SMI2

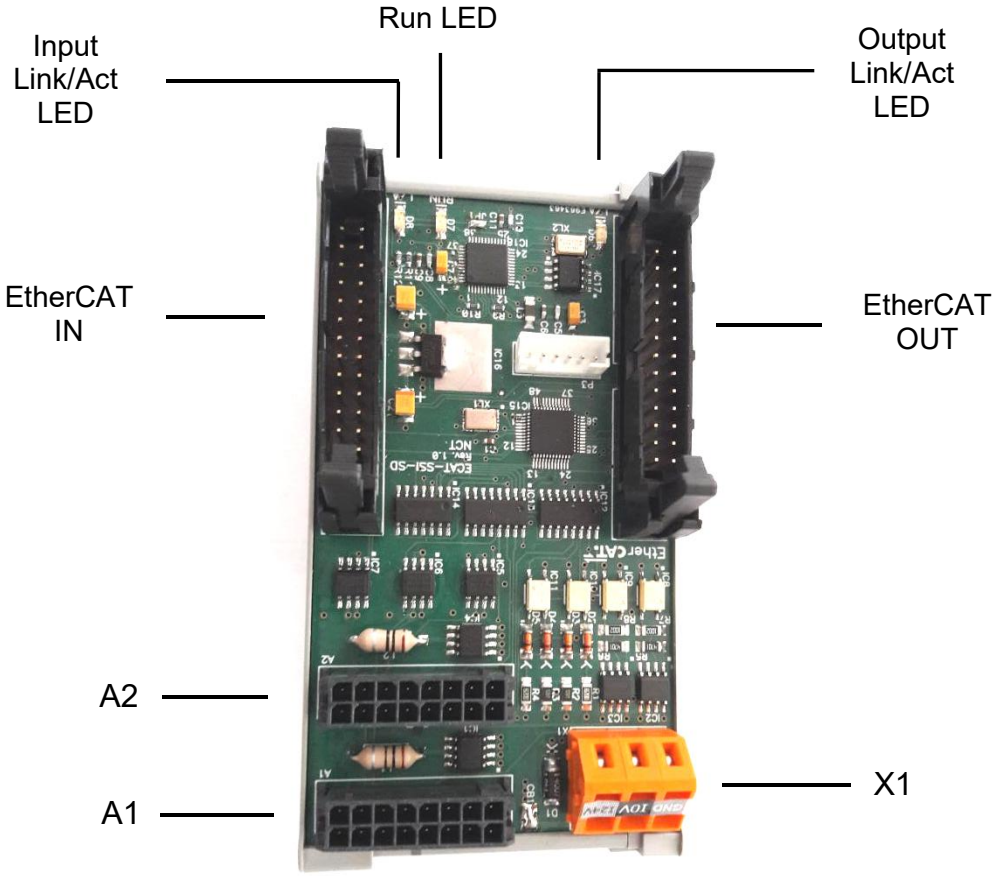
EtherCAT STEPPER MOTOR INTERFACE MODULE WITH PWM OUTPUT

Described SMI2 software version: 4.1,4.2, 4.3
Document revision: 1.5

TABLE OF CONTENTS

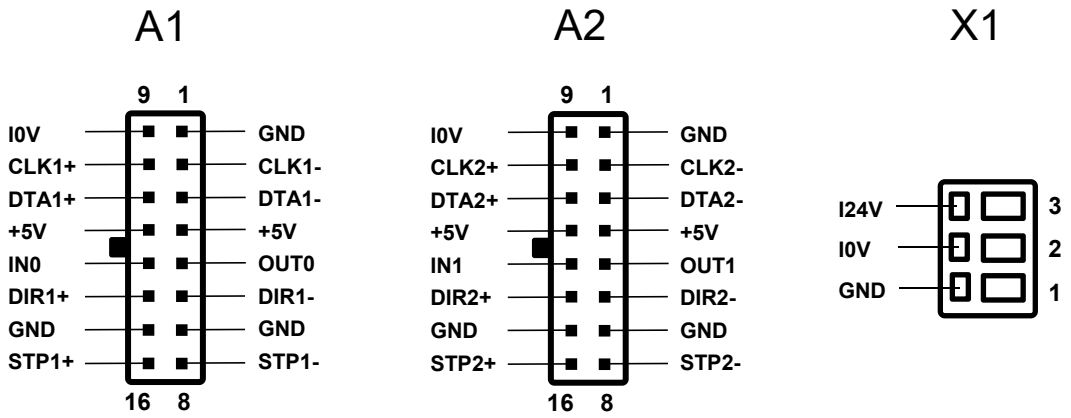
1 INTRODUCTION

The SMI2 is a two-channel stepper motor controller communicating via EtherCAT capable of receiving SSI signal encoder and generating PWM output. Each channel has one 24 V digital output and one digital input. The device can be used both in SM synchronized and DC synchronized modes.



Technical data:	SMI2
Power supply (through LVDS)	5 V / 170 mA
Encoder supply voltage	5 V
Encoder interface	SSI (RS422), 400 kHz
Stepper motor driving	STEP/DIR, max. 200 kHz, open collector
PWM output range and resolution	20Hz..20kHz, 750 ns, open collector
PWM duty cycle	0-100%
Overall dimensions	108 x 56 x 30 mm
Weight	100 g

2 CONNECTOR ASSIGNMENTS



A1/A2

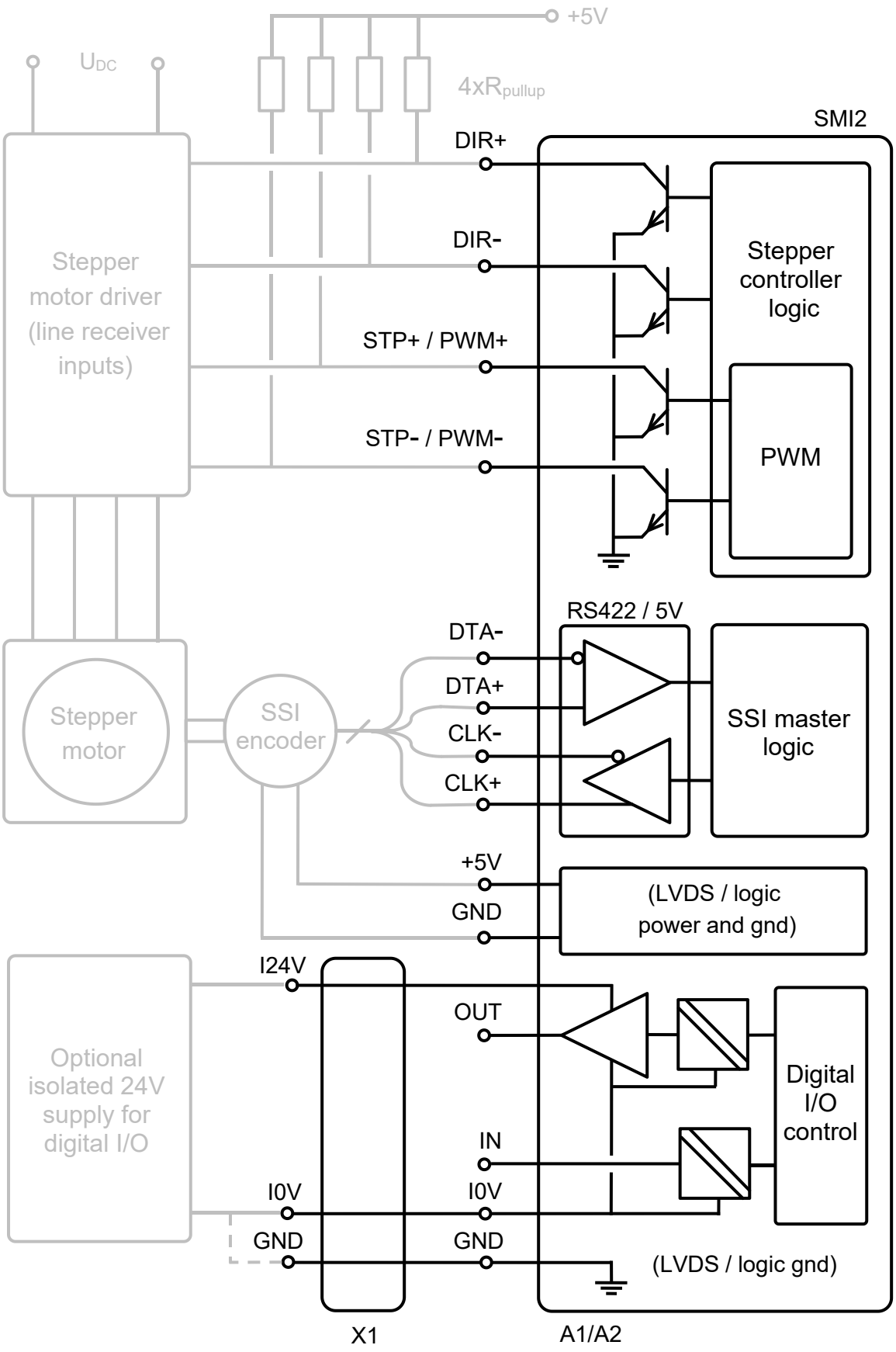
Signal	Pin	STEPPER mode	PWM mode
DTA+	11	SSI encoder data input	-
DTA-	3	SSI encoder data input inverted	-
CLK+	10	SSI encoder clock output	-
CLK-	2	SSI encoder clock output inverted	-
+5V	4, 12	5V power supply output	5V power supply output
GND	1, 7, 15	0V (LVDS / logic gnd)	0V (LVDS / logic gnd)
IN	13	Isolated 24V input	Isolated 24V input
OUT	5	Isolated 24V output	Isolated 24V output
I0V	9	Isolated 0V	Isolated 0 V
STP+	16	Open collector stepper motor driver pulse output	PWM+ Open collector pulse width modulated output
STP-	8	Open collector stepper motor driver pulse output inverted	PWM- Open collector pulse width modulated output inverted
DIR+	14	Open collector stepper motor driver direction output	-
DIR-	6	Open collector stepper motor driver direction output inverted	-

- :not used

X1

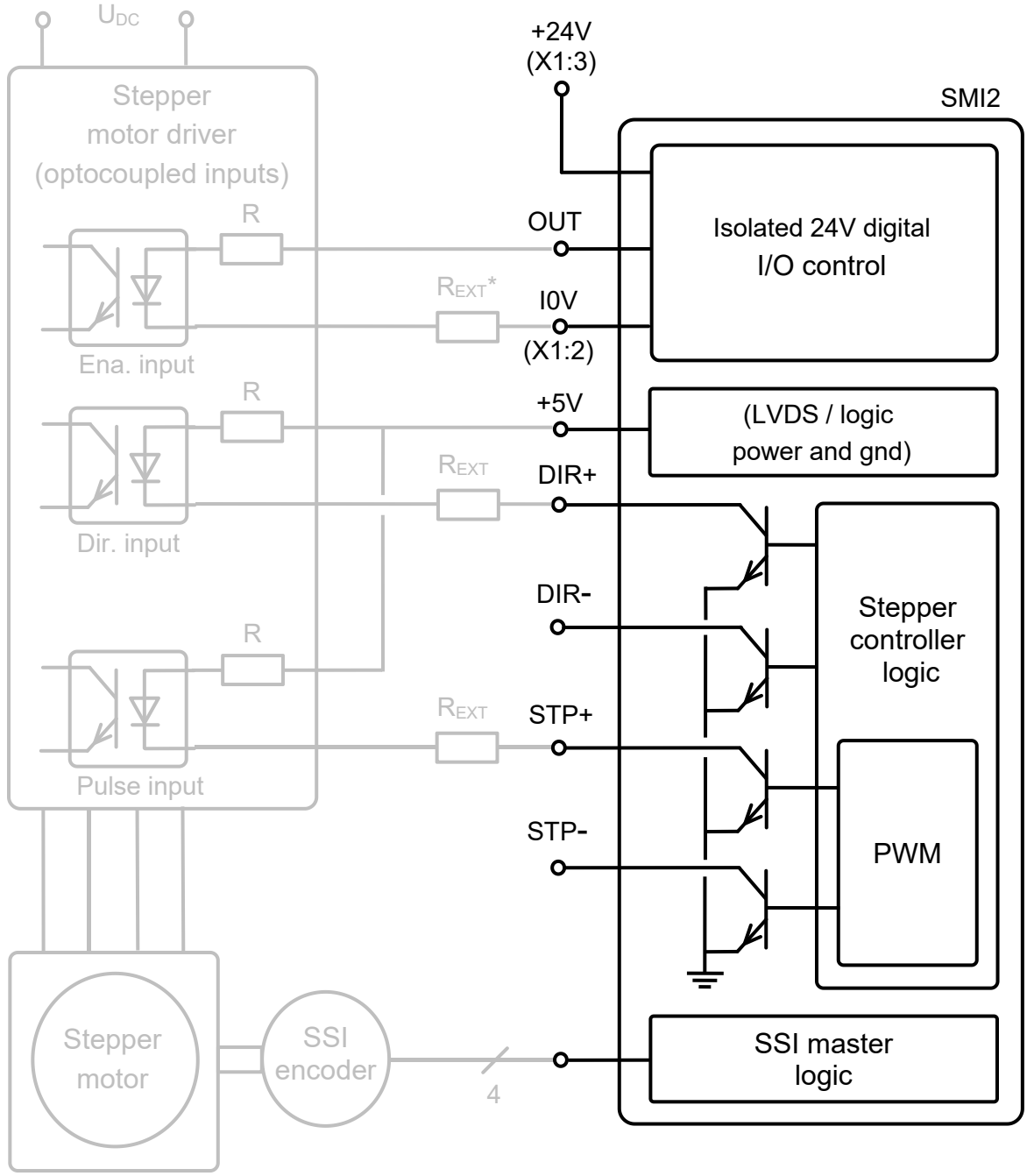
GND	1	Shield (LVDS / logikai gnd)
I0V	2	Isolated 0V (input for SMI2)
I24V	3	Isolated 24VDC (input for SMI2)

3 BLOCK DIAGRAM AND TYPICAL CONNECTION

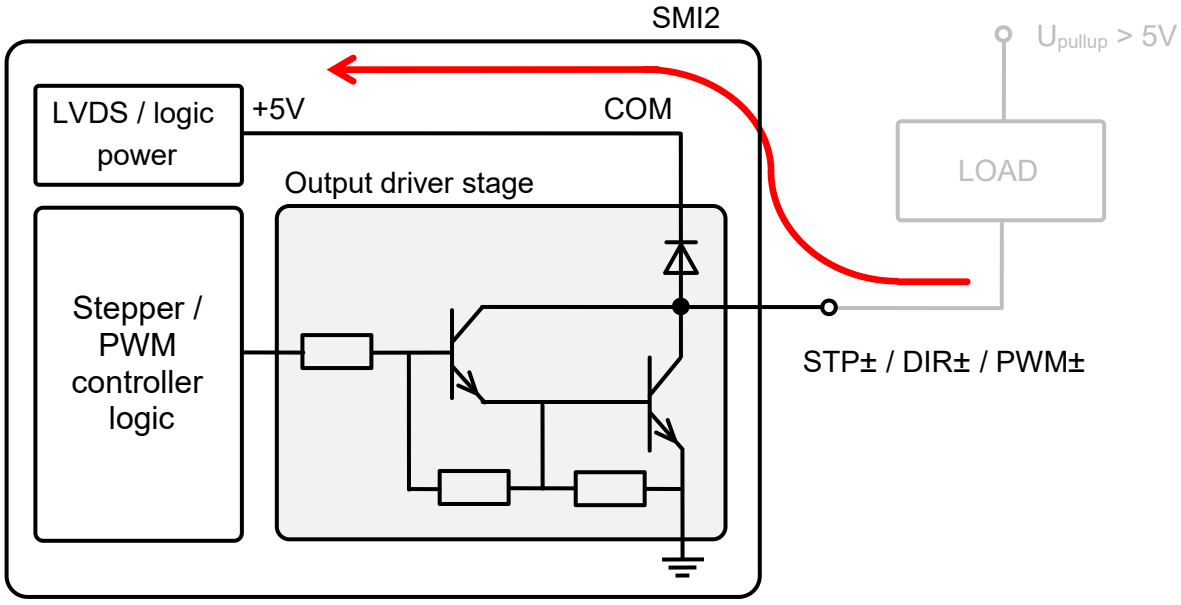


The connection diagram above is applicable when the stepper motor drive has differential RS422 line receiver inputs. In this case both STP+/STP- and DIR+/DIR- we should use pull up resistors because the SMI2 has open collector outputs. The maximal pull up voltage is 5V and the 5V power output in A1/A2 connectors can be used for this purpuse.

On stepper motor driver often we found optically isolated (optocoupled) inputs where you can use different connection as above shown.



The positive pin of the stepper motor driver's optocoupler should pull up via a pull-up resistor (the pull-up voltage can be +5V output voltage of the A1/A2 connector). The negative pin should be connected to the SMI2's DIR+/STP+ outputs (or to the DIR-/STP- inverted outputs). Depending on the pull up voltage and the built-in R resistor of the stepper motor driver there could be need to connect an optional external REXT resistor into the DIR+/STP+ branches. If the stepper motor driver's enable signal is connected to the SMI2's OUT pin (which is available via A1/A2 connector), then we should aware of the OUT sigal's voltage level is 24V. About the necessity and value of the R_{EXT} and R_{EXT}* external resistors the user guide of the actual stepper motor driver can give us information.



All the open collector outputs (STP±/DIR±/PWM±) have the same output driver stage which is shown on picture above (only one output is shown). Do not apply higher pull up voltage than 5V because an undesired balancing current can be flown form the pullup power supply into the SMI2 via the free running diode of the output drive stage.

If the pullup voltage is less or equal than 5V but it is independent form the SMI2's 5V output voltage then in case of inductive load apply an external free running diode paralell with the load because the output drive stage's built-in diode can not conduct the current which inducted by load switch off.

4 FUNCTIONAL DESCRIPTION

The SMI2 can be used in two modes: in STEPPER and PWM modes. The acually used mode is selected by the process data settings. By default, the SMI2 module start sin STEPPER mode, and in Op state, the current mode can be read out from the Output Mode (0x2000) parameter. The two modes of operation are mutually exclusive; the desired mode of operation can be selected using the PDO settings in the following way:

Mode of operation	Output PDO setting (0x1C12)	Input PDO setting (0x1C13)
STEPPER mode	0x1600 Stepper Axis Output (default)	0x1A00 Stepper Axis Input (default)
PWM mode	-	0x1601 PWM Channel Output
PWM mode with digital outputs	-	0x1601 PWM Channel Output 0x1602 Digital Outputs
PWM mode with digital inputs	0x1A01 Digital Inputs	0x1601 PWM Channel Output
PWM mode with digital inputs/outputs	0x1A01 Digital Inputs	0x1601 PWM Channel Output 0x1602 Digital Outputs

Except for the process data settings specified in the table, all other settings are invalid and result in an Invalid Input/Output Config ALStatus error. The content of the PDOs for each mode is given in the table of parameters.

In PWM mode, it is capable of generating two independent PWM signals and managing digital inputs and outputs.

In STEPPER mode, it is capable of controlling two stepper motor drives by step/dir signals, receiving SSI encoder and managing digital inputs and outputs.

The stepper driver and stepper motor handling of SMI2 is differs form the typical stepper motor solutions where the motion controller controls directly the positon (i.e. it sends the number of steps) the stepper motor driver.

The SMI2 is between the motion controller and the stepper motor driver and its set value is the SpeedSetValue variable which is proportional to frequency and it controls the speed of he stepper motor (i.e. the frequency of the STEP signal). If the value of SpeedSetValue is +2097152 (0x200000) then the frequency of the STEP signal is 200 kHz,

if the value of SpeedSetValue is +1048576 (0x100000) then the frequency of the STEP signal is 100 kHz, etc. The exact correlation between the SpeedSetValue variable and the STEP frequency is explained in chapter 4.4.1. Changing the SpeedSetValue variable changes the speed of the stepper motor i.e. constant SpeedSetValue means constant rotational speed. The real rotational speed depends on the physical step number of the stepper motor and the number of microsteps is set on stepper motor driver.

With this mode of SMI2 can be realized such a CNC control method where the CNC motion controller sends speed set value to the stepper motor drivers and receives the position form the encoders of the stepper motors, i.e. the positon loop is closed in the CNC controller.

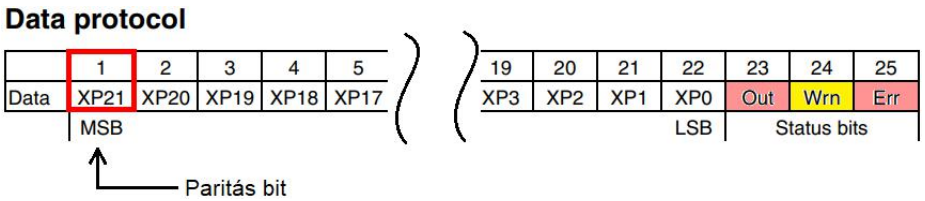
4.1 SSI INPUT

It is a two-channel input for receiving signals from encoders with SSI interface, through 5 V RS-422 interface. The frequency of the SSI clock signal is 400 Hz which is unchangeable and is the same on both channels. The encoder is read out in each EtherCAT cycle synchronized with the DC synchronous signal of the network. The read position can be read out via the process data variable ActualPosition1..2, in case of encoder error (parity error or error indicated by the encoder) the 2nd bit of the variable ErrorDword belonging to the given channel will turn on. The error bit remains turned on until the error exists.

Selection of the encoder type can be set in the parameter 0x3201:01 Encoder interface where predefined encoder types can be selected. A fully parameterized configuration mode is not supported. The selected encoder type is saved in the non-volatile memory of the device, with NCT EMA-14xx encoder type set as default. The SSI encoder handling function is part of the STEPPER mode and cannot be used in combination with PWM mode.

Encoder Type	Description
NCT EMA-14xx	Absolute mechanical multiturn NCT encoder with two or three gears.
PF PCV50-F200-SSI-V19	Pepperl-Fuchs PCV50-F200-SSI-V19 optical QR code reading head.*
KUBLER M3663.G2222	Kübler Sendix 8.M3663.XX2X.G2222 24-bit gray coded SSI encoder without parity and status bits.**
Parametrized	Free configuration of the encoder parameters is not supported.

* In the case of receiving the Pepperl Fuchs encoder, the SMI waits the SSI data in the following format:



Any of the Out and Err error bits indicates error (is set to 1), it appears on the error bit of the variable ErrorDword belonging to the encoder channel; however, the Wrn bit does not appear in the error word, therefore it is useful to display this warning bit on the general digital output of the encoder and make it available for the PLC via an EtherCAT input module.The Out, Err and Wrn bits are in the correct bit position by default,

but, for the appropriate operation, the last bit (XP21 position bit) of the encoder data has to be changed to an even parity bit in the encoder settings (using the encoder's own configuration software).

** This encoder type setting can be used for general 24-bit gray-coded SSI encoders where all the bits are position bit.

4.2 PWM OUTPUT

Two-channel pulse width modulated (pwm) square wave output function. The period time and fill factor can be set independently on the two channels. The period time can be varied between 20Hz and 20 kHz, the fill factor can be set between 0 and 100 percent in 16-bit format. The output is controlled directly by the fill factor; 0% fill factor (0) turns the output off, 100% fill (65535) turns the output on. The resolution depends on the pwm frequency; in brackets is the minimum fill factor that can be output

- below 2.5 kHz the resolution > 9 bit (D_{min} = 0.187 %)
- below 5 kHz the resolution > 8 bit (D_{min} = 0.38%)
- below 10 kHz the resolution > 7 bit (D_{min} = 0.75%)
- below 20 kHz the resolution > 6 bit (D_{min} = 1.50 %)

Variable	Data type	Description
PWM Period	UINT16	Period time is given as multiple of the resolution (750 ns). Minimum value: 66, maximum value: 65565.
PWM Duty	UINT16	Duty cycle specification. The value 0 corresponds to 0%,the value 65535 corresponds to 100% duty cycle. Illustrative values: 100% 65535 75% 49151 50% 32767 25% 16384 10% 6553
PWM Control	UINT16	It is reserved; zero has to be written.

4.3 DIGITAL INPUT AND OUTPUT

The galvanically isolated digital input is used for receiving 24 V digital signals, sensors and other devices. This function can be used both in STEPPER and PWM modes if corresponding PDOs are selected as process data. The input considers the voltage over 10 V as logical high value, which can be read in at the process data variable DigitalInput via bits IN_0 and IN_1.

DigitalInput

15	14	13	12	11	10	9	8

7	6	5	4	3	2	1	0
						IN_1	IN_0

A The digital output is a galvanically isolated 24 V digital output realized by the use of FET switch, its load current is 500 mA. The logical low value of the output is smaller than 1 V. The digital outputs are controlled via bits OUT_0 and OUT_1 of the process data variable DigitalOutput. Using of both of digital inputs and outputs the 24V external (interface) voltage should be conneced via X1 connector.

DigitalOutput

15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
						OUT_1	OUT_0

4.4 STEPPER MOTOR CONTROL OUTPUT

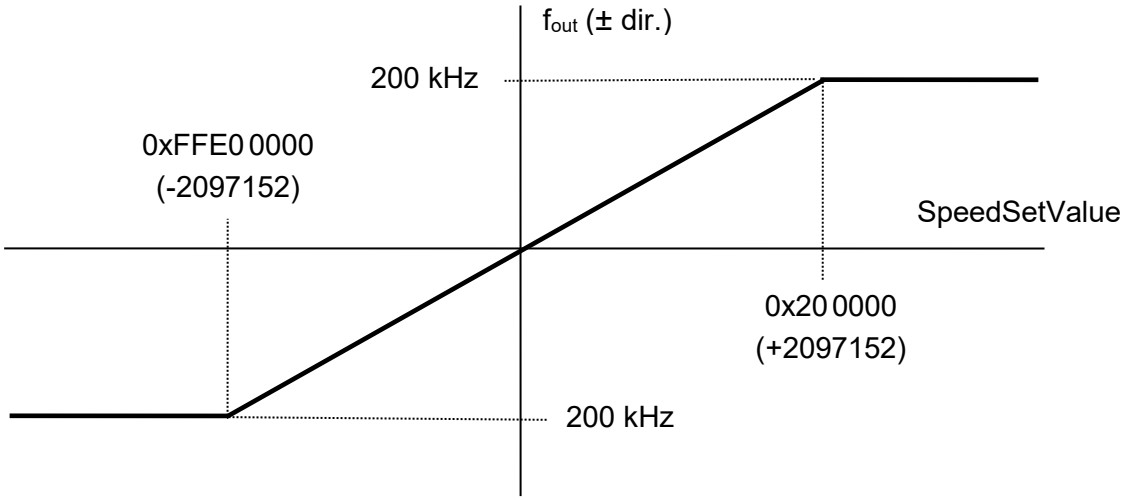
Pulse (STP) and direction signal (DIR) output for the stepper motor controller in true and false (inverted) format. The signals are issued through an open collector drive stage.

4.4.1 SPEEDSETVALUE1..2 (OUT)

Data type: INT32

31	30	29	28	27	26	25	24
SSV_31	SSV_30	SSV_29	SSV_28	SSV_27	SSV_26	SSV_25	SSV_24
23	22	21	20	19	18	17	16
SSV_23	SSV_22	SSV_21	SSV_20	SSV_19	SSV_18	SSV_17	SSV_16
15	14	13	12	11	10	9	8
SSV_15	SSV_14	SSV_13	SSV_12	SSV_11	SSV_10	SSV_9	SSV_8
7	6	5	4	3	2	1	0
SSV_7	SSV_6	SSV_5	SSV_4	SSV_3	SSV_2	SSV_1	SSV_0

Bit31..0 – SSV31..SSV0: signed speed reference signal, maximum value: 0x200000, minimum value: 0xFFE00000. In EtherCAT Op mode (the NCT controller is in NC Ready status), the value of SpeedSetValue will be valid immediately at the output of the module. The sign of the SpeedSetValue will affect only the DIR signal.



4.4.2 CONTROLDWORD1..2 (OUT)

Data type: UINT32

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
						ErrorClear	
7	6	5	4	3	2	1	0

Bit 9 - ErrorClear Clearing the errors.

When the set value of the bit ErrorClear is 1, the content of the ErrorDword will be cleared if the value of the bit ErrClearAck in the StatusDword is 0.

4.4.3 ACTUALPOSITION1..2 (IN)

Data type: UINT64

31	30	29	28	27	26	25	24
POS_31	POS_30	POS_29	POS_28	POS_27	POS_26	POS_25	POS_24
23	22	21	20	19	18	17	16
POS_23	POS_22	POS_21	POS_20	POS_19	POS_18	POS_17	POS_16
15	14	13	12	11	10	9	8
POS_15	POS_14	POS_13	POS_12	POS_11	POS_10	POS_9	POS_8
7	6	5	4	3	2	1	0
POS_7	POS_6	POS_5	POS_4	POS_3	POS_2	POS_1	POS_0

Bit 31..0 - POS_31..POS_0: current postition

4.4.4 StatusDword1..2 (in)

Data type: UINT32

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
			ErrClearAck		Abs/Incr		

Bit 2 - Abs/Incr Absolute or incremental encoder.

Bit 4 - ErrClearAck. Acknowledgement of error clearing.

A ControlDword ErrClear bit 1 értékének hatására ez az ErrClearAck bit 1 értéket vesz When the value of the bit ControlDword ErrClear is 1, this bit ErrClearAck takes the value 1 and the value remains 1 until the value of the bit ControlDword ErrClear changes to 0.

4.4.5 ErrorDword1..2 (in)

Data type: UINT32

31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
						EncoderErr	UnitErrors

Bit 0 - UnitErrors Global error indication

The bit UnitErrors takes the value 1 if any error occurs on the unit. Investigation of more error bits can explore the specific cause of the error.

Bit 1 - EncoderErr Encoder error at the connected encoder.

5 ETHERCAT INTERFACE

The module communicates with the control system via EtherCAT-LVDS bus. The module,via the LVDS input, has to be connected to the EPU unit or to a module already connected to the EPU unit. To the LVDS output, a subsequent module can be interfaced.

5.1 DESCRIPTION OF PARAMETER SET

5.1.1 DEVICE IDENTIFIER AND GENERAL PARAMETES

In the column Attribute, the features modifiability and saving can be seen. When the first part is RO, the parameter is read- only; otherwise those statuses can be seen in which the parameter can be modified (P-PreOp, S-SafeOp, O-Op). When a letter B follows the forward slash, the parameter will be saved during modification.

Index	Name	Explanation	Data type	Attr.
1000:0	Device type	Device type, for SMI it is zero.	UINT32	RO/-
1001:0	Error register	Error register, for SMI it is always zero.	UINT8	RO/-
1008:0	Device name	Device name in text.	STRING	RO/-
1009:0	Hardware version	Device hardware version in text, e.g. “V1.0 (PCB)”	STRING	RO/-
100A:0	Software version	Device software version in text, together with application software version, communication interface version and bootloader version if there is bootloader in the device, e.g. “V4.1 (APP) V5.12 (CoE) V1.3 (BOOT)”	STRING	RO/-
1018:0	Identity	Device identifier	UINT8	RO/-
1018:1	Vendor ID	Manufacturer identifier, for NCT it is 0x1326	UINT32	RO/-
1018:2	Product code	Product code, for SMI2 it is 0x37	UINT32	RO/-
1018:3	Revision	Current device version, 0x3	UINT32	RO/-
1018:4	Serial number	Individual device serial number	UINT32	RO/-
10F0:0	Backup parameter handling	Description of the parameter and its subindexes can be found at the description of saving parameters	UINT8	RO/-
10F1:0	Error Settings	Description of the parameter and its subindexes can be found at the description of DC synchronous mode.	UINT8	RO/-

5.1.2 COMMUNICATION PARAMETES

Index	Name	Explanation	Data type	Attr.
1600:0	1st RxPDO Mapping - STEPPER	RxPDO (output) mapping belonging to the STEPPER mode	UINT8	RO/-
1600:1	Subindex 001	1. PDO Mapping entry (0x3001:01, SpeedSetValue1)	UINT32	RO/-
1600:2	Subindex 002	2. PDO Mapping entry (0x3001:02, ControlDword1)	UINT32	RO/-
1600:3	Subindex 003	3. PDO Mapping entry (0x3002:01, SpeedSetValue2)	UINT32	RO/-
1600:4	Subindex 004	4. PDO Mapping entry (0x3002:02, ControlDword2)	UINT32	RO/-
1600:5	Subindex 005	5. PDO Mapping entry (0x3000:02, Digital Outputs)	UINT32	RO/-

Index	Name	Explanation	Data type	Attr.
1601:0	2nd RxPDO Mapping - PWM	RxPDO (output) mapping belonging to the PWM mode	UINT8	RO/-
1601:1	Subindex 001	1. PDO Mapping entry (0x3003:01, PWM Control1)	UINT32	RO/-
1601:2	Subindex 002	2. PDO Mapping entry (0x3003:02, PWM Period1)	UINT32	RO/-
1601:3	Subindex 003	3. PDO Mapping entry (0x3003:03, PWM Duty1)	UINT32	RO/-
1601:4	Subindex 004	4. PDO Mapping entry (0x3004:01, PWM Control2)	UINT32	RO/-
1601:5	Subindex 005	5. PDO Mapping entry (0x3004:02, PWM Period2)	UINT32	RO/-
1601:6	Subindex 006	6. PDO Mapping entry (0x3004:03, PWM Duty2)	UINT32	RO/-

Index	Name	Explanation	Data type	Attr.
1602:0	3rd RxPDO Mapping - DIGOUT	RxPDO (output) mapping belonging to digital output	UINT8	RO/-
1602:1	Subindex 001	1. PDO Mapping entry (0x3000:02, Digital Outputs)	UINT32	RO/-

Index	Name	Explanation	Data type	Attr.
1A00:0	1st TxPDO Mapping - STEPPER	TxPDO (input) mapping belonging to the STEPPER mode	UINT8	RO/-
1A00:1	Subindex 001	1. PDO Mapping entry (0x3101:01, ActualPosition1)	UINT32	RO/-
1A00:2	Subindex 002	2. PDO Mapping entry (0x3101:02, StatusDword1)	UINT32	RO/-
1A00:3	Subindex 003	3. PDO Mapping entry (0x3101:03, ErrorDword1)	UINT32	RO/-
1A00:4	Subindex 004	4. PDO Mapping entry (0x3102:01, ActualPosition2)	UINT32	RO/-
1A00:5	Subindex 005	5. PDO Mapping entry (0x3102:02, StatusDword2)	UINT32	RO/-
1A00:6	Subindex 006	6. PDO Mapping entry (0x3102:03, ErrorDword2)	UINT32	RO/-
1A00:7	Subindex 007	7. PDO Mapping entry (0x3000:01, Digital Inputs)	UINT32	RO/-

Index	Name	Explanation	Data type	Attr.
1A01:0	2nd TxPDO Mapping - DIGIN	TxPDO (input) mapping belonging to digital input	UINT8	RO/-
1A01:1	Subindex 001	1. PDO Mapping entry (0x3000:01, Digital Inputs)	UINT32	RO/-

Index	Name	Explanation	Data type	Attr.
1C12:0	Syncmanager 2 Assignment	List of RxPDOs selected as process data	UINT8	P/-
1C12:1	Subindex 001	RxPDO selected as 1st process data (0x1600, 1st RxPDO Mapping - STEPPER)	UINT16	P/-
1C12:2	Subindex 002	RxPDO selected as 2nd process data (it does not have default value)	UINT16	P/-
1C13:0	Syncmanager 3 Assignment	List of TxPDOs selected as process data	UINT8	P/-
1C13:1	Subindex 001	TxPDO selected as 1st process data (0x1A00, 1st TxPDO Mapping - STEPPER)	UINT16	P/-
1C13:2	Subindex 002	TxPDO selected as 2nd process data (it does not have default value)	UINT16	P/-

5.1.3 STANDARD COMMUNICATION PARAMETER INFORMATION

Index	Name	Explanation	Data type	Attr.
1C00:0	Sync manager type	Standard parameter information on Sync manager types.	-	RO/-
1C32:0	SM output parameter	Standard parameter about SM features of the output process data	-	RO/-
1C33:0	SM input paraméter	Standard parameter about SM features of the input process data	-	RO/-

5.1.4 APPLICATION PROCESS DATA PARAMETERS

Index	Name	Explanation	Data type	Attr.
3000:0	Digital I/O	Parameter belonging to digital output and input.	UINT8	RO/-
3000:1	Digital Inputs	Parameter of digital inputs	UINT16	RO/-
3000:2	Digital Outputs	Parameter of digital outputs	UINT16	RO/-

Index	Name	Explanation	Data type	Attr.
3001:0	Axis 1 Output Variables	Output process data variables of the axis 1 of the stepper motor	UINT8	RO/-
3001:1	SpeedSetValue1	For detailed description of the variables see the description of the functions	UINT32	P/-
3001:2	ControlDword1		UINT32	P/-
3002:0	Axis 2 Output Variables	Output process data variables of the axis 2 of the stepper motor	UINT8	RO/-
3002:1	SpeedSetValue1	For detailed description of the variables see the description of the functions	UINT32	P/-
3002:2	ControlDword1		UINT32	P/-

Index	Name	Explanation	Data type	Attr.
3003:0	Axis 1 Output Variables	Process data variables of the 1st PWM output	UINT8	RO/-
3003:1	PWM Control1	For detailed description of the variables see the description of the functions	UINT16	P/-
3003:2	PWM Period1		UINT16	P/-
3003:3	PWM Duty1		UINT16	P/-
3004:0	Axis 2 Output Variables	Process data variables of the 2nd PWM output	UINT8	RO/-
3004:1	PWM Control2	For detailed description of the variables see the description of the functions	UINT16	P/-
3004:2	PWM Period2		UINT16	P/-
3004:3	PWM Duty2		UINT16	P/-

Index	Name	Explanation	Data type	Attr.
3101:0	Axis 1 Input Variables	Input process data variables of the axis 1 of the stepper motor	UINT8	RO/-
3101:1	ActualPosition1	For detailed description of the variables see the description of the functions	UINT64	RO/-
3101:2	StatusDword1		UINT32	RO/-
3101:3	ErrorDword1		UINT32	RO/-
3102:0	Axis 2 Input Variables	Input process data variables of the axis 2 of the stepper motor	UINT8	RO/-
3102:1	ActualPosition2	For detailed description of the variables see the description of the functions	UINT64	RO/-
3102:2	StatusDword2		UINT32	RO/-
3102:3	ErrorDword2		UINT32	RO/-

5.1.5 Application setup and information parameters

Index	Name	Explanation	Data type	Attr.
3203:0	PWM Settings Ch.1	PWM function settings.	UINT8	RO/-
3203:1	Timer resolution	The PWM timer resolution in ns	UINT16	RO/-
3203:0	PWM Settings Ch.2	PWM function settings.	UINT8	RO/-
3203:1	Timer resolution	The PWM timer resolution in ns	UINT16	RO/-

Index	Name	Explanation	Data type	Attr.
2000:0	Output mode	It displays the current operating mode (STEPPER or PWM).	ENUM	RO/-

Index	Name	Explanation	Data type	Attr.
3201:0	Encoder settings	Encoder input settings, applies to both channels.	UINT8	RO/-
3201:1	Encoder interface	Selection of a predefined encoder type.	ENUM	P/B
3201:x	-	Parameters on the 2-10 subindex: parameters are not in use.	UINT32	RO/-

5.2 SAVING THE PARAMETERS

The SMI2 settings are automatically saved in the case of a parameter modification, and there is no need to use a separate save command or startup list. Parameters can be saved by the parameter 0x10F0 Backup parameter handling. In the main software version 4, only the encoder type can be set and this is the only parameter that can be saved.

Index	Name	Explanation	Data type	Attr.
10F0:0	Backup parameter handling	It contains the parameters for saving device parameters	UINT8	RO/-
10F0:1	Checksum	It shows the CRC value belonging to the last saving.	UINT32	RO/-
10F0:2	BackupResult	After flash operations (saving and loading) have completed, the value of entry will be set that contains error code in case of error or, after successful operation, zero.	UINT16	PSO/-
10F0:3	BackupVersion	Version number of saving.	UINT16	RO/-

5.2.1 THE PROCESS OF SAVING

Modification of parameters will result in automatic saving; if an error occurs during saving,an error message (08000020h, Data cannot be transferred or stored to the application) will be received when parameter is being modified. Despite of the error message, rewritingthe parameter will be executed but without saving. More exact cause of the error can be read out from the parameter BackupResult in the form of a 16-bit error code. If saving completed successfully, the value of BackupResult will be zero. List of error codes:

Error code	Name	Explanation
0xEE10	UNKNOWN	Unknown error.
0xEE11	NOTERASED	The memory is not erased.
0xEE12	FLASHERASE	Memory erasing error.
0xEE13	PROTECTED	The memory is locked.
0xEE14	VERIFING	Test error after saving!
0xEE15	EXCEEDED	Too many or too long data!
0xEE16	HSIOSCENA	HSI is not enabled!
0xEE17	NORULE	Header and header data are not consistent.
0xEE18	UNKNWNHW	Unknown hardware!
0xEE19	NOMEMORY	There is no memory enough for saving.

Deleting an error by BackupResult will occur in the case of writing the BackupResult entry into zero through the CoE interface or executing successful EEPROM operation (i.e. by saving).

The function of saving during parameter modification can be disabled by writing value 0xFFFF in the parameter BackupResult. This is useful when the device is to be used from the startup list; in this case the parameter BackupResult has to be added to the startup list with the value 0xFFFF which deactivates the saving.

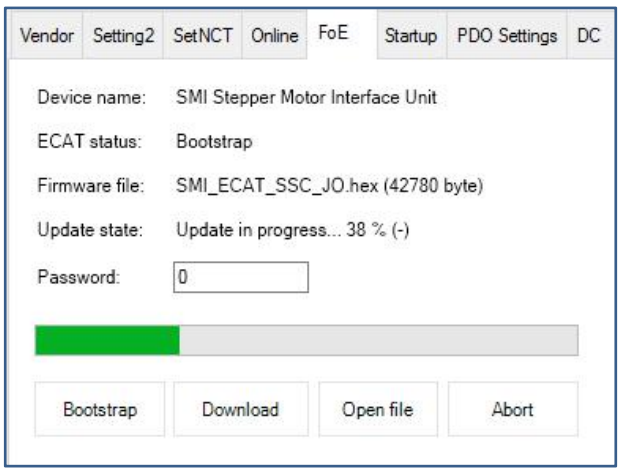
5.2.2 THE PROCESS OF LOADING

Saved parameters are read only once at power on, otherwise not. If any error occurs during the read-in, the device will refuse to change the status with the PreOp-> SafeOp state transition request with error code 0x51 ALStatus. The exact cause of the loading errors can be read from the BackupResult entry. The possible causes and codes of errors are the following:

Error code	Name	Explanation
0xEE01	EMPTY	The backup memory area is empty! No saving occurred yet.
0xEE02	CRC	CRC error!
0xEE03	VERSION	Unknown backup data version number!
0xEE04	NOMEMORY	More data is in the flash than was expected.
0xEE05	LENERROR	There is a difference between the byte lengths of the saved and to be written entry!

6 DOWNLOADING A SOFTWARE VIA ETHERCAT

The software running on the device can be replaced via EtherCAT. To do this, the device has to be stepped in bootsrap communication status and then the new software has to be downloaded via the FoE protocol.



On the NCT controller, stepping in the bootstrap status can be done in the EtherCAT window's on the Online tab using the status change buttons, or on the FoE tab using the Bootstrap button, and then the file containing the new software can be downloaded into the selected folder.

If the software change also involves a change of the device revision number, the EtherCAT EEPROM of the device must also be overwritten from the controller via the EEPROM tab of the EtherCAT window. In this case, the new EEPROM file can be found in the zip file containing the new software, as well as a changelog file that contains the information and changes made to the software. Software replacement steps in order are the following:

1. Copying the new xml (if required)
2. Downloading the software
3. Replacing the EEPROM content (if required)

If for some reason it is needed to replace the software with an earlier version, the steps have to be done in the same order. In the case of changing to a lower software version, the parameters saved in the device with the newer software may not be known to the old (replaced) software, so in such case the parameters has to backed up. Error codes can be occured during software replacement:

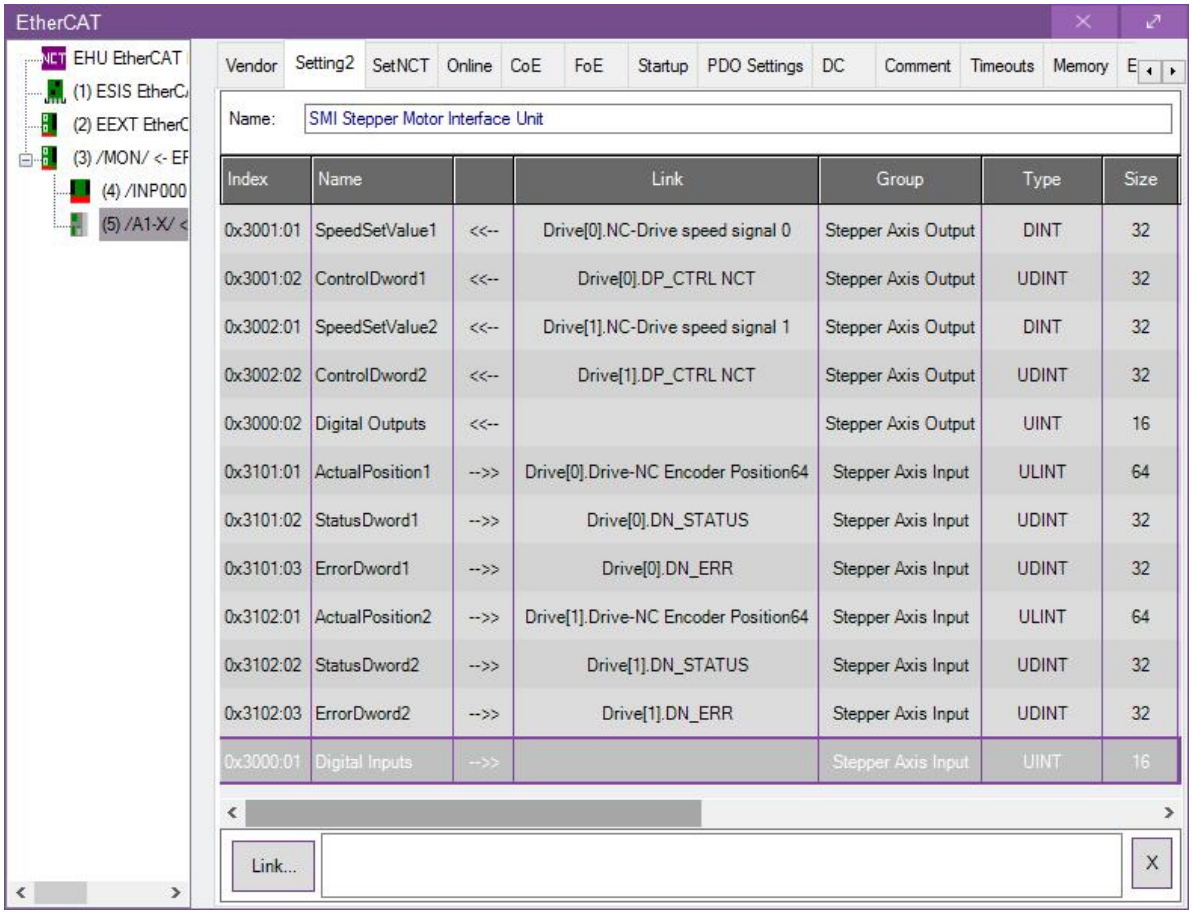
Error code	Error text	Explanation
0x8004	Invalid FoE service identifier!	Unknown FoE protocol identifier: only the WriteRequest and DataRequest identifiers are accepted.
0x8005	FoE packet number invalid!	Error in sequential continuity of the FoE packages. It is necessary to check whether there are lost packages in the network.
0x8010	Missing or invalid file header!	The header of the downloaded file is incorrect, the file format is unknown, or the file is damaged.
0x8012	Invalid file format!	The file format is unknown, the software file is not bootable or the file is damaged.
0xF0000001	Flash memory locked!	The flash is locked!
0xF0000002	Internal error: invalid FoE access state!	Invalid FoE status in case of test after FoE Response! Internal software error.
0xF0000003	Download is already running!	A new download has been initiated using command of request for write while another download is already being run.

0xF0000004	Not ready or WriteRequest missing!	Data package was received before initialization of the FoE data flow to be downloaded would have been occurred!
0xF0000005	Invalid master mailbox counter!	Invalid master mailbox counter! It is needed to enter Init status and to try to carry out downloading again. Check whether there are lost packages in the network.
0xF0000006	Allowed application memory address range exceeded!	The application indicates invalid memory domain! The file to be downloaded is incorrect or its format is unknown or it is not compatible with the device.
0xF0000007	Unknown flash writing error!	Unknown flash writing error!
0xF0000008	Flash erasing error!	Flash erasing error!
0xF0000009	Flash writing error: flash cell not erased!	Flash writing error, flash is not empty!
0xF000000A	Flash data verifying error!	Flash data verifying error!
0xF000000B	Flash address must be even!	Flash address has to be even! The file to be downloaded is incorrect or its format is unknown.
0xF000000C	Flash data block size must be even!	Flash data block size has to be even! The file to be downloaded is incorrect or its format is unknown.
0xF000000D	Wrong flash calculated CRC!	The CRC calculated on the basis of data written in the flash memory is incorrect. It is needed to enter Init status and to try to carry out downloading again.
0xF000000E	Flash error: Firmware valid marker writing error!	Error occurred when Firmware valid marker was being written in! It is needed to enter Init status and to try to carry out downloading again.
0xF000000F	Flash error: Bootmarker writing error!	Bootmarker writing error! It is needed to enter Init status and to try to carry out downloading again.
0xF0000010	Unexpected end of data stream: zero data length!	A zero-length FoE package was received unexpectedly! The file to be downloaded is incorrect or its format is unknown.
0xF0000011	Non-zero data length: end block has been received!	A block with zero data length has already been received, no more data is expected! The file to be downloaded is incorrect or its format is unknown.
0xE0000001	Unknown boot file version!	A newer boot file version at the unknown device or the file to be downloaded is incorrect or its format is unknown.

0xE0000002	Incompatible bootloader version!	A newer boot file version at the unknown device or the file to be downloaded is incorrect or its format is unknown.
0xE0000003	Incompatible hardware!	The target hardware version of the file and the device hardware version are not the same.
0xE0000004	Incompatible target product code!	The target product identifier and the device product identifier of the file are not the same; the software that can be downloaded does not belong to this device.

7 USE AS AXIS IN NCT CONTROLLER

To use the module as axis, the input and output variables of the device have to be assigned to the NC symbols in the following way:



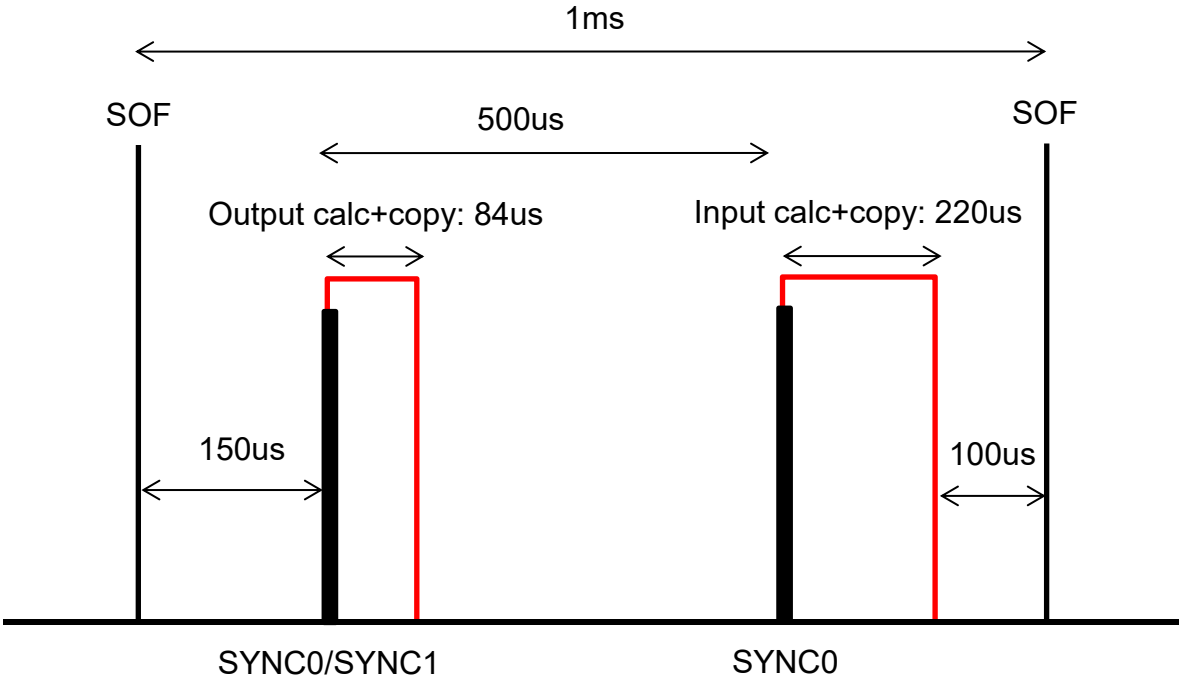
The picture above shows that the axis 1 of the module being STEPPER mode has been assigned to the drive 0 and its axis 2 to the drive 1; the digital input and output variables have not been assigned to PLC symbols.

On the SMI2 module, the scaling of the speed reference signal cannot be adjusted, so the linear speed belonging to the maximum control reference signal has to be specified in the parameter *N505 Position Calibration* of the controller in mm/min. If the drive, the motor and the ratio are known, the initial value of the parameter can be calculated.

For example: If the value of the control parameter *N502 Axis Output Type* is ECAT_NCT, the value of the maximum reference signal issued by the controller will be 2097152 (0x200000) to which 200 kHz output frequency belongs on the SMI2 module. If the number of (physical) steps of the stepper motor is 200 (α=1.8°) for one revolution and the microstep on the stepper motor drive is set to 64, 12800 pulses will be needed for one revolution. For stepper motor drives, it can be specified either the number of microsteps (12800) a revolution has to be divided into or the number of microsteps (64) a step has to be divided into. To 200 kHz frequency, 15.625 rpm speed belongs. If the pitch of the ballscrew is 5 mm, the linear speed belonging to the maximum reference signal will be 4687 mm/min. Then, the exact value of the parameter Position Calibration can be set on the controller by observing the variables *Command* and *ComVel*.

8 DC SYNCHRONOUS SCHEME

The device can be used in both SM synchronous and DC synchronous modes. In the case of the DC synchronous mode, the recommended EtherCAT cycle time is 1 ms, 2 ms and 5 ms.



9 SOFTWARE VERSIONS

Verzió	Date	Description, modifications
V4.1	2019.07.27	First version released.
V4.2	2020.08.31	Error corrected: when first switched on, a random value of 1 was found for the combined error bit (UnitErrors) on one, the other or both channels.
V4.3	2021.03.10	Handling the new encoder: Kübler M3663.G2222 “draw wire encoder”.

10 DOCUMENT REVISION

Revízió	Date	Description
V1.0	2019.07.22	First version released
V1.1	2019.11.27	- Handling the Wrn bit has been added to the Note of the Pepperl-Fuchs encoder. - Detailing the scaling of the reference signal of the stepper motor drive - Setting the parameters of the controller
V1.2	2020.01.29	- Correction of text errors - The device is renamed from SMI to SMI2.
V1.3	2021.03.10	- Correction of text errors - SMI description has been amplified with changes belonging to the version V4.3.
V1.4	2021.06.16	- Annex with a sample SSI communication, - Table containing the software versions
V1.5	2022.01.07	- Updated connection diagram for optocoupled input drives, - STP/DIR/PWM pull up voltage maximized to 5V.

ANNEX I. - EXAMPLE SSI COMMUNICATION

This is a record of an SMI2 position reading sample from a Kübler M3663.G2222 gray-coded encoder with SSI communication interface. In the picture, the signals DTA+ (data line) and CLK+ (clock signal) are shown; the idle state of both signals is high. The size of the SSI frame is 24 bits, there is no parity bit so the complete read data is position data. The read gray code data is 0x8B9D which, after decoding, corresponds to the position value 0xF2E9 (62185). At the and of the frame the monoflop time is shown that determines the minimum SSI frequency; the maximum SSI frequency is determined by the time between the leading edge and the appearance of the stable data on the data line.