NCT® 201

CNC Machine controller

Setting of EtherCAT slaves

Version 1.0.

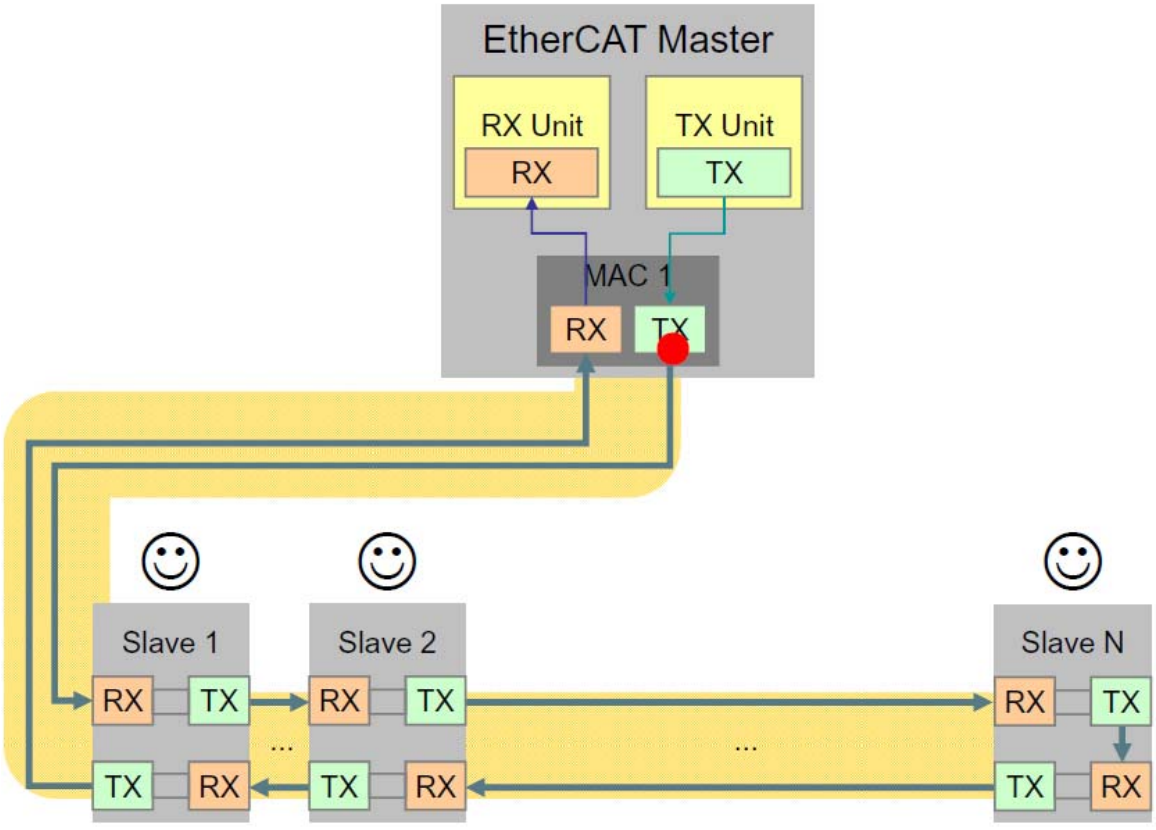
Manufacturer and developer: **NCT Ipari Elektronikai Kft**.
H-1148 Budapest, Fogarasi út 7.
 Postal address: H-1631 Budapest, P.O. Box: 26
 Telephone: (+36 1) 467 63 00
 Telefax:(+36 1) 467 63 09
E-mail: nct@nct.hu
Web page: www.nct.hu

We compiled this description with the greatest care and checked thoroughly; however, *we shall not bear any responsibility for any errors or incorrect data and the resulting damages*.

1 Introduction

1.1 ETHERCAT -REAL TIME ETHERNET CONTROL AUTOMATION TECHNOLOGY

EtherCAT is a real-time communication surface which connects the control unit with peripheries through an Ethernet network, such as with the drives, in- and output units, encoders, etc. In case of EtherCAT only the control unit (EHU - ***E****therCAT* ***H****OST* ***U****nit*) forwards EtherNET frames and the units *(Slave)* will place data in or take data out of only this frame. The control unit is the *HOST* which starts the train (data flow) and the slave units are the stations where the data exchange takes place. At the last slave element, the train turns back and runs through all stations till it arrives back to the HOST unit. The HOST schedules how frequently it will initiate communication.



Picture 1: EtherCAT communication (EHU - Slave connection)

1.2 Identification of slave elements

Every slave elements has a unique identifier set by the manufacturer. We distinguish between the elements based on these identifiers:

- Vendor ID
- Product code
- Revision number
- Serial number

These data can be read out from all slave elements and the control unit displays based on these the features of the slave, contained by a description file (XML), provided by the manufacturer to the product.
Data are stored by the slave unit in a non-volatile memory part (EEPROM) from which data typical for the unit can be read out.

Structure of the EEPROM: (16-bit addressing with words)

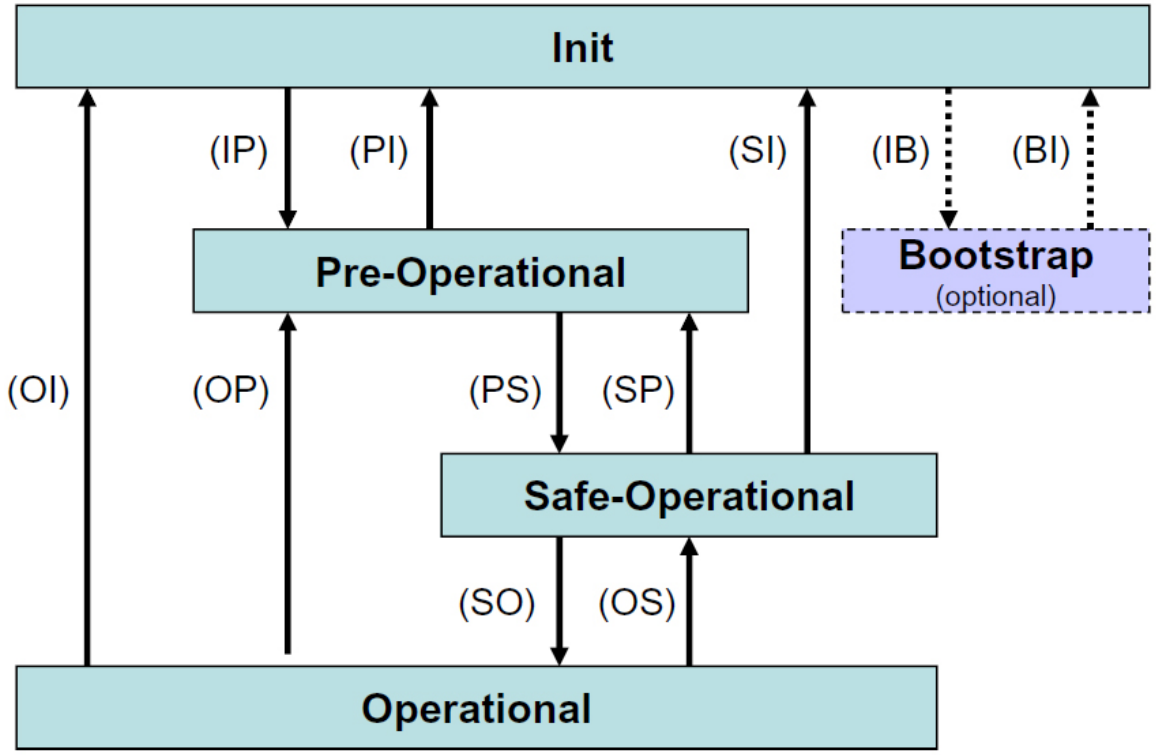
0	EtherCAT configuration area			
8	Vendor ID	Product code	Revision number	Serial number
16	Hardware delays		Bootstrap mailbox settings	
24	Mailbox SM settings			
	Reserved			
64	Other information			
	FMMU settings			
	SM settings			
	Tx / Rx PDO data			

1.3 States of slave elements

The EtherCAT slave elements have 5 states:

- **Init** - Initial state after turning on. Features and data of the device can be queried.
- Pre-Operational - Query and setting of communication surface and background data. (e.g.: Beginning of parameter download at drives)
- **Safe-Op**erational - Every communication channel is operating but the EHU is reading only data, it does not set the outputs. (e.g.: The signals of input units can already be seen but the outputs cannot be switched yet)
- **Op**erational - A state ready for operation. EHU sets the outputs and reads the inputs.
- **Boot**strap - in this state the software upload takes place. It is not supported by all slave elements.

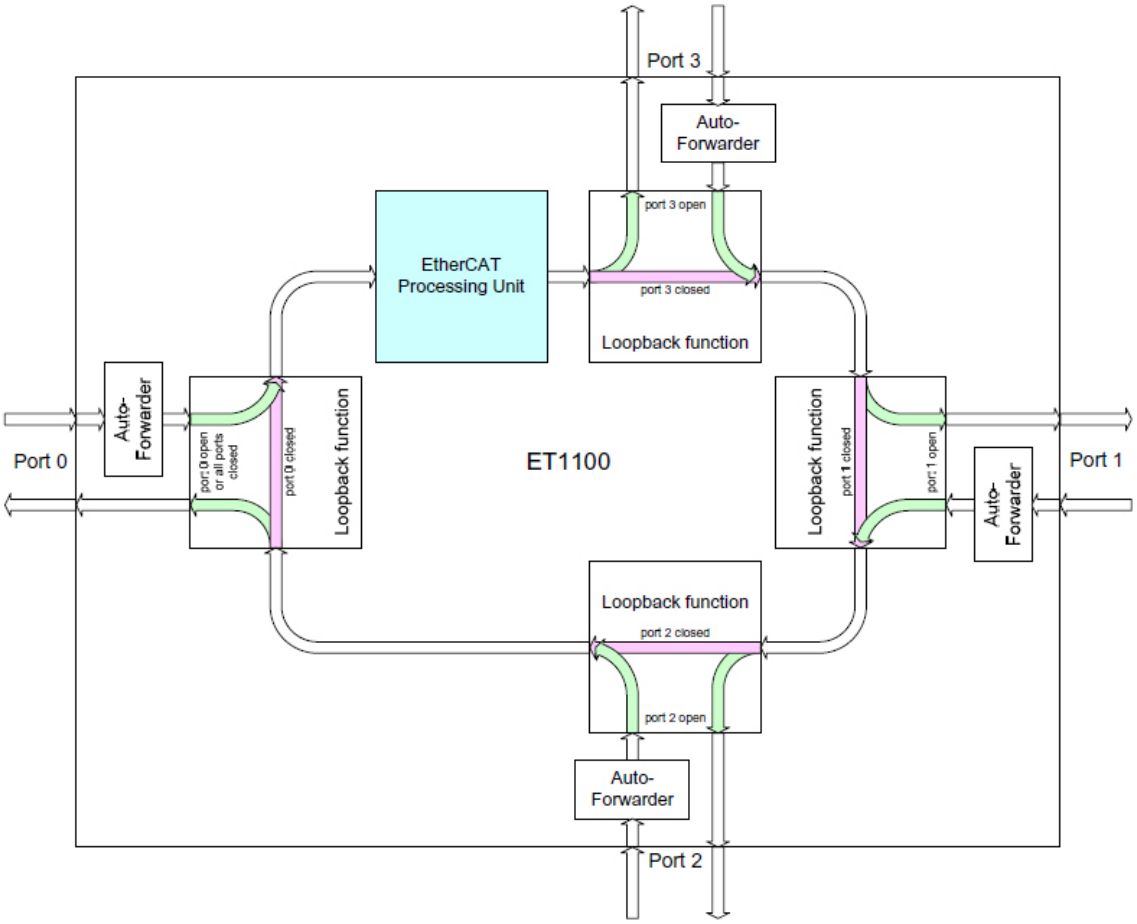
At the setting of the EtherCAT connection, the EHU leads the slave elements through these setting sections (Init -> Op). The *Operational* is the target state in which the query/setting of cyclical data operates with all tools. It controls all devices on the network together, and if it detects a problem on any of them, it does not set the other slave elements to the next state either. The XML description file, belonging to the slave elements contains the tasks to be carried out and data to be set per every level.



Picture 2: The state machine

1.4 Connection modes of slave elements

The elements have 4 ports from among which the manufacturer decides which ones to use. As the elements are connected in a row, 2 ports are used for sure on the device. A slave element may occur on which 3 ports are used (EPU periphery connecting unit), as it forwards data from the surface with a normal RJ45 connector (100BASE TX standard) to the EBUS/LVDS surface. On every port there is a monitoring unit which controls data. In case there are not any more units connected to the port, it will not send out data but forward them to the next port.



Picture 3: Data flow between ports

1.5 Communication of slave elements with the EHU

EHU is the initiator which provides the room in the EtherNET package for the slave to forward data. The handover of cycles is carried out in *Operational* mode. The description files of all elements contain the data - to be read and forwarded - of the given element (PDO - **P**rocess **D**ata **O**bject). These data are written and read out in all cycles. PDO data can be written or queried in a pre-set channel (SM - **S**ync**M**anager). The description file contains also the settings of this channel. SM receives data from the EHU and makes them available for the slave, and back. SM has an individual Watchdog timer, which, in case it is not set in time, may automatically block the outputs of the slave. Thus, in case a slave element remains alone (does not receive data from the EHU) its outputs will not remain turned on either.

1.6 EtherCAT error monitoring

In case the sending/receiving of data is not successful, this is monitored on one hand by the slave (the SM Watchdog timer expires if it does not receive data in time) and, on the other hand, by the EHU (the EtherCAT commands have to be marked by all addressed devices). The errors will be recorded in the log file. The connection becomes interrupted due to a communication error if there are 5 errors in 200 consecutive cycles.

Communication errors:

- *00810500 EtherCAT telegram index error* - The sent out data package (telegram) becomes marked by a serial number by the EHU, and the serial number of the reply telegram will not be the same with the one of the sent out telegram.
- *00810600 EtherCAT telegram WKC error* - The sent out data package was not read by all addresses.
- *00812400 EtherCAT transmit error* - EHU was not able to send out data.
- *00812500 EtherCAT receive error* - Data have become damaged. In the EtherNET frame the controlling amount is not appropriate.

The error counters of slave elements become recorded in the log, too; they may refer to circumstances disturbing the communication.

- *00811800 EtherCAT FRAME/PHYS error* - The slave element interprets the received data, too; and if it finds an error, it will step the error counter.
- *00812100 EtherCAT link error* - The slave continuously tests (pings) the connection with the devices connected to it. In case it does not receive any replies after a built-up connection, it steps the error counter.

2 EtherCAT setting window

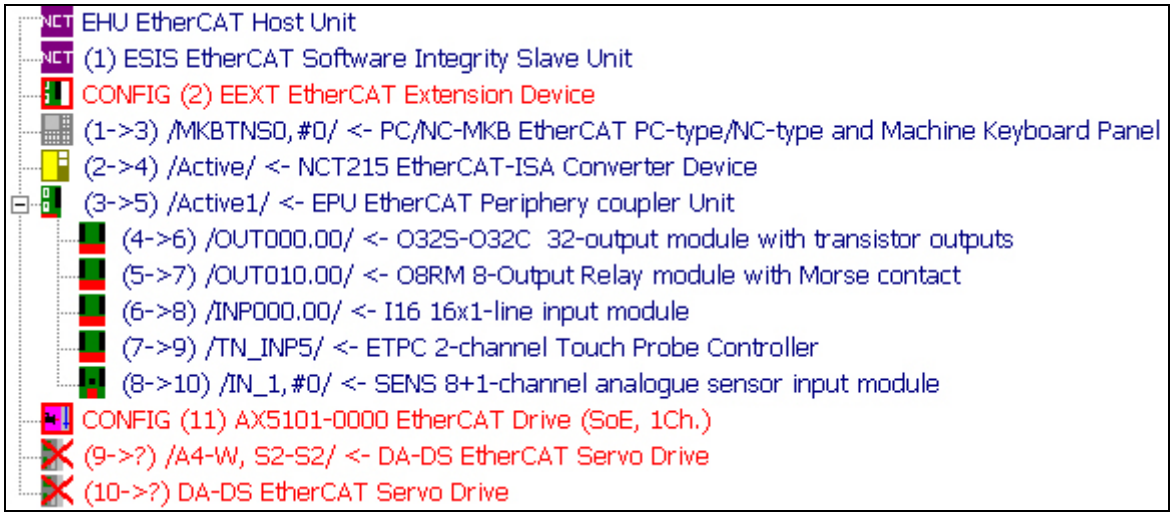
In the control unit every element has to be assigned to a system variable. The type of every element has to be indicated (input, output, drive, etc.) and one has to select, within this, under what address shall data be looked for.
The EtherCAT window is available with the button *Main menu -> Service -> EtherCAT Settings.*
By clicking on the button, the reading of EtherCAT data will start. Definition files describing the elements can be found in the subdirectory *\StorageCard\Config\XML.* At the time the window starts, it will process files found in the directory and it will display elements based on these data. Every file with an XML extension will be read. In case it does not find any definition files to the slave element, it will display it as an "*Unknown*" element.

2.1 Elements of the EtherCAT window

2.1.1 The menu

- *Save Config* - NC saves the set data and will store them in the directory *\StorageCard\Config\XML* in the file *ECATdef.sys.*
- *Load Config* - loading of a saved setting.
- *Set Config* - the control unit downloads the set data to the devices and the communication will start.
- *AutoScan* - the control unit checks what kind of devices are connected and displays them in the list of elements. It will stop the communication and it can be started again only with the button *Application of settings*.
- *Restart* - it will set all EtherCAT elements to the initial state (Init) and thereafter it will lead them through the state machine into *Operational* state.
- *Authorization* - it is necessary for displaying those data which can be modified only by persons having an authorization.

2.1.2 List of slave elements



Picture 4: The list of elements

Meaning of pictures:

- *Top element* - it is the EHU element where the communication starts and ends.
- *Element icon* - it is contained by the definition file; it may be unique at all elements. In case there is no such data, an empty icon will appear.
- *Red frame* - the element is not yet included in the group of set elements, i.e. no information is saved about it. It will be handled by the system as a not used device, and its data will not be available.
- *Red X* - In case the icon is crossed by a red X and it appears at the bottom of the list, then, although this element is defined, there is no connection with it at the moment.

Interpretation of the displayed text:

- Red colour - the element is not yet included in the group of set elements, i.e. no information is saved about it. It will be handled by the system as a not used device, and its data will not be available.
- (1->3) - The place of the element in the EtherCAT chain. The first number (1) indicates the place at which it was between the saved data, and the second number (3) indicates its current place.
  - 1->3 - The element was saved at the first place but currently it is at the third place. 2 elements were added to the chain.
  - 1 - The element is at the 1 place.
  - 9->? - The element that was saved at the 9 place cannot be found in the list. (Here a red X indicates, too that this element cannot be found.)
- /INP000.00/ - In the angled brackets we can see the PLC symbol to which the element is assigned. (This does not reflect the whole assignment, it is displayed based on the first PDO).
  - /A4-W/ - At drives the names of assigned axes can be seen (based on controlling parameters)

- In case it does not appear, it means that this element is not assigned to any system variable.

- <- Element name - the description file contains the name of the element that is displayed here. In case it does not find any data to the element, then the text "*Unknown*" will appear.

Elements displayed in red were modified in the list. An intervention of the operator is necessary to start the EtherCAT communication.
The element can be selected from the element list, the data of which appear in the right side in the data window. The selected element appears in the element list with a different background colour.

2.1.3 Data of slave elements

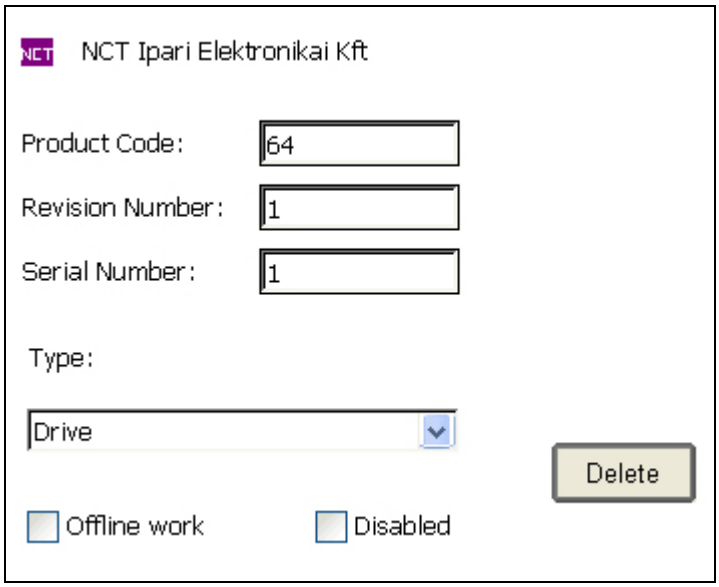
Data of elements appear in a grouped form on the sheets in the window on the right side. Data interpreted for the given group appear by selecting the sheets.

Sheets appearing without authorization:
- Vendor - Unique identifiers of the element.
- Online - it displays the state and error counters of the element.

Data bound to authorization:
- Data - Assignment of elements to the controlling data
- PDO - Displaying of cyclical data. It can be seen only if there is a data flow.
- Memory - the content of the EtherCAT memory can be queried.
- Drive - data helping the setting of NCT drives.
- SoE – Management of **S**ervo drive profile **o**ver **E**therCAT data
- FoE - **F**ile Acces **o**ver **E**therCAT

2.2 Unique identifiers of the element

Unique identifiers read out from the device appear in the *Vendor* window.

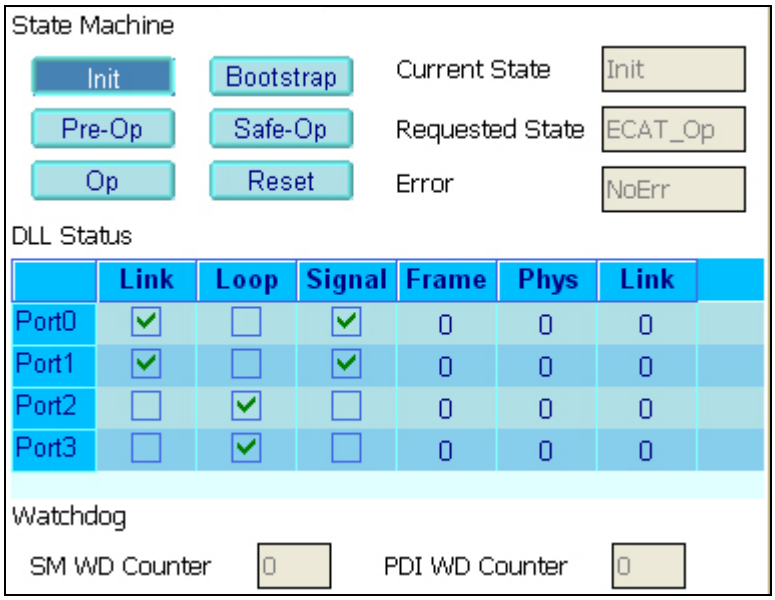


Picture 5: The vendor window

- Icon + name - the logo and name of the issuer
- Product Code - the identifier of the product (a hexadecimal number)
- Revision number - the revision number (a hexadecimal number)
- Serial Number - the serial number of the product (integer)
- Type - it indicates the type of system variable to which the element is assigned.
- Offline work- we separate the element from the system and it will not receive any data from there.
- Disabled - an element that is not used.
- Delete - it serves for the deletion of the element and it becomes active if the element is not connected to the EtherCAT chain any more.

2.3 State and error counters of the element

The *Online* window displays the state and error counters of the element.



Picture 6: The state window

- *State machine* - the buttons indicate the state in which the device is currently. The state change can be initiated on the device by pushing the button.
- *Current* - the current state of the device; it displays the state of the state machine by a text, too. *(e.g.: SafeOp* - the current state is *Safe-Operational*
- *Target state* - the state requested by the control unit. The control unit wants to send the device to this state. *(e.g.: Op* - The control unit will send the *element* to the *Operational state.*
- *Error* - Error signal. Code of the error signalled by the device. In case the device does not carry out the state change, it indicates here the error during which is has not been carried out. The name or code of the error will appear here. *(e.g.: NoFirmware* - there is no operating software in the device.
- *State of DLL* - state of the data communication surface per ports.
  - *Link* - the device is connected.
  - *Loop* - the device is not connected and there is no data traffic.
  - *Signal* - on this port there is a data traffic.
  - *Frame* - the EtherCAT package error counter of the device. It counts those errors which have occurred during physical or internal procession (too long package, controlling amount error, Ethernet header error, etc.)
  - *Phys* - error occurred on the physical surface. (A bit was lost, coding error, etc.) This error is counted also by the *Frame*.

- *Link* - the port checks whether there is another device connected or not. And, if it managed to create the connection, it will continuously monitor it and count the errors.
- *Watchdog* - expired timer counter
- *SM WD counter* - no data were delivered in time to the communication surface (**S**ync**M**anager).
- *PDI WD counter* - the internal unit was not able to read/write the data in time.

2.4 Communication data:

By selecting the EHU element in the list of elements, we can display the statistics of data currently in process in the *Online* window. In the columns we can see the sent out data packages per channels. Numbers indicate the active channels.

In the rows the following data will be displayed:

- Sent frames - counter of sent out packages.
- Lost frames - counter of defected packages.
- Tx / Rx errors - the number of those packages from among the erroneous packages in which there was a sending or receiving error.
- Index errors - the packages are marked with an identifier, and if in the given channel not the sent out package can be found, there will be an Index error.
- Number of telegrams - number of commands in the current package.
- The most telegrams - the number of biggest commands in the packages sent out so far.
- Length (byte) - the length of the current package in bytes.
- Longest frame - the size of the longest package from among those sent out so far.

It begins to count the values from the setting of the EtherCAT connection. The *AutoScan* menu option deletes the values (as in this case it begins to build up the connection from the beginning).

2.5 Setting of authorization

By clicking on the *Authorization* button of the menu, we have to enter the controlling code in a pop-up window. The setting of elements can be performed exclusively by authorized users. A controlling code can be received by the authorized persons from the distributor of the machine.
In the pop-up window the following question appears: "*Would you like more authorization?*" We have to click on the option *Yes*, and a code will appear where we have to enter the controlling code. The serial number of the control unit becomes displayed at the title *SerialNr.*

2.6 Adding of a new slave element

We have to connect the new slave element to the EtherCAT network. On every element there is an L/A (Link and activity) led per ports, indicating that the given port is connected.
States of the L/A led:

- Not lighting - it is not connected
- It lights continuously - it is connected and there is no communication
- It blinks - it is connected and there is a communication, too.

There is also a RUN led on the elements, indicating the state of the device.
States of the RUN led:

- Not lighting - Init state
- Blinking (slowly) - Pre Operational state
- Blink up - Safe Operational state
- Continuously lighting - Operational state
- Vibrating - Bootstrap state

After turning on the RUN led does not light, i.e. the device is in Init state.
After connecting the slave device, we can look for the device in the control unit by the menu option *AutoScan.* The new element will appear in the list of elements, according to the cabling order. Its icon is in a red frame and the text starts with a red CONFIG title, indicating that the element has not been set yet.

2.7 Removal of the not used slave element

It is possible to remove the not used elements; however, only if the device is not connected to the EtherCAT network. Then the element will appear in the list with an icon crossed with a red X and the text will be displayed in red, too. The removal of the element can be performed on the *Vendor* sheet by pushing the button *Delete*. The element will be deleted from the list after approving the appearing confirmation question ("Do you really want to delete this element? : the name of the slave element").
In order to save the modification we have to push the menu button *Save Config*.
In case we leave an element in the list which is not connected, the control unit will start always with an error message.

2.8 Replacement of the slave element

At the replacement of an element, the new slave element can be placed to the place of the old one, but also to a new place. The control unit stores the settings based on the unique identifiers. The new slave appears in the element list with a red CONFIG title and the old slave element with a red X. The settings of the old element have to be transferred also to the new one. After the setting we have to delete the old element from the list and we have to save the settings by the menu button *Save Config*.

In case of slave elements where the serial number is not distinguished, the system will organize them in a serial order, and their settings will be stored in a serial order, too. In case we place an element with a different serial number to the place of such an element, the setting

of the preceding elements with the same serial number will go wrong. The system stores the setting based on unique identifiers, and if the unique identifier is not unique (as it contains elements with the same serial number), the identification system will go wrong and the assignments of elements will not be clear. In case in the list there are elements of the same type, we have to check whether their identifiers are set correctly. In case we still have to replace such an element, we have to carry out the setting of all elements of this type (on which the product number, revision number and serial number was the same).

2.9 Setting of slave elements

We need an authorization for the setting of elements. Data of the element which can be set appear in the *Setting* window. The displayed data may be different based on the types of elements. Data which can be set will appear in the *Data* column in the left. The name cannot be modified. The assignment can be carried out in the *Address* column. By double-clicking on the field a drop-down menu appears from which we can select the proper setting. By the marking *Not used* the data will not be handed over to the control unit. In the *Note* field other information belonging to the system will appear.
The control unit will remember the settings only if we save them with the button *Save Config*. Modifications will become valid only with the button *Set Config*, and only if the communication was stopped (the button *AutoScan* will stop the communication and build it up once again from the beginning after the button *Set Config*).

2.9.1 Keyboard (MK machine control panel)

Assignment of data of the machine control panel with the PLC variables. Buttons, button lamps and the override switch belong here. It assigns the addresses PLC MKBTNS, MKFOVER, MKSOVER and MKLEDS with the device. Currently there can be 4 pcs of control panels defined in the control unit. The address, on which the PLC can handle the element can be indicated in the *Address* column in the *Keyboard* row.
(e.g.: Keyboard -> MKBTNS0,#0 - data of the first control panel in the PLC.

In case a manual wheel is connected to the machine control panel, we have to determine on which address it shall be handled by the PLC. The built-in handwheel that can be fit to the machine control panel, has to be addressed in the *Built-in handwheel* row.
(e.g.: Built-in handwheel -> HWBITS,#3 - the address of the last (HWBITS + 3) handwheel in the PLC

It is possible to fit up to four pieces of placed-out manual wheels through a CAN bus, which can be addressed as *External handwheel x* in the row corresponding to their set address. PLC is able to handle altogether four pieces of handwheels. HWBITS and HWMOVE data will be handed over to the PLC.

In case we enter the same addresses e.g. for both the built-in handwheel and the placed-out handwheel, the PLC is going to see only the placed-out handwheel.
(e.g.: Handwheel set to the address 4 CAN - External handwheel 4 -> HWBITS,#3 – handwheel data in the PLC.

Displayed data: *Machine control panel of the type PC/NC-MKB*

Data	Address	Comment
Keyboard	MKBTNS0,#0	
Built-in handwheel	HWBITS,#0	
External handwheel 1	Not used	
External handwheel 2	Not used	
External handwheel 3	Not used	
External handwheel 4	HWBITS,#3	

2.9.2 Head unit (EPU periphery fitter)

It fits the LVDS bus to the 100BASE-TX line. Besides this, it is able to switch a Machine On (MON) state to the relay output that can be found on it. Currently the system is able to switch 5 pcs of such elements. We can select in the *Address* column, whether we want it to switch its outputs upon the MON indicator or not.
(e.g.: MON -> Active1 - it will switch the Machine On relays on the card.

Displayed data: *EPU EtherCAT periphery connecting unit*

Data	Address	Comment
MON	Active1	

2.9.3 Input (line input module)

The input unit assigns data from the *INP000* symbolic address to the *INP071* address of the PLC. We have to select the starting address of input elements, and based on the length of data, the system will assign the corresponding addresses in a consecutive way.
E.g. on a 16-channel input card, by selecting the INP000.16 starting address in the Address column, it will assign INP000.16 -> till INP000.31.
In the *Comment* column we can see the value range of the (symbols of) calculated addresses.

Displayed data: *I16 16x1-line input module*

Data	Address	Comment
Starting address	INP001.00	INP001.00-INP001.15

2.9.4 Output (line output module)

The output unit assigns data from the *OUT000* symbolic address to the *OUT070* address of the PLC. We have to select the starting address of output elements, and based on the length of data, the system will assign the corresponding addresses in a consecutive way.
E.g. on a 32-channel output card, by selecting the OUT010.00 starting address in the Address column, it will assign OUT010.00 -> till OUT010.31.
In the *Comment* column we can see the value range of the calculated addresses.

Displayed data: *O32S-O32C 32-line transistor output module*

Data	Address	Comment
Starting address	OUT000.16	OUT000.16-OUT010.15

2.9.5 Drive (DS/DA servo amplifiers)

The drives are assigned to logical axes based on a physical identifier. The PLC is able to identify the drive based on the physical index. PLC accesses status data of the drive on the address DN_STAT and the data of the command on the address DN_CTRL. The control unit assigns the position and basic signal data to the logical axis based on this address. In the *Address* column we can determine the physical index, to which we have to refer also in the parameters of logical axes.
E.g.: axis address -> 1 - the PLC receives data on the first DN_STAT,#0 address (it is #0 because this is the first index in the PLC) and the axis handler will assign it to those logical axes on the parameters of which there is 1 set, too.

Displayed data: *DA-DS EtherCAT servo amplifier*

Data	Address	Comment
Axis address	1	

Displayed data: *TTLADO 2 pcs of TTL encoder input and 2 pcs of analog and digital output modules*

Data	Address	Comment
Address of the analog output - A	1	
Address of the analog output - B	2	
Output address of the stepping motor - A	Not used	
Output address of the stepping motor - B	Not used	
Input address of encoder - A	1	
Input address of encoder - B	2	

2.9.6 ISA-104 Fitting module

The unit receiving the cards of the NCT 104 control unit. The system handles 1 such element and it will automatically distribute the input and output data starting from the very first address.
(e.g.: ISA -> Active - the unit is used.

Displayed data: *NCT215 EtherCAT-ISA transformer unit*

Data	Address	Comment
ISA	Active	

In case there is such element in the system, it will automatically reserve the first 4 addresses for the probe. In case there is a separate probe card (ETPC) used, they have to be set for addresses 5-8.

2.9.7 Probe (ETPC)

The address of the probe unit determines on which physical input shall the signal change be monitored by the NC at the probe functions (e.g.: G31 XI10 P1). The same address assigns the values of TN_INPx and TN_OUTx variables of the PLC. The system currently handles 8 pcs of probes. The address of the probe can be selected in the *Address* column, based on the TN_INPx variable of the PLC where x indicates the selected address.
(e.g.: PROBE1 -> TN_INP5 - the system handles the probe on the fifth input and the PLC sees it on the address N_INP5.

Displayed data: *ETPC dual-channel electronics fitting probes which provide the contact*

Data	Address	Comment
PROBE 1	TN_INP5	
PROBE 2	TN_INP6	

In case there is an ISA-104 fitting element in the system, it will automatically reserve the first 4 addresses for the probe. In case there is a separate probe card (ETPC) used, they have to be set for addresses 5-8.

2.9.8 Analog (SENS)

The analog card assigns the state information of the card to the IN_1 address of the PLC. It receives commands from the address OP_1 and it displays data starting from the address ANINPUTS. Every indicator can be addressed from 0 to 31, i.e. the system is able to handle 32 pcs of channels. The assignment of the card to a channel can be carried out in the *Address* column. IN_1,#0 indicates the first, whilst IN_1,#31 indicates the last channel.
(e.g.: ANALOG -> IN_1,#5 - then the PLC handles the signals on the address IN_1 + 5).

Displayed data: *SENS 8+1 pcs of analog input fitter module*

Data	Address	Comment
ANALOG	IN_1, #0	

2.10 Service functions

The window contains certain service functions which help the user during their setting.

2.10.1 PDO data

- Cyclical data are displayed in the *PDO* window. Here we can see data which are contained in the XML description file. In the header of rows the direction of data can be seen.
- *OUT* - it is written by the NC in all cycles.
- *IN* - it is read by the NC in all cycles.

PDO data appear in the *Name* field with the name that can be found in the description file.
The current value of data can be seen in the *Data* field. The format of displayed data is indicated in the description file. At data of the type of a number, it is possible to display the hexadecimal form by ticking the check box in the *HEX* column.

The value of data of true/false state is displayed in the *Sign* column in the check box corresponding to its state.

	Name	Data	Sign	HEX
OUT	LEDs	0x00001		✓
OUT	OutputByte0	0x01		
OUT	-Bit00	TRUE	✓	
OUT	-Bit01	FALSE	☐	
OUT	-Bit02	FALSE	☐	
OUT	-Bit03	FALSE	☐	
IN	SHORT0	1	✓	

2.10.1.1 Offline work:

In the regime without connection, devices do not receive data from the NC but we can determine values in the *PDO* window for data which can be written (located at the OUT header). We can turn on the regime by the setting of the check box *offline work*. Data must be entered directly into the *Data* field. The input field opens upon a double-click. After entering the value, the setting can be made by the *Ente*r button. Elements of the true/false state can be set in the check boxes located in the *Sign* column, by a double-click.

2.10.2 Queriy of EtherCAT registers

The query of registers is made in the *Memory* window. The memory address has to be determined in the *Address* field in a hexadecimal format, in a length of maximum 4 characters. The length of data to be read has to be indicated in the Length field, in bytes. The erroneous address and length will be interpreted by the system and it will not let us leaving the field till it is not corrected.
The query starts upon the *Update* button. In case the *Auto update* check box is turned on, it will continuously update data every 100 msec.
Data will be displayed in a list. The *Byte index column* indicates the place of the queried byte. The value will be displayed in the *DEC* column in a decimal, and in the *HEX* column in a hexadecimal format.

Displayed data: e.g.: *Address: 600 Length (Byte): 8*

Byte index	DEC	HEX
0	4	4
1	0	0
2	0	0
3	1	1
4	12	C
5	0	0
6	0	0
7	7	7

In the documentation of the EtherCAT chip we can find the description, what kind of data can we found on the various addresses. In the window we can only query the values but we can not set them.

2.10.3 Software update on the device

On those elements which have an individual software and handle the **F**ile Acces **o**ver **E**therCAT (FoE) surface, it is possible to update the software.
On the *FoE* sheet the following fields can be found:

- *Device name* - the name of the device from the description file
- *ECAT state* - state of the element (Init,PreOp,SafeOp,Op,Boot)
- *Firmware file* - the selected file that can be downloaded into the device
- *Update state* - the state of the download

Buttons

- *Bootstrap* - we have to have the device stepped to the Bootstrap regime where the software update is permitted.
- *Download* - startup of software download
- *Open* - selection of the file which can be downloaded
- *Abort* - interruption of the process

Process of the software update:

1. Sending of the device to Bootstrap state - by using the *Bootstrap* button. In the ECAT status field the Bootstrap title has to appear.
2. Selection of the file that can be downloaded - by clicking on the *Open* button a file dialogue window opens in which we can select the requested file.
3. Download - it starts with the *Download* button. The download process can be seen in the *State* field and under it, and also in the process indicator.

2.11 Parameterization of drives from the control unit

The parameterization of NCT drives can be performed from the control unit in the *SoE* window. SoE indicates the **S**ervo drive profile **o**ver **E**therCAT surface through which the parameterization communication of the drive takes place. The control unit displays the parameters and sets their values.
Parameter data stored in the drive:

- *IDN* - parameters' identifier (**Id**entification **n**umber).
  - Structure of IDN (S/P – x – yyyy).
    - *S/P initial letter* - "S" indicates the standard parameters, "P" indicates the manufacturer's parameters.
    - *x* - indicates the parameter group. Its value can be between 0 and 7.
    - *yyyy* – indicates the number of the parameter. Its value can be between 0 and 4095.
- *Name* - in a textual format
- *Attribute* - the length and type of the parameter, together with the possibility of its modification and the way of its displaying.
- *Measurement unit* - in a textual format
- *Maximum and Minimum* - according to the type of the parameter value
- *Value* - the parameter value, according to the type indicated in the feature
- *Default value* - the parameter takes it up at the startup

In the group of parameters there is such a read-only parameter in which the list of parameters existing in the drive is indicated. It has a fix identifier (IDN S_0_17). The control unit displays the list of parameters on the display based on this parameter.

	IDN	Name	Set value	Actual value	Unit	Status	Save
P - -	P-0-0001	Velocity control cycle time	125	125	µs	OK	✓
P S O	P-0-0057	Eletrical commut. offset	90	90	deg		✓
P S -	P-0-0058	Motor EMF	54	54	mV/rpm	OK	✓
P - -	P-0-0059	Motor Brake	0x0	err		LdErr	✓
- - -	S-0-0131	Control word	0	0		RO	

Elements of the parameter list

- *header of rows* - modifiable states of the parameter.
  - P - can be modified in the Pre Operational state.
  - S - can be modified in the Safe Operational state.
  - O - can be modified in the Operational state.
- *IDN* - the number of the parameter identifier
- *Name* - the name of the parameter
- *Set value* - the set value, which is stored in the control unit.
- *Current value* - the current value that can be found in the drive
- *Unit* - the measurement unit of the parameter.
- *State* - the state of the parameter.
  - Empty field - no data modification was made, no operations are present on the data.
  - *cmd* - this is a command-type parameter
  - *RO* - the parameter is read-only (**R**ead **o**nly)
  - *OK* - the writing/reading of the parameter is all right
  - *ImpErr* - during the file opening (import) an error emerged on the parameter
  - *SetErr* - the download of the parameter value into the drive was not successful
  - *LdErr* - the reading out of parameter value from the drive was not successful
- *Save* - it indicates those parameters which need to be loaded into the drive at the startup. These parameters are indicated in the list of mandatory parameters to be downloaded (IDN S_0_18, IDN S_0_19).

Buttons

- *Save* - the control unit saves the mandatory parameters to be downloaded (indicated in the *Save* column)
- *Import* - we can load a previously exported list of parameters
- *Export* - saving of all parameters in the drive into a file selected by us.
- *Download* - downloading of all set parameters into the drive
- *Upload* - reading out of parameter values in the drive

Errors sent by the drive appear in the Messages window appearing under the parameter list. In case there was a writing error on the parameter, it will appear also in the *State* column of the parameter, but the character of the error will be displayed only in the messages window. (e.g.: a parameter cannot be modified as it is password protected, or the parameter value exceeds the maximum value, etc.)

2.11.1 Parameter modification

The modification of parameters can be performed in the parameter list. By double-clicking on the cell in the *Set value* column in the row of the parameter to be modified, a pop-up window appears in which we can modify its value according to the type of the parameter.

In the header of the appearing window, the parameter name can be seen, together with its measurement unit in brackets, if any.

2.11.1.1 Simple parameter type

The actual value is displayed in the input field. On the left we can see the minimum below the field, and on the right we can see the maximum value. The value has to be entered in a format corresponding to the parameter. E.g.: in case of a hexadecimal number we cannot enter decimal values. In such cases the background colour of the field turns red, indicating that the value contained in it is not appropriate. In case of an erroneous value the *Save* button will be inactive.
Data types

- *Binary* - the value appears with a *0b* prefix, but it is not mandatory to write it out. The values that can be entered can only be *0 or 1*. *(e.g.: 0b1110*
- *Integer* - only an integer can be entered on the parameter. The characters that can be entered can only be numbers between *0 and 9*. In case of a number with a sign the sign is (-). E.g.: 36 or -36
- *Hexadecimal* - the value appears with a *0x* prefix but it is not mandatory to write it out. Characters to be entered can be between *0 and 9 and A-F*. E.g.: 0xd39
- *Floating-point* - in case of real numbers we can enter also a decimal point the sign of which shall be (.). Characters to be entered can be between 0 and 9. E.g.: 123.45
- *Textual* - any characters can be entered.
- *IDN* - only values corresponding to the identifier can be entered here. It has to begin by an *S or P* by any means, and after it we can enter the parameter group, followed by the serial number separated by hyphens. It is mandatory to write out hyphens, too.
  E.g.: S-0-0015 or P-1-0113.

2.11.1.2 Complex parameter type (list element)

Data are displayed in the list. Under the list we can see the current number of elements at the *Size* title and the maximum number of elements at the *Max* title.
The elements of the list can be modified by the *Modify* button. The element can be removed from the list by the *Delete* button. We can add a new element to the end of the list by the *Add* button, or insert one into the list with the *Insert* button. At the element modification the simple parameter-entering window will appear.

The application of the modified value can be made by the *Save* button, while its disregarding can be made by the *Cancel* button. Upon the *Save* button, the value will be

downloaded also into the drive, if the element is in a state where this parameter can be written. The value will appear in the *Set value* column and after the successful download, also in the *Current value* column.

2.11.2 Saving of parameters for the startup (Startup list)

The control unit stores the set parameters to be downloaded, in a file. We can find the file in the directory \StorageCard*Config**Drive* based on the unique identifiers of the element. The file name: e.g.: 0000052E00000064000000100000028.drv

- the first 8 character - vendor ID
- 8th-15th characters - product code
- 16th-23rd characters - revision number
- 24th-32nd characters - serial number

The operation can be performed by the *Save* button. Only those parameters will be saved which are selected in the *Save* column.

2.11.3 Saving of parameters into a textual file

The saving of parameters in the drive in a textual format is made by the *Export* button. By this a textual file will be created at the place and under the name determined by us, containing all parameters and all their data, which can be found in the drive.

2.11.4 Reading of parameters from a textual file

The textual parameter file exported by the control unit can be read, too. During the reading the control unit checks whether in the current drive the read parameters can be found or not, and it compares the saved features of parameters with the features of the parameter in the drive. A hexadecimal value cannot be loaded into a floating-point variable.
It marks the erroneous parameters in the *State* column (*ImpErr*) and after the importing it notifies the user about them in a pop-up window. It will not display those parameters which cannot be found in the drive, but they will be displayed in the messages window.

2.11.5 Reading out of parameter values from the drive

We can update the current parameter values from the drive by the *Read out* button. In the parameter list the new values will appear and these data will be recorded also in the *Set values* column. In case an error occurs at the reading of some of the parameters, in the *State* column the *LdErr* indicator appears and we can check the character of the error below in the messages field.

2.11.6 Downloading of parameter values to the drive

We can forward values from the *Set value*s column to the drive by the *Download* button. The read-only parameters, the command-type parameters and the password parameter will not be downloaded. If there are parameters which are password protected, first of all we have to enter the password. Thus, during the download we will not have any problems with the password protected parameters. In case an error occurs at the writing of some of the parameters, in the *State* column the *SetErr* indicator appears and we can check the character of the error below in the messages field.

2.11.7 Modification of parameter tables

We handle the parameter tables separately in the parameter number. IDN P-**0**-0001 indicates the 0. parameter group, whilst P-**1**-0001 indicates the 1. We can copy the groups between each other by the button *Parameter table copying*. In the drop-down menu we can select from which parameter table we would like to copy into which parameter table, and by using the *Parameter table copying* button we can copy the values (they will appear in the *Set values* column). They will be seen in the drive only if we download them by the *Download* button.

2.11.8 Displaying of parameters

We can set filtering conditions for the parameters, by which we can sort the elements appearing in the list. The filtering can be made both for states and groups.
Status filtering

- *no filter* - no state filtering
- *Writable in Pre Operational state* - only those will appear which can be modified in PreOp.
- *Writable in Safe Operational state* - only those will appear which can be modified in SafeOp.
- *Writable in Operational state* - only those will appear which can be modified in Op.
- *read-only* - those parameters will appear which cannot be modified
- *Error during downloading* - elements in the *State* column of which the *SetErr* can be seen
- *Error during reading-out* - elements in the *State* column of which the *LdErr* can be seen
- *Error during reading* - elements in the *State* column of which the *ImpErr* can be seen

Group filtering

- *no filter* - no state filtering
- *parameter group x* - parameters which are in the x. group The x indicates the number of the group. (between 0 and 7).
- *commands* - procedure-type parameters will be displayed