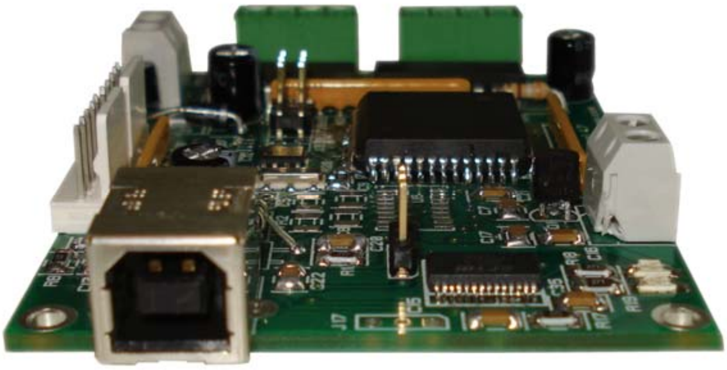
AXES CONTROL UNIT MT2USBMS
USER MANUAL

Rel. 01.00.0002
(Product Code: MT2USBMS)



Information provided in this manual is property of IPSES S.r.l. and must be considered and treated as confidential.

This publication can only be reproduced, transmitted, transcribed or translated into any human or computer language with the written consent of IPSES S.r.l.

Information in this documentation has been carefully checked and is believed to be accurate as of the date of publication; however, no responsibility is assumed of inaccuracies. IPSES will not be liable for any consequential or incidental damages arising from reliance on the accuracy of this documentation.

Information contained in this manual is subject to change without notice and does not represent a commitment on the part of IPSES. The design of this instrument is subject to continue development and improvement. Consequently, the equipment associated to this document may incorporate minor changes in detail from the information hereafter provided.

All brand or product names are trademarks or registered trademarks of their respective holders.

This manual in English is the original version.

Printed in Italy

GUARANTEE

IPSES warrants to the end-user in accordance with the following provisions that its branded hardware products, purchased by the end-user from IPSES company or an authorized IPSES distributor will be free from defects in materials, workmanship and design affecting normal use, for a period of one year as of the original purchase date. Products for which proper claims are made will, at IPSES's option, be repaired or replaced at IPSES's expense[1].

Exclusions

This Guarantee does not apply to defects resulting from: improper or inadequate installation, use or maintenance; actions or modifications by unauthorized third parties or the end-user; accidental or wilful damage or normal wear and tear.

Making a claim

Claims must be made by contacting IPSES office within the guarantee period. Please, contact:

IPSES S.r.l. - Via Lazzarotto, 10 - 20020 Cesate (MI) Italy
Tel. (+39) 02/39449519 - Fax (+39) 02/700403170
http://www.ipses.com - e-mail support@ipses.com

Limitation and Statutory Rights

IPSES makes no other warranty, guarantee or like statement other than as explicitly stated above and this Guarantee is given in place of all other guarantees whatsoever, to the fullest extent permitted by law. In the absence of applicable legislation, this Guarantee will be the end-user's sole and exclusive remedy against IPSES.

General Provisions

IPSES makes no express warranties or conditions beyond those stated in this warranty statement. IPSES disclaims all other warranties and conditions, express or implied, including without limitation implied warranties and conditions of merchantability and fitness for a particular purpose.
IPSES's responsibility for malfunctions and defects in hardware is limited to repair and replacement as set forth in this warranty statement.
IPSES does not accept liability beyond the remedies set forth in this warranty statement or liability for incidental or consequential damages, including without limitation any liability for products not being available for use or for lost data or software.

[1] With the exclusion of shipping costs for and from IPSES's development office.

WARNING!

ELECTRICAL DEVICES COULD DAMAGE EQUIPMENT OR PROPERTY OR CAUSE PERSONAL INJURY

This guide contains instructions and technical features of the MT2USBMS AXES CONTROL UNITS.
Read with attention before attempting to install.

It is the responsibility of the technician to undertake all the safety rules provided by the law during the installation and the use of this device.
For any information which is not contained in this guide, please contact:

IPSES S.r.l. - Via Lazzarotto, 10 - 20020 Cesate (MI) Italy
Tel. (+39) 02/39449519 - Fax (+39) 02/700403170
http://www.ipses.com - e-mail support@ipses.com

TABLE OF CONTENTS

REVISION HISTORY

Manual revision history

Revisione/ Data	Descrizione modifica	Autore
01.00.0000 January, 2014	First version Released	Mancuso C.
01.00.0001 February, 2016	Minor changes	Bottaccioli M.
01.00.0002 August, 2016	Added ISO 9001:20015 logo	Bottaccioli M.

GENERAL FEATURES

MT2USBMS is small size low power control devices which can control both **two bipolar and two unipolar stepper motors** (i.e. 8 and 4 lead motors, and 6 lead centred tapped motors) and their respective limit switches (one for each axis, with programmable polarity). The device accepts optical switches and in order to preserve sensors lifetime, they are powered only when the motor is moving or during their status request. Motor **control** and the device **configuration** are achieved through a **USB interface**, easily managed by the provided drivers.

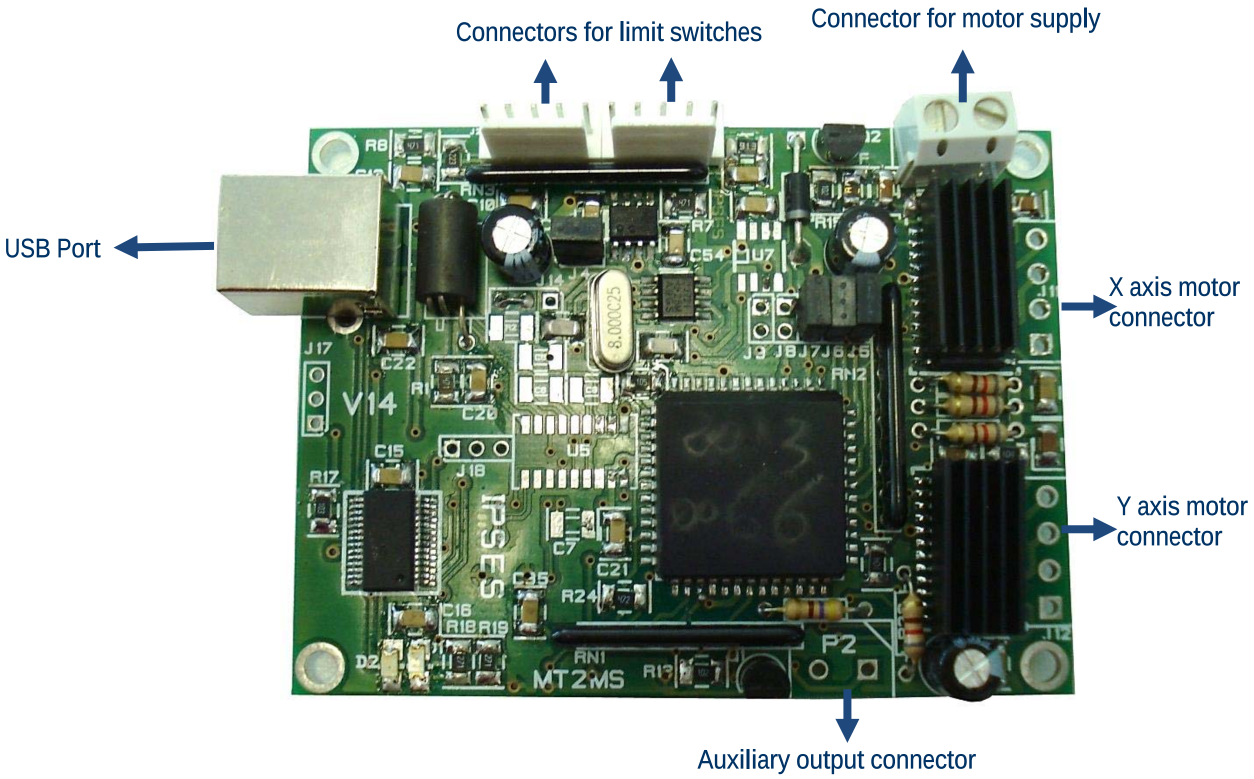
The **motor rotation speed** can be easily and **independently** configured to answer user needs, the number of **half-steps or micro steps per second** can be set as needed.

Moreover, the device has an **auxiliary output** with a dedicated control for its activation.

The MT2USBMS version is equipped with a PWM current control system on the motor phases. This device can reach a movement precision of 1/8 of step.

In

Picture 1 MT2USB card is shown with position of connectors and USB port. Connectors and USB port are in analogue position also in the microstepper card.


Connectors for limit switches
Connector for motor supply
USB Port
X axis motor connector
Y axis motor connector
Auxiliary output connector

Picture 1: MT2USB card – Position of connectors and USB port

DRIVER INSTALLATION

We recommend to execute the automatic software installation from CD before connecting the device to PC. By this way, software and USB driver are both installed, allowing the PC to automatically identify the device once you connect it.
If you use the recommend automatic software installation from CD, I do not need to follow indications contained in this chapter.

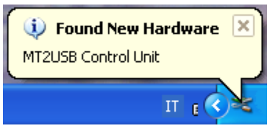
If you do not use the recommend automatic software installation from CD, to connect MT2USBMS to your PC you need to install only the USB IPSES driver that is certified for the most recent Microsoft operating systems:

- Microsoft Windows 2000 family
- Microsoft Windows XP family, x86
- Microsoft Windows Server 2003 family, x86
- Microsoft Windows Server 2003 family, x64
- Microsoft Windows XP family, x64
- Microsoft Windows Vista family, x86
- Microsoft Windows Vista family, x64
- Microsoft Windows Server 2008 family, x86
- Microsoft Windows Server 2008 family, x64
- Microsoft Windows 7
- Microsoft Windows 7 x64
- Microsoft Windows Server 2008 Release 2 family, x64
- Microsoft Windows 8 and 8.1
- Microsoft Windows 8 and 8.1 x64
- Microsoft Windows 10
- Microsoft Windows 10 x64

If your PC has an internet connection, you should follow the automatic Windows Update procedure, otherwise follow the manual installation procedure from CD.

Automatic Windows Update procedure

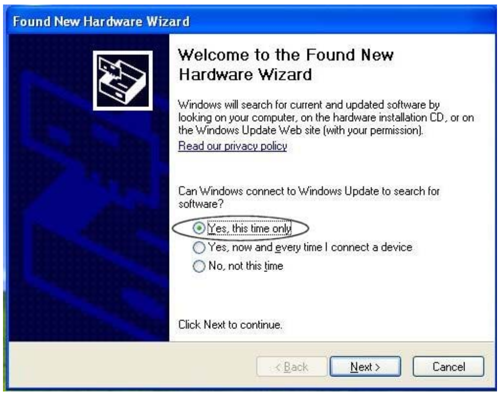
1) Connect the MT2USBMS board to PC using a USB cable. *Windows* operating system will detect a new device, showing a message similar to:



Found New Hardware

MT2USB Control Unit

2) In the following windows “found new hardware wizard” chose “Yes, this time only” and then “Next”. Wait for a complete download of the driver and its installation.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

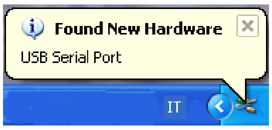
Can Windows connect to Windows Update to search for software?

● Yes, this time only

○ Yes, now and every time I connect a device

○ No, not this time

Click Next to continue.

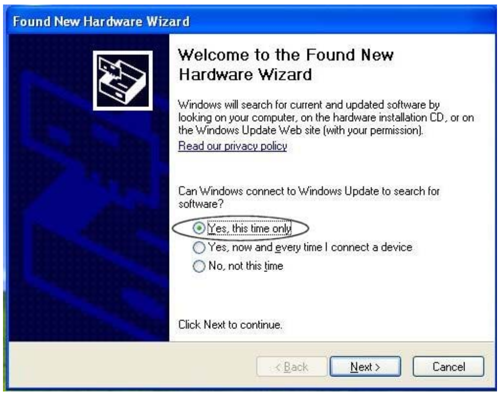


Found New Hardware

USB Serial Port

3) After a window with the message “Found New Hardware. USB Serial Port” is displayed.

4) In the following windows “found new hardware wizard” chose “Yes, this time only” and then “Next”. Wait for a complete download of the driver and its installation.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

● Yes, this time only

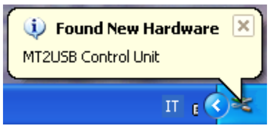
○ Yes, now and every time I connect a device

○ No, not this time

Click Next to continue.

Manual driver installation procedure

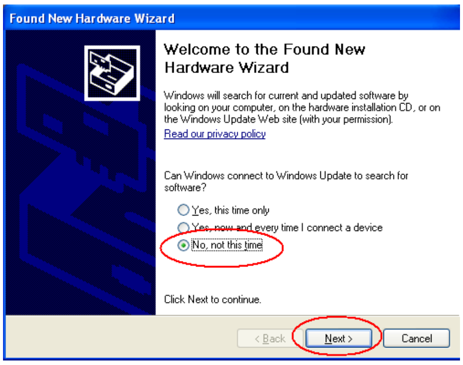
1) Connect the MT2USBMS board to the PC using a USB cable. *Windows* operating system will detect a new device, showing the message:



Found New Hardware

MT2USB Control Unit

2) In the following windows “found new hardware wizard” chose “No, not this time” and then “Next”.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

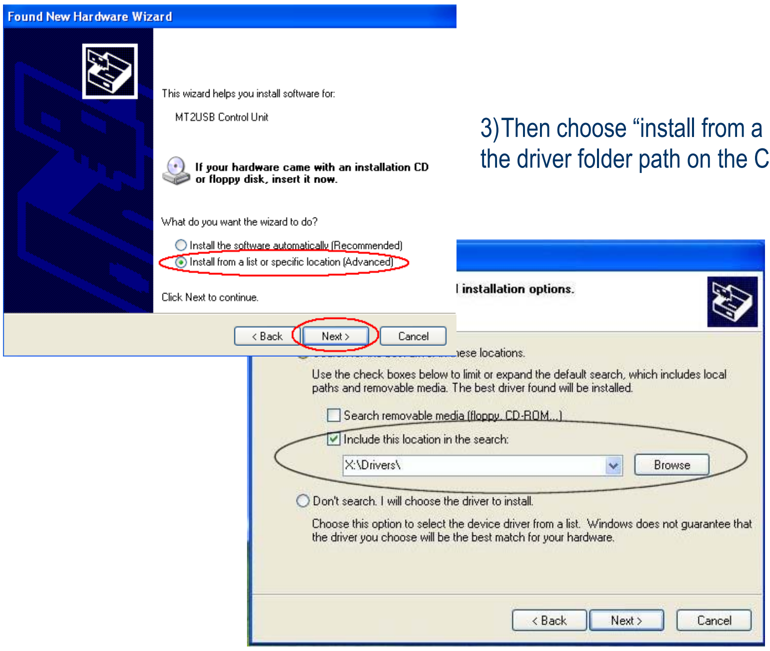
○ Yes, this time only

○ Yes, now and every time I connect a device

● No, not this time

Click Next to continue.

3)Then choose “install from a list or specific location (Advanced)” and “Next”. Then Set the driver folder path on the CD.



Found New Hardware Wizard

This wizard helps you install software for:

MT2USB Control Unit

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)

● Install from a list or specific location (Advanced)

Click Next to continue.

l installation options.

...ese locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

X:\Drivers\

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

REMOTE CONTROL COMMUNICATION PROTOCOL

The communication of control unit is achieved through a USB interface, made up by the driver, which is provided with the unit.
The commands are strings in ASCII code, terminated with <CR>. Others control characters (<*LF*>, <*VT*>, etc.) are ignored. The protocol is not case sensitive.
In case of virtual com port protocol, commands may be directly conveyed with any serial client (*Windows* HyperTerminal, for example). Moreover, application programs may be developed using standard communication functions provided. In this case, this are the settling:

➢ Communication rate:	9600 baud
➢ data bits:	8
➢ parity bit:	none
➢ stop bits:	1
➢ stream control:	none

In case of DLL choice, please refer to “FTD2XX.DLL DYNAMIC LIBRARY” manual by IPSES.

The following commands are implemented:

U	Requests the current global status of the unit (See further how the status is coded).
Pa,b	Moves the axes to an a and a b position on a coordinate grid: a and b are the absolute positions in half-steps or micro-steps. Both a and b values must be included between between –644.999 and +639.999.
Xa	Moves the X axis to an a position (absolute position in half-steps or micro-steps). The a parameter must be included between –644.999 and +639..
Yb	Moves the Y axis to a b position (absolute position in half-steps or micro-steps). The b parameter must be included between–644.999 and +639.999.
Da,b	Moves the axes by a, b movement (relative movements), where a and b are the movement values in micro steps (both values must be included between -1.289.999 and +1.279.999). It is possible to ignore the “b” parameter to move only the first axis.
L1	Activates the auxiliary output.
L0	Deactivates the auxiliary output.
B0	Deactivates the braking action when the motor is not running.
B1	Activates the braking action, with PWM current control, when the motor is not running. Be careful: to avoid motor and/or card overheating use this command with caution.
H	Moves both axes to the *home position* (*limit detection*): during the execution of this command, no more commands are received by the device.
HX	Moves the X axis to the *home position* (*limit detection*): during the execution of this command, no more commands are received by the device.
HY	Moves the Y axis to the *home position* (*limit detection*): during the execution of this command, no more commands are received by the device.
K	Stops immediately the movement of both axes, except during the home position command.
KX	Stops the movement of the X axis (except during the home position).
KY	Stops the movement of the Y axis (except during the home position).
GX,n	Perpetual motion of the X axis; when n>0 or it is omitted, this command allows forward movement, when n<0 it allows backward movement.
GY,n	Perpetual motion of the Y axis; when n>0 or it is omitted, this command allows forward movement, when n<0 it allows backward movement.
CX,n	Stepping mode of the X axis motor movement: • n = 0: whole step. • n = 1: half step. • n = 2: 1/4 of step.

	• n = 3: 1/8 of step.
CY,n	Stepping mode of the Y axis motor movement: • n = 0: whole step. • n = 1: half step. • n = 2: 1/4 of step. • n = 3: 1/8 of step.
CX?	Requests the stepping mode of X axis motor movement.
CY?	Requests the supply mode of the stepping mode of Y axis motor movement.
FX,n	Sets the current position on X axis, only when the motor is stopped (otherwise it generate an illegal command error). The n parameter must be included between –644.999 and +639.999.
FY,n	Sets the current position on Y axis, only when the motor is stopped (otherwise it generate an illegal command error). The n parameter must be included between –644.999 and +639.999.
SX,n	Sets the speed for the X axis to n micro-steps/s. This value must be included between 18 and 500. This command can be executed only when both axes are stopped and the parameter is referred to the X axis current resolution sets by C commands.
SY,n	Sets the speed for the Y axis to n micro-steps/s. This value must be included between 18 and 500. This command can be executed only when both axes are stopped and the parameter is referred to the Y axis current resolution sets by C commands.
SX?	Requests the X axis current speed.
SY?	Requests the Y axis current speed.
W	Requests the current position. The answer is an (x,y) couple, where x and y are the absolute coordinates in whole steps, half-steps, 1/4 of steps or 1/8 of steps (depending on the configured resolution with Cx,n and Cy,n commands). If the position is unknown, the answer is # character.
E1,n	Sets the maximum number of micro steps, for both axes, for the first motors movement during the home commands. The n parameter must be included between 0 and +639.999, and it is referred to the current axes resolution (see Cx,n and Cy,n commands).
E2,n	Sets the maximum number of micro steps, for both axes, for the forward motors movement after the home position has been reached first, during the home commands. The n parameter must be included between 0 and +9.999.
E3,n	Sets the maximum number of micro steps, for both axes, for the backward motors movement subsequent the forward displacement, during the home commands. This motion allows to try again the home position to prevent false contact. The n parameter must be included between 0 and +9.999.
E1?	Request the memorized number of free micro steps for the home position search.
E2?	Request the memorized number of forward micro steps during the home position search.
E3?	Request the memorized number of backward micro steps during the home position search.
?	Requests the current firmware version and the serial number of the instrument. The answer will be an ASCII string similar to "*MT2USBMS relx.x – S/Nyyyyyy*", in which *x.x* represents the firmware version of the device and *yyyyyy* is the serial number.
M	Stores the speed settings and the working mode currently set in the non-volatile memory.

It is also possible, for commands using X and Y characters, replacing X with 1 and Y with 2: for instance, the command *GX,1* is equivalent to G*1,1*.

For every request command sent (such as u, ?, Cx?, etc.) the device will return the property information string followed by <CR> and <LF> control characters (13 and 10 in decimal notation, respectively).

During the execution of a home command (refer to H, Hx and Hy commands) the motor moves in search of the home position ("Limit" signal from switches) for a maximum number of steps sets by E1,n command. The current steps resolution depends by the card version (refer to Cx,n and Cy,n commands).
If, within this phase, the limit position has not been reached an home position error will be asserted by the system. Otherwise, when the home position has been reached, the motor moves in a forward direction for the number of steps sets by E2,n command. Then it come back for the number of steps sets by E3,n command. If the limit position will be reached for the second time the motor stops.

It is important that during the second phase (forward motion), the motor moves for the number of steps enough to disable the home position signal, otherwise an error is generated. It means that the parameter of E2,n command must be greater than zero at least to a sufficient quantity, depending by the mechanic of your application.
All this procedure are intended to avoid false triggers of the home signal and to achieve better home positioning: during the execution of a home command, no more commands (status request included) are received by the device.
It is recommended to sets the E3 parameter value command equal or greater than the E2, to allow the second home position achieving. Otherwise the system will be generate an error.

All the positions and the movements are in whole, half, 1/4 and 1/8 of steps for the MT2USBMS version (depending on the configured resolution with Cx,n and Cy,n commands).

The speed request messages return the current speed for the selected axis as a number representing the speed expressed in whole steps/s, half-steps/s, 1/4 of steps/s or 1/8 of steps/s (depending on the configured resolution with Cx,n and Cy,n commands) in case of MT2USBMS card.

The status request message (“U”) forces the device to return a byte (2 hex characters) representing the actual status of the unit.

Return message interpretation table:

bit 7	error
bit 6	known Y axis position
bit 5	known X axis position
bit 4	auxiliary output active
bit 3	Y axis home position reached
bit 2	X axis home position reached
bit 1	Axes running
bit 0	*braking action* (when the motor is stopped) active.

Example: if the status command is sent during an axes movement (with start positions known), the read string will be “62” in hexadecimal code or “01100010” in binary code. Instead, if the X axis start position is unknown and the auxiliary output is active the read string will be “52” in hexadecimal code or “01010010” in binary code.

If the error bit is high (i. e. if it answers with a code like 81), then another error code is added after a comma (for example 81,02); more than one error code can be active.

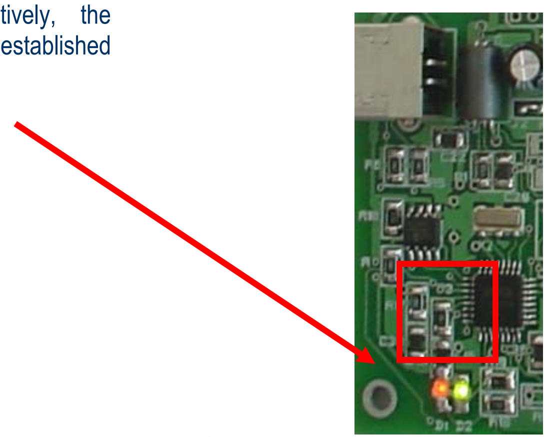
The possible error codes are:

bit 7	Reached the Y axis home position during backward movement when negative run is disabled with the appropriate *jumper*.
bit 6	Reached the X axis home position during backward movement when negative run is disabled with the appropriate *jumper*.
bit 5	Invalid *checksum* of the stored data in non-volatile memory.
bit 4	Invalid number stored in non-volatile memory.
bit 3	Time out or error during home position search.
bit 2	Out of range parameter (i. e. the set speed is out of the fixed ranges).
bit 1	Illegal command (i.e. an absolute movement request when the positions are unknown or during a movement).
bit 0	Command not acknowledged.

All the errors are reset after the state request command.

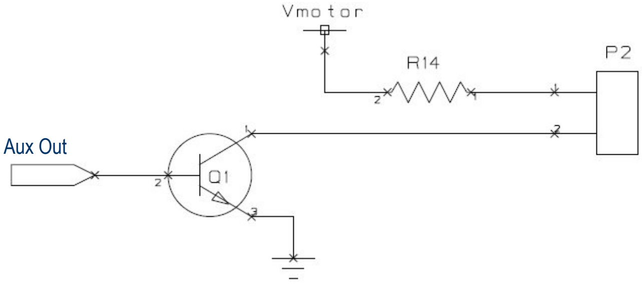
It is suggested to link the device with the USB cable only after power supply has been applied.

Two LEDS, one red and the other green, indicate, respectively, the exchanging of data between the unit and the PC an the established connection of the unit with the PC.



Picture 2: exchanging data and established connection LEDs.

Board is provided of an *open-collector* output with an internal *pull-up* managed by user (as it can be seen in the previous table), which output schematic circuit is reported in the next Picture 3:

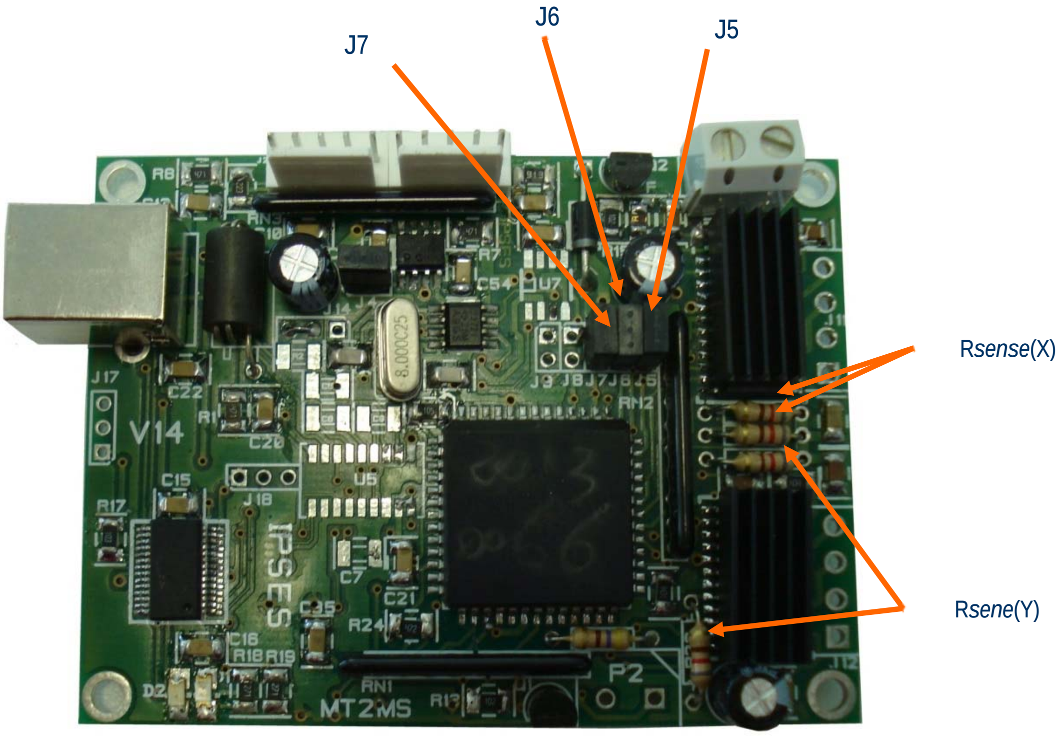


Picture 3: auxiliary output schematic

The value of R14 is 470Ω (1/4 W).

MT2USBMS CONFIGURATION INSTRUCTIONS

Through the **jumpers** placed on MT2USBMS card, as shown in Picture , it is possible to **program** the same function modes described in the previous page for the MT2USB version.



Picture 4: jumpers and sense resitors in MT2USBMS card

Through the four **sense resistors** it is possible to set the nominal current of the connected motors.

It is suggested to use resistors with tolerance less than 2%. The correct sense resistor value can be found using the following formula or the table on the next page:

$$R_{sense} = {}^{0{,}5}\!/_{I_{nom}}$$

Here below there is a resistor value table in which are reported powers and minimum/maximum current values:

Resistor	Minimum current	Maximum current
0,68 Ω (½ W)	700 mA	750 mA
0,75 Ω (½ W)	640 mA	700 mA
0,82 Ω (½ W)	580 mA	640 mA
0,91 Ω (½ W)	525 mA	580 mA
1,0 Ω (½ W)	460 mA	525 mA

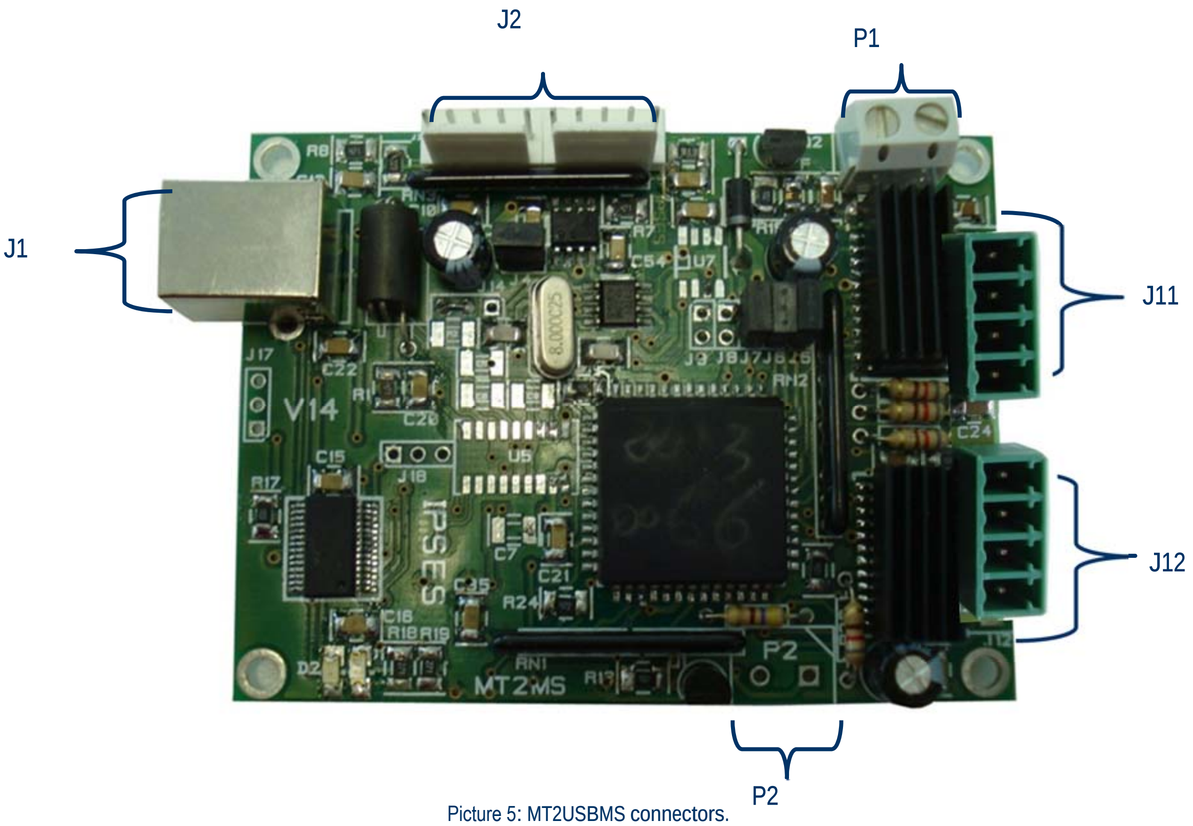
1,2 Ω (¼ W)	375 mA	460 mA
1,5 Ω (¼ W)	305 mA	375 mA
1,8 Ω (¼ W)	250 mA	305 mA
2,2 Ω (¼ W)	205 mA	250 mA
2,7 Ω (¼ W)	170 mA	205 mA
3,3 Ω (¼ W)	140 mA	170 mA
3,9 Ω (¼ W)	120 mA	140 mA
4,7 Ω (¼ W)	100 mA	120 mA
5,6 Ω (¼ W)	80 mA	100 mA
6,8 Ω (¼ W)	70 mA	80 mA
8,2 Ω (¼ W)	55 mA	70 mA
10,0 Ω (¼ W)	45 mA	55 mA

The board is equipped with four 1,2Ω (¼ W) resistor, different resistors are installed on request.

MT2USBMS CONNECTION INSTRUCTIONS

The device needs a supply for control logic and for motors driving (see P1 connector in Picture). For MT2USBMS version, power supply must be included between 4,5 and 30V.

Moreover, it is possible, to connect two limit/home detectors to the card, one for each motor (see J2 connectors).
It is also possible to use an *open collecto*r auxiliary output (see P2 connector) to communicate with a PLC or to activate externally a control electronic system.

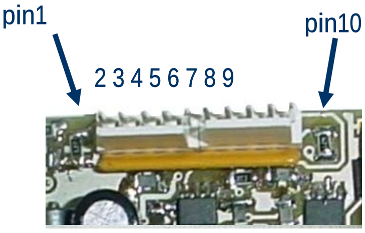


Picture 5: MT2USBMS connectors.

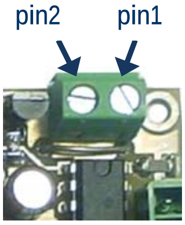
CONNECTION

J1 (USB): “B” type connector, to interface with a PC.

J2 (limit detector):

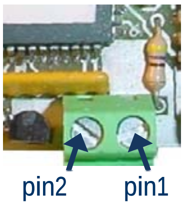


pin1: positive power supply output (5Vdc, without current limitation) to supply a possible external detection logic for negative run detection.
pin2: positive power supply for infrared LED of the Y axis (for optical limit detection).
pin3: input of the limit detection sensor of the Y axis.
pin4: GND.
pin5: negative power supply for infrared LED of the Y axis (for optical limit detection).
pin6: positive power supply output (5Vdc, without current limitation) to supply a possible external detection logic for negative run detection.
pin7: positive power supply for infrared LED of the X axis (for optical limit detection).
pin8: input of the limit detection sensor of the X axis.
pin9: GND.
pin10: negative power supply for infrared LED of the X axis (for optical limit detection).

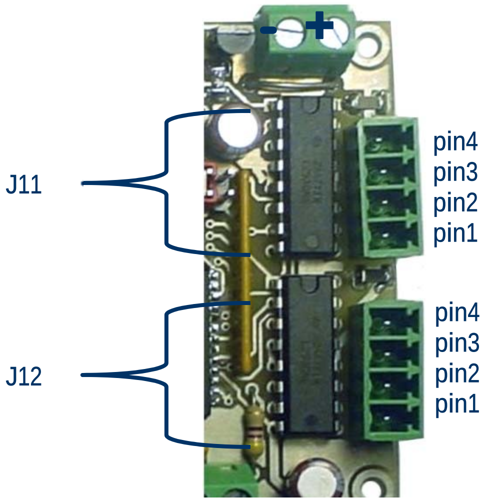


P1 (power supply): *pin1* (+): positive voltage.
pin2 (-): GND.

P2 (aux out): *pin1*: positive voltage (limited by using a resistor).
pin2: *open-collector* to GND.



J11 (X axis):	MT2USB:	*pin1*	Phase A+
		pin2	Phase B+
		pin3	Phase B-
		pin4	Phase A-
	MT2USBMS:	*pin1*	Phase A+
		pin2	Phase A-
		pin3	Phase B+
		pin4	Phase B-
J12 (Y axis):	MT2USB:	*pin1*	Phase A+
		pin2	Phase B+
		pin3	Phase B-
		pin4	Phase A-
	MT2USBMS:	*pin1*	Phase A+
		pin2	Phase A-
		pin3	Phase B+
		pin4	Phase B-



Use 3,81 mm Phoenix Contact connectors to connect motor to J11 and J12[2].

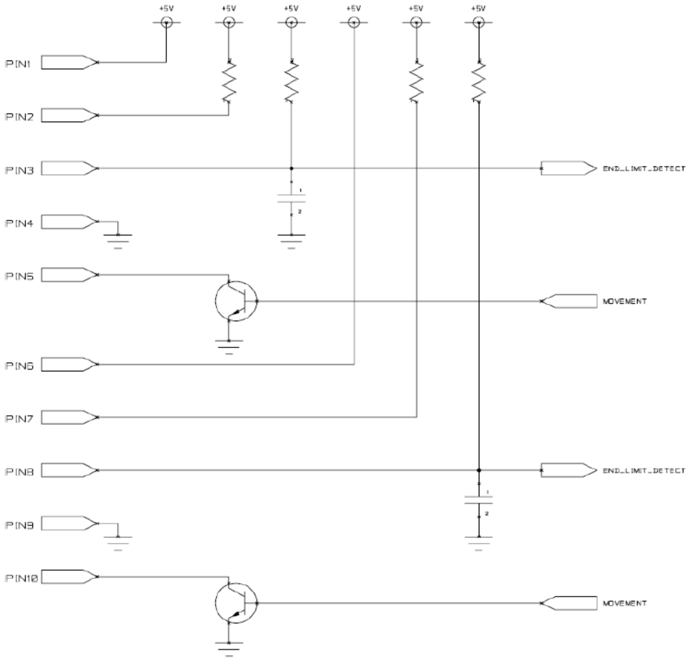
WARNING!

- Do not connect or disconnect motors or power leads with power applied!
- The +5Vdc voltage on *pin1* and *pin6* of home position connector is directly gained from the USB power supply of the PC. In case of voltage overload or shorting the PC could be damaged.

[2] RS code # 220-4670 or Distrelec code # 141128

LIMIT SWITCH CONNECTION EXAMPLES

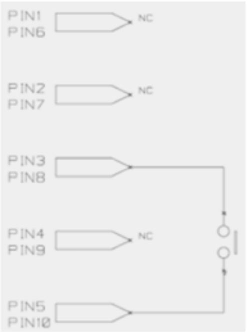
The next Picture shows implementation scheme of MT2USBMS J2 connector.



Picture 6: implementation scheme of J2 connector.

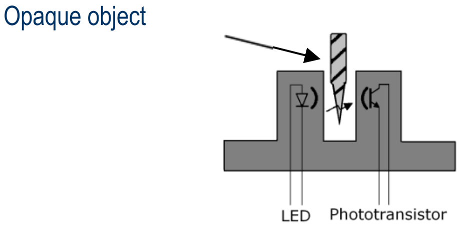
During motors running, MT2USBMS reads home position which is reached when *pin3* (for Y axis movement) or *pin8* (for X axis movement) of J2 connector changes its electric potential. The J6 connector sets the limit switch polarity: if it is inserted, the “End-of-run” signal is enabled when GND is applied; otherwise if J6 is not inserted, the “End-of-run” signal is enabled when the voltage is set at +5V.

In case of mechanics limit switches: for normally opened (N.O.) ones, J6 must be inserted; for normally close (N.C.) ones, J6 must not.



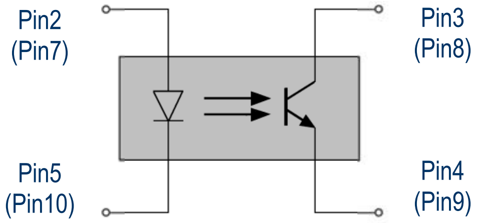
Picture 7: mechanics limit switches commutation.

The limit switch signal can be received by optical sensors. The following Picture shows the functioning of optical sensors. The LED lights an element with a beam, for example a *phototransistor*; then this enlightened element changes its electrical properties. The *phototransistor* gives a conductive path; but it does not if the beam is interrupted by the interposition of an opaque object.



Picture 8: optical limit switch.

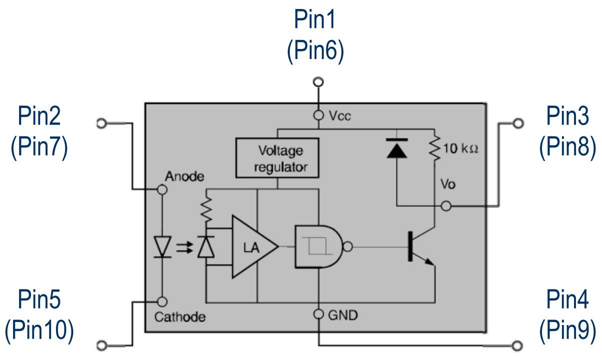
The next Picture indicates the linkage with J2 connector when a *phototransistor* output sensor is used.



Picture 9: *phototransistor* limit switch linkage with J2 pins are shown for Y axis (X axis) limit detection.

To utilize a *phototransistor* output, connector J6 must be removed.

Higher performance in terms of reliability of the reading of the home position can be reached with the adoption of optical sensors equipped with electronic circuits for a more accurate detection of the beam emitted by the LED. Picture 10 the scheme of electronic logic equipped with optical sensors is shown: the beam interruption by the interposition of an opaque object is detected by a power supplied system control. *Pin1* (Y axis) and *pin6* (X axis) give power supply.



Picture 10: electronic logic equipped optical switches; linkage with J2 pins are shown for Y axis (X axis) limit detection.

To use this kind of sensor, J6 connector must be inserted.

WARNING!
The maximum input voltage for *pin3* and *pin8* must not exceed +5Vdc.
For higher voltage the components of the board may damage irremediably.

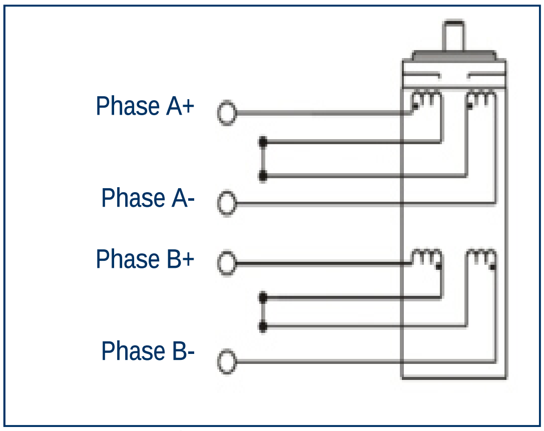
MOTORS CONNECTION

MT2USBMS card can control both two bipolar and two unipolar stepper motors, i.e. 8, 4 and 6 lead centre tapped motors. Here the possible different motor connections are showed.

MOTOR CONNECTION (8 LEAD MOTORS)

Series connection

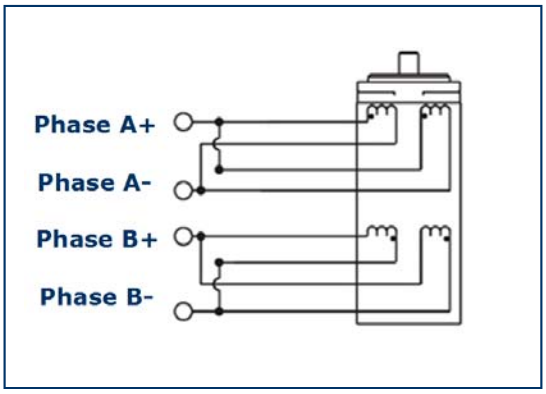
A series motor configuration would typically be used in application where a higher torque at lower speeds is required. Because this configuration has the most inductance, the performance will start to degrade at higher speeds.



Picture 11: series connection.

Parallel connection

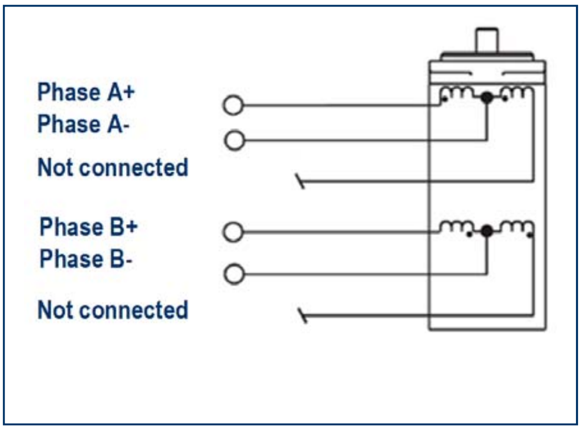
An 8 lead motor in a parallel configuration offers a more stable, but lower torque at lower speeds. But because of the lower inductance, there will be higher torque at higher speeds.



MOTOR CONNECTION (6 LEAD MOTORS)

Half coil configuration

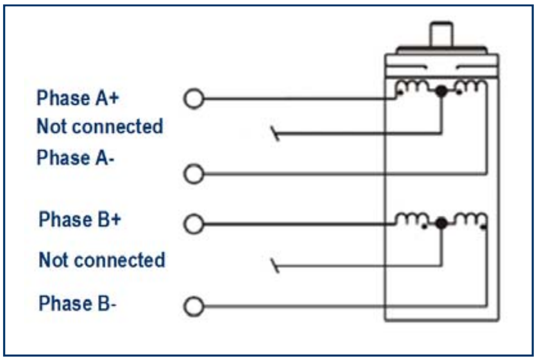
This configuration uses 50% of the motor phase windings. This gives lower inductance, hence, lower torque output. Like the parallel connection of 8 lead motor, the torque output will be more stable at higher speeds. This configuration is also referred to as **half copper**.



Picture 13: *half coil* configuration.

Full coil configuration

The full coil configuration on a 6 lead motor should be used in applications where higher torque at lower speeds is desired. This configuration is also referred to as **full copper**.

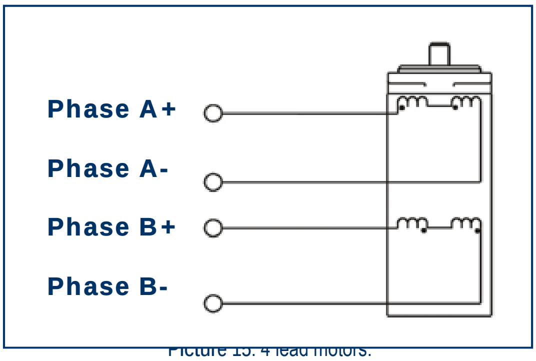


Picture 14: *full coil* configuration.

MOTOR CONNECTION (4 LEAD MOTORS)

4 lead motors

4 lead motors are the least flexible but easiest to wire. Speed and torque will depend on winding inductance.

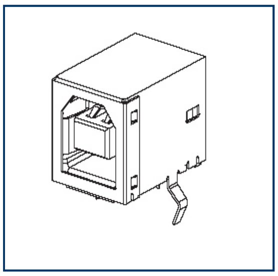


Picture 15: 4 lead motors.

NOTE

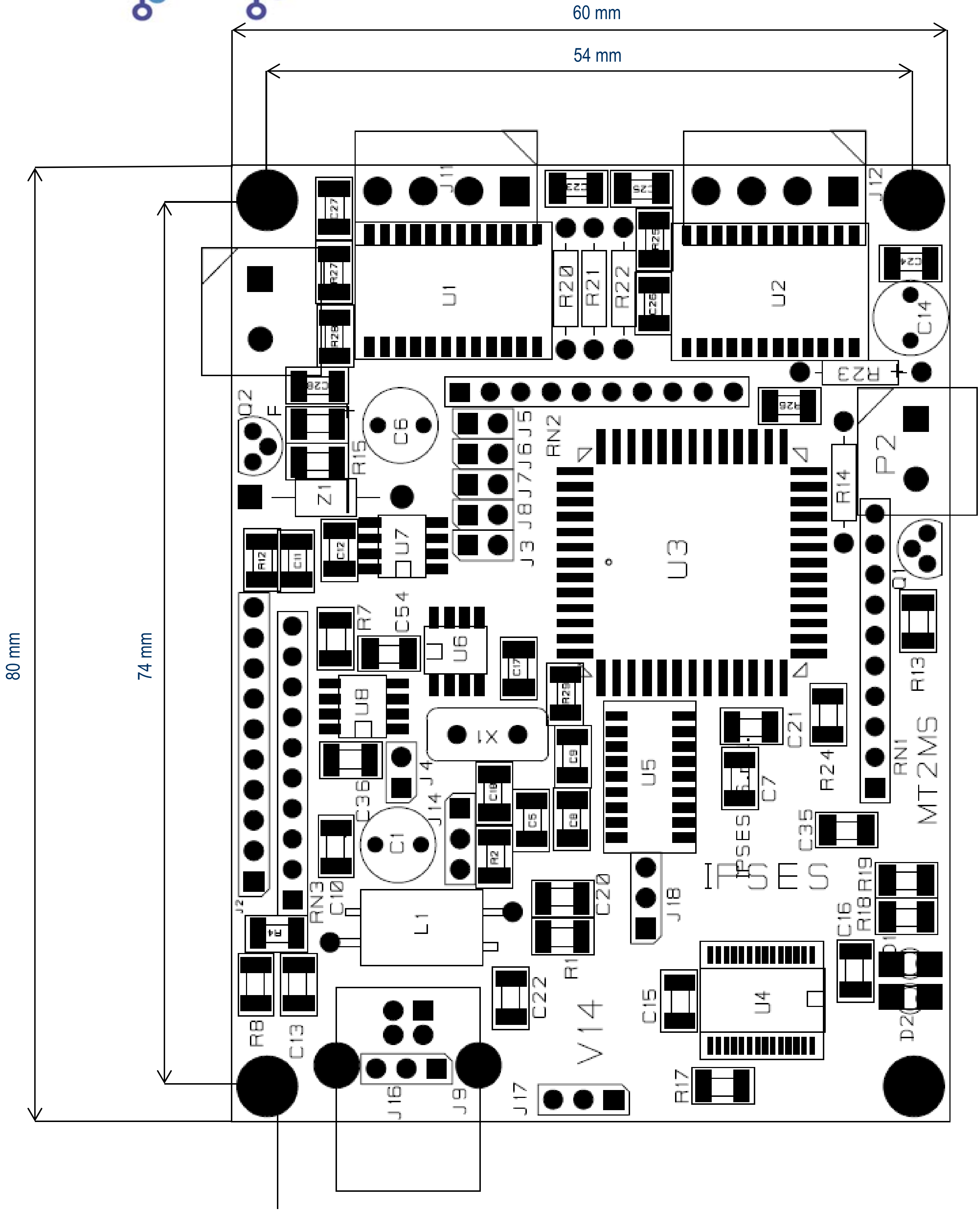
The physical direction of motors depends on connection of motor windings. To switch the direction of the motor, switch the wires on either phase A or phase B outputs.

TECHNICAL FEATURES



Power supply:	The logic is self-supplied directly from the computer USB port
Motor power supply:	from 4,5 up to 30Vdc
Output current:	max 0,75A/phase (0,85A of peak)
Interface:	USB 2.0 (B connector type)
Motor movement:	programmable from the whole step up to 1/8 of step
Auxiliary output:	open collector configuration (absorbed Imax: 200mA)
Board dimensions:	60 x 80 x 15 mm (2,36 x 3,15 x 0,59 inch) without connectors

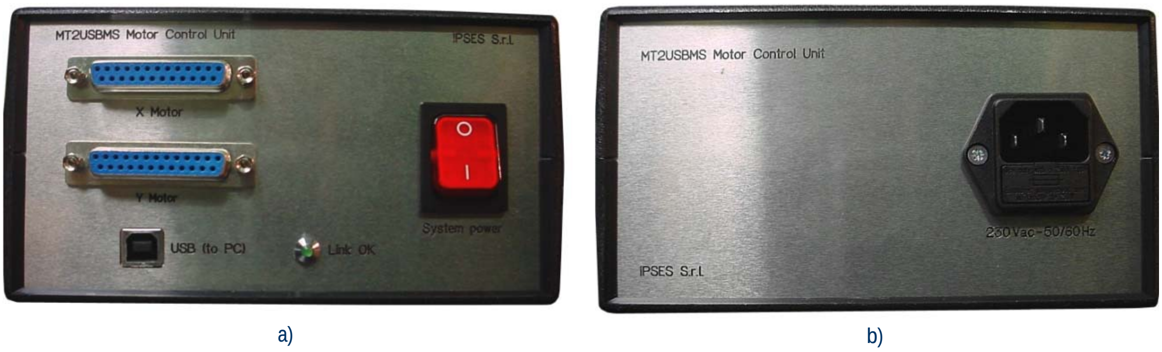
Note: the power supplied the motors is shared with the auxiliary output.



MT2USBMS card dimensions

BOX

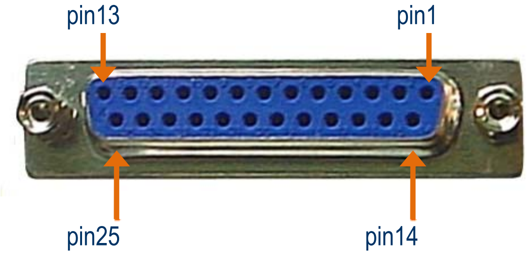
On demand, all MT2 motor control cards are available on box with power pack included. Picture shows MT2USBMS-BOX.



Picture 16 a) front and b) rear panel of MT2USBMS box

Breadth, height and length of the box are respectively 158 mm, 85 mm and 170 mm (6.2 x 3.3 x 6.7 inches). On the rear side there is the socket to link the box to the electrical network (230Vac, 50/60Hz). On the front panel there are: the general power switch, a USB socket to link the device to the PC, a LED indicating working connection, two DB25 standard 25 pin connectors to be used to connect the motors.

Pinout of the 25 poles connector is the following one:



DB25 connector pin	Motor pin
1	positive power supply output (5Vdc, without current limitation) to supply a possible external detection logic for negative run detection
2	positive power supply for infrared LED (for optical limit detection)
3	input of the limit detection sensor
4	GND
5	negative power supply for infrared LED (for optical limit detection)
6	N.C.
7	N.C.
8	N.C.

9	N.C.
10	N.C.
11	N.C.
12	N.C.
13	N.C.
14	AUX OUT: positive voltage (limited by using a resistor)
15	AUX OUT: *open-collector* to GND
16	N.C.
17	N.C.
18	N.C.
19	N.C.
20	N.C.
21	N.C.
22	Motor Phase B+
23	Motor Phase B-
24	Motor Phase A+
25	Motor Phase A-

To avoid malfunctioning, please link the device with the USB cable only after you have turned on the MT2USBMS-BOX.

DEMO SOFTWARE

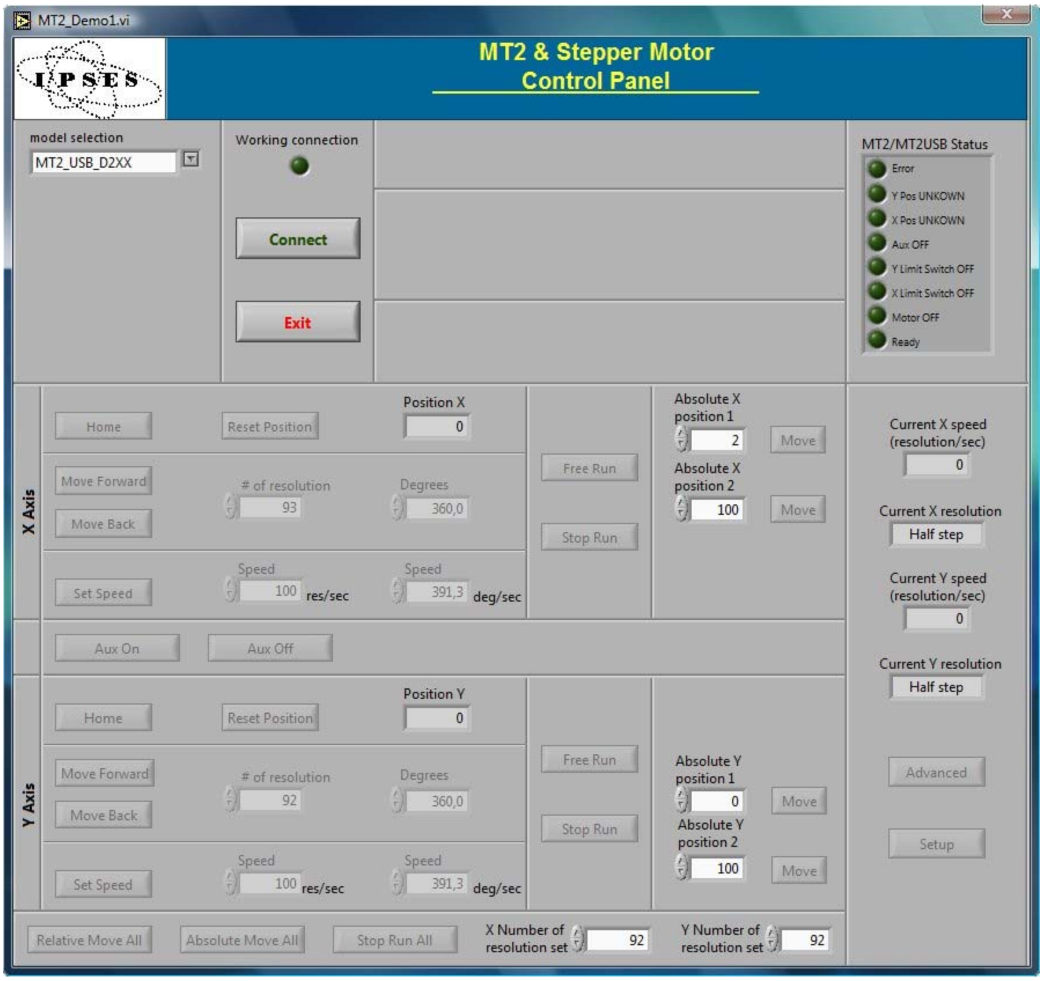
MT2_Demo is a demo software which allows MT2 device control testing from a PC. Displayed virtual control panel has intuitive functionalities which make you easy understand how it works.

INSALLATION

To install the software on your PC, execute “Setup.exe” and follow instructions displayed. Default destination folder of the executable file “MT2_Demo.exe” is “C:\Program Files\MT2_Demo”.

EXECUTION

Execute “MT2_Demo.exe”. Virtual control panel is displayed as showed in Picture 17:



model selection: MT2_USB_D2XX

Position X: 0

# of resolution: 93

Degrees: 360,0

Speed: 100 res/sec

Speed: 391,3 deg/sec

Absolute X position 1: 2

Absolute X position 2: 100

Current X speed (resolution/sec): 0

Current X resolution: Half step

Current Y speed (resolution/sec): 0

Current Y resolution: Half step

Position Y: 0

# of resolution: 92

Degrees: 360,0

Speed: 100 res/sec

Speed: 391,3 deg/sec

Absolute Y position 1: 0

Absolute Y position 2: 100

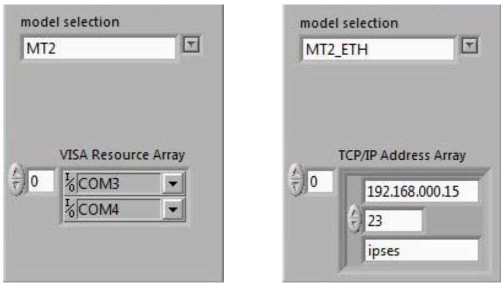
X Number of resolution set: 92

Y Number of resolution set: 92

Picture17: virtual control panel

CONNECTION TO MT2 DEVICE

To start dialogue with MT2 device you have to choose the relevant option in the field **model selection**. Specific connection parameter fields will be displayed for **Serial** (VCP) and **Ethernet** versions. For the first one must indicate the *COM port(s)*, for the second one *IP address(es)*, *IP port(s)* and *password(s)*, as shown in Picture 18. It is not possible to manage different model cards at the same time. **Connect** starts connection, while **Exit** close the Demo software.



model selection: MT2

VISA Resource Array: 0 | COM3 | COM4

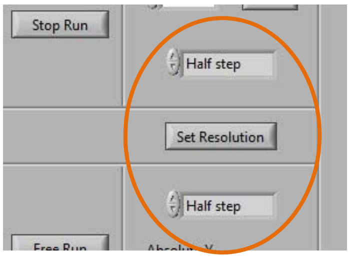
model selection: MT2_ETH

TCP/IP Address Array: 0 | 192.168.000.15 | 23 | ipses

Picture 18: a) selection parameters fields for serial protocol communication (VCP) and b) selection parameters field for Ethernet protocol communication

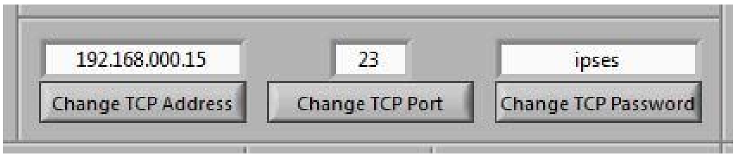
If the connection procedure successfully pass the fictitious LED **Working connection** turns on. The **S/N list** menu lists all connected devices and **Info device** field gives information about *firmware version* and *serial number* of the current selected board, as shown in Picture 18. All the panel commands are activated.

If the selected model is a *micro stepper* version, on the front panel will be showed also the selector controls to define the movement resolution of each motors as shown in Picture 19. Press **Set Resolution** to apply the current settings.



Picture19: controls to set motors' resolution

As shown in Picture 20, for all Ethernet models, fields and commands to modify the TCP/IP protocol access parameters, such as *address*, *port* and *password*, will be activated.

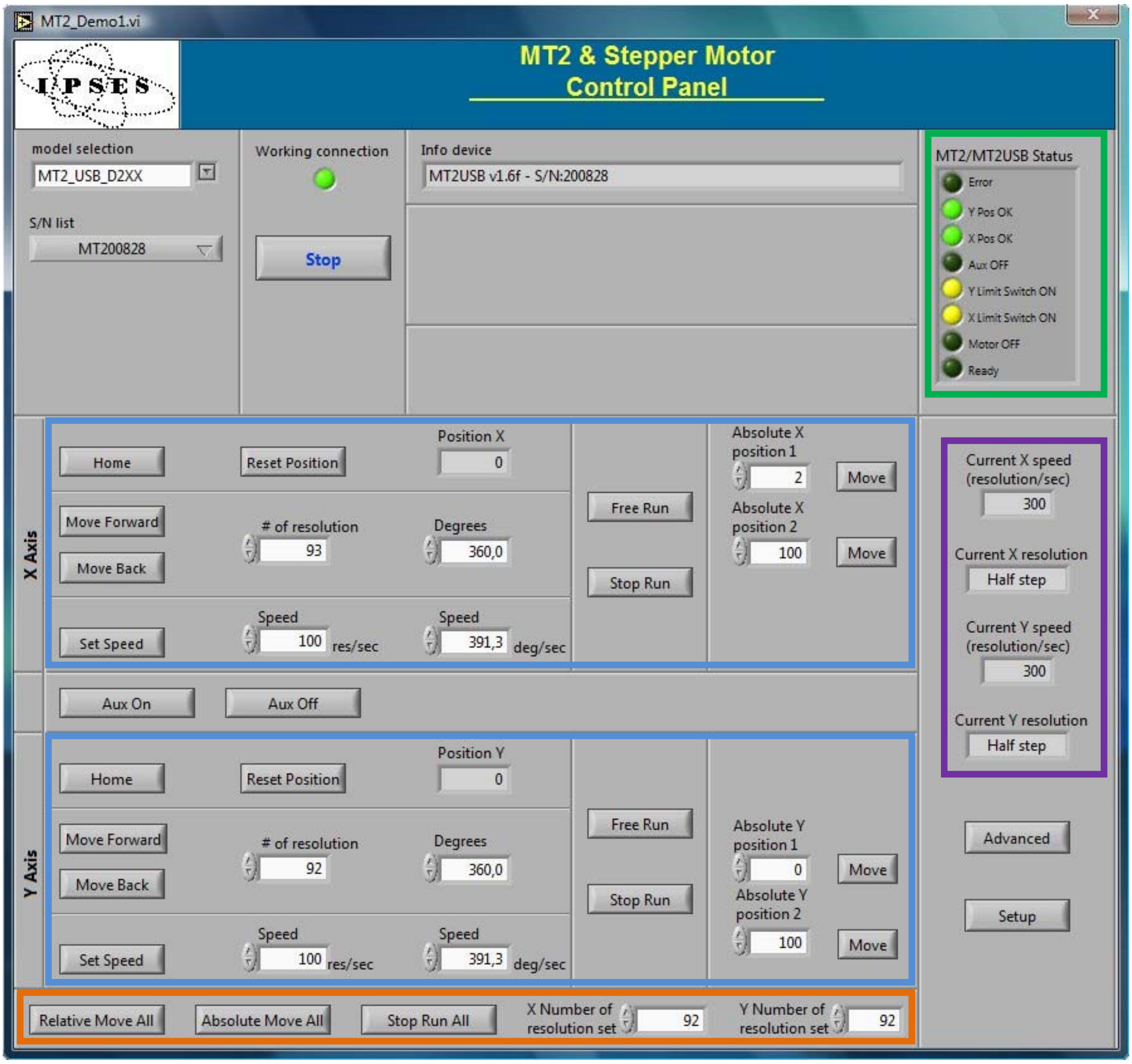


192.168.000.15 | 23 | ipses

Picture 20: controls to modify the TCP/IP protocol connection parameters

FUNCTIONALITIES

Virtual control panel is structured to make you easy understand implemented functions.



model selection: MT2_USB_D2XX

S/N list: MT200828

Info device: MT2USB v1.6f - S/N:200828

Position X: 0

# of resolution: 93

Degrees: 360,0

Speed: 100 res/sec

Speed: 391,3 deg/sec

Absolute X position 1: 2

Absolute X position 2: 100

Current X speed (resolution/sec): 300

Current X resolution: Half step

Current Y speed (resolution/sec): 300

Current Y resolution: Half step

Position Y: 0

# of resolution: 92

Degrees: 360,0

Speed: 100 res/sec

Speed: 391,3 deg/sec

Absolute Y position 1: 0

Absolute Y position 2: 100

X Number of resolution set: 92

Y Number of resolution set: 92

Picture 21: active panel

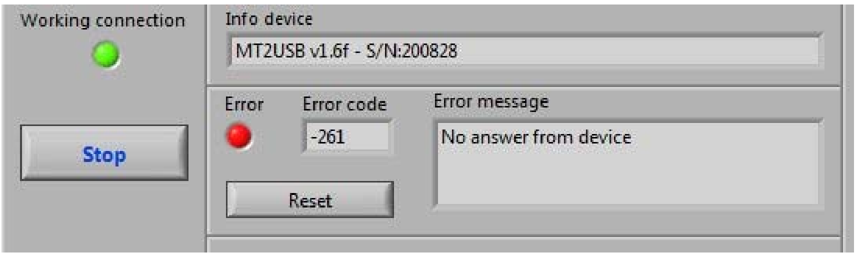
The blue boarded area in Picture 21 includes controls and commands for setting and for the movements of both X and Y axes respectively.

Consider controls and commands for the X axis.

Home command moves X axis motor to home position: during the execution of a **home** command, for serial and USB versions, it is not possible to send other commands. Otherwise, during the execution of this command, it is possible to change the active control device by selecting a different **S/N list** item, or to abort demo software by pressing the **Quit** button (which substitutes the **Stop** button under this condition).
Reset position makes the actual position equivalent to zero displacement (**Position** X axis, displacement along X, is zero).
# of resolution or **Degrees** define the relative positive or negative displacement, enabled by **Move Forward** or **Move Back** commands, respectively.
Free Run allows perpetual motion of the X axis, while **Stop Run** arrests it.
Speed can be declared in resolution per second or, similarly to the displacement, in degrees per second.
Set Speed updates the current speed values.

Absolute X position 1 and Absolute X position 2 set the amount of two absolute displacements, referred to zero position.
Move commands allow these movements.
There are analogue controls for Y motor control.
The lower panel area, surrounded in orange in picture 21, includes the commands for the simultaneously movement of both axes, either in relative and absolute terms.
Aux On enables the auxiliary output of the card, while Aux Off disables it.
The indicators, surrounded in violet in Picture 21, show the current settings for motor speeds and resolutions for both axes.

The software errors are generated as shown in Picture, where is represented an error occurred after a failed communication to the device.

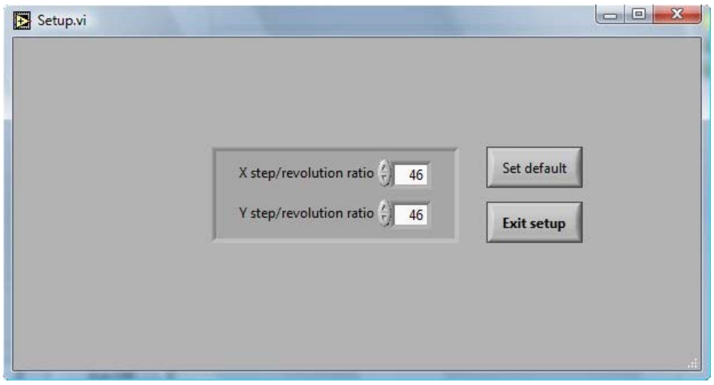


Info device: MT2USB v1.6f - S/N:200828
Error
Error code: -261
Error message: No answer from device

Picture 22: error message

The re-enabling of the application functionalities is suborder to the Reset condition.

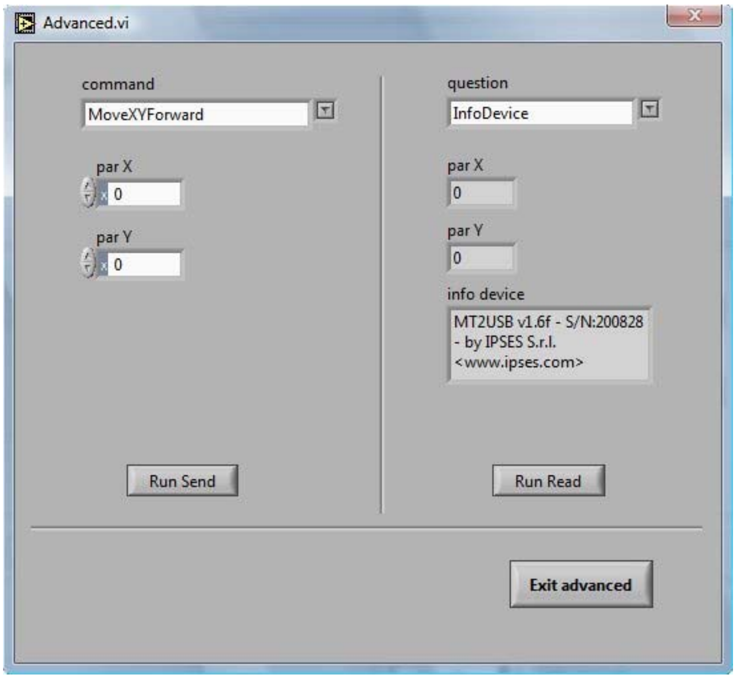
The Setup button opens a new window which allows to set the numbers of steps (X step/revolution ratio and Y step/revolution ratio) necessary for a complete motor revolution, one for each axis. These values are automatically updated at the window closure, forced by Exit Setup. Set default restores default values.



X step/revolution ratio: 46
Y step/revolution ratio: 46

Picture 23: setup window

Advanced runs homonym subroutine, as shown in Picture 24, where user can see a low level dialogue to the active device. Chosen instruction in command and associated parameter par X and par Y are communicated to the device when Run Send is set. Run read allows answer in the appropriate fields (info device, par X 1, par Y or status) to interrogations selected in question.



Advanced.vi

command

MoveXYForward

par X

0

par Y

0

question

InfoDevice

par X

0

par Y

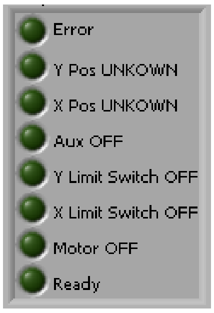
0

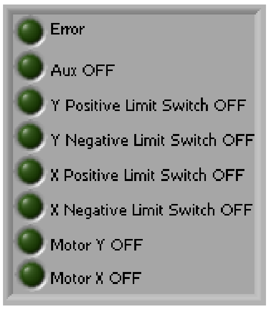
info device

MT2USB v1.6f - S/N:200828 - by IPSES S.r.l. <www.ipses.com>

Picture 24: Advanced subroutine window

The surrounded green area on front panel (see Picture) refers about device status through the stylized LEDs: when a LED lights up, the condition described by the label on its side happens (in this case OFF becomes ON). The labeling descriptions are different between serial/USB and Ethernet versions. Picture shows these differences.





Error
Aux OFF
Y Positive Limit Switch OFF
Y Negative Limit Switch OFF
X Positive Limit Switch OFF
X Negative Limit Switch OFF
Motor Y OFF
Motor X OFF

Error	Device error
Y Pos UNKNOWN	Unknown Y axis position
X Pos UNKNOWN	Unknown X axis position
Aux OFF	Auxiliary output active
Y Limit Switch OFF	Y axis home position reached
X Limit Switch OFF	X axis home position reached
Motor OFF	Motor moving
Ready	Device ready

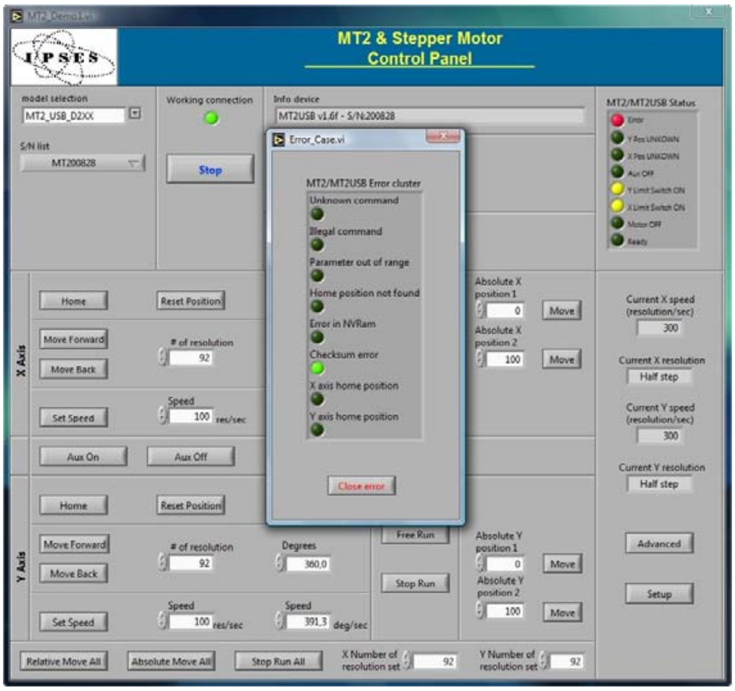
a)

Error	Device error
Aux OFF	Auxiliary output active
Y Positive Limit Switch OFF	Y positive end-of-run signal presents
Y Negative Limit Switch OFF	Y negative end-of-run signal presents
X Positive Limit Switch OFF	X positive end-of-run signal presents
X Negative Limit Switch OFF	X negative end-of-run signal presents
Motor Y OFF	Y axis motor moving
Motor Y OFF	Y axis motor moving

b)

Picture 25:a) status LEDs for serial or USB devices; b) status LEDs for Ethernet devices.

Picture 25 shows the notification of a device error message. To re-enable the operative conditions press **Close error**.



model selection: MT2_USB_D2XX

S/N list: MT200828

Info device: MT2USB v1.6f - S/N:200828

X Axis: # of resolution 92; Speed 100 res/sec

Absolute X position 1: 0

Absolute X position 2: 100

Current X speed (resolution/sec): 300

Current X resolution: Half step

Current Y speed (resolution/sec): 300

Current Y resolution: Half step

Y Axis: # of resolution 92; Degrees 360,0; Speed 100 res/sec; Speed 391,3 deg/sec

Absolute Y position 1: 0

Absolute Y position 2: 100

X Number of resolution set: 92

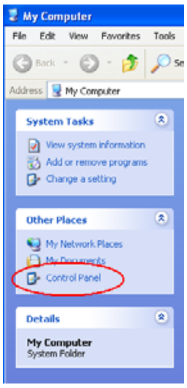
Y Number of resolution set: 92

Picture 26: device error message example

REMOVAL

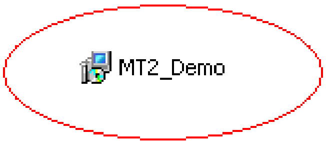
To correctly remove the software, follow the instructions listed below.

1) From Desktop, click “My Computer” icon and choose “Control Panel”.



2) Click “Add or Remove Programs” from the resource list displayed.

3) From program installed list select “MT2_Demo” and proceed removal with “Change/Remove”.



LABVIEW LIBRARY

LabVIEW development tool gives the feasibility of MT2 device remote control. This control can be achieved through the use of the eleven functions implemented in *LabVIEW 7.1* and included in the library MT2_Library: thanks to these functions you do not have to know the details of the communication protocol and the application development is quick and easy.

The functions have two development levels: *MT2_Low_Level_Communication.llb* contains six functions through which is possible to manage the connection with the MT2 card.

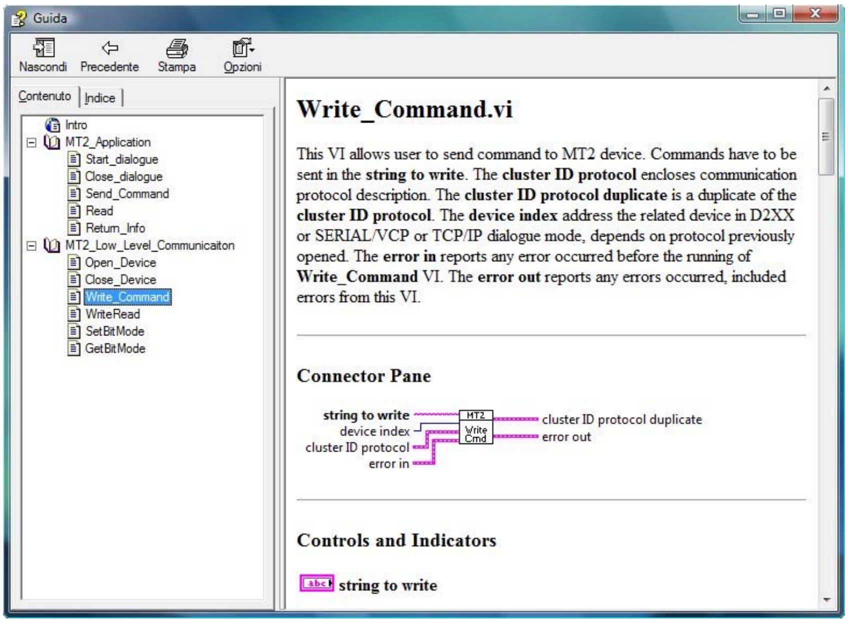
MT2_Application.llb contains other five functions realized through the use of the previous ones: these higher level functions allow the assignment of the commands recognized by the device. Use *MT2_Application.llb* for application development, while *MT2_Low_Level_Communication.llb* for maximize performances.

MT2_Application.llb
MT2_Low_Level_Communication.llb

	Function	Properties
MT2_Low_Level_Communication.llb	*Close_Device.vi*	Closes the connection established with one of the available protocols.
	Open_Device.vi	Opens the connection with one of the available protocols.
	Write&Read.vi	Sends and receives ASCII characters.
	Write_Command.vi	Sends ASCII characters.
	SetBitMode.vi	Sets the values of two more output configured as open collector (valid only for USB versions).
	GetBitMode.vi	Reads the values of two more output configured as open collector (valid only for USB versions).
MT2_Application.llb	*Close_dialogue.VI*	Ends the communication with the MT2 card.
	Read.vi	Sent a request to the device and read its answer.
	Send_Command.vi	Imparts the commands implemented on the device.
	Start_dialogue.vi	Starts dialogue session with the MT2 card.
	Return_Info.vi	Drafts the serial number list of connected devices.

MT2_Library is provided with a help file, *MT2_Help.chm*.
The help explains deeper the functions in the library.
MT2_Help.chm, information of which are available in *LabVIEW* too, gives structural description of all eleven functions. Graphical representations are realized, so that the user may easily understand how they work in the tool in which they were build. Next figure displays the help of the library.

MT2_Help.chm



Write_Command.vi

This VI allows user to send command to MT2 device. Commands have to be sent in the **string to write**. The **cluster ID protocol** encloses communication protocol description. The **cluster ID protocol duplicate** is a duplicate of the **cluster ID protocol**. The **device index** address the related device in D2XX or SERIAL/VCP or TCP/IP dialogue mode, depends on protocol previously opened. The **error in** reports any error occurred before the running of **Write_Command** VI. The **error out** reports any errors occurred, included errors from this VI.

Connector Pane

Controls and Indicators

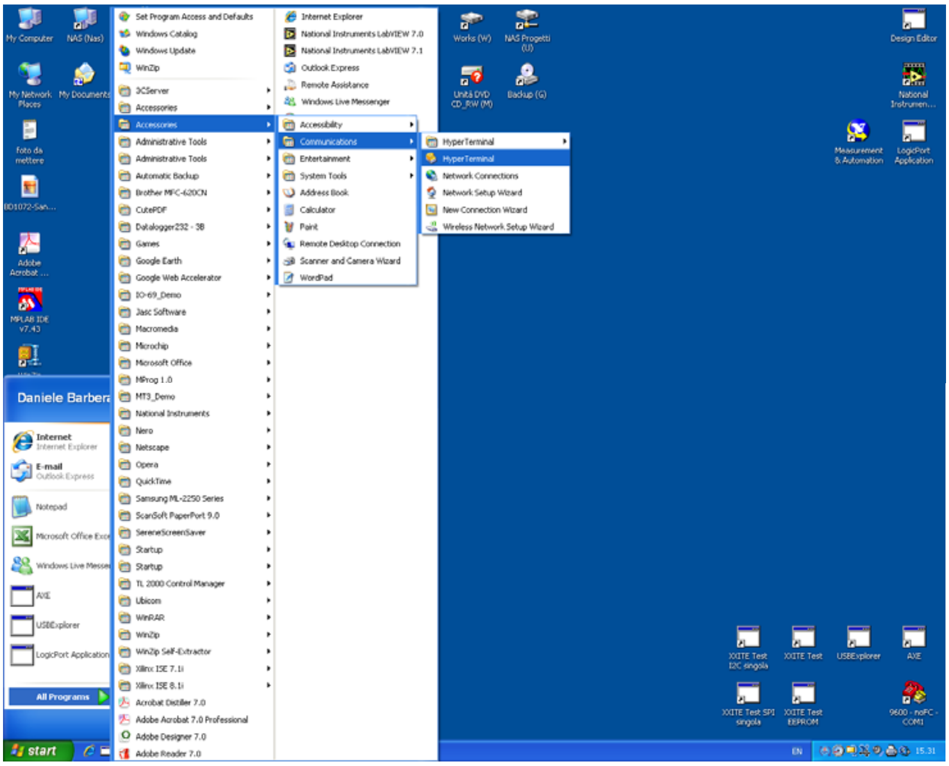
string to write

Picture 27: *LabView* functions help

MT2_Library is available on demand.

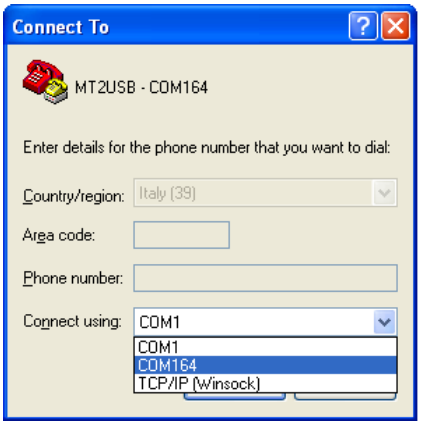
CONNECTION EXAMPLE: HYPERTERMINAL

A typical example of serial connection is represented by HyperTerminal tool, which is present in all operating systems Microsoft Windows (except Windows Vista where the program is not included in the operative system distributions, but it is easily downloadable from internet), running by menu “Start -> All Programs -> Accessories -> Communications -> Hyper Terminal” (Picture 28).

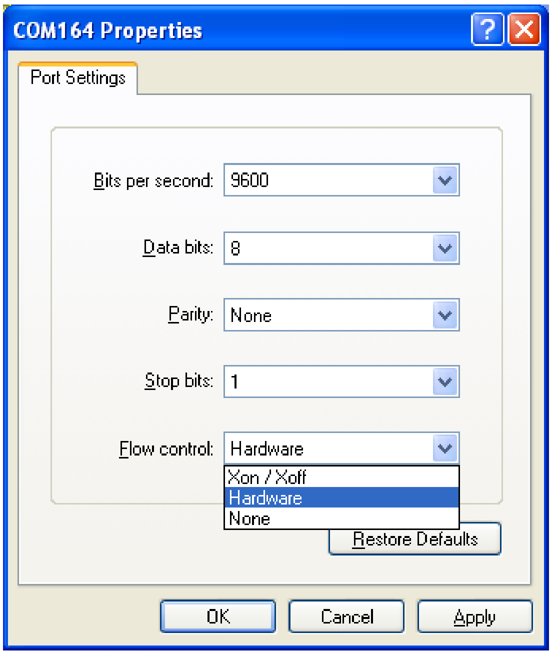


Picture 28: Hyper Terminal path

The serial communication can be used also with USB devices because the driver provides a virtual serial communication port (VCP) for each MT2 system connected. Through the VCP it can be easily established the serial communication with the board. Once the connection is established, a name and some other parameters are needed (as it can be seen in Picture). Before getting information by the board it is necessary to view characters sent by user. To do that, an echo late on what is written has to be set via program properties menu (Picture29 – Picture 31):

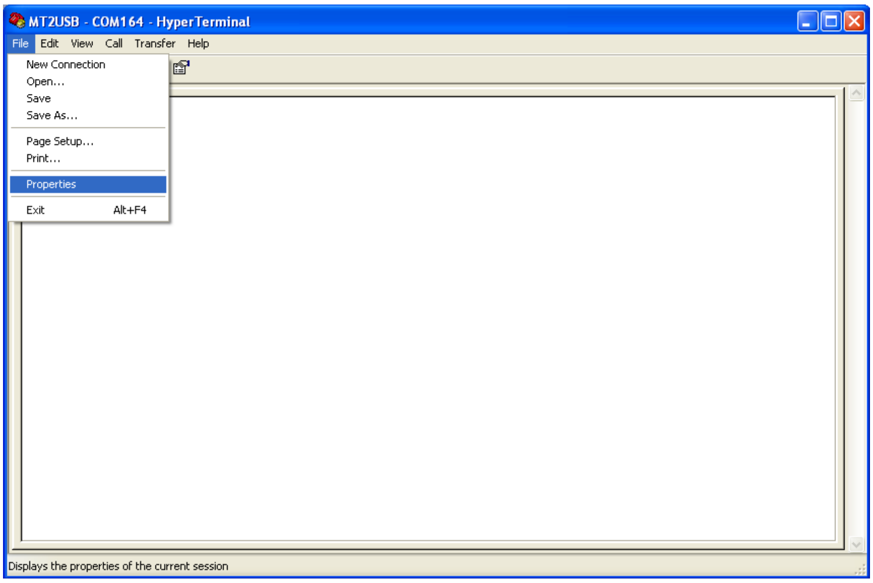


Picture 29: HyperTerminal general parameter settings



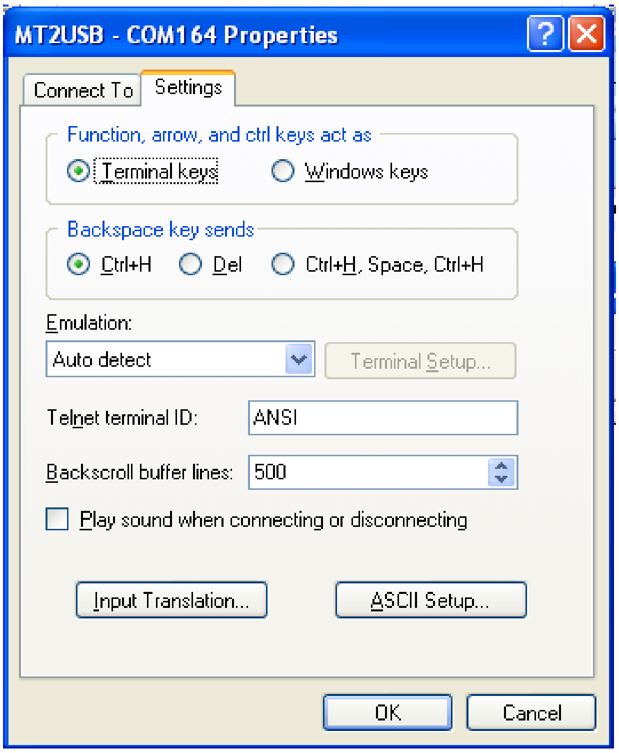
Picture 30: Hyper Terminal serial parameter settings

Before getting information by the board it is necessary to view characters sent by user. To do that, an echo late on what is written has to be set via program properties menu (Picture 31- Picture 33).



Displays the properties of the current session

Picture31: HyperTerminal properties menu



Function, arrow, and ctrl keys act as

● Terminal keys ○ Windows keys

Backspace key sends

● Ctrl+H ○ Del ○ Ctrl+H, Space, Ctrl+H

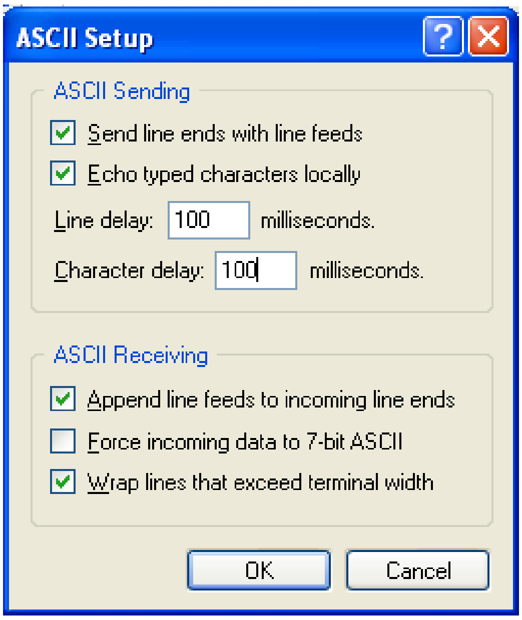
Emulation: Auto detect

Telnet terminal ID: ANSI

Backscroll buffer lines: 500

□ Play sound when connecting or disconnecting

Picture 32: by TAB “Settings”, user can select “ASCII Setup”



ASCII Setup

ASCII Sending

☑ Send line ends with line feeds

☑ Echo typed characters locally

Line delay: 100 milliseconds.

Character delay: 100 milliseconds.

ASCII Receiving

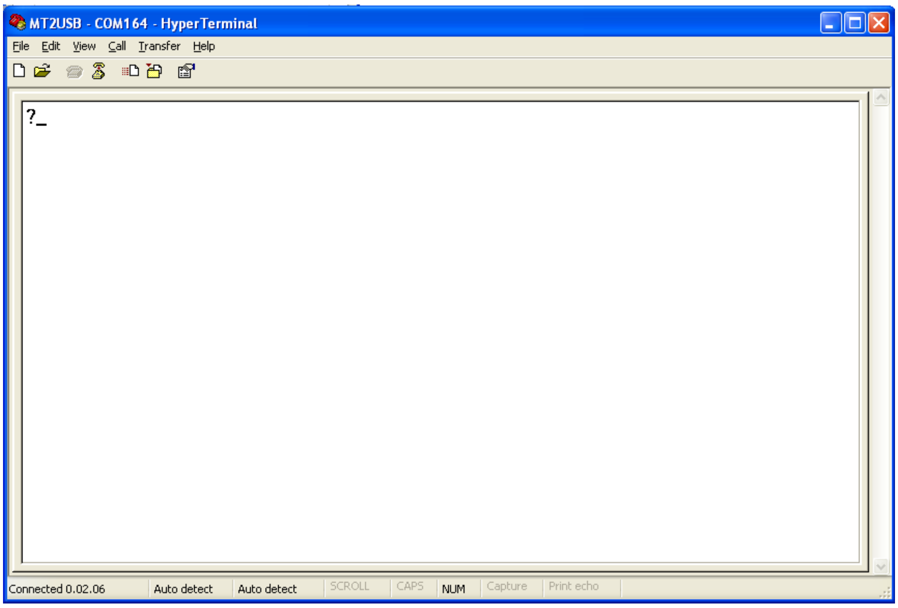
☑ Append line feeds to incoming line ends

□ Force incoming data to 7-bit ASCII

☑ Wrap lines that exceed terminal width

Picture 33: an example configuration to see ASCII characters

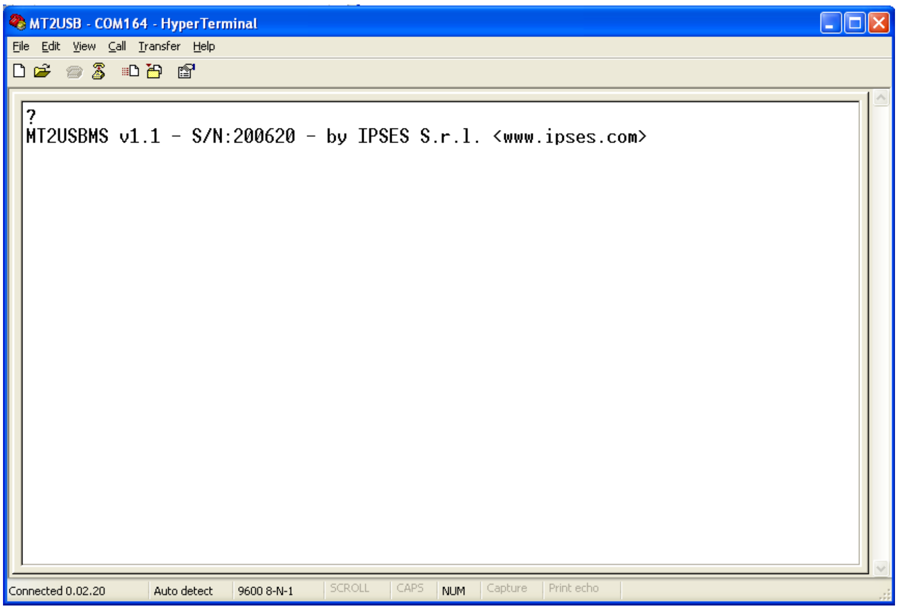
Now communication between PC and MT2 board is established with the protocol previous described. For example, if character “?” is sent by the keyboard (Picture 34), MT2 peripheral shall answer with a string similar to the one shown in Picture:



MT2USB - COM164 - HyperTerminal
File Edit View Call Transfer Help

```
?_
```

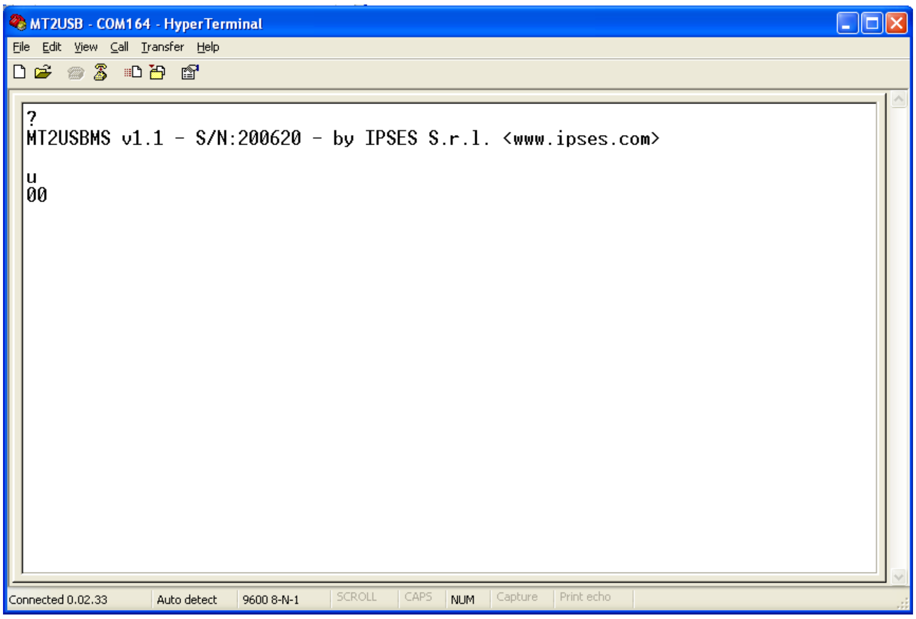
Picture 34: get firmware version command



```
?
MT2USBMS v1.1 - S/N:200620 - by IPSES S.r.l. <www.ipses.com>
```

Picture 35: firmware version string example

Sending other kind of commands in succession, it is possible to communicate and to operate with MT2USBMS via its COM port 164 (Picture 35 the get status function after the get firmware version can be seen).



```
?
MT2USBMS v1.1 - S/N:200620 - by IPSES S.r.l. <www.ipses.com>

u
00
```

Picture 36: get status register.

EXAMPLE USING THE DLL D2XX

The following code example shows how to open communication toward the 0 indexed device and how to configure transmission parameters. The code also allows the reading of *firmware* version and device's *serial number* (command "?") and the sending of X axis movement command (for 2000 steps) and the subsequently reading of device status (command "U"). At the end it closes the communication.

```
// Variables definition
unsigned long ftStatus = 0, ftHandle = 0;
unsigned long TxBytes = 0, RxBytes = 0, EventNode = 0, BytesWritten = 0, BytesReceived = 0;
char TxBuffer [16] = " ";
char RxBuffer [256] = " ";

// Open Device Communication to  0 indexed device and sets its communication parameters
ftStatus = FT_Open (0, &ftHandle);
if (ftStatus != FT_OK)
{
    //Error on opening procedure
}
else
{
    ftStatus = FT_SetBaudRate (ftHandle, 9600);
    if (ftStatus != FT_OK)
    {
        //Error on setting baud rate procedure
    }
    else
    {
        ftStatus = FT_SetDataCharacteristics (ftHandle, FT_BITS_8, FT_STOP_BITS_1, FT_PARITY_NONE
        );
        if (ftStatus != FT_OK)
        {
            //Error on setting data characteristics  procedure
        }
        else
        {
            ftStatus = FT_SetFlowControl (ftHandle, FT_FLOW_NONE, NULL, NULL);
            if (ftStatus != FT_OK)
            {
                //Error on setting flow control  procedure
            }
            else
            {
                ftStatus = FT_SetTimeouts (ftHandle, 500, 300);
                if (ftStatus != FT_OK)
                {
                    //Error on setting timeout procedure
                }
                else
                {
                    //Opening procedure successfully completed
                }
            }
        }
        }
}

//Get Info device
TxBuffer = “?/r”;
ftStatus = FT_Write (ftHandle, TxBuffer, sizeof(TxBuffer), &BytesWritten);
if (ftStatus != FT_OK){
{
        //Write error
}
else
{
        FT_GetStatus(ftHandle, &RxBytes, & TxBytes, &EventNode);
        if (RxBytes >0 )
        {
                ftStatus = FT_Read(ftHandle, RxBuffer, RxBytes, &BytesReceived);
                if (ftStatus == FT_OK)
                {
                        // successfully reading
                }
                else
                {
                        // Error reading
                }
        }
}

//Send command: movement of X axis for 2000 steps
TxBuffer = “d2000,0/r”;
ftStatus = FT_Write (ftHandle, TxBuffer, sizeof(TxBuffer), &BytesWritten);
if (ftStatus != FT_OK){
{
        //Write error
}
else
{
        //Command sent
}

//Get Status
TxBuffer = “u/r”;
ftStatus = FT_Write (ftHandle, TxBuffer, sizeof(TxBuffer), &BytesWritten);
if (ftStatus != FT_OK){
{
        //Write error
}
else
{
        FT_GetStatus(ftHandle, &RxBytes, & TxBytes, &EventNode);
        if (RxBytes >0 )
        {
                ftStatus = FT_Read(ftHandle, RxBuffer, RxBytes, &BytesReceived);
                if (ftStatus == FT_OK)
                {
                        // Status successfully reading
                }
                else
                {
                        // Error reading
                }
        }
}

//Close device
FT_Close (ftHandle);
```

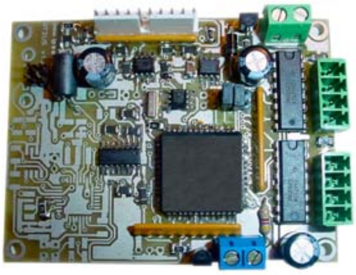
PRODUCT CODES

Code	Description
MT2USB	Two axes stepper motor control card with USB interface
MT2USBMS	Two axes microstepper motor control card with USB interface
MT2USB-box	Two axes control system with MT2USB card and power supply integrated in a case
MT2USBMS-box	Two axes control system with MT2USBMS card and power supply integrated in a case
MT2Library	LabVIEW 7.1 (and further version) library for MT2 card series
MC-connect	*Bundle* of 2 MC male connectors (for motors) + 2 AMP connectors (for limit home detectors) *(for MT2USB, MT2USBMS, MT2, MT2MS and MT2ETHMS)*
USB-A-B	USB cable for USB cards and systems (lenght: 1,8m – 70.9 inches)
USB-A-B-ill	USB cable with light end for USB cards and systems

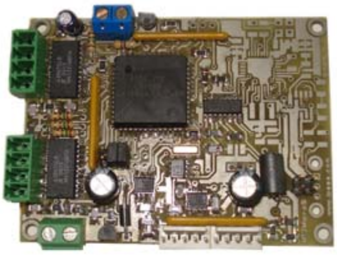
OTHER AVAILABLE MODELS

IPSES can realize customized versions of this device to answer to any clients' demand.
Particularly, it is possible to have this instrument in any size (so as to easily integrate it in any mechanical system) and with customized communication protocol.
The MT2USB card is also available in MT2USBMS version with microstepper control up to 1/8 step.
There are also available axes control cards with serial interface *RS232* (see the *MT2* and *MT2MS* systems) and with Ethernet interface (see the *MT2ETH* and *MT2ETHMS* systems). It is also available a version with USB interface and PWM control current up to 3A (MT2HC). On request we can conceive and develop system with any communication interface, according to our client's specifications.
For each version on demand is available the box configuration.

For further information, please visit the website *http://www.ipses.com.*



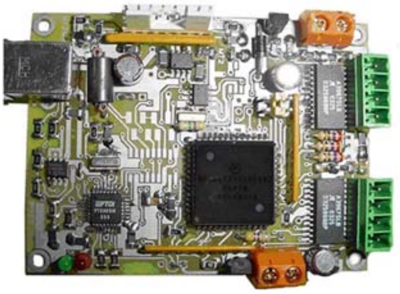
MT2



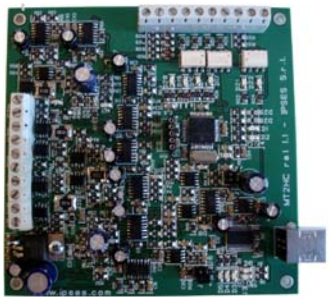
MT2MS

MT2ETH

MT2ETHMS



MT2USBMS



MT2HC

CONTACTS

IPSES S.r.l. conceives, projects and markets electronic and scientific instruments. The customized planning of our devices allows us to answer specific necessities for customers asking for embedded systems. IPSES clients enjoy access to a dedicated project engineering team, available as needed.

Our pool consists of highly competent professionals whose experience in this field is extremely strong. Thanks to constant updating and technical development, IPSES is a leading company, combining the dynamism of a young group into the competence and reliability of a qualified staff.

IPSES S.r.l.
Research and development office:
via Suor Lazzarotto, 10
20020 Cesate (MI)
Italy

tel. +39 02 39449519
+39 02 320629547
fax +39 02 700403170
e-mail: info@ipses.com

IPSES
scientific electronics

SISTEMA DI GESTIONE CERTIFICATO
ITALCERT
UNI EN ISO 9001

SUPPORT INFORMATION

The customer is at liberty to contact the relevant engineer at IPSES S.r.l. directly.
A call can be logged in a variety of ways:

Telephone	:	++39 02 39449519 ++39 02 320629547
Fax	:	++39 02 700403170
Email	:	support@ipses.com

PROBLEM REPORT

The next page is a standard template used for reporting system problems. It can be copied and send as a fax. Alternative bugs may be reported by emails, in this case please insure that the mail contains similar information as the *Engineering Problem Report* form.

ENGINEERING PROBLEM REPORT

Problem describer

Name			IPSES s.r.l. Via Lazzarotto, 10 Cesate (MI) Italy Fax ++39 02/700403170 e-mail *support@ipses.com*
Company			
Date	Tel.	Fax	

Product

Name	Version	Serial No.

Report Type (bug, change request or technical problem)

Major bug	□	Urgency:	
Minor bug	□	High	□
Change request	□	Medium	□
Technical problem	□	Low	□

Problem Description

Reproduction of Problem

IPSES s.r.l. Action notes

Received by	Date	Report No.	Action

(Product code MT2USB- Rel. 01.00.0002)

IPSES S.r.l.
Via Lazzarotto, 10
20020 CESATE (MI) - ITALY
Tel. (+39) 02/39449519
Fax (+39) 02/700403170
e-mail: info@ipses.com
support@ipses.com