1. DESCRIPTION

The 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 without forgoing high performances.

Mounted on the rear of the motor in an integrated solution, FD1 and FD2 drivers exploits a 12-bits absolute magnetic encoder.

Latest V8 firmware replaces previous versions V3 and V5 integrating both main characteristics: the driver can have a fixed motor current independent from the load (only the current reduction when the motor is not moving of V3 functioning), or it can calculate the optimum amount of current measuring the resistant torque applied to the motor (V5 functioning). The torque control gives 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 the FD motion using I/O such as start/stop, step/dir or quadrature signals, Modbus via RS-232 and RS-485 and using the CANopen – CiA 402 (including interpolated position mode).

Up to 32 motion profiles, named cycles, can be pre-programmed into the driver, selected via I/O or fieldbus and started alone or in sequence.

2. FEATURES

- Multipurpose I/O

FD1	FD2
4 DI opto-coupled	6 DI opto-coupled
2 DO pnp	2 DO opto-coupled pnp
	1 AI

- 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. Configurable speed, acceleration, deceleration, target position with linear, parabolic and s-curve motion profiles. Relative and absolute movements.
Complex cycles as homing, delta stop and delay are also selectable and started.

- Position resolution

Configurable µsteps per revolution

- Absolute multi-turn position recovery at power on

- Interfaces

Modbus via RS-232 and RS-485
CANopen

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

- Step accumulator with programmable alarm limit

Fig. 1 – FD1.5AB-2231

3. 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

4. HARDWARE

V8 firmware exploits 40 kHz PWM modulation to control two sine wave currents of the dual-H-bridge power stage. FD-family drivers are provided with high performance Hall-effect current sensors and low on-resistance MOSFETs.

The period of current is divided into a configurable number of µsteps, allowing resolutions from 400 up to 204˙800 µsteps 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 V8.
The load torque is calculated using the 12-bit magnetic encoder acquisition. This value is continuously processed to determine the best amount of current which is 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 V8 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 V8 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.

In those applications characterized by high acceleration and inertial load traditional stepper drivers 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 V8 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). As soon as the resistant torque decreases the driver recovers the accumulated steps without position loss.

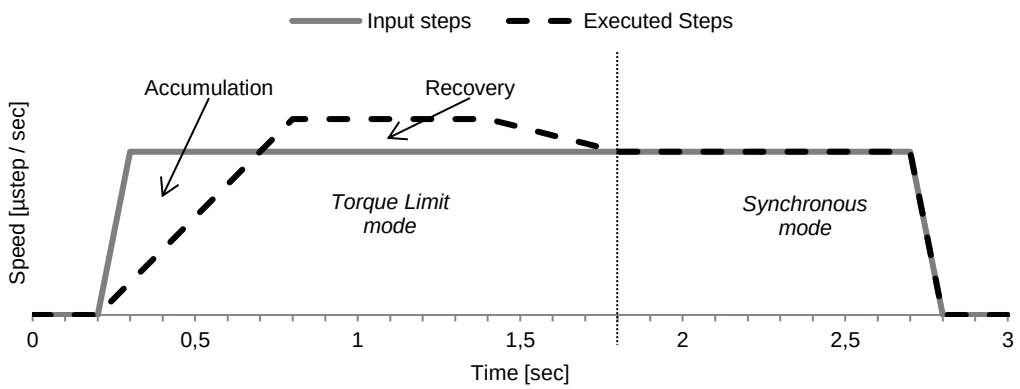


Fig. 2 – Speed profile during accumulation and recovery

If the accumulated steps exceed the configurable limit (named accumulation limit in Modbus, following error window in CANopen), the alarm Steps Accumulation Limit arises. During accumulation the red LED is steady lit and the related status words bits are set.

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

V8 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 input voltage	Integrated absolute encoder	Digital I/O	RS-232	RS-485	Analog I/O
FD1.4	20 – 36 V_{DC}	✓	✓	✓	✓	
FD1.1	20 – 80 V_{DC}	✓		✓	✓	
FD1.5		✓	✓	✓	✓	
FD1.6		✓	✓	✓	✓	
A = CANopen B = aluminum cover W = IP65 cover D = to supply the logic from V_{EXT} when V_{POW} is off (already included in FD1.6)						
FD2.1	24 – 130 V_{DC}	✓	✓	✓	✓	✓
FD2.2		✓	✓	✓		✓
A = CANopen (only applicable to FD2.1) D = to supply the logic from V_{EXT} when V_{POW} is off						

Tab. 1 – FD1 and FD2 drivers

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. DWLOADER

DwLoader is a PC application which allows the user to program the driver and test its functionality.

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.

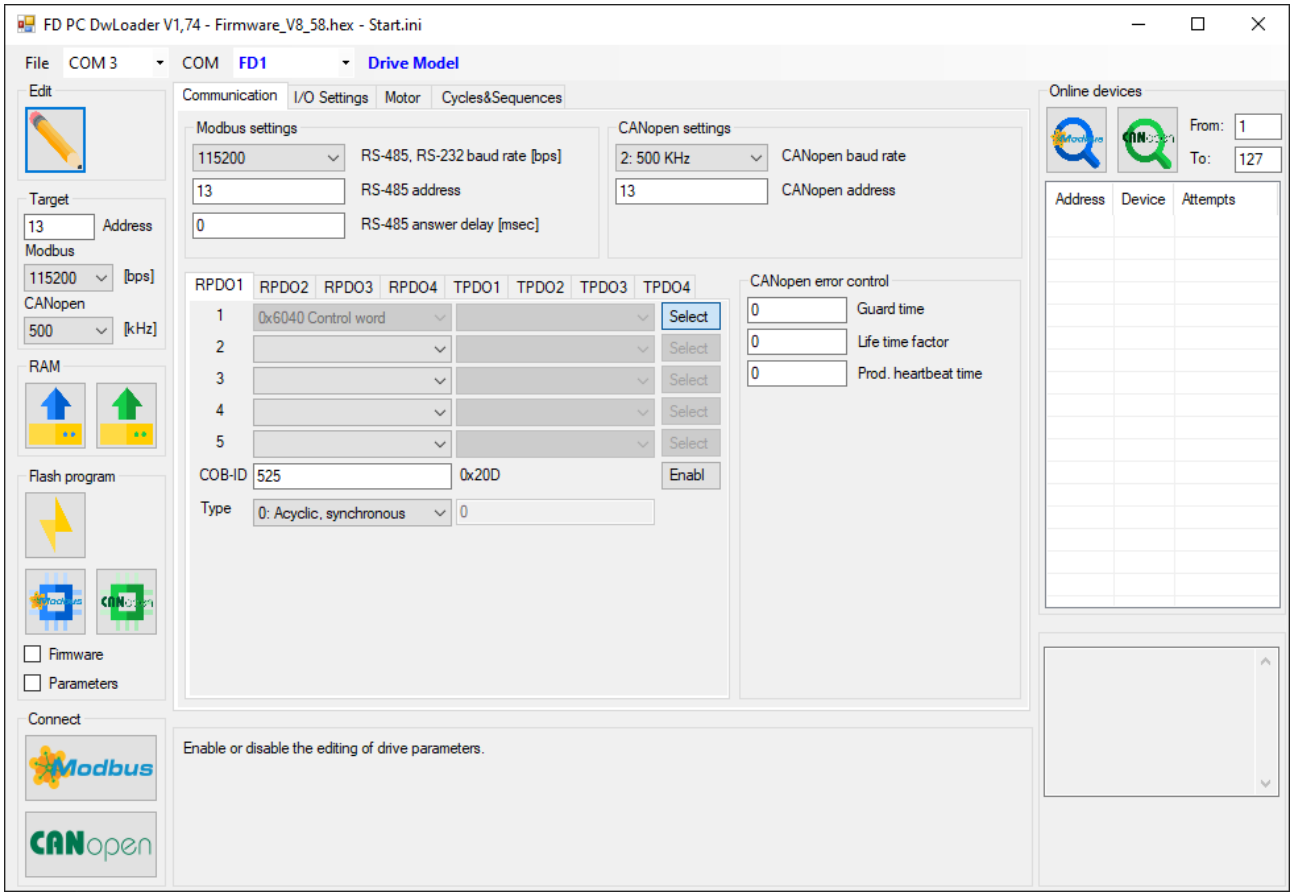


Fig. 3 – DwLoader main window

Note:
The COM port and the correct driver model FD1 or FD2 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).

CANopen DwLoader functionalities work with IXXAT VCI V4 drivers or Kvaser drivers, depending on the DwLoader type requested.

5.1. In-application programming

In-application programming (IAP) tool allows to re-program the FD firmware and the set of data using Modbus protocol via RS-485 or RS-232 (blue push-button) or CANopen protocol (green push-button). Ref. to 13.4 for Modbus protocol details and ref. to 14.2.2.4 for CANopen protocol details.

IAP allows to modify the field-bus parameters: baud rate and node address:

- Target text boxes identifies the parameters to be used during programming, those which are currently active in the device.
- Communication Modbus/CANopen baud rate and address are the new parameters, which will be programmed and will be active after reset.

Note:
FD1 driver has a programmable node address.

FD2 driver has a node address selectable via DIP switches.

Starting from version V8.37, the device bootloader has been modified to implement IAP through CANopen. If the drive is already programmed with an earlier version, before using IAP, it is necessary to update device bootloader using user flash program (ref. to 5.3).

Latest bootloader is version is V0.21.

5.2. RAM data upload/download

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

Note:
Data upload combined with In Application Programming are useful when it is needed to re-program a new driver with the same configuration of another existing.

Data upload in CANopen is available starting from firmware versions V8.37.

5.3. 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:
   a. FD1: place the flash programming jumper,
   b. FD2: turn-on the DIP switch 8.
3. Power on the driver (LEDs will be all off).
4. Click on “User Flash Program” button. The flash programming 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 not absorbing power).
6. Remove the programming mode.
7. Power on the driver again. Blinking green LED means that the programming procedure occurred properly.

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

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

5.4. 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.

5.5. Data modify

Click on the pencil symbol to modify the driver parameters.

Name	Measurement unit	Description
Communication		
RS-485, RS-232 baud rate	[bps]	Range: 4˙800 – 115˙200 bps for RS-232. Range: 4˙800 – 921˙600 bps for RS-485.
RS-485 address	[1 – 247]	Address 0 is reserved for broadcast messages in Modbus. FD1 address is programmable. FD2 address is selected via DIP switches.
RS-485 answer delay	[msec]	It is used when the master takes time to invert the line driver in order to receive the answer from the slave. Use 0 msec for faster communication.
CANopen baud rate	[0 – 8]	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.
CANopen address	[1 – 128]	FD1 address is programmable. FD2 address is selected via DIP switches.
PDO mapping		Select the objects to be mapped into the PDO (maximum 8 Bytes per PDO)
PDO COB-ID		Select the COB-ID and enable/disable Standard settings for RPDO 0x200 + Node-ID Standard settings for TPDO 0x180 + Node-ID
PDO Type		Select the transmission type: 0: Acyclic, synchronous (at every SYNC only if there is a difference) 1 – 240: synchronous (at every 1-240 SYNC) 254, 255: asynchronous (every time there is a difference in the PDO content)
TPDO Inhibit time	[0 – 65535]	It is used for asynchronous TPDO to inhibit TPDO transmission for a period of time. It is defined in multiples of 100 usec, i.e. a value of 50 inhibits the TPDO transmissions for 5 msec. 0 value means disabled.
TPDO Event timer	[0 – 65535]	It is used for asynchronous TPDO to force a TPDO transmission after a period of time is elapsed. It is defined

		in multiples of msec, i.e. a value of 50 force a TPDO transmission after 50 msec of last TPDO transmission. 0 value means disabled.
Guard time	[0 – 65535]	If the drive does not receive NMT guarding within Life time factor * guard time milli seconds, the motor is stopped and the drive reinitialize the communication. 0 value means disabled.
Life time factor	[0 – 255]	
Producer heartbeat time	[0 – 65535]	It defines the number of milliseconds between the transmissions of the drive heartbeat frame.

I/O settings for FD1		
IN2 Configuration		0: Step input frequency, 1: Start NO, 2: Start NC, 6: Quadrature input frequency A, 7: Start NO, Stop NC, 8: Start NC, Stop NO.
IN3 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 of step input frequency, 11: Cycle or sequence selection (bit 0), 12: Delta stop NO, 13: Delta Stop NC, 15: Start NO, 16: Start NC, 17: Stop NO, 18: Stop NC, 19: Enable current
IN4 Configuration		0: Direction of step input frequency, 1: Stop NO 2: Stop NC 3: Cycle or sequence selection (bit 2) 12: Homing NO, 13: Homing NC.
IN5 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 of step input frequency, 11: Cycle or sequence selection (bit 1), 12: Delta stop NO, 13: Delta stop NC,

		14: Quadrature input frequency B, 15: Start NO, 16: Start NC, 17: Stop NO, 18: Stop NC, 19: Enable current.
OUT6 Configuration		0: Drive ok, 1: Alarm, 2: Drive ok + not running, 3: Alarm + running.
OUT7 Configuration		0: Cycle running, 3: Position uploaded, 4: Axis calibrated, 5: Power on position ok, 6: Torque limit, 7: Cycle Running + /DStop.

I/O settings for FD2		
IN1/2 Configuration		0: Step input frequency, 1: Start NO, 2: Start NC, 6: Quadrature input frequency A, 7: Start NO, Stop NC, 8: Start NC, Stop NO.
IN3/4 Configuration		0: Direction of step input frequency, 1: Stop NO, 2: Stop NC, 3: Cycle or sequence selection, 6: Quadrature input frequency B, 12: Homing NO, 13: Homing NC.
IN5 Configuration IN6 Configuration		0: Disable Current, 1: Disable Frequency, 2: Homing NO, 3: Homing NC, 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 of step input frequency, 11: Cycle or sequence selection, 12: Delta stop NO, 13: Delta stop NC, 15: Start NO, 16: Start NC, 17: Stop NO, 18: Stop NC, 19: Enable current.
IN7 Configuration		11: Cycle or sequence selection.
IN8 Configuration		11: Cycle or sequence selection.
OUT9 Configuration		0: Drive ok, 1: Alarm, 2: Drive ok + not running, 3: Alarm + running.
OUT10 Configuration		0: Cycle running,

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.
Analogic IN Configuration		1: Speed set-point scaling

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). Ref. to 12
Software Limit Switch Up	[µstep]	To enable and set the software limit switch at increasing positions
Software Limit Switch Down	[µstep]	To enable and set the software limit switch at 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 slowly go back to the torque-controlled values.
Select 32 Cycles		Ref. to 6.1
Auto Full Step		When enabled the drive automatically modifies the current control method, resulting in an increment of motor's torque at medium speeds.
Maximum Motor Current	[mA]	Maximum peak value of the sine wave phase current, corresponding to maximum torque
Minimum Motor Current	[mA]	Minimum peak value of the sine wave phase current, corresponding to zero torque
Step Accumulator Limit	[µstep]	Maximum number of accumulated µsteps (upper limited at 10 revolutions)
Time Constant	[256 • 50 µsec]	Preset to 1. DO NOT MODIFY
Reference Resolution Num.		To configure the number of µsteps per revolution: $\mu step_{REV} = \frac{50 \cdot Res_{NUM}}{Res_{DEN}}$ Res_{NUM} and Res_{DEN} range from 0 to 65'535. The resolution is anyhow limited from 400 to 204'800 µstep/rev.
Reference Resolution Den.		

Cycle data		
Cycle Type	[1 – 7]	1: Jog, 2: Relative indexer, 3: Calibration, 4: Delta stop, 5: Absolute Indexer, 6: Position delay, 7: Time delay.

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

Speed	[µstep / sec]	Range: 1 – 300˙000 µstep/sec.
Position	[µstep] or [msec]	Motor µsteps to be executed in an indexer relative cycle. Position destination in an indexer absolute cycle. Maximum motor µsteps to be executed in a delta stop cycle. µsteps 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	[µstep]	Programmable µsteps of movement after delta stop input activation.
Arrows for cycle selection		To configure the movement data of 32 programmable cycles.

Sequences		
0, …, 19		Cycle number [0- 31], Stop [254], Loop [255]. 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		
Ramp Profile		Linear, parabolic or S-curve.
Start / Stop Frequency	[µstep]	Selected: it enables and set the start / stop frequency. Deselected: automatic.
Acceleration	[1˙000 µstep / sec^2]	e.g. 15 stands for an acceleration of 15˙000 µstep / sec^2.
Deceleration	[1˙000 µstep / sec^2]	e.g. 15 stands for a deceleration of 15˙000 µstep / sec^2.

Calibration		
Calibration position	[µstep]	The position loaded at the end of a calibration cycle.
Homing		Type of calibration cycle to be implemented. Ref. to 6.4.
Position Offset	[µstep]	The encoder magnet is mounted in a random angular position. This parameter modifies the zero-encoder position.
Power-On Calibration Limit	[µstep]	Programmable power-on position deviation to assess the axis calibration Ref. to 12.
Research Speed	[µstep / sec]	Speed to research the homing switch.
Release Speed	[µstep / sec]	Speed to release from the homing switch.
Time Stop	[msec]	Delay time between research and release speed in calibration cycles.

Application version		It is a free data to be used by the master to associate a version number to the current parameters setting.

Tab. 2 – Programming data

5.6. Data Exchange

Connect Modbus button establishes a continuous communication with the target in RS-232 or RS-485 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.

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

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

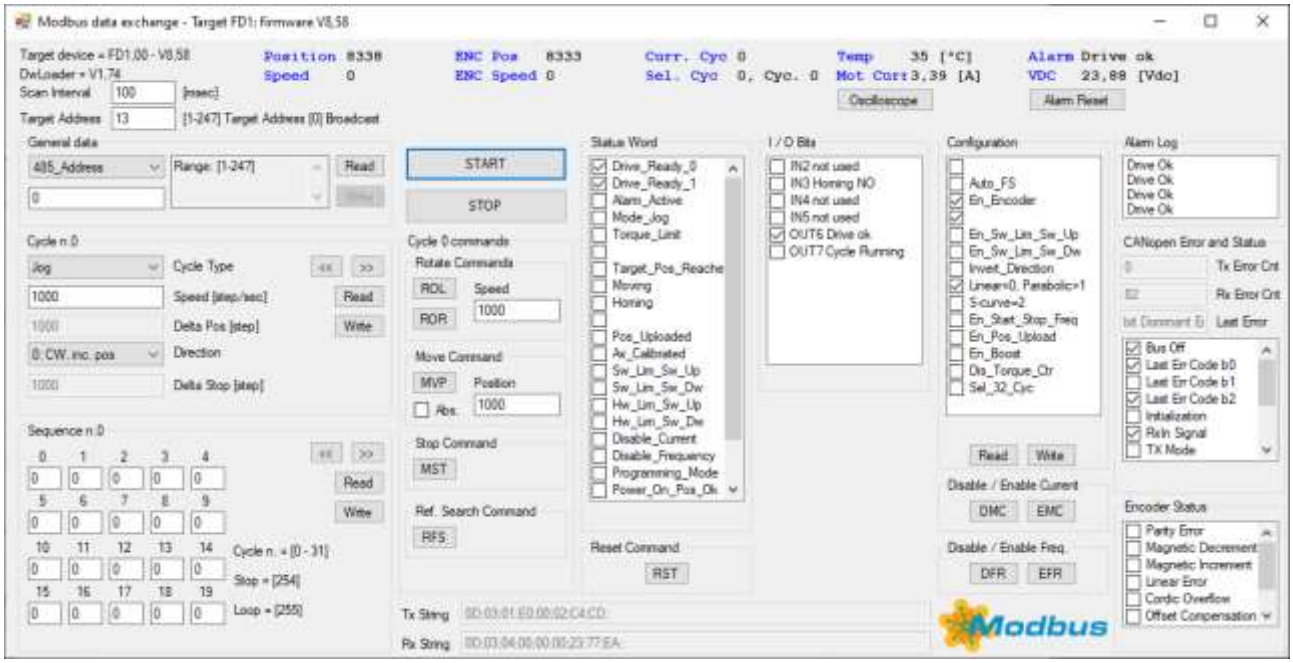


Fig. 4 – Modbus data exchange

Before clicking Connect Modbus, verify the target parameters in DwLoader windows: address (in RS-485 only) and baud rate shall be the same of those programmed in the drive, 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

Connect CANopen pushbutton will open the data exchange window which can be used to configure the PDO and test the communication protocol. It works with IXXAT USB-CAN converters libraries.

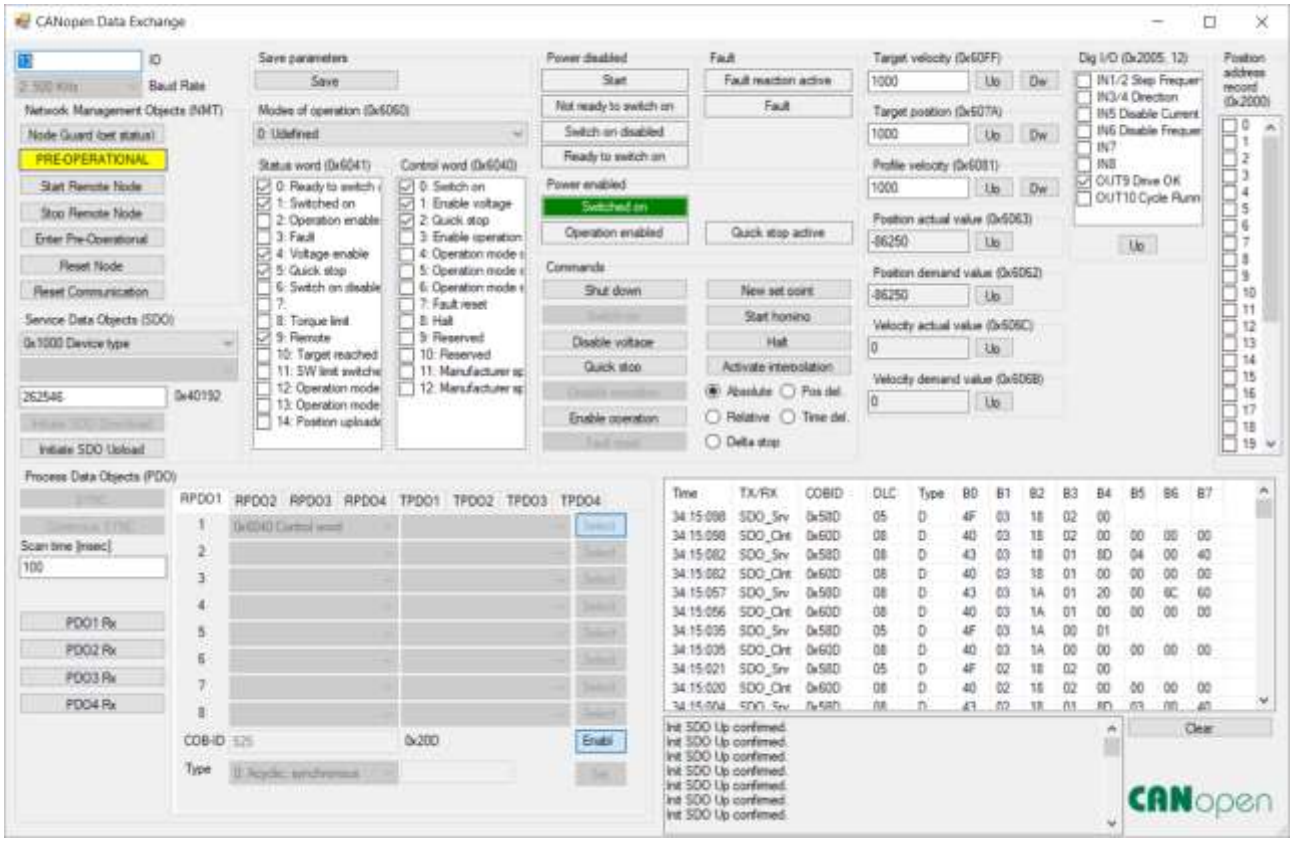


Fig. 5 – CANopen data exchange

5.7. 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 CANopen, RS-485 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 before sampling the selected data register (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.

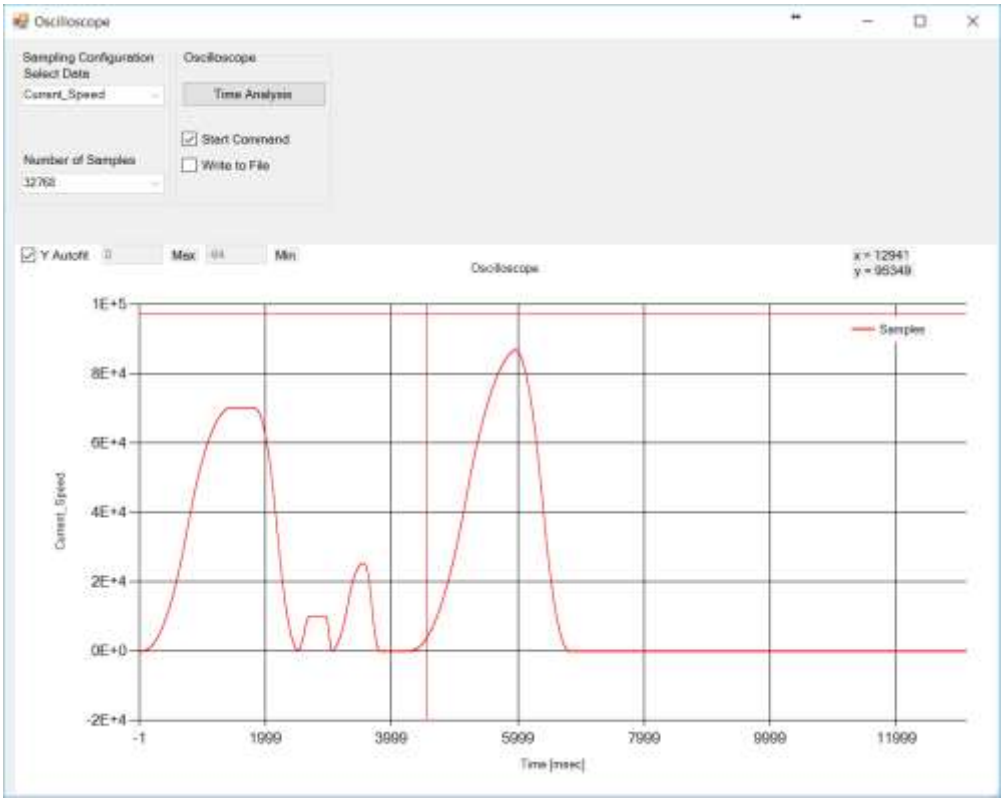


Fig. 6 – Oscilloscope

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 needs 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

6. CYCLES AND SEQUENCES

FD 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 up to 10 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).

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

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.

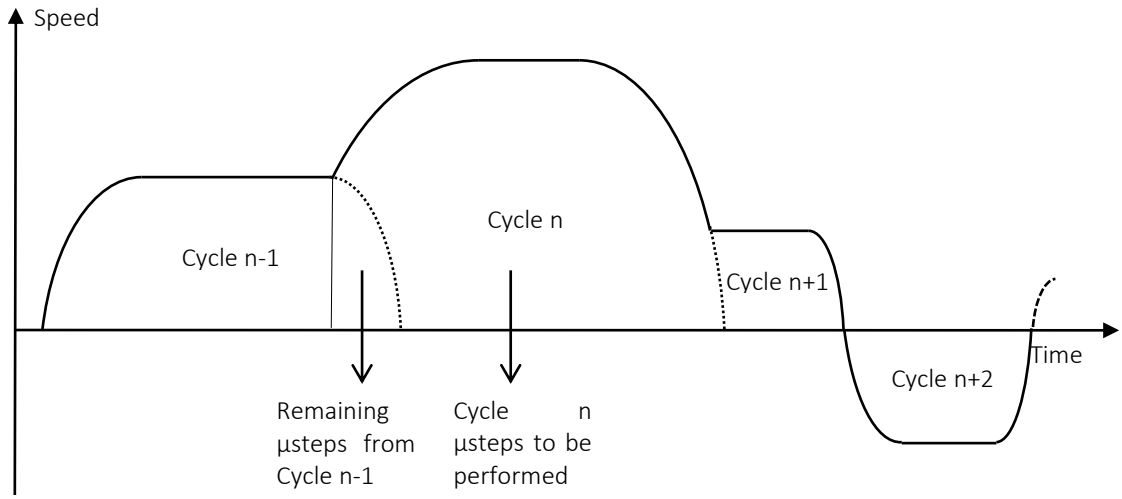


Fig. 7 – Sequence of cycles with parabolic ramps

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

6.1. Cycle and sequence selection

When no input is used as cycle or sequence selection, cycles and sequences can be selected via Modbus 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

Because FD1 has fewer digital inputs for selection than FD2, they adopt different logic:

In FD1 IN3 is cycle or sequence selection bit 0, IN5 bit 1, IN4 bit 2. The bit weight of the inputs is fixed, for example if only IN5 and IN4 are used, only sequences 0, 2, 4, 6 can be selected with CONFIG register bit 13 equals to zero. With CONFIG register bit 13 equals to one only cycles 0, 2, 4, 6.

In FD2 IN3/4, IN5, IN6, IN7 and IN8 can be used for cycle or sequence selection. Inputs don't have a pre-assigned bit weight, but the weight is ordered with the input number (weight zero is assigned to the lowest input number), e.g., if only IN5 and IN7 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 IN6 and IN7.
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 in Modbus on 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 the configuration in flash using IAP or write it via Modbus in RAM, ref. to 12.2.18-19.

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

6.2. Jog

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

During jog cycle the Modbus 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.

E.g. using FD2:

CONFIG register, bit 13	1: Select 32 cycles
IN1/2 configuration	1: Start NO,
IN3/4 configuration	1: Stop NO,
IN5 configuration	11: Cycle or sequence selection,
IN6 configuration	11: Cycle or sequence selection,
Cycle 0, Type	0: Jog,
Cycle 0, Speed	1'000 µstep/sec
Cycle 0, Direction	0: CW, increasing positions
Cycle 1, Type	0: Jog,
Cycle 1, Speed	1'000 µstep/sec
Cycle 1, Direction	1: CCW, decreasing positions
Cycle 2, Type	0: Jog,
Cycle 2, Speed	3'000 µstep/sec
Cycle 2, Direction	0: CW, increasing positions
Cycle 3, Type	0: Jog,
Cycle 3, Speed	3'000 µstep/sec
Cycle 3, Direction	1: CCW, decreasing positions

In this example the selection addresses cycles and IN5 has weight bit 0 and IN6 has weight bit 1. Jog cycles 0, 1, 2 and 3 have been configured in order to have IN5 direction, IN6 low and high speed:

IN5 = 0	CW, increasing positions
IN5 = 1	CCW, decreasing positions
IN6 = 0	Low speed 1'000 µstep/sec
IN6 = 1	High 3'000 µstep/sec

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

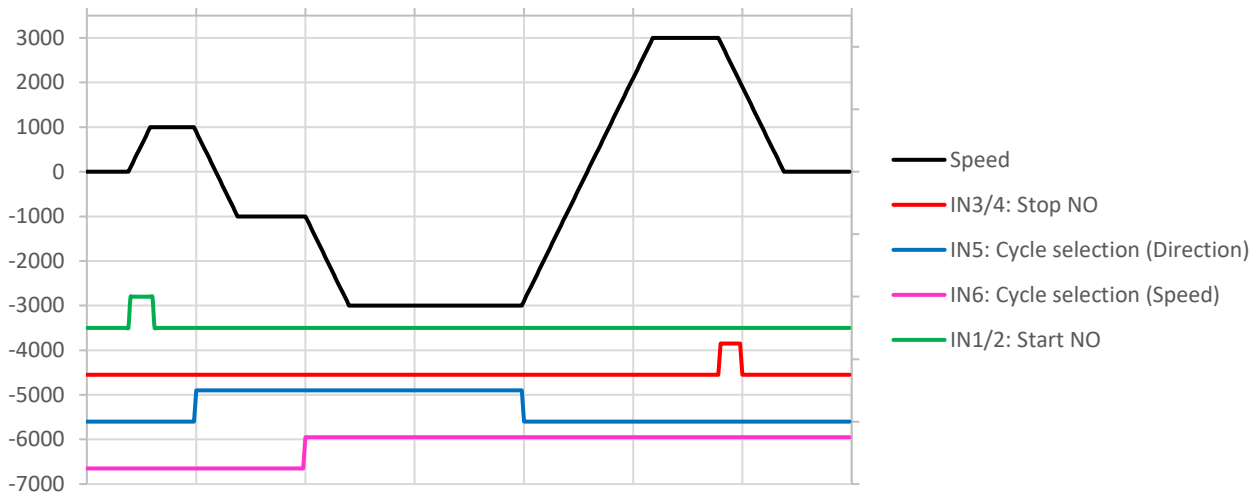


Fig. 7 – Jog

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

6.3. 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 bit 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-bit 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 this case it is anyhow 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

6.4. 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 (on FD1 and FD2 only),
5: Homing, time stop, no inversion, marker (on FD1 and FD2 only).
6: Mechanical stop,
7: Mechanical stop, 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.

FD1 IN3, IN4, IN5 and FD2 IN3/4, IN5, IN6 can be set as homing inputs.
FD1 IN3, IN5 and FD2 IN5, IN6 can be set as hardware limit switch inputs.
Calibration is possible on both configurations.

FD1 and FD2 drives can restore the multi-turn axis calibration at the power-on using the integrated absolute encoder. Ref. to 12.

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

6.4.1. 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.

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.

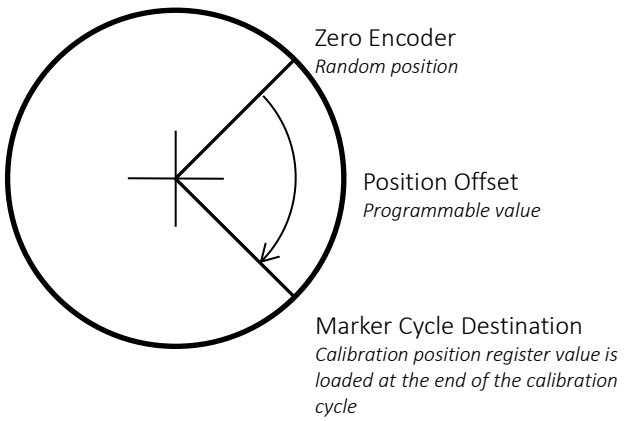


Fig. 8 – Marker Cycle

For CW and CCW directions, if the start position of the movement is very close to the destination, there might be one-revolution uncertainty in motor movement as represented in Fig. 9 for a CW movement.

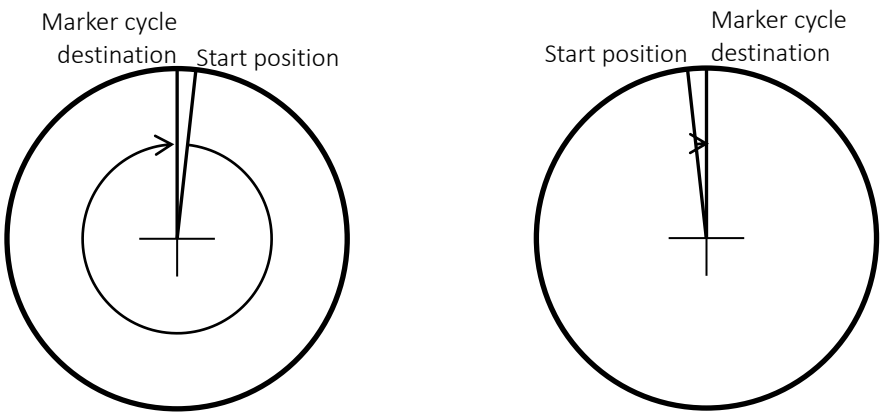


Fig. 9 – Marker cycle uncertainty

In order to detect such behavior, IX_TOO_EARLY and IX_TOO_LATE, respectively bit 21 and 22 of 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

have been implemented. IX_TOO_EARLY gets active if the marker movement takes less than 1/8 revolution, IX_TOO_LATE gets active is it takes more than 7/8 revolution.

6.4.2. 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 is concluded.

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

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.

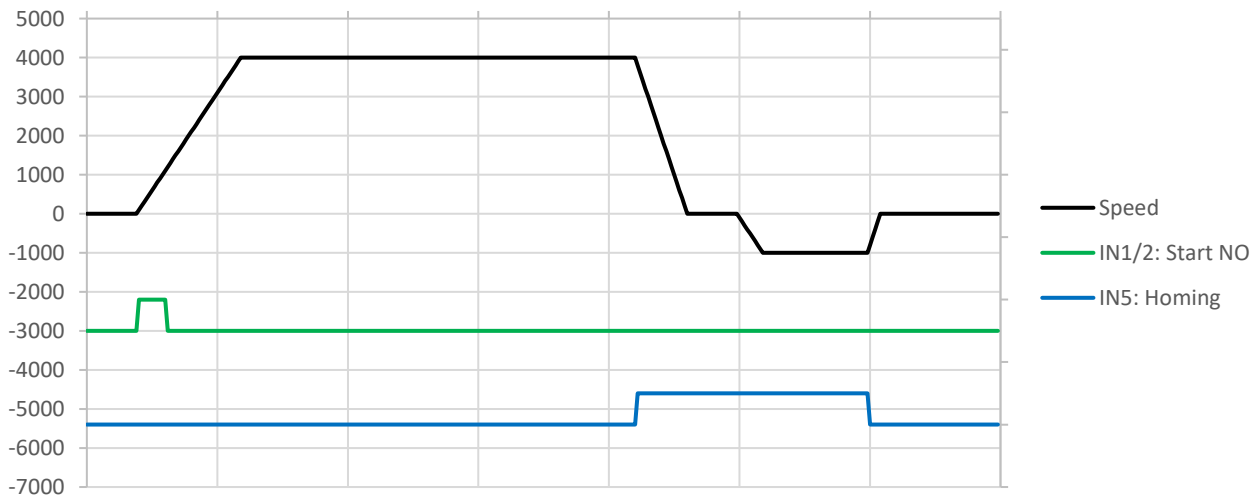


Fig. 9 – Homing invert

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.

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

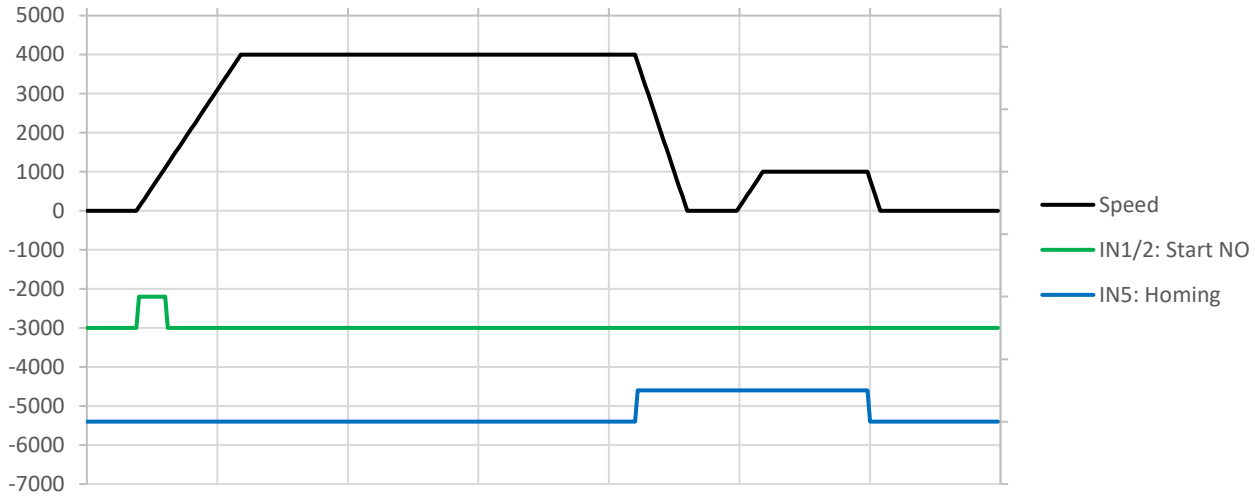


Fig. 10 – Homing no invert

This homing method allows a more precise end position, since the stop is made at V_RELEASE speed, typically much lower than V_RESEARCH speed.

6.4.4. 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.

When homing register is equal to five, the motor does not invert the direction, but runs at V_RELEASE speed at the same direction.

Note:
Absolute encoder position destination needs to be set approximately half motor revolution far from the deactivation of the switch to avoid the possible uncertainty of one revolution that would happen if the motor stops few steps before or after the encoder position destination.

6.4.1. Mechanical stop

When HOMING register is equal to six, the homing is not performed on a sensor, but against a mechanical stop.
The motor rotation takes place at V_RESEARCH speed at the cycle direction:

0: CW,
1: CCW.

when the axis movement gets stopped against a physical block, the drive continues to produce torque against the obstacle until the following error exceeds an internal limit.
When such limit is reached, the calibration is concluded.

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

6.4.2. Mechanical stop, Marker

Setting HOMING register equal to seven, a mechanical stop plus marker method is performed, i.e. after the conclusion of mechanical stop movement, a marker cycle is launched in order to calibrate the axis in the more accurate absolute encoder position.

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

6.5. Delta stop

Delta stop is a particular indexer relative cycle, used when it is necessary 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 µsteps after delta stop input activation. FD1 IN3 and IN5, FD2 IN5 and IN6 can be configured as delta stop inputs.

This cycle type is defined by speed, position, direction and delta stop µsteps.

When delta stop input gets active the motor executes exactly the µsteps 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 while running at high speed.

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 OUT7, FD2 OUT10) 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 their proper increment at the end of the cycle, i.e. when TARGET_POS, bit 6 of STATUS_WORD register, is set.

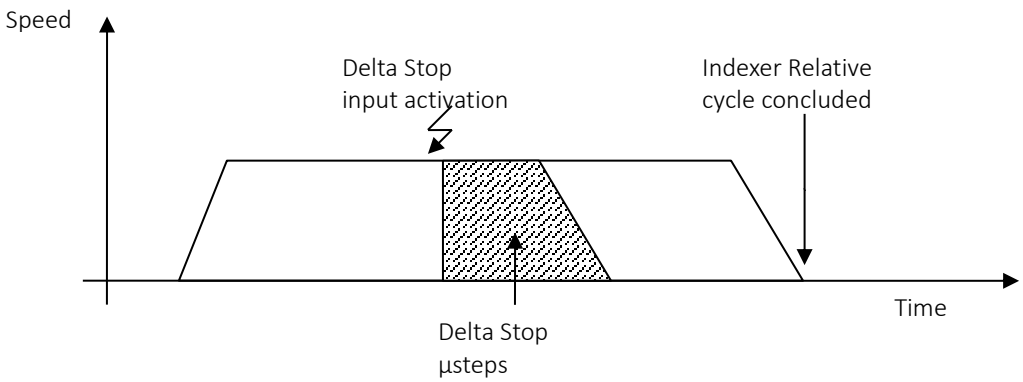


Fig. 11 – Delta Stop Cycles

Note:
The position of sensor activation shall be anticipated in respect to the wanted destination position of the delta stop steps.

To avoid steep deceleration, make sure to set Delta Stop steps parameter higher than the deceleration steps at the maximum speed used.

Ramp	*Linear*	*Parabolic*	*S-curve*
Minimum delta stop steps	$\frac{v^2}{2d}$	$\frac{4v^2}{3d}$	$\frac{v^2}{d}$

Where v is expressed as µstep/sec, and d as µstep/sec^2.

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

6.6. 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.

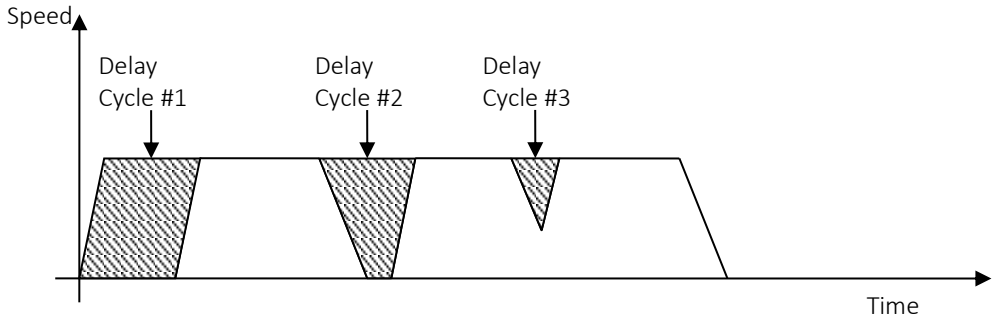


Fig. 12 – Delay Cycles

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.

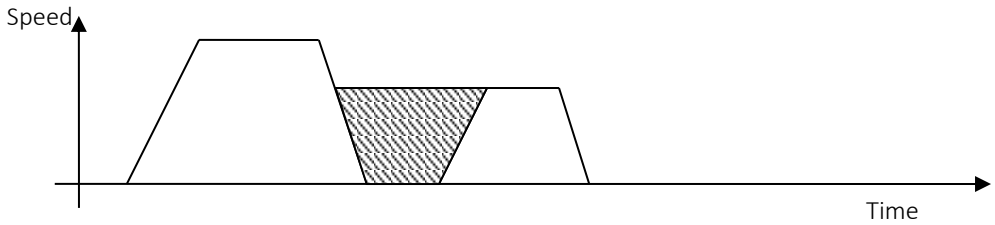


Fig. 13 – Delay cycles: speed before greater than speed after

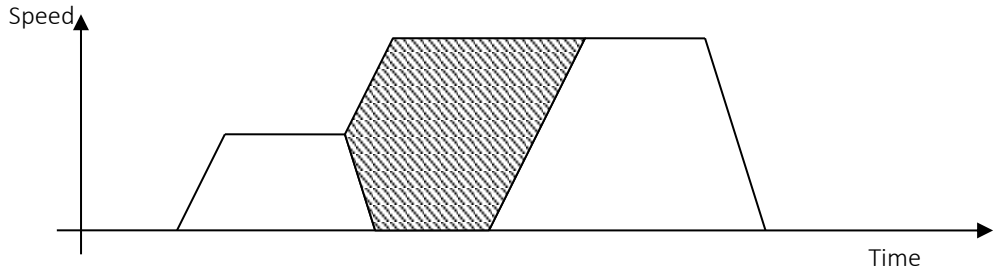


Fig. 14 – Delay cycles: speed before lower than speed after

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

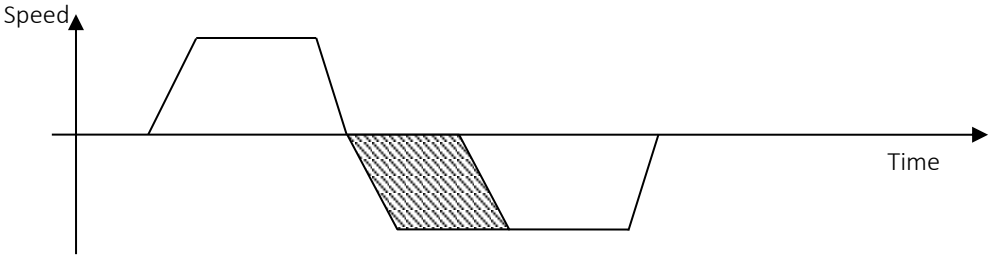


Fig. 15 – Delay cycles: direction inversion

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.

6.7. 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

7. 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.

7.1. 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.

7.2. 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.

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

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

7.4. Reference Search (RFS) EXE_FUN = 13

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

7.5. 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.

7.6. 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.

7.7. 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.

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 current disable and then current enable perform an alarm reset.

During current disable the STATUS WORD bit 16 DIS_CURR is set.

7.8. 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.

7.9. 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,

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

- 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, ...).

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. INPUTS / OUTPUTS

FD drives have configurable inputs and outputs.

Type	Digital Inputs	Digital Outputs	Analogue Inputs
FD1	4	2	
FD2	6	2	1

Tab. 3 – Inputs / Outputs

Generally, two digital inputs are configurable as input frequency, direction or start, stop. Other two digital inputs, named multipurpose inputs, are for delta stop, cycle or sequence selection, homing and limit switches, enable or disable current, etc. remaining two are for cycle or sequence selection.

The inputs name is taken from the pin on the connector: e.g., FD1 IN2 is the pin 2 of CN3 (the common is taken for all the four inputs on the pin 1 of CN3), FD2 IN3/4 is a differential input from pins 3 and 4.

FD1 OUT6 and FD2 OUT9 are used for signaling the alarm status (normally closed, i.e., drive ok, is recommended). FD1 OUT7 and FD2 OUT10 can be configured as cycle running, axle calibrated, etc.

FD2 is provided with one analogue input. By default, the only function associated to this input is to scale the speed set-point of all the cycles proportionally with the input value. Any other implementation, variable scaling, etc. are available upon request.

The status of all the inputs and outputs can be monitored from Modbus register IO_BITS. The value of analogic input from X_IN_VAL_A1.

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

8.1. Start/Stop and Input Frequency Signals

FD1 IN2 and IN4, FD2 IN1/2 and IN3/4 can be configured as step/dir, quadrature A/B input frequencies, start/stop NO/NC and other additional functionalities as cycle or sequence selection. Depending on the model the configuration changes, giving the possibilities of using also IN3 and IN5. Refer to table below.

Inputs	Configuration	Description
FD1 IN2 FD2 IN1/2	0: Step input frequency	Every pulse produces a motor µstep. The maximum input frequency allowed is 300 kHz (complementary with direction input frequency).
	1: Start NO 2: Start NC	When active it starts the selected cycle or sequence.
	6: Quadrature input frequency A	Every signal transition produces one µstep (x4)
	7: Start NO, Stop NC, 8: Start NC, Stop NO.	When active it starts the selected cycle or sequence. Inactive, it stops it.
FD1 IN4 FD2 IN3/4	0: Direction input frequency	Inactive down-counting CCW, active up-counting CW (when no direction inversion is applied). Input setup time: 200 µsec (complementary with step input frequency).
	1: Stop NO 2: Stop NC	When active it stops the selected cycle or sequence.
	3: Cycle or sequence selection	Bit weight 2 for FD1. Scalable weight for FD2. Ref. to 6.1.
FD2 IN3/4	6: Quadrature input frequency B	Every signal transition produces one µstep
FD1 IN5	14: Quadrature input frequency B	Every signal transition produces one µstep
FD1 IN3 FD1 IN5 FD2 IN5 FD2 IN6	10: Direction input frequency	Low value is down-counting CCW, high value is up-counting CW (when no direction inversion is applied). Input setup time: 200 µsec

Tab. 4 – IN2/4 inputs description

Modbus and CANopen commands are always available and managed with higher priority in respect to input frequency signals. Input frequency signals are disabled when the motor is executing a command and they are enabled again at the end of the movement.
In input frequency modes acceleration or deceleration ramps are not provided from the drive itself. The master which is supplying the signals is supposed to create them.

Quadrature input frequency 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.

Note:
Hardware and software limit switches are not active in input frequency modes.

FD1 IN4 can be set as cycle or sequence selection bit 2 plus homing. In this case, 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.

When the input is configured as input frequency (step, direction or quadrature A/B), measures to reduce external disturbances, such as shielded cables, need to be put in place. All the other's input configurations have a digital filter and a Schmitt trigger with a maximum delay of 2.1 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

8.2. Multipurpose Inputs

FD1 IN3 and IN5, FD2 IN5 and IN6 are multipurpose input channels. They can be configured as:

Input	Configuration	Description
FD1 IN3 FD1 IN5 FD2 IN5 FD2 IN6	0: Disable current	It frees the motor shaft. Using the encoder, the drive can keep on incrementing the position counter of the movement happened when current was disabled. When current is enabled again position and status word bits are restored.
	1: Disable frequency	It inhibits all the movements from commands or any input frequency. This input can be used to multiplex a single frequency output to more drives.
	2: Homing NO	Homing sensor. Ref. to 6.4.
	3: Homing NC	
	5: Hardware limit switch up NO	Motor movements are stopped in the direction of the switch and it can be used also in calibration cycles as homing cycles. They are not active in input frequency mode.
	6: Hardware limit switch up NC	
	7: Hardware limit switch down NO	
	8: Hardware limit switch down NC	
	9: Encoder position latch	The input is used to sample and latch the multi-turn encoder position at rising and falling edges of selected input. Positions can be read from ENC_LATCH_UP and ENC_LATCH_DW registers. The resolution is the same of the encoder (4˙096 steps/rev on FD1 and FD2)
	11: Cycle or sequence selection	
	12: Delta stop NO	Ref. to 6.5.
	13: Delta Stop NC	
	15: Start NO	
	16: Start NC	
	17: Stop NO	
	18: Stop NC	
	19: Enable motor current	When not active, it frees the motor shaft. By the encoder, the drive keeps on counting the position feedback. When current is enabled again position and status word bits are restored.

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 these two input channels in order to reduce noise effects.
The low pass filter is removed when these inputs are set as encoder latch or direction input frequency (setup time = 200 µsec).

8.3. Aux inputs

FD2 is provided with other two additional inputs IN7 and IN8.

INPUT	CONFIGURATION	DESCRIPTION
FD2 IN7 FD2 IN8	11: Cycle or sequence selection	

Tab. 6 – IN7 and IN8 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 these two input channels 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

8.4. Alarm output

FD1 OUT6, FD2 OUT9 are used to signal the status of the drive. When configured as drive ok, the signal is normally closed, which means at active level when no alarm is present. Doing so a cable break will generate an alarm in the host.

OUTPUT	CONFIGURATION	DESCRIPTION
FD1 OUT6 FD2 OUT9	0: Drive ok	High logic level indicates that the drive is capable to receive commands, no alarm is present
	1: Alarm	Low logic level indicates that the drive is capable to receive commands, no alarm is present
	2: Drive ok + not running	In idle, without alarms, the output logic level is high. As soon as a movement is started, the output level gets low because of "not running" condition. A timeout can be implemented: if at the end of the movement the signal does not get back high, it means that an alarm is present. This is configuration is useful to gather dual information on single wire.
	3: Alarm + running	In idle, without alarms, the output logic level is low. As soon as a movement is started, the output level gets high because of "running" condition. A timeout can be implemented: if at the end of the movement the signal does not get back low, it means that an alarm is present. This is configuration is useful to gather dual information on single wire.

8.5. Multipurpose output

FD1 OUT7, FD2 OUT10 are digital outputs normally opened. They can be configured as:

0: Cycle running
The output is high during running cycles and sequences.

3: Position uploaded
The output is high if the power off position is exactly restored after power on. Ref. to 11.

4: Axle calibrated
The output is high if the axle is calibrated.

5: Power on position ok
The output is high if the power off position is within POWER_ON_CALIBRATION_LIMIT register value after power on. Ref. to 11.

6: Torque limit
The output is high when the motor is not able to follow the ordered position.

7: Cycle running + not delta stop
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 steps, but the output remains active. The host or PLC which manage this output needs to implement a time-out on this signal deactivation.

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. ALARMS

In normal operation the bit 2 ALARM_ACTIVE of the STATUS WORD is reset, the output Drive ok is active (output active low), the ERR_FAT register value is zero, the green LED is flashing and the red LED is off.

If any of below alarm condition is triggered, motor gets immediately de-energized and the shaft freed.

9.1. Steps Accumulation Limit Alarm

ERR_FAT = 1
RED_LED = steady lit

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. If torque control is enabled, V8 is able to keep producing the maximum torque, accumulating the steps that cannot be executed, without losing the synchronism with the motor. If torque control is disabled, when the position error is higher than 1,8°, the motor loses the synchronism and steps are lost.
When the difference of ordered steps and executed steps exceeds the programmable ACCUMULATION_LIMIT register, the steps accumulation limit alarm arises and the drive stops. ACCUMULATION_LIMIT register is upper limited to 10 revolutions.
In case of alarm, the ERR_FAT register will be equal to one, the red LED becomes steady lit and the output Drive Ok turns off. This alarm can be reset using Alarm Reset command.

The red LED will also light up (not latched) when the ordered steps are out of synchronism (step accumulation exceeds ± 1.8°, TORQUE LIMIT mode), but the alarm is not active.

9.2. Over-Temperature Alarm

ERR_FAT = 2
RED_LED = high frequency flashing (5 Hz)

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.
In case of over-temperature alarm, the ERR_FAT register value will be equal to two and the red led flashes at high frequency.

9.3. Short Circuit Alarm

ERR_FAT = 3
RED_LED = low frequency flashing (0.5 Hz)

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.
The ERR_FAT register will be equal to three and the red LED will flash at low frequency.

Note:
This is a hardware alarm. On FD1 is possible to reset this type of alarm only by turning off and on the device. On FD2 it is possible to reset it using Alarm Reset command.

9.4. Over-Voltage Alarm

ERR_FAT = 4
RED_LED and GREEN_LED = steady lit

This is a latched hardware fault that occurs in the event of an over-voltage on the input DC power supply. On FD1 and FD2, being supplied by DC voltage, this alarm could appear in case of high deceleration with high inertial load and weak external output power supply capacitor.
The ERR_FAT register will be equal to four and the red and green LED will be steady active.

FD overvoltage alarms are set as:
- FD1 88 V_{DC},
- FD2 135 V_{DC},

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

9.5. Data Error

ERR_FAT = 5
RED_LED and GREEN_LED = alternate high frequency flashing (5 Hz)

This alarm arises at power on in case of wrong settings in the flash memory. This alarm cannot be reset in any case, contact the supplier.
The ERR_FAT register will be equal to five and the red and green LED will flash at high frequency alternatively.

9.6. Under Voltage

ERR_FAT = 7
Red LED short blink every 3 seconds

Hardware versions with suffix D and Z, such as FD2.1D or FD2.1Z, and FD1.6 are equipped with DC/DC converter that supplies the drive logic from V_{EXT}, even when the main power supply is off.
When the main power supply is below 17 V_{DC} 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 (exception made for FD1 short circuit, that can be reset only by shutting down completely the drive) and multiturn position is restored. If the axis was calibrated, it remains so (bit 11 AX_CALIBRATED of STATUS_WORD set).

Note:
FD1 drives: V_{EXT} at CN3.8 – the 24 V_{DC} that powers the drive logic when main power is OFF – is referred to power ground, GND_{POW}, at CN5.2 of FD1.6 and CN5.3/CN5.4 of FD1.5. GND_{EXT} shall be connected externally to GND_{POW} and GND_{POW} cannot be disconnected from the device when V_{EXT} is ON.
FD2 drives: V_{EXT} at CN3.11 is referred to GND_{EXT} at CN3.12 and it is galvanically insulated from main power.

9.7. Alarm Resetting

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

- Alarm Reset (Exe_Fun = 15)
- Disable and Enable Current (Exe_Fun = 16 then Exe_Fun = 17, or using Multipurpose Inputs configured as Disable current)
- CANopen fault reset
- System Reset (Exe_Fun = 14)
- ROL or ROR (Exe_Fun = 1 or 2)
- Turn the main power off and on

Alarm Reset and Disable/Enable Current are preferred, because after a System Reset or ROL/ROR 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	ERR_FAT = 0
Programming mode	Red and green LEDs blinking alternatively: 5 Hz communication ON 0.5 Hz communication OFF	ERR_FAT = 0 Status Word bit 18 high
Step loss	Red LED steady ON Green LED same as Drive ok status	ERR_FAT = 1
Over temperature	Red LED blinking at 5 Hz Green LED same as Drive ok status	ERR_FAT = 2
Short circuit	Red LED blinking at 0,5 Hz	ERR_FAT = 3

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

	Green LED same as Drive ok status	
Over voltage	Red LED steady ON Green LED steady ON	ERR_FAT = 4
Data error	Red and green LED blinking at 5 Hz together	ERR_FAT = 5
Under voltage	Red LED short blink every 3 seconds Green LED same as Drive ok status	ERR_FAT = 7
Absolute encoder warning (only on FD1 and FD2)	Green LED short blink at 0,25 Hz	ERR_FAT = 0 ENC_STATUS ≠ 0

Tab. 7 – Diagnostic LEDs and Errors

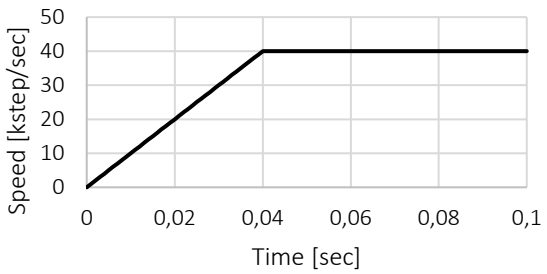
e-mail: info@auxind.com
Tel: (+39) 0522 520312 – Fax, Tel: (+39) 0522 521333
Via M. Montessori, 25 – 42123 Reggio Emilia, Italy
C.F. / P.I. 01844360352 – R.E.A. di R.E. 228913 –
Reg. Impr. 27689 / 99

10. RAMP PROFILES

V8 self generates linear, parabolic and s-curved speed ramps.

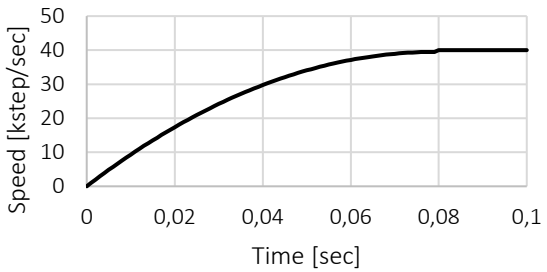
Linear ramp:
ACCELERATION and DECELERATION registers represent the speed increment / decrement every millisecond.

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



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

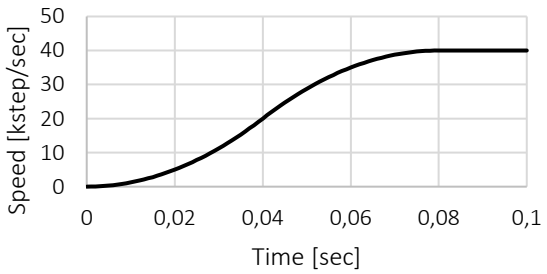
$$t_{acc} = \frac{2V}{acc}$$



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.

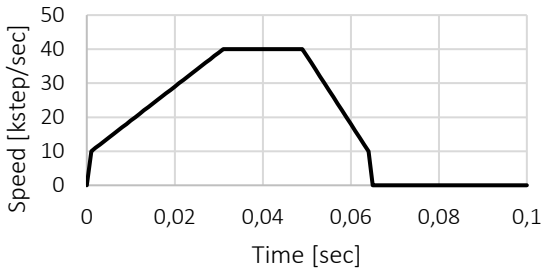
$$t_{acc} = \frac{2V}{acc}$$



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 µsteps per second – as a function of time. For all of them regime speed of 40˙000 µstep/sec and acceleration of 1˙000˙000 µstep/sec^2 have been used.

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 µstep/sec, start stop frequency of 10˙000 µstep/sec, acceleration of 1˙000˙000 µstep/sec^2 and deceleration of 2˙000˙000 µstep/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

11. POWER-ON POSITION RESTORE

When the driver is powered off, V8 saves the actual position and status.

11.1. 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, STATUS_WORD bit 11 AX_CALIBRATED and bit 6 TARGET_POS are restored and bit 10 POS_UPLOADED is set.

11.2. 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 take 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.

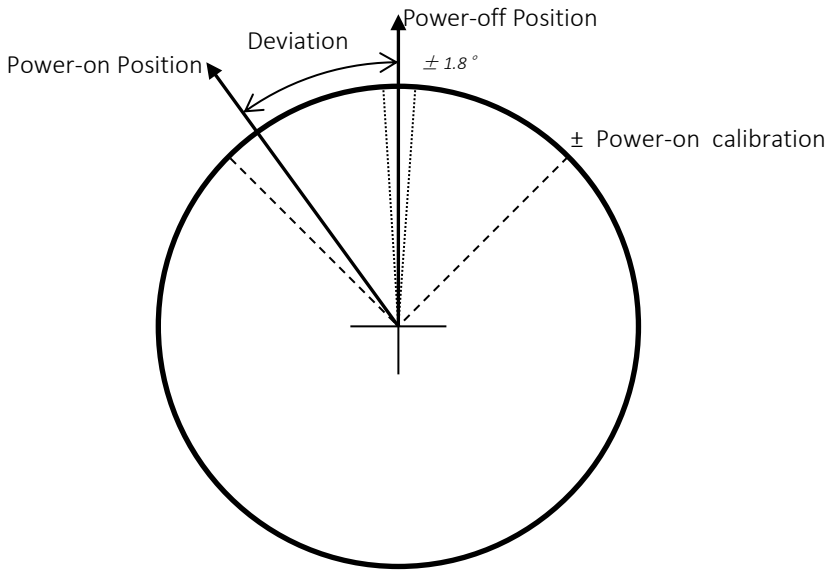


Fig. 17 – Position restore angles

11.3. 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

12. MODBUS

12.1. Holding registers

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).

	Register address[1]	
Register name	H	L
START	0	1
STOP	2	3
ACCELERATION	4	5
DECELERATION	6	7
CURR_SPEED	8	9
CURR_POSITION	10	11
CURR_CYCLE	12	13
CALIB_POSITION	14	15
POSITION_OFFSET	16	17
SEL_CYC_SEQ	18	19
TARGET_VERSION	20	21
IO_BITS	22	23
CONFIG	24	25
MODBUS_485_ADDRESS	26	27
EXE_FUN	28	29
I_MAX	30	31
MODBUS_485_DELAY	32	33
ERR_FAT	34	35
MODBUS_BAUD_RATE	36	37
STATUS_WORD	38	39
CYC_0_TYPE	40	41
CYC_0_SPEED	42	43
CYC_0_DELTA_POS	44	45
CYC_0_DIRECTION	46	47
CYC_0_DELTA_STOP	48	49
CYC_n_TYPE	2 x (20 + n x 5)	2 x (20 + n x 5) + 1
CYC_n_SPEED	2 x (21 + n x 5)	2 x (21 + n x 5) + 1
CYC_n_DELTA_POS	2 x (22 + n x 5)	2 x (22 + n x 5) + 1
CYC_n_DIRECTION	2 x (23 + n x 5)	2 x (23 + n x 5) + 1
CYC_n_DELTA_STOP	2 x (24 + n x 5)	2 x (24 + n x 5) + 1
SEQ_0_DW_0	360	361
SEQ_0_DW_1	362	363
SEQ_0_DW_2	364	365
SEQ_0_DW_3	366	367
SEQ_0_DW_4	368	369
SEQ_n_DW_0	2 x (180 + n x 5)	2 x (180 + n x 5) + 1
SEQ_n_DW_1	2 x (181 + n x 5)	2 x (181 + n x 5) + 1
SEQ_n_DW_2	2 x (182 + n x 5)	2 x (182 + n x 5) + 1
SEQ_n_DW_3	2 x (183 + n x 5)	2 x (183 + n x 5) + 1
SEQ_n_DW_4	2 x (184 + n x 5)	2 x (184 + n x 5) + 1
FD1: IN_3_CNF FD2: IN_5_CNF	460	461
FD1: IN_5_CNF	462	463

[1] The 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 address[1]	
Register name	H	L
FD2: IN_6_CNF		
HOMING	464	465
TS	466	467
V_RESEARCH	468	469
V_RELEASE	470	471
POS_SW_LS_UP	472	473
POS_SW_LS_DW	474	475
I_LIM	476	477
KNUM_NOM	478	479
TEMPERATURE	480	481
KV	482	483
ACCUMULATION_LIMIT	484	485
ENC_REV	486	487
ACCUMULATED_STEPS	488	489
I_MIN	490	491
ENC_POS	492	493
ENC_SPEED	494	495
ENC_LATCH_UP	496	497
PON_CALIB_LIM	498	499
TIME_CONST	500	501
START_STOP_FREQ	502	503
TEMP_OFF	504	505
FD1: IN_2_CNF FD2: IN_1_2_CNF	506	507
FD1: IN_4_CNF FD2: IN_3_4_CNF	508	509
FD1: OUT_6_CNF FD2: OUT_9_CNF	510	511
FD1: OUT_7_CNF FD2: OUT_10_CNF	512	513
RESOLUTION_NUM	514	515
RESOLUTION_DEN	516	517
I_MIS	518	519
CANOPEN_ADDRESS	520	521
CANOPEN_BAUDRATE	522	523
CNT_CYC	524	525
CNT_DSTOP	526	527
ENC_LATCH_DW	528	529
ERR_LOG	530	531
…		
FD2: IN_7_CNF	536	537
FD2: IN_8_CNF	538	539
ENC_STATUS	540	541
FD2: IN_CNF_A1	542	543
FD2: IN_VAL_A1	544	545
…		
CANOPEN_STATUS	550	551
V_DC	552	553
…		
APP_VERSION	582	583
RPDO1_PARAM	584	585
RPDO1_MAPPING1	586	587
RPDO1_MAPPING2	588	589

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 address[1]	
Register name	H	L
RPDO1_MAPPING3	590	591
RPDO1_MAPPING4	592	593
RPDO1_MAPPING5	594	595
RPDOn_PARAM	2 x (286 + n x 6)	2 x (286 + n x 6) + 1
RPDOn_MAPPING1	2 x (287 + n x 6)	2 x (287 + n x 6) + 1
RPDOn_MAPPING2	2 x (288 + n x 6)	2 x (288 + n x 6) + 1
RPDOn_MAPPING3	2 x (289 + n x 6)	2 x (289 + n x 6) + 1
RPDOn_MAPPING4	2 x (290 + n x 6)	2 x (290 + n x 6) + 1
RPDOn_MAPPING5	2 x (291 + n x 6)	2 x (291 + n x 6) + 1
TPDO1_PARAM	632	633
TPDO1_MAPPING1	634	635
TPDO1_MAPPING2	636	637
TPDO1_MAPPING3	638	639
TPDO1_MAPPING4	640	641
TPDO1_MAPPING5	642	643
TPDOn_PARAM	2 x (310 + n x 6)	2 x (310 + n x 6) + 1
TPDOn_MAPPING1	2 x (311 + n x 6)	2 x (311 + n x 6) + 1
TPDOn_MAPPING2	2 x (312 + n x 6)	2 x (312 + n x 6) + 1
TPDOn_MAPPING3	2 x (313 + n x 6)	2 x (313 + n x 6) + 1
TPDOn_MAPPING4	2 x (314 + n x 6)	2 x (314 + n x 6) + 1
TPDOn_MAPPING5	2 x (315 + n x 6)	2 x (315 + n x 6) + 1
TPDO1_TIME	680	681
TPDO2_TIME	682	683
TPDO3_TIME	684	685
TPDO4_TIME	686	687
…		
CAN_LIFETIME	700	701
…		
CAN_PROD_HEARTBEAT	704	705

Tab. 8 – Registers addresses

12.1.1. Start, Stop

START and STOP are two Read/Write registers normally equal to zero. Writing a value different from zero into the START register, the motor will run using selected cycle data. Writing a value different from zero into 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.

12.1.2. 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]

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

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

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

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

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

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

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

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

12.1.3. 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.

12.1.4. Calibration Position, Position Offset

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

12.1.5. 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.

12.1.6. 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.

12.1.7. 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	

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		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

12.1.8. Configuration

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

Bit number	Name	Description
1	AUTO_FULL_STEP_ENABLE	It increases motor torque at middle speed range
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 10)
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

12.1.9. Modbus address, delay and baud rate

MODBUS_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.

MODBUS_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.

MODBUS_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.

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

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 reception CAN filter via specific protocol commands. For this reason, DIP switches are read at power on and modifiable only via CANopen commands.

12.1.10. 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
105	SAVE_PARAM	Saves current RAM parameters setting into flash. This command shall be launched when the motor is stopped and without vertical loads. During flash programming, the drive disables motor current and, if it was previously enabled, it enables the current as soon as it finishes. This lasts approx. 300 msec.

Tab. 11 – Execute function register

12.1.11. 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.

When torque control is deactivated, the drive increases the motor current at I_MAX when running and decreases it at I_MIN when not running.

I_MIS is a Read Only register, which express the actual motor current sine wave peak.

12.1.12. Fatal Error and Error Log

ERR_FAT is a Read Only register used to read the status of the alarms. Ref. to 10.
ERR_LOG is a Read/Write register that stores the last 4 alarms. At Byte 0 is stored the last alarm occurred. The previous alarm at Byte 1 and so 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

12.1.13. 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 providing the maximum available torque at given speed to follow the position actual value, i.e., out of synchronism.
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 has 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.
21	IX_TOO_EARLY	Marker cycle stroke less than 1/8 of revolution
22	IX_TOO_LATE	Marker cycle stroke higher than 7/8 of revolution

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

12.1.14. 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

12.1.15. 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

12.1.16. 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
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.
12	HOMING_NO	Homing sensor input
13	HOMING_NC	

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

12.1.17. 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_UP and ENC_LATCH_DW 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	
19	CURRENT_ENABLE	

Tab. 17 – IN3/5 configuration register

12.1.18. 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

12.1.19. Output status

Output configuration register:
- FD1 OUT_6_CNF
- FD2 OUT_9_CNF

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

Value	Function	Description
0	DRIVE_OK	When the signal is low, an alarm is present
3	ALARM	When the signal is high, an alarm is present.
4	DRIVE_OK_NOT_RUN	If the signal level is low when the motor is not moving or after movement is completed, it means an alarm is present. During the movement, low signal means the drive started the movement. Timeout logic shall be implemented.
5	ALARM_RUN	If the signal level is high when the motor is not moving or after movement is completed, it means an alarm is present. During the movement, high signal means the drive started the movement. Timeout logic shall be implemented.

12.1.20. Output multi-purpose

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.
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.
7	DELTA_STOP_NOT_RUN	It is used in delta stop cycle to implement a time-out in case of delta stop sensor not received.

Tab. 19 – OUT7 and OUT10 configuration register

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

Tab. 20 – FD2 AI configuration registers

12.1.22. Homing, Time Stop, Release and Research Speeds

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, 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	Function	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	
6	MECH_STOP	
7	MECH_STOP_MRK	

Tab. 21 – Homing register

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

12.1.24. Temperature and temperature offset

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.

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

12.1.25. K

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

12.1.26. Accumulated steps and Steps Accumulation Limit

ACCUMULATION_LIMIT is a Read/Write register used for setting the step accumulation alarm limit. This register is upper limited to 10 revolutions. 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.
ACCUMULATED_STEPS is a Read Only register, signed 32, where the difference between ordered steps and position feedback is stored.
When 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.

12.1.27. 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_UP and ENC_LATCH_DW are a Read Write registers used in combination with a multipurpose input configured as encoder latch. In such configuration, when one of this input gets active (signal's rising edge) the multi-turn encoder position is saved into ENC_LATCH_UP register. When the input gets inactive (signal's falling edge) it is saved into ENC_LATCH_DW register.
ENC_REV is a Read Only register that shows the encoder position inside the revolution (it ranges from 0 to 4095).

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

12.1.28. Power On Calibration Limit

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

12.1.29. Time Constant

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

12.1.30. 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.

12.1.31. 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:

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

expressed in µstep per revolution.

12.1.32. 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

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: 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.

12.1.33. 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.

12.1.34. Encoder Status

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

Bit number	Name	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

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

12.1.35. 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.

12.1.36. Application version

APP_VERSION is a Read/Write register, which can be used to identify a specific parameters setting.
No logic is associated to this variable on drive side.

12.1.37. PDO parameter

XPDOy_PARAMETER is a Read/Write register, which is used to save in flash the COB-ID, type and status of the PDO. Compared to the PDO parameter record defined in CANopen, this is just a 32 bit register organized as follows:

Bit number	Name	Description
[0 – 11]	COB-ID	RPDO = 0x200 + Node-ID TPDO = 0x180 + Node-ID
[12 – 15]	reserved	
[16 – 23]	Type	0: acyclic, synchronous 1-240: synchronous every 1-240 sync 254, 255: asynchronous
24	Status	0: Disabled 1: Enabled
[25 – 31]	reserved	

12.1.38. PDO mapping

XPDOy_MAPPINGz is a Read/Write register, which is used to save in flash the mapping objects of the PDO. Compared to the PDO mapping record defined in CANopen, just 5 objects per PDO can be mapped. This has been chosen because for stepper motor driver applications, it is very unlikely the need of mapping 8 object of one single Byte in a PDO. Since the most of the object are word or double-word, 5 objects for a PDO of 8 Bytes are sufficient.

Bit number	Name	Description
0 – 7	Qty. of bits	It defines the length of the object to be mapped: Byte = 0x08 Word = 0x10 Double word = 0x20
8 – 15	Sub-index	
16 – 31	Index	

12.1.39. PDO timing

TPDOx_TIME is a Read/Write register, which is used to save in flash the asynchronous TPOD inhibit time and event timer. Asynchronous TPDO are transmitted every time there is a change in the mapped objects.
If position actual value is mapped, during rotation of the motor there would be too many TPDO that could stall the bus. To avoid this situation, it is possible to configure TPDO inhibit time, which is the minimum time to be elapsed between each TPDO transmission. Inhibit time is defined in multiple of 100 µsec. A value of 100, means 10 msec. Position actual value every 10 msec should not stall the bus.
Event timer is used to refresh the data. If the values of mapped objects do not change, asynchronous PDO would not be transmitted. To force a transmission within an defined time, event timer shall be used. It is defined in multiples of 1 msec. A value of 100, means that every 100 msec a TPDO is transmitted.

Bit number	Name	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

0 – 15	Inhibit time	Minimum time between every asynchronous TPDO transmission
16 – 31	Event timer	Maximum time between every asynchronous TPDO transmission

12.1.40. CANopen Lifetime

CANopen Lifetime is a Read/Write register, which is used to save in flash the node guarding configuration.
If the drive does not receive within Guard time x Life time factor milli-seconds the NMT guarding frame, the motor stops and communication is reset.

Bit number	Name	Description
0 – 15	Guard time	
16 – 31	Life time factor	

12.1.41. CANopen Producer heart-beat

CANopen producer heartbeat is a Read/Write register, which is used to save in flash the producer heartbeat configuration. The driver will transmit the heartbeat every producer heartbeat time.

Bit number	Name	Description
0 – 31	Prod Heartbeat 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

12.2. 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

12.2.1. Read Holding Register (03)

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

12.2.2. 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

12.2.3. 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

12.2.4. Write Single Holding Register (06)

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

12.2.5. 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

12.2.6. 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

12.2.7. 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.

12.2.8. 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

12.2.9. 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

12.2.10. 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.

12.2.11. 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

12.2.12. 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

12.2.13. 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];

                lIx++;
                len--;
        }

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

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

13. CANopen

13.1. Standards

FD1 and FD2 drives with suffix A and V8 firmware, implements following CANopen standards:

- CiA Draft Standard 301 V4.02
- CiA Draft Standard Proposal 402 V2.0

13.2. Communication objects

FD supports CAN standard frames with 11-bits identifier field. The default profile ID-allocation scheme consists of a functional part, which determines the object priority and a Node-ID-part, which allows to distinguish between devices of the same functionality. This allows a peer-to-peer communication between a single master device and up to 127 slave devices. It also supports the broadcasting of non-confirmed NMT and SYNC objects. Broadcasting is indicated by a Node-ID of zero.

Bit	10	9	8	7	6	5	4	3	2	1	0
COB-ID	Function Code				Node-ID						

Tab. 34 – COB-ID

CANOPEN_ADDRESS (Node-ID) can be programmed using DwLoader (FD1) or selected via DIP switches (FD2), depending on the FD type. Dynamic modifications of Node-ID are not supported.

Object	Function code	Resulting COB-ID
NMT	0000	0
SYNC	0001	128 (0x80)
EMERGENCY	0001	129 (0x81) – 255 (0xFF)
PDO1 (tx)	0011	385 (0x181) – 511 (0x1FF)
PDO1 (rx)	0100	513 (0x201) – 639 (0x27F)
PDO2 (tx)	0101	641 (0x281) – 767 (0x2FF)
PDO2 (rx)	0110	769 (0x301) – 895 (0x37F)
PDO3 (tx)	0111	897 (0x381) – 1023 (0x3FF)
PDO3 (rx)	1000	1025 (0x401) – 1151 (0x47F)
PDO4 (tx)	1001	1153 (0x481) – 1279 (0x4FF)
PDO4 (rx)	1010	1281 (0x501) – 1407 (0x57F)
SDO (tx)	1011	1409 (0x581) – 1535 (0x5FF)
SDO (rx)	1100	1537 (0x601) – 1663 (0x67F)

Tab. 35 – Communication 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

13.2.1. Network Management Objects (NMT)

The Network Management (NMT) is node oriented and follows a master-slave structure. NMT objects are used for executing NMT services, which refers to CAN network, i.e. the communication aspects. Through NMT services, nodes communication is initialized, started, monitored, reset or stopped. All nodes are regarded as NMT slaves. A NMT slave is uniquely identified in the network by its Node-ID, a value in the range of [1 – 127]. NMT requires that one device in the network fulfils the function of the NMT Master.

FD can have four NMT status:

0: Initializing,
4: Stopped,
5: Operational,
127: Pre-operational.

FD boots up in initializing status, when the initialization is complete it enters in pre-operational status and transmits the boot-up heart-beat.

COB-ID	RTR	DLC	Data 0
0x700 + NodeID	0	1	0

Tab. 36 – NMT boot-up

The drive status can be read using NMT node guard or using heartbeat mechanisms:

The guarding is achieved through transmitting guarding requests (Node guarding protocol) by the NMT Master.

COB-ID	RTR	DLC
0x700 + NodeID	1	0

Tab. 37 – NMT guarding request

The FD answers its status as:

COB-ID	RTR	DLC	Data 0
0x700 + NodeID	0	1	Toggle + status

Tab. 38 – NMT guarding answer

The only byte of data in slave answer is as follows:

- Bit 7 toggles (0 after communication reset),
- Bits 6 – 0 contains the NMT status

Life guarding is a service to be used for guarding the NMT master from the NMT slave. The slave uses the objects 0x100C, guard time multiplied by 0x100D, life time factor to calculate the node life time. If the NMT Slave is not guarded within its life time, the NMT Slave stops the motor, resets the CAN objects, NMT status becomes PRE-OPERATIONAL, drive status becomes SWITCHED ON.
If guard time and life time factor are 0 (default values), the NMT Slave does not guard the NMT Master. Guarding starts for the slave when the first remote-transmit-request for its guarding identifier is received with guard time and life time factor different than zero.

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 heartbeat protocol defines control of NMT status without need of remote frames. FD transmits a heartbeat message cyclically. One or more heartbeat consumer receive the indication. The relationship between producer and consumer is configurable via object 0x1017, producer heartbeat time.
If producer heartbeat time is zero, heartbeat is disabled. Otherwise it defines the period between each heartbeat production in milliseconds. The heartbeat message from the slave is as follows:

COB-ID	RTR	DLC	Data 0
0x700 + NodeID	0	1	Status

Tab. 39 – Heartbeat

Note:
It is not allowed for one device to use both error control mechanisms Guarding Protocol and Heartbeat Protocol at the same time. If the heartbeat producer time is different than 0 the heartbeat protocol is used only.

To modify the slave NMT status following services are implemented and always active:

- Start Remote Node: this service sets the state to operational.

COB-ID	RTR	DLC	Data 0	Data 1
0x0	0	2	0x1	NodeID

Tab. 40 – Start remote node

This service is unconfirmed (no answer from the slave).

- Stop Remote Node: this service sets the state to stop.

COB-ID	RTR	DLC	Data 0	Data 1
0x0	0	2	0x2	NodeID

Tab. 41 – Stop remote node

This service is unconfirmed (no answer from the slave).

- Enter Pre-Operational: this service sets the state to pre-operational.

COB-ID	RTR	DLC	Data 0	Data 1
0x0	0	2	0x80	NodeID

Tab. 42 – Enter pre-operational

This service is unconfirmed (no answer from the slave).

- Reset Node: this service sets the state from any state to the reset application sub-state. After completion of the service, the state of the selected remote nodes will be pre-operational.

COB-ID	RTR	DLC	Data 0	Data 1
0x0	0	2	0x81	NodeID

Tab. 43 – Reset node

- Reset Communication: this service sets the state from any state to the reset communication sub-state. After completion of the service, the state of the selected remote nodes will be PRE_OPERATIONAL.

COB-ID	RTR	DLC	Data 0	Data 1
0x0	0	2	0x82	NodeID

Tab. 44 – Reset communication

Note:

Reset and Reset Communication are theoretically unconfirmed, but a boot-up message is generated, because of the transition from initializing to pre-operational.

Objects	Initialization	Pre-operational	Operational	Stop
PDO			✓	
SDO		✓	✓	
SYNC		✓	✓	
EMCY		✓	✓	
NMT	✓	✓	✓	✓

Tab. 45 – Objects active per status

SDOs are active only in operational and pre-operational status.
PDOs are active only in operational status and they can be configured only in pre-operational.

13.2.2. Service Data Objects (SDO)

With Service Data Objects (SDOs) the access to all the entries of FD object dictionary is provided.

As these entries may contain data of various size and data type, SDOs can be used to transfer multiple data sets (each containing an arbitrary large block of data) from a client to a server and vice versa.
The client can control via a multiplexor (index and sub-index of the Object Dictionary) which data set is to be transferred. The contents of the data set are defined within the object dictionary.
Basically a SDO is transferred as a sequence of segments. Prior to transferring the segments there is an initialization phase where client and server prepare themselves for transferring the segments. For SDOs, it is also possible to transfer a data set of up to four bytes during the initialization phase. This mechanism is called an expedited transfer.

SDO Block Upload is not supported.

For all transfer types it is the client that takes the initiative for a transfer. The owner of the accessed object dictionary is the server of the SDO. Either the client or the server can take the initiative to abort the transfer of a SDO.
By means of a SDO a peer-to-peer communication channel between two devices is established.

13.2.3. Download SDO

The following services can be used:
- Initiate SDO Download
- Download SDO Segment

Through these services the client of a SDO downloads data to the server, i.e. the FD drive (owner of the object dictionary). The data, the multiplexor (index and sub-index) of the data set to be downloaded and its size (only optionally for segmented transfer) are indicated to the FD.
The service is confirmed. In case of a failure, the reason is confirmed with an Abort SDO message.
The SDO download consists of at least the Initiate SDO Download service and optional of Download SDO Segment services (for data length bigger than 4 bytes).

SDOs are downloaded as a sequence of zero or more Download SDO Segment services preceded by an Initiate SDO Download service. The sequence is terminated by:
- Initiate SDO Download request/indication with the e-bit set to 1 followed by an Initiate SDO Download response/confirm, indicating the successful completion of an expedited download sequence.
- Download SDO Segment response/confirm with the c-bit set to 1, indicating the successful completion of a normal download sequence.
- Abort SDO Transfer request/indication, indicating the unsuccessful completion of the download sequence.
- Initiate Domain Download request/indication, indicating the unsuccessful completion of the download sequence and the start of a new download sequence.

The initiate SDO Download request from the client is as follows:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	> 4	[7 – 5] 1 [4] 0 [3 – 2] n [1] e [0] s	Index		Sub-Index	d			

Tab. 46 – Initiate SDO Download request

DLC shall be at least 5.
Data 0:

- bits [7 – 5] indicate the command specifier, in this case it is 1.
- bit 1 is the transfer type:
  - 0: segmented,
  - 1: expedited.
- bit 0 is the size indicator. If it is equal to 1, the data set size is indicated. In case of expedited transfer it is indicated in n, otherwise in a segmented transfer it is indicated in Data [4 – 7].
- bits [3 – 2] only valid if e = 1 and s = 1, otherwise 0. If valid it indicates the number of bytes in d that do not contain data. Bytes [8-n, 7] do not contain data.

Data [1 – 3] Index and sub-index address the data into the object dictionary.
Data [4 – 7] depends upon the type of transfer.

- Expedited: contains the data to be downloaded. 4 – n Bytes if s = 1. Unspecified number if s = 0,
- Segmented: contains the number of bytes to be downloaded (Byte4 LSB, Byte 7 MSB).

In case of successful download FD will answer:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	8	0x60	Index		Sub-Index	0			

Tab. 472 – Initiate SDO download answer

The download SDO segment request from client is as follows:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	> 1	[7 – 5] 0 [4] t [3 – 1] n [0] c	Segmented data						

Tab. 48 – Download SDO segment request

DLC shall be at least 2.
Data 0:

- bits [7 – 5] indicate the command specifier, in this case it is 0.
- bit 4 is the toggle bit. This bit alternates for each subsequent segment that is downloaded. The first segment have the toggle bit 0. The toggle bit will be equal for the request and the response message.
- bits [3 – 1] n: indicates the number of bytes in segmented data that do not contain data. Bytes [8-n, 7] do not contain segment data. n = 0 if no segment size is indicated.
- bit 0 c: indicates whether there are still more segments to be downloaded:
  - 0: more segments to be downloaded,
  - 1: no more segments to be downloaded.

FD answers:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	8	[7 – 5] 1 [4] t [3 – 0] 0	0						

Tab. 49 – Download SDO segment answer

Note:
The segments are stored in a buffer before being written into the object dictionary. The buffer is 64 Bytes long. This is the limit of the maximum segmented transfer.

13.2.4. Upload SDO

The following services can be used:
- Initiate SDO Upload
- Upload SDO Segment

Through this service the client of a SDO uploads data from the server, i.e. reads from FD. The multiplexor (index and sub-index) of the data set that has to be uploaded is indicated to the server. The SDO upload consists of at least the Initiate SDO Upload service and optional of Upload SDO Segment services (data length > 4 bytes).
The service is confirmed. In case of a failure, an Abort SDO transfer request is executed. In case of success, the server has accepted the segment data and is ready to accept the next segment.
A successful Initiate SDO Download service with segmented transfer type must have been executed.

SDO are uploaded as a sequence of zero or more Upload SDO Segment services preceded by an Initiate SDO Upload service. The sequence is terminated by:
- Initiate SDO Upload response/confirm with the e-bit set to 1, indicating the successful completion of an expedited upload sequence.
- Upload SDO Segment response/confirm with the c-bit set to 1, indicating the successful completion of a normal upload sequence.
- Abort SDO Transfer request/indication, indicating the unsuccessful completion of the upload sequence.
- new Initiate SDO Upload request/indication, indicating the unsuccessful completion of the upload sequence and the start of a new sequence.

The initiate SDO Upload request from the client is as follows:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	8	0x40	Index		Sub-Index	0			

Tab. 50 – Initiate SDO upload request

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

FD will answer:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	> 4	[7 – 5] 2 [4] 0 [3 – 2] n [1] e [0] s	Index		Sub-Index	d			

Tab. 51 – Initiate SDO upload answer

DLC shall be at least 5.
Data 0:

- bits [7 – 5] indicate the command specifier, in this case it is 2.
- bit 1 is the transfer type:
  - 0: segmented,
  - 1: expedited.
- bit 0 is the size indicator. If it is equal to 1, the data set size is indicated. In case of expedited transfer it is indicated in n, otherwise in a segmented transfer it is indicated in Data [4 – 7].
- bits [3 – 2] only valid if e = 1 and s = 1, otherwise 0. If valid it indicates the number of bytes in d that do not contain data. Bytes [8-n, 7] do not contain data.

Data [1 – 3] Index and sub-index address the data into the object dictionary.
Data [4 – 7] depends upon the type of transfer.

- Expedited: contains the data uploaded. 4 – n Bytes if s = 1. Unspecified number if s = 0,
- Segmented: contains the number of bytes to be uploaded (Byte4 LSB, Byte 7 MSB).

If the FD answer with a segmented transfer, the client shall upload the segment using:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	8	[7 – 5] 3 [4] t [3 – 0] 0	Index		Sub-Index	0			

Tab. 52 – Upload SDO segment request

Data [0]:

- t: toggle bit. This bit alternate for each subsequent segment that is uploaded. The first segment have the toggle-bit set to 0.

FD answers:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	8	[7 – 5] 0 [4] t [3 – 1] n [0] c	Segmented data						

Tab. 53 – Upload SDO segment answer

Data [0]:

- t: toggle bit. The response message will have the same bit of request.
- n: indicates the number of bytes in seg-data that do not contain segment data. Bytes [8-n, 7] do not contain segment data. n = 0 if no segment size is indicated.
- c: indicates whether there are still more segments to be uploaded:
  - 0: more segments,
  - 1: no more segments.

Note:
FD always returns the toggle bit, which is received. It doesn't rise any alarm if it doesn't toggle.

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

13.2.5. Abort SDO

The abort SDO can be sent from the Client to stop a segmented transfer or from the FD in case of error.

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 or 0x580 + NodeID	0	8	[7 – 5] 4 [4 – 0] 0	Index		Sub-index	Abort code			

Tab. 543 – Abort SDO

Abort code	Description
0x05040001	Client/server command specifier not valid or unknown.
0x05040002	Invalid block size.
0x05040003	Invalid sequence number.
0x05040004	CRC error.
0x05040005	Out of memory
0x06010002	Attempt to write a read only object.
0x06020000	Object does not exist in the object dictionary.
0x06040041	Object cannot be mapped to the PDO.
0x06040042	The number and length of the objects to be mapped would exceed PDO length.
0x06040043	General parameter incompatibility reason.
0x06040047	General internal incompatibility.
0x06070010	Data type does not match, length of service parameter does not match
0x06090011	Sub-index does not exist.
0x06090030	Value range of parameter exceeded (only for write access).
0x06090031	Value of parameter written too high.
0x06090032	Value of parameter written too low.
0x08000000	General error

Tab. 554 – Abort codes

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

13.2.6. SDO block download

SDO can be transferred as a sequence of blocks where each block is a sequence of up to 127 segments containing a sequence number and the data.

The only objects that can be transferred in this way are the sub-indexes of 0x1F50, Download program data, which are firmware and parameters.

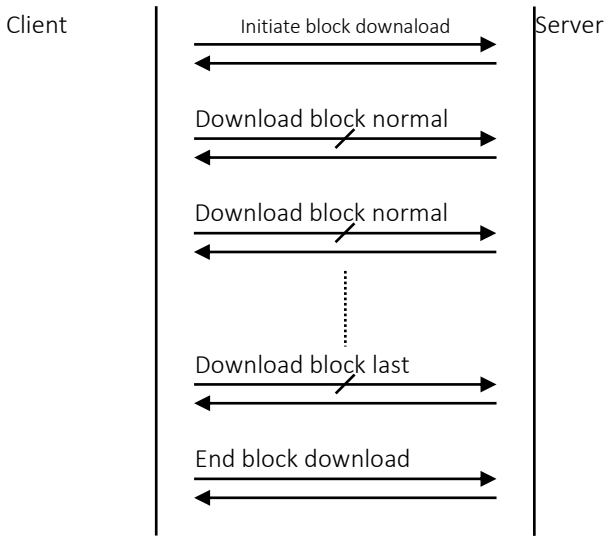
Prior to transferring the blocks there is an initialization phase where client and server prepare themselves for transferring the blocks and negotiating the number of segments in one block. After transferring the blocks there is a finalization phase where client and server can verify the correctness of the previous data transfer by comparing checksums derived from the data set.

After block download the server indicates the client the last successfully received segment of this block transfer by acknowledging this segment sequence number. Doing this the server implicitly acknowledges all segments preceding this segment. The client has to start the following block transfer with the retransmission of all not acknowledged data. Additionally the server has to indicate the number of segments per block for the next block transfer.

The SDO Download Block sequence is terminated by:

- a downloaded segment within a block with the c-bit set to 1, indicating the completion of the block download sequence.
- an 'Abort SDO Transfer' request/indication, indicating the unsuccessful completion of the download sequence.

The whole 'SDO Block Download' service is terminated with the End SDO Block Download service. If client has indicated the ability to generate a CRC during the Initiate SDO Block Download service, the server generates the CRC on the received data. If this CRC differs from the CRC generated by the client, the server indicates this with an 'Abort SDO Transfer' indication.



When the transmission of firmware and/or parameters has been successfully carried out, the client shall perform a NMT Reset command to jump back to main application.

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

Initiate SDO Block Download

Through this service the client requests the server to prepare for downloading data. FD drive will disable motor current and jump to program mode, i.e. the bootloader.

The client as well as the server indicate their ability and/or demand to verify the complete transfer with a checksum that will happen during End SDO Block Download.

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	> 4	[7 – 5] 6 [4 – 3] 0 [2] cc [1] s [0] 0	Index		Sub-Index	size			

Data 0:
- bits [7 – 5] indicate the client command specifier, in this case it is 6.
- bit 2 is the client CRC support:
  - 0: client does not support generating CRC on data,
  - 1: client supports generating CRC on data.
- bit 1 is the size indicator. If it is equal to 1, the data set size is indicated.

Data [1 – 3] Index and sub-index address the data into the object dictionary. Index can be only 0x1F50, sub index can be:
- sub-index 1: program firmware,
- sub-index 2: program parameters.

Data [4 – 7] download size in Bytes if s = 1.

FD answers:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	8	[7 – 5] 5 [4 – 3] 0 [2] sc [1 – 0] 0	Index		Sub-Index	blksize	0		

Data 0:
- bits [7 – 5] indicate the server command specifier, in this case 5
- bit 2 is the server CRC support = 1, because supported by FD drives.

Data 4: blksize, that is the number of segments per block with 0 < blksize < 128.

Download SDO Block

By this service the client supplies the data of the block to the server. The block data is transmitted to the server by a sequence of segments. Each segment consists of the data and a sequence number starting with 1 which is increased for each segment by 1 up to blksize. The parameter blksize is negotiated between server and client in the 'Initiate Block Download' protocol and can be changed by the server with each confirmation for a block transfer. The continue parameter indicates the server whether to stay in the 'Download Block' phase or to change in the 'End Download Block' phase.

The service is confirmed. In case of a success the ackseq parameter indicates the sequence number of the last segment the server has received successfully. If this number does not correspond with the sequence number of the last segment sent by the client during this block transfer the client has to retransmit all segments discarded by the server with the next block transfer. In case of a fatal failure, an Abort SDO Transfer request is executed. In case of success, the server has accepted all acknowledged segment data and is ready to accept the next block. There can be at most one Download SDO Block service outstanding for a SDO transfer.
A successful 'Initiate SDO Block Download' service must have been executed prior to this service.

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	8	[7] c [6 – 0] s	Segment 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

Data 0:

- bit 7: c, indicates whether there are still more segments to be downloaded
  - 0: more segments to be downloaded
  - 1: no more segments to be downloaded, enter End SDO block download phase
- Bits [6 – 0]: s, sequence number of segment 0 < s < 128.

FD answers:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	8	[7–5] 5 [1–0] 2	ackseq	blksize	0				

Data 0:

- bits [7-5] indicate the server command specifier, in this case 5
- bits [0-1] server subcommand, in this case 2

Data 1: ackseq: sequence number of last segment that was received successfully during the last block download. If ackseq is set to 0 the server indicates the client that the segment with the sequence number 1 was not received correctly and all segments have to be retransmitted by the client.
Data 2: blksize: Number of segments per block that has to be used by client for the following block download with 0 < blksize < 128.

End SDO Block Download

Through this service the SDO Block Download is concluded. The number of bytes not containing valid data in the last transmitted segments is indicated to the server.
If the server as well as the client have indicated their ability and demand to check the complete transfer with a checksum in 'Initiate SDO Block Download' this checksum is indicated to the server by the client. The server also has to generate a checksum which has to be compared with the one generated by the client.
The service is confirmed. The Remote Result parameter will indicate the success of the request (matching checksums between client and server if negotiated) and concludes the download of the data set. In case of a failure, an Abort SDO Transfer request must be executed.

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x600 + NodeID	0	8	[7-5] 6 [4-2] n [0] 1	CRC		0				

Byte 0:

- bits [7-5] is the client command specifier: 6 = block download
- bit 0 is the client subcommand: 1 = end block download request

Byte 1 and 2 are the CRC, Cyclic Redundancy Checksum. The check polynomial has the formula x^16 + x^12 + x^5 + 1. The calculation has to be made with an initial value of 0.

To confirm FD answers:

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x580 + NodeID	0	8	[7-5] 5 [0-1] 1	0						

13.2.7. Synchronization object (SYNC)

The Synchronization Object is broadcasted periodically by the SYNC producer, i.e. the master, and received from all the FD in the network. This SYNC provides the basic network clock. It is unconfirmed, with no data. There can be a time jitter in transmission by the SYNC producer corresponding approximately to the latency due to some other message being transmitted just before the SYNC.
In order to guarantee timely access to the CAN bus the SYNC is given a very high priority identifier, expressed in the object 0x1005, COB-ID SYNC.
Default SYNC COB-ID is 0x80.

13.2.8. Emergency object (EMCY)

Emergency objects are triggered by the occurrence of an alarm and they are transmitted from FD. An emergency object is transmitted only once per error event. As long as no new error occurs on a device no further emergency objects are transmitted.

COB-ID	RTR	DLC	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x80 + NodeID	0	8	Error code		Error register	0				

Tab. 565 – Emergency objects

Error code	Description
0x0000	No error
0x1000	Generic error
0x2300	Over current, on device output side
0x3210	Over voltage, on V_{POW}
0x3220	Under voltage, on V_{POW}
0x4310	Over temperature
0x6320	Data error
0x8100	Communication error (it does not generate EMCY)
0x8130	Life guard error
0x8611	Step loss error, following

Tab. 57 – Error codes

After initialization the FD enters the error free status. No error message is sent.
If FD goes in alarm an emergency object with the appropriate error code and error register is transmitted. The error code is also logged in the object 0x1003, pre-defined error field.
When the alarm is reset, an emergency message containing error code 0000 (Error reset) is transmitted.

13.2.9. Process Data Objects (PDO)

The real-time data transfer is performed by means of PDO. They are unconfirmed services used to write or read up to 8 Bytes without protocol overhead.
The PDOs correspond to entries in the object dictionary. Data type and mapping of objects into a PDO is determined by a corresponding PDO mapping structure within the dictionary. The mapping of objects into PDOs need to be transmitted during the PRE OPERATIONAL state by applying the SDO services to the PDO mapping objects (only Bytes or multiples can be mapped).
There are two kinds of use for PDOs. The first is data transmission and the second data reception. It is distinguished in Transmit-PDOs (TPDOs) and Receive-PDOs (RPDOs).

FD implements 4 RPDO and 4 TPDO.

The PDO communication parameter describes the communication capabilities of the PDO. The PDO mapping parameter contains information about the contents of the PDOs. The indices of the corresponding Object Dictionary entries are computed by the following formulas:

- RPDO communication parameter index = 1400h + RPDO-number-1
- TPDO communication parameter index = 1800h + TPDO-number-1
- RPDO mapping parameter index = 1600h + RPDO-number-1
- TPDO mapping parameter index = 1A00h + TPDO-number-1

The entries mentioned above are described below in Object Dictionary.

Remotely requested PDO are not implemented, as recommended by:

CiA 802 – Application note – CAN remote frames: Avoiding of usage

RPDOs default mapping is:

- RPDO 1 = *control word* 0x60400010,
- RPDO 2 = *control word* 0x60400010, *modes of operation* 0x60600008,
- RPDO 3 = *control word* 0x60400010, *target position* 0x607A0020,
- RPDO 4 = *control word* 0x60400010, *target velocity* 0x60FF0020.

TPDOs default mapping is:
- TPDO 1 = *status word* 0x60410010,
- TPDO 2 = *status word* 0x60410010, *modes of operation* 0x60600008,
- TPDO 3 = *status word* 0x60410010, *position actual value* 0x60630020,
- TPDO 4 = *status word* 0x60410010, *velocity actual value* 0x606C0020.

The structure of the mapping parameter, 8 Bytes is as follows:
- Byte 0 = object length (number of bits),
- Byte 1 = sub index,
- Bytes 2 and 3 = index.

For changing the PDO mapping first the PDO has to be deleted, the sub-index 0 must be set to 0 (mapping is deactivated), then the objects can be remapped. After all objects are mapped sub-index 0 has to be set to the valid number of mapped objects.

Finally, the PDO has to be created by setting the bit 32 in the PDO COB_ID parameter.
As only standard CAN frames are supported (Remotely requested PDO are not implemented), an attempt to set PDO COB_ID parameter bit 29 to 1 or bit 30 to 0 is responded with an abort message (abort code: 0x06090030).

The transmission type parameter of a PDO specifies the transmission mode as well as the triggering mode.
Synchronous TPDO:
- A transmission type of 0 means that the message shall be transmitted after occurrence of the SYNC but acyclic (not periodically), only if an event occurred before the SYNC.
- A transmission type of n means that the message is transmitted with every n-th SYNC object. (0 < n < 242).

Asynchronous TPDO:
- A transmission type of 254 or 255 means that the message is transmitted when the event occurs, without any relation to the SYNC.

Synchronous RPDO:
- A transmission type of n means that the message received after the occurrence of a SYNC is passed to the application with the occurrence of the following SYNC, independent of the transmission rate specified by the transmission type. (0 <= n <= 240).

Asynchronous RPDO:
- A transmission type of 254 or 255 means that the message is passed directly to the application.

Event based transmission means that the content of the PDO has a variation, i.e. only if there is a change in the content of the TPDO FD transmit it.

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

13.3. Object Dictionary

Index	Sub-index	bit	Type	Access	Object name	Description
0x1000	-	-	U32	RO	Device type	0x40192 > Stepper
0x1001	-	0	U8	RO	Error register	General alarm (always active in case of any alarm)
		1				Current error
		2				Voltage error
		3				Temperature error
		4				Communication error
		5				Data error
		7				Position error
0x1003	-	-	ARR	-	Pre-defined error field	
	0	-	U8	RW	Number of errors	Number of errors present into the array(write 0 to reset)
	1 – 8	-	U32	RO	Standard error fields	Error code (sub-ix 1 contains the most recent error)
0x1005	-	-	U32	RW	COB-ID SYNC	0x80
0x1008	-	-	STR	RO	Manufacturer device name	"FD-x"
0x1009	-	-	STR	RO	Manufacturer hardware version	"Vx.x"
0x100A	-	-	STR	RO	Manufacturer software version	"Vx.xx"
0x100C	-	-	U16	RW	Guard time	[millisec]
0x100D	-	-	U8	RW	Life time factor	
0x1010	-	-	ARR	-	Store parameters	
	0	-	U8	RO	Largest sub-index supported	1
	1	0	U32	RW	Save all parameters	Read: 1 = Device saves parameters on command
		1				Read: 0 = Device does not save parameters autonomously
		-				Write: 0x65766173 = signature to command the flash saving command
0x1014	-	-	U32	RW	COB-ID EMCY	0x80 + Node-ID
0x1017	-	-	U16	RW	Producer heartbeat time	Ref. to 14.2.1
0x1018	-	-	REC		Identity object	
	0	-	U8	RO	Number of entries	1
	1	-	U32	RO	Vendor ID	0x00001234
0x1400	-	-	REC	-	Pdo_1_Rx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_1_Rx_COB_ID	0x200
	2	-	U8	RW	Pdo_1_Rx_Type	0xFF
0x1401	-	-	REC	-	Pdo_2_Rx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_2_Rx_COB_ID	0x300
	2	-	U8	RW	Pdo_2_Rx_Type	0xFF
0x1402	-	-	REC	-	Pdo_3_Rx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_3_Rx_COB_ID	0x400
	2	-	U8	RW	Pdo_3_Rx_Type	0xFF
0x1403	-	-	REC	-	Pdo_4_Rx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_4_Rx_COB_ID	0x500
	2	-	U8	RW	Pdo_4_Rx_Type	0xFF
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

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
	1...7	-	U32	RW	Pdo_4_Rx_mapping_1...7	0x60400010; 0x60FF0020
0x1800	-	-	REC	-	Pdo_1_Tx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_1_Tx_COB_ID	0x180
	2	-	U8	RW	Pdo_1_Tx_Type	0
	3	-	U16	RW	Pdo_1_Tx_Inhibit_Time	
	5	-	U16	RW	Pdo_1_Tx_Event_Timer	
0x1801	-	-	REC	-	Pdo_2_Tx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_2_Tx_COB_ID	0x280
	2	-	U8	RW	Pdo_2_Tx_Type	0
	3	-	U16	RW	Pdo_2_Tx_Inhibit_Time	
	5	-	U16	RW	Pdo_2_Tx_Event_Timer	
0x1802	-	-	REC	-	Pdo_3_Tx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_3_Tx_COB_ID	0x380
	2	-	U8	RW	Pdo_3_Tx_Type	0
	3	-	U16	RW	Pdo_3_Tx_Inhibit_Time	
	5	-	U16	RW	Pdo_3_Tx_Event_Timer	
0x1803	-	-	REC	-	Pdo_4_Tx_Param_record	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Pdo_4_Tx_COB_ID	0x480
	2	-	U8	RW	Pdo_4_Tx_Type	0
	3	-	U16	RW	Pdo_4_Tx_Inhibit_Time	
	5	-	U16	RW	Pdo_4_Tx_Event_Timer	
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
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
0x1F50	-	-	ARR	-	Download program data	Ref. to 14.2.6
	0	-	DOM	WO	Program firmware	
	1	-	DOM	WO	Program parameters	
0x1F52	0	-	ARR	-	Verify application software	
	1	-	U32	RO	Application software date	Contains the number of days since January 1, 1984.
	2	-	U32	RO	Application software time	Contains the number of milliseconds after midnight
0x2003	-	-	U32	RW	Delta stop steps	Initialized to Cycle 0 delta stop steps
0x2005	[0-255]	-	ARR	RW	Modbus registers array 1	Ref. to 14.3.10
0x2006	[0-...]	-	ARR	RW	Modbus registers array 2	
0x603F	-	-	U16	RO	Error code	
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

Index	Sub-index	bit	Type	Access	Object name	Description
		12				Operation mode specific
0x6041	-	0	U16	RO	Status word	Ready to switch on
		1				Switched on
		2				Operation Enable
		3				Fault
		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	Ref. to 15.4.1.
0x6061	-	-	S8	RO	Modes of operation display	Same as modes of operation, but RO
0x6062	-	-	S32	RO	Position demand value	CURR_POSITION
0x6063	-	-	S32	RO	Position actual value	ENC_POS
0x6064	-	-	S32	RO	Position actual value2	ENC_POS
0x6065	-	-	U32	RW	Following error window	ACCUMULATION_LIMIT
0x6069	-	-	S32	RO	Velocity sensor actual value	ENC_SPEED signed
0x606B	-	-	S32	RO	Velocity demand value	CURR_SPEED signed
0x606C	-	-	S32	RO	Velocity actual value	ENC_SPEED signed
0x607A	-	-	S32	RW	Target position	[µstep]
0x607C	-	-	S32	RW	Home offset	[µstep]
0x607D			REC		Software position limit	
	0		U8	RO	Number of entries	[µstep]
	1		S32	RW	Min position limit	
	2		S32	RW	Max position limit	
0x607F	-	-	U32	RW	Max profile velocity	Upper limited to 300˙000 [µstep/sec]
0x6081	-	-	U32	RW	Profile velocity	Upper limited to max profile velocity [µstep/sec]
0x6083	-	-	U32	RW	Profile acceleration	[1˙000 µstep/sec^2]
0x6084	-	-	U32	RW	Profile deceleration	[1˙000 µstep/sec^2]
0x6086	-	-	S16	RW	Motion profile type	0: Linear, 1: Parabolic, 2: s-curve.
0x6098	-	-	S8	RW	Homing method	Ref. to 15.4.3
0x6099	-	-	REC	-	Homing speeds	
	0	-	U8	RO	Number of entries	2
	1		U32	RW	Homing speed research	[µstep/sec]
	2		U32	RW	Homing speed release	[µstep/sec]
0x609A	-	-	U32	RW	Homing acceleration	It sets acceleration and deceleration. Range [1- 20˙000] in [1˙000 µstep/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	
		0	U8	RO	Number of entries	2
		1	U8	RW	Interpolation time units	1
		2	S8	RW	Interpolation time index	-3
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

Index	Sub-index	bit	Type	Access	Object name	Description
		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				FD1: IN2, FD2: IN1/2
	-	17				FD1: IN3, FD2: IN3/4
	-	18				FD1: IN4, FD2: IN5
	-	19				FD1: IN5, FD2: IN6
	-	20				FD1: OUT6, FD2: IN7
	-	21				FD1: OUT7, FD2: IN8
	-	22				FD2: OUT9
	-	23				FD2: OUT10
0x60FE	-	-	REC		Digital outputs	
	0	-	U8	RO	Number of entries	
	1	-	U32	RW	Physical outputs	
	2	-	U32	RW	Bit mask	
0x60FF	-	-	S32	RW	Target velocity	Used in profile velocity mode [µstep/sec]
0x6402	-	-	U16	RO	Motor Type	9: µstepper motor
0x6502		0	U32	RO	Supported drive modes	Profile position mode = 1
		1				Velocity mode = 0
		2				Profile velocity mode = 1
		3				Profile torque mode = 0
		4				Reserved = 0
		5				Homing mode = 1
		6				Interpolated position mode = 1
0x6504	-	-	STR	RO	Drive manufacturer	
0x6505	-	-	STR	RO	Http drive catalog address	

Tab. 58 – Object dictionary

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

13.3.2. 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	Position error

Tab. 59 – Error register bits

13.3.3. 0x1003, Pre-defined error

The object at index 0x1003 holds information on 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 would 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).

13.3.4. 0x1005, COB-ID SYNC message

The object at index 0x1005 defines the COB-ID of the SYNC. Further, it defines additional information

Bit number	Value	Description
30	0	FD do not generate SYNC messages
29 – 11	0	Standard ID
10 – 0	Default 0x80	SYNC COB-ID

Tab. 59 – COB-ID Sync6

Bits 30 – 11 are not changeable, an attempt to set them is responded with an abort message (abort code: 0609 0030h).
Bit 31 is not used.

13.3.5. 0x100C / 0x100D, Guard time and Life time factor

The objects at index 100Ch and 100Dh include the guard time in milliseconds and the life time factor.
The life time factor multiplied with the guard time gives the life time for the Life Guarding Protocol. It is 0 if not used. Life guard gets active from the first NMT received since Guard time and Life time factor differs from zero.

In case of life guarding event (i.e. master didn't transmit NMT for the requested time), the slave stops all the movements and reset the object dictionary. Error code 0x8130 can be read in 0x1003, pre-defined error field.

13.3.6. 0x1010, Store parameters

The object at index 0x1010 is an array to be used to save all RAM drive parameters into flash.
Sub-index 0 is Read-Only U8. It contains the information about the largest sub-index supported, which is only one.

Sub-index 1 is Read/Write U32, which contains the device capability of saving when read:
b0 = 1: Device does saves parameters on command
b1 = 0: Device does not save parameters autonomously
b2-31 = don't care

When written with a specific signature, 0x65766173, all RAM drive parameters will be saved into flash. Any another value written generates an abort message (abort code: 0x08000020).

Flash parameters are organized in a table of the same dimension and address of Modbus registers table. When possible, CANopen parameters are initialized using such flash values.

13.3.7. 0x1014, COB-ID EMCY

The object at index 0x1014 defines the COB-ID of the EMCY object. Further, it defines additional information.

Bit number	Value	Description
31	0	0: EMCY object is activated 1: EMCY object is not active
30 – 11	0	Standard ID. Always 0
10 – 0	Default 0x80 + Node-ID	EMCY COB-ID

Tab. 60 – COB-ID Emergency7

Bits 30 – 11 are not changeable, an attempt to set them is responded with an abort message (abort code: 0609 0030h).
It is not allowed to change ID, while the object is activated (Bit 31 = 0).
By default, emergency object is activated.

13.3.8. 0x1017, Producer heartbeat time

The producer heartbeat time defines the cycle time of the heartbeat. The producer heartbeat time is 0 if it not used. The time has to be a multiple of 1ms.

13.3.9. 0x140x / 0x180x, RPDOx / TPDOx parameters

The object at index 0x140x defines the COB-ID and type of RPDOs 1, 2, 3 and 4.
The object at index 0x180x defines the COB-ID and type of TPDOs 1, 2, 3 and 4.

Sub-index 0 contains the number of entries, equal to 2, type U8.
Sub-index 1 contains the COB-ID of the PDO and its status, type U32:

Bit number	Value	Description
31	0	0: PDO is activated 1: PDO is not active
30	1	RTR not allowed
29 – 11	0	Standard ID
10 – 0	COB-ID	

Tab. 61 – PDO parameters

The PDO active/not active allows to select which PDOs are used in the operational state. There can be PDOs fully configured (e.g. by default) but not used, and therefore set to not active.
FD drives support the standard CAN frame type only and do not support Remote Frames, an attempt to set bit 29 to 1 or bit 30 to 0 is responded with an abort message (abort code: 0609 0030h).
It is not allowed to change bit 0-29 while the PDO is active (Bit 31 = 0).

Sub-index 2 contains the transmission type. Ref. to 14.2.8. Any attempt to set type higher than 240 or lower than 254 is responded with an abort message (abort code: 0609 0030h).

Sub-index 3 of TPDO parameters contains the inhibit time. Minimum time between asynchronous TPDO transmission.

Sub-index 5 of TPDO parameters contains the event timer. Maximum time between asynchronous TPDO transmission.

13.3.10. 0x160x / 0x180x, RPDOx / TPDOx Mapping record

The object at index 0x160x defines the mapping of RPDO 1, 2, 3 and 4.
The object at index 0x180x defines the mapping of TPDO 1, 2, 3 and 4.

For changing the PDO mapping first the PDO has to be deleted, the sub-index 0, number of object mapped, type U8, must be set to 0 (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 is set to the valid number of mapped objects.
Finally the PDO will be created by writing to its communication parameter COB-ID.

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.

13.3.11. 0x2003, Delta stop steps

Delta stop cycles (Ref. to 6.5) are available in profile position mode. Using this object it is possible to set the delta stop steps to be executed after sensor activation.

13.3.12. 0x2005 / 0x2006, Modbus registers

All the Modbus registers of chapter 13 have been mapped into two CANopen 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.
The object 0x2005, sub 0 is the number of entries, while 0x2005, sub 255 is FD1 IN4, FD2 IN3/4 config.
The object 0x2006, sub 0 is the number of entries, 0x2006, sub 1 is FD1 OUT6, FD2 OUT9 config.

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

13.3.13. 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. Ref. to 14.2.8 for the codes meaning.

13.3.14. 0x6040 / 0x6041, Control word and status word

0x6040, control word and 0x6041, status word is unsigned 16-bit objects that allow the master to command motor movements and check the status. Ref. to 14.4 for details.

13.3.15. 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 14.4.1 for details.

13.3.16. 0x6062 / 0x6063 / 0x6065, Positions

0x6062, position demand value and 0x6063, position actual value are a signed 32 bits read only object. The first one is the ordered position, while the second is the position feedback from the encoder.

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.

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.

When controlling FD in interpolated position mode, the master shall initialize its position counter with the object 0x6062, position demand value.

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

13.3.17. 0x6069 / 0x606B / 0x606C, Velocities

0x6069, 0x606B and 0x606C are all 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 application via Oscilloscope (ref. to 5.7).

13.3.18. 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 modes 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.

13.3.19. 0x607C, Home offset

0x607C, home offset is a signed 32 bits read write object, initialized to calibration position Modbus register. It represents the new value of 0x6062, position demand value at the end of homing methods.

13.3.20. 0x607D, Software position limits

Software position limit contains the sub-parameters *min position limit* and *max position limit*. These parameters define the absolute position limits for 0x6062, position demand value.
The limit positions are specified in µsteps. Writing the object determines its activation. They are not active in Homing cycles and when the drive is used in step/dir or quadrature step modes.

13.3.21. 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 µstep / sec.

13.3.22. 0x6081, Profile velocity

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

13.3.23. 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 kµstep/sec^2.

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

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

13.3.24. 0x6098 / 0x6099 / 0x609A, Homing method / speeds / acceleration

0x6098, homing method is a signed 32 bits read write object.

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

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 Modbus register. Its value is upper limited to 20'000 kμstep/sec^2 and used only during homing movements.

13.3.25. 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,
- 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. FD drives use instead just a single parameter, which is the position, but keep the record format defined from the standard.

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

13.3.1. 0x60F4, Following error actual value

It is a signed 32 bits read only register. It expresses the position deviation between the encoder feedback and the ordered position.

13.3.2. 0x60FD, Digital inputs

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

Bit number	FD1	FD2
0	Hardware limit switch down active	
1	Hardware limit switch up active	
2	Homing sensor active	
…		
16	IN_2	IN_1_2
17	IN_3	IN_3_4
18	IN_4	IN_5
19	IN_5	IN_6
20	OUT_6	IN_7

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

21	OUT_7	IN_8
22		OUT_9
23		OUT_10

Tab. 63 – 0x60FD, Digital inputs

13.3.3. 0x60FF, Target velocity

0x60FF, target velocity is signed 32 bits read write object, initialized to cycle 0 speed, which is the profile velocity mode speed set-point. Positive values creates movements towards increasing position, negative values towards decreasing positions. It is upper limited to 300˙000 µstep/sec and lower limited to -300˙000 µstep/sec.
Writing the object during profile velocity mode determines speed ramp up or down to 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

13.4. Device Control

The state machine, illustrated in Fig. 18, describes the device status and the possible control sequence 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.

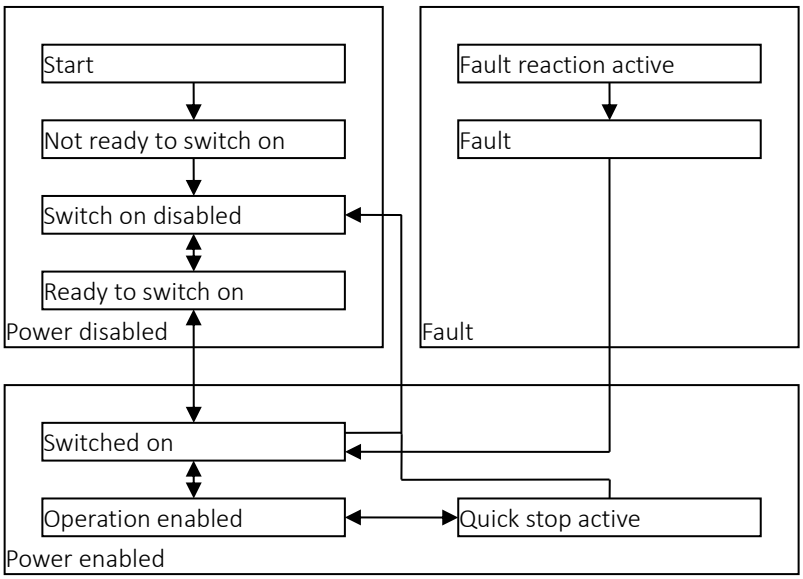


Fig. 18 – State machine

- In switch on disabled:
  - *shut down* command sets the state to *ready to switch on*. Current and frequency will be disabled.
- In 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.
- In 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.
- In 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.
- In 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.
- In fault:
  - *Fault reset* command sets the state to *switched on. Current will be enabled and frequency disabled.*

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	5	4
Profile position mode	Sub-mode	Sub-mode	Halt	Sub-mode	Change set immediately	New set point Start move
Profile velocity mode	-	-	Halt	-	-	-
Homing mode	-	-	Halt	-	-	Homing start
Position address mode	-	-	Halt	-	-	-
Interpolated position mode	-	-	-	-	-	Activate IP mode

Tab. 65 – Control word 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

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, 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.

13.4.1. Modes of operation

The object *modes of operation* (0x6060) is used to select the type of movement to be performed.

Values	Mode
0x00	Undefined
0x01	Profile position mode
0x03	Profile velocity mode
0x06	Homing mode
0x07	Interpolated position mode
0xFF	Position address mode

Tab. 68 – Operating modes

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

13.4.2. Profile Position Mode

This mode executes absolute indexer, relative indexer, delta stop, position delay or time delay cycles. The selection of the cycle type is made by the three bits 12, 11 and 6 *sub-mode* of *control word*.

To start the cycle control word bit 4 and status word bit 12 shall commutate as per below time graph. The positive edge of control word samples the movement data according to the sub-mode selected:

- *profile velocity* (0x6081),
- *max profile velocity (0x607F),*
- *profile acceleration* (0x6083),
- *profile deceleration* (0x6084),
- *control word* (0x6040) sub-mode bits,
- *target position* (0x607A),
- *delta stop steps (0x2003).*

FD confirms the sampling rising the status word bit 12, set point acknowledge. In this condition the negative edge of control word bit 12 starts the movement.

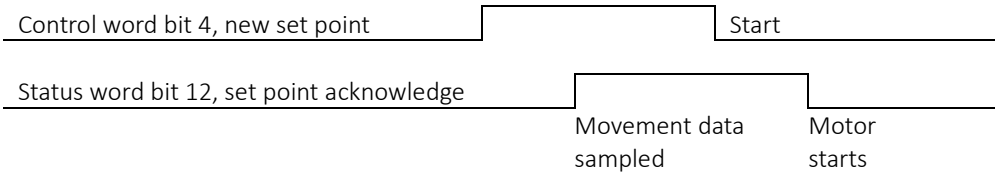


Fig. 19 – Profile position mode

If the change set immediately, bit 5 of the controlword, is set to 0, 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 next ordered 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). 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.

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

Sub-mode indicates if profile position movement is absolute, relative, delta stop, position delay or time delay:

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

If the change set immediately, bit 5 of the controlword, is set to 1, the drive will execute a single setpoint. Doing so, no FIFO buffer of movements is implemented. If a the new setpoint, bit 4 of the control word, is raised with change set immediately, bit 5 of the control word, at 1 while the motor is running, the drive will immediately adapt speed and direction to follow the new setpoint. Doing so it is possible to continuously interrupt current movement and order new set-points.

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

13.4.3. Homing mode

This mode is used to seek the home position. It is possible to specify the speeds, acceleration and the method of homing. There is a further object *home offset*, which allows to displace zero from the home position.
There are two *homing speeds*; in a typical cycle the faster speed is used to find the home switch and the slower speed is used to perform the following marker cycle or to release from the home switch.

The homing movement starts upon a positive edge of *control word* bit 4 *homing start*. Upon edge detection FD sets also acceleration and deceleration equal to *homing acceleration*. The movement stops in case of negative edges of the same bit or using the Halt bit.

Movement is completed when *homing attained*, *status word* bit 12 is set.

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,
- LS and HS define if the calibration is performed on a homing switch or limit switch. Multipurpose inputs need to be configured accordingly.
- NO and NC are normally-opened or normally-closed contact of the switch.
- MK is the marker cycle, i.e. a motor rotation to the single-turn zero encoder. An offset to the single turn zero-encoder can be configured by 0x2005:09, position offset.
- MECH is a movement towards mechanical stop.

Homing methods	Values	Description
NEAREST_MK	-33	Motor will rotate in the direction of the single-turn zero encoder following the shortest path.
UP_MECH	-18	Motor will rotate at *homing speed research* CW towards increasing positions. When the drive detects that the motor shaft is mechanically stopped, calibration is completed.
DW_MECH	-17	Motor will rotate at *homing speed research* CCW towards decreasing positions. When the drive detects that the motor shaft is mechanically stopped, homing is completed.
UP_MECH_MK	-2	Motor will rotate at *homing speed research* CW towards increasing positions. When the drive detects that the motor shaft is mechanically stopped, marker cycle in CCW direction is executed. At the end of marker cycle, homing is completed.
DW_MECH_MK	-1	Motor will rotate at *homing speed research* CCW towards decreasing positions. When the drive detects that the motor shaft is mechanically stopped, marker cycle in CW direction is executed. At the end of marker cycle, homing is completed.
DW_LS_NO_MK	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). It will decelerate to zero speed, wait *time stop* milliseconds and accelerate in opposite direction at *homing speed release* till the negative edge of limit switch signal is met and then it continues moving at the same direction and speed till the marker cycle is completed.
UP_LS_NO_MK	2	Motor will rotate at *homing speed research* CW towards increasing positions till a positive edge of hardware limit switch up input signal is met (at least one multipurpose input needs to be configured as limit switch up, i.e. input configuration register equals to 5). It will decelerate to zero speed, wait *time stop* milliseconds and accelerate in opposite direction at *homing speed release* till the negative edge of limit switch signal is met and then

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 continues moving at the same direction and speed till the marker cycle is completed.
UP_HS_NO_MK	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). 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_NC_MK	4	Motor will rotate at *homing speed research* CW towards increasing positions till a negative 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 3). It decelerates to zero speed, waits Time stop milliseconds and accelerates in opposite direction at *homing speed release* till the positive edge of the same signal is met and then it continues moving at the same direction and speed till the marker cycle is completed.
DW_HS_NO_MK	5	Motor will rotate at *homing speed research* CCW towards decreasing 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). 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.
DW_HS_NC_MK	6	Motor will rotate at *homing speed research* CCW towards decreasing positions till a negative 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 3). It decelerates to zero speed, waits Time stop milliseconds and accelerates in opposite direction at *homing speed release* till the positive edge of the same signal is met and then it continues moving at the same direction and speed till the marker cycle is completed.
-	-	
DW_LS_NO	17	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). It will decelerate to zero speed, wait *time stop* milliseconds and accelerate in opposite direction at *homing speed release* till the negative edge of limit switch signal is met.
UP_LS_NO	18	Motor will rotate at *homing speed research* CW towards increasing positions till a positive edge of hardware limit switch up input signal is met (at least one multipurpose input needs to be configured as limit switch up, i.e. input configuration register equals to 5). It will decelerate to zero speed, wait *time stop* milliseconds and accelerate in opposite direction at *homing speed release* till the negative edge of limit switch signal is met.
UP_HS_NO	19	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). 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.

UP_HS_NC	20	Motor will rotate at *homing speed research* CW towards increasing positions till a negative 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 3). It decelerates to zero speed, waits Time stop milliseconds and accelerates in opposite direction at *homing speed release* till the positive edge of the same signal is met.
DW_HS_NO	21	Motor will rotate at *homing speed research* CCW towards decreasing 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). 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.
DW_HS_NC	22	Motor will rotate at *homing speed research* CCW towards decreasing positions till a negative 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 3). It decelerates to zero speed, waits Time stop milliseconds and accelerates in opposite direction at *homing speed release* till the positive edge of the same signal is met.
-	-	
DW_MK	33	Marker cycle CCW towards decreasing positions.
UP_MK	34	Marker cycle CW towards increasing positions.
CURR_POSITION	35	Current position is set 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. Homing attained has the same meaning of Modbus status word bit axle calibrated, which can be restored from power on, ref. to 11.

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

13.4.4. Profile Velocity Mode

When entering to profile velocity mode the following movement data are sampled and if the operation is enabled the movement is directly started:

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

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 corresponds to counter-clockwise movements, decreasing the position counter value.

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

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

13.4.5. Interpolated position mode

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

The *interpolation data record* contains the interpolated positon 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. For each interpolation cycle, the drive calculates the position demand values by interpolating positions over the period of time.

There are no limit functions for speed, acceleration and deceleration applied to the interpolation data.

If the drive cannot execute the ordered command step accumulation limit alarm arises.

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

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.

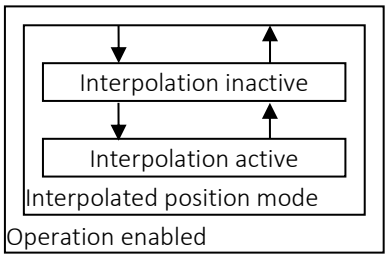


Fig. 20 – Interpolated position mode states

Interpolation inactive: this state is entered when the device is in state OPERATION ENABLE and the interpolated position mode is selected. The drive unit will accept input positions and will buffer it for interpolation calculations, but it does not move the motor.

Interpolation active: this state is entered when the device is in state OPERATIION ENABLE, the interpolated position mode is selected and enabled. The drive unit will accept input data and it moves the motor.

Entering interpolated position mode or operation enabled clears the buffer.

After the interpolation data record position is written, it is automatically added to 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.

The actual interpolation position is applied immediately, i.e. the drive will move the motor to be at that position at the next synchronization signal. An input buffer for interpolation data records is not mandatory, although it eases the data exchange between a host and a drive unit. The real-time requirements to the CAN-bus as well as to the drive unit decrease in this case, because an 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 different positions for each set of coordinates which have to be reached at sync instants and transmits them to the different axles. For each set-point FD calculates the positions in between. Each FD gets a set of positions to be processed internally independent from other axles.

Note:
Before entering IP mode, make sure that the first position transmitted is the same of current FD 0x6062, position demand value. This position is valid only if the drive is switched on.

Common PDO mapping for IP mode is a RPDO made of Control word + Interpolation data record sub-index 1 and a TPDO made of Status word + position actual value.

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

13.4.6. Position address mode

This is a manufacturer specific mode, which is used to broadcast with a single PDO several commands to multiple drives. Position address record contains four 32 bits registers at sub-indexes 1 – 4. Drive at address n checks only the nth bit of position address register.

If this bit is zero, the motor will move to 0x2001, *position 0*, while if it is one it will move to 0x2002, *position 1*.

The type of movement to be performed is specified in 0x2004, *modes of position address*.

Values	Mode
0x00	Absolute
0x01	Relative
0x02	Delta stop
0x03	Marker CW
0x04	Marker CCW
0x05	Marker shortest distance

Tab. 71 – Position address sub-modes

Note:
In case of relative or delta stop movement position 0 and 1 determines the maximum distance and the direction. Delta stop steps are the steps executed after input activation.

13.5. Device Initialization

CANopen standard is conceived to have a NMT master which initialize the network configuring all device parameters, including communication parameters via SDO transfers. This chapter wants to describe a typical initialization phase for using node 0x0D in interpolated position mode.

At power on the drive is NMT status = PRE_OPERATIONAL.

In the following tables, the first column M/S identify the master or slave originator of the message

13.5.1. Phase 1: upload of status

SDO upload 0x6041, status word

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	40	41	60	0	0	0	0	0
S	58D	6	4B	41	60	0	33	2		

Slave answer Bytes 4 and 5 contain the status of the drive, which in this case is switched-on.

SDO upload 0x6062, position demand value

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	40	62	60	0	0	0	0	0
S	58D	8	43	62	60	0	B1	DB	3	0

Slave answer (0x3DBB10) shall be used to initialize master positions generator.

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

13.5.2. Phase 2: TPDO1

SDO Download 0x1800, sub-index 1, TPDO1 disabled (B7 = 0xC0)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	18	1	8D	1	0	C0
S	58D	8	60	0	18	1	0	0	0	0

SDO Download 0x1800 sub-index 2, TPDO1 type synchronous (B4 = 0x01)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	0	18	2	1	0	0	0
S	58D	8	60	0	18	2	0	0	0	0

The slave will transmit the TPDO at every occurrence of the sync. Setting B4 = 2, the slave would transmit the TPDO every two sync. In case of B4 = 0, TPDO would by synchronous and acyclic, which means TPDO would be transmitted at every sync only if there is a change in its content. B4 = 255 means asynchronous, i.e. transmitted every time there is a change in TPDO content, which cannot be used for position because it would overload the bus.

SDO Download 0x1A00 sub-index 0, TPDO1 delete mapping

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	0	1A	0	0	0	0	0
S	58D	8	60	0	1A	0	0	0	0	0

Mapping needs to be deleted before adding new objects.

SDO Download 0x1A00 sub-index 1, TPDO1 status word mapping at first 2 Bytes

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	1A	1	10	0	41	60
S	58D	8	60	0	1A	1	0	0	0	0

SDO Download 0x1A00 sub-index 2, TPDO1 position actual value mapping on next 4 Bytes

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	1A	2	20	0	64	60
S	58D	8	60	0	1A	2	0	0	0	0

SDO Download 0x1A00 sub-index 0, TPDO1 create mapping of two objects (B4 = 2)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	0	1A	0	2	0	0	0
S	58D	8	60	0	1A	0	0	0	0	0

Since 6 Bytes in total have been used, 2 Bytes remain free.

SDO Download 0x1800 sub-index 1, TPDO1 enabled (B7 = 0x40)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	18	1	8D	1	0	40
S	58D	8	60	0	18	1	0	0	0	0

COB-ID = 0x18D

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

13.5.3. Phase 3: RPDO1

SDO Download 0x1400 sub-index 1, RPDO1 disabled (B7 = 0xC0)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	14	1	D	2	0	C0
S	58D	8	60	0	14	1	0	0	0	0

SDO Download 0x1400 sub-index 2, RPDO1 type syncronous

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	0	14	2	1	0	0	0
S	58D	8	60	0	14	2	0	0	0	0

B4 = 1 sets the PDO type syncronous, i.e. the effect of the PDO is after the SYNC reception. With type 255, asynchronous, the effect is immediate. In this case, since the position goes into the interpolation buffer, also type 255 would be correct.

SDO Download 0x1600 sub-index 0, RPDO1 delete mapping

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	0	16	0	0	0	0	0
S	58D	8	60	0	16	0	0	0	0	0

SDO Download 0x1600 sub-index 1, RPDO1 control word mapping at first 2 Bytes

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	16	1	10	0	40	60
S	58D	8	60	0	16	1	0	0	0	0

SDO Download 0x1600 sub-index 2, RPDO1 interpolated position setpoint mapping at next 4 Bytes (0x60C1 sub-index 1)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	16	2	20	1	C1	60
S	58D	8	60	0	16	2	0	0	0	0

SDO Download 0x1600 sub-index 0, RPDO1 creation mapping of two objects (B4 = 2)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	0	16	0	2	0	0	0
S	58D	8	60	0	16	0	0	0	0	0

SDO Download 0x1400 sub-index 1, RPDO1 enabled (B7 = 0x40)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	23	0	14	1	D	2	0	40
S	58D	8	60	0	14	1	0	0	0	0

COB-ID = 0x20D

Two or more drives can be set with the RPDO with the same COB-ID.

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

13.5.4. Phase 4: interpolated position mode

SDO Download 0x60C2 sub-index 1, time units, (B4 = 4 msec, it represents SYNC period)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	C2	60	1	4	0	0	0
S	58D	8	60	C2	60	1	0	0	0	0

SDO Download 0x6060, modes of operation = 0x07 (Interpolated position mode)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	60D	8	2F	60	60	0	7	0	0	0
S	58D	8	60	60	60	0	0	0	0	0

13.5.5. Phase 5: NMT

In this phase it is possible to configure also the lifeguarding, optional, this is used for the slave to control the master (when no NMT message is received within a time-out, the slave disables the operation).

NMT Start remote node.

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	0	2	1	D						

Hereafter PDO can be transmitted and received.

13.5.6. Phase 6: SYNC e PDO every 4 msec

Through PDO now it is possible control the drive to activate the interpolation and monitor its feedback using TPDO1 and RPDO1 where the status word and the control word have been mapped.

SYNC

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	80	0								

TPDO1 (B0, B1: status word = operation disabled, B2, B3, B4, B5: position actual value)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
S	18D	6	33	2	B1	DB	3	0		

RPDO1 (B0, B1: control word = enable operation, B2, B3, B4, B5: interpolated position set-point)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	20D	6	F	0	B1	DB	3	0		

SYNC

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	80	0								

TPDO1 (B0, B1: status word = operation enabled, B2, B3, B4, B5: position actual value)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
S	18D	6	37	2	B1	DB	3	0		

RPDO1 (B0, B1: control word = activate interpolation, B2, B3, B4, B5: interpolated position set-point)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	20D	6	1F	0	B1	DB	3	0		

SYNC

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	80	0								

TPDO1 (B0, B1: status word = interpolation active, B2, B3, B4, B5: position actual value)

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
S	18D	6	37	12	B1	DB	3	0		

RPDO1 (B0, B1: control word = activate interpolation, B2, B3, B4, B5: interpolated position 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

	COB-ID	DLC	B0	B1	B2	B3	B4	B5	B6	B7
M	20D	6	1F	0	B1	DB	3	0		

Hereafter, the interpolation master can change continuously the interpolated position set-point mapped on RPDO1 Bytes 2, 3, 4 e 5 and will observe the effective movements on TPDO1 Bytes 2, 3, 4 e 5.

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