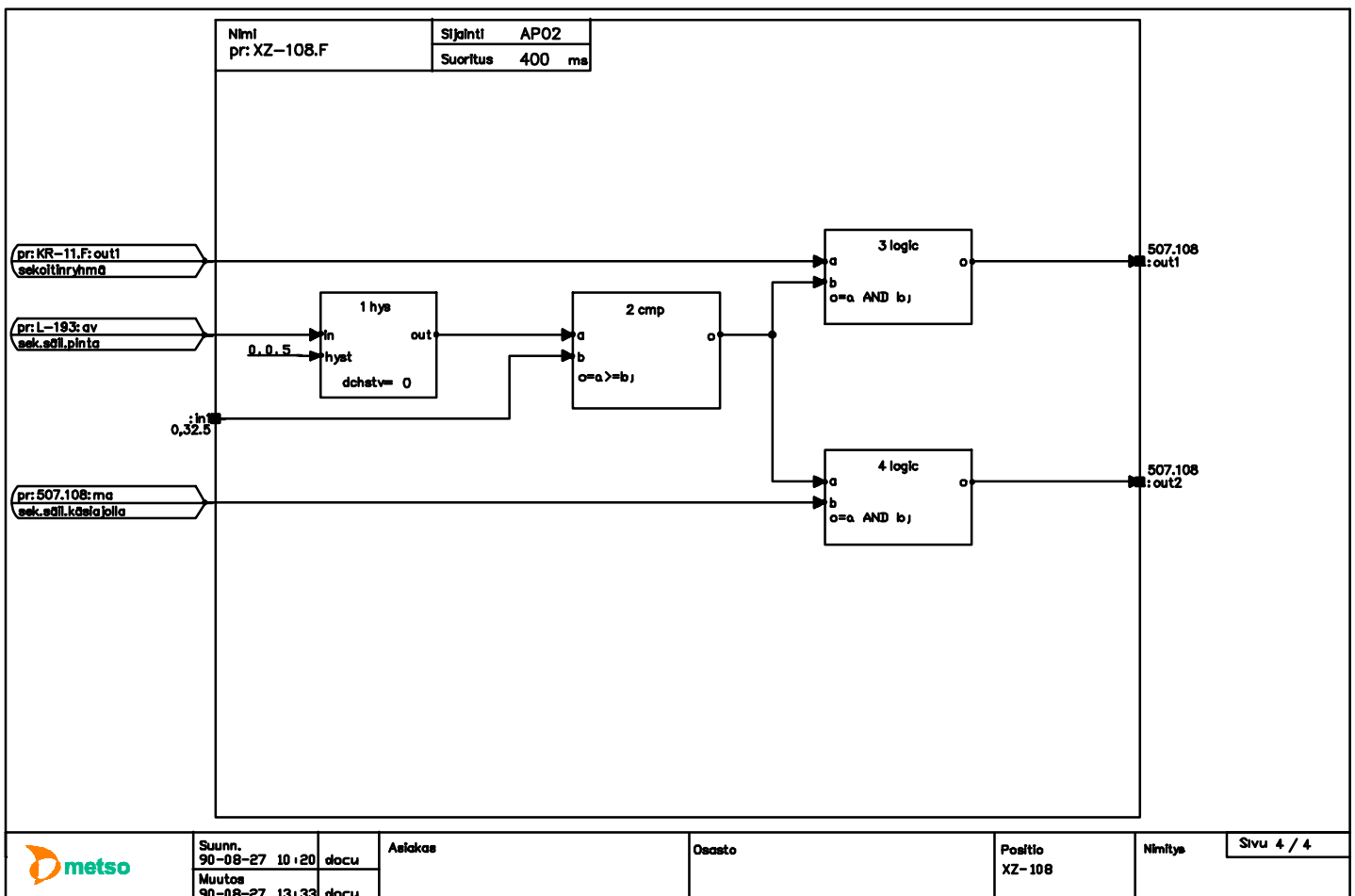
Pyydetään suorasaantiportin tietoa näkökulmalla pgd (= kaaviokuvan näkökulma), joka valikoi am-toimilohkon jäsenet av ja out (am-toimilohkon muita näkökulmia ovat mm. pt ja op).

```

ESITTELYOSA
  ULKOISET
    pr:LI-700:pgd TYYPPI am_pgd SIIRTO 192,10,0,0;
    :
    :
  
```

## 5.5 ESIMERKKI KONFIGUROIDIMODUULIN RAKENTEESTA, KYTKENNÖISTÄ JA KOMMUNIKOINNISTA

Tarkastellaan automaatiomoduaalia XZ-108 (Kuva 18).



Kuva 18

Automaatiomoduaali XZ-108

pr:L-193:av on moduulin ulkoinen analogiatyyppinen (ana) tietopiste, joka on kytketty toimilohkon 1hys tuloon in. Toimilohkon 1hys tulolle hyst on konfigurointivaiheessa annettu arvo 0,0.5. Toimilohkon lähtö out on kytketty toimilohkon 2cmp tuloon a. 2cmp:n toiseen tuloon b on kytketty rajapintaportti in1, jolle on annettu alkuarvo (0, 32.5).

Kaikki mainitut toimilohkojen kytkentäpisteet ovat rakenteista tietotyyppiä ana. Tyyppi ana rakentuu jäsenistä f, joka on johdettua alkeistyyppiä fails sekä a, joka on alkeistyyppiä float.

Edelleen 2cmp:n lähtö o on kytketty paikalliseen tietopisteeseen P1. Tämä paikallinen tietopiste on kytketty sekä toimilohkon 3logic tuloon b että toimilohkon 4logic tuloon a. Lohkossa 2cmp suoritettavan vertailun tuloksena saatava looginen arvo on tietotyyppiä bin (binääri). Tyyppi bin on johdettu alkeistyyppi (katso liite 1). Koska vain samantyyppisiä tietoja voidaan kytkeä keskenään, myös toimilohkojen 3logic ja 4logic tulot a ja b ovat tyyppiä bin. Näin myös toimilohkon 3logic tuloon a kytketty ulkoinen tieto pr:KR-11.F:out1 ja lohkon 4logic tuloon b kytketty ulkoinen tieto pr:507.108:ma ovat tyyppiä bin.

Logiikkatoimilohkoissa 3logic ja 4logic suoritettavien loogisten lausekkeiden tulokset – tyyppiä bin – on viety toimilohkojen lähdoista o moduulin rajapinnalle rajapintaportteille pr:XZ-108.F:out1 ja pr:XZ-108.F:out2. On huomattava, että lausekielisessä suunnittelussa logiikka-, vertailu- ja laskentatoimilohkojen kytkentäpisteisiin ei voi viitata suoraan, kytkentä on suoritettava paikallisen tietopisteen välityksellä.

Tarkastellussa moduulissa esiintyvät seuraavat tyypit:

```

ana
  JÄSENET
    f TYYPPI fails
    a TYYPPI float

bin    TYYPPI uns16

hys
  JÄSENET
    in TYYPPI ana
    hyst TYYPPI ana
    out TYYPPI ana

PAIKALLISET
  P1 TYYPPI bin;

RAJAPINTA
  in1 TYYPPI ana < (0,32.5);
  MODSTAT TYYPPI ktstat < (1,1,0,1,1);

```

Vertailu- ja logiikkatoimilohkojen tulojen lukumäärät sekä niiden tietotyypit määrätään konfiguroitaessa.

Esimerkkitapauksessamme lohkojen tietorakenteet ovat seuraavanlaiset (2cmp ja 3logic):

```

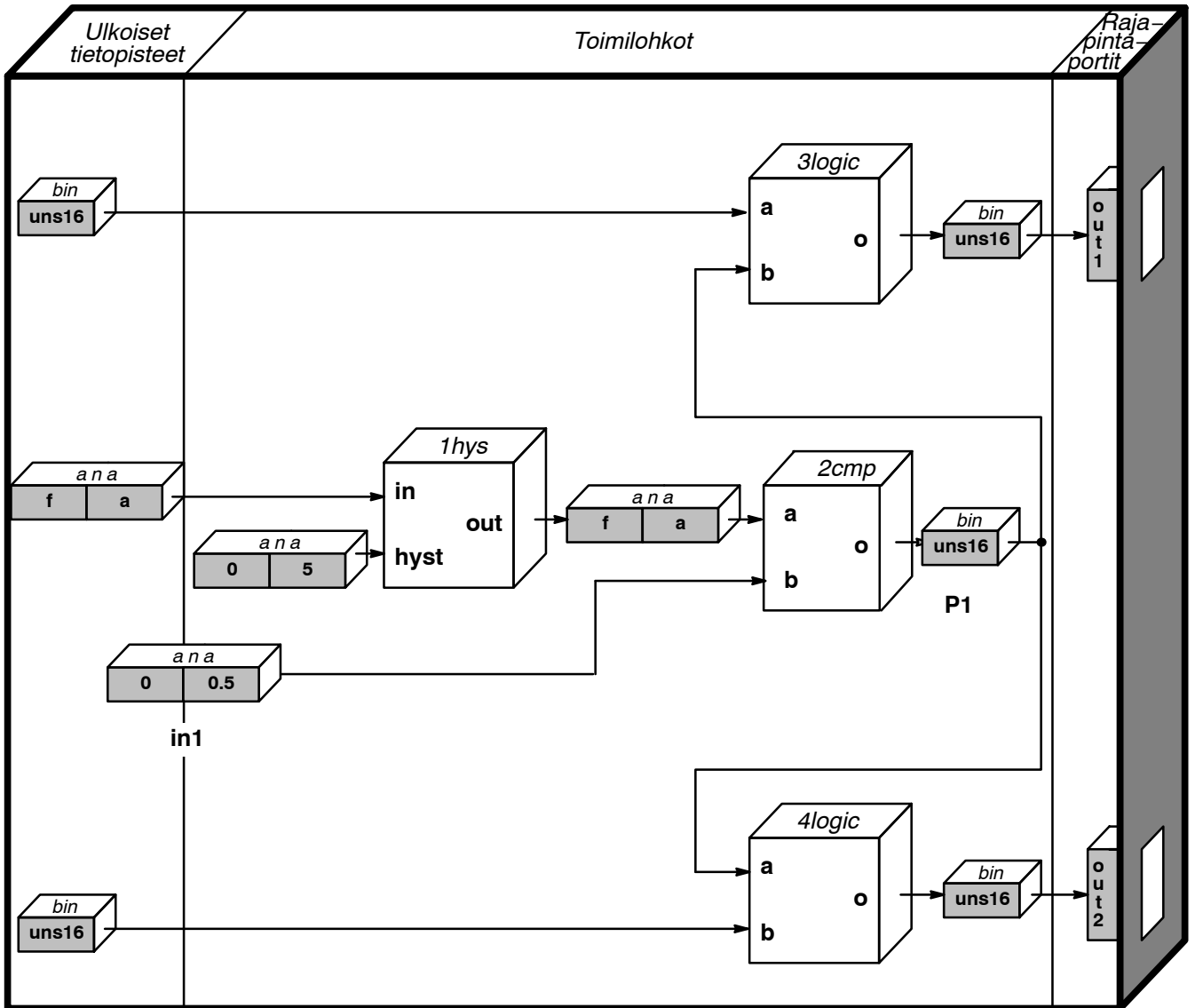
VERTAILU 2cmp
  KYTKE
    a TYYPPI ana < 1hys:out;
    b TYYPPI ana < in1;
    o TYYPPI bin > P1;
  KAAVAT
    o = a >= b;
  LOPETA 2cmp

LOGIIKKA 3logic
  KYTKE
    a TYYPPI bin < pr:50-KR-11.F:out1;
    b TYYPPI bin < P1;
    o TYYPPI bin > out1;
  KAAVAT
    o = a AND b;
  LOPETA 3logic

```

Tietotyyppien havainnollistamiseksi moduuli voidaan kuvata vielä tarkemmin seuraavasti.

### Moduuli XZ-108



Kuva 19 Moduuli XZ-108

Toimintamoduuli pr:XZ-108.F määritellään lausekielisessä muodossa seuraavasti:

```

HALLINTAOSA
NIMI:      pr:XZ-108.F
TYYPPI:    toiminta
TILA:      valmis
LUOJA:     timo
           LUOTU:      89-01-02 08:36
MUUTTAJA:  timo
           MUUTETTU:   89-01-02 08:55
SIJAINTI:  AP02
SUORITUS:  400
MONESKO:   3
KUVAUS:    "ESIMERKKI"

ESITTELYOSA
ULKOISET
pr:L-193:av TYYPPI ana SIIRTO 192,4,0,0 "Sek. säiliön pinta" ;
pr:KR-11.F:out1 TYYPPI bin SIIRTO 192,4,0,0 "Sekoitinryhmä" ;
pr:507.108:ma TYYPPI bin SIIRTO 192,4,0,0 "Sek. säil. käsiajolla" ;

PAIKALLISET
P1 TYYPPI bin;

RAJAPINTA
in1 TYYPPI ana < (0,32.5) ;
out1 TYYPPI bin "507.108"< - ;
out2 TYYPPI bin "507.108"< - ;
MODSTAT TYYPPI ktstat < (1,1,0,1,1);

TOIMINTAOSA
lhys
dchstv= (0)
hyst< (0,0.5)
in< pr:L-193:av
out> -
;

VERTAILU 2cmp
KYTKE
a TYYPPI ana< lhys:out;
b TYYPPI ana< in1;
o TYYPPI bin> P1;
KAAVAT
o = a >= b;
LOPETA 2cmp

LOGIIKKA 3logic
KYTKE
a TYYPPI bin < pr:KR-11.F:out1;
b TYYPPI bin < P1;
o TYYPPI bin > out1;
KAAVAT
o = a AND b;
LOPETA 3logic

LOGIIKKA 4logic
KYTKE
a TYYPPI bin < P1;
b TYYPPI bin < pr:507.108:ma;
o TYYPPI bin > out2;
KAAVAT
o = a AND b;
LOPETA 4logic

LOPPU

```

## 6 AUTOMAATIOKIELEN NIMEYSKÄYTÄNTÖ

### 6.1 YLEISTÄ

metsoDNA CR:ssä käytettävä automaatiokieli mahdollistaa nimipohjaisen kommunikoinnin eli tietoa metsoDNA CR:n sisällä siirretään nimien avulla.

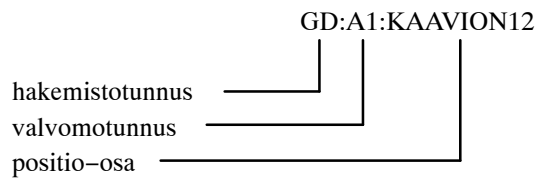
Erialaisten moduulien nimien muodostaminen on periaatteessa hyvin vapaata, mutta käytännössä on tarkoituksenmukaista käyttää yhtenäistä nimeyskäytäntöä.

Käytettäessä yhtenäistä nimeyskäytäntöä saavutetaan monia etuja:

- suunnittelussa ei tarvitse kiinnittää huomiota nimeyksen perusteisiin
- vian etsintä ja huoltotoimet ovat helpompia projektien ulkopuolisille henkilöille

### 6.2 MODUULIN NIMEN RAKENNE JA PITUUS

Moduulien nimet muodostuvat komponenteista:



Komponentit erotellaan toisistaan kaksoispisteellä (:).

Nimen kokonaispituus kaikki nimeen kuuluvat merkit huomioon ottaen voi olla 63 merkkiä.

Kaksoispisteillä erotettujen komponenttien pituus saa olla enintään 15 merkkiä.

### 6.3 MODUULIEN NIMISSÄ KÄYTETTÄVÄT MERKIT

#### Sallitut merkit

A – Z a – z 0 – 9 ' . / _ + =	Siis skandinaaviset merkit eivät ole sallittuja!
, -	Mikään komponentti ei saa alkaa näillä merkeillä!

#### Kielletyt merkit

kontrollimerkit välilyönti tabulointi # ; ” \$ & () * ? ! % < >	-
:	Komponenttien erotusmerkki

## Ei suositeltavat merkit

Seuraavien merkkien käyttöä nimissä ei kielletä, mutta niiden käyttöä ei suositella:

@ [ ] ^ { } | \ ' ~

Näiden merkkien käyttöä ei kielletä, koska 7-bittisessä merkistössä niitä käytetään kansallisia merkkeinä.

## 6.4 NIMEN HAKEMISTOTUNNUS

Jotta tunnistaissimme yksiselitteisesti automaatio-, dokumentti- tai konfigurointimoduulin nimestä, mikä moduuli tai dokumentti on kyseessä, käytetään hakemistotunnuksia erottamaan moduuleita toisistaan.

Automaatiomoduuleilla ja -kohteilla hakemistotunnus merkitään isoilla kirjaimilla. Konfigurointimoduuleissa hakemistotunnus merkitään pienillä kirjaimilla. Näin mahdollistetaan esim. automaatiomoduulin ja vastaavan konfigurointimoduulin käsittely suunnitteluympäristön työtilassa yksikäsitteisesti.

Esim. GdCADillä tehty kaaviokuva GD:A1:KAAVION3 ja vastaava automaatiokielineen konfigurointimoduuli gd:A1:KAAVION3.

### 6.4.1 Automaatiomoduulit

FbCADillä tehdyn automaatiomoduulin nimessä ei käytetä hakemistotunnusta.

LIC-100

SeqCADillä tehdyn automaatiomoduulin nimessä käytetään hakemistotunnuksena merkintää ”SQ”.

SQ:SEQ-100

GdCADillä ja GrCADillä tehdyissä automaatiomoduulien nimissä käytetään hakemistotunnuksia ”GD”, ”RP” ja ”MW”.

<b>GD:A1:KAAVION3</b>	(kaaviokuva)
<b>RP:A1:RESEPTIN4</b>	(reseptikuva)
<b>MW:A1:MWINDOW5</b>	(monitori-ikkuna-automaatiomoduuli)

### 6.4.2 Automaatiokohteet

WinTRENDillä tehdyn automaatiokohteen nimessä käytetään

trendikuville hakemistotunnuksena merkintää ”TR”.

**TR:A1:TRENDIN4**

piirturikuville hakemistotunnuksena merkintää ”RD”.

**RD:A1:RDDISP1\_1**



### 6.4.3 Konfigurointimoduulit

Konfigurointimoduulit erotetaan toisistaan käyttämällä hakemistotunnuksia ja loppuliitteitä (nimen lopussa pisteellä erotettuna).

#### HUOM!

Merkintä XX tarkoittaa valvomotunnusta, esim. A1 (kts. kohta 6.5 “Moduulin nimen valvomotunnus”).

Seuraavassa taulukossa on esitetty moduulitunnukset, moduuliliitteet ja moduulityypit sovelluspalvelintyypeittäin.

SOVELLUS-PALVELIN	TUNNUS	LIITE	MODUULITYYPPI
PCS	pr:	.F	toimintamoduuli
	pr:	.I	tulomoduuli
	pr:	.O	lähtömoduuli
	pr:	–	suorasaantiportin nimi
	sq:	.F	toimintamoduuli, sekvenssi
	il:	.F	toimintamoduuli, kaaviolamput
	il:	.O	lähtömoduuli, kaaviolamput
	re:XX:	.F	toimintamoduuli, piirturit
	re:XX:	.O	lähtömoduuli, piirturit
	ph:	–	historiamoduuli
	ph:me:	–	säätimen historiamoduuli
	ph:spa:	–	säätimen historiamoduuli
	ph:pos:	–	säätimen historiamoduuli
	rp:	.V	reseptin muunnelmamoduuli
	rp:	.P	reseptin parametrimoduuli
	rp:	.C	reseptin laskentamoduuli
rp:	.L	reseptin latausmoduuli	
DIS	xc:	.F	toimintamoduuli
	xi:	.IO	tulo-/lähtömoduuli
CIS	ci:	–	toimintamoduuli
LIS	li:	–	toimintamoduuli
SIS	si:	–	toimintamoduuli
REP	rs:	.C	raportoinnin keräys-, laskenta- ja ohjausmoduuli
	rs:	.R	raportin tallennusmoduuli
	rs:	.P	raportin tulostusmoduuli
ALP	al:XX:	.F	tapahtumamoduuli
	al:XX:	–	suorasaantiportin nimi
	sn:XX:	–	järjestelmämoduuli

OPS	ce:XX:	–	positiomoduuli
	cp:XX:	–	positiomoduulin suorasaantiportti
	od:XX:	–	operointimoduuli
	gd:XX:	–	kaaviokuvamoduuli
	gr:XX:	–	ryhmäkuvamoduuli
	tr:XX:	–	trendikuvamoduuli
	rp:XX:	–	reseptikuvamoduuli
	rd:XX:	–	piirturikuvamoduuli
	sd:XX:	.st	askelkuvamoduuli
	sd:XX:	.x	askelmoduuli
	mw:XX:	–	monitori-ikkunamoduuli
	ad:XX:	–	hälytyskuvamoduuli
	md:XX:	–	ilmoituskuvamoduuli
	td:XX:	–	testikuvamoduuli
	sn:XX:	–	järjestelmämoduuli

metsoDNA CR:ssä olevien diagnostiikka-anturien hakemistotunnus on ”di”. Antureita käytetään diagnostiikkatapahtumamoduuleissa.

## 6.5 MODUULIN NIMEN VALVOMOTUNNUS

Nimen valvomotunnusta käytetään operointipalvelimella ja hälytyskäsittelijällä olevissa moduuleissa sekä piirtureiden moduuleissa prosessinohjauspalvelimilla.

Nimessä esiintyvä valvomotunnus takaa kyseisen moduulin nimen yksikäsitteisyyden usean valvomon ja väylän verkossa. Valvomotunnus on kaksimerkkinen tunnus, joista ensimmäisen merkin on oltava kirjain ja toisen merkin numero. Tunnuksessa sallittuja merkkejä ovat kirjaimet A...Z sekä numerot 0...9.

1. merkki = prosessialueen (suunnittelalueen) tunnus A...Z
2. merkki = valvomon numero 1...9

Esim. gd:A1:KAAVION3

## 6.6 MODUULIN NIMEN POSITIO-OSA

### Positio-osan rakenne

Moduulin nimen positio-osa koostuu joko positiotunnuksesta ja laitepositiotunnuksesta loppuliitteinen tai soveltajan määrittelemästä nimestä.

Esim.

LC:LIC-100	(positio)
pr:LT-100.I	(laitespositio + loppuliite)
gd:A1:KAAVION3	(kuvan nimi)

## Positio-osan loppuliite

Positio-osaan liitetään lisäksi usein konfigurointimoduuleissa loppuliite, joka erotetaan varsinaisesta positiotunnuksesta pisteellä.

Esim.

pr:LIC-100.F	(toimintamoduuli)
pr:LT-100.I	(tulomoduuli)
pr:LV-100.O	(lähtömoduuli)

Liitteet ”F”, ”I” ja ”O” ovat yleisesti käytetyt. Niissä voi lisäksi olla mukana numero ilmaise-  
massa esimerkiksi, monesko lähtö on kyseessä:

Esim.

pr:LV-100.O2

Kuten esimerkistä huomataan, käytettäessä laitepositioita ei loppuliite ole välttämätön yksikäsitteisen nimen aikaansaamiseksi. Mutta seuraavat seikat puolustavat tällä hetkellä loppuliitteiden käyttöä:

- Loppuliitteen avulla voi makasiinista tehdä kyselyn esim. kaikista tulomoduulien nimistä. Ilman loppuliitettä tai muuta erottavaa merkintää kysely ei onnistu.
- Konfigurointiohjelmiston tarkastustyökalu käyttää loppuliitteitä laskiessaan kuormitusarvion ja tarkastaessaan, onko samaa korttiosoitetta käytetty kahdessa eri I/O-moduulissa.

Loppuliite voi myös tulla automaattisesti erityisesti konfigurointimoduulien positio-osaan suunnittelutyökaluja käytettäessä (kts. kohta 6.10 ”Moduulit työkaluittain”).

## Positio-osassa huomioitavat seikat

Positio- ja laitepositiotunnuksissa käytetään ISOJA KIRJAIMIA.

Positiotunnukseen on hyvä liittää mahdollinen prosessialueen tai osaston tunnus, jotta nimi olisi yksikäsitteinen koko metsoDNA CR:ssä. Ota tulevaisuus huomioon!

Pidä huolta, ettei positio-osasta tule 15 merkkiä pidempi. Jos tila ei riitä, niin ota positiotunnuksesta pois vähemmän merkitseviä kirjaimia:

Esim.

LIC-100 -> LC-100  
PIA-123 -> P-123

Valvomon positiotunnusena voit käyttää koko positiota ilman prosessialue- tai osastotunnusta.

## 6.7 KUVAMODUULIEN NIMEYS

Kuvamoduulit nimetään kuvatyyppin mukaan. Moduulin nimen eteen tulee hakemistotunnus ja valvomotunnus ”yy:XX”. Konfigurointimoduuleiden hakemistotunnus kirjoitetaan pienillä kirjaimilla. Hakemistotunnus muodostuu kuvatyyppistä seuraavalla tavalla:

kaaviokuva	gd	<b>g</b> raphic <b>d</b> isplay
ryhmäkuva	gr	<b>g</b> roup <b>d</b> isplay
trendikuva	tr	<b>t</b> rend <b>d</b> isplay
reseptikuva	rp	<b>r</b> ecipe <b>d</b> isplay
piirturikuva	rd	<b>r</b> ecorder <b>d</b> isplay
askelkuva	sd	<b>s</b> tep <b>d</b> isplay
hälytyskuva	ad	<b>a</b> larm <b>d</b> isplay
ilmoituskuva	md	<b>m</b> essage <b>d</b> isplay
operointikuva	od	<b>o</b> peration <b>d</b> isplay
testikuva	td	<b>t</b> est <b>d</b> isplay
monitori-ikkuna	mw	<b>m</b> onitor <b>w</b> indow

Automaatiomoduuleilla tai -kohteilla vastaavat hakemistotunnukset merkitään isoilla kirjaimilla.

## 6.8 JÄRJESTELMÄMODUULIEN NIMEYS

Järjestelmämoduulien hakemistotunnuksena käytetään tunnusta ”sn” tai tunnusta kuvatyyppin mukaan. Seuraavassa on luettelo järjestelmämoduuleista, sijainnista ja tyyppistä:

Nimi	Sijainti	Tyyppi
sn:XX:clomod	OPS,ALP	kello
sn:XX:hmod	OPS	hierarkia
sn:XX:syscrsX	OPS	aktiviteetti
sn:XX:keycnfX	OPS	näppäimistö
sn:XX:plmod	OPS	paletti
sn:XX:menumod	OPS	valikko
sn:XX:odheader	OPS	polku
sn:XX:mdheader	OPS	yliö
sn:XX:login	OPS	käyttäjätunnus
ad:XX:ALMDISPLAY	OPS	hälytyskuva
md:XX:MSGDISPLAY	OPS	ilmoituskuva
td:XX:TDWHITE	OPS	testikuva
td:XX:TDGRID	OPS	testikuva
td:XX:TDBLACK	OPS	testikuva
td:XX:TDCARPET	OPS	testikuva
td:XX:TDCREYSCA	OPS	testikuva
td:XX:TDCS	OPS	testikuva
sn:XX:ALMLIST.F	ALP	hälytyslista
sn:XX:MSGLIST.F	ALP	ilmoituslista

sn:XX:AREA.F	ALP	aluemäärittelyt
sn:XX:HORN.F	ALP	torvimäärittelyt
sn:XX:printcon	ALP	kirjoitinmäärittelyt
sn:XX:ALMPRINT.F	ALP	kirjoittimen ohjaus

## 6.9 OPEROINTIPALVELIMEN TUOTTAMIEN MODUULIEN NIMEYS

Testauksen ja vianhaun yhteydessä on syytä tietää metsoDNA CR:n itsensä sisältämien tai tuottamien moduulien nimeyskäytäntö. Tämä asia on tullut ajankohtaiseksi vapaavalintaisten trendien ja piirturien myötä.

Operointipalvelin tuottaa metsoDNA CR:ään historia-, piirtureiden skaalaus- ja systeemi-moduuleja.

Historiamoduulit syntyvät trendikuvan ja piirtureiden skaalausmoduulit piirturikuvan valintaoperointien seurauksena. Systeemimoduulit luodaan operointipalvelimen käynnistyessä.

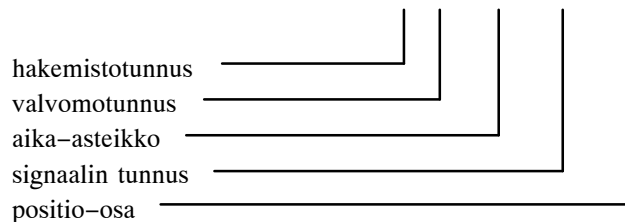
Moduulien luomiseen operointipalvelin käyttää pohjamoduuleita.

### 6.9.1 Historiamoduulien nimeys

Historiamoduulien nimi muodostuu seuraavista komponenteista:

Esim.

pt:A1:0.125h:me:FIC-100



Historiamoduulit sijoitetaan position sijainnin mukaiselle sovelluspalvelimelle.

### 6.9.2 Piirturien skaalausmoduulien nimeys

Operointipalvelimen tuottamien piirturien skaalausmoduulien nimeys ei poikkea Win-TREND-suunnittelutyökalun tuottamien moduulien nimeyksestä.

Skaalausmoduulit sijoitetaan position sijainnin mukaiselle sovelluspalvelimelle.

### 6.9.3 Systeemimoduulien nimeys

Systeemimoduulit ja niihin liittyvät tiedot varustetaan #-merkillä:

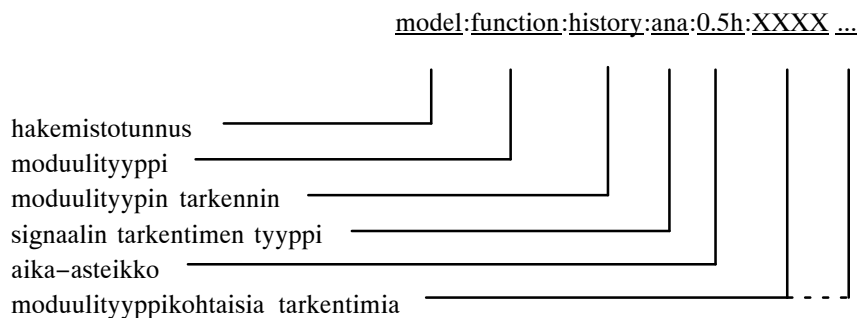
Esim.

#ce:genpos	moduulin nimi
#cp:genpos	suorasaantiportin nimi
#txt_kbd	ascii-näppäimistömoduuli

### 6.9.4 Pohjamoduulien nimeys

Pohjamoduulit ovat operointipalvelimella olevia kuva- tai toimintamoduulipohjia, joiden avulla metsoDNA CR:n sisäiset konfigurointimuutokset hoidetaan. Pohjamoduulit nimitään seuraavasti:

Esim.



Pohjamoduulit ovat osa operointipalvelimen tuotetta, joten niitä ei suunnittelija voi muuttaa.

### 6.10 MODUULIT TYÖKALUITTAIN

Seuraavassa on lueteltu työkaluittain automaatiomoduulit, -kohteet ja niistä syntyvät konfigurointimoduulit.

#### HUOM!

XX tarkoittaa valvomon tunnusta, YYYY sovelluspalvelimen tunnusta ja Z väylän tunnusta.

TYÖKALU	SYNTYVÄN MODUULIN/DOKUMENTIN NIMI	-
GdCAD	GD:XX:KAAVION3	automaatiomoduuli/kaaviokuvadokumentti
	gd:XX:KAAVION3	kaaviokuvamoduuli
	RP:XX:RESEPTIN4	automaatiomoduuli/reseptikuvadokumentti
	rp:XX:RESEPTIN4	reseptikuvamoduuli
	MW:XX:MWINDOW5	automaatiomoduuli/monitori-ikkunadokumentti
	mw:XX:MWINDOW5	monitori-ikkunamoduuli
DNAuse-Editor	gd:XX:KAAVION3	kuvamoduuli
GrCAD	GR:XX:RYHMAN2	automaatiomoduuli/ryhmäkuvadokumentti
	gr:XX:RYHMAN2	ryhmäkuvamoduuli

FbCAD	LIC-100	automaatiomoduuili/toimintokaaviodoku - menti
	pr:LIC-100.F	toimintamoduuli
	pr:LT-100.I	tulomoduuili
	pr:LV-100.O	lähtömoduuili
	il:LIC-100.F	toimintamoduuli, kaaviolamput
	il:LIC-100.O	lähtömoduuili, kaaviolamput
	ce:XX:LIC-100	positiomoduuli
	od:XX:LIC-100	operointikuvamoduuli
	al:XX:LIC-100.F	tapahtumamoduuli
	RE1_V	automaatiomoduuili/reseptin muunnelma- dokumentti
	rp:RE1.V	reseptin muunnelmamoduuli
	RE1_P	automaatiomoduuili/reseptin parametrido- kumentti
	rp:RE1.P	reseptin parametrimoduuli
	RE1_C	automaatiomoduuili/reseptin laskentado- kumentti
	rp:RE1.C	reseptin laskentamoduuli
	RE1_L	automaatiomoduuili/reseptin latausdoku- menti
	rp:RE1.L	reseptin latausmoduuili
	REP1_C	automaatiomoduuili/raportin keräys-, las- kenta- ja ohjausdokumentti
	rs:REP1.C	raportin keräys-, laskenta- ja ohjausmo- duuli
	REP1_R	automaatiomoduuili/raportin talletusdoku- menti
rs:REP1.R	raportin talletusmoduuili	
SeqCAD	SQ:SEQ-100	automaatiomoduuili/sekvenssikaaviodoku - menti
	sq:SEQ-100.F	toimintamoduuli,sekvenssi
	ce:XX:SEQ-100	positiomoduuli, pääsekvenssi
	ce:XX:SEQ-100.L	positiomoduuli, osasekvenssi
	od:XX:SEQ-100	operointimoduuli
	al:XX:SEQ-100.F	tapahtumamoduuli, pääsekvenssi
	al:XX:SEQ-100.L.F	tapahtumamoduuli, osasekvenssi
	sd:XX:SEQ-100.st	askelkuvamoduuli
	sd:XX:SEQ-100.x	askelmoduuili
CdCAD	CD:PERALAATIKKO	dokumenttimoduuli/säätökaaviodoku - menti
HwCAD	HW:50RK001	dokumenttimoduuli/laitepiirustusdoku - menti

WinTREND	TR:XX:TRENDIN4	automaatiokohde
	tr:XX:TRENDIN4_1	trendikuvamoduuli
	ph:me:LIC-100	historiamoduuli
WinTREND	RD:XX:RDDISP1_1	automaatiokohde
	rd:XX:RDDISP1_1	piirturikuvamoduuli
	rd:XX:RECGRP	piirturikuvamoduuli, ryhmänvalinta
	re:XX:01RECPEN001.F	skaalausmoduuli, piirturit
	re:XX:RECPEN001.F	toimintamoduuli, piirturit
	re:XX:RECPEN001.O	lähtömoduuli, piirturit
	re:XX:RECGRP.F	toimintamoduuli, ryhmänvalinta

## 6.11 MODUULIN SIJAINNITTIETO

Automaatio- ja konfigurointimoduuleissa sekä automaatiokohteissa käytetään sijaintitietoa ilmaisemaan, mille sovelluspalvelimelle konfigurointimoduulit metsDNA CR:ssä sijoitetaan.

Sijaintitietoa varten on varattu tila neljälle merkille, joista ensimmäisen merkin on oltava kirjain. Sijaintitiedossa sallittuja merkkejä ovat kirjaimet A...Z sekä numerot 0...9.

### 6.11.1 Sijaintitiedon rakenne

1. merkki = prosessialueen (suunnittelalueen) tunnus A...Z
2. merkki = sovelluspalvelimen tunnus A...Z tai valvomon numero 1...9
3. merkki = sovelluspalvelimen numero 0...9 tai valvomon tunnus A...Z
4. merkki = sovelluspalvelimen numero 0...9

### Sovelluspalvelinten tunnuskirjaimet

Sovelluspalvelimen nimi	Lyhenne	Tunnuskirjain	Esim.
Prosessinohjauspalvelin	PCS	P	AP01
Raportointipalvelin	REP	E	AE01
Hälytyskäsittelijä	ALP	A	A1A1,A1
Operointipalvelin	OPS	O	A1O1,A1O,A1
Diagnostiikkapalvelin	DIA	D	AD01
Varmennuspalvelin	BU	B	AB01
Damatic-liityntäpalvelin	DIS	I	AI01
Tietokone-liityntäpalvelin	CIS	C	AC01
Logiikkaliityntäpalvelin	LIS	L	AL01
Sensor-liityntäpalvelin	SIS	S	AS01
Reitityspalvelin	RTS	X	AX01



## **Sovelluspakettien nimeys operointipalvelimella ja hälytyskäsittelijällä**

Valvomon tapauksessa sijainti voi tarkoittaa yhden sovelluspalvelimen (A1O1) sijasta myös usean sovelluspalvelimen muodostamaa sovelluspalvelinryhmää (A1O). Tällöin ko. sijaintitunnuksella nimetyt moduulit menevät kaikille samaan ryhmään kuuluville sovelluspalvelimille.

Operointipalvelimella ja hälytyskäsittelijällä nimetään sovelluspalvelinryhmittäin lähetettäviä sovelluspaketteja seuraavasti:

- kuvamoduulit esim. A1O  
moduulit ladataan kaikille valvomon A1 operointipalvelimille
- positiomoduulit esim. A1  
positiomoduulit ladataan kaikille valvomon A1 operointipalvelimille sekä hälytyskäsittelijöille

## 7 VIKABITTIKÄYTÄNTÖ

metsoDNA CR:ssä muodostettavat tiedot, joita voidaan kytkeä tai joihin voidaan kytkeytyä ja siten muuttaa sovelluskonfiguraatiolla, muodostuvat pääsääntöisesti kahdesta nipussa olevasta jäsenestä: varsinaisesta perustyyppiä olevasta datasta ja vikabiteistä.

Tässä luvussa selvitetään, miten vikabitteihin tulee suhtautua.

### 7.1 VIKABITTIEN MERKITYKSET

Vikabitit kuvaavat signaalin käsittelyketjussa havaittuja vikoja. Niillä voidaan ilmaista useampaa, samanaikaisesti vaikuttavaa, eri tyyppistä vikaa. Seuraavassa on lueteltu vikabitit ja niiden lyhyet kuvaukset:

**ext**

**Ulkoinen vika (external fault):**

lähettimessä tai tulojohdossa vikaa.

**ovf**

**Ylitys/alitus (data overflow):**

tulosignaali on alittanut tai ylittänyt signaalille valitun tuloviestialueen.

**dis**

**Ei ohjattavissa (control disabled):**

signaali ei ole metsoDNA CR:n ohjattavissa, esim. lähtö on paikallisohjauksessa.

**inv**

**Kelvoton tieto (invalid data):**

signaalin arvosta ei ole mitään mittauksiin perustuvaa tietoa.

**old**

**Uudistamaton tieto (old data):**

signaalin arvo on jäädytetty.

**der**

**Johdettu vika (fault on derived data):**

kyseessä on johdettu signaali, jonka jossakin tulosuureessa on jokin vika.

**sex**

**Poikkeuksellinen tietolähde (source exceptional):**

tiedon on tuottanut jokin muu lähde kuin tavallisesti, esim. tieto on simuloitu.

### 7.2 MITÄ ERI VIKABITIT INDIKOIVAT

Toisaalta vikabiteillä kuvataan käsiteltävän arvon käyttökelpoisuutta, toisaalta taas indikoidaan tiedon kulkureitissä olevia poikkeavuuksia.

Toiset vikabitit tulkitaan vikatiloiksi, toiset taas informatiivisiksi tiloiksi. Se, onko kyseessä vika vai ”kiva tietää” –tilanne, on tapauskohtaisesti valittavaa tietoa.

Yleensä vain vioiksi tulkittavat vikabitit aiheuttavat mahdollisia jatkotoimenpiteitä.

Seuraavassa kuvataan eri vikabittien syntyä ja luetellaan toimenpiteitä, joihin niiden asettuminen voisi johtaa. Kyseessä on ohje, joten jokaisessa signaaleita käsittelevässä toiminnossa on erikseen päätettävä, mikä juuri kyseisessä tapauksessa on paras tapa reagoida eri vikabitteihin.

### 7.2.1 ext – ulkoinen vika

Syy tavallisesti:

- Liitäntäjohdot ovat poikki tai oikosulussa, tai lähtö on viritetty väärin tai se on viallinen.
- metsoDNA CR:n ulkopuolella on vika, joka vääristää metsoDNA CR:ään tulevaa viestiä tai estää metsoDNA CR:stä lähtevän ohjauksen toteutumisen.

Mikäli vika suoranaisesti vaikuttaa mitattavaan suureeseen, FBC reagoi vikaan jäädyttämällä mittusarvon vikaa edeltäneeseen arvoon ja asettaa old-vikabitin tästä merkiksi. Jos ulkopuolinen vika ei suoraan vaikuta mitattavaan suureeseen, FBC ei mittausta jäädytä. Päätelmät vian vaikutuksista mitattavaan suureeseen tehdään I/O-kortilla ja FBC:llä. Tämän takia muut toiminnot eivät mittauksissa tähän vikabittiin reagoi, ellei se muista syistä ole tarpeellista.

Takaisinluetun lähdön tapauksessa FBC ei linjaviasta takaisinluettua arvoa jäädytä, vaan arvona on se lähdön arvo, jota I/O-kortti on lähtöön kirjoittamassa. Lähdön todellista arvoa ei ulkoisen vian vuoksi tiedetä.

### 7.2.2 ovf – ylitys/alitus

Syy tavallisesti:

- Mitattava suure on ohittanut lähettimelle viritetyn alueen.

Ei yleensä tulkita viaksi.

Lähettimeltä tuleva viesti on alle 0 % tai yli 100 % valitusta viestialueesta (esim. alle 4 mA tai yli 20 mA). Signaali on tämän vuoksi epätarkka eikä sen tarkkaa arvoa pystytä ilmoittamaan, koska tulokortilla ei viestialueen ohi meneviä arvoja mitata eikä näin tiedetä, paljonko viestialue on ohitettu. Signaalin arvo on sille määritellyn skaalan ala- tai ylärajalla riippuen siitä, onko alue alitettu vai ylitetty, ellei sitä muista syistä ole jäädytetty.

Tätä vikabittiä käytetään myös esim. kun binäärisignaalin oleva aikaleima on epätarkka.

Tähän vikabittiin reagoidaan vain, jos se tarkan mittaustiedon puuttumisen vuoksi on tarpeellista.

### 7.2.3 dis – ei ohjattavissa

Syy tavallisesti:

- Lähtökortilla on signaali asetettu paikallisohjaukseen tai lähtö on FBC:llä asetettu tilaan, jossa metsoDNA CR:stä tulevaa lähdön arvoa ei viedä kentälle.
- Lähtöä ei voida metsoDNA CR:stä normaaliin tapaan ohjata, vaan signaalin kulureitti on FBC:ltä (ulkoinen ohjaus) tai I/O-kortilta (paikallisohjaus) poikki.

Vikabittiin reagoidaan, jos se ohjausmahdollisuuden puuttumisen vuoksi on tarpeellista.

### 7.2.4 inv – kelvoton tieto

Syy tavallisesti:

- Konfiguraation käynnistämisen jälkeen mittausta ei saada tai sitä ei voida hyväksyä.

Signaalista ei ole koskaan saatu mitattua kelvollista arvoa, jolloin arvona on konfiguroinnin yhteydessä annettu alkuarvo tai signaalin käyttäjille halutaan ilmoittaa, ettei signaalia saa käsitellä.

Vikabittiin reagoidaan yleensä siten, ettei signaalin arvoa käytetä hyväksi.

### 7.2.5 old – uudistumaton tieto

Signaalin tämänhetkisestä arvosta ei pystytä sanomaan mitään varmaa, koska sen arvoa ei päivitetä signaalin syntypaikalla muodostuvalla arvolla. Mikä on signaalin näytettävyyden old:in ollessa voimassa riippuu siitä, mistä tilanteesta tähän on jouduttu: onko ennen jäädyttämistä signaaliin saatu päivittämällä virheetön vai virheellinen arvo vai onko sitä päivitetty kertaakaan.

Vikabittiin reagoidaan vain, jos se signaalin päivittämättömyyden tai sen käyttökelpoisuuden vuoksi on tarpeellista.

### 7.2.6 der – johdettu vika

Signaali on johdettu muista signaaleista ja jossakin niistä on ollut vikabitti inv, old tai der asettuneena.

### 7.2.7 sex – poikkeuksellinen tietolähde

#### Mittaus

Mittauksen arvo ei tule lähettimeltä, vaan signaalin kulkureitti on katkaistu FBC:ltä. Mittauksen arvona on tulon simulointitiedossa annettu arvo.

#### Lähtö

Takaisinluetun lähdön arvo ei tule I/O-kortilta, vaan I/O-moduulista palautetaan arvona I/O-moduuliin viimeksi kirjoitetun lähdön arvo.

Myös muualla kuin FBC:llä voidaan tulo/lähtö asettaa simulointitilaan.

Ei yleensä tulkita viaksi.

Vikabittiin reagoidaan vain, jos se kulkureitin katkaisun vuoksi on tarpeellista.

Kun mittausta simuloidaan, simuloidaan samalla vikabittikin ts. simulointipisteestä ylöspäin vikatietoina annetaan käyttäjän antamat vikabitit. Vikabiteistä ei simuloiduissa mittauksissa pysty päättämään todellisen mittauksen käyttökelpoisuutta simulointipisteestä alaspäin.

## 7.3 VIKABITEISTÄ SIGNAALIHÄLYTYKSIÄ

Eräissä tietotyypeissä on varauduttu signaalihälytykseen. Se asetetaan, kun signaalissa on vikaa.

Signaalihälytystä voidaan sovellutuksessa käyttää, kun operaattoreille halutaan ilmoittaa sovellutuksen liitynnöissä havaituista vioista.

Pääsääntönä on, että signaalihälytys asetetaan, jos jokin vikabiteistä ext, inv, old tai der on asettunut. ovf, dis ja sex eivät aseta signaalihälytystä.

## 7.4 VIKABITTIEN KÄYTÖSSÄ HUOMIOITAVIA SEIKKOJA

Vikabittin inv esiintyessä yksinään tiedon arvoa ei valvomossa piirretä näyttölaitteille. Jos inv:in lisäksi on jokin muu vikabitti asettunut, arvo näytetään.

## 7.5 VIKABITTIKÄYTÄNNÖN SOVELTAMISESTA

### 7.5.1 Tyypien alkuarvot

Tyypissä määritelty alkuarvo tulee voimaan, kun soveltaja ei ole moduuliin määritellyt alkuarvoa eikä hän ole kytkenyt tietoa tai kytkentä ei ole voimassa.

### 7.5.2 Alkuarvot moduuleissa

Soveltajan moduuliin antamat alkuarvot jäävät voimaan, kun hän ei ole kytkenyt tietoa moduulin ulkopuolelle.

Jos tieto on kytketty, soveltajan antamat vikabitit ovat voimassa moduulin käynnistyessä sen aikaa, minkä kytkennän voimaan saattaminen kestää, kuitenkin seuraavassa luvussa mainituin poikkeuksin.

Jos tietoon kuuluu vikabittejä, on soveltajan aina annettava sekä tiedon arvo että vikabitit.

### 7.5.3 Kytkettävät tiedot

Niiden tietojen osalta, jotka soveltaja on kytkenyt moduulin ulkopuolelle, metsoDNA CR toimii seuraavasti:

Jos kytkentää ei saada moduulin käynnistyessä voimaan, asetetaan vikabitti old. Tällöin muut vikabitit jäävät siihen tilaan, mihin soveltaja oli ne asettanut, tai jos hän ei ole antanut alkuarvoja, muut vikabitit jäävät tyypissä määriteltyihin alkuarvoihin.

Kun kytkentä saadaan tehtyä, vikabitit päivittyvät kytkeydyttävän tiedon mukana.

Jos kytkentä katkeaa, asetetaan vikabitti old. Tällöin muut vikabitit jäävät siihen tilaan, missä ne olivat viimeksi kytkennän ollessa voimassa.



# ALKEISTYYPIT

## PERUSTYYPIT

TYYPPI	MERKITYS	LUKUALUE	KOKO TAVUINA
char	etumerkitön merkkimuuttuja	merkki	1
int8	etumerkillinen kokonaisluku	-128...+127	1
int16	etumerkillinen kokonaisluku	-32768...-32767	2
int32	etumerkillinen kokonaisluku	-2147483648... +2147483647	4
uns8	etumerkitön kokonaisluku	0...255	1
uns16	etumerkitön kokonaisluku	0...65535	2
uns32	etumerkitön kokonaisluku	0... 4294967295	4
bool8	Boolean-muuttuja, arvo on epätosi, jos muuttujan kaikki bitit ovat tilassa epätosi. Arvo on tosi, jos muuttujan yksikin bitti on tilassa tosi.	tosi/epätosi	1
bool16	Boolean-muuttuja, arvo on epätosi, jos muuttujan kaikki bitit ovat tilassa epätosi. Arvo on tosi, jos muuttujan yksikin bitti on tilassa tosi.	tosi/epätosi	2
float	yksikertaisen tarkkuuden liukuluku	$\left\{ \begin{array}{l} \text{MIN-} = -2.71050 * 10^{-20} \\ \text{MIN+} = -5.42101 * 10^{-20} \\ \text{MAX} = 9.22337 * 10^{18} \end{array} \right.$	4

**JOHDETUT TYYPIT****fails, tyyppiä uns16**

- Tyypin fails sisältää 16 bittiä (b0...b15), 16 erillistä loogista tietoa, joihin voidaan viitata automaatiokielen merkintätapaa käyttäen.
- Bittien symboliset nimet ja merkitykset ovat:

BITTI	NIMI	MERKITYS
b0	b	≡ 0 (muuttujan arvo johdetussa tyypissä bin)
b1	ext	lähettimen syöttöhäiriö tai linjavika (external)
b2	ovf	tulosignaali rajojen ulkopuolella (overflow)
b3	dis	signaali ei ole ohjattavissa (control disabled)
b4	inv	epäluotettava tieto (invalid)
b5	old	uudistumaton tieto (old)
b6	der	johdettu vika (derived)
b7	sex	poikkeuksellinen tietolähde (source exceptional)
b8	ei käytössä	
b9	”	
b10	”	
b11	”	
b12	”	
b13	”	
b14	”	
b15	”	

- Esim.: Jos muuttuja paine on rakenteista tyyppiä ana, on paine:f:inv sen vikabitti epäluotettava tieto.

**bin, tyyppiä uns16**

- Tyypin bin sisältää 16 bittiä, 16 erillistä loogista tietoa. Näistä bitti b0 on tyypin bin arvo. Viitattaessa automaatiokielellä tyyppiä bin olevaan muuttujaan viitataan juuri tähän arvoon. Bitit b1 – b15 ovat tyypin bin vikatietoja, jotka on määritelty tyypin fails yhteydessä.
- Automaatiokielellä voidaan käsitellä vikatietoja erillisinä loogisina kokonaisuuksina. Esim., jos muuttuja lim on tyyppiä bin, on lim:b sen arvo ja lim:der sen vikabitti johdettu vika.



## YLEISET RAKENTEISET TYYPIT

Tyyppi	Merkitys	Jäsenet	Jäsenten merkitys	Jäsenten tyypit
ana	analogiasignaali	f a	vikabitit arvo	fails float
ints	lyhyt kokonaisluku	f s	vikabitit arvo	fails int16
intl	pitkä kokonaisluku	f l	vikabitit arvo	fails int32
time	kellonaika ja päivämäärä	hundus tenms sec min hour wday day month year vers	100 mikrosekunnit 10 millisekunnit sekunnit minuutit tunnit viikonpäivä päivä kuukausi vuosi ajan versio (b7 – b5) ja aikavyöhyke (b4 – b0)	uns8 uns8 uns8 uns8 uns8 uns8 uns8 uns8 uns8 uns8
bo	binäärilähtö	bv pw	binäärilähdön tila binäärilähdön pulssin pituus	bin uns16
binev	aikaleimattu binäärisignaali	binstat bintime	arvo aikaleima	bin time
anaev	aikaleimattu analogiasignaali	anastat anatime	arvo aikaleima	ana time
intsev	aikaleimattu lyhyt kokonaisluku	intsstat intstime	arvo aikaleima	ints time
intllev	aikaleimattu pitkä kokonaisluku	intlstat intltime	arvo aikaleima	intl time



# GRAFIKKAKIELEN KOMENNOT

## Automaatiokielen grafiikkaosuus

Automaatiokieleen kuuluu erillinen lausekemuotoinen grafiikkakieli, jolla muodostetaan staattisia grafiikkakuvia eli kuvapohjia. Kuvapohjat yhdessä päivittyvien näyttötietojen kanssa muodostavat kuvamoduulin, joka saa aikaan kuvan valvomon monitorille.

Kaaviokuvat luodaan graafisesti GdCAD- tai DNAuseEditor-suunnittelutyökalulla.

## Grafiikkakuvan koordinaatit

metsoDNA CR:ssä grafiikkakuva tulostetaan kuva-alueelle, jonka resoluutio on x-suunnassa 640 pikseliä ja y-suunnassa 480 pikseliä.

Grafiikkageneraattori tulostaa tämän koordinaattialueelle, joka on kuvattu alla olevassa kuvassa.

(0, 959)

(639, 959)



(0, 480)

(639, 480)

Koordinaattialue on siis hiukan tavallisesta (0, 0) origokeskeisestä koordinaatistosta poikkeava. Kuvan vasen alalaita sijaitsee koordinaateissa (0, 480).

Tämän vuoksi onkin tärkeää, että konfiguroija muistaa sijoittaa kuvansa sallitulle kuva-alueelle. Suunnittelijan muodostamat kuvat eivät saa ylittää kuva-alueen koordinaatteja!

## **Grafiikkakielen komennot**

Grafiikkakielen komennot ohjaavat grafiikkatulostusta joko hallinnollisesti (esim. kuvien määräytyminen) tai määrittäen eri ominaisuuksia (miten viiva piirtyy) tai ns. suorilla grafiikkakomennoilla (muodosta ympyrä).

Grafiikkakielen syntaksi vaatii, että kaikki komennot kirjoitetaan suuraakkosin.

Esimerkki:

```
YMPYRÄ S 30 30
```

Tiettyjen komentojen parametrit annetaan absoluuttisena (A) tai suhteutettuna (S) senhetkiseen piirtopisteeseen näytöllä.

Esimerkki:

Komento

```
LIIKU A 30 30
```

siirtää piirtopisteen kohtaan (30, 30), kun taas komento

```
LIIKU S 30 30
```

siirtää piirtopisteen tämänhetkisestä kohdasta 30 pikseliä x-suuntaan ja 30 pikseliä y-suuntaan.

Tietyt kuvat voidaan tulostaa valitulla värillä täytettynä (T).

Esimerkki:

Komento

```
YMPYRÄ S T 20 20
```

muodostaa täytetyn ympyrän.

## ***KOMENNOT***

Seuraavaksi esitellään grafiikkakielen tuntemat komennot esimerkkeineen. Esimerkkien yhteydessä käytettyjä merkintöjä:

$x \hat{=}$  piirtopiste alkutilanteessa

$o \hat{=}$  piirtopiste lopputilanteessa

- KUVAILE KUVA <kuvanimi>

Grafiikkakielinen kuva alkaa aina tällä hallinnollisella komennolla.

- LOPETA KUVA

Tällä komennolla lopetetaan kuva. Myös tämä komento on hallinnollinen komento.

- KUVAILE MAKRO <makronimi>

Makron määrittelyn aloittava komento. Makrolle annetaan tässä yhteydessä myös nimi. Makron sisälle voidaan sijoittaa suhteellisia piirtokomentoja sekä ominaisuskomentoja.

- LOPETA MAKRO

Makron lopetuskomento. Makron määrittely on valmis. Seuraavassa esimerkki makron määrittelemisestä:

```
KUVAILE MAKRO motor_on
```

```
VÄRI 7
```

```
YMPYRÄ S T 20 20
```

```
TEKSTI "M"
```

```
LOPETA MAKRO
```

- VÄRI <0 – 15>  
Valitaan väripaletista piirtoväri. Oletusarvona on taustalle väri 0 ja kuvioille sekä tekstile väri 15.  
Oletusarvopalettei  
0  $\hat{=}$  taustaväri  
1  $\hat{=}$  vaaleanruskea, tekstiväri  
2  $\hat{=}$  tummanruskea, kehysväri  
3  $\hat{=}$  punainen, vilkkuväri  
4  $\hat{=}$  punainen, staattinen  
5  $\hat{=}$  oranssi, vilkkuväri  
6  $\hat{=}$  oranssi, staattinen  
7  $\hat{=}$  keltavihreä, päällä  
8  $\hat{=}$  tummanvihreä  
9  $\hat{=}$  tummansininen  
10  $\hat{=}$  vaaleansininen, cyan  
11  $\hat{=}$  keltainen  
12  $\hat{=}$  violetti, vikaväri  
13  $\hat{=}$  harmaa, ilmoitusväri  
14  $\hat{=}$  valkoinen, huomio  
15  $\hat{=}$  valkoinen, huomiovilkku
- TEKSTIKOKO <0 – 3>  
Valitaan tekstikoko numeroilla 0 – 3  
0 = pieni tekstikoko  
1 = pieni, korkea tekstikoko  
2 = suuri tekstikoko  
3 = suuri, leveä tekstikoko
- PINTAKUVIO  
Määrää pintakuvion, jolla täytettävä kuvio täyttyy. Pintakuvio 0 (= yhtenäinen pintakuvio) on ainoa valittavissa oleva pintakuvio ja siten myös oletusarvo.
- VIIVATYYPPI <0 – 3>  
Valitaan viivatyyppi, jolla viivoja ja muita grafiikkaprimitiivejä piirretään. Oletusarvona on viivatyyppi 0.  
Numeroiden merkitykset ovat seuraavat:  
0 = \_\_\_\_\_  
1 = .....  
2 = -----  
3 = .\_.\_..
- VIIVAPAKSUUS <dx dy>  
Määrittelee piirrettävien grafiikkaviivojen paksuuden x- ja y-koordinaatteina. Vaikuttaa myös pintakuvioiden leveyteen. Oletusarvona on viivapaksuus 1 1 eli viivan leveys on yksi pikseli molempiin suuntiin.
- KEHYS ON...EI  
Määrittelee täytettyjen kuvioiden kehysten. Tämä näkyy mm. täytettyjen suorakaiden reunoissa korostuksena. ON asettaa kehysten ja EI poistaa.

#### 4 Grafiikkakielen komennot

- ”KOMMENTTITEKSTI”

Grafiikkakielisten komentojen väliin voidaan kirjoittaa myös kuvaa kommentoivaa tekstiä. Aloitetaan ”-merkillä ja lopetetaan samalla merkillä.

Esimerkki:

”Tämä on esimerkkikuva”

- **LIIKU A <x y>**

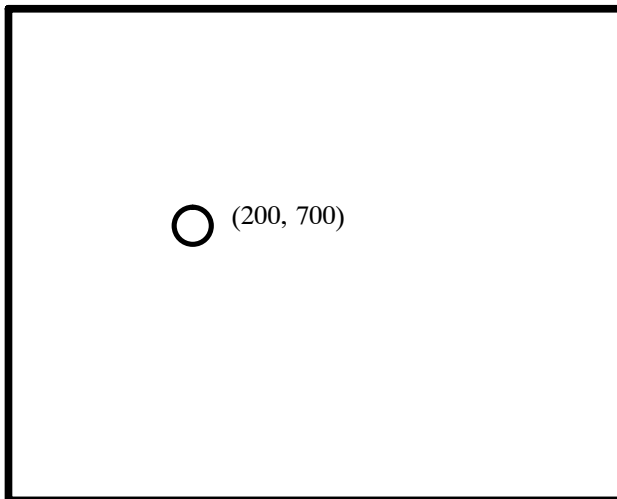
Liikutaan absoluuttisesti <x y> -koordinaattiparin määräämään pisteeseen.

Esimerkki:

LIIKU A 200 700

(0, 959)

(639, 959)



(0, 480)

(639, 480)

- **LIIKU S <dx dy>**

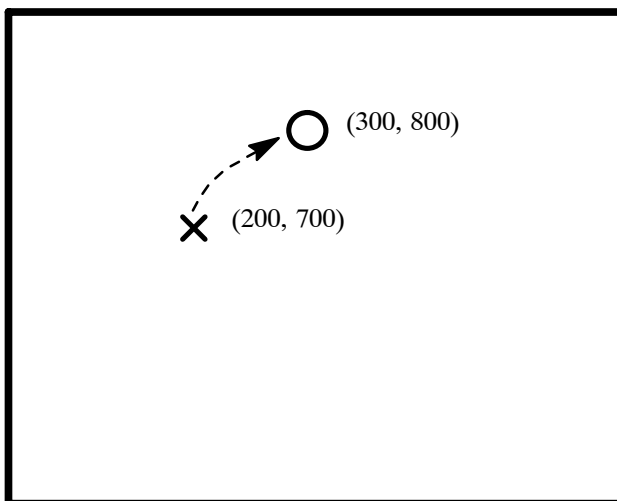
Liikutaan suhteellisesti nykyisestä piirtopisteestä siirroksen <dx dy> verran.

Esimerkki:

LIIKU S 100 100

(0, 959)

(639, 959)



(0, 480)

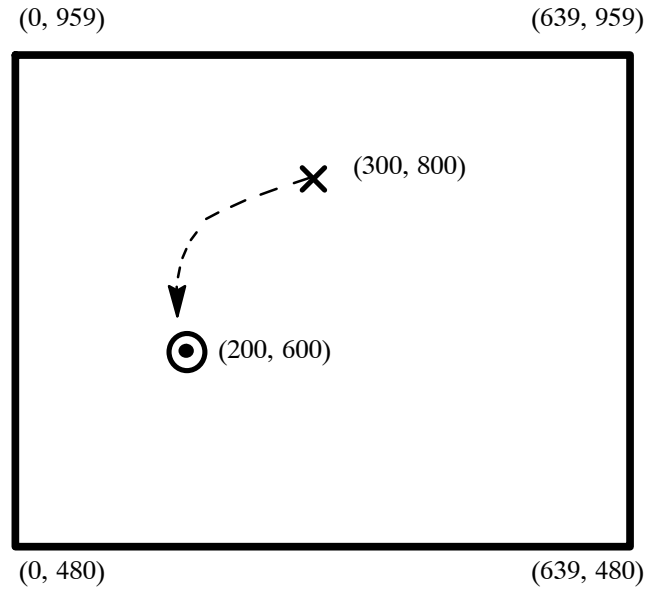
(639, 480)

- PISTE S <dx dy>

Liikutaan suhteellisesti nykyisestä piirtopisteestä siirroksen <dx dy> verran ja piirretään piste. Piirtopiste jää tähän pisteeseen.

Esimerkki:

PISTE S -100 -200

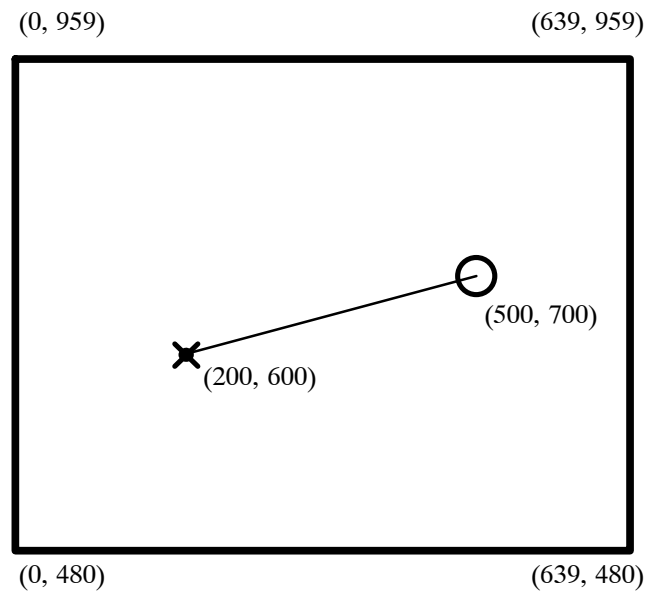


- VIIVA S <dx dy>

Piirretään suhteellisesti nykyisestä piirtopisteestä alkaen viivaa siirroksen <dx dy> verran. Piirtopiste jää viivan loppupisteeseen.

Esimerkki:

VIIVA S 300 100

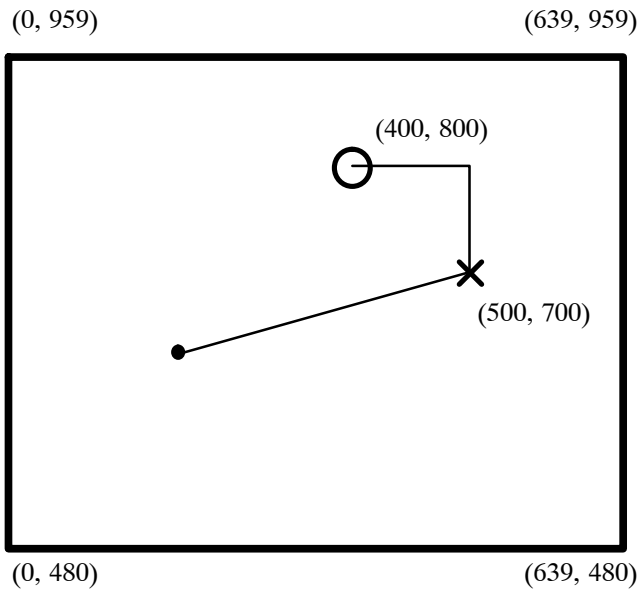


- MURTOVIIVA S <dx dy> <dx dy>...

Piirretään suhteellisesti nykyisestä piirtopisteestä alkaen murtoviivaa siirroksia <dx dy>... noudattaen. Piirtopiste jää murtoviivan loppupisteeseen.

Esimerkki:

MURTOVIIVA S 0 100 -100 0



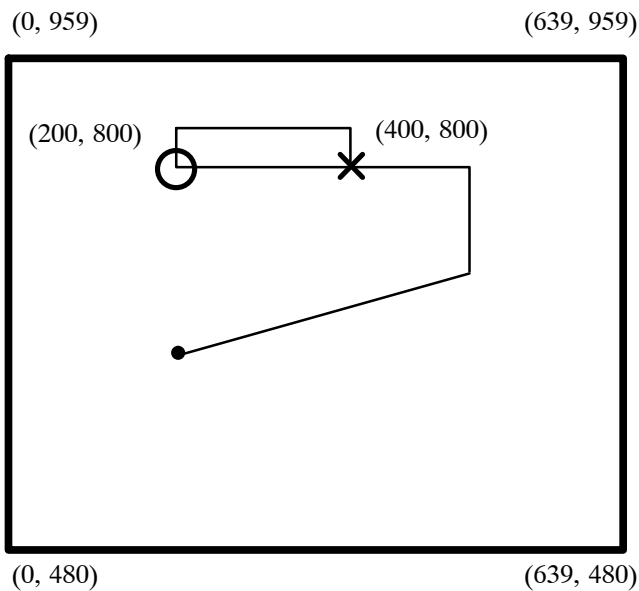
- SUORAKAIDE S <dx dy>

Piirretään suorakaide. Suorakaide määritellään kahden nurkkapisteen avulla. Yksi nurkka on nykyinen piirtopiste ja toinen nurkka määritellään suhteellisenä edellisestä nurkkapistestä.

Loppupiste jää samalle vaakasuoralle kuin alkupiste, mutta vastakkaiseen päähän suoraa.

Esimerkki:

SUORAKAIDE S -200 50





- **SUORAKAIDE S T <dx dy>**  
Piirretään täytetty suorakaide. Suorakaide määritellään kahden nurkkapisteen avulla. Yksi nurkka on nykyinen piirtopiste ja toinen nurkka määritellään suhteellisenä siirroksena edellisestä nurkkapistestä. Loppupiste jää samalle vaakasuoralle kuin alkupiste, mutta vastakkaiseen päähän suoraa.

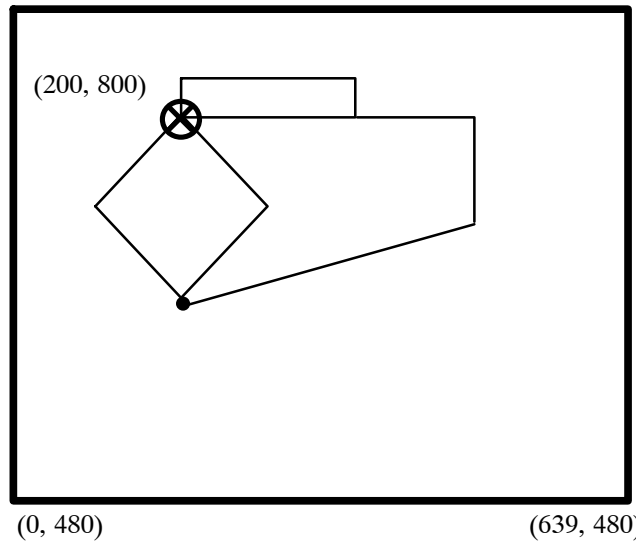
- **MONIKULMIO S <dx dy> <dx dy>...**  
Piirretään monikulmio. Piirtäminen aloitetaan nykyisestä piirtopisteestä edeten viivaa piirtäen siirroksien mukaisesti. Monikulmio saadaan tulokseksi, kun viimeisestä pisteestä piirretään viiva alkupisteeseen, johon loppupiste siis jää.

Esimerkki:

MONIKULMIO S -100 -100 100 -100 100 100 -100 100

(0, 959)

(639, 959)



- **MONIKULMIO S T <dx dy> <dx dy>...**  
Piirretään täytetty monikulmio. Piirtäminen aloitetaan nykyisestä piirtopisteestä edeten viivaa piirtäen siirroksien mukaisesti. Monikulmio saadaan tulokseksi, kun viimeisestä pisteestä piirretään viiva alkupisteeseen.

- KAARI S <dx dy> <dx dy>

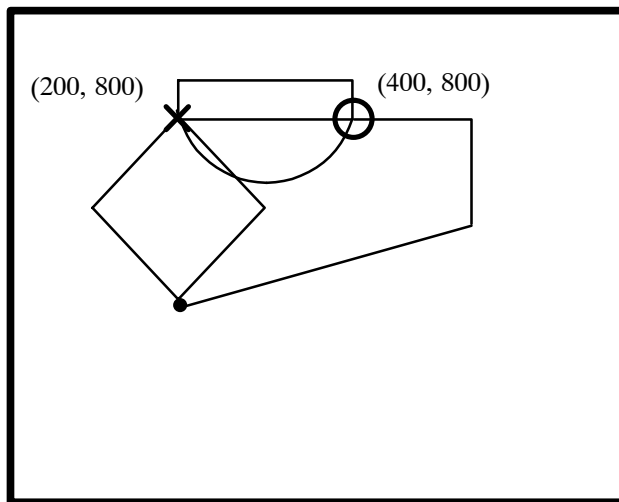
Piirretään kaari alkaen nykyisestä piirtopisteestä. Kaari määritellään kolmen pisteen avulla. Ensimmäinen piste on nykyinen piste, toinen on ensimmäisen siirroksen määrittämä piste ja loppupiste on seuraavan siirroksen määrittämä.

Esimerkki:

```
KAARI S 100 -50 100 50
```

(0, 959)

(639, 959)



(0, 480)

(639, 480)

- KAARI S T <dx dy> <dx dy>

Piirretään täytetty kaari alkaen nykyisestä piirtopisteestä. Kaari määritellään kolmen pisteen avulla. Ensimmäinen piste on nykyinen piste, toinen on ensimmäisen siirroksen määrittämä piste ja loppupiste on seuraavan siirroksen määrittämä.

- YMPYRÄ S <dx dy>

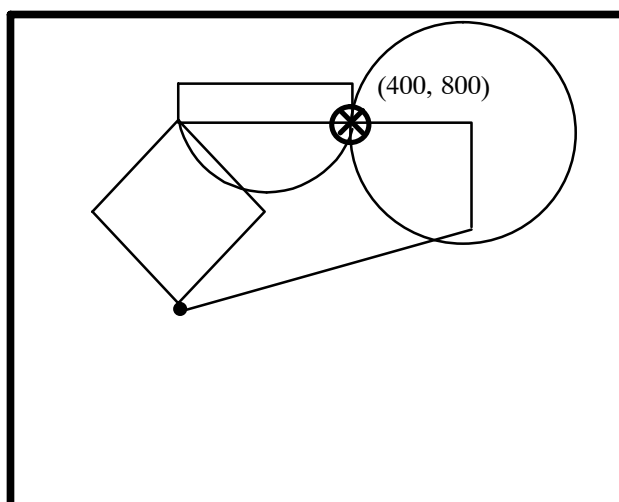
Piirretään ympyrä, jonka halkaisija määräytyy nykyisen piirtopisteen ja syötetyn siirroksen mukaisesti. Piirtopiste jää alkuperäiseen pisteeseen.

Esimerkki:

```
YMPYRÄ S 200 0
```

(0, 959)

(639, 959)



(0, 480)

(639, 480)

- **YMPYRÄ S T <dx dy>**  
Piirretään täytetty ympyrä, jonka halkaisija määräytyy nykyisen piirtopisteen ja syötetyn siirroksen mukaisesti. Piirtopiste jää alkuperäiseen pisteeseen.
- **TEKSTI ”Tulostuva teksti”**  
Piirretään tekstiä alkaen nykyisestä piirtopisteestä. Piirtopiste jää viimeisen merkin jälkeiseen merkkiposition alkupisteeseen.  
ROMAN8–merkkitaulukon merkkejä voidaan käyttää CWS–ympäristössä. Tietyt erikoismerkit voidaan korvata kansallisilla merkeillä.  
Kaikissa paikoissa ei kansallisia merkkejä voi käyttää. Sellaisissa tapauksissa saadaan ilmoitus kielletyistä merkeistä.  
Esimerkiksi USASCII – suomi –merkistön muunnos on seuraava:  

[	->	Ä
{	->	ä
\	->	Ö
	->	ö
]	->	Å
}	->	å
- **MAKRO <makronimi>**  
Piirretään määritelty makro. Makro on määriteltävä samassa kuvataustassa. Piirtopiste liikkuu makron grafiikkakomentojen mukaisesti.  
Esimerkki:  
MAKRO motor\_on

## Grafiikkaa sisältävien kuvien konfigurointi

Kuvamoduuli koostuu staattisesta kuvapohjasta ja päivittyvistä sekä operoitavista kohteista. Yksi kuvamoduuli muodostaa yhden valvomon monitorilla näkyvän kuvan.

Grafiikkaa voidaan syöttää seuraaville moduulityypeille:

- kuva
- yliö
- hierarkia
- operointi
- ikkuna
- valikko

Grafiikkaa sisältävissä kuvamoduuleissa on normaaliin automaatiokielen tapaan kolme osaa:

- hallintaosa
- esittelyosa
- toimintaosa

Moduulin hallintaosa on normaali.

Esittelyosan tietopisteeksi määritellään muodostettava kuvatausta. Kuvataustan luonti tehdään erityisellä grafiikkakielellä, joka on kuvattu edellä.

Kuvataustamäärittelyn aluksi on moduulissa oltava ”GRAFIKKA”-merkkijono. Grafiikan kuvailu alkaa komennolla KUVAILU KUVA <kuvan nimi> ja päättyy komennolla LOPETA KUVA.

Toimintaosassa on grafiikkatyyppejä ”draw”, jonka ainoaan jäsenen ”nap” kuvatausta kytetään.

Esimerkki moduulin rakenteesta:

```
HALLINTAOSA
.
.
.
ESITTELYOSA
  ULKOISET
.
.
.
  PAIKALLISET
.
.
.
  GRAFIIKKA
    KUVAILE KUVA esimerkki
      LIIKU A 100 700
      VIIVA S 20 30
      SUORAKAIDE S T 40 60
.
.
.
      LOPETA KUVA
    ;
TOIMINTAOSA
.
.
.
  ldraw
    nap < esimerkki
  ;
.
.
.
LOPPU
```