ScandiCAT™

Control system manual

Revision history

Rev no	Author	Reviewer	Approver	Description
00	FH Sign	JG Sign	Sign	Initial
01	Sign	Sign	Sign	Update
02	Sign	Sign	Sign	

Table of contents

1 Introduction

The ScandiCAT control system is used to control a Scandinova modulator. It is based on the industrial communication bus EtherCAT in combination with a PLC main controller from Beckhoff and FPGA based nodes developed by Scandinova.

The external interface to the control system is an operator GUI and by default also a Modbus-TCP interface for remote control.

2 Modulator hardware

2.1 Overview

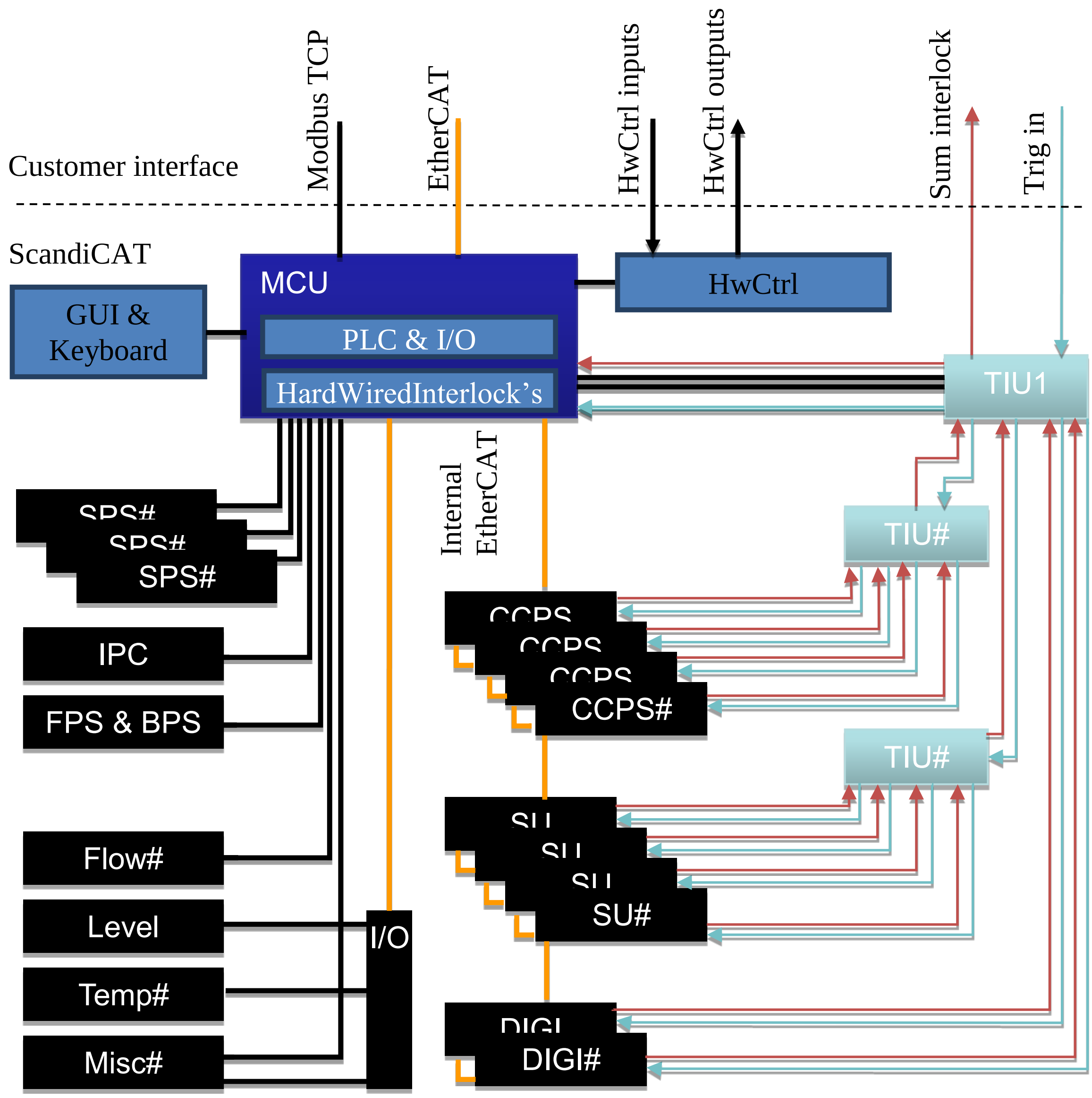


Image 1 An overview of the ScandiCAT control system.

Image 1 An overview of the ScandiCAT control system.Image 1shows an overview of the control system. At the top is the CPU, a Beckhoff embedded PC that runs the GUI and the PLC program. It controls and monitors the whole system.The actual units present and the number of each type of unit are different for different systems.

Item	Description	Chap
ModbusTCP	See chapter…	5.1
EtherCAT	See chapter…	5.2
HwCtrl inputs	See chapter…	2.3
HwCtrl outputs	See chapter…	2.3
Sum interlock	See chapter…	2.5
Trig in	See chapter…	2.5
GUI & Keyboard	See chapter…	4
PLC + I/O	See chapter…	2.1
Hardwired interlocks	See chapter…	2.4
SPS#	Solenoid Power Supply controlled via standard I/O and hard-wired circuits	2.4
IPC#	Ion Pump Controller controlled via standard I/O and hard-wired circuits	2.4
FPS	Filament Power Supply controlled via standard I/O and hard-wired circuits	2.4
BPS	Bias Power Supply controlled via standard I/O	-
FLOW#	Flow sensors controlled via standard I/O and hard-wired circuits	2.4
LEVEL	Oil level sensor	-
TEMP#	Temperature sensors	-
MISC#	Miscellaneous sensors	-
I/O	Distributed analog / digital in and output modules	-
CCPS#	Capacitor Charging Power Supplies (FPGA slave made by ScandiNova)	-
SU#	Switch Units (FPGA slave made by ScandiNova)	-
DIGI#	Digitizers (FPGA slave made by ScandiNova)	-
TIU1	Master Trig & Interlock distribution Unit (CPLD based, made by ScandiNova)	2.5
TIU#	Slave Trig & Interlock distribution Unit (CPLD based, made by ScandiNova)	2.5

Table 1 Brief introduction of all units / signals

2.2 Main Control Unit (MCU)

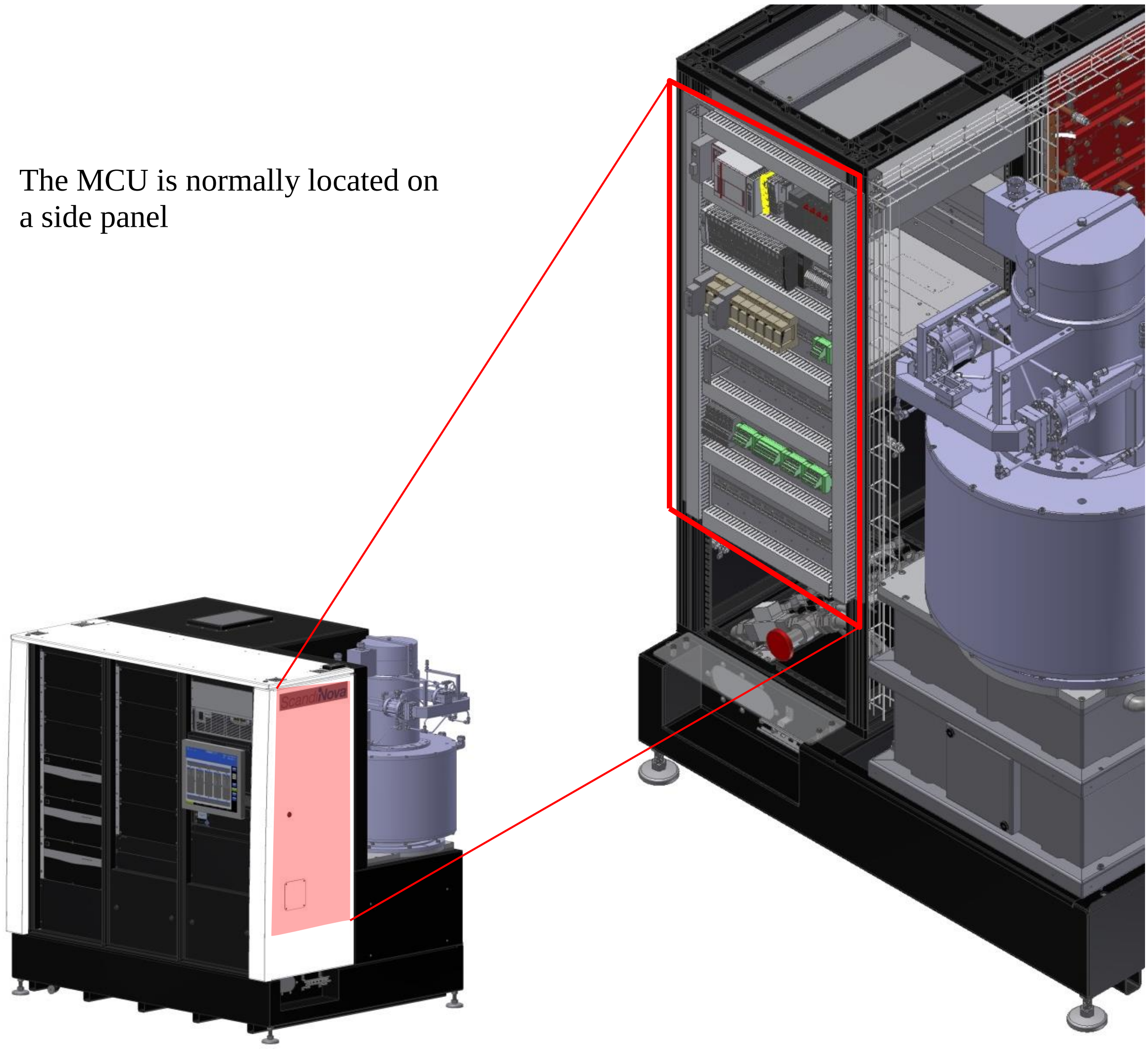


Image 2 Location and appearance of the MCU.

The MCU contains an embedded PC and its I/O units. The embedded PC has the following software installed:

- C:\ScandiCAT PLC — PLC program
- C:\TwinCAT — Soft PLC running in parallel with Windows
- C:\ScandiCAT GUI — Graphical User Interface made by ScandiNova
- C:\ScandiCAT FPGAs — Location of the FPGA firmware
- C:\Mbus — ModbusTCP test client

The PLC is the main unit of the control system. It controls all modulator hardware via EtherCAT communication and via digital and analog I/O:s. It makes a number of control variables available to the GUI and also to external systems via Modbus-TCP.

PC specification	
CPU	Beckhoff CX2030
OS	Windows Embedded Standard 7
Soft PLC (IEC 61131-3)	TwinCAT2

Table 2 PC Specification

2.2.1 Software protection

The PLC code is password protected during the warranty period if no other arrangements have been made.

The disk of the PC is locked so that only a few files can be written persistently (log files, settings files etc.). All other disk writes will be lost when the modulator is powered down. This means that the risk of a disk failure when the modulator is powered down is reduced to a minimum. It also means that everything in the operating system etc. will have the same state at each restart.

2.2.2 Time synchronization

The system time can be synchronized via NTP or SNTP. This is done in windows. See Microsoft documentation.

2.2.3 Runtime counters

Option: It is possible to have non-resettable mechanical runtime counters for:

- ***StandBy*** hours
- ***Hv*** or ***Trig*** hours

There is also a runtime counter for filament time in the software, accessible from the GUI in the *FpsVoltRead* dialog.

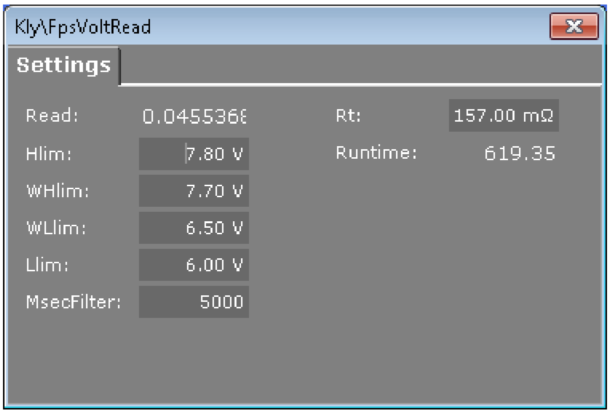


Image 3 The software runtime counter for the filament.

2.3 Hardwired Control (HwCtrl)

HwCtrl is the customer’s hardwired interface to the modulator.This interface is configurable and can contain different signals depending on customer needs.

General	
Location	Located in the MCU
Type of connection	By default the type of connection is spring-loaded wire terminals on a DIN-rail, other type of connectors are possible if requested
Isolation	All HwCtrl signals are galvanically isolated (500V) to the internal potentials of the modulator
Power supply	The customer provides external 24VDC to two spring-loaded wire terminals

Table 3 General properties of HwCtrl.

HwCtrl signal examples (the examples in the table below may not be present in all modulators)		
Name	**Type**	**Description**
External interlock	Digital input	An external interlock signal that can be used to stop the modulator by blocking trig and forcing CCPS to stop charging.
Trig enable	Digital input	Enables or disables trig without causing an interlock. Unlatched.
Hv status	Digital output	True when high voltage is enabled.
Trig status	Digital output	True when trig is enabled.
PLC trig out	Digital output	The internally generated trig signal. To use it, connect it to the electrical trig in on the master TIU. The Prf of this output is set on the main page, see section 0.
Sum alarm	Digital output	Sum interlock output.

Table 4 HwCtrl example signal

Electrical characteristics of HwCtrl I/O	
Digital inputs	OK: 11…30 V Interlock: -3…+5 V Input current: typ. 3 mA (IEC 61131-2, type 1/3) Type: Beckhoff EL1258
Digital outputs	Rated load voltage: 24 V DC (-15 %/+20 %) Load type: ohmic, inductive, lamp load Max. output current: 0.5 A (short-circuit-proof) per channel Type: Beckhoff EL2808

Table 5 Electrical characteristics of HwCtrl I/O

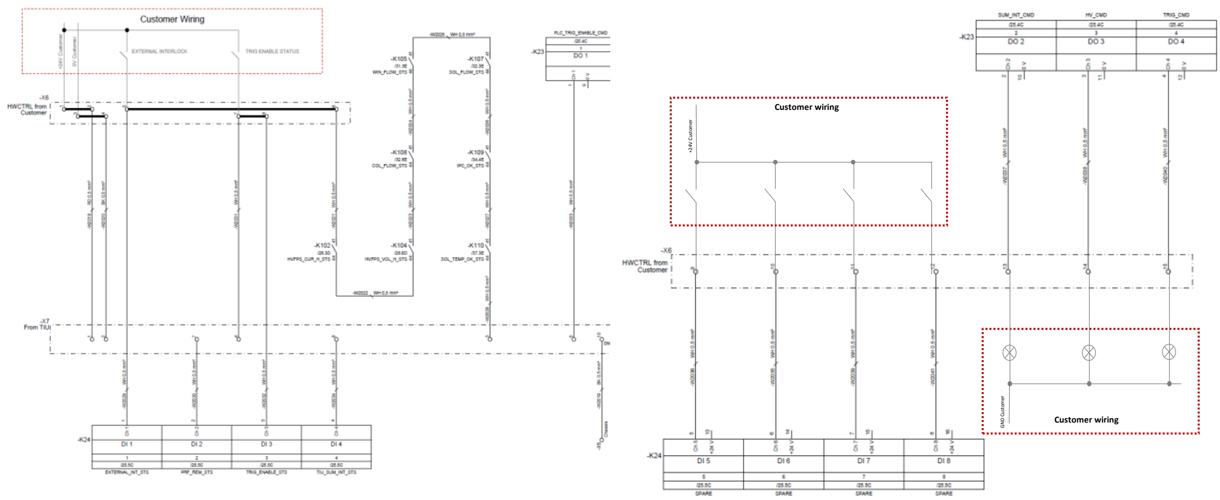


Image 4 Wiring examples

2.4 Hardwired interlocks

In addition to the PLC based interlocks, there are also some redundant hard wired interlock circuits. These cannot be turned off or be configured from the GUI.

The examples in the table below may not be present in all modulators.

Examples of hardwired interlocks		
Interlocked unit (hard interlock)	**Interlocked by**	**Corresponding SSM (will cause soft interlock of the whole modulator)**
All CCPS, SU, TIU	HwCtrl\External interlock	T&I\HwCtrlExtIntlkSts
TIU	Switch unit over current	Switch\SuSumSts
TIU, Filament PS	Ion PS vacuum set point relay	Kly\IonPsSetPointSts
TIU, Filament PS	Filament current monitoring relay	Kly\FpsHwCtrlSumSts
TIU, Filament PS	Filament voltage monitoring relay	Kly\FpsHwCtrlSumSts
TIU	Body flow status	Cool\BodyFlowSts
TIU	Collector flow status	Cool\CollectorFlowSts
TIU	Window flow status	Cool\WindowFlowSts
TIU Solenoid PS	Solenoid flow status	Cool\SolenoidFlowSts
Solenoid PS	Solenoid temperature switch	Cool\SolTempSwitch
Oil pump inverter	Oil pump temperature switch	Tank\OilPumpTempSts

Table 6 Hardwired interlock examples

For more information about PLC based interlocks, see Control logic.

2.5 Trig and interlock system (TI)

The trig and interlock system is made out of one or several Trig and Interlock Units (TIU). Each TIU can distribute optical trig and interlock signals to six other units, CCPS's, Switches, Digitizer's or more TIU's. A missing interlock signal from any connected unit will block the trig pulse to all connected units within 1-2µs.

A TIU also has an electrical external interlock input. The sum interlock signal from the master TIU is also sent to the PLC via an electrical signal to make sure that the whole system is aware of the interlock.

Image 5 Trig and Interlock unit

Signal descriptions				
Conn	**Pin**	**Signal**	**Description**	**Properties**
J1		SC_TRIG_IN	Optical trig input.	Fiber-Type: SC 820nm
J1		SC_SUM_INT_OUT	Optical sum interlock output.	Fiber-Type: SC 820nm
J2		BNC_TRIG_IN	Electrical trig input.	5-15VDC, 50Ω
J3	1	+24Vc	Opto-coupler power from customer	+24VDC
	2	0Vc	Opto-coupler power from customer	0V
	3	EXT_INT_IN	Customer External Interlock	24VDC = OK
	4	EXT_TRIG_ENA_IN	Customer External Trig Enable. Not latched.	24VDC = OK
	5	PLC_TRIG_ENA_IN	PLC External Trig Enable. Not latched.	24VDC = OK
	6	Spare input	-	-
	7	PRF_STS_OUT	Trig output to PLC. The pulse width of this signal is extended to 500µs.	0/ 24VDC
	8	SUM_INT_OUT	TIU sum interlock out to PLC.	24VDC = OK
	9	Spare output	-	-
J4…J9		SC_INT_1…6_IN	Interlock input from the connected unit.	Fiber-Type: SC 820nm
J4…J9		SC_TRIG_1…6_OUT	Trig output to the connected unit.	Fiber-Type: SC 820nm
J10	1	24VDC	Control system power	+24VDC
	2	0V	Control system power	0V

Table 7 TIU signal specification

2.5.1 Disabling the power supplies

The CCPS triggers are inverted (dark=trig). If the master “TIU\External interlock” signal is removed or the fiber link is broken the CCPS charging will be disabled in a hard-coded fashion within ~1sec.

2.5.2 Logic

The logic of a trig and interlock unit is hard-coded in a CPLD. It cannot be changed by software during runtime.

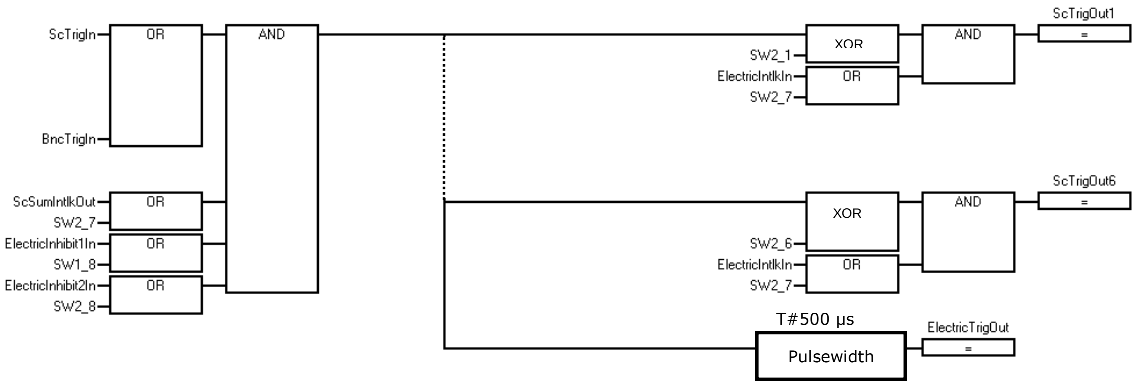


Image 6 Trig logic

The trig logic is shown in Image 6**Fel! Hittar inte referenskälla.**. The input trig signals are active high and the inhibit signals and interlock signals are active low.

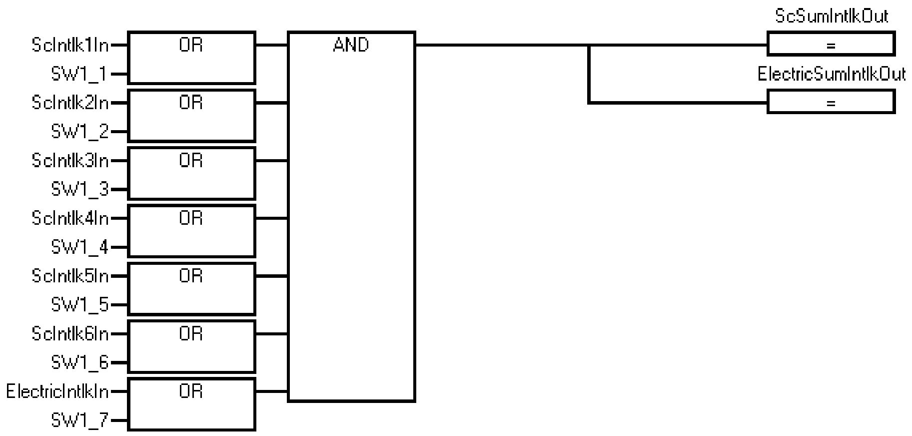


Image 7 Interlock logic

The interlock logic is shown in Image 7. The interlock signals are active low.

DIP switch settings		
Switch	**ON function**	**OFF function**
SW1_1	Disable ScIntlk1In	Enable ScIntlk1In
SW1_2	Disable ScIntlk2In	Enable ScIntlk2In
SW1_3	Disable ScIntlk3In	Enable ScIntlk3In
SW1_4	Disable ScIntlk4In	Enable ScIntlk4In
SW1_5	Disable ScIntlk5In	Enable ScIntlk5In
SW1_6	Disable ScIntlk6In	Enable ScIntlk6In
SW1_7	Disable ElectricIntlkIn	Enable ElectricIntlkIn
SW1_8	Disable ElectricInhibit1In	Enable ElectricInhibit1In
SW2_1	Invert ScTrigOut1	Do not invert ScTrigOut1
SW2_2	Invert ScTrigOut2	Do not invert ScTrigOut2
SW2_3	Invert ScTrigOut3	Do not invert ScTrigOut3
SW2_4	Invert ScTrigOut4	Do not invert ScTrigOut4
SW2_5	Invert ScTrigOut5	Do not invert ScTrigOut5
SW2_6	Invert ScTrigOut6	Do not invert ScTrigOut6
SW2_7	Interlock will not affect trig output. This mode is only used on a TIU slave where the incoming trigger is inverted (on a CCPS branch of the TI system)	Interlock will make trig output zero (dark fiber).
SW2_8	Disable ElectricInhibit2In	Enable ElectricInhibit2In

Table 8 TIU dip-switches

CAUTION! Altering the factory settings may result in damaged hardware.

2.5.3 Local trig generator

Close to the master TIU there is a BNC connector where the PLC provides a local trig generator signal which is controlled by the SSM named “T&i\TrigGenPrfSet”, this signal has to be manually looped back to the master TIU to achieve stand-alone pulsing.

2.6 Digitizer (DIGI)

A digitizer samples the pulse waveform of the output pulse, both voltage and current. It calculates several pulse parameters and it can also stream the whole pulse waveform so that it can be displayed or processed further.

Parameters			
Name	**SSM type***	**Interlock if**	**Description**
CVD read	Ai	Above high limit, Below low limit	The measured pulse output voltage.
CT read	Ai	Above high limit, Below low limit	The measured pulse output current.
Fwhm read	Ai	Above high limit, Below low limit	The measured pulse width. (Full width half maximum).
Cvd arc	MultiDi bit 0	Not ok	Arc indication by Cvd.
Ct arc	MultiDi bit 1	Not ok	Arc indication by Ct.
Cvd shape	MultiDi bit 2	N/A	Pulse shape within limits.
Cvd shape warning	MultiDi bit 3	N/A	
Ct integrate	MultiDi bit 4	Not ok	TBD
Ct shape	MultiDi bit 5	N/A	TBD
Ct shape warning	MultiDi bit 6	N/A	TBD

Table 9 Digitizer parameters

* The SSM types are described in section 0.

2.6.1 Measured pulse properties

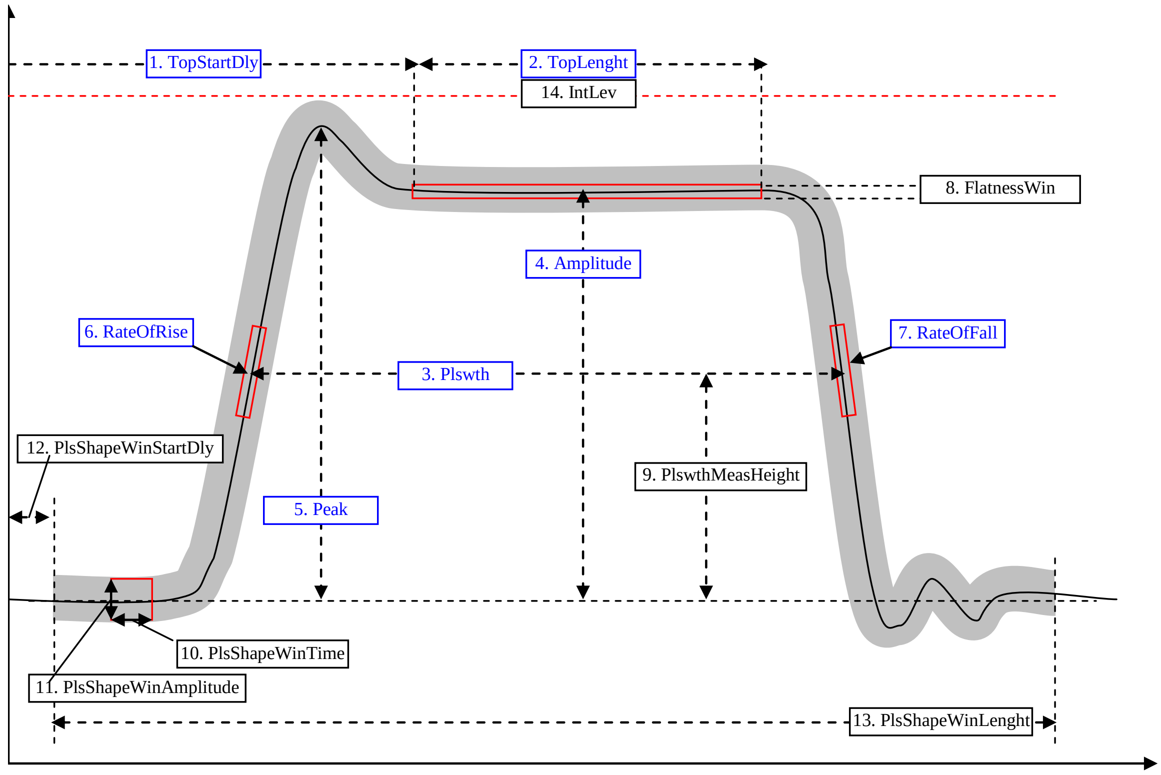


Image 8 Pulse properties measured by the digitizer

Fel! Hittar inte referenskälla. shows the different pulse properties that are available from the digitizer. These properties are measured for both the Ct pulse and the Cvd pulse.

Pulse properties	
Name	**Description**
1. TopStartDly	Time from trig to flat top start.
2. TopLength	Flat top width.
3. Plswth	The pulse width at PlswthMeasHeight.
4. Amplitude	The center amplitude value of the flat top.
5. Peak	The peak value.
6. RateOfRise	Rise rate measured at PlswthMeasHeight.
7. RateOfFall	Fall rate measured at PlswthMeasHeight.
8. FlatnessWin	Window that is used to find the flat top. The flat top is the longest row of samples that all fits within this window.
9. PlswthMeasHeight	Percentage of Amplitude where Plswth measurement should be done.
10. PlsShapeWinTime	Max allowed time variation for the pulse shape.
11. PlsShapeWinAmplitude	Max allowed amplitude variation for the pulse shape.
12. PlsShapeWinStart	Time from the trig to where the measuring window starts.
13. PlsShapeWinLength	Length of the measuring window.
14. IntLev	Absolute interlock level.

Table 10 Pulse properties.

3 Control logic

3.1 Modulator state

The modulator has a number of operational states. The number of states depends on the configuration of the modulator. Typically there are four states, but fewer or more states are possible. There are also a few other transitional states that the modulator passes through when transitioning between the states. The state machine that keeps track of the state of the modulator is called the master state machine (MSM).

There are several slave state machines (SSM). Each of these keeps track of a single I/O signal or several combined I/O signals. The main state of the master cannot be higher than the lowest slave state.

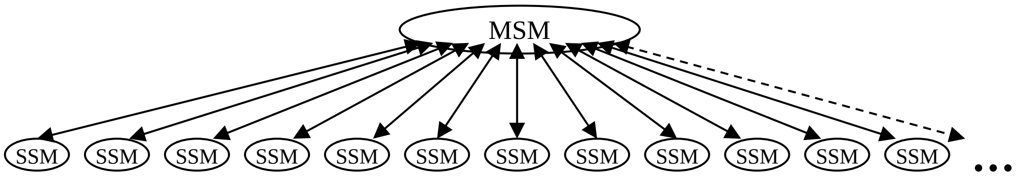


Image 9 The relation between master and slave state machines.

3.1.1 The different states

The number of states is dependent of the configuration of the modulator. Some modulators have two main states, others have four, but it is possible to have other numbers of main states.

Typically a modulator has four main states, ***Off***, ***Standby***, ***Hv*** and ***Trig***. ***Trig*** is the highest state where the modulator is fully operational and produces pulses when trigged.

3.1.2 Target state

The target state is the wanted state that is set by the operator GUI or via external communication. It will be lowered automatically when an interlock occurs.

3.1.3 State transitions

If the target state is higher than the state of the MSM, and there are no interlocks, then each SSM and the MSM will go to a transitional state. Each SSM will wait in the transitional state for its subsystem to ramp up or turn on or whatever the condition is that has been configured for it, before going to the next state. When all the SSM:s has reached the next state, the MSM will go to that state. If the target state is still higher than the MSM state the process will be repeated until the MSM has reached the target state.

If the target state is lower than the MSM state then a similar change of state will take place. All SSM:s must reach the next main state before continuing to lower the state, until the target state is reached.

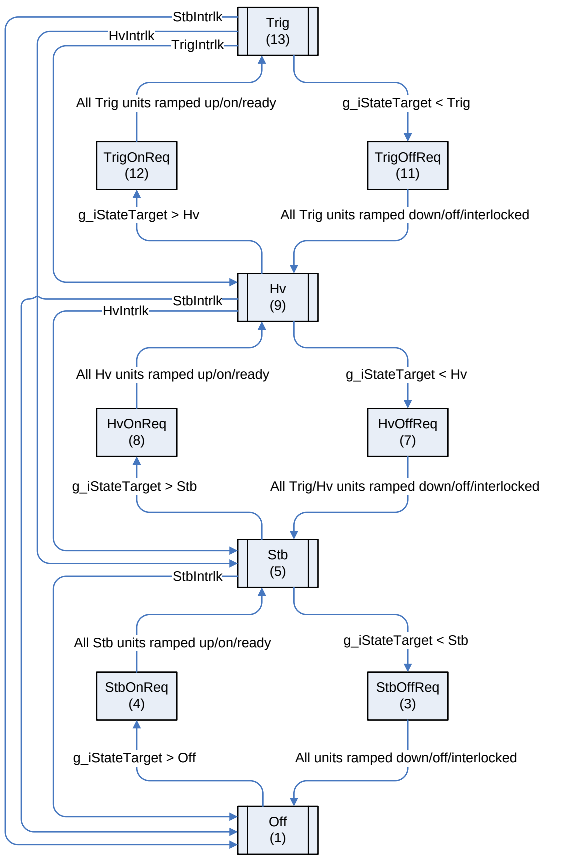


Image 10 The state transitions of a master state machine with four states.

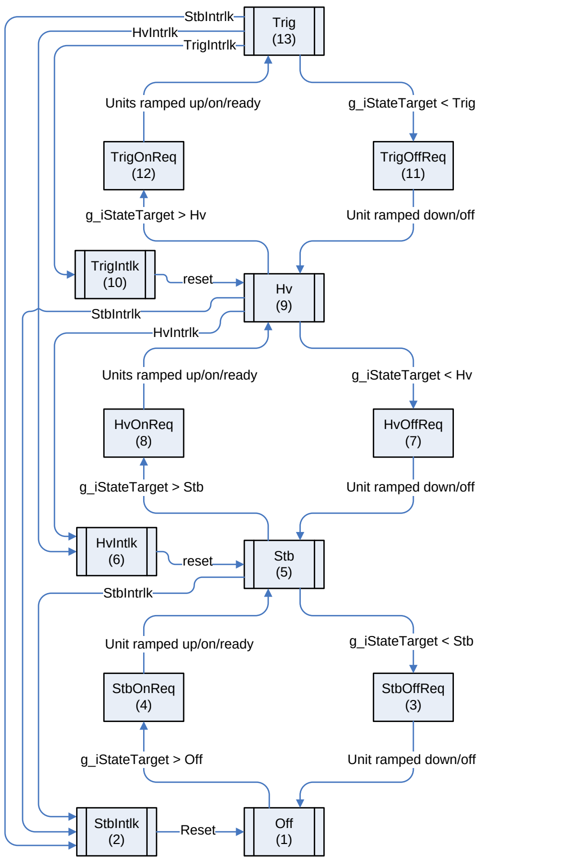


Image 11 The state transitions of a slave state machine with four states.

Fel! Hittar inte referenskälla. shows state transitions for the MSM and Image 11the state transitions of an SSM, for a modulator with four states.

3.2 Configuration

The modulator is highly configurable. Some configurations are hard coded, like the number of states. Others are configurable from the GUI.

At what state an SSM will become active and turn on its subsystem is configurable from the GUI, as can be seen in Image 12. (This is only possible to change if the access level is high enough.)

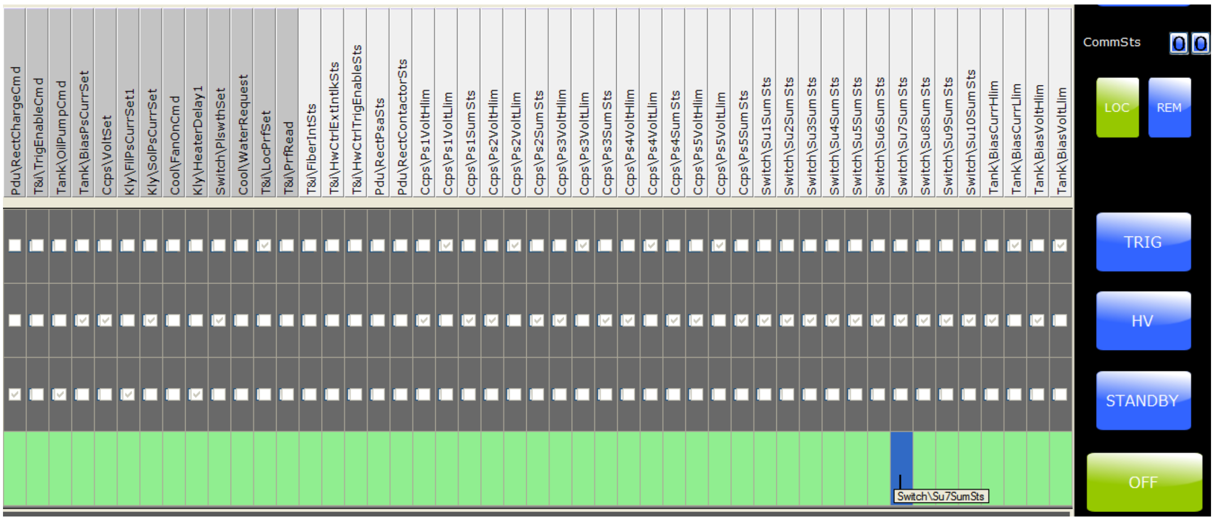


Image 12 Configuration of the states where each SSM will become active, in the matrix page of the GUI.

The interlock level of an SSM will be the level where it becomes active. When an interlock occurs the state will go to the level below the interlock level.

In the example in Image 12, the sum status of switch 7 (Switch\Su7SumSts) is configured to be active from the ***Hv*** state. This means that if it becomes not ok in ***Hv*** or any state above ***Hv***, an interlock will be generated and the state will fall to the ***StandBy*** state. If it is not ok in ***StandBy*** and an attempt is made to go to ***Hv***, the state will stay in ***StandBy***.

A list of all SSM:s and the state they are active at can be found in ref [2].

When logged in as ScandiNova it is possible to dump the whole list of SSM:s and some of their properties to a file named *MatrixConfig.txt* that is located in the *Contents* directory. The *Contents* directory is located in the same directory as *Scandinova GUI.exe*.

3.3 Interlock concept

When the signal of an active SSM is outside of its limits or a status signal is not ok, an interlock will occur and the MSM state will be lowered to the state below the state where the SSM became active. For example an SSM that is active in ***Hv*** will make the MSM fall to ***StandBy*** when interlocked.

It is not possible to go to a higher state until the condition that caused the interlock has vanished and the interlock has been reset in the GUI or via remote communication.

4 Control via GUI

The modulator can be controlled either locally or via remote desktop applications using the GUI that is available on the local touch screen.

4.1 Parts that are common to all pages

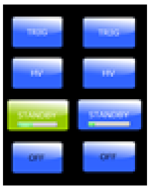
Image 13 The Scandinova GUI.

The parts of the GUI that are common to all pages are as follows (see numbering in Image 13):

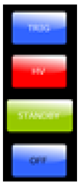
1. Current time and access level. The access level decides what settings the user is authorized to do. The access level for normal operation is “Operator”. See section 0 for more information on access levels.
2. CVD and CT values for the latest pulse.
3. Status display. Shows last warning/error or current state. Displays a timestamp of when the error or warning was issued or when the state was last changed. The trig id is a sequence number for pulses.
4. The main control panel, containing from top to bottom:
    - An interlock reset button.
      A short press will reset normal interlocks.
      A more than 3 second long press will perform a hard reset. A hard reset cycles

the power to all Scandinova nodes on the EtherCAT network (CCPS, SU, DIGI) and also resets any latched hardwired interlocks. This is done by cycling the 24V process power.

- Communication status indicators.
- Local/remote control buttons. When local is selected, the GUI controls the modulator. When remote is selected, the modulator is controlled via Modbus TCP.
- The state buttons are used to change and indicate state:



Transition between states will be indicated by button colour alternating between blue and green (it will appear to be flashing green), and if there's also a delay function a progress bar will be visible inside the button.



As a certain state is interlocked this button will be red while the actual state of the system will be downgraded to the state below.

Image 14 State buttons.

4.2 Interactive help

By pressing F1 you activate the interactive help mode. In this mode a help text will be displayed for the item under the mouse pointer. Pressing F1 again turns off the help mode.

Help texts can be found in C:\ScandiCAT GUI\Contents\Resource.xml and in this file it's located under "GUI\Help\..."

4.2.1 Access manuals

When F1 has been pressed and you are in help mode, a "browse manuals" button will appear in the lower right corner of the GUI. Press it to get access to the manuals that are delivered with the modulator.

The manual folder is located here C:\ScandiCAT GUI\Contents\Manual\ and this folder is never locked so feel free to add whatever documentation you need

4.3 The Main page

This page displays an overview of all set/read values of the modulator. Each value has a button and the symbol on the button indicates the status of the value. Typically blue or green is ok, red is interlocked and yellow is warning. To get the exact meaning of the status indication, press F1 and point on the button to get a help text. Most buttons can be pressed to get more to get more detailed information about the specific unit/SSM, for more info, see section 0.

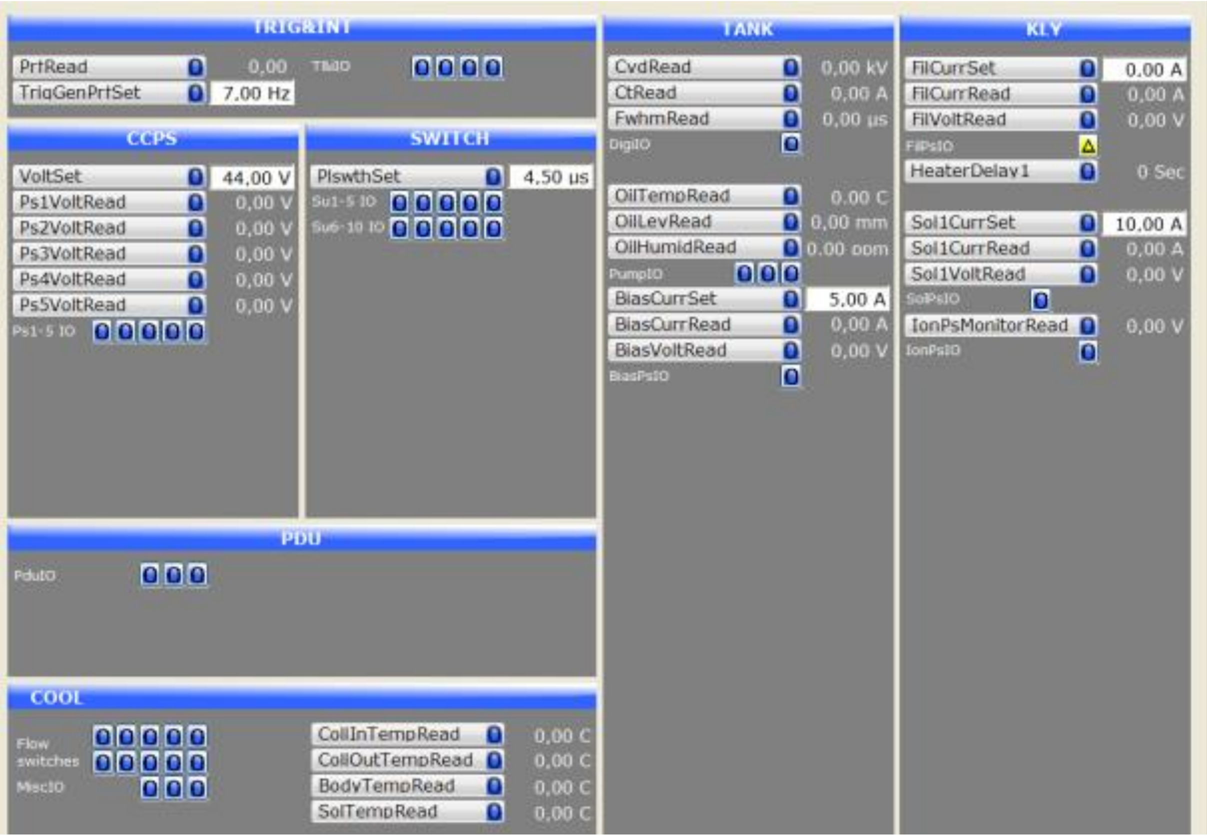


Image 15 The main page of the GUI.

Examples of main page controls	
Name	**Description**
PrfRead	The Prf read value.
TrigGenPrfSet	The frequency of the locally generated trig signal.
VoltSet	The wanted CCPS voltage.
Ps#VoltRead	The actual CCPS voltage for each CCPS.
PlswthSet	The wanted pulse width.

Table 11 Main page controls

4.4 The Digitizer page

Not yet functional.

4.5 The Event page

This page shows the event log.

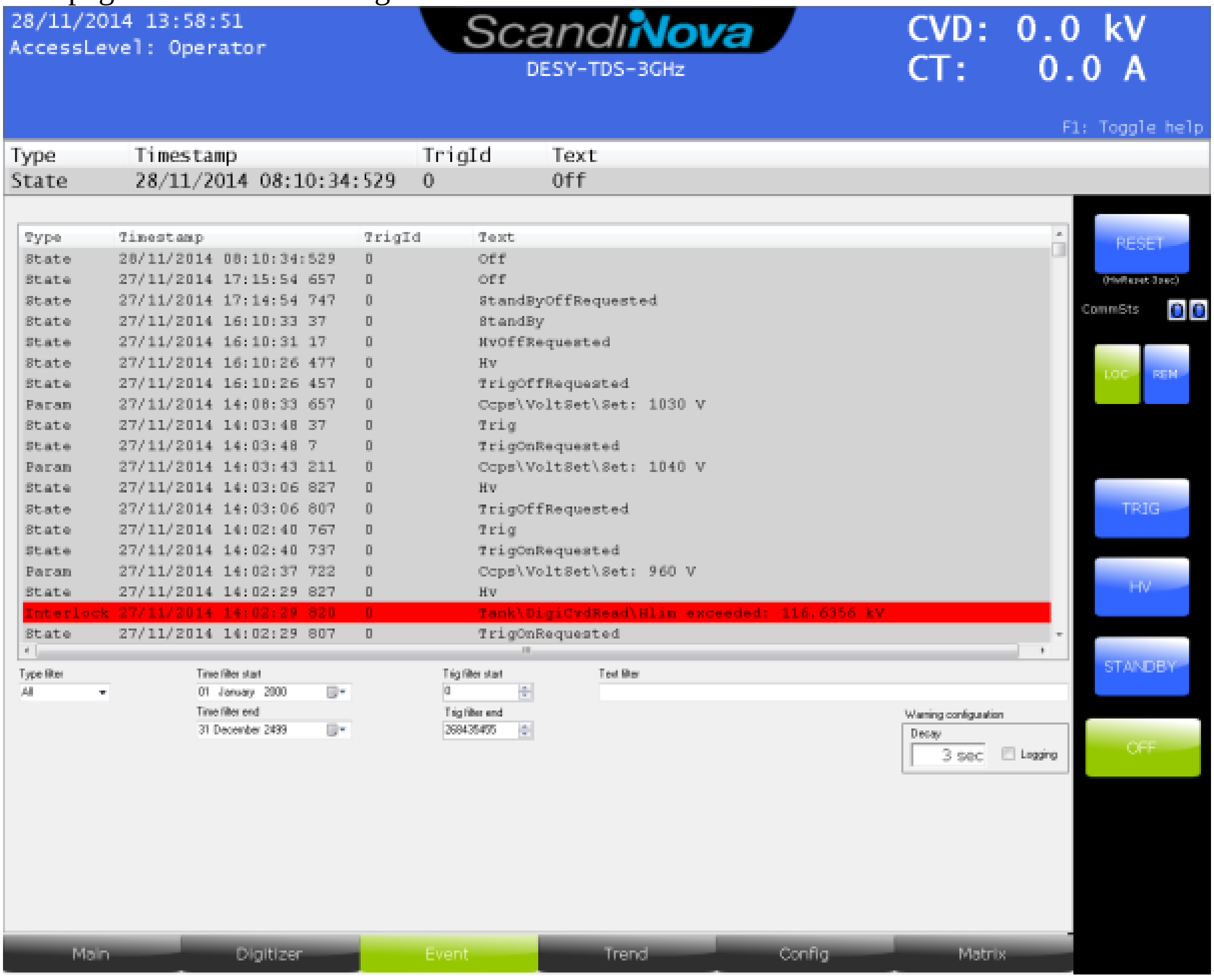


Image 16 The Event page.

The event log shows errors, warnings, interlocks and also messages about state changes, access level changes etc. The maximum number of events shown is about 2500.

The event log can be filtered. To get more detailed info about the filter controls, press F1 and point on them with the mouse to get interactive help texts.

The log is also sent to the file EventLog.txt that is located in the *Contents* directory. The *Contents* directory is located in the same directory as *Scandinova GUI.exe*.

Each event in the event log has a time stamp that is taken from the system time.

The last 50 events are stored in a FIFO in the PLC. This FIFO can be read via Modbus-TCP.

It can be configured for how long a warning message will be shown in the upper part of the GUI and also if warnings are logged or not. To get more info on these controls, use interactive help.

4.6 The Trend page

Not yet functional.

4.7 The Config page

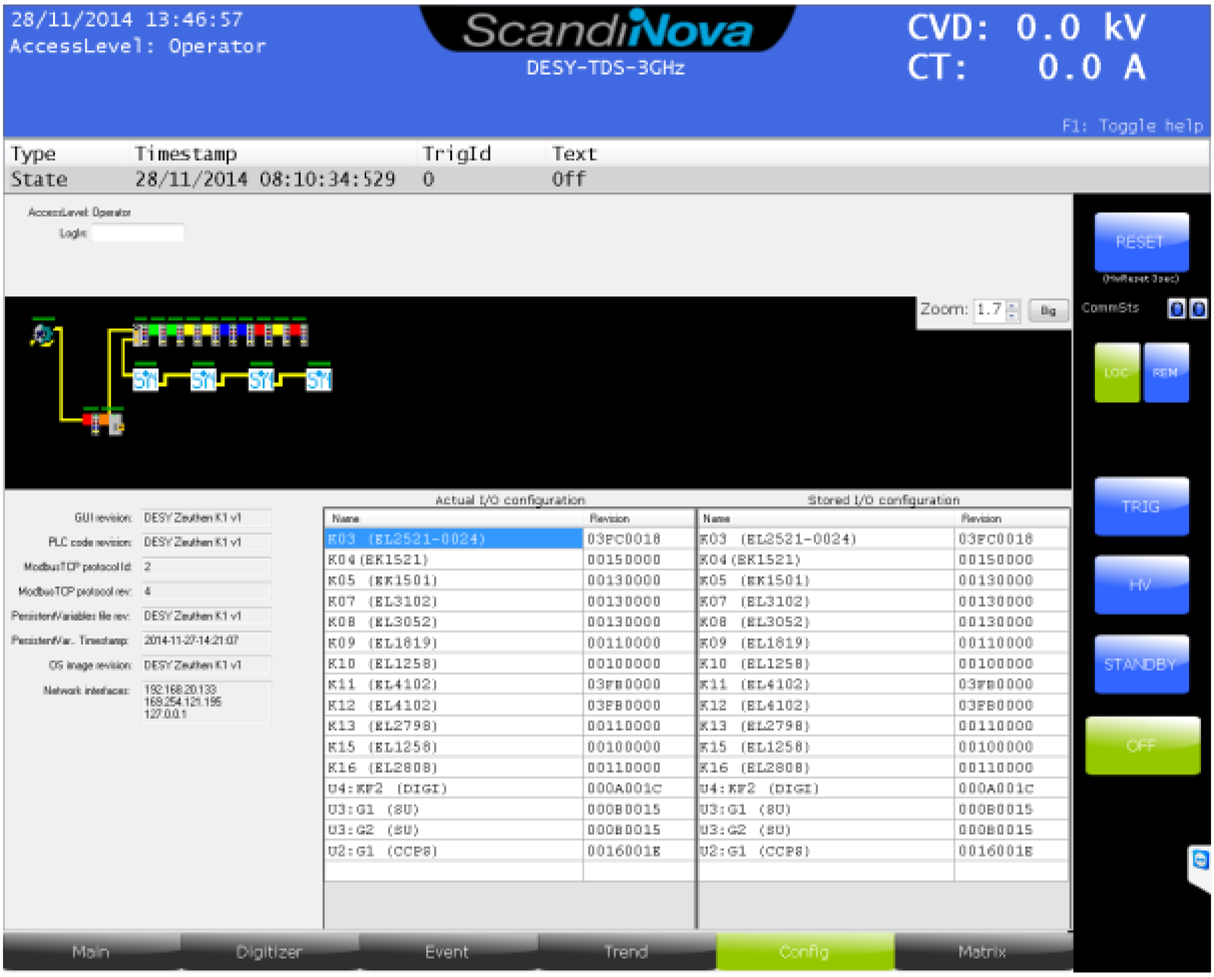


Image 17 The Config page.

Config page controls	
Control	**Description**
The login textbox	Input a password here to change access level. See section 0 for a description of the different access levels.
The EtherCAT display	This control displays all connected EtherCAT units. If communication is broken, all units that cannot be reached will be indicated with red color.
The revision textboxes	These textboxes shows the software revisions that are currently used in the modulator.
The Network interfaces textbox	This textbox displays all IP-addresses of all available network interfaces.
Actual I/O configuration	This list shows the names of all currently connected EtherCAT units and their revisions. If this list does not match the stored I/O configuration list, the modulator cannot be started. This is to make sure that all slaves have the correct software before using the modulator.
Stored I/O configuration	This list shows the stored list of all EtherCAT units and their revisions. It can be stored when logged in as Scandinova.

Table 12 Config page controls

For more detailed info press F1 and point to the control you want to know more about to get the help text for it.

4.7.1 Access levels

Level	Modbus-Value	Description
Remote	0	Used when in remote operation mode.
Operator	1	Used during normal operation.
Admin	2	Used for configuration. Password: 1234 Will be lowered to operator after a timeout of 5 min.
Scandinova	3	Used for factory configuration. Will be lowered to operator after 5 min of inactivity.

Table 13 Access levels

4.8 The Matrix page

In the matrix page there is a view of the matrix of slave state machines (SSM:s), one column per SSM. Each SSM corresponds to a subsystem unit. This is a different view of the same units that are in the main page. In the matrix page it is possible to see and configure at what state each unit is active.

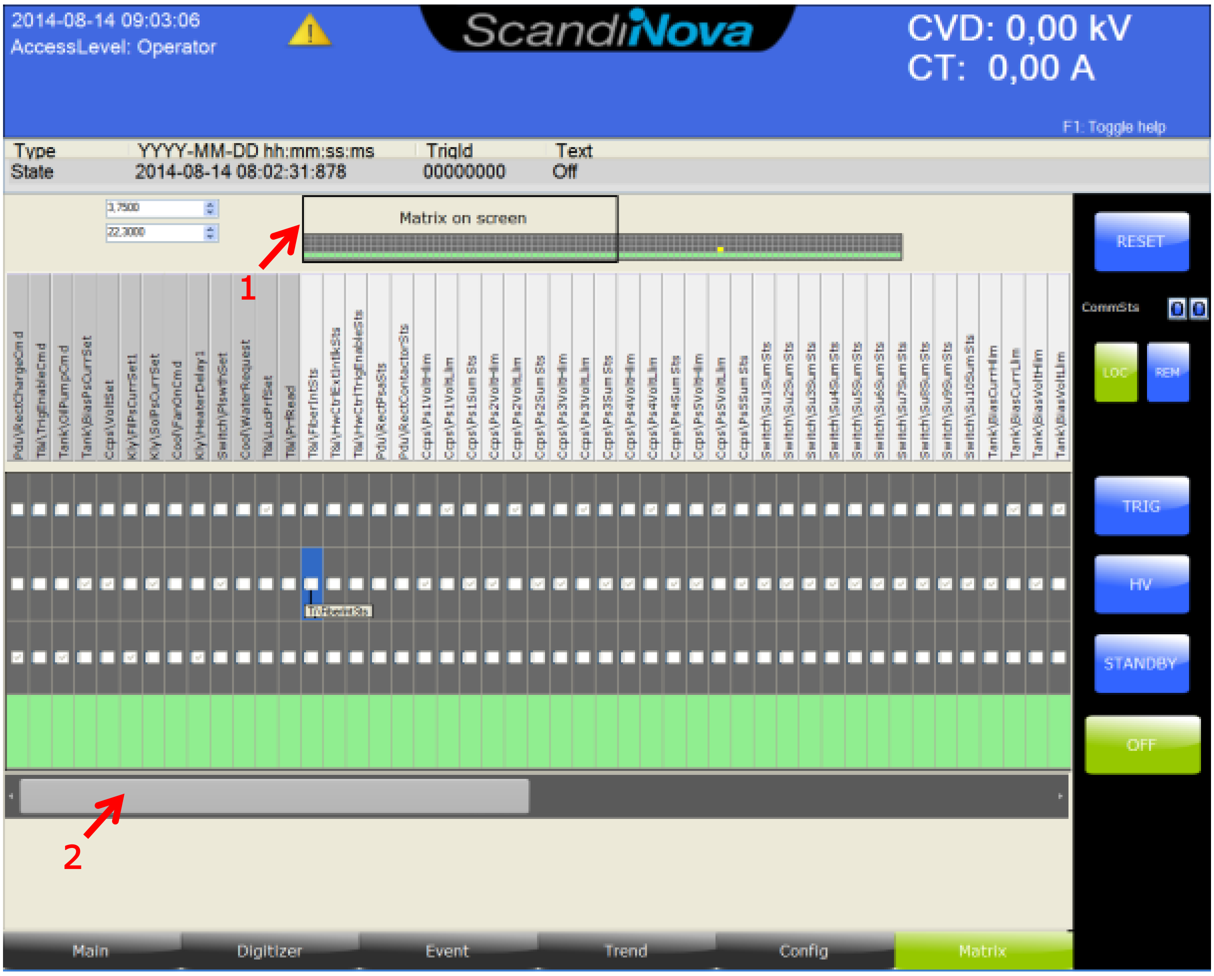


Image 18 The matrix page.

Image 18 shows the matrix page. The total number of values in the matrix does not fit on screen so at number 1 in the figure there is an indication of what part of the matrix is currently shown. The scrollbar at 2 is used to scroll the matrix and select what part is shown.

Each column in the matrix contains the name of the unit in the first row. The checkboxes in the following rows determines at what state the SSM that controls the unit will be active. If an interlock occurs due to a unit, the interlock state will be the state below the activation state for that SSM.

The state for each SSM is displayed by changing the color of the buttons of its matrix column. In the figure above all SSM:s are in the ***Off*** state (displayed with green color). In The figure below all but two are in the ***StandBy*** state. When these two reaches the ***StandBy*** state the whole modulator is in ***StandBy*** and the ***StandBy*** button will stop flashing.

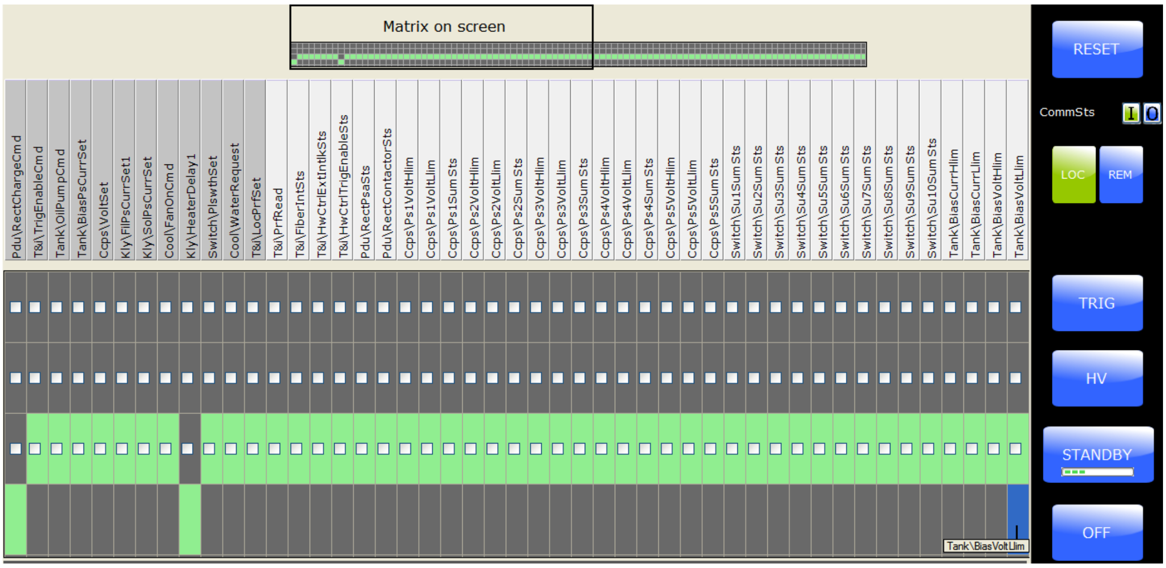


Image 19 All but two SSM:s has reached standby.

Image 20 show a SSM with a warning condition, for example a warning limit that is exceeded, this will be indicated by yellow.

Image 21 show a SSM with a interlock condition, this will be Indicated by red.

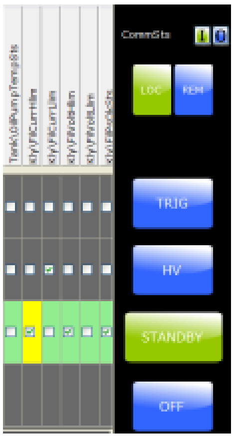


Image 20 SSM warning in Matrix.

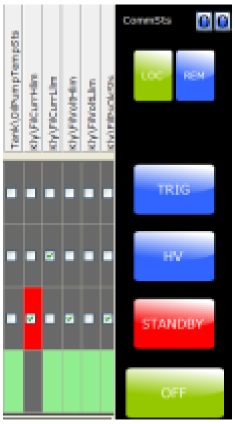


Image 21 SSM interlock in Matrix.

If a condition exists that would give you an interlock if you try to go to a higher state, you will not get further than the state below and there will be a warning at the state with the interlock condition. The warning will be indicated with yellow colour.

Press F1 and use the help system to get more detailed information about the different units in your modulator.

4.9 *The different SSM types*

4.9.1 Example of an Ai SSM

An Ai SSM is an entry in the matrix that corresponds to an analog input.

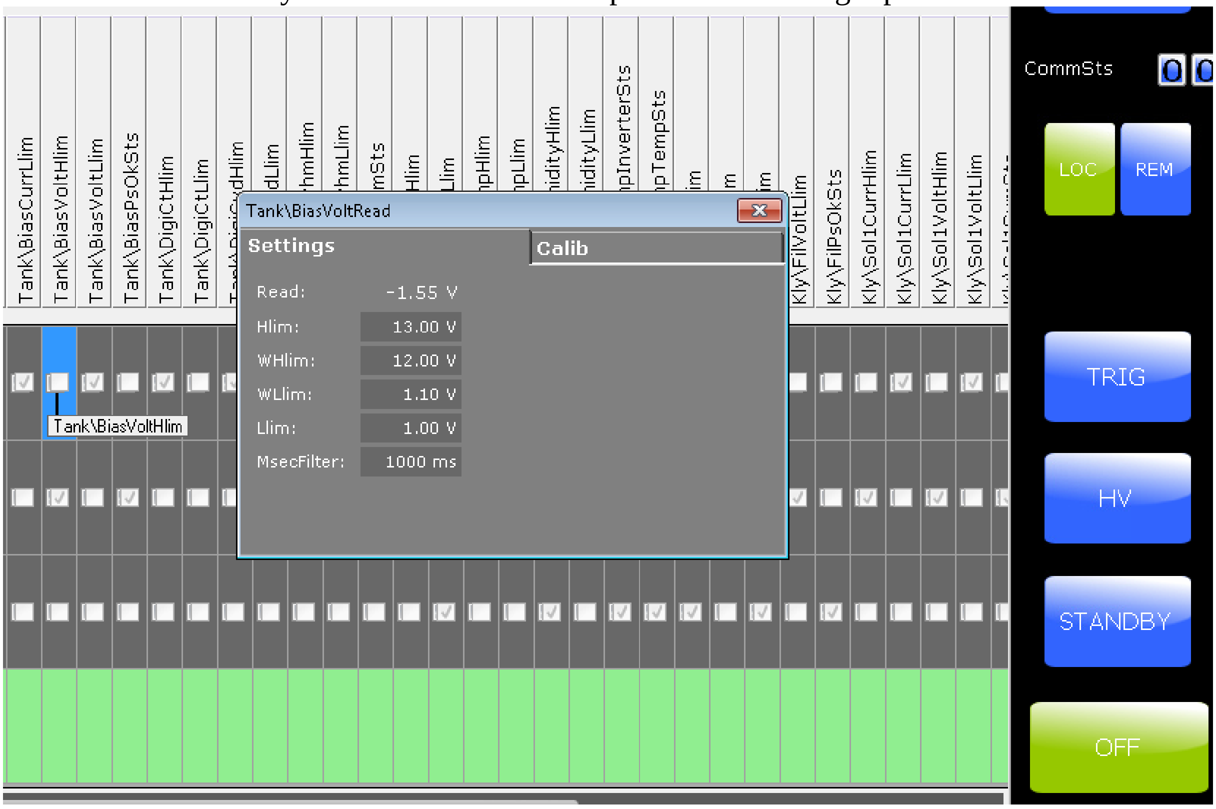


Image 22 Example of an analog input.

Image 22 shows as an example the analog input Tank/BiasVoltRead. An analog input has two entries in the matrix, a high limit entry and a low limit entry. Tank/BiasVoltRead has the entries Tank/BiasVoltHlim and Tank/BiasVoltLlim. This means that it can have two different interlocks, one when the value is higher than its high limit and one when it is lower than its low limit. These interlocks can be configured at different states. In the figure it can be seen that Tank/BiasVoltHlim is a ***HV*** interlock and Tank/BiasVoltLlim is a ***Trig*** interlock.

Ai SSM parameters	
Name	**Description**
Read	The current analog read value.
Hlim	The high limit. A read value that is over this limit will cause an interlock.
WHlim	The high warning limit. A read value that is over this limit will cause a warning.
WLlim	The low warning limit. A read value that is under this limit will cause a warning.
Llim	The Low limit. A read value that is under this limit will cause an interlock.
MsecFilter	The time of the analog averaging filter for this input.

Table 14 Ai SSM parameters

4.9.1.1 Calibration

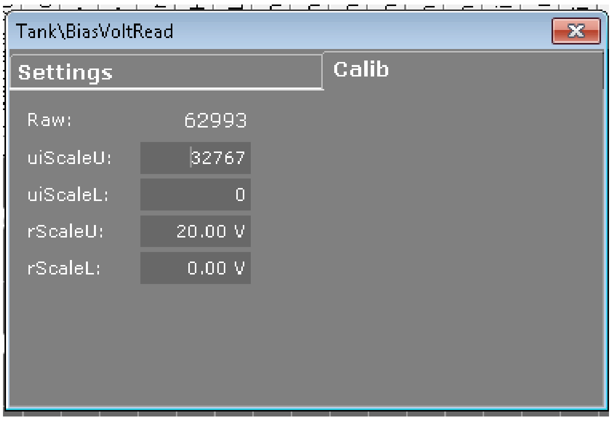


Image 23 Calibration for an analog input.

The raw analog input value is scaled linearly to a voltage range before being displayed in the GUI, using a two point calibration.

Calibration parameters	
Name	**Description**
Raw	The current raw A/D input value.
uiScaleU	The raw value for the higher calibration point.
uiScaleL	The raw value for the lower calibration point.
rScaleU	The voltage value for the higher calibration point.
rScaleL	The voltage value for the lower calibration point.

Table 15 Calibration parameters

Note that the Calib page of the dialog box is only available if the user access level is high enough.

4.9.2 Example of an Ao SSM

An Ao SSM is an entry in the matrix that corresponds to an analog output.

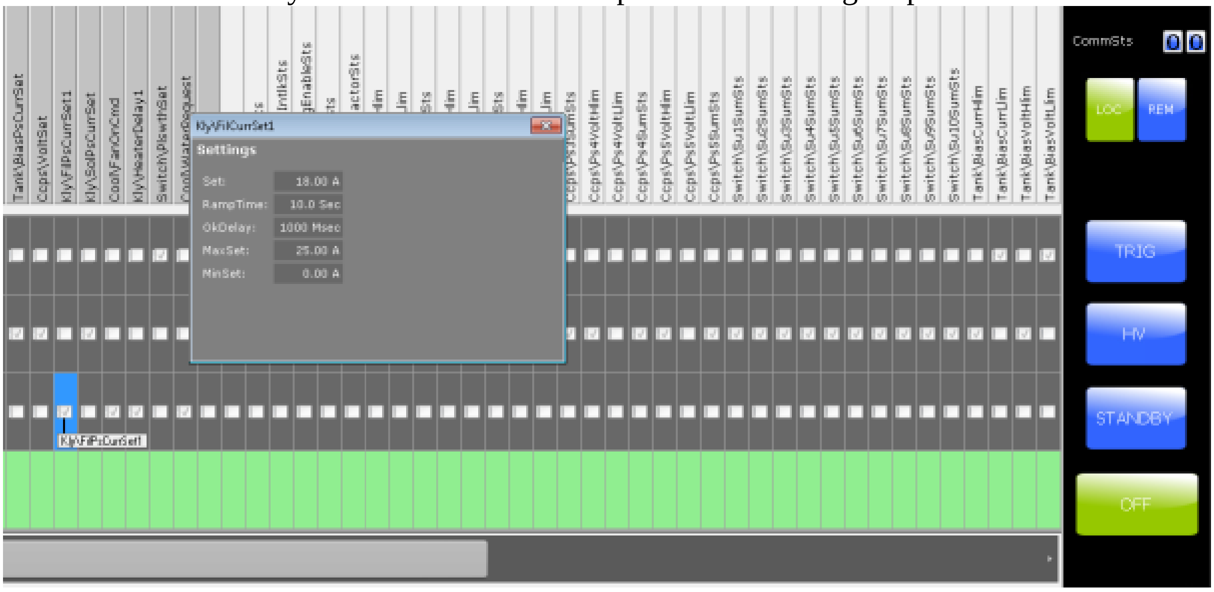


Image 24 Example of an analog output.

Settings for the analog output Kly\FilCurrSet1 can be seen in Image 24.
In the matrix it can be seen that it will be turned on at the ***StandBy*** state.

Ao SSM parameters	
Name	**Description**
Set	The set value for the analog output.
RampTime	How long time it will take for the output to ramp up to the set value after it is turned on.
OkDelay	When the ramp has reached the set value, the state will not change until after this delay.
MaxSet	The set value cannot be set higher than this value. To change this, the user access level must be high enough.
MinSet	The set value cannot be set lower than this value. To change this, the user access level must be high enough.

Table 16 Ao SSM parameters

Note that there are also hardcoded max and min limits in the PLC that cannot be exceeded. The MaxSet and MinSet values can only be used to make the max-min range narrower than the hardcoded limits.

4.9.2.1 Calibration

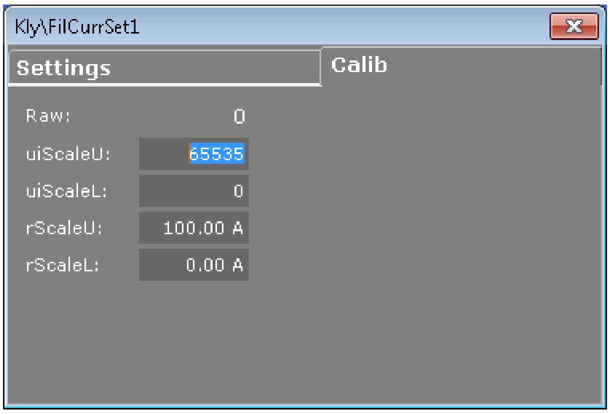


Image 25 Calibration dialog for an Ao SSM.

The current in A (in this example) from the GUI is scaled to a raw D/A output value using a two point calibration.

Calibration parameters	
Name	**Description**
Raw	The current raw D/A output value.
uiScaleU	The raw value for the higher calibration point.
uiScaleL	The raw value for the lower calibration point.
rScaleU	The current in A for the higher calibration point.
rScaleL	The current in A for the lower calibration point.

Table 17 Calibration parameters

Note that the Calib page of the dialog box is only available if the user access level is high enough.

4.9.3 Example of an AoMulti SSM

An AoMulti SSM is an entry in the matrix that corresponds to several analog outputs. It has one single set value that is common to all outputs. Each output can be calibrated individually.

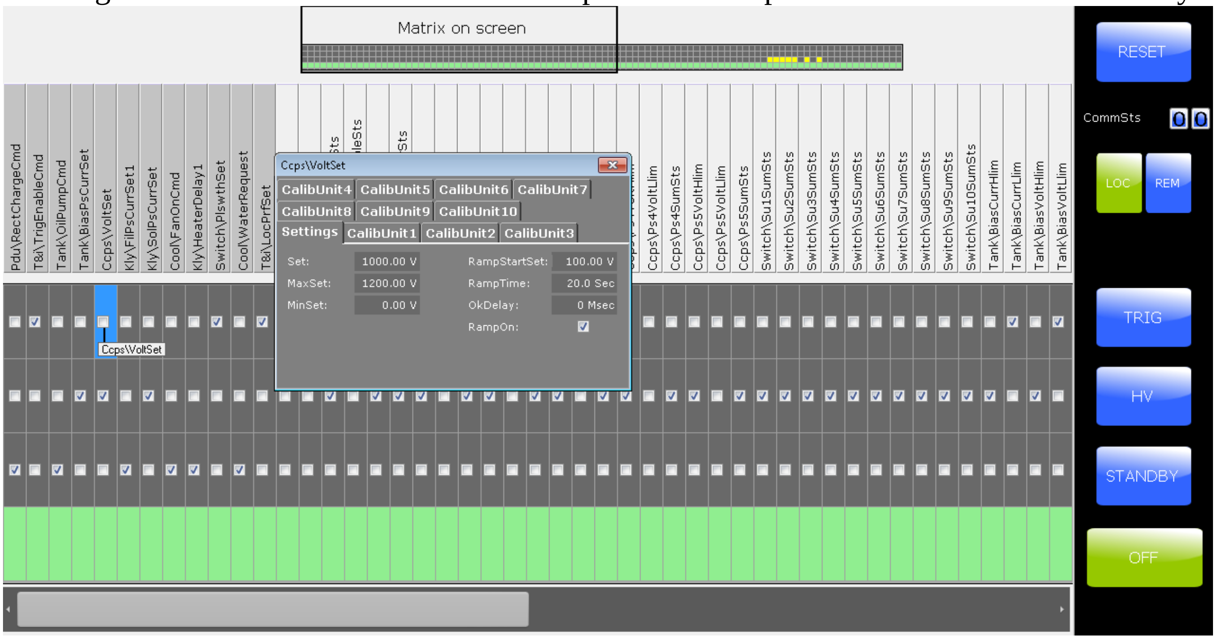


Image 26 Example of an AoMulti SSM.

AoMulti SSM parameters	
Name	**Description**
Set	The set value for the analog output.
MaxSet	The set value cannot be set higher than this value. To change this, the user access level must be high enough.
MinSet	The set value cannot be set lower than this value. To change this, the user access level must be high enough.
RampOn	Enables the RampStartSet, RampTime and OkDelay settings.
RampStartSet	The start value of the output before the ramp starts.
RampTime	How long time it will take for the output to ramp up to the set value after it is turned on.
OkDelay	When the ramp has reached the set value, the state will not change until after this delay.

Table 18 AoMulti SSM parameters

The Set, MaxSet and MinSet values are the same as for the Ao SSM. As for the Ao SSM there are also hardcoded max and min limits in the PLC.

4.9.3.1 Calibration

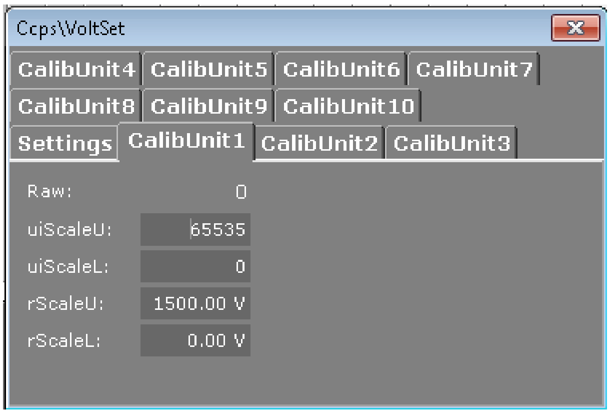


Image 27 Ao multi calibration dialog.

The voltage (in this example) from the GUI is scaled to a raw D/A output value using a two point calibration.

Calibration parameters	
Name	**Description**
Raw	The current raw D/A output value.
uiScaleU	The raw value for the higher calibration point.
uiScaleL	The raw value for the lower calibration point.
rScaleU	The voltage for the higher calibration point.
rScaleL	The voltage for the lower calibration point.

Table 19 Calibration parameters

Note that the Calib page of the dialog box is only available if the user access level is high enough.

4.9.4 Example of a Di SSM

A Di SSM is an entry in the matrix that corresponds to a digital input.

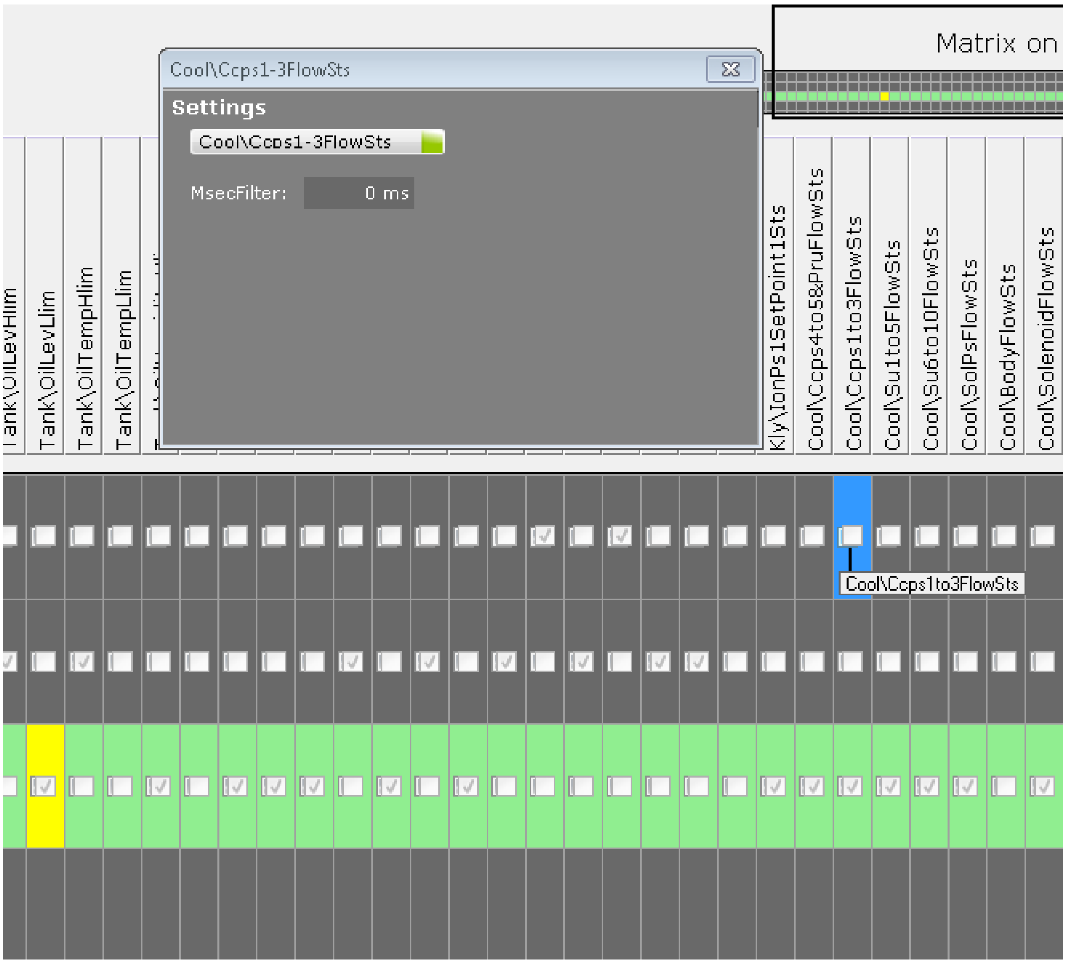


Image 28 Example of a Di SSM.

The input is filtered. It must have a constant "not ok" value for a certain time before the Di SSM can trip an interlock. The filter time is given by MsecFilter.

Di SSM parameters	
Name	**Description**
MsecFilter	Input filter time. The input must have a constant "not ok" value for this long before the Di SSM can trip an interlock.

Table 20 Di SSM parameters

4.9.5 Example of a DiMulti SSM

A DiMulti SSM is an entry in the matrix that corresponds to several combined digital inputs.

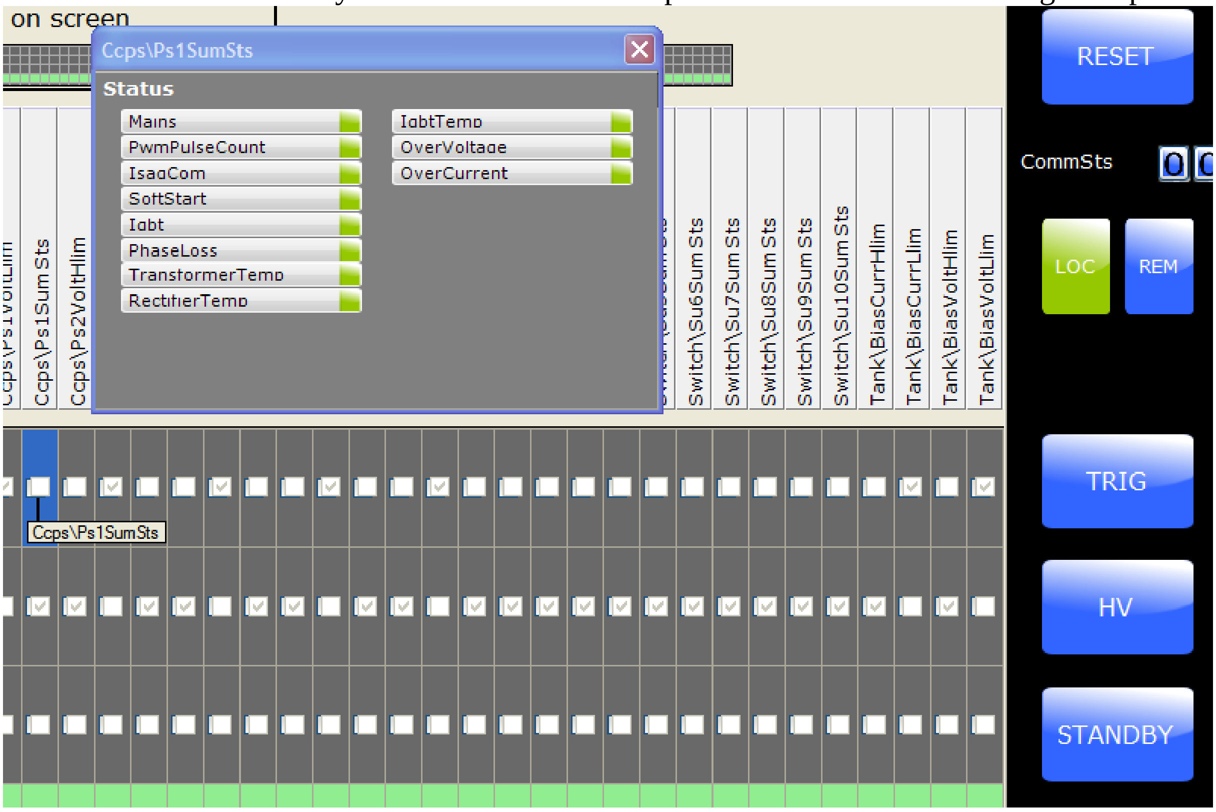


Image 29 Example of a DiMulti SSM

Image 29 shows the digital input Ccps/Ps1SumSts. As can be seen, this input is configured to be active in the ***HV*** state and above. Its value is the combined status of several status signals from the Ccps. These status signals can be seen in the dialog box. All these must be ok for the sum status to be ok. If one is not ok, then the sum status will be not ok and a ***HV*** interlock will be generated.

4.9.6 Example of a Do SSM

A Do SSM is an entry in the matrix that corresponds to a single digital output. The output will be turned on at the configured state. It has no parameters.

4.9.7 The IntComm SSM

The IntComm SSM is special because it has two activation states so it can trip interlocks at two different states.

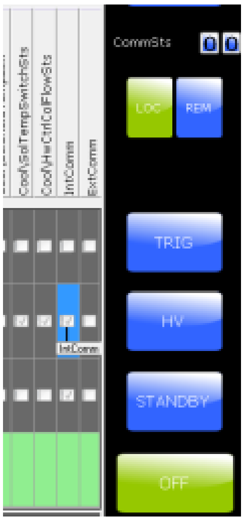


Image 30 The IntComm SSM.

The higher state interlocks if the communication is broken to any ScandiNova unis (CCPS, SU or DIGI). The lower state is interlocked after 3 seconds of no communication with a Beckhoff unit.

4.9.8 The ExtComm SSM

ExtComm monitors the Modbus TCP and (optionally) external EtherCAT communication watchdogs. If any of these are not served it will interlock. It is active in remote communication mode only.

5 Remote control

5.1 Control via Modbus-TCP

The modulator can be controlled via Modbus-TCP. A number of Modbus variables are available, which ones and their purpose can be found in the Modbus TCP specification for the modulator, ref [1].

Only input registers and holding registers are used (no coils). This means that each variable size is a multiple of 16 bits. The size of each variable can be found in ref [1]. The variables must be accessed one at a time; consecutive variables cannot be accessed together in a single operation.

ModbusTCP specification			
Protocol	Connector	Speed	Location
ModbusTCP / 100Mbit	RJ45 (option: SC fiber)	Max refresh rate 100Hz	MCU:K01:X000

Table 21 ModbusTCP specification

5.2 Control via EtherCAT

Option: When in remote control mode, some set and many read values can be controlled synchronously via EtherCAT.

ModbusTCP specification			
Protocol	Connector	Speed	Location
EtherCAT / 100Mbit	RJ45 (option: SC fiber)	Up to 1kHz	MCU:K01:X000

Table 22 EtherCAT specification

5.3 Getting started with MODBUS-TCP

This section is intended to get you started with controlling the modulator using MODBUS-TCP. It is supposed to get you up and running to the point where you can read and write registers in the PLC remotely.

The following examples assume that you use a mock ScandiCAT system that is provided as a VirtualBox image. When the communication works with that image you can set it up in a similar way with the real modulator hardware.

5.3.1 Run the VirtualBox image

A test version of the control system is available as a VirtualBox image. It contains Windows XP 32-bit, the TwinCAT PLC and the Scandinova GUI for manual control. The PLC is running a mock version of the ScandiCAT control system that is only used to test communication.

- Download and install VirtualBox: https://www.virtualbox.org/wiki/Downloads

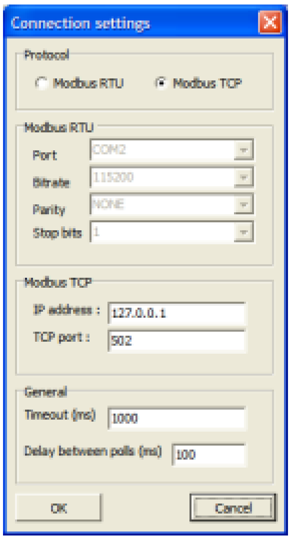
- Run the provided Windows XP image.
- Make sure that the PLC is started. When it is started the TwinCat symbol in the system tray is green.


5.3.2 Test Modbus TCP communication locally with the RMMS client

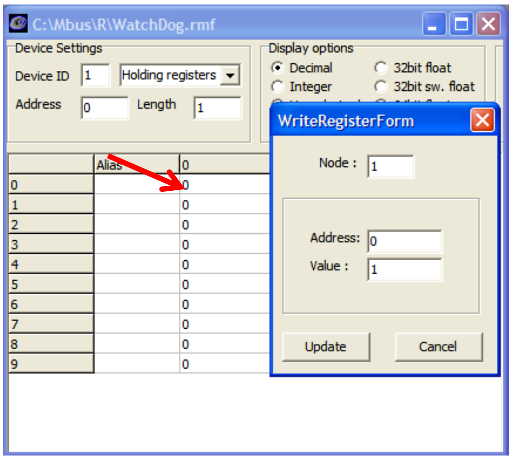
The VirtualBox image contains a generic Modbus client called RMMS. This client can be used to test Modbus communication with the ScandiCAT server.
To get familiar with it you can use it to control the mock ScandiCAT system on the image.

- Start the ScandiCAT GUI from the desktop 

.
- Start RMMS from the desktop 

.
- Go to File -> Load windows set and browse to c:\Mbus\R\GettingStarted.rms. This will open a few windows that each contains one modulator register.
- Make sure that the WatchDog and StateTargetRem windows contains holding registers and that WatchDogPrev and StateRead contains Input registers.
- Open Connection->Settings and verify that Modbus TCP is chosen, that the IP-address is 127.0.0.1 and that the port is 502. Then close the settings dialog.


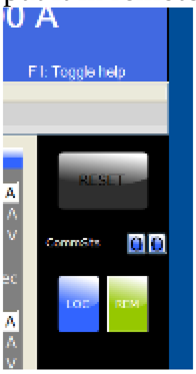
- Connect to the PLC with Connection->Connect.
- Try to write a value to the communications watchdog and verify that it can be read back. To do this, double click on the first register value in the WatchDog window (at the red arrow in the image below). Type a value in the dialog box that opens and click update. The value you typed should appear in the WatchDogPrev window. If it does, then the communication works.



5.3.3 Test remote control via Modbus TCP locally with the RMMS client

To be able to write other values than the watchdog value to the modulator it needs to be in remote control mode.

- Use the same setup as in the previous section (5.3.2). Go to the ScandiCAT GUI and put it in remote control mode by clicking the REM button:



- Set the target state value to 0, 5, 9 or 13 in RMMS in the StateTargetRem window. 5Verify that the read state value follows it in the StateRead window, but with some delay. Check also in the GUI that the state indication changes.

5.3.4 Test the communication remotely with the RMMS client

The virtual network card of the image is configured to be bridged, that means that the virtual machine is connected directly to the local network and will get its own ip-address via DHCP.

- Run the mock modulator in the VirtualBox image as in the previous sections.
- Run the RMMS client it in your host operating system on the same computer or on another computer in the same network.

- Copy the c:\Mbus\R dir in the VirtualBox image to the computer you run the RMMS client on.
- Go to File -> Load windows set and browse to GettingStarted.rms to get the same window setup as in the previous sections.
- To get the IP address of the mock modulator, go to the config page of the ScandiCAT GUI. The IP-addresses of all local network cards are shown there. The IP-address you want is the one that is within the range of your local network.
  You can also get the IP address by running *ipconfig* from the command prompt.
- In the RMMS client, go to Open Connection->Settings and input the IP-address.
- Go to Connection->Connect in the RMMS client.
- It should now be possible to write and read the watchdog value as in the previous sections.
- If you put the modulator in remote mode in the GUI it should also be possible to change the modulator state remotely, as in the previous sections.

5.3.5 Test communication with your own client

You can test the communication between your own client and the mock modulator the same way as with the RMMS client. Get the IP-address of the mock modulator from the ScandiCAT GUI or *ipconfig* on the command line and configure your client with this value.

To test the communication watchdog, write to Modbus Holding register 0 and read back the value at Modbus input register 2.

To test changing state; put the modulator in remote control mode and, write a new target state to Modbus holding register 1. Get the current state by reading the Modbus input register 3.

5.3.6 Getting started on the real system

Communication with the real system is done the same way as with the mock system, as **described in the previous sections. Get the IP address of the modulator from the Config page** of the ScandiCAT GUI. Configure the client with it. You can use RMMS for the first communication tests or you can use your own client.

Make sure that your own client is able to communicate with the ScandiCAT system in the Virtual Image before attempting to communicate with a real modulator.

When you are up and running with the communication you can configure your client to use all Modbus registers in the modulator. The list of all registers can be found in ref [1].

6 Maintenance

6.1.1 Remote access

A remote desktop application, like TeamViewer, can be run on the modulator. This makes it possible for Scandinova to perform maintenance remotely. All software in the system can be updated this way, except the CPLD software in the Trig and Interlock Units.

To enable remote access you must temporarily connect a new Ethernet cable to the CPU with internet access or set up a VPN connection and open suitable ports in your firewall only at the time of remote assistance.

6.1.2 Replacing a unit

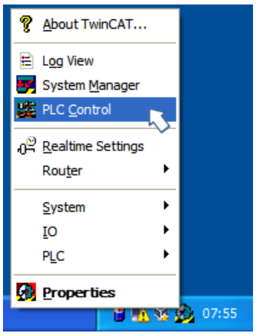
No settings are saved in the Scandinova units, (CCPS, SU, DIGI) so it is possible for the customer to replace such a unit with a spare part and program it with the latest version of its software. If needed, Scandinova support can help with the programming of the unit via remote access.

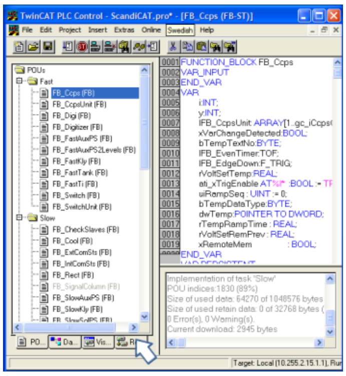
6.1.3 Back-up/restore modulator settings

Each time a parameter is changed (from GUI or via remote communication) the value is stored in flash memory.

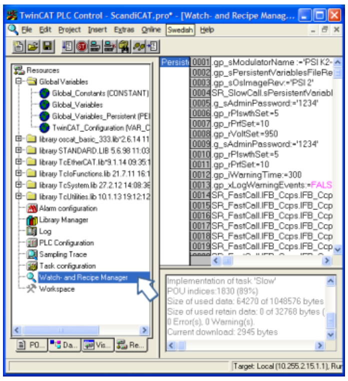
6.1.3.1 Saving a "snap-shot" of all parameters

1. Minimize the GUI (Win+D) and:
2. Click on the TwinCAT icon: 

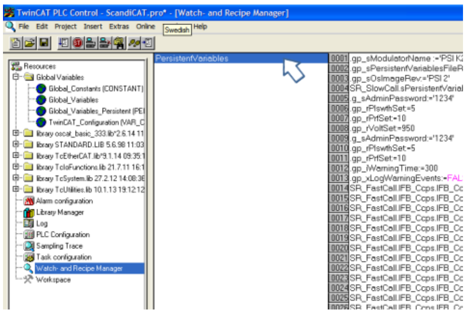
3. Select PLC Control: 



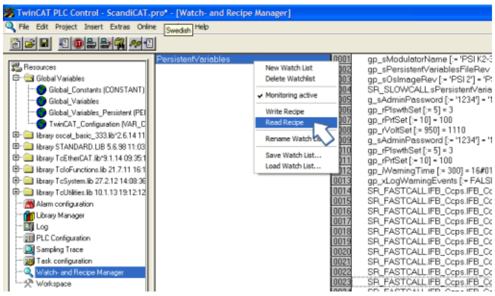
4. Click on the Resources TAB:



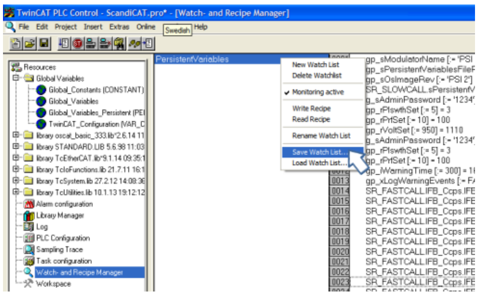
5. Double-click on the Watch- And Recipe Manager:
6. Go Online by pressing F11



7. Right-click on the WatchList PersistentVariables:



8. Selet Read Recipe:
9. Answer Yes in the following pop-up

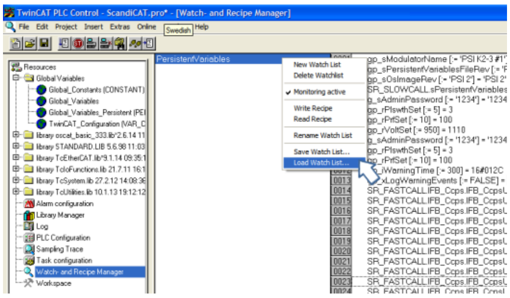
10. Finally, save the file by selecting Save Watch List: 

    (default location and name is C:\ScandiCAT PLC\PersistentVariables.wtc but you can modify the name and location of where to place the file, why not on a USB stick so you can store the settings on a external harddrive)

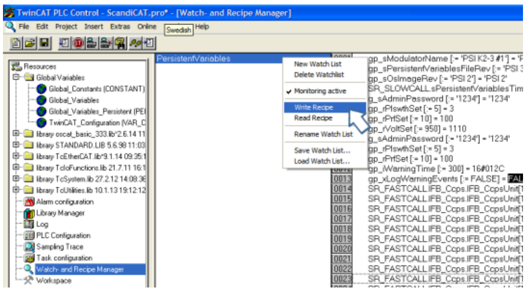
6.1.3.2 Recalling parameters

Start by performing step 1 to 7 of the previous chapter and then…

8. Delete or rename the existing watch list

9. Select Load Watch List: 

    (and point out the file you want to restore, probably located on a USB stick that you inserted)

10. Select Write Recipe: 

11. Answer Yes in the following pop-up

7 References

1 ScandiCAT Modbus TCP specification.

2 ScandiCAT State Machine Configuration.