# SKS SERVOMOOTTOREIDEN VALVOMO-OHJELMISTO

Toteutus LabView-ohjelmalla

# SKS UNIDRIVE CONTROL SYSTEM

Made with LabView-software

Teemu Järvenpää

Opinnäytetyö
Huhtikuu 2011
Sähkötekniikan koulutusohjelma
Sähkövoimatekniikka
Tampereen ammattikorkeakoulu

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

**TIIVISTELMÄ**

Työnä oli luoda valvomo tyyppinen ohjelmisto SKS Control Techniquesin servomoottorikäytölle. Ohjelman avulla voidaan monitoroida käyttöä, sekä ajaa itse määrättyjä ajoprofiileita. Ohjelman pitää myös kerätä tietoa talteen servomoottorikäytön taajuusmuuttajalta, sekä pystyä siirtämään kerätty tieto Excel-ohjelmaan. Ohjelma toteutettiin LabView-ohjelmalla.

Tuloksena syntyi valvomo-ohjelmisto, jonka avulla voidaan monitoroida SKS:n servomoottorikäyttöä sekä ajaa oma ajoprofiili laitteistolle. Lisäksi taajuusmuuttajan parametrejä voidaan muokata suoraan ohjelman välityksellä.

Työ toteutettiin Tampereen ammattikorkeakoulun sähkö laboratoriossa.

**ABSTRACT**

The assignment of this project was to create a control system for SKS Control Techniques servo drive. The control system should be such that it can be used to monitor drive values and to drive manually created drive profiles. The program needs to be able to collect data from a servo drive and to be able to transfer the collected data to a third party program as Excel. The program was created by using LabView-software.

As a result a working control system was created. The system is able to monitor drive values and to drive manually created drive profiles. In addition the program is able to configure drive parameters.

This project was executed in the electric laboratory at Tampere polytechnics.

## ALKUSANAT

Tahdon kiittää perhettäni heidän pyyteettömästä tuestaan ja siitä että jaksoivat kannustaa loppuun asti. Tahdon kiittää myös avovaimoani hänen kärsivällisyydestään ja tuestaan työn eri vaiheiden läpi. Erityiset kiitokset kuuluvat työn valvojalle Lauri Hietalahdelle, joka ei auttanut liikaa vaan pisti minut ajattelemaan ja syventymään työhön ja todella oppimaan asian.

Tampereella

21. huhtikuuta 2011

Teemu Järvenpää

**SISÄLLYSLUETTELO**

SISÄLLYSLUETTELO ..... 5
1 JOHDANTO ..... 8
2 TOIMILAITTEET ..... 9
2.1 Taajuusmuuttaja ..... 9
2.1.1 SKS Control Techniques UNI2402-taajuusmuutaja ..... 9
2.2 Moottori ..... 11
2.3 LabView ..... 11
3 YHTEYSPROTOKOLLA ..... 13
3.1 Tiedonsiirto ..... 14
3.1.1 Kehys ..... 14
3.1.2 ASCII ..... 14
3.1.3 Ohjaus merkit ..... 15
3.1.4 Osoitteet ..... 16
3.1.5 3.1.5 Parametrit ja arvot ..... 16
3.1.6 Tarkistussumma (BCC) ..... 16
3.2 Lukeminen, lähetys ja vastaanotto ..... 17
3.2.1 Parametrin lukeminen ..... 17
3.2.2 Parametrin kirjoitus ..... 18
3.2.3 Tarkistussumman laskeminen ..... 18
3.3 Lukeminen, lähetys ja vastaanotto LabView-ohjelmalla ..... 19
3.3.1 Vastaanotetun tiedon muuttaminen ymmärrettävään muotoon ..... 21
4 TAAJUUSMUUTTAJAN SISÄINEN OHJELMA ..... 23
4.1 4.1 Muutokset ..... 23
5 TYÖN TOTEUTUS ..... 24
5.1 Yhteyden muodostus ..... 24

5.2 Pääkäyttöliittymä ..... 26
5.3 Parametrit ..... 29
5.3.1 Parametrien kirjoitus ..... 30
5.4 Monitorointi tila ..... 31
5.5 Ajoarvojen asetus ..... 32
5.6 Aja ajoarvot ..... 33
6 LOPPUSANAT JA PARANNUS EHDOTUKSET ..... 35
LIITE 1 ..... 39
LIITE 2 ..... 49

## LYHENTEET

| | |
|---|---|
| AC | Alternating current, vaihtovirta |
| DC | Direct current, tasavirta |
| ANSI | American National Standards Institute |
| LabView | **Lab**oratory **v**irtual **i**nstrumentation **e**ngineering **w**orkbench |
| VI | Virtuaali instrumentti |
| EIA | Electronics Industries Association |
| Baud rate | Kellojaksoa sekunnissa. |
| Parity | Tarkistusbitti, jonka avulla ykkösten määrä tavussa asetetaan joko parilliseksi tai parittomaksi |
| ASCII | American Standard Code for Information Interchange |
| LSB | Least significant bit |
| MSB | Most significant bit |
| BCC | Tarkistus summa merkki (block check character) |
| RW | Read/Write |
| RO | Read only |

# 1 JOHDANTO

Tämä opinnäytetyö on kehitetty Tampereen ammattikorkeakoulun sähkötekniikan laboratorion opetustyössä käytettyihin servomoottorikäyttöihin.

Työn tavoitteena on kehittää taajuusmuuttajakäytön ohjausjärjestelmä, jossa voidaan PC:n avulla ohjata kuormitusservomoottoreita, sekä monitoroida kuormituskäytön tilaa ja suureita. Järjestelmä toteutetaan LabView-ohjelmistolla.

Ohjausjärjestelmän tulee olla valvomotyyppinen ohjelmisto, jonka käyttöliittymänä on ohjauspaneeli.

Työ toteutettiin SKS Control Techniques:n taajuusmuuttajaan.

# 2 TOIMILAITTEET

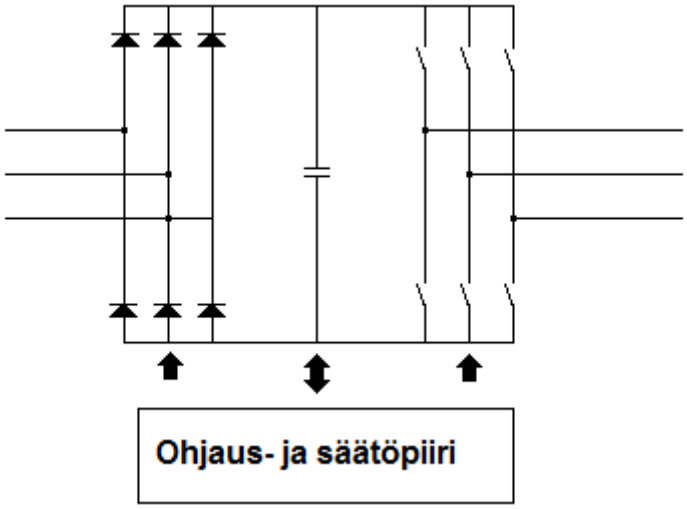
Luvussa 2. esitellään työssä käytetyt toimilaitteet.

## 2.1 Taajuusmuuttaja

Taajuusmuuttajan avulla voidaan säätää tarkasti moottorin nopeutta ja momenttia.

Yleisin taajuusmuuttajatyyppi on kolmivaiheinen jännitevälipiirillinen taajuusmuuttaja. Jännitevälipiirillisen taajuusmuuttajan tehopuolijohteet muodostavat kuuden kytkimen vaihtosuuntaussillan, jonka avulla moottorin vaiheet voidaan kytkeä joko positiiviseen tai negatiiviseen tasajännitepotentiaaliin.

Vaihtosuuntaus toteutetaan nykyään yleisimmin IGBT-transistoripareja käyttäen. Yksi komponentti kustakin parista johtaa kerrallaan nopeassa tahdissa käyttäen pulssinleveysmodulaatiota. Syntynyt jännite on muodoltaan kanttiaaltomaista. Suodattamalla jännitteestä saadaan sinimuotoista jännitettä.



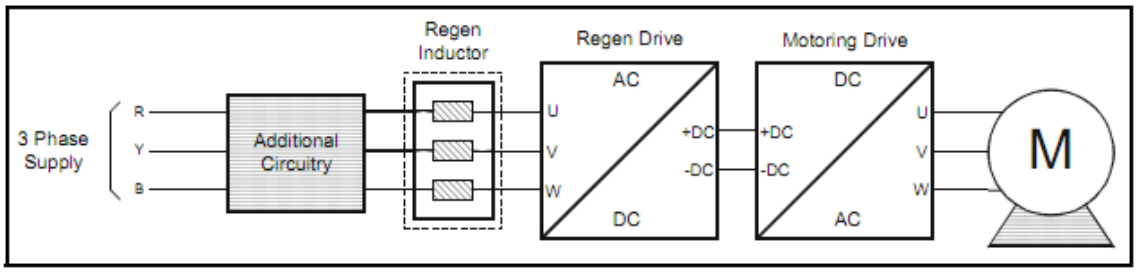

**Kuva 2.1: Taajuusmuuttajan peruskytkentä**

Tässä työssä käytettiin SKS Control Techniques:n UNI2402- taajuusmuuttajia.

### 2.1.1 SKS Control Techniques UNI2402-taajuusmuutaja

UNI2402-taajuusmuuttajalla voidaan ohjata sekä oikosulkumoottoreita, että servomoottoreita.

**Kuva 2.2: UNI2402-Taajuusmuuttaja**




**Kuva 2.3: SKS lohkokaavio**

Laitetta voidaan ohjata suoraan taajuusmuuttajan etupaneelista, servokäytön ohjauspulpetista tai sarjakaapeli liitännän kautta ANSI-protokollan avulla. Seuraavassa taulukossa (Taulukko 2.1) on esitetty taajuusmuuttajan tulot ja lähdöt. Taajuusmuuttajaan on saatavilla seuraavat kenttäväyläsovittimet lisävarusteena: PROFIBUS-DP, InterBus S, CTNET, ModBus Plus ja DeviceNet.

**Taulukko 2.1: Taajuusmuuttajan datasiirtomahdollisuudet /4/**

| Function | Qty | Programmability | Terminals |
|---|---|---|---|
| Differential analog input | 1 | Destination, mode, scaling, | 5,6 |
| Single ended analog input | 2 | Destination, mode, scaling, | 7,8 |
| Analog output | 2 | Source, mode, scaling, | 9,10 |
| Digital input | 3 | Destination, mode, | 27,28,29 |
| Digital input / outputs | 3 | Destination / source, mode | 24,25,26 |
| Relay | 1 | Source | 1,2 |
| Drive enable | 1 | External trip (latching) or inhibit (non latching) | 30 |
| 10V supply | 1 | | 4 |
| 24V supply | 1 | | 22 |
| 0V analog | 2 | | 3,11 |
| 0V digital | 2 | | 21,23,31 |

## 2.2 Moottori

Työssä käytetään SKS:n harjatonta AC-servomoottoria.

**Servomoottoreiksi** kutsutaan takaisinkytkettyjä ja säädettyjä moottoriratkaisuja. Servomoottoreita on saatavilla sekä AC- että DC-moottoreina.

Työssä käytetty moottori on harjaton AC-servomoottori, joka on kestomagnetoitu tahtimoottori.

Tahtimoottorissa, roottori pyörii staattorin synnyttämän pyörivän magneettikentän kanssa tarkalleen samalla nopeudella. Periaatteessa tahtimoottorin staattorin rakenne on samanlainen kuin epätahtikoneessa, mutta roottorin rakenne on erilainen. Kyseisessä tahtimoottorissa roottorin magneettikenttä synnytetään kestomagneettien avulla.

SKS:n servomoottorin arvot on lueteltu alla olevassa taulukossa.

**Taulukko 2.2: Servomoottorin arvot**

| | |
|---|---|
| Tyyppi | 142UME300 |
| Nimellisnopeus | 3000 rpm |
| 0-nopeusvirta | 14,63 |
| napaluku | 6 |
| Nimellisteho | 5,65 kW |

## 2.3 LabView

LabView tulee sanoista **Lab**oratory **v**irtual **i**nstrumentation **e**ngineering **w**orkbench. Se on National Instrumentsin kehittämä graafinen ohjelmointiympäristö. LabView:n graafinen ohjelmointikieli tunnetaan myös G-kielenä.

LabView julkaistiin vuonna 1986 Apple Macintosh laitteistolle ja nykyään siitä on saatavissa versiot Windows, Linux, Mac ja Solaris käyttöjärjestelmille. LabView:ia käytetään yleisesti mittaus- ja testaussovelluksissa, mutta helppokäyttöisyytensä ansiosta se soveltuu myös yleisohjelmointikieleksi.

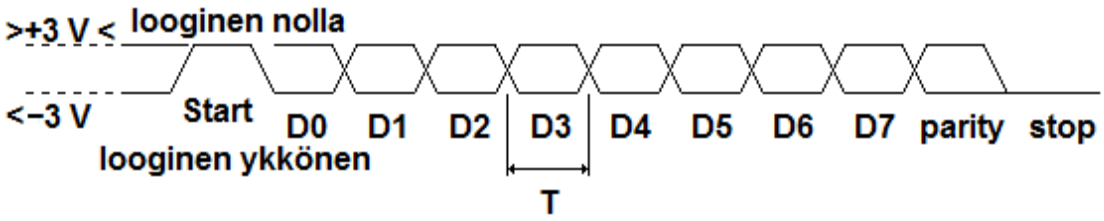
LabView ajaa tuotetun koodin siinä järjestyksessä, kun se on rakenteessa määritetty. Jokainen aliohjelma ajetaan vasta siinä vaiheessa kun kaikki sen tarvitsema tieto on saatu. Tätä kutsutaan tietovuo-ohjelmoinniksi (dataflow programming).

LabView:ssa yksittäinen ohjelmamoduuli on nimeltään ”virtuaali instrumentti” eli vi. Yksi vi koostuu teoriassa kahdesta osasta, etupaneelista (front panel) ja ohjelmasta (block diagram). Jokainen luotu vi voidaan testata erikseen, joka helpottaa virheiden etsintää huomattavasti. Laajemmin tarkasteltuna useita vi:tä voidaan käyttää samassa ohjelmassa eli osaa voidaan käyttää aliohjelmina.

# 3 YHTEYSPROTOKOLLA

Tietokoneen ja taajuusmuuttajan väliseen yhteyteen käytetään sarjaliikenne protokollaa. Käytössä on RS-232 standardit.

**RS-232** (Recommended Standard 232) on kahden tietokonelaitteen väliseen tietoliikenteeseen tarkoitettu tietoliikenneportti. Sarjaliikenne toteutetaan asynkronisesti kahden johtimen avulla (signaali ja referenssitaso). Bitit lähetetään peräkkäin alla olevan kuvan mukaisesti.




**Kuva 3.1: RS-232 sarjaliikenteen ajoitus**

Tiedonsiirto aloitetaan lähettämällä aloitusbitti (start bit), jonka jälkeen vastaanottava laite tietää alkaa tallettaa loogisia tasoja tietyn siirtonopeuden (T, baud rate) määräämin aikavälein. RS-232 on tarkoitettu kahden laitteen väliseen tiedonsiirtoon. Molempien laitteiden on tiedettävä siirtoprotokolla eli databittien ja lopetusbittien määrät sekä käytetäänkö ylimääräistä yhdenvertaisuus bittiä.

**Taulukko 3.1: RS-232 tekniset tiedot**

| | |
|---|---|
| Kaapelin suurin suositeltava pituus | 15 m |
| Siirtonopeus[1] | 20 kbaud |
| Lähettimen jännitetasot | ±5 V → ±15 V |
| Vastaanottimen herkkyys | ±3 V |
| Kuormitus | 3 kohm → 7 kohm |
| Oikosulkusuojan raja | 500 mA |

1. lyhyillä siirtoetäisyyksillä voidaan käyttää jopa 115 kbaudin siirtonopeutta

Laite voi olla joko pääte (DTE data terminal equipment) tai kommunikaatiolaite (DCE data communications equipment).

## 3.1 Tiedonsiirto

RS232:n avulla voidaan muodostaa yhteys taajuusmuuttajan UD70 korttiin. UD70-kortti käyttää sisään rakennettua ANSI x3.28-2.5-4A protokollaa.

Tieto siirretään tietyllä siirtonopeudella (baud rate) tavu muodossa. Yksi tavu koostuu tyypillisesti seitsemästä tai kahdeksasta bitistä.

### 3.1.1 Kehys

Jotta vastaanottava laite ymmärtäisi lähetettävän tiedon, asetetaan tieto kehykseen. Tämä kehys sisältää aloitus bitin, lopetus bitin ja mahdollisesti yhdenvertaisuus bitin. Ilman tätä kehystä vastaanottava laite ei ymmärrä lähetettyä tietoa.

| | Low ASCII character byte | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1st hex character | | | | 2nd hex character | | | | |
| Start bit | Seven data bits | | | | | | | Parity bit | Stop bit |
| 0 | LSB | | | | | | MSB | | 1 |

**Kuva 3.2: Tietokehys**

Tämä kehys (Kuva 3.2) tunnetaan 10-bittisenä kehyksenä, koska 10-bittiä lähetetään kokonaisuudessaan. Kehys koostuu seuraavista osista.

- 1 aloitus bitti, 7 data bittiä, even/odd/no parity, 1 lopetus bitti.
- lsb tarkoittaa vähiten merkittävää bittiä (bit 0)
- msb tarkoittaa eniten merkittävää bittiä (bit 6)
- Parity bittin avulla tarkistetaan tiedon yhtenäisyys

### 3.1.2 ASCII

Käytettävä merkistö tunnetaan nimellä ASCII. Merkistö koostuu 128 merkistä. Ensimmäiset 32 merkkiä kuvaavat erityistä koodia joilla on tietty merkitys, nämä ovat ohjaus koodeja, esimerkiksi STX (Start of text) on ASCII koodi 02.

**Taulukko 3.2: ASCII koodin merkit /4/**

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | NUL | SOH | STX | ETX | EOT | ENQ | ACK | BEL | BS | HT | LF | VT | FF | CR | SO | SI |
| 1 | DLE | DC1 | DC2 | DC3 | DC4 | NAK | SYN | ETB | CAM | EM | SUB | ESC | FS | GS | RS | US |
| 2 | | ! | " | # | $ | % | & | ´ | ( | ) | * | + | , | - | . | / |
| 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| 4 | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 5 | P | Q | R | S | T | U | V | W | X | Y | Z | [ | \ | ] | ^ | _ |
| 6 | ´ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| 7 | p | q | r | s | t | u | v | w | x | y | z | { | \| | } | ~ | DEL |

### 3.1.3 Ohjaus merkit

Kaikki käskyt ja pyynnöt lähetetään viesti paketteina ja jokainen viesti alkaa tietyllä ohjaus merkillä.

| Character | Meaning | ASCII code (decimal) | Keyed as... |
|---|---|---|---|
| EOT | Reset<br>Instructs the UD70 to prepare for a new message.<br>Also indicates parameter does not exist. | 04 | Ctrl D |
| ENQ | Enquiry<br>Used when interrogating. | 05 | Ctrl E |
| STX | Start of text<br>Used to start a command. | 02 | Ctrl B |
| ETX | End of text<br>Used at the end of a command. | 03 | Ctrl C |
| ACK | Acknowledge (message accepted) | 06 | Ctrl F |
| NAK | Negative acknowledge (message not understood) | 21 | Ctrl U |
| BS | Backspace (go to previous parameter) | 08 | Ctrl H |

**Kuva 3.3: Käytössä olevat ohjaus merkit**

Yleisesti viestin alussa käytetään EOT (end of transmission) ohjaus merkkiä. Tämä merkki käskee RS232 väylässä olevat laitteet valmistautua viestin vastaanottamiseen. Vaikka kyseinen ohjaus merkki tarkoittaa suomeksi lähetyksen lopetusta, se asetetaan viestin alkuun, jotta laite on varmasti valmis tiedon vastaanottamiseen.

### 3.1.4 Osoitteet

Jokaisella ANSI tiedonsiirto väylässä olevalla yksiköllä tulee olla ainutlaatuinen osoite, jotta vain kyseinen yksikkö vastaa lähetettyyn viestiin. Osoite muodostuu kahdesta osasta:

- Ryhmä osoite (ensimmäinen merkki)
- Yksikkö osoite (toinen merkki)

Molemmat osoitteet ilmoitetaan numeroilla 1-9:n, ryhmän tai yksikön osoite ei saa olla 0. Tämä siksi, että ANSI tiedonsiirto väylällä voidaan hallita kerralla koko ryhmän laitteita. Esimerkiksi lähetettäessä viesti ryhmän 6 kaikille yksiköille, annetaan osoitteeksi 60. Mikäli halutaan hallita koko järjestelmän kaikkia laitteita kerralla, asetetaan osoitteeksi 00.

Kutsuttaessa useampaa laitetta kerralla UD70-kortit eivät vastaa viestiin.

Tiedonsiirron varmistamiseksi kaksi numeroiset osoitteet lähetetään toistetussa muodossa. Eli osoite 23 lähetetään muodossa 2 2 3 3. Yksikön osoite tulee välittömästi ensimmäisen ohjaus merkin jälkeen.

### 3.1.5 3.1.5 Parametrit ja arvot

Kaikkiin parametreihin viitataan neljä numeroisella luvulla. Ensimmäiset kaksi merkitsevät valikon numeroa ja jälkimmäiset parametrin numeroa. Lähetettävät tai vastaanotettavat parametrien arvot ovat heti parametri numeroiden jälkeen.

Parametrien arvot esitetään numero arvona. Ensimmäinen merkki kertoo onko kyseessä heksa- vai numeroarvo. Mikäli ensimmäinen merkki on x, kyseessä on heksa-arvo.

### 3.1.6 Tarkistussumma (BCC)

Tarkistussumman avulla varmistetaan että lähetetty tieto ei vioitu tiedonsiirron yhteydessä. Tarkistussuman avulla lasketaan kaikki lähetetyt merkit yhteen ja tuloksesta saadaan tarkistussummamerkki (BCC, block check character). Jokainen lähetetty viesti päättyy tarkistussummamerkkiin.

## 3.2 Lukeminen, lähetys ja vastaanotto

### 3.2.1 Parametrin lukeminen

Kun luetaan tietyn parametrin arvot, lähetetään seuraavassa kuvassa esitetty viesti.

| Control | Address | | | | Parameter | | | | Control |
|---|---|---|---|---|---|---|---|---|---|
| **EOT** | **GA** | **GA** | **UA** | **UA** | **M1** | **M2** | **P1** | **P2** | **ENQ** |

Where:

GA = Group Address
UA = Unit Address
M1 M2 = Menu number
P1 P2 = Parameter number

**Kuva 3.4: Lue parametri**

Luettaessa parametrin arvoa ei tarkistussummamerkkiä tarvita, koska mitään tietoa ei kirjoiteta yksikköön. Vastaanotettuaan lue-viestin UD70 vastaa seuraavassa kuvassa (Kuva 3.5) esitetyllä viestillä, mikäli laite ymmärtää kyseisen viestin. Mikäli tiedusteltua parametria ei löydy, laite vastaa ohjaus merkillä EOT. Vastaanotettavan tiedon pituus voi vaihdella. Viestin pituus riippuu tiedustellusta yksiköstä ja parametrista.

| Control | Parameter | | | | Data | | | Control | BCC |
|---|---|---|---|---|---|---|---|---|---|
| **STX** | **M1** | **M2** | **P1** | **P2** | **D1** | **...** | **Dn** | **ETX** | **BCC** |

Where:

M1 M2 = Menu number
P1 P2 = Parameter number
D1...Dn = Data
First character:
+ or Space for positive values
– for negative values
**X** for hex. values
BCC = Block checksum

**Kuva 3.5:UD70-kortin vastaus lue-viestiin**

### 3.2.2 Parametrin kirjoitus

Kun tietoa kirjoitetaan, pitää lähetetyn viestin muoto olla seuraavan kuvan mukainen.

| Control | Address | | | | Control | Parameter | | | | Data | | | Control | BCC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **EOT** | **GA** | **GA** | **UA** | **UA** | **STX** | **M1** | **M2** | **P1** | **P2** | **D1** | **...** | **Dn** | **ETX** | |

Where:

GA = Group address
GU = Unit address
M1 M2 = Menu number
P1 P2 = Parameter number
D1...DN = Data
First character:
+ or Space for positive values
– for negative values
**X** for hex. values
BCC = Block checksum

**Kuva 3.6: Kirjoita-viestin muoto**

Lähetettävän kirjoita-viestin pituus riippuu kyseessä olevan parametrin arvon pituudesta.

### 3.2.3 Tarkistussumman laskeminen

Tarkistussumma lasketaan käyttämällä **poissulkeva tai** (exclusive OR, XOR) loogista operaatiota jokaiseen lähetetyn viestin merkkiin. Seuraavassa kuvassa on esitetty XOR- operaation totuustaulukko.

| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Kuva 3.7: Totuuttaulukko**

Laskettaessa tarkistussummaa ensimmäinen merkki on 0 (00110000 binäärimuodossa), jota käytetään aloitus arvona. Seuraavassa kuvassa on esimerkki kirjoitus-viesti, josta lasketaan tarkistussumma.

| Control | Address | | | | Control | Parameter | | | | Data | | | | | Control | BCC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOT | 1 | 1 | 2 | 2 | STX | 0 | 1 | 2 | 1 | - | 3 | 4 | . | 5 | ETX | 0 |
| Not included in the calculation | | | | | | Included in the calculation | | | | | | | | | | Result |

**Kuva 3.8:Esimerkki kirjoitus-viesti**

Tarkistussumman laskemisessa otetaan huomioon vain parametrin numero, lähetettävä arvo ja viimeinen ohjaus merkki. Viestin aloitusta tai osoitetta ei siis oteta huomioon. Seuraavassa kuvassa on esitetty XOR- operaation tulokset. Viimeinen XOR- arvo on tarkistussumma merkki eli BCC. Mikäli BCC on arvoltaan 32 tai vähemmän, lisätään siihen 32. Koska ASCII-koodissa merkit 00-31 ovat ohjaus merkkejä, pitää BCC:n olla arvoltaan tätä suurempi.

**Taulukko 3.3: Tarkistussumman lasku**

| Character | Binary Value | XOR result |
|---|---|---|
| 0 | 0011 0000 | – |
| 1 | 0011 0001 | 0000 0001 |
| 2 | 0011 0010 | 0011 0011 |
| 1 | 0011 0001 | 0000 0010 |
| – | 0010 1101 | 0010 1111 |
| 3 | 0011 0011 | 0001 1100 |
| 4 | 0011 0100 | 0010 1000 |
| . | 0010 1110 | 0000 0110 |
| 5 | 0011 0101 | 0011 0011 |
| ETX | 0000 0011 | 0011 0000 |

Viimeisen XOR- operaation arvo on 00110000 eli numeerisessa muodossa luku 48. ASCII-koodissa tämä luku tarkoittaa merkkiä 0, joten BCC on merkki 0. Lopullinen lähetettävä viesti on esitetty seuraavassa kuvassa.

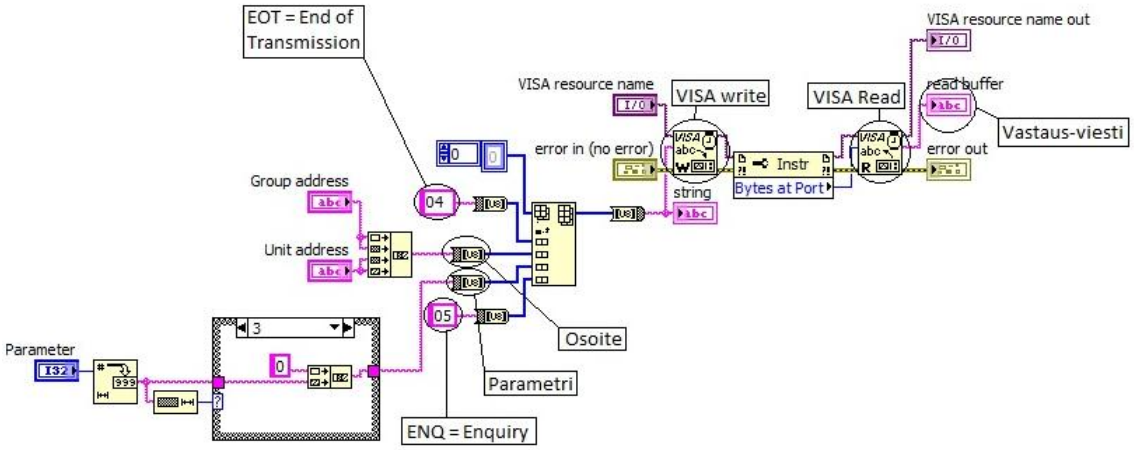
| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EOT | 1 | 1 | 2 | 2 | STX | 0 | 1 | 2 | 1 | - | 3 | 4 | . | 5 | ETX | 0 |

**Kuva 3.9:Lopullinen kirjoitus-viesti**

## 3.3 Lukeminen, lähetys ja vastaanotto LabView-ohjelmalla

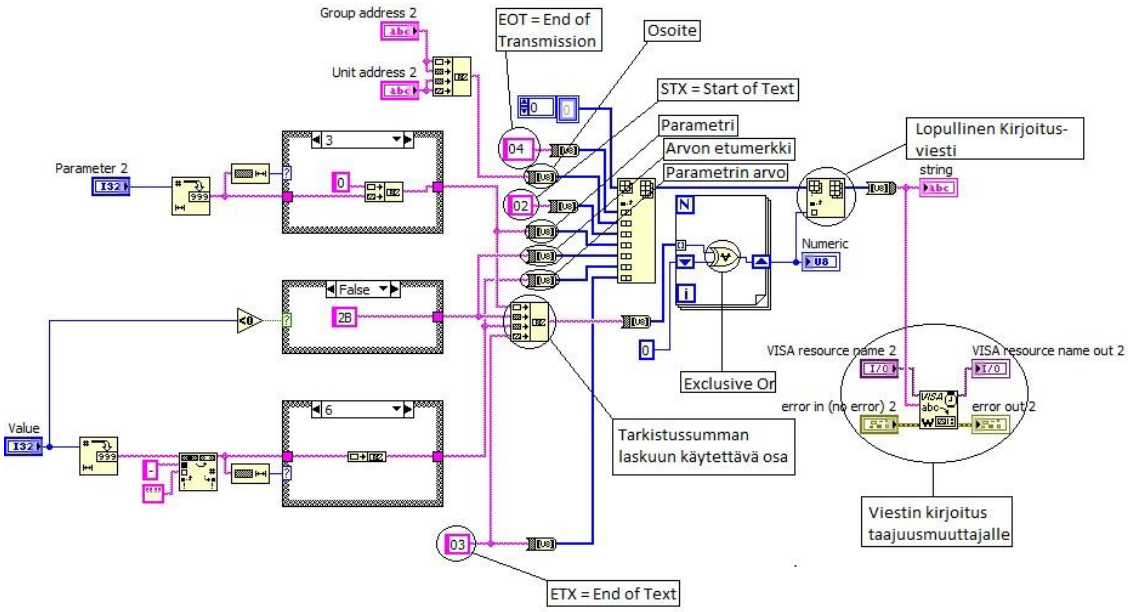
Seuraavassa kuvassa on esitetty lue-viestin lähetys ja vastaanotto LabView-ohjelmalla. Kuten kuvasta näkyy ohjelma vastaa aiemmin esitettyä järjestystä. Tiedot asetetaan binääriseen muotoon, koska muutos binäärisestä muodosta laitteen ymmärtämään hex-muotoon aiheuttaa vähemmän virheitä. Teksti muodosta hex-muotoon muutettaessa esimerkiksi numero 04:n (EOT) merkitys

muuttuu, kun ohjelma olettaa hex-merkin olevan kaksi merkkiä, 0 ja 4. Eli lähetettävän tiedon sisältö muuttuu oleellisesti.




**Kuva 3.10: Lue-viestin lähetys ja vastaanotto LabView-ohjelmalla**

Seuraavassa kuvassa (Kuva 3.11) on esitetty parametrin kirjoitus ja tarkistussumman lasku LabView-ohjelmalla.




**Kuva 3.11: Parametrin kirjoitus ja tarkistussumman lasku LabView-ohjelmalla**

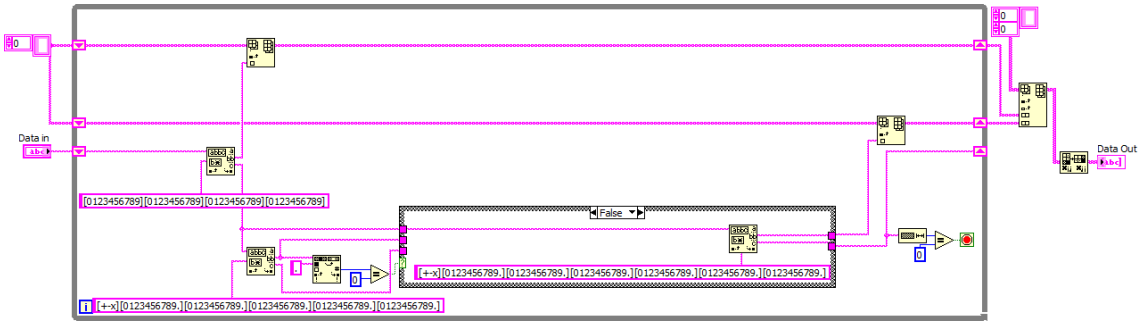
### 3.3.1 Vastaanotetun tiedon muuttaminen ymmärrettävään muotoon

Data in

```
.1040+12323.<.1028+00001.2.0101+00612.=.0111+
00001.8.0112+00001.;.0113+00000.;.0201-
00609.2.0302-00609.0.0402-000.36.3.0404-
0003.4.7.0505+00693.4.0622+00.003.3.0623+
020.51.7.1016+00000.>.1017+00000.?.1018+
00000.0.1019+00000.1.1032+00000.8.1042+
00000.?.1020+00001.:.1021+00001.;.1022+
00001.8.1023+00001.9.1024+00001.>.1025+
00001.?.1026+00001.<.1027+00001.=.1029+
00001.3.1040+12323.<.1028+00001.2.0101+
00609.7.0111+00001.8.0112+00001.;.0113+
00000.;.0201-00609.2.0302-00609.0.0402-
000.38.=.0404-0003.3.0.0505+00695.2.0622+
```

**Kuva 3.12:Vastaanotettu tieto**

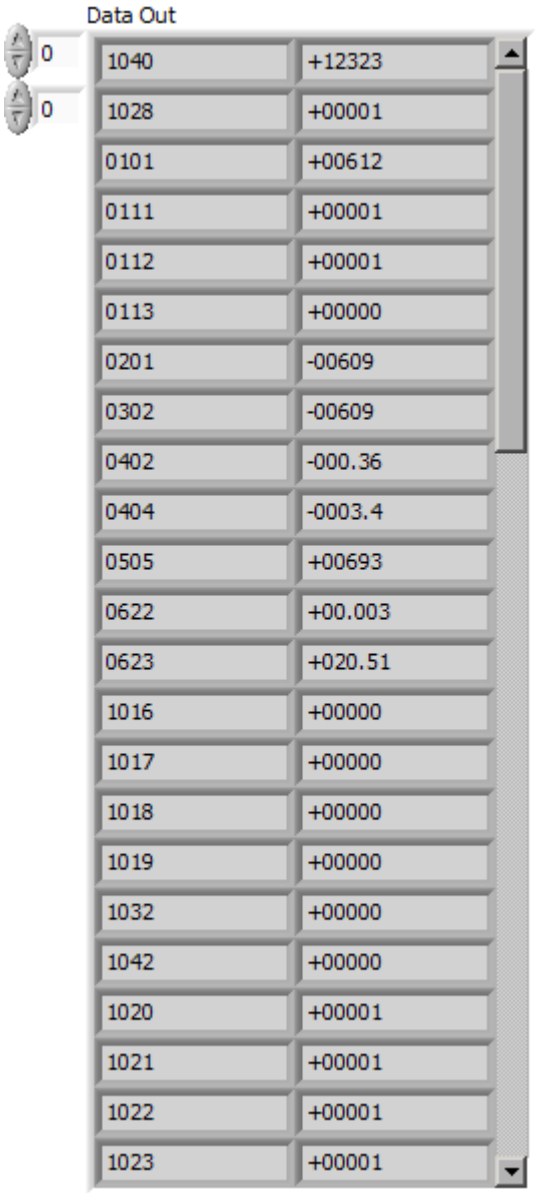
Kuvassa (Kuva 3.12) on esitetty taajuusmuuttajasta vastaanotettu tieto. Vastaanotettu tieto ei ollut valmiiksi helposti ymmärrettävässä muodossa, joten tarvittiin aliohjelma tiedon tulkitsemiseksi. Seuraavassa kuvassa on esitetty vastaanotetun tiedon parsimiseen käytetty aliohjelma.




**Kuva 3.13:Tiedon parsimiseen käytetty aliohjelma.**

Tutkimalla taajuusmuuttajalta saatua tietoa ja vertaamalla sitä aiemmin esitettyyn vastaus viestiin (Kuva 3.5), saadaan selville tulevan tiedon runko. Ohjelma etsii aluksi neljä numeroisen sarjan, joka merkitsee parametrinumeroa, ja siirtää tämän numeron taulukkoon. Tämän jälkeen ohjelma etsii etumerkeillä +, - tai x alkavaa numerosarjaa.

Taajuusmuuttajan vastauksessa parametrin arvo sisältää etumerkin jälkeen neljästä seitsemään merkkiä. Silloin kun arvo ei sisällä desimaalipistettä, on arvossa normaalisti viisi merkkiä. Joten ohjelma etsii viittä merkkiä jotka koostuvat merkeistä 0,1,2,3,4,5,6,7,8,9 ja ”.”. Mikäli ohjelma löytää desimaalipisteen, case-rakenne etsii arvon loppuosan. Saatu parametrin arvo lisätään taulukkoon parametrinumeron perään.




**Kuva 3.14: Parsittu tieto**

Yllä esitetyssä kuvassa on esitetty taajuusmuuttajalta saadut parametrit parsitussa muodossa. Kaikki tieto käsitellään string-muodossa, koska numeerisessa muodossa ohjelma poistaa automaattisesti +-etumerkin. Tämä aiheuttaisi ongelmia tiedonsiirrossa, koska taajuusmuuttaja tarvitsee parametrin arvon etumerkin tietoa siirrettäessä.

# 4 TAAJUUSMUUTTAJAN SISÄINEN OHJELMA

Taajuusmuuttajan sisällä on oma ohjelmansa, joka määrittää nopeuden ja momentin ohjauksen ohjauspulpetista. Taajuusmuuttajan sisäistä ohjelmaa täytyi muuttaa hieman, koska nopeutta ja momenttia ohjaavat parametrit oli jo sidottu ohjauspulpettiin.

SKS Control Techniques:n taajuusmuuttajan ohjelman tarkastelemiseen ja muuttamiseen käytettiin DPL toolkit ohjelmaa joka on toimitettu taajuusmuuttajan mukana.

Muutosten tekeminen ohjelmaan osoittautui hankalaksi, koska DPL toolkit-ohjelma on tehty Windows 3.1 alustalle. Ohjelma kaatui jatkuvasti ja muutosten teko piti varmistaa useaan kertaan.

## 4.1 4.1 Muutokset

Ilman muutoksia taajuusmuuttajan sisäiseen ohjelmaan, ei taajuusmuuttajaa voinut ohjata muista ulkoisista lähteistä kuin ohjauspulpetista. Taajuusmuuttajan sisäinen ohjelma on esitetty liitteessä 1. Tehdyt muutokset on merkattu keltaisella pohjavärillä.

Muutokset toteutettiin ehtolauseella ja sidottiin vapaisiin parametreihin taajuusmuuttajan sisällä. Eli, kun taajuusmuuttajan parametri 19.30 on 0, taajuusmuuttaja toimii normaalisti, ohjaus ohjauspulpetista. Kun parametri 19.30 on 1, voidaan ohjelman avulla säätää nopeutta, muu ohjaus ohjauspulpetilla. Parametrin 19.30 ollessa 2, voidaan ohjelman avulla ohjata nopeutta ja momenttia. Parametrin 19.30 ollessa 3 kyseessä on generaattori käyttö, eli ainoastaan momenttia voidaan ohjata ohjelmalla.

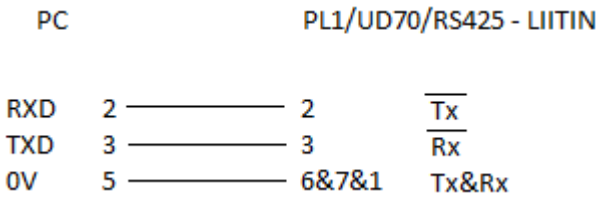
# 5 TYÖN TOTEUTUS

Tässä luvussa käydään läpi LabView- sovellus, jolla SKS:n servokäyttöä ohjataan, sekä yhteyden muodostus taajuusmuuttajaan.

Sovellus koostuu kuudesta osasta.

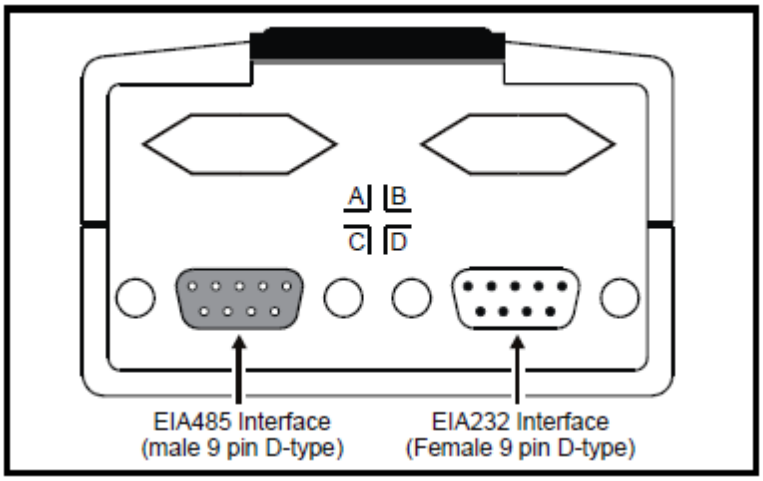
- Yhteyden muodostus
- Pääkäyttöliittymä
- Parametrien asetus
- Monitorointi-tila
- Ajoarvojen asetus
- Ajoarvojen ajo

## 5.1 Yhteyden muodostus




**Kuva 5.1: Räätälöidyn D-9 liittimen kytkentä**

Yhteys taajuusmuuttajaan muodostettiin kuvassa (Kuva 5.1) esitetyllä räätälöidyllä kaapelilla. Kaapeli kytkettiin seuraavassa kuvassa esitettyyn EIA232 liittimeen. Yhteys muodostettiin neljä johtimisen EIA232 ANSI slave-protokollan avulla.



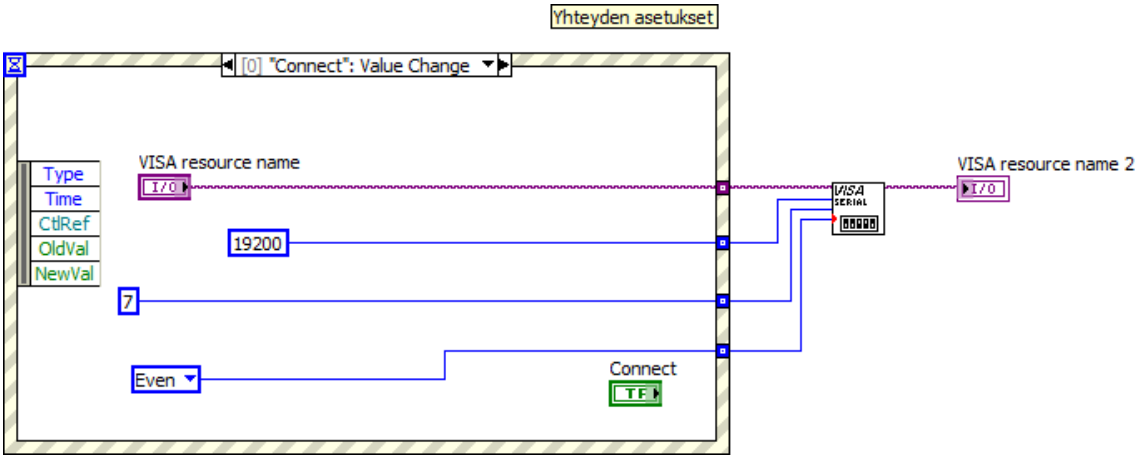

**Kuva 5.2: Sarjakaapelin liitäntä taajuusmuuttajaan**

Fyysisen sarjakaapeli yhteyden muodostamisen jälkeen LabView-sovellus muodostaa yhteyden taajuusmuuttajaan.

Seuraavassa kuvassa on esitetty yhteyden muodostus ikkuna, jonka sovellus avaa ensimmäisenä.

**Kuva 5.3:COM-portin valinta**

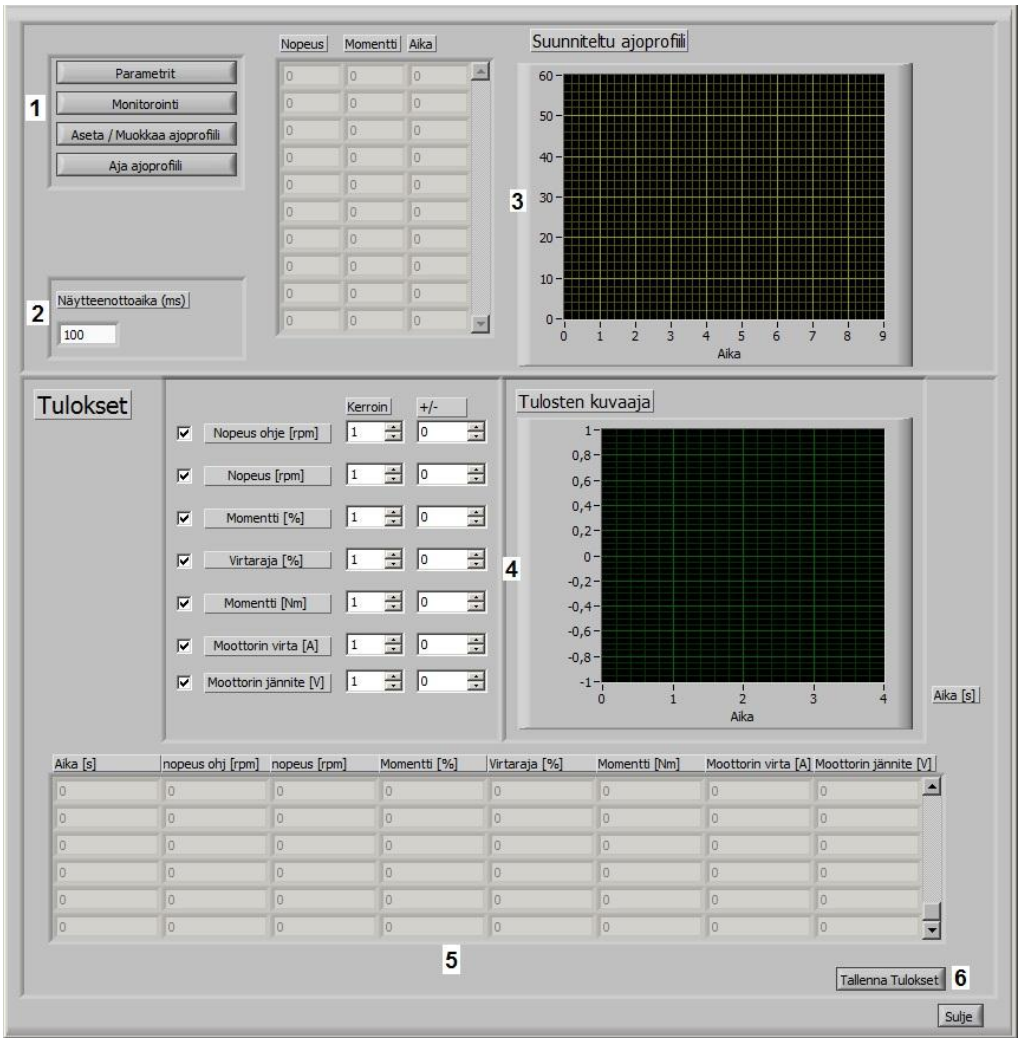
Käyttäjän tehtäväksi jää oikean sarjaportin valinta. Ohjelmaan on itsessään asetettu valmiiksi yhteyden määritykseen liittyvät asetukset. Yhteys on vakiona 19200 baudia, data bittejä on 7 ja parity on even.




**Kuva 5.4:Yhdistys aliohjelma**

Yhteyden muodostamisen jälkeen sovellus etenee pääkäyttöliittymään.

## 5.2 Pääkäyttöliittymä

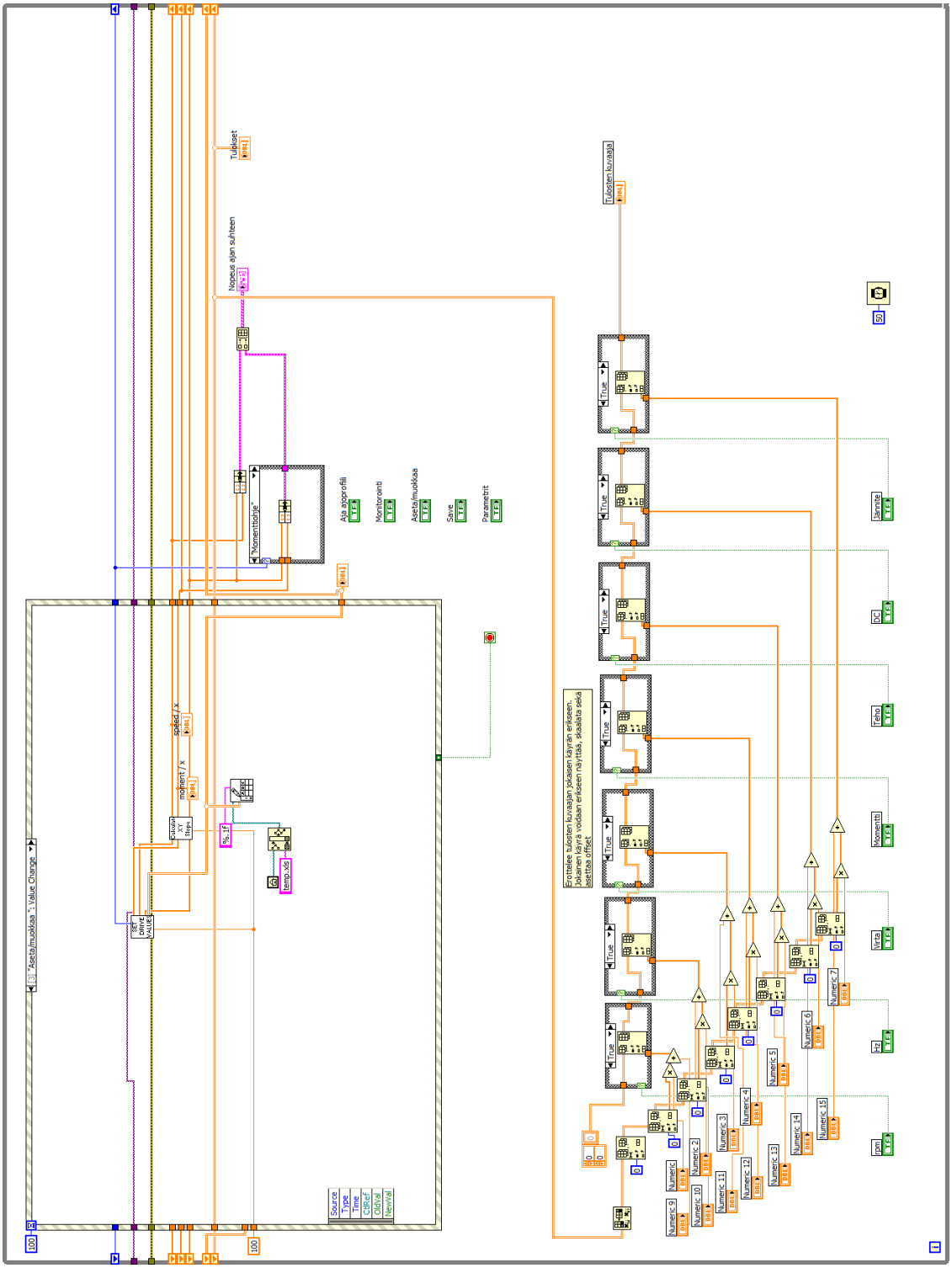



**Kuva 5.5: Pääkäyttö liittymä**

Yhteyden muodostamisen jälkeen avautuu kuvassa (Kuva 5.5) esitelty pääkäyttöliittymä. Kuvan kohdassa 1 on taajuusmuuttajan hallintaan käytettävät painikkeet. Näistä painikkeista avautuu ohjauspaneelit parametrien muuttamiseen, monitorointiin, ajoprofiilin asettamiseen ja ajamiseen. Kohdassa 2 on näytteenottoajan asetus, joka vaikuttaa mittausten ottoaikaan sekä ajoprofiilin ajan asetukseen. Kohdassa 3 näkyy suunnitellun ajoprofiilin kuvaaja ja ajoarvot.

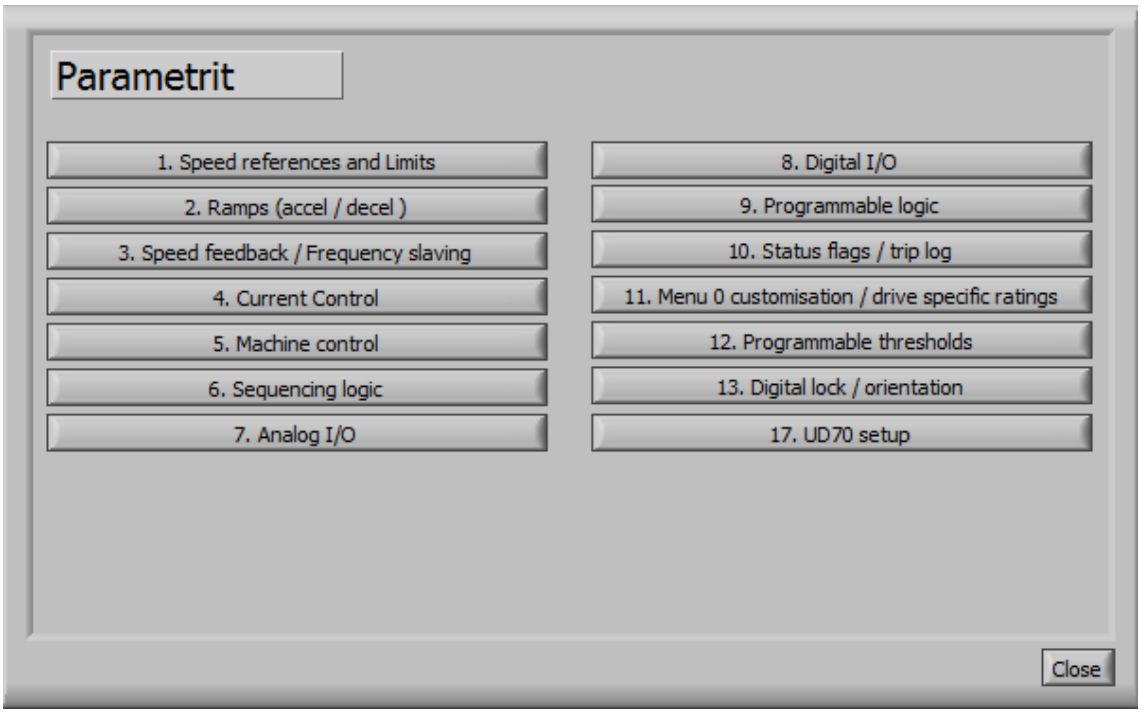
Kuvan kohdassa 4 on tulosten tarkasteluun käytettävät työkalut. Ajetun ajoprofiilin tulokset tulostuvat kuvaajaksi ja kuvaajan arvoja voi skaalata ja muokata pikaista tarkastelua varten. Kohtaan 5 ohjelma tulostaa kaikki mitatut arvot ja kohdassa 6 on mahdollisuus siirtää mitatut arvot Excel-taulukkoon.

Pääkäyttöliittymä rakennettiin while-loopin ja event-rakenteen avulla. Siirtorekistereitä tarvittiin runsaasti, koska muokattujen tietojen piti kulkea useiden eri aliohjelmien ja osien välillä. Kuvassa (Kuva 5.6) on esitetty pääkäyttöliittymän ohjelma.

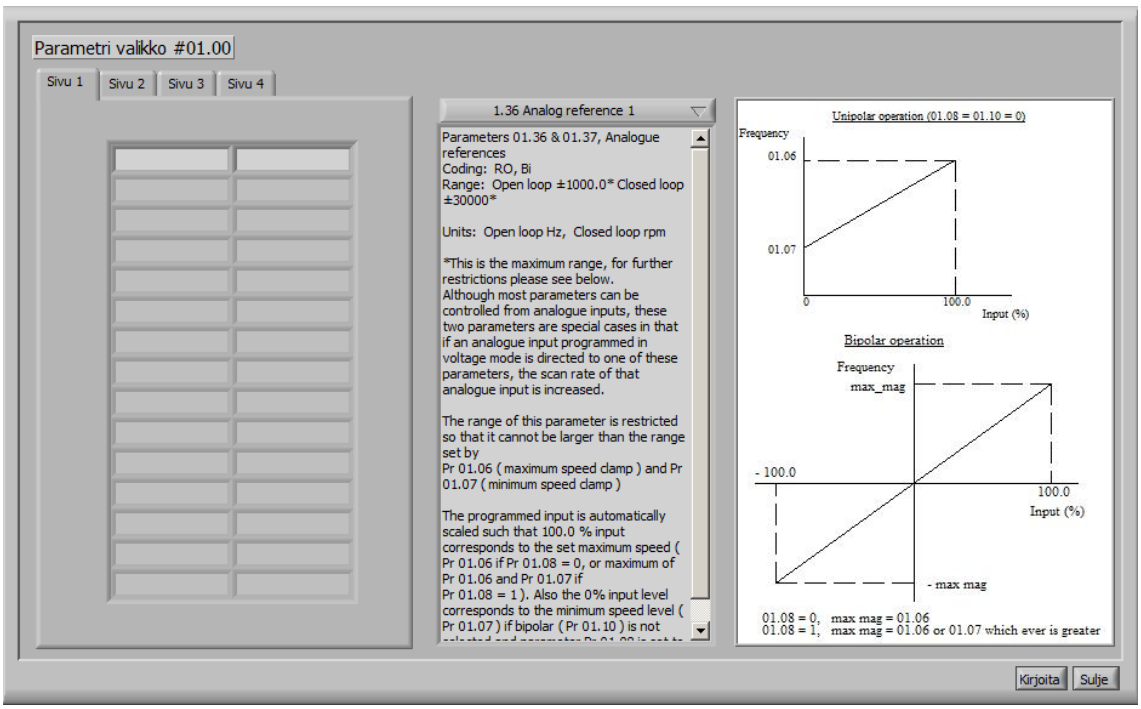



**Kuva 5.6:Pääkäyttöliittymän ohjelma**

## 5.3 Parametrit



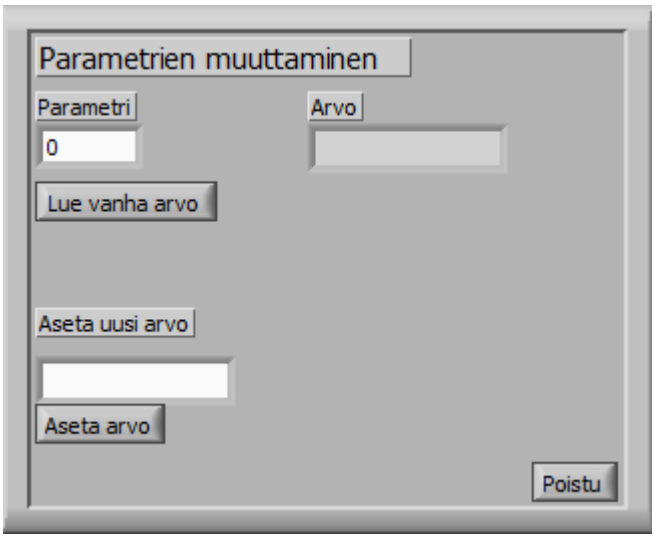

**Kuva 5.7:Parametrit-valikko**




**Kuva 5.8: Parametri valikko #01.00**

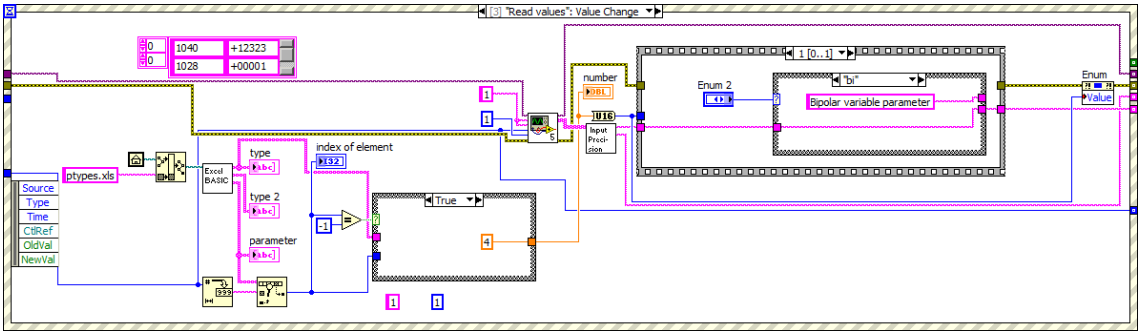
Parametri-valikon avulla valitaan haluttu parametri. Kuvassa (Kuva 5.7) on esitetty parametri-valikko. Kuvassa (Kuva 5.8) on esitetty parametri valikko 1. nopeus ohjeet ja rajat. Ohjelman ollessa kytkettynä taajuusmuuttajaan ohjelma noutaa taajuusmuuttajasta parametrit ja niiden arvot. Ensimmäisessä sarakkeessa on parametrin numero ja toisessa sen arvo. Oikealle puolella on laitteen manuaalista koottu parametrien selitykset.

### 5.3.1 Parametrien kirjoitus




**Kuva 5.9: Parametrien kirjoitus**

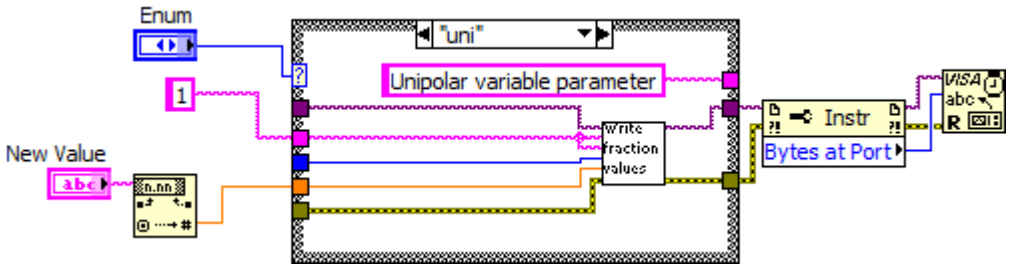
Kun painetaan ”kirjoita”-näppäintä valikosta, aukeaa kuvassa (Kuva 5.9) esitetty ikkuna. Tähän ikkunaan syötetään muutettava parametri kohtaan parametri. Tämän jälkeen ohjelma tulostaa parametrin tämän hetkisen arvon. Syötetään uusi arvo ja painetaan ”aseta arvo”-painiketta. Mikäli parametri on RO-arvo, ohjelma tulostaa tekstin ”aseta uusi arvo” kohtaan: ”RO-arvoa ei voida muuttaa.”




**Kuva 5.10: Lue vanha arvo**

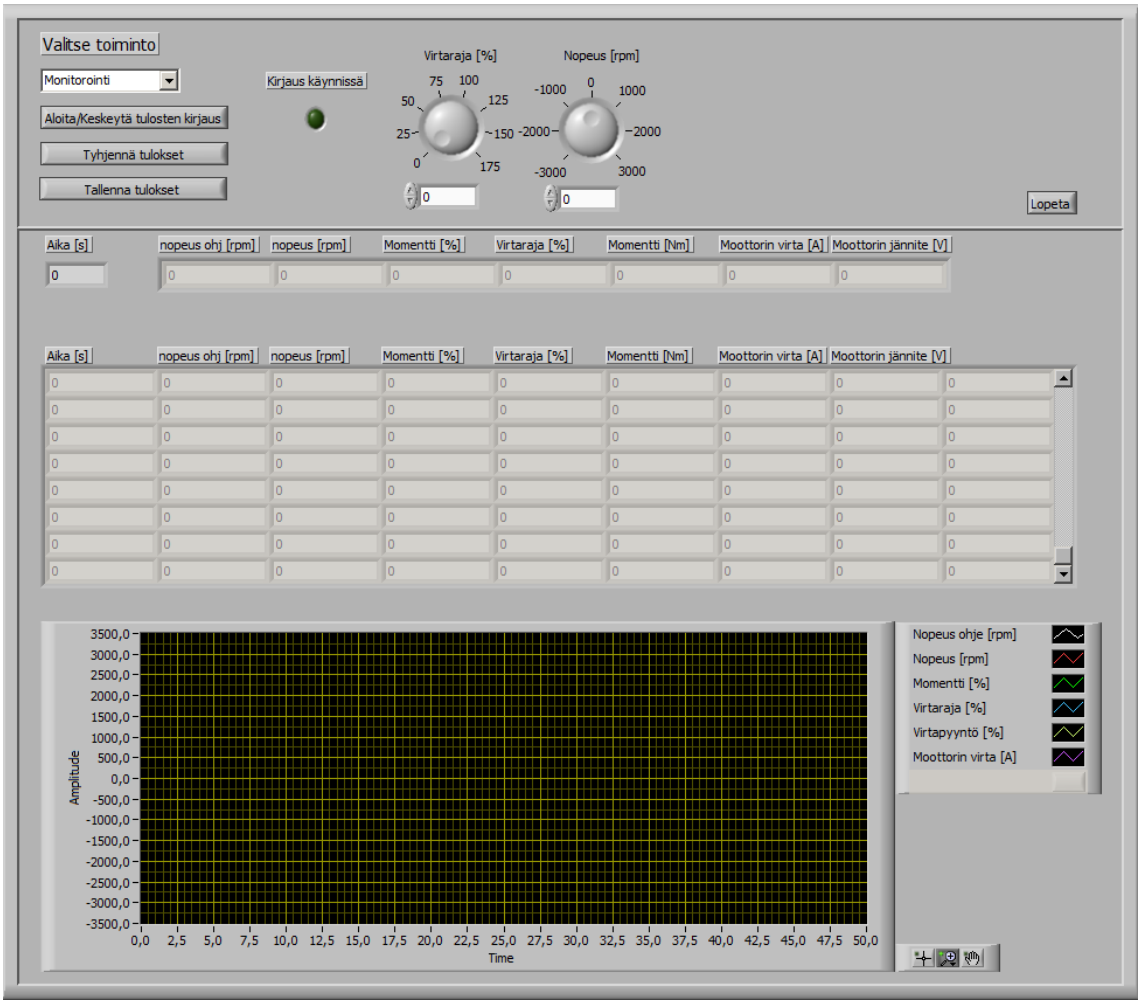
Kun painetaan ”lue vanha arvo”-painiketta ohjelma lukee parametrin laitteelta, sekä vertaa parametrin arvoa ennalta tehtyyn taulukkoon ptypes.xls. Tämän taulukon avulla selviää voidaanko parametria muuttaa ja missä muodossa parametrin arvo pitää kirjoittaa (bit, bipolar, unipolar, txt).

Kuvassa (Kuva 5.11) on esitetty kirjoitus toiminto, kun kirjoitettava parametri on unipolar muodossa.




**Kuva 5.11: Kirjoita uusi arvo**

## 5.4 Monitorointi tila




**Kuva 5.12: Monitorointi ikkuna**

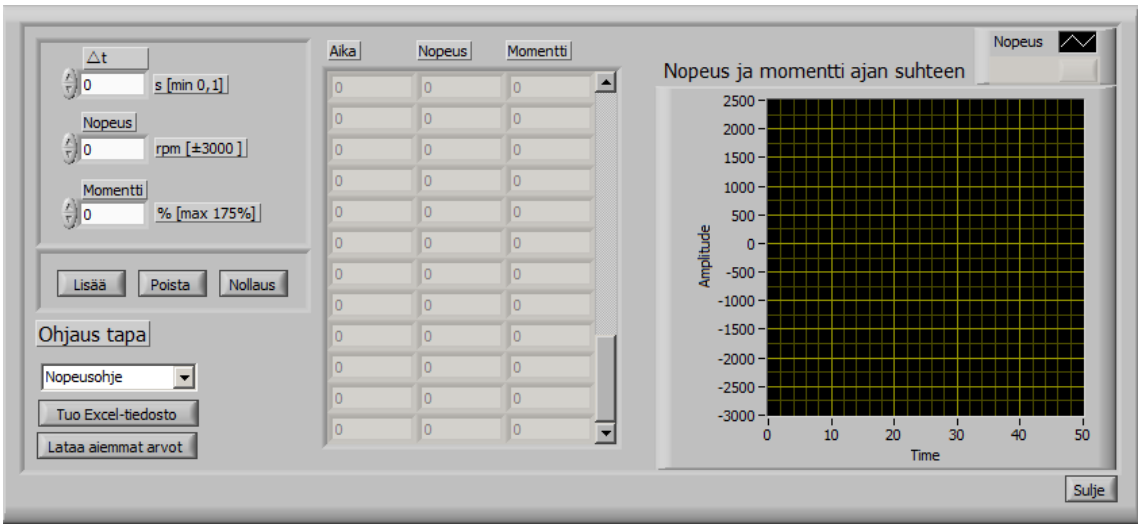
Monitorointi ikkunassa voidaan tarkastella taajuusmuuttajakäytön ajoa tai ohjata käyttöä manuaalisesti. Kuvassa (Kuva 5.12) on esitetty ohjauspaneeli. Yläosassa on ohjaus valinnat ja alaosa on tulosten tarkastelua varten. Mikäli valittuna on monitorointi, voidaan käyttöä ohjata normaalisti ohjauspulpetista.

Tulosten kirjaus painikkeella ohjelma tallentaa esitetyt arvot asetetun näytteenottoajan välein. Tallenna painikkeella kirjatut tulokset siirretään suoraan Excel-taulukkoon. Tyhjennä painike tyhjentää tähän asti kirjatut tulokset.

Valittaessa kohta nopeusohje voidaan ohjelmalla suoraan muuttaa nopeusohjetta, joko nopeusohje rullasta tai kirjoittamalla haluttu nopeus rullan alapuolella olevaan teksti kenttään.

Momenttiohje kohdassa voidaan säätää virtarajaa. Moottori tarvitsee kuitenkin nopeusohjeen ohjauspulpetilta. Nopeus- ja momenttiohje kohdassa voidaan molempia säätää ohjelman avulla.

## 5.5 Ajoarvojen asetus



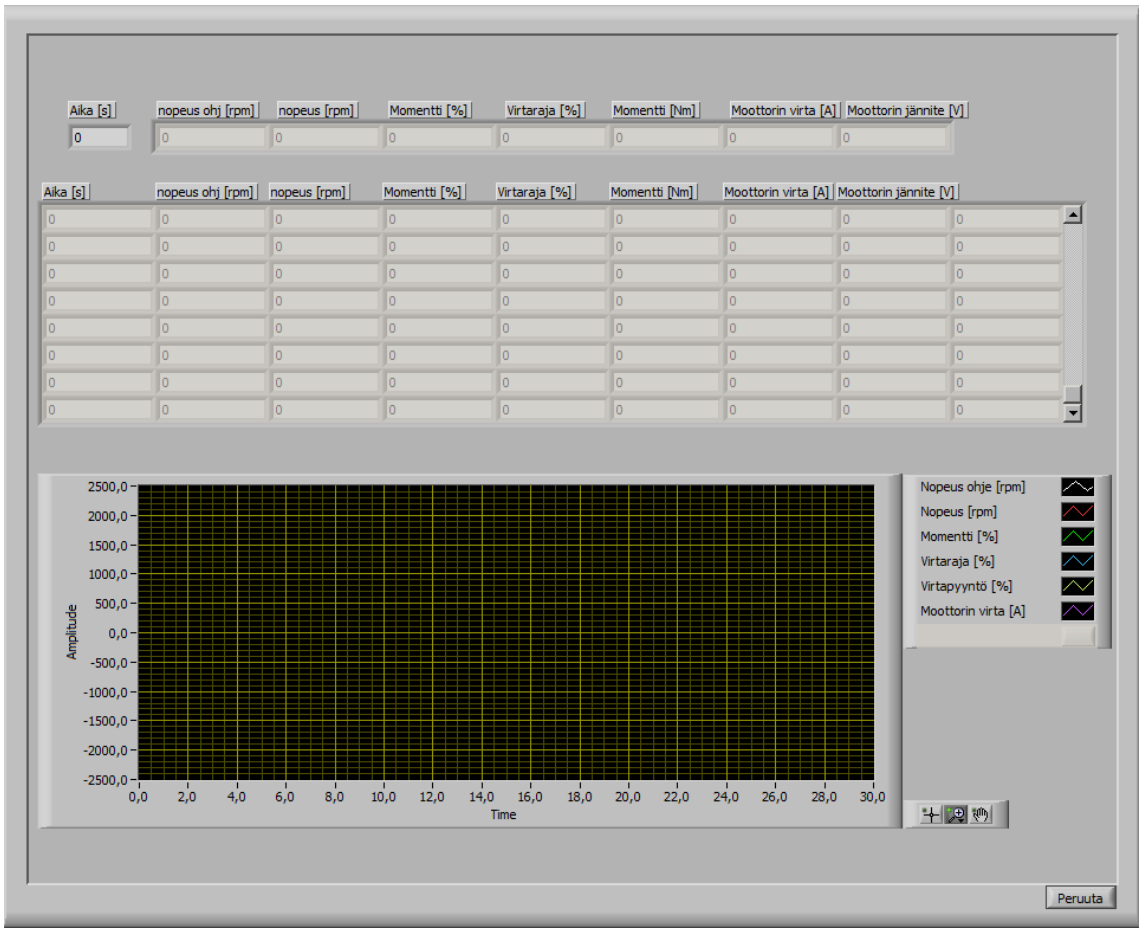

**Kuva 5.13: Ajoarvojen asetus**

Ajoarvojen asetus ikkunan avulla voidaan asettaa manuaalisesti ajettavat arvot tai ladata ajettavat arvot Excel-tiedostosta. Aika asetetaan nopeuden muutoksena. Ohjelma lisää ajan aina edelliseen aikaan. Nopeusohje asetetaan kierroksina minuutissa ja momenttiohje prosentteina. Momenttiohjeen maksimi arvoksi on asetettu 175% moottorin suojaamiseksi.

Jos ajoarvot tuodaan Excel-tiedostosta pitää arvojen olla samoilla sarakkeilla kuten yllä olevassa kuvassa (Kuva 5.13) esitetään, eli järjestys on aika, nopeus ja momentti.

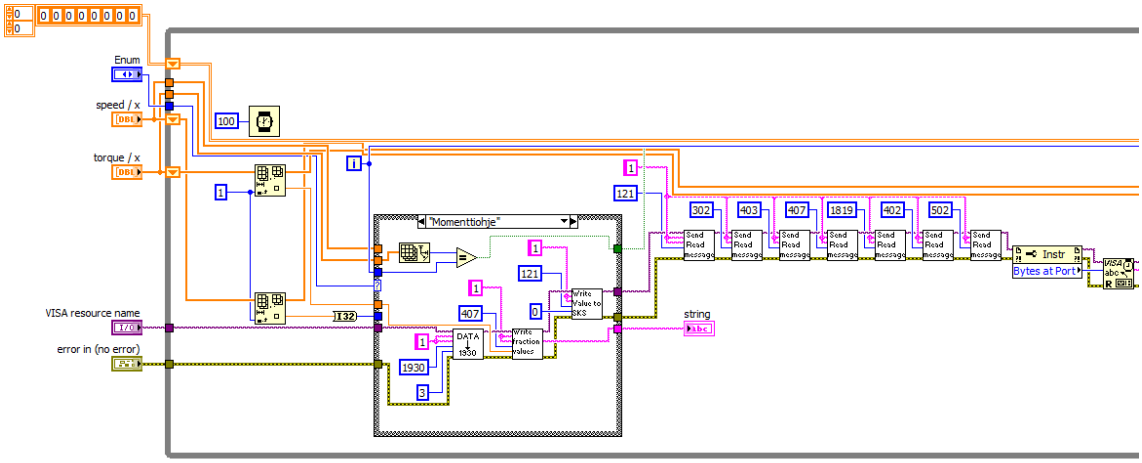
Kun ajoarvot on asetettu ja ikkuna suljetaan, arvot näkyvät pääikkunaan (Kuva 5.5).

## 5.6 Aja ajoarvot




**Kuva 5.14: Asetettujen ajoarvojen ajo.**

Aja ajoarvot ikkuna ajaa automaattisesti aiemmin asetetut ajoarvot. Ikkuna on vain tulosten seurantaa varten.




**Kuva 5.15: Aja ajoarvot, ohjelma.**

Ohjelma lähettää ajoarvot taajuusmuuttajalle ja lukee määrätyt parametrit asetetun näytteenottoajan välein. Arvojen lukemisen jälkeen ohjelma parsii

luetut parametrit ymmärrettävään muotoon. Yllä olevassa kuvassa (Kuva 5.15) on esitetty ”aja ajoarvot ohjelma”. Case-rakenteessa on kolme eri vaihtoehtoa, riippuen onko lähetettävä ohje momentti- nopeus- vai momentti- ja nopeusohje. Case-rakenteen jälkeen olevat ”send read message”-aliohjelmat noutavat niihin määrättyjen parametrien arvot taajuusmuuttajalta. Kuvasta puuttuvassa osassa tiedot muokataan helpommin ymmärrettävään muotoon.

Ajoarvojen ajon jälkeen ohjelma kokoaa saadut tulokset pääikkunaan (Kuva 5.5, 4). Tuloksia voidaan tarkastella pääikkunassa tai siirtää ne Excel-tiedostoon.

# 6 LOPPUSANAT JA PARANNUS EHDOTUKSET

Työ onnistui oikein hyvin. Puutteellisen LabView-koulutuksen vuoksi aikataulut venyivät todella paljon, mutta opetti paremmin kuin yksikään kurssi olisi voinut opettaa ohjelman käytöstä.

Työn edetessä vastaan tuli paljon ongelmia. Yksi suurimmista oli nopeus- ja momenttiohjeen lähetys taajuusmuuttajalle. Vaikka parametrien luku ja kirjoitus onnistui, nopeutta ja momenttia ohjaavia parametreja ei voitu muuttaa. Useiden tuntien selvityksen jälkeen selvisi että nämä parametrit oli jo määritetty taajuusmuuttajan sisäisessä ohjelmassa. Ohjelmisto hierarkiassa taajuusmuuttajan sisäinen ohjelma meni ulkoisten ohjelmien edelle. Esimerkiksi nopeusohjetta lähetettäessä taajuusmuuttaja vastasi että, ohje otettu vastaan, mutta ei muuttanut moottorin nopeutta. Myöhemmin selvisi että, ohje oli todella otettu vastaan, mutta taajuusmuuttajan sisäinen ohjelma korvasi välittömästi vastaanotetun ohjeen ohjauspulpetin ohjeella.

Toinen suuri ongelma oli parametrien kirjoitus. Koska parametrit voivat olla useassa eri muodossa. (txt, hex, bipolar, binääri, unipolar) Ja koska lähetettävät parametrit sisälsivät eri määrät merkkejä, riippuen muodosta ja siitä sisälsivätkö ne desimaali pisteen vai eivät, oli yhteisen ohjelman luominen parametrien kirjoittamiseen erittäin hankalaa. Ongelma saatiin ohitettua lukemalla kirjoitettavan parametrin nykyinen arvo ennen lähetystä, ja varmistamalla että, uusi parametri lähetetään samassa muodossa.

Työtä tehdessä löytyi useita parannus ideoita joita ei kerinnyt enää toteuttamaan. Esimerkiksi ohjelman avulla tehtävää ajoarvojen syöttöä olisi voinut parantaa. Nyt arvoja voi lisätä ja poistaa vain aikajärjestyksessä, eli ohjelma lisää arvon aina viimeiseksi ja poistaa vain viimeisen arvon.

Työlle asetettuihin tavoitteisiin päästiin, mutta useiden pienten asioiden muutoksia harkitsin työn eri vaiheissa. Esimerkiksi MS Excel-ohjelman käytön olisi voinut korvata OpenOfficen vastaavalla. Nyt Ohjelman käyttövaatimuksena on MS Office. Lisäksi mahdolliset päivitykset MS Officeen voivat vaikuttaa ohjelman toimintaan. Ja koska tein Excel-ohjelmaa käyttävän aliohjelman itse,

enkä käyttänyt kaupallista lisäsovellusta, ohjelma sulkee koko Excel-sovelluksen aina kun ohjelma käyttää sitä, joten kyseistä ohjelmaa ei voi käyttää samaan aikaan kun tämä ohjelma on auki.

Työtä voisi edelleen kehittää ohjaamaan useampaa moottorikäyttöä yhtä aikaa, sekä ohjaamaan muiden valmistajien moottorikäyttöjä. Lisäksi voisi olla kiinnostavaa kehittää sovellus jolla voisi etäohjata moottoreita esimerkiksi verkon ylitse.

Myös tarkkuusmittauksiin sovellusta voisi kehittää. Mikäli käytössä on kehittyneempi väylä järjestelmä kuin sarjakaapeliliitäntä, voisi sovelluksella ottaa millisekunnin tarkkuudella moottorin arvoja. Tämän avulla voitaisiin tarkkailla moottorin todellista jättämää ja ohjaustarkkuutta nopeissa nopeuden muutoksissa.

**LÄHTEET**

/1/ Juha Aaltonen, Seppo Kousa, Jyrki Stor-Pellinen, Elektroniikan perusteet.2002.

/2/ Seppo Haltsonen, Jaakko Levomäki, Esko T. Rautanen, Digitaalitekniikka. 2005.

/3/ Lauri Aura, Antti J. Tonteri, Sähkökoneet ja tehoelektroniikan perusteet. 1996.

/4/ SKS:n UNI2402 manuaali: saatavissa:

http://www.controltech.com.tr/Documents/Unidrive%20Classic/Unidrive%20User%20Guide%20Iss9.pdf

# LIITTEET

LIITE 1 Taajuusmuuttajan sisäinen ohjelma

LIITE 2 Käyttöohjeet

**LIITE 1**

```
$AUTHOR Vesa Auvinen
$COMPANY Control Techniques SKS Oy
$VERSION 1.0.2
$DRIVE Unidrive

NOTES{
 Versio 1.0.0                 03.09.-99 VA
 Sähkökoneiden testauslaitteisto
 1kpl Unidrive UNI2402 verkkoyksikkö + 1 kpl  Unidrive UNI2402 moottorikäyttö
 CTIU käyttöpääte

Versio 1.0.1      14.01.2000 VA
Lisätty tahti muuttuja (#18.41), jolla voi asettamalla=1 valita nopeuskäyttöön
myös momenttiohjaus.
Lisätty valinta% muuttuja ja kertaX% muuttujat, joilla reset tehdään vain kun on
muutos ajotapaan.
Lisätty CTIU:n lukemille nopeudelle ja momentille suoto
Lisätty ominaisuus, että duty tavassa t1 ja t2 mahdollista muuttaa ajan
aikanakin
Lisätty ominaisuus, että momentttien asetusta voi muuttaa duty tavassa aina

Versio 1.0.2                 04.02.2000 VA
Korjattu Versio 1.0.1 tahtikoneen asetuksia
Siirretty t1 ja t2 käsittely kellotaskiin, jotta voidaan muuttaa vauhdissa sekä
skaalaus kellotaskin tick aika(10 ms*100000)*asetus.
Testattu ja ladatttu sysfile 2.7.6 kanssa
}
;                TOIMINTA-kytkin
$define Toiminta_M1 #18.31
$define Toiminta_M2 #18.32
$define Toiminta_G #18.33
$define Toiminta_PROG #18.34
;                MOMENTTI-kytkin
$define Momentti_1 #18.35
$define Momentti_2 #18.36
$define Momentti_3 #18.37
$define Momentti_4 #18.38
$define Momentti_5 #18.39
;                t1/On t2/Off, Relay 2
$define t1_On #18.40
;                Analogia inputit
$define nref #01.37
$define Tref_k_T1 #18.12
$define T2 #18.14
;$define t1 #18.18
;$define t2 #18.16
;
$define eteen #6.30
$define taakse #6.32
$define tahti #18.41
$define CTIU_Momentti #19.13
```

```
$define CTIU_Nopeus #19.15
$define CTIU_Tila #19.17
INITIAL{
t1=0
t2=0
#2.04=1                    ; rampin valinta = fast ramp
#7.10=18.13                ; analogiatulon 1 kohde
#7.14=1.37                 ; analogiatulon 2 kohde
#1.14=2                    ; analogiatulo 2 =  nopeusohje
#7.11=0                    ; analogiatulon 2 toimintatapa = jännitetulo
#7.15=0                    ; analogiatulon 3 toimintatapa = jännitetulo
#7.18=18.11                ; analogiatulon 3 kohde
#16.15=18.15               ; analogiatulon 4 kohde
#16.18=18.17               ; analogiatulon 5 kohde
#7.19=3.26                 ; anlogialähdön 1 lähdeparametri =
                           moottorin pyörimisnopeus
#7.22=18.19                ; anlogialähdön 2 lähdeparametri =
                           moottorin virta*160=momentti Nm
#6.04=2                    ; sekvenssikäytön valinta = wire proof
                           mode
#8.09=1                    ; liittimen 30 toiminta = käyntilupa
#6.34=1                    ; invertoi torg input
#17.20=1                   ; menu 20 parametrien tallennus
                           käyttöjännitteen katkaisussa = päällä
#8.13=18.34                ; liittimen 25 (F2) tulon kohde = prog
                           asento
#8.16=18.33                ; liittimen 26 (F3) tulon kohde =
                           generaattorikäyttö
#8.19=6.30                 ; liittimen 27 (F4) tulon kohde = käy eteen
#8.21=6.32                 ; liittimen 28 (F5) tulon kohde = käy taakse
#16.21=18.37               ; liittimen 48 (F7) tulon kohde = momentti 3
                           asento (T=kn*n)
#16.24=18.38               ; liittimen 49 (F8) tulon kohde = momentti 4
                           asento (T=k/n)
#16.30=18.39               ; liittimen 44 (F10) tulon kohde = momentti
                           5 asento (Duty-type)
#16.32=18.35               ; liittimen 45 (F11) tulon kohde = momentti
                           1 asento (T=Tref)
#16.34=18.36               ; liittimen 46 (F12) tulon kohde = momentti
                           2 asento (T=kn)
#16.36=18.40               ; liittimen 40 (Relay 2) lähdön kohde =
                           t1/On  t2/Off
#19.12=1500                ; trendinäytön momentin skaalaus maks 15
                           Nm
#19.14=3000                ; trendinäytön nopeuden skaalaus maks
                           3000 rpm
kerta1%=0
kerta2%=0
kerta3%=0
kello1%=0
kello2%=0
tyo1%=0
```

```
tyo2%=0
suodin%=0
#18.41=0                        ;nopeussäädössä momenttisäätö
}


BACKGROUND{
top:
//#1.09=1                       offset nopeusohjeelle valittu
//#1.04=-3                      ;offsetin arvo nopeusohjeelle

IF #19.30=0 THEN                                ; Normaali käyttö, ohjelmaa ei
                                                kytketty
        IF Toiminta_G=0 AND tahti=0 THEN
                                                ; vain nopeuskäyttö valittuna
                #7.14=19.11                     ; analogiatulon 2 kohde
                                                (nopeusohje nref käytössä)
                #1.14=3
                #1.15=1
                IF #19.11<64 AND #19.11>=-64  THEN
                        #1.21=0
                        ELSE
                        #1.21=INT(#19.11/10.666666)
                ENDIF
                #8.20=0                         ; liittimen 27 (F4) tulo ei
                                                invertoitu (6.30=0)
                #7.18=0.0                       ; analogiatulon 3 kohde
                                                (momenttiohje Tref pois
                                                käytöstä)
                #4.07=175                       ; maksimi virtaraja = 175%
                vaihto%=1
        ENDIF

        IF Toiminta_G=0 AND tahti=1THEN
                                                ; tahtikoneajo valittu ja
                                                 nopeuskäyttö
                #7.14=19.11                     ; analogiatulon 2 kohde
                                                (nopeusohje nref käytössä)
                #1.14=3
                #1.15=1
                IF #19.11<64 AND #19.11>=-64  THEN
                        #1.21=0
                        ELSE
                        #1.21=INT(#19.11/10.666666)
                ENDIF
                #8.20=0                         ; liittimen 27 (F4) tulo ei
                                                 invertoitu (6.30=0)
                #7.18=18.11                     ; analogiatulon 3 kohde
                                                (momenttiohje Tref pois
                                                käytöstä)
```

```
                    #4.07=FLOAT(#18.11)/320
                                        ; maksimi virtaraja skaalattu
                                        0...100 % analogiatulo 3
                    vaihto%=2
          ENDIF

                    IF Toiminta_G=1 THEN
                    #1.14=1
                    #7.18=18.11         ; analogiatulon 3 kohde
                                        (momenttiohje Tref käytössä)
                    #8.20=1             ; liittimen 27 (F4) tulo invertoitu
                                        (6.30=1)
                    #7.14=0.0           ; analogiatulon 2 kohde
                                        (nopeusohje nref pois
                                        käytöstä)

                    Tref_k_T1=FLOAT(#18.11)/320
                                        ; analogiatulon 3 skaalaus
                                        0...100% (18.11 integer
                                        0...32000)
                    T2=FLOAT(#18.13)/182.857
                                        ; analogiatulon 1 skaalaus
                                        0...175% (18.13 integer
                                        0...32000)


                    ;t2=((FLOAT(#18.15)/999)*1000)/32
                                        ; analogiatulon 4 skaalaus
                                        0...999s (18.15 integer
                                        0...32000)


                    ;t1=((FLOAT(#18.17)/999)*1000)/32
                                        ; analogiatulon 5 skaalaus
                                        0...999s (18.17 integer
                                        0...32000)


                    vaihto%=3
          ENDIF
          apu8=Tref_k_T1

          IF #18.20=0 THEN #18.20=140   ; k/n k:n skaalaus oletusarvo
          IF #18.21=0 THEN #18.21=10    ; k*n k:n skaalaus oletusarvo
          IF #18.22=0 THEN #18.22=10    ; k*n*n k:n skaalaus oletusarvo
          IF Toiminta_G=1 AND Momentti_1=1 THEN
                     #4.07=Tref_k_T1*1.75  ; Virtaraja =Tref
                    ELSEIF Toiminta_G=1 AND Momentti_2=1 THEN

#4.07=MIN(175,((Tref_k_T1*(ABS(#3.02)))/1714.29)*#18.21/10)
                                        ; Virtaraja =k*n
                    ELSEIF Toiminta_G=1 AND Momentti_3=1 THEN
```

```
#4.07=MIN(175,((Tref_k_T1*(ABS(#3.02)*ABS(#3.02)))/5142857.143)*#18.22/1
0)
                                                    ; Virtaraja =k*n*n
                                ELSEIF Toiminta_G=1 AND Momentti_4=1 THEN

#4.07=MIN(175,((Tref_k_T1*1.75)*#18.20/MAX(ABS(#3.02),0.1)))
                                                    ; Virtaraja =k/n
                        ELSEIF Toiminta_G=1 AND Momentti_5=1 AND #8.07=1
THEN
                         CALL Tapa:                 ; Duty-type valittu ja START
                                                    päällä
                        ELSEIF Toiminta_PROG=1 THEN
                        CALL Program:               ; PROG valittu
                  ENDIF

ENDIF

IF #19.30=1 THEN                                    ; ohjelma käytössä,
                                                    analogiatulo 2 pois käytöstä
                                                    ; vain nopeuskäyttö valittuna
                            #7.14=0.0               ; analogiatulon 2 kohde
                                                    (nopeusohje nref pois
                                                    käytöstä)
                            #1.14=3
                            #1.15=1
                            #8.20=0                 ; liittimen 27 (F4) tulo ei
                                                    invertoitu (6.30=0)
                            #7.18=0.0               ; analogiatulon 3 kohde
                                                    (momenttiohje Tref pois
                                                    käytöstä)
                            #4.07=175               ; maksimi virtaraja = 175%
                            vaihto%=1

ENDIF

IF #19.30=2 THEN                                    ; ohjelma käytössä nopeus ja
                                                    momentti, analogiatulot 2 ja 3
                                                    pois käytöstä
                                                    ; tahtikoneajo valittu ja
                                                    nopeuskäyttö
                            #7.14=0.0               ; analogiatulon 2 kohde
                                                    (nopeusohje nref pois
                                                    käytöstä)
                            #1.14=3
                            #1.15=1
                            #8.20=0                 ; liittimen 27 (F4) tulo ei
                                                    invertoitu (6.30=0)
                            #7.18=0.0               ; analogiatulon 3 kohde
                                                    (momenttiohje Tref käytössä)
```

```
                                                                ;
#4.07=FLOAT(#18.11)/320 maksimi virtaraja skaalattu 0...100 % analogiatulo 3
                        vaihto%=2

ENDIF

IF #19.30=3 THEN                                ; G käyttö
            IF Toiminta_G=1 THEN
                        #1.14=1
                        #7.18=0.0               ; analogiatulon 3 poiskäytöstä
                                                (momenttiohje Tref käytössä)
                        #8.20=1                 ; liittimen 27 (F4) tulo invertoitu
                                                (6.30=1)
                        #7.14=0.0               ; analogiatulon 2 kohde
                                                (nopeusohje nref pois
                                                käytöstä)

                        Tref_k_T1=FLOAT(#18.11)/320
                                                ; analogiatulon 3 skaalaus
                                                0...100% (18.11 integer
                                                0...32000)
                        T2=FLOAT(#18.13)/182.857
                                                ; analogiatulon 1 skaalaus
                                                0...175% (18.13 integer
                                                0...32000)

                                                ;t2=((FLOAT(#18.15)/999)
                                                *1000)/32
                                                ; analogiatulon 4 skaalaus
                                                0...999s (18.15 integer
                                                0...32000)

                                                ;t1=((FLOAT(#18.17)/999)
                                                *1000)/32
                                                ; analogiatulon 5 skaalaus
                                                0...999s (18.17 integer
                                                0...32000)

                        vaihto%=3
            ENDIF
            apu8=Tref_k_T1

            IF #18.20=0 THEN #18.20=140         ; k/n k:n skaalaus oletusarvo
            IF #18.21=0 THEN #18.21=10          ; k*n k:n skaalaus oletusarvo
            IF #18.22=0 THEN #18.22=10          ; k*n*n k:n skaalaus oletusarvo
            IF Toiminta_G=1 AND Momentti_1=1 THEN
                        #4.07=Tref_k_T1*1.75    ; Virtaraja =Tref
                        ELSEIF Toiminta_G=1 AND Momentti_2=1 THEN
```

```
#4.07=MIN(175,((Tref_k_T1*(ABS(#3.02)))/1714.29)*#18.21/10)
                                                    ; Virtaraja =k*n
                            ELSEIF Toiminta_G=1 AND Momentti_3=1 THEN

#4.07=MIN(175,((Tref_k_T1*(ABS(#3.02)*ABS(#3.02)))/5142857.143)*#18.22/1
0)                                                  ; Virtaraja =k*n*n
                             ELSEIF Toiminta_G=1 AND Momentti_4=1 THEN

#4.07=MIN(175,((Tref_k_T1*1.75)*#18.20/MAX(ABS(#3.02),0.1)))
                                                    ; Virtaraja =k/n
                      ELSEIF Toiminta_G=1 AND Momentti_5=1 AND #8.07=1
THEN
                       CALL Tapa:                   ; Duty-type valittu ja START
                                                    päällä
                      ELSEIF Toiminta_PROG=1 THEN
                      CALL Program:                 ; PROG valittu
                ENDIF

ENDIF
                IF #1.21>3000 THEN #1.21=3000     ;maks.ohjearvo 3000 rpm



                IF #4.07>175 THEN #4.07=175       ; Virtarajan maksimiarvon
                                                  rajoitus 175%


                GOTO top:
                }

                Tapa:{                            ; Duty-type aliohjelma
                            DO  WHILE  kello1%<t1-1 AND #18.39=1 AND
#8.07=1
                            #4.07=FLOAT(#18.11)/320
                                                  ; virtaraja 100% asetetusta
                            tyo1%=1
                            t1_On=1

                                                  ; Relay 2 ulostulo =1
                LOOP
                tyo1%=0
                DO  WHILE  kello2%<t2-1 AND #18.39=1 AND #8.07=1
                                                  ;#4.07=FLOAT(#18.13)
                                                  /182.857
                                                  ; Virtaraja =T2 175% maks
                            #4.07=FLOAT(#18.13)/320
                                                  ; Virtaraja =T2 100%
                                                  maks.virrasta
                            tyo2%=1
                            t1_On=0               ; Relay 2 ulostulo =0
                LOOP
```

```
tyo2%=0
}

Program:{                                   ; PROG aliohjelma
dummy%=0
}

CLOCK{
IF #18.39=1 AND tyo2%=0 THEN
                kello1%=kello1%+1
                kello2%=0
ENDIF
IF #18.39=1 AND tyo1%=0 THEN
                kello2%=kello2%+1
                kello1%=0
ENDIF
IF #18.39=0 THEN
                kello1%=0
                kello2%=0
                tyo1%=0
                tyo2%=0
ENDIF

                                            ;IF kello1%>#19.20 THEN
IF kello1%>t1 THEN
                kello1%=0
ENDIF
                                            ;IF kello2%>#19.21 THEN
IF kello2%>t2 THEN
                kello2%=0
ENDIF
t2=((FLOAT(#18.15)/999)*100000)/32
                                            ; analogiatulon 4 skaalaus
                                            0...999s (18.15 integer
                                            0...32000)


t1=((FLOAT(#18.17)/999)*100000)/32
                                            ; analogiatulon 5 skaalaus
                                            0...999s (18.17 integer
                                            0...32000)
}

ENCODER{


suodin%=suodin%+1


IF suodin%>100 THEN                         ; filtteri näytöille 5.52
                                            ms*100=552 ms välein näyttö
```

```
                #18.19=FLOAT(160*#4.02)
                                        ; Momentti näyttö
                CTIU_Momentti=ABS(#18.19)
                                        ; Momentti CTIU:n
                                        trendinäytölle
                CTIU_Nopeus=ABS(#3.02)
                                        ; Nopeus CTIU:n trendinäytölle
                suodin%=0
ENDIF
IF Toiminta_G=0 AND #6.32=0 AND Toiminta_PROG=0 THEN
                                        ; Tilat CTIU:n näytölle
                CTIU_Tila=1
                ELSEIF Toiminta_G=0 AND #6.32=1 THEN
                CTIU_Tila=2
                ELSEIF Toiminta_G=1 AND  Momentti_1=1 THEN
                CTIU_Tila=3
                ELSEIF Toiminta_G=1 AND  Momentti_2=1 THEN
                CTIU_Tila=4
                ELSEIF Toiminta_G=1 AND  Momentti_3=1 THEN
                CTIU_Tila=5
                ELSEIF Toiminta_G=1 AND  Momentti_4=1 THEN
                CTIU_Tila=6
                ELSEIF Toiminta_G=1 AND  Momentti_5=1 THEN
                CTIU_Tila=7
                ELSEIF Toiminta_PROG=1 THEN
                CTIU_Tila=8
ENDIF

IF vaihto%=1 AND kerta1%=0 THEN
                                        ; valittuna nopeus ja  ei tahti
                kerta1%=1
                kerta2%=0
                kerta3%=0
                #10.38 = 100

                                        ; reset koska inputin kohde
                                        muuttuu
ENDIF

IF vaihto%=2 AND kerta2%=0 THEN
                                        ; valittuna nopeus ja tahti
                kerta1%=0
                kerta2%=1
                kerta3%=0
                #10.38 = 100

                                        ; reset koska inputin kohde
                                        muuttuu
ENDIF

IF vaihto%=3 AND kerta3%=0 THEN
                                        ; valittuna momentti
```

```
                kerta1%=0
                kerta2%=0
                kerta3%=1
                #10.38 = 100

                                        ; reset koska inputin kohde
                                        muuttuu
ENDIF

}
```

**LIITE 2**

Käyttöohje

**SKS CONTROL TECHNIQUESIN SERVOMOOTTORIKÄYTÖN VALVOMO-OHJELMISTO**

1. TURVALLISUUS OHJEET ..... 2
1.1. Ennen ohjelman käynnistystä ..... 2
1.2. Ohjelman käyttö ..... 2
1.3. Käytön aloitus ..... 2
2. ASENNUS ..... 3
3. OHJELMAN KÄYTTÖ ..... 4
3.1. Yhteyden muodostus ..... 4
3.2. Pääkäyttöliittymä ..... 5
3.3. Parametrit ..... 6
3.3.1. Parametrien muokkaus ..... 7
3.4. Monitorointi-tila ..... 8
3.5. Aseta / Muokkaa ajoarvot ..... 9
3.5.1. Ajoarvojen asetus ohjelmasta käsin ..... 9
3.5.2. Ajoarvojen lisäys Excel-tiedostosta ..... 10
3.6. Aja ajoprofiili ..... 11

# 1. TURVALLISUUS OHJEET

## 1.1. Ennen ohjelman käynnistystä

Ohjelmisto on tehty käytettäväksi vain SKS Control Techniquesin UNI2402 taajuusmuuttajan kanssa.

Varmista ennen ohjelman käynnistämistä että, moottorin läheisyydessä ei ole ylimääräisiä esineitä tai ihmisiä.

Varmista ennen ohjelman käynnistämistä että, moottorin akseli pääsee vapaasti pyörimään eikä voi tarttua kiinni mihinkään.

## 1.2. Ohjelman käyttö

Ohjelmiston käytössä tulee huomioida moottorin käyttörajoitukset.

Ohjelmistolla käytössä tulee välttää ääripäästä toiseen menemistä. Esimerkiksi moottorin nopeutta ja pyörimissuuntaa muutettaessa tulee välttää vastakkaisten pyörimissuuntien maksimi nopeuksia pienellä aika erolla. Ohjelmistolla ei tule ajaa -3000 rpm:stä +3000 rpm:n 100ms erolla.

Ennen tietojen siirtämistä, sulje kaikki Excel-sovellukset.

## 1.3. Käytön aloitus

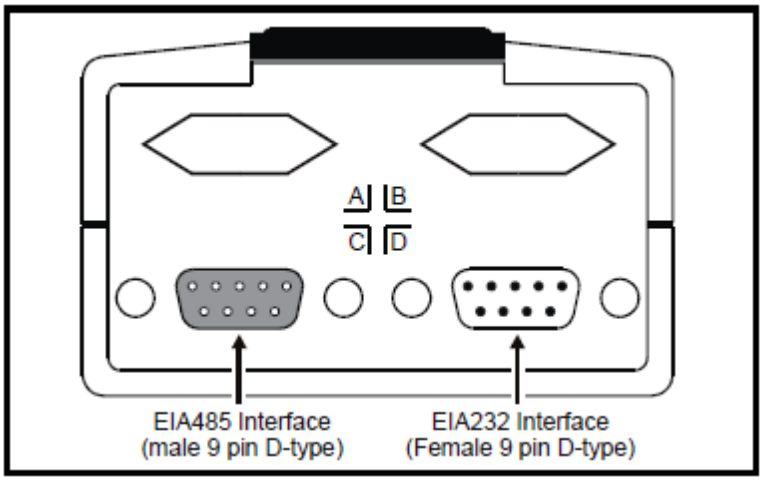
1. Tutustu ohjeisiin ennen ohjelman käyttöönottoa.
2. Kytke kaapeli taajuusmuuttajaan
3. Käynnistä taajuusmuuttaja
4. Käynnistä ohjelma
5. Älä sammuta taajuusmuuttajaa ennen kuin olet sulkenut ohjelman

## 2. ASENNUS

| PC | | PL1/UD70/RS425 - LIITIN | |
|---|---|---|---|
| RXD | 2 ———— | 2 | $\overline{Tx}$ |
| TXD | 3 ———— | 3 | $\overline{Rx}$ |
| 0V | 5 ———— | 6&7&1 | Tx&Rx |

**Kuva 2.1: Räätälöidyn D-9 liittimen kytkentä**

Yhteys taajuusmuuttajaan muodostetaan kuvassa (Kuva 2.1) esitetyllä räätälöidyllä kaapelilla. Kaapeli kytketään seuraavassa kuvassa esitettyyn EIA232 liittimeen. Yhteys muodostetaan neljä johtimisen EIA232 ANSI slave-protokollan avulla.

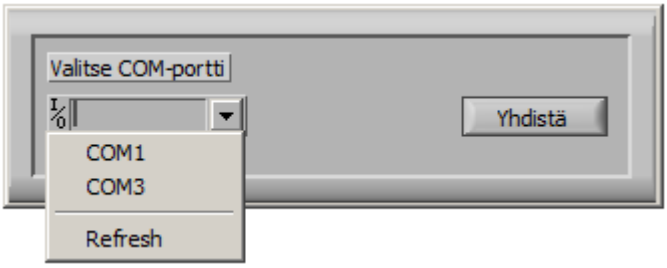



**Kuva 2.2: Sarjakaapelin liitäntä taajuusmuuttajaan**

# 3. OHJELMAN KÄYTTÖ

Tässä osassa esitellään ohjelman käyttö, yhteyden muodostus, parametrien muokkaus, monitorointi ja ajoarvojen ajo.

## 3.1. Yhteyden muodostus

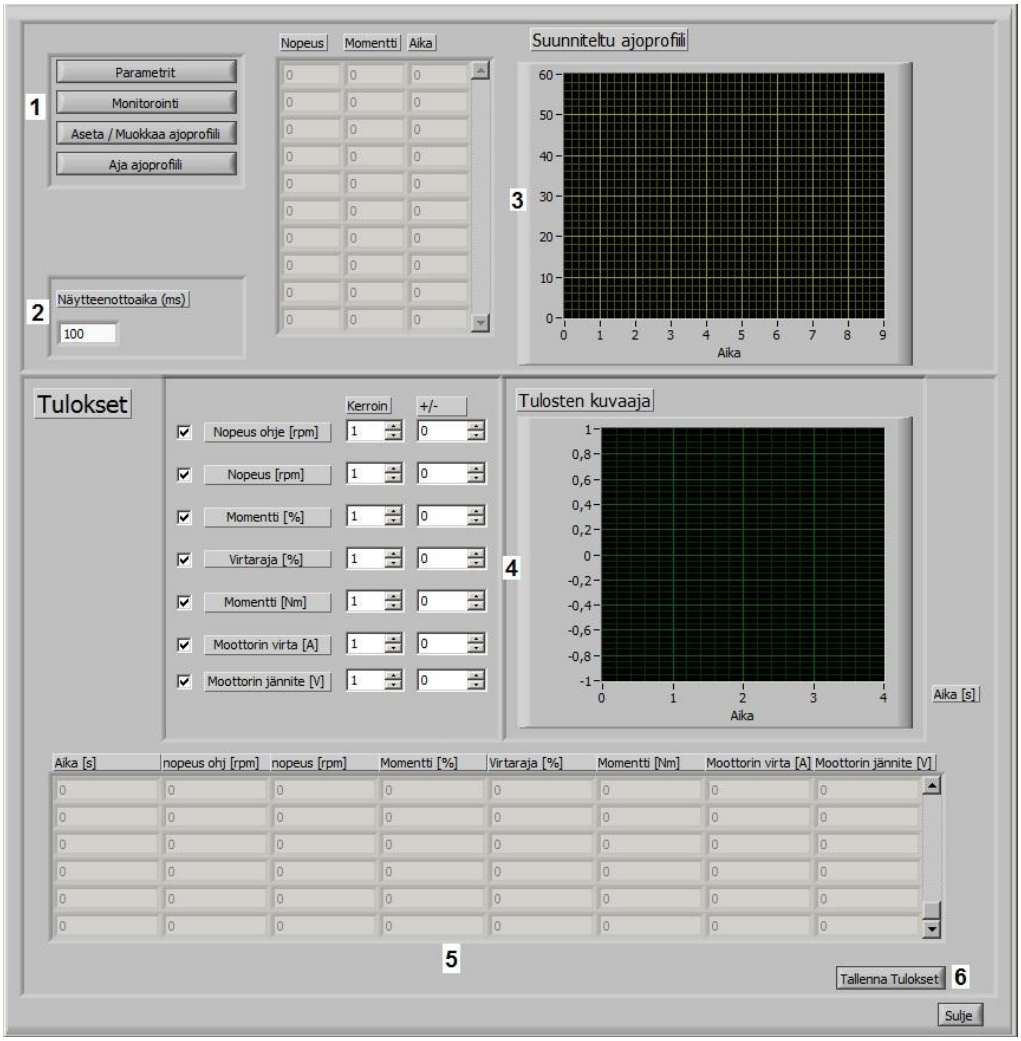



**Kuva 3.1: Yhteyden muodostus**

Ohjelman käynnistyessä avautuu yllä esitetyn kuvan (Kuva 3.1) mukainen ikkuna. Tässä ikkunassa valitaan COM-portti jolla tietokone on liitetty taajuusmuuttajaan.

### 3.2. Pääkäyttöliittymä

Yhteyden muodostuksen jälkeen avautuu pääkäyttöliittymä, joka on esitetty seuraavassa kuvassa (Kuva 3.2).




**Kuva 3.2: Pääkäyttöliittymä**

Pääkäyttöliittymän osat:

1 Taajuusmuuttajan hallintaan käytettävät painikkeet.

2 Asetetun ajoprofiilin kuvaaja ja arvot.

3 Näytteenottoajan asetus.

4 Tulosten kuvaaja ja kuvaajan hallinta työkalut.

5 Mitatut ajoarvot.

6 Tulosten tallennus Excel-taulukkolaskenta ohjelmaan.

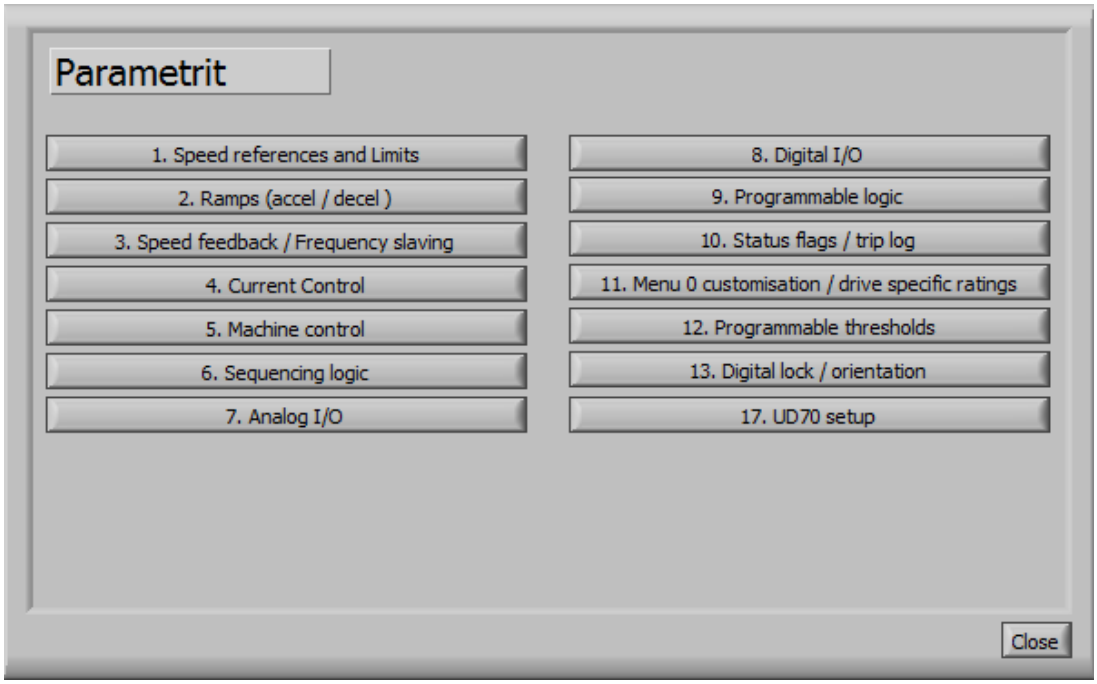
Pääkäyttöliittymässä on esitetty ohjelman kannalta tärkeimmät toiminnot. Yläosassa on taajuusmuuttajan hallintaan käytettävät työkalut ja alaosa on tulosten tarkastelua varten.

Kohdassa 1 on hallintaan käytettävät painikkeet, joiden avulla voidaan navigoida ohjelmassa, sekä käynnistys painike monitorointi tilalle ja ajoprofiilin ajolle.

Kohdan 2 avulla voidaan muokata näytteenottoaikaa. Tämä kohta vaikuttaa miltei koko ohjelman toimintaan. Se määrittelee ajoarvojen lähetysvälin ja mittausten otto välin. Tämä aika määrittelee myös ajoarvojen asettamisen minimi ajan muutoksen ($\Delta t$). Näytteenottoajan asetus tulee olla minimissään 100 ms ja myös jaollinen 100:lla.

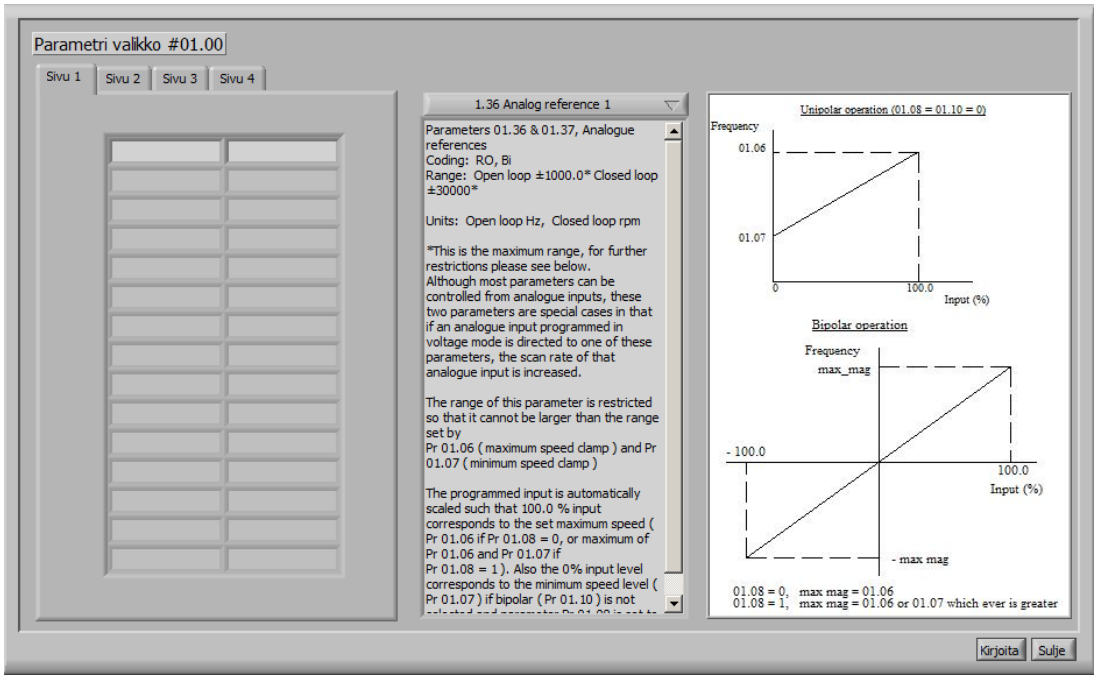
### 3.3. Parametrit

Parametri valikkoon päästään kuvan 2 kohdan 1 kautta. Tästä avautuu parametri valikko (Kuva 3.3).




**Kuva 3.3: Parametrit-valikko**

Parametrit on jaettu ryhmiin taajuusmuuttajan oman asettelun mukaisesti.

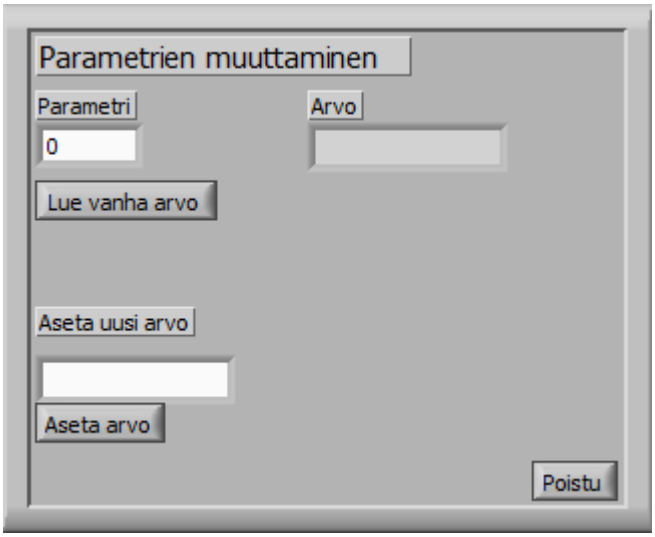



**Kuva 3.4: Parametrivalikko #01.00**

Yllä olevassa kuvassa (Kuva 3.4) on esitetty parametri ryhmä 1, nopeusohjeet ja rajat. Vasemmassa reunassa näkyvässä taulukossa esitetään parametrit ja niiden arvot. Keskellä olevan vetovalikon avulla saadaan lisätietoa parametrista.

### 3.3.1. Parametrien muokkaus

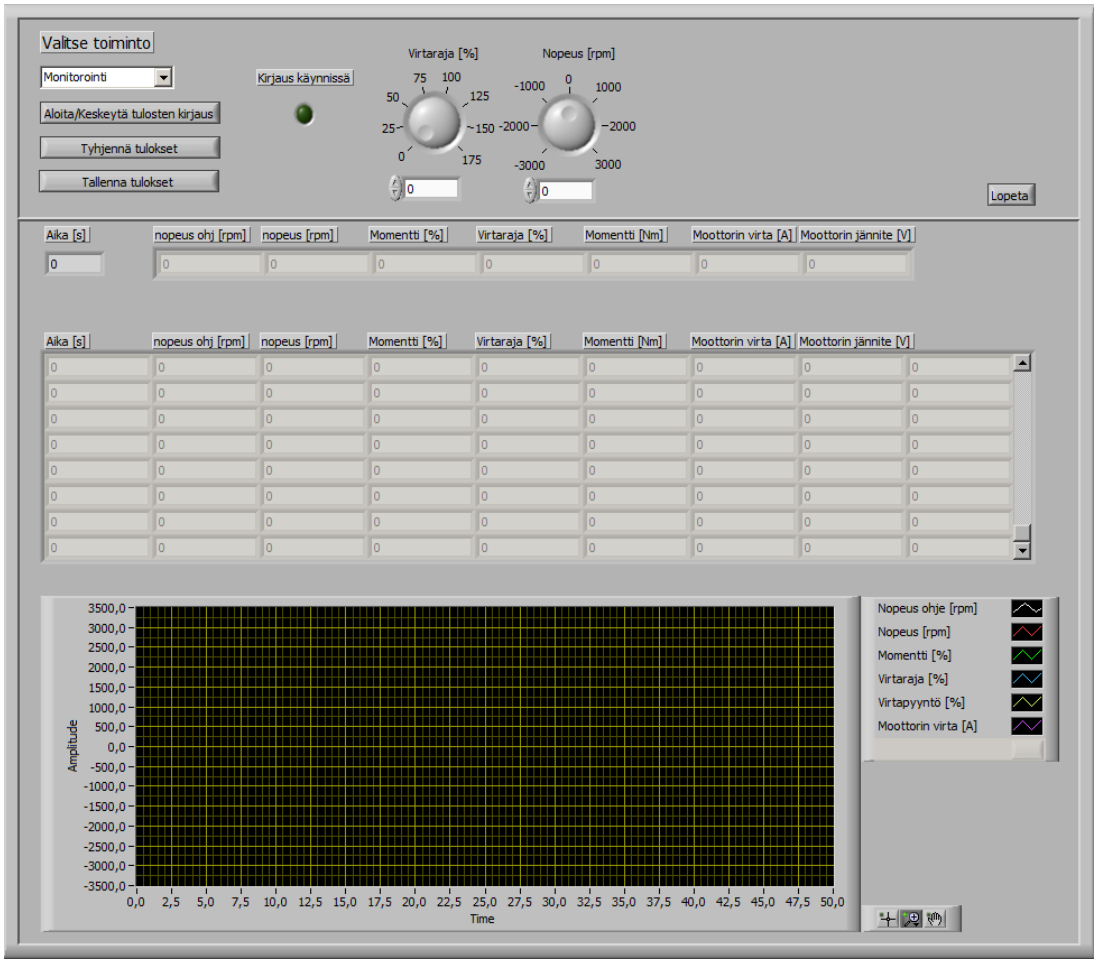
Painamalla oikeassa alalaidassa olevaa kirjoita painiketta, voidaan muokata parametrien arvoja. Kaikkia parametreja ei voida muokata. Osa parametreista on niin kutsuttuja RO- arvoja, eli Read Only (vain luku) arvoja.




**Kuva 3.5: Parametrien kirjoitus**

Parametri muutetaan syöttämällä muutettava parametri kohtaan ”parametri” ja painamalla painiketta ”Lue vanha arvo”. Tämän jälkeen ohjelma tulostaa parametrin tämän hetkisen arvon. Syötetään uusi arvo kohtaan ”aseta uusi arvo” ja painetaan ”aseta arvo”-painiketta. Mikäli parametri on RO-arvo, ohjelma tulostaa tekstin ”aseta uusi arvo” kohtaan. RO-arvoa ei voida muuttaa.

### 3.4. Monitorointi-tila




**Kuva 3.6: Monitorointi-tila**

Monitorointi tilan avulla voidaan tarkastella taajuusmuuttajan toimintaa, sekä ohjata taajuusmuuttajaa.

Oikeassa yläkulmassa olevan vetovalikon avulla voidaan valita monitorointi-ikkunan toiminto. Monitorointi tilassa ohjaus on taajuusmuuttajan ohjauspulpetilla. Ohjelma vain kerää talteen tiedot laitteelta.

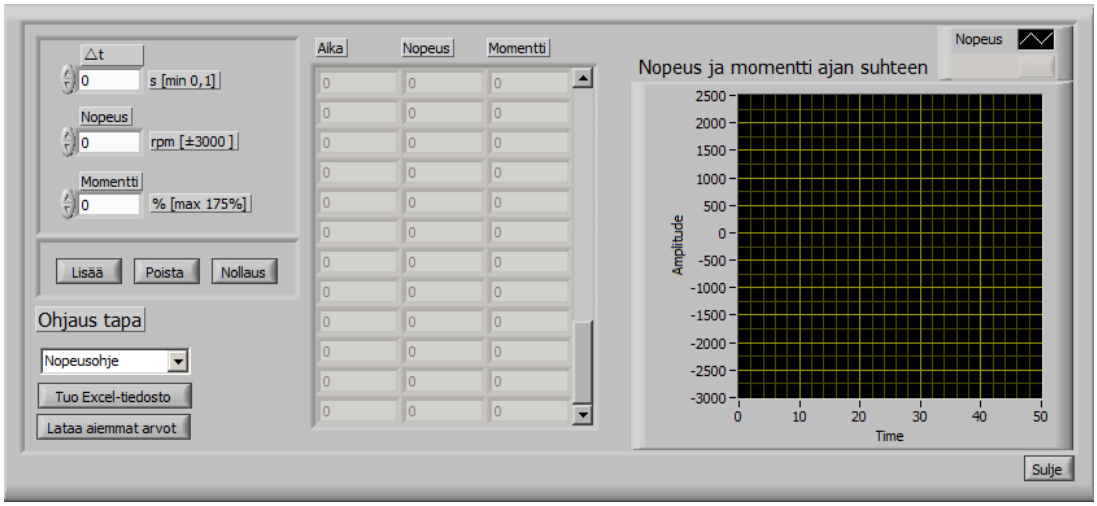
Nopeusohjaustilassa voidaan ohjata moottorin nopeutta. Nopeutta ohjataan joko syöttämällä nopeus numero arvona oikean puoleisen säätimen tekstikenttään, tai suoraan säätimestä.

Momenttiohjaustilassa säädetään moottorin maksimi momenttia, mutta moottorin nopeusohjaus toteutetaan ohjauspulpetin avulla. Momenttia säädetään joko momenttisäätimestä tai sen tekstikentän avulla.

Nopeus- ja momenttiohjaus tilassa molempia säädetään ohjelman säätimistä tai niiden tekstikenttien avulla.

Painettaessa ”Aloita/Keskeytä tulosten kirjaus”-painiketta, ohjelma kerää mitattavat arvot alla olevaan taulukkoon ja piirtää arvojen mukaan myös kuvaajaa. Nämä arvot jäävät taulukkoon ja voidaan siirtää Excel-taulukkoon painamalla ”Tallenna tulokset”- painiketta. Taulukko tyhjennetään painamalla ”Tyhjennä tulokset”- painiketta.

### 3.5. Aseta / Muokkaa ajoarvot



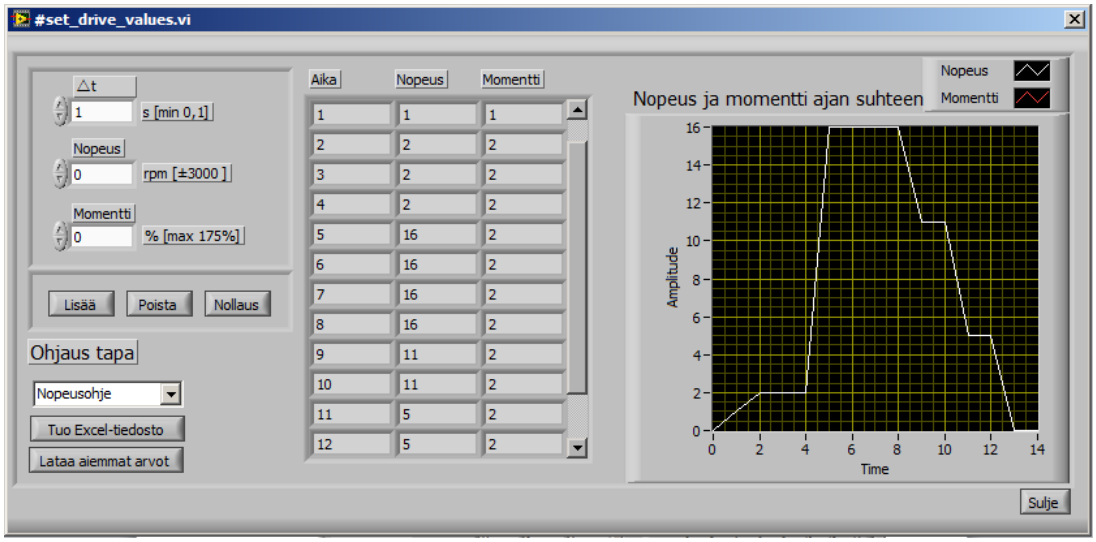

**Kuva 3.7: Aseta/muokkaa ajoarvot**

Tässä tilassa tapahtuu ajoarvojen asetus. Ajoarvot voidaan asettaa nopeusohjeena, momenttiohjeena tai nopeus- ja momentti ohjeena.

#### 3.5.1. Ajoarvojen asetus ohjelmasta käsin

Ajoarvojen lisäys käsin, tapahtuu syöttämällä ohjelmaan ajan muutos Δt, nopeus ja momentti. Aika syötetään muutoksena edelliseen aikaan. Arvot lisätään painamalla ”lisää”-painiketta. Ohjelma muuttaa itse ajan muutoksen ajan hetkeksi.

”Poista”-painike poistaa viimeisen lisätyn rivin. Nollaus tyhjentää koko taulukon.




**Kuva 3.8: Ajoarvojen asetus**

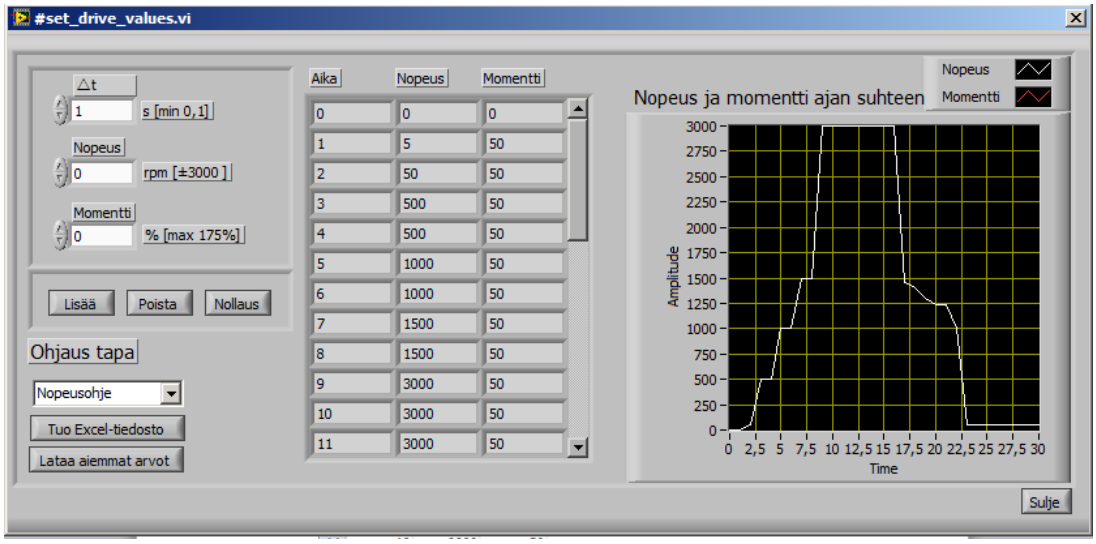
### 3.5.2. Ajoarvojen lisäys Excel-tiedostosta

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Aika | Nopeus | Momentti | |
| 2 | 1 | 5 | 50 | |
| 3 | 2 | 50 | 50 | |
| 4 | 3 | 500 | 50 | |
| 5 | 4 | 500 | 50 | |
| 6 | 5 | 1000 | 50 | |
| 7 | 6 | 1000 | 50 | |
| 8 | 7 | 1500 | 50 | |
| 9 | 8 | 1500 | 50 | |
| 10 | 9 | 3000 | 50 | |
| 11 | 10 | 3000 | 50 | |
| 12 | 11 | 3000 | 50 | |
| 13 | 12 | 3000 | 50 | |
| 14 | 13 | 3000 | 50 | |
| 15 | 14 | 3000 | 50 | |
| 16 | 15 | 3000 | 50 | |
| 17 | 16 | 3000 | 50 | |
| 18 | 17 | 1455 | 50 | |
| 19 | 18 | 1400 | 50 | |

**Kuva 3.9: Lisättävä Excel-tiedosto**

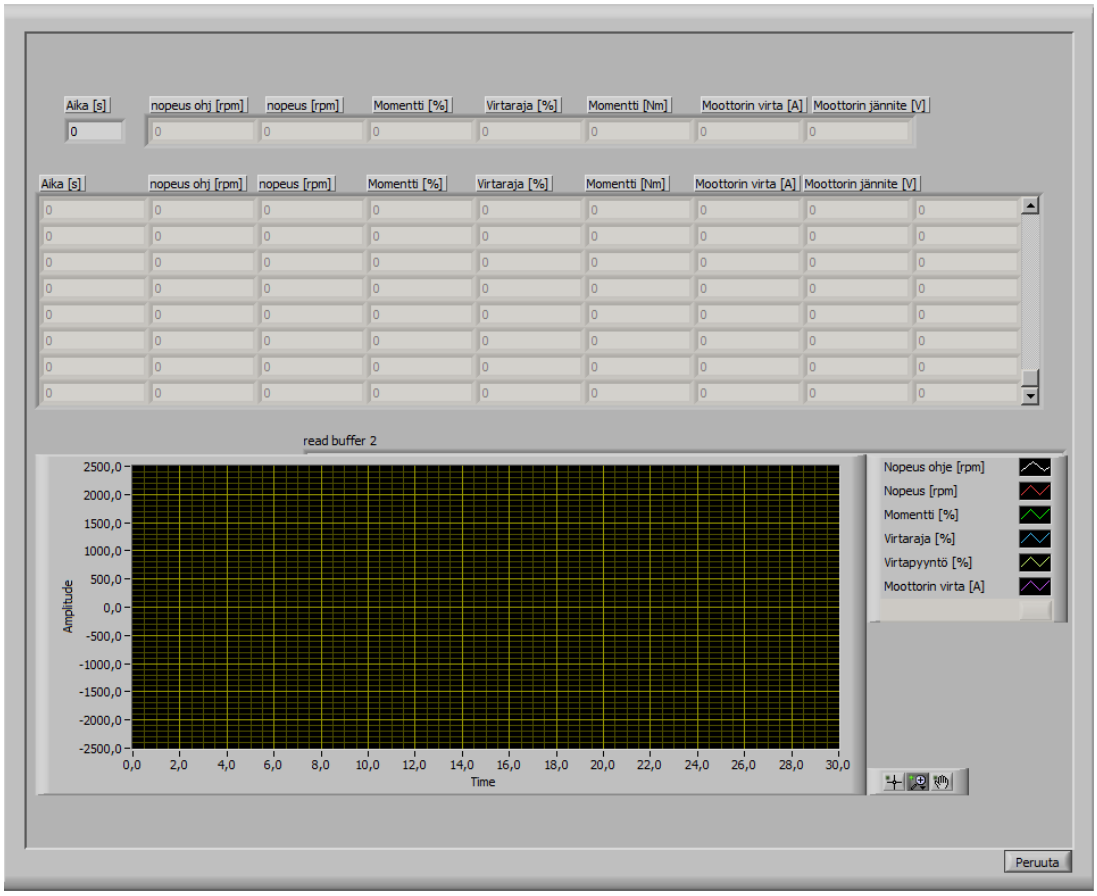
Ajoarvojen tuonti Excel-tiedostosta tapahtuu painamalla ”Tuo Excel-tiedosto”-painiketta. Excel-tiedostossa tietojen pitää olla ohjelman mukaisilla sarakkeilla, eli järjestyksessä aika, nopeus ja momentti. Kun kyseessä on pelkkä momenttiohje, jätetään nopeus sarake tyhjäksi. Ohjelma aloittaa tietojen lukemisen riviltä 2.

**!!! SULEJE EXCEL-OHJELMA ENNEN TIETOJEN SIIRTOA OHJELMAAN. OHJELMA SULKEE KOKO EXCEL-OHJELMAN TIETOJA LUETTAESSA!!!**




**Kuva 3.10: Excel-ohjelmasta tuodut arvot**

## 3.6. Aja ajoprofiili




**Kuva 3.11: Aja ajoprofiili**

Aja ajoprofiili näyttää vain ajon edistymisen. Tämä tila on vain tarkkailua varten. Kun ajoarvot on ajettu, ohjelma siirtää tulokset pääkäyttöliittymään.

Aja ajoprofiili ikkunan yläosassa näytetään tämän hetkiset mittaustulokset. Taulukossa näkyy aiemmat mittaustulokset. Tarkempi tulosten tarkastelu tehdään pääkäyttöliittymässä, tai myöhemmin Excel-ohjelmassa.