Controller & Connection Manual

Xeryon drivers for ultrasonic stages

Revision 3.26 dd. 27/12/2023

1. Introduction

Xeryon provides 5 types of controllers, all discussed in this manual. Where needed specifications are given per controller.

- The XD-A is a very small controller with housing (XD-A) or without (XD-A PCB), that can drive a single XLA.
- The XD-C is our most compact controller. It fits in the palm of your hand and can drive a single axis stage. This one comes in two versions, with housing (XD-C), or without (XD-C PCB) for a more compact OEM solution.
- The XD-M is the more advanced controller. It is available in a single-axis version (XD-M-1), but has the possibility to be daisy chained to create multi-axis systems with up to 6 stages (XD-M-2 → XD-M-6).
- The XD-19 is the rack mounted version of our most versatile controller specifically developed to fit in 19-inch server racks. Up to 12 stages can be connected and steered in this 2U high controller.
- The OEM controller is a controller that's specifically designed for OEM clients. It has a very compact format and can be easily integrated into a system.
- The integrated controller is a controller that's specifically made for driving our XLA stages in open-loop mode. The controller is integrated into the XLA itself and therefore it makes the XLA open loop version very compact.

All of our controllers come with a user-friendly Windows interface which allows for a plug-and-play experience.

Disclaimer
The product images shown may be for illustration purposes only and may not be an exact representation of the product.

2. Controller overview

	Front	Back
XD-C		
XD-C PCB		
XD-A		
XD-A PCB		
XD-M		
XD-19		
OEM controller		
Integrated controller		

2.1 Specification overview

	Compatible stages	Axis	Power supply	Control	Size	Motor/stage connector
XD-C	XLS XRTU XRTA	1 stage	48 VDC	Open and closed loop	80 x 54 x 23 mm	D-Sub 15 HD (female)
XD-C PCB					50 x 50 x 30 mm	
XD-A	XLA				80 x 54 x 23 mm	ZIF connector (12 core)
XD-A PCB					50 x 50 x 30 mm	
XD-M	XLS XRTU XRTA	Up to 6 stages			160 x 165 x 53 mm	D-Sub 15 HD (female)
XD-19		Up to 12 stages			Height: 2U, depth: 221 mm	
OEM controller	XLS XLA XRTU XRTA	Up to 16 stages	12-48 VDC		110 x 25 x 15 mm	ZIF connector (12 core)
Integrated controller	XLA open loop only	1 stage	12 VDC	Open loop	37 x 29 x 7 mm	ZIF connector (14 core)

2.2 Control overview

	USB	Digital IO	Analog IO	I2C	UART	EtherCAT	RS232/ RS422
XD-C	X						
XD-A	X						
XD-M	X	X	X				X
XD-19	X	X	X				X
XD-C PCB	X	X*	X*	X*	X*		
ssXD-A PCB	X	X*	X*	X*	X*		
OEM controller	X	X	X		X	X	
Integrated controller	X	X	X		X		

*No real connector, requires soldering to the correct pins.

Controller communication overview

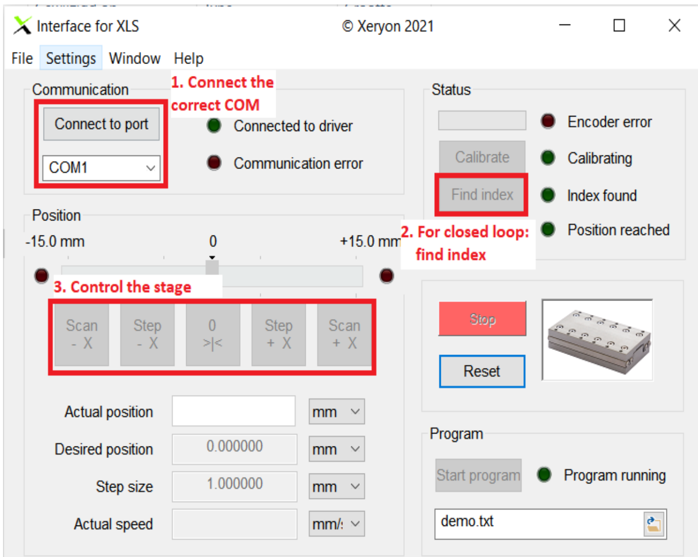
Controller type ----------------- Protocol	XUMU and XLA		Stages				Control interface	Input / Output
	Integrated controller (open-loop)	XD-OEM PCB controller (closed-loop)	XD-C PCB controller (closed-loop)	XD-C controller (closed-loop)	XD-M controller (closed-loop)	XD-19 controller (closed-loop)		
Serial communication (ASCII over virtual COM)	USB-C port on breakout board	USB mini B	USB C	USB C	USB C	USB B	Xeryon Windows Interface Python C++ Labview Matlab ASCII code (terminal) (The complete list of commands is available in the controller manual)	Input / Output
UART (TTL/CMOS)	6 pin header on breakout board	IO header	IO header	N/A	Optional	Optional		Input / Output
UART (RS232)	N/A	RS232 on piggyback board	N/A	N/A	Dsub 9 pin	Dsub 9 pin		Input / Output
Ethernet	N/A	Ethernet on piggyback board	N/A	N/A	Optional	Optional		Input / Output
RS422	N/A	RS422 on piggyback board	N/A	N/A	N/A	Optional	Contact Xeryon support team	Input / Output
I2C	N/A	Optional	IO header	N/A	Optional	Optional		Input / Output
EtherCat	N/A	EtherCat on piggyback board	N/A	N/A	N/A	N/A	EtherCat control interface (TwinCat, …)	Input / Output
Step on pulse	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25	Pulse 3,3V or 5V	Input
A quad B input (step on pulse)	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25		Input
PWM	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25		Input
Analog control (speed & direction)	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25	Voltage 0 - 10V	Input
PWM	ZIF connector / 8 pin screw terminal on breakout board	N/A	N/A	N/A	N/A	N/A	Pulse 3,3V	Input
Analog control (speed & direction)	ZIF connector / 8 pin screw terminal on breakout board	N/A	N/A	N/A	N/A	N/A	Voltage 0 - 3,3V	Input
A quad B encoder output	Optional (requires encoder)	Optional (requires encoder)	IO header	N/A	Optional	Optional	Signal: Encoder output	Output
Digital pulse 3,3V	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25	Signal: Trigger pulse on specific positions	Output
Digital signal 3,3V	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25	Signal: Index found	Output
Digital signal 3,3V	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25	Signal: Position reached	Output
Digital signal 3,3V	N/A	Optional	IO header	N/A	Optional	Optional	Signal: Limit passed	Output
Digital signal 3,3V	N/A	IO header	IO header	N/A	Dsub 15 / 25	Dsub 15 / 25	Signal: Error	Output

3. Getting started

All controllers are delivered with a power adapter, USB cable and USB-stick with software to control the piezo stage(s). This allows for plug-and-play capabilities of the product. To get started, plug all stage connectors into the controller. Connect the controller with a computer with the provided USB cable and open up the Windows interface, press SCAN or MOVE and the stages will start moving.

The USB-stick contains three files:

- a Windows interface
- a *settings_default.txt* file, this contains settings that are specific to a stage.
- a *config.txt* file, this configures the Windows interface to your setup.



A quick overview of the Windows interface:

To control a stage:

1. Connect the correct COM-port
2. For closed loop stages only: press "Find Index".
   The stage starts moving and searches for the index of the encoder.
3. You can start controlling the stage:
   a. SCAN: move with a continuous speed
   b. STEP: move with a specified step size
   c. 0: move to the zero position.

More information about the Windows interface can be found in section " 15 Windows user interface".

4. Control of a stage

The controllers can be controlled using various methods:

- Using a serial connection (over USB) — see section 5
- Using RS232 or RS422 — see section 6
- Using UART — see section 6
- Using I2C — see section 6
- Using a combination of digital and analog IO — see section 7
- Controlling the integrated controller — see section 8

5. Communication over a virtual COM port

A host computer or controller can communicate with the Xeryon controller via the USB configured as a virtual COM port. The controller firmware makes use of the LUFA library[1] for communication.

The baud rate is automatically detected by the driver and can be up to 115 200 baud. The protocol uses 8 data bits, 1 stop bit, no parity bit, no handshaking.

This section is divided into three parts:

- 5.1 Format
- 5.2 Instruction set
- 5.3 Feedback from controller

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.

[1] LUFA Library: Copyright © Dean Camera, 2015.

dean@fourwalledcubicle.com
www.lufa-lib.org

Permission to use, copy, modify, distribute, and sell LUFA software library and its documentation for any purpose is granted without fee, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of the author not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The author disclaims all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Addressing axes on the XD-M or XD-19

To address an axis, put the axis name (1 letter followed by a colon) before the command. When no axis is specified, the command goes to the first axis (stage 1). In case of a single-axis system, no axis designation is required.

Standard axis names:

- XD-M: X Y Z A B C
- XD-19: A B C D E F G H I J K L

Multiple-axes system:

- Y:DPOS=-1000 → positioning of the Y-axis.
- X:RSET → resetting the X-axis.

Single-axis system:

- DPOS=-1000
- RSET

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. In case of a multi-axes system put an axis tag in front of the command. e.g. X:STAT asks for the status word of axis X. This works best with INFO=0, otherwise the reply disappears in the constant flood of feedback data.

Units are as follows:

Type	Rotation stage	Translation stage	Resolution
Time, delays	ms		1 ms
Target position, step size	encoder units		1 encoder increment
Speed	deg/s (*)	µm/s	0.01 deg/s or 1 µm/s
Frequency	Hz		1 Hz

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

5.2. Instruction set

This is a list of all possible instructions that can be send. In this table, a difference is made between "controller commands" and "stage commands".

"Controller commands" are commands that don't need to have a stage specified in front of it. "Stage commands" on the other hand are stage specific and the stage has to be specified. eg: *INFO=0* is a "controller command" and *X:SSPD=1000* is a "stage command"

This part is divided into:

I. Driver configuration: stage selection			**stage command**
Command	**Range**	**Mode**	**Explanation**
XRT1			Configure the driver for a XRTU-30 or XRTU-40 rotation stage.
	2,		XRTU-40 with 3 µrad resolution: XRT1=2
	3,		XRTU-30 with 3 µrad resolution: XRT1=3
	18,		XRTU-40 with 19 µrad resolution: XRT1=18
	19,		XRTU-30 with 19 µrad resolution: XRT1=19
	47,		XRTU-40 with 49 µrad resolution: XRT1=47
	49,		XRTU-30 with 49 µrad resolution: XRT1=49
	73,		XRTU-40 with 73 µrad resolution: XRT1=73
	109		XRTU-30 with 109 µrad resolution: XRT1=109
XRT3			Configure the driver for a XRTU-60 rotation stage.
	3,		XRTU-60 with 3 µrad resolution: XRT3=3
	19,		XRTU-60 with 19 µrad resolution: XRT3=19
	49,		XRTU-60 with 49 µrad resolution: XRT3=49
	109		XRTU-60 with 109 µrad resolution: XRT3=109
XRTA	109		Configure the driver for a XRTA rotation stage
			XRTA=109
XLS1			Configure the driver for a XLS-1 linear stage.
	1,		XLS-1 with 1 nm resolution: XLS1=1

	5, 78, 313, 1251		XLS-1 with 5 nm resolution: XLS1=5 XLS-1 with 78 nm resolution: XLS1=78 XLS-1 with 312 nm resolution: XLS1=313 XLS-1 with 1250 nm resolution: XLS1=1251
XLS3	1, 5, 78, 313, 1251		Configure the driver for a XLS-3 linear stage. XLS-3 with 1 nm resolution: XLS3=1 XLS-3 with 5 nm resolution: XLS3=5 XLS-3 with 78 nm resolution: XLS3=78 XLS-3 with 312 nm resolution: XLS3=313 XLS-3 with 1250 nm resolution: XLS3=1251
XLA1	78 312 1250		Configure the driver for a XLA1 linear stage. XLA1 with 78 nm resolution: XLA1 = 78 XLA1 with 312 nm resolution: XLA1 = 312 XLA1 with 1250 nm resolution: XLA1 = 1250
XLA3	78 312 1250		Configure the driver for a XLA3 linear stage. XLA3 with 78 nm resolution: XLA3 = 78 XLA3 with 312 nm resolution: XLA3 = 312 XLA3 with 1250 nm resolution: XLA3 = 1250

II. Closed-loop motion			**stage command**
Command	**Range**	**Mode**	**Explanation**
INDX	0, 1	Closed loop	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 the stage reaches a mechanical limit (detected by position error > ILIM) it reverses the search direction. After finding the index, the stage is positioned at the index position. Some stages (e.g. XLS with 312 or 1250 nm resolution) have their physical index close to the end limits. For those stages a large encoder offset is used (ENCO), thus after finding the index location near the end limits, the stage is sent to the centre corresponding to the encoder offset.
HOME	-	Closed loop	Go to the home position. This equals DPOS=0
DPOS	26 bits	Closed loop	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 stage.
STEP	26 bits	Closed loop	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 stage towards higher encoder values,

			negative values send the stage towards lower encoder values. Closed-loop control is used to reach and maintain the new position.
SCAN	-1,0,1	Closed loop	Continuously move with fixed speed. The speed is maintained by closed-loop control. A positive number sends the stage towards increasing encoder values, a negative number sends the stage towards decreasing encoder values. A zero value stops the stage.
SSPD	24 bits	Closed loop	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	Closed loop	Set the speed which is used while searching the index. Unit is 1 µm/s or 0.01 deg/s.
ACCE	16 bits	Closed loop	Set acceleration for speed profile. Expressed in m/s2. Default value: 65500
DECE	0-255	Closed loop	Set deceleration for speed profile, when approaching target position. Default value: 255. Maximum value: 255 for XD-C controller / 65500 for XD-OEM controller.
LLIM	26 bits	Open and closed loop	Set low-side soft end stop. Expressed in encoder units.
HLIM	26 bits	Open and closed loop	Set high-side soft end stop. Expressed in encoder units.
ENON	1 bit	Closed loop	Enable or disable the encoder. ENON=1 enables the encoder. It's the default value.

III. Open-loop motion			**stage command**
Command	**Range**	**Mode**	**Explanation**
MOVE	-1,0,1	Open loop	Continuously move in open loop. Phase and amplitude influence the speed, but speed is not controlled. A positive number sends the stage towards increasing encoder values, a negative number sends the stage towards decreasing encoder values. A zero value stops the stage.
PHAS	16 bits	Open loop	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	Open loop	Set amplitude for open-loop piezo excitation signals.

IV. Open- and closed-loop motion			**controller command**
Command	**Range**	**Mode**	**Explanation**
STOP	-	Open and closed loop	Stop the stage.
CONT	-	Open and closed loop	Continue movement after a stop command.
ENBL	1 bit	Open & closed loop	If a stage is in error mode, you can recover from that by sending "ENBL=1". This enables the stage again.
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
PLIM			Set wether or not to react to physical limits (eg. magnetical limits on XLA). 1: Physical limits on 0: Physical limits off Default value: 0
DTIM			Determines how long (in ms) the motor must be turned off before it can be turned on again. Ensures that the stage is not over-excited when landing by switching it on and off too quickly. Default value: 0

V. Control and tuning parameters			**stage command**
Command	**Range**	**Mode**	**Explanation**
FREQ	24 bits	Closed loop	Set the frequency of the excitation signals for zone 1. Unit is Hz. Default: 170000 (Hz).
FRQ2	24 bits	Open and closed loop	Set the frequency of the excitation signals for zone 2. Unit is Hz. Also used for scanning. Default: 170000 (Hz).
LFRQ	24 bits	Open and closed loop	Set the minimum piezo excitation frequency. Unit is Hz. Used as the lower boundary for finding the optimal frequency. Default: 165 kHz.
HFRQ	24 bits	Open and closed loop	Set the maximum piezo excitation frequency. Unit is Hz. Used as the upper boundary for finding the optimal frequency. Default: 175 kHz.

PROP	16 bits	Closed loop	Proportional control factor for zone 1. Default: 100.
PRO2	16 bits	Closed loop	Proportional control factor for zone 2. Default: 100.
ZON1	26 bits	Closed loop	Width of zone 1: +/- value around target position. Expressed in encoder units. Default: 100.
ZON2	26 bits	Closed loop	Width of zone 2: +/- value around target position. Expressed in encoder units. Default: 1000.
CFRQ	16 bits	Closed loop	Control frequency. Adapt this value to obtain stable closed-loop control. The optimal control frequency depends on the mass or inertia of the load. Default: 30000 (30 000 Hz) for zero load.
DUCO	1 bit	Closed loop	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	Closed loop	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 stage, or by setting a too high speed. Do not forget to first find the index position (INDX command) to avoid that the stage travels beyond the end stops and triggers this error. Default: 10000.
ILIM	26 bits	Closed loop	Sets the following-error at which the index search algorithm reverses direction. This influences the time the stage stalls at the end position during an index search.
PTOL	16 bits	Closed loop	Position tolerance. When the stage 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	Closed loop	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	Closed loop	Set timeout time. To avoid that the stage keeps vibrating indefinitely around the desired position without 'landing', a

			timeout time can be set. The timer starts when the stage 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 50 (50 ms). Also check PTOL and PTO2.
TOU2	16 bits	Closed loop	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	Closed loop	This defines another safety timeout. When the stage 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. Status bit #
ENCR	1 bit	Closed loop	Reset the encoder by sending “ENCR=1”.

VI. Signal shape and conditioning			**stage command**
Command	**Range**	**Mode**	**Explanation**
ENBL	0-3	-	For XRTA only: 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.
COMP	12 bits	-	For XRTA only. Percentage of active error compensation. COMP=0 disables active error compensation. COMP=100 applies 100 % compensation, as set by the compensation values stored in memory. For other percentage values, the compensation is scaled accordingly. Maximum value: 4095.
ZERO	-	Open & closed loop	Force the piezo signals to zero volt.
MAMP	16 bits	Open and closed loop	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	Closed loop	Set minimum amplitude for the piezo excitation signals. See MAMP for values.
PHAC	16 bit	Open & closed loop	Phase correction. Corrects an imbalance in the motor. Such imbalance may cause a rattling or scratching noise when the stage 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	Open and closed loop	Filter speed for phase of piezo excitation signals. Default value: 1. Max. value: 255.
FILA	8 bits	Open and closed loop	Filter speed for amplitude of piezo excitation signals. Default value: 1. Max. value: 255.

VII. Directional settings			**stage command**
Command	**Range**	**Mode**	**Explanation**
ENCD	1 bit	Open and closed loop	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	Open and closed loop	Sets the encoder offset: distance between the index position and the desired zero position. In encoder units. Default value is 0.
ACTD	1 bit	Open and closed loop	Set the actuation direction. If not set correctly, the stage will move away from the desired position. Default value is 0.
PATH	1 bit	Closed loop	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

VIII. Trigger outputs			**stage command**
Command	**Range**	**Mode**	**Explanation**
TRGS	26 bits	Closed loop	Start of the trigger pulses, expressed in encoder units.
TRGW	26 bits	Closed loop	Width of the trigger pulses, expressed in encoder units. This (positive) value should be lower than the pitch (TRGP).
TRGP	26 bits	Closed loop	Pitch of the trigger pulses, expressed in encoder units. The pitch can’t be negative. To trigger pulses in the negative direction: set the TRGS at the most negative point and use a positive pitch.

TRGN	26 bits	Closed loop	Number of trigger pulses

IX. Communication			**controller command**
Command	**Range**	**Mode**	**Explanation**
INFO	4 bits	-	Select type of info to be transmitted from the driver to the master (PC). 0: Stop broadcasting info automatically 1: SRNO, SOFT, STAGE, STAT, SYNC 2: SRNO, SOFT, STAGE, 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.
UART	0, 2400, 4800, 9600, 14400, 19200, 28800 38400, 57600, 76800	-	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.
POLI	1-65535	-	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).
DLAY	16 bit	Closed loop	Sets the delay between the moment the stage reaches its target position and the moment the ‘position reached’ flag is

			raised. Expressed in milliseconds. Default: 100 (100 ms).

X. Manage settings			controller command
Command	**Range**	**Mode**	**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
FACT	-	-	Reset to factory settings
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.
GPIO	0-13	-	Select the preferred input mode using the GPIO command. For more details see section "7 Communication using digital and analog IO" GPIO=0: Control via IO pins switched off. The controller will only react to text commands sent through USB or UART. GPIO=2: Pulse and direction mode, with direction & enable pins. GPIO=3: Pulse and direction mode, with forward & backward pins. GPIO=4: A quad B input mode GPIO=8: PWM control, with direction & enable pins. GPIO=9: PWM control, with forward & backward pins. GPIO=12: Analog control, with direction & enable pins. GPIO=13: Analog control, with forward & backward pins.
SPTS	16 bits	Closed loop	Select the step size for each STEP pulse. The step size is expressed in encoder units. Default: 1 encoder unit

XI. Test			controller command
Command	**Range**	**Mode**	**Explanation**
TEST	0-1	-	Test LED indicators (XD-M and XD-19). TEST=1 switches all indicators on. TEST=0 brings them back to their function.

XII. Integrated controller (for XLA open loop) specific commands			
Command	**Range**	**Mode**	**Explanation**
VOLT	16 bits	Open loop	Set the desired voltage level expressed in mV. Default value: 48000
DICF	1 bit	Open loop	Select a way to change direction. A value of "0" means that the direction can be controlled using the "MOVE" command (default). A value of "1" means that the direction can be controlled using the "DIR" pin.
SPCF	2 bits	Open loop	Select a way to control the speed. A value of "0" means that the speed can be controlled using the "SSPD" command (default). A value of "1" means that the speed can be controlled using a PWM signal on the "SPD" pin. A value of "2" means that the speed can be controlled using an analog signal on the "SPD" pin.
PLIM	1 bit		Set wether or not to react to physical limits (eg. magnetical limits on XLA). 1: Physical limits on 0: Physical limits off Default value: 0

5.3. Feedback from controller

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

1. One character identifying the axis, followed by a colon. This only applies to multiple-axes systems. For a single-axis system the axis identification is omitted.
2. Tag: Four characters describing the type of information
3. '=' sign separating the command from the corresponding value
4. Signed value associated with that information (sign + 8 decimal places). The message is terminated with a 'new line' character (ASCII code 10).

e.g. X:EPOS=+12345678

Different types of information:

For multiple-axis systems this information is being sent for every axis. First all data for axis 1, then all data for axis 2, … . The command INFO determines which information is sent back.

Tag	Explanation

SRNO	Serial number of the driver (hardware)
SOFT	Software version installed on the driver. e.g. 20103 → 2.1.3
[STAGE]	Type of stage (XLS1, XLS3, XRT1, XRT3, XLA1, XLA3, XRTA 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 the STAT(US) Word:
The Status Word contains 24 bits:

Status bit	Name	Explanation
0	Amplifiers enabled	XRTA only: Amplifiers for phase 1 and 2 enabled
1	End stop	Stage 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 stage is currently in closed loop control.
7	Encoder at index	Indicates whether the stage 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 stage 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 stage 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.

6. Communication using UART, I2C, RS232/422

Some controllers are equipped with connections for UART, I2C and/or RS232 / RS422. The instruction set for these protocols is the same as communication over USB. This can be found in section “0

Communication over a virtual COM port”.

The connections for the different controllers can be found here:

- XD-M see section “9 Connections on the housing”
- XD-19 see section “9 Connections on the housing”
- XD-C PCB see section “10 Connections on the PCB”
- XD-A PCB see section “10 Connections on the PCB”
- Integrated controller see section “9 Connections on the housing”
- OEM controller see section “10 Connections on the PCB”

7. Communication using digital and analog IO

The connection overview for digital and analog communication can be found here:

- XD-M	see section	“9 Connections on the housing”
- XD-19	see section	“9 Connections on the housing”
- XD-C PCB	see section	“10 Connections on the PCB”
- XD-A PCB	see section	“10 Connections on the PCB”
- Integrated controller	see section	“9 Connections on the housing”
- OEM controller	see section	“10 Connections on the PCB”

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

1. Control the position of the stage using pulses. Each pulse does a step (size) in a certain direction.
2. Control the position of the stage using an encoder-like signal.
3. Control the speed of the stage using an PWM input signal.
4. Control the speed of the stage 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 “0

Communication over a virtual COM port" -> "5.2 Instruction set" -> "   ". This command can be send over USB, UART, RS232, RS422 and I2C.

The GPIO settings can be stored in memory using the "SAVE" command. That way, the GPIO mode will become active at powerup 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.

Pulse and direction mode

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

GPIO=2 enables the input signals: pulse, direction, enable, index.
GPIO=3 enables the input signals: pulse, forward, backward, index

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).

A quad B input

This mode is activated by the command GPIO=4
Input signals: A, B, enable, index

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

PWM control

This mode is activated by the command GPIO=8 or 9
GPIO=8 enables the input signals: PWM, direction, enable, index
GPIO=9 enables the input signals: PWM, forward, backward, index
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.

Analog controls

This mode is activated by the command GPIO=12 or 13
GPIO=12 enables the input signals: speed, direction, enable, index

GPIO=13 enables the input signals: speed, forward, backward, index

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.

8. Controlling the integrated controller

The integrated controller has very simple control. Therefore this has a designated section in this manual.

The integrated controller is used for the XLA open loop. At both ends of the rod of the XLA, a permanent magnet is attached. These magnets are detected with 2 HAL-effect sensors integrated in the controller. By adjusting the placement of the magnets, you can adjust the traveling range.

Note: If used without payload attached to the rod ends, an end screw with o-ring needs to be attached in order to avoid standing waves in the rod.

Control through software

One method to control our stages is with software commands. This can be done over a USB connection or by using UART. Any command line window can be used to send the desired commands. These commands are explained below. To use UART you first need to enable it. This can be done using the "UART" command.

It's also possible to request the value of a command. This can be done by sending the command with "=?" after it. For example: "FREQ=?" returns "FREQ=84000".

If the device is not found or you can not create a serial connection with the device, you should install the "STM32 Virtual COM Port Driver". You can find the download link here: https://www.st.com/en/development-tools/stsw-stm32102.html

Command	Explanation
FREQ	Set the frequency of the excitation signals. The unit is Hz. When you set a frequency with "FREQ", the parameters "FRQ1" and "FRQ2" will both be set equal to the value of "FREQ". This means that for all speeds, this one single frequency will be used.
FRQ1	Set the frequency to be use in the speed control. This is the frequency that will be used at the lowest speed. The unit is Hz. Default: 87000 Hz.
FRQ2	Set the highest frequency to be used in the speed control. This is the frequency that will be used at the highest speed. The unit is Hz. Default: 84500 Hz.
PHAS	Set the phase offset between the excitation signals. The unit is degree. A value ranging from 0 to 90° can be set. Default: 0
PHS1	Set the lowest phase offset to be used in the speed control. This corresponds to the phase that will be used at the lowest speed. The unit is degree. Default: 90.
PHS2	Set the highest phase offset to be used in the speed control. This corresponds to the phase that will be used at the highest speed. The unit is degree. Default: 90.
VOLT	Set the voltage that is used to drive the motor. The unit is mV. A value ranging from 12 000 mV to 72 000 mV can be used. Default: 48 000 mV.
DICF	Select a way to change direction. A value of "0" means that the direction can be controlled using the "MOVE" command (default). A value of "1" means that the direction can be controlled using the "DIR" pin.
SPCF	Select a way to control the speed. A value of "0" means that the speed can be controlled using the "SSPD" command (default). A value of "1" means that the speed can be controlled using a PWM signal on the "SPD" pin. A value of "2" means that the speed can be controlled using an analog signal on the "SPD" pin.

OSPD	Set the speed. This can be a value ranging from 0 (standstill) to 100 (full power).
MOVE	This command can be used to move the stage. By sending “MOVE=1” or “MOVE=-1” you move the stage. The “-1” and “1” determine the direction. By sending “MOVE=0” you stop the movements of the stage.
LDCF	Enable of disable the limit switches. LDCF=0: disable the limit switches, the stage will keep going. LDCF=1: enable the limit switches, the stage will stop at the limit.
SOFT	You can request the software number with this command by sending “SOFT=?”
SRNO	You can request the serial number with this command by sending “SRNO=?”.
RSET	You can reset the device by sending “RSET”.
UART	You can enable UART communication by sending the baudrate with this command. For example, UART=9600 enables UART communication and the baudrate is 9600.

Control through hardware

Another method to control our stages is using it’s hardware IO.

By putting either 3.3V or 0V onto the “DIR” pin, it’s possible to select the travel direction. This pin is also available from the breakouts board. For example: low is left, high is right.
To control the direction with these pins, you should first enable this feature by sending “DICF=1” to the stage. This is explained in the “Software Control” section. This only has to be done once.

The “LIM+” and “LIM-“ output pins are 0V unless the stage is at one of the limits, then the output corresponding to that end has 3.3V.

<u>The speed can be controlled in two ways:</u>

- You can control the speed with an **analog signal** on the “SPD” pin. The analog input voltage can range from 0V to 3.3V which corresponds to lowest speed (standstill) to full speed. Make sure you don’t exceed 3.3V.
  To enable this way of controlling the speed, you should first send “SPCF=2” to the controller. This is explained in the “Software Control” section. This only has to be done once.

- You can control the speed with a **PWM signal** on the “SPD” pin. This signal should not exceed 3.3V and the frequency has to stay between 200 Hz and 200 kHz. By varying the duty cycle of the PWM signal you can control the speed from the lowest speed (standstill) to full speed.
  To enable this way of controlling the speed, you first send “SPCF=1” to the controller. This is explained in the “Software Control” section. This only has to be done once.

Speed control

It’s possible to go a little bit more into depth with controlling the speed. But this is not required to start using the stage.

The speed can be controlled in three different ways: by changing the phase offset of the excitation signals, by changing the frequency of the excitation signals or using a combination of both frequency and phase.

The frequency is set to a value between FRQ1 and FRQ2 depending on the speed set. The phase is set to a value between PHS1 and PHS2 depending on the speed set. By changing the speed, the value’s of the frequency and phase are interpolated depending on the speed that’s set.

To only control the speed using the phase, set the FRQ1 and FRQ2 to the same value. This can be done using the “FREQ=” command.

To only control the speed using the frequency, set the PHS1 and PHS2 to the same value. This can be done using the “PHAS=” command.

The phase more or less controls the driving force of the stage and the frequency mostly controls the speed.

9. Connections on the housing

The piezo drive signals have an amplitude of 45 Vpp and a frequency range between 0 and 300 kHz, sufficient to drive Xeryon's ultrasonic linear and rotary stages.

In section "2 Controller overview" the connections of each controller are listed. Make sure to check which connector is available on your controller

applicable for

XD-C
XD-C PCB
XD-A
XD-A PCB
XD-M
XD-19
OEM controller
Integrated controller

- Stage connector: D-Sub HD 15 (female)

PIN #	SIGNAL	PIN #	SIGNAL
1	NC	9	CLK+/E+
2	Encoder power	10	GND P1 / GND P2 / P3
3	Encoder GND	11	Encoder Index -
4	Piezo phase 2	12	Encoder A-
5	Piezo phase 1	13	Encoder B-
6	Encoder Index +	14	NC
7	Encoder A+	15	P4
8	Encoder B+		

applicable for

XD-C
XD-C PCB
XD-A
XD-A PCB
XD-M
XD-19
OEM controller
Integrated controller

- Stage connector: ZIF (12 core)
  WARNING: Make sure to always use an **opposing contact** ZIF cable.
  Pin number 1 is the pin which is furthest from the USB-C connector.

PIN #	SIGNAL	PIN #	SIGNAL
1	Encoder Index -	7	Encoder Index +
2	Encoder B-	8	Encoder GND
3	Encoder A-	9	Piezo phase 1
4	Encoder power	10	Piezo phase 2
5	Encoder A+	11	Piezo phase 3
6	Encoder B+	12	Piezo phase 4

applicable for

XD-C
XD-C PCB
XD-A
XD-A PCB
XD-M
XD-19
OEM controller
Integrated controller

- RS232 connector D-Sub 9 (female)

PIN #	SIGNAL	PIN #	SIGNAL
1	NC	6	NC
2	RXD	7	RTS
3	TXD	8	CTS

4	NC	9	NC
5	GND		

applicable for

XD-C
XD-C PCB
XD-A
XD-A PCB
XD-M
XD-19
OEM controller
Integrated controller

- Analog IO connector D-Sub 15 (female)
  For more details about these pins, see section "7 Communication using digital and analog IO" about the GIPO. Digital signals are 3.3V logic and analog inputs are max 10V.

PIN #	SIGNAL	PIN #	SIGNAL
1	Analog speed input stage 1	9	Find index stage 1
2	Analog speed input stage 2	10	Find index stage 2
3	Analog speed input stage 3	11	Find index stage 3
4	Analog speed input stage 4	12	Find index stage 4
5	Analog speed input stage 5	13	Find index stage 5
6	Analog speed input stage 6	14	Find index stage 6
7	GND	15	5 V out
8	3,3 V out		

applicable for

XD-C
XD-C PCB
XD-A
XD-A PCB
XD-M
XD-19
OEM controller
Integrated controller

- Digital IO connector D-Sub HD 25 (female)
  For more details about these pins, see section "7 Communication using digital and analog IO" about the GIPO. Digital signals are 3.3V logic and analog inputs are max 10V.

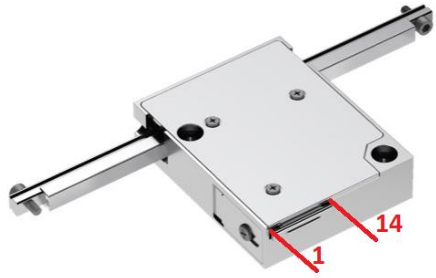
PIN #	SIGNAL	PIN #	SIGNAL
1	Pulse / A / PWM stage 1	14	Trigger pulse* stage 1
2	Direction / forward / B stage 1	15	Encoder valid signal stage 1
3	Enable / backward stage 1	16	Position reached signal stage 1
4	GND	17	GND
5	Pulse / A / PWM stage 2	18	Trigger pulse* stage 2
6	Direction / forward / B stage 2	19	Encoder valid signal stage 2
7	Enable / backward stage 2	20	Position reached sisgnal stage 2
8	GND	21	GND
9	Pulse / A / PWM stage 3	22	Trigger pulse* stage 3
10	Direction / forward / B stage 3	23	Encoder valid signal stage 3
11	Enable / backward stage 3	24	Position reached signal stage 3
12	GND	25	5 V out
13	3,3 V out		

* For the trigger pulses, see the command "TRGS".

- ZIF-connector (14 core) for the integrated controller
! This pinout is only valid for the V2 PCB. You can find the version of your PCB written on the PCB. If you need the pinout of the V1 PCB, feel free to contact us.
The pinout for the ZIF-connector is shown below. We also have a breakout board available to easily connect to this ZIF-cable. More information about this breakout board can be found in “11 Pin layout of stage connectors”.

applicable for

XD-C
XD-C PCB
XD-A
XD-A PCB
XD-M
XD-19
OEM controller
Integrated controller



PIN #	SIGNAL	FUNCTION	IN/OUT	PIN #	SIGNAL	FUNCTION	IN/OUT
1	GND		In	2	D+	USB positive data signal	In/Out
3	D-	USB negative data signal	In/Out	4	GND		In
5	ENC-I	Encoder Index signal	Out	6	ENBL	Enable signal	In
7	Rx	UART Rx*	In	8	Tx	UART Tx*	Out
9	LIM- / ENC-A	Digital IO: Left limit / Encoder A signal	Out	10	LIM+/ ENC-B	Digital IO: Right limit / Encoder B signal	Out
11	SPD	Analog IO/PWM: Speed	In	12	DIR	Digital IO: Direction	In
13	+12V	+12V	In	14	+12V	+12V	In

All digital IO’s are 3.3V logic and all analog IO’s are max 3.3V. Do not supply more than 3.6V to the stage, this will damage the actuator.

*Rx and Tx work on 3.3V. If you want to connect with RS232, higher voltages are needed. In this case a voltage convertor (buffer) should be added.

Warning: Do not open the driver. In case of a damaged connector or cable, please contact Xeryon for repair or replacement.

10. Connections on the PCB

10.1. XD-A and XD-C controller PCB

Important: The XD-C and XD-A controller look identical, but they have different electronic components. The XD-A connects with the ZIF connector to the XLA micro-actuator. The XD-C connects with the D-Sub 15 connector to the Xeryon stages. The XD-C controller should never be connected to an XLA (micro-actuator) and the XD-A controller should never be connected to the XLS, XRT (stage). This can damage the controllers and the stages.

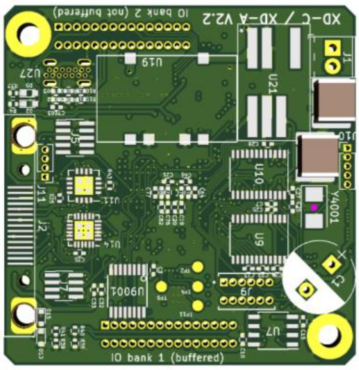
Summarized: The ZIF connector on the XD-C should not be used. The D-Sub connector on the XD-A should not be used.

All connections on the PCB can be connected to the controller housing with a dedicated connector on request by the costumer.

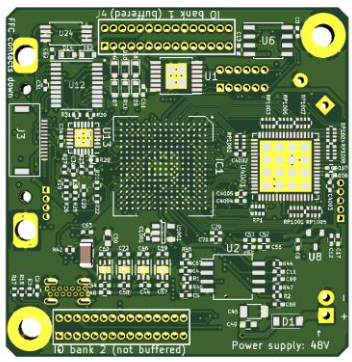
Stage connectors

The stage connector for the XD-C PCB is a D-SUB. This is the same D-SUB as the XD-C with housing and is defined in section 9.
The stage connector for the XD-A PCB is a ZIF connector. This is the same ZIF connector as the XD-A with housing and is defined in section 9.

Top and bottom view (XD-C and XD-A)

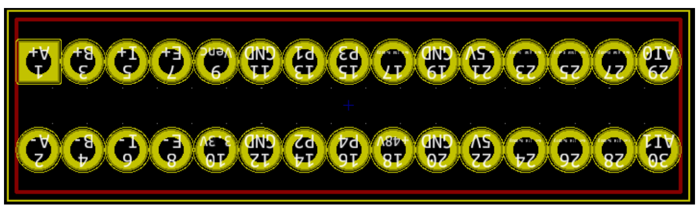


Top-view



Bottom-view

IO bank 1, buffered (XD-C and XD-A)



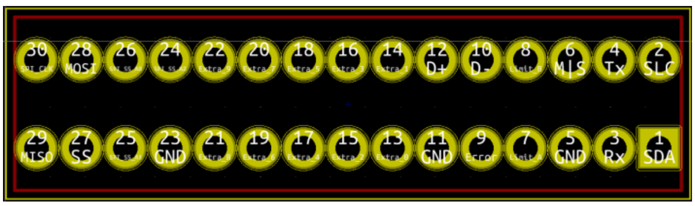
Pin #	Symbol	Function	In/Out	V	Pin #	Symbol	Function	In/Out	V
1	A+	Positive encoder signal A	In/Out	5V	2	A-	Negative encoder signal A	In/Out	5V
3	B+	Positive encoder signal B	In/Out	5V	4	B-	Negative encoder signal B	In/Out	5V

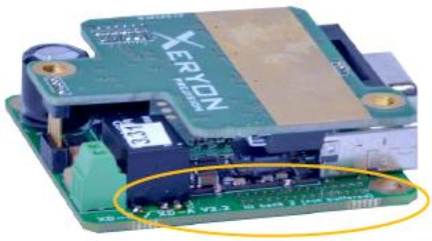
5	I+	Positive encoder index	In/Out	5V	6	I-	Negative encoder index	In/Out	5V
7	E+	Positive encoder error	In/Out	5V	8	E-	Negative encoder error	In/Out	5V
9	Venc	Encoder supply out	Out	5V	10	3.3 V	3.3 V out	Out	3.3V
11	GND	Ground	In		12	GND	Ground	In	
13	P1	Piezo phase 1	Out	48V	14	P2	Piezo phase 2	Out	48V
15	P3	Piezo phase 3	Out	48V	16	P4	Piezo phase 4	Out	48V
17	-	Not connected	-		18	48 V	Power supply	In	48V
19	GND	Ground	In		20	GND	Ground	In	
21	-5 V	-5 V out	Out	-5V	22	5 V	5 V out	Out	5V
23	DI0	Pulse / A / PWM*	In	3.3V	24	DO0	Trigger pulse (see TRGS command)	Out	3.3V
25	DI1	Direction / forward / B*	In	3.3V	26	DO1	Encoder valid signal	Out	3.3V
27	DI2	Enable / backward*	In	3.3V	28	DO2	Position reached signal	Out	3.3V
29	AI0	Analog speed input*	In	3.3V	30	AI1	Find index*	In	3.3V

* For more details about these pins, see section "7 Communication using digital and analog IO" about the GIPO. Digital signals are 3.3V logic and analog inputs are max 10V.

Connector: 1.27 mm pitch, 2 rows, 30 contact

IO bank 2, not buffered (XD-C and XD-A)





in #	Symbol	Function	In/Out	Pin #	Symbol	Function	In/Out
1	SDA	I2C data	In/Out	2	SLC	I2C clock	In/Out
3	Rx	UART Rx*	In	4	Tx	UART Tx*	Out
5	GND	Ground	In	6	M | S	Set as master or slave	In
7	Limit A	Passed position limit A	Out	8	Limit B	Passed position limit B	Out
9	Error	Error, fault	Out	10	D-	USB negative data signal	In/Out
11	GND		In	12	D+	USB positive data signal	In/Out
13	Extra 0		In	14	Extra 1		In
15	Extra 2		In	16	Extra 3		In
17	Extra 4	A+ encoder (3.3V-	Out	18	Extra 5	B+ encoder (3.3V)	Out
19	Extra 6	Index flag	Out	20	Extra 7	Motor on flag	Out
21	Extra 8	Position reached flag	Out	22	Extra 9	Index known flag	Out
23	GND	Ground	In	24	SPI A2	SPI slave address bit 2	Out
25	SPI A1	SPI slave address bit 1	Out	26	SPI A0	SPI slave address bit 0	Out
27	SPI SS	SPI slave select	In/Out	28	MOSI	SPI master out, slave in	In/Out
29	MISO	SPI master in, slave out	In/Out	30	SPI CLK	SPI clock	In/Out

*Rx and Tx work on 3.3V. If you want to connect with RS232, higher voltages are needed. In this case a voltage convertor (buffer) should be added.

Connector: 1.27 mm pitch, 2 rows, 30 contacts

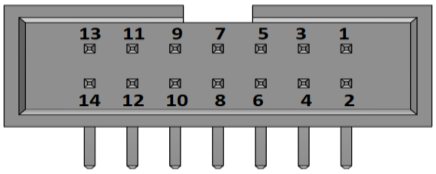
10.2. OEM Controller

- Stage 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

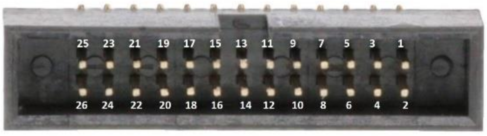
3	ENC A-	Encoder A-	IN	9	P1	Piezo phase 1	OUT
4	ENC PWR	Encoder power	OUT	10	P2	Piezo phase 2	OUT
5	ENC A+	Encoder A+	IN	11	P3	Piezo phase 3	OUT
6	ENC B+	Encoder B+	IN	12	P4	Piezo phase 4	OUT

- Stage connector: pin 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

- 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 GIPO. Digital signals are 3.3V logic and analog inputs are max 10V.
** NOTE: A voltage has to be supplied to this pin. 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).

11. Pin layout of stage connectors

! WARNING !

For the XLS stages with resolution 1 nm, 5 nm and 312 nm and the XRTU stages with a resolution finer than 109 µrad the encoder signals should be within a separate shielded wire. The encoder signals should be shielded from the motor wires and other noise sources. This is because the encoder signals are an analog signal.

11.1. Pin layout of stage connectors XLS-1

- Pin layout of standard stages (D-sub 15 HD male, 3 rows of 5 pins)

PIN #	SIGNAL	PIN #	SIGNAL
1	NC	9	Encoder Error
2	Encoder Power (5V)	10	Piezo Ground
3	Encoder Ground	11	Encoder index -
4	Piezo phase 2	12	Encoder A -
5	Piezo phase 1	13	Encoder B -
6	Encoder Index +	14	NC
7	Encoder A +	15	NC
8	Encoder B +	Shell	Shield

- Pin layout of stages with vacuum compatibility (D-sub 15 female, 2 rows of contacts)

PIN #	SIGNAL	PIN #	SIGNAL
1	Encoder Index -	9	Encoder Error
2	Encoder A -	10	Encoder B -
3	Shield	11	NC
4	Piezo phase 2	12	NC
5	Piezo phase 1	13	Piezo Ground
6	Encoder B +	14	Encoder Ground
7	Encoder A +	15	Encoder Power (5V)
8	Encoder Index +	Shell	Shield

When preparing air-side cables from a vacuum flange to the controller, take care to follow this pin layout correctly. The above numbering is for the stage connector on the vacuum side. Vacuum feedthroughs may have male pins on both sides and may therefore mirror the pin numbering for the corresponding air side connector!

To assure proper shielding of the signals inside the vacuum chamber, the shield has to be connected through the vacuum feedthrough. Plugging the connectors into the feedthrough is not always sufficient: in some feedthroughs the electrical path of the shield goes only via the locking screws. Also with PEEK connectors no electrical path for the shield exists. Therefore, pin 3 in used to connect the shield through the vacuum feedthrough. The connection of pin 3 to the shield is realised inside the connector, thus no wire in the cable is assigned for this purpose.

11.2. Pin layout of stage connectors XLS-3

- Pin layout of standard stages (D-sub 15 HD male, 3 rows of 5 pins)

PIN #	SIGNAL	PIN #	SIGNAL
1	NC	9	Encoder Error
2	Encoder Power (5V)	10	Piezo phase 3
3	Encoder Ground	11	Encoder index -
4	Piezo phase 2	12	Encoder A -
5	Piezo phase 1	13	Encoder B -
6	Encoder Index +	14	NC
7	Encoder A +	15	Piezo phase 4
8	Encoder B +	Shell	Shield

- Pin layout of stages with vacuum compatibility (D-sub 15 female, 2 rows of contacts)

PIN #	SIGNAL	PIN #	SIGNAL
1	Encoder Index -	9	Encoder Error
2	Encoder A -	10	Encoder B -
3	Shield	11	NC
4	Piezo phase 2	12	Piezo phase 4
5	Piezo phase 1	13	Piezo phase 3
6	Encoder B +	14	Encoder Ground
7	Encoder A +	15	Encoder Power (5V)
8	Encoder Index +	Shell	Shield

When preparing air-side cables from a vacuum flange to the controller, take care to follow this pin layout correctly. The above numbering is for the stage connector on the vacuum side. Vacuum feedthroughs may have male pins on both sides and may therefore mirror the pin numbering for the corresponding air side connector!

To assure proper shielding of the signals inside the vacuum chamber, the shield has to be connected through the vacuum feedthrough. Plugging the connectors into the feedthrough is not always sufficient: in some feedthroughs the electrical path of the shield goes only via the locking screws. Also with PEEK connectors no electrical path for the shield exists. Therefore, pin 3 in used to connect the shield through the vacuum feedthrough. The connection of pin 3 to the shield is realised inside the connector, thus no wire in the cable is assigned for this purpose.

11.3. Pin layout of stage connectors XRT-A / XRT-U

- Pin layout of standard stages (D-sub 15 HD male, 3 rows of 5 pins)

PIN #	SIGNAL	PIN #	SIGNAL
1	NC	9	NC
2	Encoder Power (5V)	10	Piezo Ground
3	Encoder Ground	11	NC
4	Piezo phase 2	12	NC
5	Piezo phase 1	13	NC
6	Encoder Index +	14	NC
7	Encoder A +	15	NC
8	Encoder B +	Shell	Shield

- Pin layout of stages with vacuum compatibility (D-sub 15 female, 2 rows of contacts)

PIN #	SIGNAL	PIN #	SIGNAL
1	NC	9	NC
2	NC	10	NC
3	Shield	11	NC
4	Piezo phase 2	12	NC
5	Piezo phase 1	13	Piezo Ground
6	Encoder B +	14	Encoder Ground
7	Encoder A +	15	Encoder Power (5V)
8	Encoder Index +	Shell	Shield

When preparing air-side cables from a vacuum flange to the controller, take care to follow this pin layout correctly. The above numbering is for the stage connector on the vacuum side. Vacuum feedthroughs may have male pins on both sides and may therefore mirror the pin numbering for the corresponding air side connector!

To assure proper shielding of the signals inside the vacuum chamber, the shield has to be connected through the vacuum feedthrough. Plugging the connectors into the feedthrough is not always sufficient: in some feedthroughs the electrical path of the shield goes only via the locking screws. Also with PEEK connectors no electrical path for the shield exists. Therefore, pin 3 in used to connect the shield through the vacuum feedthrough. The connection of pin 3 to the shield is realised inside the connector, thus no wire in the cable is assigned for this purpose.

11.4. Colour codes

XLS-1, XRT-U-30 and XRT-U-40 have 1 cable: a 12-core cable (AWG 36) containing encoder and piezo signals.

XLS-3 and XRT-U-60 stages have 2 cables: a 12-core cable (AWG 36) for the encoder signals, and a 4-core cable (AWG 32) for the piezo signals.

	Colour code	XRT-A-25 XRT-U-30 XRT-U-40	XRT-U-60	XLS-1	XLS-3
12-core cable AWG 36	Orange	Piezo phase 1	NC	Piezo phase 1	NC
	Yellow	Piezo phase 2	NC	Piezo phase 2	NC
	Pink	Piezo ground	NC	Piezo ground	NC
	Grey	Encoder power	Encoder power	Encoder power	Encoder power
	White	Encoder ground	Encoder ground	Encoder ground	Encoder ground
	Brown	A+	A+	A+	A+
	Red	B+	B+	B+	B+
	Black	I+	I+	I+	I+
	Metallic blue	A- (*)	A- (*)	A-	A-
	Silver	B- (*)	B- (*)	B-	B-
	Metallic purple	I- (*)	I- (*)	I-	I-
	Green	NC	NC	Encoder error (#)	Encoder error (#)
	Shield	Shield	Shield	Shield	Shield
4-core cable AWG 32	Red	NA	Piezo phase 1	NA	Piezo phase 1
	White	NA	Piezo phase 2	NA	Piezo phase 2
	Black	NA	Piezo phase 3	NA	Piezo phase 3
	Green	NA	Piezo phase 4	NA	Piezo phase 4
	Shield	NA	Shield	NA	Shield

(*) Not used for XRT-U-30-109, XRT-U-40-73 and XRT-A-25-109

(#) Not used for XRT-U-30-109, XRT-U-40-73, XRT-A-25-109, XLS-xxx-312, XLS-xxx-1250

NA: not applicable

NC: not connected

12. Connections to the breakout PCB

A breakout PCB is available for the integrated controller. It's not necessary but it allows to easily get started with this integrated controller. The connections of the ZIF cable from the controller are connected to easily accessible connectors, a USB-C connection and UART pins.

To connect the actuator with the breakout PCB:

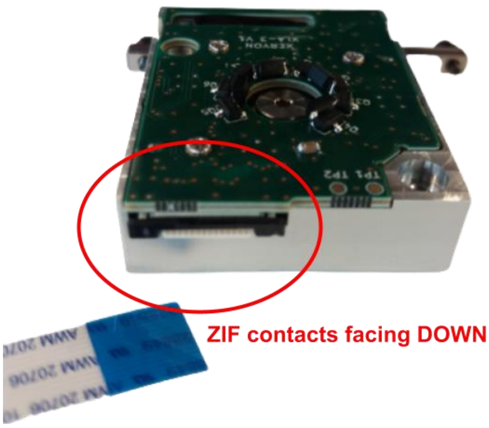
Make sure that the ZIF contacts are pointing down in the actuator and the contacts are facing up at the breakout PCB.

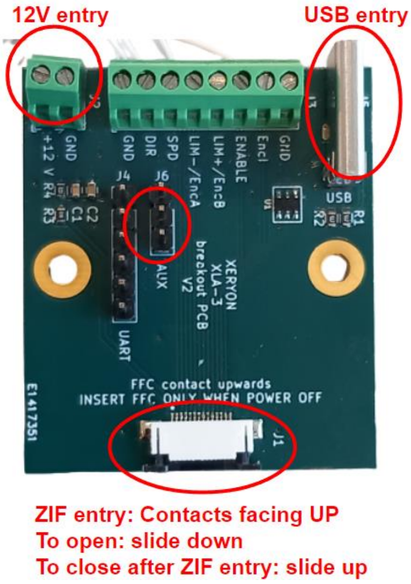
On the actuator side:

- ZIF contacts face down
- Do not connect FFC with power on, always connect with power off

On the breakout PCB side:

- ZIF contacts face up
- Do not connect FFC with power on, always connect with power off
- 12V input
- Inputs: 3.3V, **never exceed 3.6V**.





13. Explanation of the control parameters

Xeryon's stages 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 stage, load and customer-specific motion requirement. These parameters are pre-set by Xeryon upon delivery of the stage and controller. You don't need to adapt the settings right out of the box. Nevertheless, it may be required that the user modifies these parameters when conditions change. This section explains the most important parameters. To adapt these parameters, you should read "Tune your control parameters". That section explains the procedure to adapt the right parameters

The most important control parameters are: excitation frequency (FREQ), proportional factor (PROP), control frequency (CFRQ) and positioning tolerance (PTOL).

The first parameter to set is the excitation frequency. This should correspond to the resonant frequency of the piezomotor driving the stage. Resonant frequencies differ slightly among piezomotors and therefore this parameter is tuned for each individual stage. Values close to the resonant frequency generate the highest force and speed, but setting it a few kHz above the resonant frequency often leads to a more relaxed control with better 'landing' characteristics. By using different frequencies for scanning and fine-positioning, the piezomotor can be tuned to work optimally in different scanning and positioning conditions.

It is not recommended to reduce the excitation frequency far below the pre-set frequency. This may lead to unstable scanning and positioning behaviour.

The proportional control factors are a second set of parameters that are used to tune the stage for a specific situation. 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. Lower proportional factors, on the other hand, will results in a more sluggish motion response with more overshoot.

The control frequency is important when the load on the stage (mass) is significantly increased. More information for each of these parameters is given in the instruction set (See "Tune your control parameters").

To optimise closed-loop control for both speed and accuracy, two zones are defined each with a different excitation frequency and proportional factor for both scanning and fine-positioning applications. The zones are defined symmetrically around the target position, with zone 1 being the area closest to the target and zone 2 the widest. The zones are set with the commands ZON1 and ZON2. Corresponding to these zones, there are also 2 excitation frequencies and 2 proportional factors. In between the zones the values for excitation frequency and proportional factor are interpolated:

- Positioning error < ZON1: FREQ & PROP
- Positioning error > ZON2: FRQ2 & PRO2
- ZON1 < positioning error < ZON2: interpolated values

	Zone 1 (close to target)	Zone 2 (far from target)
Set zone width	ZON1	ZON2
Set excitation frequency	FREQ	FRQ2

Set proportional factor	PROP	PRO2

Typically, in zone 1 (closest to the target position), the excitation 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 excitation 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 an XSU-1 motor and between 0.5 and 2 kHz for an XSU-3 motor. The proportional factors in zone 1 (PROP) are typically 2-3 times the value of the proportional factors in zone 2.

When the stage 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.

During scanning motion, the controller only uses the parameters of zone 2 (FRQ2 and PRO2).

Homing procedures

All closed-loop stages contain a position encoder, also called a position sensor. This is a small component that is integrated in an actuator or a stage and constantly feeds back the position of the slider. Five different position sensor resolutions are offered: 1250 nm, 312 nm, 78 nm, 5 nm and 1 nm. We use optical sensors for the highest precision stages (78 nm, 5 nm and 1 nm) and inductive sensors for the medium precision stages (1250 nm and 312 nm) and for dusty environments.

The optical encoders have a physical index or reference mark, in the centre of the encoder strip. To find this index, the controller sets off in a specified direction to search. When the stage reaches a mechanical limit it reverses the search direction. When the index is found, the stage stops.

The inductive sensors contain the physical index position close to the end limit. The controller moves in a specified direction until the stage reaches a mechanical limit, where the physical index is located. For those stages a large encoder offset is used (ENCO), thus after finding the index location near the end limits, the stage is sent to the centre corresponding to the encoder offset.

The inductive sensors have no unique index or reference mark. Instead, these sensors give an index pulse each 1.28 mm. The controller slowly moves the stage in a specified direction until it reaches a mechanical limit. Then, it moves in the other direction and uses the first index pulse it passes as reference. With respect to this reference point, the zero position is defined by the parameter ENCO (in encoder units).

As a result it is important that nothing blocks the physical index of the stages.

14. Tuning the control parameters

As previously mentioned in ”Control”, the control parameters have to be set according to the type of stage and load. Each individual stage is tested by Xeryon and the parameters are pre-set for zero payload upon delivery of the stage and controller.

Nevertheless, it may be required that the user modifies these parameters when conditions change.

Adding mass to the stage

As soon as adding a mass of 100g or more, the MASS parameter should be adapted. For every 100g of mass you add, increase the MASS parameter with 100. The correlation is not 1-on-1, and stage dependant. Finding the optimal MASS parameters is usually obtained by trial-and-error.

MMAS is the software limit used for the Windows User Interface and the maximum MASS that can be set for the stage.

After changing the MASS parameters, the proportional parameters (PROP and PRO2) should also be adapted. A first good practice is too halve the values of PROP and PRO2.

Changing the dynamics

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

- SSPD is used to set the speed the stage will move from point to point or during a scan movement.
- ACCE defines the acceleration of the stage to the set speed in SSPD. The default value of 10000 means no acceleration limitation is taking into account (i.e. full acceleration).
- DECE defines the deceleration of the stage upon reaching its target. The default value of 255 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 better positioning accuracy. Both PROP and PRO2 should always be changed according to the same ratio.

When the stage needs to move and react faster to get better positioning accuracy and less overshoot, increase 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.

15. Windows user interface

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

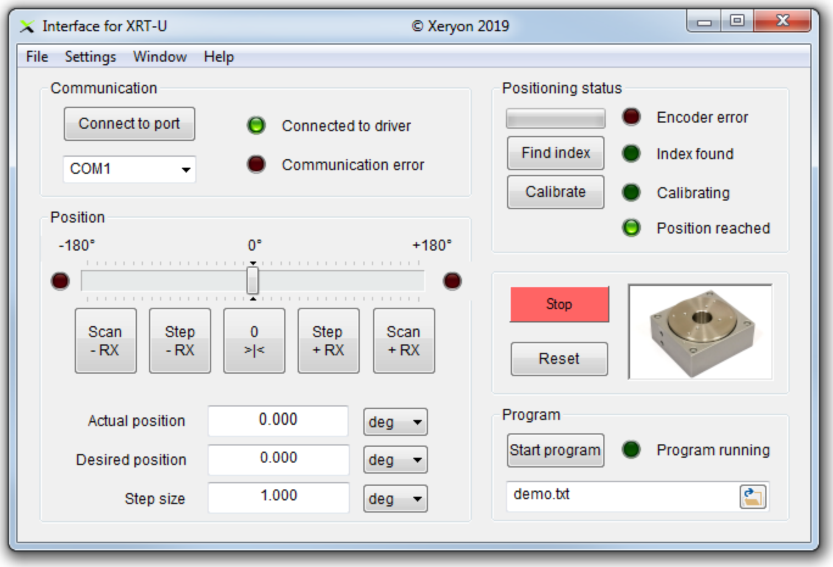
This user interface is not intended for interaction with other programs written in C, LabVIEW, MATLAB, etc. These programs should directly interact with the driver via the protocol described in previous sections. On request, code (C++, Python, LabVIEW, MATLAB) is provided to ease interfacing.

Remark: Before one can use the user interface “Xeryon_Dialog.exe”, one has to install the driver installation files. Copy the “xd-c.inf” and “xd_c_win.cat” files from the USB-stick to a folder on your hard disk, preferably in the same folder as the user interface “Xeryon_Dialog.exe” file. Don’t install the driver installation files from a USB-stick or network disk. Then, when both driver installation files are copied to your hard disk, install the files by a right mouse click on “xd-c.inf” and choose “Install”. The installation can take a while. When installation is complete, a port number will be assigned to the controller. Start the user interface, select the correct port, click ‘Connect to port’.

15.1. User interface for XD-A, XD-C PCB and XD-C

In case of a single axis system (one stage), select the correct graphical interface by setting GUI in config.txt to the following value:

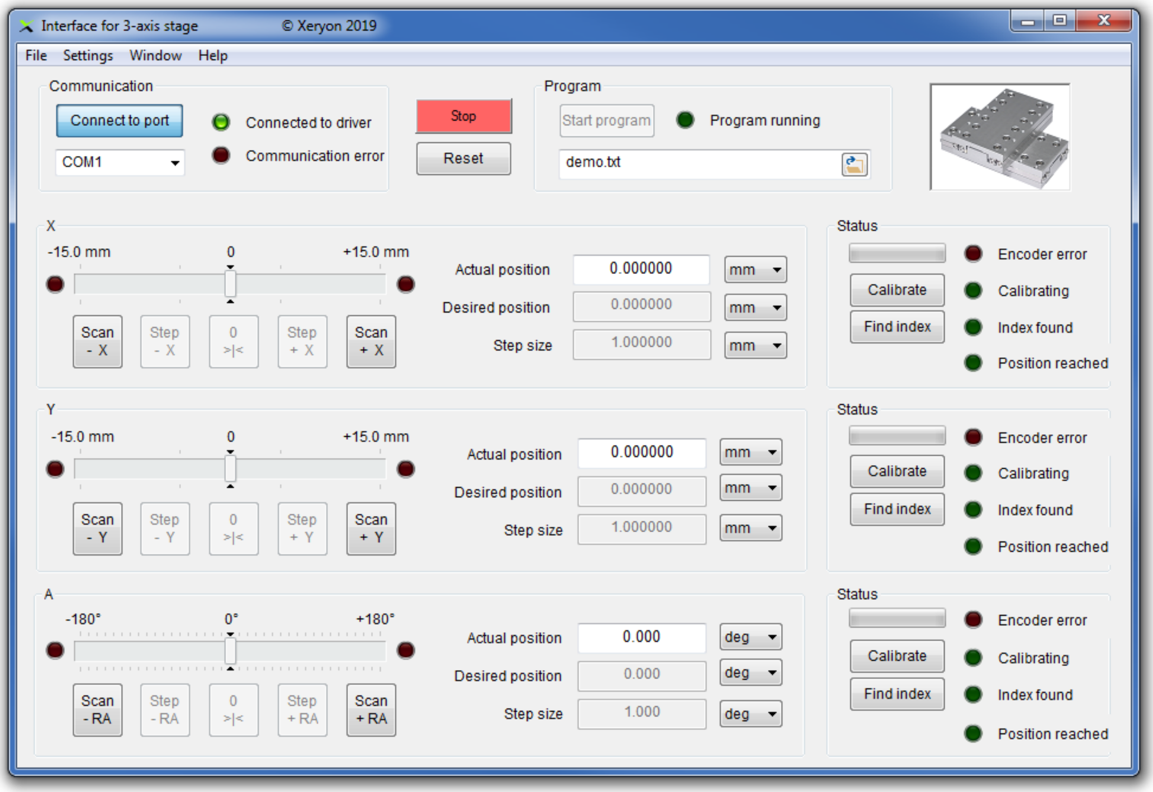
- XRTU-30: GUI=13
- XRTU-40: GUI=19
- XLS1 or XLS3: GUI=14
- XRTA: GUI=15
- XLA closed loop: GUI=21
- XLA open loop: GUI=22



15.2. User interface for XD-M

In the config.txt file, select the correct graphical interface:

- 2 axes in XY config.: GUI=17
- 2 independent axes: GUI=24
- 3 independent axes: GUI=20
- 4 independent axes: GUI=23



15.3. User interface for XD-19

The Windows Interface can handle a maximum of 4 axes. For controlling more axes, we refer to the Python interface.

15.4. Required files

The User Interface makes use of the following files:

- The executable of the User Interface: 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 User Interface 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”. The user can save and load alternative settings files via the menu. The filename is free to choose, but the file dialog window presents “settings_user.txt” as default filename.
- Several program files for which the name and content can be freely chosen. The file dialog window presents “demo.txt” as default filename.

Remark: config.txt and settings_default.txt have to be in the same folder as Xeryon_Dialog.exe.

The settings file and program files are composed of the same commands that are used to directly talk to the XD-C. An important difference is that the User Interface uses position data in degrees, millimetre etc., while encoder units have to be used when talking directly to the XD-C. The conversion is automatically made by the User Interface. Similar conversions are made for speed (deg/s or mm/s).

A few additional commands exist that affect the program flow and connection. These are not axis specific.

15.5. Commands for the Windows Interface

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 Interface 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 stage. The User Interface calculates the optimal control parameters to obtain stable operation.
MMAS	Maximum mass that can be selected in the User Interface.
MPRO	Maximum proportional factor that can be selected in the User Interface.
MSPD	Maximum speed that can be selected in the User Interface.
PORT	Default port number to appear in the User Interface.
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.

Remark: Comment text should be preceded by a percentage sign.

Example config file (config.txt)

This config file is written by Xeryon for your specific setup and should normally not be changed by the user.

GUI=19	% Version of graphical user interface
AXES=1 X	% Number of axes in the system and their names
XRT1=73	% Type of stage, e.g. XRT-U-40 with 73 µrad resolution
RANGE=360	% Range of the X-stage, e.g. 360 degrees
LEVEL=0	% User level (0-2)
CL=1	% Scan buttons for closed loop (1) instead of Move buttons for open loop (0)

Example settings file (settings_default.txt) – XD-C

INFO=2	% Select info
ZON1=0.1	% Set width of zone 1
ZON2=1	% Set width of zone 2
FREQ=178000	% Excitation frequency for zone 1
FRQ2=176000	% Excitation frequency for zone 2
HFRQ=185000	% Upper limit for excitation frequency
LFRQ=165000	% Lower limit for excitation frequency
PROP=500	% Proportional factor for zone 1
PRO2=100	% Proportional factor for zone 2
MPRO=120	% Maximum proportional factor
LLIM=-14	% Low-side soft end stop
HLIM=14	% High-side soft end stop
MASS=0	% Mass
MMAS=1000	% Maximum mass
SSPD=25	% Speed
MSPD=150	% Maximum speed
ELIM=30000	% Error limit
PTOL=5	% Positioning tolerance 1
PTO2=10	% Positioning tolerance 2
TOUT=300	% Timeout time 1
PHAC=-500	% Phase compensation

Example settings file (settings_default.txt) – XD-M

```
PORT=2          % Select COM-port 2 as default
INFO=2          % Select info, identical for all axes
X:ENCD=0        % Encoder direction for X-axis
Y:ENCD=1        % Encoder direction for Y-axis
X:FREQ=167000 % Piezo excitation frequency for X-axis
Y:FREQ=167000 % Piezo excitation frequency for Y-axis
X:SSPD=10       % Speed of X-axis
Y:SSPD=20       % Speed of Y-axis
X:PROP=3        % Proportional control factor for X-axis
Y:PROP=3        % Proportional control factor for Y-axis
X:LLIM=-28      % Lower position limit for X-axis
X:HLIM=28       % Upper position limit for X-axis
Y:LLIM=-28      % Lower position limit for Y-axis
Y:HLIM=28       % Upper position limit for Y-axis
X:MASS=100      % Load on X-axis
Y:MASS=50       % Load on Y-axis
X:PTOL=3        % Positioning tolerance in X-direction
Y:PTOL=5        % Positioning tolerance in Y-direction
```

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
DPOS=120
WAIT=100
DPOS=180
WAIT=500
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)

SCAN=-1 % Move with constant speed in negative direction

WAIT=2000 % Wait for 2 s (while scan goes on)

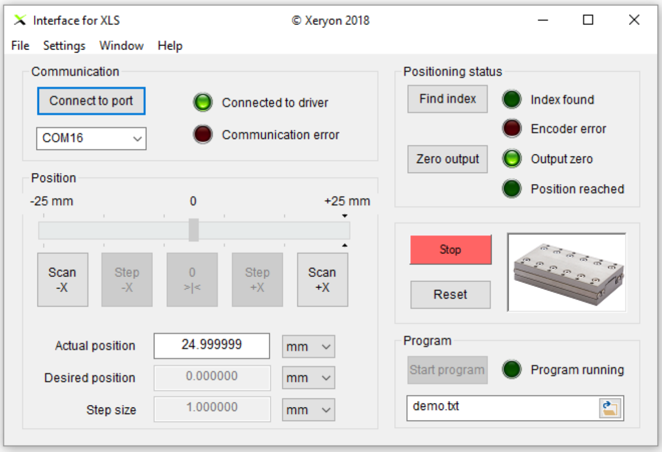
REPT=3 2 % Repeat 3 times the code above, starting from label 2

STOP % Stop stage

DPOS=0 % Finish in the centre

15.6. Using the Windows Interface

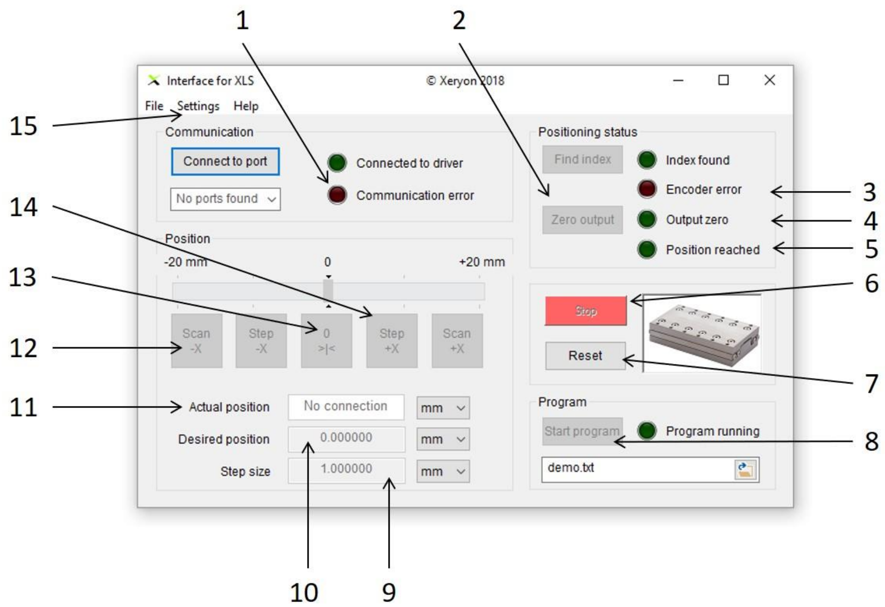
1. Switch on the XD-U driver using the on/off button (green button at the front panel). The green LED will blink when the driver is powered. The XD-C has no on/off button. The XD-C is automatically powered when the power plug is connected to the driver.
2. Install the driver installation files and start the windows interface "Xeryon_Dialog.exe". The windows interface will look like the figure below[2].
3. Select the port number in the drop down box and click on the "Connect to port" button.
4. The green LED "Connected to driver" should light up now. When the LED does not light up it means that the computer is not connected with the driver. In this case, one cannot go to the next step. Please reconnect the computer with the driver and repeat the steps described above until the green LED "Connected to driver" lights up.



5. Click "Find index". The stage will automatically search for the index and stop on the index position. The green LED "Index found" will light up when the index is found.
6. The driver and stage are now configured and ready to use.

Remark: the translation speed of the stage can be changed by altering the speed in the Settings menu.

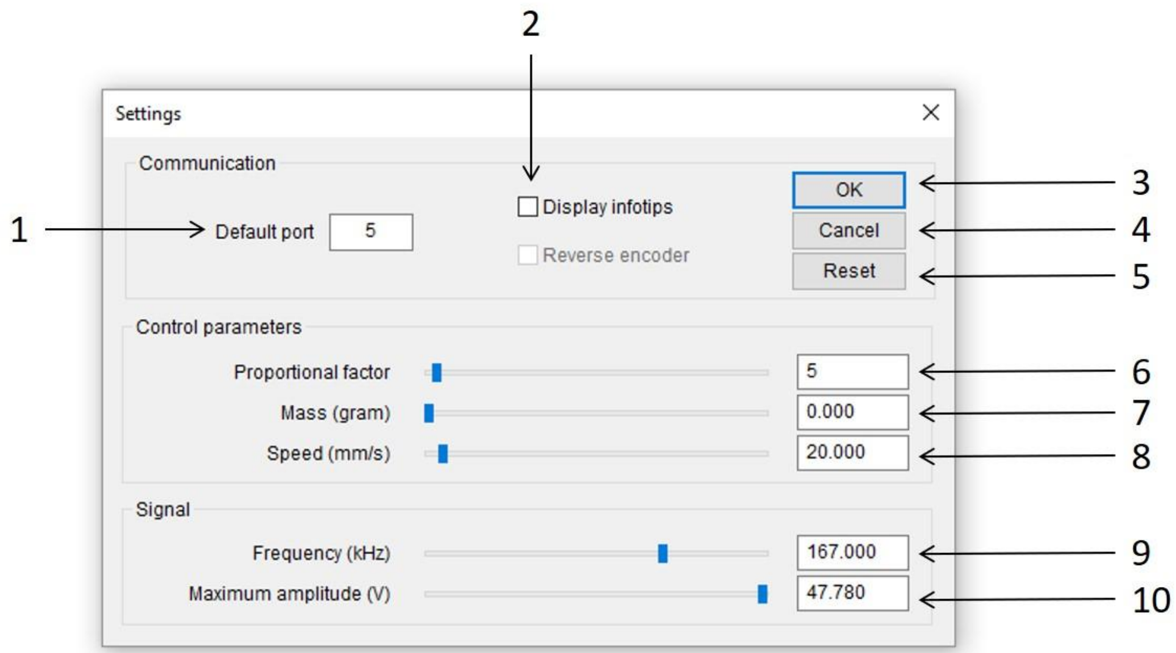
2 Some of the buttons may be disabled when you start the windows interface. Please follow the next steps to configure the driver. This will enable all buttons.



Function of buttons and status displays:

1 Communication error between the computer and driver.
2 Zero output: the voltage to the piezo motor will be set to zero.
3 Encoder error. Please contact Xeryon's customer service for assistance.
4 The voltage to the piezo motor is zero.
5 The "position reached" LED will light up when the desired position is reached.
6 The stage will stop moving when you click the "Stop" button.
7 Reset the settings and driver.
8 Run automatically the selected program. The green LED "Program running" will light up when the program is running. Via the browse button, one can search and select the program that is being executed (.txt file). Remark: the ASCII commandos to write the program are described in the manual of the driver.
9 Set the step size when you press the "Step -X" and "Step +X" button.
10 Set the desired position.
11 Actual position of the stage.
12 The stage will continuously move in the -X direction.
13 Move to the index position.
14 Step in the +X direction with the step size defined in #9.
15 Settings button:
- Load default settings: load the default settings from the "settings_default.txt" file
- Load from file: select a file to load your own user settings
- Edit: open the settings window (see below)
- Save to file: save the settings to a file you select

Settings window:



Function of buttons and status displays:

1. Set the default communication port number
2. Display info balloons when you hover over a button or text box
3. Close the settings window and keep the (adapted) settings. The settings will not be saved or overwritten in the settings file.
4. Close the settings windows and reject all changes.
5. Reset to the original settings.
6. Set the proportional control factor.
7. Set the mass of the external load on top of the stage to obtain a stable control.
8. Set the speed.
9. Set the excitation frequency of the piezo motor (typically between 166 en 168 kHz).
10. Set the maximum voltage for excitation of the piezo motor.

16. Python, MATLAB, C++ and LabVIEW.

Interaction with the Xeryon controllers and connected stages is also possible through Python, MATLAB, LabVIEW and C++.

All libraries are available from our website under “Downloads”, with a sample program.

16.1. Python & MATLAB

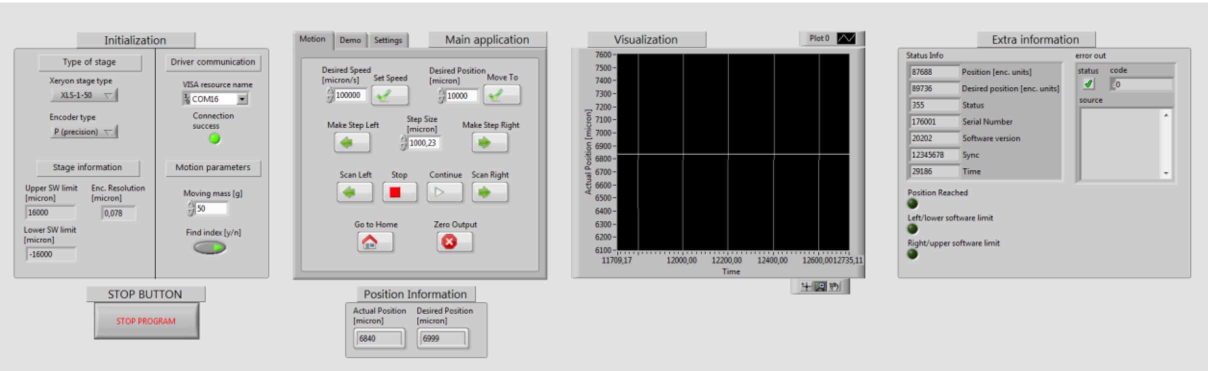
These files are available in one folder. There is an example for both Python and MATLAB. You can read more about our Python & MATLAB library on our website.

16.2. C++

We also have a C++ library available. This makes it very easy to control our stages using C++. You can read more about this library on our website.

16.3. LabVIEW

A simple LabVIEW program is available for download. A screenshot of the interface for the XLS linear stage is shown below. You can use all the individual .vi’s to make your own program. You can read more about our LabVIEW program on our website.



17. Customer service

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