



OPTO ENGINEERING

Motion controller for bipolar stepper motors

MTDV series



INSTRUCTION MANUAL



V 1.3 - eng

INDEX

1 • GENERAL INFORMATION.....	4
1.1 Installation, use and maintenance manual.....	4
1.2 Product identification data	5
1.3 Package contents	5
2 • SAFETY NOTES	6
2.1 General disclaimer	6
2.2 Forbidden use.....	6
3 • PRODUCT WARRANTY.....	8
3.1 Warranty.....	8
3.2 CE certification.....	8
4 • STORAGE AND USE.....	9
4.1 Storage conditions	9
4.1.1 Temperature.....	9
4.1.2 Humidity.....	9
4.2 Operating conditions.....	9
4.2.1 Temperature.....	9
4.2.2 Humidity.....	9
5 • CHARACTERISTICS AND DESCRIPTION	10
5.1 Working principle	10
5.2 Block diagram	10
5.3 Technical specifications.....	11
5.3.1 Mechanical dimensions.....	11
5.3.2 Specifications.....	11
5.3.3 Circuit protections.....	13
5.4 Front panel connectors and pinout.....	13
5.4.1 Power connector	13
5.4.2 Serial RS485 port.....	14
5.4.3 USB port	14
5.4.4 Ethernet port	15
5.4.5 Motion connector.....	15
5.4.6 Status RGB LED.....	19
5.5 Wiring diagrams	20
6 • Ethernet communication	22
6.1 Configuration and preliminary operations	22

6.2 Web application overview	22
6.2.1 Status information	23
6.2.2 Operation tab	23
6.2.3 Demo tool tab.....	25
6.2.4 Configuration tab	26
6.2.5 Import/Export XML tab.....	28
6.3 Command list.....	29
6.4 DLL control	29
6.4.1 Basic data structure	29
6.4.2 Functions prototypes.....	30
6.5 Firmware update	38
7 • Serial communication	40
7.1 USB communication	40
7.2 RS485 communication	40
8 • Troubleshooting	41
9 • Compatible products and accessories	43
9.1 Compatible OE products	43
9.2 Accessories.....	43

1 • GENERAL INFORMATION

1.1 Installation, use and maintenance manual

Opto Engineering s.r.l., with its registered office in Strada Circonvallazione Sud 15, 46100 Mantova (Mn) - Italy, hereinafter the **manufacturer**, provides all the necessary information in this installation, use and maintenance manual in a clear and simple way to install, use and service product MTDVxCH-XXXX.

The **recipients of this manual** are all those who have the knowledge, experience and capability of understanding the standards, prescriptions and safety measures indicated in this manual. Such people will be later identified as **qualified personnel** who are authorized to transport, install, use and service the products described in this manual.

This material can only be used by the customer whom this manual has been delivered to, in order to install, use and service the product.

The manufacturer will retain the right to modify or improve the manual and/or the product referred to in this manual without any prior notice.

The following typographical conventions are used in this document:

NOTE: notes contain important information. Highlighted outside the text to whom they refer.



CAUTION: These indications highlight procedures that, if not observed in their entirety or in part, can cause damage to the machine or to the appliances.



HAZARD: These indications highlight procedures that, if not observed in their entirety or in part, can cause injuries or affect the health of the operator.

1.2 Product identification data

The MTDVxCH-xxAx is identified with a label placed on the lateral side of the packaging box.

This label is used to identify part number, serial number and the MAC address of each device. This set of data is used to identify a single device delivered by the manufacturer.

- 1) **Manufacturer:** Opto Engineering srl.
- 2) **P/N** is the part number of the device: MTDVXCH-WYAZ. In particular:
 - o X denotes the number of the channels of the device;
 - o W denotes the communication interface;
 - o Y denotes the communication protocol;
 - o Z denotes the output maximum current.
- 3) **S/N** is the serial number of the device (5-digit number)
- 4) **MAC Address** is a 12 hexadecimal characters code
- 5) which is a unique identifier assigned to a network interface controller.
- 6) Opto Engineering logo
- 7) Compliance markings (CE marking and RoHS compliance)

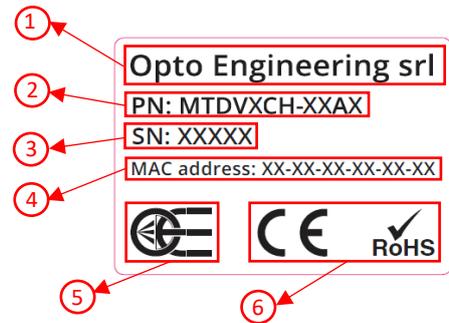


Figure 1.2.1 Product label

1.3 Package contents

The content of delivery consists of:

- MTDV module
- 1 terminal block (4 positions) for the power connection
- 1 terminal block (3 positions) for the RS485 connection

Other compatible products are sold separately (see "Compatible products" and "Accessories" sections, respectively at 9.1 and 9.2).



CAUTION: any discrepancy in the contents of the package or integrity of the packaging must immediately be reported to the manufacturer, upon receipt of the goods.

2 • SAFETY NOTES

2.1 General disclaimer

Always use and store Opto Engineering® products in the prescribed conditions in order to ensure they function properly: failing to comply with the following conditions may shorten the product lifetime and/or result in malfunctioning, performance degradation or failure.

Make sure that incorrect equipment operation cannot generate a hazardous situation or significant financial losses. It is essential that the user ensures that the operation of the controller is suitable for the application in question.

All trademarks mentioned herein belong to their respective owners.

Except where prohibited by law:

→ All hardware, software and documentation are provided on an “as is” basis.

→ Opto Engineering® accepts no liability for consequential loss, of any kind.

Upon receiving your Opto Engineering® product, visually examine the product for any damage during shipping. If the product is damaged upon receipt, please notify Opto Engineering® immediately.

2.2 Forbidden use

Please read the following notes before using this controller. Contact your distributor or dealer for any doubts or further advice.

Do not disassemble, modify or repair the product yourself. It may cause permanent malfunctioning or even fire or electric shock, possibly resulting in serious injury.

Do not place the product in dusty, humid or hot places or near flames. These conditions may cause malfunctioning and damage or even fire or electric shock, possibly resulting in serious injury.

Do not spray insecticide or apply other volatile chemicals on or around the product.

This device must not be used in an application where its failure could cause a hazard to human health or damage to other equipment. Keep in mind that if the device is used in a manner not foreseen by the manufacturer, the protection provided by its circuits and by its enclosure may be impaired.

This is a low voltage power supplied device. As such, the potential difference between any combination of applied signals must not exceed the supply voltage at any time.

Higher voltages may cause a fault and can be dangerous to human health.

This device has limited protection against transients caused by inductive loads. If necessary, use external protection devices like fast diodes or better still, specific transient protectors.

Do not allow foreign objects to enter the unit or drop into holes, terminals and other openings or gaps. This may cause fire or electric shock, possibly resulting in serious injury.

Disconnect the power cable before moving the product. Failure to comply with this precaution may damage the power cable, cause fire or electric shock, possibly resulting in serious injury.

Do not scratch, cut, open or twist the power cables. It may cause malfunctioning, fire or electric shock, possibly resulting in serious injury.

If the power cable is damaged or cracked, please contact our technical support and do not use the product. Damaged cables may cause malfunctioning, fire or electric shock, possibly resulting in serious injury.

Do not insert or remove the plug of the power cable with wet hands. It may cause electric shock, possibly resulting in serious injury.

Do not use the product in the presence of inflammable gas. It may cause outbreaks and flames, possibly resulting in serious injury.

Should you notice any abnormality such as a smell, smoke or overheating, turn off the power and disconnect the power cables. Continuing to use the product in these conditions may cause fire or electric shock, possibly resulting in serious injury.

If you have dropped the product or damaged the product case, turn off the power and disconnect the power cables. Continuing to use the product in these conditions may cause fire or electric shock, possibly resulting in serious injury.

3 • PRODUCT WARRANTY

3.1 Warranty

The device warranty is 12 months from the effective delivery date with reference to the device serial number.

The warranty covers the replacement or repairs of the defective part (components, device or part of it) with the exclusion of dismantling and shipping costs.

The replacement of one or more components does not renew the warranty period of the entire device. The manufacturer cannot be held liable for any compensation for whatever reason and the buyer renounces any claims for costs or damages to third parties due to any machine downtime.

The electronics and parts subjected to normal use or deterioration due to atmospheric agents and the external environment are excluded from the warranty. Also, all failure caused by the lack of, insufficient or incorrect maintenance performed by unskilled or unauthorized personnel or due to unintended use or unauthorized replacements, alterations or repairs is excluded from the warranty.

The general validity of the warranty depends on:

- Maintenance being performed correctly as described in the device manual;
- The intended use of the device as specified in this manual.

3.2 CE certification

Opto Engineering srl, with registered office in Strada Circonvallazione Sud 15, 46100 Mantova (Mn), Italy, states that the products described in this manual conform to the following EU directives and their amendments:

- 2011/65/EU (RoHS)
- 2014/30/EU (EMC)



2012/19/EU (WEEE directive): Products marked with this symbol cannot be disposed of as unsorted municipal waste in the European Union. For proper recycling, contact Opto Engineering srl which will provide the support for disposal.



4 • STORAGE AND USE

4.1 Storage conditions

4.1.1 Temperature

Storage environment between -20°C and 60°C.

Avoid thermal shock by not exposing the product to sudden changes in temperature.

4.1.2 Humidity

Store the product in a dry place: storage environment with relative humidity (RH) less than 85% (no condensation).

4.2 Working conditions

4.2.1 Temperature

Operating environment between 0°C and 40°C.

Extreme temperatures affect the functionality of the product, especially the electronic components.

Avoid thermal shock by not exposing the product to sudden changes in temperature.

Since the product includes electronic components, it may generate heat when functioning. Always allow enough room to dissipate the appropriate amount of heat (if necessary, operate the device with a forced air cooling system).

4.2.2 Humidity

Operate the product in a dry place: operating environment with relative humidity (RH) less than 85% (no condensation).

5 • CHARACTERISTICS AND DESCRIPTION

5.1 Working principle

The MTDV is a **motion controller** module for bipolar stepper motors with winding currents up to 1.35A rms (adjustable).

The MTDV can drive **up to four** stepper motors and has been developed to control aperture, focus and zoom of motorized lenses via **Ethernet** or **RS485/USB** interface of a PC/PLC system.

The built-in electronics has been designed specifically to accurately control the standard parameters of bipolar stepper motors: acceleration, deceleration, speed, winding current, etc...

Each motion channel has the ability to read the output of an **incremental A/B/Z encoder** with **RS422** interface. It's therefore possible to achieve a very precise positioning by means of advanced **closed-loop control**.

A demo user interface with control functions is available via web browser application. The MTDV can also be fully controlled with a dedicated *DLL* library, downloadable from the Opto Engineering® website.

There are different versions of MTDV, differing in the number of motion channels: one, two, three and four stepper motors can be controlled by the same device, each with its related encoder.

5.2 Block diagram

The modular architecture of the MTDV series is composed by:

- 1x **main board**
- 1/2/3/4x **driver boards**, depending on the number of channels

An example of block diagram scheme (for the two channels version) is here illustrated:

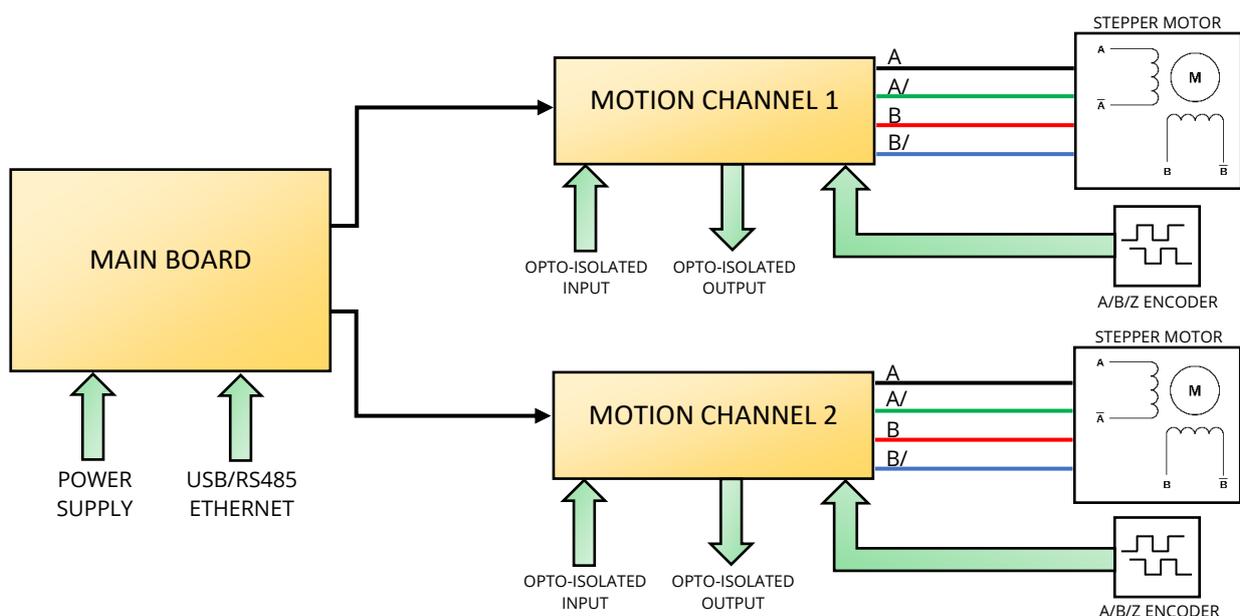
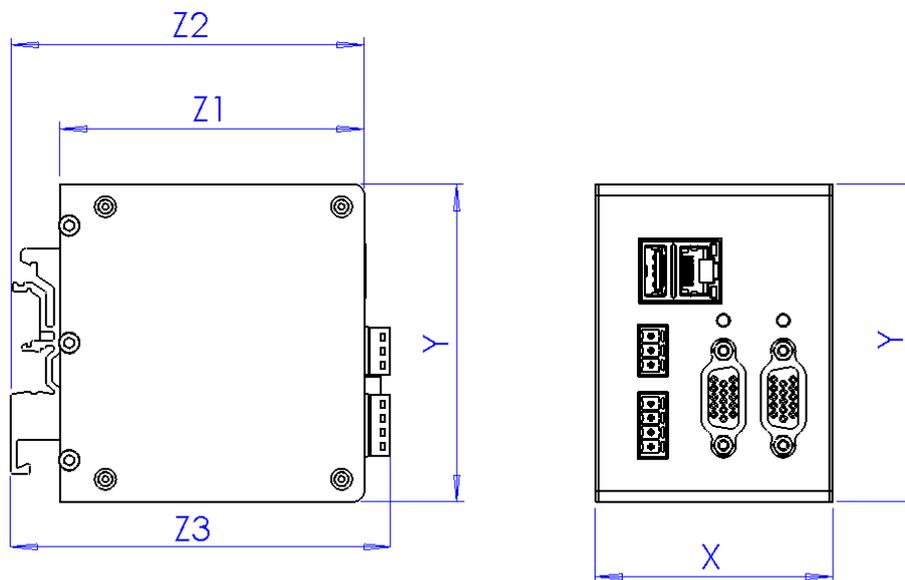


Figure 5.2.1 MTDV2CH block diagram

5.3 Technical specifications

5.3.1 Mechanical dimensions



	1 CH	2 CH	3 CH	4 CH
X (mm)	47.0	62.8	78.6	94.4
Y (mm)	84.0	84.0	84.0	84.0
Z1 (mm)	80.2	80.2	80.2	80.2
Z2 (mm)	92.8	92.8	92.8	92.8
Z3 (mm)	102.6	102.6	102.6	102.6

The MTDV series has been designed specifically following a modular architecture; from the MTDV1CH to the MTDV4CH the only difference in the product dimensions is visible along the *x-axis*: every channel added to the motion controller increases the *x-axis* dimension of 15.8 mm. The MTDV is equipped with a DIN rail support useful to settle the motion controller on a standard DIN rail. No other clamping mechanisms have been designed for this product.

5.3.2 Specifications

Model	MTDV1CH-22A2	MTDV2CH-22A2	MTDV3CH-22A3	MTDV4CH-22A4
<i>Electrical specifications</i>				
DC Voltage (V)	24			
Power consumption max (W)	38	71	96	96
Communication interfaces	Ethernet (TCP/IP), USB2.0 (custom serial), RS485 (custom serial)			
Connectors	1x RJ45, 1x USB Type A, 2x Terminal blocks, Nx DB15HD female 1			

Visual indicators	Ethernet activity LED, RGB LED for each motor status
Protections	ESD, polarity inversion, overvoltage, undervoltage, output overcurrent, shorted output, open output, overtemperature

Functions

Non volatile memory	Yes
Automatic position saving	Yes
Software	Integrated web interface (Ethernet), DLL library (Ethernet, USB and RS485)
Control type	Closed-loop, open-loop
Motion mode	Positioning mode (relative and absolute) with 2 stage velocity ramp
Special features	Advanced motor drive with StealthChop™ and 256x SpreadCycle™, extremely silent operation, adjustable RUN and HOLD currents for optimized power consumption, advanced positioning control in closed-loop operation, motor homing based on encoder reference or limit switch, reduced EMI noise ²

Motor parameters ³

Number of motors		1	2	3	4
Type		Bipolar stepper			
RMS winding current	(mA)	20 to 1350 (adj)	20 to 1350 (adj)	20 to 1250 (adj)	20 to 935 (adj)
Peak winding voltage	(V)	24 (PWM)			

Encoder parameters ³

Number of encoders		1	2	3	4
DC power supply	(V)	5			
Type		Linear / rotary, incremental			
Signal outputs		A, B, Z (index)			
Interface		RS422			

Environmental specifications

Operating temperature	(°C)	0 to 40			
Storage temperature	(°C)	-20 to 60			
Humidity	(%)	10-85 non condensing			
IP rating		-			
Installation		Indoor use only			

Mechanical specifications

Length	(mm)	47	62.8	78.6	94.4
Height	(mm)	84			
Width (case only)	(mm)	80.2			
Width (total)	(mm)	102.6			
Material		Anodized aluminum			
Mounting		DIN rail			

Compatibility ⁴

Lenses	TCZRXXX	TCZRXXX, ENMT, MZMT12x
--------	---------	------------------------

Cable 5

 CBMT002,
 CBETH003

CBMT002, CBMT003, CBETH003

Accessories

ADPT001

5.3.3 Circuit protections

The built-in electronics of the MTDV exhibits some protections against electrical and temperature faults:

- Protection diodes against electrostatic discharge (ESD), polarity inversion, over-voltage and over-current on the power supply port
- Circuits for under-voltage detection on the power supply port
- Circuits for over-temperature detection
- Electronic control of the winding motor current (output over-current and output short circuit protection).

5.4 Front panel connectors and pinout

The MTDV exhibits the following connectors and indicators:

No	DESCRIPTION
1	Power connector
2	Serial RS485 port
3	USB port
4	Ethernet port (with activity LEDs)
5	Motion connector (DB15HD female)
6	Status RGB LED

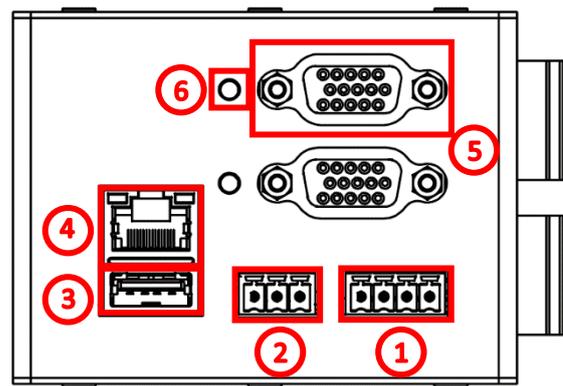


Figure 5.4.1. Front panel

A detailed description of each connector pinout and electrical specifications are reported in the following sections.

5.4.1 Power connector

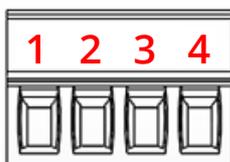


Figure 5.4.1.1 Power connector

PIN	NAME	DESCRIPTION
1	+24V	+24V DC ($\pm 5\%$)
2	GND	0V – ground reference
3		PE - Protection Earth
4		PE - Protection Earth

The device requires a DC power supply (direct current) with a voltage of $24V \pm 5\%$ (pin 1 and 2), able to deliver up to 100W (refer to *Electrical Specifications* in section 5.3.2). The power absorbed by the device may vary depending on the number of available motion channels and peripheral devices

connected to it. An **efficient earth connection** (pin 3 or 4) is also required to ensure product safety and compliance with EMC regulations.

Always check that the power supply line matches the voltage, frequency and electrical absorption of the system.

The power cable must be in pristine condition. Do not bend it or pull it, which can interrupt the conductors.

Any connectors for the connection to the power supply must be able to withstand the electrical absorption of the system with a suitable margin.

Ensure the polarity of +24V and GND (0V) is correct before turning the device on.



HAZARD: ONLY use the device with a suitable power supply and an earth connection. Failure to follow these instructions can damage the product and/or injure the operator.

5.4.2 Serial RS485 port

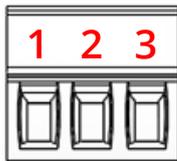


Figure 5.4.2.1 Power connector

PIN	NAME	DESCRIPTION
1	D+	RS485 positive differential positive signal
2	D-	RS485 negative differential positive signal
3	GND	0V – ground reference

The device has an RS485 communication interface with protocol that can be implemented on request. This can be used to communicate with any PLCs and/or other peripheral devices in the system.

The port is **not galvanically isolated** and so a common earth wire is needed, in addition to a twisted differential data pair used for data exchange. For this reason, it is essential that the various devices connected to the RS485 line have the same earth potential (or have a power supply isolated from the others).

We recommend using a **shielded cable** with internal wires that are grouped in twisted pairs to connect the devices to the RS485 line. This way, there is maximum rejection of electromagnetic disturbances coming from the surrounding machinery.



CAUTION: a difference in electrical potential between the grounds of the devices that are connected to the RS485 line can cause malfunctions and/or faults in the devices themselves.

5.4.3 USB port

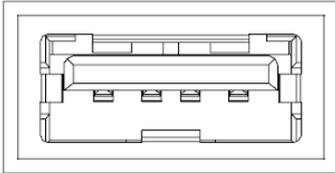


Figure 5.4.3.1 USB connector

The device has a USB port which is USB1.1 compliant and employs a standard Type-A connector. The serial communication protocol can be implemented on request. This can be used to communicate with any PC and/or other peripheral devices in the system.

NOTE: this is a device USB port, not a host port.

The port is **not galvanically isolated**. For this reason, it is essential that the different devices connected to the USB bus have the same earth potential (or have a power supply isolated from the others).

We recommend using a **shielded cable** compliant with USB1.1 specification.



CAUTION: a difference in electrical potential between the grounds of the devices that are connected to the USB bus can cause malfunctions and/or faults in the devices themselves.

5.4.4 Ethernet port

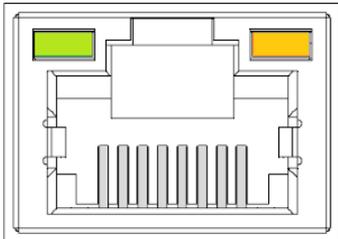


Figure 5.4.4.1 Ethernet connector

The device is equipped with a standard RJ45 connector used for 10/100 Ethernet communication. In order to guarantee the proper operation of the device we recommend using **CAT5 or higher shielded cable** compliant with 100BASE-TX specifications. It is advisable to use cables designed specifically for the industrial environment, with excellent electrical and mechanical performances. For more information regarding the Ethernet connection refer to chapter 6.

NOTE: the ethernet port is galvanically isolated and no special precautions are required if the device is connected to another 10/100/1000BASE-TX compliant device.



CAUTION: be sure to connect the device only to 10/100/1000BASE-TX compliant devices. Failure to follow these instructions can cause malfunctions and/or faults in the devices themselves.

5.4.5 Motion connector

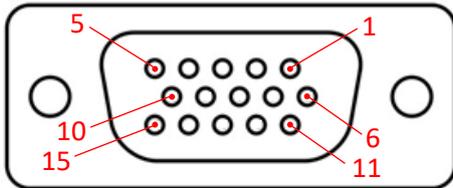


Figure 5.4.5.1 Motion connector

PIN	NAME	DESCRIPTION
1	5V	5V encoder power supply
2	GND	0V reference ground
3	ENC_A +	Encoder quadrature signal – A +
4	ENC_B +	Encoder quadrature signal – B +
5	ENC_Z +	Encoder quadrature signal – Z +
6	MOT_A +	Motor – Phase A +
7	MOT_B +	Motor – Phase B +
8	ENC_A -	Encoder quadrature signal – A -
9	ENC_B -	Encoder quadrature signal – B -
10	ENC_Z -	Encoder quadrature signal – Z -
11	MOT_A -	Motor – Phase A -
12	MOT_B -	Motor – Phase B -
13	OPTO_OUT	Opto-isolated output signal
14	OPTO_IN	Opto-isolated input signal
15	OPTO_GND	Opto-isolated 0V reference ground

Each motion channel has an I/O connector (DB15HD, 15 pins D-SUB high density connector) which carry all the signals used to control the positioning of a motorized axis. These are the stepper motor power signals, encoder RS422 interface (along with its power supply), one opto-isolated input and one opto-isolated output.



CAUTION: Always load the proper XML configuration file BEFORE connecting any Opto Engineering lens to the MTDV controller. Failing to do so can lead to damage due to incorrect motor drive.

Motor interface

The motor power signals consist of phase A and phase B coil drive current for a standard bipolar stepper motor.

ELECTRICAL SPECIFICATION	MIN	NOM	MAX
Output voltage		24 V	
Output current (adjustable)	0 mA		1350 mA*

*Overall maximum output current (all channels) = 3750 mA.



CAUTION: be sure to connect only bipolar stepper motors which are compatible with the MTDV electrical specifications. Failure to follow these instructions can cause malfunctions and/or faults in the device or the motor.

Encoder interface

The **encoder interface** consists of 3 RS422 differential inputs which can be used to connect an A/B/Z quadrature encoder. It also does provide a 5V power output which can be used to directly power the encoder circuitry (along with the GND reference ground). In any case the encoder power supply must be referenced to the GND signal. All the differential inputs meet or exceed Standard TIA/EIA-422-B and ITU Recommendation V.11.

ELECTRICAL SPECIFICATION	MIN	NOM	MAX
Encoder supply voltage	4.9 V	5 V	5.1 V
Encoder supply max current			100 mA
RS422 differential input voltage	-7 V		7 V
RS422 common mode input voltage	-7 V		7 V
High-level input voltage	2 V		5.5 V
Low-level input voltage	0 V		0.8 V

The equivalent circuit schematic (for channel A) is illustrated in the following figure:

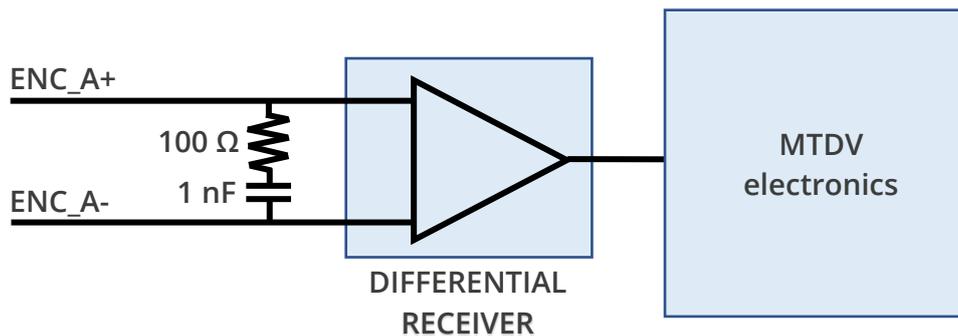


Figure 5.5.2.1 Encoder RS422 equivalent circuit

The differential input line is terminated on a 100 Ω resistor in series with a 1 nF capacitor. This will ensure very clean signal edges during fast commutations and near-zero power dissipation in static conditions.



CAUTION: be sure to connect only encoders which are compatible with the MTDV electrical specifications. In particular, check that the power supply requirements of the encoder meets the specifications described above. Failure to follow these instructions can cause malfunctions and/or faults in the device or the encoder.

Synchronization I/Os

There are also synchronization **digital input and output** (OPTO_IN and OPTO_OUT), both opto-isolated. Input and output share a common reference ground (OPTO_GND) in order to minimize the pin requirements of the connector.

The **input** pin can be used to connect a limit-switch or another kind of sensor which provide a reference for the motion channel homing procedure.

ELECTRICAL SPECIFICATION	MIN	NOM	MAX
Input voltage	0 V		24 V
Input current	5 mA		10 mA
High-level input voltage	5 V		24 V
Low-level input voltage	0 V		2 V

The equivalent circuit schematic is illustrated in the following figure:

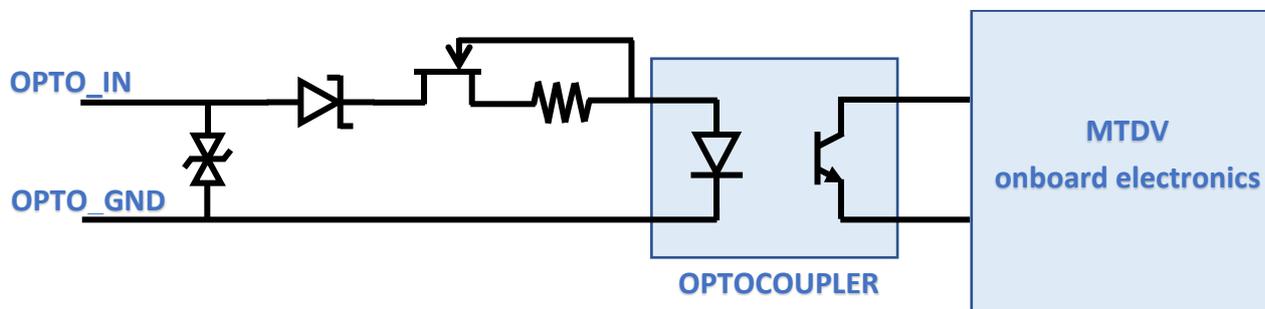


Figure 5.4.5.1 Opto-isolated input equivalent circuit

The input port is equipped with:

- ESD protection circuit
- Polarity inversion protection circuit
- Current limitation circuit

While in standard operating conditions the OPTO_IN is kept at LOW logic level or not connected, to trigger the system it's necessary to bring the OPTO_IN pin to a HIGH logic level.

The polarity of the input can be inverted by changing the appropriate configuration parameter. See section 6 for more information.

The delay time between the input signal transition (considering a step transition) and the effective system triggering is about 3.5 μ s.



CAUTION: be careful to not exceed the nominal specifications of the opto-isolated input port.

The **output** pin signals when the motor has reached the desired position by means of a rectangular pulse of 1 ms.

The equivalent circuit schematic is illustrated in the following figure:

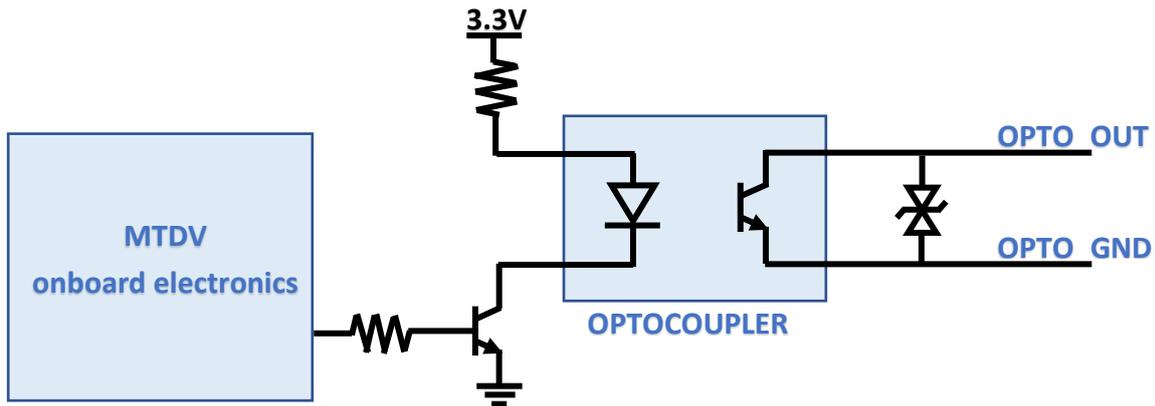


Figure 5.4.5.2 Opto-isolated output equivalent circuit

The output port is equipped with:

- ESD protection circuit

While in standard operating conditions the OPTO_OUT exhibit a high-impedance and no current can flow into the pin, when the output synchronization signal is present, OPTO_OUT is shorted to OPTO_GND by the output transistor (open-collector NPN output), thus allowing a current to flow.

The delay time between the start of the synchronization and the actual output signal transition is about of 3.5 μ s.



CAUTION: be sure to respect the signals polarity applied to the opto-isolated I/O pins. More specifically, the output circuit is polarized and doesn't tolerate inverse polarity connections.



HAZARD: do not intentionally cause short-circuits and/or wrong connections. Even if the outputs have several protection circuits, it's not possible to guarantee reliable operation with all the possible wrong cabling combinations.

5.4.6 Status RGB LED

The RGB LED indicator (next to the motion connector) is used to give a direct feedback of the motion controller operations.

The RGB LED codification is illustrated in the following table:

LED color	DESCRIPTION
None	The motor is standstill: position reached.
Blue	The motor is moving.
Red	Hardware error: check the motor and encoder connections.

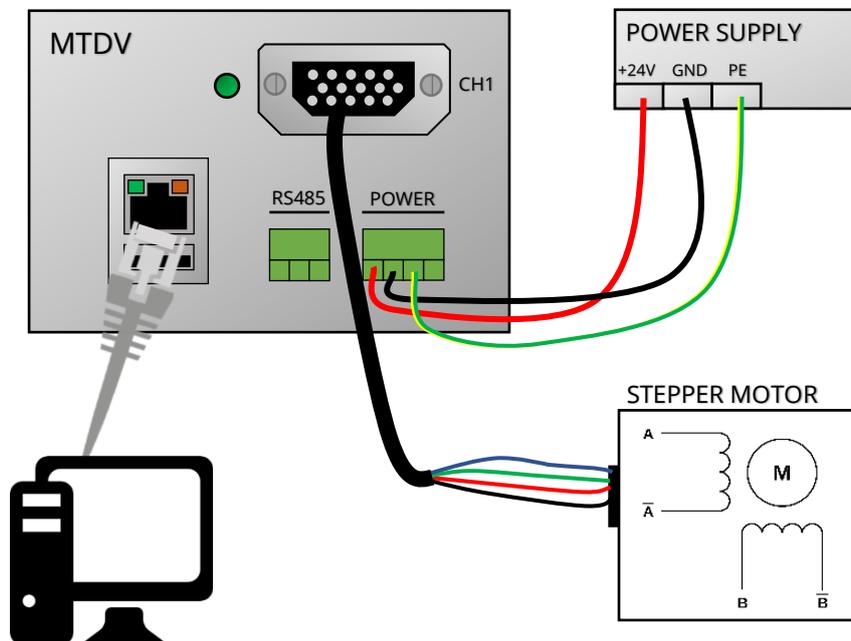
5.5 Wiring diagrams

In this section will be showed some wiring configuration examples of the motion controller module.

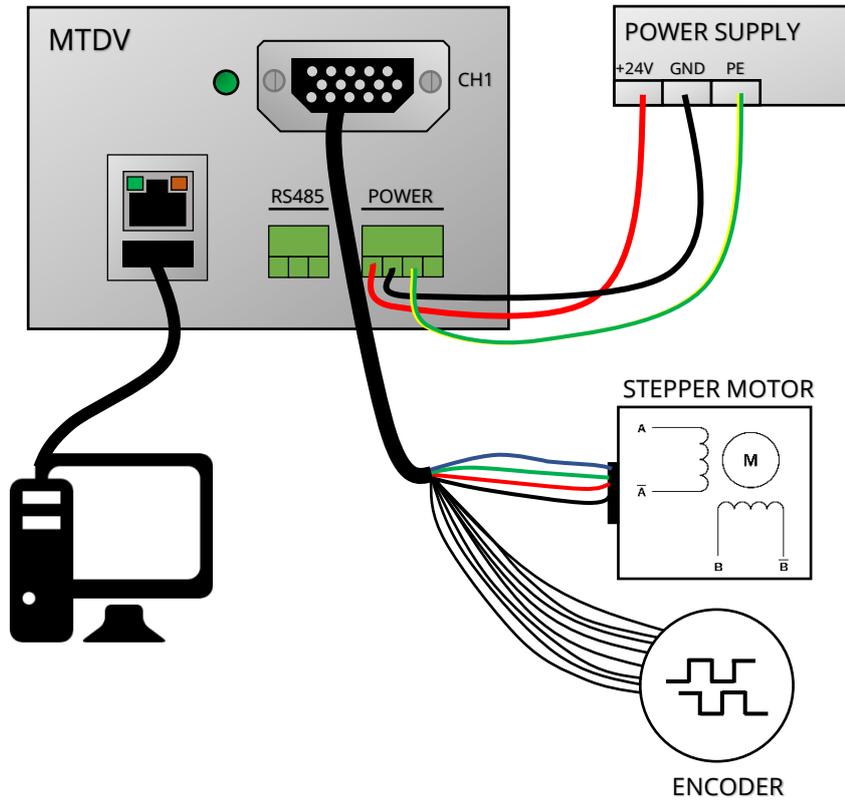


CAUTION: Always load the proper XML configuration file BEFORE connecting any Opto Engineering lens to the MTDV controller. Failing to do so can lead to damage due to incorrect motor drive.

Example 1: driving a stepper motor in open-loop (without encoder) using Ethernet control (both web application or .dll libraries).



Example 2: driving a stepper motor in closed-loop (with encoder) using USB interface.



6 • ETHERNET COMMUNICATION

6.1 Configuration and preliminary operations

The MTDV can be controlled either via ethernet or serial communication (USB and/or RS485). This paragraph describes the ethernet control method.

If the device is accessed for the first time, a preliminary configuration is required.

The default IP address of the MTDV is set equal to **192.168.1.200**. Therefore, the network card properties of the PC must be changed in order to start the communication with the controller.

In particular, the following parameters have to be set in the Network Control Panel of the PC (TCP/IPv4):

- IP Address: **192.168.1.XXX** (where XXX is a number from 1 to 255, different from 200)
- Subnet mask: **255.255.255.0**

NOTE: *be careful to avoid possible conflicts between IP addresses of different devices. Since the default IP address of the MTDV is equal to 192.168.1.200, don't use XXX = 200 in the network properties of the PC for the first use of the controller.*

However, the IP address can be changed at any moment. This can be done in different ways:

- using the web application (see section 6.2)
- using the dedicated .dll (see section 6.3)
- through the serial communication (USB or RS485) (see chapter 7 for further information)



CAUTION: *be careful, if the IP address is changed by the user and gets accidentally lost or forgotten, the web application cannot be accessed until:*

- *The IP address is resetted using the MTDV Firmware Update Utility (only for main firmware version ≥ 2.1)*
- *The IP address is retrieved using the USB or the RS485 functions*
- *The IP address is changed using the USB or the RS485 functions*

After the preliminary network configuration, the web-based interface can be easily accessed by entering the IP address 192.168.1.200 in the browser home page (or the new IP address, if changed).

Mozilla Firefox, Google Chrome and Microsoft Edge fully support the MTDV web application. Other browsers have not been tested yet.

NOTE: *please download the latest version of your browser application in order to ensure the best operation of the device.*

6.2 Web application overview

The web application is a platform that includes most of the functions implemented in the .dll library and in the USB/RS485.

The advantages of the web application are the following:

- includes all the functionalities of the motion controller but doesn't require the installation of additional software;
- it's not dependent from the PC operative system;
- it's easy and fast to access;
- includes a user-friendly and quickly understandable interface to drive the motors.

6.2.1 Status information

On the left side of the web interface the device identification data and current status are displayed.

The identification data of the MTDV can be summarized in part number (PN), firmware version, MAC address, IP address and subnet mask.

The temperature displayed on the web page refers to the internal temperature of the MTDV: note that this is NOT the actual temperature of the motors connected to the device, which can be considerably higher.

The voltage indicators allow the user to verify that the system has the correct supply voltage. In particular, in case of undervoltage in the power supply, the web page will notify this error and the system won't work properly. Please be careful to always work respecting the nominal conditions.

PN	MTDV1CH-22A2
FW Version	1.6
MAC Address	88:5F:E8:20:00:33
IP Address	192.168.1.200
Subnet Mask	255.255.255.0
Temperature	42.0 °C
Supply Voltage	23.87 V
Logic Voltage	5.03 V

Figure 6.2.1.1 MTDV identification data and status

6.2.2 Operation tab

The positions of the motion axis can be directly managed in the *Operation* tab.

On this page are displayed all the motion channels present in the motion controller. The following pictures, for example, report the web interface of a 2 channels device.

The channel number is coherent with the one reported near the physical connectors on the MTDV front panel.

For each device, next to the operation buttons, there is also a graph which shows the evolution of the motor position in function of time. In case of rotary encoder, overcoming the last position is equal to reset the position itself.

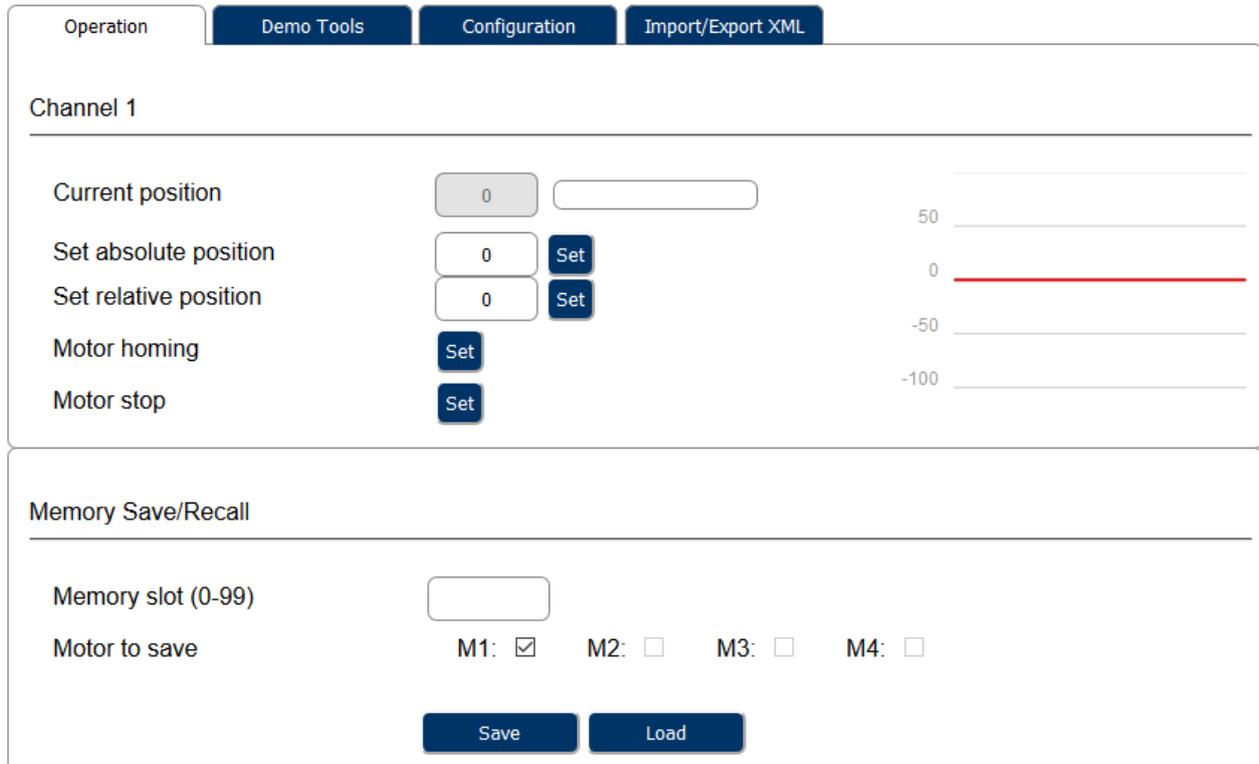


Figure 6.2.2.1 Web interface operation tab

To **move** one motor, write the desired absolute or relative (to the current one) position and press the corresponding *Set* button. In **open-loop** mode, the position is expressed in μ steps, made by the motors (each motor step is divided into 256 μ steps). In **closed-loop** mode the position is expressed in encoder steps.

The effective current position is displayed in the grey box and plotted on the corresponding graph.

To perform a **homing** procedure, press the *Set* button near the *Motor homing* label. The motor starts to rotate at constant speed (set in the *Configuration* tab) until one of the following conditions become true:

- In open-loop mode:
 - The *homing timeout* time is elapsed (it can be used with a mechanical reference which physically stops the motor)
 - A limit-switch signal is received on the opto-isolated input OPTO_IN
- In closed-loop mode:
 - The *homing timeout* time is elapsed
 - The encoder reaches the position which corresponds to its reference mark

At the end of the homing procedure the position is reset to zero.

To **stop** the motion, press the *Set* button near the *Motor stop* label.

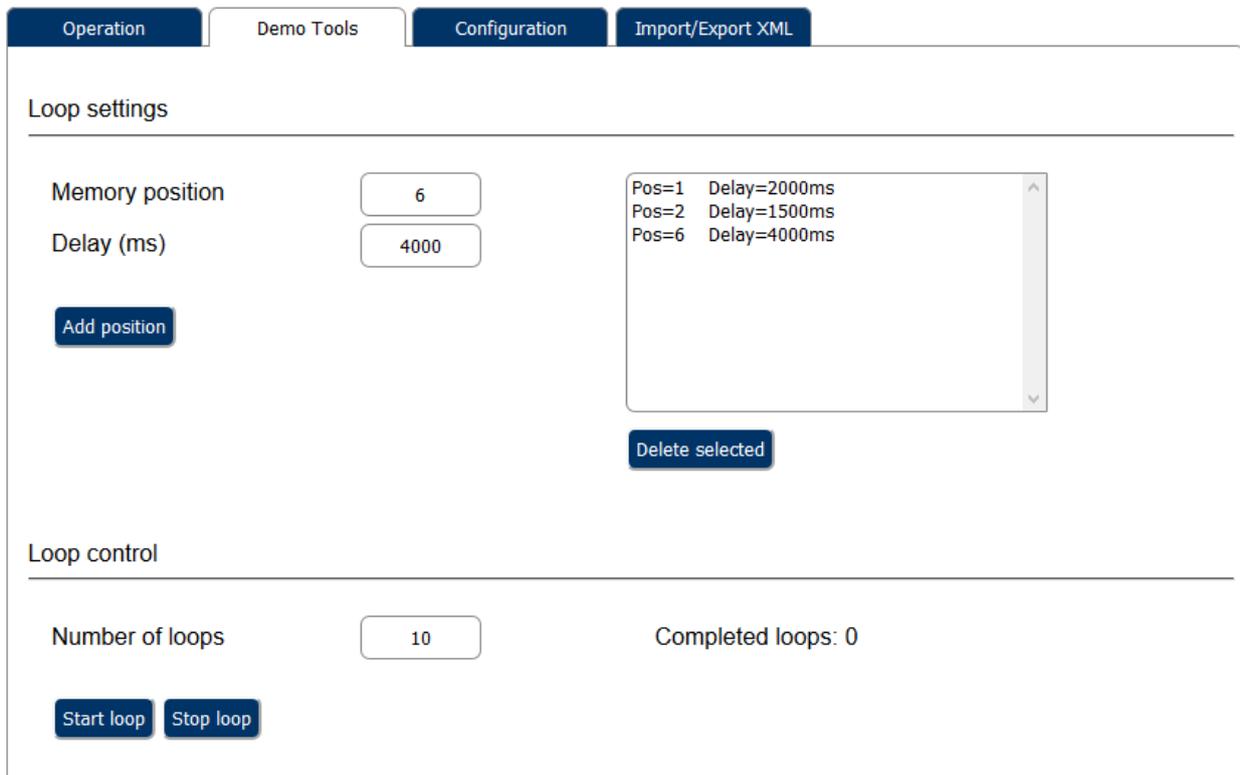
In the *Memory Save/Recall* section its possible to save or load a position of one or more motors.

To **save** the current position, insert one of the available memory slots (0-99), select the motion channels to save by checking one (or more) of the M1-M4 boxes and press the *Save* button.
 To **load** one of the saved positions, insert one of the available memory slots (0-99), and press the *Load* button.

6.2.3 Demo tool tab

The *Demo Tools* tab helps the user to implement test procedures and user-defined loops, cycling between different target positions. This is useful when it's necessary to quickly evaluate a motion sequence.

NOTE: do not rely on the Demo Tools controls in order to perform reliable and precise motion sequences, since they are strongly dependent on the browser timing functions and network management (not suitable for reliable operations).



The screenshot shows the 'Demo Tools' tab in a web interface. It features four main sections: 'Operation', 'Demo Tools', 'Configuration', and 'Import/Export XML'. The 'Demo Tools' section is active and contains two main areas: 'Loop settings' and 'Loop control'.

Loop settings:

- Memory position:** A text input field containing the value '6'.
- Delay (ms):** A text input field containing the value '4000'.
- Add position:** A blue button to add a new position to the list.
- Position List:** A scrollable list box containing three entries:
 - Pos=1 Delay=2000ms
 - Pos=2 Delay=1500ms
 - Pos=6 Delay=4000ms
- Delete selected:** A blue button to remove a selected position from the list.

Loop control:

- Number of loops:** A text input field containing the value '10'.
- Completed loops:** A label showing 'Completed loops: 0'.
- Start loop:** A blue button to begin the loop sequence.
- Stop loop:** A blue button to stop the loop sequence.

Figure 6.2.3.1 Web interface demo tab

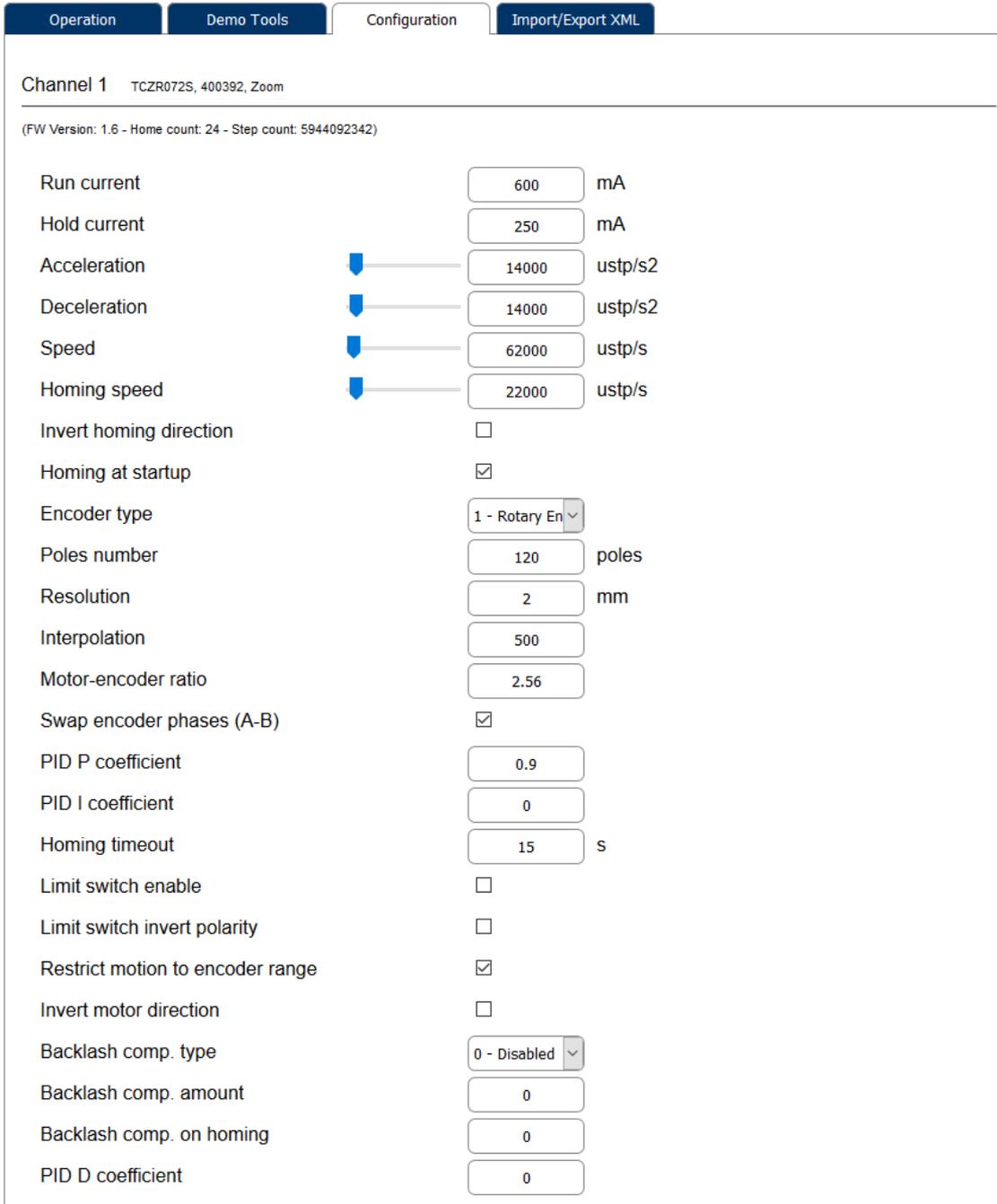
In order to start a **loop sequence** through different positions (previously saved in memory in the *Memory Save/Recall* section) perform the following operations:

1. Insert the memory slot of the desired position.
2. Insert the delay (in milliseconds) which will follow the positioning command.
3. Press the *Add position* button.
4. Repeat the steps 1, 2 and 3 for all the desired positions.
5. If needed, it's possible to select a position in the list on the right and press the *Delete selected* button in order to remove it from the loop sequence.

6. Insert the number of the desired loops.
7. Press the *Start loop* button.
8. If desired it's possible to early stop the loop sequence by pressing the *Stop loop* button

When the loop procedure is active, the current position is highlighted on the list. Another visual feedback is given by the *Completed loops* indicator, which increases every time that all the positions of the loop are scanned.

6.2.4 Configuration tab



Parameter	Value	Unit
Run current	600	mA
Hold current	250	mA
Acceleration	14000	ustp/s ²
Deceleration	14000	ustp/s ²
Speed	62000	ustp/s
Homing speed	22000	ustp/s
Invert homing direction	<input type="checkbox"/>	
Homing at startup	<input checked="" type="checkbox"/>	
Encoder type	1 - Rotary En	
Poles number	120	poles
Resolution	2	mm
Interpolation	500	
Motor-encoder ratio	2.56	
Swap encoder phases (A-B)	<input checked="" type="checkbox"/>	
PID P coefficient	0.9	
PID I coefficient	0	
Homing timeout	15	s
Limit switch enable	<input type="checkbox"/>	
Limit switch invert polarity	<input type="checkbox"/>	
Restrict motion to encoder range	<input checked="" type="checkbox"/>	
Invert motor direction	<input type="checkbox"/>	
Backlash comp. type	0 - Disabled	
Backlash comp. amount	0	
Backlash comp. on homing	0	
PID D coefficient	0	

Figure 6.2.4.1 Web interface configuration tab

The *Configuration* tab includes all the settings related to the motors and, eventually, encoders connected to each motion controller channel.

If the MTDV controller is used in combination with another **Opto Engineering product** (i.e. a motorized lens) there is **no need to configure each parameter**. It's enough to import an XML configuration file which contains all the correct parameters values for that product. See section 6.2.5 for more information.

If an XML configuration file is loaded, near the channel number is displayed the lens part number, serial number and the name of the controlled axis (i.e. "Zoom" or "Focus"). For each channel are also displayed the firmware version and a log of homing procedure count and motor step count.

See section 6.3 for further details about the configuration parameters.

6.2.5 Import/Export XML tab

In the *Import/Export XML* tab it's possible to import the MTDV settings from an XML file or to export the current MTDV settings to an XML file.

If the MTDV controller is used in combination with another Opto Engineering product (i.e. a motorized lens) there is no need to configure each parameter. It's enough to import an XML configuration file which contains all the correct parameters values for that product.

To load the configuration from an XML file, follow these steps:

1. Download the XML configuration file of your zoom lens on the web page https://www.opto-e.com/configuration_files by typing the serial number of the lens and the Opto Security Number and by clicking the submit button.
2. Click the button *Import/Export XML* (top-right of the web page) and click *Browse XML file* to select and load the XML file previously downloaded. If you use a multichannel device (i.e. MTDV2CH-22A2, MTDV3CH-22A3, MTDV4CH-22A4) you can choose which channel to associate to the zoom (see the following figures).
3. The MTDV device has 100 memory slots (from 0 to 99) to save motor positions. The XML file could contain the information of less positions (i.e. 10 positions). You can specify the memory offset to save this information. For example, if you choose "0" motor positions are saved from 0 to 9 memory slots. If you choose "21" motor positions are saved from 21 to 30 memory slots.
4. Click *Import XML* and your MTDV is configured
5. Please refresh the page to see the updated MTDV configuration.

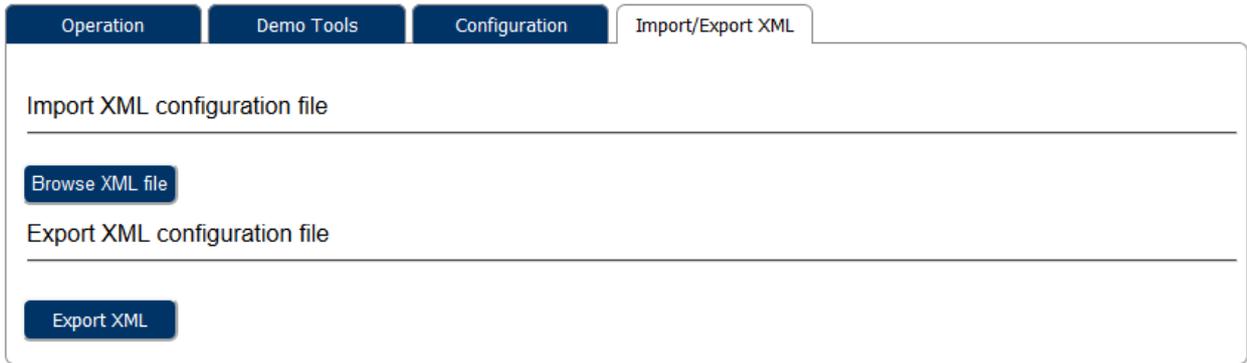


Figure 6.2.5.1 Web interface Import/Export XML tab

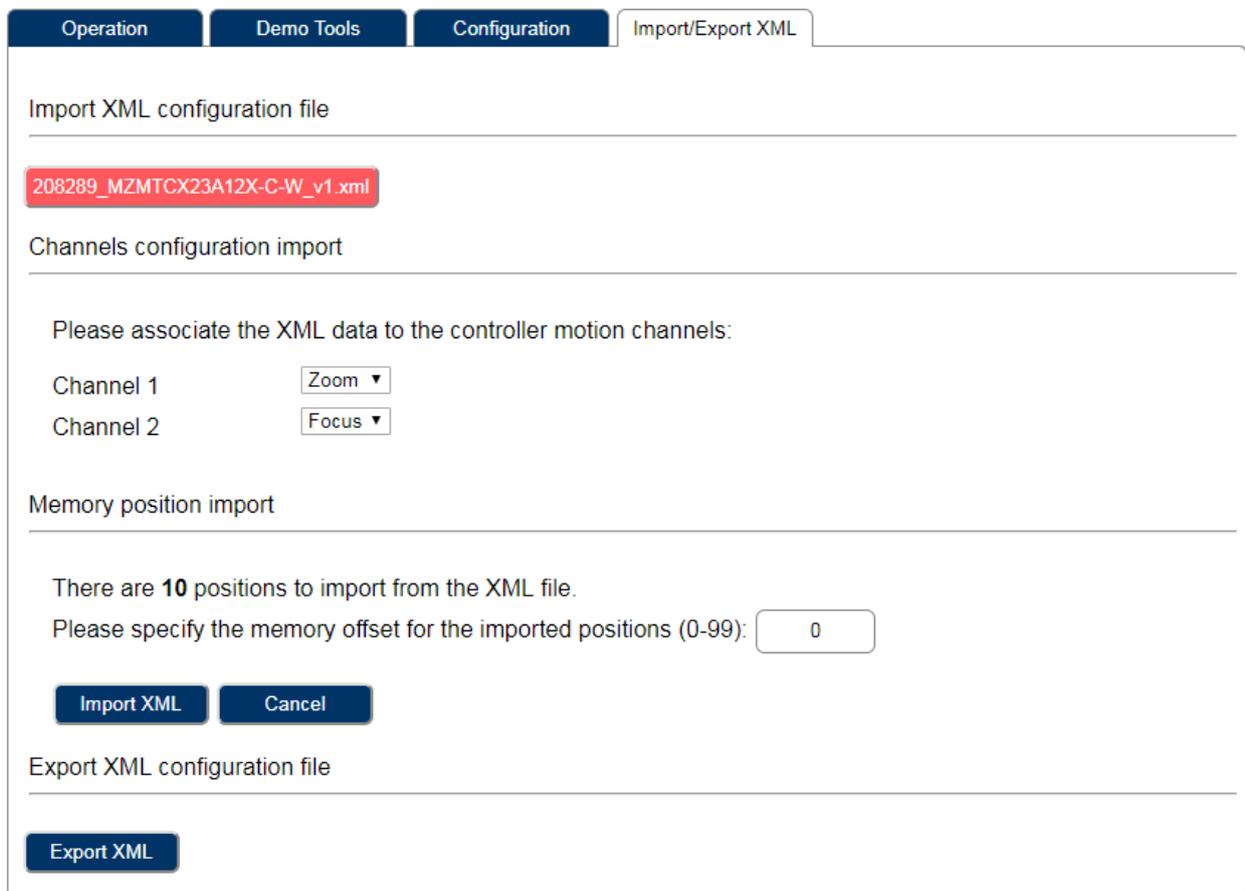


Figure 6.2.5.1 Web interface Import XML channel selection

6.3 Command list

The settings and actions of the MTDV controller are controlled by changing one of the available parameters. It's a register-like communication method, where each parameter (or register) describe an **attribute** (i.e. the motor speed) or an **action** (i.e. "start the homing procedure") of the motion controller.

When the parameter describe an **attribute**, the operation is straight-forward:

- **writing** the parameter value causes the attribute to be set immediately, affecting the behaviour of the controller;
- **reading** the parameter returns the actual value of the attribute which the controller is using.

When the parameter describe an **action** (highlighted in **bold** in the table), the operation is the following:

- **writing** the parameter value (usually "1" or boolean "TRUE") causes the action to be triggered (i.e. "start the homing procedure"). The parameter then returns to its default value (usually "0" or boolean "FALSE");
- **reading** the parameter has no effect and the value returned has no meaning.

The **INDEX** shown in the following table is the value to be used with all the DLL functions as the parameter identifier (or *param*).

In the section 6.4 there are more information on how to write and read the controller parameters.

INDEX	COMMAND	DATA TYPE	R/W	RANGE	UNIT	DESCRIPTION
0	Current position	Signed integer	R		μstep	Current position of the motor
1	Set absolute position	Signed integer	W		μstep	Perform an absolute positioning with reference to the "0" position (homing reference).
2	Set relative position	Signed integer	W		μstep	Perform an relative positioning with reference to the current position.
3	Motor homing	Unsigned integer	W	0 ÷ 1		Set to "1" to start a homing procedure. It's automatically reset to "0"
4	Motor stop	Unsigned integer	W	0 ÷ 1		Set to "1" to stop the motor. It's automatically reset to "0"
5	Run current	Unsigned integer	R/W	0 ÷ 1350	mA	Set the motor running RMS current value
6	Hold current	Unsigned integer	R/W	0 ÷ 1350	mA	Set the motor hold RMS current value (used to keep the motor still)
7	Acceleration	Unsigned integer	R/W	100 ÷ 500000	μstep/s ²	Set the motor acceleration value
8	Deceleration	Unsigned integer	R/W	100 ÷ 500000	μstep/s ²	Set the motor deceleration value
9	Speed	Unsigned integer	R/W	1000 ÷ 10000000	μstep/s	Set the motor speed

10	Homing speed	Unsigned integer	R/W	1000 ÷ 10000000	μstep/s	Set the motor homing speed
11	Invert homing direction	Boolean	R/W	0 ÷ 1	-	Select homing direction: when checked, the homing direction is inverted with respect to the default one
12	Homing at start-up	Boolean	R/W	0 ÷ 1	-	When checked, the motor position is set equal to the reference position every time the MTDV is switched on or reset
13	Encoder type	Unsigned integer	R/W	0 ÷ 2	-	Set the encoder type (if present). In particular: 0: no encoder is present 1: rotary encoder 2: linear encoder
14	Poles number	Unsigned integer	R/W	1 ÷ 100000	poles	Poles number of the encoder (i.e. magnetic poles)
15	Resolution	Float	R/W	0.1 ÷ 10.0	mm	Resolution of the encoder. Sometimes the resolution is referred also as "pole length"
16	Interpolation	Unsigned integer	R/W	1 ÷ 100000	-	Interpolation factor of the encoder ⁽¹⁾
17	Motor-encoder ratio	Float	R/W	0.001 ÷ 1000.0	-	Ratio between the motor resolution and the encoder resolution ⁽²⁾
18	Swap encoder phases (A-B)	Boolean	R/W	0 ÷ 1	-	When checked, swap the encoder A/B signals coming from the encoder
19	PID P coefficient	Float	R/W	0.0 ÷ 1000.0	-	Proportional coefficient of the PID control
20	PID I coefficient	Float	R/W	0.0 ÷ 1000.0	-	Integral coefficient of the PID control
21	Homing timeout	Unsigned integer	R/W	0 ÷ 50	s	Timeout for the homing operation
22	Limit switch enable	Boolean	R/W	0 ÷ 1	-	When checked, allow the use of a detection signal for the reference position (when no encoder is present)
23	Limit switch invert polarity	Boolean	R/W	0 ÷ 1	-	When checked, invert the polarity of the detection signal for the reference position (when no encoder is present)
24	Restrict motion to encoder range	Boolean	R/W	0 ÷ 1	-	When checked, only the encoder available values are accepted for motor position
25	Invert motor direction	Boolean	R/W	0 ÷ 1	-	When checked, invert the motor direction
26	Backlash comp. type	Unsigned integer	R/W	0 ÷ 3	μstep	Type of backlash compensation (only open loop). In particular: 0: no compensation 1: fixed compensation 2: exceed and go back CW 3: exceed and go back CCW
27	Backlash comp. amount	Signed integer	R/W	-50000 ÷ 50000	μstep	Amount of the backlash compensation

28	Backlash comp. on homing	Signed integer	R/W	-50000 ÷ 50000	μstep	Amount of the backlash compensation after positioning on the homing reference
29	PID D coefficient	Float	R/W	0.0 ÷ 1000.0	-	Differential coefficient of the PID control

- (1) The interpolation factor is defined as the ratio between the pole length and the resolution of the encoder; for example, a pole length equal to 2 mm and a resolution of 4 μm lead to an interpolation factor of 500.
- (2) Motor resolution and encoder resolution are obtained as:

$$\text{Motor resolution} = [n^\circ \text{ step per rev}] * 256 \text{ (microstepping)}$$

$$\text{Encoder resolution (ppr)} = \text{Poles number} \times \text{Interpolation} \quad (\text{rotary encoder})$$

$$\text{Resolution } (\mu\text{m}) = \frac{\text{Pole length } (\mu\text{m})}{\text{Interpolation}} \quad (\text{linear encoder})$$

6.4 DLL control

Opto Engineering offers a library which includes all the functions required to communicate with the MTDV motion controller.

The .dll file can be directly found and downloaded from the Opto Engineering website.

In the following sections are listed all the structures and the functions prototypes of the library.

6.4.1 Basic data structure

```
struct MTDVDev
{
    char ip[MAXSTRINGL];
    char description[MAXSTRINGL];
};
```

Structure which identifies an MTDV device.

ip	IP address of the device.
description [optional]	Description of the device.

```
struct MTDVData
{
    char name[MAXSTRINGL];
    unsigned int type;
    unsigned int rw;
    char unit[MAXSTRINGL];
    MTDVRawValue dataRaw;
    MTDVRawValue dataRawMin;
    MTDVRawValue dataRawMax;
};
```

Structure which represents a generic configuration parameter.

name	Name of the parameter.
type	Type of the parameter (see section <i>Parameters data types</i>).
rw	Read/write permissions flag (see section <i>Parameters RW flag</i>).
unit	Unit of measurement.
dataRaw	Raw parameter value.
dataRawMin	Raw parameter minimum value.
dataRawMax	Raw parameter maximum value.

Parameters data types

In the next table are listed the identifiers used to specify a parameter data type.

Value	Constant	Description
0	GO_TYPE_U32	Unsigned integer
1	GO_TYPE_I32	Signed integer
2	GO_TYPE_F	Float
4	GO_TYPE_BOOL	Boolean

Parameters RW flag

In the next table are listed the flags used to specify if a parameter is read-only or can also be written.

Value	Constant	Description
0	GO_RW_R	Read-only permission
1	GO_RW_RW	Read and write permission

Error flags

In the next table are listed all the errors which can be returned from the functions of section 6.3.2.

Value	Constant	Description
0	OE_MTDV_OK	No error.
1	OE_MTDV_HTTPREQ_ERR_WRONGPARAM	Wrong HTTP parameter.
2	OE_MTDV_HTTPREQ_ERR_CMDUNKNOWN	Unknown HTTP command.
3	OE_MTDV_HTTPREQ_ERR_DATARANGE	HTTP parameter data out of range.
4	HTTPREQ_ERR_DATAINTEGRITY	HTTP data integrity compromised.
5	HTTPREQ_ERR_READONLY	Attempt to write a read-only HTTP parameter.
6	HTTPREQ_ERR_DATATYPE	Wrong data type of the HTTP parameter.
5001	ERROR_WRITE	Write error on the TCP/IP socket.
5002	ERROR_READ	Read error on the TCP/IP socket.
5003	ERROR_PARSING	Parsing error of the answer parameters.
5004	ERROR_SOCKET	Error in the TCP/IP socket creation procedure.
5005	ERROR_INVALID_IP	The specified string is not a valid IP (ipv4).

6.4.2 Functions prototypes

```
int restoreFactorySettings(struct MTDVDev device);
```

Restore the factory settings of the motion controller. Any parameter and memory position is restored to its default value.

device	IN	Device identifier.
---------------	----	--------------------

```
int reboot(struct MTDVDev device);
```

Reboot the motion controller.

device	IN	Device identifier.
---------------	----	--------------------

```
int getDevInfo(struct MTDVDev device, char *pn, char *fwver,
               char *macAddress, char *ip, char *subNetMask, float &temperature,
               float &vdd1, float &vdd2);
```

Get information data of the motion controller.

Device	IN	Device identifier.
pn	OUT	Device part number.
fwver	OUT	Firmware version.
macAddress	OUT	Ethernet MAC Address.
ip	OUT	Ethernet IP Address.
subNetMask	OUT	Ethernet subnet mask.
temperature	OUT	Device temperature.
vdd1	OUT	Measured input voltage.
vdd2	OUT	Measured internal regulated logic voltage.

```
int getRawStatus (struct MTDVDev device, unsigned int &moving,
                  unsigned int &error);
```

Get the status of the motion controller.

device	IN	Device identifier.
moving	OUT	0 if all the motors are standstill, 1 if at least one motor is moving.
error	OUT	Bit-packed flag, where each bit that is set identifies a different error. See the following table for the possible values.

The following table lists the error flag returned by getRawStatus *error* parameter:

Value	Constant	Description
0x00000001	ERROR_FLAG_OVERVOLT	Input over voltage.
0x00000002	ERROR_FLAG_UNDERVOLT	Input under voltage.
0x00000004	ERROR_FLAG_OVERTEMP	Over temperature (> 90°C)
0x00000008	ERROR_FLAG_MOTOROUT1	Short-circuit on motor driver 1.
0x00000010	ERROR_FLAG_MOTOROUT2	Short-circuit on motor driver 2.
0x00000020	ERROR_FLAG_MOTOROUT3	Short-circuit on motor driver 3.
0x00000040	ERROR_FLAG_MOTOROUT4	Short-circuit on motor driver 4.
0x00000800	ERROR_FLAG_ENCERR1	Quadrature error on encoder 1.
0x00001000	ERROR_FLAG_ENCERR2	Quadrature error on encoder 2.
0x00002000	ERROR_FLAG_ENCERR3	Quadrature error on encoder 3.
0x00004000	ERROR_FLAG_ENCERR4	Quadrature error on encoder 4.

```
int getChannelsCount (struct MTDVDev device, unsigned int &channelsCount);
```

Get the number of the available motion channels.

device	IN	Device identifier.
channelsCount	OUT	Number of the available motion channels.

```
int getParametersCount (struct MTDVDev device, unsigned int channel,
    unsigned int &parametersCount);
```

Get the number of parameters available for the specified motion channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
parametersCount	OUT	Number of the available channel parameters.

```
int getChannelLog(struct MTDVDev device, unsigned int channel,
    char *fwver, unsigned int &homeCount, unsigned long int &stepCount);
```

Get log data of the specified motion channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
fwver	OUT	Firmware version.
homeCount	OUT	Number of homing procedures done.
stepCount	OUT	Number of μ steps done by the motor.

```
int getData(struct MTDVDev device, unsigned int channel,
    unsigned int param, MTDVData &data);
```

Get data of the specified parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
data	OUT	Raw data structure with the parameter data.

```
int getUIntData(struct MTDVDev device, unsigned int channel,
    unsigned int param, char *name, char *unit, bool &writable,
    unsigned int &data, unsigned int &dataMin, unsigned int &dataMax);
```

Get data of the specified *unsigned integer* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
name	OUT	Name of the parameter.
unit	OUT	Unit of measurement.
writable	OUT	Read/write permissions flag.
data	OUT	Unsigned integer parameter value.
dataMin	OUT	Unsigned integer parameter minimum value.
dataMax	OUT	Unsigned integer parameter maximum value.

```
int getIntData(struct MTDVDev device, unsigned int channel,
              unsigned int param, char *name, char *unit, bool &writable,
              int &data, int &dataMin, int &dataMax);
```

Get data of the specified *integer* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
name	OUT	Name of the parameter.
unit	OUT	Unit of measurement.
writable	OUT	Read/write permissions flag.
data	OUT	Integer parameter value.
dataMin	OUT	Integer parameter minimum value.
dataMax	OUT	Integer parameter maximum value.

```
int getFloatData(struct MTDVDev device, unsigned int channel,
                unsigned int param, char *name, char *unit, bool &writable,
                float &data, float &dataMin, float &dataMax);
```

Get data of the specified *float* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
name	OUT	Name of the parameter.
unit	OUT	Unit of measurement.
writable	OUT	Read/write permissions flag.
data	OUT	Float parameter value.
dataMin	OUT	Float parameter minimum value.
dataMax	OUT	Float parameter maximum value.

```
int getBoolData(struct MTDVDev device, unsigned int channel,
               unsigned int param, char *name, char *unit, bool &writable,
```

```
bool &data);
```

Get data of the specified *bool* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
name	OUT	Name of the parameter.
unit	OUT	Unit of measurement.
writable	OUT	Read/write permissions flag.
data	OUT	Bool parameter value.

```
int setUIntData(struct MTDVDev device, unsigned int channel,
               unsigned int param, unsigned int data, unsigned int &actualData);
```

Set a value for the specified *unsigned integer* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
data	IN	Unsigned integer parameter value.
actualData	OUT	Actual written parameter value. Can be different from the specified <i>data</i> value in order to meet the data range limitations (min and max).

```
int setIntData(struct MTDVDev device, unsigned int channel,
              unsigned int param, int data, int &actualData);
```

Set a value for the specified *integer* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
data	IN	Integer parameter value.
actualData	OUT	Actual written parameter value. Can be different from the specified <i>data</i> value in order to meet the data range limitations (min and max).

```
int setFloatData(struct MTDVDev device, unsigned int channel,
                unsigned int param, float data, float &actualData);
```

Set a value for the specified *float* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).

param	IN	Parameter number (starts from 0).
data	IN	Float parameter value.
actualData	OUT	Actual written parameter value. Can be different from the specified <i>data</i> value in order to meet the data range limitations (min and max).

```
int setBoolData(struct MTDVDev device, unsigned int channel,
               unsigned int param, bool data, bool &actualData);
```

Set a value for the specified *bool* parameter relative to the specified channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
param	IN	Parameter number (starts from 0).
data	IN	Bool parameter value.
actualData	OUT	Actual written parameter value. Can be different from the specified <i>data</i> value in order to meet the data range limitations (min and max).

```
int saveCurrentPositions(struct MTDVDev device, unsigned int memPos,
                       unsigned int channelsMask, char* name, unsigned int &actualMemPos);
```

Save the current motor positions according to the specified bit-packed mask. Bits [0-3] of *channelMask* which are set identifies the channels (from 0 to 3) of which the motor positions are saved.

device	IN	Device identifier.
memPos	IN	Device memory slot to write into.
channelMask	IN	Bit-packed mask of the channels motor to save.
name	IN	Name associate to the memory position (max 63 characters).
actualMemPos	OUT	Actual written memory slot. Can be different from the specified <i>memPos</i> value in order to meet the memory range limitations (0-99).

```
int loadPositionsAndMove(struct MTDVDev device, unsigned int memPos,
                       unsigned int &actualMemPos, char *name);
```

Load the motors positions from the specified memory slot. Each motion channel associated to a valid position immediately move its corresponding motor in order to reach it.

device	IN	Device identifier.
memPos	IN	Device memory slot to read from.
actualMemPos	OUT	Actual read memory slot. Can be different from the specified <i>memPos</i> value in order to meet the memory range limitations (0-99).

name OUT Name associate to the memory position (max 63 characters).

```
int saveSpecificPositions(struct MTDVDev device, unsigned int memPos,
    char *name, int p1, int p2, int p3, int p4, unsigned int &actualMemPos);
```

Save the specified motor positions in a motion controller memory slot. Set to 0x7FFFFFFF (invalid position) the motor positions that you don't want to save (and so, not recall with the corresponding load function).

device	IN	Device identifier.
memPos	IN	Device memory slot to write into.
name	IN	Name associate to the memory position (max 63 characters).
p1	IN	Position of the motor of channel 1.
p2	IN	Position of the motor of channel 2.
p3	IN	Position of the motor of channel 3.
p4	IN	Position of the motor of channel 4.
actualMemPos	OUT	Actual written memory slot. Can be different from the specified <i>memPos</i> value in order to meet the memory range limitations (0-99).

```
int loadPositions(struct MTDVDev device, unsigned int memPos, char*name,
    unsigned int &actualMemPos, int &p1, int &p2, int &p3, int &p4);
```

Read the motors positions from the specified memory slot. Note that the motion channels don't perform any action. If one of the position wasn't save or set to an invalid value, it is read as 0x7FFFFFFF (invalid position).

device	IN	Device identifier.
memPos	IN	Device memory slot to write into.
name	OUT	Name associate to the memory position (max 63 characters).
actualMemPos	OUT	Actual read memory slot. Can be different from the specified <i>memPos</i> value in order to meet the memory range limitations (0-99).
p1	OUT	Position of the motor of channel 1.
p2	OUT	Position of the motor of channel 2.
p3	OUT	Position of the motor of channel 3.
p4	OUT	Position of the motor of channel 4.

```
int setUIntDataLimit(struct MTDVDev device, unsigned int channel,
    unsigned int param, unsigned int dataMin, unsigned int dataMax,
    unsigned int &actualDataMin, unsigned int &actualDataMax);
```

Set new lower and upper limits for the specified *unsigned integer* parameter. This will overwrite the factory defaults "safe" limits.

<code>device</code>	IN	Device identifier.
<code>channel</code>	IN	Motion channel number (starts from 0).
<code>param</code>	IN	Parameter number (starts from 0).
<code>dataMin</code>	IN	Unsigned integer lower limit.
<code>dataMax</code>	IN	Unsigned integer upper limit.
<code>actualDataMin</code>	OUT	Actual lower limit set.
<code>actualDataMax</code>	OUT	Actual upper limit set.

```
int setIntDataLimit(struct MTDVDev device, unsigned int channel,
    unsigned int param, int dataMin, int dataMax,
    int &actualDataMin, int &actualDataMax);
```

Set new lower and upper limits for the specified *integer* parameter. This will overwrite the factory defaults "safe" limits.

<code>device</code>	IN	Device identifier.
<code>channel</code>	IN	Motion channel number (starts from 0).
<code>param</code>	IN	Parameter number (starts from 0).
<code>dataMin</code>	IN	Integer lower limit.
<code>dataMax</code>	IN	Integer upper limit.
<code>actualDataMin</code>	OUT	Actual lower limit set.
<code>actualDataMax</code>	OUT	Actual upper limit set.

```
int setFloatDataLimit(struct MTDVDev device, unsigned int channel,
    unsigned int param, float dataMin, float dataMax,
    float &actualDataMin, float &actualDataMax);
```

Set new lower and upper limits for the specified *float* parameter. This will overwrite the factory defaults "safe" limits.

<code>device</code>	IN	Device identifier.
<code>channel</code>	IN	Motion channel number (starts from 0).
<code>param</code>	IN	Parameter number (starts from 0).
<code>dataMin</code>	IN	Float lower limit.
<code>dataMax</code>	IN	Float upper limit.
<code>actualDataMin</code>	OUT	Actual lower limit set.
<code>actualDataMax</code>	OUT	Actual upper limit set.

```
int setChannelName(struct MTDVDev device, unsigned int channel,
    char *name, unsigned int &actualChannel, char* actualName);
```

Set a name for the specified motion channel.

<code>device</code>	IN	Device identifier.
<code>channel</code>	IN	Motion channel number (starts from 0).

name	IN	Name to be associate with the specified motion channel (max 63 characters).
actualChannel	OUT	Actual selected channel.
actualName	OUT	Actual name associated with the motion channel.

```
int getChannelName(struct MTDVDev device, unsigned int channel,
    unsigned int &actualChannel, char* actualName);
```

Get the name of the specified motion channel.

device	IN	Device identifier.
channel	IN	Motion channel number (starts from 0).
actualChannel	OUT	Actual selected channel.
actualName	OUT	Actual name associated with the motion channel.

6.5 Firmware update

Opto Engineering offers the possibility to update the device firmware with an easy procedure. This operation can be done directly by the user: all the necessary tools and software can be found on Opto Engineering website.

NOTE: *to proceed with the firmware update, it's necessary to check if the firewall of the PC is blocking the network connections (both in input and in output).
If the firewall is limiting the network connections, it's necessary to open ports 9, 67, 68, 69 in both directions (IN and OUT) for the UDP protocol.*

To update the firmware, take the following steps:

1. Connect the MTDV to the power supply and to the PC through an Ethernet cable.
2. Detach any peripheral connected to the motion channels DB15HD connectors (i.e. motors and encoders).
3. Extract ALL the content of the *MTDV Firmware Update Utility* ZIP file in a single folder (firmware binaries and update application).
4. Open the *MTDV Firmware Update Utility* application.
5. Insert the device MAC address (can be found on the product label), the local IP address and the device IP address.
6. Click on *Connect Device* button.
7. Click on *Program* button. The process can take a few minutes to complete. Please note that the software tries to update all the 4 motion channels. If your MTDV device has less than 4 channels, a warning is displayed. Ignore it and proceed.
8. Wait until you see the update confirmation window.
9. Close the application.
10. Open the browser and go to the web page which corresponds to the MTDV IP address (default: <http://192.168.1.200>).
11. Click on the *Reboot Device* button.
12. Wait for the device to reboot.
13. Click on the *Reset Factory Settings* button.
14. The update procedure is now completed.

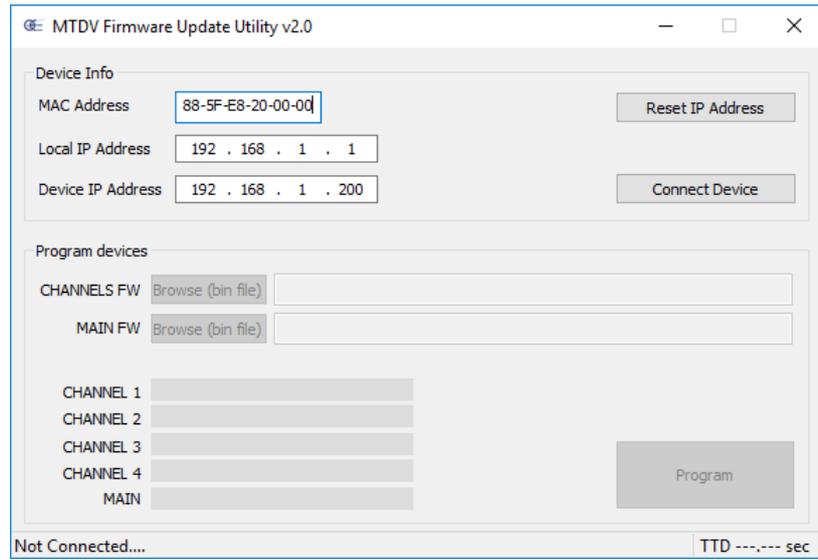


Figure 6.4.1 Update tool GUI

7 • SERIAL COMMUNICATION

7.1 USB communication

To be implemented on request.

7.2 RS485 communication

To be implemented on request.

8 • TROUBLESHOOTING

In this section are listed the solutions to possible problems or apparent causes of malfunctioning.

The MTDV doesn't turn on.

- Check the wiring configuration.
- Check that the power supply is properly connected and fully functional.

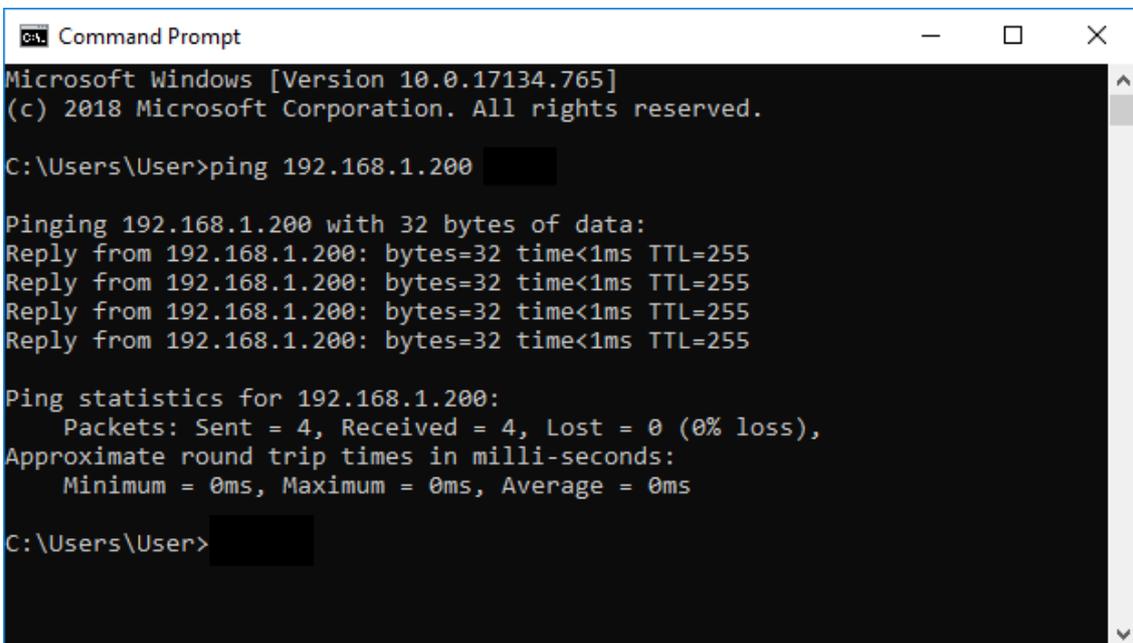
How to check if the MTDV is working properly:

- Connect an ethernet cable between the MTDV and a PC. The LAN port green led is active.
- It's possible to access the web application (see section 6.2 for further details).

The MTDV is turned on but it's impossible to access the web page application.

If the web page cannot be accessed, try the following steps (after every step try to access the web page again):

- Check the system "Network and internet connections" properties.
If the lens has the default IP address (192.168.1.200) one possible configuration is the following:
IP address: 192.168.1.10
Subnet mask: 255.255.255.0
- Try to *ping* the MTDV by the command prompt; if the communication is established the following screen appears (this function is available from FW version 2.0):



```
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\User>ping 192.168.1.200

Pinging 192.168.1.200 with 32 bytes of data:
Reply from 192.168.1.200: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\User>
```

Figure 8.1 Successful ping operation

- Check if there is an IP address conflict with other devices.
Please, also be careful about the configuration of the Company network.
- Try to use another browser.

- Try to use another Ethernet cable.

How to check if the procedure is working:

- It's possible to access the web application (see section 6.2 for further details).

An error occurs during the firmware update procedure.

If the FW update procedure doesn't work, try the following steps:

- Check that the MTDV is powered.
- Check the firewall settings: ports 9, 67, 68 and 69 have to be opened for TCP/UDP protocol, both in input and in output.
- Check that all the fields of the MTDV Firmware Update Utility are correct:
 - The MAC Address of the device can be found on the product label (or on the web page of the demo application).
 - The Local IP Address must be equal to the one set in the network properties of the PC.
 - The Device IP Address is the default one (192.168.1.200) or the one chosen by the user.
 - The .bin file has to be the correct one

How to check if the procedure is working:

- Check the FW version on the web page interface.

9 • COMPATIBLE PRODUCTS AND ACCESSORIES

9.1 Compatible OE products

The MTDV motion controller is fully compatible with the following OE products:

- TCZR S series (using CBMT002)
- MZMT12X series (using CBMT003)



CAUTION: Always load the proper XML configuration file **BEFORE** connecting any Opto Engineering lens to the MTDV controller. Failing to do so can lead to damage due to incorrect motor drive.

9.2 Accessories

The following cables are compatible with the MTDV motion controller in order to use it with other OE products:

- CBMT002 - 15 wires cable, DB15HD Male to DB15HD Female connector, 2 m.
- CBMT003 - 8 wires cable, 2x DB15HD Male to DIN EN 60529 12 Pin Female connector, 2 m.
- CBETH003 - Ethernet cable, CAT6, industrial level, high flexible cable with screw, 5 m.

Cables must be ordered separately.