XD-OEM Controller manual – 3.0

The XD-OEM is a versatile industrial piezo controller, supporting a wide variety of use cases and communication options for our actuators and stages. Designed for OEM integration, both single-axis and multi-axis (through EtherCAT) is supported. A Windows GUI is included for plug-and play testing.

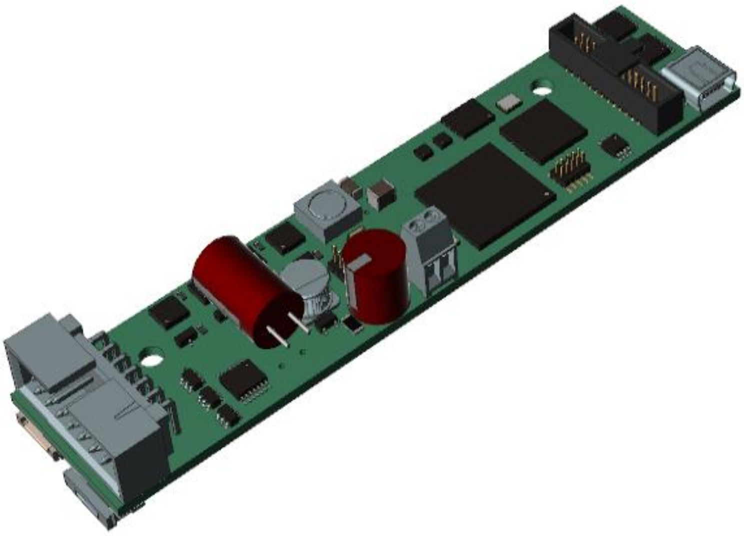


Image for illustration only. May not be an exact representation of the product.

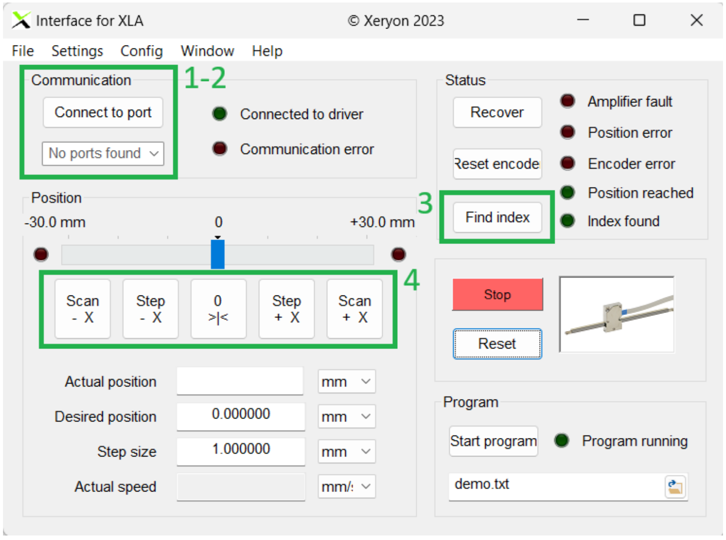
1. Key specifications

Compatible products	XLA series, XLS series, XRT series
Number of axes	Single-axis (multi-axis through EtherCAT, ch. 7)
Power supply	12 – 48 VDC (screw terminal), 1-5W
Power supply (XLA-10 only)	48V VDC (screw terminal), max. 10W
Temperature range	-30°C to +70°C
Dimensions	111 x 25 x 13.2 mm
Connector to actuator / stage	ZIF | Molex
Control types	Closed loop, open loop, hybrid
Key communication options	USB, UART, EtherCAT, GPIO, …
Software support	Windows GUI, C++, Python, TwinCAT, LABView
Motion profile	Point-to-point trapezoidal
Encoder compatibility	Optical, incremental
Supported resolutions	Up to 78 nm (linear), up to 3 µrad (rotary)
Error detection	Overcurrent, thermal, short-circuit, …

2. Content

3. Getting started

All controllers are delivered with a power adapter, USB cable and USB stick with software to control actuators and stages (from now on: 'actuator(s)') for easy plug-and-play use. To get started, plug the actuator connector into the XD-OEM. Connect the XD-OEM to a computer with the provided USB cable and open up the Windows GUI, press SCAN or MOVE and the actuator will start moving.



A quick overview of the Windows GUI:

To control an actuator:
1. Connect the correct COM port.
2. Load *settings_default.txt* from USB stick.
3. Press "Find Index".
The actuator will start moving and look for the encoder index.
4. You can start controlling the actuator:
a. SCAN: move with continuous speed.
b. STEP: move with specified step size.
c. 0: move to home position.

Using the XD-OEM controller, a Xeryon actuator can be used to create a wide variety of motion patterns, such as:

- Short- or long-range motion, high- or low-speed motion
- Repetitive or dynamic motion, smooth continuous motion
- Set-and-forget applications, controlled damping (soft landing) motion
- A combination of the above, and more

Tuning

Xeryon actuators work based on ultrasonic piezo technology. This technology is frequency-based, and differs slightly from both a) electromagnetic motors and b) other piezo systems. As a result, motion characteristics may differ from conventional motion systems. Several parameters can be used for finetuning motion behaviour. Motion trajectory is split into two zones: bulk move (or 'scanning') and final positioning. When experiencing unexpected behaviour, the following simple tuning procedures can be followed (Settings -> Edit). In-depth tuning info can be found in chapter 9.

Behaviour	**Final positioning (zone 1)**	**Bulk move / scanning (zone 2)**
Vibration	Lower proportional factor	Lower proportional factor
Noise	Adjust frequency	Adjust frequency
Overshoot	Lower deacceleration	Lower deacceleration
Force too low	-	Lower frequency
Inaccurate landing	Lower positioning tolerance	-
Slow landing	Adjust proportional factor	-

Note: for vertical or inclined use with higher payloads, advanced tuning may be necessary. For assistance, please contact support@xeryon.com. More information about the Windows GUI: ch. 9.

4. Control of an actuator

The controllers can be controlled using various methods:

- Using USB / UART — see section 5
- Using GPIO — see section 6
- Using EtherCAT — see section 7

For nanosecond multi-axis use, EtherCAT is recommended. For fast microsecond pulses: GPIO.

5. Communication using USB / UART

A host computer or controller can communicate with the XD-OEM via the USB configured as a virtual COM port. The baud rate is automatically detected by the controller and can be up to 115200. The protocol uses 8 data bits, 1 stop bit, no parity bit, no handshaking. This section consists of three parts:

- 5.1 Format
- 5.2 List of **commands**
- 5.3 Feedback from controller

Note: To enable UART, send UART=9600 command and use the following pins:

Ground (pin 2), 5V reference (5), Tx (12), Rx (18).

5.1. Format for instructions

A command line consists of maximum 16 characters followed by a 'new line' character (ASCII code 10). The command has the following fixed format:

X:DPOS=-12345678

- 1 character defining the axis, followed by a colon.
- 4 characters for the command.
- '=' sign separating the command from the corresponding value.
- Optional sign.
- Decimal value up to 8 decimal places (9 if the sign is omitted).
- Maximum total of 16 characters.

The characters have to be sent from left to right, in the example above starting with 'X' and ending with '8'. The command tags are in upper case. The instruction should be terminated with a 'new line' character (ASCII code 10). The driver processes the instruction immediately after receiving this 'new line' character.

Some instructions such as 'ZERO' and 'RSET' require no value. In that case, it is sufficient to send only the command itself, e.g. 'ZERO' followed by the 'new line' character.

Value range

There are 9 characters reserved for the value including its sign. For signed values 8 decimal places are available, giving a range from -99 999 999 to +99 999 999. For positive numbers, the '+' sign can be omitted, increasing the positive range to 999 999 999. No spaces, commas or periods should be added to the numbers. Only integers are allowed.

- X:DPOS=-99999999
- X:DPOS=+99999999
- X:DPOS=999999999

Request a value

To request the value of a certain setting, put '=?' after the parameter for which you want to know the value, e.g. EPOS=? gives the controller a request for the encoder position. FREQ=? asks the controller for the current excitation frequency. This works best with INFO=0, otherwise the reply disappears in the constant flood of feedback data.

Units are as follows:

Type	XLA / XLS series	XRT series	Resolution
Time, delays	ms		1 ms
Target position, step size	encoder units		1 encoder increment
Speed	µm/s	deg/s (*)	1 µm/s or 0.01 deg/s
Frequency	Hz		1 Hz

(*) Conversion factor of 100 required: e.g. enter SSPD=10000 for 100 deg/s.

5.2. List of commands

This is a list of all possible commands that can be sent and their use. This part is divided into:

I. Motion
II. Settings handling
III. Communication
IV. Tuning
V. Signal (advanced)
VI. Directional settings (advanced)
VII. Trigger outputs (advanced)

I.	Motion commands	
Command	Range	Explanation
INDX	0, 1	Find the index. A value of 0 or 1 indicates the initial search direction. The controller sets off in the specified direction to search. When a mechanical limit is reached (detected by position error > ILIM) it reverses the search direction. After finding index, actuator is positioned at the index position.
INDA	1 bit	Automatic detection of encoder index.
HOME	-	Go to the home position. This equals DPOS=0.
DPOS	26 bits	Set target position. Closed-loop control is used to reach and maintain the new position. The position is expressed in encoder units. Positive and negative values are allowed within the range of the actuator.
STEP	26 bits	Move relative to the current position, over a specified distance. When already in closed loop, the current desired position is used as a reference. When before in open loop, the actual position (encoder value) is used as a reference. The command value specifies the step size in encoder increments. Positive values send the actuator towards higher encoder values, negative values send the actuator towards lower encoder values. Closed-loop control is used to reach and maintain the new position.
SCAN	-1,0,1	Continuously move with fixed speed. The speed is maintained by closed-loop control. A positive number sends the actuator towards increasing encoder values, a negative number sends the actuator towards decreasing encoder values. A zero value stops the actuator.
SSPD	24 bits	Set speed. Used as scanning speed (SCAN command) and as target speed towards the next target position (DPOS and STEP). Unit is 1 µm/s or 0.01 deg/s. Default: 10000 (10 mm/s or 100 deg/s).
ISPD	24 bits	Set the speed which is used while searching the index. Unit is 1 µm/s or 0.01 deg/s.
ACCE	16 bits	Set acceleration for speed profile. Expressed in m/s2. Default value: 65500.
DECE	16 bits	Set deceleration for speed profile, when approaching target position. Default / maximum value: 65500 for XD-OEM controller.
LLIM	26 bits	Set low-side soft end stop. Expressed in encoder units.

HLIM	26 bits	Set high-side soft end stop. Expressed in encoder units.
MOVE	-1,0,1	Continuously move in open loop. Phase and amplitude influence the speed, but speed is not controlled. A positive number sends the actuator towards increasing encoder values, a negative number sends the actuator towards decreasing encoder values. A zero value stops the actuator.
PHAS	16 bits	Set the phase offset between the excitation signals. Can be used to control the speed in open loop. Input values 0-65535 correspond to a phase shift of 0-360°. Around 16384 the phase corresponds with a MOVE=1 direction, around 49152 (=-16384) it corresponds to a MOVE=-1 direction.
AMPL	16 bits	Set amplitude for open-loop piezo excitation signals.
STOP	-	Stop the actuator.
CONT	-	Continue movement after a stop command.
ENBL	1 bit	Enable drive. Or: If an actuator is in error mode, you can recover from that by sending "ENBL=1". This enables the actuator again.
ENBR	1 bit	Enable drive on power-up automatically.
FILE		Filters fast variations from the encoder values (noise). Default value: 0, sets filter to a minimum.
FILG		Extracts the spikes from the raw encoder inputs. Defines the minimum pulse width that may be responded to. Default value: 0
INTF		Controller integration factor. Default value: 1. Minimum = 1
DTIM		Determines how long (in ms) the motor must be turned off before it can be turned on again. Ensures that the actuator is not over-excited when landing by switching it on and off too quickly. Default value: 0

II.	**Manage settings**	
Command	Range	Explanation
RSET	-	Reset the driver. All piezo signals go to zero and settings are reset to their saved value.
LOAD	-	Load settings from memory.
SAVE	-	Save settings to memory.
BLCK	0-1	Blocked mode. When blocking mode is enabled (1), the ENBL command should be send to the controller if an error occurs. If blocking mode is disabled (0), the controllers executes the next command even when an error occurs.
TEST	0-1	Test LED indicators. TEST=1 switches all indicators on. TEST=0 brings them back to their function.

<table>
<tr><th colspan="3">III. Communication</th></tr>
<tr><th>Command</th><th>Range</th><th>Explanation</th></tr>
<tr><td>INFO</td><td>4 bits</td><td>Select type of info to be transmitted from the controller to the master (PC).

0: Stop broadcasting info automatically.
1: SRNO, SOFT, MODEL, STAT, SYNC
2: SRNO, SOFT, MODEL, STAT, FREQ, SYNC, EPOS, DPOS, REQUESTED PARAMETER*, TIME
3: EPOS, DPOS, STAT
4: EPOS, STAT, DPOS, TIME
5: STAT, FREQ, EPOS, DPOS, REQUESTED PARAMETER*, TIME
6: REQUESTED PARAMETER*
7: EPOS, STAT

Default: 2

e.g. INFO=7 will alternatingly send EPOS & STAT values.

* REQUESTED VALUE, when a parameter value is requested (e.g. by sending FREQ=?) the value for the parameter will be returned here.</td></tr>
<tr><td>UART</td><td>0-76800</td><td>Set UART baud rate. To switch UART off: UART=0.To set the baud rate to 9600: UART=9600. The maximum baud rate is 76800. When UART is off, the UART can only be restarted by sending the UART command via USB.</td></tr>
<tr><td>GPIO</td><td>0-13</td><td>Select the preferred input mode using the GPIO command.
For more details see section 5.
<table>
<tr><td>GPIO=0</td><td>Control via IO pins switched off. The controller will only react to text commands sent through USB or UART.</td></tr>
<tr><td>GPIO=2</td><td>Pulse and direction mode, with direction & enable pins.</td></tr>
<tr><td>GPIO=3</td><td>Pulse and direction mode, with forward & backward pins.</td></tr>
<tr><td>GPIO=4</td><td>A quad B input mode.</td></tr>
<tr><td>GPIO=8</td><td>PWM control, with direction & enable pins.</td></tr>
<tr><td>GPIO=9</td><td>PWM control, with forward & backward pins.</td></tr>
<tr><td>GPIO=12</td><td>Analog control, with direction & enable pins.</td></tr>
<tr><td>GPIO=13</td><td>Analog control, with forward & backward pins.</td></tr>
</table></td></tr>
<tr><td>STPS</td><td>16 bits</td><td>Select the step size for each STEP pulse. The step size is expressed in encoder units. Default: 1 encoder unit.</td></tr>
<tr><td>POLI</td><td>1-65535</td><td>Set polling interval. Specifies the time between data updates. Also defines the polling interval of the analog IO. The interval is expressed in milliseconds. The default value is 97 (97 ms).</td></tr>
<tr><td>DLAY</td><td>16 bit</td><td>Sets the delay between the moment the actuator reaches its target position and the moment the ‘position reached’ flag is raised. Expressed in milliseconds. Default: 100 (ms).</td></tr>
</table>

IV. Tuning		
Command	Range	Explanation
FREQ	24 bits	Set the frequency of the excitation signals for zone 1. Unit is Hz.
FRQ2	24 bits	Set the frequency of the excitation signals for zone 2. Unit is Hz. Also used for scanning.
PROP	16 bits	Proportional control factor for zone 1.
PRO2	16 bits	Proportional control factor for zone 2.
ZON1	26 bits	Width of zone 1: Expressed in mm.
ZON2	26 bits	Width of zone 2: Expressed in mm.
DUCO	1 bit	Amplitude is used in closed loop if set to 1. If set to 0, a fixed amplitude of 50% is used. Default: 1.
ELIM	20 bits	ELIM (error limit) sets the maximum following error. When the following error exceeds the value set bij ELIM, then the controller goes in safe mode and the motor signals are switched off. Recovery: RSET or ENBL=1. This error may be triggered when trying to move beyond the physical limits of the actuators, or by setting too high a speed. Do not forget to first find the index position (INDX command) to avoid that the actuators travels beyond the end stops and triggers this error. Default: 10000.
ILIM	26 bits	Sets the following error at which the index search algorithm reverses direction. This influences the time the actuator stalls at the end position during an index search.
SLIM	26 bits	Error saturation limit
PTOL	16 bits	Position tolerance. When the actuator is within +/- position tolerance of the desired position, the control is switched off and the 'position reached' flag is raised. Values are expressed in encoder units and should be in the range 0 – 65535. The range is applied symmetrically with respect to positive and negative position errors. e.g. PTOL=2 allows s errors between -2 and +2 encoder units. Default: 2. See also TOUT and PTO2.
PTO2	16 bits	Second position tolerance, similar to PTOL. Comes into action if first position tolerance PTOL fails within a timeout time TOUT. The default value is 10.
TOUT	16 bits	Set timeout time. To avoid that the actuator keeps vibrating indefinitely around the desired position without 'landing', a timeout time can be set. The timer starts when the actuator is near the desired position, within a distance of +/- PTO2. After passing the timeout time, PTO2 becomes the new position tolerance. The time is expressed in milliseconds. The default value is 1000 (ms).
TOU2	16 bits	This defines a safety timeout. When the motor is on for a time longer that the value set by TOU2, then the controller goes in safe mode and the motor

		signals are switched off. Recovery: RSET or ENBL=1 depending on BLCK. Status bit #18 goes up when this timeout is triggered. TOU2=0 disables this timeout. Any other value sets the timeout time in seconds. Maximum value: 65535 seconds.
TOU3	16 bits	This defines another safety timeout. When the actuator is trying to "land" to a specific position for longer than the value set by TOU3, then the controller goes in safe mode. The motor signals are switched off.
ENCR	1 bit	Reset the encoder by sending "ENCR=1".

V.	**Signal shaping (advanced use)**	
Command	Range	Explanation
ENBL	0-3	Enable amplifiers. Bit 0 is for piezo signal 1, bit 1 for piezo signal 2. ENBL=3 enables both amplifiers, ENBL=0 disables both amplifiers. ENBL=1 enables only amplifier 1, ENBL=2 enables only amplifier 2.
ENBR	0-1	Enable amplifiers automatically upon controller start-up.
ZERO	-	Force the piezo signals to zero volt.
MAMP	16 bits	Set maximum amplitude. The piezo excitation signals are limited to the corresponding voltages. MAMP=65535 sets them to the maximum voltage of 45 V. MAMP=36400 sets the maximum to 25 V. The relation is linear.
MIMP	16 bits	Set minimum amplitude for piezo excitation signals. See MAMP for values.
PHAC	16 bit	Phase correction. Corrects an imbalance in the motor. Such imbalance may cause a rattling or scratching noise when the actuator moves at low speed. Practical values are in the range of a few 1000, positive or negative. Default: 0 (no correction)
OFSA	12 bits	Offset on the piezo signals on piezo phase 1. OFSA=4095 corresponds to full scale (45 V), OFSA=0 produces no offset. The relation is linear.
OFSB	12 bits	Similar to OFSB, but for piezo phase 2.
FILP	8 bits	Filter speed for phase of piezo excitation signals. Default value: 1. Max. value: 255.
FILA	8 bits	Filter speed for amplitude of piezo excitation signals. Default value: 1. Max. value: 255.

VI.	**Directional settings (advanced use)**	
Command	Range	Explanation
ENCD	1 bit	Set the encoder direction. Set the counting direction with respect to the A/B signals or sin/cos signals of the encoder. Flip this bit to swap left and right, or clockwise and counter-clockwise. Default value is 0.

ENCO	32 bits	Sets the encoder offset: distance between the index position and the desired zero position. In encoder units. Default value is 0.
ACTD	1 bit	Set the actuation direction. If not set correctly, the actuator will move away from the desired position. Default value is 0.
PATH	1 bit	For rotation stages only. Selects whether the stage will follow the shortest path (PATH=1) to the target position or follow a linear approach, respecting high to low or low to high (PATH=0). Default: 1 for rotation stages, 0 for linear stages.

5.3. Feedback from controller

Status LEDs	
Green LED	Power on for controller board
Orange LED	Motor on
Red LED	Disabled (ENBL=1 to enable) OR error (position failure, amplifier fault, timeout)

Information is sent back from the Xeryon controller to the master in ASCII format. The format is as follows:

1. Tag: Four characters describing the type of information
2. '=' sign separating the command from the corresponding value
3. Signed value associated with that information (sign + 8 decimal places). The message is terminated with a 'new line' character (ASCII code 10).

e.g. EPOS=12345678

Different types of information:
The command INFO determines which information is sent back. See the example below (INFO=2).

Tag	Explanation
SRNO	Serial number of the driver (hardware)
SOFT	Software version installed on the driver. e.g. 20103 → 2.1.3
[TYPE]	Type of motion device (XLA, XLS, XRT) and its resolution e.g. XLS1=312
STAT	Status (see below)
FREQ	Excitation frequency currently in use
SYNC	Fixed value "12345678". Can be used for debugging communication issues.
EPOS	Encoder position
DPOS	Desired position
TIME	Time stamp: resolution 0,1 ms

Meaning of STAT(US):
The Status contains 24 bits:

Status bit	Name	Explanation
0	Amplifiers enabled	XRTA only: Amplifiers for phase 1 and 2 enabled
1	End stop	Actuator stopped by end stop
2	Thermal protection 1	Amplifier for phase 1 or 3 in thermal protection.
3	Thermal protection 2	Amplifier for phase 2 or 4 in thermal protection.
4	Force zero	Motor signals are currently forced to zero.
5	Motor on	The piezo motor is on.
6	Closed loop	The actuator is currently in closed loop control.
7	Encoder at index	Indicates whether the actuator is positioned exactly at the encoder index.
8	Encoder valid	Indicates whether the encoder index has been passed and therefore the encoder value reflects the absolute position, not the relative position with respect to the startup position.
9	Searching index	Indicates whether the actuator is currently searching the index position.
10	Position reached	Indicates whether the target position is reached (within tolerance limits).
11	Error compensation	Error compensation is on.
12	Encoder error	Indicates an error produced by the encoder.
13	Scanning	Indicates whether the actuator is in a scanning mode.
14	Left end stop	Indicates that the left end stop is passed.
15	Right end stop	Indicates that the right end stop is passed.
16	Error limit	Indicates that the position error has reached the limit set by ELIM. This can indicate a collision or mechanical limit (end of stroke).
17	Searching optimal frequency	The driver is searching for the optimal excitation frequency of the piezo motor.
18	Safety timeout triggered	If this is set to “1”, then the safety timeout was triggered. See the explanation of the command “TOU2” several pages back.
19	EtherCAT acknowledge	only used when control via EtherCAT
20	Emergency stop	not used
21	Position fail	If this is set to “1”, then the safety timeout TOU3 was triggered.

XERYON

6. Communication using GPIO

There are 4 different configurations for the use of the digital and analog IO pins.

1. Control the position of the actuator using pulses. Each pulse does a step (size) in a certain direction.
2. Control the position of the actuator using an encoder-like signal.
3. Control the speed of the actuator using a PWM input signal.
4. Control the speed of the actuator using an analog input.

All of these methods are described more in detail below. To select a method, the command "GPIO" is used. This can be found in section. This command can be sent over USB or UART. **Corresponding pins** can be found in section 6.
The GPIO settings can be stored in memory using the "SAVE" command. That way, the GPIO mode will become active at power-up without having to first send the GPIO command via USB, UART or any other method.
The GPIO input and commands can be used together: the controller will react to both GPIO inputs and text commands. You don't have to go back to GPIO=0 to send text commands. You can send text commands also in the GPIO=2, 3, 4, … modes. But when going from GPIO input to text commands, first send a STOP command. Recommended maximum pulsing frequency: 20 kHz.

5.1 Pulse and direction mode

This mode is activated by the command GPIO=2 or 3.

GPIO=2 enables the input signals: ground (pin 2), pulse (13), direction (14), enable (15), index (16).
GPIO=3 enables the input signals: ground (pin 2), pulse (13), forward (14), backward (15), index (16).

On each positive edge of the PULSE signal, the target position is incremented or decremented with a particular step size. The default step size is 1 (1 encoder unit). This step size can be changed with the command STPS, e.g. STPS=10.

The DIRECTION signal determines the direction: incrementing or decrementing the target position. The ENABLE signal enables the input, it has to be high in order to start moving.

An alternative to the DIRECTION and ENABLE signal is the FORWARD and BACKWARD signal. By setting one of them high, it's possible to select the direction of moving.

The INDEX signal is an analog input used as digital input. When going high (positive edge) the controller searches the index (3.3 V is sufficient, max. 10 V).

5.2 Encoder-like signal (A quad B input)

This mode is activated by the command GPIO=4
Input signals: ground (pin 2), A (13), B (14), enable (15), index (16).

Encoder-like A quad B signal used as input to adapt the target position.

5.3 PWM control

This mode is activated by the command GPIO=8 or 9
GPIO=8 enables the input signals: ground (pin 2), PWM (13), direction (14), enable (15).
GPIO=9 enables the input signals: ground (pin 2), PWM (13), forward (14), backward (15).
The frequency of the PWM signal can be set by the command PWMF. PWMF=1000 sets the PWM frequency to 1000 Hz.

The speed is proportional to the pulse width. When the pulse width is 50 %, speed is set to 50 % of the speed set by the SSPD command. When the signal is all the time high, then 100 % of SSPD is selected. Speed can be controlled from 0 to 100 %.

The use of the direction, enable, forward and backward signals is the same as in pulse and direction mode.

5.4 Analog controls

This mode is activated by the command GPIO=12 or 13
GPIO=12 enables the input signals: ground (pin 2), speed (1), direction (14), enable (15).
GPIO=13 enables the input signals: ground (pin 2), speed (1), forward (14), backward (15).

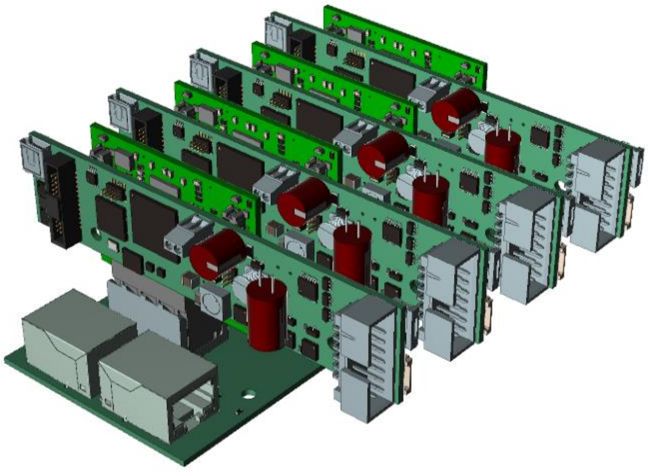
The speed input is proportional to the voltage applied to the speed input pin. 10 V corresponds to 100 % of the speed set by the SSPD command.

The use of the direction, enable, forward and backward signals is the same as in pulse and direction mode.

7. Communication using EtherCAT

EtherCAT is an Ethernet-based high-performance industrial protocol used for multi-axis, high-speed communication. Xeryon offers EtherCAT compatibility through a 4- or 16-axis motherboard, allowing multiple actuators or stages to be controlled synchronously.

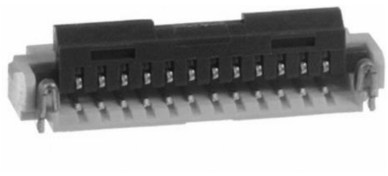
All manuals, technical drawings and 3D files and a TwinCAT 3 demo program can be found in our EtherCAT package available for download from xeryon.com. Vendor ID: 0x0000004E



4-axis EtherCAT base plate

8. Connections / pin layouts

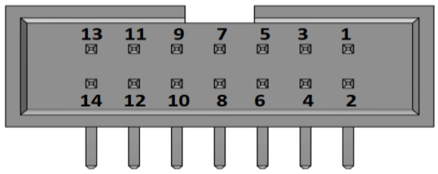
Connection to actuator



Actuator connector: ZIF 12 core
Pin 1 is defined by an arrow on the ZIF connector.
Connector: FH52E-12S-0.5SH
WARNING: Make sure to always use an opposing contact ZIF cable.

PIN #	SIGNAL	FUNCTION	IN/OUT	PIN #	SIGNAL	FUNCTION	IN/OUT
1	ENC-	Encoder Index -	IN	7	ENC+	Encoder Index +	IN
2	ENC B-	Encoder B-	IN	8	ENC GND	Encoder ground	IN
3	ENC A-	Encoder A-	IN	9	P1	Piezo phase 1	OUT
4	ENC PWR	Encoder power	OUT	10	P2	Piezo phase 2	OUT
5	ENC A+	Encoder A+	IN	11	P3	Piezo phase 3	OUT
6	ENC B+	Encoder B+	IN	12	P4	Piezo phase 4	OUT

Connection to stage

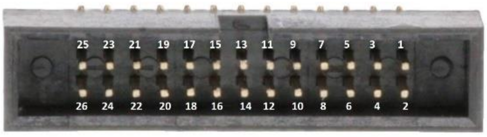


Stage connector: (14 pin, 0,1 inch pitch)
Connector: Molex 702471451
Pin 1 is defined by an arrow on the connector, see figure:

PIN #	SIGNAL	FUNCTION	IN/OUT	PIN #	SIGNAL	FUNCTION	IN/OUT
1	ENC B-	Encoder B-	IN	8	SHIELD	SHIELD	IN
2	ENC B+	Encoder B+	IN	9	ENC PWR	Encoder power	OUT
3	ENC A-	Encoder A-	IN	10	ENC GND	Encoder ground	IN
4	ENC A+	Encoder A+	IN	11	P2	Piezo phase 2	OUT
5	ENC I-	Encoder I-	IN	12	P1	Piezo phase 1	OUT
6	ENC I+	Encoder I+	IN	13	P4	Piezo phase 4	OUT
7	/	/	/	14	P3	Piezo phase 3	OUT

Connection to GPIO

IO pins (0,05 inch pitch)
connector: CNC Tech 3220-26-0300-00

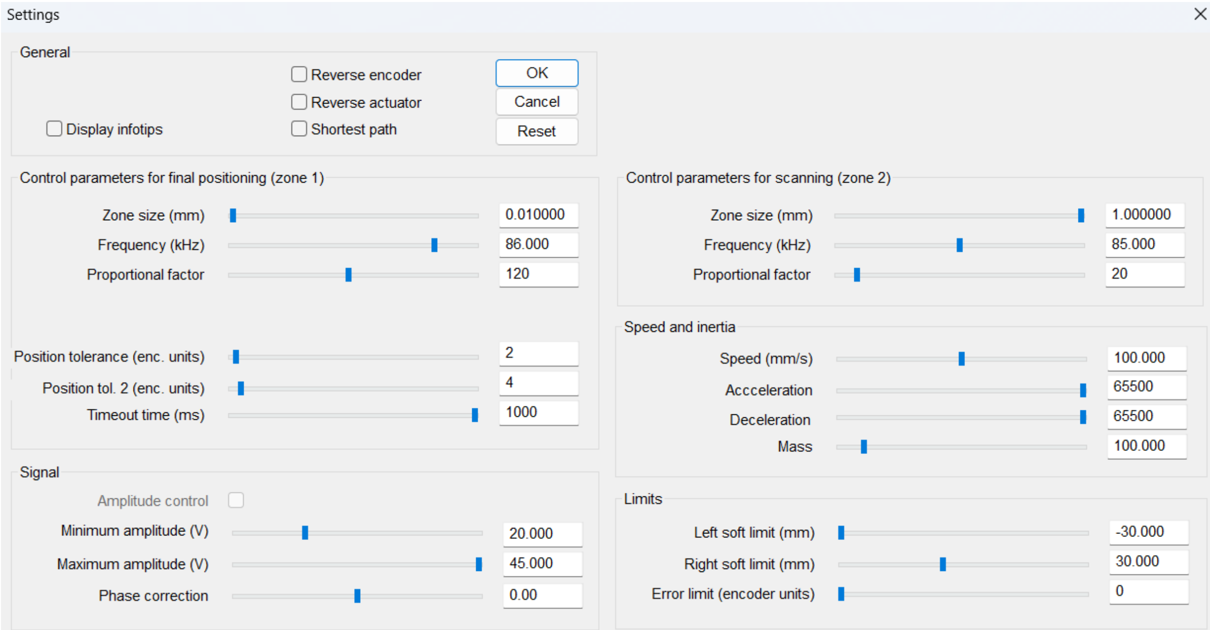


PIN #	SIGNAL	FUNCTION	IN/OUT	PIN #	SIGNAL	FUNCTION	IN/OUT
1	AI0	Analog speed input	IN	14	IN 2	Direction / forward / B*	IN
2	GND	Ground	IN	15	IN 3	Enable / backward*	IN
3	GND	Ground	IN	16	IN 4	Find index	IN
4	+5V	+5V	OUT	17	IN 5	/	
5	OUT PWR	Buffer supply voltage**	IN	18	IN 6	UART Rx	IN
6	+3,3V	+3,3V	OUT	19	/		
7	OUT 1	Trigger output	OUT	20	/		
8	OUT 2	Encoder valid flag	OUT	21	/		
9	OUT 3	Position reached flag	OUT	22	/		
10	OUT 4	Error flag	OUT	23	/		
11	OUT 5	Running flag	OUT	24	/		
12	OUT 6	UART Tx	OUT	25	GND	Ground	IN
13	IN 1	Pulse / A / PWM*	IN	26	nSRST	/	

* For more details about these pins, see section “7 Communication using digital and analog IO” about the GPIO. Digital signals are 3.3V logic and analog inputs are max 10V.
** **NOTE**: A voltage has to be supplied to this pin for UART to work. You can connect it to the available 3,3V power pin or the 5V output pin. It is also possible to supply your own voltage (max 5V). The voltage of the output pins will be determined by this pin (pin 5).

9. Tuning

Xeryon's actuators are primarily intended for closed-loop control and for this reason already have a position sensor integrated. A control algorithm is implemented on our controller and this algorithm uses multiple control parameters. These control parameters have to be set according to the type of actuator, load and customer-specific motion requirement. These parameters are pre-set by Xeryon upon delivery of the actuator and controller. In most cases, you don't need to adapt the settings right out of the box. Nevertheless, it may be required that the user modifies these parameters for optimal performance.



Below you can find the corresponding commands for the most common settings:

Setting	Command	Setting	Command
Shortest path	PATH		
Zone size, zone 1	ZON1	Zone size, zone 2	ZON2
Frequency, zone 1	FREQ	Frequency, zone 2:	FRQ2
Proportional factor, zone 1	PROP	Proportional factor, zone 2	PRO2
Position tolerance	PTOL	Speed	SSPD
Position tolerance 2	PTO2	Accel / deceleration	ACCE/DECE
Timeout time	TOUT	Mass	MASS
Amplitude control	DUCO	Left soft limit	LLIM
Min / max amplitude	MIMP/MAMP	Right soft limit	HLIM
Phase correction	PHAS	Error limit	ELIM

Frequencies/proportional values:

Typically, in zone 1 (closest to the target position), the frequency and proportional factor are both chosen higher. This gives a better 'landing' on the target position. For zone 2 (further away from the target position) the frequency is chosen lower to increase speed. At the same time the proportional factor for zone 2 typically has to be chosen lower to avoid instability. Be aware that outside a certain frequency range, the motor will have very limited force (frequency too high) or feature unstable behaviour (frequency too low). A typical frequency difference between FREQ and FRQ2 is between 1 and 3 kHz for a 1N motor and between 0.5 and 2 kHz for a 3N, 5N or 10N motor. The proportional factors in zone 1 (PROP) are typically 2-3 times the value of the proportional factors in zone 2.

When the actuator does not want to land on the target position, despite optimising frequency and proportional factor for zone 1, try to increase the positioning tolerances PTOL and PTO2. See the instruction set for more information.

To obtain the maximum power out of the actuator, the frequency in zone 2 (FRQ2) should be set as low as possible without drawing too much current, which results an amplifier fault (status bit 2 & 3)

Adding mass to the actuator

As soon as adding a mass of 100g or more, the MASS parameter can be adapted. For every 100g of mass you add, increase the MASS parameter with 100. Finding the optimal MASS parameters is usually obtained by trial-and-error. This parameter is mostly used for heavy payloads on horizontal actuators.

Changing the dynamics

Would you like your actuator to react faster or slower, the following parameters can be adapted to achieve this.

- SSPD is used to set the maximum speed the actuator. Please note, if the frequency in zone 2 is set too low, the speed can't always be reached.
- ACCE defines the acceleration of the actuator to the set speed in SSPD. The default value of 65500 means no acceleration limitation is taking into account (i.e. full acceleration).
- DECE defines the deceleration of the actuator upon reaching its target. The default value of 65500 means no deceleration limitation is taken into account (i.e. full deceleration).
- PROP & PRO2 are proportional factors used for the closed feedback loop. Both parameters are used for the two different zones (ZON1 and ZON2). Typically PROP is higher than PRO2. Increase them to get faster positioning times.

When the actuator needs to move with less overshoot, decrease the above mentioned parameters. Higher proportional factors let the controller react stronger and reduce positioning errors, but can also lead to instability or noisy operation when chosen too high.

10. Windows GUI

To provide the user with a quick way to interact with the controlled and the connected actuator, a Windows GUI is supplied with every controller. The use is simple and self-explanatory. It can be used for manual input and to run simple scripts.

10.1. Required files

The Windows GUI makes use of the following files:

- The executable of the Windows GUI: Xeryon_Dialog.exe
- A configuration file named "config.txt". This file should not be edited by the user.

- A default settings file named “settings_default.txt”. The GUI reads this file for initial settings at start-up. Replace or modify this file to alter the default settings. After saving, these values are stored on the driver.
- A settings file named “settings_user.txt”. This file will be created by the program after pressing the “save to file” button.
- A demo program. The file dialog window presents “demo.txt” as default filename.

Remark: config.txt has to be in the same folder as Xeryon_Dialog.exe.

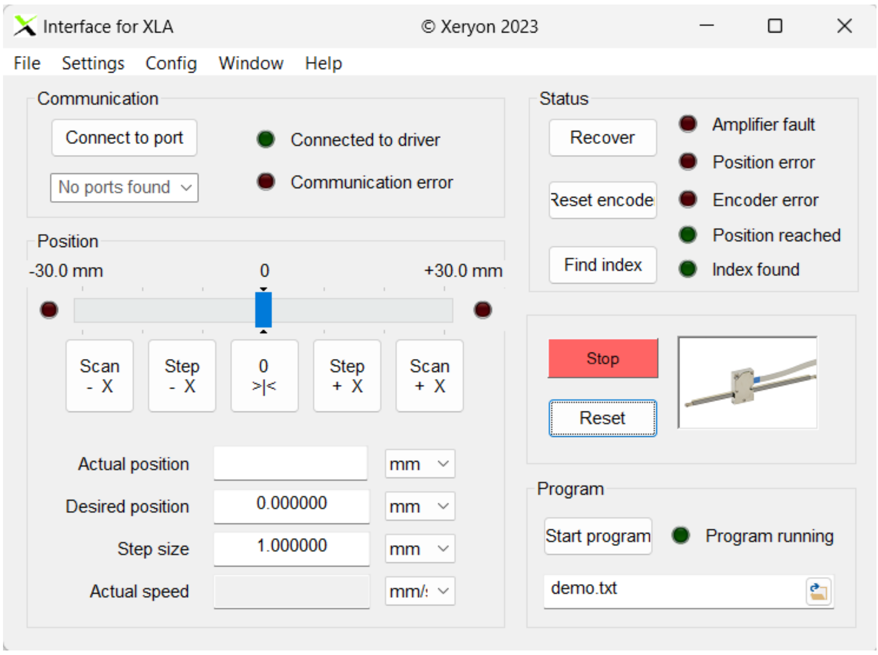
10.2. Commands for the Windows GUI demo program

Command	Explanation
BAUD	Set the baud rate for communication.
DPOL	Delay used when polling for a ‘position reached’ signal after a new target position is set. When DPOL is too small, the Windows GUI may trigger on the ‘position reached’ status flag of the previous target position due to communication delay. In that case, a succeeding WAIT command will start the timer at the start of the movement instead of after the target has been reached.
HELP	Switch help on or off. HELP=1 switches the info tips on. HELP=0 switches the info tips off.
HALT	Stop the program. (Not to be confused by the STOP command for the driver.)
LABL	Label in the program used by REPT.
LOG	Start or stop logging of data. LOG=1 switches logging on. LOG=0 switched logging off. Data is stored in datalog.csv. When datalog.csv already exists, new data is appended.
MASS	Specifies the mass/inertia of the load on the actuator. The Windows GUI calculates the optimal control parameters to obtain stable operation.
MMAS	Maximum mass that can be selected in the Windows GUI.
MPRO	Maximum proportional factor that can be selected in the Windows GUI.
MSPD	Maximum speed that can be selected in the Windows GUI.
PORT	Default port number to appear in the Windows GUI.
REPT	Repeat the above program a specified number of times. The first argument specifies the number of loops. The second argument specifies the label to jump to (label range 0 – 99). If the label does not exist, then the program jumps back to the first line. The REPT command should be placed at the end of the block to be repeated. Nesting of REPT blocks is allowed. Example: REPT=10 2 does 10 loops starting from label 2.
WAIT	Wait a specified time before proceeding to the next command. Time expressed in milliseconds. When WAIT follows a STEP or DPOS command, the timer is started when reaching the target position.

Example program file (demo.txt)

```
SSPD=100      % Set speed to 100 mm/s or 100 degrees/s
DPOS=0        % Go to position 0
WAIT=100      % Wait 100 ms after arrival at position
LABL=2        % Set label 2
```

```
DPOS=60      % Go to position 60 mm or 60 degrees
WAIT=100     % Wait 100 ms after arrival at position
SSPD=10      % Set speed to 10 mm/s or 10 degrees/s
SCAN=1       % Move with constant speed in positive direction
WAIT=2000    % Wait for 2 s (while scan goes on)
REPT=3 2     % Repeat 3 times the code above, starting from label 2
STOP         % Stop actuator
DPOS=0       % Finish in the centre
```



Function of buttons and status displays:

1 Settings button:
- Load from file: select a file to load your own user settings.
- Edit: open the settings window (see below).
- Save to file: save the current settings to a new file.
- Save on driver: save the current to the controller.

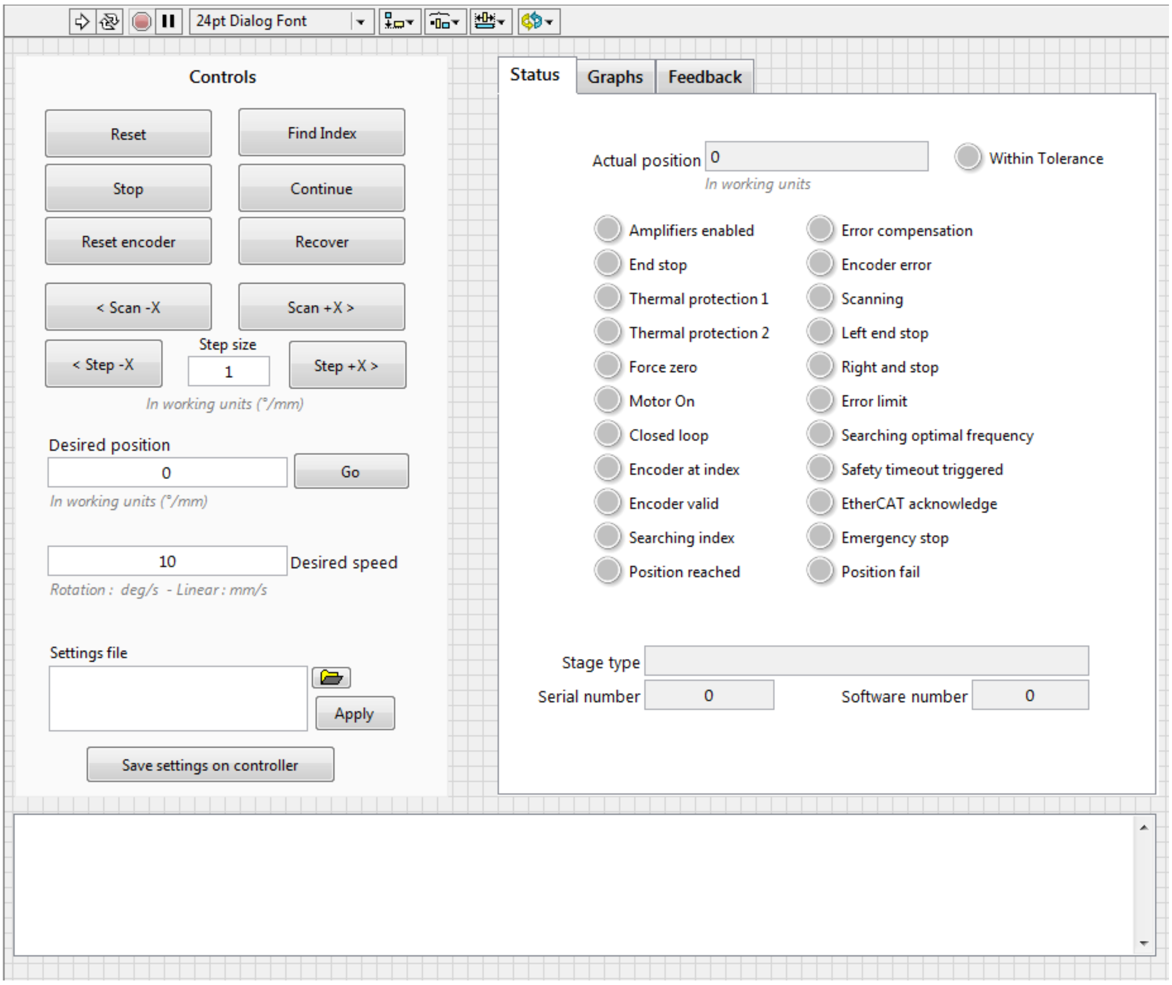
1 Save to file: save the settings to a file you select
2 Connect to port: when the port is selected, connect to the port and select settings file
3 Port: Choose COM port of XD-OEM controller.
4 Connected to driver: successfully connected to the controller.
5 Communication error: communication error between the computer and controller.
6 Recover: recover the controller after a fault/error.
7 Reset encoder: reset encoder count. Index will need to be found again.
8 Find index: start indexing procedure.
9 Amplifier fault: overcurrent, undervoltage or short-circuit error.
10 Position error: position couldn’t be reached within set timeout.
11 Encoder error: encoder not working or damaged. (commonly caused by short-circuit)
12 Position reached: desired position reached.
13 Index found: index or home position found.
14 Stop: stop the motor.
15 Reset: reset the settings back to settings_default.txt and stop the motor
16 Start program: start the program selected below
17 Demo.txt: select the demo program

18 Blue slider: use the slider to change the desired position.
19 Scan + X: the actuator will continuously move in the + X direction.
20 Step + X: the actuator will take a step in the + X direction.
21 0 button: move to the middle/index position.
22 Actual position: current position of the actuator. (can't be changed here)
23 Desired position: change the position of the actuator here.
24 Step size: choose the size of the steps.
25 Actual speed: calculated speed of the actuator. (will always be lower than actual speed due to polling delay)

11. Python, MATLAB,C++ and LabVIEW

Interaction with the XD-OEM controller and connected actuator is also possible through Python, MATLAB, LabVIEW, C++ or any other programming language that supports sending ASCII commands over a (virtual) COM port. Demo programs available for the programming languages mentioned above. The demo's include functions for advanced motion commands, error statuses, settings control and more.

All libraries are available from our website under Downloads.



LabVIEW example

12. Support

- Contact: support@xeryon.com
- Address: Interleuvenlaan 21, 3001 Leuven, Belgium
- Phone: +32 (0)16 903 904
- Website: xeryon.com