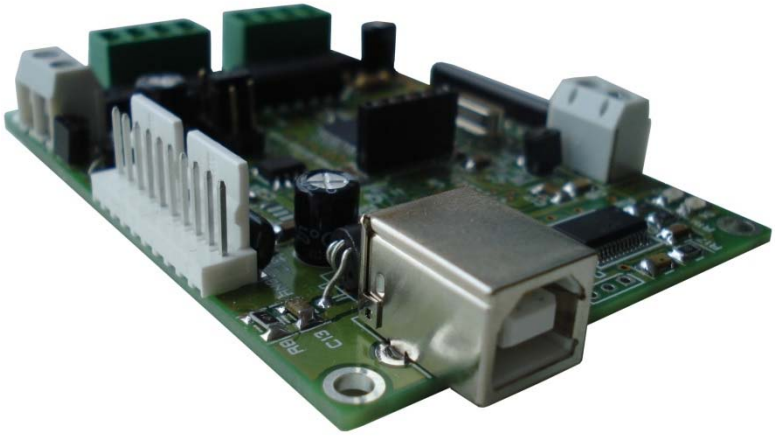
UNITÀ DI CONTROLLO ASSI
MT2USB

Rel. 01.01.0002
(Codice Prodotto: MT2USB)



Le informazioni contenute nel presente manuale sono proprietà di IPSES S.r.l. e devono essere considerate e trattate come confidenziali.
La presente pubblicazione può essere riprodotta, trasmessa, trascritta o tradotta in qualsiasi linguaggio umano o elettronico solamente dopo avere ottenuto l'autorizzazione scritta di IPSES S.r.l.

Le informazioni contenute nel presente manuale sono state accuratamente verificate e sono considerate valide alla data di pubblicazione del presente documento; IPSES non si assume comunque alcuna responsabilità di eventuali inaccuratezze. IPSES non potrà essere ritenuta responsabile per qualsiasi danno conseguente o incidentale accaduto dall'aver considerato accurata la presente documentazione.

Le informazioni contenute nel presente manuale possono subire variazioni senza preavviso e non rappresentano un impegno da parte di IPSES. Il progetto di questa apparecchiatura subisce continui sviluppi e miglioramenti. Di conseguenza, l'apparecchiatura associata al presente manuale potrebbe contenere piccole differenze di dettaglio rispetto alle informazioni fornite nel presente manuale.

Il presente manuale in italiano è la versione originale.

Stampato in Italia

Copyright © 2013-2016 IPSES S.r.l.

Tutti i diritti riservati.

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

GARANZIA

Salvo non sia diversamente stabilito, IPSES garantisce che i Prodotti contraddistinti dal suo marchio, acquistati direttamente da IPSES o da un suo rivenditore autorizzato, saranno esenti da difetti per 12 mesi dalla consegna. Nel caso di difetti del prodotto entro il periodo indicato, IPSES, a sua scelta, riparerà o sostituirà il Prodotto a proprie spese[1] in tempi ragionevoli. Sarà adottato ogni ragionevole sforzo, al fine di risolvere il problema in termini realistici, a seconda delle circostanze. IPSES interviene e ripara usando componenti nuovi o componenti equivalenti a nuovi, in conformità agli standard e alla pratica industriale.

Esclusione dalla garanzia:

IPSES non rilascia alcuna garanzia per: danni causati per installazione, uso, modifiche o riparazioni improprie effettuate da terzi non autorizzati o dall'utente finale; danni causati da qualsiasi soggetto (diverso da IPSES) o da fattori esterni; inadeguatezza a particolari scopi; danni accidentali.

Reclami:

Ogni reclamo, entro i termini di garanzia, dovrà essere inviato contattando gli uffici IPSES al seguente indirizzo:

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

Limitazioni:

IPSES non fornisce nessun altro tipo di garanzia rispetto a quanto non sia esplicitamente qui scritto. Le garanzie prestate da IPSES sostituiscono ogni altra garanzia implicita e tali garanzie implicite sono escluse, nei limiti di quanto consentito.

[1] Franco spese di spedizione alla IPSES e spese di consegna

ATTENZIONE!

LE APPARECCHIATURE ELETTRICHE POSSONO COSTITUIRE CAUSA DI PERICOLO PER COSE O PERSONE

Questa appendice illustra le nuove caratteristiche tecniche dell'UNITÀ DI CONTROLLO ASSI MT2USB, evidenziando solo le caratteristiche che differiscono dalla precedente versione. Per tutto ciò che non è descritto nella presente appendice occorre fare riferimento al manuale della scheda MT2USB.

Leggere attentamente prima di procedere all'installazione.
È responsabilità dell'installatore assicurarsi che l'installazione risponda alle normative di sicurezza previste dalla legge.
Per qualsiasi informazione non contenuta nel presente manuale rivolgersi a:

IPSES S.r.l. – Via Suor Lazzarotto, 10 - 20020 Cesate (MI) Italia
Tel. (+39) 02/39449519 - (+39) 02/700403170
e-mail support@ipses.com

INDICE

INDICE 5
REVISIONI 6
PRINCIPALI CARATTERISTICHE 7
INSTALLAZIONE DEL *DRIVER* 8
PROTOCOLLO DI COMUNICAZIONE PER IL CONTROLLO REMOTO 11
CONFIGURAZIONE DELLA SCHEDA TRAMITE JUMPER 18
CONNESSIONI DELLA SCHEDA MT2USB 20
ESEMPI DI CONNESSIONE DEI SENSORI DI FINE CORSA 23
CONNESSIONI DEI MOTORI 26
CONNESSIONE DEL MOTORE (A 8 CONDUTTORI) 26
CONNESSIONI DEL MOTORE (A 6 CONDUTTORI) 27
CONNESSIONI DEL MOTORE (A 4 CONDUTTORI) 28
CARATTERISTICHE TECNICHE 29
BOX 31
SOFTWARE DEMO 33
INSTALLAZIONE 33
ESECUZIONE 33
CONNESSIONE CON L'UNITÀ MT2 33
FUNZIONALITA' 35
RIMOZIONE 39
LIBRERIA LABVIEW 40
ESEMPIO DI CONNESSIONE: HYPERTERMINAL 42
ESEMPIO D'USO DELLA DLL D2XX 47
CODICE PRODOTTI 50
ALTRE VERSIONI DISPONIBILI 51
CONTATTI 52
INFORMAZIONI PER IL SUPPORTO TECNICO 53
RAPPORTO PROBLEMATICHE 53
ENGINEERING PROBLEM REPORT 54

REVISIONI

Revisioni manuale

Revisione/ Data	Descrizione modifica	Autore
01.00.0000 Luglio, 2013	Rilascio prima versione	Mancuso C. Pizzocolo G.
01.01.0000 Aprile 2015	Sostituzione foto con versione 2.3 Inserimento nuovi diodi LED e jumper	Bottaccioli M.
01.01.0001 Febbraio 2016	Modifiche minori	Bottaccioli M.
01.01.0002 Agosto 2016	Aggiunta logo certificazione ISO 9001:20015	Bottaccioli M.

PRINCIPALI CARATTERISTICHE

Le schede di controllo assi della serie MT2USB sono delle unità di controllo di ridotte dimensioni e basso consumo, in grado di controllare due motori passo-passo (sia bipolari, sia unipolari) e i relativi sensori di fine corsa (uno per ogni asse, con polarità programmabile). Il dispositivo riconosce segnali di fine corsa anche da sensori ottici.
I sensori ottici vengono alimentati solo quando uno dei motori è in movimento oppure nel caso in cui ne venga richiesto lo stato, al fine di preservarne la durata.
Il controllo dei motori e la configurazione dello strumento avvengono tramite interfaccia USB, facilmente gestibile mediante il *driver* fornito a corredo.
La velocità di rotazione dei motori può essere facilmente e indipendentemente configurata secondo le proprie esigenze.
Lo strumento dispone inoltre di una uscita ausiliaria che può essere attivata con apposito comando.
In Figura 1 è mostrata la scheda MT2USB con la posizione dei connettori e della porta USB.

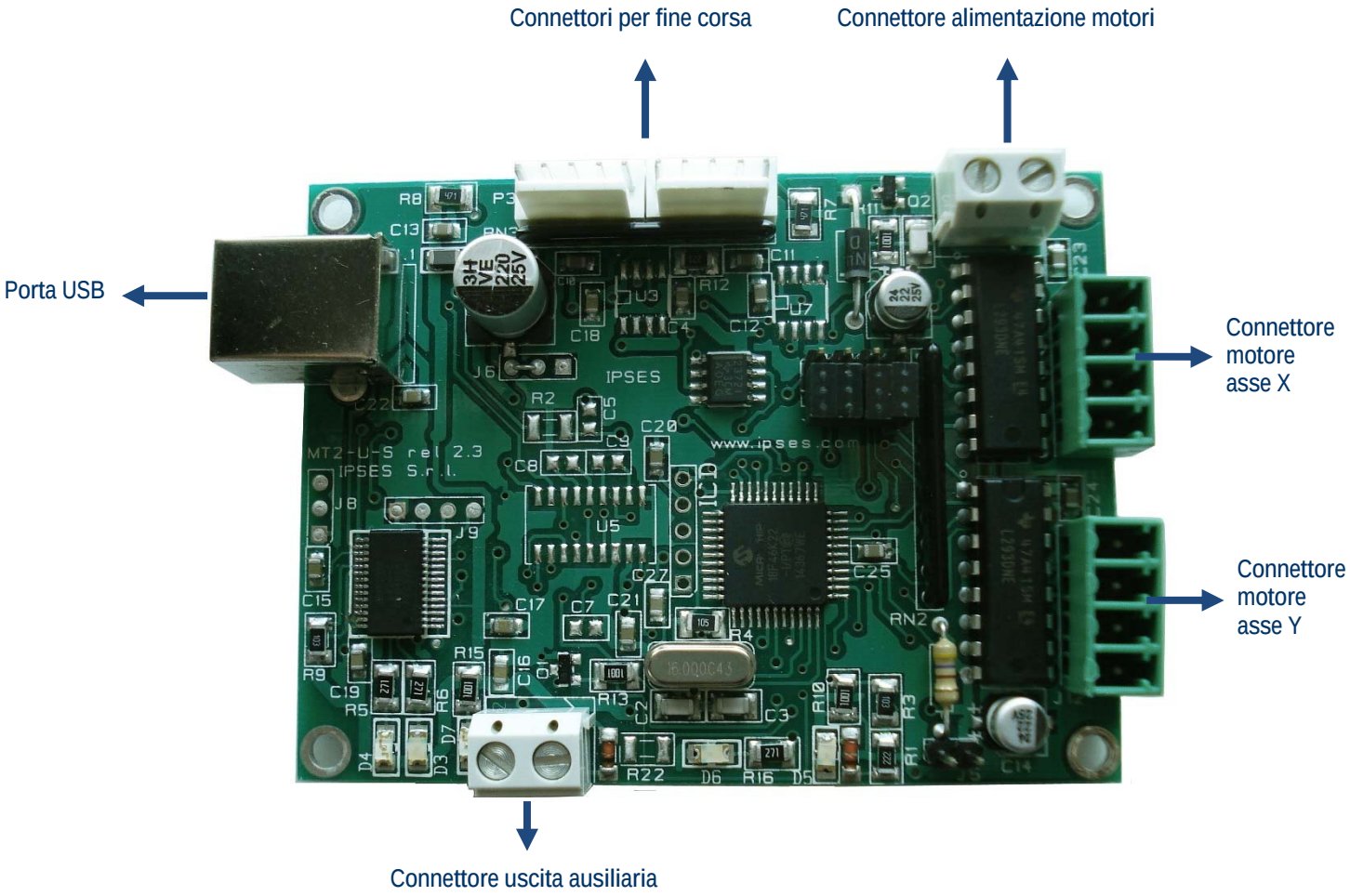


Figura 1: Scheda MT2USB - Posizione dei connettori e della porta USB

INSTALLAZIONE DEL *DRIVER*

Si consiglia di eseguire semplicemente l'installazione del *software* (prima di connettere la scheda al PC), con cui vengono installati in modo automatico anche i *driver* USB per la scheda.
In questo modo non è necessario seguire le indicazioni di questo capitolo poiché il sistema riconoscerà automaticamente il dispositivo (se connesso dopo l'installazione del *software*).

Se NON si installa il *software* fornito a corredo e si utilizza la scheda MT2USB è necessario installare solo il *driver* USB fornito da IPSES e certificato per i più recenti sistemi operativi Microsoft:

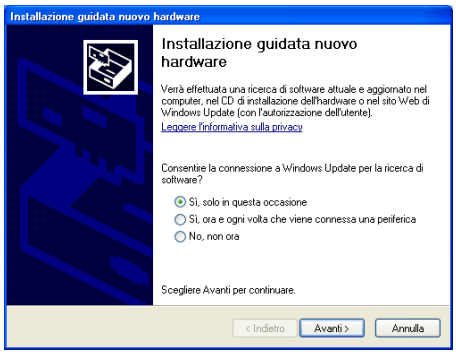
- Microsoft Windows 2000 family
- Microsoft Windows XP family, x86
- Microsoft Windows Server 2003 family, x86
- Microsoft Windows Server 2003 family, x64
- Microsoft Windows XP family, x64
- Microsoft Windows Vista family, x86
- Microsoft Windows Vista family, x64
- Microsoft Windows Server 2008 family, 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 e 8.1
- Microsoft Windows 8 e 8.1 x64
- Microsoft Windows 10
- Microsoft Windows 10 x64

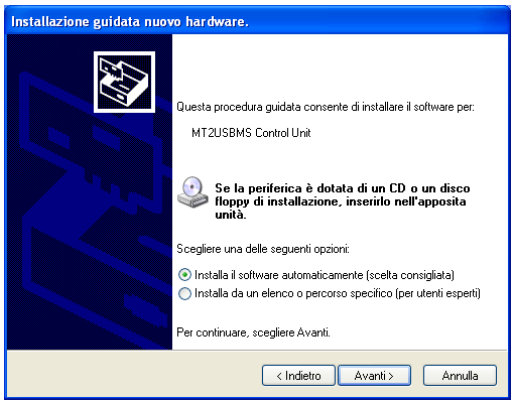
Se il PC è connesso ad Internet è possibile seguire la procedura di installazione automatica con *Windows Update*, altrimenti è necessario procedere con l'istallazione manuale da CD.

Procedura automatica con *Windows Update*

1) Collegare con il cavo USB la scheda MT2USB al PC. Il sistema operativo *Windows* rileva la presenza di un dispositivo con un messaggio simile a questo:

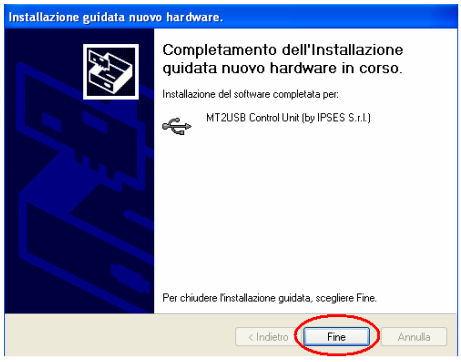
2) Nella successiva finestra "Installazione guidata nuovo hardware" scegliere "Sì, solo in questa occasione" e quindi premere il pulsante "Avanti".

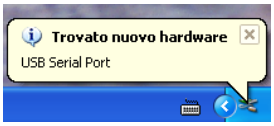




3) Successivamente, scegliere "Installa il software automaticamente (Scelta Consigliata)" e premere il pulsante "Avanti". Attendere il termine della ricerca e dell'installazione.

4) L'avvenuta installazione è segnalata dal messaggio di completamento dell'aggiornamento guidato *hardware* in corso. Per terminare, scegliere "Fine".



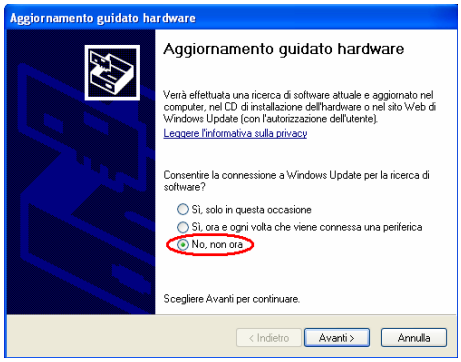


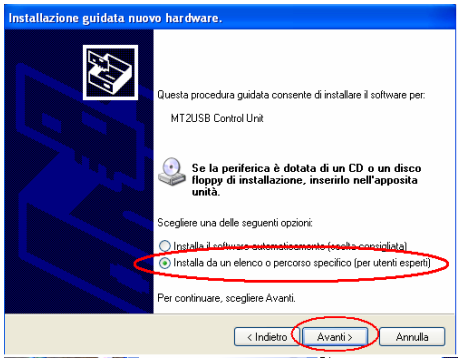
5) Terminata l'installazione dell'*hardware* descritta sopra, viene rilevata la nuova periferica "*USB Serial Port*". Ripercorrere di nuovo quanto sopra dal punto 2).

Procedura installazione *driver* manuale

1) Collegare con il cavo USB la scheda MT2USB al PC. Il sistema operativo *Windows* rileva la presenza di un dispositivo con un messaggio simile a questo:

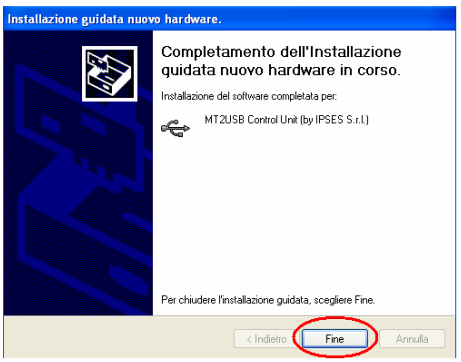
2) Nella successiva finestra "Installazione guidata nuovo hardware" scegliere "No, non ora" e quindi "Avanti".

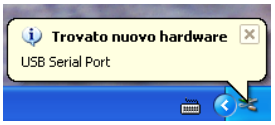




3) Successivamente, scegliere "Installa da un elenco o percorso specifico (per utenti esperti)" e "Avanti". Quindi selezionare la cartella "driver" dal CD fornito con la scheda.

4) L'avvenuta installazione è segnalata dal messaggio di completamento dell'aggiornamento guidato *hardware* in corso. Per terminare, scegliere "Fine".





Terminata l'installazione dell'*hardware* descritta sopra, viene rilevata la nuova periferica "*USB Serial Port*". Ripercorrere di nuovo quanto sopra dal punto 2).

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

PROTOCOLLO DI COMUNICAZIONE PER IL CONTROLLO REMOTO

La comunicazione con l'unità di controllo MT2USB avviene tramite l'interfaccia USB, gestibile con il *driver* di semplice utilizzo fornito a corredo.
I comandi sono stringhe in codice ASCII terminanti con il carattere <*CR*> (0x0D in esadecimale).
L'invio di altri caratteri di controllo (<*LF*>, <*VT*>, ecc.), genera un errore di sintassi.
Si può usare indifferentemente il carattere minuscolo o maiuscolo (i comandi non sono quindi *case-sensitive*).
Nel caso in cui si scelga di utilizzare la porta seriale virtuale messa a disposizione dal driver, i comandi potranno essere impartiti direttamente mediante qualsiasi client seriale (ad esempio l'HyperTerminal di Windows, a pagina 42 del presente manuale), così come sarà possibile sviluppare programmi applicativi utilizzando le normali funzioni di comunicazione messe a disposizione dal linguaggio di programmazione utilizzato.
La velocità di comunicazione della porta seriale virtuale può essere impostata a 9600 baud oppure a 19200 baud (fare riferimento al paragrafo di configurazione dei *jumper*, a pagina 18).
Gli altri i parametri di configurazione seriale da usare sono i seguenti:

- bit di dati: 8
- bit di parità: nessuno
- bit di stop: 1
- controllo di flusso: nessuno/hardware

Tramite il *jumper* J2 (descritto a pagina 18) è possibile abilitare il controllo del flusso hardware (CTS/RTS).

Nel caso in cui si scelga di utilizzare la libreria DLL di comunicazione USB a basso livello, si rimanda al manuale "LIBRERIA DINAMICA FTD2XX.DLL" di IPSES, presente al link http://www.ipses.com/PDF/IPSES-D2XX-it.pdf.

I comandi implementati sono i seguenti:

U	Richiesta di stato (vedere pag. 16 per una descrizione approfondita dello stato della scheda).
UU	Richiede lo stato globale della scheda (per ulteriori dettagli vedere pag. 16 e 16). Questo comando fornisce maggiori informazioni rispetto allo stato richiesto con il comando "U".
Pa,b	Muove gli assi alla posizione di coordinate (a,b), dove a e b sono le posizioni assolute espresse in mezzi passi. Questo comando è identico a quello implementato nella precedente versione, ma, a differenza di prima, ora sia a sia b possono essere compresi tra -999.999.999 e +999.999.999
Xa	Muove l'asse X alla posizione di coordinate a (posizione assoluta espressa in mezzi passi). Questo comando è identico a quello implementato nella precedente versione, ma, a differenza di prima, ora il parametro a può essere compresio tra -999.999.999 e +999.999.999
Yb	Muove l'asse Y alla posizione di coordinate b (posizione assoluta espressa in mezzi passi). Questo comando è identico a quello implementato nella precedente versione, ma, a differenza di prima, ora il parametro b può essere compreso tra -999.999.999 e +999.999.999
Da,b	Muove gli assi per una corsa di a e b (movimento relativo). a e b sono le corse espresse in mezzi passi. È possibile omettere il parametro b per muovere solo il primo asse. Sia a sia b possono essere compresi tra -999.999.999 e +999.999.999

L1	Attiva l'uscita ausiliaria.
L0	Disattiva l'uscita ausiliaria.
H	Porta entrambi gli assi alla *home position* (ossia a fine corsa). Durante la ricerca della *home position*, è possibile continuare a comunicare con la scheda (è possibile inviare anche il comando "K" per terminare la ricerca della *home position*).
HX	Porta l'asse X alla *home position* (ossia a fine corsa). Durante la ricerca della *home position*, è possibile continuare a comunicare con la scheda (è possibile inviare anche il comando "K" per terminare la ricerca della *home position*).
HY	Porta l'asse Y alla *home position* (ossia a fine corsa). Durante la ricerca della *home position*, è possibile continuare a comunicare con la scheda (è possibile inviare anche il comando "K" per terminare la ricerca dell'*home position*,).
K	Arresta il movimento corrente su entrambi gli assi. Questo comando è utilizzabile anche durante la ricerca della *home position*.
KX	Arresta il movimento sull'asse X (eccetto durante l'home position). Questo comando è utilizzabile anche durante la ricerca della *home position*.
KY	Arresta il movimento sull'asse Y (eccetto durante l'home position). Questo comando è utilizzabile anche durante la ricerca di *home position*.
GX,n	Attiva il moto perpetuo per l'asse X; con n maggiore o uguale a 0 fa partire il movimento in avanti, con n minore di 0 quello in indietro. Il parametro n non può essere omesso.
GY,n	Attiva il moto perpetuo per l'asse Y; con n maggiore o uguale a 0 fa partire il movimento in avanti, con n minore di 0 quello in indietro. Il parametro n non può essere omesso.
CX,n	Modalità di funzionamento dell'azione frenante (*braking action)* per l'asse X. Permette di impostare il *duty cycle* in un *range* compreso tra lo 0% e il 50% del PWM. Il valore di n deve essere compreso tra 0 e 50 Se impostato a zero, la *braking action* viene totalmente disattivata. Il valore di *default* è 0 (*braking action* disattivata). Usare questa modalità con attenzione, impostando valori di PWM inversamente proporzionali al tempo per cui si vuole lasciare attiva la funzione, in modo da evitare il surriscaldamento del motore o del sistema.
CY,n	Modalità di funzionamento dell'azione frenante (*braking action)* per l'asse Y. Permette di impostare il *duty cycle* in un *range* compreso tra lo 0% e il 50% del PWM. Il valore di n deve essere compreso tra 0 e 50 Se impostato a zero, la *braking action* viene totalmente disattivata. Il valore di *default* è 0 (*braking action* disattivata). Usare questa modalità con attenzione, impostando valori di PWM inversamente proporzionali al tempo per cui si vuole lasciare attiva la funzione, in modo da evitare il surriscaldamento del motore o del sistema.
CX?	Richiede il *duty cycle* impostato per il motore dell'asse X. Il valore restituito è in percentuale.
CY?	Richiede il *duty cycle* impostato per il motore dell'asse Y. Il valore restituito è in percentuale.
FX,n	Imposta la posizione assoluta corrente sull'asse X, con motore fermo (in caso contrario viene generato un errore di comando illegale). Il parametro n può essere compreso tra -999.999.999 e +999.999.999.
FY,n	Imposta la posizione assoluta corrente sull'asse Y, con motore fermo (in caso contrario viene generato un errore di comando illegale). Il parametro n può essere compreso tra -999.999.999 e +999.999.999.
SX,n	Imposta a n mezzi passi/s la velocità di regime per il movimento per l'asse X. Il parametro n può essere compreso tra 5 e 5.000 e può essere eseguito solo se il motore X è fermo (e non ha importanza se il motore Y si sta muovendo).
SY,n	Imposta a n mezzi passi/s la velocità di regime per il movimento per l'asse Y. Il parametro n può essere compreso tra 5 e 5.000 e può essere eseguito solo se il motore Y è fermo (e non ha importanza se il motore X si sta muovendo).
SX?	Richiede la velocità di movimento per l'asse X correntemente impostata.
SY?	Richiede la velocità di movimento per l'asse Y correntemente impostata.
W	Richiede la posizione corrente. Questo comando provoca la restituzione delle coordinate correnti come coppia (x,y) dove x ed y sono le coordinate assolute espresse in mezzi passi per la versione MT2USB. Nel caso di posizione non nota, viene restituito il carattere #.
E1,n	Imposta a n il massimo numero di mezzi passi, per entrambi gli assi, che il motore potrà compiere durante la prima ricerca del fine corsa a seguito del comando di *home*. Il parametro n può essere compresi tra 10 e 999.999. Per impostare distintamente questo valore per ogni singolo asse (X o Y) si vedano i comandi I1 e Q1.

E2,n	Imposta a n il massimo numero di mezzi passi, per entrambi gli assi, che il motore potrà compiere per eseguire un movimento in avanti rispetto alla posizione rilevata di fine corsa, al fine di prevenire la lettura di falsi contatti, durante l'esecuzione del comando di *home*. Il parametro n può essere compresi tra 5 e 32. Questo parametro deve essere inferiore al parametro impostato con il comando E3, altrimenti l'impostazione non verrà accettata. Per impostare distintamente questo valore per ogni singolo asse (X o Y) si vedano i comandi I2 e Q2.
E3,n	Imposta a n il massimo numero di mezzi passi, per entrambi gli assi, che il motore potrà compiere per eseguire un movimento all'indietro a seguito dello spostamento in avanti, atto a rilevare nuovamente la posizione di fine corsa, al fine di prevenire la lettura di falsi contatti, durante l'esecuzione del comando di *home*. Il parametro n può essere compresi tra 15 e 32.757. Questo parametro deve essere superiore al parametro impostato con il comando E2, altrimenti l'impostazione non verrà accettata. Per impostare distintamente questo valore per ogni singolo asse (X o Y) si vedano i comandi I3 e Q3.
E1?	Richiede il massimo numero di mezzi passi memorizzato per il movimento di ricerca della posizione di *home*. Diversamente dalla precedente versione, la risposta a questo comando sarà costituito da due valori numerici: uno riferito all'asse X e l'altro all'asse Y (con questa versione firmware è infatti possibile definire differenti parametri per l'asse X e l'asse Y – si vedano i comandi Ix e Qx).
E2?	Richiede il massimo numero di mezzi passi memorizzato per il movimento in avanti durante la ricerca della posizione di *home*. La risposta a questo comando sarà costituito da due valori numerici: uno riferito all'asse X e l'altro all'asse Y (con questa versione *firmware* è infatti possibile definire differenti parametri per l'asse X e l'asse Y – si vedano i comandi Ix e Qx).
E3?	Richiede il massimo numero di mezzi passi memorizzato per il movimento all'indietro durante la ricerca della posizione di *home*. La risposta a questo comando sarà costituito da due valori numerici: uno riferito all'asse X e l'altro all'asse Y (con questa versione *firmware* è infatti possibile definire differenti parametri per l'asse X e l'asse Y – si vedano i comandi Ix e Qx).
?	Richiede la versione *firmware* corrente e il *serial number* dello strumento. La risposta sarà una stringa ASCII simile a "*MT2USB vx.x – S/N:yyyyyyyy – by IPSES S.r.l. <www.ipses.com>*", dove *x.x* rappresenta la versione *firmware* dello strumento e *yyyyyy* è il *serial number*. La versione *firmware* deve essere maggiore o uguale a 2.0 e il S/N maggiore di 20130100 (altrimenti non sono applicabili le note indicate nel presente documento: contattare IPSES per maggiori informazioni)
M	Salva le impostazioni di velocità e modalità di funzionamento correntemente impostate in memoria non volatile.
Z	Richiede la tensione a cui sono alimentati i *driver* di pilotaggio dei motori. La risposta della scheda mostrerà la tensione (in *volt*) fornita dall'utente per alimentare i motori, nel formato x.y.
T1,n	Imposta la ricerca automatica della *home position* all'accensione. Il valore di n può essere o 0 o 1. Quando il valore è pari a 1, viene abilitata la ricerca della *home position*, se è 0, non viene effettuata la ricerca della *home position* all'accensione. Di *default*, il valore impostato è 0.
T2,n	Imposta la polarità dei sensori di fine corsa. Il valore di n definisce la polarità in cui i fine corsa risultano attivi (bassa o alta): quando n è impostato su 0, il sistema considererà il fine corsa attivo quando l'ingresso corrispondente è a 0V, mentre quando è impostato su 1 considererà attivo il fine corsa quando l'ingresso corrispondente ha un livello logico alto. Il comando può essere impostato solo quando entrambi i motori sono fermi. Il valore di *default* è 0.
T3,n	Impedisce il movimento degli assi oltre la posizione di fine corsa (movimento negativo del motore). Quando il valore di n è 1, il sistema ferma il motore ogni volta che viene raggiunta la posizione di fine corsa (durante un movimento negativo). Quando n vale invece 0, il motore non verrà mai fermato anche se raggiunge il fine corsa negativo. Il comando può essere impostato solo quando entrambi i motori sono fermi. Di default, il valore di n è 1.
T1?	Richiede lo stato a cui è stato impostato T1 per verificare se è stata abilitata o meno la funzionalità di ricerca automatica della *home position* all'accensione. Quando il valore è pari a 1, vuol dire che è abilitata, se invece è 0, significa che non viene effettuata la ricerca della *home position* all'accensione.
T2?	Richiede lo stato della polarità dei sensori di fine corsa. La risposta può essere 0 o 1: quando la risposta è 0 vuol dire che la polarità ha un livello logico basso, quando è 1 significa che è alto.
T3?	Richiede lo stato a cui è stato impostato T3 per verificare se il movimento oltre il fine corsa è abilitato o meno. La risposta può essere 0 o 1: quando la risposta è 0 vuol dire che non è abilitato, quando è 1 significa che è abilitato.

AX,n	Modalità di funzionamento dell'azione frenante per l'asse X. Imposta la frequenza del PWM che può essere compresa tra 80KHz e 200KHz. Il valore di n deve essere incluso tra 80 e 200: tale valore corrisponde alla frequenza in kHz. Il valore di default è 100.
AY,n	Modalità di funzionamento dell'azione frenante per l'asse Y. Imposta la frequenza del PWM che può essere compresa tra 80KHz e 200KHz. Il valore di n deve essere incluso tra 80 e 200: tale valore corrisponde alla frequenza in kHz. Il valore di default è 100.
AX?	Richiede la frequenza impostata per il PWM (in kHz) per il motore sull'asse X, quando viene attivata l'azione frenante.
AY?	Richiede la frequenza impostata per il PWM (in kHz) per il motore sull'asse Y, quando viene attivata l'azione frenante.
J1?	Richiede lo stato del *jumper* del *baud rate* (J1). 1=*jumper* non inserito, ossia velocità di 9.600 *baud*; 0=*jumper* inserito, ossia velocità di 19.200 *baud*.
J2?	Richiede lo stato del *jumper* J2. 1=jumper non inserito; 0=jumper inserito. Il *jumper* J2 è riservato.
J3?	Richiede lo stato del *jumper* J3. 1=jumper non inserito; 0=jumper inserito (all'accensione del sistema, verranno ripristinati tutti i parametri di *default*).
J4?	Richiede lo stato del *jumper* J4. 1=jumper non inserito; 0=jumper inserito. Il *jumper* J4 è riservato.
I1,n	Imposta il numero massimo di mezzi passi per l'asse X durante l'esecuzione del commando di ricerca dell'*home position*. Il valore di n deve essere incluso tra 10 e +999.999.999. L'impostazione può essere effettuata solo quando il motore dell'asse X è fermo.
I2,n	Imposta a n il massimo numero di mezzi passi, per l'asse X che il motore potrà compiere per eseguire un movimento in avanti rispetto alla posizione rilevata di fine corsa, al fine di prevenire la lettura di falsi contatti, durante l'esecuzione del comando di *home*. Questo parametro deve essere inferiore al parametro impostato con il comando I3, altrimenti l'impostazione non verrà accettata
I3,n	Imposta a n il massimo numero di mezzi passi, per l'asse X che il motore potrà compiere per eseguire un movimento all'indietro a seguito dello spostamento in avanti, atto a rilevare nuovamente la posizione di fine corsa, al fine di prevenire la lettura di falsi contatti, durante l'esecuzione del comando di *home*. Questo parametro deve essere superiore al parametro impostato con il comando I2, altrimenti l'impostazione non verrà accettata.
I1?	Richiede il massimo numero di mezzi passi memorizzato per il movimento di ricerca della posizione di *home* per l'asse X.
I2?	Richiede il massimo numero di mezzi passi memorizzato per il movimento in avanti durante la ricerca della posizione di *home* per l'asse X.
I3?	Richiede il massimo numero di mezzi passi memorizzato per il movimento all'indietro durante la ricerca della posizione di *home* per l'asse X.
Q1,n	Imposta il numero massimo di mezzi passi per l'asse Y durante l'esecuzione del comando di ricerca della *home position*. Il valore di n deve essere incluso tra 10 e +999.999.999. L'impostazione può essere effettuata solo quando il motore dell'asse Y è fermo.
Q2,n	Imposta a n il massimo numero di mezzi passi, per l'asse Y, che il motore potrà compiere per eseguire un movimento in avanti rispetto alla posizione rilevata di fine corsa, al fine di prevenire la lettura di falsi contatti, durante l'esecuzione del comando di *home*. Questo parametro deve essere inferiore al parametro impostato con il comando Q3, altrimenti l'impostazione non verrà accettata
Q3,n	Imposta a n il massimo numero di mezzi passi, per l'asse Y, che il motore potrà compiere per eseguire un movimento all'indietro a seguito dello spostamento in avanti, atto a rilevare nuovamente la posizione di fine corsa, al fine di prevenire la lettura di falsi contatti, durante l'esecuzione del comando di *home*. Questo parametro deve essere superiore al parametro impostato con il comando Q2, altrimenti l'impostazione non verrà accettata.
Q1?	Richiede il massimo numero di mezzi passi memorizzato per il movimento di ricerca della posizione di *home* per l'asse Y.
Q2?	Richiede il massimo numero di mezzi passi memorizzato per il movimento in avanti durante la ricerca della posizione di *home* per l'asse Y.
Q3?	Richiede il massimo numero di mezzi passi memorizzato per il movimento all'indietro durante la ricerca della posizione di *home* per l'asse Y.
VX,n	Imposta la velocità di partenza per il motore dell'asse X, Il valore di n deve essere incluso tra 10 e 5.000. L'impostazione può essere effettuata solo quando il motore dell'asse X è fermo. Se si imposta n a 5.000, il sistema non effettuerà mai rampe di accelerazione e decelerazioni, qualsiasi sia la velocità di regime impostata (comando SX,n).

VY,n	Imposta la velocità di partenza per il motore dell'asse Y, Il valore di n deve essere incluso tra 10 e 5.000. L'impostazione può essere effettuata solo quando il motore dell'asse Y è fermo. Se si imposta n a 5.000, il sistema non effettuerà mai rampe di accelerazione e decelerazioni, qualsiasi sia la velocità di regime impostata (comando SY,n).
VX?	Richiede la velocità di partenza impostata per il motore dell'asse X.
VY?	Richiede la velocità di partenza impostata per il motore dell'asse Y
RX,n	Imposta ogni quanti mezzi passi variare la velocità della rampa di accelerazione e decelerazione per il motore dell'asse X. Il valore di n deve essere incluso tra 20 e 100. L'impostazione può essere effettuata solo quando il motore dell'asse X è fermo. Il valore di *default* è 20
RY,n	Imposta ogni quanti mezzi passi variare la velocità della rampa di accelerazione e decelerazione per il motore dell'asse Y. Il valore di n deve essere incluso tra 20 e 100. L'impostazione può essere effettuata solo quando il motore dell'asse Y è fermo. Il valore di *default* è 20
RX?	Richiede il valore impostato di ogni quanti mezzi passi varia la velocità della rampa di accelerazione e decelerazione per il motore dell'asse X.
RY?	Richiede il valore impostato di ogni quanti mezzi passi varia la velocità della rampa di accelerazione e decelerazione per il motore dell'asse Y.

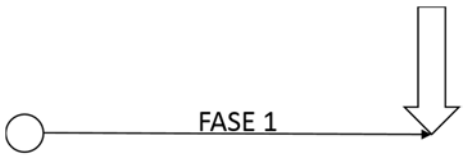
Per tutti i comandi facenti uso dei parametri *X*,*Y* è anche possibile, in alternativa, fare uso rispettivamente di *1* per l'asse *X*, e di *2* per l'asse *Y*. Ad esempio, il comando *GX,1* è equivalente al comando G*1,1*.

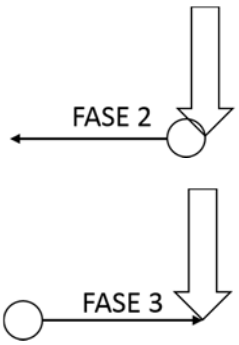
Ad ogni comando di interrogazione inviato (u?, Cx?, ecc.) il dispositivo risponderà con la stringa contenente l'informazione voluta, seguita dai caratteri di controllo <CR> e <LF> (rispettivamente 13 e 10 in notazione decimale).

Quando si esegue il comando home (si vedano i comandi H, Hx e Hy) il motore si muove all'indietro alla ricerca della posizione notificata dall'attivazione del segnale di fine corsa, per un numero massimo di passi impostati tramite il comando E1,n. Se durante questa fase non viene rilevata la posizione di fine corsa il sistema genera un errore di ricerca della *Home position*. In caso contrario, all'individuazione del fine corsa, il motore si muoverà in avanti per il numero di passi impostato con il comando E2, e, successivamente, invertirà il senso di moto nuovamente alla ricerca del segnale di fine corsa, per un numero massimo di passi impostato con il comando E3.
In questo modo se il fine corsa viene letto per la seconda volta il motore si fermerà in corrispondenza di questa lettura.
È importante che durante la seconda fase (movimento in avanti) il motore si muova di un numero di passi sufficienti a disattivare il segnale di fine corsa, in caso contrario verrà generato un errore. Risulta quindi necessario che il parametro n del comando E2 sia impostato maggiore di zero e tale per cui, al termine di questa movimentazione, la meccanica del vostro sistema non attivi la segnalazione di fine corsa.
Queste operazioni prevengono la lettura di "falsi contatti" del fine-corsa e garantiscono di essere effettivamente nella posizione di *home*: per la durata dell'esecuzione di tale comando non è possibile inviarne altri (compresa la richiesta di stato e il comando di arresto).
È a cura dell'utente impostare un valore n per il comando E3 maggiore o uguale al valore del parametro n impostato per il comando E2, al fine di evitare il mancato rilevamento della posizione di fine corsa durante l'ultima fase d'esecuzione del comando di home, in seguito alla quale verrebbe asserito l'errore di ricerca della *home position*.

Tutte le posizioni e i movimenti si devono intendere misurate in mezzi passi (la velocità del motore, ad esempio, è misurata in mezzi passi al secondo).

In figura sono esemplificate le tre fasi:





Il comando di richiesta di stato ("*U*") fa sì che venga trasmesso un numero esadecimale su due cifre che rappresenta il *byte* di stato dell'unità secondo la seguente convenzione:

bit 7	errore
bit 6	posizione asse Y nota
bit 5	posizione asse X nota
bit 4	uscita ausiliaria attiva
bit 3	raggiunto fine corsa per asse Y
bit 2	raggiunto fine corsa per asse X
bit 1	movimento in corso
bit 0	ready (posizione nota dopo un home position)

Esempio: se viene letto lo stato mentre si stanno muovendo i due assi con la posizione di partenza nota, la stringa letta sarà "62" che (convertito da esadecimale a binario) corrisponde a "01100010", numeri che confrontati con la tabella precedente danno le stesse informazioni prima descritte. Se invece è in corso un movimento di assi con coordinata X sconosciuta e uscita ausiliaria attiva, la stringa sarà "52" ("01010010").

É anche disponibile il comando di richiesta di stato esteso ("*UU*"), che fa sì che venga trasmesso un numero esadecimale su tre cifre che rappresenta il *byte* di stato dell'unità secondo la seguente convenzione:

bit 11	Uso interno, da non considerare
bit 10	Uso interno, da non considerare
bit 9	movimento in corso su asse Y
bit 8	movimento in corso su asse X
bit 7	Errore
bit 6	posizione asse Y nota
bit 5	posizione asse X nota
bit 4	uscita ausiliaria attiva
bit 3	raggiunto fine corsa per asse Y
bit 2	raggiunto fine corsa per asse X
bit 1	Asse Y pronto (posizione nota a seguito di una *home position*).
bit 0	Asse X pronto (posizione nota a seguito di una *home position*).

Se alla richiesta di stato (sia con il comando U che con il comando UU) viene restituito un *bit* di errore alto (cioè se la scheda risponde con un codice tipo 080), viene concatenato un numero esadecimale a tre cifre (per esempio 080,002) secondo la convenzione elencata di seguito. Possono essere attivi anche più codici di errori.

I codici di errore previsti per il comando di richiesta di stato U, sono i seguenti:

bit 7 Raggiunto fine corsa asse Y durante movimento all'indietro, con movimenti negativi disabilitati tramite apposito *jumper*.
bit 6 Raggiunto fine corsa asse X durante movimento all'indietro, con movimenti negativi disabilitati tramite apposito *jumper*.
bit 5 *Checksum* dei dati memorizzati sulla memoria non volatile non valido.
bit 4 Parametri non validi memorizzati nella memoria non volatile interna.
bit 3 Errore nella ricerca dell'*Home position*.
bit 2 Parametro fuori *range* (ad es. si è cercato di impostare una velocità che non rientra nei *range* stabiliti).
bit 1 Comando illegale (ad es. una richiesta di movimento assoluto quando le posizione non sono note oppure quando c'è già un movimento in corso).
bit 0 Comando non riconosciuto

I codici di errore del comando di stato esteso UU, invece, sono i seguenti:

bit 9 Uso interno, da non considerare
bit 8 Uso interno, da non considerare
bit 7 Raggiunto fine corsa asse Y durante movimento all'indietro, con movimenti negativi disabilitati tramite apposito parametro (si veda comando T3).
bit 6 Raggiunto fine corsa asse X durante movimento all'indietro, con movimenti negativi disabilitati tramite apposito parametro (si veda comando T3).
bit 5 *Checksum* dei dati memorizzati sulla memoria non volatile non valido.
bit 4 Parametri non validi memorizzati nella memoria non volatile interna.
bit 3 Errore nella ricerca dell'*Home position*.
bit 2 Parametro fuori *range* (ad es. si è cercato di impostare una velocità che non rientra nei *range* stabiliti).
bit 1 Comando illegale (ad es. una richiesta di movimento assoluto quando le posizione non sono note oppure quando c'è già un movimento in corso).
bit 0 Comando non riconosciuto

Il *reset* di tutti gli errori avviene dopo ogni richiesta di stato.

CONFIGURAZIONE DELLA SCHEDA TRAMITE JUMPER

Tramite i *jumper* presenti sulla scheda MT2USB, evidenziati in Errore. L'origine riferimento non è stata trovata. con nomenclatura e posizione, è possibile programmare le seguenti funzioni:

- *jumper* J1: selezione della velocità di comunicazione della seriale virtuale: senza *jumper* inserito verrà impostata una velocità di 9.600 baud, con *jumper* inserito verrà impostata una velocità di 19.200 baud. Lo stato del *jumper* viene letto immediatamente dopo la connessione della porta USB e il *baudrate* impostato di conseguenza: qualsiasi variazione successiva dello stato del *jumper* non avrà effetto sino a quando l'USB non verrà fisicamente disconnessa e riconnessa.
- *jumper* J2: se inserito abilita il controllo del flusso *hardware* della seriale.
- *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").
- *jumper* J4: riservato. Lasciare senza *jumper* per il normale funzionamento della scheda.
- *Jumper J5*: se inserito cortocircuita la resistenza sull'uscita ausiliaria.

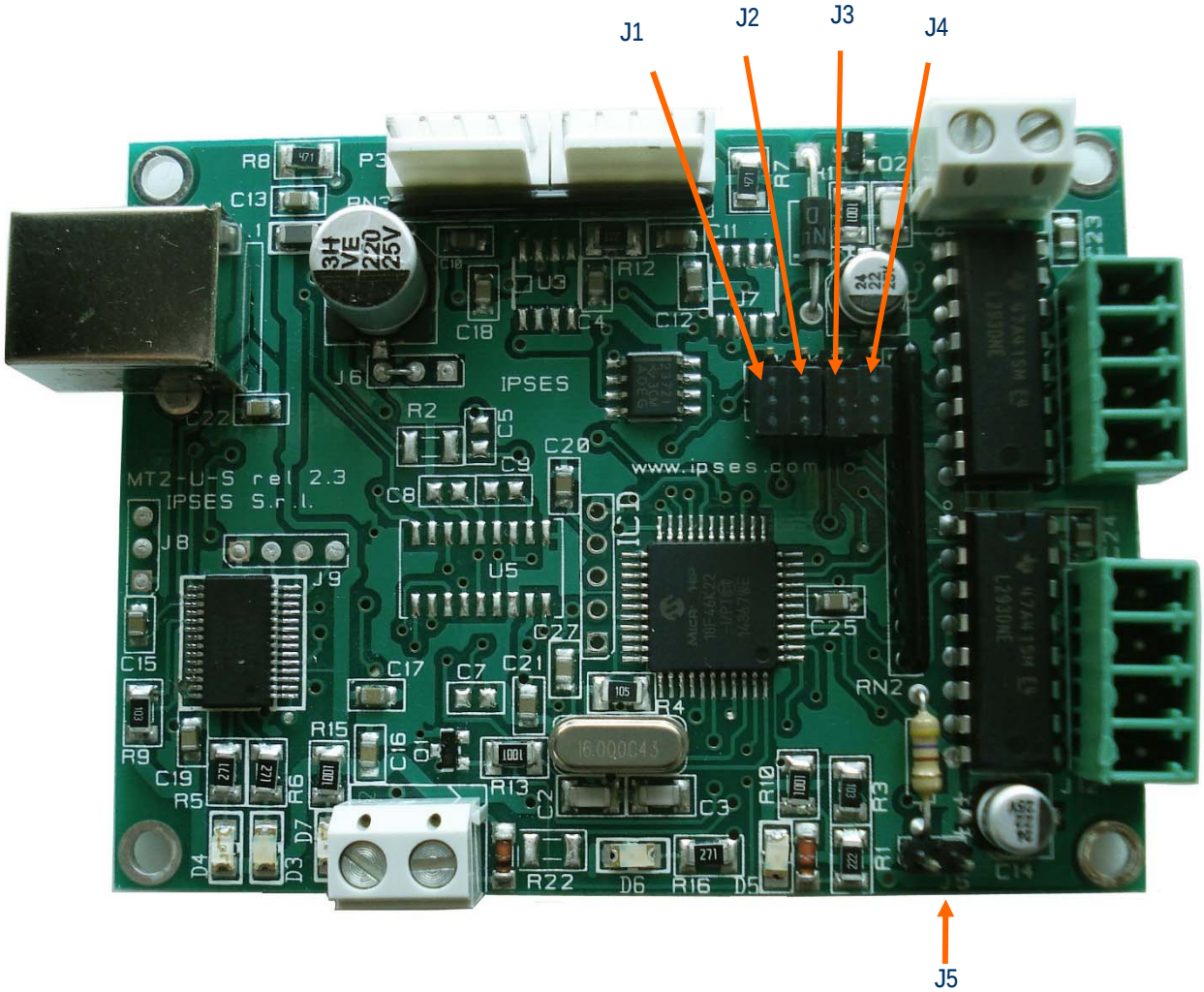


Figura 2: *jumper* della scheda MT2USB.

È possibile leggere lo stato dei *jumper* mediante i seguenti comandi:

J1?	Richiede lo stato del *jumper* del *baud rate* (J1). 1=*jumper* non inserito, ossia velocità di 9.600 *baud*; 0=*jumper* inserito, ossia velocità di 19.200 *baud*.
J2?	Richiede lo stato del *jumper* J2. 1=*jumper* non inserito; 0=*jumper* inserito (flusso hardware della seriale abilitato).
J3?	Richiede lo stato del *jumper* J3. 1=*jumper* non inserito; 0=*jumper* inserito (all'accensione del sistema, verranno ripristinati tutti i parametri di *defaut*).
J4?	Richiede lo stato del *jumper* J4. 1=*jumper* non inserito; 0=*jumper* inserito.

La funzionalità del *jumper* che permette di selezionare la ricerca della *home position* all'accensione può essere impostata mediante il comando T1.

La funzionalità del *jumper* che permette di selezionare la polarità dei fine corsa può essere impostata mediante il comando T2.

La funzionalità del *jumper* che permette di abilitare il blocco del motore al raggiungimento del fine corsa (su movimenti negativi) può essere impostata mediante il comando T3.

Per i dettagli su questi comandi si veda più su pag. 13.

CONNESSIONI DELLA SCHEDA MT2USB

La scheda è alimentata attraverso il connettore USB. È però necessario fornire al sistema l'alimentazione per pilotare i motori (utilizzando il connettore P1, visibile in figura Figura 3) che deve essere compresa tra 5 e 36V.

È inoltre possibile connettere allo strumento un sensore di fine corsa per ogni motore (connettori P3).
È anche possibile utilizzare l'uscita ausiliaria *open collector* (connettore P2) per comunicare con un PLC oppure per attivare altra elettronica di controllo.

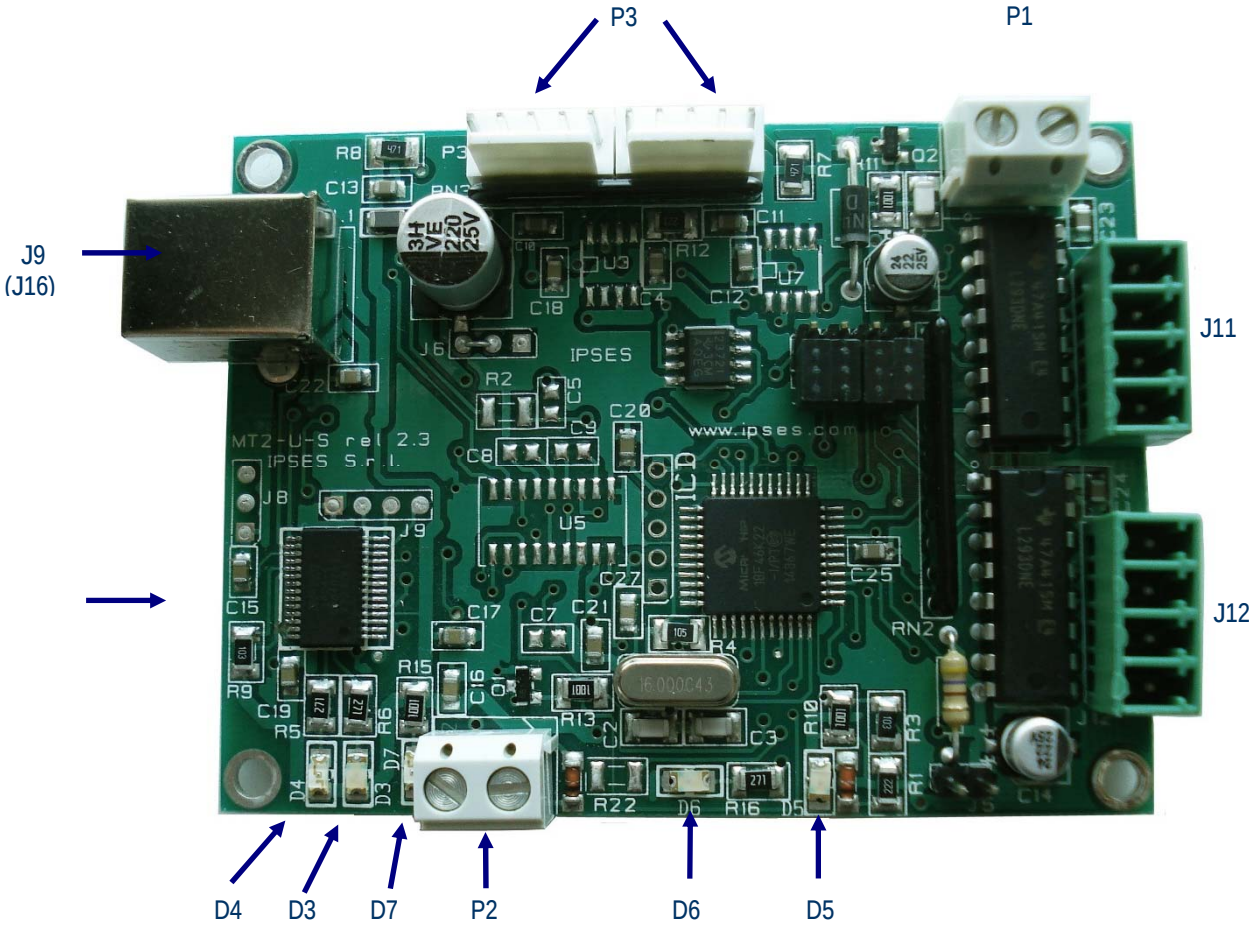


Figura 3: connettori della scheda MT2USB.

Di seguito sono illustrate tutte le connessioni della scheda:

J9	Connettore USB di tipo B per interfacciarsi al PC
P3	Sensori di fine corsa

Pin1: uscita alimentazione positiva (5Vdc, senza limitazione di corrente) per alimentare eventuale logica di rivelamento esterna del sensore di fine corsa dell'asse Y.
Pin2: uscita alimentazione positiva per LED infrarosso dell'asse Y (per fine corsa ottici).
Pin3: ingresso del sensore di fine corsa per l'asse Y.
Pin4: GND.
Pin5: uscita alimentazione negativa per LED infrarosso dell'asse Y (per fine corsa ottici).
Pin6: uscita alimentazione positiva (5Vdc, senza limitazione di corrente) per alimentare eventuale logica di rivelamento esterna del sensore di fine corsa dell'asse X.
Pin7: uscita alimentazione positiva per LED infrarosso dell'asse X (per fine corsa ottici).
Pin8: ingresso del sensore di fine corsa dell'asse X.
Pin9: GND.
Pin10: uscita alimentazione negativa per LED infrarosso dell'asse delle X (per fine corsa ottici).

P1 — Power supply
Pin1 (+): tensione positiva.
Pin2: GND.

P2 — Aux out (di cui a pagina 22 è mostrato lo schema del circuito)
Pin1: tensione positiva (limitata con una resistenza).
Pin2: *open-collector* verso GND.

J11 — Motore asse X

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

J12 — Motore asse Y

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

Per interfacciare i motori con i connettori J11 e J12 utilizzare morsetti di tipo Phoenix Contact con passo da 3,81 mm ([2]).

[2] Codice RS 220-4670 oppure codice Distrelec 141128

ATTENZIONE!
Non connettere o disconnettere il motore (o qualsiasi sua fase) quando la scheda è alimentata!
La tensione di +5Vdc presente sui *pin1* e *pin6* del connettore di fine corsa è ricavata direttamente dall'alimentazione del PC. Se sovraccaricata o cortocircuitata potrebbe quindi danneggiare il PC.

È consigliato collegare il cavo USB al dispositivo solo dopo che l'alimentazione è stata erogata allo stadio di pilotaggio dei motori.

La scheda è dotata inoltre di una uscita ausiliaria in configurazione *open-collector* (provvista di un resistore di *pull-up* connesso direttamente all'alimentazione dei motori) comandabile direttamente dall'utente (comando "L" descritto nella tabella precedente), il cui circuito dello stadio d'uscita è riportato qui di seguito:

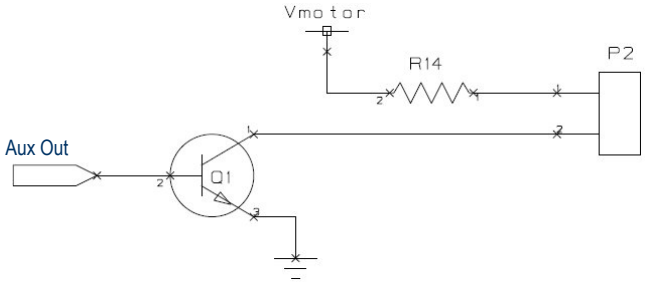


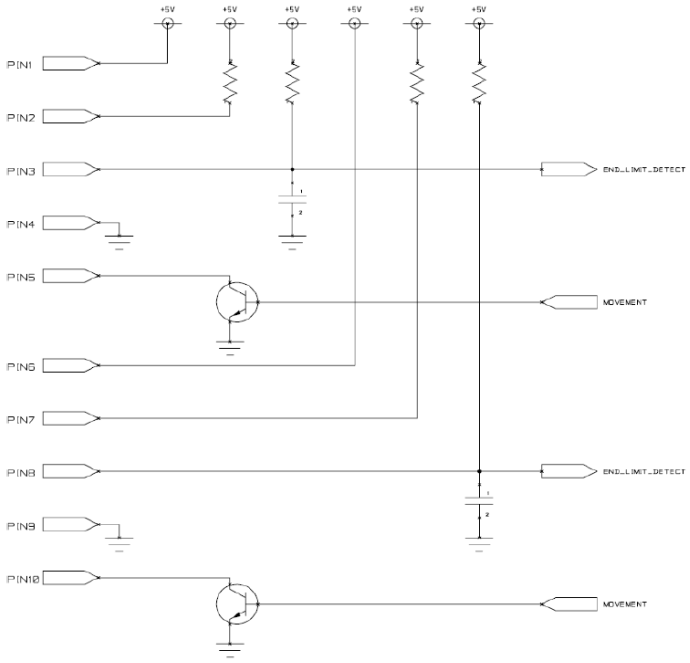
Figura 4: circuito uscita ausiliaria.

Il valore di R14 è di 470Ω (1/4 W).

I LED D4 (rosso) e D3 (verde), indicano, rispettivamente, lo scambio di dati tra lo strumento e il PC e l'avvenuta connessione dello strumento con il PC.
Il LED D5 (verde) indica la presenza dell'alimentazione dei motori.
Il LED D7 (rosso) indica l'attivazione dell'uscita ausiliaria, se alla scheda non è fornita la tensione di alimentazione dei motori, il LED rimane acceso con intensità minore.
Il LED D6 (verde) è collegato direttamente al microcontrollore e deve, di norma, rimanere spento.

ESEMPI DI CONNESSIONE DEI SENSORI DI FINE CORSA

La successiva Figura 5 mostra lo schema di implementazione del connettore P3 della scheda MT2USB per la gestione dei segnali di fine corsa.



ATTENZIONE!
La massima tensione in ingresso al *pin3* (*pin8*) non deve eccedere i +5V.

Figura 5: implementazione connettore J2.

Da questa figura si evince che, quando uno dei motori è in movimento, il dispositivo MT2USB interpreta il raggiungimento della posizione di home quando il *pin3* (per movimenti sull'asse Y), o il *pin8* (per movimenti sull'asse X) del connettore P3 subiscono un cambiamento di potenziale. Il comando *T2n* definisce la polarità dei segnali di fine corsa: quando *n* è impostato con valore 0 il segnale di fine corsa è attivo quando viene applicato GND, quando *n* è impostato su 1, invece, l'attivazione del fine corsa avviene quando sono applicati +5V.
Nel caso si impieghino fine corsa meccanici, per fine corsa normalmente aperti (N.O.), *n* deve essere impostato su 0; per fine corsa normalmente chiusi (N.C.), *n* deve invece essere impostato su 1.

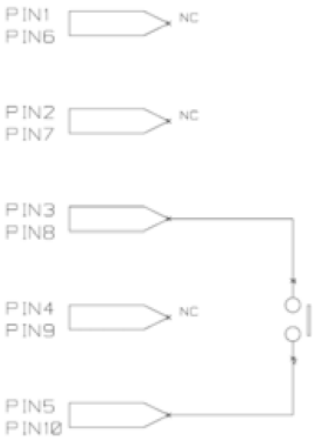


Figura 6: commutazione meccanica per il segnale di fine corsa.

Il segnale di fine-corsa può essere ottenuto con l'impiego di sensori di fine corsa ottici. La seguente Figura 7Errore. L'origine riferimento non è stata trovata. ne illustra il funzionamento. Il LED genera un fascio luminoso che illumina un elemento, ad esempio il *phototransistor*, in grado di alterare, quando irraggiato, le sue proprietà elettriche. Il *phototransistor* offre un percorso conduttivo; l'interruzione del fascio luminoso per l'interposizione di un oggetto opaco ne arresta la conduzione elettrica.

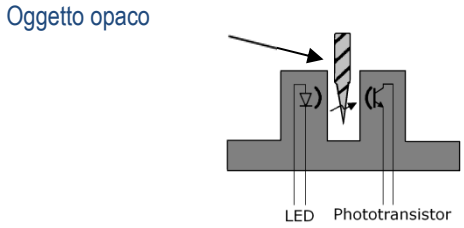


Figura 7: fine corsa ottico.

La Figura 8 indica i collegamenti da effettuare con il connettore P3 se si opta per l'adozione di un sensore ottico con l'uscita a *phototransistor*.

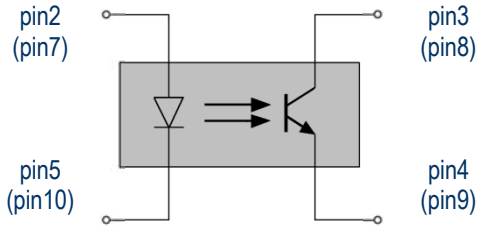


Figura 8: sensore ottico con l'uscita a *phototransistor*; sono indicati i collegamenti con i *pin* del connettore **P3** *per la* rilevazione del segnale di fine corsa lungo l'asse Y (asse X).

Per l'impiego di un sensore ottico con l'uscita a *phototransistor* a collettore aperto bisogna configurare la scheda impartendo il comando T2 con il valore impostato a 1.

Prestazioni superiori in termini di affidabilità della lettura della posizione di home sono raggiungibili con l'adozione di sensori ottici dotati di circuiti elettronici per una più accurata rilevazione del fascio emesso dal LED. In Figura 9 ne viene illustrato lo schema: si tratta di un sensore ottico dotato di logica alimentata per la rilevazione dell'interruzione del fascio per l'interposizione di un oggetto opaco. L'alimentazione alla logica è fornita tramite il *pin1*, per l'asse Y, e il *pin6*, per l'asse X.

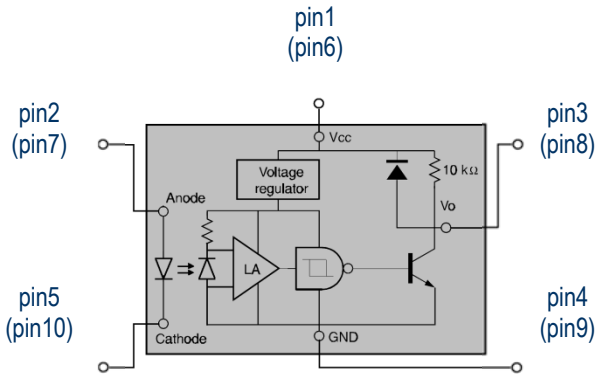


Figura 9: sensore ottico con logica alimentata per la rilevazione della presenza di un oggetto opaco; sono indicati i collegamenti con i *pin* del connettore P3 per la rilevazione del segnale di fine corsa lungo l'asse Y (asse X).

Per l'utilizzo di quest'ultimo sensore è necessario configurare la scheda impartendo il comando T2 con il valore impostato a 0.

ATTENZIONE!
La massima tensione dei segnali in ingresso ai *pin3* e *pin8* non deve mai eccedere i +5Vdc.
Valori di tensione superiori danneggerebbero irrimediabilmente i componenti della scheda.

CONNESSIONI DEI MOTORI

La scheda MT2USB può pilotare direttamente sia due motori monopolari sia due motori bipolari, tra cui motori a otto, quattro e a sei conduttori (con polo centrale). In queste pagine vengono mostrate tutte le possibili connessioni di questi motori passo-passo.

CONNESSIONE DEL MOTORE (A 8 CONDUTTORI)

Connessione in serie
Questo tipo di configurazione è utilizzata nelle applicazioni che richiedono una coppia elevata a basse velocità di rotazione. Data l'elevata induttanza che caratterizza questa configurazione, le prestazioni degradano a velocità elevate.

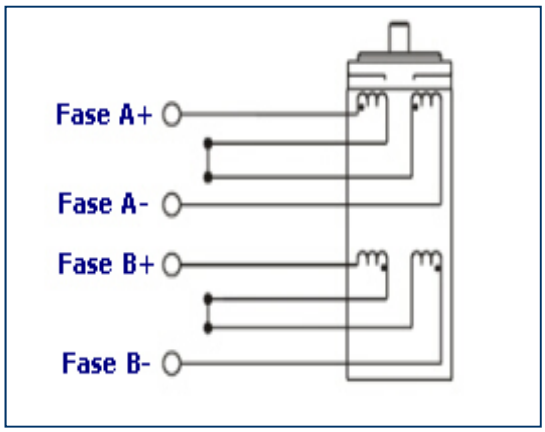


Figura 10: connessione serie.

Connessione parallela
Questa configurazione offre una coppia più stabile, ma più bassa a velocità basse rispetto alla connessione serie. Grazie al basso valore d'induttanza, le prestazioni non degenerano all'aumentare della velocità.

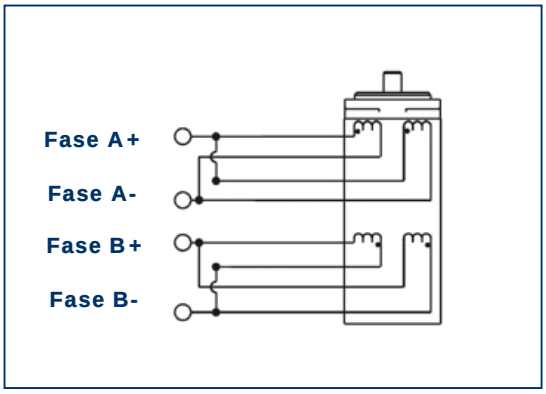


Figura 11: connessione parallela.

CONNESSIONI DEL MOTORE (A 6 CONDUTTORI)

Configurazione *half coil*
Questa configurazione, anche chiamata half copper, usa metà della bobina di ogni fase. Ciò porta a una bassa induttanza e ad una coppia ridotta, che però sarà più stabile a velocità elevate.

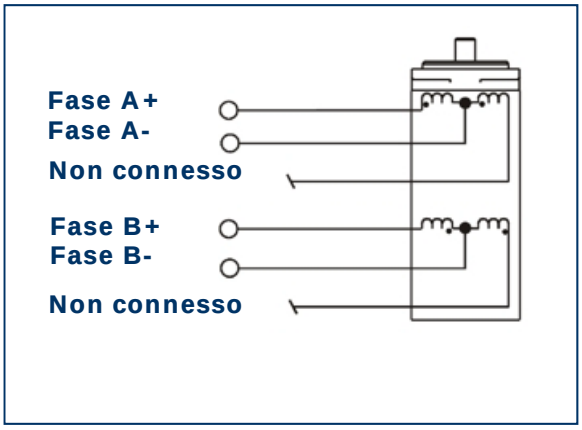


Figura 12: configurazione *half coil*.

Configurazione *full coil*
Questa configurazione, anche chiamata *full copper*, è utilizzata quando l'applicazione richiede una coppia elevata e una velocità non troppo alta.

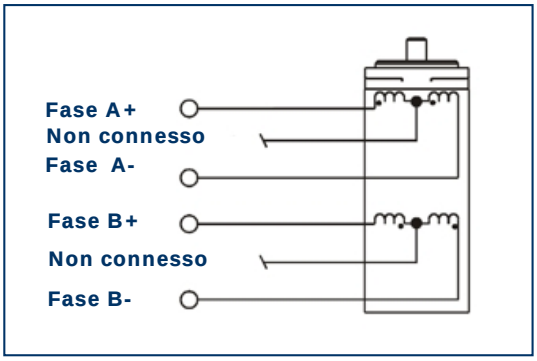


Figura 13: configurazione *full coil*.

CONNESSIONI DEL MOTORE (A 4 CONDUTTORI)

Motori a 4 conduttori
I motori a 4 conduttori sono I meno flessibili, ma i più facili da connettere. Velocità e coppia dipendono dall'induttanza caratteristica delle fasi.

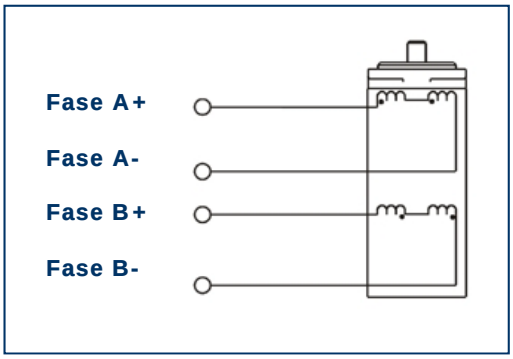


Figura 14: motori a 4 conduttori.

NOTA

Il verso di rotazione del motore dipende dalle connessioni delle fasi: per cambiarlo occorre invertire una sola coppia di fili di alimentazione delle fasi (soltanto la fase A o soltanto la fase B).

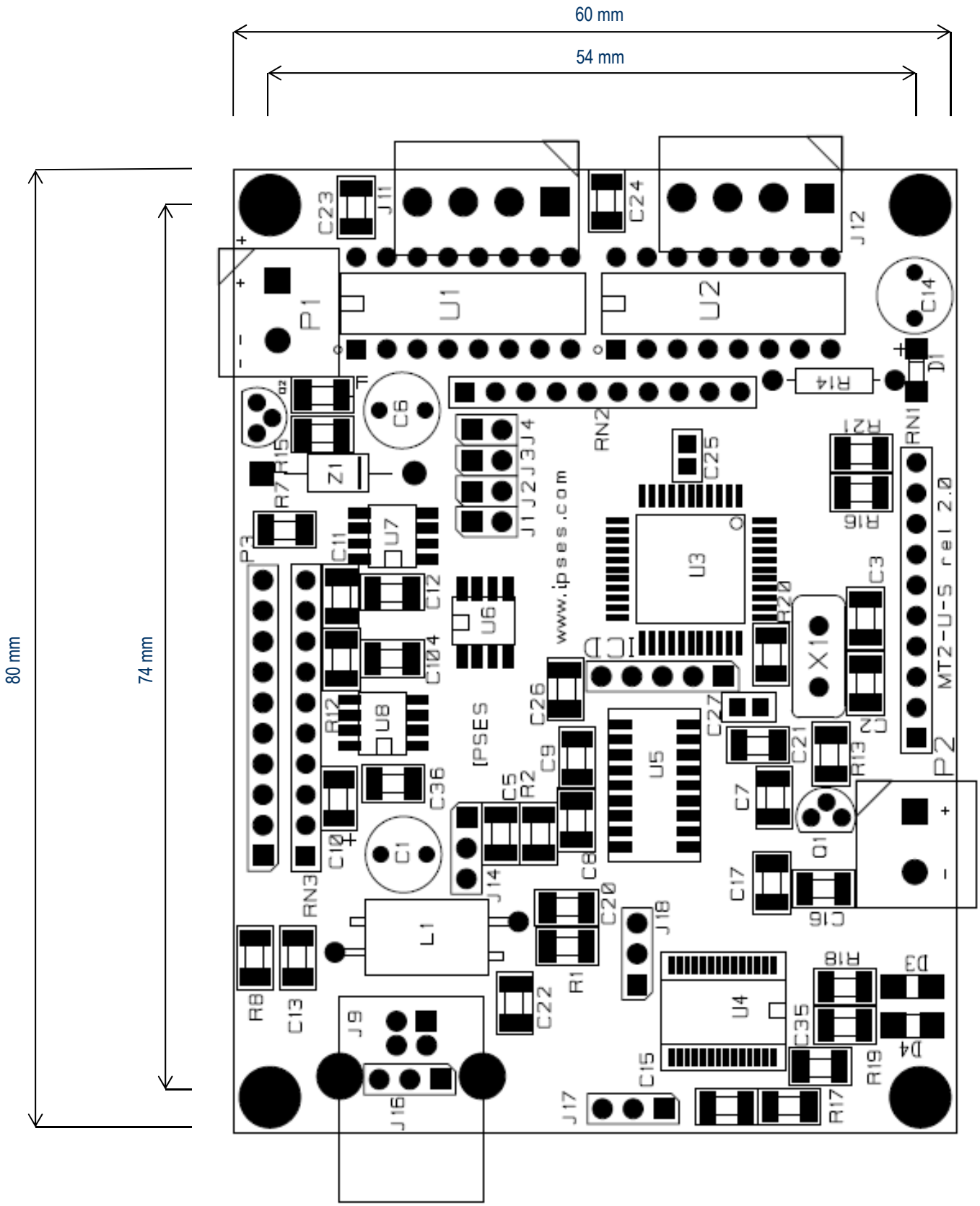
CARATTERISTICHE TECNICHE

Alimentazione:	logica autoalimentata direttamente dalla porta USB del computer
Alimentazione motori:	da 5 a 36Vdc
Corrente di uscita:	max 0,6A/fase (1,2A di picco)
Interfaccia:	USB 2.0 (connettore di tipo B)
Uscita ausiliaria:	configurazione *open collector* (Imax assorbita: 200mA).
Dimensioni:	60mm x 80mm (escluso connettori)
Movimento motori:	mezzo passo

Nota: l'alimentazione fornita ai motori è anche quella utilizzata per l'uscita ausiliaria.

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

Dimensioni scheda MT2USB



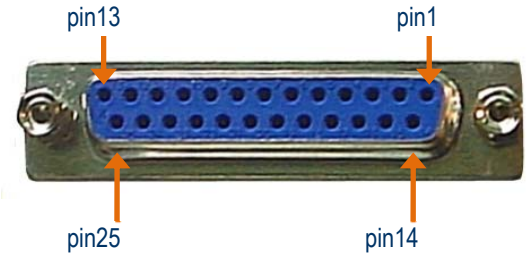
BOX

A richiesta, sono disponibili box completi di alimentatore all'interno dei quali trova alloggio la scheda MT2USB. In Figura 15 è mostrato un esempio di tali box.

a) b)

Figura 15: a) fronte e b) retro del box per MT2USBMS; analogo è il box per MT2USB.

Le dimensioni di larghezza, altezza e profondità del box sono pari rispettivamente a 158 mm, 85 mm e 170 mm. Sul retro vi è la presa che permette l'allacciamento del dispositivo alla rete elettrica (230Vac, 50/60Hz). Sul pannello frontale trovano alloggiamento l'interruttore generale per l'accensione e lo spegnimento del sistema, il connettore USB per il collegamento con il PC, un LED che indica l'avvenuta connessione e due connettori standard DB25 da 25 *pin* ciascuno per il collegamento con i due motori. Il *pinout* del connettore a 25 poli è il seguente:



DB25 connector pin	Motor pin
1	Alimentazione positiva (5Vdc, senza limitazione di corrente) per un eventuale logica di rilevamento di corsa negativa.
2	Alimentazione positiva per LED a infrarosso (per rilevamento ottico fine corsa).
3	Ingresso del sensore di fine corsa.
4	GND.
5	Alimentazione negativa per LED a infrarosso (per rilevamento ottico fine corsa).
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: tensione positiva (limitata dall'impiego di una resistenza)
15	AUX OUT: open-collector verso GND.
16	N.C.
17	N.C.
18	N.C.
19	N.C.
20	N.C.
21	N.C.
22	Fase motore B+
23	Fase motore B-
24	Fase motore A+
25	Fase motore A-

Per scongiurare malfunzionamenti del dispositivo, si consiglia di effettuare il collegamento USB solo dopo aver alimentato e acceso il box.

SOFTWARE DEMO

Il software MT2_Demo è concepito per testare mediante PC le funzionalità di tutte le schede di controllo della serie MT2. Il pannello di controllo virtuale (visualizzato in Errore. L'origine riferimento non è stata trovata. 14) può essere utilizzato in maniera molto intuitiva, consentendo così un semplice e rapido apprendimento del suo funzionamento.

INSTALLAZIONE

Per l'installazione su PC lanciare in esecuzione il file "Setup.exe" e seguire le indicazioni a video. Di default il file "MT2_Demo.exe" verrà installato nel percorso "C:\Program Files\MT2_Demo".

ESECUZIONE

Eseguire il file "MT2_Demo.exe". A video appare il pannello di controllo virtuale riportato in Figura 16:

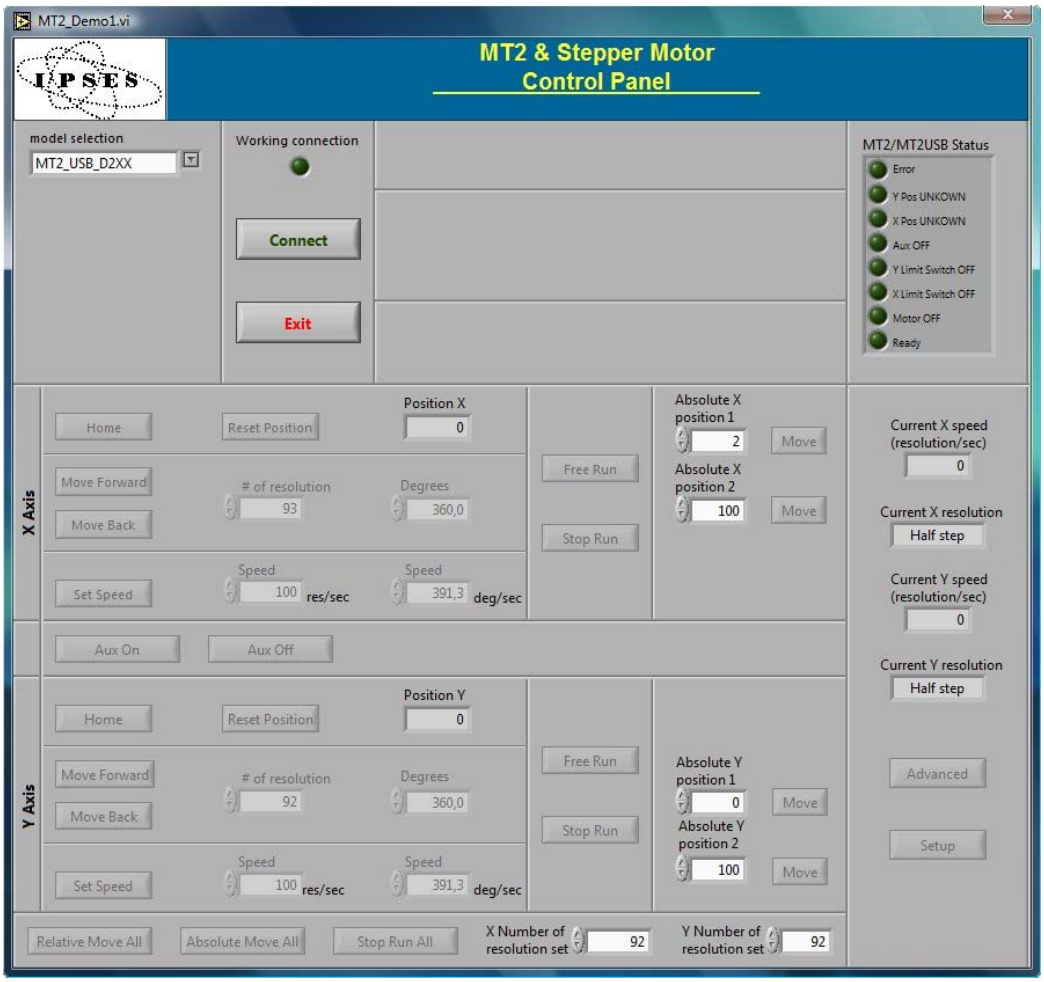


Figura 16: pannello di controllo virtuale.

CONNESSIONE CON L'UNITÀ MT2

Condizione da soddisfare per stabilire il dialogo con l'unità MT2 disponibile è la scelta opportuna della versione nel campo model selection. Per le versioni seriale (VCP) ed ethernet verranno visualizzati dei campi nei quali dovranno essere inseriti i parametri di

connessione: *COM port* e *IP address*, *IP port* e *password* rispettivamente, come mostrato in Figura 17. Non è possibile gestire contemporaneamente schede di differenti versioni.
Il pulsante Connect avvia la connessione verso i dispositivi collegati, il pulsante Exit termina l'applicativo.

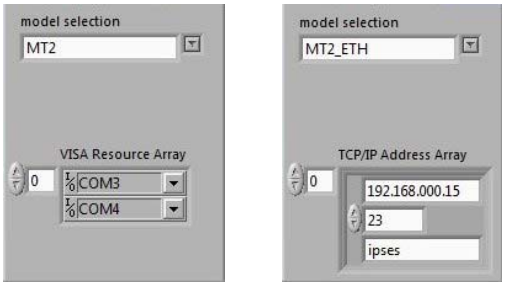


Figura 17: a) campo di selezione parametri per comunicazione seriale (VCP) e b) campo di selezione parametri per comunicazione ethernet

Se la procedura di connessione va a buon fine viene acceso il LED Working connection e vengono visualizzati il menù di selezione del dispositivo corrente S/N list e il campo Info device che riporta le informazioni circa la versione *firmware* e il *serial number* della scheda attiva, come mostrato in Figura 21. Inoltre vengono attivati tutti i comandi del pannello.
Se la versione cui ci si connette è un dispositivo micro-stepper, sul pannello di controllo appaiono i controllori per definire la risoluzione di movimento dei motori dei due assi, impostabile con Set Resolution, come mostrato in Figura 18.

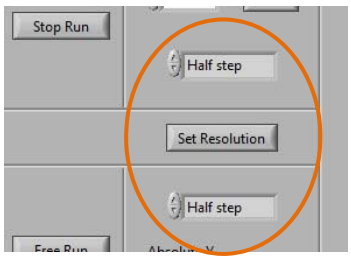


Figura 18: controllori per impostare la risoluzione dei motori

Per tutti i modelli ethernet vengono visualizzati anche i campi e i comandi che consentono di modificare i parametri di accesso del protocollo TCP/IP, ovvero *address*, *port* e *password*, come mostrato in Figura 19.

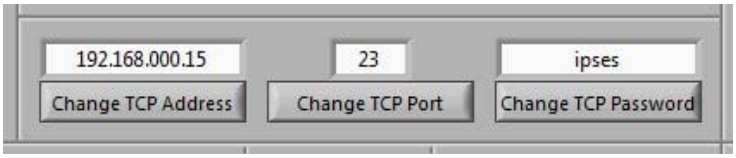


Figura 19: controlli per la modifica dei parametri di connessione del protocollo TCP/IP

FUNZIONALITA'

Il pannello di controllo virtuale è strutturato in modo da rendere immediata la comprensione delle funzioni implementate.

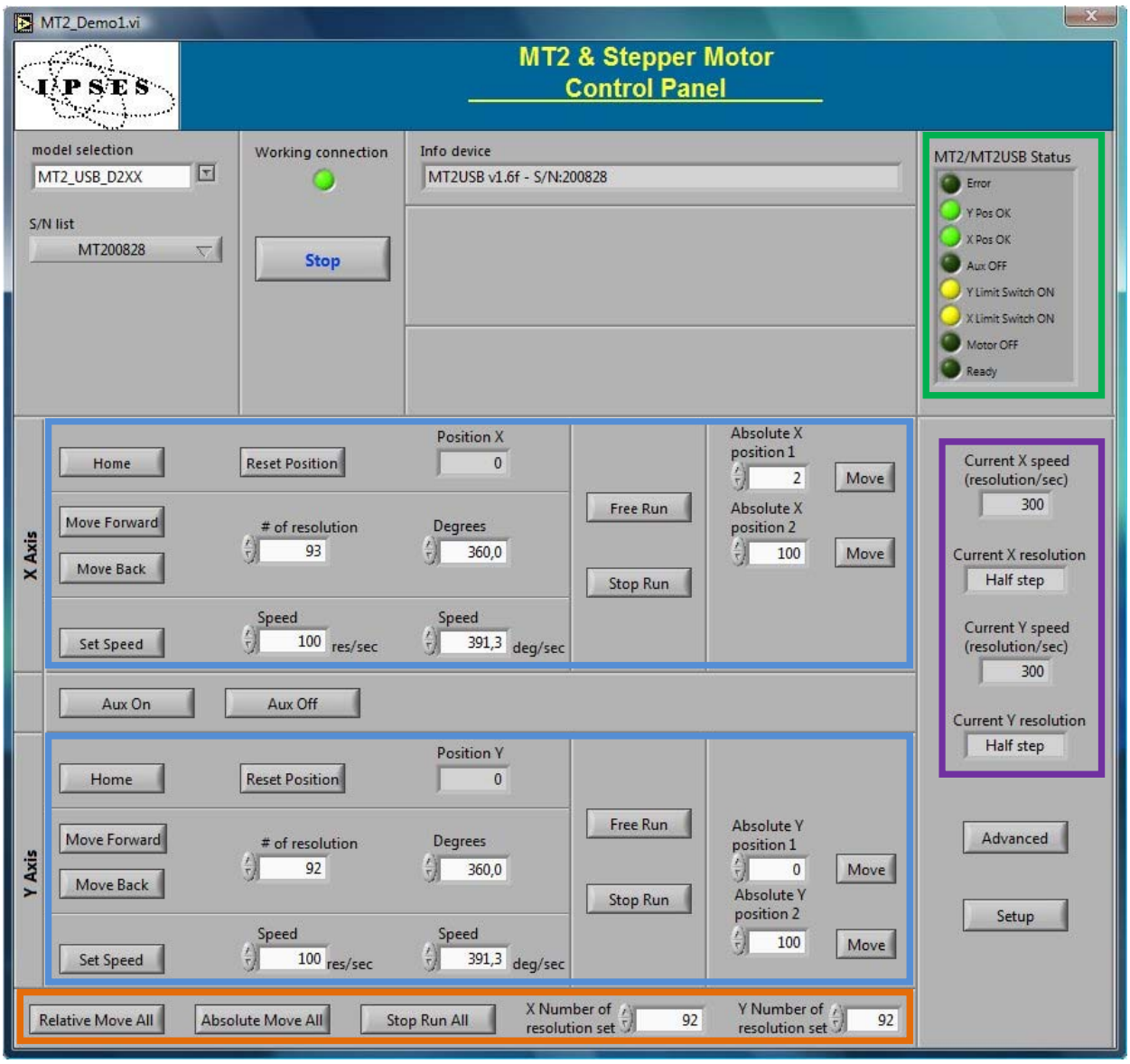


Figura 20: pannello attivo

Le zone evidenziate in azzurro in Figura 20 includono i controlli e i comandi di settaggio e movimentazione relativi a ciascun asse. Consideriamo i controlli e i comandi riferiti all'asse X.
Il comando Home attiva la richiesta di portare l'asse X all'*home position* (fine corsa): per le versioni seriale ed USB, ad eccezione delle versioni ethernet, per la durata dell'esecuzione di tale comando non è possibile inviarne altri. Durante la ricerca del fine corsa è però possibile modificare il dispositivo attivo (selezionato dal menù S/N list) oppure abortire l'applicazione con il tasto Quit che apparirà in sostituzione del tasto Stop. Il comando Reset position fa sì che la posizione corrente venga identificata con lo spostamento nullo (Position X, lo spostamento del motore in direzione X è posto a zero). L'entità dello spostamento relativo lungo l'asse è assegnata o con # of resolution in numero di risoluzioni o con Degrees in gradi: i comandi Move Forward e Move Back azionano il movimento rispettivamente in avanti e all'indietro. Il comando Free Run avvia il moto perpetuo del motore X, mentre Stop Run ne arresta il movimento. La velocità (Speed) può essere definita in risoluzioni al secondo (resolution/sec) o, analogamente allo spostamento, in gradi al secondo. Il comando Set Speed aggiorna i valori di velocità impostati.

Absolute X position 1 e Absolute X position 2 determinano l'entità di due spostamenti assoluti, riferiti alla posizione zero, avviati con il comando Move.
Quanto appena descritto è valido anche per l'asse Y.
La zona inferiore del pannello, evidenziata in arancio, include i comandi per la movimentazione contemporanea, in termini assoluto e relativo, di entrambi gli assi.
I pulsanti Aux On e Aux Off attivano e disattivano l'uscita ausiliaria della scheda.
Gli indicatori posizionati nella zona evidenziata in viola riportano le impostazioni correnti della velocità e del passo di risoluzione dei motori per entrambi gli assi.

La generazione di errori da parte del programma viene visualizzata come mostrato in Figura 21, dove è riportato come esempio l'errore causato dalla mancata comunicazione con il dispositivo.

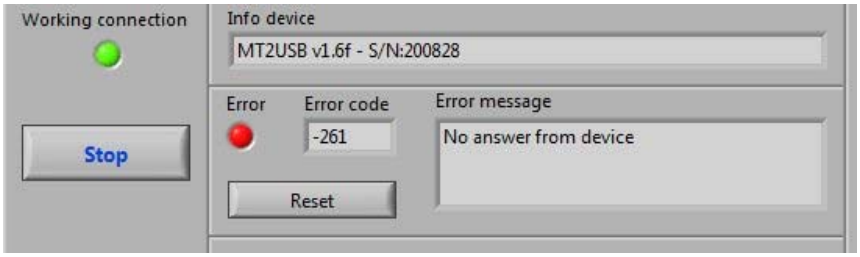


Figura 21: messaggio di errore

Il ripristino della funzionalità dell'applicativo dipende dalla presa visione del messaggio di errore dell'utente mediante il pulsante di Reset.

Il pulsante di Setup apre una finestra (mostrata in Figura 22) nella quale è possibile impostare nei campi X step/revolution ratio e Y step/revolution ratio il numero di passi per una rotazione completa dei rotori dei motori posti sull'asse X e Y. I valori vengono automaticamente aggiornati alla chiusura della finestra, con il pulsante Exit Setup. Set default ripristina i valori di default.

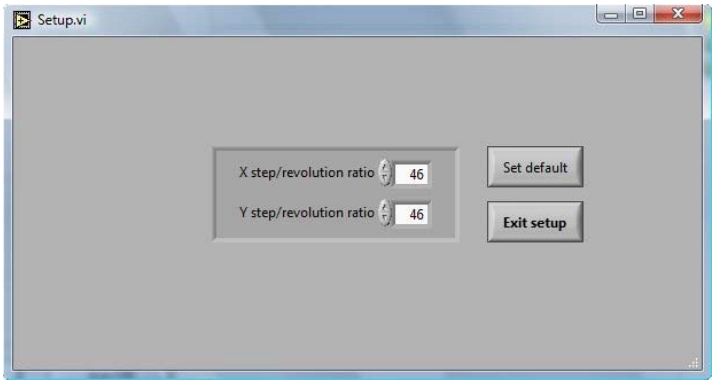


Figura 22: finestra di setup

Il pulsante Advanced lancia in esecuzione l'omonima *subroutine* (mostrata in Figura 23) che permette all'utente di vedere come viene gestito a basso livello il dialogo con il dispositivo attivo. L'istruzione scelta in command e i parametri associati in par X e par Y vengono comunicati con Run Send; parimenti Run Read consente di ricevere negli opportuni campi (info device, par X, par Y, o status) risposte alle interrogazioni selezionate in question.

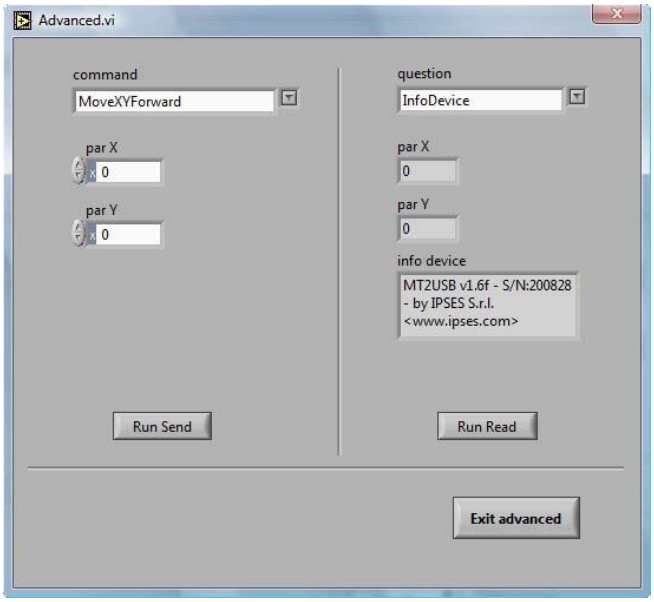
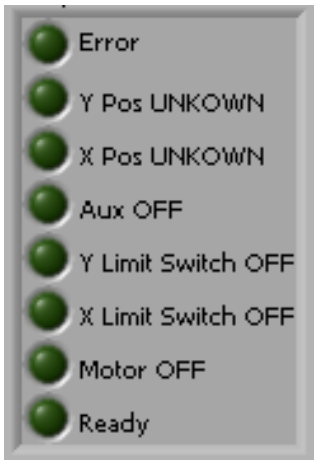
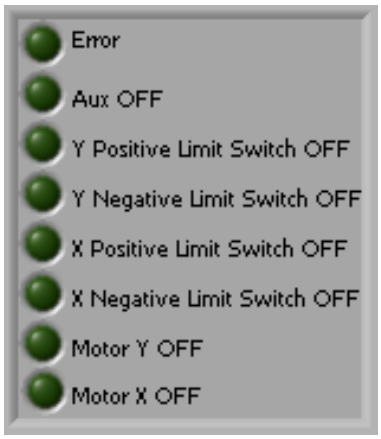


Figura 23: finestra della subroutine *Advanced*

La porzione del pannello di Figura 20 evidenziata in verde mostra i LED fittizi che informano circa lo stato della scheda: quando si illuminano riportano il verificarsi della condizione indicata dalla scritta al loro fianco (in tal evento, OFF diventa ON se presente). La descrizione di questi LED è differente per le versioni *Ethernet*: la seguente Errore. L'origine riferimento non è stata trovata. 22 mostra il confronto tra i LED di stato per i protocolli seriale e USB e i LED di stato per il protocollo *Ethernet*.





Error	Stato di errore raggiunto dal dispositivo
Y Pos UNKNOWN	Spostamento motore asse Y sconosciuto
X Pos UNKNOWN	Spostamento motore asse X sconosciuto
Aux OFF	Attivazione uscita ausiliaria
Y Limit Switch OFF	Attivazione segnale di fine corsa asse Y
X Limit Switch OFF	Attivazione segnale di fine corsa asse X
Motor OFF	Motori in movimento
Ready	Dispositivo pronto

Error	Stato di errore raggiunto dal dispositivo
Aux OFF	Attivazione uscita ausiliaria
Y Positive Limit Switch OFF	Attivazione segnale di fine corsa positiva asse Y
Y Negative Limit Switch OFF	Attivazione segnale di fine corsa negativa asse Y
X Positive Limit Switch OFF	Attivazione segnale di fine corsa positiva asse X
X Negative Limit Switch OFF	Attivazione segnale di fine corsa negativa asse X
Motor Y OFF	Motore asse Y in movimento
Motor Y OFF	Motore asse Y in movimento

a) b)

Figura 24 : LED di informazione sullo stato delle schede con protocollo a) seriale e USB; b) Ethernet

Se la scheda raggiunge uno stato d'errore questo viene visualizzato come nell'esempio in Figura 25. La presa visione dell'errore e il ripristino delle condizioni operative avviene tramite il pulsante Close error.

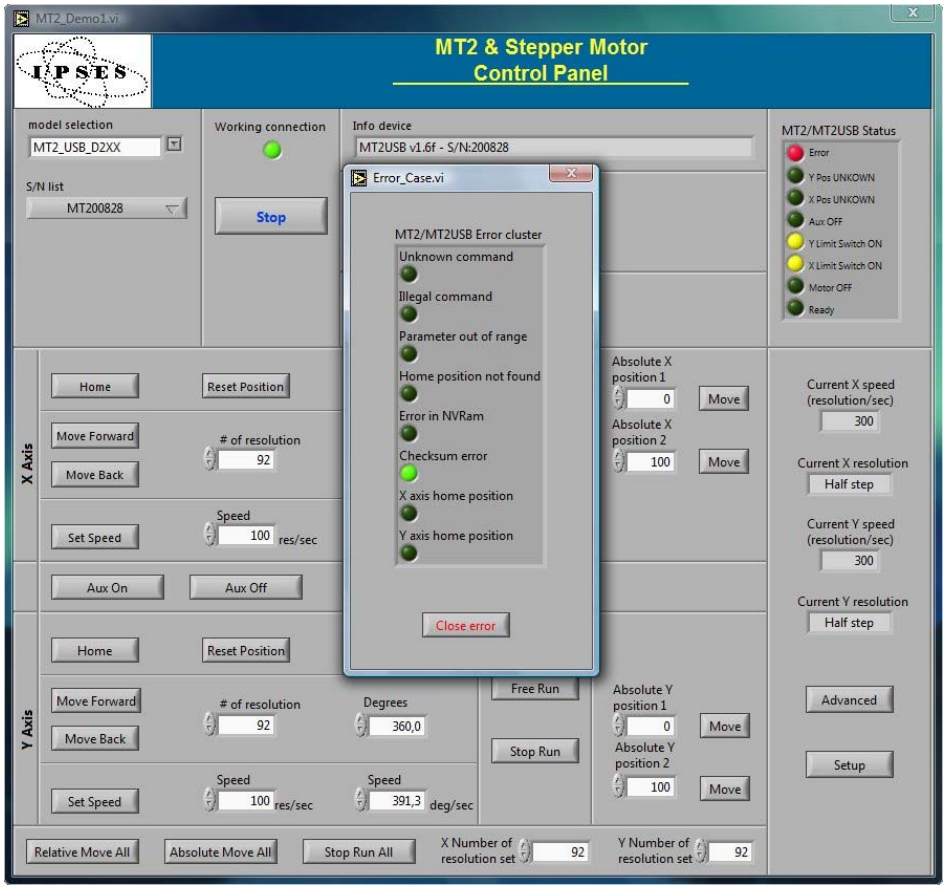
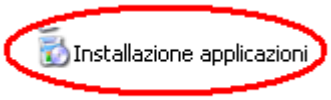


Figura 25: esempio di messaggio di errore dal dispositivo

RIMOZIONE

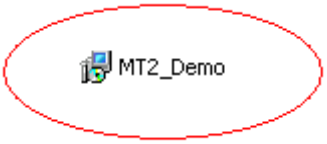
Affinché il *software* sia correttamente rimosso, procedere come indicato.

1) Ciccare sull'icona del *Desktop* "Risorse del computer" e scegliere "Pannello di controllo".



2) Nell'elenco di risorse cliccare su "Installa applicazioni".

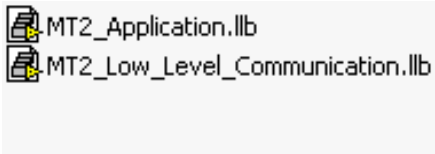
3) Nell'elenco di applicazioni installate scegliere "MT2_Demo" e procedere alla rimozione con "Cambia/Rimuovi"



LIBRERIA LABVIEW

L'ambiente di sviluppo grafico *LabVIEW* è un linguaggio di programmazione che offre la possibilità di realizzare applicazioni per il controllo da remoto di tutte le schede presentate. Il controllo di ciascuna scheda è attuabile con le undici funzioni della libreria MT2_Library realizzate in *LabVIEW7.1*: con queste funzioni è possibile implementare qualsiasi applicativo senza dover conoscere tutti i dettagli del protocollo di comunicazione, rendendo più veloce e più facile lo sviluppo.

I livelli di sviluppo delle funzioni della libreria sono due. Il file *MT2_Low_Level_Communication.llb* contiene le sei funzioni che governano la connessione con i dispositivi. *MT2_Application.llb* invece contiene le funzioni a livello superiore costruite con l'impiego delle precedenti: queste funzioni a livello superiore consentono l'assegnamento dei comandi accettati dal dispositivo. *MT2_Application.llb* è sufficiente per lo sviluppo di applicazioni, mentre *MT2_Low_Level_Communication.llb* si può utilizzare per massimizzare le prestazioni.



	Funzione	Proprietà
MT2_Low_Level_Communication.llb	*Close_Device.vi*	Chiude la connessione stabilita con un protocollo di comunicazione tra quelli disponibili.
	Open_Device.vi	Avvia la connessione con un protocollo di comunicazione tra quelli disponibili.
	Write&Read.vi	Invia e riceve caratteri in codice ASCII.
	Write_Command.vi	Invia caratteri in codice ASCII.
	SetBitMode.vi	Setta il valore di due ulteriori uscite in configurazione *open collector* (valido solo per schede USB).
	GetBitMode.vi	Legge il valore di due ulteriori uscite in configurazione *open collector* (valido solo per schede USB).
MT2_Application.llb	*Close_dialogue.VI*	Termina la comunicazione con la scheda.
	Read.vi	Invia una richiesta al dispositivo e ne legge la risposta.
	Send_Command.vi	Impartisce i comandi implementati sul dispositivo.
	Start_dialogue.vi	Avvia una sessione di dialogo con la scheda.
	Return_Info.vi	Stila l'elenco dei *serial number* dei dispositivi connessi.

La libreria MT2_Library è dotata di un help, *MT2_Help.chm*, che spiega in dettaglio l'uso di ciascuna funzione in essa contenuta. *MT2_Help.chm*, il cui contenuto informativo è accessibile anche dall'ambiente *LabVIEW*, per ciascuna delle undici funzioni fornisce una descrizione della struttura con l'impiego di rappresentazioni grafiche che semplicemente indirizzano l'utente al loro utilizzo nel contesto in cui sono state implementate. La seguente Errore. L'origine riferimento non è stata trovata. 24 mostra l'aspetto dell'*help* delle funzioni.

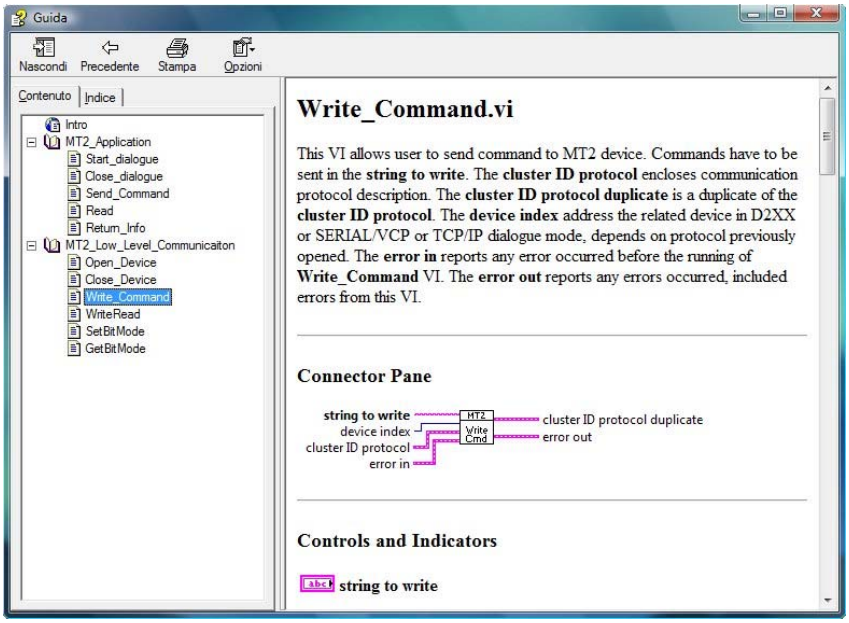


Figura 26: help delle funzioni *LabView*

La libreria MT2_Library è disponibile su richiesta.

ESEMPIO DI CONNESSIONE: HYPERTERMINAL

Un tipico esempio di connessione seriale viene rappresentato dall'HyperTerminal accessibile direttamente da tutti i PC con sistema operativo Microsoft Windows (eccetto Windows Vista per il quale il programma non è incluso negli applicativi distribuiti con il sistema operativo, ma può essere facilmente reperito in internet e installato) semplicemente richiamando dal menù "Start->Tutti i programmi -> Accessori -> Comunicazioni -> HyperTerminal" (Figura 27).

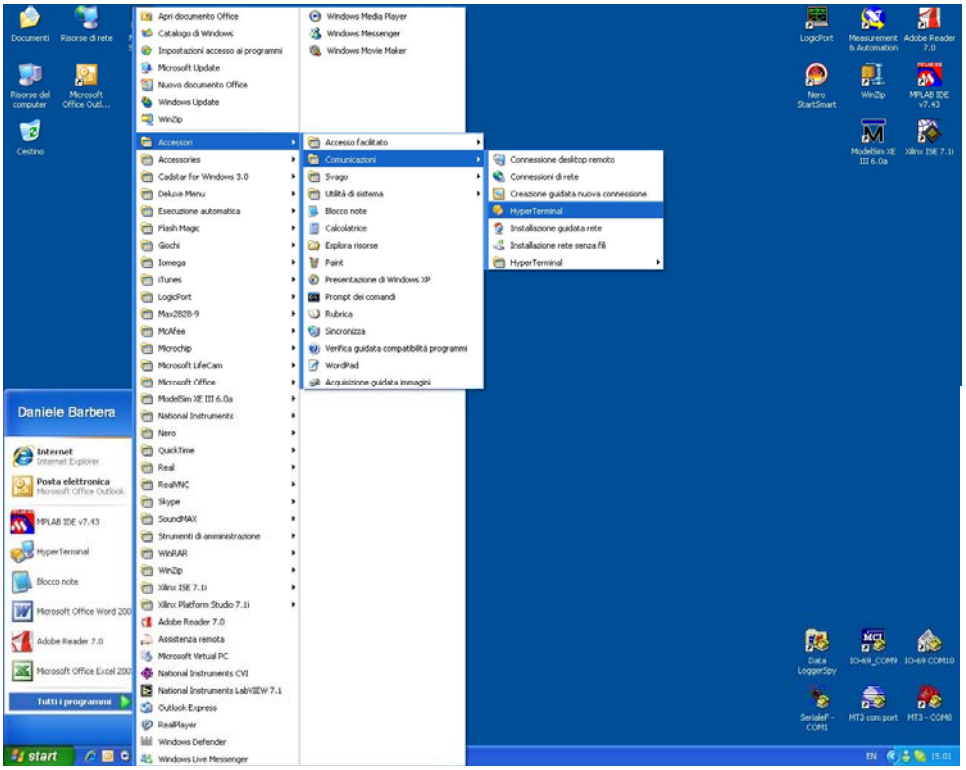


Figura 27: percorso di esecuzione di HyperTerminal

La comunicazione seriale può essere utilizzata anche con i dispositivi USB, dato che il driver rende disponibile per ogni scheda MT2 connessa una porta seriale virtuale tramite la quale si può facilmente dialogare con il dispositivo.
Una volta aperta la connessione bisogna darle un nome e parametrizzarla (come mostra l'esempio in Figura 28 e in Figura 29):

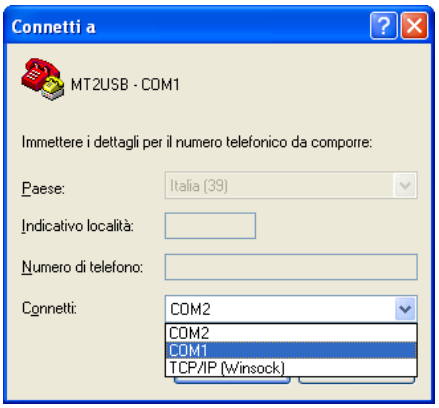


Figura 28: selezione porta di comunicazione

Figura 29: impostazione parametri di HyperTerminal

L'ultimo passaggio prima di comunicare con la scheda è quello che rende visibili i caratteri immessi dall'utente. Per far ciò basta impostare un ritardo di eco sui caratteri tramite le proprietà del programma (Figura 30 - Figura 31 -Figura 32).

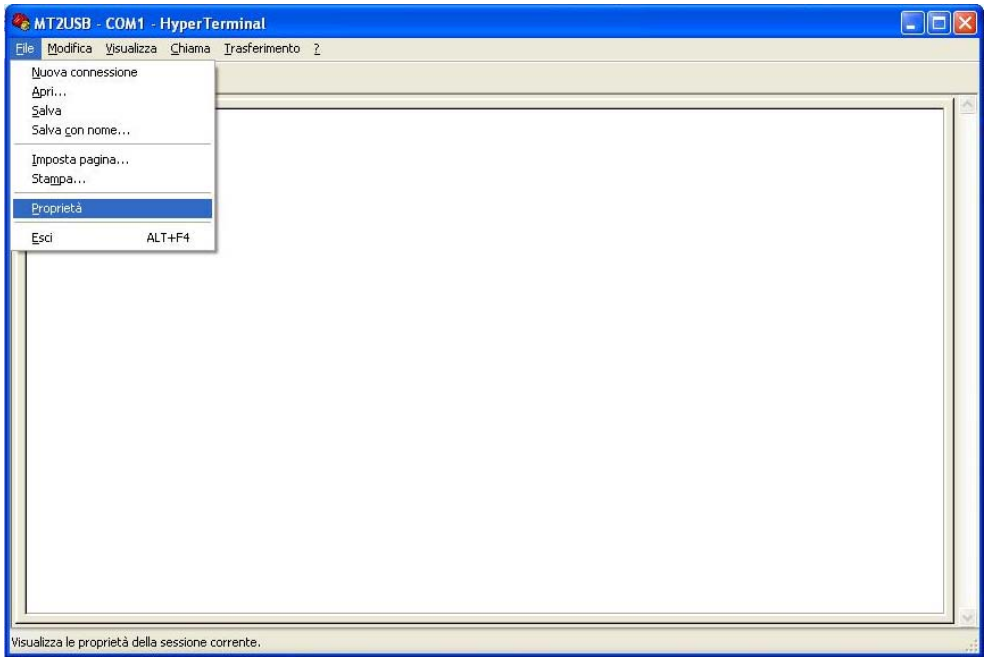


Figura 30: modifica proprietà di interfaccia utente di HyperTerminal

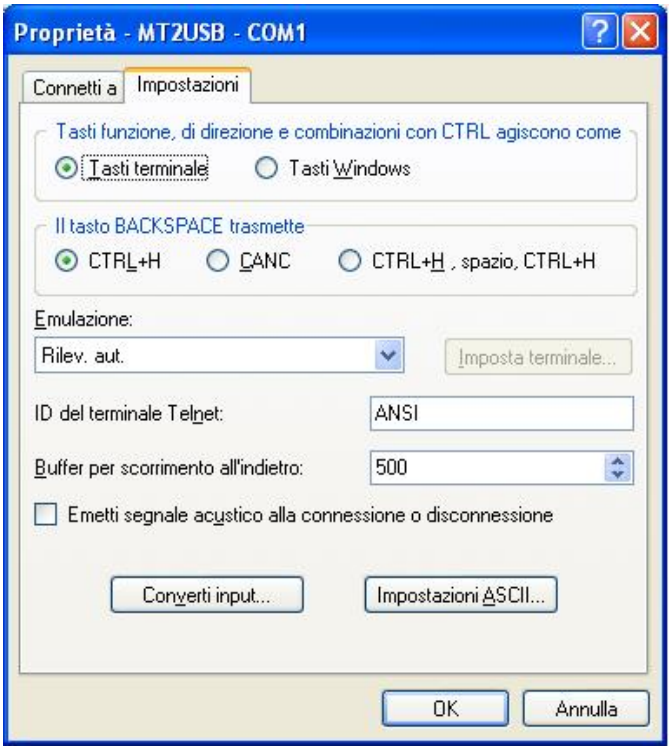


Figura 31: tramite il TAB "Impostazioni" si raggiunge l'area destinata alle "Impostazioni ASCII"

Figura 32: impostazione per la visualizzazione corretta dei caratteri ASCII

A questo punto la comunicazione tra la scheda MT2 e il PC è stabilita e segue il protocollo precedentemente descritto. Ad esempio, se si digita il carattere "?" seguito da un <ENTER> (Errore. L'origine riferimento non è stata trovata. 31), la scheda risponderà con una stringa simile a quella riportata nelle Figure Figura 33 e Figura 34:

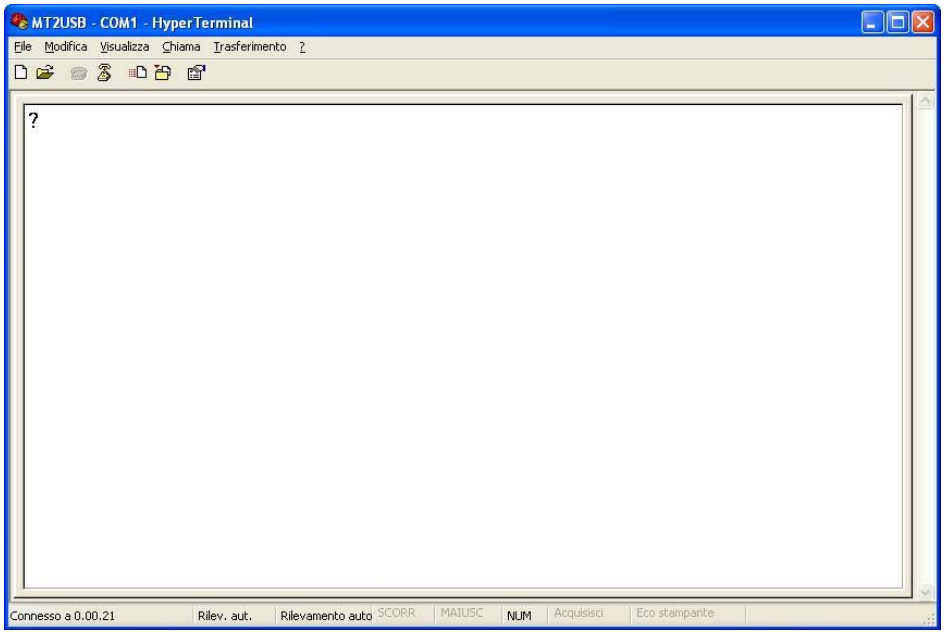


Figura 33: comando di richiesta versione firmware

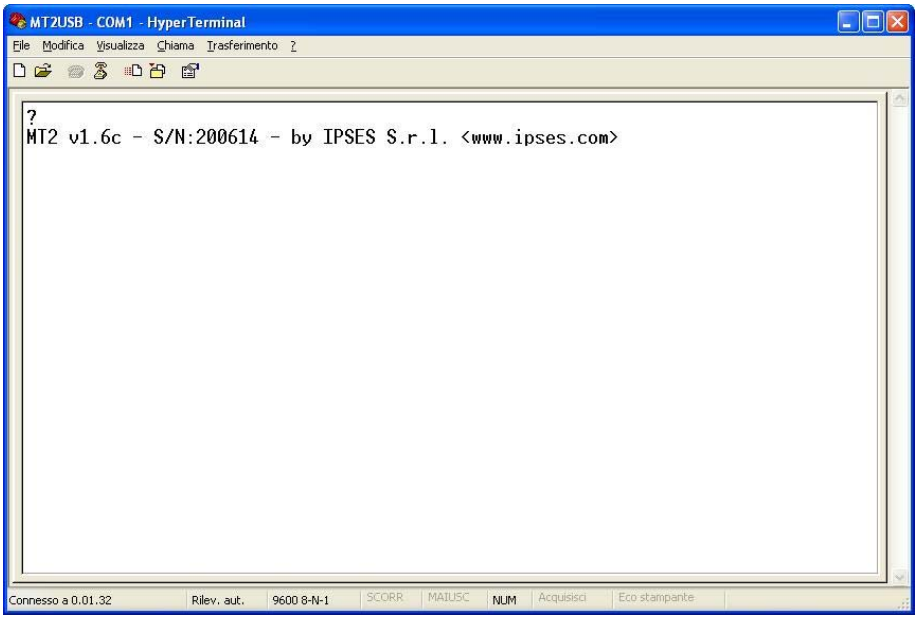


Figura 34: esempio di stringa di versione *firmware*

Digitando di seguito gli altri comandi si può quindi comunicare ed operare con la scheda MT2USB attraverso l'emulazione della porta seriale (in Figura 35 viene illustrata la richiesta del registro di stato effettuata subito dopo la richiesta firmware).

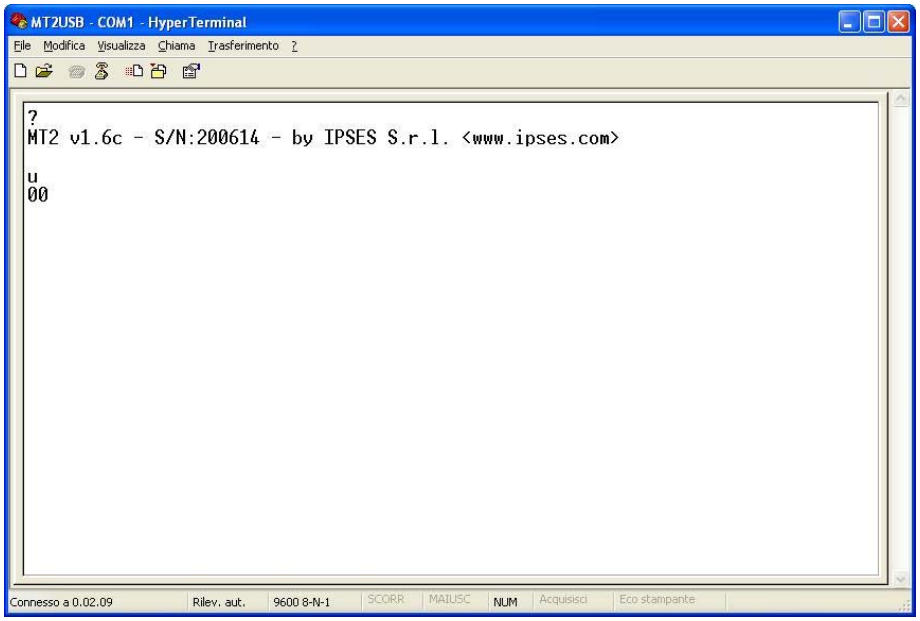


Figura 35: richiesta del registro di stato

ESEMPIO D'USO DELLA DLL D2XX

L'esempio riportato di seguito apre la comunicazione con il dispositivo di indice 0 e configura i parametri di trasmissione, legge la versione del *firmware* e il *serial number* (comando "?"), invia il comando di movimentazione del solo asse X (2000 passi) e, dopo aver letto lo stato (comando "U"), chiude la comunicazione.

```
// 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);
```

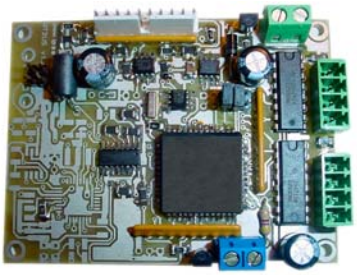
CODICE PRODOTTI

Codice	Descrizione
MT2USB	Scheda controllo a 2 assi, con interfaccia USB
MT2USB-box	Box per controllo di 2 assi con interfaccia USB e alimentazione 230V (contiene scheda MT2USB)
MT2Library	Libreria per LabVIEW 7.1 (e versioni successive) utilizzabile con tutte le schede della serie MT2
MC-connect	Bundle di 2 connettori MC a vite (per motori) + 2 connettori AMP per sensori fine corsa
USB-A-B	Cavo USB per connessione delle schede e del box USB
USB-A-B-ill	Cavo USB per connessione delle schede e del box USB, con terminazione illuminata

ALTRE VERSIONI DISPONIBILI

IPSES è in grado di realizzare **versioni personalizzate** di questo strumento, per venire incontro a qualsiasi esigenza del cliente.
In particolare è possibile richiedere versioni del prodotto di qualsiasi **dimensione** (in modo da poter essere facilmente integrato in ogni sistema meccanico) e con **protocollo di comunicazione** personalizzato.
La scheda MT2USB è disponibile anche nella versione MT2USBMS con controllo *microstepper* fino a 1/8 di passo.
Sono inoltre disponibili sistemi di controllo assi dotati di interfaccia seriale RS232 (si vedano i sistemi MT2 e MT2MS) e di interfaccia **Ethernet** (si vedano i sistemi MT2ETH e MT2ETHMS). E' inoltre disponibile una versione con interfaccia USB e controllo PWM in corrente sino a 3A (MT2HC). Per ciascuna versione è possibile richiedere la configurazione in box. A richiesta possiamo sviluppare sistemi con qualsiasi interfaccia di comunicazione definita dal cliente.

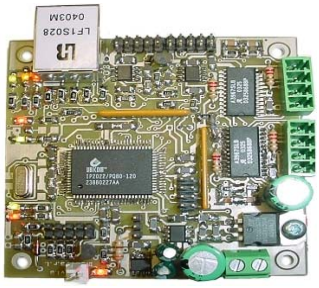
Per maggiori informazioni consultare il sito internet *http://www.ipses.com.*



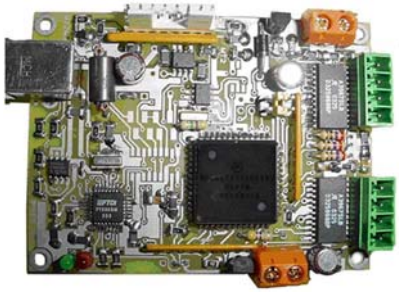
MT2

MT2MS

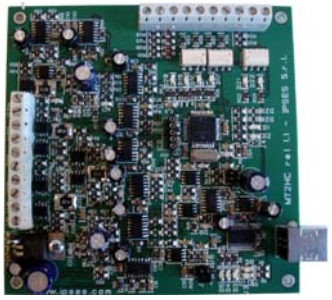
MT2ETH



MT2ETHMS



MT2USBMS



MT2HC

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

CONTATTI

IPSES S.r.l. si occupa dell'ideazione e della commercializzazione di strumenti elettronici e scientifici. La progettazione personalizzata consente di rispondere alle diverse esigenze di chi ricerca sistemi *embedded* dedicati ad applicazioni specifiche.

IPSES si avvale di uno staff con pluriennale esperienza nel settore. L'aggiornamento continuo e l'evoluzione costante rendono IPSES un'azienda all'avanguardia, capace di unire il dinamismo di una giovane impresa con la professionalità e l'affidabilità di personale qualificato.

IPSES S.r.l.

Sede operativa e centro di sviluppo:
via Suor Lazzarotto, 10
20020 Cesate (MI)
Italy

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

SISTEMA DI GESTIONE CERTIFICATO
ITALCERT
UNI EN ISO 9001

INFORMAZIONI PER IL SUPPORTO TECNICO

I nostri tecnici possono essere contattati ai seguenti recapiti:

Telephone	:	++39 02 99068453
		++39 02 320629547
Fax	:	++39 02 700403170
Email	:	support@ipses.com

RAPPORTO PROBLEMATICHE

Il modulo nella seguente pagina permette di raccogliere i dati necessari ad una corretta ricerca del problema eventualmente evidenziatosi. In caso di problemi questo modulo può essere copiato e inviato via fax, o in alternativa, allegato ad una mail.

ENGINEERING PROBLEM REPORT

Problem describer

Name			IPSES s.r.l. Via Suor 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

(Codice prodotto MT2USB REV. 01.01.0002)

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