PHASE
MOTION CONTROL

AxM II

CONFIGURABLE MOTION CONTROL PLATFORM

Parameters Reference
Firmware release 1.5
Doc. 10460-0-A-M - ENG

SUMMARY

REFERENCES.

/ 1: CiA DS301 V4.02
/ 2: CiA DSP305 V1.1
/ 3: CiA DSP402 V2.0
/ 4: Phase Motion Control AxM-II Configurable Motion Control Platform
/ 5: Phase Motion Control Cockpit II manual
/ 6: EtherCAT Communication Specification.
/ 7: LogicLab Enviroment User Manual Rev 1.0

FIGURES.

TABLES.

1. INTRODUCTION

The purpose of this document is to describe all AxM-II drive parameters and their features, divided by the different functionalities.

The system parameters are pre-defined parameters of the drive;

- used in every application;
- accessible from Cockpit configurator via the file *SysApp.par*;
- organized in hierarchical menus
- can be subdivided in read/write parameters (PAR) that could be saved in the drive, read-only variables (VAR) and working variables (WKS) used to control drive behaviour.

It's possible to modify the read/write parameters and then save them into the drive. Cockpit configurator informs when the modifications to some system parameters need a system reset to became active. Generally the parameters are structured in hierarchic sub-groups that collect all parameters related to different firmware functionalities.

The sub-groups are divided in logic blocks: there are some principal sections that collect the more commonly used parameters. The default values set by the manufacturer are appropriate in most cases: for particular applications it could be necessary to tune the parameters in order to optimize the drive performances. The user must be aware that the changes introduced assume a very deep knowledge of the drive firmware and functional control blocks implementation.
In several sections there is a "Monitor" sub-group that include all diagnostic parameters in order to allow the monitoring the AxM-II drive variables.

The complete AxM-II dictionary objects are listed here below. For each object there is a set of attributes, as follow:

Object	Mnemonic name used for parameter identification and for parameter access in the PLC user applications.
Modbus IPA:	Parameter index, as defined by the Modbus protocol.
Modbus Data Type:	Data type of the parameter. (Refer to Data type Encoding section).
CANOpen/COE Index:	This is the object index and sub-index of the parameter.
CANOpen/COE Data type:	CANOpen Data type of the parameter. (Refer to Data type Encoding section).
Attributes:	Read-only (ro), write-only (wo) or read-write (rw); could be limited to read-only depending on the state of the drive. Reset: specify if the change value of the object is affected to the reset command Retain: the object will be permanently stored in non-volatile memory Pdomap: PDO mappable: specify if the object could be mapped in a PDO
Unit	Measure unit of the object or if affected by the factor group (position, velocity and acceleration).
Default value	The value of the object has with the factory settings.
Value Range:	Value range of the parameters.

Each object is followed by a brief description for own features and characteristics.

1.1 Data type encoding

Basic data types used for accessing the object are:

- Signed8 (8 bit signed)
- Signed16(16 bit signed)
- Signed32 (32 bit signed)
- Unsigned8 (8 bit unsigned)
- Unsigned16 (16 bit unsigned)
- Unsigned32 (32 bit unsigned)
- Signed64 (64 bit signed)
- Real32: IEEE 754 floating point

The last part of the document deals AxM-II topics: the control loop schema, the faults meaning and the conversion unit for position, velocity and acceleration.

2. CORE

Object:	parPlcExeDisable	PLC program execution	
Modbus IPA:	13800	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5710h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw, reset, retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

These parameter let users choose if Plc code downloaded in the drive has to be executed at start-up or not. Such option is to be intended as debugging purpose, like having drive frozen by buggy Plc code.

Object:	parSysDriveMode	Drive mode configuration	
Modbus IPA:	22000	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5711h.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw, reset, retain		
Unit:	n/a	*Default value:*	45 (Motor Controller)
Value range:	45 (Motor Controller), 114 (Parallel Bridge Motor Control)		

Drive working mode. The 'Parallel Bridge' option is only available on double power bridge hardware and is useful to double power output. Refer to specific user manual for more information.

Object:	varSysBootErrorCode	Boot error code	
Modbus IPA:	18098	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5701h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro		
Unit:	n/a	*Default value:*	0000h
Value range:	See table above.		

System boot error code.
Please contact the technical support for all errors code, except errors code: 0003h, 0005h and 0018h that should be solve individually by the user.

Error code	*Description*
0001h	Invalid hardware configuration.
0002h	Invalid flash parameters.
0003h	Start with power failed. The auxiliary power supply voltage is under 24 Vdc voltage. AxM-II drive can't enable the power stage.
0004h	Watchdog task install failed.
0005h	Parameters error. Please check the parameters varSysWrongParCode. This parameter shows the IPA parameter value responsible for the wrong value configuration.
0010h	Fpga program failed.
0018h	Fpga program is incompatible with the DSP firmware. Try to load firmware in the drive, if error persist contact the technical support
0019h	Fpga test failed.
0100h + NT	Init task configuration failed: composed by default number (0100h) plus task number that failed own configuration.

Object:	varSysLifeTime	Life Time	
Modbus IPA:	19000	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	5209h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	sec	*Default value:*	n/a
Value range:	Full scale.		

Overall AxM-II drive life time in seconds unit from the first configuration done in Phase Motion Control production department.

Object:	varSysMaxRTExecTime	Real Time Task execution	
Modbus IPA:	19004	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5709h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro, pdomap		
Unit:	100 nsec	*Default value:*	n/a
Value range:	Full scale.		

This is the maximum value of real-time task execution time from the last reset command.
The task execution time is intend as the sum of cycle time machine and the Plc code time, that must not exceed an average of 110µsec, with allowed peak of 124µsec for a maximum of 4 consecutive time.
This parameters should help user to avoid the overtime error writing its own Plc code.

Object:	varSysWrongParCode	Wrong code parameter	
Modbus IPA:	19005	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5703h.h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	0 (no error detected)
Value range:	n/a		

This parameter show the wrong IPA value parameter code that has a wrong configuration/value.
Special error codes (only 3 number codes) are shown in table below.

Option code	*Description*
1545	Invalid sensorless back on the fly functionality (Refer to parameters parEncBEmf.BackOnTheFlySpd and parEncBEmf.SensorlessSpd).
1546	Invalid estimated sensorless speed constant (Refer to variable varEncBEmfSpeedKConversion).
1700	Invalid cooling threshold. (Refer to parameters parThermalModel.CoolingTempOn and parThermalModel.CoolingTempOff).

Object:	varSysAvgRTExecTime	Average task real time	
Modbus IPA:	19006	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	570Ah.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	100 nsec	*Default value:*	n/a
Value range:	Full scale.		

Average real-time task execution time from the last reset command.
Average is executed on 1024 samples.

Object:	varSysStat*	System Status	
Modbus IPA:	18089	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	5700.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	0
Value range:	See table above.		

This parameter shows the system status flags. The table below details all code.
When the user write its own PLC program the knowledge of the system status flags should be more useful.

Bit number	*Name*	*Description*
0	varSysStatBooting	Booting.
1	varSysStatResetting	Resetting.
2	varSysStatPowerSoftStarting	Power Ready: AxM-II is ready to execute operation.
3	varSysStatPowerZeroElFldSeek	Research electrical field orientation procedure is active.
4	varSysStatPowerEnabled	Power Enabled: voltage enabled on bridge output.
5	varSysStatFault	Fault.
6	varSysStatFaultReaction	Fault reaction.
7	varSysStatProgramFlashWriting	Program flash writing.
8	varSysStatParametersSaving	Parameters saving.
10	varSysStatLockedByBootError	Locked by boot error. Refer to varSysBootErrorCode parameter.
13	varSysStatFieldbusSyncing	When the Sync message is received (trough CANOpen or EtherCAT protocol) this bit is set for one real time cycle (125 µsec).
16	varSysStatPlcRunning	Plc running.
17	varSysStatPlcImgSlowIn	Slow task to copy image data input is running.
18	varSysStatPlcImgSlowOut	Slow task to copy image data output is running.
19	varSysStatPlcImgBackgroundIn	Background task to copy image data input is running.

20	varSysStatPlcImgBackgroundOut	Background task to copy image data output is running.
21	varSysStatAlarmsReset	Alarms reset: reset system alarm for one real time cycle.
24	varSysStatPlcRunSlow	Plc run slow task.
25	varSysStatPlcRunBackground	Plc run background task.

Object:	varSysWarnings	System Warnings.	
Modbus IPA:	18088	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	5702.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	0000 0000h
Value range:	See table above.		

This parameter shows the system active warnings flag. Each bit refers to a specific warning, as shown in the following table:

Bit mask	*Description*
0000 0001h	Locked by boot error.
0000 0004h	Access to forbidden area: write access memory not valid.
0000 0008h	Safe mode start-up requested. The AxM-II enters in this status by pressing simultaneously the command push buttons (Symbol Arrow IN) and (Symbol Arrow OUT) on the drive. In this status AxM-II has limited functionality and it's not possible enable the power stage. This status is useful since allow the user to startup the drive by ignoring all the parameters and the plc application stored into the drive.
0000 0010h	Start with power supply voltage under 24 Voltage.
0000 0020h	Undervoltage: warning if the power stage is disabled, otherwise fatal fault is generated if the power stage is enabled.
0000 0040h	Estimated sensorless module Kt value is out of range. (Refer to varEncBEmfValueKt).
0000 0080h	Estimated sensorless module speed constant is out of range. (Refer to varEncBEmfSpeedKConversion).
0000 0100h	Invalid parameters..
0000 0200h	Safe Torque Off (STO) active.
0000 1000h	Save and reset request pending.

3. MOTOR

These values describe the electrical and mechanical characteristics of the motor; usually they are found on the motor's nameplate or on the manufacturer's motor catalogue.
You can obtain the data sets for all Phase standard motors from the website: http://www.phase.eu

Object:	parMotorData.Resistance	Motor resistance.	
Modbus IPA:	27803	*Modbus Data Type:*	Real32
CANopen/COE index:	3F00h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,reset, retain		
Unit:	Ω	*Default value:*	~3.099
Value range:	0.001 to 32.767		

Motor winding resistance value.

Object:	parMotorData.Inductance	Motor inductance.	
Modbus IPA:	27804	*Modbus Data Type:*	Real32
CANopen/COE index:	3F01h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,reset, retain		
Unit:	H	*Default value:*	~0.01432
Value range:	0.00001 to 0.32767		

Motor winding inductance value.

Object:	parMotorData.KT	Motor torque constant.	
Modbus IPA:	27805	*Modbus Data Type:*	Real32
CANopen/COE index:	3F02h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,reset, retain		
Unit:	Nm/A	*Default value:*	1.5
Value range:	0.0001 to 10450		

Motor torque constant (Kt) value.

Object:	parMotorData.CurrentPeak	Motor current peak.	
Modbus IPA:	27808	*Modbus Data Type:*	Real32
CANopen/COE index:	3F03h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,reset, retain		
Unit:	Arms	*Default value:*	5.0
Value range:	0 to 214748.3647		

Motor current peak value.

Object:	parMotorData.SpeedNominal	Motor speed nominal.	
Modbus IPA:	27809	*Modbus Data Type:*	Real32
CANopen/COE index:	3F04h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,reset, retain		
Unit:	rad/sec	*Default value:*	0.0
Value range:	Full scale.		

Motor nominal speed.

Object:	parMotorData.PoleNumbers	Motor poles number.	
Modbus IPA:	27814	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3F06h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	8
Value range:	2 to Full scale, only even numbers		

Motor poles number.

Object:	parMotorData.DirectInductance	Direct-axis Inductance.	
Modbus IPA:	27818	*Modbus Data Type:*	Real32
CANopen/COE index:	3F07h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw, retain		
Unit:	H	*Default value:*	0.0
Value range:	Full scale		

The direct-axis current controls the flux of the motor and it's usually to set to zero with permanent magnet machine. This parameter sets the stator direct-axis inductance of the motor used during the field-weakening algorithm execution.
If this parameter is set to zero, the value of the direct-axis inductance is set equal to the motor winding inductance value.

4. ENCODER

The feedback source for the control loop is configured through these four flags.

Object:	parEncMgr.Flags.CntrlLoopPosMain	Feedback position for the control loop.	
Modbus IPA:	26003	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3803h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	1(Main)
Value range:	0(Aux)–1(Main)		

The position feedback to the control loop is caught or from the main encoder or from the auxiliary encoder.

Object:	parEncMgr.Flags.CntrlLoopSpeedMain	Feedback speed for the control loop.	
Modbus IPA:	26004	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3804h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	1(Main)
Value range:	0(Aux)–1(Main)		

The speed feedback to the control loop is caught or from the main encoder or from the auxiliary encoder.

Object:	parEncMgr.Flags.CntrlLoopAccelMain	Feedback acceleration for the control loop.	
Modbus IPA:	26005	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3805h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	1(Main)
Value range:	0(Aux) –1(Main)		

The acceleration feedback to the control loop is caught from the main encoder or from the auxiliary encoder.

Object:	parEncMgr.Flags.CntrlLoopElecAngleMain	Feedback Electrical Angle for the control loop.	
Modbus IPA:	26006	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3806h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	1(Main)
Value range:	0(Aux)–1(Main)		

The motor electrical angle used by current control loops is caught from the main encoder or from the auxiliary encoder.

4.1 Encoder > Main

Parameters to handle the Main Encoder features.

Object:	parEncMgr.PowerVoltage	Encoder Supply Voltage	
Modbus IPA:	26000	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3800h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset, retain		
Unit:	12.5mV	*Default value:*	416 (5.2V)
Value range:	0 to 1024 (0V to 8V)		

Encoder supply voltage :

- for Endat should be set according to the encoder requested voltage (usually 5.2V);
- for Sincos or Digital normally it is 5.2V;
- for other kind please refer to the encoder manufacturer datasheet.

Object:	parEncMgr.MainAbselection	Absolute encoder type selection.	
Modbus IPA:	26001	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3801h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	0004h
Value range:	See table above.		

Selection of the absolute tracks for the main encoder:

Option code	*Description*	*Name*
0000h	No sensor.	Null
0001h	Hall sensor.	cstEncTypeAbsHall
0003h	Analog sin/cos encoder.	cstEncTypeAbsAnalog
0004h	Absolute encoder Endat (Heidenhain encoder).	cstEncTypeAbsEndat

Absolute encoder.

The distinguishing feature of the absolute encoder is that it reports the absolute position of the encoder to the electronics immediately upon power-up with no need for indexing.

In single-turn encoders the absolute position information repeats itself with every revolution. Multiturn encoders can also distinguish between revolutions.

The Endat interface from Heidenhain is a digital, bidirectional interface for encoders: it is capable both of transmitting position values from incremental and absolute encoders as well as transmitting or updating information stored in the encoder, or saving new information. The data are transmitted in synchronism with the clock signal from the subsequent electronics.

Hall effect devices are digital On/Off sensors constructed of semiconductor material used to sense the presence of magnetic fields. In brushless servo motors, the halls are usually imbedded within the motor windings and sense the position of the rotor magnets. There is one sensor for each motor phase, aligned with the stator winding.

Sin/cos encoder, also called sinusoidal encoder, supply two sinusoidal signals, offset by 90°. The number and progress of the sine waves (interpolation and arctangent) is evaluated. Using these values, the speed and position can be determined with a very high resolution.

Usually sin/cos encoder has 2 tracks and one index signal track. Inverting the signals results in a total of six tracks. The 90° offset signal are on tracks A and B. One sine half-wawe per revolution is provided at channel track C. The track A,B,C are inverted in the encoder and provided inverted as signals on tracks A, B and C.

Object:	parEncMgr.Flags.DisableIfRelFail	Relative position is not valid.	
Modbus IPA:	26007	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3807h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw, retain		
Unit:	--	*Default value:*	0(False)
Value range:	0(False)-1(True)		

Immeditately disable the drive if data from relative position encoder are not valid, otherwise the absolute track is used to try to quick stop the motor in a controlled way.

Object:	parEncMgr.PowerStartUpDelay	Delay start-up encoder.	
Modbus IPA:	26008	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3808h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset, retain		
Unit:	msec	*Default value:*	300
Value range:	1 to 10000		

Delay in msec to startup the encoder (wait for the steady power before beginning position processing).

Object:	parEncMgr.MainRelSelection	Incremental encoder type.	
Modbus IPA:	26009	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3809h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset, retain		
Unit:	--	*Default value:*	0000h
Value range:	See table above.		

Selection of the incremental tracks used by the main encoder:

Option code	*Description*	*Name*
0000h	No sensor.	cstEncTypeNull
0005h	Sensorless BackEmf. (Refer to section (§ 4.2.7 Sensorless).	cstEncTypeRelBackEMF
0006h	Digital Incremental encoder.	cstEncTypeRelIncremental

Digital Incremental(relative) encoder.

An incremental rotary encoder, also known as a quadrature encoder or a relative rotary encoder, has two outputs called quadrature outputs. Incremental encoder works differently by providing an A and a B pulse output that provide no usable count information in their own right.
To provide useful position information, the encoder position must be referenced to the device to which it is attached, generally using an index pulse. The distinguishing feature of the incremental encoder is that it reports an incremental change in position of the encoder to the counting electronics.
A variation on the Incremental encoder is the Sinewave Encoder: instead of producing two quadrature square waves, the outputs are quadrature sine waves (a Sine and a Cosine).
Other model of relative encoder is the sensorless control algorithm. Advanced sensorless algorithms use the back-EMF generated in the stator winding to determine the rotor position.
For more details on sensorless algorithm refer to section (§ 4.2.7 Sensorless).

Object:	parEncMgr.MaxAbsRelDiff	Difference between positions.	
Modbus IPA:	26100	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	380Ah.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, retain, pdomap		
Unit:	Position factor unit.	*Default value:*	0
Value range:	0 to 32768		

Max absolute difference admitted between calculated absolute position and incremental position for redundancy control.
If differences goes above this threshold then drive generate fault status with Main Encoder Abs/Rel Track Check Failure alarm.
If the value is set to zero the checking control is disabled.

Object:	parEncMgr.Flags.DisableEPlate	Disable Endat Plate.	
Modbus IPA:	26101	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	380Bh.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0 (Off)
Value range:	0(Off)-1(On)		

Flag to disable the reading datas from the electronic plate (only for Endat encoders).
During the start-up procedure, if the electronic plate is available the firmware reads the electronic plate values and overwrites the user parameters parMotorData. The user has the possibility to change them later.
At following start-up if the firmware finds the same electronic plate does not overwrite the parMotorData; but it find another electronic plate, means that the user has changed the motor, and then the parMotorData are overwritten. With the below parameter RestoreEPlate user could force (reset) rewriting parMotorData with the right electronic plate, in case of wrong parameters are written.

Object:	parEncMgr.Flags.RestoreEPlate	Restore Endat Plate.	
Modbus IPA:	26102	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	380Ch.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off) -1(On)		

Flag to restore the data from electronic plate (only for Endat encoders).
This operation permitts to the firmware to restore the data values available on the electronic plate.

Object:	parMotorData.PhaseOffset	Phasing value correction.	
Modbus IPA:	27812	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3F05h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	0° 0' 19.78" electrical angle	*Default value:*	0
Value range:	0 to 65535 (0° to 359° 59' 59" electrical angle)		

This parameter indicates the electrical offset angle between the position feedback encoder and the motor stator windings. In Phase motors manufacturer this value is equal to zero.

4.1.1 Endat

The EnDat interface from Heidenhain is a digital, bidirectional interface for encoders. It is capable both of transmitting position values from incremental and absolute encoders as well as transmitting or updating information stored in the encoder, or saving new information. Thanks to the serial transmission method, only four signal lines are required. The data are transmitted in synchronism with the clock signal from the subsequent electronics. The type of transmission (position values, parameters, diagnostics, etc.) is selected by mode commands that the subsequent electronics send to the encoder.
Parameters to handle the Endat protocol information.

Object:	parEncMEndat.ClockFreq	Endat Clock Frequency.	
Modbus IPA:	26020	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3810h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	kHz	*Default value:*	2000
Value range:	100 – max value depending by Endat protocol type.		

Clock frequency selection for Endat protocol.

- Max Value Endat 2.1: 2000 kHz
- Max Value Endat 2.2: 8000 kHz

Object:	parEncMEndat.MTurnStartPos	Set point position.	
Modbus IPA:	26023	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3811h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,reset,retain		
Unit:	Number of turns.	*Default value:*	4294967295
Value range:	0 to 4294967295		

Set point position at start-up condition for multiturn Endat encoder.
This parameter avoid numerical issues with startup absolute multiturn position in which the bit resolution of the turns fraction of the position (typically 12bit) is less than 32bit (the maximum that drive can handle): properly setup of this parameter let user choose the threshold by which startup position (e.g. at reset) is considered positive or negative.During the initiliazation procedure, if the revolution number reached by the multiturn encoder is greater then this value, the absolute position offset became a negative number as show figure below.

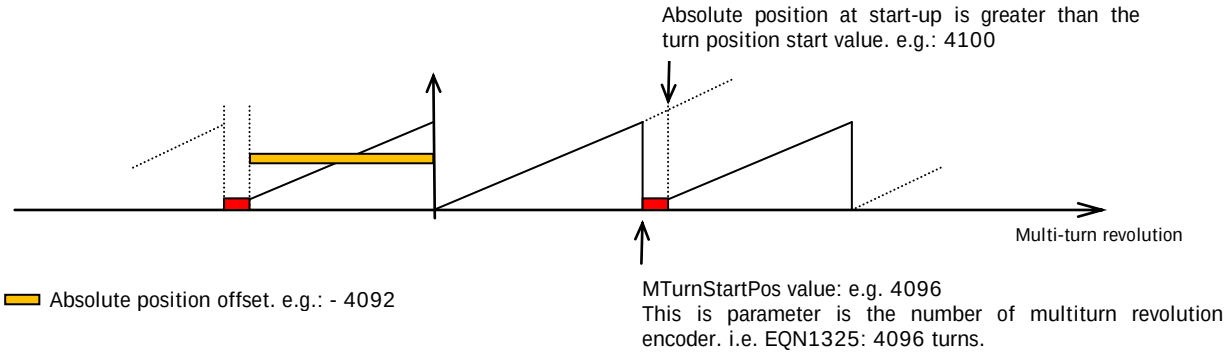

Figure 1 - Multi turn encoder: set point position.

4.1.2 Endat > Monitor

Data related to the Endat Main Encoder status.

Object:	varEncMEndatCrcErrors	CRC errors counter.	
Modbus IPA:	26021	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3818h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	--	*Default value:*	n/a
Value range:	n/a		

Endat protocol checksum value errors counter, could be used as electrical connection diagnostic checking.

Object:	varEncMEndatPropDelay	Propagation delay.	
Modbus IPA:	26022	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3819h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	nsec	*Default value:*	n/a
Value range:	n/a		

Data propagation delay: the value is calculated during Endat initialization procedure.
For more information refer to manual "Endat: VERSION 2.2 Bidirectional Synchronous-Serial Interface for Position Encoders " from Heidenhain Gmbh.

4.1.3 Absolute Analogue Traces

These parameters are referred to main type of absolute encoder: analogue sin/cos encoder or resolver type.
Resolvers are absolute analog rotary encoders which are ideal for harsh environments. Unlike the conventional optical rotary encoder (with temperature-limiting internal optoelectronics), the resolver is similar to a transformer in construction, passive and does not incorporate any electronics.
Parameters to set-up the encoder absolute analogues features.

Object:	parEncAn.PoleCounts	Poles number per revolution.	
Modbus IPA:	26052	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3840h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	--	*Default value:*	2
Value range:	2 to full scale; only even number.		

Encoder poles number for revolution setting.

Object:	parEncAn.Flags.ReverseSignal	Reverse count signal.	
Modbus IPA:	26053	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3841h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	--	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Reverse the count of the encoder on the same direction of motor rotation.

- 1(On): Angle = 360° - arctan(Sin/Cos).
- 0(Off): Angle = arctan(Sin/Cos).

Object:	parEncAn.Flags.AnalogGain	Analog gain.	
Modbus IPA:	26054	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3842h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	--	*Default value:*	1(High)
Value range:	0(Low)-1(High)		

This parameter setup hardware analog gains depending by the encoder type:

- 1(High) for Sincos encoder.
- 0(Low) for Resolver.

Object:	parEncAn.AnalogAlarmThreshold	Analog Alarm Threshold.	
Modbus IPA:	26057	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3843h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	Internal device unit (d.u.)	*Default value:*	5000
Value range:	Full scale.		

Analog tracks level (sin²+cos²) threshold.
The calculated value sin² + cos² must be greater than this value, otherwise the alarm Absolute Analog Encoder Low Tracks Level is generated. The signals of the single tracks will not be monitored to their amplitudes separately. But from the recalculation of the encoder radius the outcome is that the signal level has to greater to the analog signal threshold. The sin² + cos² greater to the threshold means that the amplitude of signal sine and cosine are correct; this represents the health of the encoder.If the value is set to zero then amplitude checking is disabled.

4.1.4 Absolute Analogue Traces > Monitor

Status of the encoder absolute analogue traces.

Object:	varEncAnChannelSin	Analog Sine channel.	
Modbus IPA:	26055	*Modbus Data Type:*	Signed16
CANopen/COE index:	3848h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Analog level of the encoder Sin channel.

Object:	varEncAnChannelCos	Analog Cosine channel.	
Modbus IPA:	26056	*Modbus Data Type:*	Signed16
CANopen/COE index:	3849h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale		

Analog level of the encoder Cos channel.

Object:	varEncAnChannelLevel	Analog channel level.	
Modbus IPA:	26058	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	384Ah.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Calculated level of (sin²+cos²).

4.1.5 Hall sensor Traces

Parameters to set-up the feedback system based on Hall effect switches.
Rotor position sensors are directly mounted on the motor stator and detect the rotor magnetic field generating a voltage signal proportional to the field intensity. The voltage signal from the Hall sensors has a periodicity in a mechanical rotation that depends on the number of pole pairs and so the mechanical resolution of the single sensor improves with the number of polar pairs while the electric resolution does not vary. The three Hall sensors are separated by 120° phase angles and triggered by the rotor magnet. They produce a switching pattern depicted with a new digital state every 60°.

Object:	parEncHall.PoleCounts	Encoder poles number.	
Modbus IPA:	26042	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3870h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	--	*Default value:*	2
Value range:	2 to full scale; only even number.		

Encoder poles number for revolution setting.
If this value is set to zero the internal software value of the encoder poles number became equals to motor poles number.

Object:	parEncHall.Flags.Enable4Wire	Enable 4 wire.	
Modbus IPA:	26043	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3871h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(0ff)-1(On)		

The 4-Wire 'Kelvin' method is used in cases when very high accuracy is required. The method is immune to the influence of lead resistance and is limited by the quality of the constant current source and voltage measurement.

- 0(Off): the 4-Wire method is disabled.
- 1(On): the 4-Wire method is enabled.

4.1.6 Hall sensor Traces > Monitor

Status of the Hall sensor traces.

Object:	varEncHallChannelSin	Analog level Sin channel	
Modbus IPA:	26044	*Modbus Data Type:*	Signed16
CANopen/COE index:	3872h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Analog level of the encoder Sin channel.

Object:	varEncHallChannelCos	Analog level Cos channel.	
Modbus IPA:	26045	*Modbus Data Type:*	Signed16
CANopen/COE index:	3873h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Analog level of the encoder Cos channel.

4.1.7 Incremental Traces

With the incremental measuring method, the graduation consists of a periodic grating structure. The position information is obtained by counting the individual increments (measuring steps) from some point of origin. Since an absolute reference is required to ascertain positions, the graduated disks are provided with an additional track that bears a reference mark.
The incremental signals are transmitted as the square-wave pulse trains U1 and U2, phase-shifted by 90° elec. The reference mark signal consists of one or more reference pulses U0, which are gated with the incremental signals.
Parameters to set-up the encoder incremental features and behaviour.

Object:	parEncMInc.LineCounts	Encoder Line Counts.	
Modbus IPA:	26030	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3820h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,reset,retain		
Unit:	Counts	*Default value:*	512
Value range:	Full scale.		

Encoder pulses number for revolution setting.
The quadrature encoder pulse (QEP) module is used for direct interface with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine for use in a high-performance motion and position-control system. Furthermore, the frequency of the clock generated by QEP

circuits is four times that of the frequency of each QEP input channel, because both the rising and falling edges of both QEP input channels are counted by the selected timer. When used for position the quadrature encoder gives a four-fold increase in resolution, in default case, 2048 counts per revolution.
For the Direction-count mode and for Down-Count Mode the encoder line counts is the same value set by the line number counts (1X mode).

Object:	parEncMInc.Flags.EnableAnalogInterp	Analog Interpolation.	
Modbus IPA:	26033	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3821h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Enable the analog sin/cos interpolation, used to select between analog incremental traces and digital incremental traces:

- 1(On): interpolation is activate, digital counts plus analog interpolation.
- 0(Off): interpolation is deactivate, only used digital counts.

Object:	parEncMInc.Flags.DisableIndexError	Disable Index Error.	
Modbus IPA:	26034	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3822h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Flag to disable the error generated by the index track checking

Object:	parEncMInc.Flags.DisableAnalogError	Disable Analog Error.	
Modbus IPA:	26035	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3823h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Flag to disable the error generated by the analog tracks levels.

Object:	parEncMInc.AnalogAlarmThreshold	Analog Signal Threshold.	
Modbus IPA:	26036	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3824h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	Internal device unit (d.u.)	*Default value:*	5000
Value range:	Full scale.		

Analog tracks level ($\sin^2+\cos^2$) threshold.
The calculated value $\sin^2 + \cos^2$ must be greater then this value, otherwise an alarm Main Incremental Encoder Low Tracks Level is generated. The signals of the single tracks will not be monitored to their amplitudes separately. But from the recalculation of the encoder radius the outcome is that the signal level has to greater to the analog signal threshold. The $\sin^2 + \cos^2$ greater to the threshold means that the amplitude of signal sine and cosine are correct; this represents the health of the encoder. If the value is set to zero the checking control is disabled.

Object:	parEncMInc.IndexErrorTolerance	Index Error Tolerance.	
Modbus IPA:	26037	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3825h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	count	*Default value:*	10
Value range:	0 to parEncMInc.LineCounts.		

Tolerance of the difference between the calcolated zero position and the index track used for index track checking.

Object:	parEncMInc.FlagsEnableIndexTrack	Enable Index Track.	
Modbus IPA:	26038	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3826h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Flag to enable index track.

Object:	parEncMInc.Flags.EnableStepDir	Enable Step Dir mode.	
Modbus IPA:	26050	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3829h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Enable step pulse (track A) and direction (track B) mode in place of standard quadrature encoder mode. Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. This mode is sometimes called in a 1X mode (a 512 line encoder will generate 512 counts per revolution).

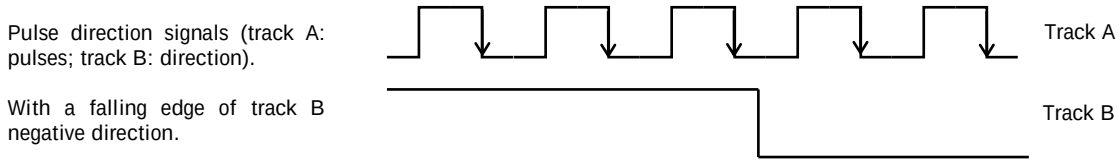

Figure 2 - Direction count-mode

Track A input will provide the clock for position counter and the track B input will have the direction information. The position counter is incremented on every rising edge of a track A input when the direction input is high and decremented when the direction input is low.

Object:	parEncMInc.Flags.EnableUpDown	Enable Up Down mode.	
Modbus IPA:	26051	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3828h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Enable up pulse (track A) and down pulse (track B) mode in place of standard quadrature encoder mode.
This is sometimes called in a 1X mode (a 512 line encoder will generate 512 counts per revolution in up/down mode).
When the Up/Down mode is selected, two input pulse signals determine the count value.

- With track B high, the count increments on the rising edge of track A.
- With track A high, the count decrements on the rising edge of track B.

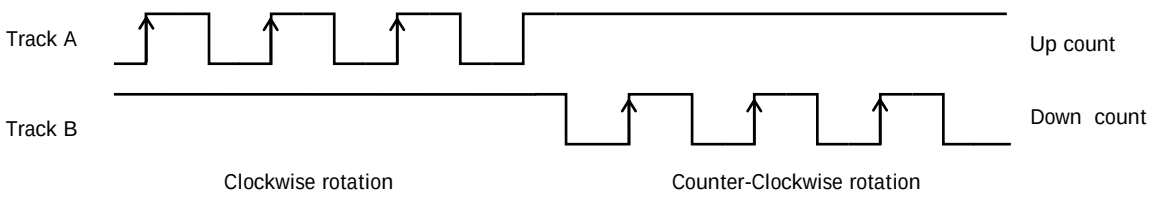

Figure 3 - Up/down count-mode

Object:	parEncMInc.Flags.SwapTracks	Swap input tracks.	
Modbus IPA:	26049	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3827h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Swap physical input tracks A and B to adjust wrong electrical connections.

4.1.8 Incremental Traces > Monitor

Status of the incremental encoder.

Object:	varEncIncChannelSin	Analog level Sin channel	
Modbus IPA:	26039	*Modbus Data Type:*	Signed16
CANopen/COE index:	3830h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Analog level of the encoder Sin channel.

Object:	varEncIncChannelCos	Analog level Cos channel.	
Modbus IPA:	26040	*Modbus Data Type:*	Signed16
CANopen/COE index:	3831h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Analog level of the encoder Cos channel.

Object:	varEncIncChannelLevels	Analog level channel.	
Modbus IPA:	26041	*Modbus Data Type:*	Signed16
CANopen/COE index:	3832h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	Internal device unit (d.u.)	*Default value:*	n/a
Value range:	Full scale.		

Calculated level of ($\sin^2+\cos^2$).
For more details refer to parEncMInc.AnalogAlarmThreshold parameter.

4.1.9 Sensorless

This algorithm is based on the estimation of rotor speed and angular position starting from the back electromotive force space vector determination without voltage sensors by using the reference voltages given by the current controllers instead of the actual ones.

WARNING: Only expert users have to use these parameters. These parameters should used to set up the sensorless algorithm or should be useful to check the system during start-up operation. Be careful as modifying the values of this object with power enabled could yield in a loss of axle control.

Parameters to handle sensorless control algorithm.

Object:	parEncBEmf.Flags.OpenLoopOnly	Open loop only.	
Modbus IPA:	26066	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3850h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

With this flag the user controls the drive in *Open Loop* without any contributes of the estimated back electromotive force value. Acceleration, speed and position of *Open loop* modality are calculated using reference speed as applied speed without any feedback information.

Object:	parEncBEmf.Flags.DynamicIqLimit	Iq Limit.	
Modbus IPA:	26067	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3851h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

- 1(On): dynamic limit applied to the quadrature current during the sensorless start-up operations.
- 0(Off): limit applied to the quadrature current depending only by the drive limits and by the Sensorless Algorithm Status.

Object:	parEncBEmf.Flags.DynamicIdLimit	Id Limit.	
Modbus IPA:	26068	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3852h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

- 1(On): dynamic limit applied to the direct current during the sensorless start-up operations.
- 0(Off): limit applied to the quadrature current depending only by the drive limits and by the Sensorless Algorithm Status.

Object:	parEncBEmf.BackOnTheFlySpd	Back on the fly speed.	
Modbus IPA:	26069	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3853h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	0.0001	*Default value:*	9000
Value range:	[0%:100%=0:10000].		

Defines the speed, expressed percentage of the nominal motor speed (parMotorData.SpeedNominal), used to try to enable the motor control while it still turning. This value has to be greater than the sum of the sensorless speed (parEncBEmf.SensorlessSpd) and of a delta value (about 10%); otherwise a Sensorless alarm is generated.

$$\text{parEncBEmf.BackOnThe FlySpd} \geq \text{parEncBEmf.SensorlessSpd} + 10\%$$

Object:	parEncBEmf.SensorlessSpd	Sensorless speed.	
Modbus IPA:	26070	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3854h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	0.0001	*Default value:*	2000
Value range:	[0%:100%=0:10000].		

Sensorless speed: percentage of the nominal motor speed (parMotorData.SpeedNominal).
This is the speed that the motor must reach in order to get the sensorless algorithm fully working.

Object:	parEncBEmf.Threshold1Spd	Threshold 1 limit.	
Modbus IPA:	26071	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3855h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	0.0001	*Default value:*	500
Value range:	[0%:100%=0:10000].		

Threshold 1 speed: percentage of the nominal motor speed (parMotorData.SpeedNominal). Used only during the sensorless initialization procedure.
This value has to be lower of parEncBEmf.SensorlessSpd value; otherwise an alarm Parameter value out of range is generated.

Object:	parEncBEmf.Threshold0Spd	Threshold 0 limit.	
Modbus IPA:	26072	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3856h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	0.0001	*Default value:*	250
Value range:	[0%:100%=0:10000].		

Threshold 0 speed: percentage of the nominal motor speed (parMotorData.SpeedNominal). Used only during the sensorless initialization procedure.
This value has to be lower of parEncBEmf.Threshold1Spd1 value; otherwise an alarm Parameter value out of range is generated.

Object:	parEncBEmf.AGlitchSpd	Anti-glitch filter.	
Modbus IPA:	26073	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3857h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	0.0001	*Default value:*	250
Value range:	[0%:100%=0:10000].		

Threshold to activate an anti-glitch filter. The filter is applied to the angle value calculated with the sensorless algorithm. The filter is useful to decrease the incidental noise from the reading system.

Object:	parEncBEmf.AGlitchFaultLimit	Anti-glitch fault limit.	
Modbus IPA:	26074	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3858h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Maximum sampling number at 125 µsec with anti-glitch filter active. When the actual counter of the anti-glitch filter is over this threshold otherwise a Sensorless alarm is generated (antiglitch filter always active).
If this parameter is zero the error checking control is disabled.

Object:	parEncBEmf.Flags.EnabledAGlitchFilt	Use anti-glitch filter.	
Modbus IPA:	26075	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3859h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	1(On)
Value range:	0(Off)-1(On)		

Parameter to enable the use of anti-glitch filter during the sensorless algorithm.

Object:	parEncBEmf.Flags.EnabledDeltaAngle	Use speed delta angle.	
Modbus IPA:	26076	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	385Ah.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

This parameter works only in full sensorless status. It's possible to use the delta angle position as feedback instead of Back EMF algorithm speed calculated. The speed delta angle is calculated us difference of mechanical angle, still every 125 µsec.

Object:	parEncBEmf.DisableAlarmMask	Disable alarm.	
Modbus IPA:	26077	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	385Bh.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0000 0000h
Value range:	0000 0000h - 0000 0001h		

Mask to disable alarms or warnings of sensorless algorithm.
Currently only one bit is available. This bit is a warning system and it informs the user that the drive is in *Open Loop* mode without any contributes of the estimated back electromotive force value.
For more details refer to varSysWarnings.

Object:	parEncBEmf.KtRefreshPeriod	Kt refresh period.	
Modbus IPA:	26078	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	385Ch.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	15
Value range:	1 to 32767.		

Time to refresh motor constant torque estimation value.

Object:	parEncBEmf.IdOpenLoop	Id value in open loop.	
Modbus IPA:	26079	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	385Dh.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	0.001	*Default value:*	100
Value range:	[0:1000=0:parMotorData.CurrentPeak]		

Max direct current value used in open loop mode. Value is a percentage of the motor maximum current peak.
To avoid current discontinuities at start-up, this value should be equal to parEncEFS.IdRampCurrent.

4.1.10 Sensorless > Monitor

Sensorless variables and algorithm status.

Object:	varEncBEmfState	Sensorless status	
Modbus IPA:	26083	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3861h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	n/a	*Default value:*	0000h
Value range:	See table above.		

Sensorless algorithm status.

Option code	*Description*
0000h	Sensorless disabled.
0001 – 0003h	Sensorless initialization procedure.
0004h	Sensorless full. The sensorless algorithm works at full capacity.

Object:	varEncBEmfAglitchCounter	Anti glitch counter.	
Modbus IPA:	26084	*Modbus Data Type:*	Signed16
CANopen/COE index:	3862h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		
Unit:	nr	*Default value:*	0
Value range:	0 to 32768.		

Actual counter of the anti-glitch filter.

Object:	varEncBEmfSpeedKConversion	Speed constant conversion.	
Modbus IPA:	26085	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3863h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	0
Value range:	0 to 65536.		

Actual speed constant conversion value calculated by the sensorless algorithm.
If the value is outside the value range (0 to 65536), a Sensorless alarm is generated.

Object:	varEncBEmfSpeedDelta	BEmf speed angle.	
Modbus IPA:	26086	*Modbus Data Type:*	Signed32
CANopen/COE index:	3864h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	rad/s	*Default value:*	0
Value range:	Full scale.		

Mechanical speed calculated as difference of mechanical angles.

Object:	varEncBEmfValueKt	Motor Kt value.	
Modbus IPA:	26087	*Modbus Data Type:*	Real32
CANopen/COE index:	3865h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	ro,pdomap		
Unit:	Nm/A	*Default value:*	0.0
Value range:	Full scale.		

Estimated motor torque constant (Kt) value.

4.1.11 Electrical Field Orientation

Parameters to manage the Electrical Field Orientation procedure.

Object:	parEncEFS.ProcedureType	Procedure type.	
Modbus IPA:	26302	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	38F0h.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw,reset,retain		
Unit:	--	*Default value:*	0(Disabled)
Value range:	0 (Disabled) – 1(IdRamp) – 2 (IdRamp with Position control)		

Procedure type:

- 0 (Disabled): the procedure is disabled.
- 1 (IdRamp): the procedure start with a ramp applied on the direct current. The ramp time and the direct current applied are both parameters. At the end of the ramp the current will reach the value set on parEncEFS.IdRampCurrent parameter.
- 2 (IdRamp with Position Control): the procedure starts performing position short movements in both directions with the final purpose to search the home position (zero position). For more details refer to (§4).

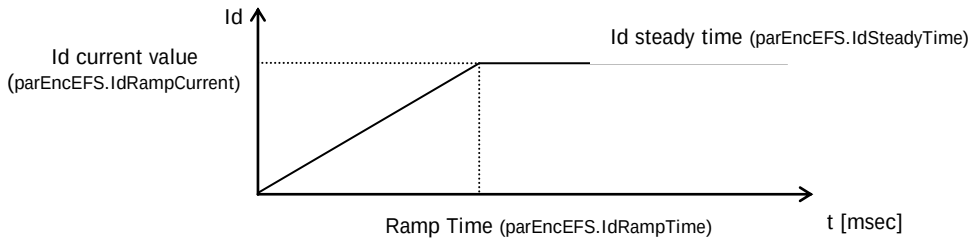

Figure 4 - Electrical field operation: IdRamp

Object:	parEncEFS.Flags.Force	Force procedure flag.	
Modbus IPA:	26303	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	38F1h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Flag to enable the electrical field orientation procedure.

Object:	parEncEFS.SpeedThreshold	Speed threshold.	
Modbus IPA:	26304	*Modbus Data Type:*	Signed32
CANopen/COE index:	38F2h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	Position factor unit.	*Default value:*	447375 (~5.24 rad/s)
Value range:	Full scale		

Before to start the procedure wait for speed goes below this threshold value.

Object:	parEncEFS.IdRampTime	Direct current ramp time.	
Modbus IPA:	26305	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	38F3h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	msec	*Default value:*	500
Value range:	Full scale		

Time ramp procedure to direct the motor rotor.

Object:	parEncEFS.IdRampCurrent	Direct current value.	
Modbus IPA:	26306	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	38F4h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	0.001	*Default value:*	300
Value range:	[0%:100%=0:parMotorData.CurrentPeak]		

Current to direct the motor rotor (percentage of motor peak current).

Object:	parEncEFS.ElecAngleFeed	Electrical Angle Feed.	
Modbus IPA:	26308	*Modbus Data Type:*	Signed32
CANopen/COE index:	38F9h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.000015	*Default value:*	1000
Value range:	[0%:100%=0:65535]		

Electrical angle feed ratio from space control loop output. This parameter refers to the procedute type IdRamp with Position Control. For more details refer to (§4).

Object:	parEncEFS.IdSteadyTime	Id Steady Time.	
Modbus IPA:	26309	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	38FAh.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	msec	*Default value:*	0
Value range:	Full scale.		

This parameter indicates the time to keep direct current steady after ramp procedure phase.

4.1.12 Monitor

Variables related to the position of the Main Encoder.

Object:	varEncMainMechHi	Feedback turns.	
Modbus IPA:	26010	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A00h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale		

This is the relative position value in mechanical turns units.
The feedback turns is the MSB 32 bit of 64 bit main encoder position value.
For more details about the format position refer to the appendix Encoder position format or to the manual (§4).

Object:	varEncMainMechLo	Feedback angle.	
Modbus IPA:	26011	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A00h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the relative position value in mechanical angle unit.
The feedback angle is the LSB 32 bit of 64 bit main encoder position value.
For more details about the format position refer to the appendix Encoder position format or to the manual (§4).

Object:	varEncMainMechAbsPosOffsetHi	Absolute position offset turns.	
Modbus IPA:	26012	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A01h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncMainMechAbsPosOffsetLo	Absolute position offset angle.	
Modbus IPA:	26013	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A01h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical angle unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncMainMechSpeed	Mechanical Speed.	
Modbus IPA:	26014	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A02h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the feedback of mechanical speed.
The speed is calculated as difference between two consecutive readings of the position encoder (125μs).

Object:	varEncMainMechAccel	Mechanical Acceleration.	
Modbus IPA:	26015	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A03h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the feedback mechanical acceleration.
The acceleration is simply computed as difference of the speed (every 125µs).

Object:	varEncMainElecAngle	Electrical angle.	
Modbus IPA:	26016	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3A04h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the feedback electrical position of the motor.

Object:	varEncMainStatus	Main Encoder Status	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A05.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	See table above.		

Main encoder status, each bit has different meaning as below:

Mask number	*Name*	*Description*
01h	cstEncStatusRelativeValid	Relative angle is valid.
02h	cstEncStatusElecAngleValid	Electrical angle is valid.
04h	cstEncStatusFatalFaultValid	Fatal fault during the calculations position.
08h	cstEncStatusNonFatalFaultValid	Non-fatal fault status. Combine status from both encoders (absolute and relative status) if one is faulty but electronic angle is valid and relative angle is valid; then result in non-fatal fault as there's at least one valid feedback.
10h	cstEncStatusAbsoluteValid	Absolute mechanical turn is valid.
20h	cstEncStatusAbsoluteWaiting	Waiting status during the calculations of absolute mechanical position for incremental encoder.

Object:	n/a	Main Encoder Position.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A06.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Main encoder position on 64 bit range calculated every 125 µsec.

Object:	n/a	Main Encoder Position Offset.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A07.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Offset value for absolute position on 64 bit range.

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncMainDeltaElecAngle	Angle phase offset.	
Modbus IPA:	26307	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	38F8h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Calculated electrical angle phase offset.
This is the result of electrical field orientation procedure.
At the end of the procedure the user have to copy this parameter into parMotorData.PhaseOffset; then *Save parameters* command and *Reset* of the drive are necessary.

4.1.13 Monitor > Absolute

Variables related to the absolute position of the Main Encoder.
The absolute position is the position of the encoder within one revolution.
These parameters are available if both absolute and relative main encoder type are selected; otherwise these values are set to zero.

Object:	varEncAbsMechHi	Feedback turns.	
Modbus IPA:	26400	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A10h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the absolute mechanical position in mechanical turns unit.
The feedback turns is the MSB 32 bit of 64 bit main encoder absolute position value.

Object:	varEncAbsMechLo	Feedback angle.	
Modbus IPA:	26401	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A10h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the absolute mechanical position in mechanical angle unit.
The feedback angle is the LSB 32 bit of 64 bit main encoder absolute position value.

Object:	varEncAbsMechAbsPosOffsetHi	Absolute position offset turns.	
Modbus IPA:	26402	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A11h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncAbsMechAbsPosOffsetLo	Absolute position offset angle.	
Modbus IPA:	26403	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A11h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncAbsMechSpeed	Absolute mechanical speed.	
Modbus IPA:	26404	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A12h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the absolute mechanical speed.
The speed is calculated as difference between two consecutive readings of the absolute position encoder (125μs).

Object:	varEncAbsMechAccel	Absolute mechanical acceleration.	
Modbus IPA:	26405	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A13h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale		

This is the absolute mechanical acceleration.
The acceleration is simply computed as difference of the speed (every 125μs).

Object:	varEncAbsElecAngle	Absolute electrical angle.	
Modbus IPA:	26406	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3A14h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale		

This is the electrical absolute position of the motor.

Object:	n/a	Main Absolute Encoder Status	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A15.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	See table above.		

Main absolute encoder status.

Mask number	*Name*	*Description*
01h	cstEncStatusRelativeValid	Relative angle is valid.
02h	cstEncStatusElecAngleValid	Electrical angle is valid.
04h	cstEncStatusFatalFaultValid	Fatal fault during the calculations position.
08h	cstEncStatusNonFatalFaultValid	Non-fatal fault status. Combine status from both encoders (absolute and relative status) if one is faulty but electronic angle is valid and relative angle is valid; then result in non-fatal fault as there's at least one valid feedback.
10h	cstEncStatusAbsoluteValid	Absolute mechanical turn is valid.
20h	cstEncStatusAbsoluteWaiting	Waiting status during the calculations of absolute mechanical position for incremental encoder.

Object:	n/a	Main Absolute Encoder Position.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A16.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Main absolute encoder position on 64 bit range calculated every 125 µsec.

Object:	n/a	Main Absolute Encoder Position Offset.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A17.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Offset value for absolute position on 64 bit range.

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

4.1.14 Monitor > Relative

Variables related to the relative position of the Main Encoder.
In principle, incremental encoders transmit relative position values (counter starts from zero at switch-on).
These parameters are available if both absolute and relative main encoder type are selected; otherwise these values are set to zero.

Object:	varEncRelMechHi	Relative mechanical turns.	
Modbus IPA:	26410	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A20h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the relative mechanical position in turns unit.
The relative mechanical turns is the 32 MSB bit of 64 bit main encoder relative position value.

Object:	varEncRelMechLo	Relative mechanical angle.	
Modbus IPA:	26411	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A20h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the relative mechanical position in angle unit.
The relative mechanical angle is the 32 LSB bit of 64 bit main encoder relative position value.

Object:	varEncRelMechAbsPosOffsetHi	Absolute position offset turns.	
Modbus IPA:	26412	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A21h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncRelMechAbsPosOffsetLo	Absolute position offset angle.	
Modbus IPA:	26413	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A21h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical angle unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncRelMechSpeed	Mechanical speed.	
Modbus IPA:	26414	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A22h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the relative mechanical speed.
The speed is calculated as difference between two consecutive readings of the relative position encoder (125µs).

Object:	varEncRelMechAccel	Mech. acceleration.	
Modbus IPA:	26415	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A23h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the relative mechanical acceleration.
The acceleration is simply computed as difference of the speed (every 125µs).

Object:	varEncRelElecAngle	Relative absolute electrical angle.	
Modbus IPA:	26416	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3A24h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the relative feedback electrical angle.

Object:	varEncRelStatus	Main Relative Encoder Status.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A25.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	See table above.		

Main relative encoder status.

Mask number	*Name*	*Description*
01h	cstEncStatusRelativeValid	Relative angle is valid.
02h	cstEncStatusElecAngleValid	Electrical angle is valid.
04h	cstEncStatusFatalFaultValid	Fatal fault during the calculations position.
08h	cstEncStatusNonFatalFaultValid	Non-fatal fault status. Combine status from both encoders (absolute and relative status) if one is faulty but electronic angle is valid and relative angle is valid; then result in non-fatal fault as there's at least one valid feedback.
10h	cstEncStatusAbsoluteValid	Absolute mechanical turn is valid.

20h	cstEncStatusAbsoluteWaiting	Waiting status during the calculations of absolute mechanical position for incremental encoder.

Object:	n/a	Main Relative Encoder Position.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A26.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Poisition Factor Unit.	*Default value:*	n/a
Value range:	Full scale.		

Main relative encoder position on 64 bit range calculated every 125 µsec.

Object:	n/a	Main Relative Encoder Position Offset.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A27.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Offset value for relative position on 64 bit range.
This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

4.2 Encoder > Auxiliary

Parameters to set-up the Auxiliary Encoder (physically connected to the AxM-II C1 connector).

Object:	parEncMgr.AuxSelection	Encoder type Auxiliary Selection	
Modbus IPA:	26002	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3802h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw		
Unit:	n/a	*Default value:*	0000h
Value range:	See specification above.		

Select the Auxiliary encoder type to use.

Option code	*Description*	*Name*
0000h	No sensor.	cstEncTypeNull
0004h	Absolute encoder Endat (Heidenhain encoder).	cstEncTypeB1AbsEndat
0006h	Digital Incremental encoder.	cstEncTypeRelIncremental
0008h	Simulation of digital incremental encoder (TTL standard)	cstEncTypeRelIncrSimulation

For more details on digital incremental encoder or on Endat encoder please refer to parEncMgr.MainAbselection and to parEncMgr.MainRelselection.

4.2.1 Endat

Parameters to manage the Auxiliary Encoder Endat configuration.

Object:	parEncAEndat.ClockFreq	Auxiliary Endat frequency protocol.	
Modbus IPA:	26120	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3910h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	kHz	*Default value:*	2000
Value range:	100 – max value depending by Endat protocol type.		

Clock frequency selection for Endat protocol.

- Max Value Endat 2.1: 2000 kHz
- Max Value Endat 2.2: 8000 kHz

Object:	parEncAEndat.MTurnStartPos	Auxiliary set point position at reset.	
Modbus IPA:	26123	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3911h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,reset,retain		
Unit:	Position factor unit.	*Default value:*	4294967295
Value range:	0 to 4294967295		

Set point position at start-up. This parameters is utilized only with Endat multiturn encoder.
For more details refer to parEncMEndat.MTurnStartPos parameter.

4.2.2 Endat > Monitor

Status of the of the Auxiliary Encoder Endat.

Object:	varEncAEndat.CrcErrors	Auxiliary CRC error counters.	
Modbus IPA:	26121	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3918h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	n/a		

Auxiliary Endat protocol checksum value errors counter, could be used as electrical connection diagnostic checking.

Object:	varEncAEndat.PropDelay	Auxiliary Propagation delay.	
Modbus IPA:	26122	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3919h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	nsec	*Default value:*	n/a
Value range:	n/a		

Data propagation delay: the value is calculated during Endat initialization procedure.
For more information read manual "Endat: VERSION 2.2 Bidirectional Synchronous-Serial Interface for Position Encoders " from Heidenhain Gmbh.

4.2.3 Incremental Traces

With the incremental measuring method, the graduation consists of a periodic grating structure. The position information is obtained by counting the individual increments (measuring steps) from some point of origin. Since an absolute reference is required to ascertain positions, the graduated disks are provided with an additional track that bears a reference mark.
The incremental signals are transmitted as the square-wave pulse trains U1 and U2, phase-shifted by 90° elec. The reference mark signal consists of one or more reference pulses U0, which are gated with the incremental signals.
Parameters to manage the Auxiliary Incremental Encoder.

Object:	parEncAInc.LineCounts	Auxiliary Encoder Line Counts	
Modbus IPA:	26130	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3920h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,reset,retain		
Unit:	count	*Default value:*	512
Value range:	Full scale.		

Encoder pulses number per revolution setting. For more details refer to parEncMInc.LineCounts.

Object:	parEncAInc.Flags.DisableIndexError	Auxiliary Disable Index Error.	
Modbus IPA:	26134	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3922h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0 (Off)
Value range:	0(Off)-1(On)		

Flag to disable the error generated by the index track checking.

Object:	parEncAInc.IndexErrorTolerance	Auxiliary Index error Tolerance.	
Modbus IPA:	26135	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3925h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	count	*Default value:*	10
Value range:	Full scale.		

Tolerance of the difference between the calcolated zero position and the index track used for index track checking.

Object:	parEncAInc.FlagsEnableIndexTrack	Auxiliary enable index track.	
Modbus IPA:	26136	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3926h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Flag to enable index track.

Object:	parEncAInc.Flags.Swaptracks	Auxiliary swap tracks.	
Modbus IPA:	26137	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	392Bh.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Swap physical input tracks A and B to adjust wrong electrical connections.

Object:	parEncAInc.Flags.EnableStepDir	Auxiliary Enable pulse and direction tracks.	
Modbus IPA:	26138	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	392Ch.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Enable step pulse (track A) and direction (track B) mode in place of standard quadrature encoder mode.
For more details refer to parEncMInc.Flags.EnableStepDir.

Object:	parEncAInc.Flags.EnableUpDown	Auxiliary enable index track.	
Modbus IPA:	26139	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	392Dh.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Flag to enable up pulse (track A) and down pulse (track B) mode in place of standard quadrature encoder mode. For more details refer to parEncMInc.Flags.EnableUpDown.

Object:	parEncSimInc.IndexLineCounts	Simulated encoder counts.	
Modbus IPA:	26146	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3928h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,reset,retain		
Unit:	counts	*Default value:*	512
Value range:	2 to 16384		

The encoder simulation signal is generated in the A quadrature B format with index marker trace. The simulated line counts for motor revolution is setting with this parameter.

The simulated encoder feature is usually used with speed or current analog references using ±10 V-interface. With encoder Simulation property the current actual position of the motor controlled by the AxM-II can be read by a higher-level control system.
The voltage level is a standard TTL signal.

Object:	parEncSimInc.IndexOffset	Simulated encoder Index offset.	
Modbus IPA:	26147	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3929h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,reset,retain		
Unit:	counts	*Default value:*	0
Value range:	0 to 65536		

The variable index offset holds the desired rate for the index pulse.

Object:	parEncSimInc.MaxPosErrTolerance	Simulated encoder Pos. Tolerance.	
Modbus IPA:	26148	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	392Ah.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	Position conversion factor.	*Default value:*	0
Value range:	Full scale.		

This parameter means a maximum tolerance of error between the main position calculated and the position simulated by the drive. When the difference between main encoder position and the calculated encoder simulation position is greater then maximun position error tolerance a Simulation Invalid Count alarm is generated.

4.2.4 Monitor

Variables related to the position of the Auxiliary Encoder.

Object:	varEncAuxMechHi	Feedback turns.	
Modbus IPA:	26110	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A30h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the auxiliary feedback position in mechanical turns unit.
The feedback turns is the MSB 32 bit of 64 bit auxiliary encoder position value.

Object:	varEncAuxMechLo	Feedback angle.	
Modbus IPA:	26111	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A30h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the auxiliary feedback position in mechanical angle unit.
The feedback angle is the LSB 32 bit of 64 bit auxiliary encoder position value.

Object:	varEncAuxMechAbsPosOffsetHi	Absolute Position Offset turns.	
Modbus IPA:	26112	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A31h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncAuxMechAbsPosOffsetLo	Absolute Position Offset angle.	
Modbus IPA:	26113	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A31h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical angle unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncAuxMechSpeed	Mechanical speed.	
Modbus IPA:	26114	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A32h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This object represents the actual value of the auxiliary velocity measurement device.
The speed is calculated as difference between two consecutive readings of the position auxiliary encoder (125µs).

Object:	varEncAuxMechAccel	Mechanical acceleration.	
Modbus IPA:	26115	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A33h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the auxiliary mechanical acceleration.
The acceleration is simply computed as difference of the auxiliary speed, still every 125µs.

Object:	varEncAuxElecAngle	Auxiliary Electrical angle.	
Modbus IPA:	26116	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3A34h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the auxiliary feedback electrical position of the motor.

Object:	varEncAuxStatus	Auxiliary Encoder Status.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A35.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	See table above.		

Auxiliary encoder status.

Mask number	*Name*	*Description*
01h	cstEncStatusRelativeValid	Relative angle is valid.
02h	cstEncStatusElecAngleValid	Electrical angle is valid.
04h	cstEncStatusFatalFaultValid	Fatal fault during the calculations position.
08h	cstEncStatusNonFatalFaultValid	Non-fatal fault status. Combine status from both encoders (absolute and relative status) if one is faulty but electronic angle is valid and relative angle is valid; then result in non-fatal fault as there's at least one valid feedback.
10h	cstEncStatusAbsoluteValid	Absolute mechanical turn is valid.
20h	cstEncStatusAbsoluteWaiting	Waiting status during the calculations of absolute mechanical position for incremental encoder.

Object:	n/a	Auxiliary Encoder Position.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A36.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Auxiliary encoder position on 64 bit range calculated every 125 µsec.

4.3 Encoder > Monitor

In this sub-menu can be read the variables related to the position of the Feedback Encoder (the one used by the Control Loops) selected by the selection flags. For more informations refer to (§4).

Object:	varEncFbMechHi	Feedback turns.	
Modbus IPA:	27020	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A40h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the feedback encoder position in mechanical turns unit.
The feedback turns is the 32 MSB bit of 64 bit feedback encoder position.

Object:	varEncFbMechLo	Feedback angle.	
Modbus IPA:	27021	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A40h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the feedback encoder position in mechanical angle unit.
The feedback angle is the 32 LSB bit of 64 bit feedback encoder position.

Object:	varEncFbMechAbsPosOffsetHi	Feedback Absolute pos. offset turns.	
Modbus IPA:	27022	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A41h.1h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncFbMechAbsPosOffsetLo	Feedback Absolute pos. offset angle.	
Modbus IPA:	27023	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3A41h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical angle unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	varEncFbMechSpeed	Mechanical speed.	
Modbus IPA:	27024	*Modbus Data Type:*	Signed32
CANopen/COE index:	606Ch.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Velocity factor unit.	*Default value:*	n/a

Value range:	Full scale.

This object represents the actual value of the velocity measurement device.
It corresponds with DSP402 profile specific object: velocity actual value.

Object:	varEncFbMechAccel	Mechanical acceleration.	
Modbus IPA:	27025	*Modbus Data Type:*	Signed32
CANopen/COE index:	3A43h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This object represents the actual value of the acceleration.
The acceleration is simply computed as difference of the speed (every 125μs).

Object:	varEncFbElecAngle	Electrical angle.	
Modbus IPA:	27026	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3A44h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the electrical position of the motor.

Object:	varEncFbStatus	Encoder Feedback status.	
Modbus IPA:	27028	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3A45.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	See table above.		

Encoder feedback status.

Mask number	*Name*	*Description*
01h	cstEncStatusRelativeValid	Relative angle is valid.
02h	cstEncStatusElecAngleValid	Electrical angle is valid.
04h	cstEncStatusFatalFaultValid	Fatal fault during the calculations position.
08h	cstEncStatusNonFatalFaultValid	Non-fatal fault status. Combine status from both encoders (absolute and relative status) if one is faulty but electronic angle is valid and relative angle is valid; then result in non-fatal fault as there's at least one valid feedback.
10h	cstEncStatusAbsoluteValid	Absolute mechanical turn is valid.
20h	cstEncStatusAbsoluteWaiting	Waiting status during the calculations of absolute mechanical position for incremental encoder.

Object:	n/a	Encoder Feedback Position.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A46.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Encoder feedback position calculated in 64 bit range.

Object:	n/a	Encoder Feedback Position Offset.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3A47.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the relative position in mechanical turns unit in order to calculate the absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

5. TORQUE LOOP

The default configuration of the Torque Loop gains is usually enough to control most of the Phase Motion Control motors. To get the best performances it's necessary to tune the torque loop using the following parameters.

Object:	parPStage.OverCurrentThreshold	Over Current Threshold	
Modbus IPA:	27404	*Modbus Data Type:*	Signed32
CANopen/COE index:	3144h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1mA	*Default value:*	103000
Value range:	See above.		

Over Current threshold is the maximum current value that the AxM-II drive can provide. The user can modify the Over Current threshold detection, but it's possible only to set a value lower than the hardware AxM-II power stage.

When the actual current has achieved an instant value higher than the maximum allowed by the drive an Over Current alarm is generated.

WARNING: If this value is set to zero, then the default hardware drive's overcurrent limit is restored. This change takes effect only after the *Save parameters* command and *Reset* of the drive.

Object:	parILoop.ILoopKi	Modulator Ki	
Modbus IPA:	27407	*Modbus Data Type:*	Real32
CANopen/COE index:	3120h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,retain,pdomap		
Unit:	--	*Default value:*	0
Value range:	Full scale.		

Current Loop Integral Gain. If set to zero, it's used the Ki automatically calculated by the firmware looking motor data parameters (refer to section §3).

Object:	parILoop.ILoopKp	Modulator Kp	
Modbus IPA:	27408	*Modbus Data Type:*	Real32
CANopen/COE index:	3121h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	rw,retain,pdomap		
Unit:	--	*Default value:*	0
Value range:	Full scale.		

Current Loop Proportional Gain. If set to zero, it's used the Kp automatically calculated by the firmware looking motor data parameters (refer to section §3).

WARNING: the values of the objects parILoop.ILoopKi and parILoop.ILoopKp could be written also during the normal drive working cycle, thus with power enabled and moving shaft. Be careful as modifying the values of this object with power enabled could yield in a loss of axle control.

Object:	parILoop.IdLimitMin	Direct Current Limit Min	
Modbus IPA:	27409	*Modbus Data Type:*	Signed32
CANopen/COE index:	3122h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1mA	*Default value:*	0.0
Value range:	0 to -214748.3647		

Direct current negative minimum limit when in *Torque Mode* profile. It's clipped to zero if the user sets a positive value.

Object:	parILoop.IdLimitMax	Direct Current Limit Max	
Modbus IPA:	27410	*Modbus Data Type:*	Signed32
CANopen/COE index:	3123h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1mA	*Default value:*	0.0
Value range:	0 to 214748.3647		

Direct current positive maximum limit when in *Torque Mode* profile. It's clipped to zero if the user sets a negative value.

Object:	parILoop.IqLimitMin	Quadrature Current Limit Min	
Modbus IPA:	27411	*Modbus Data Type:*	Signed32
CANopen/COE index:	3124h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1mA	*Default value:*	0.0
Value range:	0 to -214748.3647		

Quadrature current negative minimum limit when in *Torque Mode* profile. It's clipped to zero if the user sets a positive value.

Object:	parILoop.IqLimitMax	Quadrature Current Limit Max	
Modbus IPA:	27412	*Modbus Data Type:*	Signed32
CANopen/COE index:	3125h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1mA	*Default value:*	0.0
Value range:	0 to 214748.3647		

Quadrature current positive maximum limit when in *Torque Mode* profile. It's clipped to zero if the user sets a negative value.

Object:	parPStageEx.SoftStart	Soft-Start period	
Modbus IPA:	27416	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3145h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0004h (310 msec)
Value range:	See table above.		

Soft start is used to limit the inrush current to the dc bus capacitor bank on application of ac mains power. This is intended to prevent nuisance tripping of circuit breakers or blowing of line fuses on power-up. Time of activation of the Soft-Start circuit:

Option code	*Soft-Start time*
0000h	Soft-Start disable
0001h	38.75 msec
0002h	75.50 msec
0003h	155.00 msec
0004h	310.00 msec

WARNING: This change takes effect only after the *Save parameters* command and *Reset* of the drive.

Object:	parPStage.BridgeLayout	Bridge Layout Assignment	
Modbus IPA:	27490	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3146h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,reset,retain		
Unit:	n/a	*Default value:*	0000h (U V W)
Value range:	See table above.		

Motor wires order.
This parameter has to correspond to the motor phase U – V – W sequence on the AxM-II connector P1.

Option code	*Description*	*Name Mask*
0000h	Bridge layout: U V W	cstPStageBridgeLayoutUVW
0001h	Bridge layout: U W V	cstPStageBridgeLayoutUWV
0002h	Bridge layout: V U W	cstPStageBridgeLayoutVUW
0003h	Bridge layout: W U V	cstPStageBridgeLayoutWUV
0004h	Bridge layout: V W U	cstPStageBridgeLayoutVWU
0005h	Bridge layout: W V U	cstPStageBridgeLayoutWVU

WARNING: Use with caution. Useful parameter when it's wrong wire sequence on the motor cable and is unable to change it. This change takes effect only after the *Save parameters* command and *Reset* of the drive.

Object:	wksIdRef	User Id reference	
Modbus IPA:	27464	*Modbus Data Type:*	Signed32
CANopen/COE index:	3126h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,pdomap		
Unit:	0.1 mA	*Default value:*	0.0
Value range:	Full scale.		

Direct current target for current loop control. The direct current target can be generated from the master/host control otherwise this parameter should be configuring by the user using the Cockpit configurator. In both cases put the AxM-II drive in *Torque Mode* profile, otherwise it is ignored.

Object:	wksIqRef	User Iq reference	
Modbus IPA:	27465	*Modbus Data Type:*	Signed32
CANopen/COE index:	3127h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,pdomap		
Unit:	0.1 mA	*Default value:*	0.0
Value range:	Full scale.		

Quadrature current target for current loop control. The quadrature current target can be generated from the master/host control otherwise this parameter should be configuring by the user using the Cockpit configurator. In both cases put the AxM-II drive in *Torque Mode* profile, otherwise it is ignored.

5.1 Torque Loop Monitor

Variables related to the torque loop control.

Object:	varILoopIdFb	Id feedback	
Modbus IPA:	27432	*Modbus Data Type:*	Signed32
CANopen/COE index:	3100h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Feedback direct current generated by the Park's transformation.

Object:	varILoopIqFb	Iq feedback	
Modbus IPA:	27433	*Modbus Data Type:*	Signed32
CANopen/COE index:	3101h.0h	*CANopen/COE Data Type:*	Signed32

Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Feedback quadrature current generated by the Park's transformation.

Object:	varILoopIuFb	Iu feedback	
Modbus IPA:	27434	*Modbus Data Type:*	Signed32
CANopen/COE index:	3102h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Feedback current phase U read by the drive's current hall sensors.

Object:	varILoopIvFb	Iv feedback	
Modbus IPA:	27435	*Modbus Data Type:*	Signed32
CANopen/COE index:	3103h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Feedback current phase V read by the drive's current hall sensors.

Object:	varILoopVuOut	Output Vu	
Modbus IPA:	27438	*Modbus Data Type:*	Signed16
CANopen/COE index:	3107h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	100 mV	*Default value:*	n/a
Value range:	Full scale.		

Actual value of the voltage of the motor phase U.

Object:	varILoopVvOut	Output Vv	
Modbus IPA:	27439	*Modbus Data Type:*	Signed16
CANopen/COE index:	3108h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro,pdomap		
Unit:	100 mV	*Default value:*	n/a
Value range:	Full scale.		

Actual values of the voltage of the motor phase V.

Object:	varILoopIdRef	Id Reference	
Modbus IPA:	27443	*Modbus Data Type:*	Signed32
CANopen/COE index:	3105h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

This is the direct current reference set as torque loop input.

Object:	varILoopIqRef	Iq Reference	
Modbus IPA:	27444	*Modbus Data Type:*	Signed32
CANopen/COE index:	3106h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

This is the quadrature current reference set as torque loop input.

Object:	varILoopAutoILoopKi	Modulator Ki (auto)	
Modbus IPA:	27480	*Modbus Data Type:*	Real32
CANopen/COE index:	3109h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	ro,pdomap		
Unit:	n/a	*Default value:*	0.0
Value range:	Full scale.		

Torque loop gain calculated by the drive (based on motor inductance).

Object:	varILoopAutoILoopKp	Modulator Kp (auto)	
Modbus IPA:	27481	*Modbus Data Type:*	Real32
CANopen/COE index:	310Ah.0h	*CANopen/COE Data Type:*	Real32
Attributes:	ro,pdomap		
Unit:	n/a	*Default value:*	0.0
Value range:	Full scale.		

Torque loop gain calculated by the drive (based on motor inductance).

Object:	varILoopIdMin	Id Limit Min	
Modbus IPA:	27482	*Modbus Data Type:*	Signed32
CANopen/COE index:	310Bh.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied lower limit for the direct current.

Object:	varILoopIdMax	Id Limit Max	
Modbus IPA:	27483	*Modbus Data Type:*	Signed32
CANopen/COE index:	310Ch.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied higher limit for the direct current.

Object:	varILoopIqMin	Iq Limit Min	
Modbus IPA:	27484	*Modbus Data Type:*	Signed32
CANopen/COE index:	310Dh.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied lower limit for the quadrature current.

Object:	varILoopIqMax	Iq Limit Max	
Modbus IPA:	27485	*Modbus Data Type:*	Signed32
CANopen/COE index:	310Eh.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied higher limit for the quadrature current.

6. SPEED POSITION LOOP

These parameters control the behaviour and the performance of the position and speed loop.
It's necessary to tune these parameters according to the application and to the desired performance.

WARNING: the values of these objects could be written also during the normal drive working cycle, thus with power enabled and moving shaft. Be careful as modifying the values of this object with power enabled could yield in a loss of axes control.

6.1 Space Speed Control Loop

Object:	parSSCntrlLp.Flags.UseDifferentKp	Use different speed Kp	
Modbus IPA:	27000	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3300h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)–1(On)		

Use different coefficient for proportional gain speed reference (parSSCntrlLp.SpdKpRef) and for proportional gain speed feedback (parSSCntrlLp.SpdKPFbk).

- 0(Off): use the same proportional gain for speed reference and for speed feedback. The value of the two parameters is equal to the parSSCntrlLp.SpdKpRef parameter value.
- 1(On): use different proportional gain for speed reference and for speed feedback.

For further information refer to section (§E).

Object:	parSSCntrlLp.PosKp	Position Kp gain	
Modbus IPA:	27001	*Modbus Data Type:*	Signed16
CANopen/COE index:	3301h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Position loop: proportional gain value.
This gain is applied to the position error; i.e. the difference between the position reference and the position feedback.
In velocity profile mode, it's the integral gain of the speed control loop.

Object:	parSSCntrlLp.SpdKpRef	Speed: kp reference value.	
Modbus IPA:	27002	*Modbus Data Type:*	Signed16
CANopen/COE index:	3302h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Proportional gain value applied to the speed reference.

Object:	parSSCntrlLp.SpdKPFbk	Speed: kp feedback value.	
Modbus IPA:	27003	*Modbus Data Type:*	Signed16
CANopen/COE index:	3303h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Proportional gain value to the feedback speed reference.

Object:	parSSCntrlLp.AccKpRef	Acceleration: kp reference value.	
Modbus IPA:	27004	*Modbus Data Type:*	Signed16
CANopen/COE index:	3304h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Acceleration: reference proportional gain value.

Object:	parSSCntrlLp.AccKpFbk	Acceleration: kp feedback value.	
Modbus IPA:	27005	*Modbus Data Type:*	Signed16
CANopen/COE index:	3305h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Acceleration: feedback proportional gain value.

Object:	parSSCntrlLp.Ki	Integral Gain.	
Modbus IPA:	27006	*Modbus Data Type:*	Signed16
CANopen/COE index:	3306h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Speed loop: integral gain value.

Object:	parSSCntrlLp.PosGainShift	Position gain shift.	
Modbus IPA:	27007	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3307h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	0 to 16.		

Divisor of the position gain expressed in exponential base 2.
E.g.: with the value 2 the proportional gain is divided by 4.

Object:	parSSCntrlLp.AccGainShift	Acceleration gain shift.	
Modbus IPA:	27008	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3308h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	0 to -15.		

Multiplier of the Acceleration Gain expressed in exponential base 2.
E.g.: with the value 2 the proportional gain is multiplied by 4.

Object:	parSSCntrlLp.GlobalGainShift	Global gain shift.	
Modbus IPA:	27009	*Modbus Data Type:*	Signed16
CANopen/COE index:	3309h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	n/a	*Default value:*	0
Value range:	-8 to 16.		

Adapter of the result of the speed/position loop. Gain expressed in exponential base 2.
This parameters support maximum shift left -8 (multiplier) and maximum shift right 16 (divisor).

Object:	parSSCntrlLp.IlimitMax	Max Limit Torque.	
Modbus IPA:	27012	*Modbus Data Type:*	Signed32
CANopen/COE index:	330Ah.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1 mA	*Default value:*	2147483647
Value range:	0 to 214748.3647		

Saturation positive limit of the torque reference.

Object:	parSSCntrlLp.IlimitMin	Min Limit Torque.	
Modbus IPA:	27013	*Modbus Data Type:*	Signed32
CANopen/COE index:	330Bh.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	0.1 mA	*Default value:*	-2147483647
Value range:	-214748.3647 to 0		

Saturation negative limit of the torque reference.

6.2 Positioner

Variables related to the "Positioner" software module.

Object:	parPositioner.ProfileVel	Profile velocity.	
Modbus IPA:	27100	*Modbus Data Type:*	Signed32
CANopen/COE index:	6081h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Velocity factor unit.	*Default value:*	17895700 (~ 209.4395 rad/s)
Value range:	Full scale.		

The profile velocity is the velocity obtained at the end of the acceleration ramp during the profile move and is valid for both directions of motion.

Object:	parPositioner.ProfileAcc	Profile acceleration.	
Modbus IPA:	27101	*Modbus Data Type:*	Signed32
CANopen/COE index:	6083h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	1000000 (~ 22.8581 rad/s²)
Value range:	Full scale.		

The profile acceleration is given in user defined acceleration unit and is used to plan the acceleration slope.

Object:	parPositioner.ProfileDec	Profile deceleration.	
Modbus IPA:	27102	*Modbus Data Type:*	Signed32
CANopen/COE index:	6084h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	1000000 (~ 22.8581 rad/s²)
Value range:	Full scale.		

The profile deceleration is given in user defined acceleration unit and is used to plan the deceleration slope.

Object:	parPositioner.QuickStopDec	Quick stop deceleration.	
Modbus IPA:	27103	*Modbus Data Type:*	Signed32
CANopen/COE index:	6085h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	10000000 (~228.5809 rad/s²)
Value range:	Full scale.		

The quick stop deceleration is the deceleration used to stop the motor if the quick stop ramp is selected as option code. Refer to (§3).

Object:	parPositioner.EndVelocity	End velocity.	
Modbus IPA:	27104	*Modbus Data Type:*	Signed32
CANopen/COE index:	6082h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Velocity factor unit.	*Default value:*	0
Value range:	Full scale.		

The end velocity defines the velocity, which the drive must have on reaching the target position. Normally, the drive stops at the target position, i.e. the end velocity near to zero.

Object:	parPositioner.PositionErrorMax	Max position error allowed.	
Modbus IPA:	27105	*Modbus Data Type:*	Signed32
CANopen/COE index:	3340h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	Position factor unit.	*Default value:*	327680 (~ 1800 °)
Value range:	Full scale.		

This is the maximum following error allowed during the profile mode (position or speed) working.
Refer to parameters parDevCtrl.FollowingErrWindow and parDevCtrl.FollowingErrTimeout to understand the behaviour of the following error.

Object:	parPositioner.ZeroSpeedThreshold	Threshold speed.	
Modbus IPA:	27106	*Modbus Data Type:*	Signed32
CANopen/COE index:	3341h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw,retain,pdomap		
Unit:	Velocity factor unit.	*Default value:*	4475 (~ 0.05 rad/s)
Value range:	Full scale.		

Below this speed threshold the motor is considered standstill.

6.3 Monitor

Variables related to the position and speed loops.

Object:	varSSCntrlLpIqRef	Torque reference.	
Modbus IPA:	27029	*Modbus Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	0.1mA	*Default value:*	n/a
Value range:	Full scale.		

It represents the current reference output of the speed/position loop.

Object:	varPosRGDemandPosHi	Demand Position (turn)	
Modbus IPA:	27122	*Modbus Data Type:*	Signed32
CANopen/COE index:	3350h.2h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Reference position into mechanical turns. Represents the 32 MSB bit position input of the space speed control loop as computed by the trajectory generator.

Object:	varPosRGDemandPosLo	Demand Position (angle)	
Modbus IPA:	27123	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3350h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Reference position into mechanical angle unit. Represents the 32 LSB bit position input of the space speed control loop as computed by the trajectory generator.

Object:	varPosRGDemandSpeed	Demand Speed.	
Modbus IPA:	27124	*Modbus Data Type:*	Signed32
CANopen/COE index:	606Bh.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

It corresponds to the velocity demand value: this is the output value of the trajectory generator.

Object:	varPosRGDemandAccel	Demand acceleration.	
Modbus IPA:	27125	*Modbus Data Type:*	Signed32
CANopen/COE index:	3352h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Input acceleration reference in the speed/position loop.

Object:	varPosRGPosError	Following error.	
Modbus IPA:	27126	*Modbus Data Type:*	Signed32
CANopen/COE index:	60F4h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro,pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Following error actual value.

7. DEVICE CONTROL

Parameters defined in DSP402 profile of the CanOpen specifications.
For further information's refer to (§3).

Object:	parDevCtrl.ModeOfOperation	Mode of Operation.	
Modbus IPA:	29016	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	6060h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,pdomap		
Unit:	n/a	*Default value:*	03h
Value range:	See table above.		

This parameter switches the operation mode. The possible values are:

Option code	*Description*	*Name mask*
01h	Profile position mode	cstDevCtrlModeOfOpProfilePosition
03h	Profile velocity mode	cstDevCtrlModeOfOpProfileVelocity
06h	Homing mode	n/a
07h	Interpolated position mode	n/a
80h	Torque mode	cstDevCtrlModeOfOpTorqueMode

This value parameter is reflected in the object modes of operation display.
The AxM-II supports switching between the various modes of operation, also when the axes is moving.

Object:	parDevCtrl.QuickStopOptCode	Quick Stop Option Code.	
Modbus IPA:	29020	*Modbus Data Type:*	Signed16
CANopen/COE index:	605Ah.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain		

Unit:	n/a	*Default value:*	0002h
Value range:	See table above.		

This determines what action should be taken if the Quick stop function is executed (transition 11). The action could be one of the following:

Option code	*Description*	*Name mask*
0000h	Disable drive function	cstDevCtrlOptCodePowerOff
0001h	Slow down with slow down ramp; disable of the drive function	cstDevCtrlOptCodeSlowDown
0002h	Slow down with quick stop ramp; disable of the drive function	cstDevCtrlOptCodeQuickStop
0005h	Slow down with slow down ramp and stay in quick stop	cstDevCtrlOptCodeSlowDownAndStay
0006h	Slow down with quick stop ramp and stay in quick stop	cstDevCtrlOptCodeQuickStopAndStay

Object:	parDevCtrl.ShutdownOptCode	Shutdown Option code.	
Modbus IPA:	29021	*Modbus Data Type:*	Signed16
CANopen/COE index:	605Bh.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0000h
Value range:	See above table.		

This parameter determines what action should be taken if there is a transition from Operation enable to Ready to switch on (transition 8). The action could be one of the following:

Option code	*Description*	*Name mask*
0000h	Disable drive function	cstDevCtrlOptCodePowerOff
0001h	Slow down with slow down ramp; disable of the drive function	cstDevCtrlOptCodeSlowDown

Object:	parDevCtrl.DisableOptCode	Disable Option Code.	
Modbus IPA:	29022	*Modbus Data Type:*	Signed16
CANopen/COE index:	605Ch.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0001h
Value range:	See above table.		

This parameter determines what action should be taken if there is a transition from Operation enable to Switched on (transition 5). The action could be one of the following:

Option code	*Description*	*Name mask*
0000h	Disable drive function	cstDevCtrlOptCodePowerOff
0001h	Slow down with slow down ramp; disable of the drive function	cstDevCtrlOptCodeQuickStop

Object:	parDevCtrl.HaltOptCode	Halt Option Code.	
Modbus IPA:	29023	*Modbus Data Type:*	Signed16
CANopen/COE index:	605Dh.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0001h
Value range:	See above table.		

This determines what action should be taken if the bit 8 (halt) in the controlword is active. The action could be one of the following:

Option code	*Description*	*Name Mask*
0000h	Disable drive, motor is free to rotate	cstDevCtrlOptCodePowerOff
0001h	Slow down with slow down ramp	cstDevCtrlOptCodeSlowDown
0002h	Slow down with quick stop ramp	cstDevCtrlOptCodeQuickStop

Object:	parDevCtrl.FaultReactionOptCode	Fault Reaction Option Code.	
Modbus IPA:	29024	*Modbus Data Type:*	Signed16
CANopen/COE index:	605Eh.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain		

Unit:	n/a	*Default value:*	0002h
Value range:	See above table.		

The parameter fault reaction option code determines what action should be taken if a fault occurs in the drive. The action could be one of the following:

Option code	*Description*	*Name Mask*
0000h	Disable drive, motor is free to rotate	cstDevCtrlOptCodePowerOff
0001h	Slow down with slow down ramp	cstDevCtrlOptCodeSlowDown
0002h	Slow down with quick stop ramp	cstDevCtrlOptCodeQuickStop

Object:	varStatusWord	Status Word.	
Modbus IPA:	29001	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	6041h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	ro,pdomap		
Unit:	n/a	*Default value:*	--
Value range:	n/a		

The statusword indicates the current state of the drive and the current state of the specific operating mode.

Bit	*Name*	*Description*
0	Ready to switch on	The drive functions are disabled, the drive is ready to enable power output.
1	Switched on	The drive functions are disabled, the drive has power output enabled, and the motor shaft has no torque.
2	Operation enabled	The drive functions and power output are enabled, the torque could be applied on the motor shaft, no faults detected, and specific selected Mode Of Operation is executed.
3	Fault	A fault is occurred in the device, the drive functions and power output are disabled.
4	Voltage enabled	Power output is enabled to the drive when this bit is set to 1.
5	Quick stop	The drive functions and power output are enabled, the quick stop function is being executed or finished and the motor stopped (depending from object 605Ah.0h)
6	Switch on disabled	AxM-II initialization is complete, and then is ready to accept command, the power output and the drive functions are disabled.
7	Warning	A warning is occurred in the device.
8	reserved	
9	Remote	If set, then parameters may be modified via the CAN bus, and the drive executes the content of a command message. If the bit remote is reset, then the drive is in local mode and will not execute the command message.
10	Target reached	If set, then a set-point has been reached (not used in Torque Mode and Homing Mode). The set-point is dependent on the operating mode. The change of a target value by software alters this bit. If quick stop option code is 5 or 6 this bit is set when the quick stop operation is finished and the drive is halted. If halt occurred and the drive has halted then this bit is set too.
11	Internal limit active	It signal that the target position (if in Profile Position Mode) or the set-point (if in Interpolated Mode) was wrapped between minimum and maximum Software position limit (object 607Dh), due to exceeding value. It is reset with a new target position or set-point between the limits (not used in Torque Mode).
12	O1	
13	O2	
14	reserved	
15	Homing done	The homing is done, this bit remain active up to a node reset or a power-off.

Table 1 - Structure of the statusword

The O1, O2 are operating mode specific bits:

Bit	*Position profile*	*Velocity profile*	*Interpolated profile*	*Torque mode*	*Homing mode*
O1	Set point acknowledge	Zero speed	Ip mode active	reserved	Homing attained
O2	Following error	Max slippage error	reserved	reserved	Homing error

Table 2 - Statusword operating mode specific bits

The reserved bit is for future enhancements, it has to be ignored.

Object:	varModeOfOperationDisplay	Mode of Operation Display.	
Modbus IPA:	29017	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	6061h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	ro,pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

The mode of operation display shows the current mode of operation.

Object:	wksControlWord	Control Word.	
Modbus IPA:	29000	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	6040h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

The controlword contains the bits for controlling the state machine and for controlling the specific operating mode.

MSB LSB

reserved (7 bit)	halt	fault reset	O3	O2	O1	enable oper.	quick stop	enable volt.	switch on

Figure 5 - Structure of controlword

The O1, O2, and O3 are operating mode specific bits:

Bit	*Position profile*	*Velocity profile*	*Interpolated profile*	*Torque mode*	*Homing mode*
O1	new set-point	reserved	enable ip mode	reserved	homing operation start
O2	change set immediately	reserved	reserved	reserved	reserved
O3	abs/rel	reserved	reserved	reserved	reserved

Table 3 - Controlword operating mode specific bits

The reserved bit are for future enhancements, should be kept to 0.

7.1 Profile Position

Parameters defined in CiA DSP402 profile position of the CANOpen specifications.
For further informations refer to (§3).

Object:	parDevCtrl.FollowingErrWindow	Following Error Window.	
Modbus IPA:	29025	*Modbus Data Type:*	Signed32
CANopen/COE index:	6065h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Position factor unit.	*Default value:*	65536 (~ 360°)
Value range:	n/a		

The following error window defines the maximum tolerance on the following error; if the following error actual value is greater than following error window, a following error occurs. A following error might occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed loop coefficients.

Object:	parDevCtrl.FollowingErrTimeout	Following Error Timeout.	
Modbus IPA:	29026	*Modbus Data Type:*	Signed16
CANopen/COE index:	6066h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	ms	*Default value:*	10
Value range:	n/a		

When a following error occurs longer than the defined value of the time-out, the corresponding bit13 of statusword is set to one.

Object:	parDevCtrl.TargetPosWindow	Target Position Window.	
Modbus IPA:	29027	*Modbus Data Type:*	Signed32
CANopen/COE index:	6067h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain,pdomap		
Unit:	Position factor unit.	*Default value:*	64 (~0.35°)
Value range:	n/a		

The position window defines a symmetrical range of accepted positions relatively to the target position.

(target position - position window ; target position + position window)

If the present value of the position encoder is within the position window, this target position is regarded as reached.

Object:	parDevCtrl.TargetPosTimeout	Position Timeout.	
Modbus IPA:	29028	*Modbus Data Type:*	Signed16
CANopen/COE index:	6068h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain,pdomap		
Unit:	ms	*Default value:*	20
Value range:	n/a		

When the actual position is within the position window during the position window timeout the corresponding bit10 in the statusword is set.

Object:	wksTargetPosHi	Target position mechanical turns.	
Modbus IPA:	29002	*Modbus Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	0
Value range:	Full scale.		

User target position in mechanical turns: is the MSB 32 bit of 64 bit target position.
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The target position that is usually generated from the master control; anyway the user should be configuring it using the Cockpit configurator and the basic application of the AxM-II.
The *Positioner* application allows the user to control the AxM-II drive as programmable multi-position positioner. For more details about the format position refer to the appendix Encoder position format or to the manual (§4).

Object:	wksTargetPosLo	Target position mechanical angle.	
Modbus IPA:	29003	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	Rw		
Unit:	Position factor unit.	*Default value:*	0
Value range:	Full scale.		

User target position in mechanical angle is the LSB 32 bit of 64 bit target position.
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The target position that is usually generated from the master control; anyway the user should be configuring it using the Cockpit configurator and the basic application of the AxM-II.
The *Positioner* application allows the user to control the AxM-II drive as programmable multi-position positioner. For more details about the format position refer to the appendix Encoder position format or to the manual (§4).

7.2 Profile Velocity

Parameters defined in CiA DSP402 Profile Velocity mode of the CANOpen specifications.
For further information refer to (§3).

Object:	parDevCtrl.VelocityWindow	Velocity window.	
Modbus IPA:	29029	*Modbus Data Type:*	Signed32
CANopen/COE index:	606Dh.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, pdomap		
Unit:	Velocity factor unit.	*Default value:*	894616 (~10.47 rad/s)
Value range:	Full scale.		

The velocity window monitors whether the required process velocity has been achieved after an eventual acceleration or deceleration (braking).

Object:	parDevCtrl.VelocityTimeout	Velocity timeout.	
Modbus IPA:	29030	*Modbus Data Type:*	Signed16
CANopen/COE index:	606Eh.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, pdomap		
Unit:	msec	*Default value:*	30
Value range:	Full scale.		

The corresponding bit 10 target reached is set in the statusword when the difference between the target velocity and the velocity actual value is within the velocity window longer than the velocity window time.

Object:	parDevCtrl.VelocityThresholdWindow	Velocity threshold window.	
Modbus IPA:	29031	*Modbus Data Type:*	Signed32
CANopen/COE index:	606Fh.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, pdomap		
Unit:	Velocity factor unit.	*Default value:*	170891 (~ 2.09 rad/s)
Value range:	Full scale.		

As soon as the velocity actual value exceeds the velocity threshold than the velocity threshold time bit 12 is reset in the statusword. Below this threshold the bit is set and indicates that the axle is stationary.

Object:	parDevCtrl.VelocityThresholdTimeout	Velocity threshold timeout.	
Modbus IPA:	29032	*Modbus Data Type:*	Signed16
CANopen/COE index:	6070h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, pdomap		
Unit:	msec	*Default value:*	80
Value range:	Full scale.		

The velocity threshold time. For further information refer to the Profile Velocity Mode specification (CiA DSP402 V2.0 (§3).

Object:	wksTargetSpeed	Target velocity.	
Modbus IPA:	27118	*Modbus Data Type:*	Signed32
CANopen/COE index:	60FFh.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw, pdomap		
Unit:	Velocity factor unit.	*Default value:*	0
Value range:	Full scale.		

User target velocity: the value from this object is used only if the drive is set in velocity mode.
The target velocity is the input for the internal ramp generator:

- when the AxM-II drive works in profile position mode, the target velocity is automatically calculate by the internal ramp generator on the basis of the positioning parameters;
- when the AxM-II drive works in velocity profile the target velocity is generated from the master control that should write this object.

Anyway the user should be configuring it using the Cockpit configurator and the base application of the AxM-II. The base application allows the user to control motor's current and speed like needed in the usual classic drive control, it turns AxM-II in a versatile brushless servomotors digital control.

7.3 Interpolation mode.

The interpolated position mode is used to control multiple coordinated axes or a single axle with the need for time-interpolation of set-point data.
For synchronous operation the interpolation cycle time is defined by the object interpolation time period. Time synchronization may be done by networked dependent mechanisms. Each synchronization cycle actuates the next data record if a valid data record is available.

This object shall indicates the configured interpolation cycle time. The interpolation time period (60C2 sub-index 01h) value shall be given in $10^{\text{interpolation time index}}$ s(second). The interpolation time index (60C2 sub-index 02h) shall be dimensionless.

$$\text{time period} = \text{time units} \cdot 10^{\text{interpolation time index}}$$

The interpolation time period has to be multiple of 250µs.
For further information's refer to CiA DSP402 V2.0 (§3).

Object:	n/a	Interpolation data record	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	60C1h.1h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

The interpolation data record is the data words, which are necessary to perform the interpolation algorithm. For the linear interpolation mode each interpolation data record simply is regarded as a new position set-point. Those set-points could be optionally filtered by a user-defined 2nd order filter.

Object:	parDevCtrl.IPTimeUnits	Interpolation time units.	
Modbus IPA:	29033	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	60C2h.1h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw		
Unit:	n/a	*Default value:*	1
Value range:	Full scale.		

Interpolation time units.

Object:	parDevCtrl.IPTimeIndex	Interpolation time index.	
Modbus IPA:	29034	*Modbus Data Type:*	Signed16
CANopen/COE index:	60C2h.2h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw		
Unit:	n/a	*Default value:*	-3
Value range:	-128 to 63		

Interpolation time index.

Object:	wksIPQuotaLo	Interpolation data value angle unit.	
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position Factor Unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the interpolation data position in mechanical angle unit.
The interpolation data in angle unit is the 32 LSB bit of 64 bit interpolation data position. For more details about the format position refer to the appendix Encoder position format or to the manual (§4).

Object:	wksIPQuotaHi	Interpolation data value turns unit.	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--

Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the interpolation data encoder position in mechanical turns unit.
The interpolation data in turns unit is the 32 MSB bit of 64 bit interpolation data position. For more details about the format position refer to the appendix Encoder position format or to the manual (§4).

8. FIELDBUS

8.1 Fieldbus CANOpen

CAN is an abbreviation for the Controller Area Network. CANOpen is a higher layer protocol based on CAN (Controller Area Network), which enables the communication between devices of different manufacturers and guarantees an interchange ability of devices.
Thus, devices of different manufacturers can be accessed via the bus in exactly the same manner. In this way, a very high degree of vendor independence is achieved as the devices are interoperable and exchangeable.
CANOpen is a high speed serial interface which was designed for use in Automotive and Industrial applications. AxM-II drive is configurable as CANOpen standard slave node and it's possible select one from some CANOpen standard profiles.
For further information about CANOpen standard definition refer to (§1) and (§3).

Object:	parCAN.Controller	Can controller configuration.	
Modbus IPA:	30000	*Modbus Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0000h (Disabled)
Value range:	See table above.		

CAN controller configuration.
AxM-II drive has two configurable CAN channel.

Option code	*Description*	*Name*
0000h	Disable CAN controller.	--
0001h	Main Controller: AxM-II drive connector C1.	cstCanNode0
0002h	Aux Controller: AxM-II drive connector S1.	cstCanNode1

Object:	parCANOpen.LssNodeId	Lss Node ID	
Modbus IPA:	30001	*Modbus Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	1
Value range:	1 to 127.		

The node identifier (node ID) within a CANOpen network is set with this parameter. Each node within the network (drives, peripherals …) needs its own unique number.

WARNING: Do not assign the same node ID address to more than one slaves on the net!

Object:	parCANOpen.LssTimingIndex	Lss Timing Index	
Modbus IPA:	30002	*Modbus Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0004h (125Kbps)
Value range:	See table above.		

Baudrate CAN communication protocol.

Description	*Option code*	*Name Mask*
1Mbps	0000h	cstCanBaudRate1000Kbps
800kbps	0001h	cstCanBaudRate800Kbps
500kbps	0002h	cstCanBaudRate500Kbps
250kbps	0003h	cstCanBaudRate250Kbps
125Kbps	0004h	cstCanBaudRate125Kbps
100Kbps	0005h	cstCanBaudRate100Kbps
50Kbps	0006h	cstCanBaudRate50Kbps
20Kbps	0007h	cstCanBaudRate20Kbps
10Kbps	0008h	cstCanBaudRate10Kbps

Object:	parCANOpen.EmcyInhibitTime	Emergency Inhibit Time	
Modbus IPA:	30003	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1015h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, retain		
Unit:	100 µsec	*Default value:*	0
Value range:	Full scale.		

The inhibit time for the EMCY can be adjusted via this entry. To guarantee that no starvation on the network occurs for data objects with low priorities, data objects can be assigned an inhibit time; this defines the minimum time that has to elapse between two consecutive invocations of a transmission service for that data object. It could use only in asynchronous transmission type.
The value shall be given in multiples of 100 µs. The value zero disable the inhibit time.

Object:	parCANOpen.SyncCOB	Sync COB ID	
Modbus IPA:	30004	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0080h
Value range:	001h to 57Fh, 680h to 7FFh		

COB-ID used by Sync COB message.

Object:	parCANOpen.EmgyCOB	Emgy COB ID	
Modbus IPA:	30005	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1014h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0080h + Node ID
Value range:	n/a		

COB-ID used by Emergency COB message.
Defines the COB-ID of the EMCY. Bits 0-10 define the COB-ID, bit 31 defines if the EMCY is enabled (equal to 0) or if it is disabled (equal to 1); bits 11-30 should be leaved 0.

MSB		LSB
E	Unused (20 bit), should be 0	COB-ID (11 bit)

Figure 6 - Structure of COB-ID Emergency Message

Object:	parCANOpen.GuardTime	Guard Time.	
Modbus IPA:	30006	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	100Ch.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, retain		
Unit:	msec	*Default value:*	0
Value range:	4 to 32000		

Node guarding protocol: the node guard is used to detect communication errors in the network. With node guarding, a certain network node (NMT-master) requests the other nodes in the network with a CAN remote frame one after the other at defined intervals (guard time) to transmit a data telegram with its current communication state (stopped, operational, pre-operational) together with a toggle-bit. If the NMT Slave has not been polled during its life time, it issues an EMCY object with error code 8130h and then the action indicated in the Abort Connection (object 6007h.0h) is issued. The error is cleared either restarting polling slave or by a reset node / reset communication command. The value zero disables the guard time features.

Object:	parCANOpen.HeartbeatTime	Hearbeat Time.	
Modbus IPA:	30007	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1017h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, retain		
Unit:	msec	*Default value:*	0
Value range:	4 to 32000.		

Producer Heartbeat time.
With node monitoring according to the heartbeat principle, a node automatically transmits its communication state at regular intervals as evidence of its communication ability. The interval between two heartbeat messages (heartbeat interval) of a heartbeat producer is configured via this object.
The value shall be given in multiples of 1 ms. The value zero shall disable the producer heartbeat.

Object:	parCANOpen.LifeTimeFactor	Life Time Factor.	
Modbus IPA:	30008	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	100Dh.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0
Value range:	Full scale.		

Life time factor: multiplied with the guard time gives the life time for the Node Guarding Protocol. Set it to zero if not used.

Object:	parCANOpen.CommCyclePeriod	Communication cycle period.	
Modbus IPA:	30009	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1006h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, retain		
Unit:	µs	*Default value:*	0000 0000h
Value range:	Full scale.		

CAN protocol communication cycle period. This object defines the Sync interval in µs.
The value shall be given in multiple of µs. If the value is set to 0000 0000h the transmission of Sync messages shall be disabled. By changing the value from 0000 0000h and the synchronous counter overflow value is greater than 0 the first Sync message shall start with the counter value reset to 1.
The transmission of Sync messages shall start within one communication cycle period as given by the value after it is set to the new value.

Object:	parCAN.DisableAlarmMask	Can disable alarm mask.	
Modbus IPA:	30010	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	5730h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, retain, pdomap		
Unit:	n/a	*Default value:*	0000 0000h
Value range:	Full scale.		

If the value is different to 0000 0000h the alarm messages generated by the CAN protocol are disabled.

Object:	parCAN.Flags.ReSyncEnable	Synchronization Enable.	
Modbus IPA:	30011	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off) -1(On)		

Enable the internal machine cycle synchronization with the Sync object.

8.1.1 CANOpen RxPDO Communication Parameters

The purpose of this data structure is to define the communication parameters for all RPDO; for each RPDO exist one object, the object index range from 1400h (RPDO #1) to 1407h (RPDO #8).

Object:	n/a	COB-ID used by PDO	
Modbus IPA:	30101	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1400h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw		
Unit:	n/a	*Default value:*	NODEID+40000200h
Value range:	n/a		

Define the COB-ID and the state (enabled/disabled) of the RPDO.

Object:	n/a	RPDO transmission type	
Modbus IPA:	30102	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1400h.2h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This field defines the transmission type of RPDO and when received data should be used.

Transmission type	*cyclic*	*acyclic*	*synchronous*	*asynchronous*
0		X	X	
1-240	X		X	
255				X

For further information on RPDOs refer to (§3).

8.1.2 CANOpen RxPDO Mapping Parameters

The purpose of this data structure is to define the data mapping for all RPDO; for each RPDO exist one object, the object index range from 1600h (RPDO #1) to 1607h (RPDO #8).

Object:	n/a	Number of object mapped	
Modbus IPA:	30200	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1600h.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw		
Unit:	n/a	*Default value:*	0
Value range:	n/a		

Number of mapped object.

Object:	n/a	PDO Mapping	
Modbus IPA:	30201 - 30208	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1600h.1h – 1600h.8h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw		
Unit:	n/a	*Default value:*	
Value range:	n/a		

These entries describe the PDO contents by their index, sub-index and length.
For further information on RxPDO mapping refer to (§3).

8.1.3 CANOpen TxPDO Communication Parameters

The purpose of this data structure is to define the communication parameters for all RPDO; for each RPDO exist one object, the object index range from 1800h (RPDO #1) to 1807h (RPDO #8).

Object:	n/a	COB-ID used by PDO	
Modbus IPA:	30301	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1800h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

Define the COB-ID and the state (enabled/disabled) of the TPDO. Bits 0-10 define the COB-ID, bit 31 defines if the PDO is enabled (equal to 0) or if it is disabled (equal to 1); bit 30 defines if RTR is allowed (equal to 0) or not (equal to 1) on this PDO; bits 11-29 should be leaved 0. COB-ID have to be defined between 181h and 57Fh.

MSB			LSB
E	R	Unused (19 bit), should be 0	COB-ID (11 bit)

Figure 7 - **Structure of TPDO's COB**-ID

Object:	n/a	TPDO transmission type	
Modbus IPA:	30302	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1800h.2h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This field defines the transmission type of TPDO and when the data should be transmitted.

Transmission type	*cyclic*	*acyclic*	*synchronous*	*asynchronous*	*RTR only*
0		X	X		
1-240	X		X		
252			X		X
253				X	X
254				X	
255				X	

Object:	n/a	Inhibit Time	
Modbus IPA:	30303	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1800h.3h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw		
Unit:	100 µsec	*Default value:*	0
Value range:	Full scale.		

This defines the minimum time that has to elapse between two consecutive invocations of a transmission service for the TPDO. It is possible to set this object only for asynchronous TPDO.

Object:	n/a	Sync Start Value	
Modbus IPA:	30306	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1800h.6h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw		
Unit:	n/a	*Default value:*	0
Value range:	0 to 240		

Sub-index 06h contains the Sync start value. The Sync start value of 0 shall indicate that the counter of the Sync message shall not be processed for this PDO. The Sync start value 1 to 240 shall indicate that the counter of the Sync message shall be processed for this PDO. In case the counter of the Sync message is not enabled sub-index 06h shall be ignored. The Sync message of which the counter value equals the Sync start value shall be regarded as the first received Sync message. The value shall not be changed while the PDO exists (bit 31 of sub-index 01h is set to 0b).

8.1.4 CANOpen TxPDO Mapping Parameters

The purpose of this data structure is to define the data mapping for all RPDO; for each RPDO exist one object, the object index range from 1A00h (RPDO #1) to 1A07h (RPDO #8).

Object:	n/a	Number of object mapped	
Modbus IPA:	30400	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	1A00h.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

Number of mapped object.

Object:	n/a	PDO Mapping	
Modbus IPA:	30401 - 30408	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1A00h.1h – 1A00h.8h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

These entries describe the PDO contents by their index, sub-index and length.
For further information on TxPDO mapping refer to (§3).

8.2 Fieldbus Serial Link

Modbus Protocol is a messaging structure developed by Modicon in 1979. It is used to establish master-slave/client-server communication between intelligent devices. It is a de facto standard, truly open and the most widely used network protocol in the industrial manufacturing environment. The Modbus protocol defines how a master device polls one or more slave devices to read and write data in real time by means of RS232, RS422, or RS485 serial data communication.

Object:	parSerialZeroBaseAddress	Standard use for IPA numbering.	
Modbus IPA:	18020	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5720h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	1 (JBus)
Value range:	0 (Modbus) – 1 (JBus)		

Use:

- JBUS when the serial link is made by the "Cockpit configuration tool".
- MODBUS using a standard Modbus keyboard

JBus and Modbus are very similar; there is only a slight variance in the addressing scheme (J-bus registers start at address xxxx1, Modbus addressing starts at address xxxx0. They support the same functions (as applied by the manufacturer of your slave) and the same configuration.
The serial port default configuration is: 38400,8,N,1.

Object:	parSerialAnswerToBCast	Answer to broadcast request.	
Modbus IPA:	18029	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5721h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	1(Enabled).
Value range:	0 (Disabled)-1(Enabled)		

With this flag enabled the AxM-II drive answer to broadcast request.

Object:	parSerialBaudrate	Serial baudrate.	
Modbus IPA:	18031	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	5722h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	38400
Value range:	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 128000, 230400, 256000, 460800, 512000.		

Baudrate selection.

Object:	parSerialDataBits	Serial data bits.	
Modbus IPA:	18032	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5723h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	8
Value range:	7, 8.		

Number data bits selection.

Object:	parSerialParity	Serial parity.	
Modbus IPA:	18033	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5724h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	None
Value range:	None(0), Odd(1), Even(2).		

Parity selection.

Object:	parSerialStopBits	Serial stop bits.	
Modbus IPA:	18034	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5725h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	1
Value range:	1, 2.		

Number stop bits.

Object:	parSerialDuplex	Serial duplex selection.	
Modbus IPA:	18035	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5726h.0h	*CANopen/COE Data Type:*	Signed8
Attributes:	rw,retain		
Unit:	n/a	*Default value:*	0(Full)
Value range:	0(Full) -1(Half)		

Full duplex communication means that a device can receive and transmit data at the same time. Half duplex means that the device cannot send and receive at the same time.

Object:	parSerialEndDelay	Serial End delay.	
Modbus IPA:	18036	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5727h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	µsec	*Default value:*	0
Value range:	Full scale.		

Delay after complete transmission frame.

Object:	parSerialRxToTxDelay	Serial Rx-Tx delay.	
Modbus IPA:	18037	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5728h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw, reatin		

Unit:	µsec	*Default value:*	0
Value range:	Full scale.		

Delay between a complete receive frame and begin new frame transmission.
This delay should be useful to the user to avoid possible conflicts on the serial line.

Object:	parSerialSlaveAddress	Serial slave address.	
Modbus IPA:	18070	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	5729h.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	rw		
Unit:	n/a	*Default value:*	0
Value range:	0 to 127.		

Slave address (0 is used as broadcast).

WARNING: If one of these parameters has been changed it is necessary to align the setting of the Cockpit Configurator (menu *Target - Communication settings*) and the LogicLab environment (menu: *Communication - settings*) to avoid problems communication with the AxM-II drive. For more details refer to (§4) and to (§7).

8.3 Fieldbus EtherCAT

The EtherCAT technology overcomes the system limitations of other Ethernet solutions: the Ethernet packet is no longer received, then interpreted and copied as process data at every connection. Instead, the Ethernet frame is processed on the fly: the newly developed FMMU (fieldbus memory management unit) in each slave node reads the data addressed to it, while the telegram is forwarded to the next device. Similarly, input data is inserted while the telegram passes through. The telegrams are only delayed by a few nanoseconds. EtherCAT can provide the same communication mechanisms as the familiar CANOpen mechanisms: object dictionary, PDO (process data objects) and SDO (service data objects) - even the network management is comparable. For more details refer to (§6).
AxM-II parameters for EtherCAT configuration are show below.

Object:	parECAT.Flags.EnableModule	Enable EtherCAT module	
Modbus IPA:	30800	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0(Off)
Value range:	0(Off)-1(On)		

Enable EtherCAT software module.

Object:	parECAT.Flags.ReSyncEnable	EtherCAT Enable Sync	
Modbus IPA:	30801	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	1(On)
Value range:	0(Off)-1(On)		

Enable the internal machine cycle synchronization with the EtherCAT fieldbus.

8.4 Fieldbus Sync Manager

The AxM-II monitor continuously the time period of the Sync object, giving the user the ability to have a feedback on the quality of the Sync object; this is given in the form of three parameters, the minimum cycle time, the maximum cycle time and the average cycle time. Those parameters are updated every user specified amount of time (default 500 msec), giving back the cycle time quality of the past period and letting the user never miss any discontinuity of the Sync (e.g. missing transmission of Sync objects).

Object:	parSyncMgr.TSFIlter	Time constant filter	
Modbus IPA:	31000	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw,retain		

Unit:	n/a	*Default value:*	64000
Value range:	Full scale.		

Time constant of the filter applied to the Sync signal. If the value is set to zero the filter is disabled.

Object:	parSyncMgr.ReSyncDelta	Delta for sync signal	
Modbus IPA:	31001	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	nsec	*Default value:*	110184
Value range:	Full scale.		

Time shift used to re-calculated the centring value of the Sync signal.

Object:	parSyncMgr.PeakNDiscard	Discard number	
Modbus IPA:	31004	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	16
Value range:	Full scale.		

Maximum number of Sync signal consecutive sample that can be discarded. When the number of Sync signal discard is greater than this number a CAN alarm is generated.

Object:	parSyncMgr.PeakThreshold	Peak threshold	
Modbus IPA:	31005	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	8192
Value range:	Full scale		

Threshold for Sync signal peak detection. When the actual value of Sync signal is lower than this value a Sync signal estimated is considered a discard sample.

8.4.1 Fieldbus > Sync Manager > Monitor

Object:	varSyncMgr_SyncTime		
Modbus IPA:	31010	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	nsec	*Default value:*	n/a
Value range:	Full scale.		

Period of the Sync signal.

Object:	varSyncMgr_SyncMax		
Modbus IPA:	31011	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	nsec	*Default value:*	n/a
Value range:	Full scale.		

Maximum Sync signal period measured.

Object:	varSyncMgr_SyncMin		
Modbus IPA:	31012	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	nsec	*Default value:*	n/a
Value range:	Full scale.		

Minimum Sync signal period measured.

Object:	varSyncMgr_Valid		
Modbus IPA:	31013	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	0(Off) -1(On)		

Sync signal is correct and the internal machine cycle is synchronized with the master unit.

9. SYSTEM

9.1 System Brake Unit

Variables to set brake resistor behaviour and dc bus link voltage limit. These parameters are saved into Eeprom chip memory of the AxM-II power stage board.
The AxM-II drive can be equipped, depending on the model size, with an internal or external regeneration resistor. The algorithm controlling the duty cycle of the resistor ensures that the power dissipated by the resistor is maintained at the resistor rating provided by the user.
The recommended regeneration resistors are detailed in (§4).

Object:	parPStage.VBrakeLow	Vdc Brake Low Threshold.	
Modbus IPA:	27400	*Modbus Data Type:*	Signed16
CANopen/COE index:	3140h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw, pdomap		
Unit:	100 mV	*Default value:*	7200
Value range:	0 to max operative limits.		

Set the threshold to deactivate the recovery circuit mechanism.
If the value is set lower to zero an alarm Parameter value out of range is generated.
If the value is set to zero a default value will be used.

Object:	parPStage.VBrakeHigh	Vdc Brake High Threshold.	
Modbus IPA:	27401	*Modbus Data Type:*	Signed16
CANopen/COE index:	3141h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw, pdomap		
Unit:	100 mV	*Default value:*	7500
Value range:	0 to max operative limits.		

Set the threshold to activate the recovery circuit mechanism.
If the value is set lower to zero or lower to parPStage.VBrakeLow an alarm Parameter value out of range is generated.
If the value is set to zero a default value will be used.

Object:	parPStage.UndervoltageThreshold	Undervoltage threshold.	
Modbus IPA:	27402	*Modbus Data Type:*	Signed16
CANopen/COE index:	3142h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw, pdomap		
Unit:	100 mV	*Default value:*	0
Value range:	0 to max operative limits.		

This parameter defines the undervoltage threshold. If the dc bus voltage is lower this value if the drive is disabled a warning is generated, otherwise an Undervoltage alarm is generated.

Object:	parPStage.OvervoltageThreshold	Overvoltage threshold.	
Modbus IPA:	27403	*Modbus Data Type:*	Signed16
CANopen/COE index:	3143h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw, pdomap		
Unit:	100 mV	*Default value:*	9500
Value range:	0 to max operative limits.		

Over Voltage threshold is the maximum dc bus voltage value that the AxM-II drive can reach. The user can modify the Over voltage threshold detection, but it's possible only to set a value lower than the hardware AxM-II power stage.
If the dc bus voltage is greater than this value an alarm Overvoltage is generated

WARNING: If this value is set to zero, then the default hardware drive's overvoltage limit is restored. This change takes effect only after the *Save parameters* command and *Reset* of the drive.

Object:	parThermalModel.BrakeResistorValue	Brake resistor value.	
Modbus IPA:	28100	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3220h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	100 mΩ	*Default value:*	0
Value range:	Full scale.		

The default value matches to the AxM-II drive internal resistance. This value is used to calculate the brake power consumption.

WARNING: If an external brake resistor is used, it's necessary to set here the own value.

Object:	parThermalModel.BrakeResistorPower	Brake resistor power.	
Modbus IPA:	28101	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	3221h.0h	*CANopen/COE Data Type:*	Unsigned16
Attributes:	rw,retain		
Unit:	100 mW	*Default value:*	0
Value range:	Full scale.		

Maximum power wasted by the brake resistor. Defines the edge limit of the Brake Error alarm.

WARNING: The internal resistor max power depends on the AxM-II drive model. The internal braking resistor is able to dissipate a maximum of 10W braking power. If the power dissipation is higher, it becomes necessary the use of an external resistor. Refer to (§4) for the connection of the external resistor.

Object:	parThermalModel.BrakeResistorEnergy	Brake resistor energy.	
Modbus IPA:	28107	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	3225h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw,retain		
Unit:	Joule	*Default value:*	0
Value range:	Full scale.		

Maximum brake resistor energy.

9.1.1 Brake Unit Monitor

Object:	varPStageVdcBus	Dc bus voltage.	
Modbus IPA:	27440	*Modbus Data Type:*	Signed16
CANopen/COE index:	3104h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		

Unit:	100 mV	*Default value:*	n/a
Value range:	Full scale.		

Dc bus link actual voltage.

Object:	varThModelRBrakePower	Power dissipated.	
Modbus IPA:	28167	*Modbus Data Type:*	Signed32
CANopen/COE index:	3207h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	100 mW	*Default value:*	n/a
Value range:	Full scale.		

Actual power dissipated by the brake resistor.

Object:	varThModelRBrakeEnergy	Energy dissipated.	
Modbus IPA:	28179	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	320Ah.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	Joule	*Default value:*	n/a
Value range:	Full scale.		

Actual energy dissipated by the brake resistor.

9.2 System Thermal Model

Limit for thermal model functionalities.

Object:	parThermalModel.CoolingTempOn	On cooling temperature.	
Modbus IPA:	28104	*Modbus Data Type:*	Signed16
CANopen/COE index:	3222h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	100 m°C	*Default value:*	500
Value range:	Full scale.		

Temperature to turn on the bridge fan: e.g. 50°C.

WARNING: The value parameter has to be greater parThermalModel.CoolingTempOff value; otherwise an alarm Parameter value out of range is generated.

Object:	parThermalModel.CoolingTempOff	Off cooling temperature.	
Modbus IPA:	28105	*Modbus Data Type:*	Signed16
CANopen/COE index:	3223h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,reatin,pdomap		
Unit:	100 m°C	*Default value:*	450
Value range:	Full scale.		

Temperature to turn off the bridge fan: e.g. 45°C.

Object:	parThermalModel.MotorOverTemp	Motor overtemperature.	
Modbus IPA:	28106	*Modbus Data Type:*	Signed16
CANopen/COE index:	3224h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	rw,retain,pdomap		
Unit:	100 m°C	*Default value:*	1500
Value range:	Full scale.		

Motor temperature threshold. When motor temperature it's greater than this limit a Motor overtemperature alarm is generated.

9.2.1 Thermal Model Monitor

Object:	varThModelTNtc	Igbt NTC temperature.	
Modbus IPA:	28154	*Modbus Data Type:*	Signed16
CANopen/COE index:	3200h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		
Unit:	100m°C	*Default value:*	n/a
Value range:	Full scale.		

Igbt temperature measured by NTC sensor placed into the Igbt power block.

Object:	varThModelTjMax	Temperature junction.	
Modbus IPA:	28166	*Modbus Data Type:*	Signed16
CANopen/COE index:	3201h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		
Unit:	100m°C	*Default value:*	n/a
Value range:	Full scale.		

Igbt maximum temperature junction measured by the thermal model.

Object:	varThModelTHeatSink	Heatsink temperature.	
Modbus IPA:	28168	*Modbus Data Type:*	Signed16
CANopen/COE index:	3208h.0h	*CANopen/COE Data Type:*	Signed16
Attributes:	ro, pdomap		
Unit:	100m°C	*Default value:*	n/a
Value range:	Full scale.		

Heatsink temperature measured by the thermal model.

Object:	varThModelIqLimMax	Iq Limit Max.	
Modbus IPA:	28169	*Modbus Data Type:*	Signed32
CANopen/COE index:	3203h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Maximum allowed quadrature current calculated by the thermal model.

Object:	varThModelIqLimMin	Iq Limit Min.	
Modbus IPA:	28170	*Modbus Data Type:*	Signed32
CANopen/COE index:	3204h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Minimum allowed quadrature current calculated by the thermal model.

Object:	varThModelIdLimMax	Id Limit Max.	
Modbus IPA:	28171	*Modbus Data Type:*	Signed32
CANopen/COE index:	3205h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Maximum allowed direct current calculated by the thermal model.

Object:	varThModelIdLimMin	Id Limit Min.	
Modbus IPA:	28172	*Modbus Data Type:*	Signed32
CANopen/COE index:	3206h.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Minimum allowed direct current calculated by the thermal model.

Object:	varThModelTMotor	Motor Temperature.	
Modbus IPA:	28174	*Modbus Data Type:*	Real32
CANopen/COE index:	3209h.0h	*CANopen/COE Data Type:*	Real32
Attributes:	ro, pdomap		
Unit:	°C	*Default value:*	n/a
Value range:	Full scale.		

Motor temperature actual value.
When this value is greater than motor over temperature limit (parThermalModel.MotorOverTemp) a Motor overtemperature alarm is generated.

WARNING: This parameter can be used only with motor equipped by NTC and KTY sensor type. In case of PTC sensor this value doesn't have a reliable value.

10. ADDITIONAL PARAMETERS

The following objects are not present in the menu selection of the Cockpit configurator but could be used by the user. They are collected in CANOpen dictionary and are accessible through CANOpen SDO protocol. Moreover some of these objects are also connected to the window Target Information activated by the menu *Service* of the Cockpit configurator.

10.1 CANOpen additional parameters

Object:	n/a	Identity Device type.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1000h.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	0002 0192h
Value range:	n/a		

Describes the type of device and its functionality. It is composed of a 16-bit (LSB) field, which describes the device profile that is used, and a second 16-bit (MSB) field, which gives additional information about optional functionality of the device. In this case the device profile is 402 (0192h) and the additional information indicate that is a servo drive (0002h).

MSB	LSB
Additional information (16 bit)	Device profile type (16 bit)

Figure 8 - Structure of Device Type

Object:	n/a	CiA301 Errors register.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1001h.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This object is an error register for the AxM-II drive. It is a part of the EMCY object.
Every bit in the error register refer to a specific faults of the motion controller; more than one bit at time could be set to 1, meaning that more than one fault is active.

Bit	*Meaning*
0	generic error
1	current
2	voltage
3	temperature
4	communication error (overrun, error state)
5	device profile specific
7	manufacturer specific

Table 4 - Error register reference

Object:	n/a	CiA301 Vendor ID.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1018h.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This is a unique value assigned to each manufacturer by CIA, in this case for Phase Motion Control is 0000 00D9h.

Object:	n/a	Product Code.	
Modbus IPA:	--	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1018h.2h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This is the product code of the device.
This parameter is not present in table parameter, but it's available using the Target Information service from Cockpit configurator.

Object:	n/a	Revision Number.	
Modbus IPA:	--	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1018h.3h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This is the firmware release number, with the subfields structured as follow:

MSB		LSB
major (16 bit)	mid (8 bit)	minor (8 bit)

Figure 9 - Structure of revision number

These parameters are not present in table parameter, but they're available using the Target Information service from Cockpit Configurator.

Object:	n/a	Serial number.	
Modbus IPA:	--	*Modbus Data Type:*	Unsigned32
CANopen/COE index:	1018h.4h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This is the serial number of the drive; the same appears on the plate of the AxM-II drive.
This parameter is not present in table parameter, but it's available using the Target Information service from Cockpit configurator.

Object:	n/a	Manufacturer Status Register	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1002.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro, pdomap		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This is the common status register specific for the manufacturer. It is a part of the EMCY object.

Object:	n/a	COB-ID Sync Message	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1005.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw, retain		
Unit:	n/a	*Default value:*	0000 0080h
Value range:	n/a		

Defines the COB-ID of the Synchronization Object.
Bits 0-10 define the COB-ID, bits 11-31 should be leaved 0.

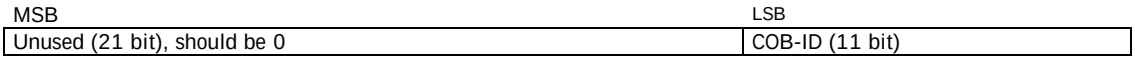

Figure 10 - Structure of COB-ID Sync Message

Object:	n/a	Manufacturer Device Name	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1008.0h	*CANopen/COE Data Type:*	string
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

Contain the application name of the code currently using by the AxM-II drive.

Object:	varSysFirmwareInfo	Manufacturer Software version.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	100A.0h	*CANopen/COE Data Type:*	visible_string
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

Contain the software release number and release date.
This parameter is a data structure composed by different fields, and it's not present in table parameter.
Its members are avalaible using the Target Information service from Cockpit configurator.

Description:	Application type.	*Default Value:*	1
	Major version.		1
	Minor version.		5
	Incremental build number version.		n/a
	Version date.		(YYYYMMDD).
	Version description.		AxX System Application

Object:	wksSysSaveParameters	Store Parameters	
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	1010.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This object let the drive to save all parameters in non-volatile memory. By read access the device provides information about its saving capabilities.

Sub-index:	0h	Large sub-index supported	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1010.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

The large sub-index supported for this object, in this case 1.

Sub-index:	1h	Store All	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1010.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This command let the drive store all parameters that have the attribute NV storage. In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-Index. The signature is the string save (or the 32 bit number 6576 6173h). On read the drive provides information about its storage functionality, in this case storage is executed only on command, not autonomously.

Object:	wksSysRestoreParameters	Store Parameters	
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	1011.0h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

With this object the default values of parameters according to the communication or device profile are restored.

Sub-index:	0h	Large sub-index supported	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1011.0h	*CANopen/COE Data Type:*	Unsigned8
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

The large sub-index supported for this object, in this case 1.

Sub-index:	1h	Restore all defaults	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	1011.1h	*CANopen/COE Data Type:*	Unsigned32
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This command let the drive restore all parameters to the factory settings. In order to avoid restore of parameters by mistake, restore is only executed when a specific signature is written to the appropriate sub-Index. The signature is the string load (or the 32 bit number 6461 6F6Ch). This command has to be completed by issuing a reset command.

10.2 Position additional parameters

Object:	n/a	Target Position	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	607A.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	rw, pdomap		
Unit:	Position factor unit	*Default value:*	n/a
Value range:	Full scale		

This the 32 bit target position used in positioner mode of operation.
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc.
At start-up the content is unforeseeable, and then the first positioning should be only absolute.

Object:	n/a	Position actual value	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	6064.0h	*CANopen/COE Data Type:*	Signed32
Attributes:	ro, pdomap		
Unit:	Position factor unit	*Default value:*	n/a
Value range:	Full scale		

This object represents the actual value of the position measurement device, normalized with home offset and polarized with the direction object.

Object:	n/a	Target position 64 bit	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3000.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	rw, pdomap		
Unit:	Position factor unit	*Default value:*	n/a
Value range:	Full scale		

This the 64 bit target position used in positioner mode of operation.
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc.
At start-up the content is unforeseeable, and then the first positioning should be only absolute.

Object:	n/a	Demand Position.	
Modbus IPA:	--	*Modbus Data Type:*	--
CANopen/COE index:	3356.0h	*CANopen/COE Data Type:*	Signed64
Attributes:	ro, pdomap		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale		

Position output from positioner software module, this is the reference position for the space speed control loop.

10.3 System additional parameter

These variables are used to reset the drive and to read alarms and warnings.

Object:	wksSysReset	System reset	
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0(False)
Value range:	0 (False), 1 (True).		

Flag to command system reset.

- 0(False): no system reset.
- 1(True): enable system reset.

Object:	varSysAlarms	System alarms	
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	Full scale.		

Active alarm mask.
For more details refer to (§C).

Object:	varSysAlarmSubCode	System alarms subcode	
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	ro		
Unit:	n/a	*Default value:*	n/a
Value range:	Full scale.		

Last alarms subcode.
For more details refer to (§C).

10.4 Physical Digital/Analog Input/Output

These variables are used to set the status of digital output and analog output.
For electrical specification of digital and analog input/output refer to (§4).

Object:	wksIODigOutputs	Digital Outputs signal.	
Modbus IPA:	--	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Array of8 digital outputs signal. The hardware digital output is physically four channels.
Their value has source to 0-3 array channels; while 4-7 virtual channels are virtual channels handled by Control Panel Logical Outputs from the Cockpit configurator.
Wks variable to command digital state of output signal.

- 0(False): digital output value is low.
- 1(True): digital output value is high.

Object:	wksIOAnaOutputs	Analog Outputs signal.	
Modbus IPA:	--	*Modbus Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	mV	*Default value:*	0
Value range:	0 -10000		

Array of 4 analog outputs signals. The hardware analog outputs are physically two channels.
Their value has source to 0-1 array channels; while 2-3 array channels are virtual channels handled by Control Panel Logical Outputs from the Cockpit configurator.

Wks variable to set the value of analog signal outputs.

- value corresponds to the ADC scale appropriate [0-10000 means 0-10V].

Object:	varIODigInputs	Digital Inputs signal	
Modbus IPA:	--	*Modbus Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True).		

Array of 8 digital inputs signal. The hardware digital inputs are physically eight channels.
Their value can be source to 0-7 channels (D0-D7) or can be source to 0-7 array virtual channels handled by Control Panel Logical Outputs from the Cockpit configurator. For more detail refer to (§4).
Variable to read digital state of input signal.

- 0(False): digital input value is low.
- 1(True): digital input value is high.

Object:	varIOAnaInputs	Physical Analog Inputs	
Modbus IPA:	--	*Modbus Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	mV	*Default value:*	0
Value range:	± 10000		

Array of 3 analog inputs signal. The hardware analog inputs are physically two channels.
Their value can be source to 0-1 array channels or can be source to 2 array virtual channels handled by Control Panel Logical Outputs from the Cockpit configurator. For more detail refer to (§4).

Variables to read the value of analog signal inputs.

- value corresponds to the ADC scale appropriate [± 10000 means – ±10V].

11. PLC VARIABLES

Following tables describe the system variables available in the file *SysApp.tgt* divided for group identifier and data type. These system variables are also reported in LogicLab environment under *Target variables* tab of *Library tools* windows. The *Target variables* are the interface between the firmware and Plc application code.
Each group is divided in constant, parameters, monitor variables and working variables (WKS) used to control drive behaviour.
LogicLab is an IEC61131-3 Integrated Development Environment supporting the whole range of languages defined in the standard. Use the software development tool LogicLab in order to create a dedicated application for the AxM-II drives family. For more detail refer to (§4) or refer to (§7).

Basic data types used for accessing the object from Plc program application are:
par*: the parameter variables are data structures based on drive parameters.
cst*: the constant variable should be used by the user to write own program Plc application.
wks*: the working variables are used to set and control the AxM-II drive and the tasks behaviour.
var*: the monitor variables.

11.1 Globals Group

Globals Group data structure describes the electrical and mechanical features of the motor.

Object:	parMotorData		
IEC address:	%MB205.0	*LogicLab Type:*	PAR_MOTORDATA
Attributes:	rw		
Description:	Motor specifications.		
Structure members:	SerialNumber	*Description:*	Motor serial number (s/n).
	Productiondate		Production date: (YYYYMMDD).
	Model		Motor model.
	Resistance		Motor winding resistance [Ω].
	Inductance		Motor winding inductance [H].
	Kt		Motor constant torque [Nm/Arms].
	CurrentNominalZeroSpeed		Nominal current at zero speed [Arms].
	CurrentNominal		Motor nominal current [Arms].
	CurrentPeak		Motor peak current [Arms].
	SpeedNominal		Motor nominal speed [rad/s].
	ThermalConstant		Thermal constant in air [s].
	StatorInertia		Motor stator inertia [kg*m²].
	PhaseOffset		Electrical phase offset [0:65535=0:359electrical°].
	Type		Motor type.
	PoleNumbers		Motor poles number.
	CoolingTempOn		Start cooling at (if available) [100m°C].
	CoolingTempOff		Stop cooling at (if available) [100m°C].
	MaximumTemp		Maximum coil temperature [100m°C].
	DirectInductance		Synchronous or direct inductance [H].

These parameters should be memorized into the non volatile memory of the Endat encoder.

If there are not informations saved into Endat electronic plate, these parameters will assume the default value or their value saved in the non volatile memory of AxM-II drive.
For more details refer to parEncMgr.Flags.DisableEPlate.

11.2 PLC Group

Constants to set the task configuration of AxM-II drive.

Constant name	*Description*	*Data Type*	*Mask value*
cstPlcTaskCfgAuto	Task configuration default: handle by the firmware.	Unsigned8	00h
cstPlcTaskCfgOff	Task configuration disable: task is switched-off.	Unsigned8	01h
cstPlcTaskCfgPlc	Task configuration handles by PLC program.	Unsigned8	02h

Wks variables are used to set and control the AxM-II drive tasks behaviour.

Object:	wksPlcSlowTaskCopyOutImmediate		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Copy output data of slow task immediately at the end of execution time.

Object:	wksPlcPwrDisOnOverTime		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

If enabled, in case of task overtime all power outputs are immediately set in safe state.

Object:	wksPlcSlowTaskPeriod		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	msec	*Default value:*	8
Value range:	1 to 64		

Plc program slow task execution period. This variable has to be set during pre-boot task of Plc program.

Object:	wksPlcTaskCfg		
IEC address:	%MB1003.0	*LogicLab Type:*	DRVTSKCTRL_PLC_CONFIG
Attributes:	rw		
Description	Tasks configuration parameters		
Structure members:	ubRtHookPosition	*Description:*	Hook Position.
	ubModActiveFrontEnd		Active front-end task.
	ubModPiDcBusCntrlLp32bit		PI control loop on DC bus task.
	ubPositioner		Positioner task.
	ubSpaceSpeedCntrLp		Space speed control loop task.
	ubThermalModel		Thermal model task.
	ubMotorHandler		Motor handler task.
	ubMotionController		Motion controller task (CiA402 state machine and profile mode).
	ubActiveFrontEndController		Active front-end controller task.
	ubEncoderManager		Encoder manager task.
	ubEncFeedBackSelection		Encoder feedback selection task.

This variable contains the list of tasks available on the AxM-II drive and can be used during pre-boot task of Plc program. Its purpose is to define the input tasks configuration through the use of the constant Plc variables. The input task should be generated from Plc program or from firmware drive.
Plc program example code:

wksPlcTaskCfg.ubMotorHandler := cstPlcTaskCfgPlc ; task input managed by PLC program
wksPlcTaskCfg.ubMotionController := cstPlcTaskCfgAuto ; task input managed by drive firmware
wksPlcTaskCfg.ubEncoderManager := cstPlcTaskCfgOff ; the task is switched-off.

For more details about the AxM-II task manager configuration refer to (§4) and (§7).

11.3 Encoder Group

Constants to get the encoder Endat status information.

Constant name	*Description*	*Data Type*	*Mask value*
cstEndatSelMain	Endat encoder on main C1 connector.	Unsigned16	0001h
cstEndatSelAux	Endat encoder on Auxialiry E1 connector.	Unsigned16	0000h
cstEndatErrMaskNoModule	Endat encoder communication error.	Unsigned32	0100 0000h
cstEndatErrMaskTimeout	Timeout error on Endat protocol.	Unsigned32	8000 0000h
cstEndatErrMaskCRCError	CRC error on Endat protocol.	Unsigned32	4000 0000h

Constants to enable the field orientation procedure.

Constant name	*Description*	*Data Type*	*Mask value*
cstEncEfsTypeDisabled	Disable the electrical field orientation.	Unsigned8	00h
cstEncEfsTypeIdRamp	Enable the electrical field orientation.	Unsigned8	01h
cstEncEfsTypeIdRampPosition	Enable the electrical field orientation.	Unsigned8	02h

These data structures are referred to encoder main and auxiliary configuration.

Object:	parEncMEndat		
IEC address:	%MB401.0	*LogicLab Type:*	PAR_ENDAT
Attributes:	rw		
Description:	Main Encoder Endat parameters.		
Structure members:	ClockFreq	*Description:*	Clock frequency selector (KHz).
	MTurnStartPos		Offset for multiturn position at start-up.
	Dummy		Dummy parameter.

Object:	parEncAEndat		
IEC address:	%MB402.0	*LogicLab Type:*	PAR_ENDAT
Attributes:	rw		
Description:	Auxiliary Encoder Endat parameters.		
Structure members:	ClockFreq	*Description:*	Clock frequency selector (KHz).
	MTurnStartPos		Offset for multiturn position at start-up.
	Dummy		Dummy parameter.

Object:	parEncAn		
IEC address:	%MB403.0	*LogicLab Type:*	PAR_SINCOS
Attributes:	rw		
Description:	Encoder main: absolute encoder parameters.		
Structure members:	PoleCounts	*Description:*	Encoder number of poles.
	AnalogAlarmThreshold		SinCos Level Alarm threshold.
	Dummy		Dummy parameter.
Structure flags:	AnalogGain		If enabled, Angle = 360° - arctan (Sin/Cos), otherwise Angle = arctan (Sin/Cos).
	ReverseSignal		High for Sincos encoder, low for Resolver.

Object:	parEncMInc		
IEC address:	%MB405.0	*LogicLab Type:*	PAR_INC
Attributes:	rw		
Description	Encoder main: incremental traces.		
Structure members:	LineCounts	*Description:*	Encoder line counts per turn.
	AnalogAlarmThreshold		Analog Tracks Level Alarm Threshold.
	IndexErrorTolerance		Index Error tolerance.
	EnableAnalogInterp		Enable incremental analog tracks interpolation.
	DisableIndexError		Disable Index Error.
	DisableAnalogError		Disable Analog Tracks Levels Error.
	EnableIndexTrack		Enable Index Track.
	SwapTracks		Swap input tracks A and B.
	EnableStepDir		Enable step pulse (track A) and direction (track B) mode.
	EnableUpDown		Enable up pulse (track A) and down pulse (track B) mode.

Object:	parEncAInc		
IEC address:	%MB406.0	*LogicLab Type:*	PAR_INC
Attributes:	rw		
Description	Encoder auxiliary: incremental traces.		
Structure members:	LineCounts	*Description:*	Encoder line counts per turn.
	AnalogAlarmThreshold		Analog Tracks Level Alarm Threshold.
	IndexErrorTolerance		Index Error tolerance.
	EnableAnalogInterp		Enable incremental analog tracks interpolation.
	DisableIndexError		Disable Index Error.
	DisableAnalogError		Disable Analog Tracks Levels Error.
	EnableIndexTrack		Enable Index Track.
	SwapTracks		Swap input tracks A and B.
	EnableStepDir		Enable step pulse (track A) and direction (track B) mode.
	EnableUpDown		Enable up pulse (track A) and down pulse (track B) mode.

Object:	parEncHall		
IEC address:	%MB404.0	*LogicLab Type:*	PAR_HALL
Attributes:	rw		
Description	Encoder main: hall sensor type.		
Structure members:	Enable4Wire	*Description*:	4 wire type enable

Object:	parEncSimInc		
IEC address:	%MB408.0	*LogicLab Type:*	PAR_SIMINC
Attributes:	rw		
Description	Encoder auxiliary: encoder simulation (incremental traces).		
Structure members:	IndexLineCounts	*Description*:	Encoder line counts per turn
	IndexOffset		Analog Tracks Level Alarm Threshold
	Maxpositionerror tolerance		Index Error tolerance

Plc program application should generate the feedback encoder position.
Plc name variables: plcEncFbSet*[1]. The meaning of these variables are the same of IPA parameters: varEncFb*.

Plc program application should simulate an incremental encoder (TTL standard).
The encoder position, the electrical angle position and the other features necessary to the simulated incremental encoder have generated by the Plc program application.
Plc name variables: plcEncSim*.

Object:	plcEncSimSetMechLo		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the simulated encoder position in mechanical turns unit.

[1] *Symbol: denotes parameters have the same initial truncation name (e.g.: plcEncFbSetMechSpeed, varEncFbMechSpeed).

The simulated turns is the 32 MSB bit of 64 bit simulated encoder position. For more details about the format position refer to the manual (§4).

Object:	plcEncSimSetMechHi		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the simulated encoder position in mechanical angle unit.
The simulated turns is the 32 LSB bit of 64 bit simulated encoder position. For more details about the format position refer to the manual (§4).

Object:	plcEncSimSetMechSpeed		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This object represents the actual value of the velocity measurement device.

Object:	plcEncSimSetMechAccel		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This object represents the actual value of the acceleration.

Object:	plcEncSimSetElecAngle		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the simulated electrical position of the motor.

Object:	plcEncSimSetMechAbsPosOffsetLo		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the simulated encoder relative position in mechanical turns unit in order to calculate the simulated encoder absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	plcEncSimSetMechAbsPosOffsetHi		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

This is the offset from the simulated encoder relative position in mechanical angle unit in order to calculate the simulated encoder absolute position (if apply, depending from the encoder configuration).
For more details about the format position refer to the manual (§4).

Object:	plcEncSimSetStatus		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned8
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	Full scale.		

Simulated encoder information status, each bit has different meaning as below:

Mask number	*Name*	*Description*
01h	cstEncStatusRelativeValid	Relative angle is valid.
02h	cstEncStatusElecAngleValid	Electrical angle is valid.
04h	cstEncStatusFatalFaultValid	Fatal fault during the calculations position.
08h	cstEncStatusNonFatalFaultValid	Non-fatal fault status. Combine status from both encoders (absolute and relative status) if one is faulty but electronic angle is valid and relative angle is valid; then result in non-fatal fault as there's at least one valid feedback.
10h	cstEncStatusAbsoluteValid	Absolute mechanical turn is valid.
20h	cstEncStatusAbsoluteWaiting	Waiting status during the calculations of absolute mechanical position for incremental encoder.

11.4 Torque Loop Group

This data structure is referred to the torque loop configuration.

Object:	parILoop		
IEC address:	%MB21.0	*LogicLab Type:*	PAR_MOTHLR
Attributes:	rw		
Description:	Torque loop and power stage control parameters.		
Structure members:	VBrakeLow	*Description:*	Dc bus voltage brake low threshold.
	VBrakeHigh		Dc bus voltage brake high threshold.
	UnderVoltageThreshold		Dc bus Under Voltage threshold.
	OverVoltageThreshold		Dc bus Over Voltage threshold.
	OverCurrentThreshold		Dc bus Over Current thrsehold.
	ILoopKi		Modulator Ki.
	ILoopKp		Modulator Kp.
	Dummy0		Dummy parameter.
	IqLimit		Quadrature current limits.
	IdLimit		Direct current limits.
	Dummy1		Dummy parameter.
	Bridgelayout		Bridge Layout Assignment.

Through Plc program application it should possible to realize a *Torque Mode* profile. In this way the torque reference and the limits of the torque loop are managed by Plc program instead of AxM-II firmware.
Plc variables: plcILoop.*.

Object:	plcILoopIdRef	Direct current reference	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	0.1 mA	*Default value:*	0
Value range:	Full scale.		

Direct current target for current loop control.

Object:	plcILoopIqRef	Quadrature current reference	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	0.1 mA	*Default value:*	0
Value range:	Full scale.		

Quadrature current target for current loop control.

Object:	plcILoopElecAngle		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	0
Value range:	Full scale.		

Electrical angle.

Object:	plcILoopIqLimMax	Iq Lim Max	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied upper limit for the quadrature current.

Object:	plcILoopIqLimMin	Iq Lim Min	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied lower limit for the quadrature current.

Object:	plcILoopIdLimMax	Id Lim Max	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied upper limit for the direct current.

Object:	plcILoopIdLimMin	Id Lim Min	
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	0.1 mA	*Default value:*	n/a
Value range:	Full scale.		

Applied lower limit for the direct current.

11.5 Power Stage Group

Constants to set the power stage configuration.

Constant name	*Description*	*Data Type*	*Mask Value*
cstPStageOptDisableBrakeDrive	Disable Brake resistor.	Unsigned16	0001h
cstPStageOptBackEMFDataEnable	Enable BEMF calculation.	Unsigned16	0002h
cstPStageOptBackEMFATanEnable	Calculate Motor BackEMF	Unsigned16	0003h

In this case the variable points to the same data structure of the Torque Loop Group.

Object:	parPStage		
IEC address:	%MB21.0	*LogicLab Type:*	PAR_MOTHLR
Attributes:	rw		
Description	Torque loop and power stage control parameters.		
Structure members:	VBrakeLow	*Description:*	Dc bus voltage brake low threshold.
	VBrakeHigh		Dc bus voltage brake high threshold.
	UnderVoltageThreshold		Dc bus Under Voltage threshold.
	OverVoltageThreshold		Dc bus Over Voltage threshold.
	OverCurrentThreshold		Dc bus Over Current thrsehold.
	ILoopKi		Modulator Ki.
	ILoopKp		Modulator Kp.
	Dummy0		Dummy parameter.
	IqLimit		Quadrature current limits.
	IdLimit		Direct current limits.
	Dummy1		Dummy parameter.
	Bridgelayout		Bridge Layout Assignment

The power stage features can be controlled through Plc program.

Object:	plcPStagePowerEnable		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Parameter to enable the AxM-II power section; the power is applied to the motor.

- 0(False): disable the AxM-II power section.
- 1(True): enable the AxM-II power section.

Object:	plcPStageReferenceEnable		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Parameter to enable the reference of the AxM-II; the drive function is enabled.

- 0(False): disable the reference.
- 1(True): enable the reference.

Object:	plcPStageDisableBrakeDrive		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Parameter to define if the break resistor is handling or not by the firmware.

- 0(False): disable the break resistor handling.
- 1(True): enable the break resistor handling. This is possible only on event Power Enabled occurred.

11.6 Space Speed Control Loop Group

This data structure is referred to the space speed control loop configuration.

Object:	parSSCntrLp		
IEC address:	%MB31.0	*LogicLab Type:*	PAR_SSCNTRLP
Attributes:	rw		
Description	Space speed control loop.		
Structure members:	UseDifferentKp	*Description:*	Use different speed proportional gain.
	PosKp		Position proportional gain.
	SpdKpRef		Speed: proportional gain reference value.
	SpdKpFbk		Speed: proportional gain feedback avlue.
	AccKpRef		Acceleration: proportional gain reference value.
	AccKpFbk		Acceleration: proportional gain feedback value.
	Ki		Integral gain.
	PosGainShift		Position gain shift.
	AccGainShift		Acceleration gain shift.
	ILimit		Maximum and Minimum limit torque.

For more details about parameters refer to Loop Schema (§E).

The Space speed control loop should receive its references (demand position, demand speed and demand acceleration) from a Plc program application instead of AxM-II firmware positioner module.
Plc variables: plcSSCntrLp.*.

Object:	plcSSCntrlRefMechLo		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Reference position into mechanical angle unit. Represents the 32 LSB bit position input of the speed/position loop as computed by the Plc program. For more details about the format position refer to the manual (§4).

Object:	plcSSCntrlRefMechHi		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Reference position into mechanical turns. Represents the 32 MSB bit position input of the speed/position loop as computed by the Plc program. For more details about the format position refer to the manual (§4).

Object:	plcSSCntrlRefMechSpeed		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Input velocity reference in the speed/position loop.

Object:	plcSSCntrlRefMechAccel		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale.		

Input acceleration reference in the speed/position loop.

Object:	plcSSCntrlEnableIntegralGain		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1(True)		

- 0 (False): disable the integral gain use of the space speed control loop.
- 1 (True): enable the integral gain use of the space speed control loop.

11.7 Positioner Group

This data structure is referred to the configuration and set-up of the positioner module.

Object:	parPositioner		
IEC address:	%MB41.0	*LogicLab Type:*	PAR_POSITIONER
Attributes:	rw		
Description	Positioner and ramp generator.		
Structure members:	EndVelocity	*Description:*	End Velocity.
	ProfileVel		Profile velocity.
	ProfileAcc		Profile acceleration.
	ProfileDec		Profile deceleration.
	QuickStopDec		Quick stop deceleration.
	PositionErrorMaxLo		Max position error (32 LSB).
	PositionErrorMaxHi		Max position error (32 MSB).
	ZeroSpeedThreshold		Threshold speed.

Positioner software module should receive its inputs from Plc program application.
Plc variables: plcPosRG.*.

Object:	plcPosRGQuickStop		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Flag to activate the quick stop functionality.
- 0(False): disable the quick stop functionality.
- 1(True): enable the quick stop functionality.

Object:	plcPosRGVelocity		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Flag to activate the ramp generator.
- 0(False): disable the ramp generator functionality.
- 1(True): enable the ramp generator functionality.

Object:	plcPosRGPosition		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Flag to activate the positioner module.
- 0(False): disable the positioner module.
- 1(True): enable the positioner module.

Object:	plcPosRGNewTarget		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Flag to know at the positioner module that it's available a new target.

- 0(False): no new target to the positioner module.
- 1(True): new target to the positioner module.

Object:	plcPosRGEnablePositioner		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Flag to enable the positioner module operation.

- 0(False): disable the operation of the positioner module.
- 1(True): enable the operation of the positioner module.

Object:	plcPosRGEnableInterpolation		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	0 (False)
Value range:	0 (False) – 1 (True)		

Flag to activate the interpolated module.

- 0(False): disable the interpolated module.
- 1(True): enable the interpolated module.

Object:	plcPosRGTargetPosLo		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale		

Reference position into mechanical angle unit. Represents the 32 LSB bit position input of the speed/position loop as computed by the Plc program. For more details about the format position refer to the manual (§4).

Object:	plcPosRGTargetPosHi		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	Full scale		

Reference position into mechanical turns unit. Represents the 32 MSB bit position input of the speed/position loop as computed by the Plc program. For more details about the format position refer to the manual (§4).

Object:	plcPosRGTargetSpeed		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Velocity factor unit.	*Default value:*	n/a
Value range:	Full scale		

It corresponds to the target velocity computed by the Plc program.

Object:	plcPosRGQuickStopDec		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Acceleration factor unit.	*Default value:*	n/a
Value range:	Full scale		

The quick stop deceleration is the deceleration used to stop the motor if the quick stop ramp is selected as option code.

11.8 Device Control Group

Constants to set the option code to determine what action should be taken by the AxM-II drive to move itself to power disable, fault or quick stop states.

Constant name	*Description*	*Data Type*	*Mask Value*
cstDevCtrlOptCodePowerOff	Disable drive function.	Unsigned16	0000h
cstDevCtrlOptCodeSlowDown	Slow down with slow down ramp; disable of the drive function.	Unsigned16	0001h
cstDevCtrlOptCodeQuickStop	Slow down with quick stop ramp; disable of the drive function.	Unsigned16	0002h
cstDevCtrlOptCodeSlowDownAndStay	Slow down with slow down ramp and stay in quick stop.	Unsigned16	0005h
cstDevCtrlOptCodeQuickStopAndStay	Slow down with quick stop ramp and stay in quick stop.	Unsigned16	0006h

In this case the par variables should be used to set and control the CiA402 device control.

Object:	parDevCtrl		
IEC address:	%MB51.0	*LogicLab Type:*	PAR_DEVCTRL
Attributes:	rw		
Description	CiA402 device control		
Structure members:	ModeOfOperation	*Description:*	CiA402 mode of operation.
	Dummy		Dummy.
	QuickStopOptCode		Quick Stop Option Code.
	ShutdownOptCode		ShutDown Option Code.
	DisableOperationOptCode		Disable Operation Code.
	FaultReactionOptCode		Fault reaction Option Code.
	FollowingErrWindowLo		Following error window (32 LSB).
	FollowingErrWindowHi		Following error window (32 MSB).
	FollowingErrTimeout		Following error timeout.
	TargetPosWindowLo		Target Postion (32 LSB).
	TargetPosWindowHi		Target Position (32 MSB).
	TargetPosTimeout		Target Position Timeout.
	VelocityWindow		Velocity Window.
	VelocityTimeout		Velocity Timeout.
	VelocityThresholdWindow		Velocity Threshold Window.
	VelocityThresholdTimeout		Velocity Threshold Timeout.
	IPTimeUnits		Interpolated Mode Time Units.
	IPTimeIndex		Interpolated Mode Time Index.

The CiA402 finite state machine and the position target should be realized through Plc program application.

Object:	plcDevCtrlControlWord		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

This variable contains the bits for controlling the state machine and the specific operating mode.
For more details about Control word refer to (§1) or to (§3) or to wksControlWord variables.

Object:	plcDevCtrlTargetPosLo		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	n/a		

User target position in mechanical angle.
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The target position that is usually generated from the master control; anyway the user should be configuring it using the Cockpit configurator and the basic application of the AxM-II.
The *Positioner* application allows the user to control the AxM-II drive as programmable multi-position positioner.

Object:	plcDevCtrlTargetPosHi		
Modbus IPA:	--	*LogicLab Data Type:*	Signed32
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	Position factor unit.	*Default value:*	n/a
Value range:	n/a		

User target position in mechanical turns.
The target position is the position that the drive should move to in position profile mode using the current settings of motion control parameters such as velocity, acceleration, deceleration, motion profile type etc. The target position that is usually generated from the master control; anyway the user should be configuring it using the Cockpit configurator and the basic application of the AxM-II.
The *Positioner* application allows the user to control the AxM-II drive as programmable multi-position positioner.

11.9 Thermal Model Group

This data structure is referred to the thermal model configuration.

Object:	parThermalModel		
IEC address:	%MB72.0	*LogicLab Type:*	PAR_THERMALMODEL
Attributes:	rw		
Description	Thermal Model		
Structure members:	BrakeResistorValue	*Description:*	Brake resistor value [100 mΩ].
	BrakeResistorPower		Brake resistor power [100 mW].
	CoolingTempOn		On cooling temperature [100m °C].
	CoolingTempOff		Off cooling temperature [100m °C].
	MotorOverTemp		Motor overtemperature [°C].
	BrakeResistorEnergy		Brake resistor energy [J].

11.10 CANOpen Group

Constants to set the CiA301 state machine for the network management (NMT) objects. Through NMT services, CANOpen devices are initialized, started, monitored, reset or stopped. All CANOpen devices are regarded as NMT slaves.

Constant to command the NMT state machine.

Constant name	*Description*	*Data Type*	*Mask Value*
cstCanOpenNMTCStart	NMT command start	Unsigned8	01h
cstCanOpenNMTCStop	NMT command stop	Unsigned8	02h
cstCanOpenNMTCEnterPreoperational	NMT command to pre-operational	Unsigned8	80h
cstCanOpenNMTCResetNode	NMT command to reset the CAN node	Unsigned8	81h
cstCanOpenNMTCResetCommunication	NMT command to reset communication	Unsigned8	82h

Constant to read state of NMT Can communication.

Constant name	*Description*	*Data Type*	*Mask Value*
cstCanOpenNMTSBootup	NMT state initialization	Unsigned8	00h

cstCanOpenNMTSStopped	NMT state stopped	Unsigned8	04h
cstCanOpenNMTSPreoperational	NMT state pre-operational	Unsigned8	7Fh
cstCanOpenNMTSOperational	NMT state operational	Unsigned8	05h

For more details refer to (§1).

This data structure is referred to the CANOpen communication set-up.

Object:	parCANOpen		
IEC address:	%MB303.0	*LogicLab Type:*	PAR_CANOPEN
Attributes:	rw		
Description	CANOpen parameters.		
Structure members:	Dummy1	*Description:*	Dummy parameter.
	LssNodeId		Lss Node ID.
	LssTimingIndex		Lss Timing Index.
	Dummy2		Dummy parameter.
	tSyncCOB		COB-ID used by SYNC COB.
	EmcyCOB		COB-ID used by Emergency COB.
	EmcyInhibitTime		Emergency COB Inhibit Time [100usec].
	GuardTime		Guard Time [ms].
	HeartbeatTime		Heartbeat [ms].
	LifeTimeFactor		Life Time Factor.
	Dummy3		Dummy parameter.
	CommCyclePeriod		Communication Cycle Period [usec].

This variable is used to monitor the state machine of the network management (NMT).

Object:	varCANOpenNMTStatus		
Modbus IPA:	--	*LogicLab Data Type:*	Unsigned16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	n/a		

NMT state machine actual status.
To evaluate the NMT state machine actual status the user have use the constant to read NMT state machine.

11.11 CAN Group

Constants to set and control the CAN physical layer properties.

Constant name	*Description*	*Data Type*	*Mask Value*
cstCanNode0	Number Node 0	Unsigned16	0000h
cstCanNode1	Number Node 1	Unsigned16	0001h
cstCanDrvRun	Driver CAN in run mode.	Unsigned16	0001h
cstCanDrvErrPasv	Driver CAN in error passive mode.	Unsigned16	0002h
cstCanDrvBusOff	Driver CAN in bus off status.	Unsigned16	0004h
cstCanDrvHwOverrun	Driver CAN in hardware over-run status.	Unsigned16	0008h

This data structure is referred to the CAN hardware configuration and communication set-up.

Object:	parCAN		
IEC address:	%MB301.0	*Logic Lab Type:*	PAR_CAN
Attributes:	rw		
Description	CAN bus parameters.		
Structure members:	ReSyncEnable	*Description:*	Enable pwm synchronization with filedbus
	CanController		CAN controller selection
	DisableCanAlarmMask		CAN alarm disable mask

11.12 EtherCAT Group

This data structure is referred to activation of EtherCAT protocol communication module.

Object:	parECAT		
IEC address:	%MB800.0	*LogicLab Type:*	PAR_ECAT
Attributes:	rw		
Description	EtherCAT parameters		
Structure members:	ReSyncEnable	*Description:*	Enable pwm synchronization with filedbus
	EnableModule		Enable EtherCAT module (reset required)
	CfgStationAlias		Configured station alias

For more details about EtherCAT fieldbus please refer to (§6).

11.13 Sync Manager Group

This data structure is referred to the Sync manager status and configuration.

Object:	parSyncMgr		
IEC address:	%MB218.0	*LogicLab Type:*	PAR_SYNCMGR
Attributes:	rw		
Description	SyncManager parameters		
Structure members:	TSFilter	*Description:*	Time constant of the filter (0=no filter).
	ReSyncDelta		Time shift for Sync point [nsec].
	FiltKp		PLL proportional gain.
	FiltKd		PLL derivative gain.
	PeakNDiscard		Number of consecutive discardable sample.
	PeakThreshold		Threshold for peak detection.

The following variables are used to monitor the Sync signal status.

Object:	varSyncMgr_Correction		
Modbus IPA:	--	*LogicLab Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	Full scale.		

Instantaneous value used for the correction of the Sync signal status.

Object:	varSyncMgr_LastSamplePoint		
Modbus IPA:	--	*LogicLab Data Type:*	Signed16
CANopen/COE index:	--	*CANopen/COE Data Type:*	--
Attributes:	rw		
Unit:	n/a	*Default value:*	n/a
Value range:	Full scale.		

Last fieldbus sample point, averaged.

A. ENCODER POSITION FORMAT

The encoder position is calculated on 64 bit range, subdivide us show in table above.
This means that the internal device unit (d.u.) for the position is expressed as: the MSB 32 bit give the number of turns, the LSB 32 bit give the angular position in one turn, giving the relations shown in the conversion factor group unit (§4).

MSB	LSB
Position Turn (MSB 32 bit)	*Angular position (LSB 32 bit)*

Figure 11 - Encoder Position

Position Turn.
Signed total number of mechanical turns.

Angular position.
The position angle calculated over one turn, where: 2^{32} means 360° mechanical degrees.

When the position reach upper or lower boundary the AxM-II drive automatically wrap the position to the opposite boundary.

The position error is calculated as 64 bit difference from the reference and feedback positions; then drive chooses fastest direction to reach the target position.

The speed is calculated as difference between two consecutive reading of the position encoder (125μs) and then filtering to improve the quality of the speed loop.
The acceleration is simply computed as difference of the speed, still every 125μs.

B. PHYSICAL UNITS VS. INTERNAL DEVICE UNITS CONVERSION

Current:	$I[d.u.] =\sim 1.0 \quad 10^{4} \quad I[Arms]$	$I[Arms] =\sim 1.0 \quad 10^{-4} \quad I[d.u.]$
Position:	$\theta[d.u.] =\sim 1.193 \quad 10^{7} \quad \theta[°]$	$\theta[°] =\sim 8.382 \quad 10^{-8} \quad \theta[d.u.]$
Velocity:	$\omega[d.u.] =\sim 8.544 \quad 10^{4} \quad \omega[rad/s]$	$\omega[rad/s] =\sim 1.170 \quad 10^{-5} \quad \omega[d.u.]$
Acceleration:	$\dot{\omega}[d.u.] =\sim 4.374 \quad 10^{4} \quad \dot{\omega}[rad/s^2]$	$\dot{\omega}[rad/s^2] =\sim 2.285 \quad 10^{-5} \quad \dot{\omega}[d.u.]$
Electrical Angle:	$\varphi \ [d.u.] =\sim 1.820 \quad 10^{2} \quad \varphi \ [°]$	$\varphi[°] =\sim 5.49 \quad 10^{-3} \quad \varphi[d.u]$

C. PLC MAPPING APPLICATION PARAMETERS

Inside a Plc program it is possible to use variables which can be managed by an external configuration or supervision programs (e.g. Cockpit configurator). These variables must be located at defined memory addresses and linked to an index that allow them to be integrated into the system database.
Some variables can keep their value permanently: these particular types of variables are defined application parameters. Other variables are not saved in flash and they are used just for data exchange. This type of variables are defined application works variable.
The application parameters and the application works variable can be transferred and modified by the dedicated application. The AxM-II firmware provides different memory zones (data block) for the application parameters and works definition, different by type (16bit, 32bit, 8bit, bit) and write/read properties (r/w). Every time the user defines a new parameter, by default it allocates a variable in the drive firmware (associated variable) in the memory zone dedicated to the declared parameter type.

All the application parameters are grouped in a database format, split in different groups and accessible by the *Library window* and by *Global Variables* tab of the LogicLab environment (§7).

The tables resume the application parameters and works data blocks.

Imaged Object	*Img IEC addr*	*No Imaged Object*	*No img IEC addr*	*Modbus IPA*	*Modbus Type*	*COE idx.subidx*	*COE Type*
usrParamInt	%MW100.0-127	usrNoImgParamInt	%MW140.0-127	14000-14127	Signed16	2900h. 01-80h	Signed16
usrParamDInt	%MD101.0-127	usrNoImgParamDInt	%MD141.0-127	14500-14627	Signed32	2901h. 01-80h	Signed32
usrParamReal	%MD102.0-127	usrNoImgParamReal	%MD142.0-127	15000-15127	Real32	2902h. 01-80h	Real32
usrParamBool	%MX103.0-127	usrNoImgParamBool	%MX143.0-127	15500-15627	Signed16	2903h. 01-80h	Signed8
usrWorksInt	%MW120.0-127	usrNoImgWorksInt	%MW110.0-127	16000-16127	Signed16	2908h. 01-80h	Signed16
usrWorksDInt	%MD121.0-127	usrNoImgWorksDInt	%MD111.0-127	16500-16627	Signed32	2909h. 01-80h	Signed32
usrWorksReal	%MD122.0-127	usrNoImgWorksReal	%MD112.0-127	17000-17127	Real32	290Ah. 01-80h	Real32
usrWorksBool	%MX123.0-127	usrNoImgWorksBool	%MX113.0-127	17500-17627	Signed16	290Bh. 01-80h	Signed8
usrWorksSInt	%MB124.0-63	usrNoImgWorksSInt	%MB114.0-63	17750-17813	Signed16	290Ch. 01-40h	Signed8

It must be observed that in most part of the system variables, the compiler generates a process image of the variable themselves. This means that the application task making use of the imaged system variable, it doesn't operate directly on the firmware, but on a copy of it. Only when the performance task terminates, the copy is automatically transferred on the system variable.

WARNING: Only for expert users there are also available parameters and working variables, without the image concept, named usrNoImgParam* and usrNoImgWorks*. This means that the application task uses directly the data on the firmware and not a copy of them. These variables should be connected to the same IPA value of the user parameters and of the user working variables.

Example of Plc variable mapping criteria.
For the fourth element of user working variables long data type the Plc mapping results:

usrWorkDInt[3] <=> IEC address: %MD111.3 <=> Modbus IPA: 16503 <=> COE Idx.subidx: 2909h.04h

For more details how to create and to use the user parameters and the user working variables refer to (§4) and to (§7).

D. AXM-II DRIVE ALARMS LIST.

If more alarms are together active, the AxM-II drive visualizes through leds status only the more important one. (E.g.: if the alarm n° 16 "Endat alarm" and n°20 "Thermal model fail" are active, only the Endat alarm is visualized). To get a complete list of the active alarms, use the function *Active Alarms*  of the Cockpit configurator. The alarm type messages are referred to the alarm indicated by the Cockpit configurator.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0000 0001h	Software Fault		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0008 0000h	FPGA WDT fail	Fpga watchdog mechanism control failure.	Contact the PMC Service.
0010 0000h	Fpga Crc fail	Fpga CRC checksum algorithm calculation failure.	Contact the PMC Service.
0020 0000h	Fpga lockout	Fpga locked. The power section will be locked.	Contact the PMC Service.
0040 0000h	Fpga PLL loss of clock	Fpga PLL circuit loss the reference clock signal.	Contact the PMC Service.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0000 0002h	PLC Tasks overtime		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	PLC Slow Task overtime	The run time of Slow task is greater than its activation period (1-64 msec).	If a user application is active, optimize the Slow run time. If Base app or SpeedV application is active, then contact the technical PMC Service.
0000 0002h	PLC Fast Task overtime	The run time of Fast task is greater than its programmed activation period 125 µsec.	If a user application is active, optimize the Fast run time. If Base app or SpeedV applications are active, then contact the technical PMC Service.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0000 0004h	Boot Failure		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	Invalid HW Configuration	Error occurred during the boot firmware: invalid hardware configuration. The hardware configuration flash sector is damaged.	Try to reload software and parameters table on the drive. Then contact PMC service.
0000 0002h	Invalid Flash Parameters.	Error occurred during the boot firmware: invalid Flash Parameters. The flash sector where the parameters are saved is damaged.	Try to reload software and parameters table on the drive. Then contact PMC service.
0000 0003h	due to power fail.	Error occurred during the boot firmware: boot Failure due to power fail.	Measure the input power supply. If the problem persist also when the power supply value is correct then contact PMC service.
0000 0010h	FPGA Programming Failed.	Error occurred during the boot firmware: the FPGA firmware download procedure is failed.	Try to reload firmware. If the problem persists, contact the PMC service.
0000 0018h	due to incompatible FPGA.	Error occurred during the boot firmware: boot failure due to incompatible firmware release and FPGA hardware revision. For more detail refer to §(4).	Try to reload software and parameters table on the drive. Then contact PMC service.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0000 0008h	System boot locked by parameter(s) value out of range.		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
--	--	Boot system fail for invalid parameter configuration.	Check the value of varSysWrongParCode parameter that should indicate which parameter has the wrong value. Correct parameter, save and reset the drive should solve the error.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0000 0010h	Power Section Fault.		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
--	--	Fault on the IGBT control of the main power bridge.	Reset the drive and check if the error returns. If the problem persists, contact the PMC Service.
0000 0001h	Bridge 2	Fault on the IGBT control of the second power bridge (if supported).	Reset the drive and check if the error returns. If the problem persists, contact the PMC Service.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0000 0020h	Internal Flash Memory Failure.		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	Flash SysLog	The flash sector is damaged.	Try to reload firmware and parameters table. If the problem persists, contact the PMC service

Group Alarm Code	Alarm Type		Leds Panel
0000 0040h	Safe Torque Off (STO) activated.		
SubCode Alarm	Subcode Identification	Description	Suggested Action
--	--	The Safe torque off mechanism is activated. Safe Torque Off performs a safety function such that when the Enable input is not asserted, i.e. open-circuit or set at nominally 0V, the drive will not develop torque in the motor.	Verify the presence of 24 Vdc power supply for the Safe Torque circuit.

Group Alarm Code	Alarm Type		Leds Panel
0000 0100h	Overcurrent Fault.		
SubCode Alarm	Subcode Identification	Description	Suggested Action
--	--	The actual AxM-II drive current has got an instant value higher than the maximum value allowed by the AxM-II drive.	Check the current loop gains setting, the possible mechanical obstacle and the correct motor size for the use.
0000 0001h	Bridge 2	The actual AxM-II drive current on the second power bridge has got an instant value higher than the maximum value allowed by the AxM-II drive.	Check the current loop gains setting, the possible mechanical obstacle and the correct motor size for the use.

Group Alarm Code	Alarm Type		Leds Panel
0000 0200h	Overvoltage Fault.		
SubCode Alarm	Subcode Identification	Description	Suggested Action
--	--	Overvoltage fault: too high voltage level on DC bus voltage has been detected.	Check if the brake resistor is present and connected.

Group Alarm Code	Alarm Type		Leds Panel
0000 0400h	Undervoltage Fault.		
SubCode Alarm	Subcode Identification	Description	Suggested Action
--	--	The DC bus voltage is under the threshold when the drive is enabled.	Verify if the power supply is present.

Group Alarm Code	Alarm Type		Leds Panel
0000 8000h	Encoder Fault.		
SubCode Alarm	Subcode Identification	Description	Suggested Action
0000 0001h	Main Encoder Abs/Rel Track Check Failure	The difference between absolute and incremental (relative) position value is too high.	Check the encoder connections (E1) and the configuration of the encoder parameters.
0000 0002h	Encoder Supply Voltage Failure	The encoder power supply voltage is out of valid range.	Check the power supply voltage, the encoder connections (E1) and the configuration of the encoder parameters.

Group Alarm Code	Alarm Type		Leds Panel
0001 0000h	Main (Aux) Endat Encoder Fault		
SubCode Alarm	Subcode Identification	Description	Suggested Action
0000 0001h	Endat Init Fail	Error during the initialization procedure of the Endat configuration.	Check the encoder connections (E1) and the Endat parameters configuration.
0000 0002h	Endat Alarm Flag	Endat encoder device in alarm condition.	Check the encoder connections (E1) and the Endat parameters configuration.
0000 0004h	Endat CRC Error	Error in Endat communication. The CRC control algorithm failed.	Check the encoder connections (E1) and the Endat parameters configuration.
0000 0008h	Endat Data Overtime	Error in Endat communication: the read position command doesn't terminate its procedure within the time period calculated during the Endat configuration procedure.	Check the encoder connections (E1) and the Endat parameters configuration.
0000 0010h	Endat Unsupported Clock Frequency	Unsupported clock frequency. The supported values are: 2 kHz and 8 kHz.	Check the encoder connections (E1) and the Endat parameter parEncMEndat.ClockFreq configuration.

Group Alarm Code	Alarm Type		Leds Panel
0002 0000h	Absolute Analog Encoder Fault		
SubCode Alarm	Subcode Identification	Description	Suggested Action
0000 0001h	Invalid line Count	The drive has detected an incorrect index position.	Check the encoder connections (E1).
0000 0002h	Low Tracks Level	The voltage ripple of the encoder analogic channels was higher than the maximum limit set by the parameter.	Check the encoder connections (E1). Verify also the parameter related to the configuration of threshold alarm: parEncAn.AnalogAlarmThreshold

Group Alarm Code	Alarm Type		Leds Panel
0004 0000h	Main Incremental Encoder fault		
SubCode Alarm	Subcode Identification	Description	Suggested Action
0000 0001h	Invalid line count	The AxM-II drive has detected an incorrect index position.	Check the encoder cabling (E1) and the shield connection. Verify also the parameters related to the index tolerance: parEncMInc.Flags.DisableIndexError and parEncMInc.IndexErrorTolerance.

0000 0002h	Low Tracks level	The voltage ripple of the encoder analogic channels was higher than the maximum limit set by the parameter.	Check the encoder cabling (E1) and the shield connection. Verify also the parameters related to the threshold alarm: parEncMInc.Flags.DisableAnalogError and parEncMInc.AnalogAlarmThreshold.
0000 0004h	Invalid Digital Decoding	Invalid transition state in digital decoding algorithm.	Check the encoder cabling (E1) and the shield connection.
0000 0010h	Invalid Analog/Digital Decoding	Incremental encoder fault in conversion analog/digital algorithm.	Check the encoder cabling (E1) and the shield connection.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0004 0000h	Aux Incremental Encoder		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0008h	Simulation Invalid Count	When the difference between main encoder position and the calculated encoder simulation position is greater than maximum position error tolerance an auxiliary incremental encoder fault is generated.	Check the encoder connection (C1 connector) and the shield connection. Verify also the parameters related to the simulated encoder parEncSimInc.MaxPosErrTolerance.
0000 8001h	Invalid Line Count	The AxM-II drive has detected an incorrect index position.	Check the auxiliary encoder connection (C1 connector) and the shield connection. Verify also the parameters related to the index tolerance: parEncAInc.Flags.DisableIndexError and parEncAInc.IndexErrorTolerance.
0000 8004h	Invalid Digital Decoding	Invalid transition state in digital decoding algorithm.	Check the encoder connection (C1 connector) and the shield connection.
0000 8008h	Simulation Failure	The AxM-II drive has detected an error on encoder simulation stage.	Check the auxiliary encoder connection (C1 connector) and the shield connection.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0008 0000h	Hall sensors Encoder Fault		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	Encoder Count Loss.	Encoder hall sensor: invalid line counts.	Check parameters configuration and the encoder connections (E1 connector).
0000 0002h	Encoder invalid State.	For 4 Wire configurations, invalid hall sensor state.	Check parameters configuration and the encoder connection (E1 connector).

Group Alarm Code	*Alarm Type*		*Leds Panel*
0010 0000h	Thermal Model Fault		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	IGBT NTC Fail	The power module has reached an excessive temperature.	Too heavy work cycle.
0000 0002h	Junction Overtemperature	The estimate junction temperature reached an excessive temperature.	Too heavy work cycle.
0000 0004h	Fan Cooler Blocked	The drive reports a bad performance of the cooling system.	Verify possible obstructions of the cooling air flux or fan hindrance.
0000 0008h	Heatsink Overtemperature	The heatsink has reached an excessive temperature.	Too heavy work cycle.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0040 0000h	Sensorless		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	KT Value Out of Range	The estimate torque constant is out of range (± 25 % of Kt motor).	Check parameters configuration, motor phases connections and drive AC line input.
0000 0002h	Invalid KT Ratio	Invalid estimated sensorless speed constant.	Check parameters configuration, motor phases connections and drive AC line input.
0000 0008h	Sync Lost	Synchronization lost between the electrical excitation and the rotor position.	Check parameters configuration and the working cycle.
0000 0010h	Antiglitch Fault	The anti-glitch filter is active for more than 4 cycles.	Check parameters configuration and the working cycle.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0080 0000h	Brake Fault		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
0000 0001h	Brake Drive Always On	The braking circuit is always active.	The supply voltage is too high or the clamp activation voltage is too low. Verify the parameters parPStage.VBrakeLow and parPStage.VBrakeHigh.
0000 0002h	Brake Resistor Overpower	The power dissipated in the braking resistor is higher than the maximum allowed.	Connect an external resistor of higher power rating.
0000 0004h	Brake IGBT desaturation	Brake circuit failure.	Braking resistor too low value or shorted or braking IGBT failure.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0100 0000h	CANOpen Fieldbus Fault		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
	--	CAN fieldbus physical layer error.	Check parameters configuration, hardware connections, CANOpen master configuration and messages overrun.
0000 0001h	CAN Module HW Overrun	CAN module hardware overrun.	Reduce network load for the slave.
0000 0002h	CAN Module Passive Mode	CAN controller entered error in passive mode.	Noisy network environment or incorrect bus termination, refer to (§4).
0000 0004h	CAN Module BusOff	Recover from CAN controller bus-off.	Extremely noisy network environment.
0000 0008h	CANOpen Error Control Protocol Parameters Error	There are two types of errors control protocol: Node Guarding Protocol and Heartbeat protocol. Either mechanisms control has generated a fault.	Check parameters configuration and CANOpen master configuration.
0000 0010h	CANOpen Error Control Protocol Timeout	SYNC PDO processing overtime.	The time slot assigned to the synchronous PDOs is not enough to process all user defined PDOs, reduce the number of PDO or the number of objects inside them.
0000 0020h	CANOpen RX PDO Length Error	RxPDO length error.	The length of RPDO does not match with the internally calculated length, refer to (§1).
0000 0080h	CAN Module SW Overrun	CAN Module SW Overrun.	The node has received a new instance of one RPDO before processing the old one.
0000 0100h	CANOpen EMCY Wrong COB-ID	The COB-ID for the EMCY message is wrong.	Check parameters configuration of the EMGY message. The COB-ID has to be composed by 080h+node-ID.
0000 0800h	CAN Module TX Overrun	CAN Transmission Overrun.	Baud rate mismatch, CAN network without proper terminating resistor, improper CAN network cabling can cause this error.
0001 0000h	CANOpen PDO Invalid COB-ID	CANOpen PDO Invalid COB-ID. The COB-ID is not in the range value defined by the CiA301 specification.	The COB-ID value is not valid. Check parameters configuration and CANOpen master configuration,
0002 0000h	CANOpen PDO RTR Not Valid	CANOpen PDO RTR Not Valid. RTR not allowed on this PDO.	Request for RTR command not valid. Check parameters configuration.
0004 0000h	CANOpen PDO Invalid TX Type	CANOpen PDO Invalid TX Type.	Check parameters configuration and CANOpen master configuration,
0008 0000h	CANOpen PDO Invalid Map Count	TxPDO or RxPDO should be configured without any objects or the firmware is not able to compose the configured TxPDO or RxPDO.	Verify configured index value, index value should be within supported range and with read access. Also corrupt TPDO configuration can cause this error, in such case reload the CANOpen master configuration.
0010 0000h	CANOpen PDO Out Of Memory	PDO out of memory.	Due to internal handling of PDOs, reduce the number of PDO or the number of objects inside them or the order of these objects; all PDOs are not created, thus unavailable.
0020 0000h	CANOpen PDO Invalid Sync Start Value	Invalid data for Sync start value.	Check the COB-ID value for Sync object type. The COB-ID has to set to the value 80h.
0040 0000h	CANOpen PDO Internal Error	CANOpen PDO Internal Error. The TxPDO or RxPDO configuration is wrong.	Check parameters configuration and CANOpen master configuration,
0080 0000h	CANOpen PDO Length Exceed	TxPDO length error. The length of TxPDO does not match with the internally calculated length.	Check parameters configuration and CANOpen master configuration.
0100 0000h	CANOpen PDO Hook Internal Error	CANOpen internal software error.	Check parameters configuration and CANOpen master configuration.
0200 0000h	CANOpen PDO Duplicated COB-ID	CANOpen PDO duplicated COB-ID identification. The COB-ID identification is already exists.	Check parameters configuration and CANOpen master configuration,

Group Alarm Code	*Alarm Type*		*Leds Panel*
0200 0000h	EtherCAT Failure		
SubCode Alarm	*Subcode Alarm*	*Description*	*Suggested Action*
--	--	EtherCAT fieldbus physical layer error.	Check Ethernet cables connections, check EtherCAT master running status.

Group Alarm Code	*Alarm Type*		*Leds Panel*
0080 0000h	Motor Overtemperature		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
--	--	The PTC sensor has detected a too high motor temperature.	Check the connection of the motor PTC to the correct drive terminals and the real motor temperature.

Group Alarm Code	*Alarm Type*		
1000 0000h	User Defined Alarm		
SubCode Alarm	*Subcode Identification*	*Description*	*Suggested Action*
--	--	Error generated from the customer PLC application.	Depend on PLC application. Set at True the parameters parPlcExeDisable value and Reset the drive. If the error disappears the user will be sure that the problem is own Plc program application.

Note: leds meaning.

Fault. The number inside specifics the weight of the bit error. E.g.:  means 0x1Bh error code.

No fault. E: drive enable. C: Modbus communication is working.

1 Hz blinking led means that the AxM-II drive is in alarm or in error condition.

E. AXM-II DEFAULT PDO PARAMETERS

Those are the default PDO communication and mapping parameters for the AxM-II drive.

PDO	RPDO #1	
COB-ID	4000 0200h+node-ID (enabled)	
Type	255 (asynchronous)	
COB-ID	B0	B1
200h+node-ID	Controlword 6040h.0h	

PDO	RPDO #2		
COB-ID	4000 0300h+node-ID (enabled)		
Type	255 (asynchronous)		
COB-ID	B0	B1	B2
300h+node-ID	Controlword 6040h.0h		Mode of operation 6060h.0h

PDO	RPDO #3					
COB-ID	4000 0400h+node-ID (enabled)					
Type	255 (asynchronous)					
COB-ID	B0	B1	B2	B3	B4	B5
400h+node-ID	Controlword 6040h.0h		Target position 607Ah.0h			

PDO	RPDO #4					
COB-ID	4000 0500h+node-ID (enabled)					
Type	255 (asynchronous)					
COB-ID	B0	B1	B2	B3	B4	B5
500h+node-ID	Controlword 6040h.0h		Target velocity 60FFh.0h			

PDO	RPDO #5
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)

PDO	RPDO #6
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)

PDO	RPDO #7
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)

PDO	RPDO #8
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)

RPDO from #5 to #8 have no default mapping parameters.

PDO	TPDO #1	
COB-ID	4000 0180h+node-ID (enabled)	
Type	255 (asynchronous)	
Inhibit Time	0	
COB-ID	B0	B1
180h+node-ID	Statusword 6041h.0h	

PDO	TPDO #2
COB-ID	4000 0280h+node-ID (enabled)
Type	0 (synchronous acyclic)
Inhibit Time	0

COB-ID	B0	B1	B2
280h+node-ID	Statusword 6041h.0h		Mode of op.display 6061h.0h

PDO	TPDO #3
COB-ID	4000 0380h+node-ID (enabled)
Type	0 (synchronous acyclic)
Inhibit Time	0

COB-ID	B0	B1	B2	B3	B4	B5
380h+node-ID	Statusword 6041h.0h		Position actual value 6064h.0h			

PDO	TPDO #4
COB-ID	4000 0480h+node-ID (enabled)
Type	0 (synchronous acyclic)
Inhibit Time	0

COB-ID	B0	B1	B2	B3	B4	B5
480h+node-ID	Statusword 6041h.0h		Velocity actual value 606Ch.0h			

PDO	TPDO #5
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)
Inhibit Time	0

PDO	TPDO #6
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)
Inhibit Time	0

PDO	TPDO #7
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)
Inhibit Time	0

PDO	TPDO #8
COB-ID	C000 0000h (disabled)
Type	255 (asynchronous)
Inhibit Time	0

TPDO from #5 to #8 have no default mapping parameters.

F. SPACE SPEED CONTROL LOOP SCHEMA

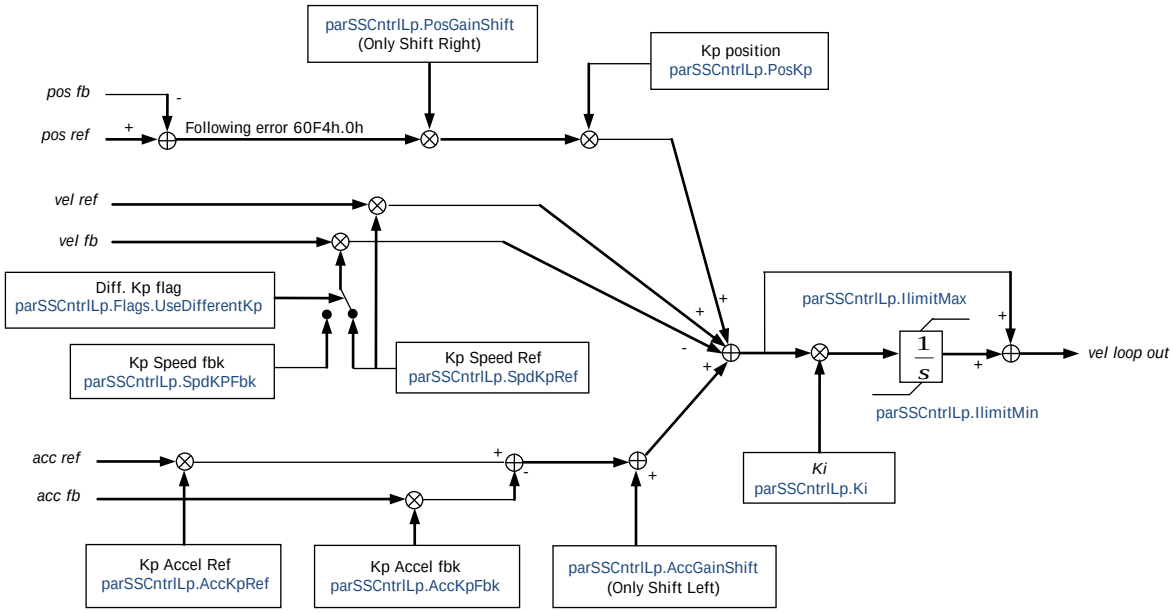

Figure 12 - Speed loop main schema

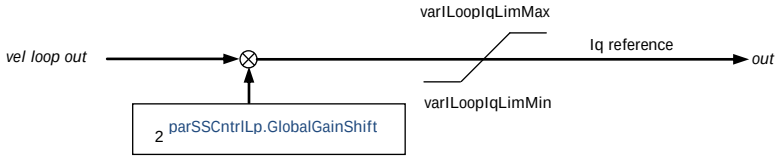

Figure 13 - Speed loop output schema

G. AXM-II FIRMWARE DIAGRAM

AxM-II firmware diagram and interaction with Plc program application (§7).

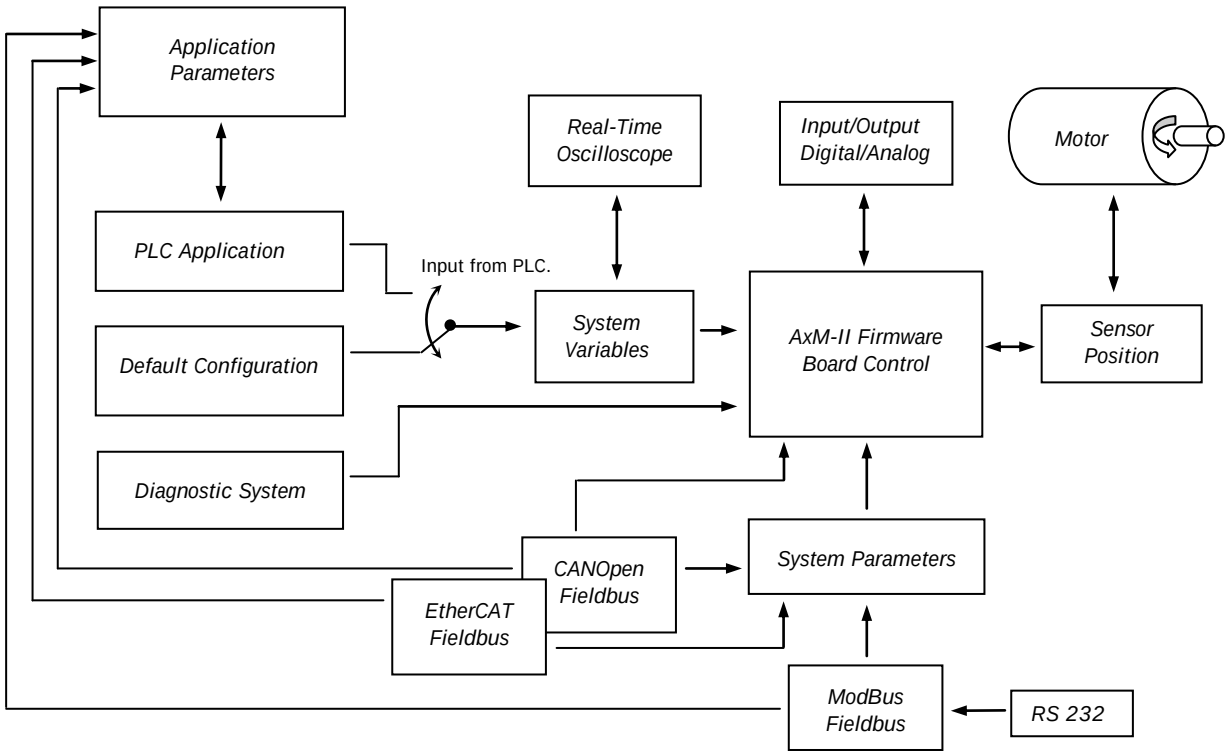

Figure 14 - AxM-II firmware diagram.

H. IPA PARAMETERS SORTED INDEX

I. NAME OBJECTS SORTED INDEX

J. CANOPEN/COE OBJECTS SORTED INDEX