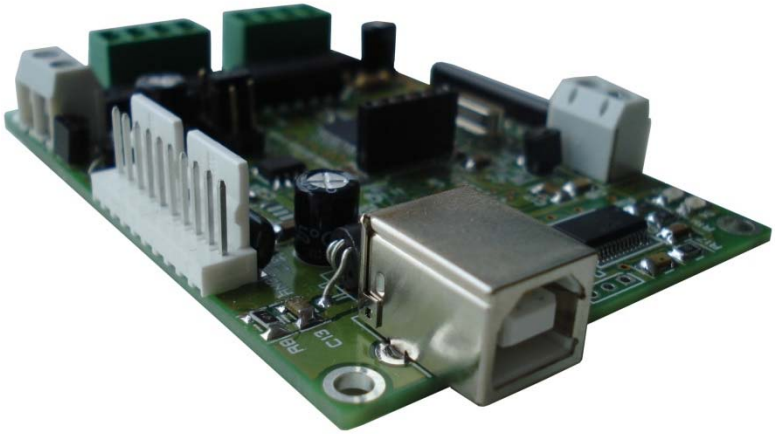
IPSES®
scientific electronics

AXES CONTROL UNIT
MT2USB

Rel. 01.00.0002
(Product Code: MT2USB)



This manual in English is the original version.

Printed in Italy

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 info@ipses.com

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 MT2USB and 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

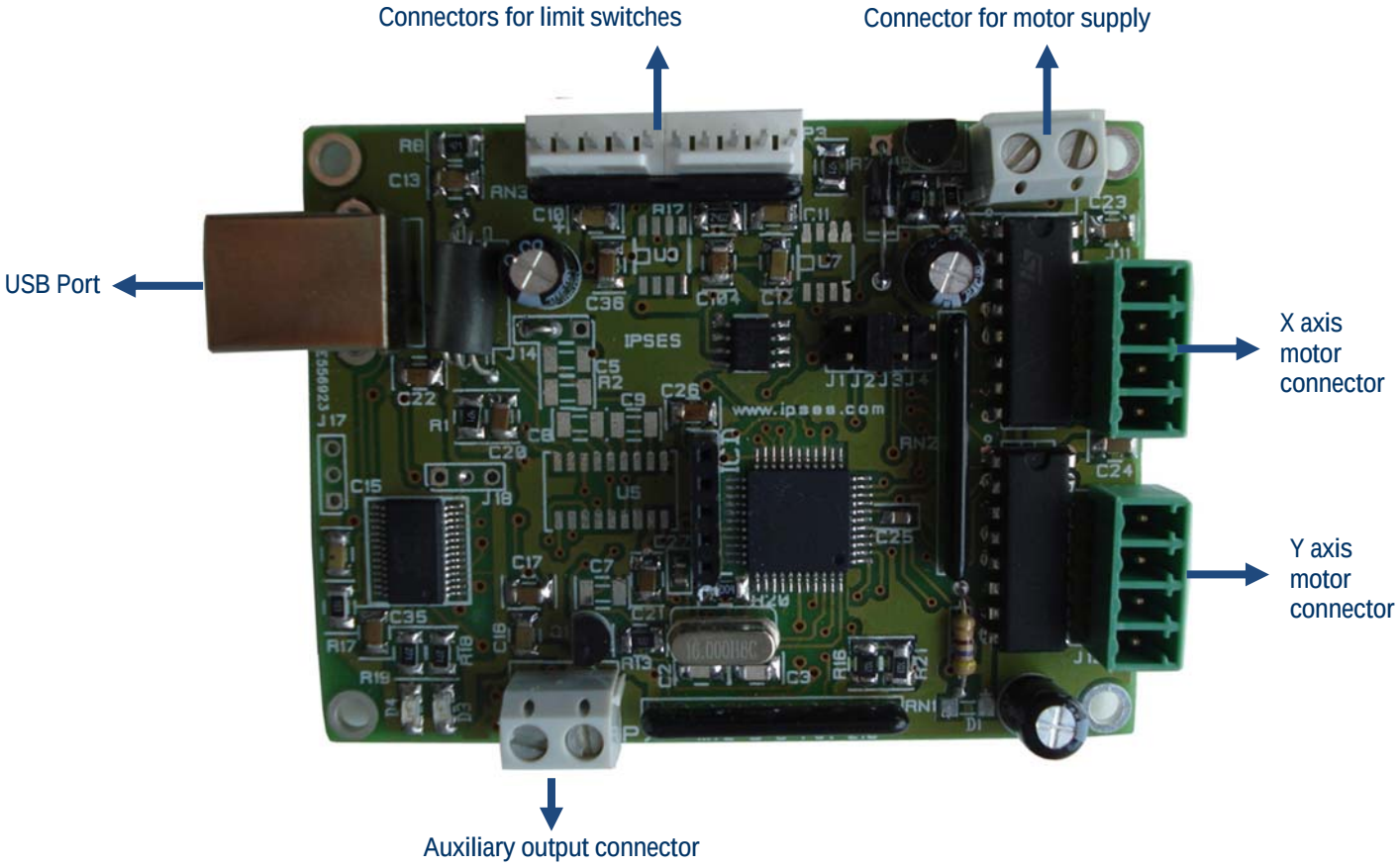
Manual revision history

Revisione/ Data	Descrizione modifica	Autore
01.00.0000 October, 2013	First version released	Mancuso C. Pizzocolo G.
01.00.0001 February 2016	Minor changes	Bottaccioli M.
01.00.0002 August, 2016	Added ISO 9001:20015 logo	Bottaccioli M.

GENERAL FEATURES

MT2USB is a small size low power control device 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 limit 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 per second can be set as needed.
Moreover, the device has an auxiliary output with a dedicated control for its activation. In Picture 1 MT2USB card is shown with position of connectors and USB port.



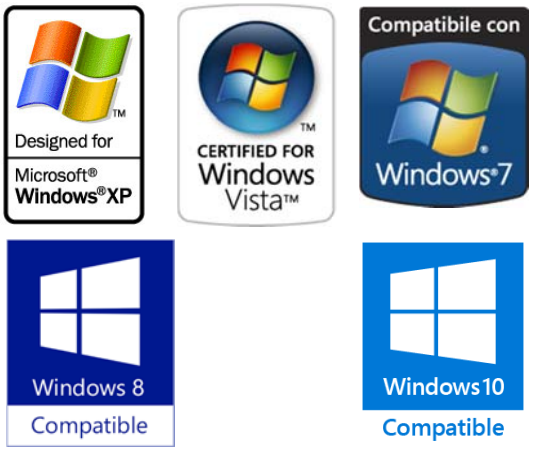
Picture 1: MT2USB card - Position of connectors and USB port

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 MT2USB(MS) 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 MT2USB board to PC using a USB cable. *Windows* operating system will detect a new device, showing a message similar to:

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.

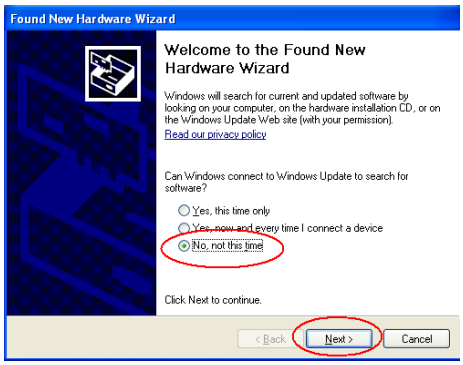
3) After a window with the message "Found New Hardware. USB Serial Port" is displayed.

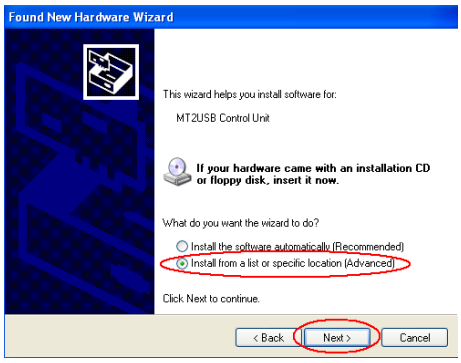
4) In the following windows "found new hardware wizard" chose "Yes, this time only" and then "Next". Wait for a complete download of the driver and its installation.

Manual driver installation procedure

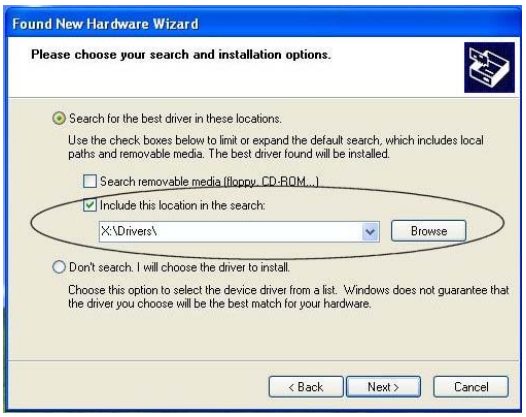
1) Connect the MT2USB board to the PC using a USB cable. *Windows* operating system will detect a new device, showing the message:

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





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



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>. (0x0D in hexadecimal).
Sending other control characters (<*LF*>, <*VT*>, etc.), generates a syntax error. 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 see page 40 of this manual). Moreover, application programs may be developed using standard communication functions provided with.
Communication speed of the virtual serial port can be set at 9600 baud of 19200 baud (see further the *jumper configuration* paragraph at page 17).
The other serial configuration parameters to be used are as following ones:

- Data bits: 8
- Parity bits: none
- Stop bits: 1
- stream control: none/hardware

Through J2 *jumper* (described at page 17.) it is possibile to enable the hardware stream control (CTS/RTS).

In case of DLL choice, please refer to "FTD2XX.DLL DYNAMIC LIBRARY" manual by IPSES, available at the following link http://www.ipses.com/PDF/IPSES-D2XX-it.pdf.

The following commands are implemented:

U	Requests the status (see page 15 for a detailed description of the card status).
UU	Requests the global status of the unit (for further details see page 15). This command provides more detailed information than the command "U".
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 and they must be included between -999.999.999 and+999.999.
Xa	Moves the X axis to an a position (absolute position in half-steps). a value must be included between -999.999.999 and+999.999.
Yb	Moves the Y axis to an G position (absolute position in half-steps). b value must be included between -999.999.999 and+999.999.
Da,b	Moves the axes by a, b movement (relative movements), where a and b are the movement values in half steps. Sia Bothe a and b must be included between -999.999.999 and+999.999
L1	Activates the auxiliary output.
L0	Deactivates the auxiliary output.
H	Moves both axes to the home position (limit detection): during the search of the home position, you can continue to communicate with the card (you can also send the command "K" to end the search of the home position).
HX	Moves X axis to the home position (limit detection): during the search of the home position, you can continue to communicate with the card (you can also send the command "K" to end the search of the home position).
HY	Moves Y axis to the home position (limit detection): during the search of the home position, you can continue to communicate with the card (you can also send the command "K" to end the search of the home position).
K	Stops the current motion on both axes. This command can also be used during the search of the home position.
KX	Stops the movement of the X axis (except during home position). This command can also be used during the search of the home position.
KY	Stops the movement of the Y axis (except during home position). This command can also be used during the search of the home position.
GX,n	Activate the perpetual motion for the X axis, with n greater than or equal to 0 starts the forward motion, with n less than 0 the backward motion. The parameter n cannot be omitted.
GY,n	Activate the perpetual motion for the Y axis, with n greater than or equal to 0 starts the forward motion, with n less than 0 the backward motion. The parameter n cannot be omitted.

CX,n	Mode of operation of the braking action for the X axis. It allows to set the duty cycle in a range between 0% and 50% of the PWM. The value of n must be between 0 and 50 If it is set to zero, the braking action is totally disabled. The default value is 0 (disabled braking action). Use this mode with care, setting PWM values inversely proportional to the time for which you want to leave activated, in order to avoid overheating of the engine or system.
CY,n	Mode of operation of the braking action for the Y axis. It allows to set the duty cycle in a range between 0% and 50% of the PWM. The value of n must be between 0 and 50 If it is set to zero, the braking action is totally disabled. The default value is 0 (disabled braking action). Use this mode with care, setting PWM values inversely proportional to the time for which you want to leave activated, in order to avoid overheating of the engine or system.
CX?	Requires the set duty cycle for the X-axis motor. The return value is a percentage.
CY?	Requires the set duty cycle for the Y-axis motor. The return value is a percentage.
FX,n	Sets the actual absolute position on the X axis, with the motor off (otherwise an error of illegal command is generated). The parameter n can be set between -999 999 999 and +999,999,999.
FY,n	Sets the actual absolute position on the Y axis, with the motor off (otherwise an error of illegal command is generated). The parameter n can be set between -999 999 999 and +999,999,999.
SX,n	Sets the steady speed for the X axis to n half-steps/s. n value must be included between 5 and 5.000 This command can be executed only when X axis is stopped (no matter if Y motor is moving).
SY,n	Sets the steady speed for the Y axis to n half-steps/s. n value must be included between 5 and 5.000 This command can be executed only when Y axis is stopped (no matter if X motor is moving).
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 half-steps. If the position is unknown, the answer is # character.
E1,n	Sets to n the maximum number of half steps, for both axes, for the first motors movement during the home commands. The n parameter must be included between 10 and 999.999. To set this value separately for each axis (X or Y) see commands I1 and Q1.
E2,n	Sets to n the maximum number of half steps, for both axes, for the forward motor movement after the home position has been reached, so to prevent the reading of false contacts, during the execution of the home command. n can be set between 5 and 32. Its value must be less than the one set with the E3 command, otherwise the setting will not be accepted. To set this value separately for each axis (X or Y) see commands I2 and Q2.
E3,n	Sets to n the maximum number of half-steps, for both axes, for the backward motors movement subsequent the forward displacement, so to prevent the reading of false contacts, during the home commands. n value can be set between 15 and 32.757. Its value must be upper than the one set with the E2 command, otherwise the setting will not be accepted. To set this value separately for each axis (X or Y) see commands I3 and Q3.
E1?	Requires the maximum number of saved half steps for the home position search movement. The response to this command will consist of two numerical values: one referred to X axis and the other Y axis (with this firmware version is possible to set different parameters for the X axis and the Y axis - see commands Ix and Qx).
E2?	Requires the maximum number of saved half steps for the forward movement during the home position search. The response to this command will consist of two numerical values: one referred to X axis and the other Y axis (with this firmware version is possible to set different parameters for the X axis and the Y axis - see commands Ix and Qx).
E3?	Requires the maximum number of saved half steps for the backward movement during the home position search. The response to this command will consist of two numerical values: one referred to X axis and the other Y axis (with this firmware version is possible to set different parameters for the X axis and the Y axis - see commands Ix and Qx).
?	Requires the current firmware version and the serial number of the device. The answer will be an ASCII string like "MT2USB Vx.x - S / N: yyyyyyyy - by IPSES Srl <www.ipses.com> ", where xx is the firmware version of the instrument and yyyyyy is the serial number. The firmware version must be greater than or equal to 2.0 and the S / N greater than 20.1301 million (otherwise do not apply the instructions included in this document: contact IPSES for further information)
M	Stores the speed settings and the working mode currently set in the non-volatile memory.
Z	Requires the voltage at which the driving motor drivers are powered on. The response of the board will show the voltage (in volts) supplied by the user to power the engines, in the format x.y.

T1,n	Sets the automatic search of the home position at power on. The n value can be either 0 or 1. When the value is equal to 1, this enables the search of the home position, if it is 0, it is disabled. By default, the value is set to 0.
T2,n	Sets the polarity of the limit switches. The value of n defines the polarity when the limit switches are active (low or high): when n is set to 0, the system will consider the limit switch actives when the corresponding input is 0V, and when it is set to 1, it will consider the limit switch actives when the corresponding input has a high logic level. The command can be set only when both motors are stopped. The default value is 0.
T3,n	Prevents the movement of the axes beyond the end position (negative movement of the motor). When the value of n is 1, the system stops the motor every time it reaches the end of home position (during a negative movement). When n is 0, instead, the motor will never be stopped even if it reaches its negative limit position. The command can be set only when both motors are stopped. By default, the value of n is 1.
T1?	Requires the status in which T1 was set to see if it was enabled or not the automatic search of home position at power on. When the value is 1, it means that it is enabled, if it is 0, it means it is disabled.
T2?	Requires the status of polarity of the limit switches. The answer can be 0 or 1: when the answer is 0, it means the polarity has a low logic level, when it is 1, it means that it is high.
T3?	Requires the status in which T3was set to see if the movement beyond limit position is enabled or not. The answer can be 0 or 1: when the answer is 0, it means that it is not enabled, when 1 means it is enabled.
AX,n	Mode of operation of the braking action to the X axis. Sets the frequency of the PWM which can be between 80KHz and 200KHz. The n value must be included between 80 and 200: this value corresponds to the frequency in kHz. The default value is 100.
AY,n	Mode of operation of the braking action to the Y axis. Sets the frequency of the PWM which can be between 80KHz and 200KHz. The n value must be included between 80 and 200: this value corresponds to the frequency in kHz. The default value is 100.
AX?	Requires the frequency set for the PWM (in kHz) for the motor on the X axis, when braking action is activated.
AY?	Requires the frequency set for the PWM (in kHz) for the motor on the Y axis, when braking action is activated.
J1?	Requests the status of the baud rate jumper (J1). 1 = jumper not inserted, that means a speed of 9600 baud; 0 = jumper inserted, i.e. a speed of 19.200 baud.
J2?	Requests the status of jumper J2. 1 = jumper not inserted, 0 = jumper inserted. The jumper J2 is reserved.
J3?	Requests the status of jumper J3. 1 = jumper not inserted, 0 = jumper inserted (at the power on all default parameters will be restored).
J4?	Requests the status of jumper J4. 1 = jumper not inserted, 0 = jumper inserted. The jumper J4 is reserved.
I1,n	Sets to n the maximum number of half steps for the X axis during the execution of the home-position search command. The value of n must be included between 10 and +999,999,999. The setting can be made only when the X-axis motor is stopped.
I2,n	Sets to n the maximum number of half steps for the X axis for the forward motor movement after the limit position has been reached, so to prevent the reading of false contacts, during the execution of the home command. Its value must be less than the one set with the I3 command, otherwise the setting will not be accepted.
I3,n	Sets to n the maximum number of half steps for the X axis for the backward motor movement after the limit position has been reached, so to prevent the reading of false contacts, during the execution of the home command. Its value must be upper than the one set with the I2 command, otherwise the setting will not be accepted.
I1?	Requires the maximum number of saved half steps for the home position search movement for the X axis
I2?	Requires the maximum number of saved half steps for the forward movement during the home position search for the X axis.
I3?	Requires the maximum number of saved half steps for the backward movement during the home position search for the X axis.
Q1,n	Sets to n the maximum number of half steps for the Y axis during the execution of the home-position search command. The value of n must be included between 10 and +999,999,999. The setting can be made only when the Y-axis motor is stopped.
Q2,n	I Sets to n the maximum number of half steps for the Y axis for the forward motor movement after the limit position has been reached, so to prevent the reading of false contacts, during the execution of the home command. Its value must be less than the one set with the Q3 command, otherwise the setting will not be accepted.
Q3,n	Sets to n the maximum number of half steps for the Y axis for the backward motor movement after the limit position has been reached, so to prevent the reading of false contacts, during the execution of the home command. Its value must be upper than the one set with the Q2 command, otherwise the setting will not be accepted.
Q1?	Requires the maximum number of saved half steps for the home position search movement for the Y axis

Q2?	Requires the maximum number of saved half steps for the forward movement during the home position search for the Y axis.
Q3?	Requires the maximum number of saved half steps for the backward movement during the home position search for the Y axis.
VX,n	Sets the starting speed of the X-axis motor. The n value must be included between 10 and 5,000. The setting can be made only when the X-axis motor is stationary. If you set n to 5,000, the system will never perform acceleration and deceleration, whatever the operating speed set (command SX, n).
VY,n	Sets the starting speed of the Y-axis motor. The n value must be included between 10 and 5,000. The setting can be made only when the Y-axis motor is stationary. If you set n to 5,000, the system will never perform acceleration and deceleration, whatever the operating speed set (command SY, n).
VX?	Requests the set starting speed of the X-axis motor
VY?	Requests the set starting speed of the Y-axis motor
RX,n	Set in half steps when changing the speed of acceleration and deceleration for the X-axis motor. The n value must be included between 20 and 100. The setting can be made only when the X-axis motor is stationary. The default value is 20
RY,n	Set in half steps when changing the speed of acceleration and deceleration for the Y-axis motor. The n value must be included between 20 and 100. The setting can be made only when the Y -axis motor is stationary. The default value is 20
RX?	Requests the set value in half steps when changing the speed of acceleration and deceleration for the X-axis motor
RY?	Requests the set value in half steps when changing the speed of acceleration and deceleration for the Y-axis motor

For all commands using of the X, Y parameters, it is also possible, alternatively, make use respectively of 1 for the X axis, and 2 for the Y axis. For example, the command GX, 1 is equivalent to the command G1, 1.

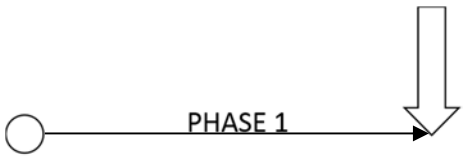
For each query command sent (u?, Cx?, Etc..) the device will respond with a string containing the desired information, followed by the <CR> and <LF> control characters (13 and 10 in decimal notation).

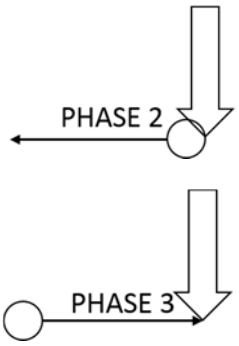
When the home command is executed (see the H, Hx and Hy commands), the motor moves backward searching the position notified by the activation of the limit switch signal for a maximum number of steps set through the command E1, n.
If during this phase the home position is not detected, the system generates a search error of the home position. Otherwise, if the home position is identified, the motor will move forward for the number of steps set with the E2 command, and, subsequently, it will reverse the direction of motion again in search of the home position, for a maximum number of steps set with the E3 command.
By this way, if the home position is read for a second time, the motor will stop there.

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 is therefore necessary that the parameter n of the E2 command is set greater than zero and with a value that, at the end of this movement, the mechanics of your system will not activate the limit home position detection.
These operations prevent the reading of "fake contacts" of the home position and guarantee of actually being in the home position: during the execution of this command is not possible to send other ones (including the status request and the stop command).
It is by the user to set a n value for the E3 command greater than or equal to the n value set for E2, in order to avoid the non- detection of the home position during the last stage of execution of the home command: in case of E3 n value lower than E2 n value an error of home position search will be alleged.

All the positions and movements are to be understood measured in half steps (the speed of the motor, for example, is measured in half steps per second).

In the picture are the 3 phases are shown:





The U command of status request causes a hexadecimal number that is transmitted on two-digit code representing the status byte of the unit according to the following convention:

bit 7	error
bit 6	known Y axis position
bit 5	known X axis position
bit 4	auxiliary output active
bit 3	Y axis limit position reached
bit 2	X axis linit position reached
bit 1	Axes running
bit 0	ready (Known position after an home position)

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.

It is also available the UU extendend status request command which causes to be transmitted a three-digit hexadecimal number that represents the status byte of the unit according to the following convention:

bit 11	Internal use, do not consider
bit 10	Internal use, do not consider
bit 9	Y Axis running
bit 8	X Axis running
bit 7	Error
bit 6	Known Y-axis position
bit 5	Known X-axis position
bit 4	auxiliary output active
bit 3	Y axis limit position reached
bit 2	X axis limit position reached
bit 1	Y-axis ready (Known position after an home position).
bit 0	X-axis ready (Known position after an home position).

If at the status request (using either the U command and the UU command) an error high bit is returned (i.e., if the board responds with code similar to 080), a three-digit hexadecimal number is concatenated (for example, 080,002) according to the convention listed below. Even more error codes can be active.

The possible error codes for the U command are:

bit 7	Reached the Y axis limit position during backward movement when negative run is disabled with the appropriate jumper.

bit 6	Reached the X axis limit 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.

The possible error codes for the UU command are:

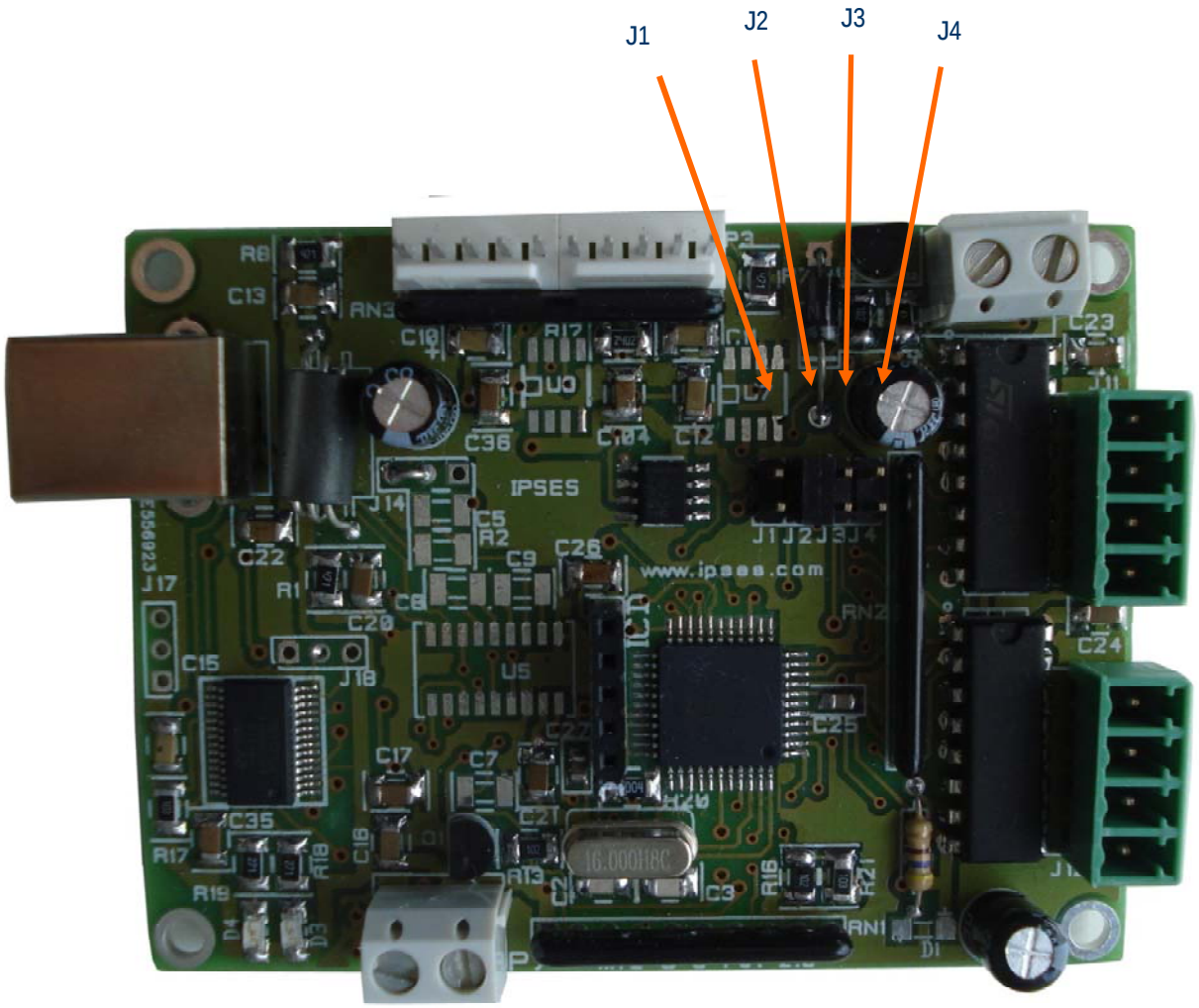
bit 9	Internal use, do not consider
bit 8	Internal use, do not consider
bit 7	Reached the Y axis limit position during backward movement when negative run is disabled with the appropriate parameter (seeT3 command).
bit 6	Reached the X axis limit position during backward movement when negative run is disabled with the appropriate parameter (seeT3 command).
bit 5	Invalid checksum of the stored data in non-volatile memory.
bit 4	Invalid parameters of the stored data in non-volatile memory.
bit 3	Time out or error during home position search.
bit 2	Out of range parameter (i.e. to set a speed that does not fall within the stated 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.

Reset all errors occurs after each status request.

MT2USB CONFIGURATION INSTRUCTIONS BY JUMPERS

Through the jumpers on the MT2USB card (see picture 2 for their nomenclature and position), you can program the following functions:

- *jumper* J1: allows selection of the communication speed of the Virtual Serial Port: no jumper will set a speed of 9600 baud, with jumper the speed will set at 19.200 baud. The status of the jumper is read immediately after the connection of the USB port and baud rate set accordingly: any subsequent change in the status of the jumper will not take effect until the USB will not be physically disconnected and reconnected.
- *jumper* J2: if inserted enable hardware flow control of the serial.
- *jumper* J3: ripristino della configurazione di *default*. All'accensione della scheda, se il *jumper* è inserito, verranno caricati tutti i valori di *default* che possono essere modificati dall'utente (ad esempio tutti i parametri modificabili con i comandi "Tx"). restores the default configuration parameters. At the board switching on, if the jumper is inserted, all modifiable values are restored to default parameters (for example, all parameters changed with the commands "Tx").
- *jumper* J4: reserved. Do not insert any jumper for the normal operation of the board.



Picture 2: *jumper* of MT2USB board.

You can read the status of the jumper using the following commands:

J1?	Requests the status of the baud rate jumper (J1). 1 = jumper not inserted, that means a speed of 9600 baud; 0 = jumper inserted, i.e. a speed of 19.200 baud.
J2?	Requests the status of jumper J2. 1 = jumper not inserted, 0 = jumper inserted. The jumper J2 is reserved.
J3?	Requests the status of jumper J3. 1 = jumper not inserted, 0 = jumper inserted (at the power on all default parameters will be restored).
J4?	Requests the status of jumper J4. 1 = jumper not inserted, 0 = jumper inserted. The jumper J4 is reserved.

The function of the jumper allowing you to select the search of the home position at power on, can be set using the T1 command.

The function of the jumper allowing you to select the polarity of the limit switches can be set using the T2 command.

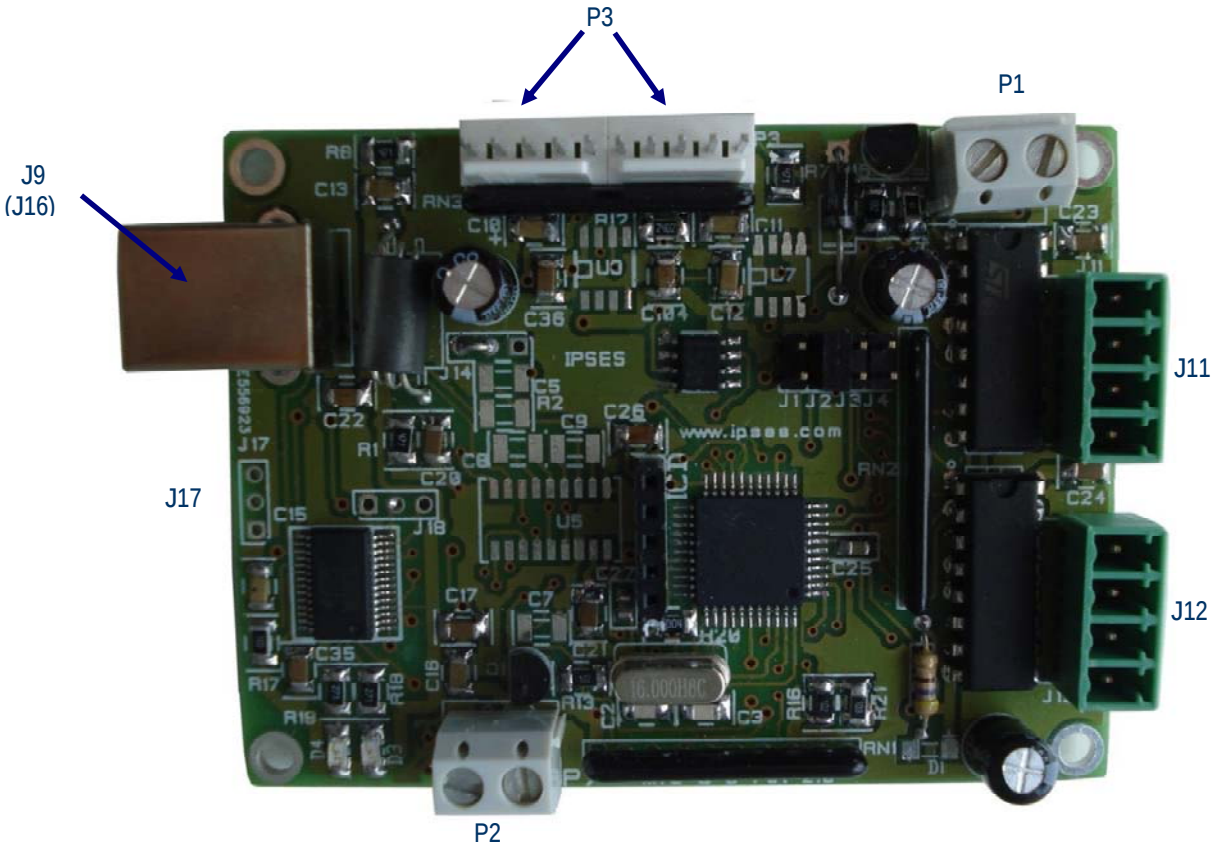
The function of the jumper allowing to enable the stop of the motor when limit position is reached (on negative movements) can be set using the T3 command.

For details on these commands, see more on p. 12.

MT2USBMS CONNECTION INSTRUCTIONS

The board is self-powered through the USB connector. It is however necessary to provide the system with the power to drive the motors (using the P1 connector, visible in Picture 3) which must be between 5 and 36V.

It is also connect to the device a limit switch for each motor (P3 connectors).
Besides, it is possible to use the open collector auxiliary output (P2 connector) to communicate with a PLC or to activate other control electronics.



Picture 3: MT2USB board connectors.

Following all the connections of the card are listed:

J9	type B USB connector to interface to the PC
P3	Limit switch connectors (one for each motor)

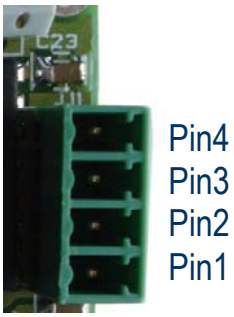
Pin1: positive output supply (5Vdc, without current limitation) to power any external detection logic of the limit switch of the Y axis.

Pin2: positive output supply for infrared LED for Y axis (for optical limit switch).
Pin3: input for the limit switch of the Y axis.
Pin4: GND.
Pin5: negative output supply for infrared LED for Y axis (for optical limit switch).
Pin6: positive output supply (5Vdc, without current limitation) to power any external detection logic of the limit switch of the X axis.
Pin7: positive output supply for infrared LED for X axis (for optical limit switch).
Pin8: input for the limit switch of the X axis.
Pin9: GND.
Pin10: negative output supply for infrared LED for X axis (for optical limit switch).

P1 Power supply
Pin1 (+): positive voltage.
Pin2: GND.

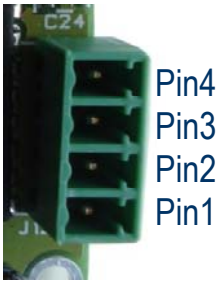
P2 Aux out (see at page 21 the circuit diagram is shown)
Pin1: positive voltage (limited by a resistor).
Pin2: *open-collector* toward GND.

J11 Motor X axis



Pin1: phase A+.
Pin2: phase B+.
Pin3: phase B-.
Pin4: phase A-.

J12 Motor Y axis



Pin1: phase A+.
Pin2: phase B+.
Pin3: phase B-.
Pin4: phase A-.

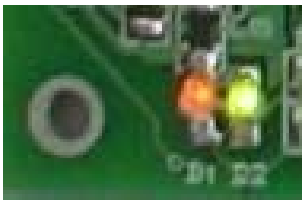
To interface the motors with J11 and J12 use connectors type Phoenix Contact with a step of 3.81 mm [2].

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

WARNING!
Do not connect or disconnect the motor (or any stage of it) when the board is powered!
The +5 Vdc voltage present on pin 1 and pin6 of the limit switch connector is obtained directly by the supply of the PC. If overloaded or short-circuited it could therefore damage the PC.

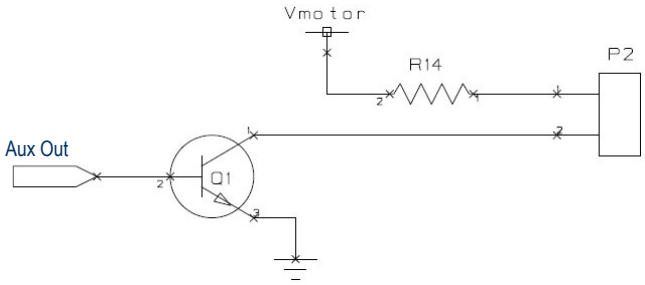
It is recommended to connect the USB cable to the device only after to the driving stage motors have been supplied.

Two LEDs, red and green, show, respectively, the exchange of data between the instrument and the PC and the successful connection of the instrument to the PC.



Picture 4: LED showing data exchange and made connection.

The board is also equipped with an auxiliary open-collector output (provided with a pull-up resistor connected directly to the motor supply) that can be controlled directly by the user ("L" command described in the table above). The circuit of the output stage is shown below:

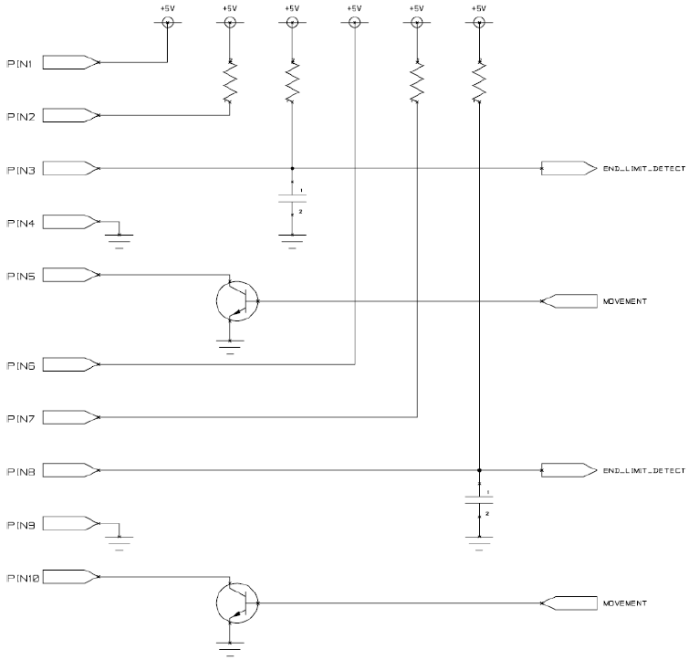


Picture 5: circuit of the output stage.

The value of R14 is equal to 470Ω (1/4 W).

LIMIT SWITCH CONNECTION EXAMPLES

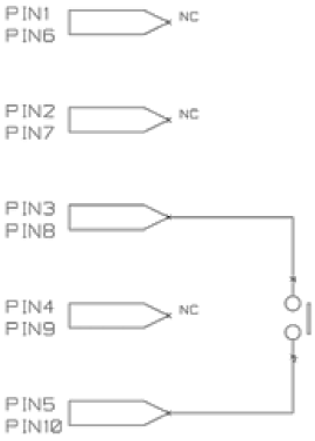
Following Picture 6 shows the diagram of implementation of the P3 connector of the MT2USB board for the management of the limit switch signals.



WARNING!
The maximum input voltage to pin3 (pin8) must not exceed +5 V.

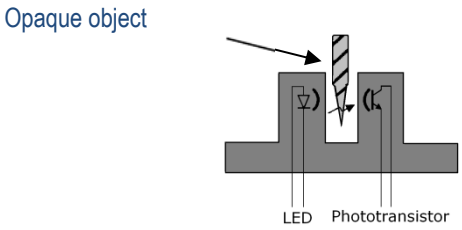
Picture 6: implementation of J2 connector.

From this picture it is also clear that when one motor is running, the MT2USB device interprets the reaching of the home position when the pin 3 (for movement along the Y axis), or pin8 (for movement along the X axis) of the connector P3 undergo a change of potential. The T2n command defines the polarity of the signals: when n is set to 0, the signal of limit switch is active when GND is applied; when n is set to 1, however, the activation of the limit switch occurs when +5 V are applied.
In case you employ mechanical limit switches, for normally open (NO) limit switches, n must be set to 0; for normally closed (NC) limit switches n must instead be set to 1.



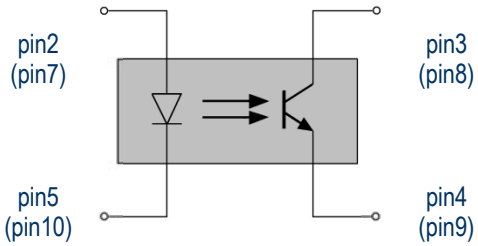
Picture 7: commutation for mechanical limit switches

The signal of the limit switch can be obtained with the use of optical limit switches. The following Picture 8 shows how it works. The LED generates a light beam that illuminates an element, such as the phototransistor, able to alter
, when irradiated, its electrical properties. The phototransistor provides a conductive path, the interruption of the light beam for the interposition of an opaque object and stops electric conduction.



Picture 8: optical limit switch.

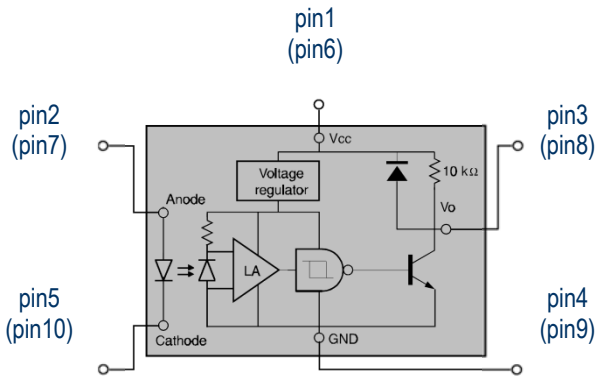
Picture 9 shows the connections to be made with the P3 connector if you opt for the adoption of an optical switch with phototransistor output.



Picture 9: Optical limit switch with phototransistor output; connection with the P3 connector pins for signal detection limit along the Y axis (X axis) is shown.

For the use of an optical sensor with open collector phototransistor output, user must configure the board by issuing the T2command with the value set to 1.

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. In Picture 10 it is shown the relevant diagram: it shows an optical sensor equipped with powered logic for the detection of the interruption of the beam for the interposition of an opaque object. The power supply of the logic is provided via pin1, for the Y axis, and via pin6, for the X axis



Picture 10: Optical sensor with powered logic to detect the presence of an opaque object; the connection with the connector pins P3 for signal detection limit along the Y axis (X axis) is shown.

To use this kind of sensor it is necessary to configure the board by issuing the T2 command with the value set to 0.

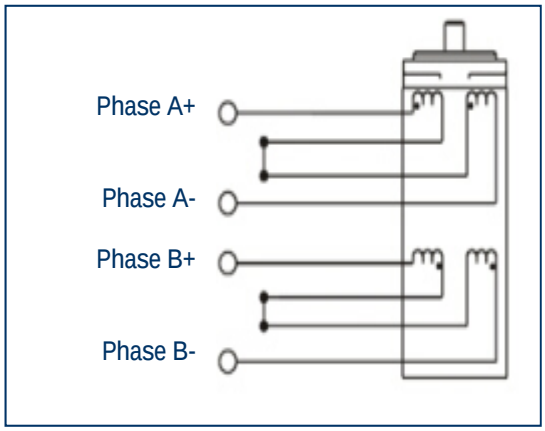
WARNING!
The maximum voltage of the input signals to pin3 and pin8 should never exceed the +5 Vdc.
Voltages higher cause irreparable damage to the components on the board.

MOTORS CONNECTION

The MT2USB card can control both two bipolar and two unipolar stepper motors, i.e. 8, 4 and 6 lead center tapped motors. Here the possible different motor connections are showed.

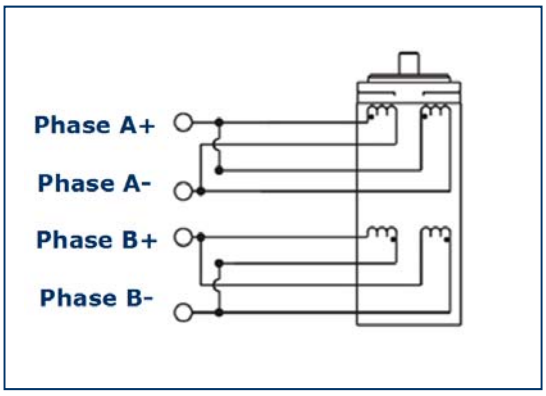
MOTOR CONNECTION (8 LEAD MOTORS)

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.

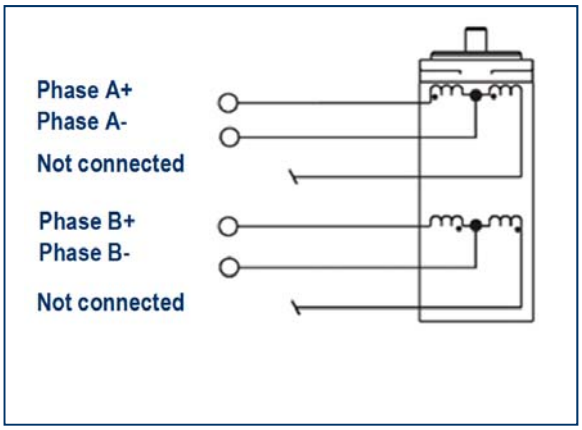


Picture 12: parallel connection.

MOTOR CONNECTION (6 LEAD MOTORS)

Half coil configuration

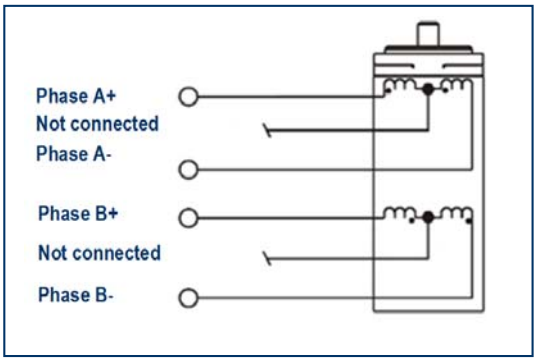
This configuration uses 50% of the motor phase windings. This gives lower inductance, hence, lower torque output. Like the parallel connection of 8 lead motor, the torque output will be more stable at higher speeds. This configuration is also referred to as half copper.



Picture 13: half coil configuration

Full coil configuration

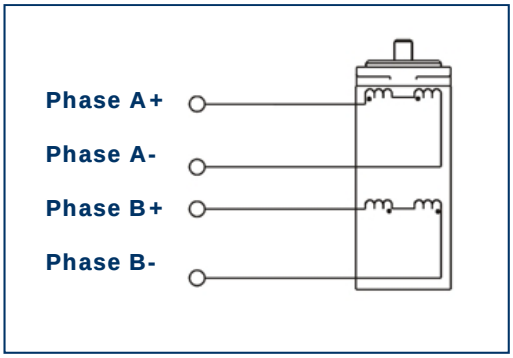
The full coil configuration on a 6 lead motor should be used in applications where higher torque at lower speeds is desired. This configuration is also referred to as full copper.



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

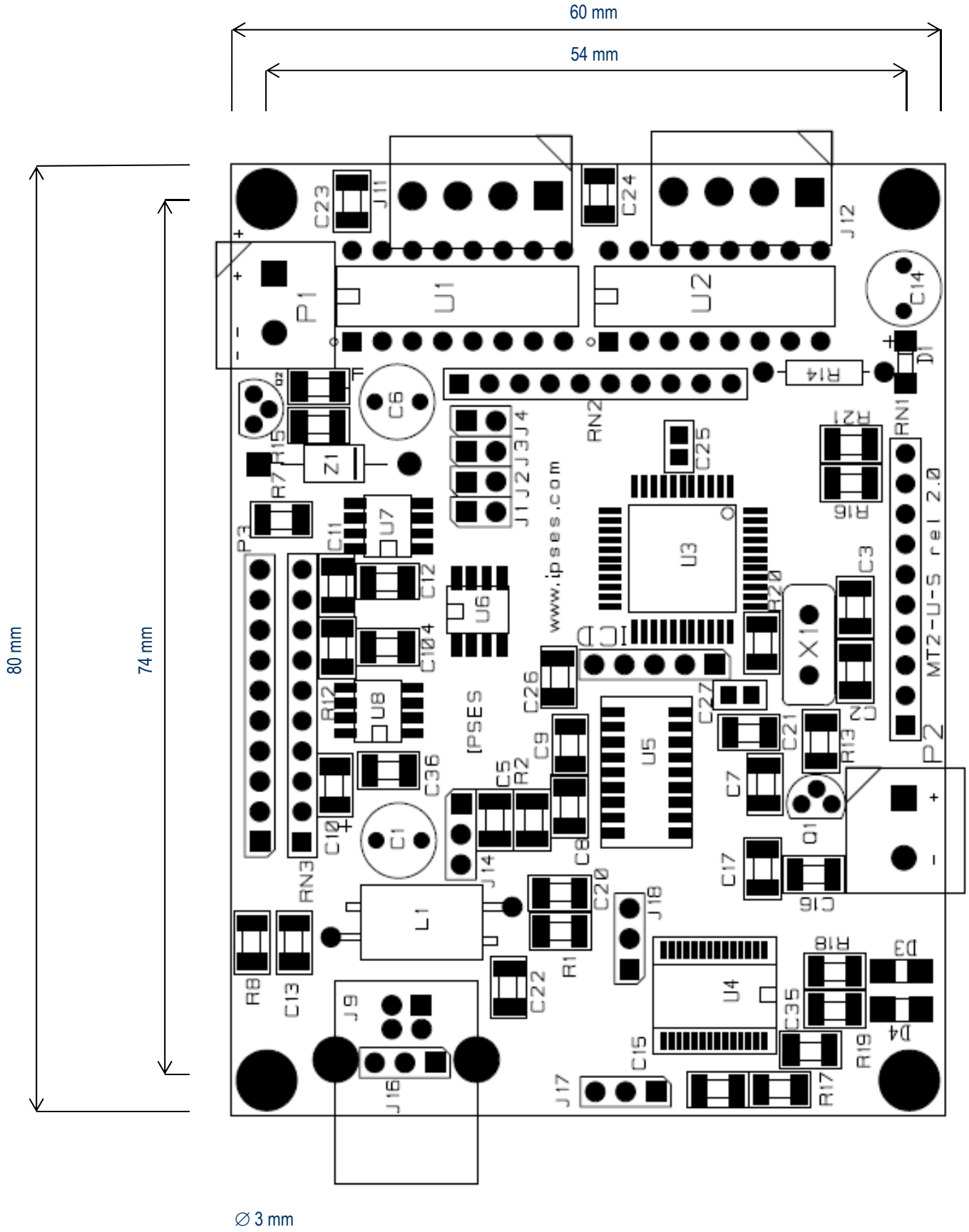
The physical direction of motors depends on connection of motor windings. To switch the direction of the motor, switch the wires on either phase A or phase B outputs.

TECHNICAL FEATURES

Power supply:	The logic is self-supplied directly from the computer USB port
Motor power supply:	from 5 up to 36Vdc
Output current:	max 0,6A/phase (1,2A of peak)
Interface:	USB 2.0 (B connector type)
Motor movement:	half-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.

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 info@ipses.com



Dimensions of MT2USB board

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 info@ipses.com

BOX

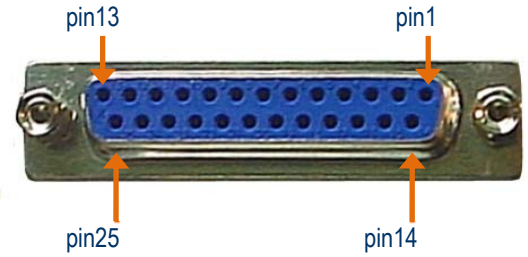
On demand, all MT2 motor control cards are available on box with power pack included. Picture 18 shows MT2USBMS-BOX; analogue solution is realized for MT2USB-BOX.

a) b)

Picture 16: a) front and b) rear panel of MT2USB 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 MT2USB-BOX.

DEMO SOFTWARE

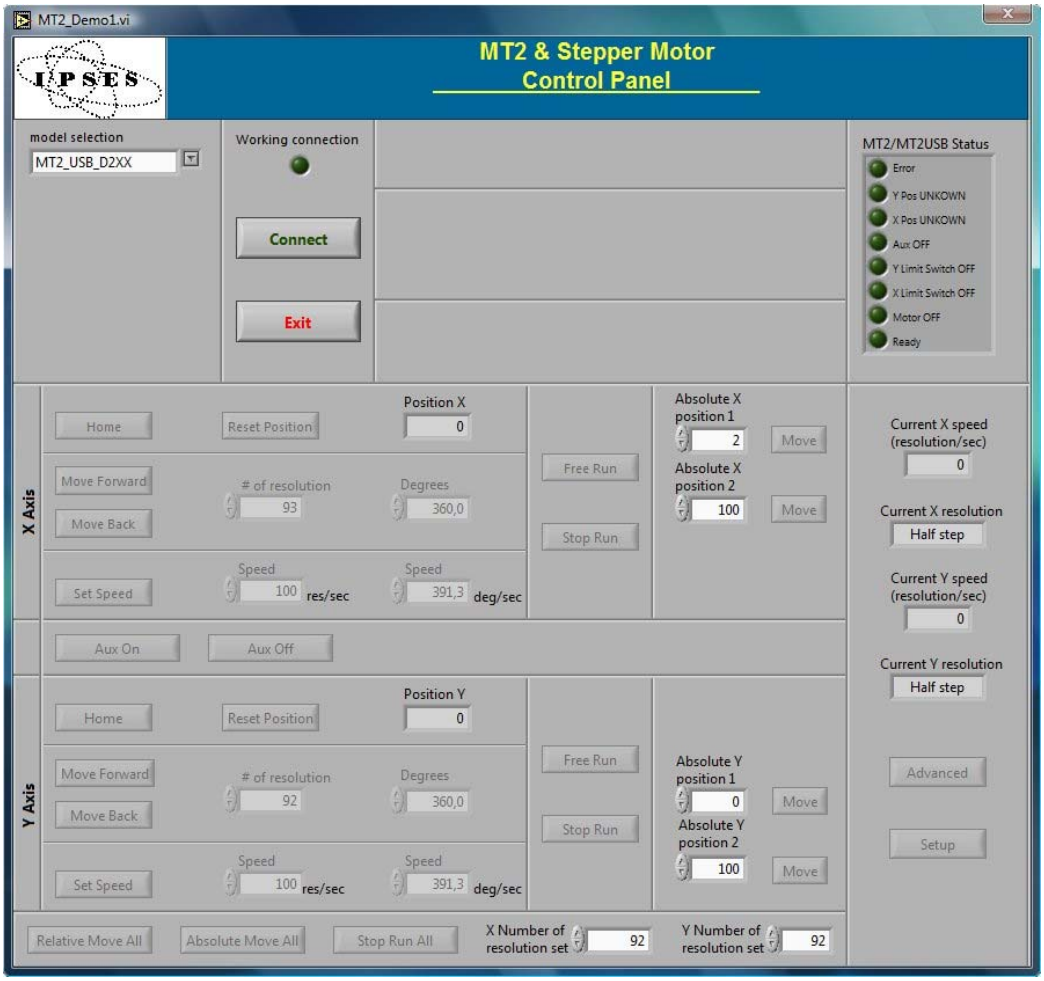
MT2_Demo is a demo software which allows MT2 device control testing from a PC. Displayed virtual control panel has intuitive functionalities which make you easy understand how it works.

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

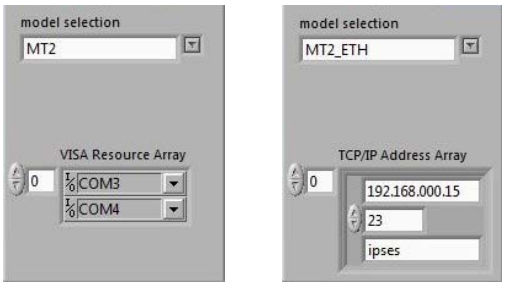
Execute "MT2_Demo.exe". Virtual control panel is displayed as showed in Picture 17:



Picture 17: virtual control panel.

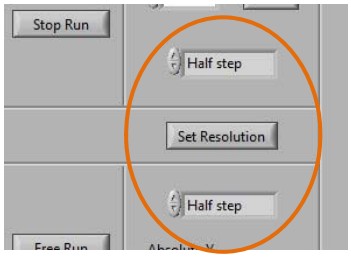
CONNECTION TO MT2 DEVICE

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.



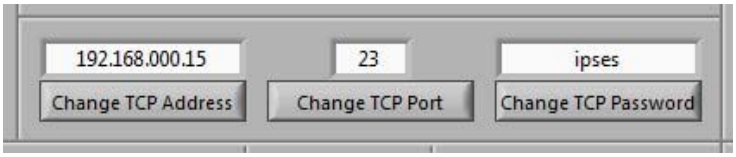
Picture 18: a) selection parameters fields for serial protocol communication (VCP) and b) selection parameters field for Ethernet protocol communication

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 22. 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



Picture 19: 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.



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

FUNCTIONALITIES

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

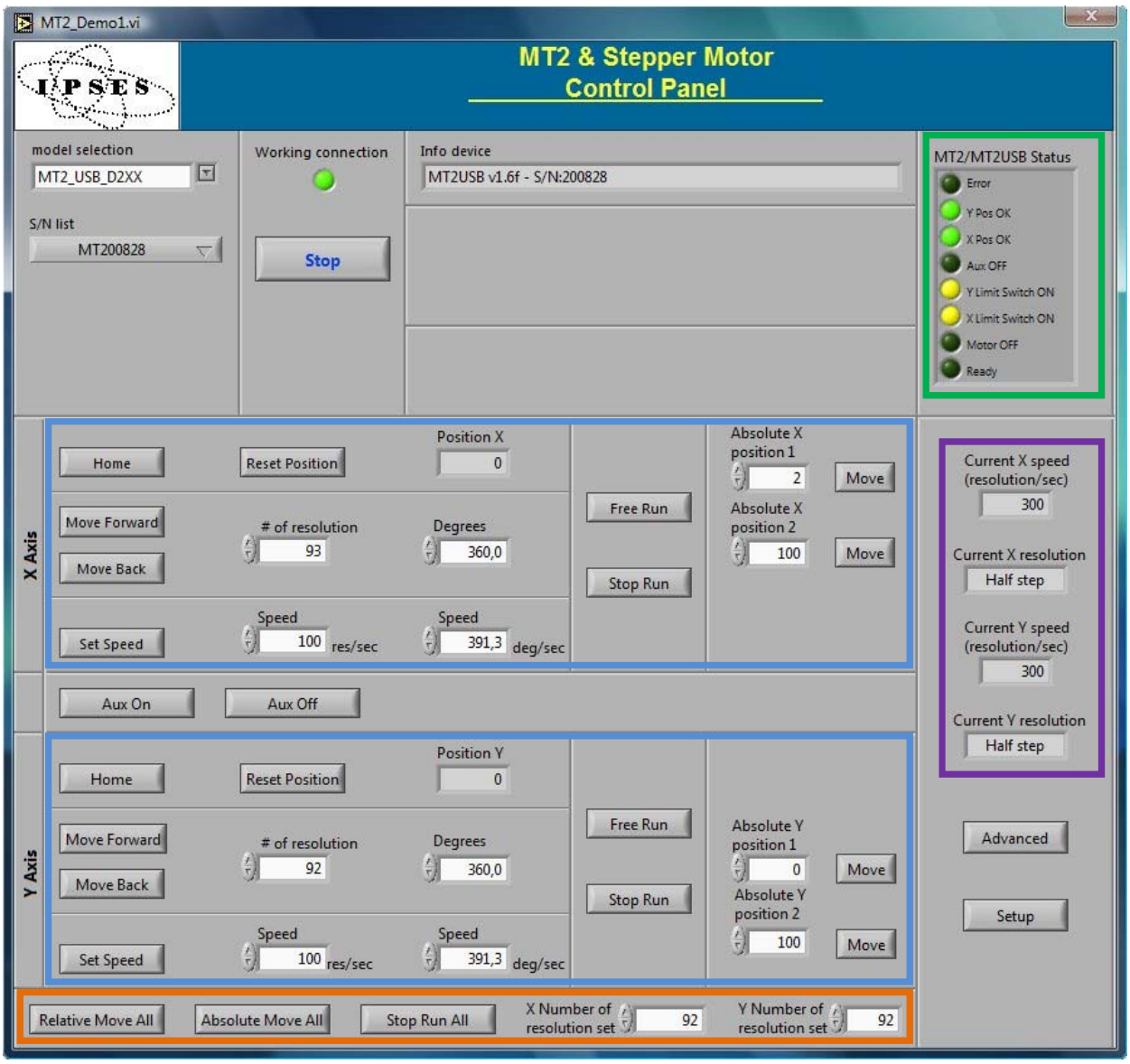


Figura 21: active panel

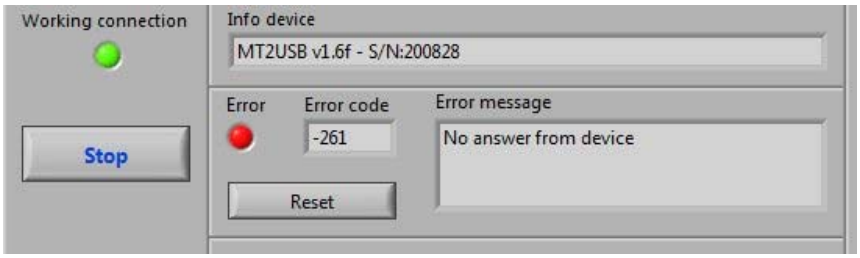
The blue boarded area in Picture 21 includes controls and commands for setting and for the movements of both X and Y axes respectively.
Consider controls and commands for the X axis.
Home command moves X axis motor to home position: during the execution of a home command, for serial and USB versions, 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 22, where is represented an error occurred after a failed communication to the device.

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.

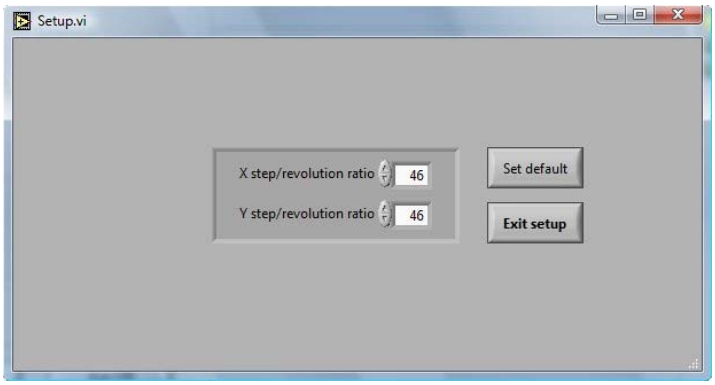
The software errors are generated as shown in Picture 22, where is represented an error occurred after a failed communication to the device.



Picture 22: error message

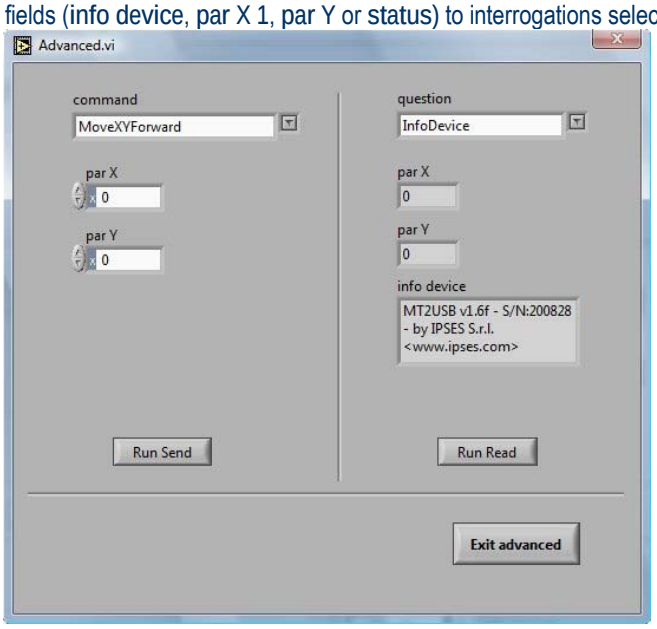
The re-enabling of the application functionalities is suborder to the Reset condition.

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.



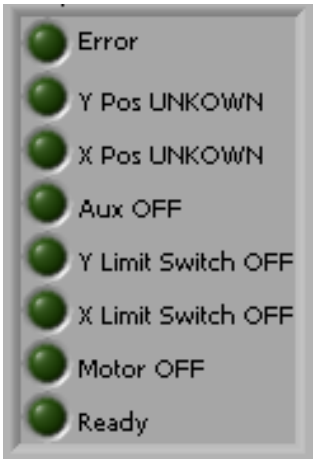
Picture 23: setup window

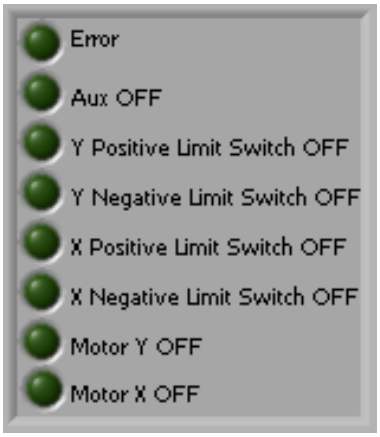
Advanced runs homonym subroutine, as shown in Picture 24, where user can see a low level dialogue to the active device. Chosen instruction in command and associated parameter par X and par Y are communicated to the device when Run Send is set. Run read allows answer in the appropriate fields (info device,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



Picture 24: Advanced subroutine window

The surrounded green area on front panel (see Picture 21) 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 25 shows these differences.





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

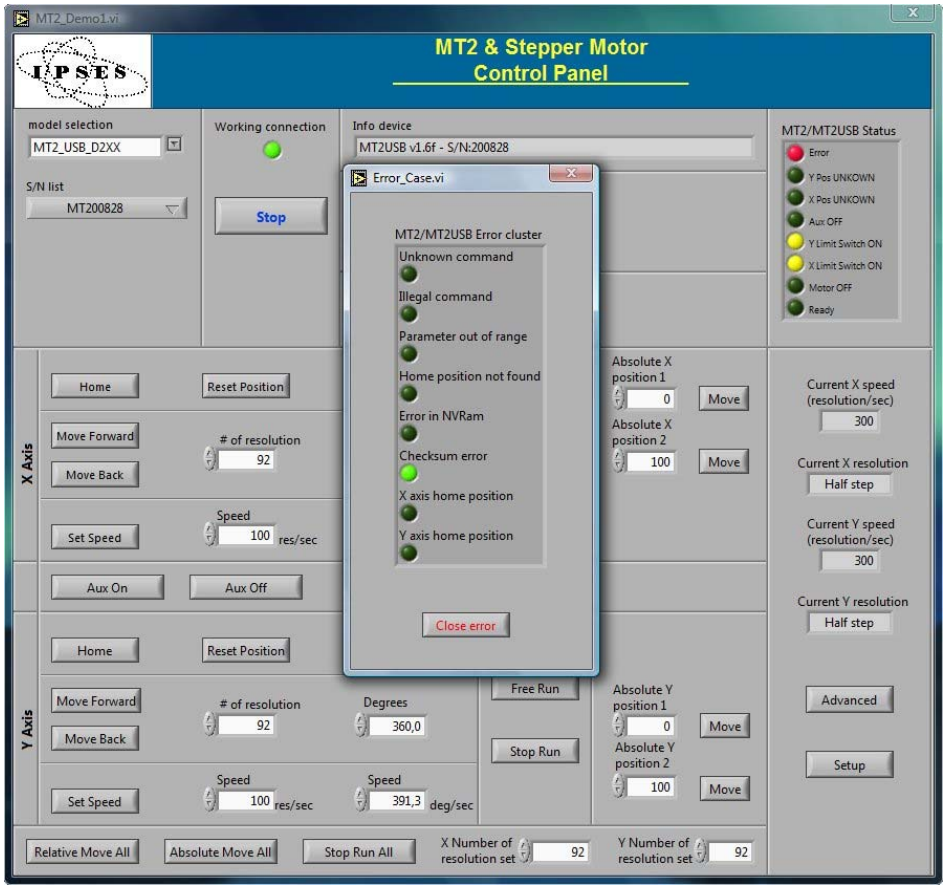
a)

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.

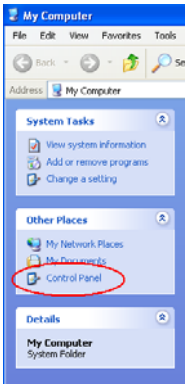
Picture 26 shows the notification of a device error message. To re-enable the operative conditions press Close error.



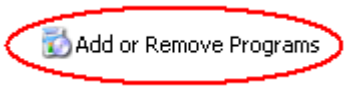
Picture 26: device error message example

REMOVAL

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

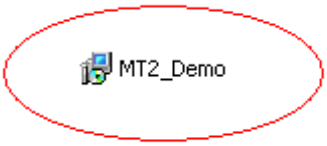


1) From Desktop, click “My Computer” icon and choose “Control Panel”.



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.

	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.

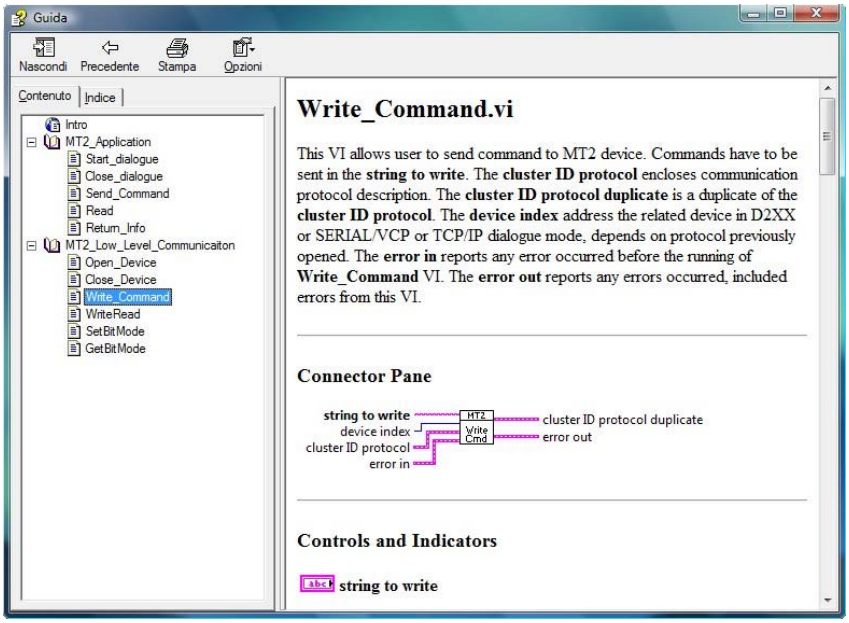
MT2_Library is provided with a help file, *MT2_Help.chm*.

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.

.

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 info@ipses.com

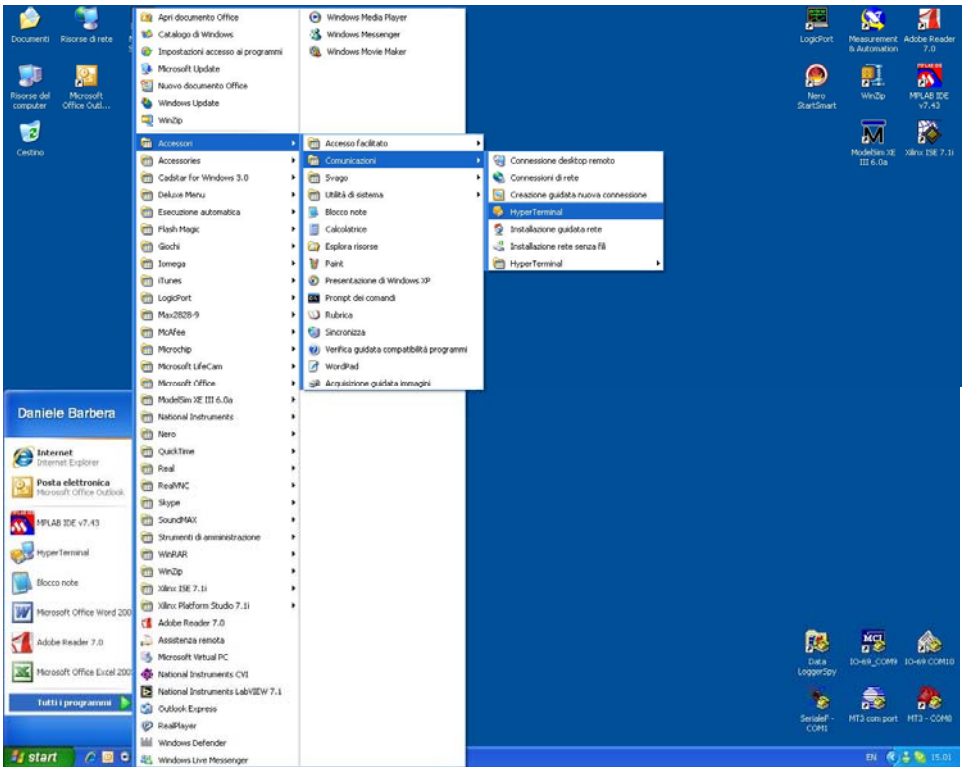


Picture 27: *LabView* functions help

MT2_Library is available on demand.

CONNECTION EXAMPLE: HYPERTERMINAL

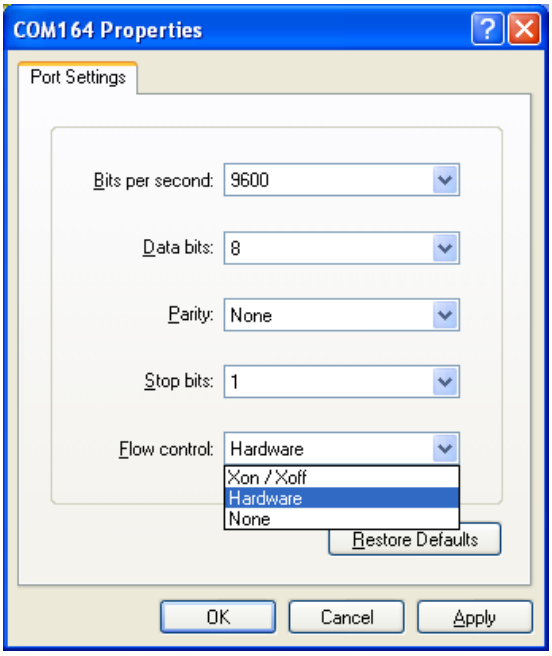
A typical example of serial connection is represented by HyperTerminal tool, which is present in all operating systems Microsoft Windows (except Windows Vista where the program is not included in the operative system distributions, but it is easily downloadable from internet), running by menu “Start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



Picture28: Hyper Terminal path

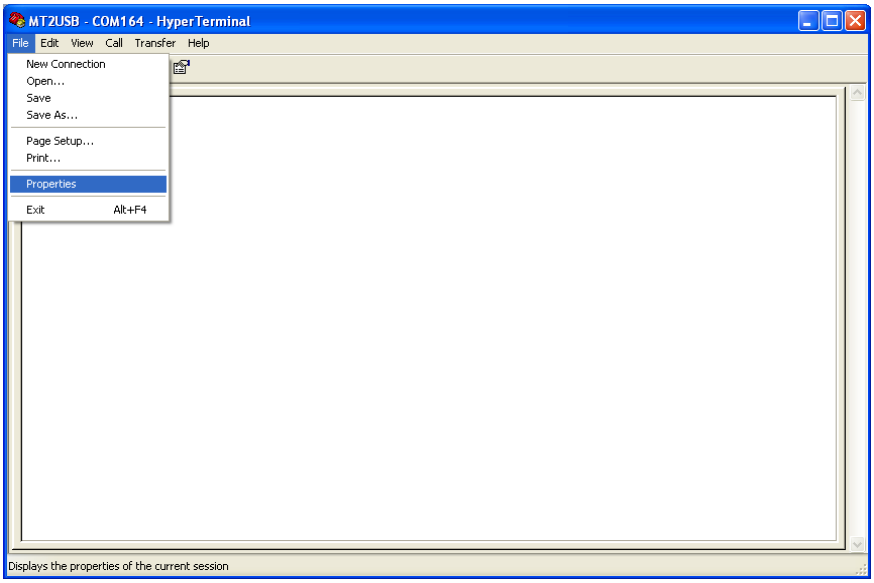
The serial communication can be used also with USB devices because the driver provides a virtual serial communication port (VCP) for each MT2 system connected. Through the VCP it can be easily established the serial communication with the board. Once the connection is established, a name and some other parameters are needed (as it can be seen in Picture 28 and Picture 30):

Picture 29: HyperTerminal general parameter settings

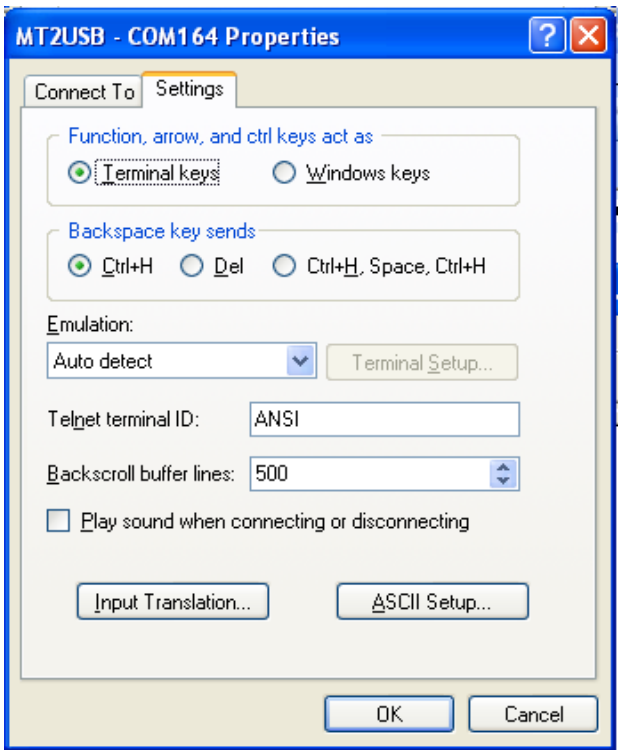


Picture30: Hyper Terminal serial parameter settings

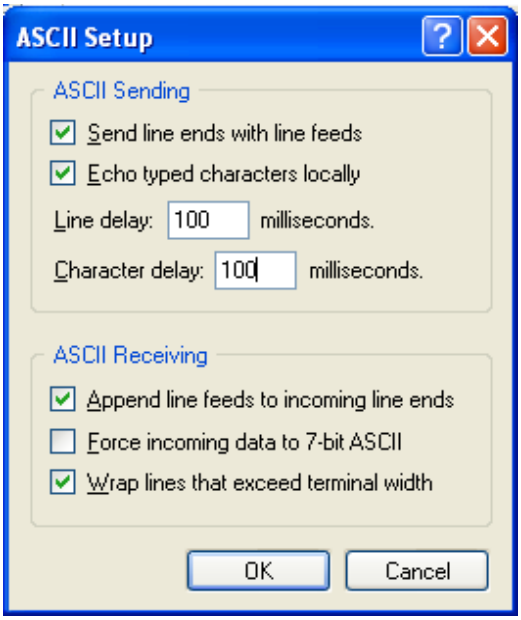
Before getting information by the board it is necessary to view characters sent by user. To do that, an echo late on what is written has to be set via program properties menu (Picture 31 - Picture 32 -Picture 33).



Picture 31: HyperTerminal properties menu

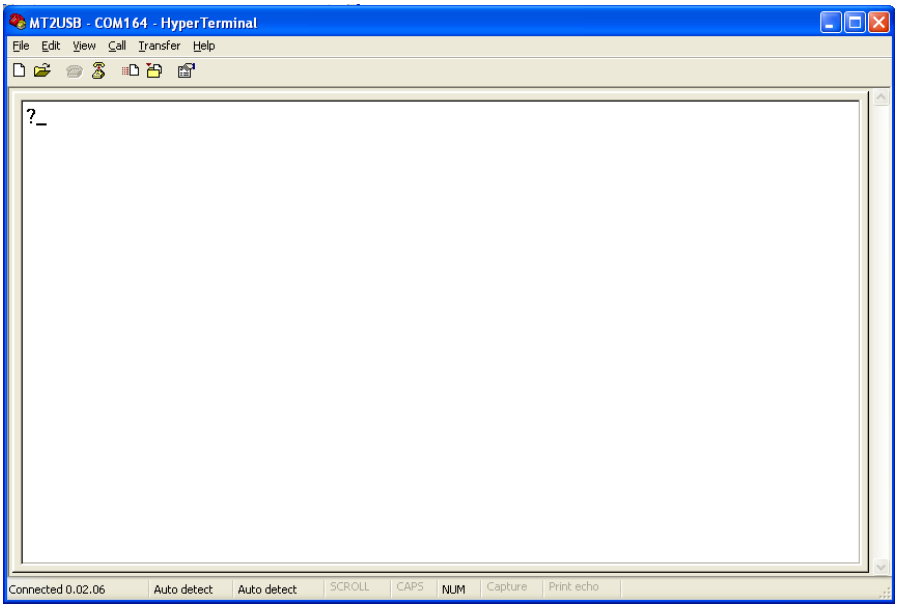


Picture 32: by TAB "Settings", user can select "ASCII Setup"



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 35:



Picture 34: get firmware version command

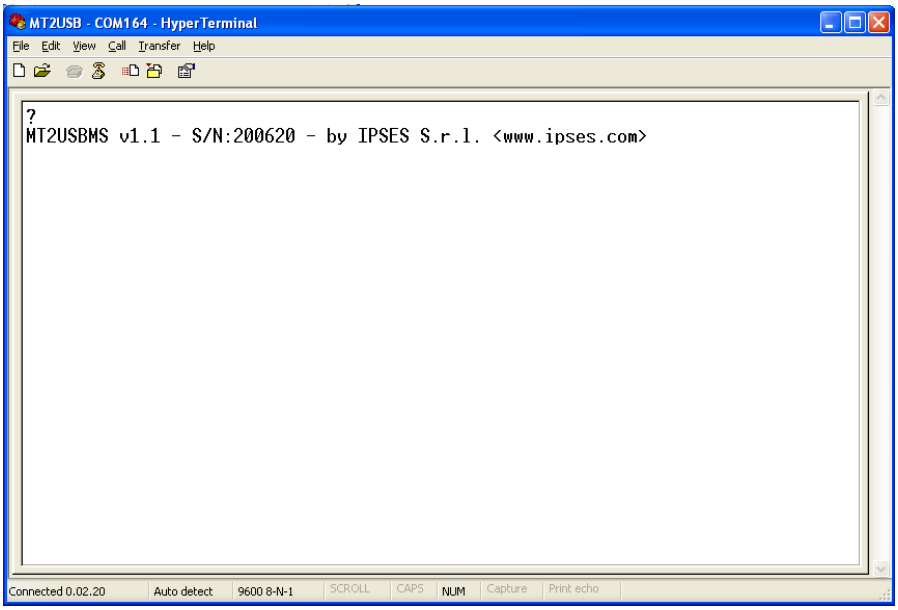
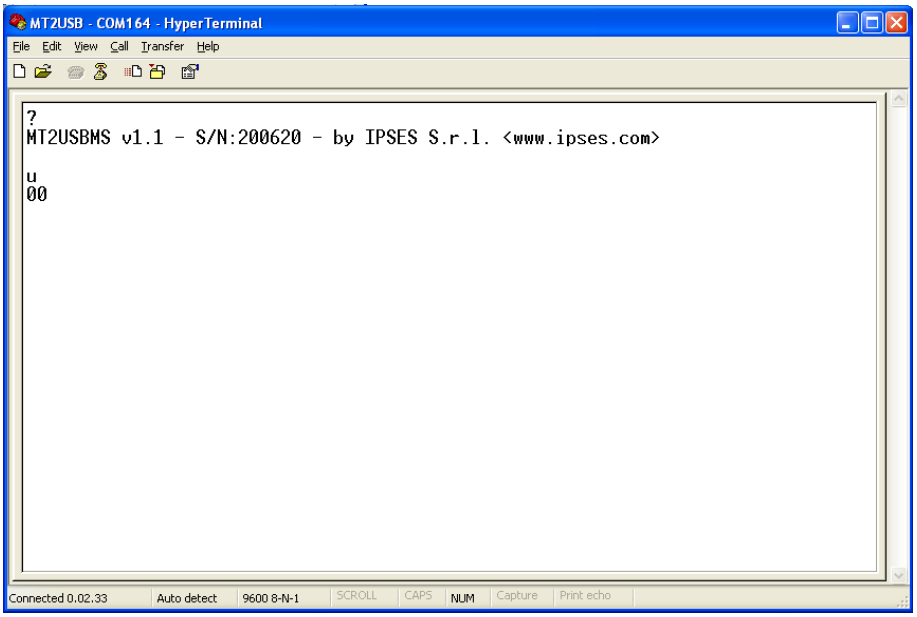


Figura 35: firmware version string example

Sending other kind of commands in succession, it is possible to communicate and to operate with MT2USBvia its COM port 164 (Picture 36 the get status function after the get firmware version can be seen).



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);
```

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 info@ipses.com

PRODUCT CODES

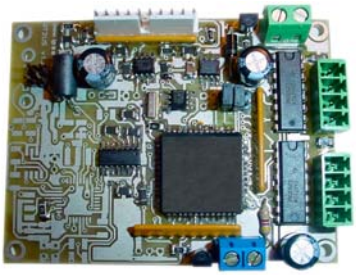
Code	Description
MT2USB	Two axes stepper motor control card with USB interface
MT2USB-box	Two axes control system with MT2USB card and power supply integrated in a case. Power supply at 230V
MT2Library	LabVIEW 7.1 (and further version) library for MT2 card series
MC-connect	Bundle of 2 MC male connectors (for motors) + 2 AMP connectors (for limit home detectors)
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

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 info@ipses.com

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.*

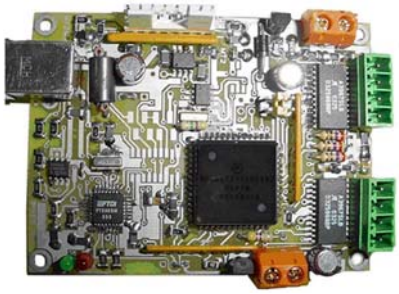


MT2

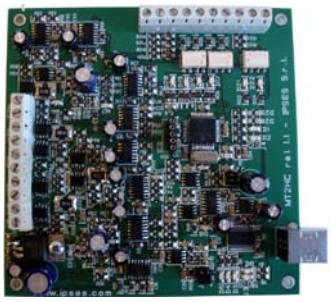
MT2MS

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.

v

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

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