

MiniStep

Azionamento stepper per PC/PLC con Modbus / protocollo di testo semplice su RS-485

Indice

Introduzione.....	3
Alimentazione.....	4
LED di stato.....	4
Uscite: connessione motore e segnale Ready/BUSY.....	4
Ingressi digitali (24 volt).....	5
Interfaccia RS-485 e microinterruttori.....	5
Dip switch 3 e 4.....	6
Configurazione di Ministep.....	7
Reset della porta RS-485.....	7
Impostazione dei parametri di comunicazione.....	7
Impostazione dell'indirizzo Slave.....	7
Impostazione del ritardo di risposta.....	8
Impostazione del timeout di Modbus.....	8
Salvataggio permanente delle impostazioni.....	8
Altre impostazioni (via Modbus).....	8
Usare Ministep da un PC o PLC.....	9
Modo Passo/Direzione (Step/Direction).....	9
Modo RS-485.....	9
Watch-dog interno (controllo di sicurezza).....	9
Interfaccia Modbus.....	10
Ingressi digitali, latenze e contatori.....	10
Condizionamento degli ingressi.....	11
Contatori sugli ingressi.....	11
Altri Discrete Inputs.....	11
Input Registers (parole di sola lettura).....	12
Registri Single Coil.....	12
Registri Holding Registers.....	13
Procedura per muovere il motore.....	14
Primo, impostare i parametri.....	14
Secondo, assegnare la posizione (quota) all'asse.....	14
Terzo, comandare i movimenti.....	14
Anatomia di un movimento.....	15
La funzione Modbus Collect.....	16
Protocollo (comandi) di testo semplice.....	17
Indirizzamento slave.....	18
Comandi di lettura (modo testo).....	18
Device.....	18
Address.....	18
Flags.....	18
Valim.....	18
Xword.....	19
X1 .. X16.....	19
Xcount1 .. Xcount3.....	19



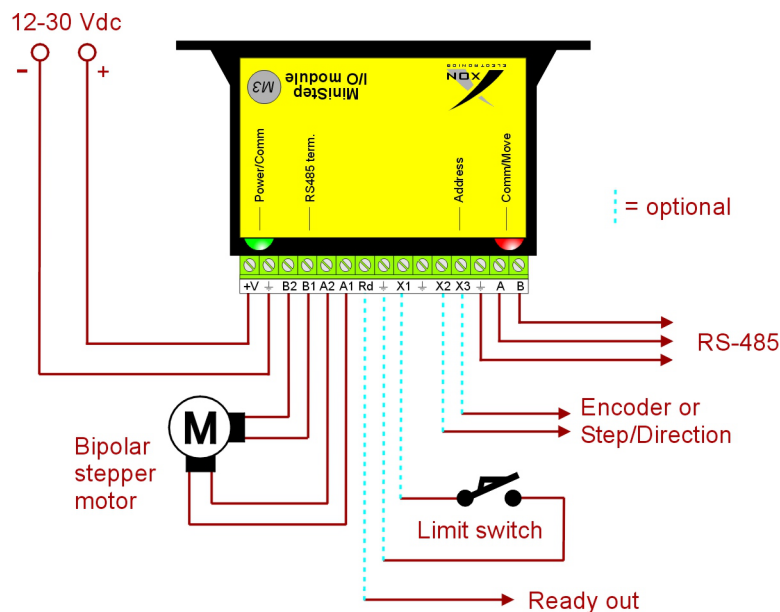
Yword.....	19
Y1 .. Y16.....	19
WDTtime.....	19
X1latup .. X3latup.....	19
X1latdn .. X3latdn.....	19
Pos, Mark.....	19
Encoder.....	19
Maxspeed, Minspeed, Accel, Decel.....	19
Currhold, Currrun, Curracc, Currdec.....	20
FSspeed.....	20
MotMode.....	20
Comandi di scrittura parametri (modo testo).....	20
Address, Serline, RdelayT, RdelayM, Wconf.....	20
Yword.....	20
Y1 .. Y16.....	20
WDTtime.....	20
Wflags.....	20
Xcount1 .. Xcount3.....	20
X1latup .. X3latup.....	21
X1latdn .. X3latdn.....	21
Y1pulse, Y2pulse.....	21
Mark.....	21
Pos.....	21
Preset.....	21
Encoder.....	21
Maxspeed, Minspeed, Accel, Decel.....	21
Currhold, Currrun, Curracc, Currdec.....	21
FSspeed.....	21
MotMode.....	21
Muovere il motore con comandi di testo semplice.....	22
Operazioni su Y1 .. Y16.....	22
Lettura degli stati da X1 .. X16.....	23

Introduzione

Ministep è un azionamento per motori stepper bipolari con interfaccia RS-485 (protocollo Modbus e testo semplice). Dispone anche di I/O per facilitare l'utilizzo. I movimenti del motore possono essere comandati con la RS-485, ad alto livello (cioè, posizionamenti), oppure usando i segnali Passo/Direzione collegati alla morsetteria.

Le caratteristiche di Ministep sono:

- Alimentazione da 12 a 34 volt C.C.
- Fino a 3A continui di corrente per il motore (6A di picco)
- Passo intero, mezzo passo, microstepping fino a 1/16
- Correnti configurabili per fermo, marcia, accelerazione e decelerazione
- Profili di velocità pienamente programmabili (crociera, rampa)
- Comandi per posizionamenti assoluti, relativi, e altre funzioni avanzate
- 3 ingressi digitali per uso generale, fine corsa o encoder; 1 uscita digitale (Ready/Busy)
- Configurabile con selettore rotante, microinterruttori, e memoria flash interna
- Watch-dog interno per assicurare affidabilità e sicurezza
- 2 LED di stato
- Circuito elettronico completamente resinato in contenitore plastico con fori di fissaggio



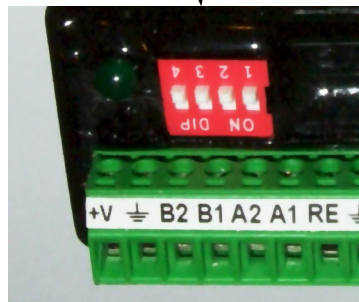
Esempio di connessione. Le linee tratteggiate indicano collegamenti opzionali.

L'interfaccia RS-485 può essere usata per controllare pienamente il motore, senza bisogno di connessioni opzionali – ma possono essere usate per interfacciare altri dispositivi come fine corsa o pulsanti. Altrimenti, la RS-485 può essere usata per impostare i parametri e memorizzarli nella memoria interna: da quel momento il motore può essere comandato con Passo/Direzione senza più bisogno dell'interfaccia RS-485.

Alimentazione



Il dispositivo va alimentato con una tensione continua da 10 a 34 volt. Il consumo di corrente è di 150 mA per il solo dispositivo, più la corrente erogata al motore (fino a 6A di picco).



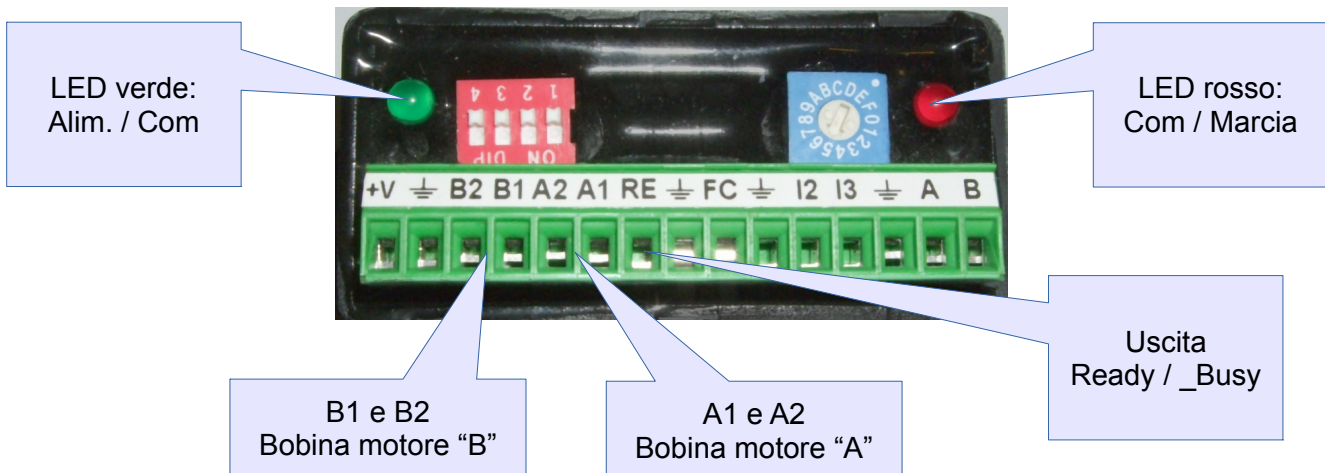
+V è il terminale del positivo dell'alimentazione.

Il simbolo di Terra è il negativo comune.

LED di stato

Il Ministep ha due LED (vedi immagine sotto). Il LED verde a sinistra è acceso per segnalare la presenza dell'alimentazione; si spegne brevemente alla ricezione di un comando da seriale; inoltre, lampeggia velocemente per segnalare l'intervento del watch-dog. Il LED rosso a destra si è acceso quando il motore sta girando; emette un lampeggio positivo o negativo quando il modulo manda una risposta al master.

Uscite: connessione motore e segnale Ready/BUSY



Il segnale Ready/BUSY è un'uscita open collector, con impedenza di 100 ohm. Viene chiuso (0 volt) per indicare che il Ministep è impegnato (busy) – per esempio, durante un posizionamento; viene rilasciato aperto quando non ci sono operazioni in corso. Può essere usato, ad esempio, per rilevare velocemente il momento in cui un posizionamento è terminato.

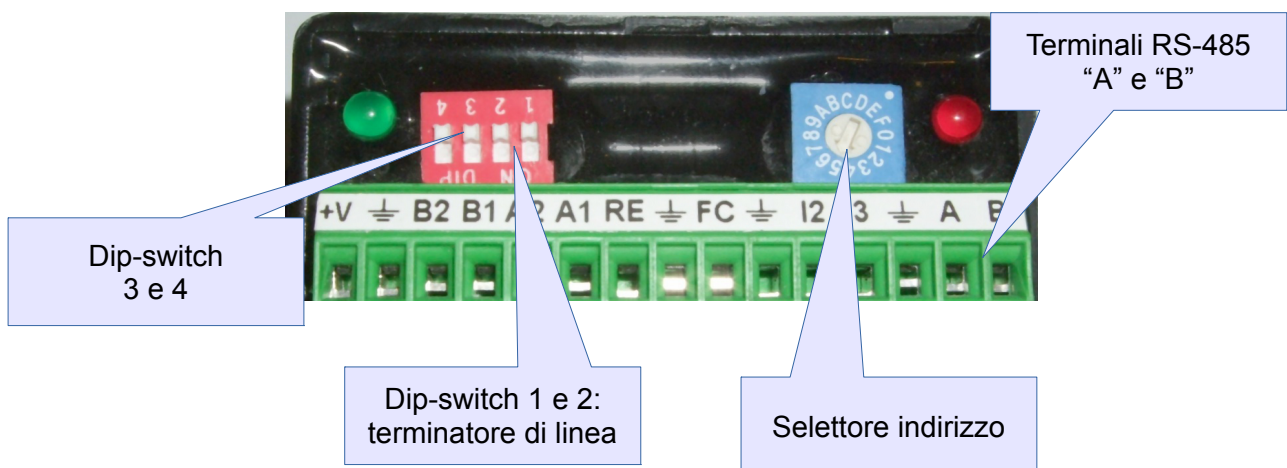
Ingressi digitali (24 volt)



I tre ingressi digitali (FC, I2, I3, anche chiamati X1..X3) hanno una impedenza d'ingresso di 5 Kohm e un circuito di protezione per i componenti interni. Devono essere pilotati con una tensione positiva per attivarli (lasciati liberi, sono in stato *off*). Le funzioni degli input sono le seguenti:

- | | |
|------------|--|
| 1. FC (X1) | Usò generale / Fine corsa (azzeramento) / Basso pilotaggio (modo Step) |
| 2. I2 (X2) | Usò generale / Canale encoder A / Comando Step (modo Step) |
| 3. I3 (X3) | Usò generale / Canale encoder B / Comando Direzione (modo Step) |

Interfaccia RS-485 e microinterruttori



La linea RS-485 è formata da due segnali bilanciati, A e B, e la relativa massa di riferimento. Usare cavo intrecciato e, se il collegamento è lungo e/o scorre in ambienti disturbati, impiegare un cavo intrecciato e schermato con calza messa a terra (NON a massa). A ogni estremità del cavo deve essere applicata una resistenza di terminazione – i valori delle resistenze devono corrispondere; il dispositivo Ministep contiene al suo interno due resistenze di terminazione che si abilitano tramite gli interruttori 1 e 2, secondo la tabella seguente:

Switch 2	Switch 1	Terminazione	Commento
OFF	OFF	<i>Assente</i>	Configurazione multidrop, quando Ministep NON è l'ultimo dispositivo topologico del bus.
OFF	ON	680 ohms	Terminazione per linee corte (<50 metri)
ON	OFF	330 ohms	Terminazione per lunghezze medie
ON	ON	220 ohms	Per distanze ancora maggiori (circa 1 Km)

Lo standard RS-485 indica 120 ohm come terminazione, tuttavia buoni risultati si possono ottenere anche con valori superiori, con un certo risparmio energetico. Se occorre usare 120 ohm, disabilitare le resistenze interne a Ministep e usarne una esterna.

La ruota di selezione indirizzo assegna lo *Slave address* per il protocollo Modbus. In questo protocollo l'indirizzo è obbligatorio e deve essere compreso tra 1 e 247 (il *Master* ha sempre indirizzo 0). Se sulla linea invece di Modbus si usa il protocollo di tipo testo, l'uso dell'indirizzo è opzionale (ma un indirizzo è comunque presente). L'indirizzo assegnato alla periferica è quello indicato dalla ruota, tranne che le lettere da A a F corrispondono agli indirizzi da 10 a 15; la posizione 0 assegna l'indirizzo 16.

Sulla linea seriale si possono usare, anche contemporaneamente, il protocollo Modbus RTU e il protocollo di testo semplice, come un terminale seriale. I parametri di comunicazione seriale sono normalmente 19200 baud, 8 bit di dati, parità pari e 1 stop bit (raccomandazione Modbus). Si può variare questi parametri con comandi di testo, insieme ad altre opzioni; le impostazioni poi possono essere salvate in modo permanente. Infine, è presente un meccanismo per re-impostare la porta RS-485 a un valore conosciuto; se, mentre il dispositivo è acceso, si sposta lo switch 3, la porta viene reimpostata a 19200 baud, parità pari e uno stop bit.

Dip switch 3 e 4

Il microinterruttore dip-switch 3 è usato per reimpostare i parametri di comunicazione, quando spostato mentre il Ministep è acceso; diversamente, è ignorato (ma può essere letto da RS-485).

Il dip-switch 4 deve essere impostato su ON per usare i segnali Passo/Direzione (Step/Direction) per comandare Ministep. Quando è ON, la RS-485 può ancora essere usata per impostare parametri, ma non per comandare movimenti del motore.

Configurazione di Ministep

Di fabbrica, il modulo contiene parametri “standard”, specialmente quelli di comunicazione, come raccomandato dal protocollo Modbus. Però è possibile, usando comandi di testo, modificare i parametri; dopo aver eseguito le impostazioni, è possibile e raccomandato di salvarle in modo permanente nella memoria flash (non volatile) interna.

DOPO LA MODIFICA DI PARAMETRI, ricordare di memorizzarli.

Per cambiare i parametri, connettere il modulo a un computer tramite la linea RS-485 e parametri di comunicazione corretti. Utilizzare quindi un normale emulatore di terminale per “parlare” con Ministep.

Reset della porta RS-485

I parametri di comunicazione di fabbrica sono 19200 baud, 8 bit di dati, parità pari e 1 bit di stop. E' possibile reimpostare questi valori temporaneamente, spostando il dip-switch SW3 mentre Ministep è alimentato. Questa operazione reimposta i parametri, ma alla prossima riaccensione verranno usati nuovamente i valori salvati come preferenza.

Impostazione dei parametri di comunicazione

Inviare a Ministep un comando come il seguente:

```
>SERLINE=19200E
```

(simbolo di maggiore, poi SERLINE=19200, poi lettera “E”), seguito da CR (il tasto Invio/Enter). In questo modo la porta è impostata a 19200 baud con parità pari e 1 stop bit. Invece di 19200 è possibile indicare 4800, 9600, 38400 e 57600.

Invece della lettera finale “E” (che significa parità “pari”), è possibile usare la lettera “O” (parità dispari), “N” per nessuna parità 2 stop bit, o “X” per nessuna parità e un solo stop bit.

Se nessuna lettera finale è specificata, parità e stop bit rimangono invariati.

Se il comando è corretto, Ministep risponde con “OK”. Se la sintassi o gli argomenti non sono accettabili, la risposta è “Error.”; se nessuna risposta viene ricevuta, allora i parametri di comunicazione sono errati o il comando non è stato iniziato con il simbolo di maggiore “>”. Ministep risponde usando gli stessi parametri di comunicazione usati al momento; solo dopo la risposta OK i nuovi parametri diventano effettivi.

Impostazione dell'indirizzo Slave

L'indirizzo slave di Ministep dipende dalla posizione del selettore rotante e da uno spiazzamento (offset) chiamato MBADD, memorizzato nelle preferenze, sommato al selettore. In questo modo è possibile usare qualsiasi indirizzo tra 1 e 255 (Modbus accetta solo da 1 a 247). L'impostazione di fabbrica per MBADD è zero, cosicché l'indirizzo è quello specificato dal selettore. Inviando il comando:

```
>ADDRESS=180
```

(simbolo “>”, ADDRESS=180, poi Invio/Enter) si assegna l'indirizzo 180. Ciò che accade realmente è che MBADD assume il valore 180 *meno* il valore del selettore, così resta possibile usarlo per cambiare indirizzo. E' consigliabile spostare il selettore rotante su “0” prima di usare questo comando, per facilitar successivi cambi d'indirizzo.

Impostazione del ritardo di risposta

Dato che la linea RS-485 è *half-duplex*, le trasmissioni da due dispositivi vanno fatte in tempi diversi. Ministep attende 1/100 di secondo, prima di rispondere, per dar modo al master di liberare la linea; è possibile aumentare questo ritardo usando il seguente comando:

>RDELAYT=nnn

dove nnn è un numero da 0 a 200, e rappresenta il ritardo in centesimi di secondo.

Impostazione del timeout di Modbus

Il protocollo Modbus specifica che il termine di un messaggio è segnalato da un silenzio sulla linea di almeno 28 bit, e Ministep rispetta questo valore. Se però il master ha qualche problema di latenza, e inserisce silenzi non voluti fra un carattere e l'altro, è possibile aumentare questo timeout di 28 bit di un numero di *decimi di millisecondo* compreso fra 0 e 200 inviando il comando:

>RDELAYM=nnn

dove nnn è il numero di decimi di millisecondo (0,0001 secondi) da attendere in più per interpretare la fine di un messaggio.

Salvataggio permanente delle impostazioni

Dopo aver modificato le preferenze, esse devono essere memorizzate per essere conservate quando il dispositivo è spento. Inviare il comando:

>WCONF=1

per scrivere le impostazioni nella memoria interna. Ministep risponde con “OK”, oppure “Error 1” se si verifica un errore.

Altre impostazioni (via Modbus)

Ministep salva nella memoria interna diversi altri parametri riguardanti la gestione del motore. Questi parametri sono:

Nome	Indir.	Descrizione
MAXSPEED	93	Velocità di crociera in passi/s (default 800)
MINSPEED	94	Velocità minima in passi/s (default 0)
ACC	95	Accelerazione in passi/secondo quadro (default 1600)
DEC	96	Decelerazione in passi/secondo quadro (default 1600)
CURRHOLD	97	Corrente a motore fermo (mA, default 300)
CURRRUN	98	Corrente in crociera (mA, default 1000)
CURRACC	99	Corrente in accelerazione (default 1000)
CURRDEC	100	Corrente in decelerazione (default 1000)
MODSPD	101	Velocità di passaggio a Passo Intero (default 1000)
STEPMODE	104	Configurazione del modo di pilotaggio (default 0x2082)

In ogni momento, via Modbus, è possibile memorizzare tali parametri scrivendo 1 nel Discrete Coil 162.

Usare Ministep da un PC o PLC

Modo Passo/Direzione (Step/Direction)

Con questo modo attivo, ogni fronte di discesa dell'ingresso 2 (X2) produce un *passo logico* del motore, nella direzione indicata da X3 (basso=+, alto=-). Il modo Passo/Direzione può essere comandato via RS-485 oppure portando il dip-switch 4 a ON. Il tipo di passo, insieme con gli altri parametri, può essere impostato in qualsiasi momento tramite la RS-485, oppure può essere pre-configurato una volta sola, e memorizzato nella memoria interna. I passi vengono eseguiti con corrente CURRRUN, ma la corrente CURRHOLD (ridotta) può essere selezionata applicando tensione all'ingresso X1.

Modo RS-485

Via RS-485 è possibile impostare parametri, eseguire movimenti con profili di velocità (rampe) e correnti di pilotaggio diverse in ogni fase. I movimenti possono essere indicati come posizioni assolute (quote) o incrementali, oppure rotazioni indeterminate (senza destinazione), e sono disponibili anche comandi speciali come Invio a Riposo, o Azzeramento su finecorsa. Ministep può essere interrogato per conoscere la posizione (quota) del motore, lo stato degli ingressi, o il valore dell'encoder opzionale (se connesso).

Il modulo Ministep deve essere pilotato come *slave* da un PC o un PLC che agisce da *master*. Sono utilizzabili due protocolli di comunicazione, anche contemporaneamente, sullo stesso bus o periferica (ma con i parametri di comunicazione corretti, chiaramente).

Il protocollo Modbus prevede che il master invii un comando a uno slave particolare, il quale processa il comando e rimanda al master una risposta (l'assenza di una risposta segnala un errore di comunicazione). I comandi Modbus sono principalmente di due tipi: scrittura di un valore e lettura di un valore. Per la scrittura di valori lo slave risponde con una conferma di “operazione eseguita”; per la lettura risponde con i valori richiesti. Il protocollo di testo semplice è basato sullo stesso principio: vi sono due tipi di comandi, impostazione e interrogazione, e le risposte dello slave seguono gli stessi principi. Si noti che in entrambi i protocolli le varie richieste sono indirizzate a uno slave specifico, tramite l'uso dell'*indirizzo slave*; tutte le periferiche sul bus “sentono” il comando inviato dal master ma solo lo slave con l'indirizzo selezionato esegue la richiesta e invia la risposta. Entrambi i protocolli prevedono la possibilità d'inviare un comando a *tutti* gli slave contemporaneamente (broadcast); in questo caso, tutti gli slave eseguono il comando, ma nessuno di essi deve rispondere. L'utilizzo di questi broadcast non è raccomandato, perché non è possibile determinare se tutti gli slave hanno capito ed eseguito il comando.

Watch-dog interno (controllo di sicurezza)

In un controllo di processo è desiderabile che determinati problemi vengano rilevati e che il processo venga portato in uno stato sicuro, cioè con una disposizione delle uscite non pericolosa. Il watch-dog, se abilitato, controlla che la comunicazione con il master sia continua (entro un tempo configurabile di timeout) e, nel caso che la comunicazione cessi, ferma il motore. Oltre che per la mancanza di comunicazione, il watch-dog può scattare anche per una diagnosi interna. All'accensione di Ministep il watch-dog è disabilitato.

Interfaccia Modbus

Ministep implementa le seguenti funzioni Modbus per leggere e scrivere *BIT*:

- 1 Read Coils (0XXXX)
- 2 Read Discrete Inputs (1XXXX)
- 5 Write Single Coil
- 15 Write Multiple Coils

e le seguenti funzioni per leggere e scrivere parole di 16 bit:

- 3 Read Holding Registers (4XXXX)
- 4 Read Input Registers (3XXXX)
- 6 Write Single Holding Register
- 16 Write Multiple Holding Registers
- 22 Mask Write Register

Leggendo e scrivendo i registri preposti, è possibile accedere a tutte le funzioni che Ministep implementa. Tutti gli indirizzi si contano partendo da 0 (il primo registro ha indirizzo 0).

Ingressi digitali, latenze e contatori

Gli ingressi possono essere letti come Discrete Inputs dagli indirizzi 0 .. 15. I primi tre bit corrispondono agli ingressi fisici X1, X2 e X3; i bit successivi sono invece bit virtuali usati per conoscere lo stato del modulo:

- X4 A livello alto 1 (*On*) quando il motore è in rotazione - 0 altrimenti.
- X5 *On* quando il motore non sta girando
- X6 *On* quando il motore gira in direzione “+” (la posizione s’incrementa)
- X7 *On* quando il motore gira in direzione “-”
- X8 *On* se il motore è scollegato (“Hiz” - alta impedenza): il motore non genera coppia
- X9 Segnalazione di Busy; è *On* quando Ministep sta eseguendo un comando o una rampa (non è *on* quando il motore gira in modo continuo/libero)
- X10 Segnala che il motore è alla posizione/quota 0 (Home)
- X13 *On* quando è stata rilevata una perdita di passo
- X14 *On* quando è stata rilevata una caduta dell'alimentazione d'ingresso
- X15 *On* quando è stato rilevato un consumo di corrente eccessivo (OCD)
- X16 Surriscaldamento (solo avvertimento; in seguito l'elettronica si auto protegge)

I tre ingressi X1 X2 X3 sono condizionati (manipolati), cioè sottoposti a latenza/antirimbalo. Tutti e tre dispongono di un contatore d'impulsi associato.

Condizionamento degli ingressi

Quando usati come GPIO (uso generico), i tre ingressi sono letti da Ministep una volta al millisecondo, e le variazioni riflesse su X1..X3 con una latenza programmabile per i fronti sia di salita che di discesa. XLATUP è la latenza per i fronti di salita, e XLATDN per i fronti di discesa. Queste latenze filtrano i disturbi, fanno l'antirimbando, e possono anche allungare gli impulsi. Supporre che un pulsante P1 normalmente aperto sia connesso a X1; quando P1 è pigiato, una tensione viene applicata a X1; quando rilasciato, X1 è a 0. Su una linea molto disturbata, un breve picco potrebbe azionare l'ingresso; impostando però una latenza di 20 ms su XLATUP, disturbi più brevi di 20ms vengono cancellati. Una latenza di 20ms per il fronte di discesa può eliminare i rimbalzi di contatti sporchi. Infine una latenza di discesa molto grande (maggiore del tempo di scansione del PLC), come 300 ms, rende il PLC capace di leggere impulsi più corti del suo stesso ciclo di scansione (l'impulso verrebbe ignorato se cominciasse e finisse tra una scansione e l'altra).

I tempi di latenza, in millisecondi, risiedono negli Holding Registers agli indirizzi 19, 20 e 21 per i fronti di salita, e agli indirizzi 35, 36 e 37 per quelli di discesa. Questi valori, che vanno da 0 a 65535 (16 bit), all'accensione sono di 3 millisecondi.

Contatori sugli ingressi

Ministep dispone di tre contatori di fronti di salita, uno per ogni ingresso. Questi contatori sono leggibili e scrivibili negli Holding Registers agli indirizzi 11, 12 e 13.

Altri Discrete Inputs

La tabella seguente elenca alcuni ingressi virtuali, che espongono alcuni stati interni dell'azionamento:

Nome	Ind.	Descrizione
PENDEVENTS	17	Ci sono eventi disponibili per la funzione <i>Collect</i>
GENERICFAIL	19	La diagnostica ha rilevato un errore interno generico
ALWAYS0	20	Questo bit è sempre a 0 (usato come prova di lettura)
ALWAYS1	21	Questo bit è sempre a 1
FLASHERR	22	La memoria flash ha fallito la scrittura della configurazione
WDTFBACKS	25	Il watch dog è intervenuto a causa di fattori interni
WDTCOMMTM	26	Il watch dog è intervenuto per un timeout di comunicazione
COREDROVF	29	L'elettronica di controllo uscite non funziona
DIPSW3	128	Valore del dip-switch 3
DIPSW4	129	Valore del dip-switch 4

Input Registers (parole di sola lettura)

Questi registri sono valori a 16 bit, che possono essere letti ma non impostati (come da specifiche Modbus). Ministep pone i Discrete Inputs nella stessa zona di questi registri, cosicché da questi registri è possibile leggere le stesse informazioni. La tabella seguente mostra i registri:

Nome	Ind.	Descrizione
INPUTS	0	I 16 ingressi X1..X16 descritti prima
ROTSELECTOR	9	Valore del selettore rotante degli indirizzi
VALIM	12	La tensione di alimentazione in decimi di volt

Registri Single Coil

Ministep usa i Single Coil per eseguire diversi comandi, descritti in questa tabella:

Nome	Ind.	Descrizione
RUNPLUS	0	Mettere a 1 per far ruotare il motore in modo libero (<i>free run</i>) in direzione +, o portare da 1 a 0 per fermare la rotazione.
RUNMINUS	1	Come sopra, per la direzione inversa (“-”).
STOP	2	Impostare a 1 per fermare il movimento libero.
GOHOME	3	Impostare 1 per mandare il motore in HOME (posizione 0).
GOSTEPDIR	4	Impostare 1 per attivare il modo Passo/Direzione.
GOSWITCH	5	Manda il motore sul fine corsa collegato all'ingresso X1.
RELSWITCH	6	Esegue un rilascio lento del fine corsa collegato a X1.
HIZ	7	Sconnette elettricamente il motore, azzerando perciò la coppia.
SWRMINUS	9	Imposta direzione “-” per i comandi GOSWITCH/RELSWITCH.
SWRZERO	11	Imposta opzione per i comandi GOSWITCH/RELSWITCH.
STPLOSS	12	Segnala una perdita di passo (azzerabile).
VALIMLOW	13	Segnala una perdita di alimentazione (azzerabile).
OVCURR	14	Segnala una sovracorrente (azzerabile).
THSHUTDOWN	15	Segnala che la temperatura troppo elevata ha disconnesso l'elettronica. Può essere azzerato (dopo il raffreddamento).

Registri Holding Registers

Come gli Input Registers, gli Holding Registers sono sovrapposti ai registri Single Coil. Quindi, i bit del paragrafo precedente possono essere letti e scritti con una singola operazione a 16 bit all'indirizzo 0. In aggiunta, ci sono altri registri usati per l'impostazione dei parametri di funzionamento. La tabella seguente mostra parametri e indirizzi:

Nome	Ind.	Descrizione
YWORD	0	Corrisponde ai primi 16 <i>Single Coils</i> .
XCOUNT1..3	11..13	Conteggio dei fronti dei 3 ingressi (azzerabile).
XLATUP1..3	19..21	Latenza dei fronti di salita dei tre ingressi.
XLATDN1..3	35..37	Latenza dei fronti di discesa.
RELPLUS (Y1PULSE)	81	(intero 16 bit senza segno) Quando scritto, genera un movimento incrementale in direzione "+", pari al valore indicato.
RELMINUS (Y2PULSE)	82	(intero 16 bit senza segno) Quando scritto, genera un movimento incrementale in direzione inversa ("-").
POSMARK	87-88	(LONGINT) Valore (coordinata) della posizione MARK.
POSMOT	89-90	(LONGINT) Quota corrente (posizione) del motore. Impostando un nuovo valore, si comanda il movimento del motore.
POSPRESET	91-92	(LONGINT) Preset dell'asse. Scrivendo un valore, il registro POSMOT viene modificato <i>senza</i> muovere l'asse.
MAXSPEED	93	Velocità massima (crociera) in passi/s per gli spostamenti del motore.
MINSPEED	94	Velocità minima in passi/s.
ACC	95	Accelerazione per le rampe (passi/s ²) (default 1600).
DEC	96	Decelerazione (passi/s ²) (default 1600).
CURRHOLD	97	Corrente di pilotaggio a motore fermo (mA, default 300).
CURRRUN	98	Corrente di pilotaggio in crociera (mA, default 1000).
CURRACC	99	Corrente in accelerazione (default 1000).
CURRDEC	100	Corrente in decelerazione (default 1000).
FSSPEED	101	Velocità di commutazione automatica a Passo Intero (default 1000).
STEPMODE	104	Configurazione di vari parametri (default 0x2082).
ENCODER	105-106	(LONGINT) Posizione corrente letta dall'encoder.

Nota: i parametri segnati con (**LONGINT**) sono interi di 32 bit con segno in formato little-endian: i 16 bit bassi sono memorizzati all'indirizzo più basso; gli altri 16 bit risiedono all'indirizzo Modbus successivo.

Procedura per muovere il motore

Per gestire correttamente l'asse (motore), i seguenti passi devono essere eseguiti:

Primo, impostare i parametri

Scegliere i valori desiderati per le correnti di pilotaggio, le velocità e le rampe. Impostare tali valori negli Holding Register corrispondenti del capitolo precedente. All'accensione Ministep carica un insieme di parametri (il default, o gli ultimi salvati). I parametri sono usati per ogni successivo movimento, e vanno cambiati solo a motore fermo. E' possibile eseguire movimenti molto diversi (come corrente, o rampe) uno dopo l'altro – basta cambiare parametri tra un movimento e l'altro. Questi parametri ritengono sempre l'ultimo valore impostato da RS-485. I parametri da impostare sono MAXSPEED, MINSPEED, ACC, DEC, CURRHOLD, CURRRUN, CURRACC e CURRDEC.

Secondo, assegnare la posizione (quota) all'asse

I motori stepper sono intrinsecamente *digitali* e *discreti*: l'azionamento principalmente ne impone la posizione (quota), e solo marginalmente la coppia. La velocità risulta controllata come diretta conseguenza del controllo di posizione. E' conveniente pensare al movimento di questi motori come un cambiamento della loro quota (coordinata): l'asse raggiungerà automaticamente la destinazione assegnata in configurazione ad anello aperto.

Comunque è consigliabile, dopo l'accensione del sistema, spostare il motore in una posizione fisica rilevabile, per esempio tramite un fine corsa; oppure spingere dolcemente l'asse contro un fine corsa meccanico. Una volta raggiunta tale posizione, si deve assegnare all'asse la coordinata prescelta scrivendone il valore nel registro POSPRESET. Da quel momento, l'asse può essere posizionato ovunque semplicemente indicando la nuova posizione da raggiungere.

Terzo, comandare i movimenti

Ci sono diversi modi per comandare un movimento, usando i parametri (velocità, rampe, correnti) attivi al momento:

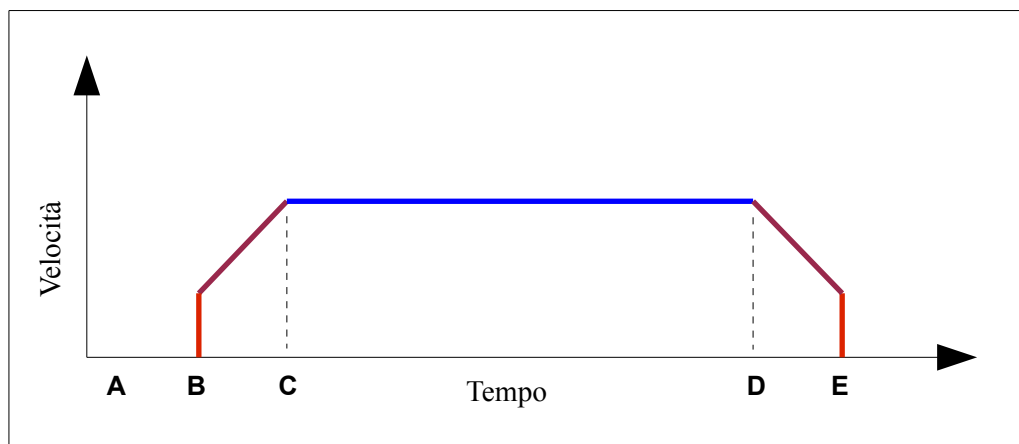
- Nuova quota: basta scrivere la nuova quota assoluta nel registro POSMOT. Il motore eseguirà una rampa di avviamento, userà la massima velocità per il tempo/spazio necessario e infine, dopo la rampa di rallentamento, si fermerà nella nuova posizione.
- Usare GOHOME: il comando è esattamente uguale a richiedere la posizione (quota) 0.
- Movimento incrementale: scrivere il numero di passi da eseguire nel registro RELPLUS; l'asse andrà nella posizione POSMOT+RELPLUS (quindi in positivo). In modo simile, scrivere il valore nel registro RELMINUS per eseguire un movimento in direzione inversa.
- Rotazione libera: il motore può girare indefinitamente, senza una destinazione assegnata. Durante tale operazione l'azionamento conteggia ugualmente la posizione dell'asse, ma per fermarlo occorre un comando esplicito. Scrivere 1 nel single coil RUNPLUS per far girare il motore in direzione positiva, o scrivere 1 in RUNMINUS per farlo girare in senso opposto. Scrivere 1 nel coil STOP per decelerare e fermare. E' possibile impostare RUNPLUS mentre RUNMINUS è attivo, o viceversa: l'azionamento esegue automaticamente decelerazione e accelerazione senza movimenti bruschi.
- Usare il comando GOSWITCH per raggiungere il fine corsa (**se presente**) connesso all'ingresso X1. Impostare prima SWRMINUS per indicare la direzione di moto (0=+, 1=-), ed SWRZERO (vedi dopo). Poi, scrivere 1 in GOSWITCH. L'asse inizierà un movimento, fino a ingaggiare il fine corsa (segnalato da X1 alto). A quel punto, esegue un arresto con rampa. Se SWRZERO era

1, il registro POSMOT viene azzerato; se era SWRZERO era 0, POSMOT rimane intatto e copiato nel registro POSMARK.

- Usare il comando RELSWITCH (*release switch*). Il motore usa la minima velocità per *lasciare* il fine corsa (X1 va basso). Di nuovo, SWRMINUS impone la direzione del movimento, ed SWRZERO indica se azzerare POSMOT o impostare POSMARK. Questa funzione è più precisa della precedente: si usa una velocità bassa e il motore viene fermato senza rampe.

Anatomia di un movimento

I motori stepper sono speciali perché, come tutti i motori, hanno bisogno di una rampa per cambiare velocità; ma essi dispongono di una certa velocità, relativamente bassa, che possono raggiungere *istantaneamente*. Si può utilizzare questa caratteristica impostando MINSPEED a un valore diverso da zero (non troppo alto, secondo diverse condizioni: per esempio, 200 passi/s o meno). Nel caso in cui MINSPEED è diverso da 0, il seguente profilo è usato per eseguire un posizionamento:



All'istante **A**, il motore è fermo. Quando si comanda un movimento, in **B**, comincia a girare a MINSPEED passi al secondo *immediatamente* e, inoltre, comincia ad aumentare la velocità fino a raggiungere MAXSPEED in **C**. La velocità massima è mantenuta per il tempo necessario; in **D** la velocità scende fino a MINSPEED, e poi il motore si arresta bruscamente. In altri termini, il motore non gira mai a una velocità inferiore a MINSPEED passi/secondo. In ogni fase è usata una diversa corrente (cioè coppia) di pilotaggio. Quando il motore è fermo, in **A** e dopo **E**, viene usata CURRHOLD. Questo valore dovrebbe essere abbastanza basso, per non surriscaldare il motore, però sufficiente a mantenere il sistema in posizione. Durante la fase di MAXSPEED, da **C** a **D**, viene usata CURRRUN. Questa corrente dovrebbe essere grande abbastanza da vincere l'attrito del sistema e la gravità (se il movimento non è orizzontale). Durante l'accelerazione (**B-C**), viene usata CURRACC; questa può essere uguale a CURRRUN, ma può essere (presumibilmente) più alta per considerare la massa inerziale del sistema. Durante la decelerazione (CURRDEC) avviene invece l'opposto: l'attrito aiuta, ma la massa ostacola il rallentamento. I valori corretti delle correnti possono essere calcolati in anticipo, ma quasi sempre sono necessarie prove sul campo.

Troppa corrente imposta a un motore stepper può causare uno stallo (*risonanza*, una caratteristica di tali motori), oppure vibrazioni e rumorosità. Per questa ragione, le correnti di pilotaggio dovrebbero essere calcolate con cura e verificate bene sul campo.

La funzione Modbus Collect

Nel caso in cui si debbano raccogliere gli stati degli I/O da molti dispositivi, il tempo d'interrogazione degli stessi può diventare eccessivamente lungo. Questo è ancor più grave se gran parte delle scansioni fornisce risultati identici poiché, alla fine, non c'è stata alcuna variazione. La funzione Collect permette d'interrogare una serie di dispositivi con indirizzo contiguo, e ottenere una risposta dal primo di essi che ha una variazione da comunicare. Collect va ripetuta fino a che non si ottengono più risposte. A seconda dei casi, il tempo d'interrogazione di molti dispositivi può ridursi anche al 5% del tempo normalmente richiesto. Ministep memorizza le variazioni in una coda di eventi, e un evento in coda viene tolto quando il master segnala di aver ricevuto tale evento.

La funzione Collect ha formato (in byte): DST, FCN, FIRST, LAST, SLVACK, SLVSEQ.

DST è lo slave di destinazione (di solito broadcast, quindi 0). FCN è la funzione, di solito 70.

FIRST e LAST definiscono l'intervallo di indirizzi ai quali la funzione è rivolta; per esempio FIRST=1 e LAST=16 consente ai soli slave di indirizzo tra 1 e 16 di rispondere se hanno eventi in coda. Per ogni slave sono disponibili 3 ms per rispondere, quindi un comando Collect per 10 slave ha un timeout nominale di 30 millisecondi (più 1 come margine di sicurezza).

Quando il master emette il comando Collect, la serie di slave indicata è autorizzata a rispondere, in ordine di indirizzo crescente, nel tempo stabilito di 3 ms ognuno. Se nessun slave ha variazioni da comunicare, il comando Collect non ottiene risposta dopo il timeout di 3 ms per ogni slave indicato nella finestra.

Se uno degli slave risponde, lo fa con un messaggio la cui forma è SLAVE, FCN (=70), SLVSEQ, TYPE (2=ingressi, 1=uscite), DATH, DATL, 0. Dopo la ricezione della risposta dello slave, il master deve mandare un comando Collect che indica gli opportuni SLVACK e SLVSEQ per eliminare l'evento dalla coda. Questo ACK può essere mandato in una fase di interrogazione successiva; per esempio, se una prima interrogazione è rivolta agli slave 1-16 e il 4 risponde, occorre una successiva interrogazione per gli slave rimanenti (5-16); questa ulteriore interrogazione dovrebbe riportare SLVACK ed SLVSEQ per lo slave che aveva risposto (il 4).

Protocollo (comandi) di testo semplice

In alternativa al protocollo Modbus, semplice ma binario, è possibile eseguire la maggior parte delle operazioni usando un protocollo di testo, adatto anche a file batch o script in diversi linguaggi.

La logica di fondo è la stessa: il master (PC o PLC) invia un comando, e attende una risposta per alcuni centesimi di secondo. Ministep risponde senza alcun ritardo, tranne che per il comando "WCONF=1" che può impiegare, in alcuni casi, qualche decimo di secondo.

Ogni comando da master verso Ministep è composta da una serie di caratteri ASCII terminata da un CR (Carriage return, carattere numero 13 corrispondente all'Invio della tastiera). Gli eventuali caratteri SPAZIO (numero 32) ed LF (Line feed, carattere numero 10) sono ignorati; Ministep, inoltre, non fa differenza tra lettere maiuscole e minuscole.

La lunghezza massima di un pacchetto è di 76 caratteri: ulteriori caratteri, fino al prossimo CR, vengono ignorati. Comunque, nessun pacchetto valido può essere così lungo.

Tutti i pacchetti iniziano con un carattere "?" (punto interrogativo, interrogazione) o ">" (simbolo di maggiore, impostazione). La forma del comando d'interrogazione è la seguente:

Esempio di comando di interrogazione / lettura				
?	X	1	<CR>	(<LF>)
Comando (print)	Argomento del comando (X1)	Terminatore		LF ignorato se inviato

L'argomento del comando, in questo esempio **X1**, è un identificatore conosciuto da Ministep; vedere più avanti l'elenco di tutti gli identificatori riconosciuti. Nel caso che il pacchetto sia formato correttamente e che l'identificatore sia valido, Ministep risponde con il valore richiesto:

Risposta a interrogazione / lettura				
X	1	=	0	<CR>
Identificatore (X1)	Elemento sintattico	Valore	Terminatore	

Dopo il singolo di uguale si trovano sempre 1, 3 o 5 cifre - dipende dal tipo di identificatore.

Nel caso dell'impostazione (di una uscita, o di un identificatore interno) la forma è la seguente:

Esempio di comando di impostazione						
>	Y	1	=	1	<CR>	(<LF>)
Comando (set)	Argomento del comando (Y1)	Elemento sintattico	Valore	Terminatore	LF ignorato se presente	

Il valore dopo il segno di uguale deve essere formato da interi positivi (senza segno); non è necessario immettere zeri non significativi. Però, il numero deve ricadere nell'intervallo ammissibile per l'identificatore specificato (per una uscita Y, solo 0 e 1; per l'insieme di uscite YWORD, un valore da 0 a 65535, anche se diversi bit non sono implementati).

Alla ricezione del comando, dopo il CR, Ministep invia la risposta:

Risposta positiva a comando d'impostazione		
O	K	<CR>
Risposta affermativa (nessun errore)		Terminatore

Se il pacchetto è mal formato (non inizia con "?" o ">"), non si riceve risposta. Se contiene altri errori:

Risposta negativa a pacchetto con errori					
E	r	r	o	r	<CR>
Risposta negativa (errore)					Terminatore

Indirizzamento slave

I comandi riportati prima non indirizzano esplicitamente un singolo slave – corrispondono al broadcast del Modbus e sono da utilizzare quando sulla linea è collegato un dispositivo solo. Quando vi sono più dispositivi collegati è necessario anteporre al comando la selezione dello slave, cioè l'indirizzo slave al quale il comando è rivolto. Questo si fa usando il simbolo AT (chiocciola) seguito dal numero di slave, seguito ancora dal comando come visto prima:

@25?DEVICE

significa “allo slave 25: print device”. Si noti che il numero dopo la chiocciola non deve essere zero.

Comandi di lettura (modo testo)

Ognuno dei comandi seguenti deve essere preceduto da un punto interrogativo e seguito da un CR (Invio) come spiegato in precedenza. Per brevità tali caratteri non sono riportati. Molti dei seguenti identificatori sono di sola lettura, ma altri sono anche scrivibili e verranno elencati nuovamente nel prossimo capitolo.

Device

Risponde con nome e versione firmware del prodotto; al momento è “Ministp3 1.2”.

Address

(es. ?ADDRESS): riporta l'indirizzo slave del modulo (es. ADDRESS=034).

Flags

Legge una maschera di bit dello stato interno del dispositivo. Vedere [Altri discrete inputs](#) per il significato dei singoli bit.

Valim

Risponde col valore (in decimi di volt) della tensione d'alimentazione, letta dal convertitore ADC interno.

Xword

Legge una maschera di 16 bit dello stato degli ingressi (virtuali e non). Per esempio, “XWORD=00003” significa che X1 e X2 sono on, e X3 è off. Riferirsi a [Ingressi digitali](#) per i dettagli.

X1 .. X16

(es. ?X1): ritorna lo stato del singolo input specificato: 1=on, 0=off. Riferirsi a [Ingressi digitali](#) per la spiegazione.

Xcount1 .. Xcount3

Risponde col contenuto del contatore indicato, che conta i fronti di salita dell'ingresso associato (da X1 a X3).

Yword

Legge un numero (maschera) di 16 bit, come Xword, che è lo stato delle uscite (tutte virtuali). Vedere [Uscite Single Coil](#) per i dettagli.

Y1 .. Y16

Mostra lo stato dell'uscita indicata: 1=on, 0=off. In Ministep, tutte le uscite sono virtuali, non sono associate ad alcun morsetto. Vedere [Uscite Single Coil](#) per maggiori informazioni.

WDTtime

Mostra il timeout, della comunicazione con il master, che fa intervenire il watch-dog. Se è 0, non c'è timeout. Se diverso da 0, Ministep si aspetta un messaggio dal master almeno ogni WDTTIME centesimi di secondo, altrimenti assume un errore del master e il watch-dog interviene.

X1latup .. X3latup

Legge la latenza per i fronti di salita, in millisecondi, per gli ingressi X1 .. X3 rispettivamente.

X1latdn .. X3latdn

Come sopra, per i fronti di discesa.

Pos, Mark

Leggono il contenuto dei registri POSMOT e POSMARK, relativi alla quota (posizione) dell'asse. Fare riferimento a [Registri Holding Registers](#) per i dettagli.

Encoder

L'encoder opzionale, se connesso agli ingressi X2 e X3, legge la posizione dell'asse in modo indipendente dall'azionamento. Leggendo Encoder, è possibile verificare la posizione meccanica *reale*, in contrasto con quella interna - calcolata in anello aperto, e quindi possibilmente inaccurata in determinate situazioni (per esempio a causa di una perdita di passo).

Maxspeed, Minspeed, Accel, Decel

Leggono i parametri del profilo di velocità dei movimenti. Vedere [Registri Holding Registers](#) per una spiegazione approfondita.

Currhold, Currrun, Curracc, Currdec

Leggono il valore, in mA, delle correnti di pilotaggio usate in differenti momenti di un movimento. Vedere [Anatomia di un movimento](#) per i dettagli.

FSspeed

Legge la velocità oltre la quale l'azionamento usa comunque il modo di pilotaggio Passo Intero. Sotto questa velocità viene usato, eventualmente, uno dei modi di microstepping.

MotMode

Numero a sedici bit che contiene diversi parametri di funzionamento dell'azionamento.

Comandi di scrittura parametri (modo testo)

Per eseguire il comando, i seguenti identificatori devono essere preceduti da simbolo di maggiore (“>”) e seguiti da un segno di uguale (“=”) con il relativo valore, e infine un carattere CR (*Carriage return*, Invio), come mostrato precedentemente. La sintassi qui non è più mostrata per brevità. Quando il comando è eseguito, Ministep risponde con “OK”.

Address, Serline, RdelayT, RdelayM, Wconf

Questi comandi sono usati per impostare le preferenze, e sono spiegati nel capitolo [“Configurazione di Ministep”](#).

Yword

Imposta tutti gli output virtuali usando il numero specificato come maschera di bit. Per esempio, “>YWORD=3” attiva gli output virtuali Y1 e Y2 e spegne tutti gli altri. Per usare correttamente questi output, fare riferimento a [Registri Single Coil](#).

Y1 .. Y16

Attiva o disattiva un'uscita virtuale singola: “>Y2=1” accende la seconda uscita (facendo girare il motore in direzione *meno*), mentre “>Y3=1” ferma la rotazione. Dopo il segno di uguale “=”, solo un singolo 0 o 1 è valido. Riferirsi a [Registri Single Coil](#).

WDTtime

Imposta il tempo, in centesimi di secondo, per l'intervento del watch-dog sulla comunicazione seriale con il master. Quando questo timeout è maggiore di 0, Ministep deve ricevere un comando valido almeno ogni WDTTIME centesimi, altrimenti il watch-dog interviene. I valori validi vanno da 0 (nessun timeout) a 65535 (poco meno di 11 minuti).

Wflags

Scriva i flag interni del modulo. Lo scopo principale è quello di resettare lo stato di watch-dog intervenuto inviando un “>WFLAGS=0”.

Xcount1 .. Xcount3

Questi sono i contatori associati agli ingressi X1..X3. Scrivendo questi registri si fa il *preset* del valore –

di solito si azzerano per cominciare un nuovo conteggio, come “>XCOUNT1=0”.

X1latup .. X3latup

Imposta la latenza (o ritardo), in millisecondi, dei fronti di salita degli ingressi X1..X3. I numeri validi vanno da 0 a 65535.

X1latdn .. X3latdn

Come sopra, per i fronti di discesa.

Y1pulse, Y2pulse

Scrivendo un valore maggiore di 0 in Y1pulse, un movimento relativo viene eseguito in direzione “+”. Per esempio, “>Y1pulse=200” sposterà l'asse in direzione + di 200 passi. Y2pulse fa la stessa cosa, ma in direzione inversa (negativa).

Mark

Imposta una quota nel registro POSMARK. Normalmente questa istruzione non si usa.

Pos

Scrivendo una quota in questo registro, si comanda il movimento che porta l'asse alla quota specificata.

Preset

Assegna una quota all'asse, senza muoverlo fisicamente. Questa operazione è chiamata “preset”.

Encoder

L'encoder opzionale, connesso agli ingressi X2 e X3, legge la posizione dell'asse in modo indipendente dall'azionamento. Eseguendo una scrittura su questo registro, si esegue un *preset* del valore.

Maxspeed, Minspeed, Accel, Decel

Questi parametri s'impostano per regolare il profilo di velocità del motore durante gli spostamenti. Riferirsi al capitolo [Registri Holding Registers](#) per una spiegazione approfondita.

Currhold, Currrun, Curracc, Currdec

Impostano i valori, in mA, delle correnti (di picco) usate nelle diverse fasi di un movimento. Vedere [Anatomia di un movimento](#) per i dettagli.

FSspeed

Imposta la velocità oltre la quale il controllo del motore è sempre fatto con Passo Intero. Sotto questa velocità il motore può essere comandato in modo microstepping, se configurato.

MotMode

Questo registro contiene diversi parametri di configurazione dell'azionamento.

Muovere il motore con comandi di testo semplice

Dopo aver impostato i parametri desiderati di funzionamento (vedere le sezioni relative), tutti i movimenti possono essere comandati impostando a 1 **una** delle uscite virtuali **Y1..Y16** oppure scrivendo nel registro POS; lo stato del sistema può essere esaminato leggendo le stesse uscite e/o gli ingressi virtuali X1 .. X16. Le tabelle seguenti riassumono le diverse operazioni:

Operazioni su Y1 .. Y16

Operazione	Nome	Descrizione
RUNPLUS	Y1	Impostare a 1 per imporre il modo “rotazione libera” in senso positivo. Quando letto, indica che il motore gira in senso +.
RUNMINUS	Y2	Come prima, per la direzione negativa (“-”).
STOP	Y3	Impostare a 1 per fermare il motore (qualsiasi comando). Quando letto, è a uno se il motore è fermo.
GOHOME	Y4	Scrivere 1 per mandare il motore alla posizione HOME (coordinata 0).
GOSTEPDIR	Y5	Abilita il modo Passi/Direzione (da X2/X3).
GOSWITCH	Y6	Porta l'asse sul finecorsa FC/X1.
RELSWITCH	Y7	Porta l'asse fuori dal finecorsa FC/X1.
HIZ	Y8	Disabilita il pilotaggio del motore: scollegato, non genera coppia.
SWRMINUS	Y10	Seleziona la direzione “-” per i comandi GOSWITCH/RELSWITCH.
SWRZERO	Y12	Sceglie l'operazione per i comandi GOSWITCH/RELSWITCH.
STPLOSS	Y13	Segnala una perdita di passo; impostare a zero per riarmare il segnale.
VALIMLOW	Y14	Segnala una caduta dell'alimentazione; si può resettare.
OVCURR	Y15	Segnala una extra-corrente; si può resettare.
THSHUTDOWN	Y16	Segnala lo spegnimento del pilotaggio a causa di un riscaldamento eccessivo. Si può resettare, dopo il tempo necessario al raffreddamento.

Letture degli stati da X1 .. X16

Parametro	Nome	Descrizione
X1	X1	Riflette lo stato dell'ingresso I1 (X1)
X2	X2	Riflette lo stato dell'ingresso I2 (X2)
X3	X3	Riflette lo stato dell'ingresso I3 (X3)
RUNNING	X4	E' a On (1) quando l'asse è in movimento
STOPPED	X5	On (1) quando il motore è fermo
RUNPLUS	X6	On quando il motore gira in senso positivo +
RUNMINUS	X7	On quando il motore gira in senso inverso ("-")
HIZ	X8	A 1 (on) quando il motore è elettricamente scollegato
BUSY	X9	A 1 quando Ministep sta eseguendo qualche comando. Questo non comprende il modo "rotazione libera".
ATHOME	X10	A 1 (on) quando l'asse è in posizione/quota 0
ATMARK	X11	On (1) quando il motore è alla quota del registro POSMARK
STPLOSS	X13	Segnala una perdita di passo
VALIMLOW	X14	Segnala una caduta della tensione di alimentazione
OVCURR	X15	Segnala una extra-corrente
THWARN	X16	Segnala un inizio di surriscaldamento. Questo bit è solo un avvertimento: se la temperatura continua a salire l'azionamento si auto protegge scollegando il motore.



XON ELECTRONICS SRL
WWW.XONELECTRONICS.IT
INFO@XONELECTRONICS.IT

Segnalare errori o imprecisioni a web@xonelectronics.it

Per acquistare Ministep online: www.xonelectronics.eu