V6 control firmware FD1E and FD2E

Stepper motor driver control firmware

02-10-2018 first issue

1. DESCRIPTION

FD-family drivers are an innovative line of fully-digital dual-H-bridge stepper motor drivers, specifically developed to meet the needs of low vibrations and efficient power consumption, maximizing motor performances.

FD1E and FD2E expand the available interfaces, implementing EtherCAT communication through industrial circular connectors (M8 and M12).
Mounted on the rear of the motor, FD drivers exploits a 12-bits per revolution absolute (single-turn) magnetic encoder.

V6 firmware implements the classic current control method, which consist of fixed motor current independent from the load (only the current reduction when the motor is not moving), or it can also calculate the optimum amount of current measuring the resistant torque applied to the motor. This latter torque control method offers two main benefits:

- Motor and driver work with higher efficiency, reducing the power losses and operating temperature.
- Ordered steps, which cannot be executed because of a sudden increment of torque demand above maximum motor torque, are accumulated, allowing a slowdown of the motor that consequently is able to distribute higher torque for taking over the obstacle. As soon as the resistant torque decreases, the motor recovers the steps accumulated, without any position loss (an alarm can be configured when the following error exceeds a programmable value).

The master can control FD motion using EtherCAT (CoE) protocol or the standard digital inputs (step/dir, quadrature steps A/B, select/start/stop cycle). Modbus RTU via RS-232 is available as easy troubleshooting communication port and also for firmware and parameter reprogramming.

- EtherCAT (CoE)

- Multipurpose I/O

FD1.1E	FD2.1E
2 single-ended DI	2 fast differential DI
1 pnp DO	3 single-ended DI
	3 pnp DO

- Torque Control loop

Adjustable I_{MAX} (current at maximum torque)
Adjustable I_{MIN} (current at no torque)

- 12-bit absolute encoder

- 32 programmable cycles, 10 cycles sequences

Cycle or sequence of cycle's selection, start and stop using DI or fieldbus. Configurable speed, acceleration, deceleration, target position with linear, parabolic and s-curve motion profiles. Several movements available.

- Position resolution

Configurable steps per revolution

- Absolute multi-turn position recovery at power on

- Over temperature (100 °C), over voltage, under voltage and short circuit alarms

- Step accumulator with programmable alarm limit

Fig. 1 – FD2.1E-1404

2. INDEX

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

3. HARDWARE

FD-family drivers measure the motor current through high performance Hall-effect sensors equipped with fast overcurrent detection. The current control takes place via dual H-bridge power stage made of very low on-resistance MOSFETs. The bridges are controlled with 40 kHz PWM modulation in order to output two orthogonal sine-wave currents, one per each motor winding.

The period of sine-wave current is divided into a configurable number of steps, allowing resolutions from 400 up to 204˙800 steps per revolution in a bi-phase 1.8° motor.

The advanced current loop is able to maintain a very flat torque/speed characteristic in all the speed operating range.

Those application characterized by a variable load torque can achieve a significant power consumption reduction and higher torque availability with the torque control loop available in V6.
The load torque is calculated using the 12-bit magnetic encoder acquisition. This value is continuously processed to determine the best amount of current needed to move the load inside motor operating limits (the sine wave peak current value will range between the programmed values I_{MIN} and I_{MAX}).

Another feature of V6 is the possibility to accumulate the steps which cannot be executed because of a sudden resistant torque above the maximum motor torque. In this case V6 maintains the maximum motor torque and, when the load torque decreases, the motor can recover the steps accumulated, accelerating and reaching the reference position. The engage, which is the change from following mode to synchronous mode, takes place through bump-less speed adjustment and without vibrations.

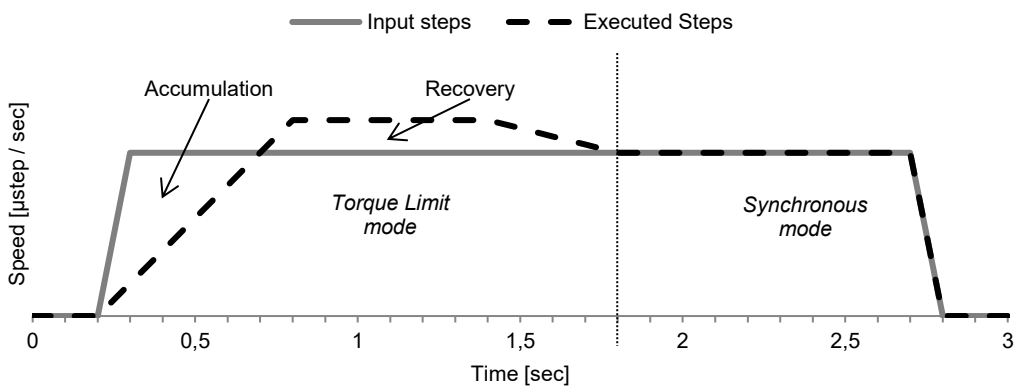


Fig. 2 – Speed profile with step accumulation and recovery during acceleration

In those motions characterized by high acceleration and inertial load, traditional stepper systems need to have sufficient torque margins, so that in case of an increment of the load, the motor does not lose the synchronism with consequent step loss (or even stop if the frequency is above the start/stop frequency). In other words, with the traditional stepper driver, it is necessary to oversize motor and driver.
With V6 control firmware, instead, the driver increases current and torque until the maximum set value. In case of higher resistant torque, the resulting speed and acceleration reduction is managed through the accumulation of the input steps not been executed (configurable alarm limit of input steps accumulation is implemented). As soon as the resistant torque decreases the driver recovers the accumulated steps without position loss.

If the accumulated steps exceed the configurable limit, the alarm steps accumulation limit arises. During accumulation the red LED is steady lit and the related status words bits are set.

V6 control firmware combines together the benefits of stepper systems: low cost, simplicity (no PID tuning), very low position overshoot, high torque/motor size ratio and the benefits of brushless systems: high efficiency (current adjustment with the load) and position retention.

Protections such as over-voltage, over-current and over-temperature are also implemented (low-voltage protection

inhibits the driver when the supply voltage is below a minimum allowed value).

Type	Rated voltage	Abs. Enc.	Digital I/O	Analog I/O	Ether CAT	Torque off	RS-232	RS-485	CAN open
FD1.1E	24 – 60 V_{DC}	✓	2 IN 1 OUT		✓		✓		
FD2.1E	24 – 130 V_{DC}	✓	7 IN 3 OUT		✓		✓		
FD2.1ET	24 – 130 V_{DC}	✓	5 IN 3 OUT		✓	✓	✓		

Tab. 1 – FD E drivers

Note:
FD1 and FD2 drives are also available with RS-485 Modbus and CANopen protocols.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

4. DWLOADER

DwLoader is a PC application which allows the user to program the driver and test its functionality. It can be useful to monitor drive parameters and solve issues, while it is controlled by the EtherCAT master.

It does not need any installation, just move the CD content into a PC folder (avoid blank spaces in folder path). Double click on “DwLoader.exe” and the main window will appear.

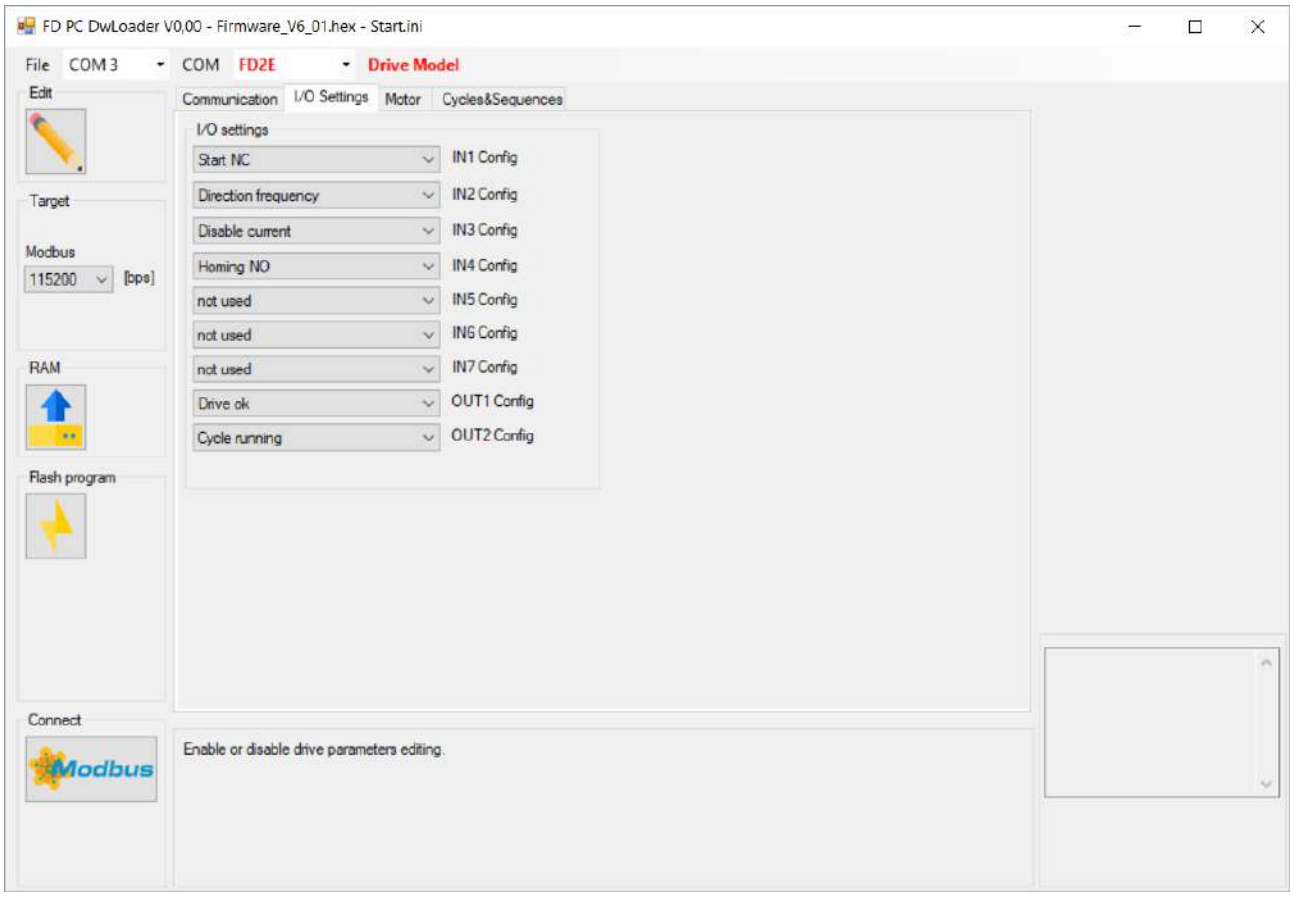


Fig. 3 – DwLoader main window

Note:
The COM port and the correct driver model in the menu strip of the window need to be selected (use Windows Device Manager to select the proper COM to be used).

The Microsoft .NET Framework 4.0 needs to be present (it is provided into the CD together with the DwLoader).

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

4.1. RAM data upload/download

Data upload reads from the target RAM all the data and update the DwLoader boxes accordingly.

4.2. User flash program

User flash program allows to completely reprogram the driver microprocessor (bootloader, firmware and parameters). Just a common PC and a 9 poles pin-to-pin RS-232 cable (or a simple USB to RS-232 converter) are needed.

1. Power off the driver.
2. Set the driver in programming mode by turning on DIP switch 8.
3. Power on the driver (FD1E: all LEDs off. FD2E: red LED will be on, all other LEDs will be all off).
4. Click on "User Flash Program" button. The command window will appear. In case of error, i.e. target does not answer or COM does not exist, a proper message will be shown.
5. Power off the driver (it might take time to discharge the energy accumulated in the capacitors, as the driver is absorbing very low power in programming mode).
6. Remove the programming mode, i.e. DIP switch 8 off.
7. Power on the driver again. Blinking green LED means that the programming procedure occurred properly.

Note:
Folder path cannot contain blank space characters (" ").

4.3. Data savings

When DwLoader is launched, it loads the parameters from previously used .ini file (if such file does not exist, the Start.ini file inside the DwLoader folder will be used). It is possible to load new parameters from File ⇨ Open and save them from File ⇨ Save / SaveAs. When saving a set of parameters into .ini file, Start.ini is automatically updated.

The actual file in use is displayed in the top bar of DwLoader window.

When closing the program, if any modification from the program start-up settings is present, it will ask to save.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

4.4. Data modify

Click on the pencil symbol to modify the driver parameters.

Name	Unit	Description
Communication		
RS-232 baud rate	[bps]	Range: 4˙800 – 115˙200 bps for RS-232

I/O settings			
		FD1.1E	FD2.1E
IN1 configuration		0: Disable current 1: Disable frequency 2: Homing NO 3: Homing NC 4: not used 5: Hardware limit switch up NO 6: Hardware limit switch up NC 7: Hardware limit switch down NO 8: Hardware limit switch down NC 9: Encoder position latch 10: Step frequency 11: Select cycle or sequence 12: Delta stop NO 13: Delta stop NC 14: Quadrature step A 15: Start NO 16: Start NC 17: Stop NO 18: Stop NC 19: Enable Current 20: Start NO, Stop NC 21: Start NC, Stop NO	0: Step frequency 1: Start NO 2: Start NC 3-5: not used 6: Quadrature step A 7: Start NO, Stop NC 8: Start NC, Stop NO
IN2 configuration		0: Disable current 1: Disable frequency 2: Homing NO 3: Homing NC 4: not used 5: Hardware limit switch up NO 6: Hardware limit switch up NC 7: Hardware limit switch down NO 8: Hardware limit switch down NC 9: Encoder position latch 10: Direction frequency 11: Select cycle or sequence 12: Delta stop NO 13: Delta stop NC 14: Quadrature step B 15: Start NO 16: Start NC 17: Stop NO 18: Stop NC 19: Enable Current 20: Start NO, Stop NC 21: Start NC, Stop NO	0: Direction frequency 1: Stop NO 2: Stop NC 3: Select cycle or sequence 4-5: not used 6: Quadrature step B 7: Start NO, Stop NC 8: Start NC, Stop NO
IN3 configuration		n.a.	0: Disable current 1: Disable Frequency 2: Homing NO

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

			3: Homing NC 4: not used 5: Hardware limit switch up NO 6: Hardware limit switch up NC 7: Hardware limit switch down NO 8: Hardware limit switch down NC 9: Encoder position latch 10: Direction frequency 11: Select cycle or sequence 12: Delta stop NO 13: Delta stop NC 14: not used 15: Start NO 16: Start NC 17: Stop NO 18: Stop NC 19: Enable current
IN4 configuration		n.a.	0: Disable current 1: Disable Frequency 2: Homing NO 3: Homing NC 4: not used 5: Hardware limit switch up NO 6: Hardware limit switch up NC 7: Hardware limit switch down NO 8: Hardware limit switch down NC 9: Encoder position latch 10: Direction frequency 11: Select cycle or sequence 12: Delta stop NO 13: Delta stop NC 14: not used 15: Start NO 16: Start NC 17: Stop NO 18: Stop NC 19: Enable current
IN5 configuration		n.a.	0-10: not used 11: Select cycle or sequence
IN6 configuration		n.a.	0-10: not used 11: Select cycle or sequence *Note:* *In FD2.1ET IN6 works as torque off 1*
IN7 configuration		n.a.	0-10: not used 11: Select cycle or sequence *Note:* *In FD2.1ET IN7 works as torque off 2*
OUT1 configuration		0: Cycle running, 1: Encoder index 50 msec 2: Encoder index Π 3: Position uploaded 4: Axis calibrated 5: Power on position ok 6: Torque limit 7: Cycle Running + /Dstop 8: Forced L 9: Forced H 10: Drive ok 11: Drive alarm	0: Drive ok 1: Drive alarm
OUT2 configuration		n.a.	0: Cycle running 1: Encoder index 50 msec 2: Encoder index Π

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

			3: Position uploaded 4: Axis calibrated 5: Power on position ok 6: Torque limit 7: Cycle Running + /Dstop 8: Forced L 9: Forced H
OUT3 configuration		n.a.	*t.b.d.* *Note:* *In FD2.1ET OUT3 works as torque feedback*

Motor		
Encoder enable		To enable encoder functionalities.
Position upload		Selected: the exact multi-turn power off position and currents configuration will be reloaded during power on (if the position deviation is inside ±1.8 °). Deselected: the power on position will be the power off position plus the position deviation (if the deviation is inside the Power-on calibration limit).
Software limit switch up	[step]	To enable and set the software limit switch towards increasing positions
Software limit switch down	[step]	To enable and set the software limit switch towards decreasing positions
Disable torque control		This function is used to disable the torque control and accumulation of steps. Just current reduction at motor stopped is implemented
Invert direction		To invert the direction of rotation: e.g. from CCW movement towards decreasing positions to CW movement towards decreasing positions
Starting boost		If checked, at every start of movement the motor will be driven with higher current to win static frictions and load inertia. Using this feature the torque control is anticipated allowing higher accelerations. After the boost, the current demand will go back to the torque-controlled values
Select 32 cycles		When cycle or sequence selection is made from a binary combination of digital inputs (configuring an input as Sel. Cyc. Seq. overrides fieldbus selection): select_32_cycles disabled: inputs select sequences [0-9] + cycles [10-31] select_32_cycles enabled: inputs select cycles [0-31]
Maximum motor current	[mA]	Maximum peak value of the sine wave phase current *Note: depends upon motor selection. Typ. value is $\sqrt{2}$ times motor rated current*
Minimum motor current	[mA]	Minimum peak value of the sine wave phase current *Note: approximately half of maximum motor current*
Step accumulation limit	[step]	Maximum number of accumulated steps (upper limited at 10 revolutions)
Time constant	[256 • 50 sec]	Preset to 1. DO NOT MODIFY
Reference resolution num.		To configure the number of steps per revolution: $$step_{REV} = \frac{50 \cdot Res_{NUM}}{Res_{DEN}}$$ Res_{NUM} and Res_{DEN} range from 0 to 65'535. The resolution is anyhow limited in steps per revolution from 400 to 204'800 step/rev.
Reference resolution den.		

Cycles		
Type	[1 – 7]	1: Jog 2: Relative indexer 3: Calibration 4: Delta stop 5: Absolute Indexer 6: Position delay 7: Time delay
Speed	[step / sec]	Range: 1 – 300'000 step/sec *Note: if higher angular velocities [rev/sec] are needed, slightly decrement the resolution [step/rev]*
Position	[step] or [msec]	Motor steps to be executed in an indexer relative cycle. Position destination in an indexer absolute cycle. Maximum motor steps to be executed in a delta stop cycle.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

		Steps of delay in a position delay cycle. Time delay in a time delay cycle.
Direction	[0-1]	When the direction is not inverted: 0: CW rotation to increasing positions 1: CCW rotation to decreasing positions
Delta stop	[step]	Programmable steps of movement after delta stop input activation.
Arrows for cycle selection	<< >>	To configure the movement data of 32 programmable cycles.

Sequences		
Sequence parameters [0, …, 19]		When a sequence is started, the motor will execute its parameters. Each of them can be one of the following: 0 – 31: Cycle number to be executed in sequence 254: Stop sequence 255: Loop sequence (restart the sequence from parameter number 0). When no stop and no loop are present in a sequence, the next sequence will be executed.
Arrows for sequence selection	<< >>	To configure the 10 programmable sequences.

Ramp profile		
Type		Linear, parabolic or S-curve.
Start / stop frequency	[step]	Selected: it enables and set the start / stop frequency. Deselected: automatic.
Acceleration	[1'000 step / sec^2]	e.g. 15 stands for an acceleration of 15'000 step / sec^2.
Deceleration	[1'000 step / sec^2]	e.g. 15 stands for a deceleration of 15'000 step / sec^2.

Calibration		
Calibration position	[step]	The position loaded in position counter at the end of a calibration cycle.
Homing		Type of calibration cycle to be implemented.
Position offset	[step]	The encoder magnet is mounted in a random angular position. This parameter modifies the zero-encoder position.
Power-on calibration limit	[step]	Programmable power-on position deviation to assess the axis calibration and recalculate the multi-turn axis position.
Research speed	[step / sec]	Speed to research the homing switch.
Release speed	[step / sec]	Speed to release from the homing switch.
Time stop	[msec]	Delay time between research and release speed in calibration cycles.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

4.5. Data Exchange

Connect Modbus button establishes a continuous communication with the target in RS-232 using Modbus protocol.

Data Exchange window appears, through which it is possible to perform current position, speed, alarms and temperature monitoring, commands execution, RAM data exchange, etc.

Faster green LED on the target means that communication is active.

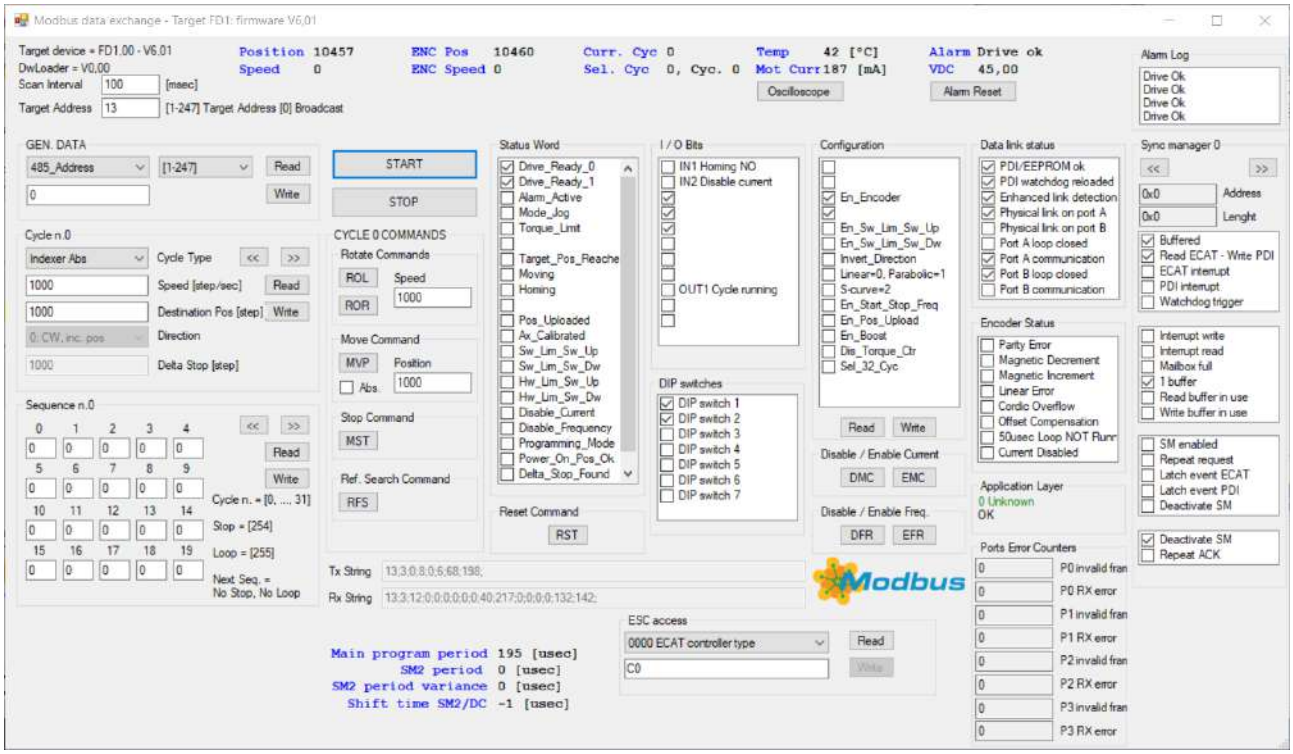


Fig. 4 – Modbus data exchange

Before clicking Connect Modbus, verify the baud rate in DwLoader window, otherwise no communication will take place and a timeout message will arise.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

4.6. Oscilloscope

Via Modbus data exchange window, clicking on Oscilloscope button (located in the bottom-left part of the window), it is possible to have a graphical representation of all the Modbus registers values with a time resolution of 1 msec.

For example, it is possible to control the drive via EtherCAT or I/O's, while plotting with DwLoader in RS-232 the selected data.

Using Start Command checkbox, the Time Analysis button will launch a Start command after 50 msec of selected data register sampling (the data buffer contains approximately 50 samples before Start command).

Write to File checkbox create a .txt file in the current folder with the time / data sampled.

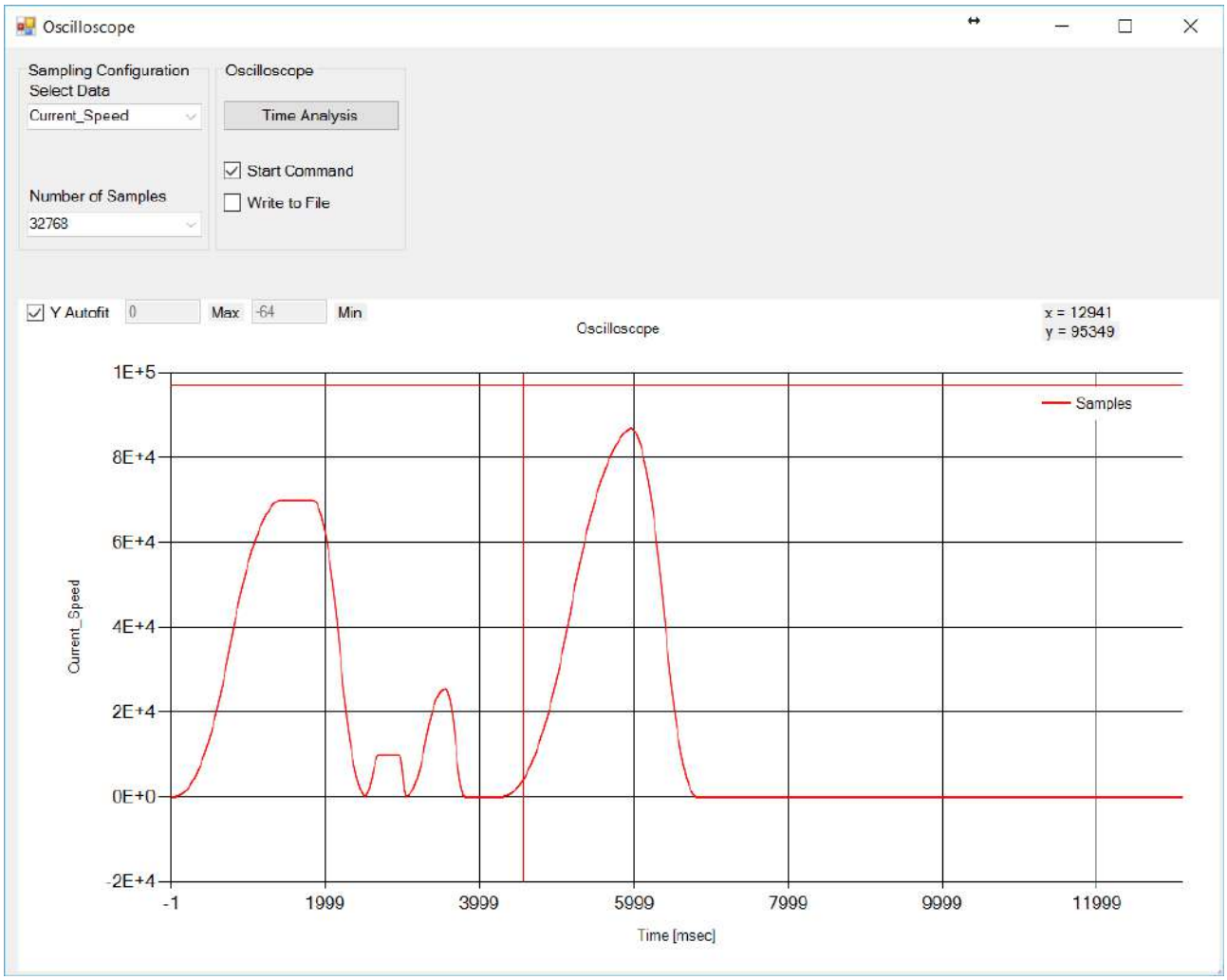


Fig. 6 – Oscilloscope

Note:
Modbus baud rate cannot be lower than 115˙200 bps.

Because many records are overlapped in time and then reconstructed, Number of Samples does not reflect the exact number of samples plotted. During the reconstruction, if more than 1% of samples are missing, record will be discarded. Otherwise missing samples will be interpolated.

Serial port need to be set with minimum latency (check Device Manager ⇨ Ports (COM & LPT) … ⇨ Advanced properties).

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

5. CONTROL METHODS

FD1E and FD2E drives can be controlled from an external system using following methods:

1. Step / dir or quadrature steps A/B

Two digital signals are transmitted from a motion-controller, which needs to be capable to generate acceleration and deceleration ramps following given position / velocity setpoint. In step/dir mode, at every positive edge of step signal the motor will rotate a configurable angle in CW or CCW depending on direction signal. In quadrature steps mode, the sequence of transitions of the two inputs A and B is evaluated to generate count pulses as well as the direction signal. Depending on the sequence the motor will move CW or CCW.

2. Start / stop and sequence selection

Digital inputs can be configured to select, start and stop up to 32 predefined movement, named cycles or 10 sequences of cycles (Ref. to 7). The control system can be a simple PLC, the drive itself will generate the acceleration ramps (linear, parabolic or s-curve (jerk). Motion profiles can be saved in flash or dynamically modified in RAM using fieldbus.

3. Modbus RTU

Over RS-232 cycles and sequences can be selected, started and stopped via Modbus RTU protocol.

4. EtherCAT

FD1E and FD2E implements CAN over EtherCAT (CoE), with following modes of operation:

- Profile position,
- Profile velocity,
- Homing mode,
- Interpolated position,
- Cyclic synchronous position (CSP),
- Cyclic synchronous velocity (CSV),
- Other custom modes,
- All Modbus registers are implemented as EtherCAT objects.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

6. INPUTS / OUTPUTS

FD drives have configurable inputs and outputs.

Type	Digital Inputs	Digital Outputs	Analogue Inputs
FD1E	2	1	-
FD2E	5	3	-

Tab. 3 – Inputs / Outputs

FD1E inputs and outputs are limited in number and all the possible configurations are available for each of them.
FD2E inputs are separated in groups, normally configured as: first two inputs are step modes or start/stop. Other two inputs are multipurpose, such as: homing, limit switch, delta stop, enable or disable current, etc. remaining inputs are for cycle or sequence selection. FD2E OUT1 is used for signaling the alarm status (normally closed or normally opened). OUT2 is multipurpose, i.e. cycle running, axle calibrated, etc. OUT3 is used as torque off feedback.

The status of all the inputs and outputs can be monitored from register IO_BITS.

6.1. Start/Stop and Input Frequency Signals

IN1 and IN2 can be configured as step/dir, quadrature A/B input frequencies, start/stop NO/NC and other extra functionalities such as cycle or sequence selection. Refer to table below:

Function	Drive: Input	Config. Reg. value: NO/NC	Description
Step frequency	FD1E: IN1	10	Every pulse produces a motor step. The maximum input frequency allowed is 300 kHz.
	FD2E: IN1	0	
Direction frequency	FD1E: IN2	10	Inactive down-counting CCW, Active up-counting CW. *Note:* *With invert dir is applied, motor sense of rotation is inverted.*
	FD2E: IN2	0	
Quadrature steps	FD1E: IN1, 2	14	Position counter counts up/down at every edge of signal depending on the level of the other signal.
	FD2E: IN1, 2	6	
Start	FD1E: IN1, 2 FD2E: IN3, 4	15: NO 16: NC	When active it starts the selected cycle or sequence.
	FD2E: IN1	1: NO 2: NC	
Stop	FD1E: IN1, 2 FD2E: IN3, 4	17: NO 18: NC	When active it stops the running cycle and/or sequence.
	FD2E: IN2	1: NO 2: NC	
Start, /Stop	FD1E: IN1, 2	20: Start NO, Stop NC 21: Start NC, Stop NO	When active it starts the selected cycle or sequence. Inactive, it stops it.
	FD2E: IN1, 2	7: Start NO, Stop NC 8: Start NC, Stop NO	

Tab. 4 – inputs description

Notes:
Fieldbus start commands are always available and managed with higher priority compared to step signals. Step signals are disabled when the motor is executing a command and they are enabled again at the end of the movement.
In step/dir. and quad. steps modes acceleration or deceleration ramps are not provided from the drive itself. The motion controller which is supplying the signals shall take care of.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Quadrature step signals can be used to connect an encoder Master to several drives, which will move synchronously to the encoder. They can be enabled/disabled using frequency enable signal and their speed can be modified thanks to the configurable resolution.

Hardware and software limit switches are not active in step/dir. and quad. step modes.

When the input is configured as step, measures to reduce external disturbances, such as shielded cables, need to be put in place. All the others input configurations have a digital filter and a Schmitt trigger with a maximum delay of 2.1 msec.

6.2. Multipurpose Inputs

Following functions are implemented:

Function	Drive: Input	Config. Reg. value	Description
Disable current	FD1E: IN1, 2 FD2E: IN3, 4	0	When active/inactive, it frees the motor shaft. Using the magnetic encoder, the drive can keep on counting the position of the shaft. As soon as the current is enabled again, position and status word bits are restored.
Enable current	FD1E: IN1, 2 FD2E: IN3, 4	19	
Disable frequency	FD1E: IN1, 2 FD2E: IN3, 4	1	When active, it inhibits all the movements from fieldbus commands or input steps, but the motor keeps on producing holding torque.
Homing	FD1E: IN1, 2 FD2E: IN3, 4	2: NO 3: NC	Homing sensor
Hardware limit switches	FD1E: IN1, 2 FD2E: IN3, 4	5: Up NO 6: Up NC 7: Down NO 8: Down NC	Motor movements are stopped in the direction of the switch. It can be used also in calibration cycles as homing sensor. They are not active in input step modes.
Encoder position latch	FD1E: IN1, 2 FD2E: IN3, 4	9	Rising and falling edges of configured input trigger a very fast interrupt, that samples and latches the multi-turn encoder position. Positions can be read from ENC_LATCH_RIS and ENC_LATCH_FAL registers.
Cycle, sequence selection	FD1E: IN1, 2 FD2E: IN3, 4	11	Binary selection of the cycle/sequence
Delta stop	FD1E: IN1, 2 FD2E: IN3, 4	12: NO 13: NC	Refer to delta stop cycle description.

Tab. 5 – Multipurpose inputs configuration

A digital low pass filter combined with a Schmitt trigger with 2.1 msec maximum delay has been applied to this two input channels in order to reduce noise effects. The low pass filter is removed when these inputs are set as delta stop, encoder latch or direction input frequency (setup time = 200 sec).

6.3. Aux inputs

FD2E is provided with another additional input IN5

Function	Drive: Input	Config. Reg. value	Description
Cycle, sequence selection	FD2E: IN5	11	Binary selection of the cycle/sequence

Tab. 6 – IN5 configuration

Other inputs configurations are available upon request.

A digital low pass filter combined with a Schmitt trigger with 2.1 msec maximum delay has been applied to this input channel in order to reduce noise effects.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

6.4. Torque off

FD2.1ET drive uses two inputs, IN6 and IN7, to disable motor current and free the shaft. They deactivate the torque-generating energy and prevent unintentional starting. This state is monitored internally by the drive.

Torque off can be used wherever the drive will be brought to a standstill in a sufficiently short time by the load torque or friction or where coasting down of the drive is not relevant (stop category 0).

This function cannot be used for safety, nor functional nor electrical (circuits remain powered).

The implementation is based on two digital inputs that independently deactivate the MOSFETs gate driver's supply voltage.

These two channels are continuously monitored by the drive microprocessor to verify the proper functioning.

PLC and other stop modules can as well periodically test the functionality of these channels receiving from OUT3 the feedback torque off active. As soon as one or both input gets active and a testing voltage goes off, OUT3 torque off feedback gets active, but the real deactivation of power stage has a delay, so that if the pulses are shorter than 100 msec, the power stage remains active and the motor keeps on being energized.

If only one input gets active, after 150 msec the microprocessor removes the power to the motor and torque off alarm is given. This alarm is not resettable. Power off/on cycle shall be performed.

Inputs are opto-isulated 24V active low, output is pnp type as shown on FD2.1E datasheet.

6.5. Outputs

OUT1 is a digital output. When it is used to monitor the drive status, it is recommended to be used in normally closed mode, i.e. 0: drive ok. Nevertheless, it can also be used in the normally opened mode, i.e. 1: alarm.

Function	Drive: Output	Config. Reg. value	Description
Drive ok	FD1E: OUT1	10	The output drive ok is normally closed and it opens in case of alarm (recommended)
	FD2E: OUT1	0	
Alarm	FD1E: OUT1	11	The output alarm is normally opened and it closes in case of alarm
	FD2E: OUT1	1	
Running	FD1E: OUT1 FD2E: OUT2	0	The output is closed during running cycles and sequences.
Encoder index 50 msec	FD1E: OUT1 FD2E: OUT2	1	The output is closed for 50 msec when the encoder position corresponds to the zero-encoder position (the zero-encoder position can be modified using POSITION_OFFSET register).
Encoder index Π	FD1E: OUT1 FD2E: OUT2	2	The output is high for half revolution and low for the other half. The commutation from low to high occurs at zero encoder position.
Position uploaded	FD1E: OUT1 FD2E: OUT2	3	The output is high if the power off position is exactly restored after power on. Ref. to 11.
Axle calibrated	FD1E: OUT1 FD2E: OUT2	4	The output is high if the axle is calibrated.
Power on position ok	FD1E: OUT1 FD2E: OUT2	5	The output is high if the power off position is within POWER_ON_CALIBRATION_LIMIT register value after power on. Ref. to 11.
Torque limit	FD1E: OUT1 FD2E: OUT2	6	The output is high when the motor is not able to follow the demand position.
Cycle running + not delta stop	FD1E: OUT1 FD2E: OUT2	7	This output activates at every start movement, as normal cycle running does, and it deactivates at the end of the movement, with the exception of delta stop cycles. If delta stop cycle is running, the output deactivates only if delta stop input gets active during movement. If the sensor is not found, the motor stops after the position parameter

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

			steps, but the output remains active. The host or PLC which manage this output needs to implement a time-out on this signal deactivation.
Forced	FD1E: OUT1 FD2E: OUT2	8: Open 9: Closed	Output can be forced to control/actuate an external device.
ECAT frame IRQ	FD1E: OUT1 FD2E: OUT1 FD2E: OUT2	31	Output gets on when the frame is received. It gets off when the RPDOs have been read from the ESC
ECAT DC SYNC	FD1E: OUT1 FD2E: OUT1 FD2E: OUT2	32: SYNC0 33: SYNC1	Output gets on when the SYNC is received. It gets off when the TPDO have been written into the ESC

FD2E implements also OUT3, which can be used as torque feedback (refer to torque off description).

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

7. ALARMS

In normal operation Modbus and EtherCAT error registers values are zero, status word alarm bit is off, output drive ok is closed, the status green LED is flashing and the alarm red LED is off.

If any of below alarm condition is triggered, status word, output, Modbus and EtherCAT error registers and LEDS signal the anomaly, motor gets immediately de-energized and the shaft freed.

In case of alarm, the red LED number of consequent blinks identify the Modbus ERR_FAT register value (when steady lit it express 1, step accumulation limit).

7.1. Steps accumulation limit

Modbus ERR_FAT = 1
EtherCAT error register = 0x81

FD drivers are provided with a 12-bit rotatory encoder, which measures the position of a shaft-mounted magnet.

When the motor is not able to execute the ordered steps, for example because it has not enough torque to win the load or because the axis is mechanically blocked, the difference between rotor position and rotating magnetic field position increases. Current control loop is able to keep producing the maximum motor torque, accumulating the steps that cannot be executed and chasing the ordered position. When the difference of ordered steps and executed steps exceeds the programmable limit, the steps accumulation limit alarm arises and the drive stops.
The limit is defined in Modbus as ACCUMULATION_LIMIT register, in EtherCAT as 0x6065, following error window. Such limit, i.e. the maximum position deviation allowed, depends from the application, it is expressed in steps, upper limited to 10 revolutions (128˙000 steps when resolution is configured as 12˙800 steps/rev), default value is one revolution.

When the load torque is so high to lose the synchronism with ordered steps, before that the step accumulation limit is reached, the motor keeps on provide maximum output torque, the red LED will turn on, torque limit bits (Modbus status word bit 4, EtherCAT status word bit 8) will be high.
As soon as the position deviation exceeds the limit, motor gets de-energized, Modbus ERR_FAT register will be equal to 1, EtherCAT status word fault bit will be activated (error register = 0x81, error code register = 0x8611), the red LED will be steady lit and the output Drive Ok turns off.

7.2. Over-Temperature Alarm

Modbus ERR_FAT = 2
EtherCAT error register = 0x09

The FD drives are provided with a temperature sensor. When the temperature exceeds 100 °C the over-temperature alarm arises and the driver is stopped.

Motor temperature depends on current settings, applied voltage, duty cycle and rotational speed.
Power dissipated for Joule effect in motor copper coils has quadratic trend with motor current. For this reason, to avoid over-temperatures, do not oversize the current settings.

Increasing rotating speed, the frequency of magnetic field increases, causing power dissipation through Foucault's currents in the iron. To avoid over-temperatures, do not oversize the speed reduction ratio.

In case of over-temperature alarm, Modbus ERR_FAT register value will be equal to 2, EtherCAT status word fault bit will be activated (error register = 0x09, error code register = 0x4310), and the red will make two flashes followed by a pause of two seconds.

7.3. Short Circuit Alarm

ERR_FAT = 3
EtherCAT error register = 0x03

This is a latched hardware fault that occurs if the current exceeds the maximum allowed value of the drive, which happens for example in the event of a short circuit on the motor wiring.
In case of short circuit alarm, Modbus ERR_FAT register will be equal to 3, EtherCAT status word fault bit will be activated

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

(error register = 0x03, error code register = 0x2300) and the red LED will flash three times followed by a pause of two seconds.

Note:
This is a hardware alarm. On FD1E is possible to reset this type of alarm only by turning off and on the device. On FD2E it is possible to reset it using Alarm Reset command.

7.4. Over-Voltage Alarm

ERR_FAT = 4
EtherCAT error register = 0x05

This is a latched hardware fault that occurs in the event of an over-voltage on the input DC power supply. This alarm could be caused by high deceleration of big inertial load, with weak external power supply capacitor.
Modbus ERR_FAT register will be equal to four, EtherCAT status word fault bit will be activated (error register = 0x05, error code register = 0x3210) and red LED will flash four times followed by a pause of two seconds.

FD overvoltage alarms are set as:

- FD1E 88 V_{DC},
- FD2E 135 V_{DC},

7.5. Data Error

ERR_FAT = 5
EtherCAT error register = 0x21

This alarm arises at power on in case of wrong settings in the flash memory. This alarm cannot be reset, contact the supplier.
Modbus ERR_FAT register will be equal to five, EtherCAT status word fault bit will be activated (error register = 0x21, error code register = 0x6320) and red LED will flash five times followed by a pause of two seconds.

7.6. Under Voltage

ERR_FAT = 7
EtherCAT error register = 0x05

FD1E logic is powered from both V_{EXT} and main power supply V_{POW} in parallel.
FD2E logic, instead, is powered from V_{EXT} only.

When V_{POW} is below 17 V_{DC} movement is stopped and under voltage alarm is triggered (if no other alarm comes first, e.g. step accumulation limit if the motor was rotating at high speeds).

Once the main power supply is restored (above 20 V_{DC}), alarms are automatically reset and multiturn position is restored. If the axis was calibrated, it remains so.

In case of under-voltage, Modbus ERR_FAT register will be equal to 7, EtherCAT status word fault bit will be activated (error register = 0x05, error code register = 0x3220) and red LED will flash seven times followed by a pause of two seconds.

7.7. Torque off error

ERR_FAT = 8
EtherCAT error register = 0x21

On the FD2E equipped with torque off functionality, 2 optocouplers are controlled on two channels via the TO1/ and TO2/ terminals. When requesting the TO via an external safety switching device, the auxiliary voltage supply channels of the power stage control circuits are switched off. Therefore, the power MOSFETs for the motor current cannot longer be switched on.
The microprocessor detects the failure of the optocoupler circuit of a channel by always checking both channels for similarity and it periodically activates the switches for the auxiliary voltage. If the hardware monitor detects a discrepancy for a defined time, or the switches don't work, the torque off error will be activated.

The processor signals this error externally by Modbus ERR_FAT register equal to 8, EtherCAT status word fault bit will be activated (error register = 0x21, error code register = 0x5493) and red LED will flash eight times followed by a pause of

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

two seconds the error code.

The only way to reset this alarm is to switch off and on the drive logic.

7.8. Alarm Resetting

In order to reset software alarms, the following possibilities are offered:

- Fault reset (EtherCAT bit 7 of 0x6040, control word)
- Alarm Reset (Modbus EXE_FUN = 15)
- Disable and enable current (using multipurpose inputs configured as disable/enable current, or by EXE_FUN = 16 then EXE_FUN = 17)
- System reset (EXE_FUN = 14) (refresh of RAM data from flash)
- ROL or ROR (EXE_FUN = 1 or 2) (refresh of RAM data from flash)
- Turn V_{POW} off and on
- Turn V_{EXT} off and on (refresh of RAM data from flash)

Fault/Alarm reset, disable/enable current and V_{POW} off/on are preferred, because after a System Reset, ROL/ROR and V_{EXT} off/on the alarm is reset, but the RAM memory will be re-initialized to Flash programmed data as it happens during a normal power on.

Status	LEDs	Error register
Drive ok	Red LED OFF Green LED blinking: 5 Hz communication ON 0.5 Hz communication OFF	Error = 0
~~Programming mode~~	~~Red and green LEDs blinking alternatively:~~ ~~5 Hz communication ON~~ ~~0.5 Hz communication OFF~~	~~Error = 0~~ ~~Modbus status word bit 18 high~~
Alarm	Red LED blinking (number of blinks represents the alarm code) Green LED same as Drive ok status	Error ≠ 0
Absolute encoder warning	Green LED short blink at 0,25 Hz	Error = 0 ENC_STATUS ≠ 0

Tab. 7 – Diagnostic LEDs and Errors

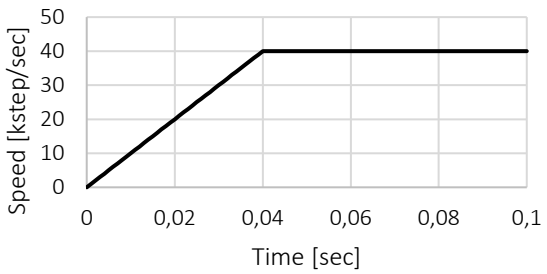
e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

8. RAMP PROFILES

V6 self generates linear, parabolic and s-curved speed ramps.

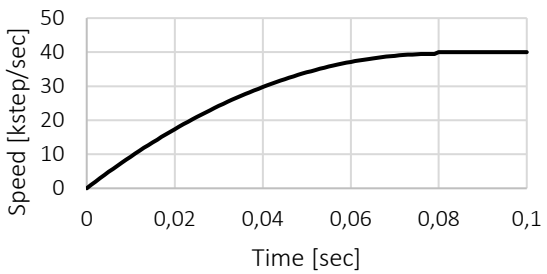
Linear ramp:
ACCELERATION and DECELERATION registers represent the speed increment / decrement every millisecond.

$$t_{acc} = \frac{V}{acc}$$



Parabolic ramp:
ACCELERATION and DECELERATION registers represent the speed increment / decrement per millisecond at zero speed. Increment and decrement are reduced at higher speeds.

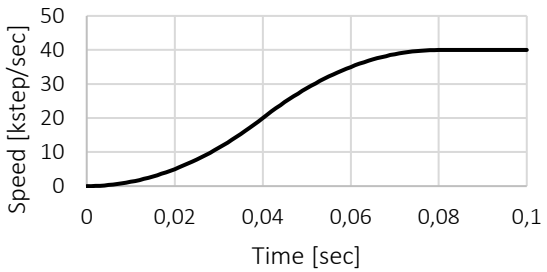
$$t_{acc} = \frac{2V}{acc}$$



Parabolic ramps, characterized by lower speed variations at higher speeds, are preferred in high speed and inertial load applications.

S-curve ramps:
ACCELERATION and DECELERATION registers represent the speed increment / decrement per millisecond at half speed. Increment and decrement are reduced at lower and higher speeds.

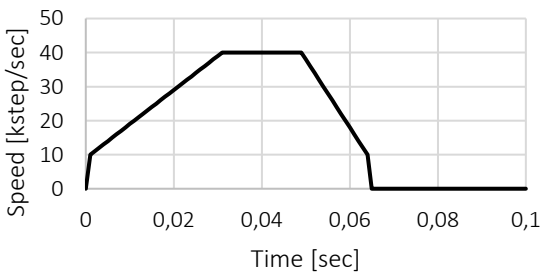
$$t_{acc} = \frac{2V}{acc}$$



Being the acceleration and deceleration a continuous function of time, motion is characterized by very smooth speed variations with the benefit of mechanical vibrations, wear and tear reduction.

Above graphs represent speed – expressed as thousands of steps per second – as a function of time. For all of them regime speed of 40˙000 step/sec and acceleration of 1˙000˙000 step/sec^2 have been used (ACCELERATION = 1000).

Start Stop Frequency defines the minimum frequency used in the ramp for both acceleration and deceleration calculation. This value must be set into the START_STOP_FREQ register and it shall be enabled setting the bit 9 of CONFIG register. Lower speeds are always available even if a higher start stop frequency is used.
In the graph regime speed of 40˙000 step/sec, start stop frequency of 10˙000 step/sec, acceleration of 1˙000˙000 µstep/sec^2 and deceleration of 2˙000˙000 step/sec^2.



When high acceleration is requested, starting boost functionality will allow a fast increment of motor current at motor start.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

9. POWER-ON POSITION RESTORE

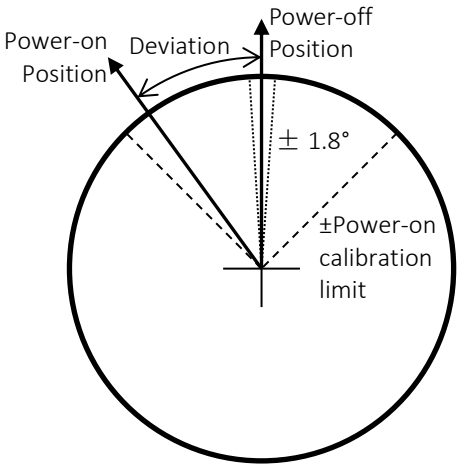


Fig. 17 – Position restore angles

Before the driver is completely powered off, V6 saves the actual encoder position and the status.
During the following power on, it is possible to evaluate the deviation with the new single-turn encoder position and the one saved with the scope of recovering the multiturn position and status from before. Following options are available.

9.1. Multi-turn position upload

Setting bit 11 POS_UPLOAD of CONFIG register the exact power-off position and status can be recovered at power-on. Stepper motors have a detent torque, which moves the rotor in the position of minimum reluctance when the drive is off (if the detent torque is bigger than the resistant torque). Thanks to the power off position saving, when the drive is turned on it compares the new absolute encoder position with the saved one. If the motor movement is inside ±1.8° threshold, it energizes the motor with the exact position before power off: the rotating field position and the multi-turn position registers will be exactly the same, as if it has never been powered off, Modbus STATUS_WORD bit 11 AX_CALIBRATED, and bit 6 TARGET_POS are restored and bit 10 POS_UPLOADED is set; EtherCAT 0x6041 bit 14 position uploaded is set, bit 12 homing attained in homing mode and bit 10 target reached in profile position mode are restored.

9.2. Multi-turn position corrected by deviation

Otherwise, if the first verification result is negative or position upload is disabled (bit 11 POS_UPLOAD of CONFIG register is reset) a second verification takes place: the position deviation (i.e. the motor movement when off) will be compared with the programmable POWER_ON_CALIBRATION_LIMIT register. If the deviation is inside the programmed limit the multi-turn position and the current configuration will be corrected using such deviation; AX_CALIBRATED bit will be restored, POWER_ON_POS_OK bit will be set and POS_UPLOADED bit will be reset.

9.3. Position inside the revolution

When both previous verification results are negative (the motor has been moved when off beyond power-on calibration limit), the new position value will be the absolute encoder position inside the revolution (i.e. not a multi-turn position), the AX_CALIBRATED and POS_UPLOADED bits will be reset.

It is possible to verify this feature moving the shaft when the motor is off: once powered on in a position beyond the Power On Calibration Limit the multi-turn position is lost.

Setting POWER_ON_CALIBRATION_LIMIT to zero and resetting bit 11 POS_UPLOAD of CONFIG register, the power on position will always be the absolute encoder position inside the revolution.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10. ETHERCAT

This chapter describes the installation, setup, range of functions and EtherCAT protocols for FD1E and FD2E Auxind drives.

10.1. Introduction

EtherCAT is an open high-speed fieldbus system that conforms to Ethernet (IEEE 802.3) for real-time distributed control, now standardized as IEC 61158.

EtherCAT is a registered trademark of Beckhoff Automation Gmbh (Germany). EtherCAT technology is protected by patents.

Each node achieves a short cycle time by transmitting Ethernet frames at high speed. A mechanism that allows sharing clock information enables high-precision synchronization control with low communications jitter.

The EtherCAT communication speed is up to 100 Mbps full duplex and can include a maximum of 65,535 stations in a single network configuration without using switches.

The frame is originated by the master and it is transmitted to the first slave in the network, traditionally connected in a ring topology. Each slave mounts an EtherCAT Slave Controller (ESC) that processes “on-the-fly” the frame. Each node in the network reads and writes the data from the frame as it passes through them. The frame is then sent back to the master, that can collect all the data written from the slaves.

Following graph describes a network of EtherCAT slaves in a ring topology. The Master controls the traffic in the network by initiating the transactions.

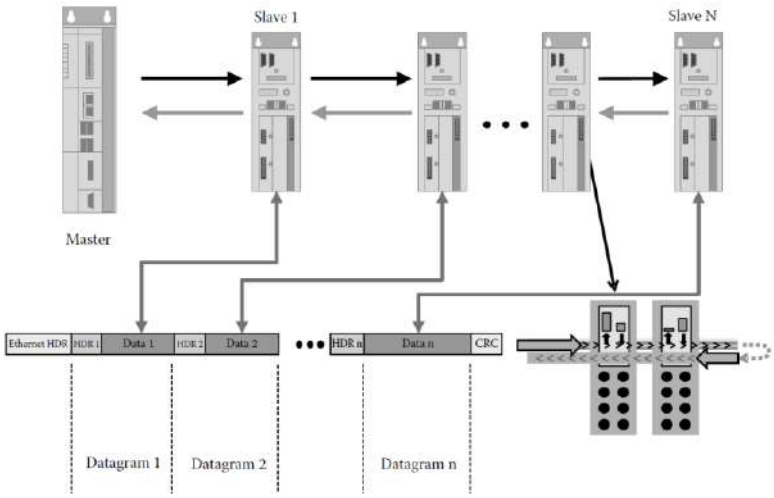


FIGURE 18.1 EtherCAT typical topology, with the *on-the-fly* frame processing.

The frames only delay by a fraction of a microsecond in each node. Via EtherCAT, the entire network can be addressed with just one frame.

EtherCAT telegram is encapsulated in an Ethernet frame and includes one or more EtherCAT datagrams destined to the EtherCAT slaves. Each datagram is a command that consists of a header, data and a working counter. The header and data are used to specify the operation that the slave must perform, and the working counter is updated by the slave to let the master to know that a slave has processed the command.

A control system requires the following in periodic time intervals:

- Inputs: messages from the slaves to the master such as status word, position actual value, speed, analogue or digital inputs, etc.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

- Outputs: messages from the master to the slave with control word, target position, target speed, etc.

The specific nature of the data transferred via the network depends on the mode of operation of the slave drive.

10.2. Main specifications

Item	Speficification
Communications standard	IEC 61158 Type 12, IEC 61800-7 CiA 402 Drive Profile
Physical layer	100BASE-TX (IEEE802.3)
Connectors	Circular connectors M12, 4 poles, D-coded, female
Communications media	Ethernet Category 5 (100BASE-TX) or higher (twisted-pair cable with double, aluminum tape and braided shielding) is recommended.
Communications distance	Distance between each node: 100 m max.
Process data	Configurable PDO mapping
Mailbox (CoE)	SDO requests, SDO responses, and SDO information
Distributed clock (DC)	Synchronization in DC mode. 1 ms, 2 ms, 4 ms
CiA402 Drive Profile	Cyclic synchronous position mode (CSP) Cyclic synchronous velocity mode (CSV) Profile potion mode (PP) Profile velocity mode (PV) Interpolated position mode (IP) Homing mode (HM)

10.3. Node address settings

The node address DIP switches (1-7) are used to set the EtherCAT node address.

DIP switch settings	Description
0	The EtherCAT master sets the node address
1 to 127	DIP switch setting is used as node address

Configured Station Alias Register 0x0012:
If DIP switches value is 0 nothing is done (value in register 0x0012 is loaded from EEPROM, independent if “0” or different value). If DIP switches value is different than 0, the DIP switches value is copied to configured station alias register during initialization phase.

Requesting ID mechanism:
FD drives use AL Status Code register 0x0134 for Device Identification value. The EtherCAT master can request the Device Identification value by setting bit 5 of AL Control register (0x0120.5 = ID Request). FD drive loads the current Device Identification Value to the AL Status Code register and set bit 5 of AL Status register (0x0130.5 = ID loaded) to indicate that its Device Identification Value is loaded. A changed Device Identification value, e.g. because the DIP switches are set to a different value, is loaded only with a new ID Request by the EtherCAT Master.

10.4. EtherCAT LED indicators

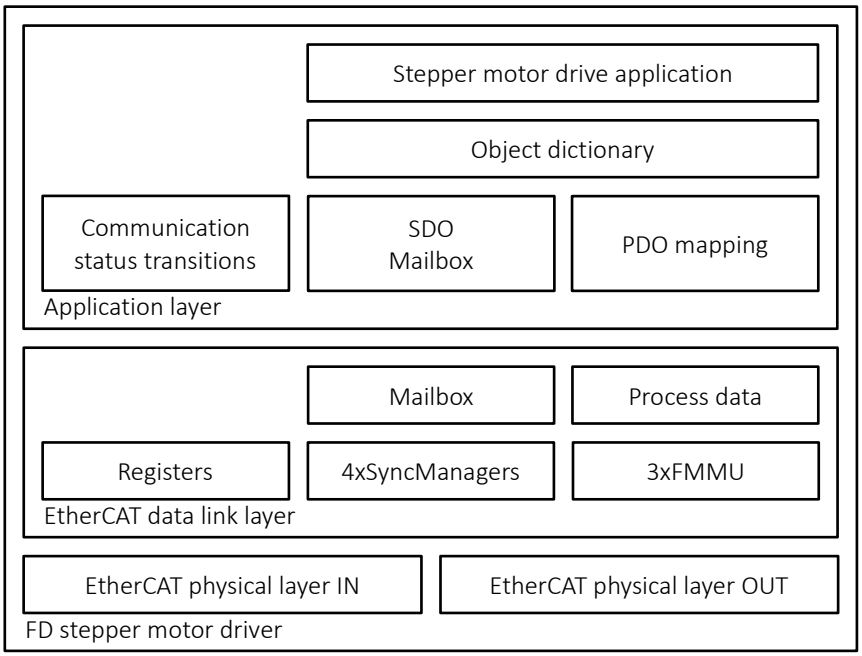
Name	Color	Status	Description
Run	Green	Off	Init
		Blinking	Pre-operational
		Single flash	Safe-operational
		On	Operational
Link/Activity IN/OUT	Green	Off	Link not established in physical layer
		On	Link established in physical layer, without activity
		Flickering	In operation

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Flickering is 50 msec on, 50 msec off.
Blinking is 200 msec on, 200 msec off.

10.1. CANopen over EtherCAT (CoE)

The structure of CANopen application over EtherCAT is represented in the following graph:



Normally, multiple protocols can be transmitted using EtherCAT. The IEC 61800-7 (CiA 402) drive profile is used.
The object dictionary in the application layer contains parameters and application data as well as information on the PDO mapping between the process data servo interface and stepper motor drive application.
The process data object (PDO) consists of objects in the object dictionary that can be mapped to the PDO. The contents of the process data are defined by the PDO mapping.
Process data communications cyclically reads and writes the PDO. Mailbox communications (SDO), instead, uses asynchronous message communications where all objects in the object dictionary can be read and written.

10.2. EtherCAT Slave Controller (ESC)

FD1E and FD2E are equipped with LAN9252 from Microchip, which is used as a 2 port ESC with dual integrated ethernet PHY. Each PHY contains a full-duplex 100BASE-TX transceiver and support 100 Mbps operation. The LAN9252 supports HP Auto-MDIX, allowing the use of direct connect or cross-over LAN cables.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

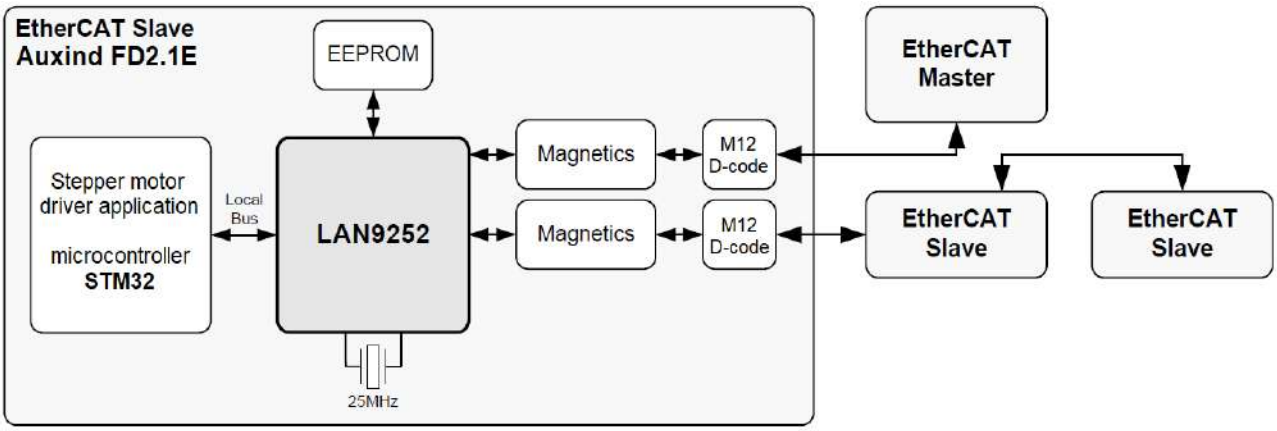


Fig. 1 - EtherCAT slave block diagram

The LAN9252 includes 4 KB of Dual Port memory (DPRAM) and 3 Fieldbus Memory Management Units (FMMUs). Each FMMU performs the task of mapping logical addresses to physical addresses.

The EtherCAT slave controller includes 4 SyncManagers to allow the exchange of data between the EtherCAT master and the stepper motor driver application. Each SyncManager's direction and mode of operation is configured by the EtherCAT master.

Two modes of communication are available: buffered mode or mailbox mode.
In the buffered mode, both the local microcontroller and EtherCAT master can write to the device concurrently. The buffer within the LAN9252 will always contain the latest data. If newer data arrives before the old data can be read out, the old data will be dropped. This mode is used for PDO.
In mailbox mode, access to the buffer by the local microcontroller and the EtherCAT master is performed using handshakes, guaranteeing that no data will be dropped. This mode is used for SDO.

10.2.1. FMMU

Fieldbus Memory Management Units (FMMU) convert logical addresses into physical addresses by the means of internal address mapping. Thus, FMMUs allow to use logical addressing for data segments that span several slave devices: one datagram addresses data within several arbitrarily distributed ESCs. Each FMMU channel maps one continuous logical address space to one continuous physical address space of the slave.

10.2.2. SyncManagers

LAN9252 is equipped with 4 KB of DPRAM. Stepper motor driver application and the master access this common memory area to communicate. SyncManagers enable consistent and secure data exchange between the EtherCAT master and the application and they are capable to generate interrupts to inform both sides of changes.

FD stepper motor drivers are equipped with 4 SyncManagers, the communication direction (in / out) and mode (mailbox / buffered) is configured as:
SyncManager 0 = master mailbox output
SyncManager 1 = master mailbox input
SyncManager 2 = master buffered output
SyncManager 3 = master buffered input

The mailbox mode implements a handshake mechanism for data exchange, so that no data will be lost. Each side, EtherCAT master or drive application, will get access to the buffer only after the other side has finished its access. At first, the producer writes to the buffer. Then, the buffer is locked for writing until the consumer has read it out. Afterwards, the producer has write access again, while the buffer is locked for the consumer. The mailbox mode is typically used for application layer protocol (SDO).

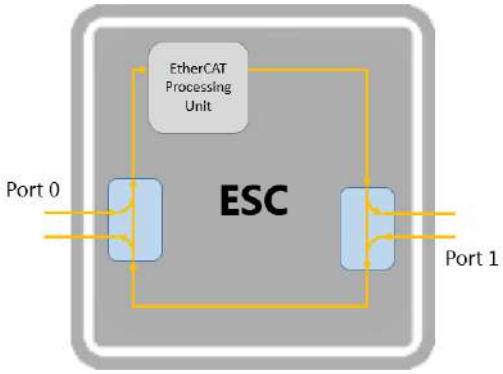
The mailbox mode only allows alternating reading and writing. This assures all data from the producer reaches the consumer. The mailbox mode uses just one buffer of the configured size. At first, after initialization/activation, the mailbox is writeable. Once it is written completely, write access is blocked, and the buffer can be read out by the other side. After it was completely read out, it can be written again. The time it takes to read or write the mailbox does not matter.

On the other hand, the buffered mode allows both sides, EtherCAT master and drive application, to access the communication buffer at any time. The consumer always gets the latest consistent buffer which was written by the producer, and the producer can always update the content of the buffer. The buffered mode allows writing and reading data simultaneously without interference. If the buffer is written faster than it is read out, old data will be dropped. The buffered mode is typically used for cyclic process data (PDO).

The buffered mode is also known as 3-buffer-mode. Physically, 3 buffers of identical size are used for buffered mode. The start address and size of the first buffer is configured in the SyncManager configuration. The addresses of this buffer have to be used by the master and the local application for reading/writing the data. Depending on the SyncManager state, accesses to the first buffer (0) address range are redirected to one of the 3 buffers.
One buffer of the three is allocated to the producer (for writing), one buffer to the consumer (for reading), and the third buffer keeps the last consistently written data of the producer.

10.3. EtherCAT functional overview

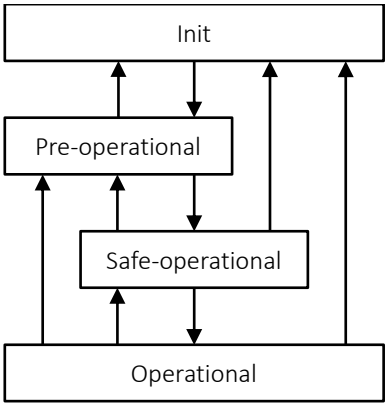
Each port receives an Ethernet frame, performs frame checking and forwards it to the next port. Time stamps of received frames are generated when they are received. The Loop-back function of each port forwards Ethernet frames to the next logical port if there is either no link at a port, or if the port is not available, or if the loop is closed for that port. The Loop-back function of port 0 forwards the frames to the EtherCAT Processing Unit. The loop settings can be controlled by the EtherCAT master.
Packets are forwarded in the following order: Port 0 → EtherCAT Processing Unit → Port 1.



The EtherCAT Processing Unit (EPU) receives, analyses and processes the EtherCAT data stream. The main purpose of the EtherCAT Processing unit is to enable and coordinate access to the internal registers and the memory space of the ESC, which can be addressed both from the EtherCAT master and from the drive application. Data exchange between master and slave application is comparable to a dual-ported memory (process memory), enhanced by special functions.
Distributed Clocks (DC) allow for precisely synchronized generation of output signals and input sampling, as well as
time stamp generation of events.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.4. EtherCAT state machine



The EtherCAT State Machine (ESM) of the EtherCAT slave is controlled by the EtherCAT Master and it is responsible for the coordination of master and slaves at start up and during operation.

In the following table, the available services in each state (RPDO are the PDO received: written from the master, read from the slave, e.g. control word; TPDO are the PDO transmitted: read from the master, written from the slave, e.g. status word).

State	SDO	RPDO	TPDO	Description
Init	X	X	X	Communications are being initialized. Communications are not possible. The master uses init state to initialize a set of configuration register. Mailbox sync managers are also configured in this state.
Pre-operational	✓	X	X	Only mailbox communications are possible in this state. This state is entered after initialization has been completed and mailbox has been setup. It is used to initialize network settings (configuration of SM and PDO mapping)
Safe-operational	✓	X	✓	In this state, PDO transmissions (not reception) are possible in addition to mailbox communications. DC mode cyclic communications can be used to send information such as status from the drive.
Operational	✓	✓	✓	This is a normal operating state. DC mode cyclic communications can be used to control the motor

State changes are requested by the master. The master requests a write to AL control register, which results in a register event in the drive. The drive responds to the change in 0x0120, AL control register through 0x0122, AL status register after a successful or failed state change. If the requested state change failed, the drive responds with the error flag set in AL status and with AL status code register that details the error occurred.

Requests of state transitions through 0x0120, AL control register are expressed as per following table:

Parameter	Data type	Value
State	4 bits	1: init 2: pre-operational 3: boot 4: safe-operational 8: operational
Acknowledge	1 bit	0: parameter Change of slave AL status register remains unchanged 1: reset parameter Change of slave AL status register
ID request	1 bit	0: ID is not requested 1: request of ID
reserved	2 bits	
reserved	1 Byte	

The drive answers via 0x0130, AL status register:

Parameter	Data type	Value
State	4 bits	1: init 2: pre-operational 3: boot

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

		4: safe-operational 8: operational
Change	1 bit	0: state transition successful 1: state transition not successful
ID loaded	1 bit	0: AL status code value does not represent ID 1: AL status code value represents drive ID
reserved	2 bits	
reserved	1 Byte	

In case of not successful state transitions, the drive explains the reason on 16 bits, 0x0134, AL status code register

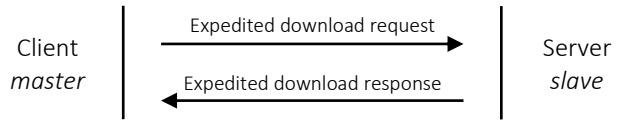
Code	Description	Current state	Resulting state
0x0000	No error	Any	Any
0x0011	Invalid request state change	I→S, I→O, P→O, O→B, S→B, P→B	I + error, P + error, S + error
0x0012	Unknown requested state	Any	I + error, P + error, S + error
0x0013	Boot not supported	I→B	I + error
0x0016	Invalid mailbox configuration	I→P	I + error
0x0017	Invalid sync manager configuration	P→S, S→O	Current state + error
0x001B	Sync manager watchdog	O, S	S + error
0x001D	Invalid output configuration	O, S, P→S	P + error
0x001E	Invalid input configuration	O, S, P→S	P + error
0x0036	Invalid SYNC0 cycle time	P→S	P + error
0x0037	Invalid SYNC1 cycle time	P→S	P + error
0x0061	Device identification value updated	P	P + error

10.5. SDO

With the SDO services the entries of the object dictionary can be read or written.
The first Byte of the first segment specifies the flow control information (download, upload, size, expedited, etc.), the next three Bytes specifies the index and sub-index, the last bytes are size or data.
When the data size exceeds the mailbox size, other segments shall be transmitted. In such segments, the first Byte specifies control information and remaining Bytes, the data.
The receiver, confirms each segment or each block of segments, so that a peer-to-peer communication (client/server) takes place.

10.5.1. SDO download expedited

Following graph shows the primitives between client and server in case of a successful single SDO download expedited sequence:



All the information can be grouped in standard SDO frame part, made of 8 Bytes.

The message sent from the master to the slave:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

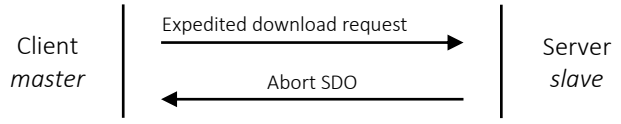
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	Size indicator	1 b	0x01, Data set size is specified
	Transfer type	1 b	0x01, Expedited transfer
	Data set size	2 b	0x00, 4 Bytes 0x01, 3 Bytes 0x02, 2 Bytes, 0x03, 1 Byte
	Complete access	1 b	0x00, entry addressed with index and sub-index will be downloaded 0x01, complete object will be downloaded, sub-index shall be zero (sub-index 0 included) or one (sub-index 0 excluded)
	Command specifier	3 b	0x01, download request
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	Data	4 Bytes	Data of the object, unused Bytes shall be 0

The successful answer from the slave to the master:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x03, SDO response
SDO	Size indicator	1 b	0x00
	Transfer type	1 b	0x00
	Data set size	2 b	0x00
	Complete access	1 b	0x00
	Command specifier	3 b	0x03, download response
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	reserved	4 Bytes	

10.5.2. SDO abort

The unsuccessful answer is through an Abort SDO transfer, whose primitives are represented in the following graph:



Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	Size indicator	1 b	0x00
	Transfer type	1 b	0x00
	Data set size	2 b	0x00
	Complete access	1 b	0x00
	Command specifier	3 b	0x04, abort transfer request
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object
	Abort code	4 Bytes	Abort code specified in following table

The implemented abort codes are:

Name	Code	Meaning
TOGGLE_NOT_CHANGED	0x05030000	Toggle bit not changed
COMMAND_NOT_VALID	0x05040001	Client/server command specifier not valid or unknown
UNSUPPORTED	0x06010000	Unsupported access to an object
READ_ONLY	0x06010002	Attempt to write a read only object
SUBIX_ZERO_NOT_ZERO	0x06010003	Sub-index cannot be written, sub-index 0 must be 0 for write access
OBJ_NOT_EXIST	0x06020000	The object does not exist in the object dictionary
OBJ_NOT_MAPPABLE	0x06040041	The object cannot be mapped into the PDO
MAPPING_TOO_LONG	0x06040042	The number and length of the objects to be mapped would exceed the PDO length
PARAM_INCOMPATIBLE	0x06040043	General parameter incompatibility reason
GEN_INTERN_INCOMP	0x06040047	General internal incompatibility in the device
DATA_TYPE_LENGHT_NOT_OK	0x06070010	Data type does not match, length of service does not match
SUB_IX_NOT_EXIST	0x06090011	Sub-index does not exist
WRITE_RANGE_EXCEEDED	0x06090030	Value range of parameter exceeded (only for write access)
VALUE_TOO_HIGH	0x06090031	Value of parameter written too high
VALUE_TOO_LOW	0x06090032	Value of parameter written too low
GENERAL_ERROR	0x08000000	General error
NOT_IN_THIS_STATE	0x08000022	Data cannot be transferred or stored to the application because of present device state

10.5.3. SDO download normal

With this type of service, the length is variable, doing so the data can be transferred all in one frame if the mailbox is big enough, otherwise via segmented transfers.

The SDO download normal request sent from the master to the slave is:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n ≥ 0x0A, length of mailbox service data
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	Size indicator	1 b	0x01, Data set size is specified
	Transfer type	1 b	0x00, normal transfer
	Data set size	2 b	0x00
	Complete access	1 b	0x00, entry addressed with index and sub-index will be downloaded 0x01, complete object will be downloaded, sub-index shall be zero (sub-index 0 included) or one (sub-index 0 excluded

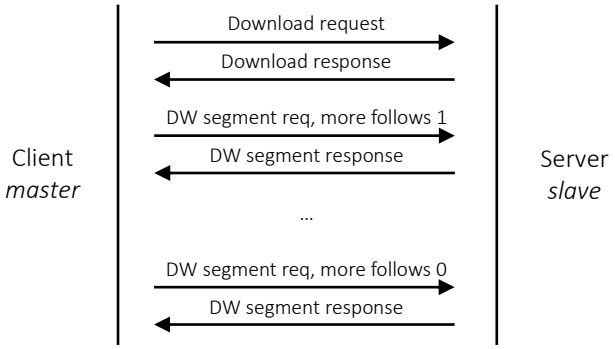
e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

	Command specifier	3 b	0x01, download request
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	Complete size	4 Bytes	Complete data size of the object
	Data	n-10 Bytes	If ((Length-10) ≥ complete size): data of the object If ((Length-10) < complete size): first data part of the object, Download SDO Segment is following

The attribute types and coding of SDO download normal response are the same as of SDO download expedited response.

When the dimension of the downloaded object is larger than the dimension of the mailbox, segmented transfers take place.

In case of segment, following primitive is implemented:



The SDO segment request sent from the master to the slave is:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n ≥ 0x0A, length of mailbox service data
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	More follows	1 b	0x00: Download SDO Segment is following 0x01: last Download SDO Segment
	SegData size	3 b	Defines how much of the last 7 Bytes (which always has to be send) contain data (only applicable if Length is 0x0A, otherwise shall be 0x00): 0x00: 7 Bytes 0x01: 6 Bytes 0x02: 5 Bytes 0x03: 4 Bytes 0x04: 3 Bytes 0x05: 2 Bytes 0x06: 1 Byte 0x07: 0 Bytes
	Toggle	1 b	Shall toggle with every Download SDO Segment Request, starting with 0x00
	Command specifier	3 b	0x00: Download Segment Request
	Data	n-3 Bytes	data of the object

The SDO segment response sent from the slave to the master is:

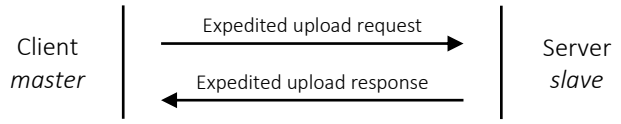
Frame part	Data field	Type	Description

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Mailbox header	Length	Word	0x0A, length of mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x03, SDO response
SDO	reserved	4 b	0x00
	Toggle	1 b	Shall be the same as for the corresponding Download SDO Segment Request
	Command specifier	3 b	0x01: Download Segment Response
	Data	7 Bytes	0x00

10.5.4. SDO upload expedite

Following graph shows the primitives between client and server in case of a successful single SDO upload expedited sequence:



All the information can be grouped in standard SDO frame part, made of 8 Bytes.

The message sent from the master to the slave:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	Reserved	4 b	0x00
	Complete access	1 b	0x00, entry addressed with index and sub-index will be uploaded 0x01, complete object will be uploaded, sub-index shall be zero (sub-index 0 included) or one (sub-index 0 excluded)
	Command specifier	3 b	0x02, upload request
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	Reserved	4 Bytes	0x00

The successful answer from the slave to the master:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, lowest priority
	Type	4 b	0x03, CoE

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x03, SDO response
SDO	Size indicator	1 b	0x01, size of data in data set size specified
	Transfer type	1 b	0x01, expedited transfer
	Data set size	2 b	0x00, 4 Bytes 0x01, 3 Bytes 0x02, 2 Bytes, 0x03, 1 Byte
	Complete access	1 b	0x00, entry addressed with index and sub-index will be downloaded 0x01, complete object will be uploaded, sub-index shall be zero (sub-index 0 included) or one (sub-index 0 excluded)
	Command specifier	3 b	0x02, upload response
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	Data	4 Bytes	Data of the object

10.5.5. SDO upload normal

With this type of service, the length is variable, doing so the data can be transferred all in one frame if the mailbox is big enough, otherwise via segmented transfers.

The SDO upload normal request sent from the master to the slave is the same as SDO upload expedited request:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	Reserved	4 b	0x00
	Complete access	1 b	0x00, entry addressed with index and sub-index will be uploaded 0x01, complete object will be uploaded, sub-index shall be zero (sub-index 0 included) or one (sub-index 0 excluded)
	Command specifier	3 b	0x02, upload request
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	Reserved	4 Bytes	0x00

The SDO upload normal response from the slave to the master, if the message fits in the mailbox is:

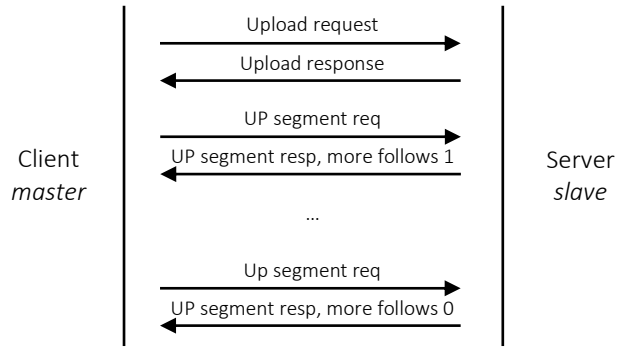
Frame part	Data field	Type	Description
Mailbox header	Length	Word	n ≥ 0x0A, length of mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

	reserved	3 b	0x00
	Service	4 b	0x03, SDO response
SDO	Size indicator	1 b	0x01
	Transfer type	1 b	0x00, normal transfer
	Data set size	2 b	0x00
	Complete access	1 b	0x00, entry addressed with index and sub-index will be downloaded 0x01, complete object will be uploaded, sub-index shall be zero (sub-index 0 included) or one (sub-index 0 excluded)
	Command specifier	3 b	0x02, upload response
	Index	Word	Index of the object
	Sub-index	Byte	Sub-index of the object 0 or 1 if complete access is used
	Complete size	4 Bytes	Complete size of data of the object
	Data	n-10 Bytes	If ((Length-10) >= Complete Size): data of the object If ((Length-10) < Complete Size): first data part of the object, upload SDO segment is following

When the dimension of the uploaded object is larger than the dimension of the mailbox, segmented transfers take place.

In case of segment, following primitive is implemented:



The upload SDO segment request sent from the master to the slave is:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x02, SDO request
SDO	Reserved	4 b	0x00
	Toggle	1 b	Shall toggle with every upload SDO segment request, starting with 0x00
	Command specifier	3 b	0x03: Upload Segment Request
	Reserved	7 Bytes	

The upload SDO segment response sent from the slave to the master is:

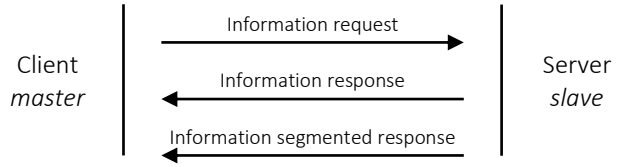
Frame part	Data field	Type	Description
Mailbox header	Length	Word	n ≥ 0x0A, length of mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x03, SDO response
SDO	More follows	1 b	0x00: Upload SDO Segment is following 0x01: last Upload SDO Segment
	SegData Size	3 b	Defines how much of the last 7 Bytes (which always has to be send) contain data (only applicable if Length is 0x0A, otherwise shall be 0x00): 0x00: 7 Bytes 0x01: 6 Bytes 0x02: 5 Bytes 0x03: 4 Bytes 0x04: 3 Bytes 0x05: 2 Bytes 0x06: 1 Byte 0x07: 0 Bytes
	Toggle	1 b	Shall be the same as for the corresponding Upload SDO Segment Request
	Command specifier	3 b	0x00: Upload Segment Response
	Data	n-3 Bytes	Data part of the object

10.5.6. SDO information

With SDO information services the object dictionary of a server can be read by a client. The primitives of the SDO information services are mapped to the primitives of the mailbox transmission services.



Frame part	Data field	Type	Description
Mailbox header	Length	Word	n > 0x06: length of the mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x01, get OD list request 0x02, get OD list response 0x03, get object description request 0x04, get object description response 0x05, get entry description request 0x06, get entry description response 0x07, SDO info error request
	Incomplete	1 b	0: last SDO information fragment 1: SDO information fragments will follow
	reserved	8 b	
	Fragments left	Word	Number of fragments which follow
SDO info service data	Data	Byte[n-6]	SDO information service data

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.5.7. Get OD list

Get OD list request is as follows:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n = 0x08: length of the mailbox service data
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x01, get OD list request
	Incomplete	1 b	0x00
	Reserved	8 b	0x00
	Fragments left	Word	0x00
SDO info service data	List type	Word	0x00: get number of objects in the 5 different lists 0x01: all objects of the object dictionary shall be delivered in response The other options are not supported.

The response, containing the number of entries or the list is:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n >= 0x08: length of the mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x02, get OD list response
	Incomplete	1 b	0: last SDO information fragment 1: SDO information fragments will follow
	Reserved	8 b	0
	Fragments left	Word	Number of fragments which follow SDO info header will be sent with every fragment, SDO info data will be sent fragmented
SDO info service data	List type	Word	0x00: get number of objects in the 5 different lists 0x01: all objects of the object dictionary shall be delivered in response The other options are not supported.
	Index list	Words [n-8/2]	5 words with the length of the list types if list type is 0. List of object indexes if list type is 1.

The SDO info service data of the get OD list response is segmented and the data are fragmented.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.5.8. Get object description

Get object description request is as follows:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n = 0x08: length of the mailbox service data
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x03, get object description request
	Incomplete	1 b	0x00
	reserved	8 b	0x00
	Fragments left	Word	0x00
SDO info service data	Index	Word	Index of the requested object description

The response is:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n > 0x0A: length of the mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x04, get object description response
	Incomplete	1 b	0: last SDO information fragment
	reserved	8 b	0x00
	Fragments left	Word	Number of fragments which follow
SDO info service data	Index	Word	Index of object
	Data type	Byte	Reference to data type list
	Max sub-index	Byte	Maximum number of sub-indexes of the object
	Object code	Byte	7: variable 8: array 9: record
	Name	Char[n-12]	Name of the object

Data type list implemented:

Index (hex)	Type
0x0002	Integer 8
0x0003	Integer 16
0x0004	Integer 32
0x0005	Unsigned 8
0x0006	Unsigned 16
0x0007	Unsigned 32
0x0009	Visible string
0x001B	Unsigned 64

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.5.9. Get entry description

Get entry description request is as follows:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A: length of the mailbox service data
	Address	Word	Slave address
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x05, get entry description request
	Incomplete	1 b	0x00
	Reserved	8 b	0x00
	Fragments left	Word	0x00
SDO info service data	Index	Word	Index of the requested object description
	Sub-index	Byte	Sub-index of the requested object description
	Value info	Byte	The value info includes which elements shall be in the response: b3: unit type b4: default value b5: minimum value b6: maximum value

The response is as follow:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	n >= 0x10: length of the mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x06, get entry description response
	Incomplete	1 b	0: last SDO information fragment 1: SDO information fragments will follow
	Reserved	8 b	0
	Fragments left	Word	Number of fragments which follow
SDO info service data	Index	Word	Index of object
	Sub-index	Byte	Sub-index of the requested object description
	Max sub-index	Byte	Maximum number of sub-indexes of the object
	Value info	Byte	The value info includes which elements shall be in the response: b3: unit type b4: default value b5: minimum value b6: maximum value
	Data type	Word	Index of the data type of the object
	Bit length	Word	Bit length of the object
	Object access	Word	b0: read in preop b1: read in safeop b2: read in op b3: write in preop b4: write in safeop b5: write in op b6: mappable in RPDO

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

			b7: mappable in TPDO
	Data	Byte[n-16]	If requested: The unit type of the object (double word) The default value (same data type of the object) The minimum value (same data type of the object) The maximum value (same data type of the object) If the length is greater than the described response parameter, the description is following (array of char)

10.5.10. Unit types

Unit types are 32 bits information, defined in ETG.1004, organized as per following scheme:

31 – 24	23 – 16	15 – 8	7 – 0
Prefix	Numerator	Denominator	reserved
MSB			LSB

Prefixes implemented are:

Name	Power	Code
-	10^0	0x00
kilo	10^3	0x03
centi	10^{-2}	0xFE
milli	10^{-3}	0xFD
nano	10^{-9}	0xF7

Following notation index, expressed as numerator and denominator, are implemented:

Name	Notations	Code
-	-	0x00
seconds	s	0x03
Ampere	A	0x04
Hertz	Hz	0x20
Volt	V	0x26
Celsius	°C	0x2D
seconds square	s^2	0x57

10.5.11. SDO info error

This service is used from the slave to signal to the master that something is wrong in its request.

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A: length of the mailbox service data
	Address	Word	0x00
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7, counter of mailbox services
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x08, SDO information
SDO info header	Opcode	7 b	0x08, SDO info error request
	Incomplete	1 b	0x00
	Reserved	8 b	0x00
	Fragments left	Word	0x00
SDO info service data	Abort code	4 Bytes	Abort code

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.5.12. Complete access

Compared to CAN, where the number of Bytes for SDO protocol is limited to 8 only, EtherCAT allows larger mailbox size. Auxind drives implements 128 Bytes in reception on SyncManager 0 and 128 Bytes in transmission on SyncManager 1 (headers included).
Thanks to larger mailbox a complete record can be read/written within a single service, this is achieved through complete access: all sub-indexes of an object are uploaded or downloaded with a single SDO service, while with CAN it is possible to read a single sub- index at a time. Consequently, complete access minimizes boot-up time of the device.
Following rules are used for the correct alignment of the data:

1. Sub-index 0 is padded to 16 bits
2. Sub-index of bit data type (BOOLEAN and BIT1 to BIT7) will be filled over the byte border, the next non-bit data type will start at the next BYTE address e.g. object 0x10F3 SI4 = BOOL, SI5 = UINT16 → Bit-Offset of SI5 = 48 (not 41)
3. To define gaps in the byte stream of a complete access, it is allowed to define gap entries (Data_Type = 0, Bit_Length = gap size in bits) The value of gaps shall be 0.
4. Sub-indexes that don't exist will not need space
5. The complete access can start with sub-index 0 or sub-index 1, other sub- indexes are not allowed
6. All objects that fit in the mailbox are accessible by SDO with complete access
7. With segmented SDO transfer also bigger objects are accessible by SDO with complete access
8. Objects with dynamic entries (i.e. "OCTET_STRING", "UNICODE_STRING", "ARRAY_OF_*" or "VISIBLE_STRING") cannot be accessed by complete access.
9. Objects that cannot be accessed return the error code 0x06010000 (Unsupported Access) or 0x6010004 (Complete access not supported).
10. The length of the responded Complete Access data shall either match to the current byte length of the object or to the Bit Size of the object description.
11. The current maximum sub-index may exceed the number of entry descriptions of the offline object dictionary.
12. The SDO download complete access data length shall always match the full current object size (defined by sub-index 0).
13. Entries of data type string have a fixed length in the entry description to be used for complete access. The string itself can be shorter. Unused Bytes are filled with 0.

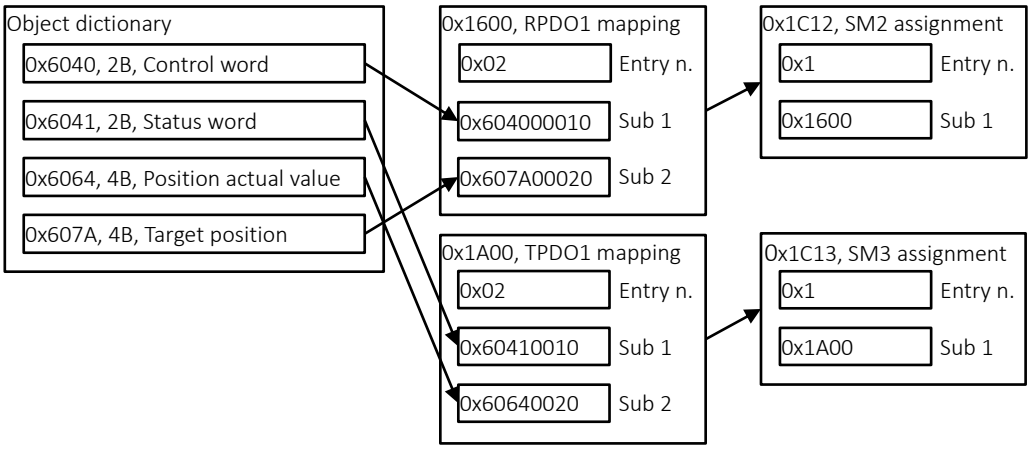
SDO Complete Access Example with SubIndex 1			SDO Complete Access Example with SubIndex 0		
Sub-index	Data type	Bit offset	Sub-index	Data type	Bit offset
1	BOOLEAN	0	0	UNSIGNED8	0
2	BIT2	1	1	BOOLEAN	16
3	BIT3	3	2	BIT2	17
4	0 = Gap, Length = 2	6	3	BIT3	19
10	UNSIGNED8	8	4	0 = Gap, Length = 2	22
11	UNSIGNED16	16	10	UNSIGNED8	24
12	UNSIGNED32	32	11	UNSIGNED16	32
13	UNSIGNED8	64	12	UNSIGNED32	48
14	OCTET-STRING[4]	72	13	UNSIGNED8	80
15	VISIBLE-STRING[7]	104	14	OCTET-STRING[4]	88
16	INTEGER32	160	15	VISIBLE-STRING[7]	120
17	UNSIGNED64	192	16	INTEGER32	176

10.6. PDO mapping

The process data objects, PDOs, are used to transfer data during cyclic communications in real-time. PDO can be reception PDO (RPDO), which receive data from the controller, or transmission PDO, (TPDO), which send status from the drive to the host controller.
The EtherCAT application layer can hold multiple objects to enable process data. The contents of the process data are described in the PDO mapping object and the SyncManager PDO assignment object, addressed in following indexes via SDO:

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

- 0x1600 – 0x1603 Receive PDO 1 to 4 mapping
- 0x1A00 – 0x1A03 Transmit PDO 1 to 4 mapping
- 0x1C12, 0x1C13 SyncManager 2 and 3 PDO assignment



Mapping parameter configured in RPDO and TPDO mapping is made of:

31 … 16	15 … 8	7 … 0
Index	Sub-index	Number of bits

MSB LSB

To change the content of these objects the following procedure shall be done:
1. set sub-index 0 = 0. The object is considered disabled when sub-index 0 has the value 0.
2. configure the mapping information in the mapping entries sub-index 1 … 8
3. set sub-index 0 = number of used mapping entries

Each PDO mapping record can also be configured using complete access in a single SDO download service.

FD drives implements 4 RPDO, and 4 TPDO each of them contains a maximum number of 8 objects for a total of 8 Bytes, and a total limit of 4*64 Bytes in reception and 4*64 Bytes in transmission.

The object mapping can be changed only when the EtherCAT communications state is in pre-operational. Since the mappings is not saved in EEPROM, the master shall transmit the configuration each time the drive is turned on.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.7. Emergency messages

Emergency messages are triggered by the occurrence of drive internal errors. The transmission is executed via the mailbox interface (SyncManager 1).

The message is structured as follows:

Frame part	Data field	Type	Description
Mailbox header	Length	Word	0x0A
	Address	Word	0x00, Master
	Channel	6 b	0x00
	Priority	2 b	0x00, Lowest priority
	Type	4 b	0x03, CoE
	Counter	3 b	1 – 7
	reserved	1 b	0x00
CoE header	Number	9 b	0x00
	reserved	3 b	0x00
	Service	4 b	0x01, Emergency
Emergency	Error code	Word	Ref. to object 0x603F
	Error register	Byte	Ref. to object 0x1001
	Data	5 Bytes	Error code 0000 – 9FFF: manufacturer specific error field Error code A000 – EFFF: diagnostic data Error code A000 – EFFF: manufacturer specific error field
	reserved	n-10 Bytes	0x00

Refer to 0x603F, error code for the list of implemented codes.

10.8. Synchronization

Following synchronization modes are available:

- Free run

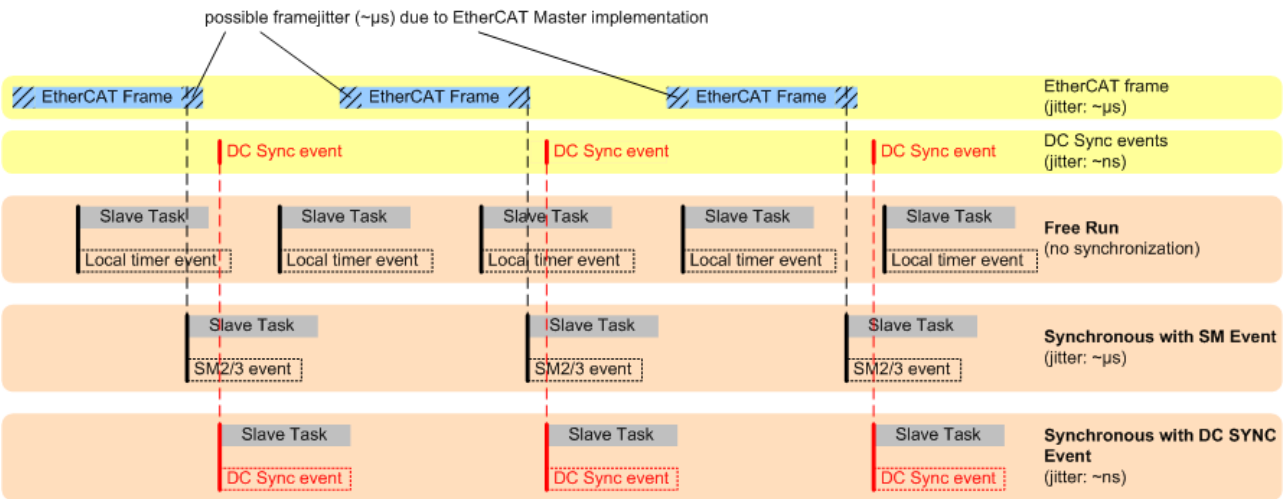
Drive's application is not synchronized with EtherCAT

- Synchronous with sync manager (SM2) event

Drive's application is synchronized with SM2 event. SM2 events are based on the time an EtherCAT frame is received. This time can jitter in the range of a few microseconds due to the EtherCAT Master implementation (delay in Stack, PHY & MAC Delay, etc).

- Synchronous with DC SYNC event

Slave's application is synchronized to the SYNC0 or SYNC1 event, which are based on the distributed clocks (DC) unit. The jitter could be reduced to a few nanoseconds.



e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

The different synchronization types can be identified by the different combinations of the sub-indexes of 0x1C32 and 0x1C33.

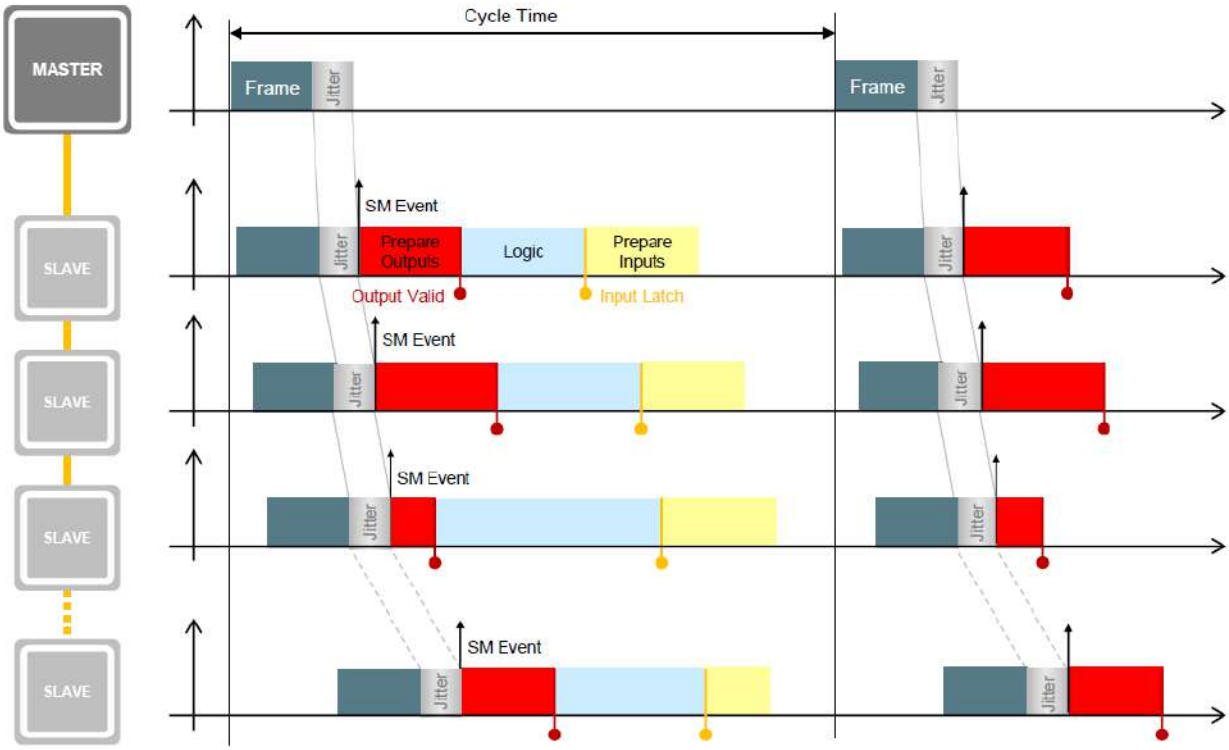
10.8.1. Free run

When the drive is controlled using I/O's, it is possible to connect EtherCAT for changing parameter's and monitoring. In this condition there is no need of synchronization with EtherCAT master and the data exchange can be asynchronous. Nevertheless, this type of functioning is not recommended, synchronization with sync manager (SM2) event is preferred. In order to operate in free run, objects 0x1C32:1, SM2 synchronization type and 0x1C33:1, SM3 synchronization type shall be written equal to zero.

10.8.2. SM synchronous

Drive's process data handling is started when the frame on SM2 is received. This lead to inaccuracy, because:

- cyclic frames are received by the slaves with the same jitter, which affects the master in sending them.
- even with no jitter, due to finite hardware propagation delays the last slaves of the ring topology will receive the cyclic frames later with respect to the first ones



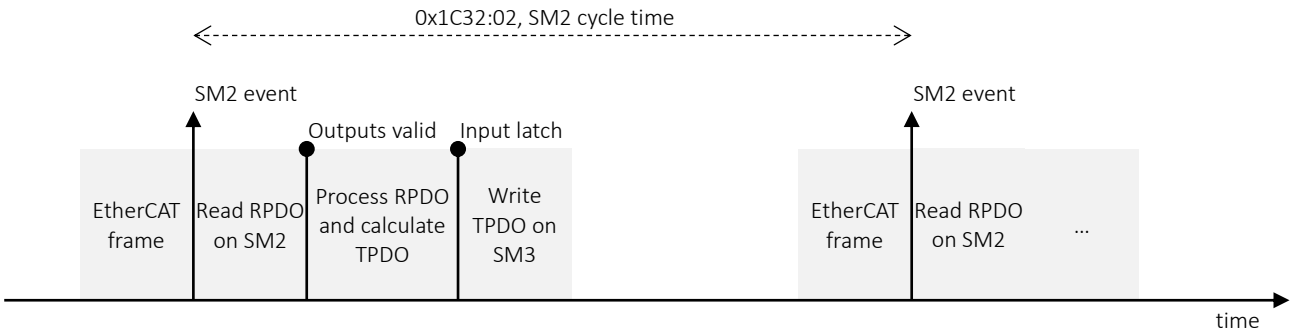
FD drives can withstand jitter on SM event, provided that it is less than a quarter of SM2 cycle time (time period between two frames). This is achieved thanks to internal PLL, capable to create a new time base that is used in cyclic synchronous modes (CSP and CSV). Because modern CPUs have a jitter of approx. 5 µsec, this mode of functioning is acceptable in most of the cases.

Nevertheless, the propagation delay, that each node of the ring introduces, introduces a frame time shift, which, with a high number of slaves, can cause a not well synchronized network. To give an idea, for simplicity the total slave forwarding delay and the cable propagation delay associated with a single slave can be considered of approx. 1 µsec.

Considerations on the need to use more complex synchronization methods depends on the number of slaves and application requirements.

On the slave point of view "prepare outputs" is the reading of SM2, once RPDO data is valid, it is immediately processed, (e.g. on cyclic synchronous velocity mode (CSV) 0x60FF, target velocity is set on drive frequency generator). Feedbacks

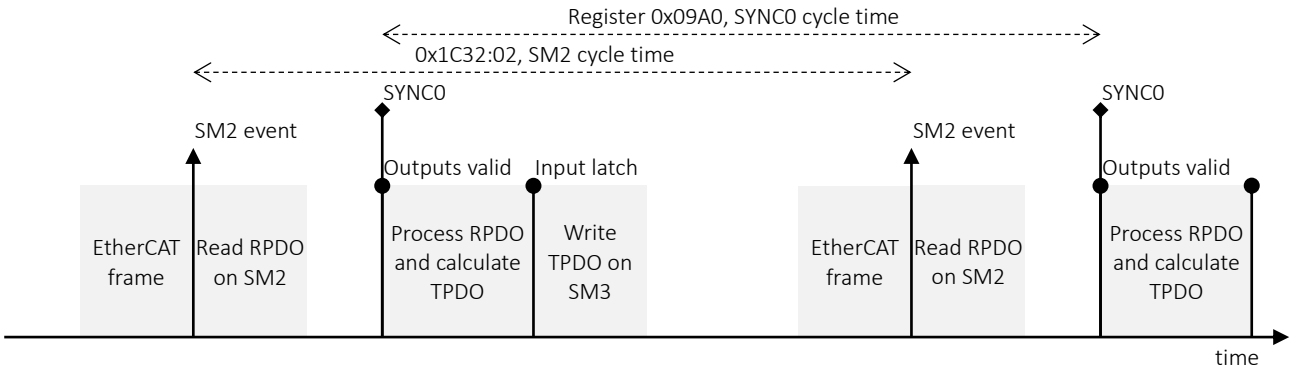
are updated and latched, then written on SM3.



The cycle time, which is the time period between every frame can be explicit writing object 0x1C32:02, SM2 cycle time, but anyhow the internal PLL is able to calculate autonomously the correct period of SM2.
Reading the object 0x1C32:02, SM2 cycle time allows to monitor the calculated PLL period.
Object 0x1C32:05, SM2 minimum cycle time express the minimum period in nsec, that for FD1E and FD2E is 1 msec. This value might depend upon the quantity of Bytes that are mapped into the PDO's.

10.8.3. Distributed clocks (DC)

A mechanism named distributed clock (DC) is used to synchronize EtherCAT communications. The DC mode is used to perform highly accurate control in a multi-axis system.
In DC mode, the master and slaves are synchronized by sharing the same clock. Interrupts (SYNC0 and SYNC1) are generated in the slaves at precise intervals based on this clock.
The communications cycle is determined by setting the SYNC0 signal output cycle. The jitter can be reduced to a few nanoseconds.



DC synchronization is more complicated than SM synchronization, but it offers better performances. The SM2 event is affected by jitter of EtherCAT frames (~µsec), while SYNC0's jitter is very small (~nsec). Furthermore, with DC synchronization all the slaves share a common time base using distributed PLL, that produces the same network SYNC signal to each slave, while on the contrary EtherCAT frames suffers of propagation delay.
It is important that the master transmits the EtherCAT frame with sufficient time before SYNC0 occurs, to allow the slave to read it, otherwise the drive application would not have RPDO data to process when the SYNC arrives.
DC synchronization is activated from the master writing on ESC register
0x1C32:1, SM2 synchronization type is equal to 2, i.e. synchronous with DC SYNC0.
0x1C32:2, SM2 cycle time is taken from master's settings to 0x09A0, SYNC0 cycle time ESC register

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.9. Device Control

The state machine, illustrated in Fig. 18, describes the device status and the possible control sequences of the drive. States can be changed using the control word and/or internal events. The current state can be read using the status word.

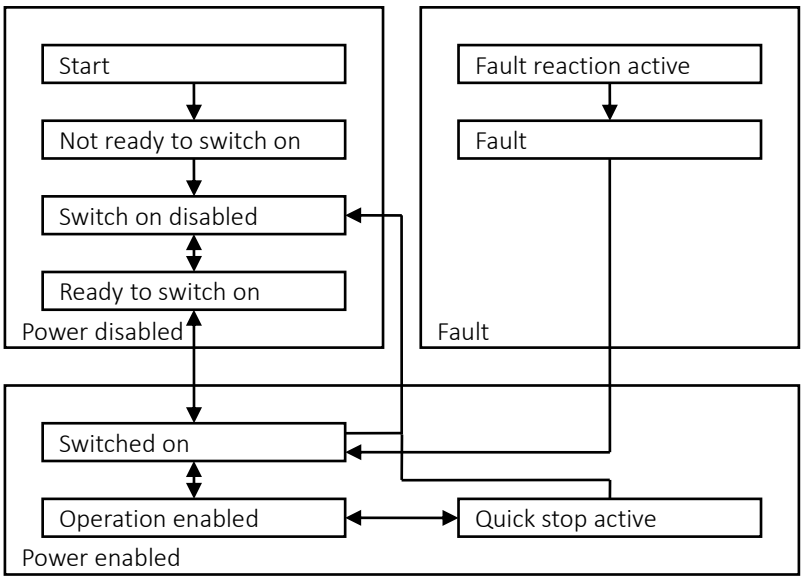


Fig. 18 – State machine

- Switch on disabled:
  - *shut down* command sets the state to *ready to switch on*. Current and frequency will be disabled.
- Ready to switch on:
  - *quick stop* or *disable voltage* commands set the state to *switch on disabled*. Current and frequency will be disabled.
  - *switch on* command sets the state to *switched on*. Current will be enabled, frequency disabled.
- Switched on:
  - *shut down* command sets the state in *ready to switch on.* Current and frequency will be disabled.
  - *quick stop* or *disable voltage* commands set the state to *switch on disabled*. Current and frequency will be disabled.
  - *enable operation* command sets the state in *operation enabled*. Current and frequency will be enabled.
- Operation enabled:
  - *shut down* command sets the state in *ready to switch on*. Current and frequency will be disabled.
  - *disable voltage* command sets the state to *switch on disabled*. Current and frequency will be disabled.
  - *disable operation* command sets the state to in *switched on.* Current will be enabled, frequency disabled.
  - *quick stop* command sets the state to *quick stop active*. Current and frequency will be enabled.
- Quickstop active:
  - *enable operation* command sets the state to *operation enabled.* Current and frequency will be enabled
  - *disable voltage* command sets the state to *switch on disabled.* Current and frequency will be disabled.
- Fault:
  - Fault reset command sets the state to switched on. Current will be enabled and frequency disabled.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Commands are activated through writes in control word bits as per Tab. 63.

Commands	Control word bits				
	7	3	2	1	0
	Fault reset	Enable operation	Quick stop	Enable voltage	Switch on
Shut down	0	X	1	1	0
Switch on	0	X	1	1	1
Disable voltage	0	X	X	0	X
Quick stop	0	X	0	1	X
Disable operation	0	0	1	1	1
Enable operation	0	1	1	1	1
Fault reset	Positive edge	X	X	X	X

Tab. 64 – Control word commands

Note:
Bits marked X are irrelevant.

Bits 12, 11, 8, 6 and 4 have effect only in operation enabled status and they assume a meaning that depends upon operation mode selected, as per Tab. 65

Operation modes	Control word bits				
	12	11	8	6	4
Profile position mode	Sub-mode	Sub-mode	Halt	Sub-mode	New set point Start move
Profile velocity mode	-	-	Halt	-	-
Homing mode	-	-	Halt	-	Homing start
Position address mode	-	-	Halt	-	-
Interpolated position mode	-	-	-	-	Activate IP mode
Cyclic synchronous position mode	-	-	-	-	-
Cyclic synchronous velocity mode	-	-	-	-	-

Tab. 65 – Control word commands

The status word bits return the status of the drive.

Bit number	Status
0	Ready to switch on
1	Switched on
2	Operation enabled
3	Fault
4	Voltage enabled
5	Quick stop
6	Switch on disabled
7-8	-
9	Remote (always at 1)
10	Target reached
11	SW Limit switches active
12	Profile position mode: set point acknowledge, Profile velocity mode: speed,

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

	Homing mode: homing attained, IP mode: interpolated position mode active, Position address mode: last position valid
13	Step loss alarm in profile position, velocity and position address Homing error in homing mode
14	Position uploaded

Tab. 66 – Status word bits

Values (binary)	States
xxxx xxxx x0xx 0000	Not ready to switch on
xxxx xxxx x1xx 0000	Switch on disabled
xxxx xxxx x01x 0001	Ready to switch on
xxxx xxxx x01x 0011	Switched on
xxxx xxxx x01x 0111	Operation enabled
xxxx xxxx x00x 0111	Quick stop active
xxxx xxxx x00x 1111	Fault reaction
xxxx xxxx x0xx 1000	Fault

Tab. 67 – Status

In all the modes the *target reached* bit is set when the deceleration due to Halt command has reached zero speed. In profile position it is also set when the absolute or relative indexer movement have finished.

10.9.1. Modes of operation

The control device writes the object *modes of operation* (0x6060) in order to select the operation mode. The drive provides the *modes of operation display* (0x6061) object to indicate the actual activated operation mode. Control word, status word, and set-points are used mode-specific.

Values	Mode
0x00	Undefined
0x01	Profile position mode
0x03	Profile velocity mode
0x06	Homing mode
0x07	Interpolated position mode
0x08	Cyclic synchronous position mode
0x09	Cyclic synchronous velocity mode

Tab. 68 – Operating modes

After reset or power on *modes of operation* is set equal to 0x00.

10.9.1. Profile position (1)

In this mode the drive internally generates a position trajectory based on input parameters. The type of trajectory can be absolute indexer, relative indexer, delta stop, position delay or time delay cycles. The selection of the type is made by the three bits 12, 11 and 6 *sub-mode* of *control word*:

Control word bits			Movement
12	11	6	
0	0	0	Absolute indexer
0	0	1	Relative indexer
0	1	0	Delta stop
0	1	1	Position delay
1	0	0	Time delay

Tab. 69 – Sub-modes

From *switched on* status, the master shall set modes of operation to 1, and then write control word 0x0F to enable

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

operation. In order to start the motor movement control word bit 4 and status word bit 12 shall make a hand-shake as per below time graph.

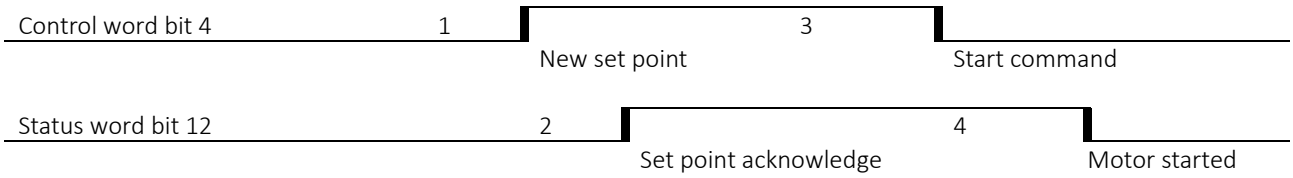


Fig. 19 – Profile position mode

1. The positive edge of control word samples the movement data according to the sub-mode selected:
   - *control word* (0x6040) sub-mode bits,
   - profile velocity (0x6081),
   - profile acceleration (0x6083),
   - profile deceleration (0x6084),
   - target position (0x607A),
   - delta stop steps (0x2003).

2. FD confirms the data sampling rising the status word bit 12, set point acknowledge.

3. In this condition the negative edge of control word bit 12 starts the movement.

4. FD signals the motor movement and readiness of sampling new setpoints with a falling edge of status word bit 12.

It is possible to create a FIFO buffer of movements giving a *new set point* while the current movement is running or while the Halt bit is active. As soon as the current movement is finished or the Halt is released the cycle starts.
All the movements will be executed in stream, one after the other, till the FIFO is not empty.
FIFO is circular, while the motor is moving it is possible to insert new movements.

Setting the Halt bit during a movement, the motor stops using profile deceleration and resets the FIFO (same effect as acting on quick stop control word bits).

The buffer is 31 cycles long, if it is exceeded the *set point acknowledge* bit will not be released to zero until the cycle in executions is not completed.

It is recommended not to use the command disable operation while the motor is running, as this command instantly disable the frequency, without deceleration ramp.

Recommended PDO mapping:

PDO#	B0	B1	B2	B3	B4	B5	B6	B7	Mapping
RPDO1	Control word								1600:0 = 0 1600:1 = 60400010 1600:0 = 1
RPDO2	Profile velocity				Target position				1601:0 = 0 1601:1 = 60810020 1601:2 = 607A0020 1601:0 = 2
RPDO3	Profile acceleration				Profile deceleration				1602:0 = 0 1602:1 = 60830020 1602:2 = 60840020 1602:0 = 2
TPDO1	Status word								1A00:0 = 0 1A00:1 = 60410010 1A00:0 = 1
TPDO2	Velocity actual value				Position actual value				1A01:0 = 0

			1A01:1 = 606C0020 1A01:2 = 60640020 1A01:0 = 2

SyncManagers PDO assignment:

SyncManagers	PDOs				Assignment
SM2	RPDO1	RPDO2	RPDO3		1C12:0 = 0 1C12:1 = 1600 1C12:2 = 1601 1C12:3 = 1602 1C12:0 = 3
SM3	TPDO1	TPDO2			1C13:0 = 0 1C13:1 = 1A00 1C13:2 = 1A01 1C13:0 = 2

10.9.2. Profile velocity (3)

In this mode the drive generates a velocity trajectory. When entering to profile velocity mode the following movement data are sampled:

- profile acceleration (0x6083),
- profile deceleration (0x6084),
- target velocity (0x60FF).

From *switched on* status, the master shall set modes of operation to 3, and then write control word 0x0F to enable operation. As soon as the operation is enabled, the drive will accelerate the motor to target velocity.

Writing on *target velocity* object it is possible to vary the motor speed and direction (this command performs also a new sampling of *profile acceleration* and *profile deceleration* objects). When the sign of *target velocity* is inverted the motor will decelerate to zero speed and it will accelerate to the opposite direction. Positive values correspond to clockwise movements, increasing the position counter value, while negative values correspond to counter-clockwise movements, decreasing the position counter value.

The movement stops setting the control word halt bit or writing target velocity 0.

Once the speed set-point is reached, status word bit 12 *speed* is set.

Velocity actual value is obtained through differentiation from the position encoder.

Recommended PDO mapping:

PDO#	B0	B1	B2	B3	B4	B5	B6	B7	Mapping
RPDO1	Control word		Target velocity						1600:0 = 0 1600:1 = 60400010 1600:2 = 60FF0020 1600:0 = 2
RPDO2	Profile acceleration				Profile deceleration				1601:0 = 0 1601:1 = 60830020 1601:2 = 60840020 1601:0 = 2
TPDO1	Status word								1A00:0 = 0 1A00:1 = 60410010 1A00:0 = 1
TPDO2	Velocity actual value				Position actual value				1A01:0 = 0 1A01:1 = 606C0020 1A01:2 = 60640020 1A01:0 = 2

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

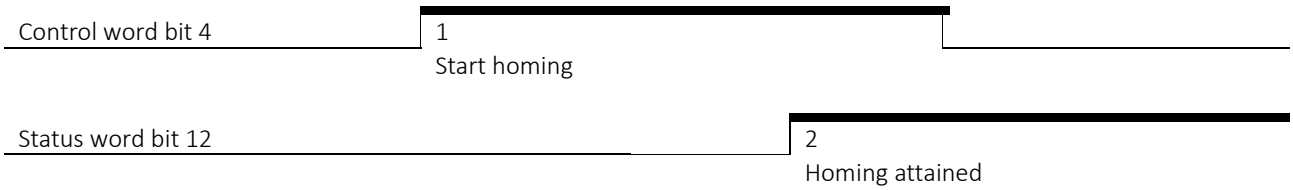
SyncManagers PDO assignment:

SyncManagers	PDOs				Assignment
SM2	RPDO1	RPDO2			1C12:0 = 0 1C12:1 = 1600 1C12:2 = 1601 1C12:0 = 2
SM3	TPDO1	TPDO2			1C13:0 = 0 1C13:1 = 1A00 1C13:2 = 1A01 1C13:0 = 2

10.9.3. Homing (6)

This mode is used to seek the home position. It is possible to specify the speeds, acceleration and the method of homing. A further object, *home offset*, allows to associate a specific value for the home position.
Two *homing speeds* are available: in a typical cycle the faster speed is used to find the home switch and the slower speed is used to release the home switch or to perform the following marker cycle.

From *switched on* status, the master shall set modes of operation to 6, write control word 0x0F to enable operation. Following signals are used:



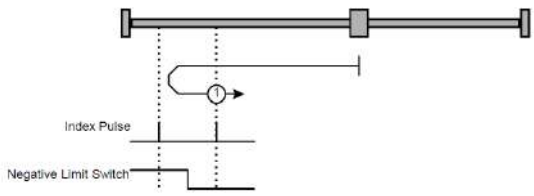
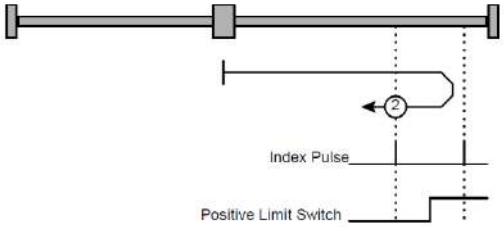
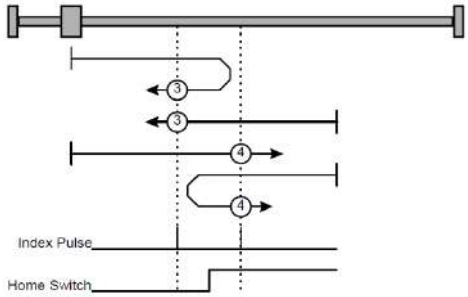
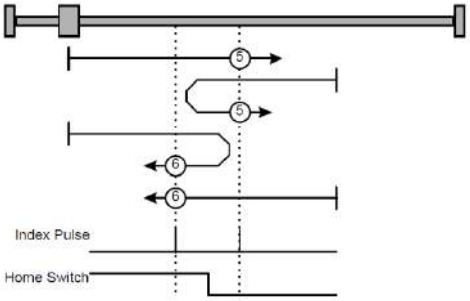
1. The homing movement starts upon a high level of *control word* bit 4, *start homing*. Upon edge detection FD sets also acceleration and deceleration equal to *homing acceleration*. The movement stops in case of low level of the same bit or in case of the *halt* bit.
2. Once the drive completes the homing procedure, status word bit 12, *homing attained*, and bit 10, *target reached* are raised.
3. In case of error during homing status word bit 13, *homing error* is raised.

The *homing methods* available are presented in Tab. 70. The method name is composed by up to four abbreviations:

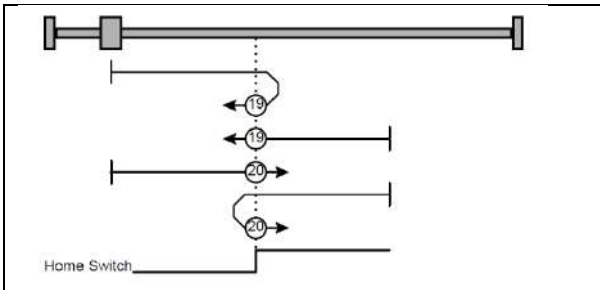
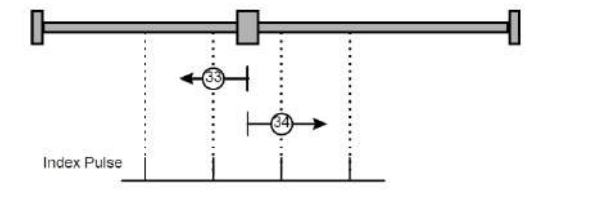
- UP and DW define the motor direction (UP is CW towards increasing positions, DW is CCW towards decreasing positions when no inversion is applied)
- LS and HS define if the calibration is performed on a homing switch or limit switch. Multipurpose inputs need to be configured accordingly.
- MK is the marker cycle, i.e. a motor rotation to the absolute encoder position destination.
- The number 1 or 2 determines the directions of the homing input edge and/or the movement.

Homing methods	Values	Description
DW_LS_MK Homing on negative limit switch and index pulse	1	Motor will rotate at *homing speed research* CCW towards decreasing positions till a positive edge of hardware limit switch down input signal is met (at least one multipurpose input needs to be configured as limit switch down, i.e. input configuration register equals to 7 or 8).

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

		It will decelerate to zero speed, wait *time stop* milliseconds and accelerate in opposite direction at *homing speed release* till the negative edge of the same signal is met and then it continues moving at the same direction and speed till the marker cycle is completed. *Note:* *It is important the switch does not commutate in the same position of the zero encoder, otherwise there can be uncertainty of one revolution.*
UP_LS_MK Homing on positive limit switch and index pulse	2	It is similar to DW_LS_MK, but if the switch is off, motor will start CW towards increasing position till a positive edge of hardware limit switch up input signal is met (at least one input need to be configured as hardware limit switch up, i.e. multipurpose input configuration register equals to 5 or 6). It will decelerate to zero speed, wait *time stop* milliseconds and accelerate in opposite direction at *homing speed release* till the negative edge of the same signal is met and then it continues moving at the same direction and speed till the marker cycle is completed. *Note:* *It is important the switch does not commutate in the same position of the zero encoder, otherwise there can be uncertainty of one revolution.*
		
UP_HS_MK_1 Homing on positive home switch and index pulse	3	Motor will rotate at *homing speed research* CW towards increasing positions till a positive edge of homing switch input signal is met (at least one input needs to be configured as homing switch, i.e. input configuration register equals to 2 or 3). It decelerates to zero speed, waits Time stop milliseconds and accelerates in opposite direction at *homing speed release* till the negative edge of the same signal is met and then it continues moving at the same direction and speed till the marker cycle is completed.
		
UP_HS_MK_2 Homing on positive home switch and index pulse	4	Same as UP_HS_MK_1, but towards decreasing positions and till a negative edge of homing switch input signal.
DW_HS_MK_1	5	Same as UP_HS_MK_1, but towards decreasing positions. Till a positive edge of homing switch input signal.
		
DW_HS_MK_2	6	Same as UP_HS_MK_1 towards increasing positions, but till a negative edge of IN3 or IN5.
-	-	
DW_LS	17	Same as DW_LS_MK, but without the marker cycle.
UP_LS	18	Same as UP_LS_MK, but without the marker cycle.
UP_HS_1	19	Same as UP_HS_MK_1, but without the marker cycle.
UP_HS_2	20	Same as UP_HS_MK_2, but without the marker cycle.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

		
DW_HS_1	21	Same as DW_HS_MK_1, but without the marker cycle.
DW_HS_2	22	Same as DW_HS_MK_2, but without the marker cycle.
-	-	
DW_MK	33	Marker cycle CCW towards decreasing positions
UP_MK	34	Marker cycle CW towards increasing positions
		
CURR_POSITION	35	It sets current position equal to *home offset*.

Tab. 70 – Homing methods

Notes:
The homing attained bit is reset only when the homing movement starts. The information is kept passing to another mode of operation. This means that it is possible to switch back to homing mode and find the homing attained active prior starting the homing movement. As soon as the homing movement is started, homing attained is reset. Homing attained has the same meaning of Modbus status word bit axle calibrated, which can be restored from power on, ref. to 11.

10.9.4. Interpolated position (7)

The interpolated position mode is used to control multiple coordinated axles or a single axle with the need for time-synchronization of positions. The interpolated position mode uses the SYNC object as time synchronization mechanisms for a time coordination of the related drive units.

The master shall implement a path generation function, to provide the axle positions at every sync instant.

The *interpolation data record* contains the interpolated position set-points, expressed as absolute multi-turn position. This object has the dimensions of a record, because it would eventually be possible to specify a vector, e.g. position, speed and acceleration that the motor shall assume at each sync. FD1 requires just positions, that's why this object is a record with only two registers: number of entries and position. The record size is fixed and defined in the *size of data record* as sub-index of the *interpolation data configuration*.

The interpolated position mode allows a host controller to transmit a stream of interpolation data with an explicit time reference to a drive unit. The period between every sync objects is pre-defined by the object *interpolation time period*.

FD drives support an input buffer of 32 positions, the interpolation data may be sent in bursts rather than continuously in real time. The available and the maximum size of the input buffer can be requested by a host using the *interpolation data configuration*. The buffer size is the number of interpolation data records which may be sent to a drive to fill the input buffer and it is not the size in Bytes.

The linear interpolation algorithm is defined in the interpolation sub mode select. This requires only one interpolation data item at the time, i.e. the position. During the interpolation cycle, the drive calculates the intermediate position demand values over the period of time.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

There is no limit function for speed, acceleration and deceleration applied to the interpolation data. If the drive cannot execute the ordered command step accumulation limit alarm arises.

To activate the interpolated position mode, modes of operation shall be set to 7, operation shall be enabled and control word bit 4 shall be high. Slave will answer rising status word bit 12.

Control word bit 4	1 Activate interpolation
Status word bit 12	2 Interpolation active

When the drive status is operation enable and the interpolated position mode is selected, the drive enters in interpolation inactive. The drive unit will accept input positions and will buffer it for interpolation calculations, but it does not move the motor. Interpolation active state is entered when the device is in operation enable state, the interpolated position mode is selected and activated with control word bit 4. The drive unit will accept input data and it will move the motor unwinding the buffered positions at every SYNC.

Entering interpolated position mode or operation enabled clears the buffer.

Writing the interpolation position (0x60C1:1), data is loaded into the input buffer, organized as a FIFO, and the pointer of the buffer is incremented to the next buffer position.

Writing 0 to sub-index *buffer clear* of *interpolation data configuration* clears all positions buffered.

Once the interpolation is activated and sync is transmitted, interpolation position is immediately consumed, without delay, i.e. the drive will move the motor to be at that position at next sync signal. An input buffer for interpolation data records is not mandatory, although it eases the data exchange between a host and a drive unit when transmission of data has jitter in respect to sync signal. The real-time requirements of the EtherCAT network decrease in this case, because the input buffer decouples the data processing in the drive from the data transmission via the bus line.

In order to follow a two- or more-dimensional curve through the space with a defined speed, a host (an interpolation controller or a PLC) calculates the positions for each set of coordinates which have to be reached at sync instants and transmits them to the axles.

Interpolated position mode can be operated via SyncManager synchronization or via DC synchronization.

Recommended PDO mapping:

PDO#	B0	B1	B2	B3	B4	B5	B6	B7	Mapping
RPDO1	Control word		Interpolated data record, sub-ix 1						1600:0 = 0 1600:1 = 60400010 1600:2 = 60C10120 1600:0 = 2
TPDO1	Status word		Position actual value						1A00:0 = 0 1A00:1 = 60410010 1A00:2 = 60640020 1A00:0 = 2

SyncManagers PDO assignment:

SyncManagers	PDOs				Assignment
SM2	RPDO1				1C12:0 = 0 1C12:1 = 1600 1C12:0 = 1
SM3	TPDO1				1C13:0 = 0

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

					1C13:1 = 1A00 1C13:0 = 1

Note:
Before entering IP mode, make sure that the first position that will be transmitted is the same or in proximity of 0x6062, position demand value (valid only if the drive is switched on).

10.9.1. Cyclic synchronous position (8)

This mode of operation is very similar to interpolated position mode, with some differences. In order control the motor using cyclic synchronization, the master shall still generate a position trajectory. This means that cyclically the master shall calculate for the drive its position at every synchronization period. On the other hand, the drive interpolates between each received target position with a resolution of 50 usec and it controls the torque based by the encoder speed and position.

Recommended PDO mapping:

PDO#	B0	B1	B2	B3	B4	B5	B6	B7	Mapping
RPDO1	Control word		Target position						1600:0 = 0 1600:1 = 60400010 1600:2 = 607A0020 1600:0 = 2
TPDO1	Status word		Position actual value						1A00:0 = 0 1A00:1 = 60410010 1A00:2 = 60640020 1A00:0 = 2

SyncManagers PDO assignment:

SyncManagers	PDOs				Assignment
SM2	RPDO1				1C12:0 = 0 1C12:1 = 1600 1C12:0 = 1
SM3	TPDO1				1C13:0 = 0 1C13:1 = 1A00 1C13:0 = 1

The target position is an absolute value. Feedbacks, such as position actual value and velocity actual value can be used to monitor motor movement, there is no need to close the position loop on master side. The drive itself calculate autonomously the correct amount of speed and torque needed to follow the target position setpoint.

CSP can be operated with SyncManager synchronization or with DC synchronization.

Note:
Compared to interpolated position mode, there is no buffer of setpoints, sync period is not explicit, instead of using a record, position is transmitted directly to an integer object target position.

10.9.2. Cyclic synchronous velocity (9)

In this mode of operation, master shall generate a velocity trajectory, hence provide the target velocity to the drive using cyclic synchronization. This means that cyclically the master shall calculate for the drive its velocity at every synchronization period.

Compared to profile velocity mode, where the drive receives just the speed regime setpoint and autonomously accelerate/decelerate using profile acceleration/deceleration parameters, in cyclic synchronous velocity mode the master shall provide the instantaneous velocity during acceleration and deceleration at every synchronization period.

Recommended PDO mapping:

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

PDO#	B0	B1	B2	B3	B4	B5	B6	B7	Mapping
RPDO1	Control word		Target velocity						1600:0 = 0 1600:1 = 60400010 1600:2 = 60FF0020 1600:0 = 2
TPDO1	Status word		Velocity actual value						1A00:0 = 0 1A00:1 = 60410010 1A00:2 = 606C0020 1A00:0 = 2

SyncManagers PDO assignment:

SyncManagers	PDOs				Assignment
SM2	RPDO1				1C12:0 = 0 1C12:1 = 1600 1C12:0 = 1
SM3	TPDO1				1C13:0 = 0 1C13:1 = 1A00 1C13:0 = 1

Feedbacks, such as position actual value and velocity actual value shall be used to monitor motor movement, but there is no need to close the velocity, torque loops on master side. The drive itself calculate autonomously the correct amount of speed and torque needed to follow the target velocity setpoint.

CSV can be operated with SyncManager synchronization or with DC synchronization.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.10. Object dictionary

Auxind FD1E and FD2E drives with EtherCAT Communications, use the object dictionary for CAN application protocol over EtherCAT (CoE) as a base for communications.

All objects are assigned four-digit hexadecimal numbers in the areas shown in the following table.

Indexes	Area	Description
0x0000 – 0x0FFF	Data types	Data types definitions
0x1000 – 0x1FFF	CoE communication	Definitions of variables that can be used by all servers for designated communications
0x2000 – 0x2FFF	Manufacturer specific	Variables with common definitions for all Auxind products
0x6000 – 0x9FFF	Device profile specific	CiA402 servo drive profile

10.10.1. CoE communication

Index	Sub-index	bit	Type	Access	Object name	Description
0x1000	-	-	U32	RO	Device type	0x40192 = Stepper
0x1001	-	0	U8	RO	Error register	General alarm
		1				Current
		2				Voltage
		3				Temperature
		5				Drive error (device profile specific)
		7				Position error (manufacturer specific)
0x1003	0		ARR	RW	Pre-defined error	Number of errors
	1 – 8					Error codes
0x1008	-	-	STR	RO	Manufacturer device name	“Auxind – FD2.1E” or “Auxind – FD1.1E”
0x1009	-	-	STR	RO	Manufacturer hardware version	“2.1E” or “1.1E”
0x100A	-	-	STR	RO	Software version	“V6.02”
0x1010	-	-	ARR	RW	Store parameters	
	0	-	U8	RO	Number of entries	1
	1	-	U32	RW	Save all parameters	R: 1, Device saves parameters on command W: 0x65766173, signature to save all parameters
0x1011	-	-	ARR	RW	Restore default parameters	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Restore all default parameters	R: 1, Device restores parameters W: 0x64616F6C, signature to restore all parameters
0x1018	-	-	REC		Identity object	
	0	-	U8	RO	Number of entries	1
	1	-	U32	RO	Vendor ID	0x000004D8
	2	-	U32	RO	Product code	0x000004D8
	3	-	U32	RO	Revision number	0
	4	-	U32	RO	Serial number	0
0x1600	-	-	REC	-	Pdo_1_Rx_Mapping_record	
	0	-	U8	RW	Number of entries	1
	1...7	-	U32	RW	Pdo_1_Rx_mapping_1...7	0x60400010
0x1601	-	-	REC	-	Pdo_2_Rx_Mapping_record	
	0	-	U8	RW	Number of entries	2
	1...7	-	U32	RW	Pdo_2_Rx_mapping_1...7	0x60400010; 0x60600008
0x1602	-	-	REC	-	Pdo_3_Rx_Mapping_record	
	0	-	U8	RW	Number of entries	2
	1...7	-	U32	RW	Pdo_3_Rx_mapping_1...7	0x60400010; 0x607A0020
0x1603	-	-	REC	-	Pdo_4_Rx_Mapping_record	
	0	-	U8	RW	Number of entries	2
	1...7	-	U32	RW	Pdo_4_Rx_mapping_1...7	0x60400010; 0x60FF0020
0x1A00	-	-	REC	-	Pdo_1_Tx_Mapping_record	
	0	-	U8	RW	Number of entries	1
	1...7	-	U32	RW	Pdo_1_Tx_mapping_1...7	0x60410010

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Index	Sub-index	bit	Type	Access	Object name	Description
0x1A01	-	-	REC	-	Pdo_2_Tx_Mapping_record	
	0	-	U8	RW	Number of entries	2
	1...7	-	U32	RW	Pdo_2_Tx_mapping_1...7	0x60410010; 0x60610008
0x1A02	-	-	REC	-	Pdo_3_Tx_Mapping_record	
	0	-	U8	RW	Number of entries	2
	1...7	-	U32	RW	Pdo_3_Tx_mapping_1...7	0x60410010; 0x60630020
0x1A03	-	-	REC	-	Pdo_4_Tx_Mapping_record	
	0	-	U8	RW	Number of entries	2
	1...7	-	U32	RW	Pdo_4_Tx_mapping_1...7	0x60410010; 0x606C0020
0x1C00	-	-	REC	-	SyncManagers comm. type	
	0	-	U8	RO	Number of entries	4
	1	-	U8	RO	SM0 communication type	1: Mailbox Out
	2	-	U8	RO	SM1 communication type	2: Mailbox In
	3	-	U8	RO	SM2 communication type	3: Buffer Out
	4	-	U8	RO	SM3 communication type	4: Buffer In
0x1C12	-	-	REC	-	SM2 PDO assignment	
	0	-	U8	RW	Number of entries	2
	1...4	-	U16	RW	SM2 PDO assignment_1...4	0x1600; ...
0x1C13	-	-	REC	-	SM3 PDO assignment	
	0	-	U8	RW	Number of entries	2
	1...4	-	U16	RW	SM3 PDO assignment_1...4	0x1A00; ...
0x1C32	-	-	REC	-	Sync manager 2 parameter	
	0	-	U8	RO	Number of entries	
	1		U16	RW	SM2 synchronization type	0x00: free run 0x01: synchronous with SM 0x02: DC SYNC0 0x03: DC SYNC1
	2		U32	RW	SM2 cycle time	Expressed in nsec. Free run: not used Synchronous with SM2: time between two SM2 events DC SYNC0 and DC SYNC1: time between two SYNC0.
	4	0	U16	RO	SM2 sync types supported	1: Free run supported
		1				1: Synchronous with SM2 supported
		2				1: Synchronous with DC SYNC0 supported
		3				1: Synchronous with DC SYNC1 supported
		5				Shift settings
		6				0: no output shift supported
	5	-	U32	RO	SM2 minimum cycle time	1'000'000 nsec
0x1C33	-	-	REC	-	Sync manager 3 parameter	
	0	-	U8	RO	Number of entries	
	1		U16	RW	SM3 synchronization type	0x00: free run 0x01: synchronous with SM 0x02: DC SYNC0 0x03: DC SYNC1
	2		U32	RW	SM3 cycle time	Expressed in nsec. Free run: not used Synchronous with SM2: time between two SM2 events DC SYNC0 and DC SYNC1: time between two SYNC0.
	4	0	U16	RO	SM3 sync types supported	1: Free run supported
		1				1: Synchronous with SM supported
		2				1: Synchronous with DC SYNC0 supported
		3				1: Synchronous with DC SYNC1 supported
		5				Shift settings
		6				0: no output shift supported
	5	-	U32	RO	SM3 minimum cycle time	1'000'000 nsec

10.10.2. 0x1000, Device type

Read only unsigned 32, it is composed of lower 16-bit field which describes the device profile that is used (DSP 402) and

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

higher 16 bits which describes the device type, which for stepper motor is equal to 0x4. Hence 0x40192.

10.10.3. 0x1001, Error register

FD drives map internal errors in this byte. It is part of an Emergency object. Only the fault reset command is able to reset this register.

Bit number	Description
0	Generic error
1	Current error
2	Voltage error
3	Temperature error
4	Communication error
5	Parameter error
6	Always 0
7	Step accumulation limit

Tab. 59 – Error register bits

10.10.4. 0x1003, Pre-defined error

The object at index 0x1003 holds the alarms that have occurred and have been signaled transmitting EMCY object, plus all the communication errors. In doing so, it provides an error history.

The entry at sub-index 0 contains the number of actual errors that are recorded in the array starting at sub-index 1. It is RW and ranges from 0 to 8. Writing a 0 to sub-index 0 deletes the entire error history (empties the array). Values higher than 0 are not allowed to write. This lead to an abort message (error code: 0x06090030).

Every new error is stored at sub-index 1, the older ones move down the list, made of maximum 8 elements.

The error numbers are of type U32. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB).

10.10.5. 0x1010, Store parameters

This object supports the saving of parameters in flash. By read access it provides information about its saving capabilities. 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 0x65766173.
On reception of the correct signature in sub-index 1, the drive stores the current parameters in flash, then confirms the SDO transmission (initiate download response). If the storing failed, the device responds with an Abort SDO Transfer (abort code: 0x06060000). If a wrong signature is written, the device refuses to store and responds with Abort SDO Transfer (abort code: 0x0800002x).

10.10.6. 0x1011, Restore default parameters

With this object all the default values of parameters are restored. By read access the device provides information about its capabilities to restore these values.
In order to avoid the restoring of default parameters by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. The signature is 0x 64 61 6F 6C.

On reception of the correct signature in the appropriate sub-index the device restores the default parameters and then confirms the SDO transmission (initiate download response). If the restoring failed, the device responds with an Abort SDO Transfer (abort code: 0x06060000). If a wrong signature is written, the device refuses to restore the defaults and responds with an Abort SDO Transfer (abort code: 0x0800002x).

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.10.7. 0x160x / 0x1A0x, RPDOx / TPDOx mapping records

The objects at indexes 0x1600, 0x1601, 0x1602, 0x1603 define the mapping of RPDO 1, 2, 3 and 4.
The objects at indexes 0x1A00, 0x1A01, 0x1A02, 0x1A03 define the mapping of TPDO 1, 2, 3 and 4.

To create the PDO object mapping, first the PDO has to be deleted, the sub-index 0, number of object mapped, type U8, must be set to 0 (PDO mapping is deactivated).
Then the sub-indexes 1-8, type U32, can be remapped writing object index, sub-index and number of bits of the object to be mapped into the PDO (number of bits can be only a multiple of 8, i.e. only Bytes can be mapped).
After all objects are mapped sub index 0 shall be set to the valid number of mapped objects.

Byte 3	Byte 2	Byte 1	Byte 0
Index_H	Index_L	Sub-index	Number of bits

Tab. 62 – PDO mapping records

Ref. to default mapping for having an example of the format.

If the change of the PDO mapping cannot be executed (e.g. the PDO length is exceeded or the SDO client attempts to map an object that cannot be mapped) FD responds with an Abort SDO Transfer Service.

PDO mapping can be performed only in pre-operational state.

10.10.8. 0x1C12, 0x1C13, SM2 and SM3 PDO assignment records

After mapping the objects into the PDOs, the PDOs have to be assigned to the SyncManagers, more precisely RPDOs shall be assigned to SM2, TPDOs to SM3.
The assignment is made only for those PDOs that are used in buffered mode, SyncManagers 0 and 1 are used in mailbox mode and for this reason they don't have PDO assigned.
Since there are 4 RPDOs and 4 TPDOs, objects 0x1C12 and 0x1C13 have a maximum of 4 entries each.
In a similar way of PDO mapping, SM assignment shall be performed by first setting to zero the sub-index 0, writing the PDOs to sub-indexes, then finally writing to sub-index 0 the number of PDOs assigned.

10.10.9. 0x1C32, 0x1C33, SM2 and SM3 parameters

There are mainly three types of synchronizations: Free run (not synchronized), synchronization with SM2 and synchronization with DC.
By default, the drive works synchronized with SM2. In case of need, the master can configure the ESC registers 0x0981, SYNC activation and 0x09A0, SYNC0 cycle time to activate the DC mode in the passage between pre-operational and safe operational. Reading from the ESC these registers, the drive microcontroller is able to recognize the type and period of synchronization to be used.
The received configuration can also be monitored and configured by the master via SDOs through records 0x1C32, 0x1C33 for SM2 and SM3.
EtherCAT offers many different types of synchronization, only a part of them relevant for stepper motor application has been implemented:

- Synchronization type

0 = free run,
1 = SM2 synchronous
2 = DC SYNC0 synchronous
3 = DC SYNC1 synchronous

- Cycle time

Period of time between two events in nsec.

- Minimum cycle time

Minimum period of time between two events, i.e. 1˙000˙000 msec.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.10.10. Manufacturer specific

Index	Sub-index	bit	Type	Access	Object name	Description
0x2003	-	-	U32	RW	Delta stop steps	Initialized to Cycle 0 delta stop steps
0x2004	-	-	U8	RW	Modes of position address	0
0x2005	[0-255]	-	ARR	RW	Modbus registers array 1	Ref. to 14.3.10
0x2006	[0-...]	-	ARR	RW	Modbus registers array 2	
0x200C	-	-	ARR	-	Encapsulated SDO	
	0	-	U8	RO	Number of entries	2
	1	-	U64	RW	SDO Request	
	2	-	U64	RO	SDO Response	
0x2010			ARR		Encoder position latch	It is used to measure the position of a sensor activation/deactivation
	0	-	U8	RO	Number of entries	2
	1		S32	RW	Position at rising edge	
	2		S32	RW	Position at falling edge	

10.10.11. 0x2003, Delta stop steps

Delta stop steps are used on delta stop cycles. This type of movement allows very accurate and fast positioning based on the presence of a sensor, detected on interrupt, to avoid time consuming homing cycles.
For further information, refer to ch. 11.5.

10.10.12. 0x2005 / 0x2006, Modbus register arrays

All the Modbus registers of ch. 13.1 have been mapped into two arrays (they are two, since the maximum size of an array is 255 elements).
Hence object 0x2005, sub 1 contains the Modbus register START, 0x2005, sub 2 contains STOP and so on.
Sub-index 0 of both arrays 0x2005 and 0x2006 is the number of entries.

All the objects are 32 bits long, the same length of Modbus registers.

10.10.13. 0x200C, SDO encapsulated

SDO request and SDO response are two sub-indexes of the record SDO encapsulated. They are used to access the complete dictionary using PDO's, instead of normal SDO's. The main advantage is that PDOs are faster and the user does not have to implement SDOs to monitor not essential information such as motor current, voltage, temperature, etc.

Their structure is very similar to SDOs, they are made of 8 Bytes:

Byte	Description
0	Command (Download + toggle / Upload): 0x40: Upload request 0x40: Upload response 0x10: Download toggle 0x20: Download request 0x60: Download response 0x80: Abort
1	Index
2	
3	Sub-index
4 – 7	Data

E.g., in order to transmit the read command of 0x2006:05 Motor current, the content of the PDO shall be as follows:

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

B0	B1	B2	B3	B4	B5	B6	B7
0x40	0x06	0x20	0x05	0x00	0x00	0x00	0x00
Upload request	Index		Sub-index	Don't care			

The response of the slave for a current of 5'000 milli Ampere will be:

B0	B1	B2	B3	B4	B5	B6	B7
0x40	0x06	0x20	0x05	0x88	0x13	0x00	0x00
Upload response	Index		Sub-index	0x1388 = 5000			

E.g., in order to transmit the write command of 0x6083:00, profile acceleration = 2000 Kstep/sec^2, the content of the PDO shall be as follows:

B0	B1	B2	B3	B4	B5	B6	B7
0x30	0x83	0x60	0x00	0xD0	0x07	0x00	0x00
Download request + Toggle	Index		Sub-index	0x07D0 = 2000			

The slave's response is:

B0	B1	B2	B3	B4	B5	B6	B7
0x70	0x83	0x60	0x00	0x00	0x00	0x00	0x00
Download response + Toggle	Index		Sub-index				

With the first write the toggle verification is positive, whether 0 or 1.

In case of abort, the slave's response is:

B0	B1	B2	B3	B4	B5	B6	B7
0x80	0x83	0x60	0x00	0x00	0x00	0x00	0x00
Download response + Toggle	Index		Sub-index	Abort code.			

10.10.14. CiA402 servo drive profile

Index	Sub-index	bit	Type	Access	Object name	Description
0x603F	-	-	U16	RO	Error code	Refer to detailed description
0x6040	-	0	U16	RW	Control word	Switch on
		1				Enable Voltage
		2				Quick Stop
		3				Enable Operation
		4				Operation mode specific
		5				Operation mode specific
		6				Operation mode specific
		7				Fault Reset
		8				Halt
		9				Reserved
		10				Reserved
		11				Operation mode specific
		12				Operation mode specific
0x6041	-	0	U16	RO	Status word	Ready to switch on
		1				Switched on
		2				Operation Enable
		3				Fault

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

		4				Voltage Enable
		5				Quick Stop
		6				Switch on disabled
		8				Torque Limit
		9				Remote
		10				Target reached
		11				SW Limit Switches
		12				Operation mode specific
		13				Operation mode specific
		14				Position Upload
0x6060	-	-	S8	RW	Modes of operation	Refer to detailed description
0x6061	-	-	S8	RO	Modes of operation display	Same as modes of operation, but RO
0x6062	-	-	S32	RO	Position demand value	
0x6063	-	-	S32	RO	Position actual internal value	
0x6064	-	-	S32	RO	Position actual value	
0x6065	-	-	U32	RW	Following error window	
0x606B	-	-	S32	RO	Velocity demand value	
0x606C	-	-	S32	RO	Velocity actual value	
0x6073	-	-	U16	RW	Max current	
0x6078	-	-	U16	RO	Current actual value	
0x6079	-	-	U32	RO	DC link circuit voltage	
0x607A	-	-	S32	RW	Target position	
0x607C	-	-	S32	RW	Home offset	
0x607D			REC		Software position limit	
	0		U8	RO	Number of entries	
	1		S32	RW	Min position limit	
	2		S32	RW	Max position limit	
0x607F	-	-	U32	RW	Max profile velocity	Upper limited to 300'000 [step/sec]
0x6081	-	-	U32	RW	Profile velocity	Upper limited to max profile velocity [step/sec]
0x6083	-	-	U32	RW	Profile acceleration	[kstep/sec^2]
0x6084	-	-	U32	RW	Profile deceleration	[kstep/sec^2]
0x6086	-	-	S16	RW	Motion profile type	0: Linear, 1: Parabolic, 2: s-curve.
0x6098	-	-	S8	RW	Homing method	
0x6099	-	-	REC	-	Homing speeds	
	0	-	U8	RO	Number of entries	2
	1		U32	RW	Homing speed research	[step/sec]
	2		U32	RW	Homing speed release	[step/sec]
0x609A	-	-	U32	RW	Homing acceleration	It sets both acceleration and deceleration. Range [1 - 20'000] in [kstep/sec^2]
0x60C0	-	-	S16	RW	Interpolation sub mode select	0: Linear interpolation
0x60C1			REC		Interpolation data record	
		0	U8	RO	Number of entries	1
		1	S32	RW	Interpolated position	
0x60C2	-	-	REC		Interpolation time period	Expressed as: units x 10^{index}
		0	U8	RO	Number of entries	2
		1	U8	RW	Interpolation time units	1
		2	S8	RW	Interpolation time index	-3, i.e. milliseconds
0x60C4			REC		Interpolation data configuration	
		0	U8	RO	Number of entries	
		1	U32	RO	Maximum buffer size	32
		2	U32	RO	Actual buffer size	0
		3	U8	RO	Buffer organization	0
		4	U16	RO	Buffer position	0
		5	U8	RO	Size of data record	1
		6	U8	RW	Buffer clear	0
0x60F4			S32	RO	Following error actual value	

0x60FD	-	0	U32	RO	Digital inputs	Negative limit switch
	-	1				Positive limit switch
	-	2				Home switch
	-	16				IN1
	-	17				IN2
	-	18				IN3
	-	19				IN4
	-	20				IN5
	-	21				IN6
	-	22				IN7
	-	23				OUT1
	-	24				OUT2
	-	25				OUT3
0x60FE	-	-	REC		Digital outputs	
	0	-	U8		Number of entries	2
	1	16	U32	RW	OUT1 physical	
		17			OUT2 physical	
		18			OUT3 physical	
	2	16	U32	RW	OUT1 mask	
		17			OUT2 mask	
		18			OUT3 mask	
0x60FF	-	-	S32	RW	Target velocity	Used in profile velocity mode and cyclic synchronous velocity [step/sec].
0x6402	-	-	U16	RO	Motor type	9: micro-stepper motor
0x6403	-	-	STR	RO	Motor catalog number	
0x6404	-	-	STR	RO	Motor manufacturer	
0x6405	-	-	STR	RO	Motor catalogue address	
0x6502		0	U32	RO	Supported drive modes	Profile position = 1
		1				Velocity = 0
		2				Profile velocity = 1
		3				Profile torque = 0
		4				Reserved = 0
		5				Homing = 1
		6				Interpolated position = 1
		7				Cyclic synchronous position = 1
		8				Cyclic synchronous velocity = 1
0x6503	-	-	STR	RO	Drive catalogue number	
0x6505	-	-	STR	RO	Http drive catalog address	

Tab. 58 – Object dictionary

10.10.15. 0x603F, Error code

It is an unsigned 16 bits, read only objects, which contains the actual error of the device. It is zero if no error is present. Once a new error take place, an emergency message is transmitted and the code is logged into 0x1003, predefined error field.

Value	Description
0x0000	Drive ok
0x1000	Generic error
0x2300	Motor over-current
0x3210	DC link over-voltage
0x3220	DC link under-voltage
0x3230	Open motor cable
0x4310	Over-temperature
0x6320	Parameter error
0x5493	Torque off error
0x8611	Following error (step accumulation limit)

Tab. 59 – Error code values

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.10.16. 0x6040 / 0x6041, Control word and status word

0x6040, control word and 0x6041, status word are unsigned 16 bit objects to command motor movements and check the status. Ref. to 14.4 for details.

10.10.17. 0x6060 / 0x6061, Modes of operation

0x6060, Mode of operation is a signed 8 bit object read write register, which can be used to change from one mode to another. 0x6061, Mode of operation display is a signed 8 bit read only object, which shows the actual mode in use. Ref. to 0 for details.

10.10.18. 0x6062 / 0x6063 / 0x6064 / 0x6065, Positions

0x6062, position demand value, 0x6063, position actual internal value and 0x6064 position actual value are a signed 32 bits read only object. The first one is the ordered position, while second and third are the position feedback from the encoder. All of them are expressed in the same resolution (step/revolution).

0x6065, following error window is an unsigned 32 bits read write object, which can be configured to set the maximum deviation between the ordered steps and the feedback. As well as Modbus step accumulation limit, it is upper limited to 10 motor revolutions.

0x60F4, following error actual value, is the difference between the ordered position and the feedback. When the motor is not moving, the value should not exceed 1.8°, i.e. 1 / 200 revolution. In this condition the drive is supplying the maximum current to the motor and consequently the motor is providing the maximum torque to the load. This condition can persist until the following error window is met or if the position deviation decreases.

During the motor movement, the position deviation in correspondence with the maximum motor torque, might vary upon speed, inertia of the load, parameters of motor winding, etc.

Notes:
While in Modbus the current position is read-write, in CANopen it is read only. In order to force position demand value to a new value, it is necessary to use homing mode: by writing on 0x6060, modes of operation = 6 and 0x607C, homing mode = 35. In this way it is possible to choose the new position value writing on 0x607C, home offset and commanding a start homing acting on 0x6040, control word. Position demand value will be set equal to home offset and position actual value will maintain the small difference, without any movement of the motor. Another method is to write directly on Modbus register, 0x2005: 0x06.

When controlling FD in interpolated position (IP) or cyclic synchronous position (CSP) modes the master shall initialize its position counter with the object 0x6062, position demand value.

10.10.19. 0x606B / 0x606C, Velocities

0x606B and 0x606C are signed 32 bits read only objects.
0x606B, velocity demand value represents the actual speed set-point; during acceleration and deceleration its value increases and decreases following the selected ramp profile.
0x606C is the velocity actual value, which is calculated from the encoder using a 20 msec filter (at low speeds the measurement can be noisy).

Note:
Since 0x606C, velocity actual value is measured directly from the encoder, it can be used to observe the maximum allowed speed of a specific movement.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.10.20. 0x6073 / 0x6078, Motor current

0x6073, Maximum current is an unsigned 16 bits object, which represents the maximum peak current that the drive supplies to the motor. Normally the rated current on the datasheet of a stepper motor is expressed as bi-polar, both phases on, i.e. the current supplied by full-step drives, where both phases are energized with the same current and by inverting one at a time their sign, the motor spins. Such rated current is established in the conditions of motor not running and both phases energized at rated current, so that after a certain period of time the motor will not exceed the maximum temperature.
FD stepper drives, instead, work in sinusoidal micro-step, hence the maximum current parameter represents the peak of the sine wave of the motor current. In order to be compared with the motor rated current, its r.m.s. (√2 of the peak) shall be used.

When the motor is not running, the main power dissipation is given by the Joule effect, RI^2. When rotating at high speeds fast changing of magnetic fields generates Foucault's currents, which sum up to increase the thermal power dissipation.
In order to avoid motor over-heating, it is also important to evaluate the duty cycle of the motor (continuous, intermittent, etc.).

Since FD drives exploits the integrated magnetic encoder to detect the torque applied to the motor, they are able to decrease the current when not needed, in order to let it cool down. The current setting ranges from 0x6073, Maximum current to 0x2005: 0xF6, Minimum current, depending on the torque applied.

0x0678, current actual value represents a filtered measurement of the milli-amp that flows into the motor.

10.10.21. 0x6079, DC link circuit voltage

0x6079, DC link circuit voltage is a unsigned 32 bit object that represents the voltage on the DC power supply. It is represented as 10^{-2} V.

10.10.22. 0x607A, Target position

0x607A, target position is a signed 32 bits read write object. Its default value is initialized to Cycle 0 position.

It is used in profile position mode with the meaning of:
- position destination in absolute sub-mode,
- movement distance in relative sub-mode,
- maximum movement distance in delta stop sub-mode,
- position delay and time delay in delay sub-modes.
-

In cyclic synchronous position mode, it represents the actual position set-point that is interpolated by the drive.

10.10.23. 0x607C, Home offset

0x607C, home offset is a signed 32 bits read write object, initialized to calibration position register. It represents the value of position counter at the end of homing mode movement.

10.10.24. 0x607D, Software position limits

Software position limit contains the sub-indexes *min position limit* and *max position limit*. These parameters define the absolute limits for demand position counter.
Writing one sub-index cause its activation. They are not active in homing, interpolated position cyclic synchronous position, cyclic synchronous velocity modes and when the drive is used in step/dir or quadrature step modes.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.10.25. 0x607F, Maximum profile velocity

It is an unsigned 32 bits read write object, which upper limit the register 0x6081, profile velocity. Its default value is equal to 300˙000 step / sec and it cannot be exceeded.

10.10.26. 0x6081, Profile velocity

It is an unsigned 32 bits read write object, initialized with cycle 0 speed. It represents the speed set-point of profile position modes.

10.10.27. 0x6083 / 0x6084 / 0x6086, Profile acceleration / deceleration / type

0x6083, profile acceleration and 0x6084, profile deceleration are unsigned 32 bits read write objects, and initialized with acceleration and deceleration Modbus registers. They represent the maximum speed increment / decrement every millisecond.
Their value is upper limited to 20˙000 kstep/sec^2.

0x6086, motion profile type is a signed 16 bits read write object, initialized to bits 7 and 8 of configuration register. Its value determines the speed ramp:

- 0: linear,
- 1: parabolic,
- 2: s-curve.

10.10.28. 0x6098 / 0x6099 / 0x609A, Homing method / speeds / acceleration

0x6098, homing method is a signed 32 bits read write object.
0x6099, homing speeds is an array of 3 elements:

- Sub-index 0: unsigned 8 bits read only, number of entries = 2
- Sub-Index 1: unsigned 32 bits read write, research speed
- Sub-index 2: unsigned 32 bits read write, release speed

0x609A, homing acceleration is an unsigned 32 bits read write object, initialized with acceleration register. Its value is upper limited to 20˙000 kstep/sec^2 and it used only during homing movements.

For a detailed description refer to ch. 10.9.3.

10.10.29. 0x60C0 / 0x60C1 / 0x60C2 / 0x60C4, Interpolation

0x60C0, interpolation sub-mode select is a signed 16 bits read only object, currently equal to 0: linear interpolation.
0x60C1, interpolation data record is a record of just two objects:

- Sub-index 0: unsigned 8 bits read only, number of entries = 1,
- Sub-index 1: signed 32 bits read write object is the interpolated position, which constitutes the entry point of the interpolated positions buffer.

0x60C2, interpolation time period is used to define the time distance between two synchronized interpolated positions.

- Sub-index 0: unsigned 8 bits read only, number of entries = 2,
- Sub-index 1: unsigned 8 bits read write, interpolation time units, initialized to 1,
- Sub-index 2: signed 8 bits read write, interpolation time index, initialized to-3.

The period is calculated as *interpolation time unit* · $10^{\text{interpolation time index}}$ seconds.
0x60C4, interpolation data record is used to define the buffer of interpolated points:

- Sub-index 0: unsigned 8 bits read only, number of entries = 6,
- Sub-index 1: unsigned 32 bits read only, maximum buffer size = 32,
- Sub-index 2: unsigned 32 bits read only, actual buffer size,
- Sub-index 3: unsigned 8 bits read only, buffer organization = 0,
- Sub-index 4: unsigned 16 bits read only, buffer position = 0,

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

- Sub-index 5: unsigned 8 bits read only, size of data record = 1,
- Sub-index 6: unsigned 8 bits write only, write 0 to clear the buffer.

Notes:
The object 0x60C1 is a record because CANopen standard offers the possibility to define the interpolated motion with a set of multiple parameters for each interpolation point, e.g. position, speed, motor current, etc. FD drives use instead just a single parameter, which is the position, but keep the record format defined from the standard.

EtherCAT simplifies this approach using cyclic synchronous position mode, where only the 0x607A, target position is transmitted.

Interpolation time period generally varies in the range 1 to 10 milliseconds.

10.10.30. 0x60FD, Digital inputs

It is unsigned 32 bits read only object, whose bits assume the meaning of below table:

Bit number	FD1E	FD2E
0	Hardware limit switch down active	
1	Hardware limit switch up active	
2	Homing sensor active	
…		
16	IN1	IN1
17	IN2	IN2
18		IN3
19		IN4
20		IN5
21		IN6
22		IN7
23	OUT1	OUT1
24		OUT2
25		OUT3

Tab. 63 – 0x60FD, Digital inputs

10.10.31. 0x60FE, Digital outputs

Digital outputs is a record used to force the outputs or to read their status.
- Sub-index 0: unsigned 8 bits read only, number of entries = 2,
- Sub-index 1: unsigned 32 bits read write, physical outputs. When read it represents the status of the output, when written, if the corresponding bit on the mask is 1, it forces open or close the output.
- Sub-index 2: unsigned 32 bits read write, bit mask. When its bits are written, it enables (1) or disables (0) the forcing of physical outputs.

10.10.32. 0x60FF, Target velocity

0x60FF, target velocity is signed 32 bits read write object, initialized to cycle 0 speed, which is the profile velocity and cyclic synchronous velocity modes speed set-point. Positive values create movements towards increasing position, negative values towards decreasing positions. It is upper limited to 300'000 step/sec and lower limited to -300'000 step/sec.
Writing the object during profile velocity mode determines speed ramp up or down to new speed set-point using profile acceleration and deceleration.
Writing the object during cyclic synchronous velocity mode immediately takes the speed setpoint, hence the master shall gradually increase the value during acceleration and decrease it during decelerations.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10.11. ESC registers

Refer to LAN9252 datasheet for further information.

Offset	Size [Bytes]	Bits	Name	R/W	Description
ESC Information					
0x0000	1	7:0	EtherCAT Controller Type	RO	0xC0 = Microchip
0x0001	1	7:0	EtherCAT Controller Revision	RO	0x02
0x0002	2	15:0	EtherCAT Controller Build	RO	0x0000
0x0004	1	7:0	Supported FMMUs	RO	0x03
0x0005	1	7:0	Supported SyncManagers	RO	0x04
0x0006	1	7:0	Process Data RAM Size	RO	0x04
0x0007	1	7:6	Port 3 Configuration	RO	00: Not implemented
		5:4	Port 2 Configuration	RO	01: Not configured
		3:2	Port 1 Configuration	RO	11: MII/RMII
		1:0	Port 0 Configuration	RO	11: MII/RMII
0x0008	2	11	Fixed FMMU/SyncManager Configuration	RO	0: Variable configuration
		10	EtherCAT Read/Write Command Support	RO	0: Supported
		9	EtherCAT LRW Command Support	RO	0: Supported
		8	Enhanced DC SYNC Activation	RO	1: Available
		7	Separate Handling of FCS Errors	RO	1: Supported, frame with wrong FCS and additional nibble will be counted separately in Forwarded RX Counter
		6	Enhanced Link Detection MII	RO	1: Available
		5	Enhanced Link Detection EBUS	RO	0: Not available
		4	Low Jitter EBUS	RO	0: Not available, standard jitter
		3	Distributed Clocks (width)	RO	1: 64-bit
		2	Distributed Clock	RO	1: Available
		0	FMMU Operation	RO	0: Bit oriented
Station Address					
0x0010	2	15:0	Configured Station Address	R/W	0x0000, This field contains the address used for node addressing (FPxx commands)
0x0012	2	15:0	Configured Station Alias Address	RO	0x0000, This field contains the alias address used for node addressing (FPxx commands). The use of this alias is activated by the Station Alias bit of the ESC DL Control Register. EEPROM value is only taken over at first EEPROM load after lower-on reset.
Write Protection					
0x0020	1	0	Write Register Enable	R/W	If write protection is enabled, this register must be written in the same Ethernet frame (value is a don't care) before other writes to this station are allowed. Write protection is still active after this frame (if the Write Register Protection Register is not changed)
0x0021	1	0	Write Register Protection	R/W	0: Protection disabled 1: Protection enabled Note: Registers 0000h-0F0Fh are write protected, except for 0030h.
0x0030	1	0	ESC Write Register Enable	R/W	If ESC write protection is enabled, this register must be written in the same Ethernet frame (value is a don't care) before other writes to this station are allowed. ESC write protection is still active after this frame (if the ESC Write Register Protection Register is not changed)
0x0031	1	0	ESC Write Register Protection	R/W	0: Protection disabled 1: Protection enabled Note: All areas are write protected, except for 0030h.
Data Link Layer					

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Offset	Size [Bytes]	Bits	Name	R/W	Description
0x0040	1	7:0	Write: ESC Reset ECAT	W	A reset is asserted after writing 52h (”R”), 45h (“E”), and 53h (“S”) in this register with 3 consecutive commands.
		1:0	Read: Reset Procedure Progress	R	01: After writing 52h 10: After writing 45h (if 52h previously written) 00: Else
0x0100	4	DL Control Register			
		24	Station Alias	R/W	Default 0: Ignore station alias 1: Alias can be used for all configured address command types (FPRD, FPWR, etc.)
		19	EBUS Low Jitter	R/W	Default 0: Normal jitter 1: Reduced jitter
		18:16	RX FIFO Size/RX Delay Reduction (ESC delays start of forwarding until FIFO is at least half full)		111: Default
		13:12	Loop Port 2	R/W	Default 00: Auto. 01: Auto Close. 10: Open. 11: Closed.
		11:10	Loop Port 1	R/W	Default 00: Auto. 01: Auto Close. 10: Open. 11: Closed.
		9:8	Loop Port 0	R/W	Default 00: Auto. 01: Auto Close. 10: Open. 11: Closed.
		1	Temporary Use of Register 0101h Settings	R/W	0: Permanent Use 1: Temporarily use for ~1 s, then revert to previous settings.
		0	Forwarding Rule	R/W	0: EtherCAT frames are processed, Non-EtherCAT frames are forwarded without processing Default 1: EtherCAT frame are processed, Non-EtherCAT frames are destroyed. The source MAC address is changed for every frame (SOURCE_MAC[1] is set to 1 - locally administered address) regardless of the forwarding rule.
0x0108	2	15:0	Physical Read/Write Offset		Default: 0 Offset of R/W commands (FPRW, APRW) between Read address and Write address. RD_ADR- ADR and WR_ADR = ADR + R/W-offset.
0x0110	2	DL Status Register			
		11	Communication on Port 1	RO	0: No stable communication 1: Communication established
		10	Loop Port 1	RO	0: Open. 1: Closed.
		9	Communication on Port 0	RO	0: No stable communication 1: Communication established
		8	Loop Port 0	RO	0: Open. 1: Closed.
		5	Physical Link on Port 1	RO	0: No link 1: Link detected
		4	Physical Link on Port 0	RO	0: No link 1: Link detected
		2	Enhanced Link Detection	RO	0: Deactivated for all ports 1: Activated for at least one port
		1	PDI Watchdog Status	RO	0: Watchdog expired 1: Watchdog reloaded

Offset	Size [Bytes]	Bits	Name	R/W	Description
		0	PDI Operational/EEPROM Loaded Correctly	RO	0: EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1: EEPROM loaded correctly, PDI operational (access to Process Data RAM)
Application Layer					
0x0120	2	AL control register			
		4	Error Ind Ack	R/W	0: No Ack of Error Ind in AL status register 1: Ack of Error Ind in AL status register
		3:0	Initiate State Transition of Device State Machine	R/W	1: Request Init State 2: Request Pre-Operational State 3: Request Bootstrap State 4: Request Safe-Operational State 8: Request Operational State
0x0130	2	AL status register			
		4	Error Ind	RO	0: Device is in state as requested or Flag cleared by command 1: Device has not entered requested state or changed state as a result of a local action
		3:0	Actual State of the Device State Machine	RO	1: Init State 2: Pre-Operational State 3: Bootstrap State 4: Safe-Operational State 8: Operational State
0x0134	2	15:0	AL status code register	RO	Default: 0
0x0138	1	RUN LED Override register			
		4	RUN Override	R/W	0: Override disabled 1: Override enabled
		3:0	RUN LED Code	R/W	0x0: Off 0x1-0xC: Flash 1x-12x 0xD: Blinking 0xE: Flickering 0xF: On
Interrupts					
0x0200	2	15:0	EtherCAT Event Mask	R/W	ECAT event masking of the ECAT Event Request register Events for mapping into the ECAT event fields of EtherCAT frames. 0: Corresponding ECAT Event Request register bit is not mapped 1: Corresponding ECAT Event Request register bit is mapped
0x0210	2	EtherCAT Event Request			
		7	SyncManager Ch.3 Status Mirror	RO	This bit mirrors the value of the SyncManager Ch.x Status. 0: No Sync Ch.x Event 1: Sync Ch.x Event Pending
		6	SyncManager Ch.2 Status Mirror	RO	
		5	SyncManager Ch.1 Status Mirror	RO	
		4	SyncManager Ch.0 Status Mirror	RO	
		3	AL Status Event	RO	0: No change in AL Status 1: AL Status Change Note: This bit is cleared by reading the AL Status Register from EtherCAT.
		2	DL Status Event	RO	0: No change in DL Status 1: DL Status Change Note: This bit is cleared by reading the ESC DL Status Register from EtherCAT.
		0	DC Latch Event	RO	0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs Note: This bit is cleared by reading the DC Latch event times from EtherCAT for EtherCAT controlled Latch Units, so that the LATCH0 Status Register/LATCH1 Status Register indicates no event.

Offset	Size [Bytes]	Bits	Name	R/W	Description
Error Counters					
0x0300	2	15:8	Port 0 RX Error Counter	R/WC	Counting is stopped when 0xFF is reached. This register is cleared if any one of the RX Error Counter Registers is written.
		7:0	Port 0 Invalid Frame Counter	R/WC	
0x0302	2	15:8	Port 1 RX Error Counter	R/WC	
		7:0	Port 1 Invalid Frame Counter	R/WC	
0x0308	1	7:0	Port 0 Forwarded RX Error Counter	R/WC	
0x0309	1	7:0	Port 1 Forwarded RX Error Counter	R/WC	
0x030C	1	7:0	ECAT Processing Unit Error Counter	R/WC	Counting is stopped when 0xFF is reached. This field counts the errors of frames passing the Processing Unit (e.g., FCS error or datagram structure error).
0x0310	1	7:0	Port 0 Lost Ling Counter	R/WC	Counting is stopped when 0xFF is reached. This counter only counts if port loop is Auto or Auto-Close. This register is cleared if any one of the Lost Link Counter Registers is written.
0x0311	1	7:0	Port 1 Lost Ling Counter	R/WC	
Watchdogs					
0x0400	2	15:0	Watchdog Divider	R/W	Default: 0x09C2 Number of 25MHz ticks (minus 2) that represents the basic watchdog increment. (default value is 100 us = 2498)
0x0410	2	15:0	Watchdog Time PDI	R/W	Default: 0x03E8 Number of basic watchdog increments (default value with Watchdog Divider of 100 us results in 100 ms watchdog). The watchdog is disabled if Watchdog Time PDI is set to 0000h. Watchdog is restarted with every PDI access.
0x0420	2	15:0	Watchdog Time Process Data	R/W	Default: 0x03E8 Number of basic watchdog increments (default value with Watchdog Divider of 100 us results in 100 ms watchdog). There is one watchdog for all SyncManagers. The watchdog is disabled if Watchdog Time PDI is set to 0x0000. The watchdog is restarted with every write access to the SyncManagers with the Watchdog Trigger Enable bit set.
0x0440	2	15:0	Watchdog Status of Process Data	RO	(triggered by SyncManagers) 0: Watchdog Process Data expired 1: Watchdog Process Data is active or disabled
0x0442	1	7:0	Watchdog Counter Process Data	R/WC	Counting is stopped when FFh is reached. Counts if Process Data Watchdog expires. This field is cleared if one of the Watchdog counters (0442h-0443h) is written.
EEPROM Interface					
0x0500	1	EEPROM Configuration			
		1	Force ECAT Access	R/W	0: Do not change 1: Reset
		0	PDI EEPROM Control	R/W	0: No 1: Yes (PDI has EEPROM control) EtherCAT controls the SII EEPROM interface if the PDI EEPROM Control bit of the EEPROM Configuration Register is 0 and the Access to EEPROM bit of the EEPROM PDI Access State Register is 0. Otherwise, PDI controls the EEPROM interface.
0x0501	1	0	EEPROM PDI Access State	RO	0: Do not change 1: Reset
		EEPROM Control/Status Register			
0x0502	2	15	Busy	RO	0: EEPROM interface is idle 1: EEPROM interface is busy

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Offset	Size [Bytes]	Bits	Name	R/W	Description
		14	Error Write Enable	RO	0: No error 1: Write Command without Write enable
		13	Error Acknowledge/Command	RO	0: No error 1: Missing EEPROM acknowledge or invalid command
		12	EEPROM Loading Status	RO	0: EEPROM loaded, device information okay 1: EEPROM not loaded, device information not available (EEPROM loading in-progress or finished with a failure)
		11	Checksum Error in ESC Configuration Area	RO	0: Checksum okay 1: Checksum error
		10:8	Command Register	R/W	Write: Initiate command Read: Currently executed command 000: No command/EEPROM idle (clear error bits) 001: Read 010: Write 100: Reload Others: RESERVED / invalid commands (no not issue)
		7	Selected EEPROM Algorithm	RO	1: 2 address bytes (32Kbit- 4Mbit EEPROMs)
		6	Supported Number of EEPROM Bytes	RO	0: 4 Bytes 1: 8 Bytes
		5	EEPROM Emulation	RO	0: Normal operation (I2C interface used) 1: PDI emulates EEPROM (I2C not used)
		0	ECAT Write Enable	R/W	0: Write requests are disabled 1: Write requests are enabled
0x0504	4	31:0	EEPROM Address	R/W	
0x0508	4	31:0	EEPROM Read/Write Data	R/W	Only lower two Bytes can be written. All four Bytes can be read.
MII Management Interface					
0x0510	2	MII Management Control/Status			
		15	Busy	RO	0: MI control state machine is idle 1: MI control state machine is active
		14	Command Error	RO	0: Last command was successful 1: Invalid command or write command without write enable Note: Cleared with a valid command or by writing "00" to Command Register.
		13	Read Error	R/W	0: No read error 1: Read error occurred (PHY or register bi available)
		9:8	Command Register	R/W	Write: Initiate command. Read: Currently executed command Commands: 00: No command / MI Idle (clear error bits) 01: Read 10: Write 11: RESERVED (do not issue)
		7:3	PHY Address Offset	RO	Deafult: 0x0000
		2	MI Link Detection	RO	0: Not available 1: MI Link Detection Active
		1	Management Interface Control	RO	0: ECAT control only Default 1: MPDI control possible (MII Management ECAT Access State Register and MII Management PDI Access State Register)
		0	Write Enable	R/W	0: Write Disabled 1: Write Enabled
0x0512	1	4:0	Phy Address	R/W	Default: 0x0000
0x0513	1	4:0	Address of PHY Register to be Read/Written	R/W	
0x0516	1	0	Access to MII Management (ECAT)	R/W	0: ECAT enables PDI takeover of MII management

Offset	Size [Bytes]	Bits	Name	R/W	Description
					control 1: ECAT claims exclusive access to MII management
0x0517	1	1	Force PDI Access State	R/W	0: Do not change Access to MII Management (PDI) bit 1: Reset Access to MII Management (PDI) bit
		0	Access to MII Management (PDI)	RO	0: ECAT has access to MII management 1: PDI has access to MII management
0x0518	1	5	Port x Lost Link Counter	R/WC	0: No Update 1: PHY Configuration was Updated Note: Cleared by writing any value to at least one of the PHY Port Status Registers.
		4	Port x Link Partner Error	RO	0: No Error Detected 1: Link Partner Error
		3	Port x Read Error	R/WC	0: No Read Error Detected 1: Read Error has Occurred Note: Cleared by writing any value to at least one of the PHY Port Status Registers.
		2	Port x Link Status Error	RO	0: No Error 1: Link Error, Link Inhibited
		1	Port x Link Status	RO	(100 Mbit/s, Full-Duplex, Auto-negotiation) 0: No Link 1: Link Detected
		0	Port x Physical Link	RO	(PHY Status Register 1.2) 0: No Physical Link 1: Physical Link Detected
FMMU					
LAN9252 includes 3 FMMUs. Each FMMU is described in 16 Bytes, starting at 0x0600. FMMU0 base address 0x0600; FMMU1 base address 0x0610; FMMU2 base address 0x0620. The subsequent FMMU registers will be referenced as an offset from these various base addresses. The variable "x" is used in the following descriptions to represent FMMUs 0 through 2.					
+ 0x0	4	31:0	Logical Start Address	R/W	Logical start address within the EtherCAT address space.
+ 0x4	2	15:0	Length	R/W	Offset from the first logical FMMU byte to the last FMMU Byte + 1 (e.g., if two bytes are used, then this parameter shall contain 2).
+ 0x6	1	2:0	Logical Start Bit	R/W	Logical starting bit that shall be mapped (bits are counted from least significant bit (0) to most significant bit (7)).
+ 0x7	1	2:0	Logical Stop Bit	R/W	Last logical bit that shall be mapped (bits are counted from least significant bit (0) to most significant bit (7)).
+ 0x8	2	15:0	Physical Start Address	R/W	Mapped to logical start address
+ 0xA	1	2:0	Physical Start Bit	R/W	Physical starting bit as target of logical start bit mapping (bits are counted from least significant bit (0) to most significant bit (7)).
+ 0xB	1	FMMU Type			
		0	Write Access Mapping	R/W	0: Ignore mapping for write accesses 1: Use mapping for write accesses
		1	Read Access Mapping	R/W	0: Ignore mapping for read accesses 1: Use mapping for read accesses
+0xC	1	0	FMMU Activation	R/W	0: FMMUx Deactivated 1: FMMUx Activated. FMMUx checks logical addressed blocks to be mapped according to the configured mapping.
SyncManager					
LAN9252 includes 4 SyncManagers. Each SyncManager is described in 8 Bytes, starting at 0800h. SyncManager 0 base address 0x0800; SyncManager 1 base address 0x0808; SyncManager 2 base address 0x0810; SyncManager 3 base address 0x0818. The subsequent SyncManager registers will be referenced as an offset from these various base addresses. The variable "x" is used in the following descriptions to represent SyncManagers 0 through 3. Configuration registers can only be written if SyncManager x is disabled via the SyncManager Enable/Disable bit of the SyncManager x Activate Register.					
+ 0x0	2	15:0	Physical Start Address	R/W	Specifies the first byte that will be handled by SyncManager x.

Offset	Size [Bytes]	Bits	Name	R/W	Description
+ 0x2	2	15:0	Length	R/W	Number of bytes assigned to SyncManager x. This field shall be greater than 1, otherwise the SyncManager is not activated. If set to 1, only Watchdog Trigger is generated, if configured.
+0x4	1	SyncManager x Control Register			
		6	Watchdog Trigger Enable	R/W	0: Disabled 1: Enabled
		5	Interrupt in PDI Event Request Register	R/W	0: Disabled Default 1: Enabled
		4	Interrupt in ECAT Event Request Register	R/W	0: Disabled Default 1: Enabled
		3:2	Direction	R/W	00: Read: EtherCAT master read access 01: Write: ECAT master write access
		1:0	Operation Mode	R/W	00: Buffered (3 buffer mode) 10: Mailbox (single buffer mode)
+ 0x5	1	SyncManager x Status Register			
		7	Write Buffer in Use	RO	
		6	Read Buffer in Use	RO	
		5:4	Buffer Status	RO	00: 1. buffer 01: 2. buffer 10: 3. buffer 11: No buffer written
		3	Mailbox Status	RO	0: Mailbox Empty 1: Mailbox Full
		1	Interrupt Read	RO	0: Interrupt cleared after first byte of buffer was written 1: Interrupt after buffer was completely and successfully read
		0	Interrupt Write	RO	0: Interrupt cleared after first byte of buffer was read 1: Interrupt after buffer was completely and successfully written
+ 0x6	1	SyncManager x Activate Register			
		7	Latch Event PDI	R/W	0: No 1: Generate latch events if stepper application issues a buffer exchange or if it accesses buffer start address.
		6	Latch Event ECAT	R/W	0: No 1: Generate latch event if EtherCAT master issues a buffer exchange.
		1	Repeat Request	R/W	A toggle of Repeat Request indicates that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)
		0	SyncManager Enable/Disable	R/W	0: Disable: Access to memory without SyncManager control 1: Enable: SyncManager is active and controls memory area set in configuration.
Distributed Clocks- Receive Times					
0x0900	4	31:0	Receive Time Port 0	R/W	Write: A write access to register 090xh with BWR, APWR (any address) or FPWR (configured address) latches the local time of the beginning of the receive frame (start first bit of preamble) at each port. Read: Local time of the beginning of the last receive frame containing a write access to this register. Note: The time stamps cannot be read in the same frame in which this register was written.
0x0904	4	31:0	Receive Time Port 1	R/W	
Distributed Clocks- Time Loop Control Unit					

Offset	Size [Bytes]	Bits	Name	R/W	Description
0x0910	8	System Time			
		63:0	Read Access	RO	Local copy of the System Time when the frame passed the reference clock (i.e., including System Time Delay). Time latched at beginning of the frame (Ethernet SOF delimiter).
		31:0	Write Access	W	Written value will be compared with the local copy of the system time. The result is an input to the time control loop.
0x0918	8	63:0	Receive Time EtherCAT Processing Unit	RO	Local time of the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to Receive Time Port 0 Register (0900h).
0x0920	8	63:0	System Time Offset	R/W	Difference between local time and System Time. Offset is added to local time.
0x0928	4	31:0	System Time Delay	R/W	Delay between Reference Clock and the ESC
0x092C	4	31	System Time Difference	RO	0: Local copy of System Time greater than or equal to received System Time 1: Local copy of System Time smaller than received System Time
		30:0		RO	Mean difference between local copy of System Time and received System Time values.
…			Speed counter…		
Distributed Clocks- Cyclic Unit Control					
0x0980	1	Cyclic Unit Control Register			
		5	Latch In Unit 1		Default 0: EtherCAT master controlled 1: Stepper application controlled
		4	Latch In Unit 0		
		0	Sync Out Unit Control	R/W	
Distributed Clocks- SYNC Out Unit					
0x0981	1	Activation			
		7	SyncSignal Debug Pulse (Vasili Bit)	R/W	Default 0: Deactivated 1: Immediately generate a single debug ping on SYNC0 and SYNC1 according to bits 2 and 1 of this register.
		6	Near Future Configuration (approx.)	R/W	Default 0: 1/2 DC width future 1: 2.1 sec future
		5	Start Time Plausibility Check	R/W	0: Disabled. SyncSignal generation if Start Time is reached. 1: Immediate SyncSignal generation if Start Time is outside Near Future Configuration (approx.).
		4	Extension of Start Time Cyclic Operation	R/W	0: No extension 1: Extend 32-bit written Start Time to 64-bit
		3	Auto-activation	R/W	0: Disabled 1: Auto-activation enabled. Sync Out Unit Activation is set automatically after Start Time is written.
		2	SYNC1 Generation	R/W	0: Deactivated 1: SYNC1 pulse is generated
		1	SYNC0 Generation	R/W	0: Deactivated 1: SYNC0 pulse is generated
		0	Sync Out Unit Activation	R/W	0: Deactivated 1: Activated
0x0982	2	15:0	Pulse Length of Sync Signals	RO	Default 0. In units of 10ns, a value of 0 is used for Acknowledge Mode: SyncSignal will be cleared by reading the SYNC0 Status Register/SYNC1 Status Register.
0x0984	1	Activation Status			
		2	Start Time Cyclic Operation plausibility check result when Sync Out Unit was activated	RO	0: Start Time was within near future 1: Start Time was out of near future

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Offset	Size [Bytes]	Bits	Name	R/W	Description
		1	SYNC1 Activation State	RO	0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending
		0	SYNC0 Activation State	RO	0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending
0x098E	1	0	SYNC0 State for Acknowledge Mode	RO	SYNCx, in Acknowledge Mode, is cleared when read by local application.
0x098F	1	0	SYNC1 State for Acknowledge Mode	RO	
0x0990	8	63:0	Start Time Cyclic Operation	R/W	Write: Start time (System Time) of cyclic operation in ns. Read: System time of next SYNC0 pulse in ns.
0x0990			Start Time Cyclic Operation		
0x0998			Next SYNC1 Pulse		
0x09A0			SYNC0 Cycle Time		
0x09A4			SYNC1 Cycle Time		

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

11. CYCLES AND SEQUENCES

FD drives can be programmed with up to 32 user defined cycles. Every cycle is identified by five parameters: type, speed, position, direction and delta stop µsteps. Acceleration and deceleration are common for all the cycles. Types are:

1: Jog,
2: Indexer relative,
3: Calibration,
4: Delta stop,
5: Indexer absolute,
6: Position delay,
7: Time delay.

Cycles can be combined in sequences and executed in stream, one after the other. Each sequence is described by 20 parameters, each of them identifies a cycle number or a command (Stop or Loop). A total of 10 sequences made of 20 parameters each are available.

Cycles and sequences can be selected, started and stopped using dedicated Modbus registers, digital inputs or EtherCAT objects.

When a sequence is selected, the start command launches the cycle identified by the first sequence parameter. As soon as this cycle finishes, the motor performs the cycle identified by the second parameter and so on.

When a Stop parameter (254) is met in the sequence, the motor decelerates till stopping; when it is a Loop parameter (255) the sequence restarts from the first cycle.
When neither Loop nor Stop are present, as the drive finishes one sequence of 20 cycles, it starts to perform the next sequence. This means a maximum loop sequence of 200 cycles or, if the last parameter in the last sequence is a Stop, a single sequence made of 199 cycles.

Every sequence is interruptible via stop command. In this case the sequence pointer is reset and a future start runs the first cycle.

Cycles and sequences parameters can be modified in RAM.

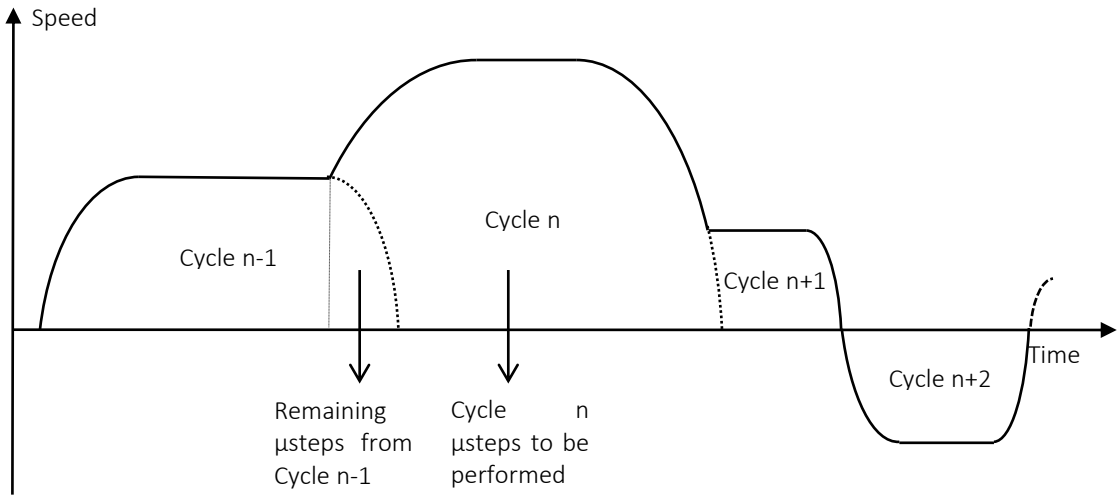


Fig. 7 – Sequence of cycles with parabolic ramps

Note:
The single cycle is considered finished when the remaining µsteps are equal to the deceleration ramp area, in this condition the next cycle starts, performing its own µsteps and the µsteps left from the previous cycle, as shown in Fig. 6 motor speed ramps up or down to the new speed set-point.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

11.1. Cycle and sequence selection

When no input is used as cycle or sequence selection, cycles and sequences can be selected using SEL_CYC_SEQ register:

Select Cycle [0 – 31] → SEL_CYC_SEQ from 0 to 31
Select Sequence [0 – 9] → SEL_CYC_SEQ from 32 to 41

Otherwise, when the selection is made using digital inputs, the meaning associated to inputs combination depends on CONFIG register bit 13, 32_CYCLES.

CONFIG register bit 13 = 0 → DI selects 10 Sequence and 22 Cycles
CONFIG register bit 13 = 1 → DI selects 32 Cycles

IN2, IN3, IN4, IN5 can be used for cycle or sequence selection. Inputs don't have a pre-assigned bit weight, the weight is ordered with the input number (weight zero is assigned to the lowest input number), e.g. if only IN3 and IN5 are used for sequence selection (CONFIG register bit 13 equal to zero), only sequences 0, 1, 2, 3 can be selected. The same sequences can be selected using only IN2 and IN4.
Using all the four inputs and CONFIG register bit 13 equal to zero, it is possible to select sequences from 0 to 9 and the remaining bits configurations from 10 to 15 are used to select single cycles from 10 to 15.

Note:
When the cycle or sequence selection is made using DI, writing SEL_CYC_SEQ register has no effect, as the selected value will be immediately overwritten by DI configuration.

To configure an input as cycle or sequence selection the register IN_x_CNF shall be set to 11. To do so it is possible to program such configuration in flash using IAP or write it via fieldbus in RAM.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

11.2. Jog

When the jog cycle is started the motor accelerates at the selected speed and direction.

During jog cycle the STATUS_WORD register bit 3 MODE_JOG is set.

For all the other types of cycles the start command samples the cycle or sequence selection and the selection is not anymore read till the end of the movement. For Jog cycles, instead, it is possible to switch between them just by changing the selection, without passing through motor stop. An example using FD2E follows:

CONFIG register, bit 13	1: Select 32 cycles
IN1 configuration	1: Start NO,
IN2 configuration	1: Stop NO,
IN3 configuration	11: Cycle or sequence selection,
IN4 configuration	11: Cycle or sequence selection,

Cycle 0, Type	0: Jog,	Cycle 2, Type	0: Jog,
Cycle 0, Speed	1'000 µstep/sec	Cycle 2, Speed	3'000 µstep/sec
Cycle 0, Direction	0: CW, increasing positions	Cycle 2, Direction	0: CW, increasing positions
Cycle 1, Type	0: Jog,	Cycle 3, Type	0: Jog,
Cycle 1, Speed	1'000 µstep/sec	Cycle 3, Speed	3'000 µstep/sec
Cycle 1, Direction	1: CCW, decreasing positions	Cycle 3, Direction	1: CCW, decreasing positions

In this example the selection addresses cycles and IN3 has weight bit 0 and IN4 has weight bit 1. Jog cycles 0, 1, 2 and 3 have been configured in order to have IN4 direction, IN5 low and high speed:

IN3 = 0	CW, increasing positions
IN3 = 1	CCW, decreasing positions
IN4 = 0	Low speed 1'000 µstep/sec
IN4 = 1	High 3'000 µstep/sec

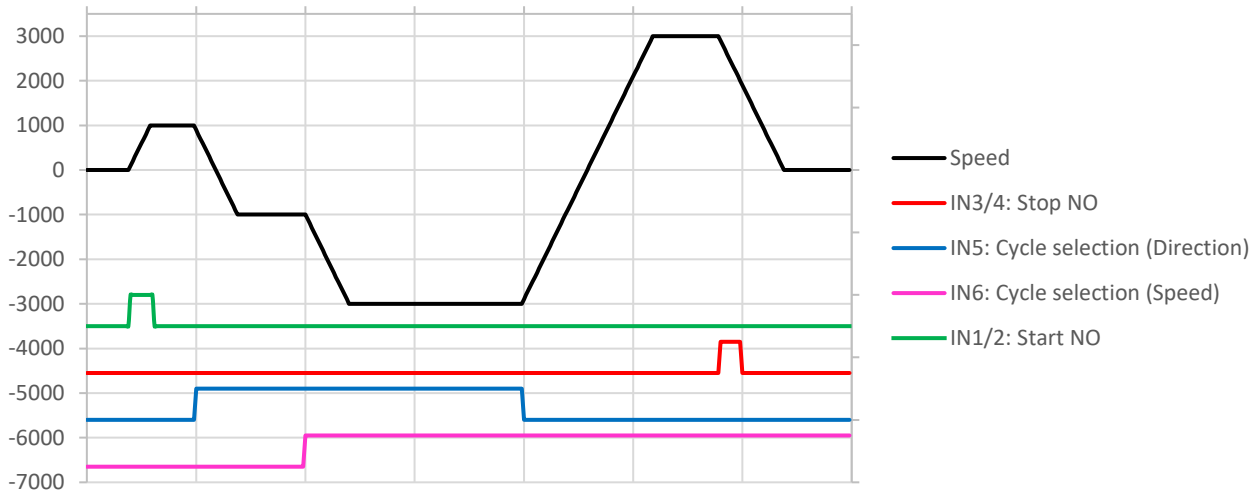


Fig. 7 – Jog

Note:
When a cycle selection involves a change of direction, the motor speed ramps down to zero and then ramp up to the new speed set-point in the opposite direction. If the selection changes again during the deceleration to another Jog at the original direction, anyhow the motor will decelerate to zero speed and then it reaccelerates at the last selected direction.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

11.3. Indexer

Indexer cycles are motor movements to a precise position destination. The destination can be relative, expressed as an increment or decrement of the current position, or absolute, expressed as the final position of the movement.

Indexer relative cycles are defined by speed, position and direction. Position is an unsigned 32 bits register and direction can be:

0: CW towards increasing positions,
1: CCW towards decreasing positions.

Indexer absolute cycles are defined by speed and position. Position is a signed 32 bits register.

When the movement is completed, the target position is compared with the current position and, if the comparison is positive (target position reached), bit 6 TARG_POS of the STATUS_WORD is set.

Note:
When the single input is configured as Start NO / Stop NC for indexer relative movements, make sure that the signal does not bounce to avoid improper motor movements during signal bouncing. In any case it is recommended to split Start and Stop signals in two different inputs or to use indexer absolute cycles.

Sequences made of several indexer cycles can be used to configure custom made speed profiles.

In the transition between two indexer cycles of a sequence configured in the same direction, the speed increases or decreases to reach the new speed set-point, without passing through zero speed. If the zero-speed crossing is wanted, just interpose a third cycle in between at the opposite direction with zero µsteps.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

11.4. Calibration

FD drives supports several methods of calibration cycles. Methods are selected via HOMING register. They can be:

0: Marker
1: Homing simple
2: Homing, time stop, inversion
3: Homing, time stop, no inversion
4: Homing, time stop, inversion, marker
5: Homing, time stop, no inversion, marker

CALIB_POSITION will be the new position loaded in the CURR_POSITION register at the end of calibration.

During calibration cycles the STATUS_WORD bit 8 HOMING is set and bit 11 AX_CALIBRATED will be reset.
If the calibration succeeds, at the end of the cycle bit 11 AX_CALIBRATED will be set. AX_CALIBRATED will remain set until no alarm arises or no calibration cycle is launched again.

IN3 and IN4 can be set as homing inputs or hardware limit switch inputs. Calibration is possible on both configurations.

FD drives can restore the multi-turn axis calibration during power-on using the integrated absolute encoder.

11.4.1. Marker

When the HOMING register is equal to zero, which corresponds to a marker cycle, the calibration consists in a motor rotation to the absolute encoder position destination. The motor will rotate at V_RELEASE speed.

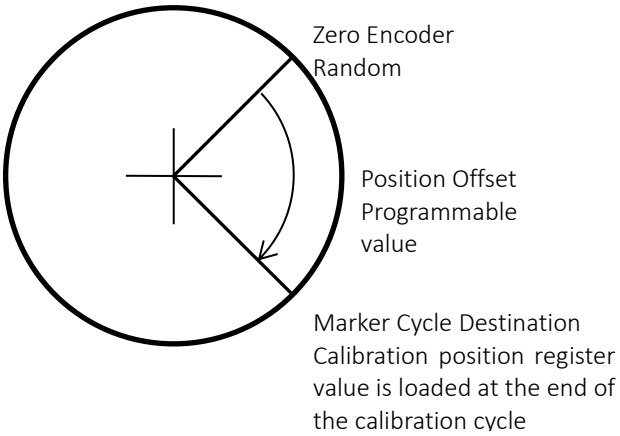


Fig. 8 – Marker Cycle

The cycle direction can be:

0: CW,
1: CCW,
2: Shortest distance.

POSITION_OFFSET register modifies the absolute encoder position destination. When POSITION_OFFSET is a zero value, the marker cycle will stop at the zero-encoder position. The zero-encoder position is a random position, which depends upon the encoder mounting.

Customer can request to have zero encoder programmed in a particular position. In this case all the drives will be delivered with such precise encoder mounting.

11.4.2. Homing simple

When the HOMING register is equal to one, which corresponds to homing simple cycle, the calibration consists in a motor rotation at V_RESEARCH speed.
The cycle direction can be:

0: CW,
1: CCW.

As soon as the homing or the hardware limit switches inputs get active the motor starts to decelerate and the calibration

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

is concluded.

11.4.3. Homing, time stop and inversion / no-inversion

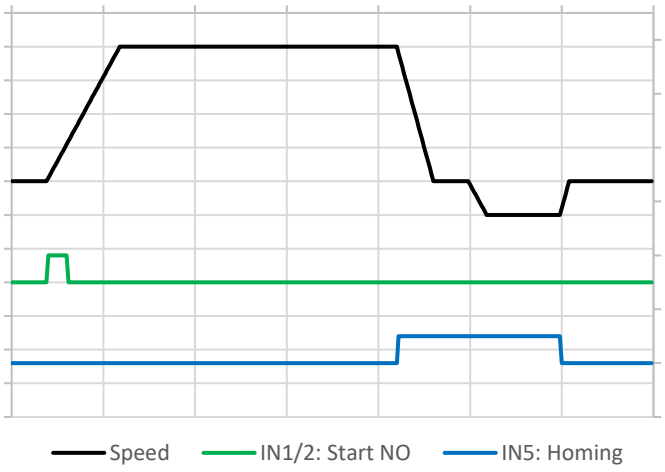


Fig. 9 – Homing invert

When the HOMING register is equal to two, which corresponds to homing, time stop and inversion cycle, the calibration consists in a motor rotation at V_RESEARCH speed.

The cycle direction can be:

0: CW,
1: CCW.

As soon as the homing or the hardware limit switches inputs get active the motor starts to decelerates, it waits TIME STOP milliseconds and then it inverts the direction running at V_RELEASE speed. The motor starts the deceleration to zero speed when the switch input gets inactive.

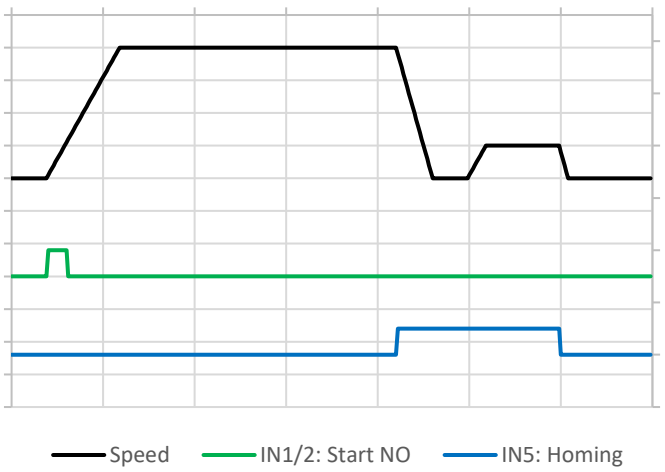


Fig. 10 – Homing no invert

If homing, time stop and no-inversion is selected (HOMING register equal to three) the motor, after waiting time stop, will accelerate to the same direction till the switch gets inactive.

11.4.4. Homing, time stop, inversion / no-inversion and marker

When the HOMING register is equal to four, which corresponds to homing, time stop, inversion and marker cycle, the calibration consists in a motor rotation at V_RESEARCH speed.

The cycle direction can be:
0: CW,
1: CCW.

As soon as the homing or the hardware limit switches inputs get active the motor will start the deceleration, it will wait TIME STOP milliseconds and then it will invert the direction running at V_RELEASE speed.
When the switch input gets inactive the motor will perform a marker cycle to the absolute encoder position destination defined in POSITION_OFFSET register.

Homing, time stop, no-inversion and marker is selected with HOMING register equal to five.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Note:
Absolute encoder position destination needs to be set approximately half motor revolution far from the deactivation of the switch to avoid the possible error of one revolution.

11.5. Delta stop

Delta stop is a particular indexer relative cycle, used when it is needed to stop the motor in a very accurate position after the activation of a sensor. A normal stop would decelerate the motor till zero speed without controlling the destination position, which would depend upon regime speed and deceleration. Delta stop cycles, instead, executes a programmed number of steps after delta stop input activation. IN3 and IN4 can be configured as delta stop inputs.

It is defined by speed, position, direction and delta stop steps.

When delta stop input gets active the motor executes exactly the steps identified by delta stop parameter and STATUS_WORD bit 20 DELTA_STOP_OK is set. Delta stop input triggers a dedicated microprocessor interrupt routine to ensure fast position sampling.

If delta stop input does not get active during the movement or it gets active only during deceleration, the cycle terminates at the indexer relative quote configured in position parameter.
To detect this event, it is possible to:

- configure multipurpose output (FD1 OUT1, FD2 OUT2) as RUNNING + /DSTOP, which would maintain the output in high state if no sensor activation is detected (timeout logic to be implemented on master side);
- evaluate cycle counter CNT_CYC and delta stop counter CNT_DSTOP registers to detect if their increment at the end of the cycle, i.e. when TARGET_POS, bit 6 of STATUS_WORD register, is set.
- Read status word delta stop ok bit.

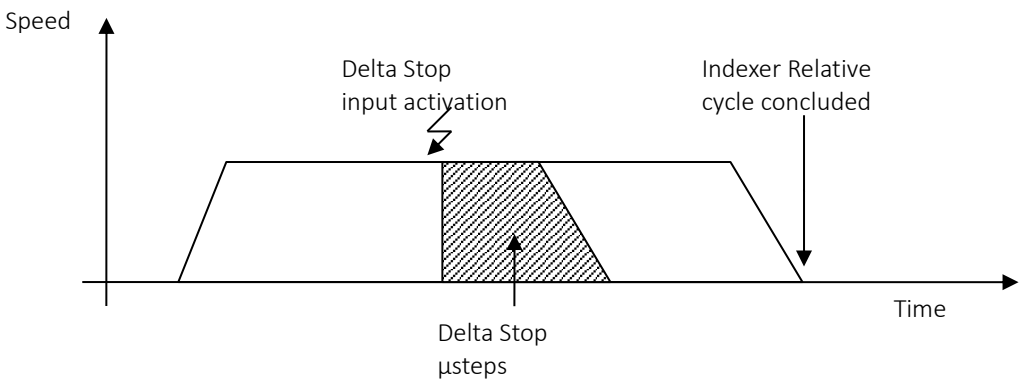


Fig. 11 – Delta Stop Cycles

Note:
The position of sensor activation shall be anticipated in respect to the wanted destination position of the delta stop steps. To avoid too steep deceleration, make sure to set Delta Stop steps parameter higher than the deceleration steps at the maximum speed used.

Being a fast input, it is needed to use shielded cables and all the relevant measures to avoid disturbances.

Delta stop cycles can also be used inside the sequences instead of relative indexer cycles in order to jump from the current cycle to the next one activating Delta Stop input.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

11.6. Position delay

Position delay cycle is used inside sequences to perform a delay in “µsteps not done” in respect to the movement which would happened without the delay cycle. Delay is configured using speed and position.

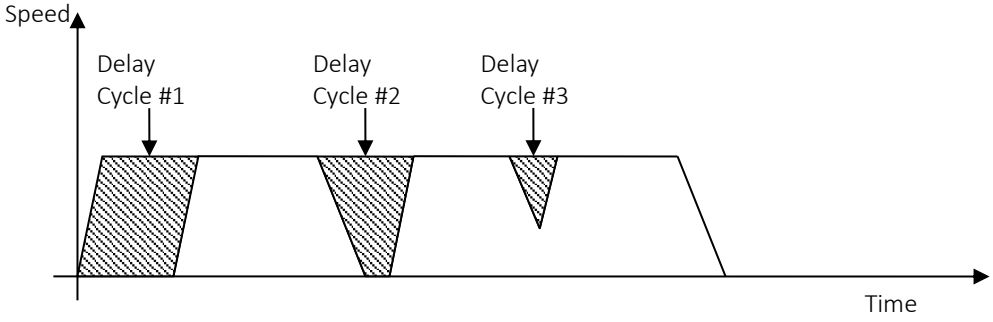


Fig. 12 – Delay Cycles

A common application of this cycle is a motor which rotates an object where to apply in pre-defined angular positions a product. The host shall give a start command to both motors: the first one will rotate at constant speed, while the second, which applies the product, will decelerate and reaccelerate in correspondence to pre-defined angular positions of product application. Angular position can be set by varying the µsteps of delay cycles and indexer cycles.

If the delay cycle is the first cycle of the sequence- Fig. 12 delay cycle #1- it works as time delay calculated as position parameter divided by speed parameter.

If the delay cycle is between two other cycles. The area of delay depends on the speeds of the cycles before and after.
If the speed of the cycle before is the same as the cycle after- Fig. 12 delay cycle #2- the time delay will be calculated using such speed. If the delay steps are not enough to reach zero speed, the motor will decelerate and reaccelerate to move back just of the delay steps- Fig. 12 delay cycle #3.
If the speed of the cycle before differs from the one indicated in the cycle after, the delay µsteps represent the grey area of Fig. 13 and Fig. 14.

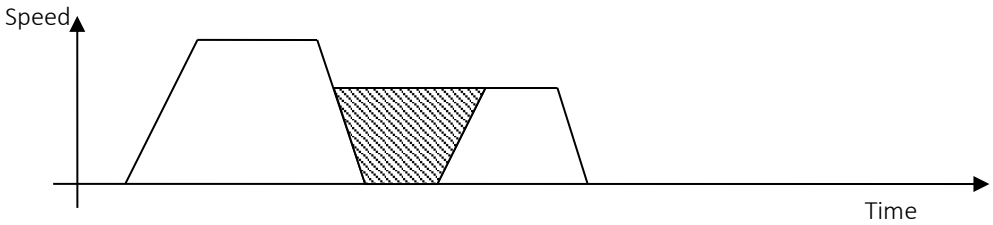


Fig. 13 – Delay cycles: speed before greater than speed after

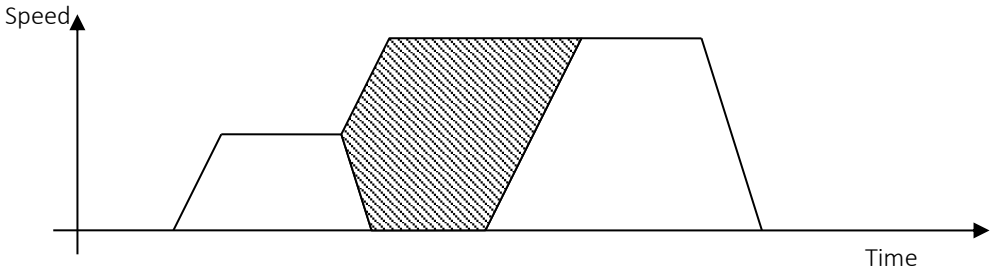


Fig. 14 – Delay cycles: speed before lower than speed after

If the cycle before and after have two different directions, the delay is treated as the first cycle of the sequence- Fig. 15 - it works as time delay calculated as position parameter divided by speed parameter.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

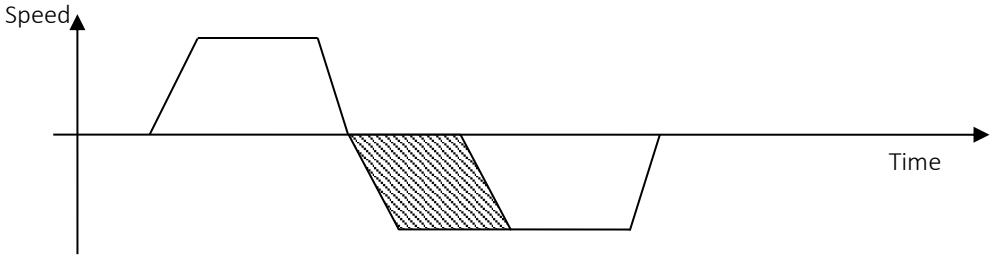


Fig. 15 – Delay cycles: direction inversion

Note:
When varying the speed with analogic input (only available in FD2) the µsteps between the motor running at regime speed and the motor running with delay cycles are respected, even when the acceleration and deceleration ramps don't allow the zero speed crossing (ref. to Delay Cycle #3 in Fig. 12).

The time delay is limited to 5 seconds.

Delay cycles are accurate with linear ramps only.

During delay, even if motor speed is zero, output running will remain active.

11.7. Time delay

When it is needed to configure a simple time delay in a sequence between two cycles, it is possible to use time delay cycles.

Position parameter will express the milliseconds of motor stop between previous and next cycle.

Note:
During delay, even if motor speed is zero, output running will remain active.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

12. COMMANDS

To reduce the overhead of messages transmitted, the start command and cycle selection has been put together in a single write command on EXE_FUN register. This command selects Cycle 0, modifies its parameters accordingly and then starts the cycle or performs below detailed functions. Only single cycles, not sequences can be started.

12.1. Rotate Right (ROR), Rotate Left (ROL)

EXE_FUN = 1, 2
or EXE_FUN = 31, 32 with no reset

The rotate right command launches a jog movement in clockwise direction (DIR = "0") at the selected speed, increasing the position counter value. When the direction is inverted (bit 6 of CONFIG register) the motor will rotate in counterclockwise direction always increasing the position counter value.

The rotate left command launches a JOG movement in counter-clockwise direction (DIR = "1") at the selected speed, decreasing the position counter value. When the direction is inverted (bit 6 of CONFIG register) the motor will rotate in clockwise direction decreasing the position counter value.

When the motor is running it is possible to modify speed and direction giving ROL/ROR commands. Giving ROL after a previous ROR command and vice versa produce a motor deceleration and re-acceleration to the opposite direction.

When the drive is in alarm, ROR and ROL commands perform a system reset. As a consequence of this type of reset, the drive will be re-initialized at the flash programmed parameters (FD1 short circuit alarm can be reset only by powering off).

In case this type of reset is not wanted, ROR_NO_RST (EXE_FUN = 31) and ROL_NO_RST (EXE_FUN = 32) can be used instead.

During jog movements the bit 3 MODE_JOG of the STATUS_WORD register is set.

12.2. Motor Stop (MST)

EXE_FUN = 3

The stop command launches a motor deceleration to zero speed.

If a calibration cycle is stopped the bit 11 AX_CALIBRATED of the STATUS_WORD is reset.

12.3. Move to Position (MVP)

EXE_FUN = 10, 11

This command launches an indexer relative cycle. When the DwLoader ABS check box is selected an absolute movement is performed and the POSITION parameter corresponds to destination position. When deselected, the motor will perform a relative movement of the total amount of µsteps defined in POSITION parameter.

In Modbus protocol two commands are used:

- MVP_ABS: Exe_Fun = 10 for absolute indexer cycles,
- MVP_REL: Exe_Fun = 11 for relative indexer cycles.

Starting at the position 10˙000, a command "MVP_REL,-1˙000" produces a motor rotation till position 9˙000.
Starting at the position 10˙000, a command "MVP_ABS,-1˙000" produces a motor rotation till position-1˙000.

After an MVP_ABS command Cycle 0 position and direction are overwritten according to a relative indexer cycle.

This command can be executed only when the motor is stopped. The cycle uses acceleration and deceleration configured in general data, speed and delta position configured in Cycle 0 data.

When the movement is completed, the target position is compared with the current position and, if the comparison is

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

positive (target position reached), bit 6 TARG_POS of the STATUS_WORD is set.

12.4. Reference Search (RFS) EXE_FUN = 13

This command launches the calibration cycle. Type of homing shall be selected using HOMING register.

12.5. Reset (RST) EXE_FUN = 14

This command performs a System Reset. As a consequence of the reset all the RAM data will return to their Flash programmed values and the alarms will be reset.

When no alarm is present the current position, the bit 11 AX_CALIBRATED and the bit 6 TARG_POS will be reloaded after reset.

12.6. Alarm Reset (ALR) EXE_FUN = 15

Alarm Reset command performs the software alarm reset.

When the Step Accumulation alarm is reset, the drive is able to restore the multi-turn position. If the axis was calibrated before that alarm occurred, the STATUS_WORD bit 11 AX_CALIBRATED will be restored. Ref. to 11.

Note:
FD1 short circuit alarm cannot be reset via software, nor with RST, nor with ALR. Ref. to 9.

12.7. Disable/Enable Motor Current (DMC/EMC) EXE_FUN = 16, 17

Disable/Enable Motor Current commands can be used to free the motor and move it to a desired position. During current disable the STATUS WORD bit 16 DIS_CURR is set.

When the current is re-enabled the system is able to calculate the correct position and currents configuration taking into account the movement performed during disable. STATUS_WORD bit 11 AX_CALIBRATED will be restored.

If an alarm is present the commands DMC followed by EMC perform an alarm reset.

If an input is configured as 0: Disable Current, its status will prevail over EXE_FUN commands.

12.8. Disable/Enable Frequency (DFR/EFR) EXE_FUN = 18, 19

Disable Frequency command can be used to deselect certain drives when using a broadcast start command or when distributing the same Step Frequency Input to more drives. Motor movements are inhibited, but the current and hence the holding torque are still present.

During frequency disable the STATUS WORD bit 17 DIS_FREQ is set.

If an input is configured as 1: Disable Frequency, its status will prevail over EXE_FUN commands.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

12.9. Enter Programming Mode (PRG) EXE_FUN = 20

Enter Programming Mode command is used when it is necessary to re-program the firmware of the drive. FD will automatically reset. The pointer of the main program will jump to a dedicated flash memory block, where firmware re-programming code is installed.

In programming mode bit 18 of status word register is set. All the movements and motor current are disabled. Using Write File Record (21) it is possible to reprogram the firmware and the user data.

Two files numbers are supported:

- FD Firmware = 1,
- User Data = 2.

It is not necessary to set the drive in programming mode when programming file 2, User Data.

The first Write File Record request must begin with Record Number equal to zero. The next requests must have a sequential record numbers (1, 2, 3, ...).

Last valid record received

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13. MODBUS

13.1. Registers addresses

Modbus protocol addresses word registers (16-bits). Data is 32-bits type and it is then organized into two 16 bits registers (low and high words).
The same data are also mapped inside EtherCAT objects at indexes 0x2005 and 0x2006 and sub-indexes as per below table.

Register name	Modbus register address[1]		EtherCAT Index: subindex [hex]
	H	L	
START	0	1	0x2005: 0x01
STOP	2	3	0x2005: 0x02
ACCELERATION	4	5	0x2005: 0x03
DECELERATION	6	7	0x2005: 0x04
CURR_SPEED	8	9	0x2005: 0x05
CURR_POSITION	10	11	0x2005: 0x06
CURR_CYCLE	12	13	0x2005: 0x07
CALIB_POSITION	14	15	0x2005: 0x08
POSITION_OFFSET	16	17	0x2005: 0x09
SEL_CYC_SEQ	18	19	0x2005: 0x0A
TARGET_VERSION	20	21	0x2005: 0x0B
IO_BITS	22	23	0x2005: 0x0C
CONFIG	24	25	0x2005: 0x0D
…			
EXE_FUN	28	29	0x2005: 0x0F
I_MAX	30	31	0x2005: 0x10
ERR_FAT	34	35	0x2005: 0x12
MB_BAUD_RATE	36	37	0x2005: 0x13
STATUS_WORD	38	39	0x2005: 0x14
CYC_0_TYPE	40	41	0x2005: 0x15
CYC_0_SPEED	42	43	0x2005: 0x16
CYC_0_DELTA_POS	44	45	0x2005: 0x17
CYC_0_DIRECTION	46	47	0x2005: 0x18
CYC_0_DELTA_STOP	48	49	0x2005: 0x19
CYC_n_TYPE	2 × (20 + 5 × n)	2 × (20 + 5 × n) + 1	0x2005: (0x1A + 5 × n)
CYC_n_SPEED	2 × (21 + 5 × n)	2 × (21 + 5 × n) + 1	0x2005: (0x1B + 5 × n)
CYC_n_DELTA_POS	2 × (22 + 5 × n)	2 × (22 + 5 × n) + 1	0x2005: (0x1C + 5 × n)
CYC_n_DIRECTION	2 × (23 + 5 × n)	2 × (23 + 5 × n) + 1	0x2005: (0x1D + 5 × n)
CYC_n_DELTA_STOP	2 × (24 + 5 × n)	2 × (24 + 5 × n) + 1	0x2005: (0x1E + 5 × n)
SEQ_0_DW_0	360	361	0x2005: 0xB5
SEQ_0_DW_1	362	363	0x2005: 0xB6
SEQ_0_DW_2	364	365	0x2005: 0xB7
SEQ_0_DW_3	366	367	0x2005: 0xB8
SEQ_0_DW_4	368	369	0x2005: 0xB9
SEQ_n_DW_0	2 × (180 + 5 × n)	2 × (180 + 5 × n) + 1	0x2005: (0xBA + 5 × n)
SEQ_n_DW_1	2 × (181 + 5 × n)	2 × (181 + 5 × n) + 1	0x2005: (0xBB + 5 × n)
SEQ_n_DW_2	2 × (182 + 5 × n)	2 × (182 + 5 × n) + 1	0x2005: (0xBC + 5 × n)
SEQ_n_DW_3	2 × (183 + 5 × n)	2 × (183 + 5 × n) + 1	0x2005: (0xBD + 5 × n)
SEQ_n_DW_4	2 × (184 + 5 × n)	2 × (184 + 5 × n) + 1	0x2005: (0xBE + 5 × n)
IN_3_CNF	460	461	0x2005: 0xE7
IN_4_CNF	462	463	0x2005: 0xE8
HOMING	464	465	0x2005: 0xE9
TS	466	467	0x2005: 0xEA

[1] Register addresses of the Modbus holding registers are calculated as per following example:
006B hex = 107 , + 40001 offset = input #40108

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Register name	Modbus register address[1]		EtherCAT Index: subindex [hex]
	H	L	
V_RESEARCH	468	469	0x2005: 0xEB
V_RELEASE	470	471	0x2005: 0xEC
POS_SW_LS_UP	472	473	0x2005: 0xED
POS_SW_LS_DW	474	475	0x2005: 0xEE
I_LIM	476	477	0x2005: 0xEF
KNUM_NOM	478	479	0x2005: 0xF0
TEMPERATURE	480	481	0x2005: 0xF1
ACCUMULATION_LIMIT	484	485	0x2005: 0xF3
ENC_REV	486	487	0x2005: 0xF4
ACCUMULATED_STEPS	488	489	0x2005: 0xF5
I_MIN	490	491	0x2005: 0xF6
ENC_POS	492	493	0x2005: 0xF7
ENC_SPEED	494	495	0x2005: 0xF8
ENC_LATCH_RIS	496	497	0x2005: 0xF9
PON_CALIB_LIM	498	499	0x2005: 0xFA
TIME_CONST	500	501	0x2005: 0xFB
START_STOP_FREQ	502	503	0x2005: 0xFC
TEMP_OFF	504	505	0x2005: 0xFD
IN_1_CNF	506	507	0x2005: 0xFE
IN_2_CNF	508	509	0x2005: 0xFF
OUT_1_CNF	510	511	0x2006: 0x01
OUT_2_CNF	512	513	0x2006: 0x02
RESOLUTION_NUM	514	515	0x2006: 0x03
RESOLUTION_DEN	516	517	0x2006: 0x04
I_MIS	518	519	0x2006: 0x05
ECAT_ADDRESS	520	521	0x2006: 0x06
…			
CNT_CYC	524	525	0x2006: 0x08
CNT_DSTOP	526	527	0x2006: 0x09
ENC_LATCH_FAL	528	529	0x2006: 0x0A
ERR_LOG	530	531	0x2006: 0x0B
FRAME_PERIOD_AV	532	533	0x2006: 0x0C
FRAME_PERIOD_VAR	534	535	0x2006: 0x0D
IN_5_CNF	536	537	0x2006: 0x0E
…			
ENC_STATUS	540	541	0x2006: 0x10
IN_6_CNF	542	543	0x2006: 0x11
IN_7_CNF	544	545	0x2006: 0x12
OUT_3_CNF	546	547	0x2006: 0x13
…			
MAIN_PERIOD	550	551	0x2006: 0x15
V_DC	552	553	0x2006: 0x16
SM0_0	554	555	0x2006: 0x17
SM0_1	556	557	0x2006: 0x18
SM1_0	558	559	0x2006: 0x19
SM1_1	560	561	0x2006: 0x1A
SM2_0	562	563	0x2006: 0x1B
SM2_1	564	565	0x2006: 0x1C
SM3_0	566	567	0x2006: 0x1D
SM3_1	568	569	0x2006: 0x1E
AL_STATUS	570	571	0x2006: 0x1F

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

Register name	Modbus register address[1]		EtherCAT Index: subindex [hex]
	H	L	
ESCREG_CMD_ADD	576	577	0x2006: 0x20
ESCREG_DATA_0	578	579	0x2006: 0x21
ESCREG_DATA_1	580	581	0x2006: 0x22
APPLICATION_VERSION	582	583	0x2006: 0x23
SHIFT_SM2_DC	548	585	0x2006: 0x24

Tab. 8 – Registers addresses

13.2. Registers Description

13.2.1. Start, Stop

START and STOP are two Read/Write registers normally equal to zero. Setting the START register, the motor will run using selected Cycle data. Setting the STOP register, the motor will decelerate till stopping.

Once START and STOP registers are processed, they are automatically reset by software.

If the START is set again during the running cycle, at the end of the movement the selected cycle starts. This is valid only for single cycles.

13.2.2. Acceleration, Deceleration

ACCELERATION and DECELERATION are two Read/Write registers. These values are expressed as thousands µ-steps per second square.

Calculation example using linear ramp:
Requested speed: v_{rpm}= 300 [r.p.m.]
Requested acceleration time: Δt = 50 [msec]

$$v_{\mu step/sec} = 300 \cdot \frac{12'800}{60} = 64'000 \left[\frac{step}{sec}\right],$$

$$a_{\mu step/sec^2} = \frac{v_{\mu step/sec}}{\Delta t} = \frac{64'000}{0.05} = 1'280'000 \left[\frac{step}{sec^2}\right],$$

$$ACCELERATION = \frac{a_{\mu step/sec^2}}{1'000} = 1'280 \left[\frac{1'000 \cdot step}{sec^2}\right].$$

Calculation example using parabolic ramp:
Requested speed: v_{rpm}= 300 [r.p.m.]
Requested acceleration time: Δt = 50 [msec]

$$v_{\mu step/sec} = 300 \cdot \frac{12'800}{60} = 64'000 \left[\frac{step}{sec}\right],$$

$$a_{\mu step/sec^2} = 2 \cdot \frac{v_{\mu step/sec}}{\Delta t} = \frac{128'000}{0.05} = 2'560'000 \left[\frac{step}{sec^2}\right],$$

$$ACCELERATION = \frac{a_{\mu step/sec^2}}{1'000} = 2'560 \left[\frac{1'000 \cdot step}{sec^2}\right].$$

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.3. Current Speed, Position, Cycle

CURR_SPEED and CURR_CYCLE are Read Only registers, while CURR_POSITION is a Read/Write register. These registers are updated every millisecond.
During sequences CURR_CYCLE indicates the cycle number that is currently running in the sequence.

13.2.4. Calibration Position, Position Offset

CALIB_POSITION and POSITION_OFFSET are two Read/Write registers used in calibration cycles.

13.2.5. Select Cycle Sequence

SEL_CYC_SEQ is a Read/Write register. It has to be set from 0 to 31 to select the single cycle 0-31 and from 32 to 41 to select the sequence 0-9. Ref. to 6.1.

13.2.6. Target Version

TARGET_VERSION is a Read Only register, which indicates in the low word the Vx.xx firmware version and in the high word the hardware FDx.xx version.

13.2.7. I/O Bits

IO_BITS is a Read Only register used to read the status of the I/O port.

Bit number	FD1	FD2	Description
0	IN_2	IN_1_2	Ref. to 8.
1	IN_3	IN_3_4	
2	IN_4	IN_5	
3	IN_5	IN_6	
4	OUT_6	IN_7	
5	OUT_7	IN_8	
6		OUT_9	
7		OUT_10	
16		DIP sw 1.1	
17		DIP sw 1.2	
18		DIP sw 1.3	
19		DIP sw 1.4	
20		DIP sw 1.5	
21		DIP sw 1.6	
22		DIP sw 1.7	

Tab. 9 – I/O bits register

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.8. Configuration

CONFIG is a Read / Write register used to configure driver features.

Bit number	Name	Description
2	ENCODER_ENABLE	It enables encoder functionalities.
4	SW_LS_UP_ENABLE	Software limit switch at increasing positions.
5	SW_LS_DW_ENABLE	Software limit switch at decreasing positions.
6	INVERT_DIR	0: normal direction: e.g. ROL = counter-clockwise towards decreasing positions, 1: inverted direction: e.g. ROL = clockwise towards decreasing positions.
7-8	RAMP	0: linear, 1: parabolic, 2: s-curve.
9	START_STOP_FREQ	0: automatic, 1: start stop frequency enabled.
10	POS_UPLOAD	Ref. to 12.
11	EN_BOOST	It increases the motor current during movement starts (ref. to 8)
12	DIS_TORQUE_CONTROL	0: Torque control and step accumulation enabled 1: Torque control and step accumulation disabled. Just step loss detection and current reduction are enabled.
13	SELECT_32_CYC	0: Inputs selects 10 sequences 1: Inputs selects 32 cycles

Tab. 10 – Configuration register

13.2.9. RS-485 Address, Delay

485_ADDRESS is a Read/Write register used to set the slave address into Modbus network.

FD1 address is a programmable. Being a R/W register, the value can be modified in RAM writing a new value.
FD2 address is selected via DIP switches.

485_DELAY is a Read/Write register used to configure the target answer delay after host transmission. It is used when the host needs time to invert the line driver in high impedance. 0 value is suggested for higher speed communication.

Note:
FD2 DIP switches are read during functioning, so that Modbus address can be modified at any moment changing the switches configuration. CANopen address instead, even if it is taken from the same DIP switches, requires modification of the reception CAN filter via specific protocol commands. For this reason DIP are read at power on and modifiable only via CANopen commands.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.10. Execute Function

EXE_FUN is a Read / Write register to launch cycle 0 commands. Write following values to activate the specific function.

Value	Function	Description
1	ROR	Rotate towards increasing positions
2	ROL	Rotate towards decreasing positions
3	MST	Motor stop
10	MVP_ABS	Move versus position absolute
11	MVP_REL	Move versus position relative
13	RFS	Reference search (calibration)
14	RST	Reset
15	ALR	Alarm reset
16	DMC	Disable motor current
17	EMC	Enable motor current
18	DFR	Disable frequency
19	EFR	Enable frequency
20	PRG	Programming mode
31	ROR_NO_RST	Rotate towards increasing positions – w/o system reset
32	ROL_NO_RST	Rotate towards decreasing positions – w/o system reset

Tab. 11 – Execute function register

13.2.11. Maximum, Minimum and Limit Currents

I_MAX is a Read/Write register used to set the maximum sine wave peak current per phase expressed in milliamps. A protection has been implemented in order to prevent damages to the drive and motor in case of improper settings. I_MAX value is upper limited to I_LIM, a Read Only register, which cannot be modified.

I_MIN is a Read/Write register used to set the minimum sine wave peak current per phase. I_MIN represents the amount of current that pass through the windings if no movement is. A low I_MIN value reduces power dissipation and temperature.

I_MIS is a Read Only register, which express the actual motor current sine wave peak. To convert this value in Ampere it is necessary to scale it using the following formula:

$$\text{Motor phase peak current} = \frac{\sqrt{\text{I\_MIS}}}{136.5}$$

13.2.12. Fatal Error

ERR_FAT is a Read Only register used to read the status of the alarms. Ref. to 10.

13.2.13. Modbus Baud Rate

MB_BAUD_RATE is a Read/Write register used to configure the RS-232 and RS-485 baud rates. It ranges from 4˙800 up to 115˙200 bps for RS-232 and from 4˙800 up to 921˙600 bps for RS-485.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.14. Status Word

STATUS_WORD is a Read Only register which shows the driver operating conditions.

Bit number	Name	Description
0	DRIVER_READY_0	Drive initialization succeeded.
1	DRIVER_READY_1	Current and frequency are enabled.
2	ALARM_ACTIVE	Ref. to 10.
3	MODE_JOG	The drive is executing a jog cycle.
4	TORQUE_LIMIT	The drive is following the position actual value.
6	TARGET_POS	The motor is arrived at the correct destination position.
7	CYCLE_RUNNING	The motor is running a cycle, including delay cycles
8	HOMING	The motor is performing a calibration cycle.
10	POS_UPLOADED	Exact power-off position have been successfully uploaded during power-on.
11	AX_CALIBRATED	The axis is calibrated. This bit is set after a calibration cycle.
12	SW_LS_UP_ACTIVE	Software Limit Switch Up is reached. The motor is not able to execute increasing position movements.
13	SW_LS_DW_ACTIVE	Software Limit Switch Down is reached. The motor is not able to execute decreasing position movements.
14	HW_LS_UP_ACTIVE	Hardware Limit Switch Up is reached. The motor is not able to execute increasing position movement.
15	HW_LS_DW_ACTIVE	Hardware Limit Switch Down is reached. The motor is not able to execute decreasing position movement.
16	DIS_CURR	Current is disabled.
17	DIS_FREQ	Frequency is disabled.
18	PRG_MODE	Drive is in programming mode.
19	POWER_ON_POS_OK	The deviation of the power on absolute position is inside the POWER_ON_CALIBRATION_LIMIT register. The multi-turn position, AX_CALIBRATED and the current configuration is restored.
20	DELTA_STOP_OK	Delta stop gets active during DS cycle.

Tab. 12 – Status Word register

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.15. Cycles Data

Cycle registers are Read/Write registers which are used to define the motion parameters to be selected and started/stopped using registers or digital inputs.

Values	Registers	Description
[1 – 7]	CYC_n_TYPE	1: Jog, 2: Indexer relative, 3: Calibration, 4: Delta stop, 5: Indexer absolute, 6: Position delay, 7: Time delay
[1 – 300 000]	CYC_n_SPEED	Speed expressed in µstep per second
[0 – 2^{32}-1] or [-2^{31} – 2^{31}-1]	CYC_n_DELTA_POS	Jog: not used Indexer relative: steps to be executed (unsigned 32 bits) Calibration: not used Delta stop: maximum steps to be executed (unsigned 32 bits). Indexer absolute: destination position (signed 32 bits). Position delay: steps to be delayed. Time delay: milliseconds delayed.
[0, 1]	CYC_n_DIRECTION	When bit 6 INVERT_DIR of CONFIG register is zero: 0: CW rotation towards increasing positions 1: CCW rotation towards decreasing positions
[0 – 2^{32}-1]	CYC_n_DELTA_STOP	Delta stop: steps executed after delta stop input activation (unsigned 32 bits)

Tab. 13 – Cycle n register

13.2.16. Sequence

Sequence registers are Read/Write registers which can be used to define 10 sequences made of up to 20 cycles or commands (Stop / Loop) each. Each sequence is described by 5 Double Words (32 bits) sequence parameters. The ordered sequence of cycles and commands are addressed in Bytes inside the 5 DW's. The lowest addressable Byte into the sequence identifies the first cycle to be executed.

Values	Name	Description
[0-255][0-255][0-255][0-255]	SEQ_n_DW_0	Sequence parameter 0, 1, 2, 3
[0-255][0-255][0-255][0-255]	SEQ_n_DW_1	Sequence parameter 4, 5, 6, 7
[0-255][0-255][0-255][0-255]	SEQ_n_DW_2	Sequence parameter 8, 9, 10, 11
[0-255][0-255][0-255][0-255]	SEQ_n_DW_3	Sequence parameter 12, 13, 14, 15
[0-255][0-255][0-255][0-255]	SEQ_n_DW_4	Sequence parameter 16, 17, 18, 19

Tab. 14 – Seq n register

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.17. Start, Stop and Input Frequency Configuration

Start and step inputs configuration registers:

- FD1 IN_2_CNF
- FD2 IN_1_2_CNF

Stop and direction inputs configuration registers:

- FD1 IN_4_CNF,
- FD2 IN_3_4_CNF

are Read / Write registers, which can be configured as follows.

Value	Function	Description
0	STEP	It is used in STEP / DIR mode. Every pulse produces one µstep.
1	START_NO	To start the selected cycle or sequence.
2	START_NC	
6	QUAD_STEP_A	It is used in quadrature input frequency mode. Every edge produces one µstep.
7	START_NO_STOP_NC	To start the selected cycle or sequence and to stop the running one.
8	START_NC_STOP_NO	

Tab. 15 – IN2 and IN1/2 configuration register

Value	Function	Description
0	DIR	It is used in STEP / DIR mode. 0: CW towards increasing positions 1: CCW towards decreasing positions
1	STOP NO	To stop the running cycle or sequence.
2	STOP NC	
3	SEL_CYC_SEQ	Ref. to 6.1
4	FD1: SEL_CYC_SEQ_HOMIN_NO	Only on FD1 IN_4_CNF. When no calibration cycle is running IN4 behaves as cycle or sequence selection, while, when the calibration cycle is running, IN4 works as homing sensor input.
5	FD1: SEL_CYC_SEQ_HOMIN_NC	
6	FD2: QUAD_STEP_B	Only on FD2 IN_3_4_CNF. It is used in quadrature input frequency mode. Every edge produces one µstep.

Tab. 16 – IN4 and IN3/4 configuration register

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.18. Multipurpose Inputs Configuration

Multipurpose inputs configuration registers:

- FD1 IN_3_CNF and IN_5_CNF,
- FD2 IN_5_CNF and IN_6_CNF,

are Read / Write registers, which can be set as follows.

Value	Function	Description
0	CURRENT_DISABLE	This input disables the motor current, freeing the shaft. In this situation if the shaft is manually moved beyond synchronous mode limit (± 1.8°) the red led will be steady on. If further moved beyond the step accumulation limit the related alarm arises. The re-activation of the current will reset the alarms.
1	FREQUENCY_DISABLE	It disables all the movement commands. The motor remains in torque, i.e. current keeps on flowing in the motor windings, but it does not move and the position counter does not change. This input can be used as a selection input, when the same frequency signal is multiplexed to several drive.
2	HOMING_NO	Homing Switch normally open / close.
3	HOMING_NC	
5	HW_LIMIT_SWITCH_UP_NO	Hardware Limit Switch up / down normal open / close. When it is active all the movements towards increasing or decreasing positions are inhibited.
6	HW_LIMIT_SWITCH_UP_NC	
7	HW_LIMIT_SWITCH_DW_NO	
8	HW_LIMIT_SWITCH_DW_NC	
9	ENC_LATCH	It latches the multi-turn encoder position sampled at 1 kHz. Higher frequencies are available if required. Latched value is saved into ENC_LATCH_RIS and ENC_LATCH_FAL registers, depending on the input signal edge. Note: it keeps the resolution of the encoder: 4096 steps per revolution.
10	DIR	It is used in STEP / DIR mode. 0: CW towards increasing positions, 1: CCW towards decreasing positions.
11	SEL_CYC_SEQ	Ref. to 6.1.
12	DELTA_STOP_NO	Ref. to 6.5.
13	DELTA_STOP_NC	
14	FD1: QUAD_STEP_B	Only on FD1 IN_5_CNF. It is used in quadrature input frequency mode. Every edge produces one µstep.
15	START NO	To start and stop the selected cycle or sequence.
16	START NC	
17	STOP NO	
18	STOP NC	

Tab. 17 – IN3/5 configuration register

13.2.19. Aux Inputs

FD2 is equipped with two additional inputs IN_7_CNF and IN_8_CNF. They are Read / Write registers, which can be configured as follows.

Value	Function	Description
11	SEL_CYC_SEQ	Ref. to 6.1.

Tab. 18 – Aux inputs configuration

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.20. Output

Output configuration register:
- FD1 OUT_7_CNF
- FD2 OUT_10_CNF

are Read / Write registers, which can be configured as follows.

Value	Function	Description
0	CYCLE_RUNNING	Active on running cycles or sequences.
1	ENC_INDEX_50	It produces one pulse of 50msec at POSITION_OFFSET angular position.
2	ENC_INDEX_Π	It is high at half revolution starting from POSITION_OFFSET angular position.
3	POS_UPLOADED	Exact power-off position has been restored.
4	AX_CALIB	Axle is calibrated.
5	POWER_ON_POS_OK	Position is restored compensated with the power-off position deviation
6	TORQUE_LIM	Drive is following the ordered position, out of synchronism.

Tab. 19 – OUT7 and OUT10 configuration register

13.2.21. Analogic Input

FD2 is equipped with analogic input. At the moment only one function can be associated to this input, but, if requested, all the registers can be scaled with AI and other functions can be implemented with it. Value of the analogic input can be monitored from the register IN_VAL_A1 with a 12 bit resolution. Speed scaling can be activated setting IN_ A1_CNF as per table below.

Value	Configuration	Description
1	Speed scaled from AI	The values defined in every cycles defines the maximum associated speed. Real motor speed is this maximum value scaled with the analogic input voltage.

Tab. 20 – FD2 AI configuration registers

13.2.22. Homing, Time Stop, Release and Research Speeds

HOMING, TS, V_RESEARCH, V_RELEASE are Read/Write registers. These registers are used to setup the calibration movement to be launched with a calibration cycle.
TS (Time Stop) indicates the waiting time between V_RESEARCH and V_RELEASE speeds (it is expressed in milliseconds).
CALIB_POSITION is loaded into the position register and STATUS_WORD bit 11 AX_CALIB is set at the successful conclusion of all the homing cycles.

VALUE	NAME	DESCRIPTION
0	MARKER	Ref. to 6.4.
1	HOME_SIMPLE	
2	HOME_TS_INV	
3	HOME_TS_NO_INV	
4	HOME_TS_INV_MRK	
5	HOME_TS_NO_INV_MRK	

Tab. 21 – Homing register

13.2.23. Position Software Limit Switch Up/Down

POS_SW_LS_UP and POS_SW_LS_DW are Read/Write registers that can be used to setup the software limit switches.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

They need to be enabled by setting CONFIG register bits 4, 5.

13.2.24. Temperature

TEMP is a Read Only register that shows the microcontroller temperature in Celsius degrees. When the temperature rises above 100 °C, the temperature alarm is given.

13.2.25. Temperature Offset

TEMP_OFF is a Read Only register used to calibrate the microcontroller temperature sensor. It cannot be modified.

13.2.26. K

KNUM_NOM and KV are Read Only registers used for setting up the current loop control. They cannot be modified.

13.2.27. Steps Accumulation Limit

ACCUMULATION_LIMIT is a Read/Write register used for setting the step accumulation alarm limit. This register is upper limited to 128˙000. If the accumulated steps exceed the ACCUMULATION_LIMIT value the motor stops, the ERR_FAT will be set to 1 and the red LED will be steady on.

13.2.28. Encoder Position, Speed, Latch and Revolution

ENC_POS and ENC_SPEED are two Read Only registers that show the encoder position in µsteps and the encoder speed in µsteps per second.
ENC_LATCH_RIS and ENC_LATCH_FAL are a Read Write registers used in combination with the multipurpose inputs encoder latch setting. When one of this input is configured as encoder latch and it gets active, the multi-turn encoder position is saved into ENC_LATCH_RIS register, while when the input gets inactive it is saved into ENC_LATCH_FAL register.
ENC_REV is a Read Only register that shows the encoder position inside the revolution (it range from 0 to 4095).

13.2.29. Power On Calibration Limit

PON_CALIB_LIM is a Read/Write register used for power on position restore. Ref. to 12.

13.2.30. Time Constant

TIME_CONST is a Read/Wirte register that shall not be modified.

13.2.31. Start Stop Frequency

START_STOP_FREQ is a Read/Write register used for setting the start stop frequency. It is expressed in µsteps per second.

13.2.32. Resolution

RESOLUTION_NUM and RESOLUTION_DEN are two Read-Only registers, which can be modified only during programming. They are unsigned 16 bits each. Their ratio shall be in the range [8 – 4 096], allowing a resolution in the range [400 – 204 800] µstep per revolution. Following formula to be used:

$$\text{Resolution}=50\cdot\frac{\text{RESOLUTION}_{\text{NUM}}}{\text{RESOLUTION}_{\text{DEN}}},$$

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

expressed in µstep per revolution.

13.2.33. CANopen Baud Rate, Address and Status

CANOPEN_BAUDRATE and CANOPEN_ADDRESS are two Read/Write registers.
CANOPEN_BAUDRATE need to be set using integer numbers from 0 to 8 that correspond to the following frequencies:

0: 1000 KHz, 1: 800 KHz, 2: 500 KHz, 3: 250 KHz, 4: 125 KHz, 5: 100 KHz, 6: 50 KHz, 7: 20 KHz, 8: 10 KHz.

FD1 address is programmable.
FD2 address is selected via DIP switches.

After modifying above values it is necessary to give a NMT command: Reset node or Reset communication to make the modification effective.

CANOPEN_STATUS is a Read-Only register, which is used to monitor the CAN peripheral status.

Bit number	Name	Description
[0 – 7]	TEC	Transmit error counter. The implementing part of the fault confinement mechanism of the CAN protocol.
[8 – 15]	REC	Receive error counter. In case of an error during reception, this counter is incremented by 1 or by 8 depending on the error condition as defined by the CAN standard. After every successful reception the counter is decremented by 1 or reset to 120 if its value was higher than 128. When the counter value exceeds 127, the CAN controller enters the error passive state.
16	BUS_OFF	Bus off, i.e. TEC greater than 255. Once this condition is reached, the bus off state is left automatically by hardware once 128 occurrences of 11 recessive bits have been monitored.
[17 – 19]	LAST_ERROR	0: No Error 1: Stuff Error 2: Form Error 3: Acknowledgment Error 4: Bit recessive Error 5: Bit dominant Error 6: CRC Error
20	ACK_INIT	CAN hardware cannot synchronize (monitor a sequence of 11 consecutive recessive bits on the CAN RX signal)
21	RX_SIGNAL	0: recessive 1: dominant
22	TX_MODE	CAN hardware is in transmission
23	RX_MODE	CAN hardware is in reception
24	ARBITRATION_LOST	Previous TX failed due to an arbitration lost
25	TX_ERROR	Previous TX failed due to an error
26	RX_FIFO_0_OVERRUN	This bit is set by hardware when a new message cannot be received because the FIFO 0 or FIFO 1 were full.
27	RX_FIFO_1_OVERRUN	

Tab. 22 – CANopen error and status

Note:
FD2 DIP switches are read during functioning, so that Modbus address can be modified at any moment changing the switches configuration. CANopen address instead, even if it is taken from the same DIP switches, requires modification of the reception CAN filter via specific protocol commands. For this reason DIP are read at power on and modifiable only via CANopen commands.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

13.2.34. Cycles Counters

CNT_CYC and CNT_DSTOP are two Read/Write registers used to check the correct execution of ordered command.

Every time an indexer movement is completed (relative, absolute or delta stop) V8 compares the arrival position with the wanted destination to make sure the target is reached. If they are identical CNT_CYC register is incremented. If the cycle is a delta stop cycle and the delta stop input activation has been the reason of motor stopping, also CNT_DSTOP register is incremented.

13.2.35. Encoder Status

ENC_STATUS is a Read/Write register used to monitor the encoder status.

BIT NUMBER	NAME	ON STATE DESCRIPTION
0	Parity Error	0: Ok, 1: parity error in the reading of absolute position.
1	Magnetic Decrement	0: Ok, 1: decrement of the magnetic field of the encoder sensor.
2	Magnetic Increment	0: Ok, 1: increment of the magnetic field of the encoder sensor.
3	Linearity Error	0: Ok, 1: encoder reading linearity error.
4	CORDIC Overflow	0: Ok, 1: Overflow of the Coordinate Rotation Digital Computer (CORDIC) that calculates the angle and the intensity of the rotating field.
5	Offset Compensation	0: Ok, 1: Offset compensation not finished.
6	50 µsec Loop Not Running	0: Ok, 1: 50 µsec current control loop has entered with a delay longer of 250 µsec.
7	Current Disabled	0: Ok, 1: MOSFET gate drivers are disabled.

Tab. 23 – Encoder Status

13.2.36. Supply voltage

V_DC is a Read-Only register, which shows the DC bus voltage applied to the drive, expressed with a resolution of 10 mV.

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

14. MODBUS PROTOCOL

Modbus protocol defines the format and the modality of communication between a master and one or more slaves that answer to master's requests.

V8 implements Modbus protocol as a slave with:

- RTU mode,
- 8 data bits,
- 1 stop bit,
- no parity bit.

The Modbus message is composed of:

- Device address (from 1 up to 247)
- Function code: V8 implements Read Holding Registers (03), Write Single Holding Register (06), Write Multiple Register (16), Write File Record (21).
- Data
- Error analysis (CRC16 algorithm)

If an error is present (format error or CRC16 error), the message is considered not valid and discarded and the slave will produce a 5 Bytes error message. In case of CRC error, no answer will take place.

Use Address 0 to send broadcast messages (no drive will answer).

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

14.1. Read Holding Register (03)

This function allows reading the values of 16 bits registers. V8 implements up to 125 registers reading at a time.

14.1.1. Master Read Request

In order to read the current position, speed and cycle of the driver number 13 the following read command have to be performed.

Slave address	Function code	Address		Quantity of words		CRC16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
13	03	0	8	0	6	68	198

Tab. 24 – CURR_POS, CURR_SPEED and CURR_CYCLE read request

14.1.2. Slave Read Answer

For example if:

- current speed is 30˙000 (corresponding to words: 0 H and 30˙000 L)
- current position is 230˙113 steps (corresponding to words: 3 H and 33˙505 L),
- current cycle is 29

the slave answers would be:

Slave Add	Function Code	Quantity of Bytes	Current Position [32 bits]				Current Speed [32 bits]				Current Cycle [32 bits]				CRC 16	
1 B	1 B	1 B	4 Bytes				4 Bytes				4 Bytes				2 Bytes	
13	3	12	0	0	117	48	0	3	130	225	0	0	0	29	136	114

Tab. 25 – CURR_SPEED, CURR_POS, CURR_CYCLE read answer

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

14.2. Write Single Holding Register (06)

This function allows to write the value of one 16 bits register.

14.2.1. Master Write Request

In order to write 100˙000 to Cycle 0 Speed register (32 bits) (corresponding to words: 1 H and 34˙464 L) of the drive number 13 the two following message (High and Low parts of the double word CYC_0_SPEED) have to be generated.

Slave address	Function code	Address		Data		CRC16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
13	06	0	42	0	1	105	14

Tab. 26 – CYC_0_SPEED_H write request

Slave address	Function code	Address		Data		CRC16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
13	06	0	43	134	160	155	22

Tab. 27 – CYC_0_SPEED_L write request

14.2.2. Slave Write Answer

The slave write answer consists of the re-transmission of the received message.

Slave address	Function code	Address		Data		CRC16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
13	06	0	42	0	1	105	14

Tab. 28 – CYC_0_SPEED_H write answer

Slave address	Function code	Address		Data		CRC16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
13	06	0	43	134	160	155	22

Tab. 29 – CYC_0_SPEED_L write answer

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

14.3. Write Multiple Holding Register (16)

This function allows to write the values of multiple 16 bits registers in the same message. V8 implements up to 123 registers writing at a time.

14.3.1. Master Write Request

In order to write the Cycle 0 Type, Speed, Position, Direction and Delta Stop registers:

- Type equal to Indexer (corresponding to words: 0 H and 2 L)
- Speed equal to 100˙000 (corresponding to words: 1 H and 34˙464 L)
- Position equal to 270˙000 (corresponding to words: 4 H and 7˙856 L)
- Direction equal to 1 (corresponding to words: 0 H and 1 L)
- Delta Stop equal to 1˙000 (corresponding to words: 0 H and 1˙000 L)

of the driver number 13 the following message have to be generated.

Slave Add	Function	Register Address		Words Count		Byte Count	Type [32 bits]				Speed [32 bits]				Position [32 bits]				Direction [32 bits]				Delta Stop [32 bits]				CRC16	
1B	1B	2 B		2 B		1B	4 Bytes				4 Bytes				4 Bytes				4 Bytes				4 Bytes				2 B	
13	16	0	40	0	10	20	0	0	0	2	0	1	134	160	0	4	30	176	0	0	0	1	0	0	3	232	21	205

Tab. 30 – CYC_0_DATA write request

14.3.2. Slave Write Answer

Slave Add	Function	Register Address		Word Count		CRC 16	
1 Byte	1 Byte	2 Bytes		2 Bytes		2 Bytes	
13	16	0	40	0	10	192	202

Tab. 31 – CYC_0_DATA write answer

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

14.4. Write File Record (21)

This function code is used to perform a file record write.

Request Length is limited to 251, as the Modbus message cannot be longer than 256 Bytes.

Request Type is always equal to 6.

Two files numbers are supported:

- FD Firmware = 1,
- User Data = 2.

Before re-programming the FD Firmware, it is necessary to set the drive in programming mode.
This is performed by setting the EXE_FUN register equal to 20. Consequently, bit 18 of status word register will be set. It is not necessary to set the drive in programming mode when programming file 2, User Data.

A file is an organization of records. The first Write File Record request must begin with Record Number equal to zero. The next requests must have a sequential record numbers (1, 2, 3, ...).

All Request Lengths are provided in terms of number of Bytes and all Record Lengths are provided in terms of the number of 16-bit words.

At the end of the file, when the programming is complete, it is necessary to write the RST command into EXE_FUN register.

14.4.1. Master Write File Request

Slave Add	Function	Req Length	Req Type	File Number		Record Number		Record Length		Record Data												CRC16	
1B	1B	1B	1B	2 B		2B		2B		12B												2 B	
13	21	19	6	0	1	0	0	0	6	35	96	32	0	33	69	8	0	115	237	8	0	119	209

Tab. 32 – Master write file request

14.4.2. Slave Write File Answer

The normal response is an echo of the request.

Slave Add	Function	Req Length	Req Type	File Number		Record Number		Record Length		Record Data												CRC16	
1B	1B	1B	1B	2 B		2B		2B		12B												2 B	
13	21	19	6	0	1	0	0	0	6	35	96	32	0	33	69	8	0	115	237	8	0	119	209

Tab. 33 – Slave write file answer

e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

14.5. Checksum Calculation

```
//CRC Tables
const u8 auchCRCHi_exp[] = {0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1,0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80,
0x41, 0x00, 0xc1, 0x81, 0x40, 0x00,0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80,
0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1, 0x81,
0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80,
0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80,
0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x00, 0xc1, 0x81,
0x40, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81, 0x40, 0x01, 0xc0, 0x80, 0x41, 0x01, 0xc0, 0x80, 0x41, 0x00, 0xc1, 0x81,
0x40};

const u8 auchCRCLo_exp[] = {0x00, 0xc0, 0xc1, 0x01, 0xc3, 0x03, 0x02, 0xc2, 0xc6, 0x06, 0x07, 0xc7, 0x05, 0xc5, 0xc4,
0x04, 0xcc, 0x0c, 0x0d, 0xcd, 0x0f, 0xcf, 0xce, 0x0e, 0x0a, 0xca, 0xcb, 0x0b, 0xc9, 0x09, 0x08, 0xc8, 0xd8, 0x18, 0x19, 0xd9,
0x1b, 0xdb, 0xda, 0x1a, 0x1e, 0xde, 0xdf, 0x1f, 0xdd, 0x1d, 0x1c, 0xdc, 0x14, 0xd4, 0xd5, 0x15, 0xd7, 0x17, 0x16, 0xd6,
0xd2, 0x12, 0x13, 0xd3, 0x11, 0xd1, 0xd0, 0x10, 0xf0, 0x30, 0x31, 0xf1, 0x33, 0xf3, 0xf2, 0x32, 0x36, 0xf6, 0xf7, 0x37, 0xf5,
0x35, 0x34, 0xf4, 0x3c, 0xfc, 0xfd, 0x3d, 0xff, 0x3f, 0x3e, 0xfe, 0xfa, 0x3a, 0x3b, 0xfb, 0x39, 0xf9, 0xf8, 0x38, 0x28, 0xe8,
0xe9, 0x29, 0xeb, 0x2b, 0x2a, 0xea, 0xee, 0x2e, 0x2f, 0xef, 0x2d, 0xed, 0xec, 0x2c, 0xe4, 0x24, 0x25, 0xe5, 0x27, 0xe7,
0xe6, 0x26, 0x22, 0xe2, 0xe3, 0x23, 0xe1, 0x21, 0x20, 0xe0, 0xa0, 0x60, 0x61, 0xa1, 0x63, 0xa3, 0xa2, 0x62, 0x66, 0xa6,
0xa7, 0x67, 0xa5, 0x65, 0x64, 0xa4, 0x6c, 0xac, 0xad, 0x6d, 0xaf, 0x6f, 0x6e, 0xae, 0xaa, 0x6a, 0x6b, 0xab, 0x69, 0xa9, 0xa8,
0x68, 0x78, 0xb8, 0xb9, 0x79, 0xbb, 0x7b, 0x7a, 0xba, 0xbe, 0x7e, 0x7f, 0xbf, 0x7d, 0xbd, 0xbc, 0x7c, 0xb4, 0x74, 0x75,
0xb5, 0x77, 0xb7, 0xb6, 0x76, 0x72, 0xb2, 0xb3, 0x73, 0xb1, 0x71, 0x70, 0xb0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52,
0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9c, 0x5c, 0x5d, 0x9d, 0x5f, 0x9f, 0x9e, 0x5e, 0x5a, 0x9a, 0x9b,
0x5b, 0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4b, 0x8b, 0x8a, 0x4a, 0x4e, 0x8e, 0x8f, 0x4f,0x8d, 0x4d, 0x4c,
0x8c, 0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40};



 /*[FUNCTION      ]*****************************************************
  *    FUNCTION NAME: Crc16Modbus                          *
  *    DESCRIPTION: /                                *
  *    PARAMETER LIST: /                               *
  *    RETURNED VALUE: /                                *
  *    SIDE EFFECTS ON GLOBAL VAR.:                        *
  ***********************************************************************/
u16 Crc16Modbus(u8 *chkbufBase, u8 IxStart, u8 len)
{
u8      uchCRCHi=0xff;
u8      uchCRCLo=0xff;
u16     uIndex;
u16     temp_code;
u8      *lPtr;
u8      lIx;

        lIx = IxStart;

        while(len)
        {
                lPtr = &chkbufBase[lIx];
```

```
            uIndex=(u16)(uchCRCHi^*lPtr);

            uchCRCHi=(u8)(uchCRCLo^auchCRCHi_exp[uIndex]);
            uchCRCLo=auchCRCLo_exp[uIndex];

            llx++;
            len--;
        }

        temp_code=(u16)uchCRCHi;
        temp_code=(u16)(temp_code<<8);

        return(u16) (temp_code | uchCRCLo);
}
```


e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99